To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

User's Manual

32

SH-2E SH7055S F-ZTAT™

Hardware Manual

Renesas 32-bit RISC Microcomputer
SuperH™ RISC engine Family/
SH7000 Series

Renesas Electronics
www.renesas.com

Rev.2.00 2003.7

Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.

2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions may occur due to the false recognition of the pin state as an input signal. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

The SH7055SF is a single-chip RISC (reduced instruction set computer) microcomputer that has an original 32-bit RISC type CPU as its core, and also includes peripheral functions necessary for system configuration.

The SH7055SF is equipped with on-chip peripheral functions necessary for system configuration, including a floating-point unit (FPU), large-capacity ROM and RAM, a direct memory access controller (DMAC), timers, a serial communication interface (SCI), Controller area network (HCAN), A/D converter, interrupt controller (INTC), and I/O ports, therefore, it can be used as a microprocessor built in a high-level control system.

The SH7055SF is an F-ZTAT™* (Flexible Zero Turn-Around Time) version with flash memory as its on-chip ROM, and it can rapidly and flexibly deal with each situation on an application system with fluid specifications from an early stage of mass production to full-scale production.

Note: F-ZTAT™ is a trademark of Renesas Technology Corp.

Target users: This manual was written for users who will be using the SH7055S F-ZTAT in the design of application systems. Users of this manual are expected to understand the fundamentals of electrical curcuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the SH7055S F-ZTAT to the above users.
Refer to the SH-2E Programming Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.

- In order to understand the details of the CPU's functions
  Read the SH-2E Programming Manual.

Rule: Bit order: The MSB (most significant bit) is on the left and the LSB (least significant bit) is on the right.

Releated Manuals: The latest versions of all related manuals are available from our web site.
Please ensure you have the latest versions of all documents you require.
http://www.renesas.com/

SH7055S F-ZTAT Hardware Manual	This manual
SH-2E Programming Manual	

Users manuals for development tools:

Manual Title	ADE No.
SH Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	ADE-702-246
SH Series Simulator/Debugger (for Windows) User's Manual	ADE-702-186
SH Series Simulator/Debugger (for UNIX) User's Manual	ADE-702-203
High-Performance Embedded Workshop User's Manual	ADE-702-201

Application note:

Manual Title	ADE No.
C/C++ Compiler	

Item	Page	Revisions (See Manual for Details)
2.4.1 Instruction Set by Classification Table 2.16 Branch Instructions	53	Table amended BF/S label 10001111dddddddd Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop 2/1* —
3.6 Usage Notes 3. Restrictions of the FADD and FSUB instructions	69, 70	Newly added
5.3.1 Connecting a Crystal Oscillator Figure 5.3 Connecting of Crystal Oscillator(Example) Table 5.3 Damping Resistance Values(Recommended Values)	75, 76	Recommended value amended CL1=CL2=18-22 pF (recommended value) Table amended Frequency (MHz) Parameter 5 10 Rd (Ω) 500 0

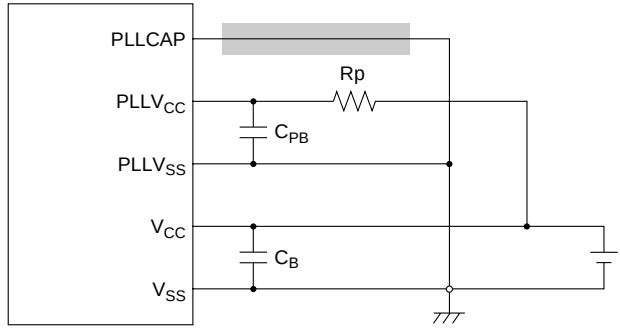
PLL Oscillation Power Supply

Figure 5.7 Points for Caution in PLL Power Supply Connection

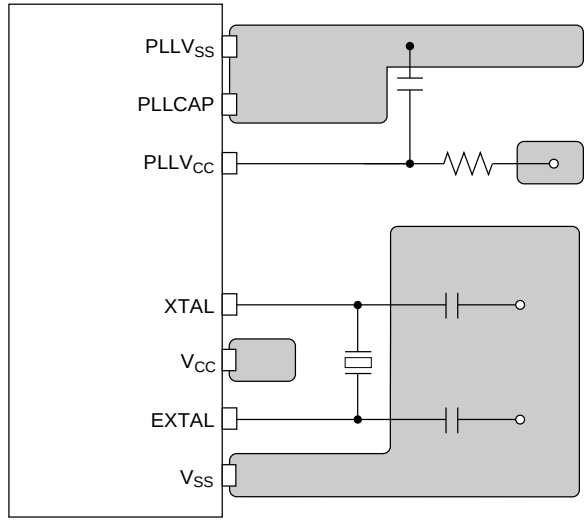
Figure 5.8 Actual Example of Board Design

PLL Oscillation Power Supply: Separate $PLLV_{CC}$ and $PLLV_{SS}$ from the other V_{CC} and V_{SS} lines at the board power supply source, …

Figures amended



Recommended values
C_{PB}, C_B: 0.1µF
Rp: 200Ω



6.7 Stack Status after Exception Processing Ends

Table 6.11 Stack Status After Exception Processing Ends

Table amended

General illegal instruction	SP → Address of general illegal instruction	32 bits
	SR	32 bits

Table 11.1 ATU-II functions — 196

Channel 1	Channel 2	Channels 3–5
(ϕ–ϕ/32) × (1/2n)	(ϕ–ϕ/32) × (1/2n)	(ϕ–ϕ/32) × (1/2n)
(n = 0–5)	(n = 0–5)	(n = 0–5)
TCLKA, TCLKB, AGCK, AGCKM	TCLKA, TCLKB, AGCK, AGCKM	TCLKA, TCLKB, AGCK, AGCKM

GR10G
OCR10AH, OCR10AL, OCR10B, NCR10, TCCLR10

11.1.3 Register Configuration — 201

Table 11.3 ATU-II Registers — Table amended

TSTR1	R/W	H'00
TSTR2	R/W	H'00
TSTR3	R/W	H'00
PSCR1	R/W	H'00
PSCR2	R/W	H'00
PSCR3	R/W	H'00
PSCR4	R/W	H'00

11.2.2 Prescaler Registers(PSCR) — 227 — Bit Table amended

Bit:	7	6	5	4	3	2	1	0
	—	—	—	PSCxE	PSCxD	PSCxC	PSCxB	PSCxA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

x = 1 to 4

11.2.4 Timer I/O Control Registers(TIOR) — 246, 247 — Table amended

Timer I/O Control Registers 3A, 3B, 4A, 4B, 5A, 5B(TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B)

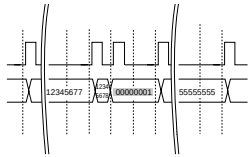
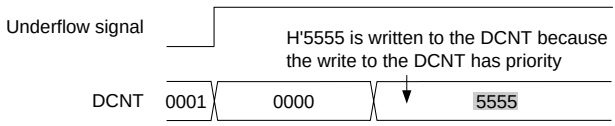
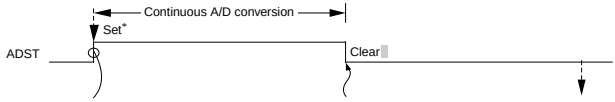
Bits 6 to 4

1	0	0	GR is an input capture register (input capture by channel 3 and 9 compare-match enabled)	Input capture disabled (In channel 3 only, GR cannot be written to)
		1		Input capture in GR on rising edge at TIOxx pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIOxx pin (GR cannot be written to)

Bits 2 to 0

1	0	0	GR is an input capture register (input capture by channel 3 and 9 compare-match enabled)	Input capture disabled (In channel 3 only, GR cannot be written to)
		1		Input capture in GR on rising edge at TIOxx pin (GR connot be written to)
	1	0		Input capture in GR on falling edge at TIOxx pin (GR connot be written to)

Cycle Registers (CYLR6A to CYLR6D, CYLR7A to CYLR7D)		At the same time, the buffer register (BFR) value is transferred to the duty register (DTR). Output pin (TO6A to TO6D, TO7A to TO7D) of corresponding channnel will be 0 when H'0000 of BFR is 0 output and otherwise will
11.2.26 Channel 10 Registers Counters(TCNT) Free-Running Counter 10AH,AL (TCNT10AH, TCNT10AL)	338	Description amended …an input clock and is cleared to the initial value by input capture input (TI10)(AGCK).
11.2.26 Channel 10 Registers Registers (TCNT) Input Capture Register 10AH, AL (ICR10AH, ICR10AL)	342	Description amended At the same time, ICF10A in timer status register 10 (TSR10) is set to 1.
11.3.1 Overview Channels 6 and 7	355	Description amended Do not set a value in DTR that will result in the condition DTR > CYLR. When H'0000 is set to DTR, do not have DTR directly read H'0000. Set BFR to H'0000 and set H'0000 by forwarding from BFR to DTR. If H'0000 is directly set to DTR, duty may not be 0%.
11.3.8 Twin-Capture Function	365	Description amended Line 4 When TCNT0, TCNT1A, and TCNT2A in channel 0, channel 1, and channel 2 are started by a setting in the timer status register (TSR), and an edge detection is carried out by the ICR0A input as a trigger signal, the TCNT1A value is transferred to OSBR1, and the TCNT2A value to OSBR2. Edge detection is as described in section 11.3.4, Input Capture Function.

Figure 11.21 PWM Timer Operation	367	If the DTR value is H'0000, the output does not change (0% duty). However, when H'0000 is set to DTR, do not directly write H'0000 to DTR. Set H'0000 to BFR and forward it from BFR to DTR. If H'0000 is directly set to DTR, duty may not be 0%. A duty of 100% is specified by setting DTR = CYLR. Do not set a value in DTR that will result in the condition DTR > CYLR. Figure amended • TO6A amended PWM output does not change for one cycle after activation* Note added * PWM output is not guaranteed because retained value is output for one cycle after activation.
11.3.9 PWM Timer Function Figure 11.22 Complementary PWM Mode Operation	368	Figure replaced
11.3.12 Channel 10 Functions Inter-Edge Measurement Function and Edge Input Cessation Detection Function: Figure 11.28 TCNT10A Capture Operation and Compare-Match Operation	372	Figure amended 
11.7 Usage Notes Contention between DCNT Write and Counter Clearing by Underflow: Figure 11.72 Contention between DCNT Write and Underflow	414	Note added Figure amended 
11.7 Usage Notes ATU Pin Setting:	418	Description amended When a port is set to the ATU pin function, the following points must be noted because input capture or count operation may occur.
17.4.2 Scan Mode Figure 17.4 Example of A/D Converter Operation(Scan Mode(Single-Cycle Scan), Channels AN0 to AN11 Selected)	608	Figure amended 

• Programming/erasing time • Number of programming		• Programming/erasing time The flash memory programming time is t_p ms (typ) in 128-byte simultaneous programming and t_p/128ms per byte. The erasing time is t_E s (typ) per block. • Number of programming The number of flash memory programming can be up to N_{WEC} times.
22.4.3 Programming/Erasing Interface Parameters (2) Programming/Erasing Initialization	758	Description added Line 13 The general registers R8 to R15 are stored. The general registers R0 to R7 can be used without being stored.
22.5.3 User Boot Mode (1) User Boot Mode Initiation	782	Description added Line 3 When the reset start is executed in user boot mode, the check routine for flash-memory related registers runs. While the check routine is running, the RAM area about 1.2 kbytes from H'FFFF6800 is used by the routine and 4 bytes from H'FFFFDFFC is used as a stack area. NMI and all other interrupts cannot be accepted. Neither can the AUD be used in this period. This period is approximately 100 µs while operating at an internal frequency of 40 MHz.
22.7 Flash Memory Emulation in RAM	791	Note: Description added Note: Setting the RAMS bit to 1 puts all the blocks in flash memory in the programming/erasing-protected state regardless of the values of the RAM2 to RAM0 bits (emulation protection). Clear the RAMS bit to 0 before actual programming or erasure. RAM emulation can be performed when the user boot MAT is selected. However, programming/erasing user boot MAT can be performed only in boot mode or program mode.

2. User branch processing intervals		**Minimum Interval** Approximately 19 µs Approximately 19 µs
Table 22.11 Initiation Intervals of User Branch Processing		
Table 22.12 Required Period for Initiating User Branch Processing		**Processing** / **Max.** / **Min.** Programming / Approximately 113 µs / Approximately 113 µs Erasing / Approximately 85 µs / Approximately 45 µs
4. State in which AUD operation is disabled and interrupts are ignored		Description added — Checking the flash-memory related registers immediately after user boot mode is initiated (Approximately 100 µs when operation with internal frequency of 40 MHz is carried out after the reset signal is released.)
22.10.1 Serial Communication Interface Specification for Boot Mode (3) Memory read	827	Description added Command: Read start address (four bytes): Size of data to be read Error response: H'2A: Address error The start address for reading is not in the MAT. H'2B: Size error The read size exceeds the MAT, the last address for reading calculated from the start address for reading and the read size is not in the MAT, or read size is 0.
24.3.1 Transition to Hardware Standby Mode	854	Description added The chip enters hardware standby mode when the HSTBY and RES pins go low. Set the pins following to mode setup pin shown in section 4, Operating modes. Operation with other pin set up are not guaranteed. Hardware standby mode reduces power consumption drastically by halting all SH7055SF functions

Table 26.1 Absolute Maximum Ratings — 862

Power supply voltage*	$PV_{cc}1$ and $PV_{cc}2$ pins	PV_{cc}	–0.3 to +6.5
Operating temperature ** (except writing or erasing on-chip flash memory)		Topr	–40 to +125
Operating temperature (writing or erasing on-chip flash memory)		TWEopr	–40 to +85
Storage temperature		Tstg	–55 to +125

Note added

Temperature Range for Operation	Accumulated Time
85 to 105 °C	3000 hours

26.2 DC Characteristics — 869

Table amended

Table 26.2 Correspondence between Power Supply Names and Pins

169	PC4	$\overline{IRQ0}$

Table 26.4 DC Characteristics

[Operating precautions]

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory,
T_a = –40°C to 85°C.

Input high-level voltage (except Schmitt trigger input voltage)	RES, NMI, FWE, MD2–0, HSTBY	V_{IH}	V_{CC} – 0.5	—	5.8	V	2.7 V ≤ V_{CC} ≤ 3.6V
Input low-level voltage (except Schmitt trigger input voltage)	RES, NMI, FWE, MD2–0, HSTBY, TRST, AUDRST, AUDMD	V_{IL}	–0.3	—	0.5	V	2.7 V ≤ V_{cc} ≤ 3.6V
	PG0, PL11		–0.3	—	$PV_{cc}2$ × 0.3	V	
	Other input pins		–0.3	—	0.8	V	
Input leak current	RES, NMI, FWE, MD2–0, HSTBY,	| Iin |	—	—	3.0*1 6.0*2	µA	Vin = 0.5 V to 5.8 V
	EXTAL (Standby)		—	—	3.0*1 6.0*2	µA	Vin = 0.5 V to V_{cc} – 0.5 V
	TMS, TRST, TDI, TCK (Standby)		—	—	3.0*1 6.0*2	µA	Vin = 0.5 V to V_{cc} – 0.5 V
	AUDMD, AUDCK, AUDSYNC, AUDATA3–0 (Standby)		—	—	3.0*1 6.0*2	µA	Vin = 0.5 V to $PV_{cc}2$ – 0.5 V
	AUDRST (Standby)		—	—	3.0*1 6.0*2	µA	Vin = 0.5 to AV $PV_{cc}2$ – 0.5 V
	A/D port		—	—	0.2*1 0.4*2	µA	Vin = 0 to AV_{cc}
Input leak current	D15–D0, WAIT, BREQ	| Iin |	—	—	3.0*1 6.0*2	µA	Vin = 0.5 V to $PV_{cc}1$ – 0.5 V $PV_{cc}1$ = 3.3 V ± 0.3 V
	PE15–PE0, PF15–PF0, PH15–PH0 (When in MCU expansion mode)		—	—	3.0*1 6.0*2	µA	Vin = 0.5 V to $PV_{cc}1$ – 0.5 V $PV_{cc}1$ = 3.3 V ± 0.3 V
	Other input pins		—	—	3.0*1 6.0*2	µA	Vin = 0.5 V to $PV_{cc}2$ – 0.5 V
Input pull-up MOS current	TMS, TRST, TDI, TCK (pull-up characteristic)	–Ipu	—	—	350	µA	Vin = 0 V
	AUDMD, AUDCK, AUDSYNC, AUDATA3-0 (pull-up characteristic)		—	—	800	µA	Vin = 0 V
Input pull-down MOS current	AUDRST (pull-down characteristic)	Ipd	—	—	500	µA	Vin = $PV_{cc}2$
Three-state leak current (while OFF)	A21–A0, D15–D0, CS3–CS0, WRH, WRL, RD, BACK (When in MCU expansion mode)	| Its |	—	—	3.0*1 6.0*2	µA	Vin = 0.5 to $PV_{cc}1$ – 0.5 V $PV_{cc}1$ = 3.3 V ± 0.3 V

Table 26.4 DC Characteristics [Operating precautions]	878					
	Standby		—	50	200	µA	$T_a \leq 50°C$
			—	—	500	µA	50°C < Ta≤105°C
			—	—	1000	µA	Ta > 105°C
	Write operation		—	60	90	mA	V_{cc} = 3.3 V f = 40 MHz
Analog supply current	During A/D conversion	AI_{CC}	—	1.2	5	mA	
	Awaiting A/D conversion		—	1	30	µA	

Reference power supply current	During A/D conversions	Alref	—	13	5	mA	AV_{ref} = 5 V
	Awaiting A/D conversion		—	11	10	µA	
RAM standby voltage		V_{RAM}	2.7	—	—	V	V_{CC}

Notes: *1 Ta ≤ 105°C
*2 Ta >105°C

Description added

[Operating precautions]

2. The current consumption is measured when V_{IH}min = V_{CC} – 0.3 V/PV_{CC} – 0.3 V, V_{IL} = 0.3 V, with all output pins unloaded.

26.2 DC Characteristics

Table 26.5 Permitted Output Current Values

879

Table amended

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V,
AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Output low-level permissible current (per pin)	I_{OL}	—	—	6	mA
Output low-level permissible current (total)	Σ I_{OL}	—	—	80	mA
Output high-level permissible current (per pin)	I_{OH}	—	—	2	mA
Output high-level permissible current (total)	Σ I_{OL}	—	—	25	mA

power supply on/off

Table 26.6 Timing for switching the power supply on/off

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V,
AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, Ta = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, Ta = –40°C to 85°C.

26.3.2 Clock Timing 881 Conditions amended

Table 26.7 Clock Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V,
AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, Ta = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, Ta = –40°C to 85°C.

26.3.3 Control Signal Timing 883 Table amended

Table 26.8 Control Signal Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V,
AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	Figure 26.5
$\overline{RES}$ setup time	t_{RESS}	40	—	ns	
MD2–MD0 setup time	t_{MDS}	20	—	t_{cyc}	

26.3.4 Bus Timing 886 Conditions amended

Table 26.9 Bus Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V,
AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Controller Timing Table 26.10 Advanced Timer Unit Timing and Advanced Pulse Controller Timing		Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C. When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$. When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.
26.3.6 I/O Port Timing Table 26.11 I/O Port Timing	892	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C. When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$. When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.
26.3.7 Watchdog Timer Timing Table 26.12 Watchdog Timer Timing	893	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C. When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$. When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.
26.3.8 Serial Communication Interface Timing Table 26.13 Serial Communication Interface Timing	894	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C. When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$. When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Table 26.14 HCAN Timing		Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C. When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$. When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.
26.3.10 A/D Converter Timing	897	Conditions amended
Table 26.15 A/D Converter Timing		Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C. When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$. When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.
26.3.11 H-UDI Timing	899	Conditions amended
Table 26.16 H-UDI Timing		Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C. When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$. When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.
26.3.12 AUD Timing	901	Conditions amended
Table 26.17 AUD Timing		Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C. When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$. When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Table 26.18 UBC Trigger Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V,
AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

26.4 A/D Converter Characteristics — 905 — Conditions and table amended

Table 26.19 A/D Converter Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V,
AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

	CSK = 0: fop = 10–20 MHz			CSK = 1: fop = 10 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	bit
A/D conversion time	—	—	13.3	—	—	13.4	µs
Analog input capacitance	—	—	20	—	—	20	pF
Permitted analog signal source impedance	—	—	3	—	—	3	kΩ
Non-linear error	—	—	±1.5*1	—	—	±1.5*1	LSB
			±2.5*2			±2.5*2	
Offset error	—	—	±1.5*1	—	—	±1.5*1	LSB
			±2.5*2			±2.5*2	
Full-scale error	—	—	±1.5*1	—	—	±1.5*1	LSB
			±2.5*2			±2.5*2	
Quantization error	—	—	±0.5	—	—	±0.5	LSB
Absolute error	—	—	±2.0*1	—	—	±2.0*1	LSB
			±2.5*2			±2.5*2	

Note: *1 Ta ≤105°C
*2 Ta >105°C

Characteristics

Table 26.20 Flash Memory Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V,
AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 105°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Typ	Max	Unit
Programming time*1*2	t_P	—	20	200	ms/128 bytes
Erase time*1*3	t_E	—	1	10	s/block
Reprogramming count	N_{WEC}	—	—	100	Times

Note: *1 Use the on-chip programming/erasing routine for programming/erasure.
*2 When all 0 are programmed.
*3 64 kbytes of block

26.6.1 Notes on Connecting External Capacitor for Current Stabilization

Figure 26.29 Connection of V_{CL} Capacitor

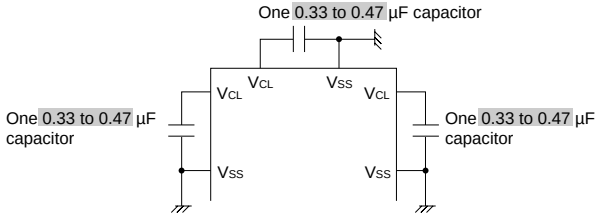
907

26.6.1 Title added

Description added

…power supply (V_{CL} pin) and the V_{ss} pin, an capacitor (0.33 to 0.47 µF) for stabilizing the internal voltage….

Figure amended



Do not apply any power supply voltage to the V_{CL} pin. Use multilayer ceramics capacitors (one 0.33 to 0.47 µF capacitor for each V_{CL} pin), which should be located near the pin.

26.6.2 Notes on Mode Pin Input

907 to 908

Newly added

A.2 Register States in Reset and Power-Down States

Table A.2 Register States in Reset and Power-Down States

953 to 954

Table amended

Serial communication interface (SCI)	SMR0 to SMR4	Initialized	Initialized	Held	Held
	BRR0 to BRR4				
	SCR0 to SCR4				
	TDR0 to TDR4			Intialized	
	SSR0 to SSR4				
	RDR0 to RDR4				
	SDCR0 to SDCR4			Held	
I/O ports	PADR, PBDR, PCDR PDDR, PEDR, PFDR PGDE, PHDR, PJDR PKDR, PLDR	Initialized	Initialized	Held	Held
	PAPR, PBPR, PDPR, PJPR, PLPR	Pin value	Held	Held	Pin value

1.1 Features

The SH7055SF is a single-chip RISC microcontroller that integrates a RISC CPU core using an original Renesas architecture with peripheral functions required for system configuration.

The CPU has a RISC-type instruction set. Basic instructions can be executed in one state (one system clock cycle), which greatly improves instruction execution speed. In addition, the 32-bit internal architecture enhances data processing power. With this CPU, it has become possible to assemble low-cost, high-performance/high-functionality systems even for applications such as real-time control, which could not previously be handled by microcontrollers because of their high-speed processing requirements.

In addition, the SH7055SF includes on-chip peripheral functions necessary for system configuration, such as a floating-point unit (FPU) , ROM , RAM, a direct memory access controller (DMAC), timers, a serial communication interface (SCI), controller area network (HCAN), A/D converter, interrupt controller (INTC), and I/O ports.

ROM and SRAM can be directly connected by means of an external memory access support function, greatly reducing system cost.

On-chip ROM is available as flash memory in the F-ZTAT™* (Flexible Zero Turn Around Time) version. The flash memory can be programmed with a programmer that supports SH7055SF programming, and can also be programmed and erased by software. Since the programming/erasing control program is included as firmware, programming and erasing can be performed by calling this program with a user program. This enables the chip to be programmed at the user site while mounted on a board.

The features of the SH7055SF are summarized in table 1.1.

Note: * F-ZTAT is a trademark of Renesas Technology Corp.

CPU	• Maximum operating frequency: 40 MHz • Original Hitachi SH-2E CPU • 32-bit internal architecture • General register machine — Sixteen 32-bit general registers — Three 32-bit control registers — Four 32-bit system registers • Instruction execution time: Basic instructions execute in one state (25 ns/instruction at 40 MHz operation) • Address space: Architecture supports 4 Gbytes • Five-stage pipeline
Operating states	• Operating modes — Single-chip mode — 8/16-bit bus expanded mode • Mode with on-chip ROM • Mode with no on-chip ROM • Processing states — Reset state — Program execution state — Exception handling state — Bus-released state — Power-down state • Power-down state — Sleep mode — Software standby mode — Hardware standby mode — Module standby
Multiplier	• $32 \times 32 \rightarrow 64$ multiply operations executed in two to four cycles $32 \times 32 + 64 \rightarrow 64$ multiply-and-accumulate operations executed in two to four cycles

Floating-point unit	• SuperH architecture coprocessor • Supports single-precision floating-point operations • Supports a subset of the data types specified by the IEEE standard • Supports invalid operation and division-by-zero exception detection (subset of IEEE standard) • Supports Round to Zero as the rounding mode (subset of IEEE standard) • Sixteen 32-bit floating-point data registers • Supports the FMAC instruction (multiply-and-accumulate instruction) • Supports the FDIV instruction (divide instruction) • Supports the FLDI0/FLDI1 instructions (constant 0/1 load instructions) • Instruction delay time: Two cycles for each of FMAC, FADD, FSUB, and FMUL instructions • Execution pitch: One cycle for each of FMAC, FADD, FSUB, and FMUL instructions
Clock pulse generator (CPG/PLL)	• On-chip clock pulse generator (maximum operating frequency: 40 MHz) • Independent generation of CPU system clock and peripheral clock for peripheral modules • On-chip clock-multiplication PLL circuit (×4) Internal clock frequency range: 5 to 10 MHz
Interrupt controller (INTC)	• Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$) • 115 internal interrupt sources (ATU-II × 75, SCI × 20, DMAC × 4, A/D × 3, WDT × 1, UBC × 1, CMT × 2, HCAN × 8, H-UDI × 1) • 16 programmable priority levels
User break controller (UBC)	• Requests an interrupt when the CPU or DMAC generates a bus cycle with specified conditions (interrupt can also be masked) • Trigger pulse output (UBCTRG) on break condition — Selection of trigger pulse width (φ ×1, ×4, ×8, ×16) • Simplifies configuration of an on-chip debugger

Bus state controller (BSC)	• Supports external memory access (SRAM and ROM directly connectable) — 8/16-bit bus space • 3.3 V bus interface • 16 MB address space divided into four areas, with the following parameters settable for each area: — Bus size (8 or 16 bits) — Number of wait cycles — Chip select signals ($\overline{CS0}$ to $\overline{CS3}$) output for each area • Wait cycles can be inserted using an external $\overline{WAIT}$ signal • External access in minimum of two cycles • Provision for idle cycle insertion to prevent bus collisions
Direct memory access controller (DMAC) (4 channels)	• DMA transfer possible for the following devices: — External memory, on-chip memory, on-chip peripheral modules (excluding DMAC, UBC, BSC) • DMA transfer requests by on-chip modules — SCI, A/D converter, ATU-II, HCAN • Cycle steal or burst mode transfer • Dual address mode — Direct transfer mode — Indirect transfer mode (channel 3 only) • Address reload function (channel 2 only) • Transfer data width: Byte/word/longword
Advanced timer unit-II (ATU-II)	• Maximum 65 inputs or outputs can be processed — Four 32-bit input capture inputs — Thirty 16-bit input capture inputs/output compare outputs — Sixteen 16-bit one-shot pulse outputs — Eight 16-bit PWM outputs — Six 8-bit event counters — One gap detection function • I/O pin output inversion function
Advanced pulse controller (APC)	• Maximum eight pulse outputs on reception of ATU-II (channel 11) compare-match signal

Watchdog timer (WDT) (1 channel)	• Can be switched between watchdog timer and interval timer function • Internal reset, external signal, or interrupt generated by counter overflow • Two kinds of internal reset — Power-on reset — Manual reset
Compare-match timer (CMT) (2 channels)	• Selection of 4 counter input clocks • A compare-match interrupt can be requested independently for each channel
Serial communication interface (SCI) (5 channels)	• Selection of asynchronous or synchronous mode • Simultaneous transmission/reception (full-duplex) capability • Serial data communication possible between multiple processors (asynchronous mode) • Clock inversion function • LSB-/MSB-first selection function for transmission
Controller area network (HCAN) (2 channels)	• CAN version: Bosch 2.0B active compatible • Buffer size (per channel): Transmit/receive × 15, receive-only × 1 • Receive message filtering capability
A/D converter	• Thirty-two channels • Three sample-and-hold circuits — Independent operation of 12 channels × 2 and 8 channels × 1 • Selection of two conversion modes — Single conversion mode — Scan mode • Continuous scan mode • Single-cycle scan mode • Can be activated by external trigger or ATU-II compare-match • 10-bit resolution • Accuracy: ±2 LSB
High-Performance user debug interface (H-UDI)	• Five dedicated pins • Bypass mode (test mode compliant with IEEE1149.1) • H-UDI interrupt

Advanced user debugger (AUD)	• Eight dedicated pins • RAM monitor mode — Data input/output frequency: ϕ/4 or less — Possible to read/write to a module connected to the internal/external bus • Branch address output mode
I/O ports (including timer I/O pins, address and data buses)	• Dual-function input/output pins: 149 • Schmitt input pins: NMI, $\overline{\text{IRQn}}$, $\overline{\text{RES}}$, $\overline{\text{HSTBY}}$, FWE, TCLK, IC, IC/OC, SCK, $\overline{\text{ADTRG}}$ • Input port protection
ROM	• 512-kbyte flash memory • 512 kbytes divided into 16 blocks — Small blocks: 4 kB × 8 — Medium block: 32 kB × 1 — Large blocks: 64 kB × 7 • RAM emulation function (using 4 KB small block) • Programming/erasing control program included as firmware • Flash memory programming methods — Boot mode — User boot mode — User program mode — Programmer mode
RAM	• 32 kB SRAM

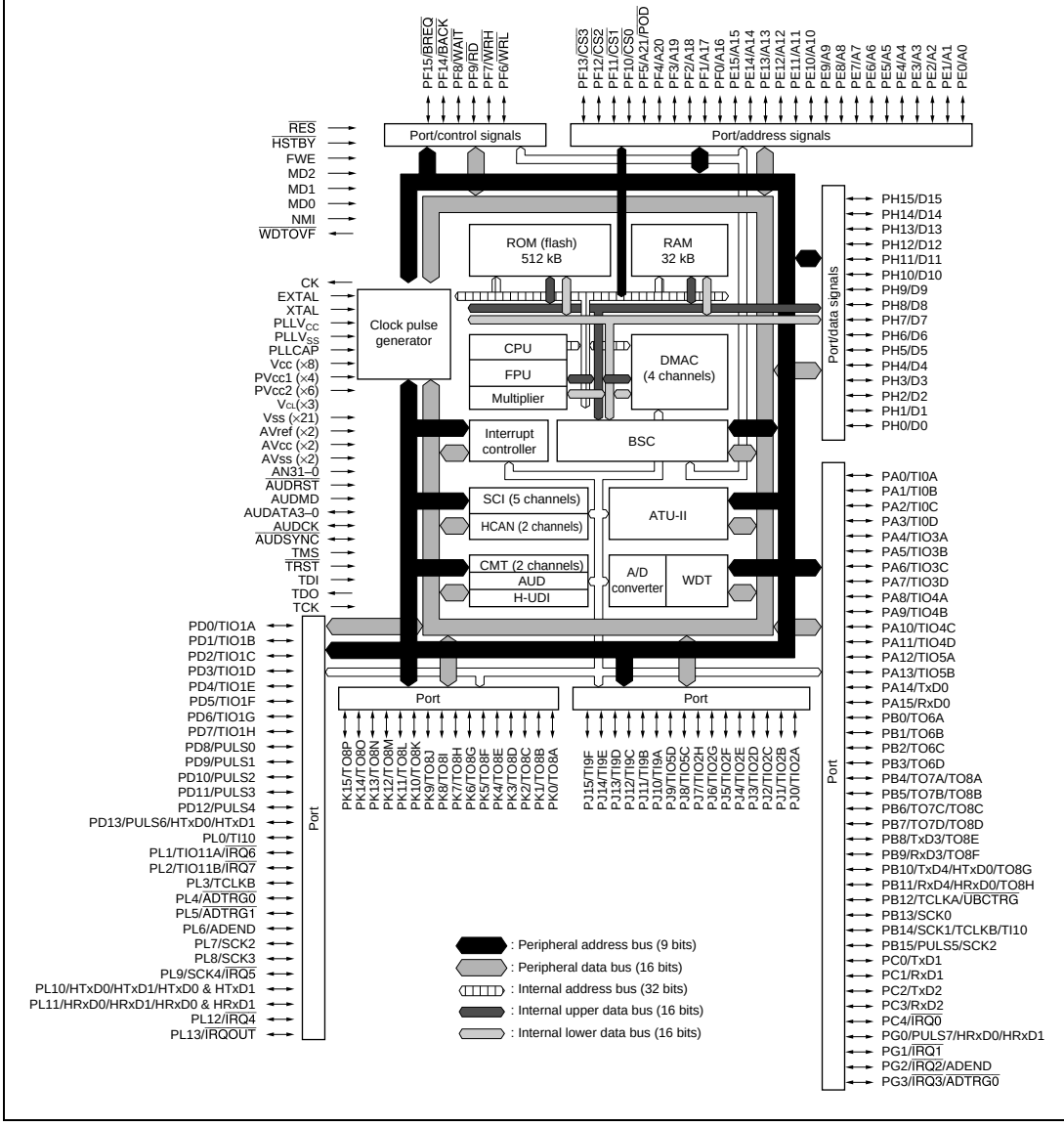


Figure 1.1 Block Diagram

1.3.1 Pin Arrangement

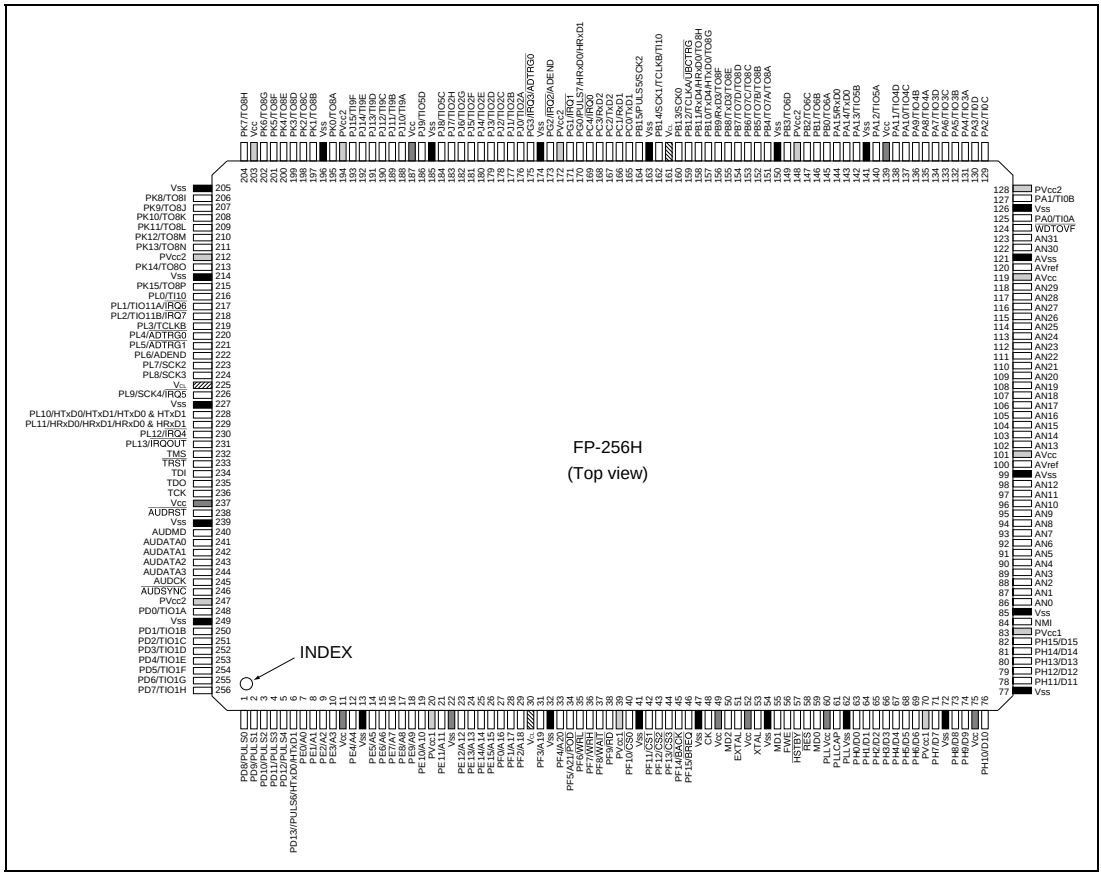


Figure 1.2 Pin Arrangement

Table 1.2 summarizes the pin functions.

Table 1.2 Pin Functions

Type	Symbol	Pin No.	I/O	Name	Function
Power supply	V_{CC}	11, 49, 52, 75, 139, 187, 203, 237	Input	Power supply	Power supply for chip-internal and system ports ($\overline{RES}$, MD2–MD0, FWE, $\overline{HSTBY}$, NMI, CK, EXTAL, XTAL, H-UDI port). Connect all V_{CC} pins to the system power supply. The chip will not operate if there are any open pins.
	$PV_{CC}1$	20, 39, 70, 83	Input	Port power supply 1	Power supply for bus ports (ports E, F, and H). Connect all $PV_{CC}1$ pins to the system bus power supply. The chip will not operate if there are any open pins.
	$PV_{CC}2$	128, 148, 172, 194, 212, 247	Input	Port power supply 2	Power supply for peripheral module ports (ports A, B, C, D, G, J, K, and L, the AUD port, and $\overline{WDTOVF}$). Connect all $PV_{CC}2$ pins to the system peripheral module power supply. The chip will not operate if there are any open pins.
	V_{CL}	30, 161, 225	Input	Internal step-down power supply	Pins for connection to a capacitor used for stablizing the voltage of the internal step-down power supply. Connect this pin to V_{SS} throhgh a capacitor. The capacitor should be located near the pin. Do not connect to an external power supply.

		41, 47, 54, 72, 77, 85, 126, 141, 150, 163, 174, 185, 196, 205, 214, 227, 239, 249			Connect all V_{SS} pins to the system ground. The chip will not operate if there are any open pins.
Flash memory	FWE	56	Input	Flash write enable	Connected to ground in normal operation. Apply V_{CC} during on-board programming.

Clock	$PLLV_{cc}$	60	Input	PLL power supply	On-chip PLL oscillator power supply. For power supply connection, see section 5, Clock Pulse Generator (CPG).
	$PLLV_{ss}$	62	Input	PLL ground	On-chip PLL oscillator ground. For power supply connection, see section 5, Clock Pulse Generator (CPG).
	PLLCAP	61	Input	PLL capacitance	On-chip PLL oscillator external capacitance connection pin. For external capacitance connection, see section 5, Clock Pulse Generator (CPG).
	EXTAL	51	Input	External clock	For connection to a crystal resonator. An external clock source can also be connected to the EXTAL pin.
	XTAL	53	Input	Crystal	For connection to a crystal resonator.
	CK	48	Output	System clock	Supplies the system clock to peripheral devices.
System control	$\overline{RES}$	58	Input	Power-on reset	Executes a power-on reset when driven low.
	$\overline{WDTOVF}$	124	Output	Watchdog timer overflow	WDT overflow output signal.
	$\overline{BREQ}$	46	Input	Bus request	Driven low when an external device requests the bus.
	$\overline{BACK}$	45	Output	Bus request acknowledge	Indicates that the bus has been granted to an external device. The device that output the $\overline{BREQ}$ signal recognizes that the bus has been acquired when it receives the $\overline{BACK}$ signal.

Operating mode control	MD0 to MD2	59, 55, 50	Input	Mode setting	These pins determine the operating mode. Do not change the input values during operation.
	$\overline{\text{HSTBY}}$	57	Input	Hardware standby	When driven low, this pin forces a transition to hardware standby mode.
Interrupts	NMI	84	Input	Nonmaskable interrupt	Nonmaskable interrupt request pin. Acceptance on the rising edge or falling edge can be selected.
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$	169, 171, 173, 175, 230, 226, 217, 218	Input	Interrupt requests 0 to 7	Maskable interrupt request pins. Level input or edge input can be selected.
	$\overline{\text{IRQOUT}}$	231	Output	Interrupt request output	Indicates that an interrupt has been generated. Enables interrupt generation to be recognized in the bus-released state.
Address bus	A0–A21	7–10, 12, 14–19, 21, 23–29, 31, 33, 34	Output	Address bus	Address output pins.
Data bus	D0–D15	63–69, 71, 73, 74, 76, 78–82	Input/ output	Data bus	16-bit bidirectional data bus pins.
Bus control	$\overline{\text{CS0}}$–$\overline{\text{CS3}}$	40, 42–44	Output	Chip select 0 to 3	Chip select signals for external memory or devices.
	$\overline{\text{RD}}$	38	Output	Read	Indicates reading from an external device.
	$\overline{\text{WRH}}$	36	Output	Upper write	Indicates writing of the upper 8 bits of external data.
	$\overline{\text{WRL}}$	35	Output	Lower write	Indicates writing of the lower 8 bits of external data.
	$\overline{\text{WAIT}}$	37	Input	Wait	Input for wait cycle insertion in bus cycles during external space access.

Advanced timer unit-II (ATU-II)	TCLKA TCLKB	159, 162, 219	Input	ATU-II timer clock input	ATU-II counter external clock Input pins.
	TI0A–TI0D	125, 127, 129, 130	Input	ATU-II input capture (channel 0)	Channel 0 input capture input pins.
	TIO1A–TIO1H	248, 250–256	Input/ output	ATU-II input capture/output compare (channel 1)	Channel 1 input capture input/output compare output pins.
	TIO2A–TIO2H	176–183	Input/ output	ATU-II input capture/output compare (channel 2)	Channel 2 input capture input/output compare output pins.
	TIO3A–TIO3D	131–134	Input/ output	ATU-II input capture/output compare/ PWM output (channel 3)	Channel 3 input capture input/output compare/PWM output pins.
	TIO4A–TIO4D	135–138	Input/ output	ATU-II input capture/output compare/ PWM output (channel 4)	Channel 4 input capture input/output compare/PWM output pins.
	TIO5A–TIO5D	140, 142, 184, 186	Input/ output	ATU-II input capture/output compare/ PWM output (channel 5)	Channel 5 input capture input/output compare/PWM output pins.
	TO6A–TO6D	145–147, 149	Output	ATU-II PWM output (channel 6)	Channel 6 PWM output pins.
	TO7A–TO7D	151–154	Output	ATU-II PWM output (channel 7)	Channel 7 PWM output pins.
	TO8A–TO8P	151–158, 195, 197–202, 204, 206–211, 213, 215	Output	ATU-II one-shot pulse (channel 8)	Channel 8 down-counter one-shot pulse output pins.

Advanced timer unit-II (ATU-II)	TI9A–TI9F	188–193	Input	ATU-II event input (channel 9)	Channel 9 event counter input pins.
	TI10	162, 216	Input	ATU-II multiplied clock generation (channel 10)	Channel 10 external clock input pin.
	TIO11A, TIO11B	217, 218	Input/ output	ATU-II input capture/output compare	Channel 11 input capture input/output compare output pins.
Advanced pulse controller (APC)	PULS0–PULS7	1–6, 164, 170	Output	APC pulse outputs 0 to 7	APC pulse output pins.
Serial communication interface (SCI)	TxD0–TxD4	143, 165, 167, 155, 157	Output	Transmit data (channels 0 to 4)	SCI0 to SCI4 transmit data output pins.
	RxD0–RxD4	144, 166, 168, 156, 158	Input	Receive data (channels 0 to 4)	SCI0 to SCI4 receive data input pins.
	SCK0–SCK4	160, 162, 223, 224, 226, 164	Input/ output	Serial clock (channels 0 to 4)	SCI0 to SCI4 clock input/output pins.
Controller area network (HCAN)	HTxD0, HTxD1	157, 228, 6	Output	Transmit data	CAN bus transmit data output pins.
	HRxD0, HRxD1	158, 229, 170	input	Receive data	CAN bus receive data input pins.
A/D converter	AV_{CC}	101, 119	Input	Analog power supply	A/D converter power supply.
	AV_{SS}	99, 121	Input	Analog ground	A/D converter power supply.
	AV_{ref}	100, 120	Input	Analog reference power supply	Analog reference power supply input pins.
	AN0–AN31	86–98, 102–118, 122, 123	Input	Analog input	Analog signal input pins.
	$\overline{\text{ADTRG0}}$, $\overline{\text{ADTRG1}}$	175, 220, 221	Input	A/D conversion trigger input	External trigger input pins for starting A/D conversion.
	ADEND	173, 222	Output	ADEND output	A/D2 channel 31 conversion timing monitor output pins.

User break controller (UBC)	$\overline{\text{UBCTRG}}$	159	Output	User break trigger output	UBC condition match trigger output pin.
High-Performance user debug interface (H-UDI)	TCK	236	Input	Test clock	Test clock input pin.
	TMS	232	Input	Test mode select	Test mode select signal input pin.
	TDI	234	Input	Test data input	Instruction/data serial input pin.
	TDO	235	Output	Test data output	Instruction/data serial output pin.
	$\overline{\text{TRST}}$	233	Input	Test reset	Initialization signal input pin.
Advanced user debugger (AUD)	AUDATA0–AUDATA3	241–244	Input/ output	AUD data	Realtime trace mode: Branch destination address output pins. RAM monitor mode: Monitor address input / data input/output pins.
	$\overline{\text{AUDRST}}$	238	Input	AUD reset	Reset signal input pin.
	AUDMD	240	Input	AUD mode	Mode select signal input pin. Realtime trace mode: Low RAM monitor mode: High
	AUDCK	245	Input/ output	AUD clock	Realtime trace mode: Serial clock output pin. RAM monitor mode: Serial clock input pin.
	$\overline{\text{AUDSYNC}}$	246	Input/ output	AUD synchronization signal	Realtime trace mode: Data start position identification signal output pin. RAM monitor mode: Data start position identification signal input pin.
I/O ports	$\overline{\text{POD}}$	34	Input	Port output disable	Input pin for port pin drive control when general port is set for output.
	PA0–PA15	125, 127, 129–138, 140, 142–144	Input/ output	Port A	General input/output port pins. Input or output can be specified bit by bit.

I/O ports	PB0–PB15	145–147, 149, 151–160, 162, 164	Input/ output	Port B	General input/output port pins. Input or output can be specified bit by bit.
	PC0–PC4	165–169	Input/ output	Port C	General input/output port pins. Input or output can be specified bit by bit.
	PD0–PD13	248, 250–256, 1–6	Input/ output	Port D	General input/output port pins. Input or output can be specified bit by bit.
	PE0–PE15	7–10, 12, 14–19, 21, 23–26	Input/ output	Port E	General input/output port pins. Input or output can be specified bit by bit.
	PF0–PF15	27–29, 31, 33–38, 40, 42–46	Input/ output	Port F	General input/output port pins. Input or output can be specified bit by bit.
	PG0–PG3	170, 171, 173, 175	Input/ output	Port G	General input/output port pins. Input or output can be specified bit by bit.
	PH0–PH15	63–69, 71, 73, 74, 76, 78–82	Input/ output	Port H	General input/output port pins. Input or output can be specified bit by bit.
	PJ0–PJ15	176–184, 186, 188–193	Input/ output	Port J	General input/output port pins. Input or output can be specified bit by bit.
	PK0–PK15	195, 197–202, 204, 206–211, 213, 215	Input/ output	Port K	General input/output port pins. Input or output can be specified bit by bit.
	PL0–PL13	216–224, 226, 228–231	Input/ output	Port L	General input/output port pins. Input or output can be specified bit by bit.

Table 1.5 Pin Assignments

Pin No.	MCU Mode	Programmer Mode
1	PD8/PULS0	N.C
2	PD9/PULS1	N.C
3	PD10/PULS2	N.C
4	PD11/PULS3	N.C
5	PD12/PULS4	N.C
6	PD13/PULS6/HTxD0/HTxD1	N.C
7	PE0/A0	A0
8	PE1/A1	A1
9	PE2/A2	A2
10	PE3/A3	A3
11	Vcc	Vcc
12	PE4/A4	A4
13	Vss	Vss
14	PE5/A5	A5
15	PE6/A6	A6
16	PE7/A7	A7
17	PE8/A8	A8
18	PE9/A9	A9
19	PE10/A10	A10
20	PVcc1	Vcc
21	PE11/A11	A11
22	Vss	Vss
23	PE12/A12	A12
24	PE13/A13	A13
25	PE14/A14	A14
26	PE15/A15	A15
27	PF0/A16	A16
28	PF1/A17	A17
29	PF2/A18	A18
30	V_{CL}	V_{CL}

31	PF3/A19	A19
32	Vss	Vss
33	PF4/A20	N.C.
34	PF5/A21/$\overline{\text{POD}}$	N.C.
35	PF6/$\overline{\text{WRL}}$	N.C.
36	PF7/$\overline{\text{WRH}}$	N.C.
37	PF8/$\overline{\text{WAIT}}$	Vcc
38	PF9/$\overline{\text{RD}}$	N.C.
39	PVcc1	Vcc
40	PF10/$\overline{\text{CS0}}$	N.C.
41	Vss	Vss
42	PF11/$\overline{\text{CS1}}$	Vcc
43	PF12/$\overline{\text{CS2}}$	Vcc
44	PF13/$\overline{\text{CS3}}$	Vss
45	PF14/$\overline{\text{BACK}}$	N.C.
46	PF15/$\overline{\text{BREQ}}$	Vcc
47	Vss	Vss
48	CK	N.C.
49	Vcc	Vcc
50	MD2	Vss
51	EXTAL	EXTAL
52	Vcc	Vcc
53	XTAL	XTAL
54	Vss	Vss
55	MD1	Vcc
56	FWE	FWE
57	$\overline{\text{HSTBY}}$	Vcc
58	$\overline{\text{RES}}$	$\overline{\text{RES}}$
59	MD0	Vcc
60	PLLVcc	PLLVcc
61	PLLCAP	PLLCAP

62	PLLVss	PLLVss
63	PH0/D0	D0
64	PH1/D1	D1
65	PH2/D2	D2
66	PH3/D3	D3
67	PH4/D4	D4
68	PH5/D5	D5
69	PH6/D6	D6
70	PVcc1	Vcc
71	PH7/D7	D7
72	Vss	Vss
73	PH8/D8	N.C.
74	PH9/D9	N.C.
75	Vcc	Vcc
76	PH10/D10	N.C.
77	Vss	Vss
78	PH11/D11	N.C.
79	PH12/D12	N.C.
80	PH13/D13	N.C.
81	PH14/D14	N.C.
82	PH15/D15	N.C.
83	PVcc1	Vcc
84	NMI	Vss
85	Vss	Vss
86	AN0	N.C.
87	AN1	N.C.
88	AN2	N.C.
89	AN3	N.C.
90	AN4	N.C.
91	AN5	N.C.
92	AN6	N.C.

93	AN7	N.C.
94	AN8	N.C.
95	AN9	N.C.
96	AN10	N.C.
97	AN11	N.C.
98	AN12	N.C.
99	AVss	Vss
100	AVref	Vcc
101	AVcc	Vcc
102	AN13	N.C.
103	AN14	N.C.
104	AN15	N.C.
105	AN16	N.C.
106	AN17	N.C.
107	AN18	N.C.
108	AN19	N.C.
109	AN20	N.C.
110	AN21	N.C.
111	AN22	N.C.
112	AN23	N.C.
113	AN24	N.C.
114	AN25	N.C.
115	AN26	N.C.
116	AN27	N.C.
117	AN28	N.C.
118	AN29	N.C.
119	AVcc	Vcc
120	AVref	Vcc
121	AVss	Vss
122	AN30	N.C.
123	AN31	N.C.

124	$\overline{\text{WDTOVF}}$	N.C.
125	PA0/TI0A	N.C.
126	Vss	Vss
127	PA1/TI0B	N.C.
128	PVcc2	Vcc
129	PA2/TI0C	N.C.
130	PA3/TI0D	N.C.
131	PA4/TIO3A	N.C.
132	PA5/TIO3B	N.C.
133	PA6/TIO3C	N.C.
134	PA7/TIO3D	N.C.
135	PA8/TIO4A	N.C.
136	PA9/TIO4B	N.C.
137	PA10/TIO4C	N.C.
138	PA11/TIO4D	N.C.
139	Vcc	Vcc
140	PA12/TIO5A	N.C.
141	Vss	Vss
142	PA13/TIO5B	N.C.
143	PA14/TxD0	N.C.
144	PA15/RxD0	N.C.
145	PB0/TO6A	N.C.
146	PB1/TO6B	N.C.
147	PB2/TO6C	N.C.
148	PVcc2	Vcc
149	PB3/TO6D	N.C.
150	Vss	Vss
151	PB4/TO7A/TO8A	N.C.
152	PB5/TO7B/TO8B	N.C.
153	PB6/TO7C/TO8C	N.C.
154	PB7/TO7D/TO8D	N.C.

155	PB8/TxD3/TO8E	N.C.
156	PB9/RxD3/TO8F	N.C.
157	PB10/TxD4/HTxD0/TO8G	N.C.
158	PB11/RxD4/HRxD0/TO8H	N.C.
159	PB12/TCLKA/$\overline{\text{UBCTRG}}$	N.C.
160	PB13/SCK0	N.C.
161	V_{CL}	V_{CL}
162	PB14/SCK1/TCLKB/TI10	N.C.
163	Vss	Vss
164	PB15/PULS5/SCK2	N.C.
165	PC0/TxD1	N.C.
166	PC1/RxD1	N.C.
167	PC2/TxD2	N.C.
168	PC3/RxD2	N.C.
169	PC4/$\overline{\text{IRQ0}}$	N.C.
170	PG0/PULS7/HRxD0/HRxD1	N.C.
171	PG1/$\overline{\text{IRQ1}}$	N.C.
172	PVcc2	Vcc
173	PG2/$\overline{\text{IRQ2}}$/ADEND	N.C.
174	Vss	Vss
175	PG3/$\overline{\text{IRQ3}}$/$\overline{\text{ADTRG0}}$	N.C.
176	PJ0/TIO2A	N.C.
177	PJ1/TIO2B	N.C.
178	PJ2/TIO2C	N.C.
179	PJ3/TIO2D	N.C.
180	PJ4/TIO2E	N.C.
181	PJ5/TIO2F	N.C.
182	PJ6/TIO2G	N.C.
183	PJ7/TIO2H	N.C.
184	PJ8/TIO5C	N.C.
185	Vss	Vss

186	PJ9/TIO5D	N.C.
187	Vcc	Vcc
188	PJ10/TI9A	N.C.
189	PJ11/TI9B	N.C.
190	PJ12/TI9C	N.C.
191	PJ13/TI9D	N.C.
192	PJ14/TI9E	N.C.
193	PJ15/TI9F	N.C.
194	PVcc2	Vcc
195	PK0/TO8A	N.C.
196	Vss	Vss
197	PK1/TO8B	N.C.
198	PK2/TO8C	N.C.
199	PK3/TO8D	N.C.
200	PK4/TO8E	N.C.
201	PK5/TO8F	N.C.
202	PK6/TO8G	N.C.
203	Vcc	Vcc
204	PK7/TO8H	N.C.
205	Vss	Vss
206	PK8/TO8I	N.C.
207	PK9/TO8J	N.C.
208	PK10/TO8K	N.C.
209	PK11/TO8L	N.C
210	PK12/TO8M	N.C.
211	PK13/TO8N	N.C.
212	PVcc2	Vcc
213	PK14/TO8O	N.C.
214	Vss	Vss
215	PK15/TO8P	N.C.
216	PL0/TI10	N.C.

217	PL1/TIO11A/$\overline{\text{IRQ6}}$	N.C.
218	PL2/TIO11B/$\overline{\text{IRQ7}}$	$\overline{\text{CE}}$
219	PL3/TCLKB	N.C.
220	PL4/$\overline{\text{ADTRG0}}$	N.C.
221	PL5/$\overline{\text{ADTRG1}}$	N.C.
222	PL6/ADEND	N.C.
223	PL7/SCK2	N.C.
224	PL8/SCK3	N.C.
225	V_{CL}	V_{CL}
226	PL9/SCK4/$\overline{\text{IRQ5}}$	$\overline{\text{WE}}$
227	Vss	Vss
228	PL10/HTxD0/HTxD1/HTxD0 & HTxD1	N.C.
229	PL11/HRxD0/HRxD1/HRxD0 & HRxD1	N.C.
230	PL12/$\overline{\text{IRQ4}}$	$\overline{\text{OE}}$
231	PL13/$\overline{\text{IRQOUT}}$	N.C.
232	TMS	N.C.
233	$\overline{\text{TRST}}$	N.C.
234	TDI	N.C.
235	TDO	N.C.
236	TCK	N.C.
237	Vcc	Vcc
238	$\overline{\text{AUDRST}}$	N.C.
239	Vss	Vss
240	AUDMD	N.C.
241	AUDATA0	N.C.
242	AUDATA1	N.C.
243	AUDATA2	N.C.
244	AUDATA3	N.C.
245	AUDCK	N.C.
246	$\overline{\text{AUDSYNC}}$	N.C.
247	PVcc2	Vcc

248	PD0/TIO1A	N.C.
249	Vss	Vss
250	PD1/TIO1B	N.C.
251	PD2/TIO1C	N.C.
252	PD3/TIO1D	N.C.
253	PD4/TIO1E	N.C.
254	PD5/TIO1F	N.C.
255	PD6/TIO1G	N.C.
256	PD7/TIO1H	N.C.

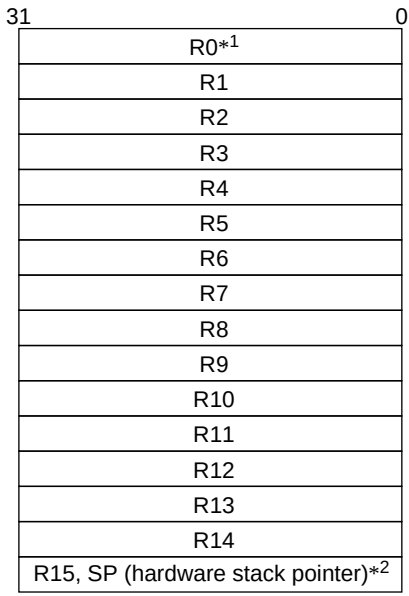
2.1 Register Configuration

The register set consists of sixteen 32-bit general registers, three 32-bit control registers and four 32-bit system registers.

In addition, the FPU has eighteen internal registers: sixteen 32-bit floating-point registers and two 32-bit floating-point system registers.

2.1.1 General Registers (Rn)

The sixteen 32-bit general registers (Rn) are numbered R0–R15. General registers are used for data processing and address calculation. R0 is also used as an index register. Several instructions have R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Saving and recovering the status register (SR) and program counter (PC) in exception processing is accomplished by referencing the stack using R15. Figure 2.1 shows the general registers.



Notes: *1. R0 functions as an index register in the indirect indexed register addressing mode and indirect indexed GBR addressing mode. In some instructions, R0 functions as a fixed source register or destination register.
*2. R15 functions as a hardware stack pointer (SP) during exception processing.

Figure 2.1 General Registers

The 32-bit control registers consist of the 32-bit status register (SR), global base register (GBR), and vector base register (VBR). The status register indicates processing states. The global base register functions as a base address for the indirect GBR addressing mode to transfer data to the registers of on-chip peripheral modules. The vector base register functions as the base address of the exception processing vector area (including interrupts). Figure 2.2 shows the control registers.

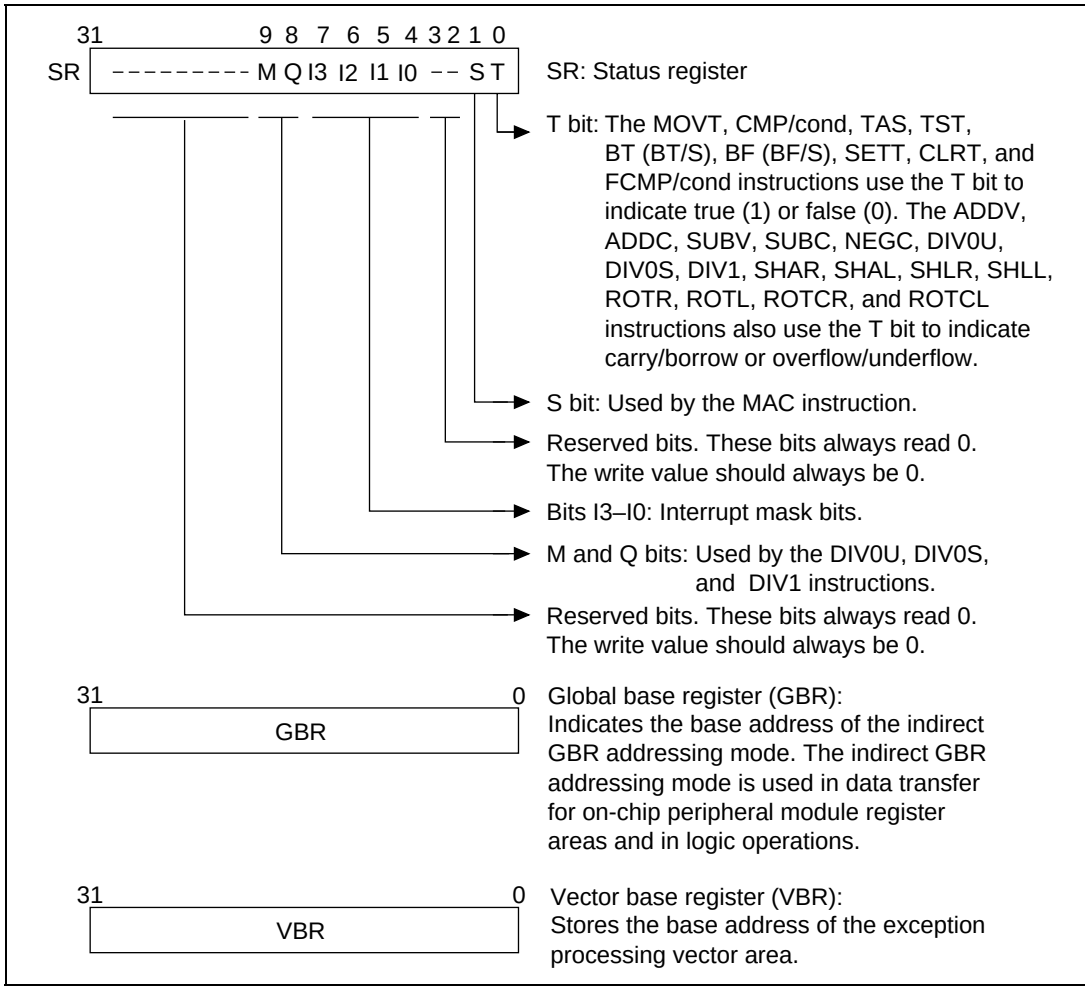


Figure 2.2 Control Register Configuration

System registers consist of four 32-bit registers: high and low multiply-and-accumulate registers (MACH and MACL), the procedure register (PR), and the program counter (PC). The multiply-and-accumulate registers store the results of multiply-and-accumulate operations. The procedure register stores the return address from a subroutine procedure. The program counter stores program addresses to control the flow of the processing. Figure 2.3 shows the system registers.

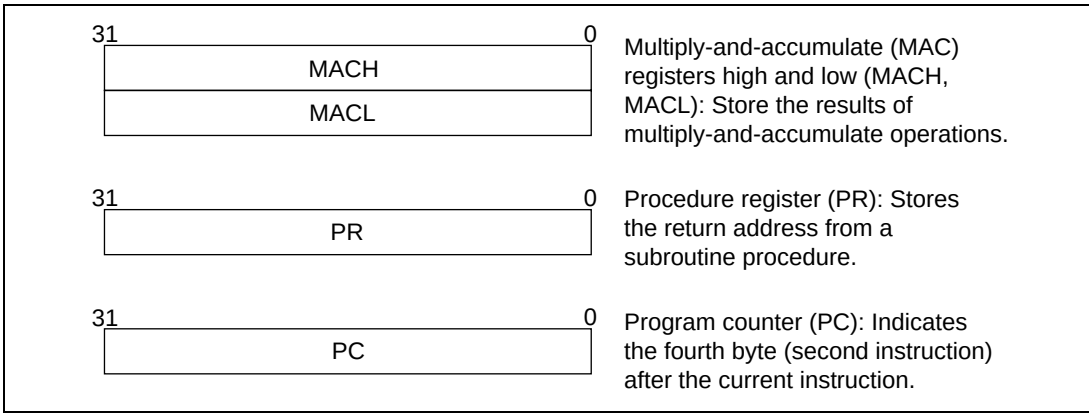


Figure 2.3 System Register Configuration

There are sixteen 32-bit floating-point registers, designated FR0 to FR15, which are used by floating-point instructions. FR0 functions as the index register for the FMAC instruction. These registers are incorporated into the floating-point unit (FPU). For details, see section 3, Floating-Point Unit (FPU).

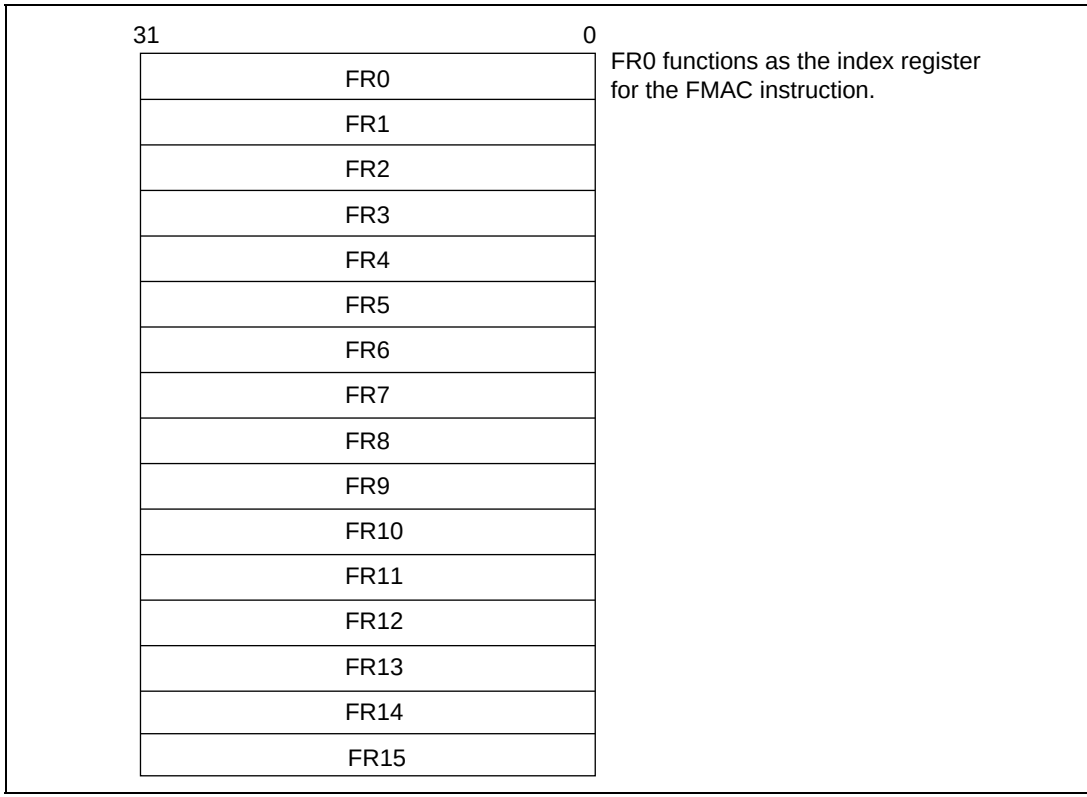


Figure 2.4 Floating-Point Registers

There are two 32-bit floating-point system registers: the floating-point communication register (FPUL) and the floating-point status/control register (FPSCR). FPUL is used for communication between the CPU and the floating-point unit (FPU). FPSCR indicates and stores status/control information relating to FPU exceptions.

These registers are incorporated into the floating-point unit (FPU). For details, see section 3, Floating-Point Unit (FPU).

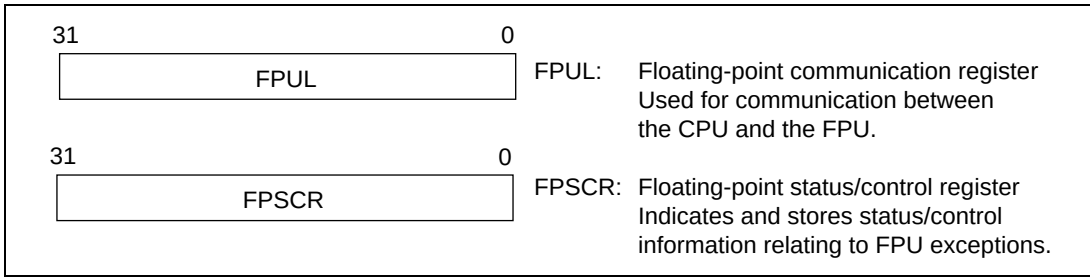


Figure 2.5 Floating-Point System Registers

2.1.6 Initial Values of Registers

Table 2.1 lists the values of the registers after reset.

Table 2.1 Initial Values of Registers

Classification	Register	Initial Value
General registers	R0–R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I3–I0 are 1111 (H'F), reserved bits are 0, and other bits are undefined
	GBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector address table
Floating-point registers	FR0–FR15	Undefined
Floating-point system registers	FPUL	Undefined
	FPSCR	H'00040001

2.2.1 Data Format in Registers

Register operands are always longwords (32 bits). When the memory operand is only a byte (8 bits) or a word (16 bits), it is sign-extended into a longword when loaded into a register (figure 2.6).

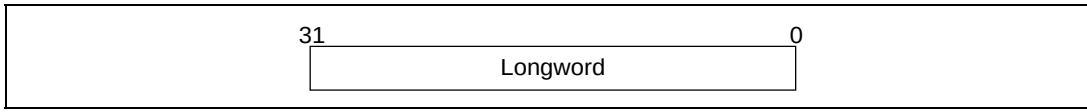


Figure 2.6 Data Format in Registers

2.2.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Byte data can be accessed from any address, but an address error will occur if an attempt is made to access word data starting from an address other than 2n or longword data starting from an address other than 4n. In such cases, the data accessed cannot be guaranteed. The hardware stack area, referred to by the hardware stack pointer (SP, R15), uses only longword data starting from address 4n because this area holds the program counter and status register (figure 2.7).

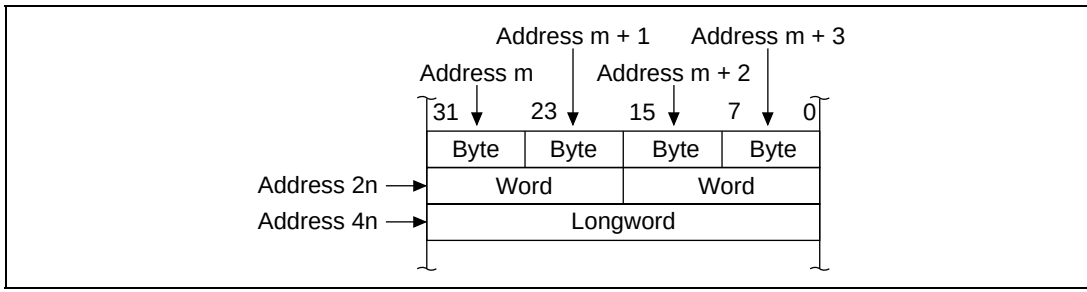


Figure 2.7 Data Formats in Memory

2.2.3 Immediate Data Format

Byte (8 bit) immediate data resides in an instruction code. Immediate data accessed by the MOV, ADD, and CMP/EQ instructions is sign-extended and handled in registers as longword data. Immediate data accessed by the TST, AND, OR, and XOR instructions is zero-extended and handled as longword data. Consequently, AND instructions with immediate data always clear the upper 24 bits of the destination register.

PC relative addressing mode with displacement.

2.3 Instruction Features

2.3.1 RISC-Type Instruction Set

All instructions are RISC type. This section details their functions.

16-Bit Fixed Length: All instructions are 16 bits long, increasing program code efficiency.

One Instruction per Cycle: The microprocessor can execute basic instructions in one cycle using the pipeline system. Instructions are executed in 25 ns at 40 MHz.

Data Length: Longword is the standard data length for all operations. Memory can be accessed in bytes, words, or longwords. Byte or word data accessed from memory is sign-extended and handled as longword data. Immediate data is sign-extended for arithmetic operations or zero-extended for logic operations. It also is handled as longword data (table 2.2).

Table 2.2 Sign Extension of Word Data

SH7055SF CPU		Description	Example of Conventional CPU
`MOV.W`	`@(disp,PC),R1`	Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction.	`ADD.W #H'1234,R0`
`ADD`	`R1,R0`		
	`.........`		
`.DATA.W`	`H'1234`		

Note: @(disp, PC) accesses the immediate data.

Load-Store Architecture: Basic operations are executed between registers. For operations that involve memory access, data is loaded to the registers and executed (load-store architecture). Instructions such as AND that manipulate bits, however, are executed directly in memory.

Delayed Branch Instructions: Unconditional branch instructions are delayed branch instructions. With a delayed branch instruction, the branch is taken after execution of the instruction following the delayed branch instruction. There are two types of conditional branch instructions: delayed branch instructions and ordinary branch instructions.

BRA	TRGET	Executes the ADD before branching to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

Multiply/Multiply-and-Accumulate Operations: 16-bit × 16-bit → 32-bit multiply operations are executed in one to two cycles. 16-bit × 16-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to three cycles. 32-bit × 32-bit → 64-bit multiply and 32-bit × 32-bit + 64bit → 64-bit multiply-and-accumulate operations are executed in two to four cycles.

T Bit: The T bit in the status register changes according to the result of the comparison, and in turn is the condition (true/false) that determines if the program will branch. The number of instructions that change the T bit is kept to a minimum to improve the processing speed (table 2.4).

Table 2.4 T Bit

SH7055SF CPU		Description	Example of Conventional CPU	
CMP/GE	R1,R0	T bit is set when R0 = R1. The program branches to TRGET0 when R0 = R1 and to TRGET1 when R0 < R1.	CMP.W	R1,R0
BT	TRGET0		BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#1,R0	T bit is not changed by ADD. T bit is set when R0 = 0. The program branches if R0 = 0.	SUB.W	#1,R0
CMP/EQ	#0,R0		BEQ	TRGET
BT	TRGET			

Immediate Data: Byte (8-bit) immediate data resides in the instruction code. Word or longword immediate data is not input via instruction codes but is stored in a memory table. An immediate data transfer instruction (MOV) accesses the memory table using the PC relative addressing mode with displacement (table 2.5).

8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOV.W	@(disp,PC),R0	MOV.W	#H'1234,R0
		.....		
	.DATA.W	H'1234		
32-bit immediate	MOV.L	@(disp,PC),R0	MOV.L	#H'12345678,R0
		.....		
	.DATA.L	H'12345678		

Note: @(disp, PC) accesses the immediate data.

Absolute Address: When data is accessed by absolute address, the value already in the absolute address is placed in the memory table. Loading the immediate data when the instruction is executed transfers that value to the register and the data is accessed in the indirect register addressing mode (table 2.6).

Table 2.6 Absolute Address Accessing

Classification	SH7055SF CPU		Example of Conventional CPU	
Absolute address	MOV.L	@(disp,PC),R1	MOV.B	@H'12345678,R0
	MOV.B	@R1,R0		
		.....		
	.DATA.L	H'12345678		

Note: @(disp,PC) accesses the immediate data.

16-Bit/32-Bit Displacement: When data is accessed by 16-bit or 32-bit displacement, the pre-existing displacement value is placed in the memory table. Loading the immediate data when the instruction is executed transfers that value to the register and the data is accessed in the indirect indexed register addressing mode (table 2.7).

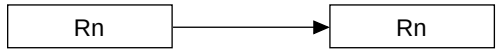
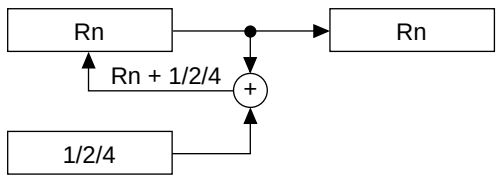
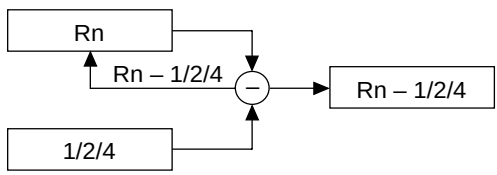
Table 2.7 Displacement Accessing

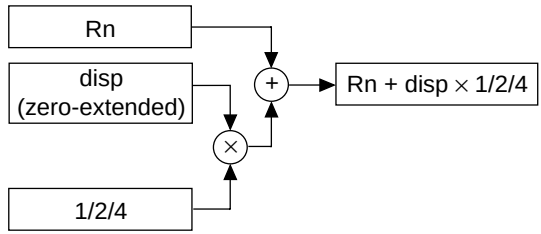
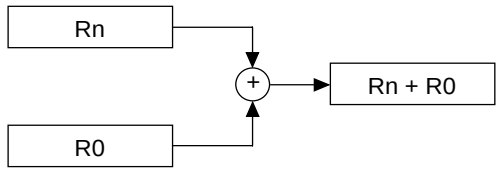
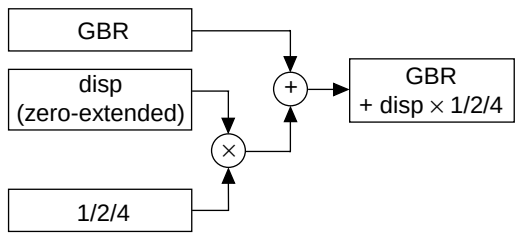
Classification	SH7055SF CPU		Example of Conventional CPU	
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W	@(H'1234,R1),R2
	MOV.W	@(R0,R1),R2		
		.....		
	.DATA.W	H'1234		

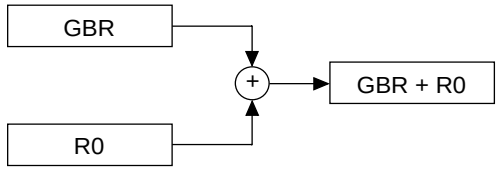
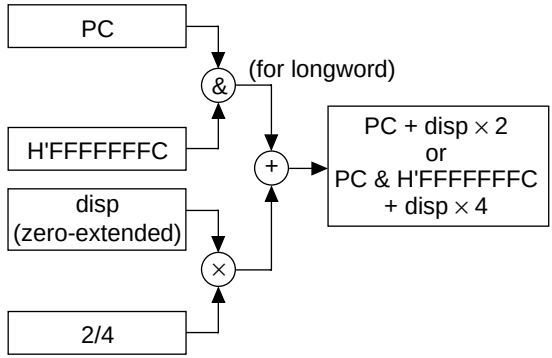
Note: @(disp,PC) accesses the immediate data.

Table 2.8 describes addressing modes and effective address calculation.

Table 2.8 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Direct register addressing	Rn	The effective address is register Rn. (The operand is the contents of register Rn.)	—
Indirect register addressing	@Rn	The effective address is the contents of register Rn. 	Rn
Post-increment indirect register addressing	@Rn+	The effective address is the contents of register Rn. A constant is added to the content of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Rn (After the instruction executes) Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn
Pre-decrement indirect register addressing	@–Rn	The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction executed with Rn after calculation)

Mode	Format	Effective Address Calculation	Equation
Indirect register addressing with displacement	@(disp:4, Rn)	The effective address is Rn plus a 4-bit displacement (disp). The value of disp is zero-extended, and remains the same for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Indirect indexed register addressing	@(R0, Rn)	The effective address is the Rn value plus R0. 	Rn + R0
Indirect GBR addressing with displacement	@(disp:8, GBR)	The effective address is the GBR value plus an 8-bit displacement (disp). The value of disp is zero-extended, and remains the same for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4

Mode	Format	Effective Address Calculation	Equation
Indirect indexed GBR addressing	@(R0, GBR)	The effective address is the GBR value plus R0. 	GBR + R0
Indirect PC addressing with displacement	@(disp:8, PC)	The effective address is the PC value plus an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword†operation, the lowest two bits of the PC value are masked. 	Word: PC + disp × 2 Longword: PC & H'FFFFFFFC + disp × 4

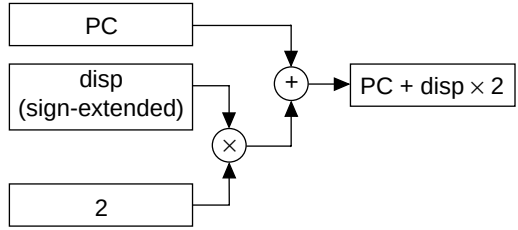
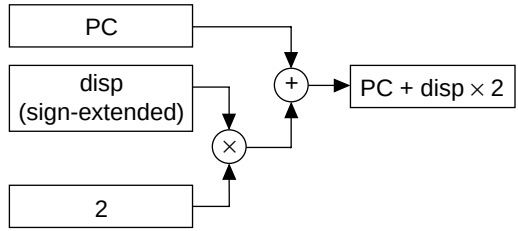
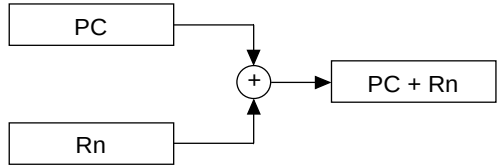
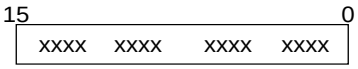
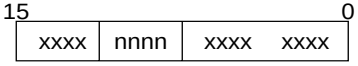
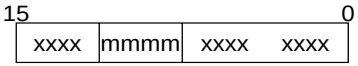
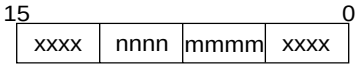
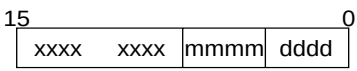
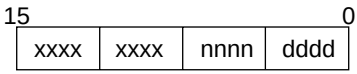
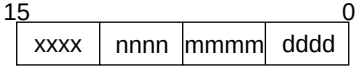
Mode	Format	Effective Addresses Calculation	Equation
PC relative addressing	`disp:8`	The effective address is the PC value sign-extended with an 8-bit displacement (disp), doubled, and added†to the PC value. 	PC + disp × 2
	`disp:12`	The effective address is the PC value sign-extended with a 12-bit displacement (disp), doubled, and added to the PC value. 	PC + disp × 2
	Rn	The effective address is the register PC value plus Rn. 	PC + Rn
Immediate addressing	`#imm:8`	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions is zero-extended.	—
	`#imm:8`	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions is sign-extended.	—
	`#imm:8`	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and quadrupled.	—

Table 2.9 lists the instruction formats for the source operand and the destination operand. The meaning of the operand depends on the instruction code. The symbols used are as follows:

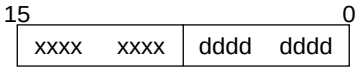
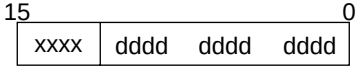
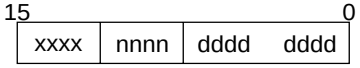
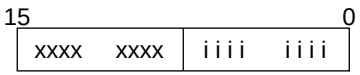
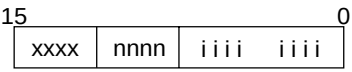
1. xxxx: Instruction code
2. mmmm: Source register
3. nnnn: Destination register
4. iiii: Immediate data
5. dddd: Displacement

Table 2.9 Instruction Formats

Instruction Formats	Source Operand	Destination Operand	Example
0 format 	—	—	`NOP`
n format 	—	nnnn: Direct register	`MOVT Rn`
	Control register or system register	nnnn: Direct register	`STS MACH,Rn`
	Control register or system register	nnnn: Indirect pre-decrement register	`STC.L SR,@-Rn`
m format 	mmmm: Direct register	Control register or system register	`LDC Rm,SR`
	mmmm: Indirect post-increment register	Control register or system register	`LDC.L @Rm+,SR`
	mmmm: Direct register	—	`JMP @Rm`
	mmmm: PC relative using Rm	—	`BRAF Rm`

Instruction Formats	Source Operand	Operand	Example
nm format 	mmmm: Direct register	nnnn: Direct register	ADD Rm,Rn
	mmmm: Direct register	nnnn: Indirect register	MOV.L Rm,@Rn
	mmmm: Indirect post-increment register (multiply-and-accumulate) nnnn*: Indirect post-increment register (multiply-and-accumulate)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: Indirect post-increment register	nnnn: Direct register	MOV.L @Rm+,Rn
	mmmm: Direct register	nnnn: Indirect pre-decrement register	MOV.L Rm,@-Rn
	mmmm: Direct register	nnnn: Indirect indexed register	MOV.L Rm,@(R0,Rn)
md format 	mmmmdddd: Indirect register with displacement	R0 (Direct register)	MOV.B @(disp,Rn),R0
nd4 format 	R0 (Direct register)	nnnndddd: Indirect register with displacement	MOV.B R0,@(disp,Rn)
nmd format 	mmmm: Direct register	nnnndddd: Indirect register with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: Indirect register with displacement	nnnn: Direct register	MOV.L @(disp,Rm),Rn

Note: * In multiply-and-accumulate instructions, nnnn is the source register.

Instruction Formats	Source Operand	Operand	Example
d format 	ddddddddd: Indirect GBR with displacement	R0 (Direct register)	MOV.L @(disp,GBR),R0
	R0 (Direct register)	ddddddddd: Indirect GBR with displacement	MOV.L R0,@(disp,GBR)
	ddddddddd: PC relative with displacement	R0 (Direct register)	MOVA @(disp,PC),R0
	—	ddddddddd: PC relative	BF label
d12 format 	—	ddddddddddddd: PC relative	BRA label (label = disp + PC)
nd8 format 	ddddddddd: PC relative with displacement	nnnn: Direct register	MOV.L @(disp,PC),Rn
i format 	iiiiiiii: Immediate	Indirect indexed GBR	AND.B #imm,@(R0,GBR)
	iiiiiiii: Immediate	R0 (Direct register)	AND #imm,R0
	iiiiiiii: Immediate	—	TRAPA #imm
ni format 	iiiiiiii: Immediate	nnnn: Direct register	ADD #imm,Rn

2.4 Instruction Set by Classification

2.4.1 Instruction Set by Classification

Table 2.10 lists the instructions according to their classification.

Classification	Types	Code	Function	Instructions
Data transfer	5	MOV	Data transfer, immediate data transfer, peripheral module data transfer, structure data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	
Arithmetic operations	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Initialization of signed division	
		DIV0U	Initialization of unsigned division	
		DMULS	Signed double-length multiplication	
		DMULU	Unsigned double-length multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate, double-length multiply-and-accumulate operation	
		MUL	Double-length multiply operation	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	

Classification	Types	Code	Function	Instructions
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	10	ROTL	One-bit left rotation	14
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
Branch	9	BF	Conditional branch, conditional branch with delay (Branch when T = 0)	11
		BT	Conditional branch, conditional branch with delay (Branch when T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	

Classification	Types	Code	Function	Instructions
System control	11	CLRT	T bit clear	31
		CLRMAC	MAC register clear	
		LDC	Load to control register	
		LDS	Load to system register	
		NOP	No operation	
		RTE	Return from exception processing	
		SETT	T bit set	
		SLEEP	Transition to power-down mode	
		STC	Store control register data	
		STS	Store system register data	
		TRAPA	Trap exception handling	
Floating-point instructions	15	FABS	Floating-point absolute value	22
		FADD	Floating-point addition	
		FCMP	Floating-point comparison	
		FDIV	Floating-point division	
		FLDI0	Floating-point load immediate 0	
		FLDI1	Floating-point load immediate 1	
		FLDS	Floating-point load into system register FPUL	
		FLOAT	Integer-to-floating-point conversion	
		FMAC	Floating-point multiply-and-accumulate operation	
		FMOV	Floating-point data transfer	
		FMUL	Floating-point multiplication	
		FNEG	Floating-point sign inversion	
		FSTS	Floating-point store from system register FPUL	
		FSUB	Floating-point subtraction	
		FTRC	Floating-point conversion with rounding to integer	
FPU-related CPU instructions	2	LDS	Load into floating-point system register	8
		STS	Store from floating-point system register	
Total:	79			172

Table 2.11 Instruction Code Format

Item	Format	Explanation
Instruction	`OP.Sz SRC,DEST`	OP: Operation code Sz: Size (B: byte, W: word, or L: longword) SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*1
Instruction code	MSB ↔ LSB	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 . . . 1111: R15 iiii: Immediate data dddd: Displacement
Operation	→, ←	Direction of transfer
	(xx)	Memory operand
	M/Q/T	Flag bits in the SR
	&	Logical AND of each bit
	|	Logical OR of each bit
	^	Exclusive OR of each bit
	~	Logical NOT of each bit
	<<n	n-bit left shift
	>>n	n-bit right shift
Execution cycles	—	Value when no wait states are inserted*2
T bit	—	Value of T bit after instruction is executed. An em-dash (—) in the column means no change.

Notes: *1. Depending on the operand size, displacement is scaled ×1, ×2, or ×4. For details, see the *SH-2E Programming Manual*.

*2. Instruction execution cycles: The execution cycles shown in the table are minimums. The actual number of cycles may be increased when (1) contention occurs between instruction fetches and data access, or (2) when the destination register of the load instruction (memory → register) and the register used by the next instruction are the same.

Instruction	Instruction Code	Operation	tion Cycles	T Bit
MOV #imm,Rn	1110nnnniiiiiiii	#imm → Sign extension → Rn	1	—
MOV.W @(disp,PC),Rn	1001nnnndddddddd	(disp × 2 + PC) → Sign extension → Rn	1	—
MOV.L @(disp,PC),Rn	1101nnnndddddddd	(disp × 4 + PC) → Rn	1	—
MOV Rm,Rn	0110nnnnmmmm0011	Rm → Rn	1	—
MOV.B Rm,@Rn	0010nnnnmmmm0000	Rm → (Rn)	1	—
MOV.W Rm,@Rn	0010nnnnmmmm0001	Rm → (Rn)	1	—
MOV.L Rm,@Rn	0010nnnnmmmm0010	Rm → (Rn)	1	—
MOV.B @Rm,Rn	0110nnnnmmmm0000	(Rm) → Sign extension → Rn	1	—
MOV.W @Rm,Rn	0110nnnnmmmm0001	(Rm) → Sign extension → Rn	1	—
MOV.L @Rm,Rn	0110nnnnmmmm0010	(Rm) → Rn	1	—
MOV.B Rm,@–Rn	0010nnnnmmmm0100	Rn–1 → Rn, Rm → (Rn)	1	—
MOV.W Rm,@–Rn	0010nnnnmmmm0101	Rn–2 → Rn, Rm → (Rn)	1	—
MOV.L Rm,@–Rn	0010nnnnmmmm0110	Rn–4 → Rn, Rm → (Rn)	1	—
MOV.B @Rm+,Rn	0110nnnnmmmm0100	(Rm) → Sign extension → Rn,Rm + 1 → Rm	1	—
MOV.W @Rm+,Rn	0110nnnnmmmm0101	(Rm) → Sign extension → Rn,Rm + 2 → Rm	1	—
MOV.L @Rm+,Rn	0110nnnnmmmm0110	(Rm) → Rn,Rm + 4 → Rm	1	—
MOV.B R0,@(disp,Rn)	10000000nnnndddd	R0 → (disp + Rn)	1	—
MOV.W R0,@(disp,Rn)	10000001nnnndddd	R0 → (disp × 2 + Rn)	1	—
MOV.L Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm → (disp × 4 + Rn)	1	—
MOV.B @(disp,Rm),R0	10000100mmmmdddd	(disp + Rm) → Sign extension → R0	1	—
MOV.W @(disp,Rm),R0	10000101mmmmdddd	(disp × 2 + Rm) → Sign extension → R0	1	—
MOV.L @(disp,Rm),Rn	0101nnnnmmmmdddd	(disp × 4 + Rm) → Rn	1	—
MOV.B Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm → (R0 + Rn)	1	—

Instruction		Instruction Code	Operation	tion Cycles	T Bit
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm → (R0 + Rn)	1	—
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0 + Rm) → Sign extension → Rn	1	—
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0 + Rm) → Sign extension → Rn	1	—
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0 + Rm) → Rn	1	—
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0 → (disp + GBR)	1	—
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0 → (disp × 2 + GBR)	1	—
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0 → (disp × 4 + GBR)	1	—
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp + GBR) → Sign extension → R0	1	—
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp × 2 + GBR) → Sign extension → R0	1	—
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1	—
MOVA	@(disp,PC),R0	11000111dddddddd	disp × 4 + PC → R0	1	—
MOVT	Rn	0000nnnn00101001	T → Rn	1	—
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm → Swap bottom two bytes → Rn	1	—
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm → Swap two consecutive words → Rn	1	—
XTRCT	Rm,Rn	0010nnnnmmmm1101	Rm: Middle 32 bits of Rn → Rn	1	—

Instruction		Instruction Code	Operation	tion Cycles	T Bit
ADD	Rm,Rn	0011nnnnmmmm1100	Rn + Rm → Rn	1	—
ADD	#imm,Rn	0111nnnniiiiiiii	Rn + imm → Rn	1	—
ADDC	Rm,Rn	0011nnnnmmmm1110	Rn + Rm + T → Rn, Carry → T	1	Carry
ADDV	Rm,Rn	0011nnnnmmmm1111	Rn + Rm → Rn, Overflow → T	1	Overflow
CMP/EQ	#imm,R0	10001000iiiiiiii	If R0 = imm, 1 → T	1	Comparison result
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	If Rn = Rm, 1 → T	1	Comparison result
CMP/HS	Rm,Rn	0011nnnnmmmm0010	If Rn=Rm with unsigned data, 1 → T	1	Comparison result
CMP/GE	Rm,Rn	0011nnnnmmmm0011	If Rn = Rm with signed data, 1 → T	1	Comparison result
CMP/HI	Rm,Rn	0011nnnnmmmm0110	If Rn > Rm with unsigned data, 1 → T	1	Comparison result
CMP/GT	Rm,Rn	0011nnnnmmmm0111	If Rn > Rm with signed data, 1 → T	1	Comparison result
CMP/PL	Rn	0100nnnn00010101	If Rn > 0, 1 → T	1	Comparison result
CMP/PZ	Rn	0100nnnn00010001	If Rn = 0, 1 → T	1	Comparison result
CMP/STR	Rm,Rn	0010nnnnmmmm1100	If Rn and Rm have an†equivalent byte, 1 → T	1	Comparison result
DIV1	Rm,Rn	0011nnnnmmmm0100	Single-step division (Rn ÷ Rm)	1	Calculation result
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calculation result
DIV0U		0000000000011001	0 → M/Q/T	1	0

Instruction		Instruction Code	Operation	tion Cycles	T Bit
DMULS.L	Rm,Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2 to 4*	—
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2 to 4*	—
DT	Rn	0100nnnn00010000	Rn – 1 → Rn, when Rn is 0, 1 → T. When Rn is nonzero, 0 → T	1	Comparison result
EXTS.B	Rm,Rn	0110nnnnmmmm1110	Byte in Rm is sign-extended → Rn	1	—
EXTS.W	Rm,Rn	0110nnnnmmmm1111	Word in Rm is sign-extended → Rn	1	—
EXTU.B	Rm,Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended → Rn	1	—
EXTU.W	Rm,Rn	0110nnnnmmmm1101	Word in Rm is zero-extended → Rn	1	—
MAC.L	@Rm+,@Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 32 × 32 + 64 → 64 bits	3/(2 to 4)*	—
MAC.W	@Rm+,@Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 16 × 16 + 64 → 64 bits	3/(2)*	—
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL, 32 × 32 → 32 bits	2 to 4*	—
MULS.W	Rm,Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	1 to 3*	—
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	1 to 3*	—
NEG	Rm,Rn	0110nnnnmmmm1011	0 – Rm → Rn	1	—
NEGC	Rm,Rn	0110nnnnmmmm1010	0 – Rm – T → Rn, Borrow → T	1	Borrow

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
SUB	Rm,Rn	0011nnnnmmmm1000	Rn – Rm → Rn	1	—
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn – Rm – T → Rn, Borrow → T	1	Borrow
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn – Rm → Rn, Underflow → T	1	Overflow

Note: * The normal minimum number of execution cycles. (The number in parentheses is the number of cycles when there is contention with following instructions.)

Table 2.14 Logic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
AND	Rm,Rn	0010nnnnmmmm1001	Rn & Rm → Rn	1	—
AND	#imm,R0	11001001iiiiiiii	R0 & imm → R0	1	—
AND.B	#imm,@(R0,GBR)	11001101iiiiiiii	(R0 + GBR) & imm → (R0 + GBR)	3	—
NOT	Rm,Rn	0110nnnnmmmm0111	~Rm → Rn	1	—
OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	1	—
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	1	—
OR.B	#imm,@(R0,GBR)	11001111iiiiiiii	(R0 + GBR) | imm → (R0 + GBR)	3	—
TAS.B	@Rn	0100nnnn00011011	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)	4	Test result
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm; if the result is 0, 1 → T	1	Test result
TST	#imm,R0	11001000iiiiiiii	R0 & imm; if the result is 0, 1 → T	1	Test result
TST.B	#imm,@(R0,GBR)	11001100iiiiiiii	(R0 + GBR) & imm; if the result is 0, 1 → T	3	Test result
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	1	—
XOR.B	#imm,@(R0,GBR)	11001110iiiiiiii	(R0 + GBR) ^ imm → (R0 + GBR)	3	—

Instruction		Instruction Code	Operation	tion Cycles	T Bit
ROTL	Rn	0100nnnn00000100	T ← Rn ← MSB	1	MSB
ROTR	Rn	0100nnnn00000101	LSB → Rn → T	1	LSB
ROTCL	Rn	0100nnnn00100100	T ← Rn ← T	1	MSB
ROTCR	Rn	0100nnnn00100101	T → Rn → T	1	LSB
SHAL	Rn	0100nnnn00100000	T ← Rn ← 0	1	MSB
SHAR	Rn	0100nnnn00100001	MSB → Rn → T	1	LSB
SHLL	Rn	0100nnnn00000000	T ← Rn ← 0	1	MSB
SHLR	Rn	0100nnnn00000001	0 → Rn → T	1	LSB
SHLL2	Rn	0100nnnn00001000	Rn<<2 → Rn	1	—
SHLR2	Rn	0100nnnn00001001	Rn>>2 → Rn	1	—
SHLL8	Rn	0100nnnn00011000	Rn<<8 → Rn	1	—
SHLR8	Rn	0100nnnn00011001	Rn>>8 → Rn	1	—
SHLL16	Rn	0100nnnn00101000	Rn<<16 → Rn	1	—
SHLR16	Rn	0100nnnn00101001	Rn>>16 → Rn	1	—

Instruction		Instruction Code	Operation	tion Cycles	T Bit
BF	label	10001011dddddddd	If T = 0, disp × 2 + PC → PC; if T = 1, nop	3/1*	—
BF/S	label	10001111dddddddd	Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop	2/1*	—
BT	label	10001001dddddddd	If T = 1, disp × 2 + PC → PC; if T = 0, nop	3/1*	—
BT/S	label	10001101dddddddd	Delayed branch, if T = 1, disp × 2 + PC → PC; if T = 0, nop	2/1*	—
BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2	—
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2	—
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	2	—
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm†+†PC → PC	2	—
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	2	—
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	2	—
RTS		0000000000001011	Delayed branch, PR → PC	2	—

Note: * One state when the program does not branch.

Instruction		Instruction Code	Operation	tion Cycles	T Bit
CLRT		0000000000001000	0 → T	1	0
CLRMAC		0000000000101000	0 → MACH, MACL	1	—
LDC	Rm,SR	0100mmmm00001110	Rm → SR	1	LSB
LDC	Rm,GBR	0100mmmm00011110	Rm → GBR	1	—
LDC	Rm,VBR	0100mmmm00101110	Rm → VBR	1	—
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	3	LSB
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	3	—
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	3	—
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—
NOP		0000000000001001	No operation	1	—
RTE		0000000000101011	Delayed branch, stack area → PC/SR	4	—
SETT		0000000000011000	1 → T	1	1
SLEEP		0000000000011011	Sleep	3*	—
STC	SR,Rn	0000nnnn00000010	SR → Rn	1	—
STC	GBR,Rn	0000nnnn00010010	GBR → Rn	1	—
STC	VBR,Rn	0000nnnn00100010	VBR → Rn	1	—
STC.L	SR,@–Rn	0100nnnn00000011	Rn – 4 → Rn, SR → (Rn)	2	—
STC.L	GBR,@–Rn	0100nnnn00010011	Rn – 4 → Rn, GBR → (Rn)	2	—
STC.L	VBR,@–Rn	0100nnnn00100011	Rn – 4 → Rn, BR → (Rn)	2	—
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1	—
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—

Instruction		Instruction Code	Operation	tion Cycles	T Bit
STS.L	MACH,@–Rn	0100nnnn00000010	Rn – 4 → Rn, MACH → (Rn)	1	—
STS.L	MACL,@–Rn	0100nnnn00010010	Rn – 4 → Rn, MACL → (Rn)	1	—
STS.L	PR,@–Rn	0100nnnn00100010	Rn – 4 → Rn, PR → (Rn)	1	—
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	8	—

Note: * The number of execution cycles before the chip enters sleep mode: The execution cycles shown in the table are minimums. The actual number of cycles may be increased when (1) contention occurs between instruction fetches and data access, or (2) when the destination register of the load instruction (memory → register) and the register used by the next instruction are the same.

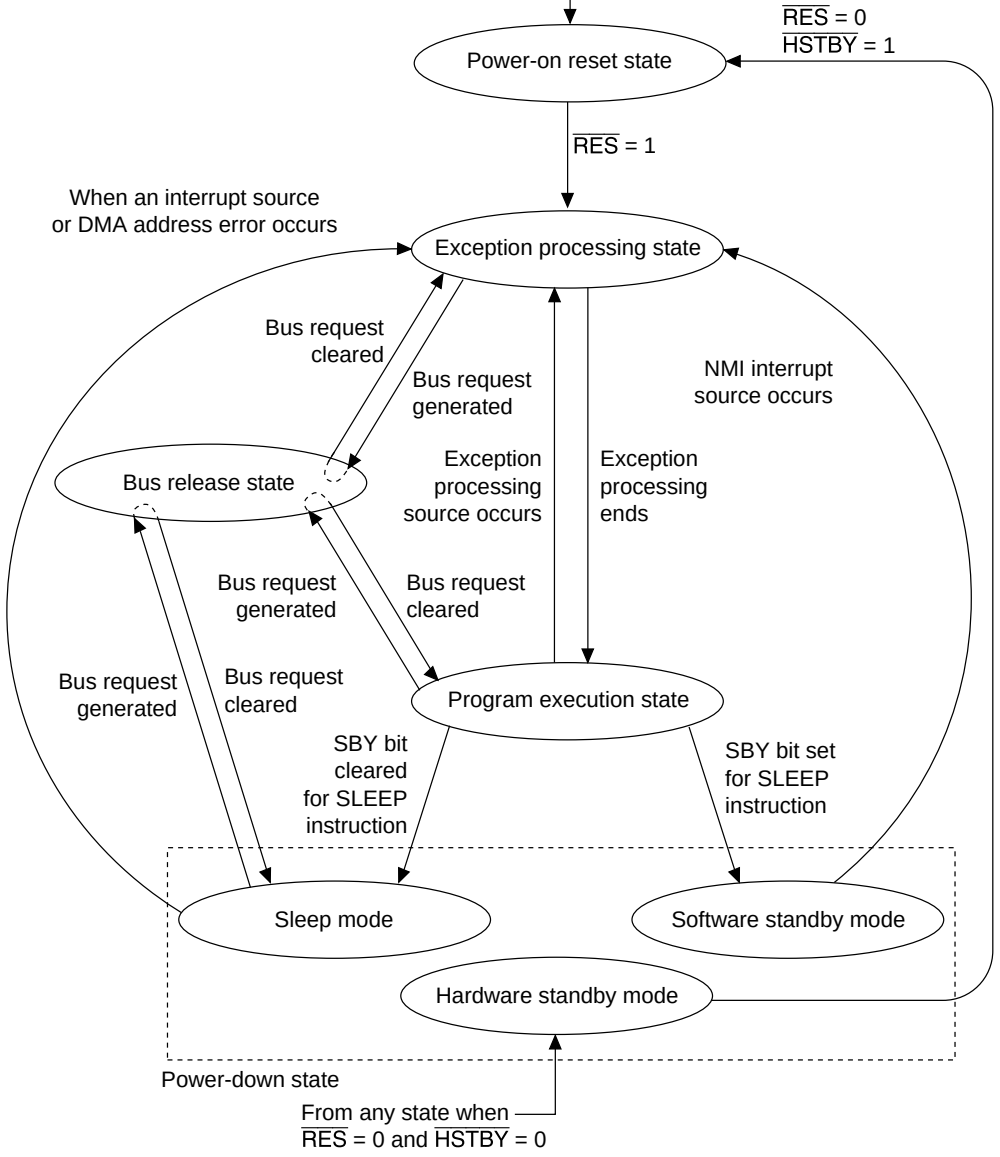
Instruction		Instruction Code	Operation	tion Cycles	T Bit
FABS	FRn	1111nnnn01011101	|FRn| → FRn	1	—
FADD	FRm,FRn	1111nnnnmmmm0000	FRn + FRm → FRn	1	—
FCMP/EQ	FRm,FRn	1111nnnnmmmm0100	(FRn = FRm)? 1:0 → T	1	Comparison result
FCMP/GT	FRm,FRn	1111nnnnmmmm0101	(FRn > FRm)? 1:0 → T	1	Comparison result
FDIV	FRm,FRn	1111nnnnmmmm0011	FRn/FRm → FRn	13	—
FLDI0	FRn	1111nnnn10001101	0x00000000 → FRn	1	—
FLDI1	FRn	1111nnnn10011101	0x3F800000 → FRn	1	—
FLDS	FRm,FPUL	1111mmmm00011101	FRm → FPUL	1	—
FLOAT	FPUL,FRn	1111nnnn00101101	(float) FPUL → FRn	1	—
FMAC	FR0,FRm,FRn	1111nnnnmmmm1110	FR0 × FRm + FRn → FRn	1	—
FMOV	FRm, FRn	1111nnnnmmmm1100	FRm → FRn	1	—
FMOV.S	@(R0,Rm),FRn	1111nnnnmmmm0110	(R0 + Rm) → FRn	1	—
FMOV.S	@Rm+,FRn	1111nnnnmmmm1001	(Rm) → FRn, Rm+ = 4	1	—
FMOV.S	@Rm,FRn	1111nnnnmmmm1000	(Rm) → FRn	1	—
FMOV.S	FRm,@(R0,Rn)	1111nnnnmmmm0111	FRm → (R0 + Rn)	1	—
FMOV.S	FRm,@-Rn	1111nnnnmmmm1011	Rn– = 4, FRm → (Rn)	1	—
FMOV.S	FRm,@Rn	1111nnnnmmmm1010	FRm → (Rn)	1	—
FMUL	FRm,FRn	1111nnnnmmmm0010	FRn × FRm → FRn	1	—
FNEG	FRn	1111nnnn01001101	–FRn → FRn	1	—
FSTS	FPUL,FRn	1111nnnn00001101	FPUL → FRn	1	—
FSUB	FRm,FRn	1111nnnnmmmm0001	FRn – FRm → FRn	1	—
FTRC	FRm,FPUL	1111mmmm00111101	(long) FRm → FPUL	1	—

Instruction		Instruction Code	Operation	tion Cycles	T Bit
LDS	Rm,FPSCR	0100mmmm01101010	Rm → FPSCR	1	—
LDS	Rm,FPUL	0100mmmm01011010	Rm → FPUL	1	—
LDS.L	@Rm+, FPSCR	0100mmmm01100110	@Rm → FPSCR, Rm+ = 4	1	—
LDS.L	@Rm+, FPUL	0100mmmm01010110	@Rm → FPUL, Rm+ = 4	1	—
STS	FPSCR, Rn	0000nnnn01101010	FPSCR → Rn	1	—
STS	FPUL,Rn	0000nnnn01011010	FPUL → Rn	1	—
STS.L	FPSCR,@-Rn	0100nnnn01100010	Rn– = 4, FPCSR → @Rn	1	—
STS.L	FPUL,@-Rn	0100nnnn01010010	Rn– = 4, FPUL → @Rn	1	—

2.5 Processing States

2.5.1 State Transitions

The CPU has five processing states: power-on reset, exception processing, bus release, program execution and power-down. Figure 2.8 shows the transitions between the states.



Note: An internal reset due to the WDT causes a transition from the program execution state or sleep mode to the exception processing state.

Figure 2.8 Transitions between Processing States

Exception Processing State: The exception processing state is a transient state that occurs when exception processing sources such as resets or interrupts alter the CPU's processing state flow.

For a reset, the initial values of the program counter (PC) (execution start address) and stack pointer (SP) are fetched from the exception processing vector table and stored; the CPU then branches to the execution start address and execution of the program begins.

For an interrupt, the stack pointer (SP) is accessed and the program counter (PC) and status register (SR) are saved to the stack area. The exception service routine start address is fetched from the exception processing vector table; the CPU then branches to that address and the program starts executing, thereby entering the program execution state.

Program Execution State: In the program execution state, the CPU sequentially executes the program.

Power-Down State: In the power-down state, the CPU operation halts and power consumption declines. The SLEEP instruction places the CPU in the sleep mode or the software standby mode. If the $\overline{\text{HSTBY}}$ pin is driven low when the $\overline{\text{RES}}$ pin is low, the CPU will enter the hardware standby mode.

Bus Release State: In the bus release state, the CPU releases access rights to the bus to the device that has requested them.

3.1 Overview

The SH7055SF has an on-chip floating-point unit (FPU), The FPU's register configuration is shown in figure 3.1.

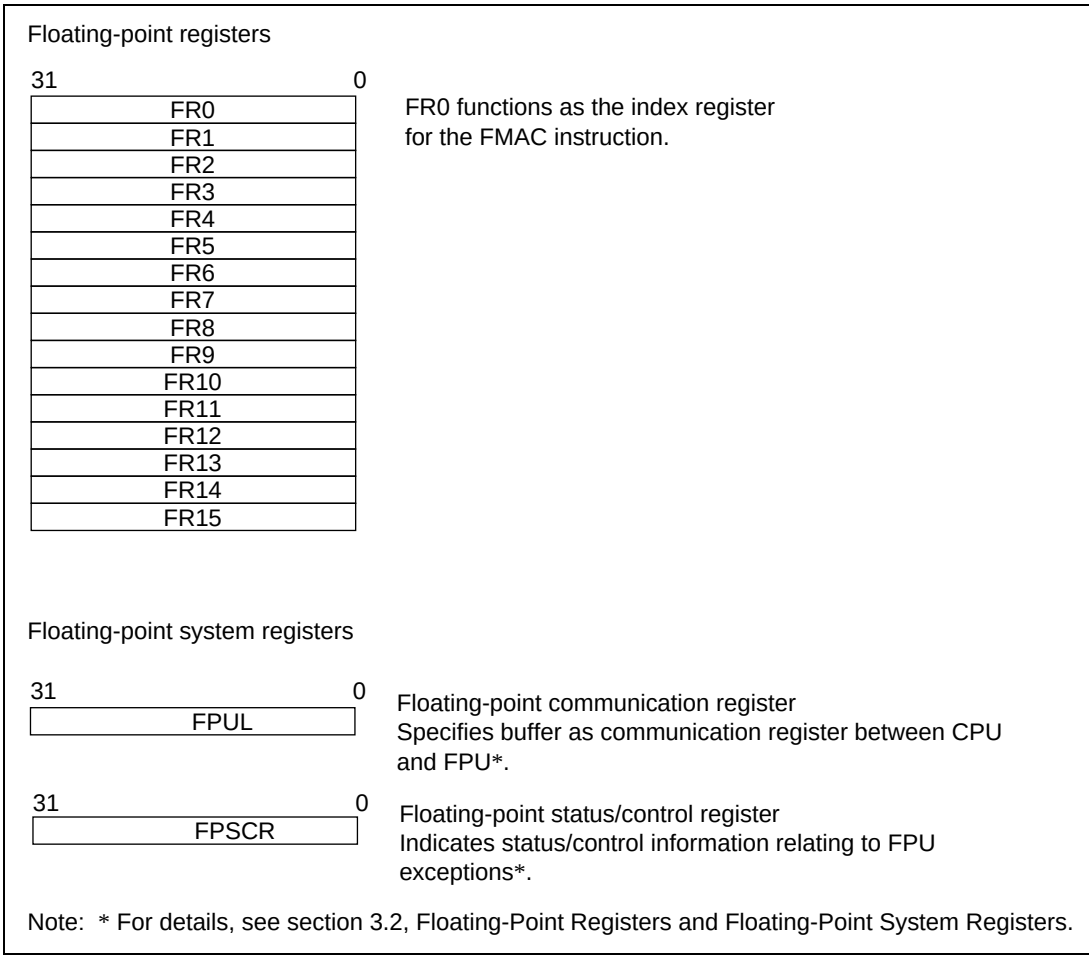


Figure 3.1 Overview of Register Configuration (Floating-Point Registers and Floating-Point System Registers)

3.2.1 Floating-Point Register File

The SH7055SF has sixteen 32-bit single-precision floating-point registers. Register specifications are always made as 4 bits. In assembly language, the floating-point registers are specified as FR0, FR1, FR2, and so on. FR0 functions as the index register for the FMAC instruction.

3.2.2 Floating-Point Communication Register (FPUL)

Information for transfer between the FPU and the CPU is transferred via the FPUL communication register, which resembles MACL and MACH in the integer unit. The SH7055SF is provided with this communication register since the integer and floating-point formats are different. The 32-bit FPUL is a system register, and is accessed by the CPU by means of LDS and STS instructions.

3.2.3 Floating-Point Status/Control Register (FPSCR)

The SH7055SF has a floating-point status/control register (FPSCR) that functions as a system register accessed by means of LDS and STS instructions (figure 3.2). FPSCR can be written to by a user program. This register is part of the process context, and must be saved when the context is switched. It may also be necessary to save this register when a procedure call is made.

FPSCR is a 32-bit register that controls the storage of detailed information relating to the rounding mode, asymptotic underflow (denormalized numbers), and FPU exceptions. The module stop bit that disables the FPU itself is provided in the module standby control register (MSTCR). For details, see section 24, Power-Down State. After a reset start, the FPU is enabled.

Table 3.1 shows the flags corresponding the five kinds of FPU exception. A sixth flag is also provided as an FPU error flag that indicates an floating-point unit error state not covered by the other five flags.

Table 3.1 Floating-Point Exception Flags

Flag	Meaning	Support in SH7055SF
E	FPU error	—
V	Invalid operation	Yes
Z	Division by zero	Yes
O	Overflow (value not expressed)	—
U	Underflow (value not expressed)	—
I	Inexact (result not expressed)	—

according to whether or not an exception state occurred during execution of a single instruction.

The bits in the enable field specify the kinds of exception to be enabled, allowing the flow to be changed to exception processing. If the cause bit corresponding to an enable bit is set by the currently executing instruction, an exception occurs.

The bits in the flag field are used to keep a tally of all exceptions that occur during a series of instructions. Once one of these bits is set by an instruction, it is not reset by a subsequent instruction. The bits in this field can only be reset by the explicit execution of a store operation on FPSCR.

Reserved	DN	CE	CV	CZ	CO	CU	CI	EV	EZ	EO	EU	EI	FV	FZ	FO	FU	FI	RM

DN: Denormalized bit
In the SH7055SF this bit is always set to 1, and the source or destination operand of a denormalized number is 0. This bit cannot be modified even by an LDS instruction.

CV: Invalid operation cause bit
When 1: Indicates that an invalid operation exception occurred during execution of the current instruction.
When 0: Indicates that an invalid operation exception has not occurred.

CZ: Division-by-zero cause bit
When 1: Indicates that a division-by-zero exception occurred during execution of the current instruction.
When 0: Indicates that a division-by-zero exception has not occurred.

EV: Invalid operation exception enable
When 1: Enables invalid operation exception generation.
When 0: An invalid operation exception is not generated, and a qNAN is returned as the result.

EZ: Division-by-zero exception enable
When 1: Enables exception generation due to division-by-zero during execution of the current instruction.
When 0: A division-by-zero exception is not generated, and infinity with the sign (+ or –) of the current expression is returned as the result.

FV: Invalid operation exception flag bit
When 1: Indicates that an invalid operation exception occurred during instruction execution.
When 0: Indicates that an invalid operation exception has not occurred.

FZ: Division-by-zero exception flag bit
When 1: Indicates that a division-by-zero exception occurred during instruction execution.
When 0: Indicates that a division-by-zero exception has not occurred.

RM: Rounding bit.
In the SH7055SF, the value of these bits is always 01, meaning that rounding to zero (RZ mode) is being used. These bits cannot be modified even by an LDS instruction.

In the SH7055SF, the cause field EOUI bits (CE, CO, CU, and CI), enable field OUI bits (EO, EU, and EI), and flag field OUI bits (FO, FU, and FI), and the reserved area, are preset to 0, and cannot be modified even by using an LDS instruction.

Figure 3.2 Floating-Point Status/Control Register

3.3.1 Floating-Point Format

The SH7055SF supports single-precision floating-point operations, and fully complies with the IEEE754 floating-point standard.

A floating-point number consists of the following three fields:

- Sign (s)
- Exponent (e)
- Fraction (f)

The exponent is expressed in biased form, as follows:

$$e = E + bias$$

The range of unbiased exponent E is $E_{min} - 1$ to $E_{max} + 1$. The two values $E_{min} - 1$ and $E_{max} + 1$ are distinguished as follows. $E_{min} - 1$ indicates zero (both positive and negative sign) and a denormalized number, and $E_{max} + 1$ indicates positive or negative infinity or a non-number (NaN). In a single-precision operation, the bias value is 127, E_{min} is –126, and E_{max} is 127.

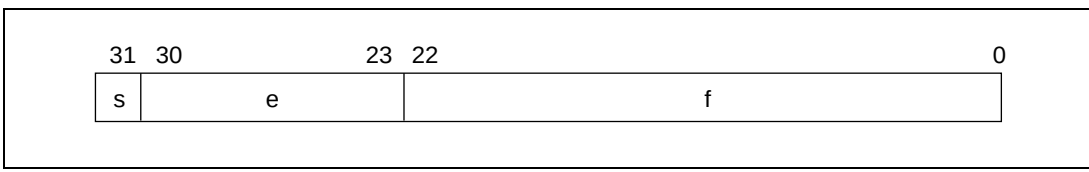


Figure 3.3 Floating-Point Number Format

Floating-point number value v is determined as follows:

If $E = E_{max} + 1$ and f! = 0, v is a non-number (NaN) irrespective of sign s
If $E = E_{max} + 1$ and f = 0, $v = (-1)^s$ (infinity) [positive or negative infinity]
If $E_{min} <= E <= E_{max}$, $v = (-1)^s 2^E$ (1.f) [normalized number]
If $E = E_{min} - 1$ and f! = 0, $v = (-1)^s 2^{Emin}$ (0.f) [denormalized number]
If $E = E_{min} - 1$ and f = 0, $v = (-1)^s 0$ [positive or negative zero]

With non-number (NaN) representation in a single-precision operation value, at least one of bits 22 to 0 is set. If bit 22 is set, this indicates a signaling NaN (sNaN). If bit 22 is reset, the value is a quiet NaN (qNaN).

The bit pattern of a non-number (NaN) is shown in the figure below. Bit N in the figure is set for a signaling NaN and reset for a quiet NaN. x indicates a don't care bit (with the proviso that at least one of bits 22 to 0 is set). In a non-number (NaN), the sign bit is a don't care bit.

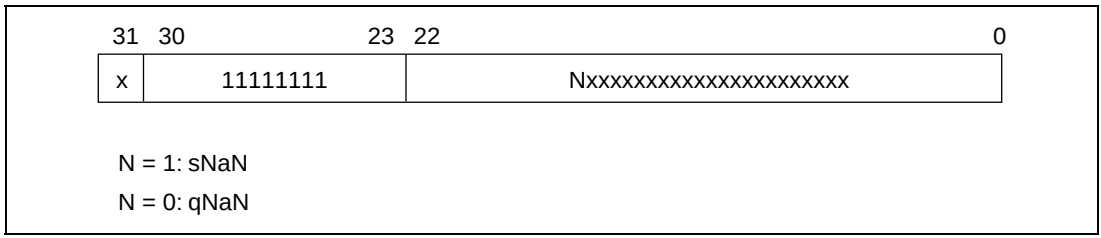


Figure 3.4 NaN Bit Pattern

If a non-number (sNaN) is input in an operation that generates a floating-point value:

- When the EV bit in the FPSCR register is reset, the operation result (output) is a quiet NaN (qNaN).
- When the EV bit in the FPSCR register is set, an invalid operation exception will be generated. In this case, the contents of the operation destination register do not change.

If a quiet NaN is input in an operation that generates a floating-point value, and a signaling NaN has not been input in that operation, the output will always be a quiet NaN irrespective of the setting of the EV bit in the FPSCR register. An exception will not be generated in this case.

Refer to the SH-2E Programming Manual for details of floating-point operations when a non-number (NaN) is input.

3.3.3 Denormalized Number Values

For a denormalized number floating-point value, the biased exponent is expressed as 0, the fraction as a non-zero value, and the hidden bit as 0. In the SH7055SF's floating-point unit, a denormalized number (operand source or operation result) is always flushed to 0 in a floating-point operation that generates a value (an operation other than copy).

Floating-point value representations include the seven different kinds of special values shown in table 3.2.

Table 3.2 Representation of Special Values in Single-Precision Floating-Point Operations Specified by IEEE754 Standard

Value	Representation
+0.0	0x00000000
–0.0	0x80000000
Denormalized number	As described in 3.3.3, Denormalized Number Values
+INF	0x7F800000
–INF	0xFF800000
qNaN (quiet NaN)	As described in 3.3.2, Non-Numbers (NaN)
sNaN (signaling NaN)	As described in 3.3.2, Non-Numbers (NaN)

3.4.1 Enable State Exceptions

Invalid operation and division-by-zero exceptions are both placed in the enable state by setting the enable bit. All exceptions generated by the FPU are mapped as the same exception event. The meaning of a particular exception is determined by software by reading system register FPSCR and analyzing the information held there.

3.4.2 Disable State Exceptions

If the EV enable bit is not set, a qNaN will be generated as the result of an invalid operation (except for FCMP and FTRC). If the EZ enable bit is not set, division-by-zero will return infinity with the sign (+ or –) of the current expression. Overflow will generate a finite number which is the largest value that can be expressed by an absolute value in the format, with the correct sign. Underflow will generate zero with the correct sign. If the operation result is inexact, the destination register will store that inexact result.

3.4.3 FPU Exception Event and Code

All FPU exceptions have a vector table address offset in address H'00000034 as the same general exception event; that is, an FPU exception.

3.4.4 Floating-Point Data Arrangement in Memory

Single-precision floating-point data is located in memory at a 4-byte boundary; that is, it is arranged in the same form as an SH7055SF long integer.

3.4.5 Arithmetic Operations Involving Special Operands

All arithmetic operations involving special operands (qNaN, sNaN, +INF, –INF, +0, –0) comply with the specifications of the IEEE754 standard. Refer to the SH-2E Programming Manual for details.

Synchronization with CPU: Floating-point instructions and CPU instructions are executed in turn, according to their order in the program, but in some cases operations may not be completed in the program order due to a difference in execution cycles. When a floating-point instruction accesses only FPU resources, there is no need for synchronization with the CPU, and a CPU instruction following an FPU instruction can finish its operation before completion of the FPU operation. Consequently, in an optimized program, it is possible to effectively conceal the execution cycle of a floating-point instruction that requires a long execution cycle, such as a divide instruction. On the other hand, a floating-point instruction that accesses CPU resources, such as a compare instruction, must be synchronized to ensure that the program order is observed.

Floating-Point Instructions That Require Synchronization: Load, store, and compare instructions, and instructions that access the FPUL or FPSCR register, must be synchronized because they access CPU resources. Load and store instructions access a general register. Post-increment load and pre-decrement store instructions change the contents of a general register. A compare instruction modifies the T bit. An FPUL or FPSCR access instruction references or changes the contents of the FPUL or FPSCR register. These references and changes must all be synchronized with the CPU.

3.6 Usage Notes

1. When using the FPU (using FPU instructions or FPU-related CPU instructions)
   a. Limitations on using the BT and BF instructions on the SH7055F are abolished. The BT and BF instructions can be used on the SH7055SF.
   b. The branch destination of TRAP instruction and interrupt/exception handling must be located at a 4n address. In this case, do not place an FPU instruction or FPU-related CPU instruction at address 4n or 4n+2.
2. When not using the FPU (not using FPU instructions or FPU-related CPU instructions)
   After a power-on reset, the FPU should be placed in the module standby state until a DMAC or AUD bus cycle is generated.
   Specifically, write 1 to bit 1 in the module standby control register.
   This operation is also effective in reducing current dissipation.
   When the FPU enters the module standby state, any subsequent FPU instruction or FPU-related CPU instruction will be subjected to exception handling as an illegal instruction.
3. Restrictions of the FADD and FSUB instructions
   In this FPU, values calculated by the following two arithmetic operations with a special operand have a sign which is different from values' expected in the IEEE Standard 754.
   1) FADD FRm, FRn FRm = –INF(0xFF80000)
      FRn = MAX(0x7F7FFFFF)

2) FSUB FRm, FRn FRm = –INF(0xFF80000)
FRn = MAX(0x7F7FFFFF)

At this time, + INF (0x7F800000) is generated as a result to the expected value –INF (0xFF800000) in the IEEE754.

4.1 Operating Mode Selection

The SH7055SF has five operating modes that are selected by pins MD2 to MD0 and FWE. The mode setting pins should not be changed during operation of the SH7055SF, and only the setting combinations shown in table 4.1 should be used.

The $PV_{CC}1$ power supply voltage must be within the range shown in table 4.1.

Table 4.1 Operating Mode Selection

Operating Mode No.	Pin Settings				Mode Name	On-Chip ROM	Area 0 Bus Width	$PV_{CC}1$ Voltage
	FWE	MD2	MD1	MD0				
Mode 0	0	1	0	0	MCU expanded mode	Disabled	8 bits	3.3 V ±0.3 V
Mode 1	0	1	0	1			16 bits	
Mode 2	0	1	1	0		Enabled	Set by BCR1	
Mode 3	0	1	1	1	MCU single-chip mode	Enabled	—	5.0 V ±0.5 V
Mode 4	1	1	0	0	Boot mode	Enabled	Set by BCR1	3.3 V ±0.3 V
Mode 5	1	1	0	1			—	5.0 V ±0.5 V
Mode 6	1	1	1	0	User program mode	Enabled	Set by BCR1	3.3 V ±0.3 V
Mode 7	1	1	1	1			—	5.0 V ±0.5 V
Mode 8	1	0	0	1	User boot mode	Enabled	Set by BCR1	3.3 V ±0.3 V
Mode 9	1	0	0	1			—	5.0 V ±0.5 V
—	0/1	0	1	1	Programmer mode	—	—	3.3 V ±0.3 V

There are two MCU operating modes: MCU single-chip mode and MCU expanded mode.

Modes in which the flash memory can be programmed are boot mode, user boot mode and user program mode (the two on-board programming modes) and programmer mode in which programming is performed with an EPROM programmer (a type which supports programming of this device).

For details, see section 22, ROM.

5.1 Overview

The clock pulse generator (CPG) supplies clock pulses inside the SH7055SF chip and to external devices. The SH7055SF CPG consists of an oscillator circuit and a PLL multiplier circuit. There are two methods of generating a clock with the CPG: by connecting a crystal resonator, or by inputting an external clock. The oscillator circuit oscillates at the same frequency as the input clock. A chip operating frequency of 4 times the oscillator frequency is generated by the PLL multiplier circuit.

The CPG is halted in software standby mode and hardware standby mode.

5.1.1 Block Diagram

A block diagram of the clock pulse generator is shown in figure 5.1.

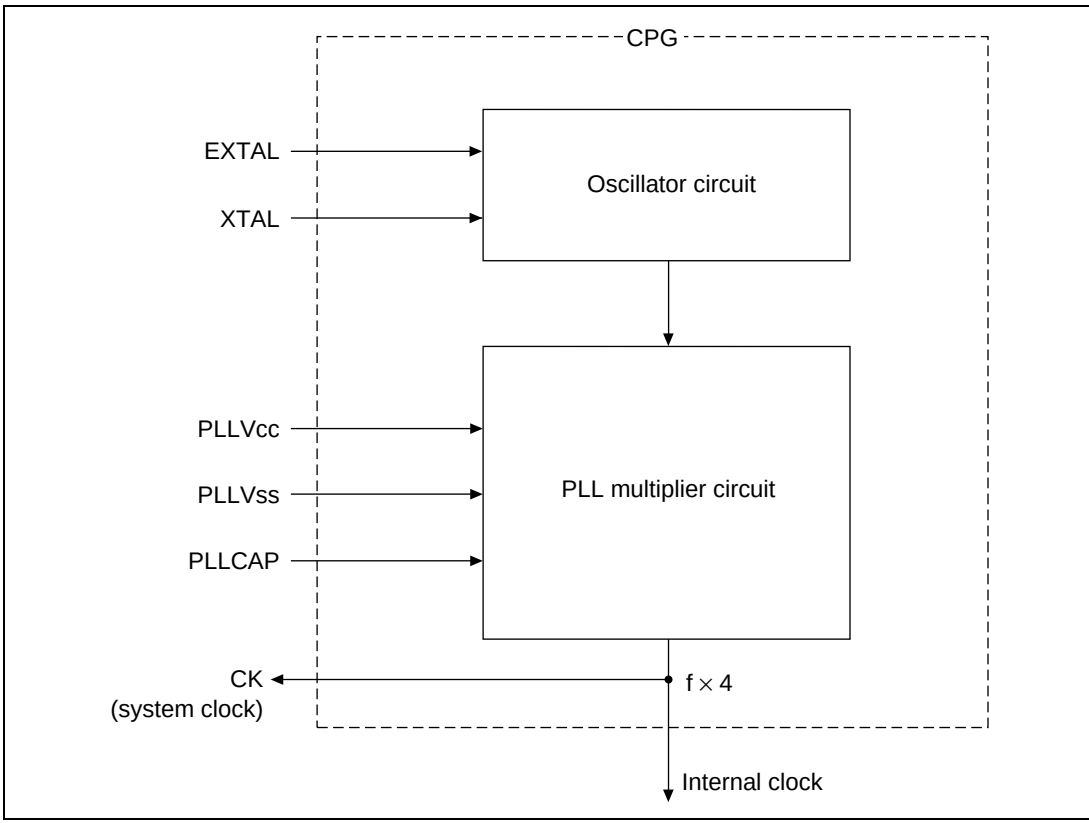


Figure 5.1 Block Diagram of Clock Pulse Generator

The pins relating to the clock pulse generator are shown in table 5.1.

Table 5.1 CPG Pins

Pin Name	Abbreviation	I/O	Description
External clock	EXTAL	Input	Crystal resonator or external clock input
Crystal	XTAL	Input	Crystal resonator connection
System clock	CK	Output	System clock output
PLL power supply	$PLLV_{CC}$	Input	PLL multiplier circuit power supply
PLL ground	$PLLV_{SS}$	Input	PLL multiplier circuit ground
PLL capacitance	PLLCAP	Input	PLL multiplier circuit oscillation external capacitance pin

5.2 Frequency Ranges

The input frequency and operating frequency ranges are shown in table 5.2.

Table 5.2 Input Frequency and Operating Frequency

Input Frequency Range (MHz)	PLL Multiplication Factor	Operating Frequency Range (MHz)
5–10	×4	20–40

Note: Crystal resonator and external clock input

For the chip operating frequency, a frequency of 4 times the input frequency (EXTAL pin) is generated as the internal clock (ϕ) by the on-chip PLL circuit. The system clock (CK pin) output frequency is the same as that of the internal clock (ϕ).

Some on-chip peripheral modules operate on a peripheral clock (Pϕ) obtained by dividing the internal clock (ϕ) by 2. Figure 5.2 shows the relationship between the various clocks. As regards the system clock, since the input clock is multiplied by the PLL multiplier circuit, the phases of both clocks are not determined uniformly.

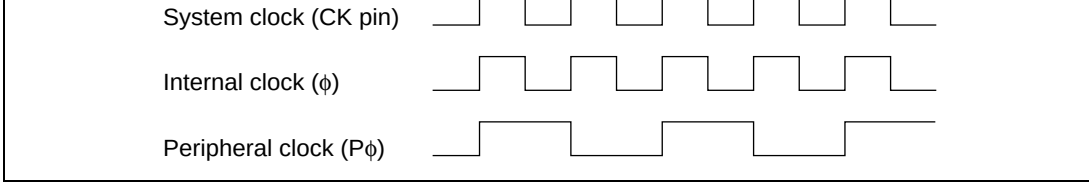


Figure 5.2 Input Clock and System Clock

5.3 Clock Source

Clock pulses can be supplied from a connected crystal resonator or an external clock.

5.3.1 Connecting a Crystal Oscillator

Circuit Configuration: Figure 5.3 shows an example of connecting a crystal resonator. Use the damping resistance (Rd) shown in table 5.3. An AT-cut parallel-resonance type crystal resonator should be used. Load capacitors (CL1, CL2) must be connected as shown in the figure.

The clock pulses generated by the crystal resonator and internal oscillator are sent to the PLL multiplier circuit, where a multiplied frequency is selected and supplied inside the SH7055SF chip and to external devices.

The crystal oscillator manufacturer should be consulted concerning the compatibility between the crystal oscillator and the chip.

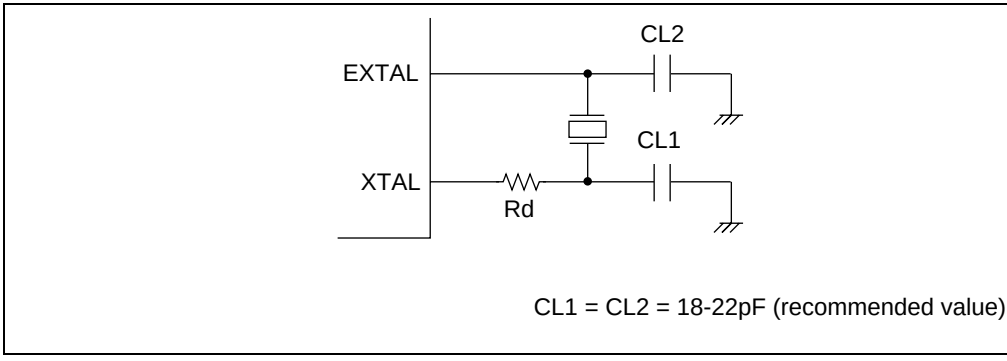


Figure 5.3 Connection of Crystal Oscillator (Example)

Parameter	5	10
Rd (Ω)	500	0

Crystal Oscillator: Figure 5.4 shows an equivalent circuit of the crystal oscillator. Use a crystal oscillator with the characteristics listed in table 5.4.

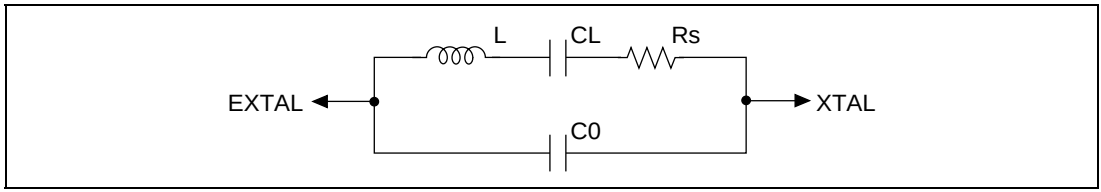


Figure 5.4 Crystal Oscillator Equivalent Circuit

Table 5.4 Crystal Oscillator Parameters (Recommended Values)

	Frequency (MHz)	
Parameter	**5**	**10**
Rs max (Ω)	100	50
C0 max (pF)	7	7

The crystal oscillator manufacturer should be consulted concerning the compatibility between the crystal oscillator and the chip.

5.3.2 External Clock Input Method

An example of external clock input connection is shown in figure 5.5.

When the XTAL pin is placed in the open state, the parasitic capacitance should be 10 pF or less.

Even when an external clock is input, provide for a wait of at least the oscillation settling time when powering on or exiting standby mode in order to secure the PLL settling time.

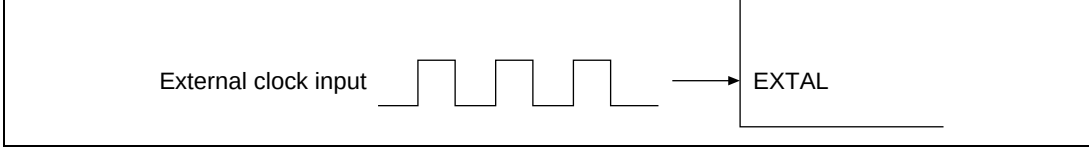


Figure 5.5 External Clock Input Method (Example)

5.4 Usage Notes

Notes on Board Design: When connecting a crystal oscillator, observe the following precautions:

- To prevent induction from interfering with correct oscillation, do not route any signal lines near the oscillator circuitry (figure 5.6).
- When designing the board, place the crystal oscillator and its load capacitors as close as possible to the XTAL and EXTAL pins.

Figure 5.6 shows the precautions regarding oscillator circuit system board design.

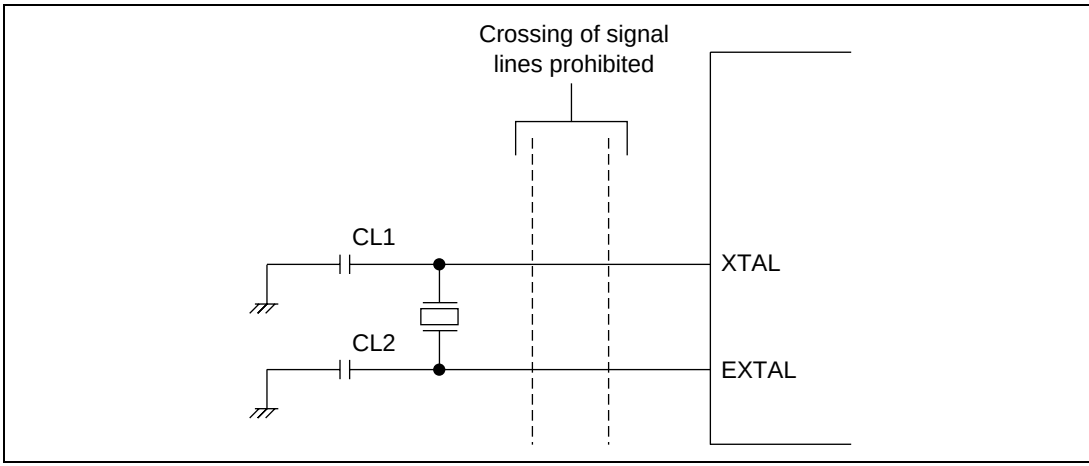


Figure 5.6 Precautions for Oscillator Circuit System Board Design

PLL Oscillation Power Supply: Separate $PLLV_{CC}$ and $PLLV_{SS}$ from the other V_{CC} and V_{SS} lines at the board power supply source, and be sure to insert bypass capacitors C_{PB} and C_B close to the pins.

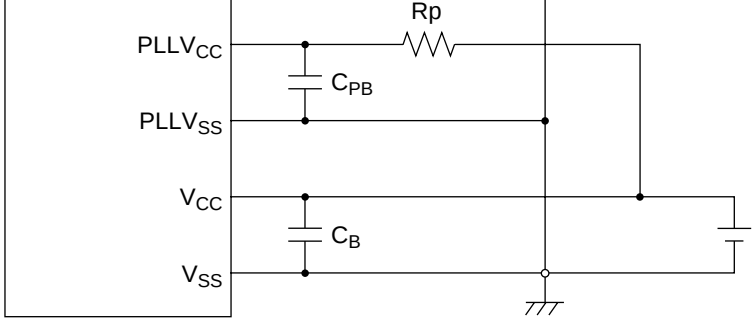


Figure 5.7 Points for Caution in PLL Power Supply Connection

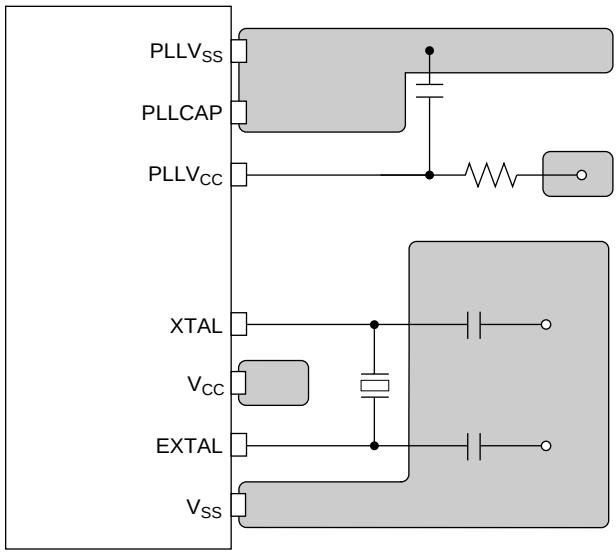


Figure 5.8 Actual Example of Board Design

6.1 Overview

6.1.1 Types of Exception Processing and Priority

Exception processing is started by four sources: resets, address errors, interrupts and instructions and have the priority shown in table 6.1. When several exception processing sources occur at once, they are processed according to the priority shown.

Table 6.1 Types of Exception Processing and Priority Order

Exception	Source		Priority
Reset	Power-on reset		High
	Manual reset		
Address error	CPU address error		
	DMAC address error		
Instructions	FPU exception		
Interrupt	NMI		
	User break		
	H-UDI		
	IRQ		
	On-chip peripheral modules:	• Direct memory access controller (DMAC) • Advanced timer unit-II (ATU-II) • Compare match timer 0 (CMT0) • A/D converter channel 0 (A/D0) • Compare match timer 1 (CMT1) • A/D converter channel 1 (A/D1) • A/D converter channel 2 (A/D2) • Serial communication interface (SCI) • Controller area network 0 (HCAN0) • Watchdog timer (WDT) • Controller area network 1 (HCAN 1)	Low

Instructions	Trap instruction (TRAPA instruction)	High
	General illegal instructions (undefined code)	↕
	Illegal slot instructions (undefined code placed directly after a delay branch instruction*1 or instructions that rewrite the PC*2)	Low

Notes: *1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF.

*2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF.

6.1.2 Exception Processing Operations

The exception processing sources are detected and begin processing according to the timing shown in table 6.2.

Table 6.2 Timing of Exception Source Detection and Start of Exception Processing

Exception	Source	Timing of Source Detection and Start of Processing
Reset	Power-on reset	Starts when the $\overline{\text{RES}}$ pin changes from low to high or when the WDT overflows.
	Manual reset	Starts when the WDT overflows.
Address error		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Interrupts		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code anytime except after a delayed branch instruction (delay slot).
	Illegal slot instructions	Starts from the decoding of undefined code placed in a delayed branch instruction (delay slot) or of instructions that rewrite the PC.
	Floating point instructions	Starts when a floating-point instruction causes an invalid operation exception (IEEE754 specification) or division-by-zero exception.

The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception processing vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses for power-on resets and the H'00000008 and H'0000000C addresses for manual resets). See section 6.1.3, Exception Processing Vector Table, for more information. H'00000000 is then written to the vector base register (VBR) and H'F (1111) is written to the interrupt mask bits (I3–I0) of the status register (SR). The program begins running from the PC address fetched from the exception processing vector table.

2. Exception processing triggered by address errors, interrupts and instructions:

   SR and PC are saved to the stack indicated by R15. For interrupt exception processing, the interrupt priority level is written to the SR's interrupt mask bits (I3–I0). For address error and instruction exception processing, the I3–I0 bits are not affected. The start address is then fetched from the exception processing vector table and the program begins running from that address.

6.1.3 Exception Processing Vector Table

Before exception processing begins running, the exception processing vector table must be set in memory. The exception processing vector table stores the start addresses of exception service routines. (The reset exception processing table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception processing, the start addresses of the exception service routines are fetched from the exception processing vector table, which is indicated by this vector table address.

Table 6.3 shows the vector numbers and vector table address offsets. Table 6.4 shows how vector table addresses are calculated.

Table 6.3 Exception Processing Vector Table

Exception Sources		Vector Numbers	Vector Table Address†Offset
Power-on reset	PC	0	H'00000000–H'00000003
	SP	1	H'00000004–H'00000007
Manual reset	PC	2	H'00000008–H'0000000B
	SP	3	H'0000000C–H'0000000F
General illegal instruction		4	H'00000010–H'00000013
(Reserved by system)		5	H'00000014–H'00000017

Exception Sources		Numbers	Vector Table Address†Offset
Slot illegal instruction		6	H'00000018–H'0000001B
(Reserved by system)		7	H'0000001C–H'0000001F
		8	H'00000020–H'00000023
CPU address error		9	H'00000024–H'00000027
DMAC address error		10	H'00000028–H'0000002B
Interrupts	NMI	11	H'0000002C–H'0000002F
	User break	12	H'00000030–H'00000033
FPU exception		13	H'00000034–H'00000037
H-UDI		14	H'00000038–H'0000003B
(Reserved by system)		16	H'0000003C–H'00000043
		:	:
		31	H'0000007C–H'0000007F
Trap instruction (user vector)		32	H'00000080–H'00000083
		:	:
		63	H'000000FC–H'000000FF
Interrupts	IRQ0	64	H'00000100–H'00000103
	IRQ1	65	H'00000104–H'00000107
	IRQ2	66	H'00000108–H'0000010B
	IRQ3	67	H'0000010C–H'0000010F
	IRQ4	68	H'00000110–H'00000113
	IRQ5	69	H'00000114–H'00000117
	IRQ6	70	H'00000118–H'0000011B
	IRQ7	71	H'0000011C–H'0000011F
On-chip peripheral module*		72	H'00000120–H'00000124
		:	:
		255	H'000003FC–H'000003FF

Note: * The vector numbers and vector table address offsets for each on-chip peripheral module interrupt are given in table 7.3, Interrupt Exception Processing Vectors and Priorities, in section 7, Interrupt Controller (INTC).

Resets	Vector table address	= (vector table address offset) = (vector number) × 4
Address errors, interrupts, instructions	Vector table address	= VBR + (vector table address offset) = VBR + (vector number) × 4

Notes: 1. VBR: Vector base register
2. Vector table address offset: See table 6.3.
3. Vector number: See table 6.3.

6.2.1 Types of Reset

A reset is the highest-priority exception processing source. There are two kinds of reset, power-on and manual. As shown in table 6.5, the CPU state is initialized in both a power-on reset and a manual reset. On-chip peripheral module registers are also initialized by a power-on reset, but not by a manual reset.

Table 6.5 Exception Source Detection and Exception Processing Start Timing

	Conditions for Transition to Reset State		Internal States		
Type	**RES**	**WDT Overflow**	**CPU/MULT/ FPU/INTC**	**On-Chip Peripheral Modules**	**PFC, IO Port**
Power-on reset	Low	—	Initialized	Initialized	Initialized
	High	Power-on reset	Initialized	Initialized	Not initialized
Manual reset	High	Manual reset	Initialized	Not initialized	Not initialized

6.2.2 Power-On Reset

Power-On Reset by Means of $\overline{\text{RES}}$ Pin: When the $\overline{\text{RES}}$ pin is driven low, the chip enters the power-on reset state. To reliably reset the chip, the $\overline{\text{RES}}$ pin should be kept at the low level for at least the duration of the oscillation settling time at power-on or when in standby mode (when the clock is halted), or at least 20 t_{cyc} when the clock is running. In the power-on reset state, the CPU's internal state and all the on-chip peripheral module registers are initialized. See Appendix B, Pin States, for the state of individual pins in the power-on reset state.

In the power-on reset state, power-on reset exception processing starts when the $\overline{\text{RES}}$ pin is first driven low for a set period of time and then returned to high. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception processing vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception processing vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3-I0) of the status register (SR) are set to H'F (1111).
4. The values fetched from the exception processing vector table are set in the PC and SP, and the program begins executing.

Be certain to always perform power-on reset processing when turning the system power on.

on reset state.

The pin function controller (PFC) registers and I/O port registers are not initialized by the reset signal generated by the WDT (these registers are only initialized by a power-on reset from off-chip).

If reset caused by the input signal at the $\overline{\text{RES}}$ pin and a reset caused by WDT overflow occur simultaneously, the $\overline{\text{RES}}$ pin reset has priority, and the WOVF bit in RSTCSR is cleared to 0. When WDT-initiated power-on reset processing is started, the CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception processing vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception processing vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3-I0) of the status register (SR) are set to H'F (1111).
4. The values fetched from the exception processing vector table are set in the PC and SP, and the program begins executing.

6.2.3 Manual Reset

When a setting is made for a manual reset to be generated in the WDT's watchdog timer mode, and the WDT's TCNT overflows, the chip enters the power-on reset state.

When WDT-initiated manual reset processing is started, the CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception processing vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception processing vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3-I0) of the status register (SR) are set to H'F (1111).
4. The values fetched from the exception processing vector table are set in the PC and SP, and the program begins executing.

When a manual reset is generated, the bus cycle is retained, but if a manual reset occurs while the bus is released or during DMAC burst transfer, manual reset exception processing will be deferred until the CPU acquires the bus. However, if the interval from generation of the manual reset until the end of the bus cycle is equal to or longer than the internal manual reset interval of 512 cycles, the internal manual reset source is ignored instead of being deferred, and manual reset exception processing is not executed.

6.3.1 Address Error Sources

Address errors occur when instructions are fetched or data read or written, as shown in table 6.6.

Table 6.6 Bus Cycles and Address Errors

Bus Cycle Type	Bus Master	Bus Cycle Description	Address Errors
Instruction fetch	CPU	Instruction fetched from even address	None (normal)
		Instruction fetched from odd address	Address error occurs
		Instruction fetched from other than on-chip peripheral module space*	None (normal)
		Instruction fetched from on-chip peripheral module space*	Address error occurs
		Instruction fetched from external memory space when in single chip mode	Address error occurs
Data read/write	CPU or DMAC	Word data accessed from even address	None (normal)
		Word data accessed from odd address	Address error occurs
		Longword data accessed from a longword boundary	None (normal)
		Longword data accessed from other than a long-word boundary	Address error occurs
		Byte or word data accessed in on-chip peripheral module space*	None (normal)
		Longword data accessed in 16-bit on-chip peripheral module space*	None (normal)
		Longword data accessed in 8-bit on-chip peripheral module space*	Address error occurs
		External memory space accessed when in single chip mode	Address error occurs

Note: * See section 9, Bus State Controller (BSC), for details of the on-chip peripheral module space.

When an address error occurs, the bus cycle in which the address error occurred ends. When the executing instruction then finishes, address error exception processing starts up. The CPU operates as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the address error that occurred and the program starts executing from that address. The jump that occurs is not a delayed branch.

6.4 Interrupts

6.4.1 Interrupt Sources

Table 6.7 shows the sources that start up interrupt exception processing. These are divided into NMI, user breaks, H-UDI, IRQ, and on-chip peripheral modules.

Table 6.7 Interrupt Sources

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller	1
H-UDI	High-performance user debug interface	1
IRQ	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$ (external input)	8
On-chip peripheral module	Direct memory access controller (DMAC)	4
	Advanced timer unit (ATU-II)	75
	Compare match timer (CMT)	2
	A/D converter	3
	Serial communication interface (SCI)	20
	Watchdog timer (WDT)	1
	Controller area network (HCAN)	8

Each interrupt source is allocated a different vector number and vector table offset. See table 7.3, Interrupt Exception Processing Vectors and Priorities, in section 7, Interrupt Controller (INTC), for more information on vector numbers and vector table address offsets.

The interrupt priority order is predetermined. When multiple interrupts occur simultaneously (overlap), the interrupt controller (INTC) determines their relative priorities and starts up processing according to the results.

The priority order of interrupts is expressed as priority levels 0–16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, so it is always accepted. The user break interrupt and H-UDI interrupt priority level is 15. IRQ interrupts and on-chip peripheral module interrupt priority levels can be set freely using the INTC's interrupt priority registers A through L (IPRA to IPRL) as shown in table 6.8. The priority levels that can be set are 0–15. Level 16 cannot be set. See section 7.3.1, Interrupt Priority Registers A–L (IPRA-IPRL), for details of the interrupt priority registers.

Table 6.8 Interrupt Priority Order

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User break	15	Fixed priority level.
H-UDI	15	Fixed priority level.
IRQ	0–15	Set with interrupt priority level setting registers A through L (IPRA to IPRL).
On-chip peripheral module	0–15	Set with interrupt priority level setting registers A through L (IPRA to IPRL).

6.4.3 Interrupt Exception Processing

When an interrupt occurs, its priority level is ascertained by the interrupt controller (INTC). NMI is always accepted, but other interrupts are only accepted if they have a priority level higher than the priority level set in the interrupt mask bits (I3–I0) of the status register (SR).

When an interrupt is accepted, exception processing begins. In interrupt exception processing, the CPU saves SR and the program counter (PC) to the stack. The priority level value of the accepted interrupt is written to SR bits I3–I0. For NMI, however, the priority level is 16, but the value set in I3–I0 is H'F (level 15). Next, the start address of the exception service routine is fetched from the exception processing vector table for the accepted interrupt, that address is jumped to and execution begins. See section 7.4, Interrupt Operation, for further details.

6.5.1 Types of Exceptions Triggered by Instructions

Exception processing can be triggered by trap instructions, general illegal instructions, and illegal slot instructions, and floating-point instructions, as shown in table 6.9.

Table 6.9 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instructions	TRAPA	
Illegal slot instructions	Undefined code placed immediately after a delayed branch instruction (delay slot) and instructions that rewrite the PC	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF
General illegal instructions	Undefined code anywhere besides in a delay slot	
Floating-point instructions	Instruction causing an invalid operation exception defined in the IEEE754 standard or a division-by-zero exception	FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FNEG, FABS, FTRC

6.5.2 Trap Instructions

When a TRAPA instruction is executed, trap instruction exception processing starts up. The CPU operates as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the vector number specified in the TRAPA instruction. That address is jumped to and the program starts executing. The jump that occurs is not a delayed branch.

An instruction placed immediately after a delayed branch instruction is said to be placed in a delay slot. When the instruction placed in the delay slot is undefined code, illegal slot exception processing starts up when that undefined code is decoded. Illegal slot exception processing also starts up when an instruction that rewrites the program counter (PC) is placed in a delay slot. The processing starts when the instruction is decoded. The CPU handles an illegal slot instruction as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code or the instruction that rewrites the PC.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the exception that occurred. That address is jumped to and the program starts executing. The jump that occurs is not a delayed branch.

6.5.4 General Illegal Instructions

When undefined code placed anywhere other than immediately after a delayed branch instruction (i.e., in a delay slot) is decoded, general illegal instruction exception processing starts up. The CPU handles general illegal instructions in the same way as illegal slot instructions. Unlike processing of illegal slot instructions, however, the program counter value stored is the start address of the undefined code.

When the FPU has been stopped by means of the module stop bit, floating-point instructions and FPU-related CPU instructions are treated as illegal instructions.

6.5.5 Floating-Point Instructions

When the V or Z bit is set in the enable field of the FPSCR register, an FPU exception occurs. This indicates that a floating-point instruction has caused an invalid operation exception defined in the IEEE754 standard or a division-by-zero exception. Floating-point instructions which can cause an exception are as follows:

FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FNEG,
FABS, FTRC

An FPU exception occurs only if the corresponding enable bit is set. When the FPU detects an exception source, FPU operation is suspended and the occurrence of the exception is reported to the CPU. When exception processing is started, the CPU saves the SR and PC contents to the stack (the PC value saved is the start address of the instruction following the last instruction executed), and branches to VBR + H'00000034.

FPSCR cause bits change each time an FPU instruction is executed.

Exception events other than those defined in the IEEE754 standard (i.e., underflow, overflow, and inexact exceptions) are detected by the FPU but do not result in the generation of any kind of exception. Neither is an FPU exception generated by a floating-point instruction relating to data transfer, such as FLOAT.

6.6 When Exception Sources Are Not Accepted

When an address error or interrupt is generated after a delayed branch instruction or interrupt-disabled instruction, it is sometimes not accepted immediately but stored instead, as shown in table 6.10. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 6.10 Generation of Exception Sources Immediately after a Delayed Branch Instruction or Interrupt-Disabled Instruction

	Exception Source		
Point of Occurrence	**Bus Error**	**Interrupt**	**FPU Exception**
Immediately after a delayed branch instruction*1	Not accepted	Not accepted	Not accepted
Immediately after an interrupt-disabled instruction*2	Not accepted*4	Not accepted	Accepted
Immediately after an FPU instruction*3	Not accepted	Not accepted	Accepted

Notes: *1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

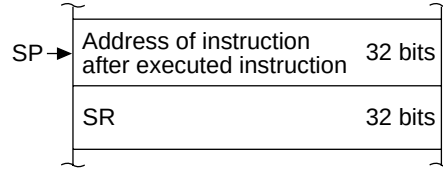
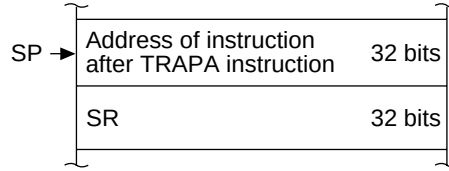
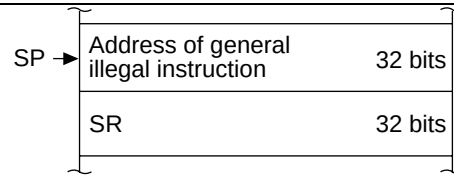
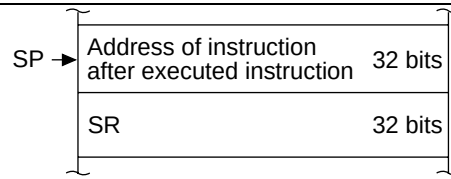
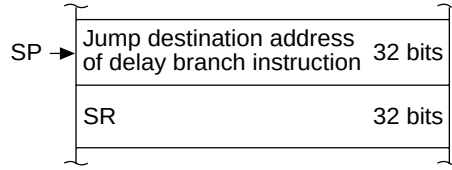
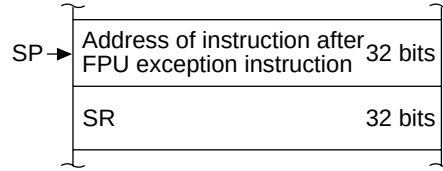
*2. Interrupt-disabled instructions: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS, STS.L

*3. FPU instructions: Table 2.18, Floating-Point Instructions, and table 2.19, FPU-Related CPU Instructions, in section 2.4.1, Instruction Set by Classification.

*4. In the SH-2 a bus error is accepted.

The status of the stack after exception processing ends is as shown in table 6.11.

Table 6.11 Stack Status After Exception Processing Ends

Exception Type	Stack Status
Address error	
Trap instruction	
General illegal instruction	
Interrupt	
Illegal slot instruction	
FPU exception	

6.8.1 Value of Stack Pointer (SP)

The value of the stack pointer must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception processing.

6.8.2 Value of Vector Base Register (VBR)

The value of the vector base register must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception processing.

6.8.3 Address Errors Caused by Stacking of Address Error Exception Processing

When the stack pointer is not a multiple of four, an address error will occur during stacking of the exception processing (interrupts, etc.) and address error exception processing will start up as soon as the first exception processing is ended. Address errors will then also occur in the stacking for this address error exception processing. To ensure that address error exception processing does not go into an endless loop, no address errors are accepted at that point. This allows program control to be shifted to the address error exception service routine and enables error processing.

When an address error occurs during exception processing stacking, the stacking bus cycle (write) is executed. During stacking of the status register (SR) and program counter (PC), the SP is decremented by 4 for both, so the value of SP will not be a multiple of four after the stacking either. The address value output during stacking is the SP value, so the address where the error occurred is itself output. This means the write data stacked will be undefined.

6.8.4 Interrupt Processing Timing Gap Caused in SCO Processing

If an interrupt processing is generated in an SCO processing, the interrupt generation timing is different because the interrupt processing is started after the SCO* processing. For details on the arbitration with the SCO processing, refer to section 22.8.2(1).

Note: SCO is the processing to download the flash memory programming/erasing program on the on-chip RAM.

7.1 Overview

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC has registers for setting the priority of each interrupt which can be used by the user to order the priorities in which the interrupt requests are processed.

7.1.1 Features

The INTC has the following features:

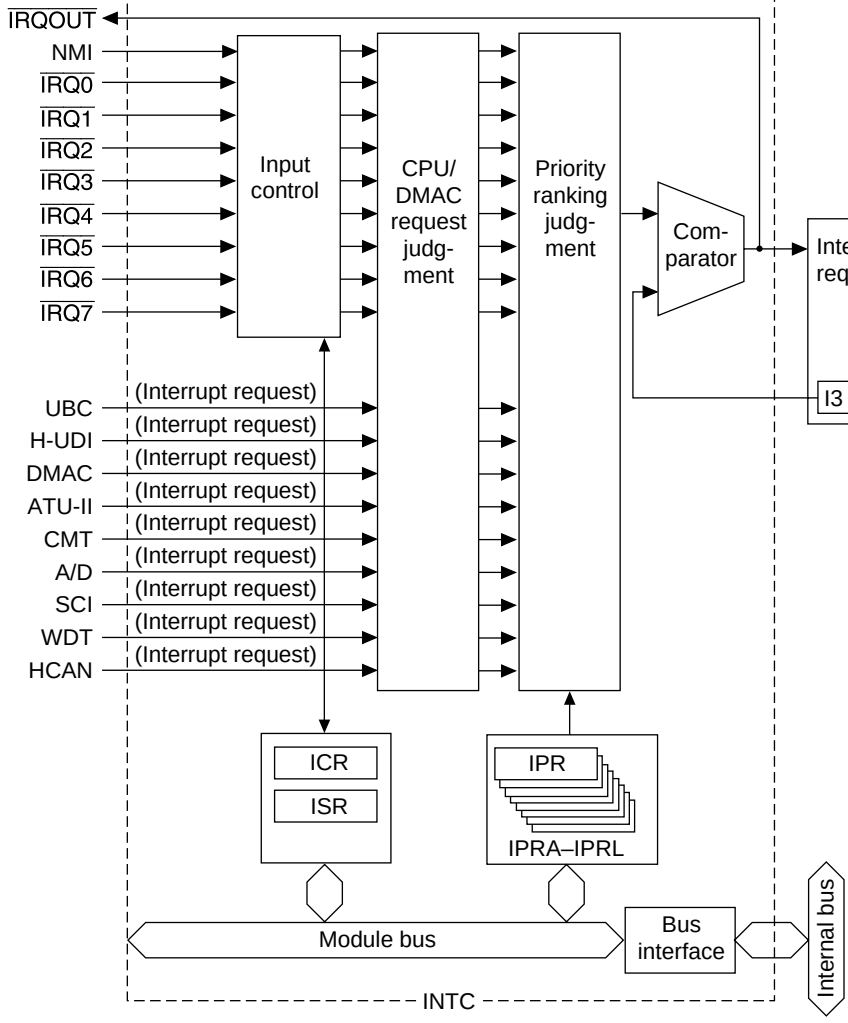
- 16 levels of interrupt priority

  By setting the twelve interrupt-priority level registers, the priorities of IRQ interrupts and on-chip peripheral module interrupts can be set in 16 levels for different request sources.
- NMI noise canceler function

  NMI input level bits indicate the NMI pin status. By reading these bits with the interrupt exception service routine, the pin status can be confirmed, enabling it to be used as a noise canceler.
- Notification of interrupt occurrence can be reported externally ($\overline{\text{IRQOUT}}$ pin)

  For example, it is possible to request the bus if an external bus master is informed that a peripheral module interrupt has occurred when the chip has released the bus.

Figure 7.1 is a block diagram of the INTC.



UBC: User break controller
H-UDI: High-performance user debug interface
DMAC: Direct memory access controller
ATU-II: Advanced timer unit
CMT: Compare match timer
A/D: A/D converter
SCI: Serial communication interface
WDT: Watchdog timer
HCAN: Controller area network
ICR: Interrupt control register
ISR: IRQ status register
IPRA–IPRL: Interrupt priority level setting registers A to L
SR: Status register

Figure 7.1 INTC Block Diagram

Table 7.1 shows the INTC pin configuration.

Table 7.1 Pin Configuration

Name	Abbreviation	I/O	Function
Non-maskable interrupt input pin	NMI	I	Input of non-maskable interrupt request signal
Interrupt request input pins	$\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$	I	Input of maskable interrupt request signals
Interrupt request output pin	$\overline{\text{IRQOUT}}$	O	Output of notification signal when an interrupt has occurred

7.1.4 Register Configuration

The INTC has the 14 registers shown in table 7.2. These registers set the priority of the interrupts and control external interrupt input signal detection.

Table 7.2 Register Configuration

Name	Abbr.	R/W	Initial Value	Address	Access Sizes
Interrupt priority register A	IPRA	R/W	H'0000	H'FFFF ED00	8, 16, 32
Interrupt priority register B	IPRB	R/W	H'0000	H'FFFF ED02	8, 16, 32
Interrupt priority register C	IPRC	R/W	H'0000	H'FFFF ED04	8, 16, 32
Interrupt priority register D	IPRD	R/W	H'0000	H'FFFF ED06	8, 16, 32
Interrupt priority register E	IPRE	R/W	H'0000	H'FFFF ED08	8, 16, 32
Interrupt priority register F	IPRF	R/W	H'0000	H'FFFF ED0A	8, 16, 32
Interrupt priority register G	IPRG	R/W	H'0000	H'FFFF ED0C	8, 16, 32
Interrupt priority register H	IPRH	R/W	H'0000	H'FFFF ED0E	8, 16, 32
Interrupt priority register I	IPRI	R/W	H'0000	H'FFFF ED10	8, 16, 32
Interrupt priority register J	IPRJ	R/W	H'0000	H'FFFF ED12	8, 16, 32
Interrupt priority register K	IPRK	R/W	H'0000	H'FFFF ED14	8, 16, 32
Interrupt priority register L	IPRL	R/W	H'0000	H'FFFF ED16	8, 16, 32
Interrupt control register	ICR	R/W	*[1]	H'FFFF ED18	8, 16, 32
IRQ status register	ISR	R(W)*[2]	H'0000	H'FFFF ED1A	8, 16, 32

Notes: Three access cycles are required for byte access and word access, and six cycles for longword access.

*1. The value when the NMI pin is high is H'8000; when the NMI pin is low, it is H'0000.

*2. Only 0 can be written, in order to clear flags.

There are five types of interrupt sources: NMI, user breaks, H-UDI, IRQ, and on-chip peripheral modules. Each interrupt has a priority expressed as a priority level (0 to 16, with 0 the lowest and 16 the highest). Giving an interrupt a priority level of 0 masks it.

7.2.1 NMI Interrupts

The NMI interrupt has priority 16 and is always accepted. Input at the NMI pin is detected by edge. Use the NMI edge select bit (NMIE) in the interrupt control register (ICR) to select either the rising or falling edge. NMI interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to level 15.

7.2.2 User Break Interrupt

A user break interrupt has a priority of level 15, and occurs when the break condition set in the user break controller (UBC) is satisfied. User break interrupt requests are detected by edge and are held until accepted. User break interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to level 15. For more information about the user break interrupt, see section 8, User Break Controller (UBC).

7.2.3 H-UDI Interrupt

A serial debug interface (H-UDI) interrupt has a priority level of 15, and occurs when an H-UDI interrupt instruction is serially input. H-UDI interrupt requests are detected by edge and are held until accepted. H-UDI exception processing sets the interrupt mask level bits (I3-I0) in the status register (SR) to level 15. For more information about the H-UDI interrupt, see section 18, High-Performance User Debug Interface (H-UDI).

7.2.4 IRQ Interrupts

IRQ interrupts are requested by input from pins $\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$. Set the IRQ sense select bits (IRQ0S–IRQ7S) of the interrupt control register (ICR) to select low level detection or falling edge detection for each pin. The priority level can be set from 0 to 15 for each pin using interrupt priority registers A and B (IPRA–IPRB).

When IRQ interrupts are set to low level detection, an interrupt request signal is sent to the INTC during the period the IRQ pin is low. Interrupt request signals are not sent to the INTC when the IRQ pin becomes high. Interrupt request levels can be confirmed by reading the IRQ flags (IRQ0F–IRQ7F) of the IRQ status register (ISR).

detection results are maintained until the interrupt request is accepted. Confirmation that IRQ interrupt requests have been detected is possible by reading the IRQ flags (IRQ0F–IRQ7F) of the IRQ status register (ISR), and by writing a 0 after reading a 1, IRQ interrupt request detection results can be withdrawn.

In IRQ interrupt exception processing, the interrupt mask bits (I3–I0) of the status register (SR) are set to the priority level value of the accepted IRQ interrupt.

7.2.5 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are interrupts generated by the following on-chip peripheral modules:

- Direct memory access controller (DMAC)
- Advanced timer unit (ATU-II)
- Compare match timer (CMT)
- A/D converter (A/D)
- Serial communication interface (SCI)
- Watchdog timer (WDT)
- Controller area network (HCAN)

A different interrupt vector is assigned to each interrupt source, so the exception service routine does not have to decide which interrupt has occurred. Priority levels between 0 and 15 can be assigned to individual on-chip peripheral modules in interrupt priority registers C–L (IPRC–IPRL).

On-chip peripheral module interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to the priority level value of the on-chip peripheral module interrupt that was accepted.

Table 7.3 lists interrupt sources and their vector numbers, vector table address offsets and interrupt priorities.

Each interrupt source is allocated a different vector number and vector table address offset. Vector table addresses are calculated from vector numbers and address offsets. In interrupt exception processing, the exception service routine start address is fetched from the vector table indicated by the vector table address. See table 6.4, Calculating Exception Processing Vector Table Addresses, in section 6, Exception Processing.

IRQ interrupts and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers A–L (IPRA–IPRL). The ranking of interrupt sources for IPRC–IPRL, however, must be the order listed under Priority within IPR Setting Range in table 7.3 and cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts and on-chip peripheral module interrupts. If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, their priority order is the default priority order indicated at the right in table 7.3.

Interrupt Source		Vector No.	Vector Table Address Offset	Priority (Initial Value)	Corre-sponding IPR (Bits)	Setting within IPR Range		Default Priority
NMI		11	H'0000002C to H'0000002F	16	—	—		High
UBC		12	H'00000030 to H'00000033	15	—	—		
H-UDI		14	H'00000038 to H'00000038	15	—	—		
IRQ0		64	H'00000100 to H'00000103	0 to 15 (0)	IPRA (15–12)	—		
IRQ1		65	H'00000104 to H'00000107	0 to 15 (0)	IPRA (11–8)	—		
IRQ2		66	H'00000108 to H'0000010B	0 to 15 (0)	IPRA (7–4)	—		
IRQ3		67	H'0000010C to H'0000010F	0 to 15 (0)	IPRA (3–0)	—		
IRQ4		68	H'00000110 to H'00000113	0 to 15 (0)	IPRB (15–12)	—		
IRQ5		69	H'00000114 to H'00000117	0 to 15 (0)	IPRB (11–8)	—		
IRQ6		70	H'00000118 to H'0000011B	0 to 15 (0)	IPRB (7–4)	—		
IRQ7		71	H'0000011C to H'0000011F	0 to 15 (0)	IPRB (3–0)	—		
DMAC0	DEI0	72	H'00000120 to H'00000123	0 to 15 (0)	IPRC (15–12)	↑	1	
DMAC1	DEI1	74	H'00000128 to H'0000012B	0 to 15 (0)		↓	2	
DMAC2	DEI2	76	H'00000130 to H'00000133	0 to 15 (0)	IPRC (11–8)	↑	1	
DMAC3	DEI3	78	H'00000138 to H'0000013B	0 to 15 (0)		↓	2	Low

Interrupt Source			Vector No.	Vector Table Address Offset	Priority (Initial Value)	Corre-sponding IPR (Bits)	within IPR Setting Range		Default Priority
ATU0	ATU01	ITV1/ ITV2A/ ITV2B	80	H'00000140 to H'00000143	0 to 15 (0)	IPRC (7–4)			High
	ATU02	ICI0A	84	H'00000150 to H'00000153	0 to 15 (0)	IPRC (3–0)	↑	1	
		ICI0B	86	H'00000158 to H'0000015B			↓	2	
	ATU03	ICI0C	88	H'00000160 to H'00000163	0 to 15 (0)	IPRD (15–12)	↑	1	
		ICI0D	90	H'00000168 to H'0000016B			↓	2	
	ATU04	OVI0	92	H'00000170 to H'00000173	0 to 15 (0)	IPRD (11–8)			
ATU1	ATU11	IMI1A/CMI1	96	H'00000180 to H'00000183	0 to 15 (0)	IPRD (7–4)	↑	1	
		IMI1B	97	H'00000184 to H'00000187				2	
		IMI1C	98	H'00000188 to H'0000018B				3	
		IMI1D	99	H'0000018C to H'0000018F			↓	4	
	ATU12	IMI1E	100	H'00000190 to H'00000193	0 to 15 (0)	IPRD (3–0)	↑	1	
		IMI1F	101	H'00000194 to H'00000197				2	
		IMI1G	102	H'00000198 to H'0000019B				3	
		IMI1H	103	H'0000019C to H'0000019F			↓	4	
	ATU13	OVI1A/ OVI1B	104	H'000001A0 to H'000001A3	0 to 15 (0)	IPRE (15–12)			Low

Interrupt Source			Vector No.	Vector Table Address Offset	Priority (Initial Value)	Corre-sponding IPR (Bits)	within IPR Setting Range		Default Priority
ATU2	ATU21	IMI2A/CMI2A	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPRE (11–8)	↑	1	High
		IMI2B/CMI2B	109	H'000001B4 to H'000001B7				2	
		IMI2C/CMI2C	110	H'000001B8 to H'000001BB				3	
		IMI2D/CMI2D	111	H'000001BC to H'000001BF			↓	4	
	ATU22	IMI2E/CMI2E	112	H'000001C0 to H'000001C3	0 to 15 (0)	IPRE (7–4)	↑	1	
		IMI2F/CMI2F	113	H'000001C4 to H'000001C7				2	
		IMI2G/CMI2G	114	H'000001C8 to H'000001CB				3	
		IMI2H/CMI2H	115	H'000001CC to H'000001CF			↓	4	
	ATU23	OVI2A/OVI2B	116	H'000001D0 to H'000001D3	0 to 15 (0)	IPRE (3–0)			
ATU3	ATU31	IMI3A	120	H'000001E0 to H'000001E3	0 to 15 (0)	IPRF (15–12)	↑	1	
		IMI3B	121	H'000001E4 to H'000001E7				2	
		IMI3C	122	H'000001E8 to H'000001EB				3	
		IMI3D	123	H'000001EC to H'000001EF			↓	4	
	ATU32	OVI3	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPRF (11–8)			Low

Interrupt Source			Vector No.	Vector Table Address Offset	Priority (Initial Value)	Corre-sponding IPR (Bits)		within IPR Setting Range	Default Priority
ATU4	ATU41	IMI4A	128	H'00000200 to H'00000203	0 to 15 (0)	IPRF (7–4)	↑	1	High
		IMI4B	129	H'00000204 to H'00000207				2	
		IMI4C	130	H'00000208 to H'0000020B				3	
		IMI4D	131	H'0000020C to H'0000020F			↓	4	
	ATU42	OVI4	132	H'00000210 to H'00000213	0 to 15 (0)	IPRF (3–0)			
ATU5	ATU51	IMI5A	136	H'00000220 to H'00000223	0 to 15 (0)	IPRG (15–12)	↑	1	
		IMI5B	137	H'00000224 to H'00000227				2	
		IMI5C	138	H'00000228 to H'0000022B				3	
		IMI5D	139	H'0000022C to H'0000022F			↓	4	
	ATU52	OVI5	140	H'00000230 to H'00000233	0 to 15 (0)	IPRG (11–8)			
ATU6		CMI6A	144	H'00000240 to H'00000243	0 to 15 (0)	IPRG (7–4)	↑	1	
		CMI6B	145	H'00000244 to H'00000247				2	
		CMI6C	146	H'00000248 to H'0000024B				3	
		CMI6D	147	H'0000024C to H'0000024F			↓	4	Low

Interrupt Source			Vector No.	Vector Table Address Offset	Priority (Initial Value)	Corre-sponding IPR (Bits)	within IPR Setting Range		Default Priority
ATU7		CMI7A	148	H'00000250 to H'00000253	0 to 15 (0)	IPRG (3–0)	↑	1	High
		CMI7B	149	H'00000254 to H'00000257				2	
		CMI7C	150	H'00000258 to H'0000025B				3	
		CMI7D	151	H'0000025C to H'0000025F			↓	4	
ATU8	ATU81	OSI8A	152	H'00000260 to H'00000263	0 to 15 (0)	IPRH (15–12)	↑	1	
		OSI8B	153	H'00000264 to H'00000267				2	
		OSI8C	154	H'00000268 to H'0000026B				3	
		OSI8D	155	H'0000026C to H'0000026F			↓	4	
	ATU82	OSI8E	156	H'00000270 to H'00000273	0 to 15 (0)	IPRH (11–8)	↑	1	
		OSI8F	157	H'00000274 to H'00000277				2	
		OSI8G	158	H'00000278 to H'0000027B				3	
		OSI8H	159	H'0000027C to H'0000027F			↓	4	
	ATU83	OSI8I	160	H'00000280 to H'00000283	0 to 15 (0)	IPRH (7–4)	↑	1	
		OSI8J	161	H'00000284 to H'00000287				2	
		OSI8K	162	H'00000288 to H'0000028B				3	
		OSI8L	163	H'0000028C to H'0000028F			↓	4	Low

Interrupt Source			Vector No.	Vector Table Address Offset	Priority (Initial Value)	Corre-sponding IPR (Bits)	Priority within IPR Setting Range		Default Priority
ATU8	ATU84	OSI8M	164	H'00000290 to H'00000293	0 to 15 (0)	IPRH (3–0)	↑	1	High
		OSI8N	165	H'00000294 to H'00000297				2	
		OSI8O	166	H'00000298 to H'0000029B				3	
		OSI8P	167	H'0000029C to H'0000029F			↓	4	
ATU9	ATU91	CMI9A	168	H'000002A0 to H'000002A3	0 to 15 (0)	IPRI (15–12)	↑	1	
		CMI9B	169	H'000002A4 to H'000002A7				2	
		CMI9C	170	H'000002A8 to H'000002AB				3	
		CMI9D	171	H'000002AC to H'000002AF			↓	4	
	ATU92	CMI9E	172	H'000002B0 to H'000002B3	0 to 15 (0)	IPRI (11–8)	↑	1	
		CMI9F	174	H'000002B8 to H'000002BB			↓	2	
ATU10	ATU101	CMI10A	176	H'000002C0 to H'000002C3	0 to 15 (0)	IPRI (7–4)	↑	1	
		CMI10B	178	H'000002C8 to H'000002CB			↓	2	
	ATU102	ICI10A/CMI10G	180	H'000002D0 to H'000002D3	0 to 15(0)	IPRI (3–0)			
	ATU11	IMI11A	184	H'000002E0 to H'000002E3	0 to 15 (0)	IPRJ (15–12)	↑	1	
		IMI11B	186	H'000002E8 to H'000002EB				2	
		OVI11	187	H'000002EC to H'000002EF			↓	3	Low

Interrupt Source		Vector No.	Vector Table Address Offset	Priority (Initial Value)	Corre-sponding IPR (Bits)	within IPR Setting Range		Default Priority
CMT0	CMTI0	188	H'000002F0 to H'000002F3	0 to 15 (0)	I PRJ (11–8)	↑	1	High
A/D0	ADI0	190	H'000002F8 to H'000002FB			↓	2	
CMT1	CMTI1	192	H'00000300 to H'00000303	0 to 15 (0)	IPRJ (7–4)	↑	1	
A/D1	ADI1	194	H'00000308 to H'0000030B			↓	2	
A/D2	ADI2	196	H'00000310 to H'00000313	0 to 15 (0)	IPRJ (3–0)			
SCI0	ERI0	200	H'00000320 to H'00000323	0 to 15 (0)	IPRK (15–12)	↑	1	
	RXI0	201	H'00000324 to H'00000327				2	
	TXI0	202	H'00000328 to H'0000032B				3	
	TEI0	203	H'0000032C to H'0000032F			↓	4	
SCI1	ERI1	204	H'00000330 to H'00000333	0 to 15 (0)	IPRK (11–8)	↑	1	
	RXI1	205	H'00000334 to H'00000337				2	
	TXI1	206	H'00000338 to H'0000033B				3	
	TEI1	207	H'0000033C to H'0000033F			↓	4	
SCI2	ERI2	208	H'00000340 to H'00000343	0 to 15 (0)	IPRK (7–4)	↑	1	
	RXI2	209	H'00000344 to H'00000347				2	
	TXI2	210	H'00000348 to H'0000034B				3	
	TEI2	211	H'0000034C to H'0000034F			↓	4	Low

Interrupt Source		Vector No.	Vector Table Address Offset	Priority (Initial Value)	Corre-sponding IPR (Bits)	Setting within IPR Range		Default Priority
SCI3	ERI3	212	H'00000350 to H'00000353	0 to 15 (0)	IPRK (3–0)	↑	1	High
	RXI3	213	H'00000354 to H'00000357				2	
	TXI3	214	H'00000358 to H'0000035B				3	
	TEI3	215	H'0000035C to H'0000035F			↓	4	
SCI4	ERI4	216	H'00000360 to H'00000363	0 to 15 (0)	IPRL (15–12)	↑	1	
	RXI4	217	H'00000364 to H'00000367				2	
	TXI4	218	H'00000368 to H'0000036B				3	
	TEI4	219	H'0000036C to H'0000036F			↓	4	
HCAN0	ERS0	220	H'00000370 to H'00000373	0 to 15 (0)	IPRL (11–8)	↑	1	
	OVR0	221	H'00000374 to H'00000377				2	
	RM0	222	H'00000378 to H'0000037B				3	
	SLE0	223	H'0000037C to H'0000037F			↓	4	
WDT	ITI	224	H'00000380 to H'00000383	0 to 15 (0)	IPRL (7–4)			
HCAN1	ERS1	228	H'00000390 to H'00000393	0 to 15 (0)	IPRL (3–0)	↑	1	
	OVR1	229	H'00000394 to H'00000397				2	
	RM1	230	H'00000398 to H'0000039B				3	
	SLE1	231	H'0000039C to H'0000039F			↓	4	Low

7.3.1 Interrupt Priority Registers A–L (IPRA–IPRL)

Interrupt priority registers A–L (IPRA–IPRL) are 16-bit readable/writable registers that set priority levels from 0 to 15 for IRQ interrupts and on-chip peripheral module interrupts. Correspondence between interrupt request sources and each of the IPRA–IPRL bits is shown in table 7.4.

Bit:	15	14	13	12	11	10	9	8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.4 Interrupt Request Sources and IPRA–IPRL

	Bits			
Register	**15–12**	**11–8**	**7–4**	**3–0**
Interrupt priority register A	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register B	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register C	DMAC0, 1	DMAC2, 3	ATU01	ATU02
Interrupt priority register D	ATU03	ATU04	ATU11	ATU12
Interrupt priority register E	ATU13	ATU21	ATU22	ATU23
Interrupt priority register F	ATU31	ATU32	ATU41	ATU42
Interrupt priority register G	ATU51	ATU52	ATU6	ATU7
Interrupt priority register H	ATU81	ATU82	ATU83	ATU84
Interrupt priority register I	ATU91	ATU92	ATU101	ATU102
Interrupt priority register J	ATU11	CMT0, A/D0	CMT1, A/D1	A/D2
Interrupt priority register K	SCI0	SCI1	SCI2	SCI3
Interrupt priority register L	SCI4	HCAN0	WDT	HCAN1

from H'0 (0000) to H'F (1111) in each of the four-bit groups 15–12, 11–8, 7–4, and 3–0. Interrupt priority rank becomes level 0 (lowest) by setting H'0, and level 15 (highest) by setting H'F. If multiple on-chip peripheral modules are assigned to the same bit (DMAC0 and DMAC1, DMAC2 and DMAC3, CMT0 and A/D0, and CMT1 and A/D1), those multiple modules are set to the same priority rank.

IPRA–IPRL are initialized to H'0000 by a reset and in hardware standby mode. They are not initialized in software standby mode.

7.3.2 Interrupt Control Register (ICR)

ICR is a 16-bit register that sets the input signal detection mode of the external interrupt input pin NMI and $\overline{\text{IRQ0}}$ –$\overline{\text{IRQ7}}$ and indicates the input signal level at the NMI pin. A reset and hardware standby mode initialize ICR but the software standby mode does not.

Bit:	15	14	13	12	11	10	9	8
	NMIL	—	—	—	—	—	—	NMIE
Initial value:	*	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	IRQ0S	IRQ1S	IRQ2S	IRQ3S	IRQ4S	IRQ5S	IRQ6S	IRQ7S
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * When NMI input is high: 1; when NMI input is low: 0

- Bit 15—NMI Input Level (NMIL): Sets the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified.

Bit 15: NMIL	Description
0	NMI input level is low
1	NMI input level is high

- Bits 14 to 9—Reserved: These bits always read 0. The write value should always be 0.

0	Interrupt request is detected on falling edge of NMI input (Initial value)
1	Interrupt request is detected on rising edge of NMI input

- Bits 7 to 0—IRQ0–IRQ7 Sense Select (IRQ0S–IRQ7S): These bits set the IRQ0–IRQ7 interrupt request detection mode.

Bits 7-0: IRQ0S–IRQ7S	Description
0	Interrupt request is detected on low level of IRQ input (Initial value)
1	Interrupt request is detected on falling edge of IRQ input

7.3.3 IRQ Status Register (ISR)

ISR is a 16-bit register that indicates the interrupt request status of the external interrupt input pins $\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$. When IRQ interrupts are set to edge detection, held interrupt requests can be withdrawn by writing 0 to IRQnF after reading IRQnF = 1.

A reset and hardware standby mode initialize ISR but software standby mode does not.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	IRQ0F	IRQ1F	IRQ2F	IRQ3F	IRQ4F	IRQ5F	IRQ6F	IRQ7F
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 8—Reserved: These bits always read 0. The write value should always be 0.

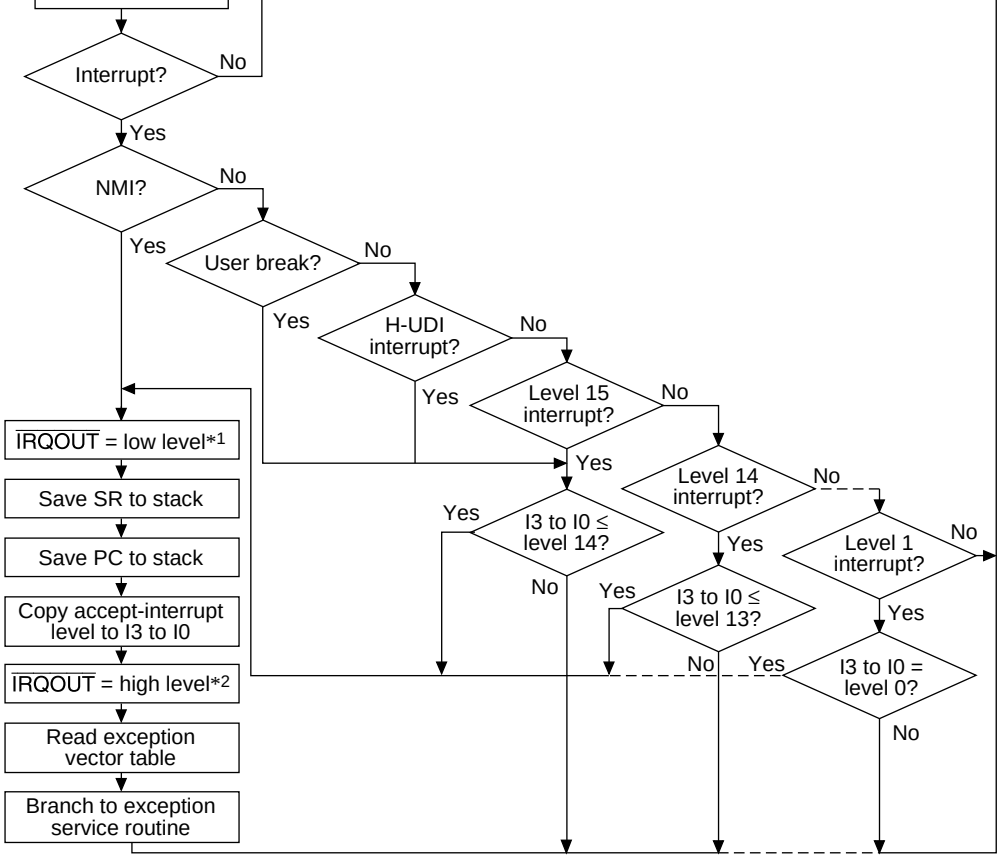
Bits 7-0: IRQ0F–IRQ7F	Detection Setting	Description
0	Level detection	No IRQn interrupt request exists [Clearing condition] When $\overline{\text{IRQn}}$ input is high
	Edge detection	No IRQn interrupt request was detected (Initial value) [Clearing conditions] • When 0 is written after reading IRQnF = 1 • When IRQn interrupt exception processing has been executed
1	Level detection	An IRQn interrupt request exists Setting condition: When $\overline{\text{IRQn}}$ input is low
	Edge detection	An IRQn interrupt request was detected Setting condition: When a falling edge occurs at an $\overline{\text{IRQn}}$ input

n = 7 to 0

7.4.1 Interrupt Sequence

The sequence of interrupt operations is explained below. Figure 7.2 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest priority interrupt in the interrupt requests sent, following the priority levels set in interrupt priority registers A–L (IPRA–IPRL). Lower-priority interrupts are ignored. They are held pending until interrupt requests designated as edge-detect type are accepted. For IRQ interrupts, however, withdrawal is possible by accessing the IRQ status register (ISR). See section 7.2.4, IRQ Interrupts, for details. Interrupts held pending due to edge detection are cleared by a power-on reset or a manual reset. If two of these interrupts have the same priority level or if multiple interrupts occur within a single module, the interrupt with the highest default priority or the highest priority within its IPR setting range (as indicated in table 7.3) is selected.
3. The interrupt controller compares the priority level of the selected interrupt request with the interrupt mask bits (I3–I0) in the CPU's status register (SR). If the request priority level is equal to or less than the level set in I3–I0, the request is ignored. If the request priority level is higher than the level in bits I3–I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. When the interrupt controller accepts an interrupt, a low level is output from the $\overline{\text{IRQOUT}}$ pin.
5. The CPU detects the interrupt request sent from the interrupt controller when it decodes the next instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception processing (figure 7.4).
6. SR and PC are saved onto the stack.
7. The priority level of the accepted interrupt is copied to the interrupt mask level bits (I3 to I0) in the status register (SR).
8. When the accepted interrupt is sensed by level or is from an on-chip peripheral module, a high level is output from the $\overline{\text{IRQOUT}}$ pin. When the accepted interrupt is sensed by edge, a high level is output from the $\overline{\text{IRQOUT}}$ pin at the point when the CPU starts interrupt exception processing instead of instruction execution as noted in 5 above. However, if the interrupt controller accepts an interrupt with a higher priority than one it is in the process of accepting, the $\overline{\text{IRQOUT}}$ pin will remain low.
9. The CPU reads the start address of the exception service routine from the exception vector table for the accepted interrupt, jumps to that address, and starts executing the program there. This jump is not a delay branch.



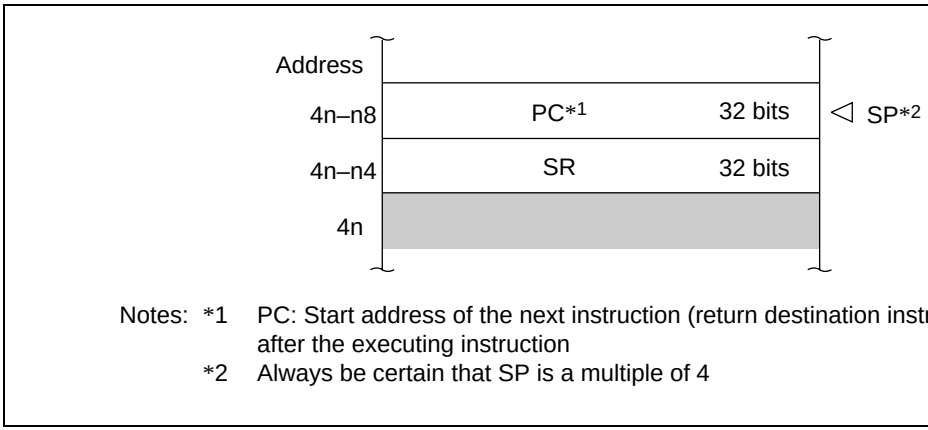
I3 to I0: Interrupt mask bits of status register

Notes: *1. $\overline{\text{IRQOUT}}$ is the same signal as the interrupt request signal to the CPU (see figure 7.1). Thus, it is output when there is a higher priority interrupt request than the one in the I3 to I0 bits of SR.

*2. When the accepted interrupt is sensed by edge, the $\overline{\text{IRQOUT}}$ pin becomes high level at the point when the CPU starts interrupt exception processing instead of instruction execution (before SR is saved to the stack).
If the interrupt controller has accepted another interrupt with a higher priority and has output an interrupt request to the CPU, the $\overline{\text{IRQOUT}}$ pin will remain low.

Figure 7.2 Interrupt Sequence Flowchart

Figure 7.3 shows the stack after interrupt exception processing.



Notes: *1 PC: Start address of the next instruction (return destination instruction) after the executing instruction
*2 Always be certain that SP is a multiple of 4

Figure 7.3 Stack after Interrupt Exception Processing

Table 7.5 indicates the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception processing starts and fetching of the first instruction of the interrupt service routine begins. Figure 7.4 shows an example of pipeline operation when an IRQ interrupt is accepted.

Table 7.5 Interrupt Response Time

Item		Number of States: NMI, Peripheral Module	Number of States: IRQ	Notes
DMAC activation judgment		0 or 1	0	1 state required for interrupt signals for which DMAC activation is possible
Compare identified interrupt priority with SR mask level		2	3	
Wait for completion of sequence currently being executed by CPU		X (≥ 0)		The longest sequence is for interrupt or address-error exception processing (X = 4 + m1 + m2 + m3 + m4). If an interrupt-masking instruction follows, however, the time may be even longer.
Time from start of interrupt exception processing until fetch of first instruction of exception service routine starts		5 + m1 + m2 + m3		Performs the PC and SR saves and vector address fetch.
Interrupt response time	Total:	(7 or 8) + m1 + m2 + m3 + X	8 + m1 + m2 + m3 + X	
	Minimum:	10	11	0.25 to 0.28 µs at 40 MHz
	Maximum:	12 + 2 (m1 + m2 + m3) + m4	12 + 2 (m1 + m2 + m3) + m4	0.48 µs at 40 MHz*

Note: * When m1 = m2 = m3 = m4 = 1

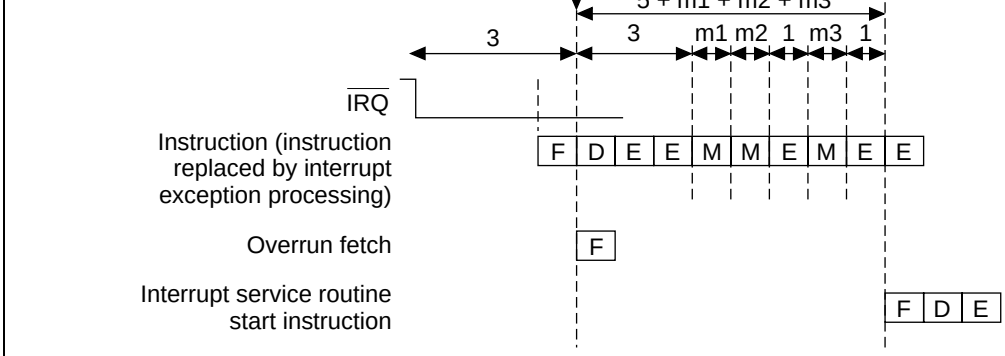
m1–m4 are the number of states needed for the following memory accesses.

m1: SR save (longword write)

m2: PC save (longword write)

m3: Vector address read (longword read)

m4: Fetch first instruction of interrupt service routine



F: Instruction fetch (instruction fetched from memory where program is stored).
D: Instruction decoding (fetched instruction is decoded).
E: Instruction execution (data operation and address calculation is performed according to the results of decoding).
M: Memory access (data in memory is accessed).

Figure 7.4 Example of Pipeline Operation when an IRQ Interrupt is Accepted

The following data transfer can be carried out using interrupt request signals:

- Activate DMAC only, without generating CPU interrupt

Among interrupt sources, those designated as DMAC activating sources are masked and not input to the INTC. The masking condition is as follows:

Mask condition = DME • (DE0 • source selection 0 + DE1 • source selection 1 + DE2 • source selection 2 + DE3 • source selection 3)

7.6.1 Handling CPU Interrupt Sources, but Not DMAC Activating Sources

1. Either do not select the DMAC as a source, or clear the DME bit to 0.
2. Activating sources are applied to the CPU when interrupts occur.
3. The CPU clears interrupt sources with its interrupt processing routine and performs the necessary processing.

7.6.2 Handling DMAC Activating Sources but Not CPU Interrupt Sources

1. Select the DMAC as a source and set the DME bit to 1. CPU interrupt sources are masked regardless of the interrupt priority level register settings.
2. Activating sources are applied to the DMAC when interrupts occur.
3. The DMAC clears activating sources at the time of data transfer.

8.1 Overview

The user break controller (UBC) provides functions that simplify program debugging. Break conditions are set in the UBC and a user break interrupt is generated according to the conditions of the bus cycle generated by the CPU or DMAC. This function makes it easy to design an effective self-monitoring debugger, enabling the chip to easily debug programs without using a large in-circuit emulator.

8.1.1 Features

The features of the user break controller are:

- The following break compare conditions can be set:
  - Address
  - CPU cycle/DMA cycle
  - Instruction fetch or data access
  - Read or write
  - Operand size: byte/word/longword
- User break interrupt generated upon satisfying break conditions
  A user-designed user break interrupt exception processing routine can be run.
- Select either to break in the CPU instruction fetch cycle before the instruction is executed or after.
- Satisfaction of a break condition can be output to the $\overline{\text{UBCTRG}}$ pin.

Figure 8.1 shows a block diagram of the UBC.

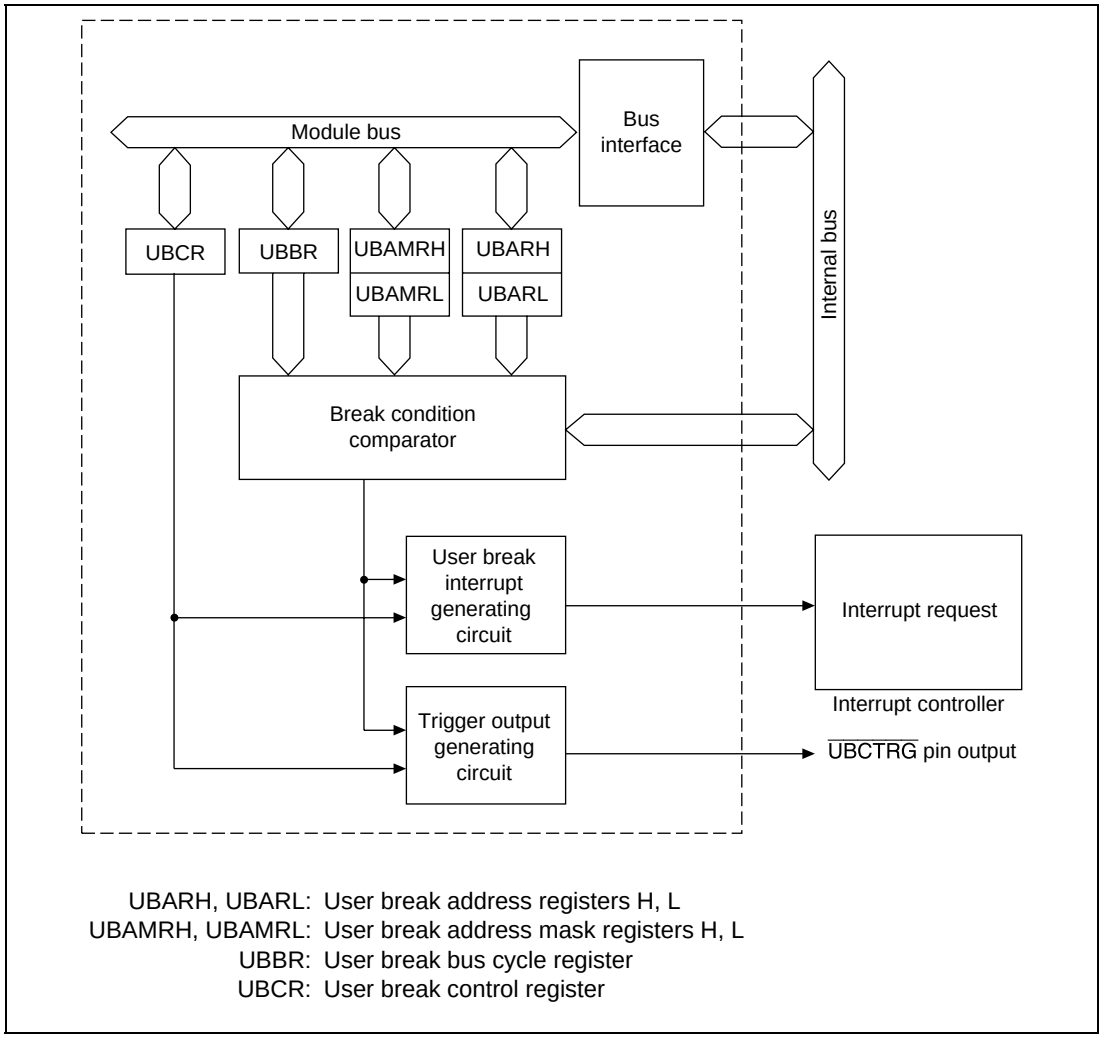


Figure 8.1 User Break Controller Block Diagram

The UBC has the six registers shown in table 8.1. Break conditions are established using these registers.

Table 8.1 Register Configuration

Name	Abbr.	R/W	Initial Value	Address*	Access Size
User break address register H	UBARH	R/W	H'0000	H'FFFFEC00	8, 16, 32
User break address register L	UBARL	R/W	H'0000	H'FFFFEC02	8, 16, 32
User break address mask register H	UBAMRH	R/W	H'0000	H'FFFFEC04	8, 16, 32
User break address mask register L	UBAMRL	R/W	H'0000	H'FFFFEC06	8, 16, 32
User break bus cycle register	UBBR	R/W	H'0000	H'FFFFEC08	8, 16, 32
User break control register	UBCR	R/W	H'0000	H'FFFFEC0A	8, 16, 32

Note: * In register access, three cycles are required for byte access and word access, and six cycles for longword access.

8.2 Register Descriptions

8.2.1 User Break Address Register (UBAR)

UBARH:

Bit:	15	14	13	12	11	10	9	8
	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8
	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The user break address register (UBAR) consists of user break address register H (UBARH) and user break address register L (UBARL). Both are 16-bit readable/writable registers. UBARH stores the upper bits (bits 31 to 16) of the address of the break condition, while UBARL stores the lower bits (bits 15 to 0). UBARH and UBARL are initialized to H'0000 by a power-on reset and in module standby mode. They are not initialized in software standby mode.

- UBARH Bits 15 to 0—User Break Address 31 to 16 (UBA31 to UBA16): These bits store the upper bit values (bits 31 to 16) of the address of the break condition.
- UBARL Bits 15 to 0—User Break Address 15 to 0 (UBA15 to UBA0): These bits store the lower bit values (bits 15 to 0) of the address of the break condition.

8.2.2 User Break Address Mask Register (UBAMR)

UBAMRH:

Bit:	15	14	13	12	11	10	9	8
	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The user break address mask register (UBAMR) consists of user break address mask register H (UBAMRH) and user break address mask register L (UBAMRL). Both are 16-bit readable/writable registers. UBAMRH designates whether to mask any of the break address bits established in UBARH, and UBAMRL designates whether to mask any of the break address bits established in UBARL. UBAMRH and UBAMRL are initialized to H'0000 by a power-on reset and in module standby mode. They are not initialized in software standby mode.

- UBAMRH Bits 15 to 0—User Break Address Mask 31 to 16 (UBM31 to UBM16): These bits designate whether to mask the corresponding break address 31 to 16 bits (UBA31 to UBA16) established in UBARH.
- UBAMRL Bits 15 to 0—User Break Address Mask 15 to 0 (UBM15 to UBM0): These bits designate whether to mask the corresponding break address 15 to 0 bits (UBA15 to UBA0) established in UBARL.

Bit 15–0: UBMn	**Description**
0	Break address UBAn is included in the break conditions (Initial value)
1	Break address UBAn is not included in the break conditions

Note: n = 31 to 0

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The user break bus cycle register (UBBR) is a 16-bit readable/writable register that selects from among the following four break conditions:

1. CPU cycle/DMA cycle
2. Instruction fetch/data access
3. Read/write
4. Operand size (byte, word, longword)

UBBR is initialized to H'0000 by a power on reset and in module standby mode. It is not initialized in software standby mode.

- Bits 15 to 8—Reserved: These bits always read 0. The write value should always be 0.
- Bits 7 and 6—CPU Cycle/DMA Cycle Select (CP1, CP0): These bits designate break conditions for CPU cycles or DMA cycles.

Bit 7: CP1	Bit 6: CP0	Description	
0	0	No user break interrupt occurs	(Initial value)
	1	Break on CPU cycles	
1	0	Break on DMA cycles	
	1	Break on both CPU and DMA cycles	

Bit 5: ID1	Bit 4: ID0	Description	
0	0	No user break interrupt occurs	(Initial value)
	1	Break on instruction fetch cycles	
1	0	Break on data access cycles	
	1	Break on both instruction fetch and data access cycles	

- Bits 3 and 2—Read/Write Select (RW1, RW0): These bits select whether to break on read and/or write cycles.

Bit 3: RW1	Bit 2: RW0	Description	
0	0	No user break interrupt occurs	(Initial value)
	1	Break on read cycles	
1	0	Break on write cycles	
	1	Break on both read and write cycles	

- Bits 1 and 0—Operand Size Select (SZ1, SZ0): These bits select operand size as a break condition.

Bit 1: SZ1	Bit 0: SZ0	Description	
0	0	Operand size is not a break condition	(Initial value)
	1	Break on byte access	
1	0	Break on word access	
	1	Break on longword access	

Note: When breaking on an instruction fetch, clear the SZ0 bit to 0. All instructions are considered to be word-size accesses (even when there are instructions in on-chip memory and two instruction fetches are performed simultaneously in one bus cycle).

Operand size is word for instructions or determined by the operand size specified for the CPU/DMAC data access. It is not determined by the bus width of the space being accessed.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	CKS1	CKS0	UBID
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

The user break control register (UBCR) is a 16-bit readable/writable register that (1) enables or disables user break interrupts and (2) sets the pulse width of the $\overline{\text{UBCTRG}}$ signal output in the event of a break condition match.

UBCR is initialized to H'0000 by a power-on reset and in module standby mode. It is not initialized in software standby mode.

- Bits 15 to 3—Reserved: These bits always read 0. The write value should always be 0.
- Bits 2 and 1—Clock Select 1 and 0 (CKS1, CKS0): These bits specify the pulse width of the $\overline{\text{UBCTRG}}$ signal output in the event of a condition match.

Bit 2: CKS1	Bit 1: CKS0	Description	
0	0	$\overline{\text{UBCTRG}}$ pulse width is ϕ	(Initial value)
	1	$\overline{\text{UBCTRG}}$ pulse width is $\phi/4$	
1	0	$\overline{\text{UBCTRG}}$ pulse width is $\phi/8$	
	1	$\overline{\text{UBCTRG}}$ pulse width is $\phi/16$	

Note: ϕ: Internal clock

- Bit 0—User Break Disable (UBID): Enables or disables user break interrupt request generation in the event of a user break condition match.

Bit 0: UBID	Description	
0	User break interrupt request is enabled	(Initial value)
1	User break interrupt request is disabled	

8.3.1 Flow of the User Break Operation

The flow from setting of break conditions to user break interrupt exception processing is described below:

1. The user break addresses are set in the user break address register (UBAR), the desired masked bits in the addresses are set in the user break address mask register (UBAMR) and the breaking bus cycle type is set in the user break bus cycle register (UBBR). If even one of the three groups of the UBBR's CPU cycle/DMA cycle select bits (CP1, CP0), instruction fetch/data access select bits (ID1, ID0), and read/write select bits (RW1, RW0) is set to 00 (no user break generated), no user break interrupt will be generated even if all other conditions are in agreement. When using user break interrupts, always be certain to establish bit conditions for all of these three groups.
2. The UBC uses the method shown in figure 8.2 to judge whether set conditions have been fulfilled. When the set conditions are satisfied, the UBC sends a user break interrupt request signal to the interrupt controller (INTC). At the same time, a condition match signal is output at the $\overline{\text{UBCTRG}}$ pin with the pulse width set in bits CKS1 and CKS0.
3. The interrupt controller checks the accepted user break interrupt request signal's priority level. The user break interrupt has priority level 15, so it is accepted only if the interrupt mask level in bits I3–I0 in the status register (SR) is 14 or lower. When the I3–I0 bit level is 15, the user break interrupt cannot be accepted but it is held pending until user break interrupt exception processing can be carried out. Consequently, user break interrupts within NMI exception service routines cannot be accepted, since the I3–I0 bit level is 15. However, if the I3–I0 bit level is changed to 14 or lower at the start of the NMI exception service routine, user break interrupts become acceptable thereafter. See section 7, Interrupt Controller (INTC), describes the handling of priority levels in greater detail.
4. The INTC sends the user break interrupt request signal to the CPU, which begins user break interrupt exception processing upon receipt. See section 7.4, Interrupt Operation, for details on interrupt exception processing.

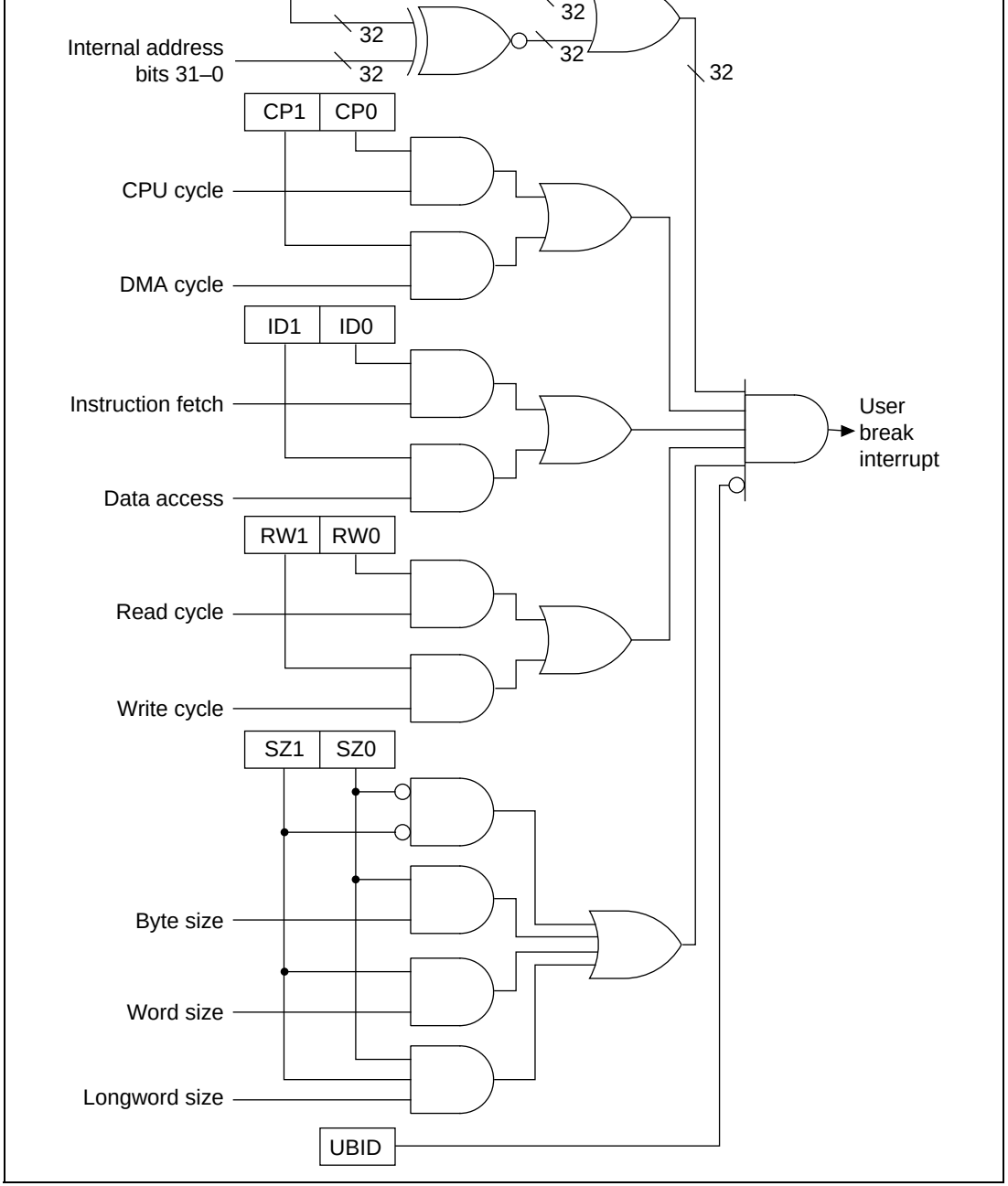


Figure 8.2 Break Condition Judgment Method

On-chip memory (on-chip ROM and/or RAM) is always accessed as 32 bits in one bus cycle. Therefore, two instructions can be retrieved in one bus cycle when fetching instructions from on-chip memory. At such times, only one bus cycle is generated, but by setting the start addresses of both instructions in the user break address register (UBAR) it is possible to cause independent breaks. In other words, when wanting to effect a break using the latter of two addresses retrieved in one bus cycle, set the start address of that instruction in UBAR. The break will occur after execution of the former instruction.

8.3.3 Program Counter (PC) Values Saved

Break on Instruction Fetch: The program counter (PC) value saved to the stack in user break interrupt exception processing is the address that matches the break condition. The user break interrupt is generated before the fetched instruction is executed. If a break condition is set in an instruction fetch cycle placed immediately after a delayed branch instruction (delay slot), or on an instruction that follows an interrupt-disabled instruction, however, the user break interrupt is not accepted immediately, but the break condition establishing instruction is executed. The user break interrupt is accepted after execution of the instruction that has accepted the interrupt. In this case, the PC value saved is the start address of the instruction that will be executed after the instruction that has accepted the interrupt.

Break on Data Access (CPU/DMA): The program counter (PC) value is the top address of the next instruction after the last instruction executed before the user break exception processing started. When data access (CPU/DMA) is set as a break condition, the place where the break will occur cannot be specified exactly. The break will occur at the instruction fetched close to where the data access that is to receive the break occurs.

8.4 Examples of Use

8.4.1 Break on CPU Instruction Fetch Cycle

1. Register settings: UBARH = H'0000
   UBARL = H'0404
   UBBR = H'0054
   UBCR = H'0000

   Conditions set: Address: H'00000404
   Bus cycle: CPU, instruction fetch, read
   (operand size not included in conditions)
   Interrupt requests enabled

   A user break interrupt will occur before the instruction at address H'00000404. If it is possible for the instruction at H'00000402 to accept an interrupt, the user break exception processing

2. Register settings: UBARH = H'0015
   UBARL = H'389C
   UBBR = H'0058
   UBCR = H'0000

   Conditions set: Address: H'0015389C
   Bus cycle: CPU, instruction fetch, write
   (operand size not included in conditions)
   Interrupt requests enabled

   A user break interrupt does not occur because the instruction fetch cycle is not a write cycle.

3. Register settings: UBARH = H'0003
   UBARL = H'0147
   UBBR = H'0054
   UBCR = H'0000

   Conditions set: Address: H'00030147
   Bus cycle: CPU, instruction fetch, read
   (operand size not included in conditions)
   Interrupt requests enabled

   A user break interrupt does not occur because the instruction fetch was performed for an even address. However, if the first instruction fetch address after the branch is an odd address set by these conditions, user break interrupt exception processing will be carried out after address error exception processing.

8.4.2 Break on CPU Data Access Cycle

1. Register settings: UBARH = H'0012
   UBARL = H'3456
   UBBR = H'006A
   UBCR = H'0000

   Conditions set: Address: H'00123456
   Bus cycle: CPU, data access, write, word
   Interrupt requests enabled

   A user break interrupt occurs when word data is written into address H'00123456.

2. Register settings: UBARH = H'00A8
   UBARL = H'0391
   UBBR = H'0066
   UBCR = H'0000

   Conditions set: Address: H'00A80391
   Bus cycle: CPU, data access, read, word
   Interrupt requests enabled

8.4.3 Break on DMA Cycle

1. Register settings: UBARH = H'0076
   UBARL = H'BCDC
   UBBR = H'00A7
   UBCR = H'0000

   Conditions set: Address: H'0076BCDC
   Bus cycle: DMA, data access, read, longword
   Interrupt requests enabled

   A user break interrupt occurs when longword data is read from address H'0076BCDC.
2. Register settings: UBARH = H'0023
   UBARL = H'45C8
   UBBR = H'0094
   UBCR = H'0000

   Conditions set: Address: H'002345C8
   Bus cycle: DMA, instruction fetch, read
   (operand size not included in conditions)
   Interrupt requests enabled

   A user break interrupt does not occur because no instruction fetch is performed in the DMA cycle.

8.5 Usage Notes

8.5.1 Simultaneous Fetching of Two Instructions

Two instructions may be simultaneously fetched from on-chip memory. If a break condition is set on the second of these two instructions but the contents of the UBC break condition registers are changed so as to alter the break condition immediately after the first of the two instructions is fetched, a user break interrupt will still occur when the second instruction is fetched.

8.5.2 Instruction Fetches at Branches

When a conditional branch instruction or TRAPA instruction causes a branch, the order of instruction fetching and execution is as follows:

1. When branching with a conditional branch instruction: BT and BF instructions
   When branching with a TRAPA instruction: TRAPA instruction

instruction fetch

Instruction execution order: Branch instruction execution → branch destination instruction execution

2. When branching with a delayed conditional branch instruction: BT/S and BF/S instructions

   Instruction fetch order: Branch instruction fetch → next instruction fetch (delay slot) → overrun fetch of instruction after next → branch destination instruction fetch

   Instruction execution order: Branch instruction execution → delay slot instruction execution → branch destination instruction execution

Thus, when a conditional branch instruction or TRAPA instruction causes a branch, the branch destination instruction will be fetched after an overrun fetch of the next instruction or the instruction after next. However, as the instruction that is the object of the break does not break until fetching and execution of the instruction have been confirmed, the overrun fetches described above do not become objects of a break.

If data accesses are also included as break conditions in addition to instruction fetch breaks, a break will occur because the instruction overrun fetch is also regarded as satisfying the data break condition.

8.5.3 Contention between User Break and Exception Processing

If a user break is set for the fetch of a particular instruction, and exception processing with higher priority than a user break is in contention and is accepted in the decode stage for that instruction (or the next instruction), user break exception processing may not be performed after completion of the higher-priority exception service routine (on return by RTE).

Thus, if a user break condition is applied to the branch destination instruction fetch after a branch (BRA, BRAF, BT, BF, BT/S, BF/S, BSR, BSRF, JMP, JSR, RTS, RTE, exception processing), and that branch instruction accepts exception processing with higher priority than a user break interrupt, user break exception processing is not performed after completion of the higher-priority exception service routine.

Therefore, a user break condition should not be set for the fetch of the branch destination instruction after a branch.

8.5.4 Break at Non-Delay Branch Instruction Jump Destination

When a branch instruction with no delay slot (including exception processing) jumps to the jump destination instruction on execution of the branch, a user break will not be generated even if a user break condition has been set for the first jump destination instruction fetch.

Information on internal bus condition matches monitored by the UBC is output as UBCTRG. The trigger width can be set with clock select bits 1 and 0 (CKS1, CKS0) in the user break control register (UBCR).

If a condition matches occurs again during trigger output, the $\overline{\text{UBCTRG}}$ pin continues to output a low level, and outputs a pulse of the length set in bits CKS1 and CKS0 from the cycle in which the last condition match occurs.

The trigger output conditions differ from those in the case of a user break interrupt when a CPU instruction fetch condition is satisfied. When a condition occurs in an overrun fetch instruction as described in section 8.5.2, Instruction Fetch at Branches, a user break interrupt is not requested but a trigger is output from the $\overline{\text{UBCTRG}}$ pin.

In other CPU data accesses and DMAC bus cycles, pulse output is performed under conditions similar to user break interrupt conditions.

Setting the user break interrupt disable (UBID) bit to 1 in UBCR enables trigger output to be monitored externally without requesting a user break interrupt.

8.5.6 Module Standby

After a power-on reset the UBC is in the module standby state, in which the clock supply is halted. When using the UBC, the module standby state must be cleared before making UBC register settings. Module standby is controlled by the module standby control register (MSTCR). See section 24.2.3, Module Standby Control Register (MSTCR), for further details.

9.1 Overview

The bus state controller (BSC) divides up the address spaces and outputs control for various types of memory. This enables memories like SRAM and ROM to be linked directly to the chip without external circuitry, simplifying system design and enabling high-speed data transfer to be achieved in a compact system.

9.1.1 Features

The BSC has the following features:

- Address space is divided into four spaces
  - A maximum linear 2 Mbytes for on-chip ROM effective mode, and a maximum 4 Mbytes for on-chip ROM disabled mode, for address space CS0
  - A maximum linear 4 Mbytes for each of address spaces CS1–CS3
  - Bus width can be selected for each space (8 or 16 bits)
  - Wait states can be inserted by software for each space
  - Wait state insertion with $\overline{\text{WAIT}}$ pin in external memory space access
  - Outputs control signals for each space according to the type of memory connected
- On-chip ROM and RAM interfaces
  - On-chip RAM access of 32 bits in 1 state
  - On-chip Rom access of 32 bits in 1 state for a read and 2 states for a write

Figure 9.1 shows the BSC block diagram.

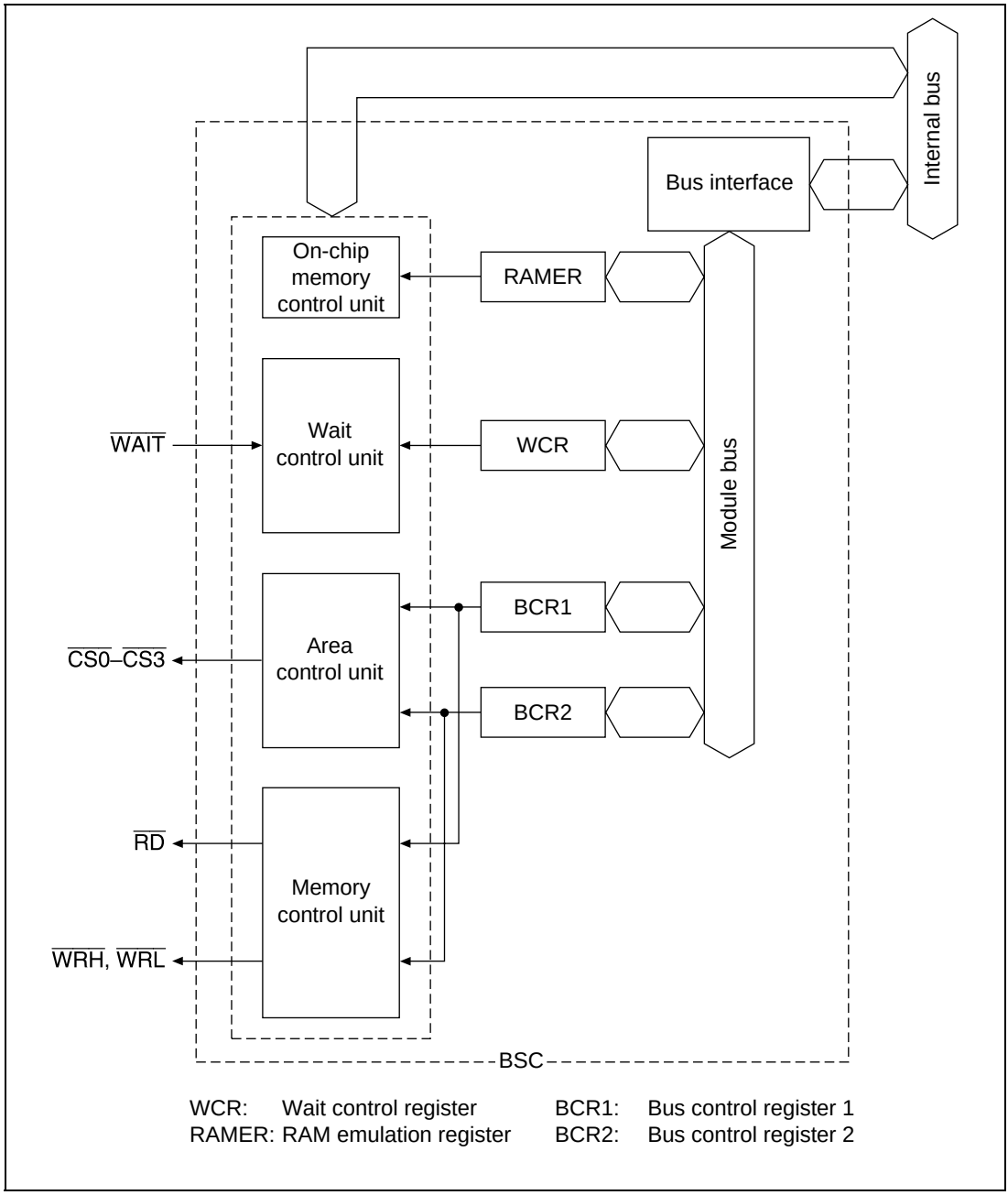


Figure 9.1 BSC Block Diagram

Table 9.1 shows the bus state controller pin configuration.

Table 9.1 Pin Configuration

Name	Abbr.	I/O	Description
Address bus	A21–A0	O	Address output
Data bus	D15–D0	I/O	16-bit data bus
Chip select	$\overline{CS0}$–$\overline{CS3}$	O	Chip select signals indicating the area being accessed
Read	$\overline{RD}$	O	Strobe that indicates the read cycle for ordinary space/multiplex I/O
Upper write	$\overline{WRH}$	O	Strobe that indicates a write cycle to the upper 8 bits (D15–D8)
Lower write	$\overline{WRL}$	O	Strobe that indicates a write cycle to the lower 8 bits (D7–D0)
Wait	$\overline{WAIT}$	I	Wait state request signal
Bus request	$\overline{BREQ}$	I	Bus release request input
Bus acknowledge	$\overline{BACK}$	O	Bus use enable output

Note: When an 8-bit bus width is selected for external space, WRL is enabled.
When a 16-bit bus width is selected for external space, $\overline{WRH}$ and $\overline{WRL}$ are enabled.

9.1.4 Register Configuration

The BSC has four registers. These registers are used to control wait states, bus width, and interfaces with memories like ROM and SRAM, as well as refresh control. The register configurations are listed in table 9.2.

All registers are 16 bits. All BSC registers are all initialized by a power-on reset and in hardware standby mode. Values are retained in a manual reset and in software standby mode.

Table 9.2 Register Configuration

Name	Abbr.	R/W	Initial Value	Address	Access Size
Bus control register 1	BCR1	R/W	H'000F	H'FFFFEC20	8, 16, 32
Bus control register 2	BCR2	R/W	H'FFFF	H'FFFFEC22	8, 16, 32
Wait state control register	WCR	R/W	H'7777	H'FFFFEC24	8, 16, 32
RAM emulation register	RAMER	R/W	H'0000	H'FFFFEC26	8, 16, 32

Note: In register access, three cycles are required for byte access and word access, and six cycles for longword access.

Figure 9.2 shows the address format used by the SH7055F.

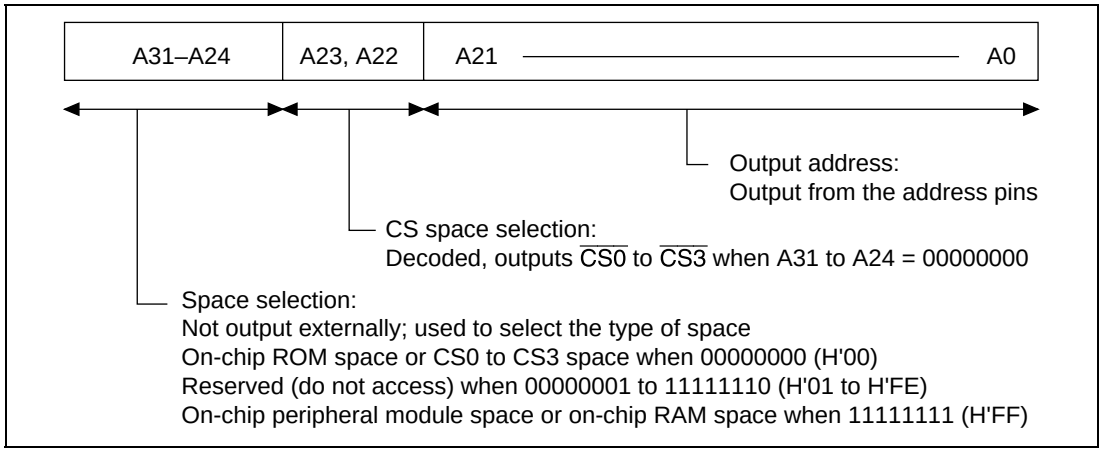


Figure 9.2 Address Format

This chip uses 32-bit addresses:

- Bits A31 to A24 are used to select the type of space and are not output externally.
- Bits A23 and A22 are decoded and output as chip select signals ($\overline{CS0}$ to $\overline{CS3}$) for the corresponding areas when bits A31 to A24 are 00000000.
- A21 to A0 are output externally.

- On-chip ROM enabled mode

Address	Space	Memory	Size	Bus Width
H'0000 0000 to H'000F FFFF	On-chip ROM	On-chip ROM	512 kB	32 bits
H'0010 0000 to H'001F FFFF	Reserved	Reserved		
H'0020 0000 to H'003F FFFF	CS0 space	External space	2 MB	8, 16 bits*[1]
H'0040 0000 to H'007F FFFF	CS1 space	External space	4 MB	8, 16 bits*[1]
H'0080 0000 to H'00BF FFFF	CS2 space	External space	4 MB	8, 16 bits*[1]
H'00C0 0000 to H'00FF FFFF	CS3 space	External space	4 MB	8, 16 bits*[1]
H'0100 0000 to H'FFFE FFFF	Reserved	Reserved		
H'FFFF 0000 to H'FFFF BFFF	On-chip RAM	On-chip RAM	32 kB	32 bits
H'FFFF C000 to H'FFFF FFFF	On-chip peripheral module	On-chip peripheral module	8 kB	8, 16 bits

- On-chip ROM disabled mode

Address	Space	Memory	Size	Bus Width
H'0000 0000 to H'003F FFFF	CS0 space	External space	4 MB	8, 16 bits*[2]
H'0040 0000 to H'007F FFFF	CS1 space	External space	4 MB	8, 16 bits*[1]
H'0080 0000 to H'00BF FFFF	CS2 space	External space	4 MB	8, 16 bits*[1]
H'00C0 0000 to H'00FF FFFF	CS3 space	External space	4 MB	8, 16 bits*[1]
H'0100 0000 to H'FFFE FFFF	Reserved	Reserved		
H'FFFF 0000 to H'FFFF BFFF	On-chip RAM	On-chip RAM	32 kB	32 bits
H'FFFF C000 to H'FFFF FFFF	On-chip peripheral module	On-chip peripheral module	8 kB	8, 16 bits

Notes: *1. Selected by on-chip register (BCR1) settings.

*2. Selected by the mode pin.

Do not access reserved spaces. Operation cannot be guaranteed if they are accessed.

9.2.1 Bus Control Register 1 (BCR1)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	A3SZ	A2SZ	A1SZ	A0SZ
Initial value:	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

BCR1 is a 16-bit readable/writable register that specifies the bus size of the CS spaces.

Write bits 15–0 of BCR1 during the initialization stage after a power-on reset, and do not change the values thereafter. In on-chip ROM enabled mode, do not access any of the CS spaces until after completion of register initialization. In on-chip ROM disabled mode, do not access any CS space other than CS0 until after completion of register initialization.

BCR1 is initialized to H'000F by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

- Bits 15–4—Reserved: The write value should always be 0. Operation cannot be guaranteed if 1 is written to these bits.
- Bit 3—CS3 Space Size Specification (A3SZ): Specifies the CS3 space bus size. A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 3: A3SZ	Description	
0	Byte (8-bit) size	
1	Word (16-bit) size	(Initial value)

- Bit 2—CS2 Space Size Specification (A2SZ): Specifies the CS2 space bus size. A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 2: A2SZ	Description	
0	Byte (8-bit) size	
1	Word (16-bit) size	(Initial value)

Bit 1: A1SZ	Description	
0	Byte (8-bit) size	
1	Word (16-bit) size	(Initial value)

- Bit 0—CS0 Space Size Specification (A0SZ): Specifies the CS0 space bus size A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 0: A0SZ	Description	
0	Byte (8-bit) size	
1	Word (16-bit) size	(Initial value)

Note: A0SZ is valid only in on-chip ROM enabled mode. In on-chip ROM disabled mode, the CS0 space bus size is specified by the mode pin.

9.2.2 Bus Control Register 2 (BCR2)

Bit:	15	14	13	12	11	10	9	8
	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCR2 is a 16-bit readable/writable register that specifies the number of idle cycles and $\overline{CS}$ signal assert extension of each CS space.

BCR2 is initialized to H'FFFF by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

- Bits 15–8—Idles between Cycles (IW31, IW30, IW21, IW20, IW11, IW10, IW01, IW00): These bits specify idle cycles inserted between consecutive accesses when the second one is to a different CS area after a read. Idles are used to prevent data conflict between ROM (and other memories, which are slow to turn the read data buffer off), fast memories, and I/O interfaces. Even when access is to the same area, idle cycles must be inserted when a read access is followed immediately by a write access. The idle cycles to be inserted comply with

IW31, IW30 specify the idle between cycles for CS3 space; IW21, IW20 specify the idle between cycles for CS2 space; IW11, IW10 specify the idle between cycles for CS1 space and IW01, IW00 specify the idle between cycles for CS0 space.

Bit 15: IW31	Bit 14: IW30	Description	
0	0	No CS3 space idle cycle	
	1	Inserts one idle cycle	
1	0	Inserts two idle cycles	
	1	Inserts three idle cycles	(Initial value)

Bit 13: IW21	Bit 12: IW20	Description	
0	0	No CS2 space idle cycle	
	1	Inserts one idle cycle	
1	0	Inserts two idle cycles	
	1	Inserts three idle cycles	(Initial value)

Bit 11: IW11	Bit 10: IW10	Description	
0	0	No CS1 space idle cycle	
	1	Inserts one idle cycle	
1	0	Inserts two idle cycles	
	1	Inserts three idle cycles	(Initial value)

Bit 9: IW01	Bit 8: IW00	Description	
0	0	No CS0 space idle cycle	
	1	Inserts one idle cycle	
1	0	Inserts two idle cycles	
	1	Inserts three idle cycles	(Initial value)

- Bits 7–4—Idle Specification for Continuous Access (CW3, CW2, CW1, CW0): The continuous access idle specification makes insertions to clearly delineate the bus intervals by once negating the CSn signal when performing consecutive accesses to the same CS space. When a write immediately follows a read, the number of idle cycles inserted is the larger of the two values specified by IW and CW. Refer to section 9.4, Waits between Access Cycles, for details.

specifies the continuous access idles for CS0 space.

Bit 7: CW3	Description	
0	No CS3 space continuous access idle cycles	
1	One CS3 space continuous access idle cycle	(Initial value)

Bit 6: CW2	Description	
0	No CS2 space continuous access idle cycles	
1	One CS2 space continuous access idle cycle	(Initial value)

Bit 5: CW1	Description	
0	No CS1 space continuous access idle cycles	
1	One CS1 space continuous access idle cycle	(Initial value)

Bit 4: CW0	Description	
0	No CS0 space continuous access idle cycles	
1	One CS0 space continuous access idle cycle	(Initial value)

- Bits 3–0—$\overline{\text{CS}}$ Assert Extension Specification (SW3, SW2, SW1, SW0): The $\overline{\text{CS}}$ assert cycle extension specification is for making insertions to prevent extension of the $\overline{\text{RD}}$ signal, $\overline{\text{WRH}}$ signal, or $\overline{\text{WRL}}$ signal assert period beyond the length of the $\overline{\text{CSn}}$ signal assert period. Extended cycles insert one cycle before and after each bus cycle, which simplifies interfaces with external devices and also has the effect of extending the write data hold time. Refer to section 9.3.3, $\overline{\text{CS}}$ Assert Period Extension, for details.

  SW3 specifies the $\overline{\text{CS}}$ assert extension for CS3 space access; SW2 specifies the $\overline{\text{CS}}$ assert extension for CS2 space access; SW1 specifies the $\overline{\text{CS}}$ assert extension for CS1 space access and SW0 specifies the $\overline{\text{CS}}$ assert extension for CS0 space access.

Bit 3: SW3	Description	
0	No CS3 space $\overline{\text{CS}}$ assert extension	
1	CS3 space $\overline{\text{CS}}$ assert extension	(Initial value)

Bit 2: SW2	Description	
0	No CS2 space $\overline{\text{CS}}$ assert extension	
1	CS2 space $\overline{\text{CS}}$ assert extension	(Initial value)

1	CS1 space $\overline{CS}$ assert extension	(Initial value)

Bit 0: SW0	Description	
0	No CS0 space $\overline{CS}$ assert extension	
1	CS0 space $\overline{CS}$ assert extension	(Initial value)

Bit:	15	14	13	12	11	10	9	8
	W33	W32	W31	W30	W23	W22	W21	W20
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	W13	W12	W11	W10	W03	W02	W01	W00
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WCR is a 16-bit readable/writable register that specifies the number of wait cycles for each CS space.

WCR is initialized to H'FFFF by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

- Bits 15–12—CS3 Space Wait Specification (W33, W32, W31, W30): These bits specify the number of waits for CS3 space access.

Bit 15: W33	Bit 14: W32	Bit 13: W31	Bit 12: W30	Description	
0	0	0	0	No wait (external wait input disabled)	
0	0	0	1	1 wait external wait input enabled	
...					
1	1	1	1	15 wait external wait input enabled	(Initial value)

- Bits 11–8—CS2 Space Wait Specification (W23, W22, W21, W20): These bits specify the number of waits for CS2 space access.

Bit 11: W23	Bit 10: W22	Bit 9: W21	Bit 8: W20	Description	
0	0	0	0	No wait (external wait input disabled)	
0	0	0	1	1 wait external wait input enabled	
...					
1	1	1	1	15 wait external wait input enabled	(Initial value)

- Bits 7–4—CS1 Space Wait Specification (W13, W12, W11, W10): These bits specify the number of waits for CS1 space access.

W13	W12	W11	W10	Description	
0	0	0	0	No wait (external wait input disabled)	
0	0	0	1	1 wait external wait input enabled	
...					
1	1	1	1	15 wait external wait input enabled	(Initial value)

- Bits 3–0—CS0 Space Wait Specification (W03, W02, W01, W00): These bits specify the number of waits for CS0 space access.

Bit 3: W03	Bit 2: W02	Bit 1: W01	Bit 0: W00	Description	
0	0	0	0	No wait (external wait input disabled)	
0	0	0	1	1 wait external wait input enabled	
...					
1	1	1	1	15 wait external wait input enabled	(Initial value)

9.2.4 RAM Emulation Register (RAMER)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

The RAM emulation register (RAMER) is a 16-bit readable/writable register that selects the RAM area to be used when emulating realtime programming of flash memory.

RAMER is initialized to H'0000 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

- Bits 15 to 4—Reserved: Only 0 should be written to these bits. Operation cannot be guaranteed if 1 is written.
- Bit 3—RAM Select (RAMS): Used together with bits 2 to 0 to select or deselect flash memory emulation by RAM (table 9.4).

  When 1 is written to this bit, all flash memory blocks are write/erase-protected.

  This bit is ignored in modes with on-chip ROM disabled.
- Bits 2 to 0—RAM Area Specification (RAM2 to RAM0): These bits are used together with the RAMS bit to designate the flash memory area to be overlapped onto RAM (table 9.4).

Table 9.4 RAM Area Setting Method

RAM Area	Bit 3: RAMS	Bit 2: RAM2	Bit 1: RAM1	Bit 0: RAM0
H'FFFF6000 to H'FFFF6FFF	0	*	*	*
H'00000000 to H'00000FFF	1	0	0	0
H'00001000 to H'00001FFF	1	0	0	1
H'00002000 to H'00002FFF	1	0	1	0
H'00003000 to H'00003FFF	1	0	1	1
H'00004000 to H'00004FFF	1	1	0	0
H'00005000 to H'00005FFF	1	1	0	1
H'00006000 to H'00006FFF	1	1	1	0
H'00007000 to H'00007FFF	1	1	1	1

*: Don’t care

A strobe signal is output in external space accesses to provide primarily for SRAM or ROM direct connections.

9.3.1 Basic Timing

Figure 9.3 shows the basic timing of external space access. External access bus cycles are performed in 2 states.

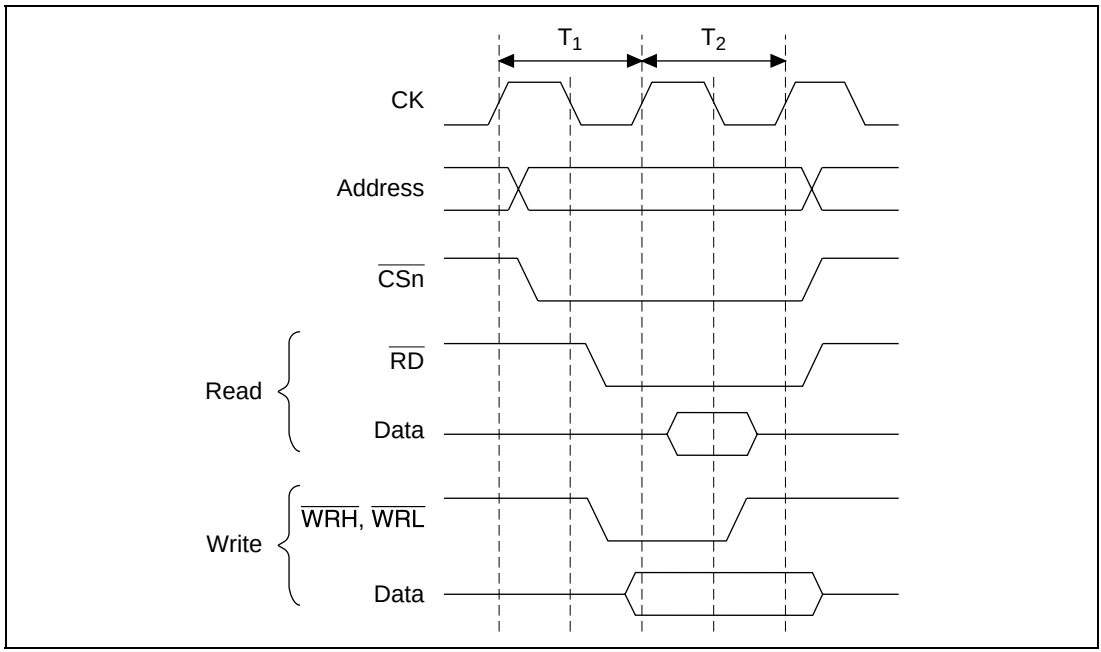


Figure 9.3 Basic Timing of External Space Access

The number of wait states inserted into external space access states can be controlled using the WCR settings (figure 9.4). The specified number of T_w cycles are inserted as software cycles at the timing shown in figure 9.4.

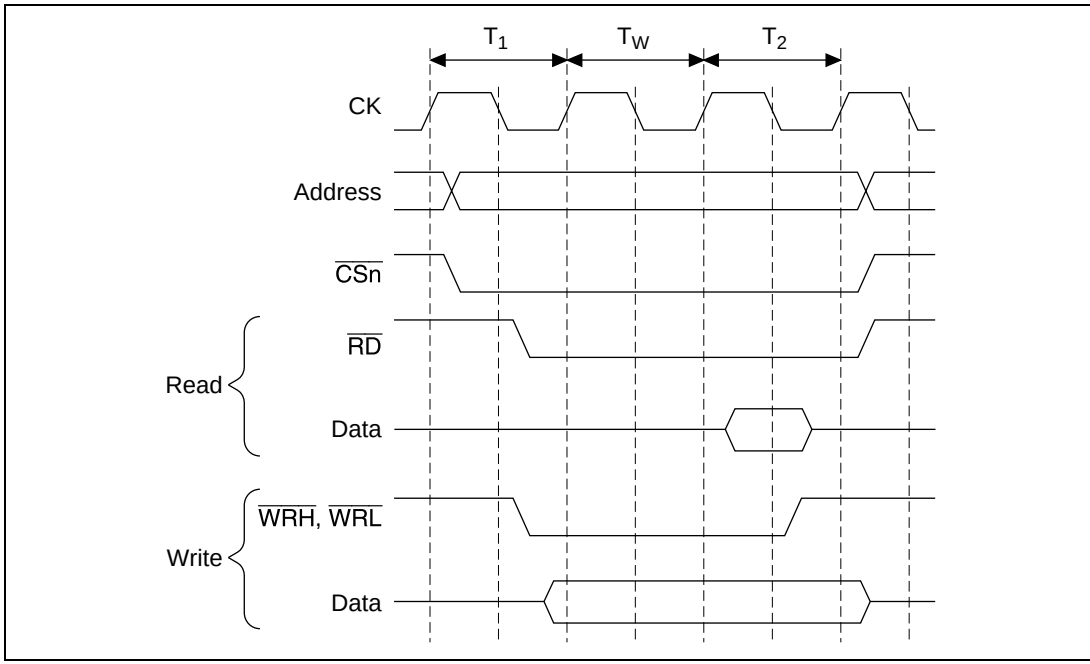


Figure 9.4 Wait State Timing of External Space Access (Software Wait Only)

rise one cycle before the clock rise when the T_w state shifts to the T_2 state. When using external waits, use a WCR setting of 1 state or more when extending $\overline{CS}$ assertion, and 2 states or more otherwise.

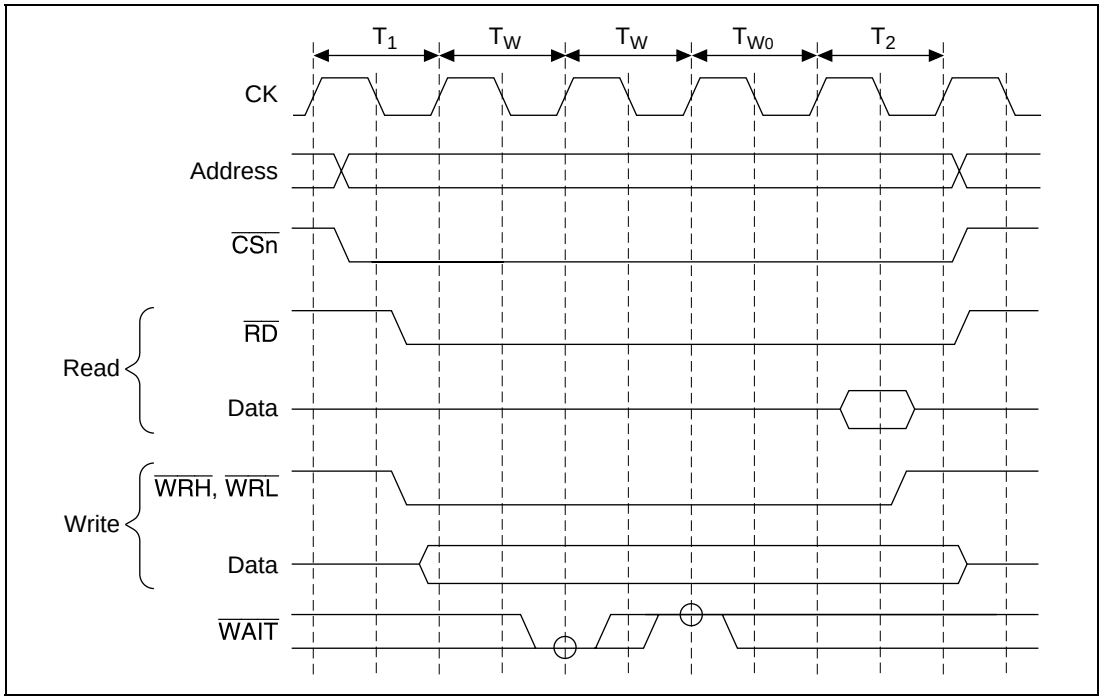


Figure 9.5 Wait State Timing of External Space Access (Two Software Wait States + $\overline{WAIT}$ Signal Wait State)

beyond the length of the $\overline{CSn}$ signal assert period by setting the SW3–SW0 bits of BCR2. This allows for flexible interfaces with external circuitry. The timing is shown in figure 9.6. T_h and T_f cycles are added respectively before and after the ordinary cycle. Only $\overline{CSn}$ is asserted in these cycles; $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ signals are not. Further, data is extended up to the T_f cycle, which is effective for gate arrays and the like, which have slower write operations.

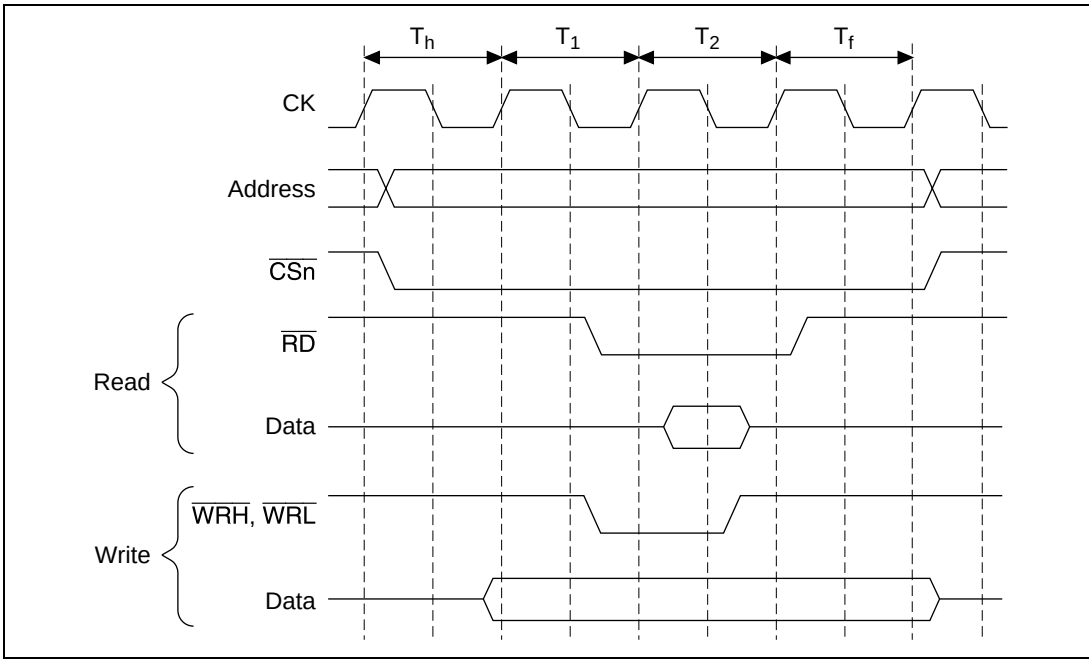


Figure 9.6 $\overline{CS}$ Assert Period Extension Function

When a read from a slow device is completed, data buffers may not go off in time to prevent data conflicts with the next access. If there is a data conflict during memory access, the problem can be solved by inserting a wait in the access cycle.

To enable detection of bus cycle starts, waits can be inserted between access cycles during continuous accesses of the same CS space by negating the $\overline{\text{CSn}}$ signal once.

9.4.1 Prevention of Data Bus Conflicts

For the two cases of write cycles after read cycles, and read cycles for a different area after read cycles, waits are inserted so that the number of idle cycles specified by the IW31 to IW00 bits of BCR2 occur. When idle cycles already exist between access cycles, only the number of empty cycles remaining beyond the specified number of idle cycles are inserted.

Figure 9.7 shows an example of idles between cycles. In this example, one idle between CSn space cycles has been specified, so when a CSm space write immediately follows a CSn space read cycle, one idle cycle is inserted.

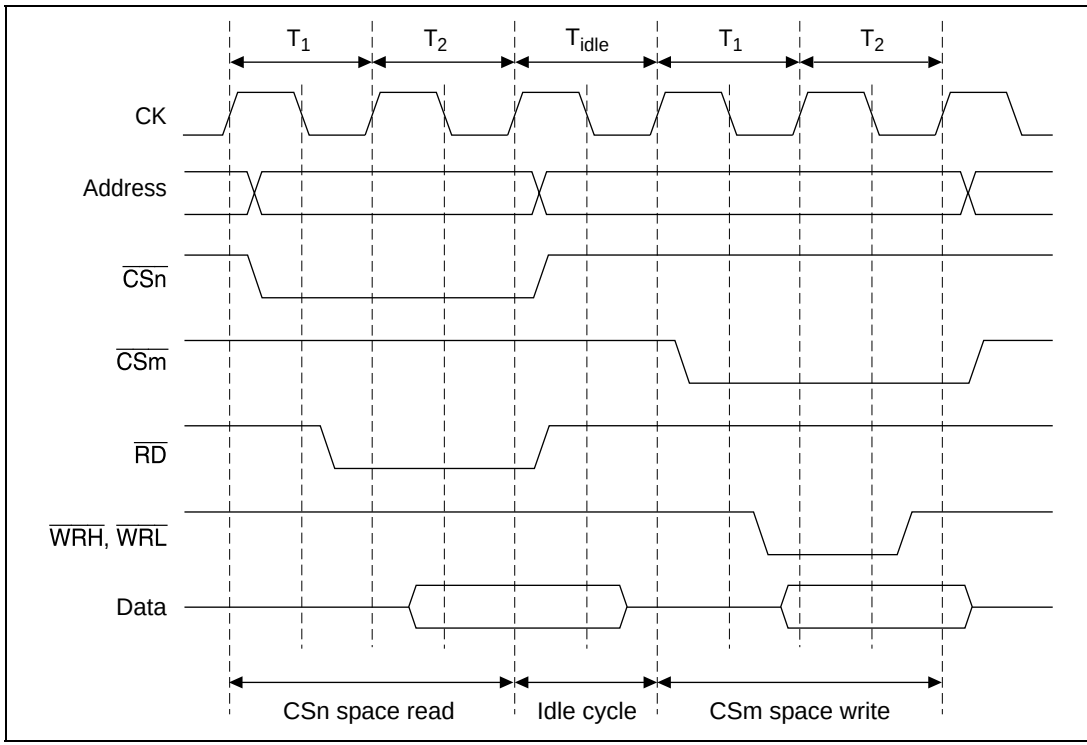


Figure 9.7 Idle Cycle Insertion Example

IW20 specify the number of idle cycles after a CS2 space read, IW11 and IW10, the number after a CS1 space read, and IW01 and IW00, the number after a CS0 space read. 0 to 3 idle cycles can be specified.

9.4.2 Simplification of Bus Cycle Start Detection

For consecutive accesses to the same CS space, waits are inserted to provide the number of idle cycles designated by bits CW3 to CW0 in BCR2. However, in the case of a write cycle after a read, the number of idle cycles inserted will be the larger of the two values designated by the IW and CW bits. When idle cycles already exist between access cycles, waits are not inserted. Figure 9.8 shows an example. A continuous access idle is specified for CSn space, and CSn space is consecutively write-accessed.

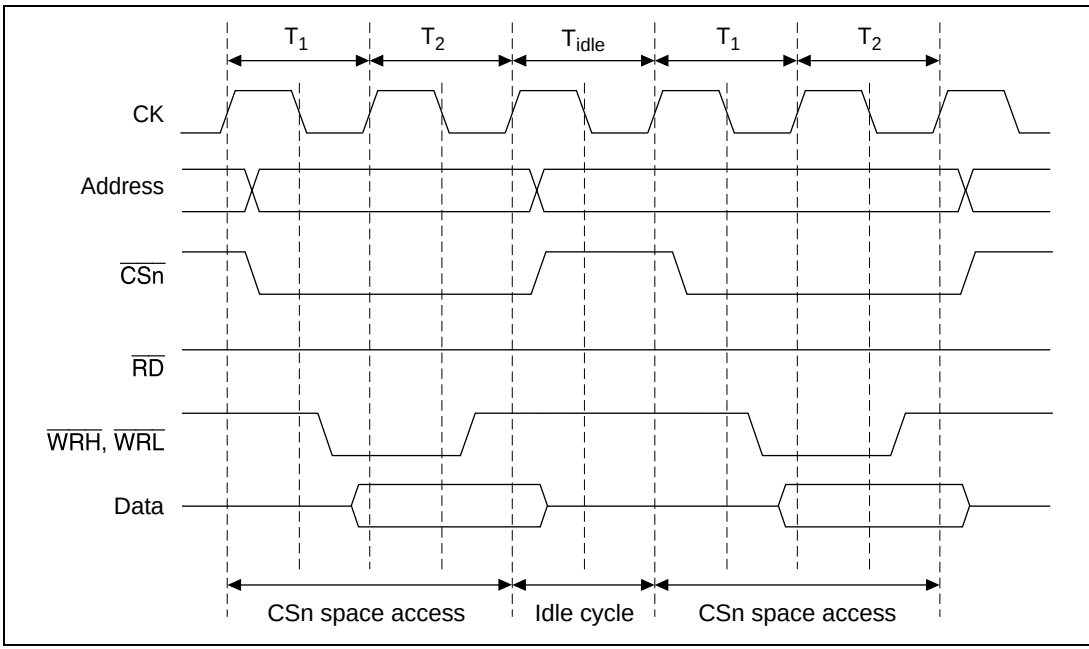


Figure 9.8 Same Space Consecutive Access Idle Cycle Insertion Example

The SH7058 has a bus arbitration function that, when a bus release request is received from an external device, releases the bus to that device. It also has three internal bus masters, the CPU, DMAC, and AUD. The priority ranking for determining bus right transfer between these bus masters is:

Bus right request from external device > AUD > DMAC > CPU

Therefore, an external device that generates a bus request is given priority even if the request is made during a DMAC burst transfer.

The AUD does not acquire the bus during DMAC burst transfer, but at the end of the transfer. When the CPU has possession of the bus, the AUD has higher priority than the DMAC for bus acquisition.

A bus request by an external device should be input at the $\overline{\text{BREQ}}$ pin. The signal indicating that the bus has been released is output from the $\overline{\text{BACK}}$ pin.

Figure 9.9 shows the bus right release procedure.

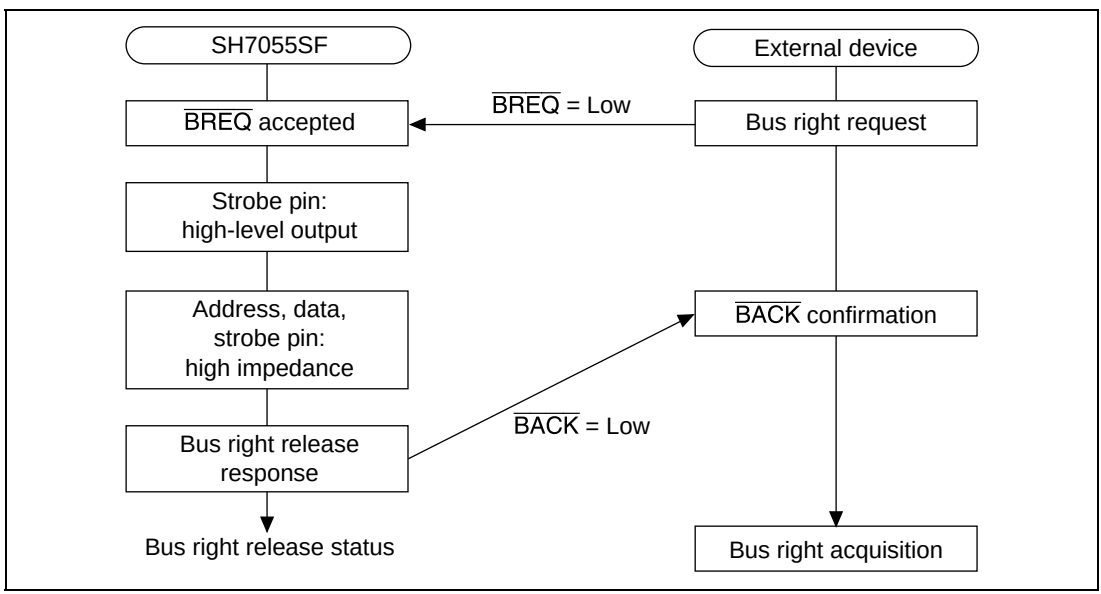


Figure 9.9 Bus Right Release Procedure

Figures 9.10–9.13 show examples of the memory connections.

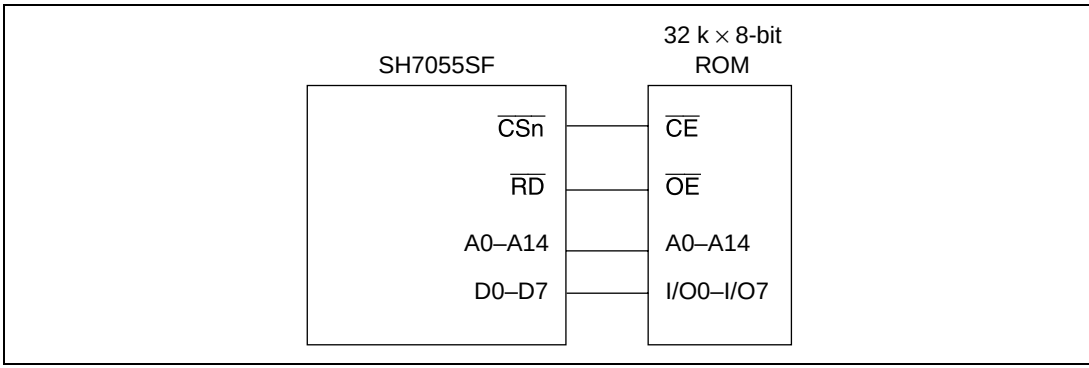


Figure 9.10 Example of 8-Bit Data Bus Width ROM Connection

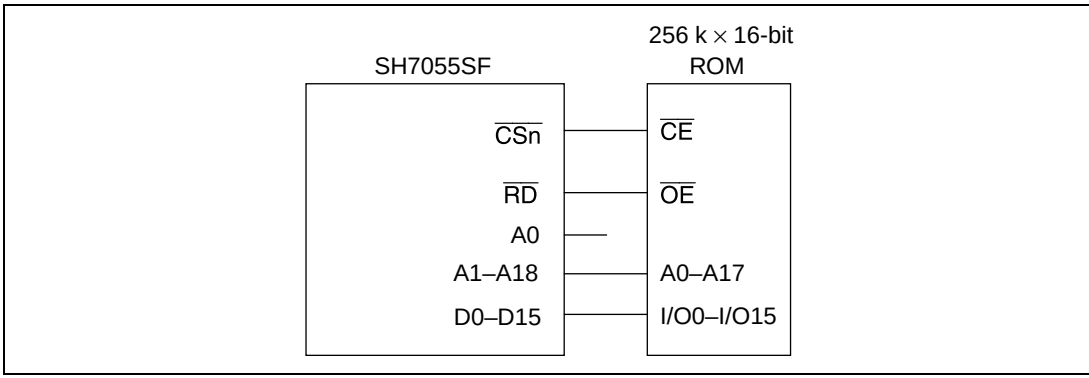


Figure 9.11 Example of 16-Bit Data Bus Width ROM Connection

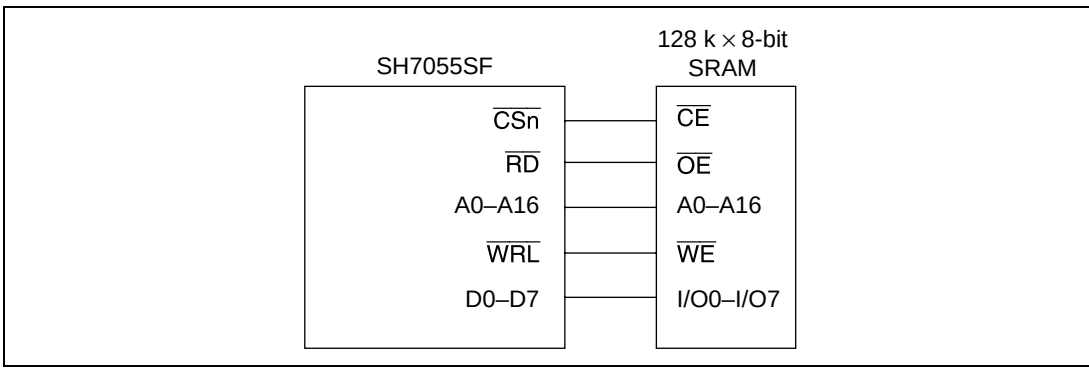


Figure 9.12 Example of 8-Bit Data Bus Width SRAM Connection

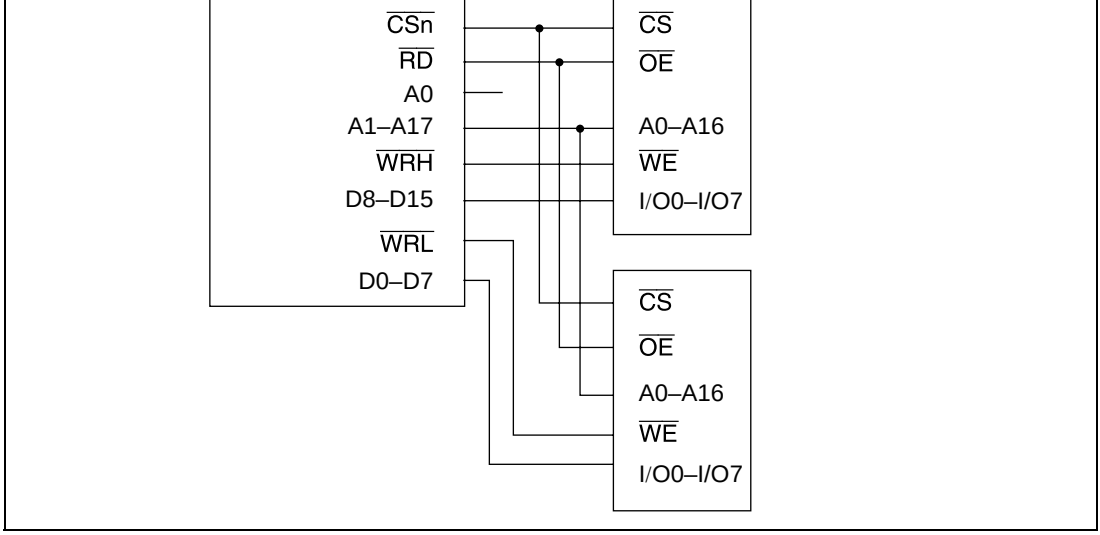


Figure 9.13 Example of 16-Bit Data Bus Width SRAM Connection

10.1 Overview

The SH7055SF includes an on-chip four-channel direct memory access controller (DMAC). The DMAC can be used in place of the CPU to perform high-speed data transfers among external memories, memory-mapped external devices, and on-chip peripheral modules (except for the DMAC, BSC, and UBC). Using the DMAC reduces the burden on the CPU and increases the operating efficiency of the chip as a whole.

10.1.1 Features

The DMAC has the following features:

- Four channels
- 4-Gbyte address space in the architecture
- 8-, 16-, or 32-bit selectable data transfer length
- Maximum of 16 M (6,777,216) transfers
- Address modes

  Both the transfer source and transfer destination are accessed by address. There are two transfer modes: direct address and indirect address.

  — Direct address transfer mode: Values set in a DMAC internal register indicate the accessed address for both the transfer source and transfer destination. Two bus cycles are required for one data transfer.

  — Indirect address transfer mode: The value stored at the location pointed to by the address set in the DMAC internal transfer source register is used as the address. Operation is otherwise the same as for direct access. This function can only be set for channel 3. Four bus cycles are required for one data transfer.
- Channel function: Dual address mode is supported on all channels.

  Channel 2 has a source address reload function that reloads the source address every fourth transfer. Direct address transfer mode or indirect address transfer mode can be specified for channel 3.
- Reload function

  Enables automatic reloading of the value set in the first source address register every fourth DMA transfer. This function can be executed on channel 2 only.
- Transfer requests

  There are two DMAC transfer activation requests, as indicated below.

  — Requests from on-chip peripheral modules: Transfer requests from on-chip modules such as the SCI or A/D. These can be received by all channels.

  — Auto-request: The transfer request is generated automatically within the DMAC.

- CPU can be interrupted when the specified number of data transfers are complete.

10.1.2 Block Diagram

Figure 10.1 is a block diagram of the DMAC.

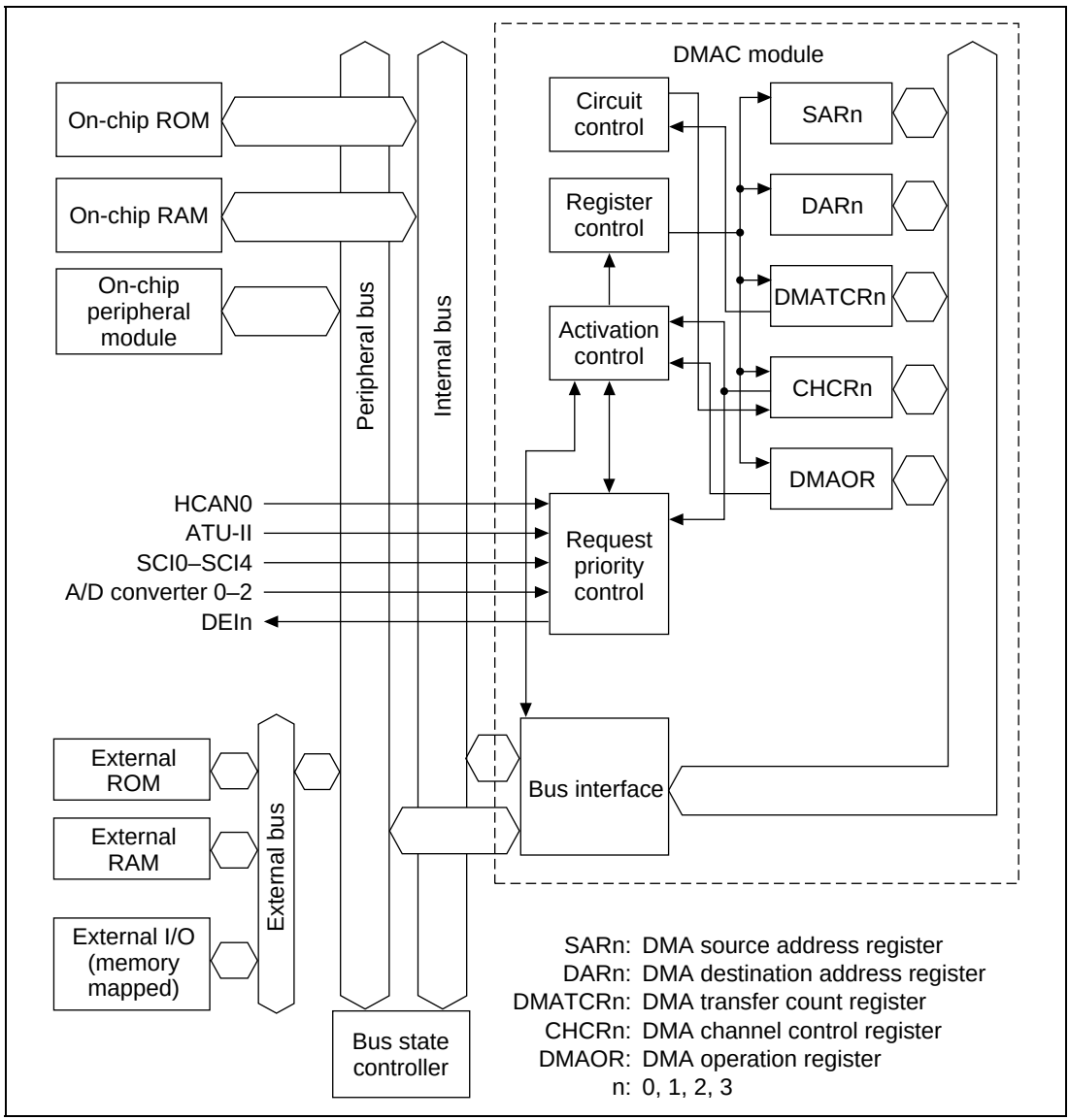


Figure 10.1 DMAC Block Diagram

Table 10.1 summarizes the DMAC registers. The DMAC has a total of 17 registers. Each channel has four registers, and one overall DMAC control register is shared by all channels.

Table 10.1 DMAC Registers

Channel	Name	Abbr.	R/W	Initial Value	Address	Register Size	Access Size
0	DMA source address register 0	SAR0	R/W	Undefined	H'FFFFECC0	32 bits	16, 32*[2]
	DMA destination address register 0	DAR0	R/W	Undefined	H'FFFFECC4	32 bits	16, 32*[2]
	DMA transfer count register 0	DMATCR0	R/W	Undefined	H'FFFFECC8	32 bits	16, 32*[2]
	DMA channel control register 0	CHCR0	R/W*[1]	H'00000000	H'FFFFECCC	32 bits	16, 32*[2]
1	DMA source address register 1	SAR1	R/W	Undefined	H'FFFFECD0	32 bits	16, 32*[2]
	DMA destination address register 1	DAR1	R/W	Undefined	H'FFFFECD4	32 bits	16, 32*[2]
	DMA transfer count register 1	DMATCR1	R/W	Undefined	H'FFFFECD8	32 bits	16, 32*[3]
	DMA channel control register 1	CHCR1	R/W*[1]	H'00000000	H'FFFFECDC	32 bits	16, 32*[2]
2	DMA source address register 2	SAR2	R/W	Undefined	H'FFFFECE0	32 bits	16, 32*[2]
	DMA destination address register 2	DAR2	R/W	Undefined	H'FFFFECE4	32 bits	16, 32*[2]
	DMA transfer count register 2	DMATCR2	R/W	Undefined	H'FFFFECE8	32 bits	16, 32*[3]
	DMA channel control register 2	CHCR2	R/W*[1]	H'00000000	H'FFFFECEC	32 bits	16, 32*[2]

Channel	Name	Abbr.	R/W	Value	Address	Size	Size
3	DMA source address register 3	SAR3	R/W	Undefined	H'FFFFECF0	32 bits	16, 32*2
	DMA destination address register 3	DAR3	R/W	Undefined	H'FFFFECF4	32 bits	16, 32*2
	DMA transfer count register 3	DMATCR3	R/W	Undefined	H'FFFFECF8	32 bits	16, 32*3
	DMA channel control register 3	CHCR3	R/W*1	H'00000000	H'FFFFECFC	32 bits	16, 32*2
Shared	DMA operation register	DMAOR	R/W*1	H'0000	H'FFFFECB0	16 bits	16*4

Notes: Word access to a register takes 3 cycles, and longword access 6 cycles.
Do not attempt to access an empty address, as operation canot be guaranteed if this is done.

*1 Write 0 after reading 1 in bit 1 of CHCR0–CHCR3 and in bits 1 and 2 of DMAOR to clear flags. No other writes are allowed.

*2 For 16-bit access of SAR0–SAR3, DAR0–DAR3, and CHCR0–CHCR3, the 16-bit value on the side not accessed is held.

*3 DMATCR has a 24-bit configuration: bits 0–23. Writing to the upper 8 bits (bits 24–31) is invalid, and these bits always read 0.

*4 Do not use 32-bit access on DMAOR.

10.2 Register Descriptions

10.2.1 DMA Source Address Registers 0–3 (SAR0–SAR3)

DMA source address registers 0–3 (SAR0–SAR3) are 32-bit readable/writable registers that specify the source address of a DMA transfer. These registers have a count function, and during a DMA transfer, they indicate the next source address.

Specify a 16-bit boundary when performing 16-bit data transfers, and a 32-bit boundary when performing 32-bit data transfers. Operation cannot be guaranteed if any other addresses are set.

The initial value after a power-on reset and in standby mode is undefined.

Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	…	…	2	1	0
				…	…			
Initial value:	—	—	—	…	…	—	—	—
R/W:	R/W	R/W	R/W	…	…	R/W	R/W	R/W

10.2.2 DMA Destination Address Registers 0–3 (DAR0–DAR3)

DMA destination address registers 0–3 (DAR0–DAR3) are 32-bit readable/writable registers that specify the destination address of a DMA transfer. These registers have a count function, and during a DMA transfer, they indicate the next destination address.

Specify a 16-bit boundary when performing 16-bit data transfers, and a 32-bit boundary when performing 32-bit data transfers. Operation cannot be guaranteed if any other addresses are set.

The value after a power-on reset and in standby mode is undefined.

Bit:	31	30	29	28	27	26	25	24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	23	22	21	…	…	2	1	0
				…	…			
Initial value:	—	—	—	…	…	—	—	—
R/W:	R/W	R/W	R/W	…	…	R/W	R/W	R/W

10.2.3 DMA Transfer Count Registers 0–3 (DMATCR0–DMATCR3)

DMA transfer count registers 0–3 (DMATCR0–DMATCR3) are 24-bit read/write registers that specify the transfer count for the channel (byte count, word count, or longword count) in bits 23 to 0. Specifying H'000001 gives a transfer count of 1, while H'000000 gives the maximum setting, 16,777,216 transfers. During DMAC operation, these registers indicate the remaining number of transfers.

The upper 8 bits of DMATCR always read 0. The write value, also, should always be 0.

	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	23	22	21	20	19	18	17	16
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.2.4 DMA Channel Control Registers 0–3 (CHCR0–CHCR3)

DMA channel control registers 0–3 (CHCR0–CHCR3) are 32-bit readable/writable registers that designate the operation and transmission of each channel. CHCR register bits are initialized to H'00000000 by a power-on reset and in standby mode.

Bit:	31	30	29	28	27	26	25	24
	—	—	—	DI	—	—	—	RO
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W*[2]	R	R	R	R/W*[2]

Bit:	23	22	21	20	19	18	17	16
	—	—	—	RS4	RS3	RS2	RS1	RS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W*[1]	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	TS1	TS0	TM	IE	TE	DE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/(W)*[1]	R/W

Notes: *1. TE bit: Allows only a 0 write after reading 1.
*2. The DI and RO bits may be absent, depending on the channel.

- Bits 31–29, 27–25, 23–21, 15, 14, 11, 10, 7, 6—Reserved: These bits are always read as 0, and should only be written with 0.
- Bit 28—Direct/Indirect Select (DI): Specifies either direct address mode operation or indirect address mode operation for the channel 3 source address. This bit is valid only in CHCR3. It always reads 0 in CHCR0–CHCR2, and should always be written with 0.

Bit 28: DI	Description	
0	Direct access mode operation for channel 3	(Initial value)
1	Indirect access mode operation for channel 3	

- Bit 24—Source Address Reload (RO): Selects whether to reload the source address initial value during channel 2 transfer. This bit is valid only for channel 2. It always reads 0 in CHCR0, CHCR1, and CHCR3, and should always be written with 0.

Bit 24: RO	Description	
0	Does not reload source address	(Initial value)
1	Reloads source address	

0	0	0	0	0	No request* (Initial value)
				1	SCI0 transmission
			1	0	SCI0 reception
				1	SCI1 transmission
		1	0	0	SCI1 reception
				1	SCI2 transmission
			1	0	SCI2 reception
				1	SCI3 transmission
	1	0	0	0	SCI3 reception
				1	SCI4 transmission
			1	0	SCI4 reception
				1	On-chip A/D0
		1	0	0	On-chip A/D1
				1	On-chip A/D2
			1	0	No request*
				1	HCAN0 (RM0)
1	0	0	0	0	No request*
				1	ATU-II (ICI0A)
			1	0	ATU-II (ICI0B)
				1	ATU-II (ICI0C)
		1	0	0	ATU-II (ICI0D)
				1	ATU-II (CMI6A)
			1	0	ATU-II (CMI6B)
				1	ATU-II (CMI6C)
	1	0	0	0	ATU-II (CMI6D)
				1	ATU-II (CMI7A)
			1	0	ATU-II (CMI7B)
				1	ATU-II (CMI7C)
		1	0	0	ATU-II (CMI7D)
				1	No request*
			1	0	No request*
				1	Auto-request

Note: * Setting prohibited. For details, see No.12 in section 10.5, Usage Notes.

Bit 13: SM1	Bit 12: SM0	Description	
0	0	Source address fixed	(Initial value)
0	1	Source address incremented (+1 during 8-bit transfer, +2 during 16-bit transfer, +4 during 32-bit transfer)	
1	0	Source address decremented (–1 during 8-bit transfer, –2 during 16-bit transfer, –4 during 32-bit transfer)	
1	1	Setting prohibited	

When the transfer source is specified at an indirect address, specify in source address register 3 (SAR3) the actual storage address of the data to be transferred as the data storage address (indirect address).

During indirect address mode, SAR3 obeys the SM1/SM0 setting for increment/decrement. In this case, SAR3's increment/decrement is fixed at +4/–4 or 0, irrespective of the transfer data size specified by TS1 and TS0.

- Bits 9 and 8—Destination Address Mode 1, 0 (DM1, DM0): These bits specify increment/decrement of the DMA transfer source address.

Bit 9: DM1	Bit 8: DM0	Description	
0	0	Destination address fixed	(Initial value)
0	1	Destination address incremented (+1 during 8-bit transfer, +2 during 16-bit transfer, +4 during 32-bit transfer)	
1	0	Destination address decremented (–1 during 8-bit transfer, –2 during 16-bit transfer, –4 during 32-bit transfer)	
1	1	Setting prohibited	

- Bits 5 and 4—Transfer Size 1, 0 (TS1, TS0): These bits specify the size of the data for transfer.

Bit 5: TS1	Bit 4: TS0	Description	
0	0	Specifies byte size (8 bits)	(Initial value)
0	1	Specifies word size (16 bits)	
1	0	Specifies longword size (32 bits)	
1	1	Setting prohibited	

0	Cycle-steal mode	(Initial value)
1	Burst mode	

- Bit 2—Interrupt Enable (IE): When this bit is set to 1, interrupt requests are generated after the number of data transfers specified in DMATCR (when TE = 1).

Bit 2: IE	Description
0	Interrupt request not generated on completion of DMATCR-specified number of transfers (Initial value)
1	Interrupt request enabled on completion of DMATCR-specified number of transfers

- Bit 1—Transfer End (TE): This bit is set to 1 after the number of data transfers specified by DMATCR. At this time, if the IE bit is set to 1, an interrupt request is generated.

  If data transfer ends before TE is set to 1 (for example, due to an NMI or address error, or clearing of the DE bit or DME bit of DMAOR) TE is not set to 1. With this bit set to 1, data transfer is disabled even if the DE bit is set to 1.

Bit 1: TE	Description
0	DMATCR-specified number of transfers not completed (Initial value) [Clearing condition] 0 write after TE = 1 read, power-on reset, standby mode
1	DMATCR-specified number of transfers completed

- Bit 0—DMAC Enable (DE): DE enables operation in the corresponding channel.

Bit 0: DE	Description
0	Operation of the corresponding channel disabled (Initial value)
1	Operation of the corresponding channel enabled

Transfer is initiated if this bit is set to 1 when auto-request is specified (RS4–RS0 settings). With an on-chip module request, when a transfer request occurs after this bit is set to 1, transfer is initiated. If this bit is cleared during a data transfer, transfer is suspended.

If the DE bit has been set, but TE = 1, then if the DME bit of DMAOR is 0, and the NMIF or AE bit of DMAOR is 1, the transfer enable state is not entered.

DMAOR is a 16-bit readable/writable register that controls the overall operation of the DMAC.

Register values are initialized to H'0000 by a power-on reset and in standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	AE	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W)*	R/W

Note: * 0 write only is valid after 1 is read at the AE and NMIF bits.

- Bits 15–3—Reserved: These bits are always read 0 and should always be written with 0.
- Bit 2—Address Error Flag (AE): Indicates that an address error has occurred during DMA transfer. If this bit is set during a data transfer, transfers on all channels are suspended. The CPU cannot write a 1 to the AE bit. Clearing is effected by a 0 write after a 1 read.

Bit 2: AE	Description	
0	No address error, DMA transfer enabled	(Initial value)
	[Clearing condition]	
	Write AE = 0 after reading AE = 1	
1	Address error, DMA transfer disabled	
	[Setting condition]	
	Address error due to DMAC	

channels are suspended. The CPU is unable to write a 1 to the NMIF. Clearing is effected by a 0 write after a 1 read.

Bit 1: NMIF	Description	
0	No NMI interrupt, DMA transfer enabled	(Initial value)
	[Clearing condition]	
	Write NMIF = 0 after reading NMIF = 1	
1	NMI has occurred, DMC transfer disabled	
	[Setting condition]	
	NMI interrupt occurrence	

- Bit 0—DMAC Master Enable (DME): This bit enables activation of the entire DMAC. When the DME bit and DE bit of the CHCR register for the corresponding channel are set to 1, that channel is transfer-enabled. If this bit is cleared during a data transfer, transfers on all channels are suspended.

  Even when the DME bit is set, when the TE bit of CHCR is 1, or its DE bit is 0, transfer is disabled if the NMIF or AE bit in DMAOR is set to 1.

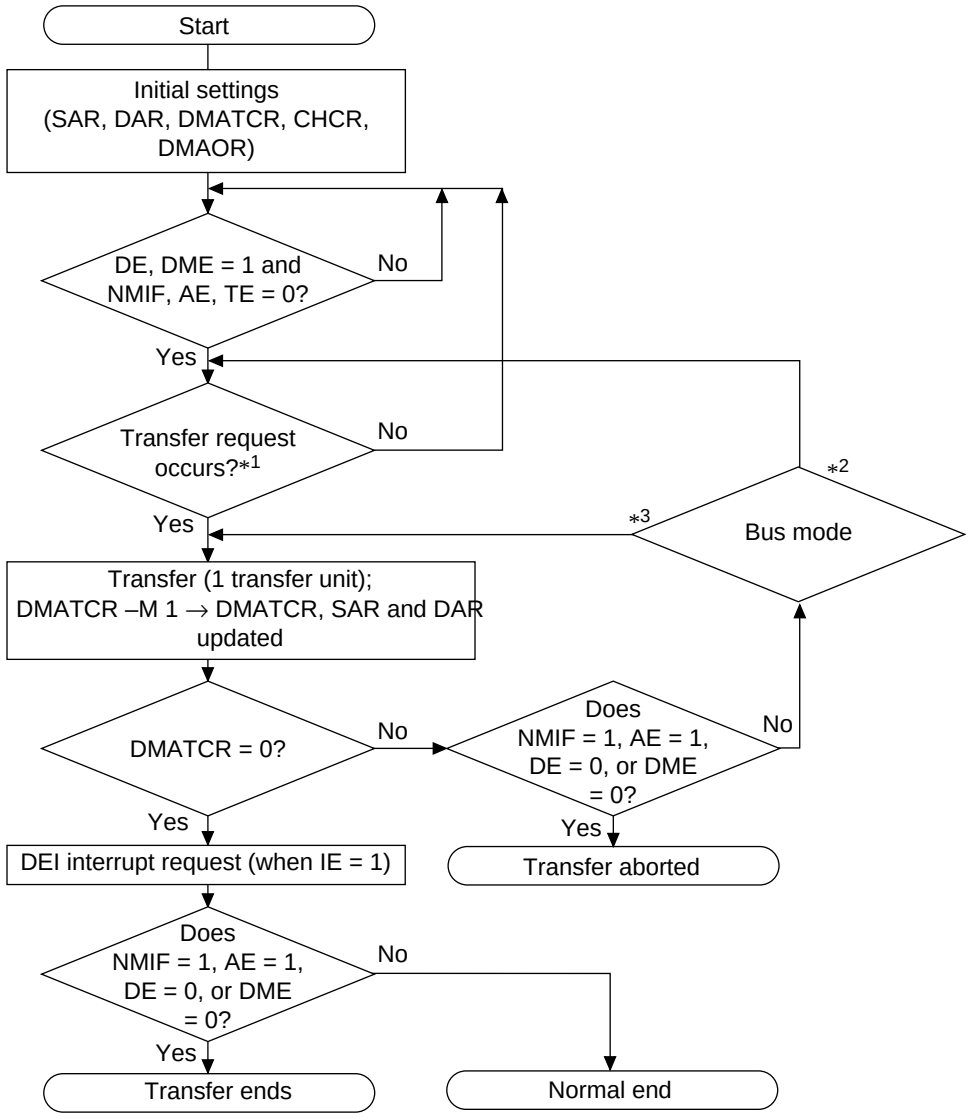
Bit 0: DME	Description	
0	Operation disabled on all channels	(Initial value)
1	Operation enabled on all channels	

When there is a DMA transfer request, the DMAC starts the transfer according to the channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in two modes: auto-request and on-chip peripheral module request. Transfer is performed only in dual address mode, and either direct or indirect address transfer mode can be used. The bus mode can be either burst or cycle-steal.

10.3.1 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count register (DMATCR), DMA channel control registers (CHCR), and DMA operation register (DMAOR) are set to the desired transfer conditions, the DMAC transfers data according to the following procedure:

1. The DMAC checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0).
2. When a transfer request comes and transfer has been enabled, the DMAC transfers 1 transfer unit of data (determined by the TS0 and TS1 setting). For an auto-request, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented by 1 upon each transfer. The actual transfer flows vary by address mode and bus mode.
3. When the specified number of transfers have been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit of CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
4. When an address error occurs in the DMAC or an NMI interrupt is generated, the transfer is aborted. Transfer is also aborted when the DE bit of CHCR or the DME bit of DMAOR is cleared to 0.



Notes: *1 In auto-request mode, transfer begins when NMIF, AE, and TE are all 0, and the DE and DME bits are set to 1.
*2 Cycle-steal mode
*3 Burst mode

Figure 10.2 DMAC Transfer Flowchart

DMA transfer requests are generated in either the data transfer source or destination. Transfers can be requested in two modes: auto-request and on-chip peripheral module request. The request mode is selected in the RS4–RS0 bits of DMA channel control registers 0–3 (CHCR0–CHCR3).

Auto-Request Mode: When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits of CHCR0–CHCR3 and the DME bit of DMAOR are set to 1, the transfer begins (so long as the TE bits of CHCR0–CHCR3 and the NMIF and AE bits of DMAOR are all 0).

On-Chip Peripheral Module Request Mode: In this mode a transfer is performed at the transfer request signal (interrupt request signal) of an on-chip peripheral module. As indicated in table 10.2, there are 26 transfer request signals: 12 from the advanced timer unit (ATU-II), which are compare match or input capture interrupts; the receive data full interrupts (RXI) and transmit data empty interrupts (TXI) of the five serial communication interfaces (SCI); the receive interrupt of HCAN0; and the A/D conversion end interrupts (ADI) of the three A/D converters. When DMA transfers are enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0), a transfer is performed upon the input of a transfer request signal.

When the transfer request is set to RXI (transfer request because the SCI's receive data register is full), the transfer source must be the SCI's receive data register (RDR). When the transfer request is set to TXI (transfer request because the SCI's transmit data register is empty), the transfer destination must be the SCI's transmit data register (TDR). If the transfer request is set to the A/D converter, the data transfer source must be the A/D converter register; if set to HCAN0, the transfer source must be HCAN0 message data.

RS4	RS3	RS2	RS1	RS0	Transfer Request Source	DMAC Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
0	0	0	0	1	SCI0 transmit block	TXI0 (SCI0 transmit-data-empty transfer request)	Donít care*	TDR0	Burst/cycle-steal
			1	0	SCI0 receive block	RXI0 (SCI0 receive-data-full transfer request)	RDR0	Donít care*	Burst/cycle-steal
				1	SCI1 transmit block	TXI1 (SCI1 transmit-data-empty transfer request)	Donít care*	TDR1	Burst/cycle-steal
		1	0	0	SCI1 receive block	RXI1 (SCI1 receive-data-full transfer request)	RDR1	Donít care*	Burst/cycle-steal
				1	SCI2 transmit block	TXI2 (SCI2 transmit-data-empty transfer request)	Donít care*	TDR2	Burst/cycle-steal
			1	0	SCI2 receive block	RXI2 (SCI2 receive-data-full transfer request)	RDR2	Donít care*	Burst/cycle-steal
				1	SCI3 transmit block	TXI3 (SCI3 transmit-data-empty transfer request)	Donít care*	TDR3	Burst/cycle-steal
	1	0	0	0	SCI3 receive block	RXI3 (SCI3 receive-data-full transfer request)	RDR3	Donít care*	Burst/cycle-steal
				1	SCI4 transmit block	TXI4 (SCI4 transmit-data-empty transfer request)	Donít care*	TDR4	Burst/cycle-steal
			1	0	SCI4 receive block	RXI4 (SCI4 receive-data-full transfer request)	RDR4	Donít care*	Burst/cycle-steal
				1	A/D0	ADI0 (A/D0 conversion end interrupt)	ADDR0–ADDR11	Donít care*	Burst/cycle-steal
		1	0	0	A/D1	ADI1 (A/D1 conversion end interrupt)	ADDR12–ADDR23	Donít care*	Burst/cycle-steal
				1	A/D2	ADI2 (A/D2 conversion end interrupt)	ADDR24–ADDR31	Donít care*	Burst/cycle-steal
			1	1	HCAN0	RM0 (HCAN0 receive interrupt)	MD0–MD15	Donít care*	Burst/cycle-steal

RS4	RS3	RS2	RS1	RS0	Transfer Request Source	DMAC Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
1	0	0	0	1	ATU-II	ICI0A (ICR0A input capture generation)	Donít care*	Donít care*	Burst/cycle-steal
			1	0	ATU-II	ICI0B (ICR0B input capture generation)	Donít care*	Donít care*	Burst/cycle-steal
				1	ATU-II	ICI0C (ICR0C input capture generation)	Donít care*	Donít care*	Burst/cycle-steal
		1	0	0	ATU-II	ICI0D (ICR0D input capture generation)	Donít care*	Donít care*	Burst/cycle-steal
				1	ATU-II	CMI6A (CYLR6A compare-match generation)	Donít care*	Donít care*	Burst/cycle-steal
			1	0	ATU-II	CMI6B (CYLR6B compare-match generation)	Donít care*	Donít care*	Burst/cycle-steal
				1	ATU-II	CMI6C (CYLR6C compare-match generation)	Donít care*	Donít care*	Burst/cycle-steal
	1	0	0	0	ATU-II	CMI6D (CYLR6D compare-match generation)	Donít care*	Donít care*	Burst/cycle-steal
				1	ATU-II	CMI7A (CYLR7A compare-match generation)	Donít care*	Donít care*	Burst/cycle-steal
			1	0	ATU-II	CMI7B (CYLR7B compare-match generation)	Donít care*	Donít care*	Burst/cycle-steal
				1	ATU-II	CMI7C (CYLR7C compare-match generation)	Donít care*	Donít care*	Burst/cycle-steal
		1	0	0	ATU-II	CMI7D (CYLR7D compare-match generation)	Donít care*	Donít care*	Burst/cycle-steal

SCI0, SCI1, SCI2, SCI3, SCI4: Serial communication interface channels 0–4

A/D0, A/D1, A/D2: A/D converter channels 0–2

HCAN0: Controller area network channel 0

ATU-II: Advanced timer unit

TDR0, TDR1, TDR2, TDR3, TDR4: SCI0–SCI4 transmit data registers

RDR0, RDR1, RDR2, RDR3, RDR4: SCI0–SCI4 receive data registers

ADDR0–ADDR11: A/D0 data registers

ADDR12–ADDR23: A/D1 data registers

ADDR24–ADDR31: A/D2 data registers

10.3.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to the following priority order:

- CH0 > CH1 > CH2 > CH3

10.3.4 DMA Transfer Types

The DMAC supports the transfers shown in table 10.3. It operates in dual address mode, in which both the transfer source and destination addresses are output. The dual address mode consists of a direct address mode, in which the output address value is the object of a direct data transfer, and an indirect address mode, in which the output address value is not the object of the data transfer, but the value stored at the output address becomes the transfer object address. The actual transfer operation timing varies with the bus mode. The DMAC has two bus modes: cycle-steal mode and burst mode.

Table 10.3 Supported DMA Transfers

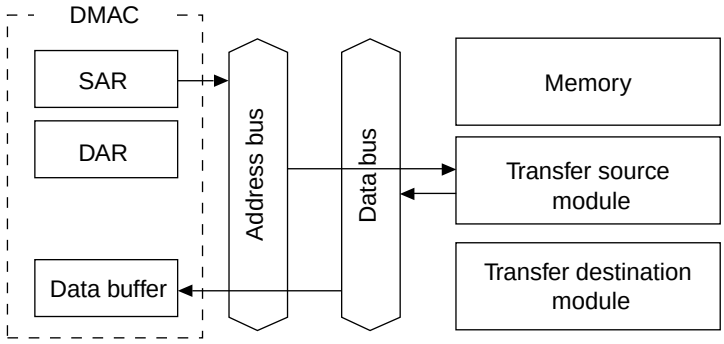
	Transfer Destination			
Transfer Source	**External Memory**	**Memory-Mapped External Device**	**On-Chip Memory**	**On-Chip Peripheral Module**
External memory	Supported	Supported	Supported	Supported
Memory-mapped external device	Supported	Supported	Supported	Supported
On-chip memory	Supported	Supported	Supported	Supported
On-chip peripheral module	Supported	Supported	Supported	Supported

10.3.5 Dual Address Mode

Dual address mode is used for access of both the transfer source and destination by address. Transfer source and destination can be accessed either internally or externally. Dual address mode is subdivided into two other modes: direct address transfer mode and indirect address transfer mode.

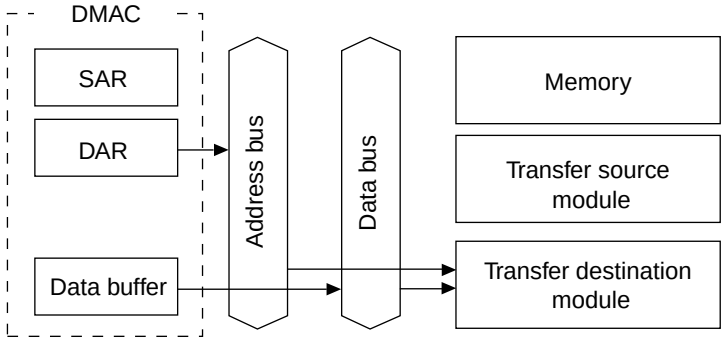
cycles. At this time, the transfer data is temporarily stored in the DMAC. With the kind of external memory transfer shown in figure 10.3, data is read from one of the memories by the DMAC during a read cycle, then written to the other external memory during the subsequent write cycle. Figure 10.4 shows the timing for this operation.

1st bus cycle



The SAR value is taken as the address, and data is read from the transfer source module and stored temporarily in the DMAC.

2nd bus cycle



The DAR value is taken as the address, and data stored in the DMAC's data buffer is written to the transfer destination module.

Figure 10.3 Direct Address Operation in Dual Address Mode

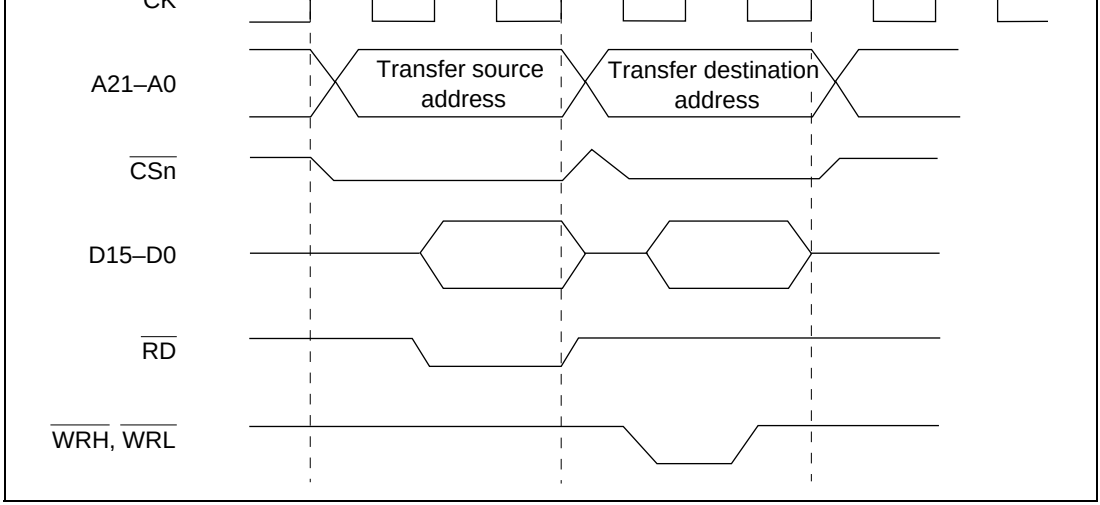
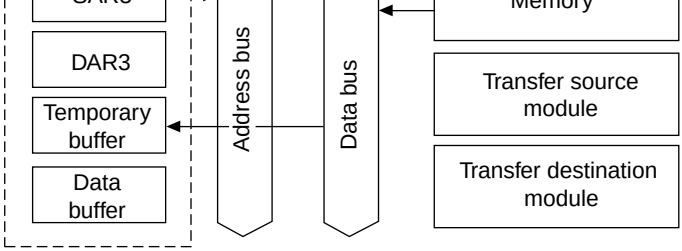


Figure 10.4 Direct Address Transfer Timing in Dual Address Mode

Indirect Address Transfer Mode: In this mode the memory address storing the data actually to be transferred is specified in the DMAC internal transfer source address register (SAR3). Therefore, in indirect address transfer mode, the DMAC internal transfer source address register value is read first. This value is first stored in the DMAC. Next, the read value is output as the address, and the value stored at that address is again stored in the DMAC. Finally, the subsequent read value is written to the address specified by the transfer destination address register, ending one cycle of DMAC transfer.

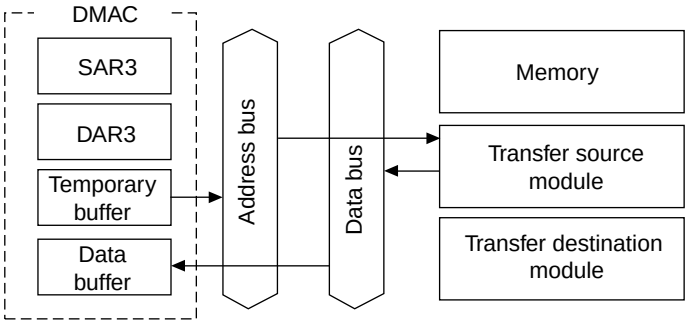
In indirect address mode (figure 10.5), the transfer destination, transfer source, and indirect address storage destination are all 16-bit external memory locations, and transfer in this example is conducted in 16-bit or 8-bit units. Timing for this transfer example is shown in figure 10.6.

In indirect address mode, one NOP cycle (figure 10.6) is required until the data read as the indirect address is output to the address bus. When transfer data is 32-bit, the third and fourth bus cycles each need to be doubled, giving a required total of six bus cycles and one NOP cycle for the whole operation.



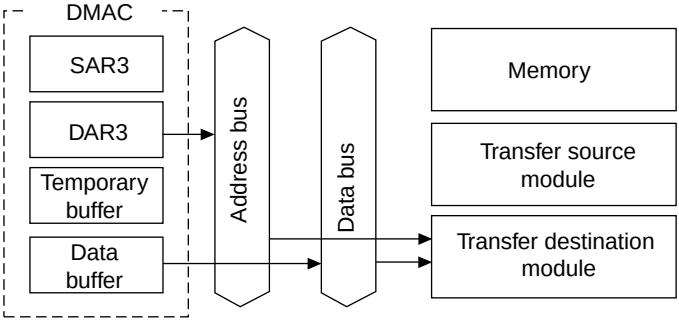
The SAR3 value is taken as the address, memory data is read, and the value is stored in the temporary buffer. Since the value read at this time is used as the address, it must be 32 bits. If data bus is 16 bits wide when accessed to an external memory space, two bus cycles are necessary.

3rd bus cycle



The value in the temporary buffer is taken as the address, and data is read from the transfer source module to the data buffer.

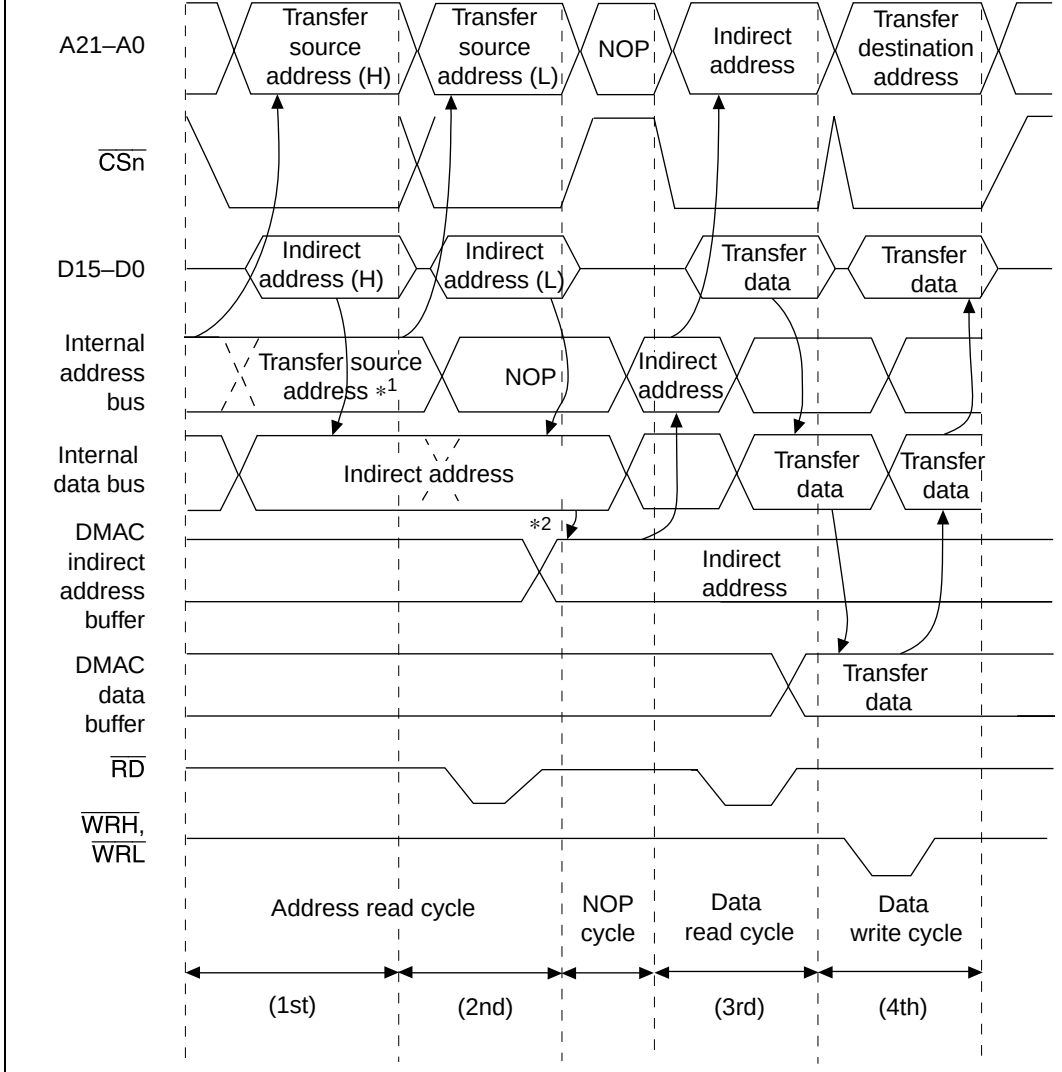
4th bus cycle



The DAR3 value is taken as the address, and the value in the data buffer is written to the transfer destination module.

Note: Memory, transfer source, and transfer destination modules are shown here.
In practice, any connection can be made as long as it is within the address space.

Figure 10.5 Dual Address Mode and Indirect Address Operation (16-Bit-Width External Memory Space)



Notes: *1 The internal address bus is controlled by the port and does not change.
*2 The DMAC does not latch the value until 32-bit data is read from the internal data bus.

Figure 10.6 Dual Address Mode and Indirect Address Transfer Timing Example 1
External Memory Space → External Memory Space
(External memory space has 16-bit width)

peripheral module with 2-cycle access space, and transfer data is 8-bit.

Since the indirect address storage destination and the transfer source are in internal memory, these can be accessed in one cycle. The transfer destination is 2-cycle access space, so two data write cycles are required. One NOP cycle is required until the data read as the indirect address is output to the address bus.

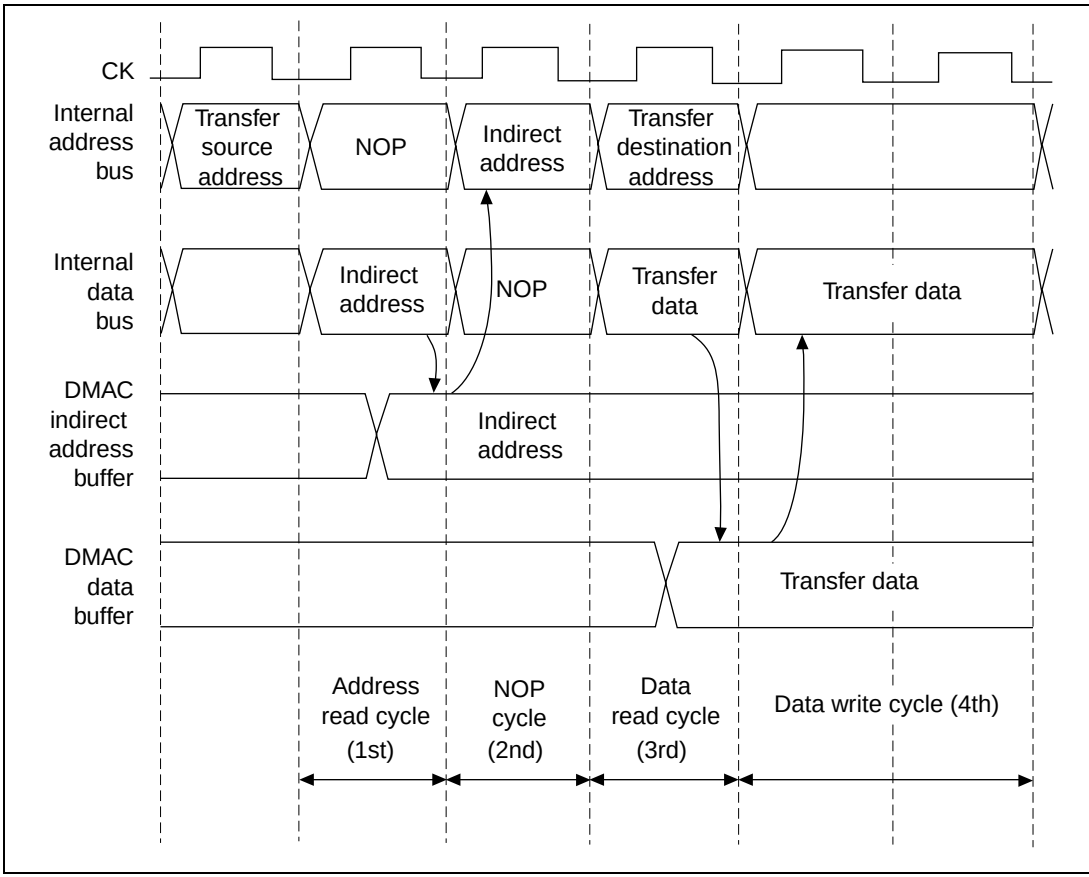


Figure 10.7 Dual Address Mode and Indirect Address Transfer Timing Example 2
Internal Memory Space → Internal Memory Space

Select the appropriate bus mode in the TM bits of CHCR0–CHCR3. There are two bus modes: cycle-steal and burst.

Cycle-Steal Mode: In cycle-steal mode, the bus right is given to another bus master after each one-transfer-unit (8-bit, 16-bit, or 32-bit) DMAC transfer. When the next transfer request occurs, the bus right is obtained from the other bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus right is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

Cycle-steal mode can be used with all categories of transfer destination, transfer source and transfer request. Figure 10.8 shows an example of DMA transfer timing in cycle-steal mode.

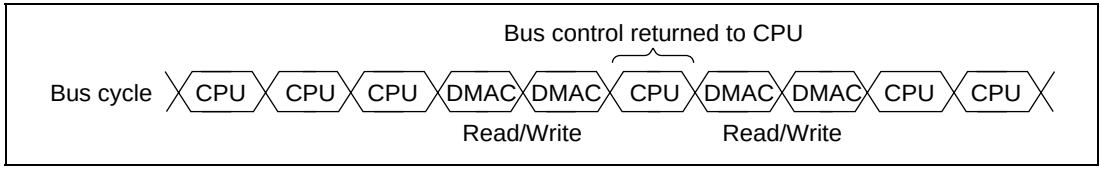


Figure 10.8 DMA Transfer Timing Example in Cycle-Steal Mode

Burst Mode: Once the bus right is obtained, transfer is performed continuously until the transfer end condition is satisfied.

Figure 10.9 shows an example of DMA transfer timing in burst mode.

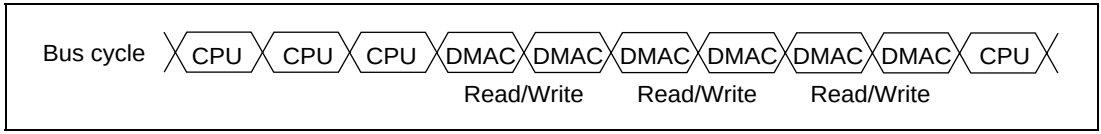


Figure 10.9 DMA Transfer Timing Example in Burst Mode

Table 10.4 shows the relationship between request modes and bus modes by DMA transfer category.

Table 10.4 Relationship between Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus*[5] Mode	Transfer Size (Bits)	Usable Channels
Dual	External memory and external memory	Any*[1]	B/C	8/16/32	0–3
	External memory and memory-mapped external device	Any*[1]	B/C	8/16/32	0–3
	Memory-mapped external device and memory-mapped external device	Any*[1]	B/C	8/16/32	0–3
	External memory and on-chip memory	Any*[1]	B/C	8/16/32	0–3
	External memory and on-chip peripheral module	Any*[2]	B/C*[3]	8/16/32*[4]	0–3
	Memory-mapped external device and on-chip memory	Any*[1]	B/C	8/16/32	0–3
	Memory-mapped external device and on-chip peripheral module	Any*[2]	B/C*[3]	8/16/32*[4]	0–3
	On-chip memory and on-chip memory	Any*[1]	B/C	8/16/32	0–3
	On-chip memory and on-chip peripheral module	Any*[2]	B/C*[3]	8/16/32*[4]	0–3
	On-chip peripheral module and on-chip†peripheral module	Any*[2]	B/C*[3]	8/16/32*[4]	0–3

B: Burst, C: Cycle-steal

Notes: *1 Auto-request or on-chip peripheral module request enabled. However, in the case of an on-chip peripheral module request, it is not possible to specify the SCI, HCAN0, or A/D converter for the transfer request source.

*2 Auto-request or on-chip peripheral module request possible. However, if the transfer request source is also the SCI, HCAN0, or A/D converter, the transfer source or transfer destination must be same as the transfer source.

*3 When the transfer request source is the SCI, only cycle-steal mode is possible.

*4 Access size permitted by the on-chip peripheral module register that is the transfer source or transfer destination.

If, for example, a transfer request is issued for channel 0 while transfer is in progress on lower priority channel 1 in burst mode, transfer is started immediately on channel 0.

In this case, if channel 0 is set to burst mode, channel 1 transfer is continued after completion of all transfers on channel 0. If channel 0 is set to cycle-steal mode, channel 1 transfer is continued only if a channel 0 transfer request has not been issued; if a transfer request is issued, channel 0 transfer is started immediately.

10.3.9 Source Address Reload Function

Channel 2 has a source address reload function. This returns to the first value set in the source address register (SAR2) every four transfers by setting the RO bit of CHCR2 to 1. Figure 10.10 illustrates this operation. Figure 10.11 is a timing chart for use of channel 2 only with the following transfer conditions set: burst mode, auto-request, 16-bit transfer data size, SAR2 incremented, DAR2 fixed, reload function on.

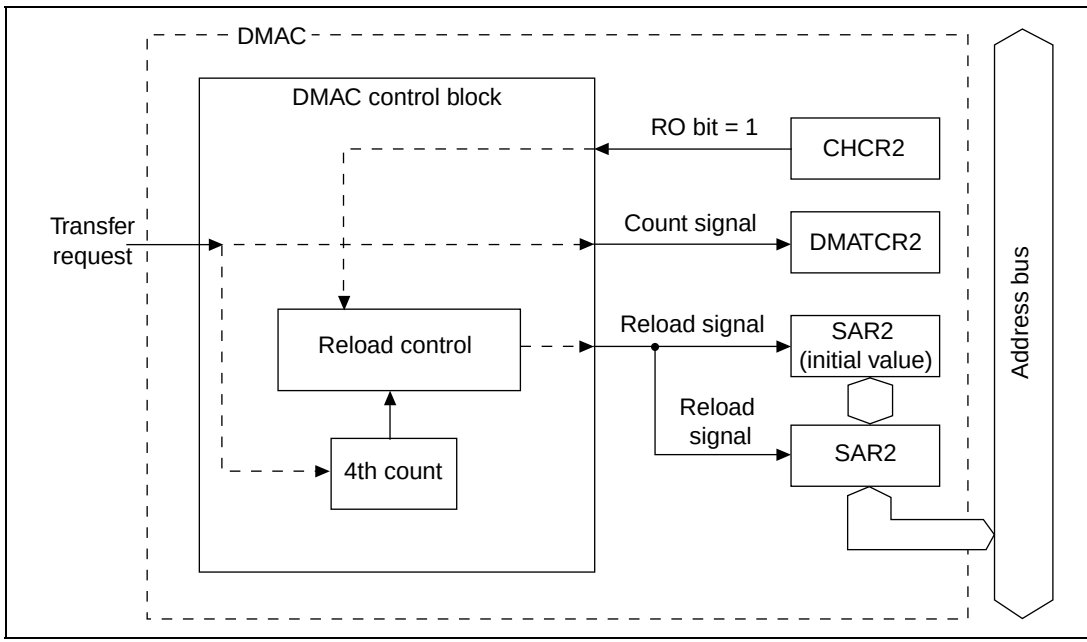


Figure 10.10 Source Address Reload Function

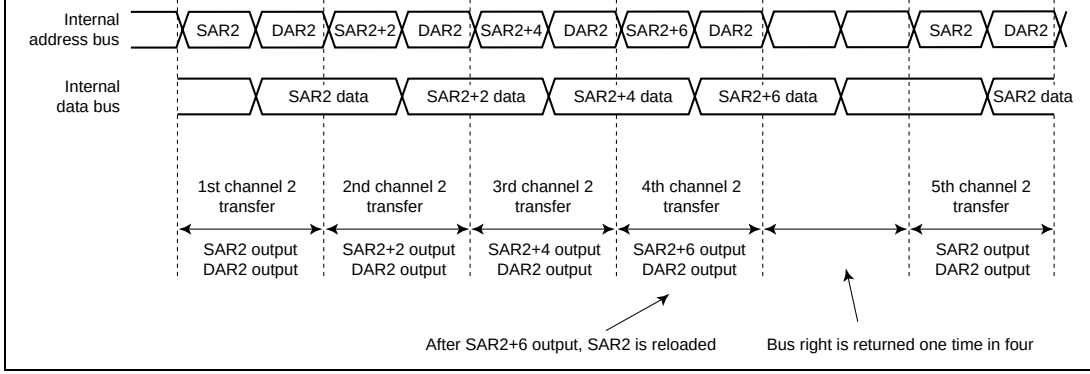


Figure 10.11 Source Address Reload Function Timing Chart

The reload function can be executed whether the transfer data size is 8, 16, or 32 bits.

DMATCR2, which specifies the number of transfers, is decremented by 1 at the end of every single-transfer-unit transfer, regardless of whether the reload function is on or off. Therefore, when using the reload function in the on state, a multiple of 4 must be specified in DMATCR2. Operation will not be guaranteed if any other value is set. Also, the counter which counts the occurrence of four transfers for address reloading is reset by clearing of the DME bit in DMAOR or the DE bit in CHCR2, setting of the transfer end flag (the TE bit in CHCR2), NMI input, and setting of the AE flag (address error generation in DMAC transfer), as well as by a reset and in software standby mode, but SAR2, DAR2, DMATCR2, and other registers are not reset. Consequently, when one of these sources occurs, there is a mixture of initialized counters and uninitialized registers in the DMAC, and incorrect operation may result if a restart is executed in this state. Therefore, when one of the above sources, other than TE setting, occurs during use of the address reload function, SAR, DAR2, and DMATCR2 settings must be carried out before re-execution.

10.3.10 DMA Transfer Ending Conditions

The DMA transfer ending conditions vary for individual channels ending and for all channels ending together.

Individual Channel Ending Conditions: There are two ending conditions. A transfer ends when the value of the channel's DMA transfer count register (DMATCR) is 0, or when the DE bit of the channel's CHCR is cleared to 0.

bit has been set, a DMAC interrupt (DEI) request is sent to the CPU.

- When DE of CHCR is 0: Software can halt a DMA transfer by clearing the DE bit in the channel's CHCR. The TE bit is not set when this happens.

Conditions for Ending on All Channels Simultaneously: Transfers on all channels end when the NMIF (NMI flag) bit or AE (address error flag) bit is set to 1 in DMAOR, or when the DME bit in DMAOR is cleared to 0.

- When the NMIF or AE bit is set to 1 in DMAOR: When an NMI interrupt or DMAC address error occurs, the NMIF or AE bit is set to 1 in DMAOR and all channels stop their transfers. The DMAC obtains the bus right, and if these flags are set to 1 during execution of a transfer, DMAC halts operation when the transfer processing currently being executed ends, and transfers the bus right to the other bus master. Consequently, even if the NMIF or AE bit is set to 1 during a transfer, the DMA source address register (SAR), designation address register (DAR), and transfer count register (DMATCR) are all updated. The TE bit is not set. To resume the transfers after NMI interrupt or address error processing, the NMIF or AE flag must be cleared. To avoid restarting a transfer on a particular channel, clear its DE bit to 0 in CHCR.

  When the processing of a one-unit transfer is complete: In a dual address mode direct address transfer, even if an address error occurs or the NMI flag is set during read processing, the transfer will not be halted until after completion of the following write processing. In such a case, SAR, DAR, and DMATCR values are updated. In the same manner, the transfer is not halted in indirect address transfers until after the final write processing has ended.
- When DME is cleared to 0 in DMAOR: Clearing the DME bit to 0 in DMAOR aborts the transfers on all channels. The TE bit is not set.

10.3.11 DMAC Access from CPU

The space addressed by the DMAC is 3-cycle space. Therefore, when the CPU becomes the bus master and accesses the DMAC, a minimum of three basic clock cycles are required for one bus cycle. Also, since the DMAC is located in word space, while a word-size access to the DMAC is completed in one bus cycle, a longword-size access is automatically divided into two word accesses, requiring two bus cycles (six basic clock cycles). These two bus cycles are executed consecutively; a different bus cycle is never inserted between the two word accesses. This applies to both write accesses and read accesses.

10.4.1 Example of DMA Transfer between On-Chip SCI and External Memory

In this example, on-chip serial communication interface channel 0 (SCI0) receive data is transferred to external memory using DMAC channel 0.

Table 10.5 indicates the transfer conditions and the set values of each of the registers.

Table 10.5 Transfer Conditions and Register Set Values for Transfer between On-chip SCI and External Memory

Transfer Conditions	Register	Value
Transfer source: RDR0 of on-chip SCI0	SAR0	H'FFFFF005
Transfer destination: external memory	DAR0	H'00400000
Transfer count: 64 times	DMATCR0	H'00000040
Transfer source address: fixed	CHCR0	H'00020105
Transfer destination address: incremented		
Transfer request source: SCI0 (RDR0)		
Bus mode: cycle-steal		
Transfer unit: byte		
Interrupt request generation at end of transfer		
DMAC master enable on	DMAOR	H'0001

10.4.2 Example of DMA Transfer between A/D Converter and On-Chip Memory (Address Reload On)

In this example, on-chip A/D converter channel 0 is the transfer source and on-chip memory is the transfer destination, and the address reload function is on.

Table 10.6 indicates the transfer conditions and the set values of each of the registers.

Transfer Conditions	Register	Value
Transfer source: on-chip A/D converter ch1 (A/D1)	SAR2	H'FFFFF820
Transfer destination: on-chip memory	DAR2	H'FFFF6000
Transfer count: 128 times (reload count 32 times)	DMATCR2	H'00000080
Transfer source address: incremented	CHCR2	H'010C110D
Transfer destination address: incremented		
Transfer request source: A/D converter ch1 (A/D1)		
Bus mode: burst		
Transfer unit: byte		
Interrupt request generation at end of transfer		
DMAC master enable on	DMAOR	H'0001

When address reload is on, the SAR2 value returns to its initially set value every four transfers. In the above example, when a transfer request is input from the A/D1, the byte-size data is first read in from the H'FFFFF820 register of on-chip A/D1 and that data is written to internal address H'FFFF6000. Because a byte-size transfer was performed, the SAR2 and DAR2 values at this point are H'FFFFF821 and H'FFFF6001, respectively. Also, because this is a burst transfer, the bus right remains secured, so continuous data transfer is possible.

When four transfers are completed, if address reload is off, execution continues with the fifth and sixth transfers and the SAR2 value continues to increment from H'FFFFF824 to H'FFFFF825 to H'FFFFF826 and so on. However, when address reload is on, DMAC transfer is halted upon completion of the fourth transfer and the bus right request signal to the CPU is cleared. At this time, the value stored in SAR2 is not H'FFFFF823 → H'FFFFF824, but H'FFFFF823 → H'FFFFF820, a return to the initially set address. The DAR2 value always continues to be decremented regardless of whether address reload is on or off.

The DMAC internal status, due to the above operation after completion of the fourth transfer, is indicated in table 10.7 for both address reload on and off.

SAR2	H'FFFFF820	H'FFFFF824
DAR2	H'FFFF6004	H'FFFF6004
DMATCR2	H'0000007C	H'0000007C
Bus right	Released	Retained
DMAC operation	Halted	Processing continues
Interrupts	Not issued	Not issued
Transfer request source flag clear	Executed	Not executed

Notes: 1. Interrupts are executed until the DMATCR2 value becomes 0, and if the IE bit of CHCR2 is set to 1, are issued regardless of whether address reload is on or off.
2. If transfer request source flag clears are executed until the DMATCR2 value becomes 0, they are executed regardless of whether address reload is on or off.
3. Designate burst mode when using the address reload function. There are cases where abnormal operation will result if it is used in cycle-steal mode.
4. Designate a multiple of four for the DMATCR2 value when using the address reload function. There are cases where abnormal operation will result if anything else is designated.

To execute transfers after the fifth transfer when address reload is on, have the transfer request source issue another transfer request signal.

10.4.3 Example of DMA Transfer between External Memory and SCI1 Transmitting Side (Indirect Address on)

In this example, DMAC channel 3 is used, indirect address designated external memory is the transfer source, and the SCI1 transmitting side is the transfer destination.

Table 10.8 indicates the transfer conditions and the set values of each of the registers.

Transfer Conditions	Register	Value
Transfer source: external memory	SAR3	H'00400000
Value stored in address H'00400000	—	H'00450000
Value stored in address H'00450000	—	H'55
Transfer destination: on-chip SCI TDR1	DAR3	H'FFFFF00B
Transfer count: 10 times	DMATCR3	H'0000000A
Transfer source address: incremented	CHCR3	H'10031001
Transfer destination address: fixed		
Transfer request source: SCI1 (TDR1)		
Bus mode: cycle-steal		
Transfer unit: byte		
Interrupt request not generated at end of transfer		
DMAC master enable on	DMAOR	H'0001

When indirect address mode is on, the data stored in the address set in SAR is not used as the transfer source data. In the case of indirect addressing, the value stored in the SAR address is read, then that value is used as the address and the data read from that address is used as the transfer source data, then that data is stored in the address designated by DAR.

In the table 10.8 example, when a transfer request from TDR1 of SCI1 is generated, a read of the address located at H'00400000, which is the value set in SAR3, is performed first. The data H'00450000 is stored at this H'00400000 address, and the DMAC first reads this H'00450000 value. It then uses this read value of H'00450000 as an address and reads the value of H'55 that is stored in the H'00450000 address. It then writes the value H'55 to address H'FFFFF00B designated by DAR3 to complete one indirect address transfer.

With indirect addressing, the first executed data read from the address set in SAR3 always results in a longword size transfer regardless of the TS0 and TS1 bit designations for transfer data size. However, the transfer source address fixed and increment or decrement designations are according to the SM0 and SM1 bits. Consequently, despite the fact that the transfer data size designation is byte in this example, the SAR3 value at the end of one transfer is H'00400004. The write operation is exactly the same as an ordinary dual address transfer write operation.

1. Only word (16-bit) access can be used on the DMA operation register (DMAOR). All other registers can be accessed in word (16-bit) or longword (32-bit) units.
2. When rewriting the RS0–RS4 bits of CHCR0–CHCR3, first clear the DE bit to 0 (clear the DE bit to 0 before modifying CHCR).
3. When an NMI interrupt is input, the NMIF bit of DMAOR is set even when the DMAC is not operating.
4. Clear the DME bit of DMAOR to 0 and make certain that any transfer request processing accepted by the DMAC has been completed before entering standby mode.
5. Do not access the DMAC, BSC, or UBC on-chip peripheral modules from the DMAC.
6. When activating the DMAC, make the CHCR settings as the final step. Abnormal operation may result if any other registers are set last.
7. After the DMATCR count becomes 0 and the DMA transfer ends normally, always write 0 to DMATCR, even when executing the maximum number of transfers on the same channel. Abnormal operation may result if this is not done.
8. Designate burst mode as the transfer mode when using the address reload function. Abnormal operation may result in cycle-steal mode.
9. Designate a multiple of four for the DMATCR value when using the address reload function, otherwise abnormal operation may result.
10. Do not access empty DMAC register addresses. Operation cannot be guaranteed when empty addresses are accessed.
11. If DMAC transfer is aborted by NMIF or AE setting, or DME or DE clearing, during DMAC execution with address reload on, the SAR2, DAR2, and DMATCR2 settings should be made before re-executing the transfer. The DMAC may not operate correctly if this is not done.
12. Do not set the DE bit to 1 while bits RS0 to RS4 in CHCR0 to CHCR3 are still set to “no request.”

11.1 Overview

The SH7055SF has an on-chip advanced timer unit-II (ATU-II) with one 32-bit timer channel and eleven 16-bit timer channels.

11.1.1 Features

ATU-II features are summarized below.

- Capability to process up to 65 pulse inputs and outputs
- Prescaler

— Input clock to channels 0 and 10 scaled in 1 stage, input clock to channels 1 to 8 and 11 scaled in 2 stages

— 1/1 to 1/32 clock scaling possible in initial stage for channels 0 to 8, 10, and 11

— 1/1, 1/2, 1/4, 1/8, 1/16, or 1/32 scaling possible in second stage for channels 1 to 8 and 11

— External clock TCLKA, TCLKB selection also possible for channels 1 to 5 and 11

— Channels 1 to 5 enable TI10 pin input, multiple the TI10 pin input (correction), and select AGCK and AGCKM.

- Channel 0 has four 32-bit input capture lines, allowing the following operations:

— Rising-edge, falling-edge, or both-edge detection selectable

— DMAC can be activated at capture timing

— Channel 10 compare-match signal can be captured as a trigger

— Interval interrupt generation function generates three interval interrupts as selected. CPU interruption or A/D converter (AD0, 1, 2) activation possible

— Capture interrupt and counter overflow interrupt can be generated

- Channel 1 has one 16-bit output compare register, eight general registers, and one dedicated input capture register. The output compare register can also be selected for one-shot pulse offset in combination with the channel 8 down-counter.

— General registers (GR1A–H) can be used as input capture or output compare registers

— Waveform output by means of compare-match: Selection of 0 output, 1 output, or toggle output

— Input capture function: Rising-edge, falling-edge, or both-edge detection

— Channel 0 input signal (TI0A) can be captured as trigger

— Provision for forcible cutoff of channel 8 down-counters (DCNT8A–H)

— Compare-match interrupts/capture interrupts and counter overflow interrupts can be generated

offset in combination with the channel 8 down-counter.

— General registers (GR2A–H) can be used as input capture or output compare registers
— Waveform output by means of compare-match: Selection of 0 output, 1 output, or toggle output
— Input capture function: Rising-edge, falling-edge, or both-edge detection
— Channel 0 input signal (TI0A) can be captured as trigger
— Provision for forcible cutoff of channel 8 down-counters (DCNT8I to P)
— Compare-match interrupts/capture interrupts and counter overflow interrupts can be generated

- Channels 3 to 5 each have four general registers, allowing the following operations:

— Selection of input capture, output compare, PWM mode
— Waveform output by means of compare-match: Selection of 0 output, 1 output, or toggle output
— Input capture function: Rising-edge, falling-edge, or both-edge detection
— Channel 9 compare-match signal can be captured as trigger (channel 3 only)
— Compare-match interrupts/capture interrupts can be generated

- Channels 6 and 7 have four 16-bit duty registers, four cycle registers, and four buffer registers, allowing the following operations:
  — Any cycle and duty from 0 to 100% can be set
  — Duty buffer register value transferred to duty register every cycle
  — Interrupts can be generated every cycle
  — Complementary PWM output can be set (channel 6 only)
- Channel 8 has sixteen 16-bit down-counters for one-shot pulse output, allowing the following operations:
  — One-shot pulse generation by down-counter
  — Down-counter can be rewritten during count
  — Interrupt can be generated at end of down-count
  — Offset one-shot pulse function available
  — Can be linked to channel 1 and 2 output compare functions
  — Reload function can be set to eight 16-bit down counters (DCNT8I to DCNT8P)
- Channel 9 has six event counters and six general registers, allowing the following operations:
  — Event counters can be cleared by compare-match
  — Rising-edge, falling-edge, or both-edge detection available for external input
  — Compare-match signal can be input to channel 3

counter, and output compare register, and one 16-bit reload counter, allowing the following operations:

- Capture on external input pin edge input
- Reload count possible with 1/32, 1/64, 1/128, or 1/256 times the captured value
- Internal clock generated by reload counter underflow can be used as 16-bit free-running counter input
- Channels 1 and 2 free-running counter clearing capability

- Channel 11 has one 16-bit free-running counter and two 16-bit general registers, allowing the following operations:
  - Two general registers can be used for input capture/output compare
  - Waveform output at compare match: 0 output, 1 output, and toggle output selectable
  - Input capture function: Detection at rising edge, falling edge, and both edges
  - Compare-match signal can be output at the APC by using a general register as a output compare register
- High-speed access to internal 16-bit bus
  - High-speed access to 16-bit bus for 16-bit registers: timer counters, compare registers, and capture registers
- 75 interrupt sources
  - Four input capture interrupt requests, one overflow interrupt request, and one interval interrupt request for channel 0
  - Sixteen dual input capture/compare-match interrupt requests and two counter overflow interrupt requests for channels 1 and 2
  - Twelve dual input capture/compare-match interrupt requests and three overflow interrupt requests for channels 3 to 5
  - Eight compare-match interrupts for channels 6 and 7
  - Sixteen one-shot end interrupt requests for channel 8
  - Six compare-match interrupts for channel 9
  - Two compare-match interrupts and one dual-function input capture/compare-match interrupt for channel 10
  - Two dual input capture/compare-match interrupt requests and one overflow interrupt request for channel 11
- Direct memory access controller (DMAC) activation
  - The DMAC can be activated by a channel 0 input capture interrupt (ICI0A–D)
  - The DMAC can be activated by a channel 6 cycle register 6 compare-match interrupt (CMI6A–D)
  - The DMAC can be activated by a channel 7 cycle register 7 compare-match interrupt (CMI7A–D)

Interval interrupt request registers (ITVRR1, ITVRR2A, ITVRR2B)

Table 11.1 lists the functions of the ATU-II.

Counter configuration	Clock sources	ϕ–ϕ/32	(ϕ–ϕ/32) × (1/2n) (n = 0–5) TCLKA, TCLKB, AGCK, AGCKM	(ϕ–ϕ/32) × (1/2n) (n = 0–5) TCLKA, TCLKB, AGCK, AGCKM	(ϕ–ϕ/32) × (1/2n) (n = 0–5) TCLKA, TCLKB, AGCK, AGCKM
	Counters	TCNT0H, TCNT0L	TCNT1A, TCNT1B	TCNT2A, TCNT2B	TCNT3–5
	General registers	—	GR1A–H	GR2A–H	GR3A–D, GR4A–D, GR5A–D
	Dedicated input capture	ICR0AH, ICR0AL, ICR0BH, ICR0BL, ICR0CH, ICR0CL, ICR0DH, ICR0DL	OSBR1	OSBR2	—
	Dedicated output compare	—	OCR1	OCR2A–2H	—
	PWM output	—	—	—	Duty: GR3A–C, GR4A–C, GR5A–C Cycle: GR3D, GR4D, GR5D
Input pins		TI0A–D	—	—	—
I/O pins		—	TIO1A–H	TIO2A–H	TIO3A–D, TIO4A–D, TIO5A–D
Output pins		—	—	—	—
Counter clearing function		—	—	—	O
Interrupt sources		6 sources Interval × 1, input capture × 4, overflow × 1	9 sources Dual input capture/ compare-match × 8, overflow × 1	9 sources Dual input capture/ compare-match × 8, overflow × 1* (* Same vector)	15 sources Dual input capture/ compare-match × 12, overflow × 3
Inter-channel and inter-module connection signals		A/D converter activation by interval interrupt request, DMAC activation by input capture interrupt, channel 10 compare-match signal capture trigger input	Compare-match signal trigger output to channel 8 one-shot pulse output down-counter Channel 10 compare-match signal counter clear input	Compare-match signal trigger output to channel 8 one-shot pulse output down-counter Channel 10 compare-match signal counter clear input	Channel 9 compare-match signal input to capture trigger (Channel 3 only)

Counter configu-ration	Clock sources	(ϕ–ϕ/32) × (1/2n) (n = 0–5)	(ϕ–ϕ/32) × (1/2n) (n = 0–5)	—	(ϕ–ϕ/32)	(ϕ–ϕ/32) × (1/2n) (n = 0-5) TCLKA, TCLKB
	Counters	TCNT6A–D, TCNT7A–D	DCNT8A–P	ECNT9A–F	TCNT10AH, TCNT10AL, TCNT10B–H	TCNT11
	General registers	—	—	—	—	GR11A, GR11B
	Dedicated input capture	—	—	—	ICR10AH, ICR10AL	—
	Dedicated output compare	—	—	GR9A–F	GR10G OCR10AH, OCR10AL, OCR10B, NCR10, TCCLR10	—
	PWM output	CYLR6A–D, CYLR7A–D, DTR6A–D, DTR7A–D, BFR6A–D, BFR7A–D	—	—	—	—
Input pins		—	—	TI9A–F	TI10	—
I/O pins		—	—	—	—	TIO11A, TIO11B
Output pins		TO6A–D, TO7A–D	TO8A–P	—	—	—
Counter clearing function		O	—	O	O	—
Interrupt sources		8 sources Compare-match × 8	16 sources Underflow × 16	6 sources Compare-match × 6	3 sources Compare-match × 2, dual input capture/compare-match × 1	3 sources Dual input capture/compare-match × 2, overflow × 1
Inter-channel and inter-module connection signals		DMAC activation compare-match signal output	Channel 1 and 2 compare-match signal trigger input to one-shot pulse output down-counter	Compare-match signal channel 3 capture trigger output	Compare-match signal channel 0 capture trigger output Channel 1 and 2 counter clear output	Compare-match signal output to APC

O: Available
—: Not available

Table 11.2 shows the pin configuration of the ATU-II. When these external pin functions are used, the pin function controller (PFC) should also be set in accordance with the ATU-II settings. If there are a number of pins with the same function, make settings so that only one of the pins is used. For details, see section 20, Pin Function Controller (PFC).

Table 11.2 ATU-II Pins

Channel	Name	Abbreviation	I/O	Function
Common	Clock input A	TCLKA	Input	External clock A input pin
	Clock input B	TCLKB	Input	External clock B input pin
0	Input capture 0A	TI0A	Input	ICR0AH, ICR0AL input capture input pin
	Input capture 0B	TI0B	Input	ICR0BH, ICR0BL input capture input pin
	Input capture 0C	TI0C	Input	ICR0CH, ICR0CL input capture input pin
	Input capture 0D	TI0D	Input	ICR0DH, ICR0DL input capture input pin
1	Input capture/output compare 1A	TIO1A	Input/ output	GR1A output compare output/input capture input
	Input capture/output compare 1B	TIO1B	Input/ output	GR1B output compare output/input capture input
	Input capture/output compare 1C	TIO1C	Input/ output	GR1C output compare output/input capture input
	Input capture/output compare 1D	TIO1D	Input/ output	GR1D output compare output/input capture input
	Input capture/output compare 1E	TIO1E	Input/ output	GR1E output compare output/input capture input
	Input capture/output compare 1F	TIO1F	Input/ output	GR1F output compare output/input capture input
	Input capture/output compare 1G	TIO1G	Input/ output	GR1G output compare output/input capture input
	Input capture/output compare 1H	TIO1H	Input/ output	GR1H output compare output/input capture input

2	Input capture/output compare 2A	TIO2A	Input/ output	GR2A output compare output/input capture input
	Input capture/output compare 2B	TIO2B	Input/ output	GR2B output compare output/input capture input
	Input capture/output compare 2C	TIO2C	Input/ output	GR2C output compare output/input capture input
	Input capture/output compare 2D	TIO2D	Input/ output	GR2D output compare output/input capture input
	Input capture/output compare 2E	TIO2E	Input/ output	GR2E output compare output/input capture input
	Input capture/output compare 2F	TIO2F	Input/ output	GR2F output compare output/input capture input
	Input capture/output compare 2G	TIO2G	Input/ output	GR2G output compare output/input capture input
	Input capture/output compare 2H	TIO2H	Input/ output	GR2H output compare output/input capture input
3	Input capture/output compare 3A	TIO3A	Input/ output	GR3A output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 3B	TIO3B	Input/ output	GR3B output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 3C	TIO3C	Input/ output	GR3C output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 3D	TIO3D	Input/ output	GR3D output compare output/input capture input
4	Input capture/output compare 4A	TIO4A	Input/ output	GR4A output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 4B	TIO4B	Input/ output	GR4B output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 4C	TIO4C	Input/ output	GR4C output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 4D	TIO4D	Input/ output	GR4D output compare output/input capture input

5	Input capture/output compare 5A	TIO5A	Input/ output	GR5A output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 5B	TIO5B	Input/ output	GR5B output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 5C	TIO5C	Input/ output	GR5C output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 5D	TIO5D	Input/ output	GR5D output compare output/input capture input
6	Output compare 6A	TO6A	Output	PWM output pin
	Output compare 6B	TO6B	Output	PWM output pin
	Output compare 6C	TO6C	Output	PWM output pin
	Output compare 6D	TO6D	Output	PWM output pin
7	Output compare 7A	TO7A	Output	PWM output pin
	Output compare 7B	TO7B	Output	PWM output pin
	Output compare 7C	TO7C	Output	PWM output pin
	Output compare 7D	TO7D	Output	PWM output pin
8	One-shot pulse 8A	TO8A	Output	One-shot pulse output pin
	One-shot pulse 8B	TO8B	Output	One-shot pulse output pin
	One-shot pulse 8C	TO8C	Output	One-shot pulse output pin
	One-shot pulse 8D	TO8D	Output	One-shot pulse output pin
	One-shot pulse 8E	TO8E	Output	One-shot pulse output pin
	One-shot pulse 8F	TO8F	Output	One-shot pulse output pin
	One-shot pulse 8G	TO8G	Output	One-shot pulse output pin
	One-shot pulse 8H	TO8H	Output	One-shot pulse output pin
	One-shot pulse 8I	TO8I	Output	One-shot pulse output pin
	One-shot pulse 8J	TO8J	Output	One-shot pulse output pin
	One-shot pulse 8K	TO8K	Output	One-shot pulse output pin
	One-shot pulse 8L	TO8L	Output	One-shot pulse output pin
	One-shot pulse 8M	TO8M	Output	One-shot pulse output pin
	One-shot pulse 8N	TO8N	Output	One-shot pulse output pin

8	One-shot pulse 8O	TO8O	Output	One-shot pulse output pin
	One-shot pulse 8P	TO8P	Output	One-shot pulse output pin
9	Event input 9A	TI9A	Input	GR9A event input
	Event input 9B	TI9B	Input	GR9B event input
	Event input 9C	TI9C	Input	GR9C event input
	Event input 9D	TI9D	Input	GR9D event input
	Event input 9E	TI9E	Input	GR9E event input
	Event input 9F	TI9F	Input	GR9F event input
10	Input capture	TI10	Input	ICR10AH, ICR10AL input capture input
11	Input capture/output compare 11A	TIO11A	Input/ output	GR11A output compare output/input capture input
	Input capture/output compare 11B	TIO11B	Input/ output	GR11B output compare output/input capture input

Table 11.3 summarizes the ATU-II registers.

Table 11.3 ATU-II Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits)	Section No.
Common	Timer start register 1	TSTR1	R/W	H'00	H'FFFFF401	8, 16, 32	11.2.1
	Timer start register 2	TSTR2	R/W	H'00	H'FFFFF400		
	Timer start register 3	TSTR3	R/W	H'00	H'FFFFF402		
	Prescaler register 1	PSCR1	R/W	H'00	H'FFFFF404	8	11.2.2
	Prescaler register 2	PSCR2	R/W	H'00	H'FFFFF406		
	Prescaler register 3	PSCR3	R/W	H'00	H'FFFFF408		
	Prescaler register 4	PSCR4	R/W	H'00	H'FFFFF40A		
0	Free-running counter 0H	TCNT0H	R/W	H'0000	H'FFFFF430	32	11.2.15
	Free-running counter 0L	TCNT0L	R/W	H'0000			
	Input capture register 0AH	ICR0AH	R	H'0000	H'FFFFF434		11.2.19
	Input capture register 0AL	ICR0AL	R	H'0000			
	Input capture register 0BH	ICR0BH	R	H'0000	H'FFFFF438		
	Input capture register 0BL	ICR0BL	R	H'0000			
	Input capture register 0CH	ICR0CH	R	H'0000	H'FFFFF43C		
	Input capture register 0CL	ICR0CL	R	H'0000			
	Input capture register 0DH	ICR0DH	R	H'0000	H'FFFFF420		
	Input capture register 0DL	ICR0DL	R	H'0000			
	Timer interval interrupt request register 1	ITVRR1	R/W	H'00	H'FFFFF424	8	11.2.7
	Timer interval interrupt request register 2A	ITVRR2A	R/W	H'00	H'FFFFF426		

Channel	Name	tion	R/W	Value	Address	Size (Bits)	No.
0	Timer interval interrupt request register 2B	ITVRR2B	R/W	H'00	H'FFFFF428	8	11.2.7
	Timer I/O control register	TIOR0	R/W	H'00	H'FFFFF42A		11.2.4
	Timer status register 0	TSR0	R/(W)*	H'0000	H'FFFFF42C	16	11.2.5
	Timer interrupt enable register 0	TIER0	R/W	H'0000	H'FFFFF42E		11.2.6
1	Free-running counter 1A	TCNT1A	R/W	H'0000	H'FFFFF440	16	11.2.15
	Free-running counter 1B	TCNT1B	R/W	H'0000	H'FFFFF442		
	General register 1A	GR1A	R/W	H'FFFF	H'FFFFF444		11.2.20
	General register 1B	GR1B	R/W	H'FFFF	H'FFFFF446		
	General register 1C	GR1C	R/W	H'FFFF	H'FFFFF448		
	General register 1D	GR1D	R/W	H'FFFF	H'FFFFF44A		
	General register 1E	GR1E	R/W	H'FFFF	H'FFFFF44C		
	General register 1F	GR1F	R/W	H'FFFF	H'FFFFF44E		
	General register 1G	GR1G	R/W	H'FFFF	H'FFFFF450		
	General register 1H	GR1H	R/W	H'FFFF	H'FFFFF452		
	Output compare register 1	OCR1	R/W	H'FFFF	H'FFFFF454		11.2.18
	Offset base register 1	OSBR1	R	H'0000	H'FFFFF456		11.2.21
	Timer I/O control register 1A	TIOR1A	R/W	H'00	H'FFFFF459	8, 16	11.2.4
	Timer I/O control register 1B	TIOR1B	R/W	H'00	H'FFFFF458		
	Timer I/O control register 1C	TIOR1C	R/W	H'00	H'FFFFF45B		
	Timer I/O control register 1D	TIOR1D	R/W	H'00	H'FFFFF45A		
	Timer control register 1A	TCR1A	R/W	H'00	H'FFFFF45D		11.2.3
	Timer control register 1B	TCR1B	R/W	H'00	H'FFFFF45C		

Channel	Name	tion	R/W	Value	Address	Size (Bits)	No.
1	Timer status register 1A	TSR1A	R/(W)*	H'0000	H'FFFFF45E	16	11.2.5
	Timer status register 1B	TSR1B	R/(W)*	H'0000	H'FFFFF460		
	Timer interrupt enable register 1A	TIER1A	R/W	H'0000	H'FFFFF462		11.2.6
	Timer interrupt enable register 1B	TIER1B	R/W	H'0000	H'FFFFF464		
	Trigger mode register	TRGMDR	R/W	H'00	H'FFFFF466	8	11.2.8
2	Free-running counter 2A	TCNT2A	R/W	H'0000	H'FFFFF600	16	11.2.15
	Free-running counter 2B	TCNT2B	R/W	H'0000	H'FFFFF602		
	General register 2A	GR2A	R/W	H'FFFF	H'FFFFF604		11.2.20
	General register 2B	GR2B	R/W	H'FFFF	H'FFFFF606		
	General register 2C	GR2C	R/W	H'FFFF	H'FFFFF608		
	General register 2D	GR2D	R/W	H'FFFF	H'FFFFF60A		
	General register 2E	GR2E	R/W	H'FFFF	H'FFFFF60C		
	General register 2F	GR2F	R/W	H'FFFF	H'FFFFF60E		
	General register 2G	GR2G	R/W	H'FFFF	H'FFFFF610		
	General register 2H	GR2H	R/W	H'FFFF	H'FFFFF612		
	Output compare register 2A	OCR2A	R/W	H'FFFF	H'FFFFF614		11.2.18
	Output compare register 2B	OCR2B	R/W	H'FFFF	H'FFFFF616		
	Output compare register 2C	OCR2C	R/W	H'FFFF	H'FFFFF618		
	Output compare register 2D	OCR2D	R/W	H'FFFF	H'FFFFF61A		
	Output compare register 2E	OCR2E	R/W	H'FFFF	H'FFFFF61C		
	Output compare register 2F	OCR2F	R/W	H'FFFF	H'FFFFF61E		

Channel	Name	tion	R/W	Value	Address	Size (Bits)	No.
2	Output compare register 2G	OCR2G	R/W	H'FFFF	H'FFFFF620	16	11.2.18
	Output compare register 2H	OCR2H	R/W	H'FFFF	H'FFFFF622		
	Offset base register 2	OSBR2	R	H'0000	H'FFFFF624		11.2.21
	Timer I/O control register 2A	TIOR2A	R/W	H'00	H'FFFFF627	8, 16	11.2.4
	Timer I/O control register 2B	TIOR2B	R/W	H'00	H'FFFFF626		
	Timer I/O control register 2C	TIOR2C	R/W	H'00	H'FFFFF629		
	Timer I/O control register 2D	TIOR2D	R/W	H'00	H'FFFFF628		
	Timer control register 2A	TCR2A	R/W	H'00	H'FFFFF62B		11.2.3
	Timer control register 2B	TCR2B	R/W	H'00	H'FFFFF62A		
	Timer status register 2A	TSR2A	R/(W)*	H'0000	H'FFFFF62C	16	11.2.5
	Timer status register 2B	TSR2B	R/(W)*	H'0000	H'FFFFF62E		
	Timer interrupt enable register 2A	TIER2A	R/W	H'0000	H'FFFFF630		11.2.6
	Timer interrupt enable register 2B	TIER2B	R/W	H'0000	H'FFFFF632		
3–5	Timer status register 3	TSR3	R/(W)*	H'0000	H'FFFFF480	16	11.2.5
	Timer interrupt enable register 3	TIER3	R/W	H'0000	H'FFFFF482		11.2.6
	Timer mode register	TMDR	R/W	H'00	H'FFFFF484	8	11.2.9
3	Free-running counter 3	TCNT3	R/W	H'0000	H'FFFFF4A0	16	11.2.15
	General register 3A	GR3A	R/W	H'FFFF	H'FFFFF4A2		11.2.20
	General register 3B	GR3B	R/W	H'FFFF	H'FFFFF4A4		
	General register 3C	GR3C	R/W	H'FFFF	H'FFFFF4A6		
	General register 3D	GR3D	R/W	H'FFFF	H'FFFFF4A8		

Channel	Name	tion	R/W	Value	Address	Size (Bits)	No.
3	Timer I/O control register 3A	TIOR3A	R/W	H'00	H'FFFFF4AB	8, 16	11.2.4
	Timer I/O control register 3B	TIOR3B	R/W	H'00	H'FFFFF4AA		
	Timer control register 3	TCR3	R/W	H'00	H'FFFFF4AC	8	11.2.3
4	Free-running counter 4	TCNT4	R/W	H'0000	H'FFFFF4C0	16	11.2.15
	General register 4A	GR4A	R/W	H'FFFF	H'FFFFF4C2		11.2.20
	General register 4B	GR4B	R/W	H'FFFF	H'FFFFF4C4		
	General register 4C	GR4C	R/W	H'FFFF	H'FFFFF4C6		
	General register 4D	GR4D	R/W	H'FFFF	H'FFFFF4C8		
	Timer I/O control register 4A	TIOR4A	R/W	H'00	H'FFFFF4CB	8, 16	11.2.4
	Timer I/O control register 4B	TIOR4B	R/W	H'00	H'FFFFF4CA		
	Timer control register 4	TCR4	R/W	H'00	H'FFFFF4CC	8	11.2.3
5	Free-running counter 5	TCNT5	R/W	H'0000	H'FFFFF4E0	16	11.2.15
	General register 5A	GR5A	R/W	H'FFFF	H'FFFFF4E2		11.2.20
	General register 5B	GR5B	R/W	H'FFFF	H'FFFFF4E4		
	General register 5C	GR5C	R/W	H'FFFF	H'FFFFF4E6		
	General register 5D	GR5D	R/W	H'FFFF	H'FFFFF4E8		
	Timer I/O control register 5A	TIOR5A	R/W	H'00	H'FFFFF4EB	8, 16	11.2.4
	Timer I/O control register 5B	TIOR5B	R/W	H'00	H'FFFFF4EA		
	Timer control register 5	TCR5	R/W	H'00	H'FFFFF4EC	8	11.2.3
6	Free-running counter 6A	TCNT6A	R/W	H'0001	H'FFFFF500	16	11.2.15
	Free-running counter 6B	TCNT6B	R/W	H'0001	H'FFFFF502		
	Free-running counter 6C	TCNT6C	R/W	H'0001	H'FFFFF504		
	Free-running counter 6D	TCNT6D	R/W	H'0001	H'FFFFF506		

Channel	Name	tion	R/W	Value	Address	Size (Bits)	No.
6	Cycle register 6A	CYLR6A	R/W	H'FFFF	H'FFFFF508	16	11.2.22
	Cycle register 6B	CYLR6B	R/W	H'FFFF	H'FFFFF50A		
	Cycle register 6C	CYLR6C	R/W	H'FFFF	H'FFFFF50C		
	Cycle register 6D	CYLR6D	R/W	H'FFFF	H'FFFFF50E		
	Buffer register 6A	BFR6A	R/W	H'FFFF	H'FFFFF510		11.2.23
	Buffer register 6B	BFR6B	R/W	H'FFFF	H'FFFFF512		
	Buffer register 6C	BFR6C	R/W	H'FFFF	H'FFFFF514		
	Buffer register 6D	BFR6D	R/W	H'FFFF	H'FFFFF516		
	Duty register 6A	DTR6A	R/W	H'FFFF	H'FFFFF518		11.2.24
	Duty register 6B	DTR6B	R/W	H'FFFF	H'FFFFF51A		
	Duty register 6C	DTR6C	R/W	H'FFFF	H'FFFFF51C		
	Duty register 6D	DTR6D	R/W	H'FFFF	H'FFFFF51E		
	Timer control register 6A	TCR6A	R/W	H'00	H'FFFFF521	8, 16	11.2.3
	Timer control register 6B	TCR6B	R/W	H'00	H'FFFFF520		
	Timer status register 6	TSR6	R/(W)*	H'0000	H'FFFFF522	16	11.2.5
	Timer interrupt enable register 6	TIER6	R/W	H'0000	H'FFFFF524		11.2.6
	PWM mode register	PMDR	R/W	H'00	H'FFFFF526	8	11.2.10
7	Free-running counter 7A	TCNT7A	R/W	H'0001	H'FFFFF580	16	11.2.15
	Free-running counter 7B	TCNT7B	R/W	H'0001	H'FFFFF582		
	Free-running counter 7C	TCNT7C	R/W	H'0001	H'FFFFF584		
	Free-running counter 7D	TCNT7D	R/W	H'0001	H'FFFFF586		
	Cycle register 7A	CYLR7A	R/W	H'FFFF	H'FFFFF588		11.2.22
	Cycle register 7B	CYLR7B	R/W	H'FFFF	H'FFFFF58A		
	Cycle register 7C	CYLR7C	R/W	H'FFFF	H'FFFFF58C		
	Cycle register 7D	CYLR7D	R/W	H'FFFF	H'FFFFF58E		

Channel	Name	tion	R/W	Value	Address	Size (Bits)	No.
7	Buffer register 7A	BFR7A	R/W	H'FFFF	H'FFFFF590	16	11.2.23
	Buffer register 7B	BFR7B	R/W	H'FFFF	H'FFFFF592		
	Buffer register 7C	BFR7C	R/W	H'FFFF	H'FFFFF594		
	Buffer register 7D	BFR7D	R/W	H'FFFF	H'FFFFF596		
	Duty register 7A	DTR7A	R/W	H'FFFF	H'FFFFF598		11.2.24
	Duty register 7B	DTR7B	R/W	H'FFFF	H'FFFFF59A		
	Duty register 7C	DTR7C	R/W	H'FFFF	H'FFFFF59C		
	Duty register 7D	DTR7D	R/W	H'FFFF	H'FFFFF59E		
	Timer control register 7A	TCR7A	R/W	H'00	H'FFFFF5A1	8, 16	11.2.3
	Timer control register 7B	TCR7B	R/W	H'00	H'FFFFF5A0		
	Timer status register 7	TSR7	R/(W)*	H'0000	H'FFFFF5A2	16	11.2.5
	Timer interrupt enable register 7	TIER7	R/W	H'0000	H'FFFFF5A4		11.2.6
8	Down-counter 8A	DCNT8A	R/W	H'0000	H'FFFFF640	16	11.2.16
	Down-counter 8B	DCNT8B	R/W	H'0000	H'FFFFF642		
	Down-counter 8C	DCNT8C	R/W	H'0000	H'FFFFF644		
	Down-counter 8D	DCNT8D	R/W	H'0000	H'FFFFF646		
	Down-counter 8E	DCNT8E	R/W	H'0000	H'FFFFF648		
	Down-counter 8F	DCNT8F	R/W	H'0000	H'FFFFF64A		
	Down-counter 8G	DCNT8G	R/W	H'0000	H'FFFFF64C		
	Down-counter 8H	DCNT8H	R/W	H'0000	H'FFFFF64E		
	Down-counter 8I	DCNT8I	R/W	H'0000	H'FFFFF650		
	Down-counter 8J	DCNT8J	R/W	H'0000	H'FFFFF652		
	Down-counter 8K	DCNT8K	R/W	H'0000	H'FFFFF654		
	Down-counter 8L	DCNT8L	R/W	H'0000	H'FFFFF656		
	Down-counter 8M	DCNT8M	R/W	H'0000	H'FFFFF658		
	Down-counter 8N	DCNT8N	R/W	H'0000	H'FFFFF65A		
	Down-counter 8O	DCNT8O	R/W	H'0000	H'FFFFF65C		
	Down-counter 8P	DCNT8P	R/W	H'0000	H'FFFFF65E		

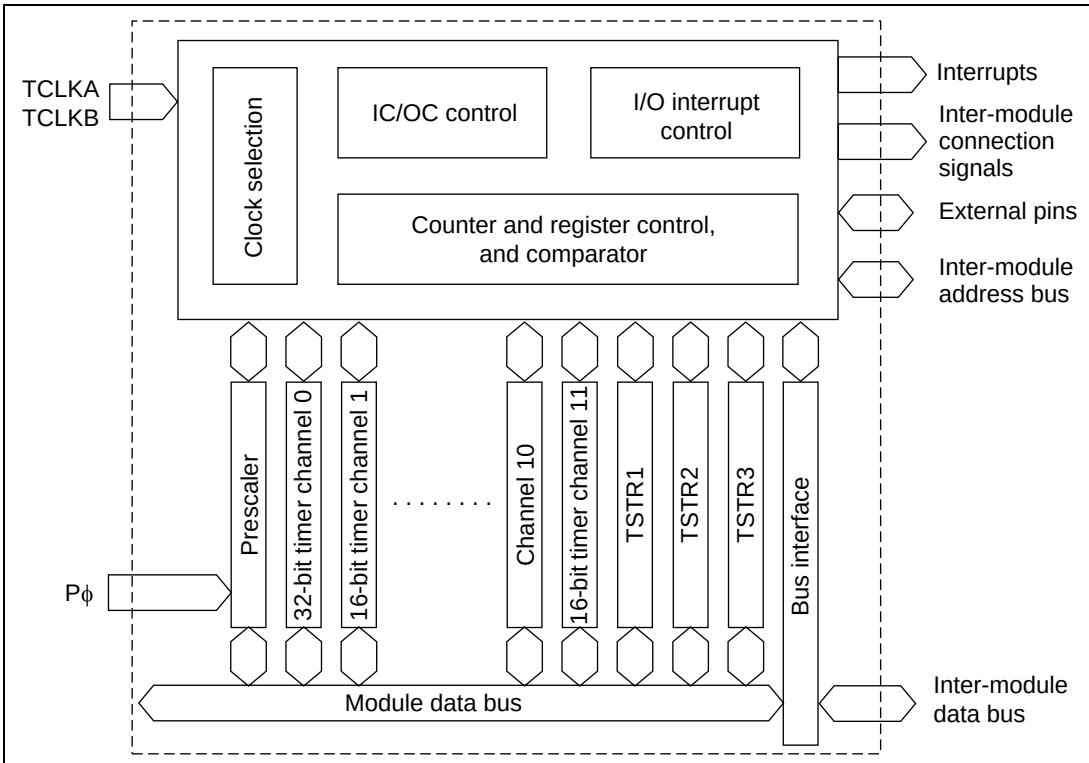
Channel	Name	tion	R/W	Value	Address	Size (Bits)	No.
8	Reload register 8	RLDR8	R/W	H'0000	H'FFFFF660	16	11.2.25
	Timer connection register	TCNR	R/W	H'0000	H'FFFFF662		11.2.12
	One-shot pulse terminate register	OTR	R/W	H'0000	H'FFFFF664		11.2.13
	Down-count start register	DSTR	R/W	H'0000	H'FFFFF666		11.2.11
	Timer control register 8	TCR8	R/W	H'00	H'FFFFF668	8	11.2.3
	Timer status register 8	TSR8	R/(W)*	H'0000	H'FFFFF66A	16	11.2.5
	Timer interrupt enable register 8	TIER8	R/W	H'0000	H'FFFFF66C		11.2.6
	Reload enable register	RLDENR	R/W	H'00	H'FFFFF66E	8	11.2.14
9	Event counter 9A	ECNT9A	R/W	H'00	H'FFFFF680	8	11.2.17
	Event counter 9B	ECNT9B	R/W	H'00	H'FFFFF682		
	Event counter 9C	ECNT9C	R/W	H'00	H'FFFFF684		
	Event counter 9D	ECNT9D	R/W	H'00	H'FFFFF686		
	Event counter 9E	ECNT9E	R/W	H'00	H'FFFFF688		
	Event counter 9F	ECNT9F	R/W	H'00	H'FFFFF68A		
	General register 9A	GR9A	R/W	H'FF	H'FFFFF68C		11.2.20
	General register 9B	GR9B	R/W	H'FF	H'FFFFF68E		
	General register 9C	GR9C	R/W	H'FF	H'FFFFF690		
	General register 9D	GR9D	R/W	H'FF	H'FFFFF692		
	General register 9E	GR9E	R/W	H'FF	H'FFFFF694		
	General register 9F	GR9F	R/W	H'FF	H'FFFFF696		
	Timer control register 9A	TCR9A	R/W	H'00	H'FFFFF698		11.2.3
	Timer control register 9B	TCR9B	R/W	H'00	H'FFFFF69A		
	Timer control register 9C	TCR9C	R/W	H'00	H'FFFFF69C		
	Timer status register 9	TSR9	R/(W)*	H'0000	H'FFFFF69E	16	11.2.5
	Timer interrupt enable register 9	TIER9	R/W	H'0000	H'FFFFF6A0		11.2.6

Channel	Name	tion	R/W	Value	Address	Size (Bits)	No.
10	Free-running counter 10AH	TCNT10AH	R/W	H'0000	H'FFFFF6C0	32	11.2.26
	Free-running counter 10AL	TCNT10AL	R/W	H'0001			
	Event counter 10B	TCNT10B	R/W	H'00	H'FFFFF6C4	8	
	Reload counter 10C	TCNT10C	R/W	H'0001	H'FFFFF6C6	16	
	Correction counter 10D	TCNT10D	R/W	H'00	H'FFFFF6C8	8	
	Correction angle counter 10E	TCNT10E	R/W	H'0000	H'FFFFF6CA	16	
	Correction angle counter 10F	TCNT10F	R/W	H'0001	H'FFFFF6CC		
	Free-running counter 10G	TCNT10G	R/W	H'0000	H'FFFFF6CE		
	Input capture register 10AH	ICR10AH	R	H'0000	H'FFFFF6D0	32	
	Input capture register 10AL	ICR10AL	R	H'0000			
	Output compare register 10AH	OCR10AH	R/W	H'FFFF	H'FFFFF6D4		
	Output compare register 10AL	OCR10AL	R/W	H'FFFF			
	Output compare register 10B	OCR10B	R/W	H'FF	H'FFFFF6D8	8	
	Reload register 10C	RLD10C	R/W	H'0000	H'FFFFF6DA	16	
	General register 10G	GR10G	R/W	H'FFFF	H'FFFFF6DC		
	Noise canceler counter 10H	TCNT10H	R/W	H'00	H'FFFFF6DE	8	
	Noise canceler register 10	NCR10	R/W	H'FF	H'FFFFF6E0		
	Timer I/O control register 10	TIOR10	R/W	H'00	H'FFFFF6E2		
	Timer control register 10	TCR10	R/W	H'00	H'FFFFF6E4		

Channel	Name	tion	R/W	Value	Address	Size (Bits)	No.
10	Correction counter clear register 10	TCCLR10	R/W	H'0000	H'FFFFF6E6	16	11.2.26
	Timer status register 10	TSR10	R/(W)*	H'0000	H'FFFFF6E8		
	Timer interrupt enable register 10	TIER10	R/W	H'0000	H'FFFFF6EA		
11	Free-running counter 11	TCNT11	R/W	H'0000	H'FFFFF5C0	16	11.2.15
	General register 11A	GR11A	R/W	H'FFFF	H'FFFFF5C2		11.2.20
	General register 11B	GR11B	R/W	H'FFFF	H'FFFFF5C4		
	Timer I/O control register 11	TIOR11	R/W	H'00	H'FFFFF5C6	8	11.2.4
	Timer control register 11	TCR11	R/W	H'00	H'FFFFF5C8		11.2.3
	Timer status register 11	TSR11	R/(W)*	H'0000	H'FFFFF5CA	16	11.2.5
	Timer interrupt enable register 11	TIER11	R/W	H'0000	H'FFFFF5CC		11.2.6

Note: * 0 write after a read

Overall Block Diagram of ATU-II: Figure 11.1 shows an overall block diagram of the ATU-II.



Legend:
TSTR1, 2, 3: Timer start registers (8 bits)

Interrupts:
ITV0–ITV2, OVI0, OVI1A, OVI1B, OVI2A, OVI2B, OVI3–OVI5, OVI11, ICI0A–ICI0D, IMI1A–IMI1H, CMI1, IMI2A–IMI2H, CMI2A–CMI2H, IMI3A–IMI3D, IMI4A–IMI4D, IMI5A–IMI5D, CMI6A–CMI6D, CMI7A–CMI7D, OSI8A–OSI8P, CMI9A–CMI9F, CMI10A, CMI10B, ICI10A, CMI10G, IMI11A, IMI11B

External pins:
TI0A–TI0D, TIO1A–TIO1H, TIO2A–TIO2H, TIO3A–TIO3D, TIO4A–TIO4D, TIO5A–TIO5D, TO6A–TO6D, TO7A–TO7D, TO8A–TO8P, TI9A–TI9F, TI10, TIO11A–TIO11B

Inter-module connection signals:
Signals to A/D converter, signals to direct memory access controller (DMAC), signals to advanced pulse controller (APC)

Figure 11.1 Overall Block Diagram of ATU-II

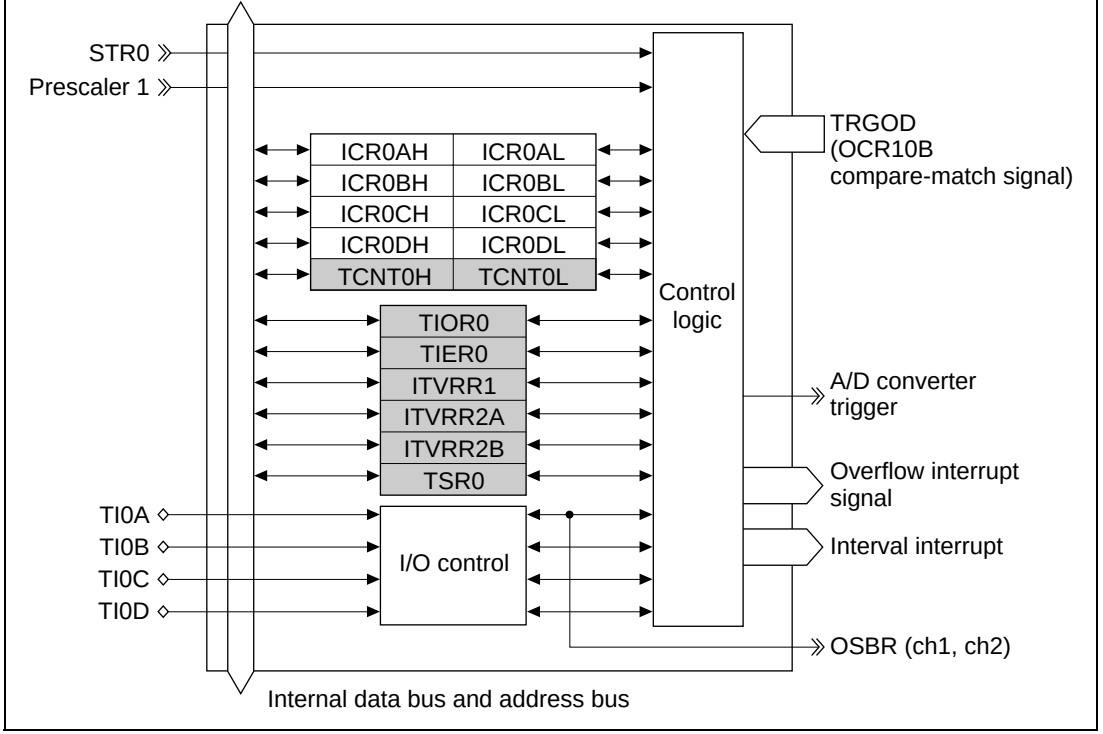


Figure 11.2 Block Diagram of Channel 0

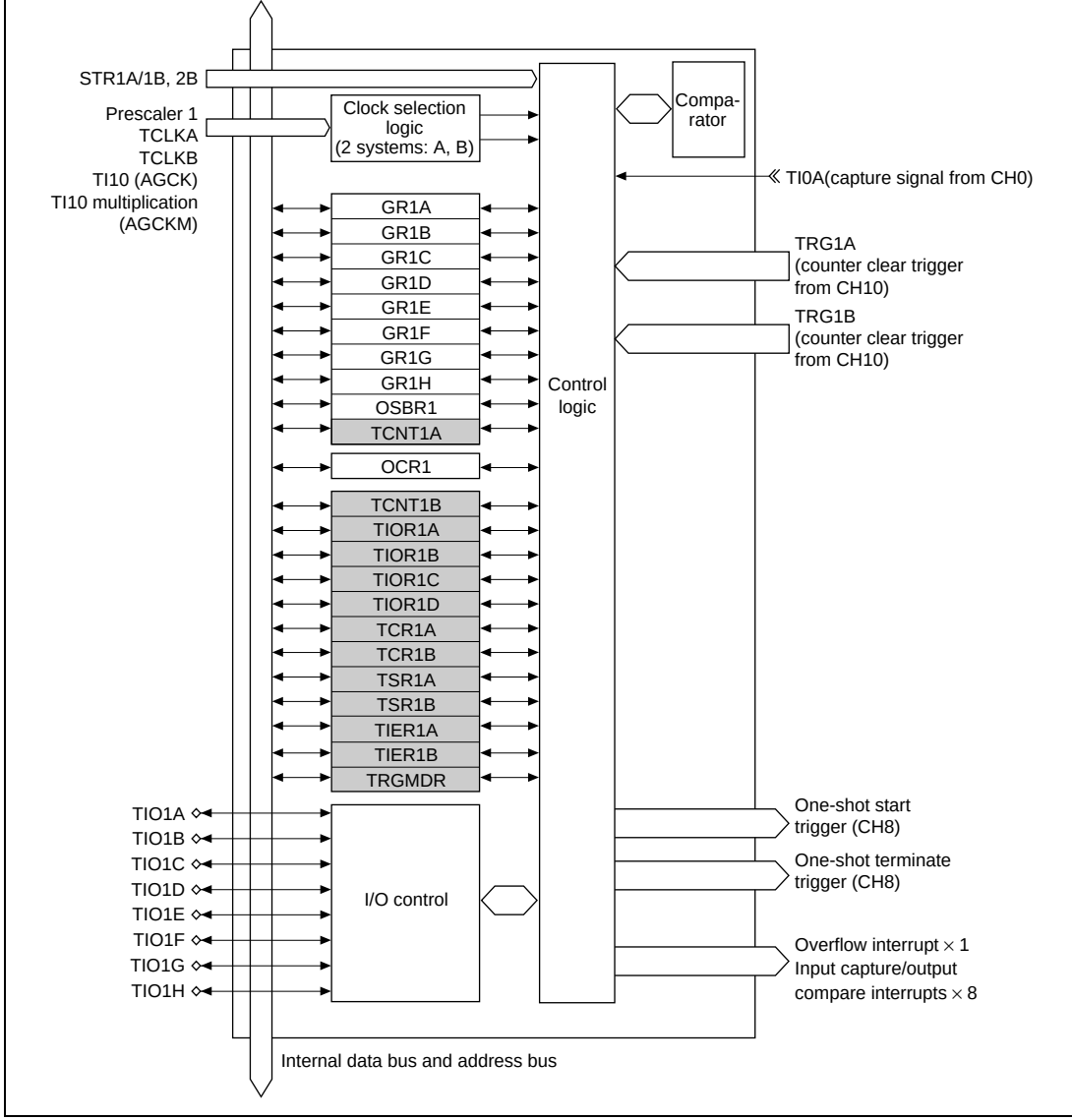


Figure 11.3 Block Diagram of Channel 1

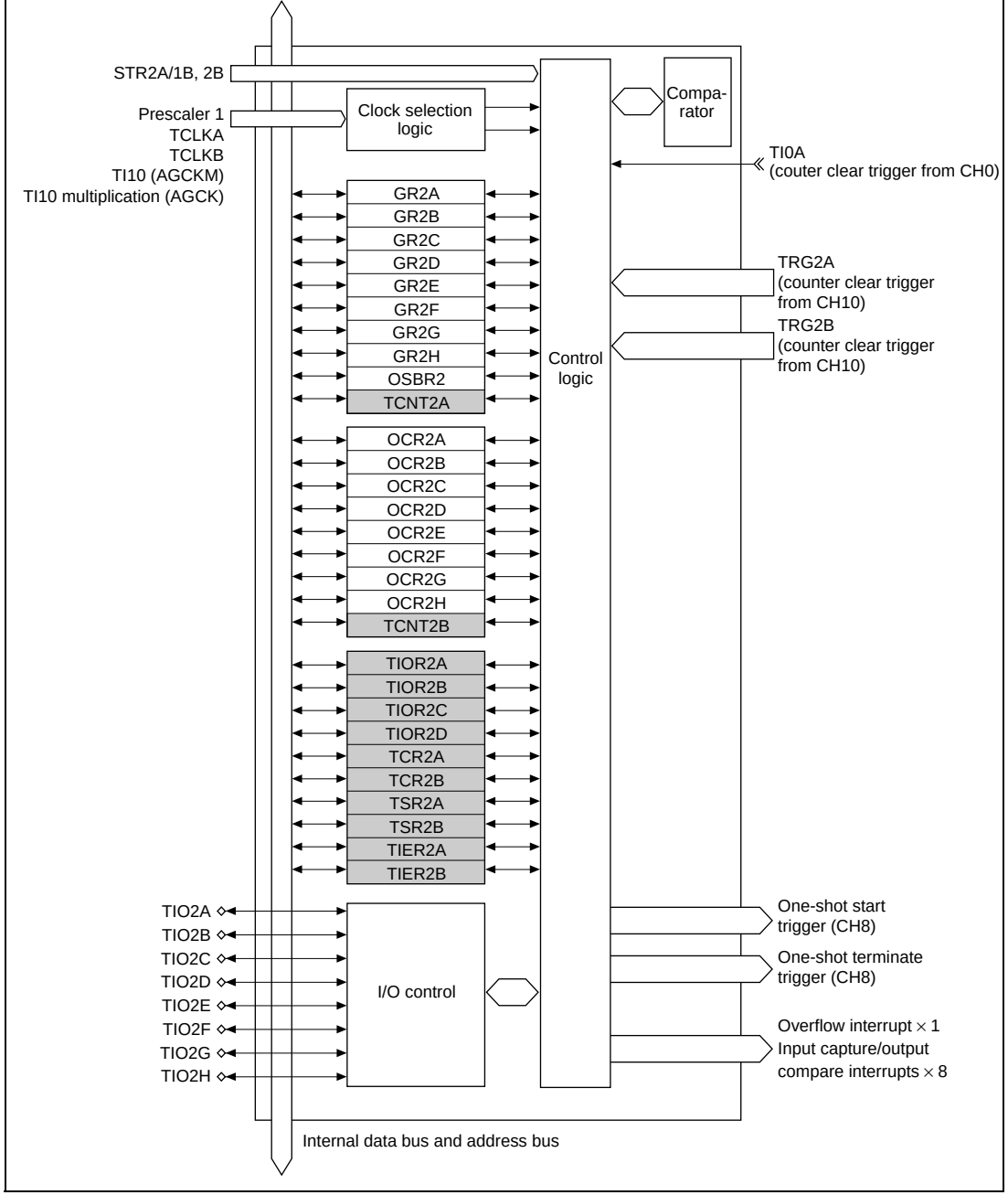


Figure 11.4 Block Diagram of Channel 2

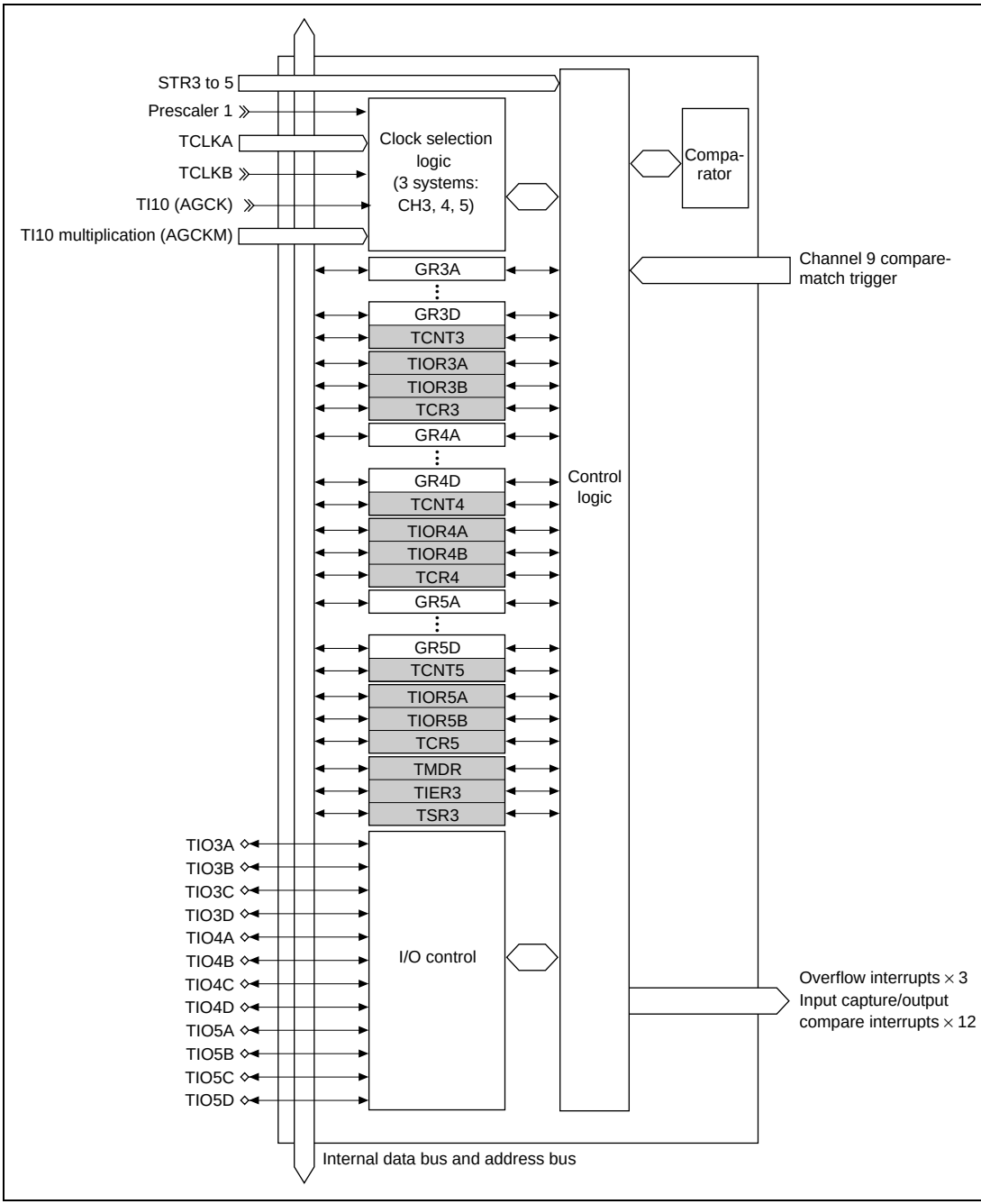


Figure 11.5 Block Diagram of Channels 3 to 5

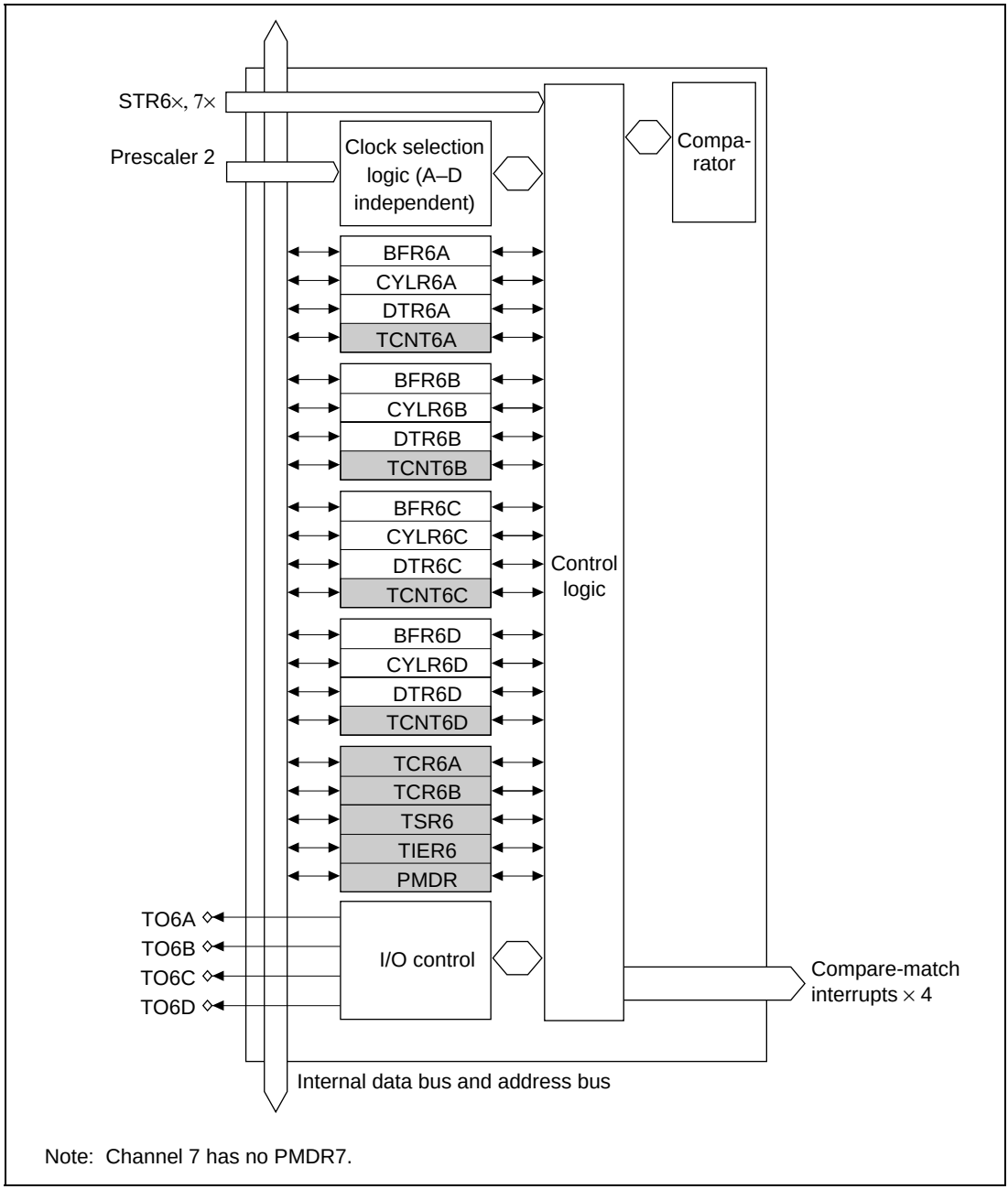


Figure 11.6 Block Diagram of Channel 6 (Same Configuration for Channel 7)

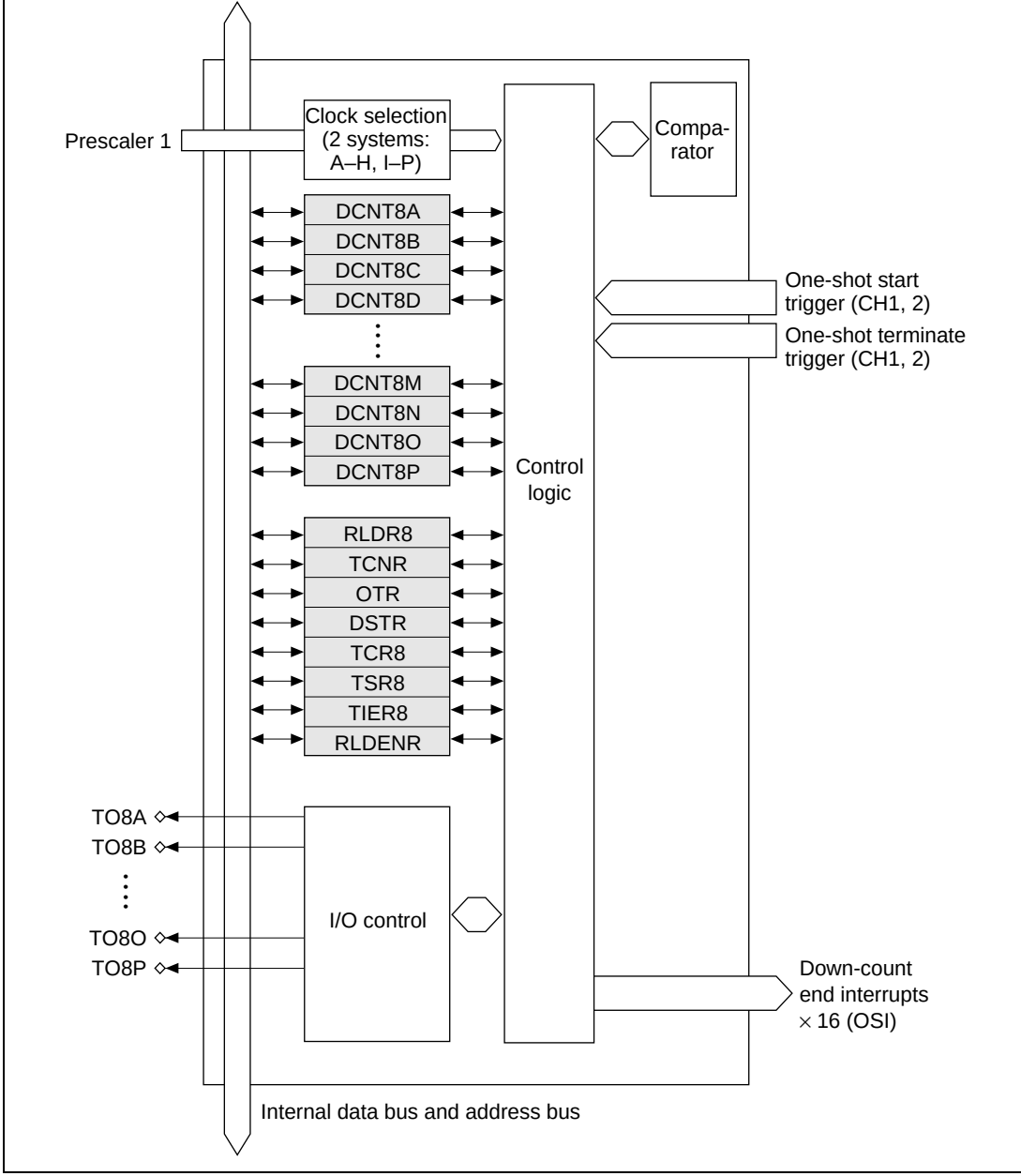


Figure 11.7 Block Diagram of Channel 8

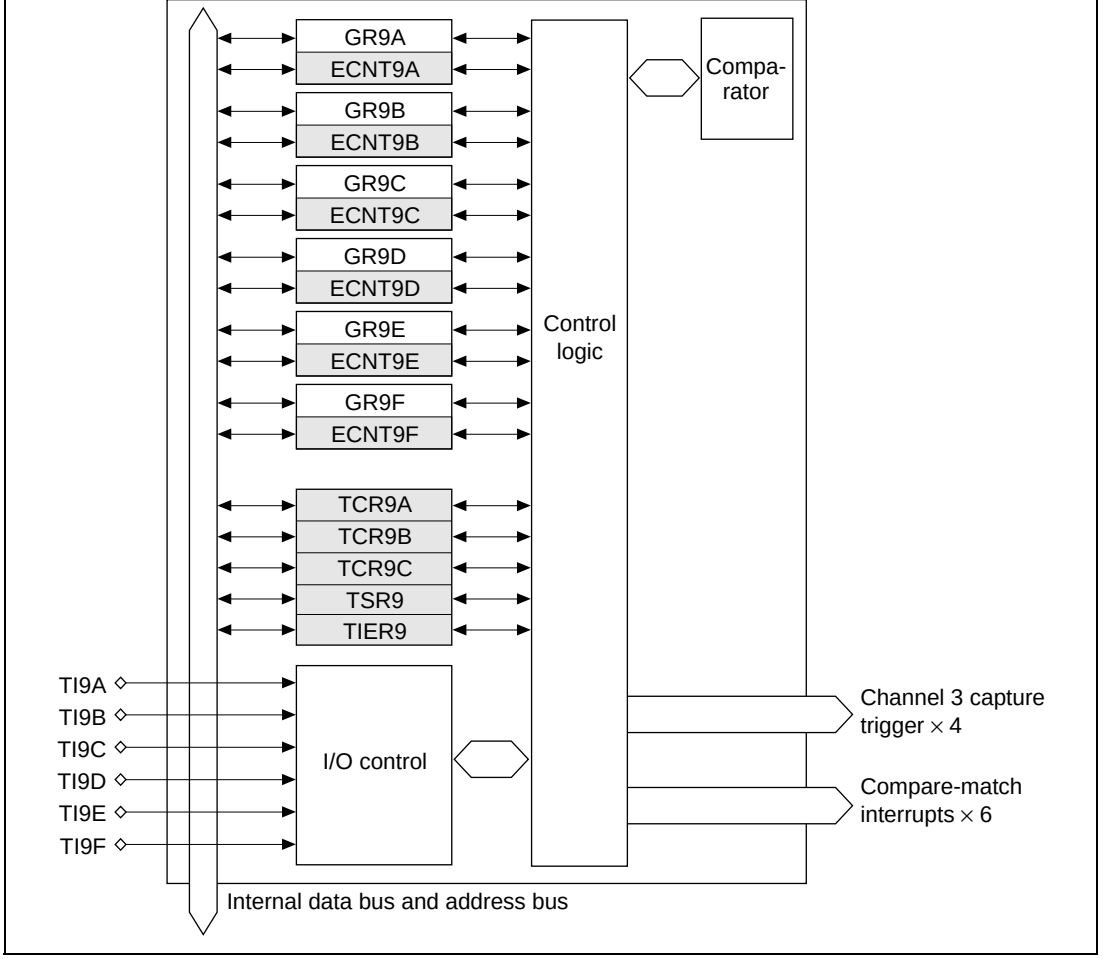


Figure 11.8 Block Diagram of Channel 9

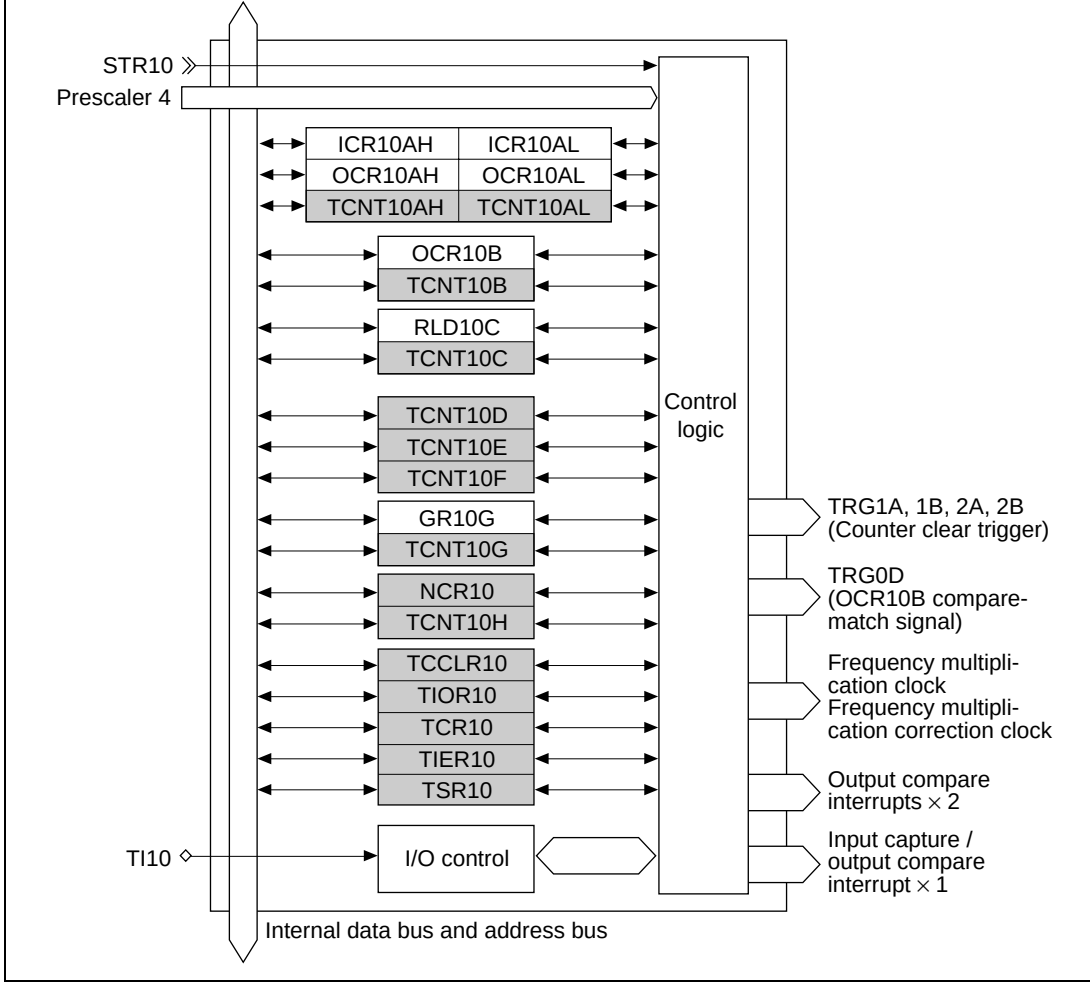


Figure 11.9 Block Diagram of Channel 10

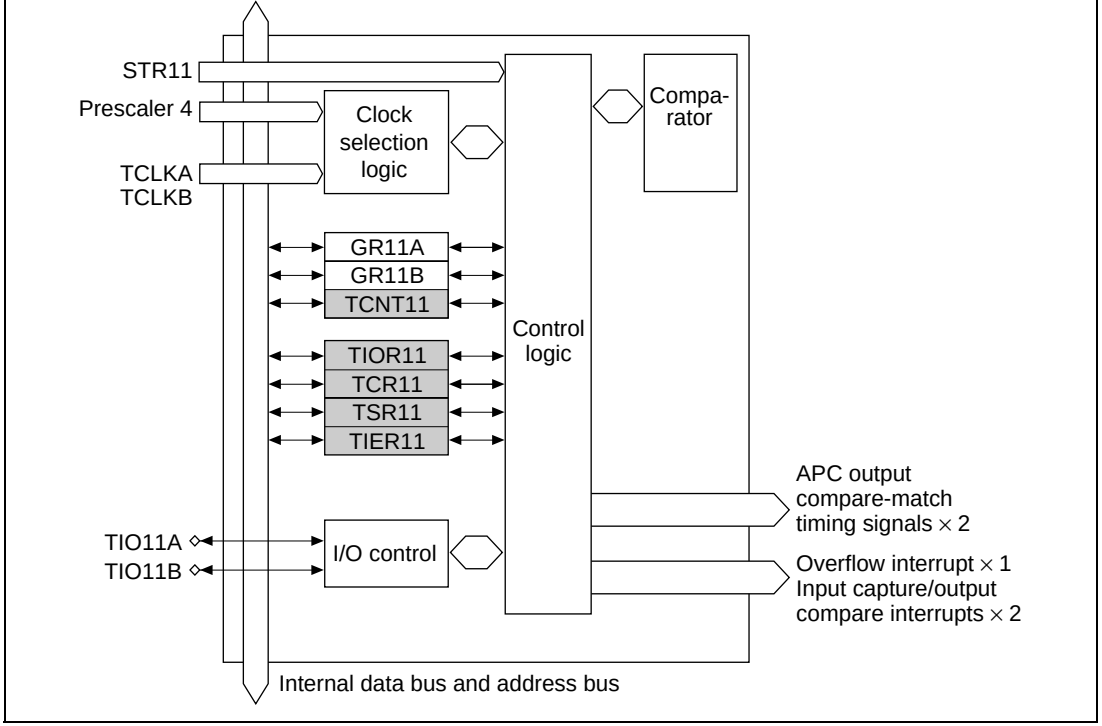


Figure 11.10 Block Diagram of Channel 11

Figure 11.11 shows the connections between channels and between modules in the ATU-II.

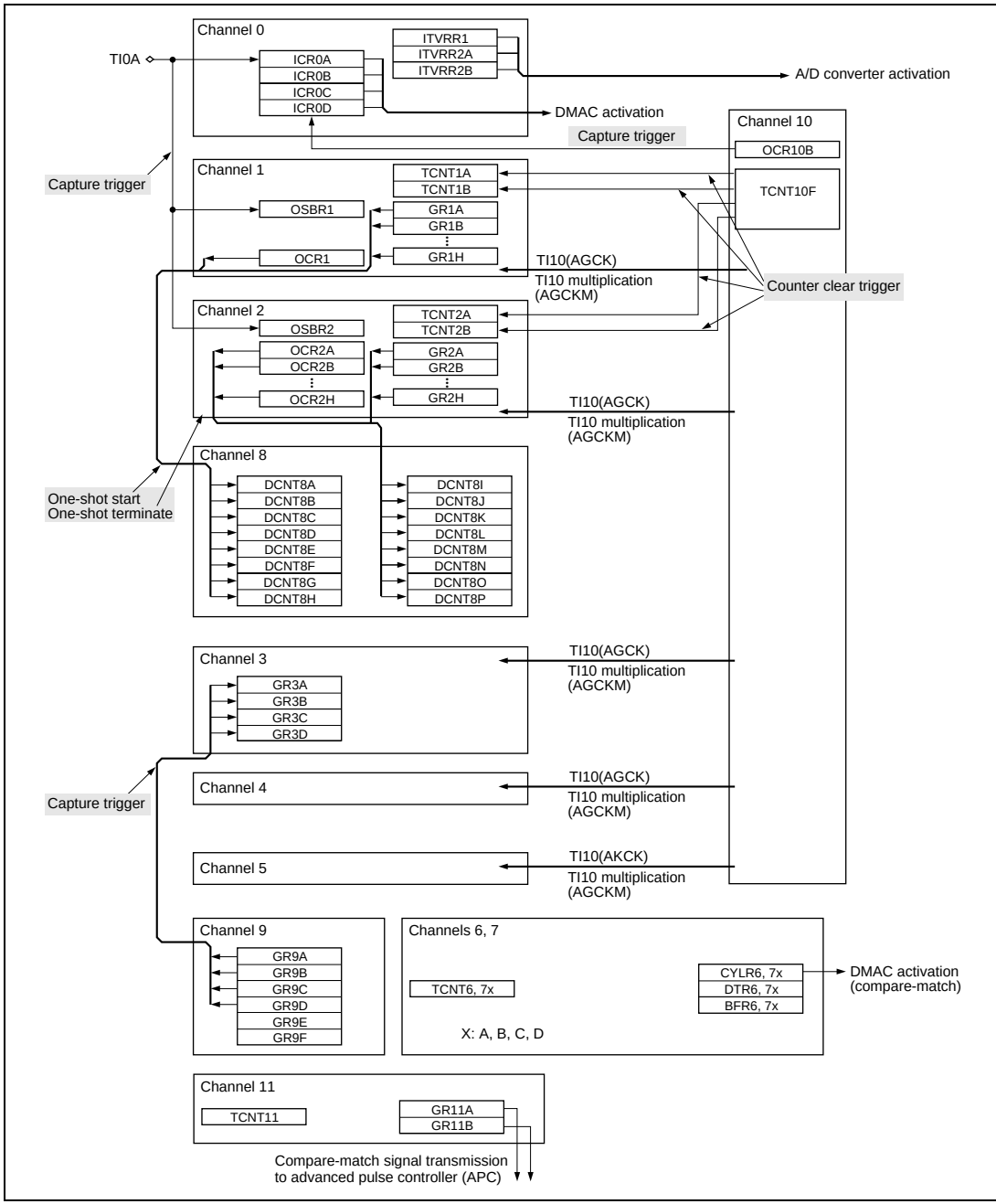


Figure 11.11 Inter-Module Communication Signals

Figure 11.12 shows a diagram of the ATU-II prescalers.

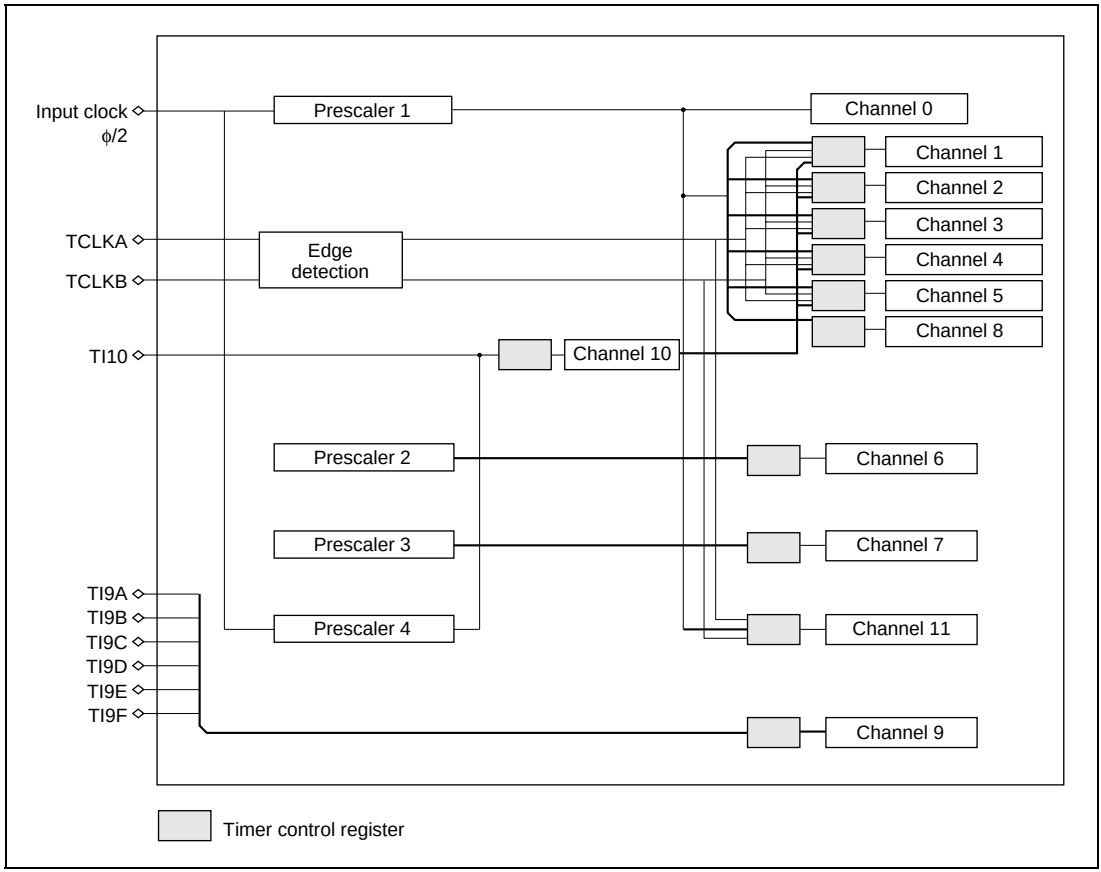


Figure 11.12 Prescaler Diagram

11.2.1 Timer Start Registers (TSTR)

The timer start registers (TSTR) are 8-bit registers. The ATU-II has three TSTR registers.

Channel	Abbreviation	Function
0, 1, 2, 3, 4, 5, 10	TSTR1	Free-running counter operation/stop setting
6, 7	TSTR2	
11	TSTR3	

Timer Start Register 1 (TSTR1)

Bit:	7	6	5	4	3	2	1	0
	STR10	STR5	STR4	STR3	STR1B, 2B	STR2A	STR1A	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TSTR1 is an 8-bit readable/writable register that starts and stops the free-running counter (TCNT) in channels 0 to 5 and 10.

TSTR1 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Counter Start 10 (STR10): Starts and stops channel 10 counters (TCNT10A, 10C, 10D, 10E, 10F, and 10G). TCNT10B and 10H are not stopped.

Bit 7: STR10	Description	
0	TCNT10 is halted	(Initial value)
1	TCNT10 counts	

- Bit 6—Counter Start 5 (STR5): Starts and stops free-running counter 5 (TCNT5).

Bit 6: STR5	Description	
0	TCNT5 is halted	(Initial value)
1	TCNT5 counts	

0	TCNT4 is halted	(Initial value)
1	TCNT4 counts	

- Bit 4—Counter Start 3 (STR3): Starts and stops free-running counter 3 (TCNT3).

Bit 4: STR3	Description	
0	TCNT3 is halted	(Initial value)
1	TCNT3 counts	

- Bit 3—Counter Start 1B, 2B (STR1B, STR2B): Starts and stops free-running counters 1B and 2B (TCNT1B, TCNT2B).

Bit 3: STR1B, STR2B	Description	
0	TCNT1B and TCNT2B are halted	(Initial value)
1	TCNT1B and TCNT2B count	

- Bit 2—Counter Start 2A (STR2A): Starts and stops free-running counter 2A (TCNT2A).

Bit 2: STR2A	Description	
0	TCNT2A is halted	(Initial value)
1	TCNT2A counts	

- Bit 1—Counter Start 1A (STR1A): Starts and stops free-running counter 1A (TCNT1A).

Bit 1: STR1A	Description	
0	TCNT1A is halted	(Initial value)
1	TCNT1A counts	

- Bit 0—Counter Start 0 (STR0): Starts and stops free-running counter 0 (TCNT0).

Bit 0: STR0	Description	
0	TCNT0 is halted	(Initial value)
1	TCNT0 counts	

	STR7D	STR7C	STR7B	STR7A	STR6D	STR6C	STR6B	STR6A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TSTR2 is an 8-bit readable/writable register that starts and stops the free-running counter (TCNT) in channels 6 and 7.

TSTR2 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Counter Start 7D (STR7D): Starts and stops free-running counter 7D (TCNT7D).

Bit 7: STR7D	Description	
0	TCNT7D is halted	(Initial value)
1	TCNT7D counts	

- Bit 6—Counter Start 7C (STR7C): Starts and stops free-running counter 7C (TCNT7C).

Bit 6: STR7C	Description	
0	TCNT7C is halted	(Initial value)
1	TCNT7C counts	

- Bit 5—Counter Start 7B (STR7B): Starts and stops free-running counter 7B (TCNT7B).

Bit 5: STR7B	Description	
0	TCNT7B is halted	(Initial value)
1	TCNT7B counts	

- Bit 4—Counter Start 7A (STR7A): Starts and stops free-running counter 7A (TCNT7A).

Bit 4: STR7A	Description	
0	TCNT7A is halted	(Initial value)
1	TCNT7A counts	

0	TCNT6D is halted	(Initial value)
1	TCNT6D counts	

- Bit 2—Counter Start 6C (STR6C): Starts and stops free-running counter 6C (TCNT6C).

Bit 2: STR6C	Description	
0	TCNT6C is halted	(Initial value)
1	TCNT6C counts	

- Bit 1—Counter Start 6B (STR6B): Starts and stops free-running counter 6B (TCNT6B).

Bit 1: STR6B	Description	
0	TCNT6B is halted	(Initial value)
1	TCNT6B counts	

- Bit 0—Counter Start 6A (STR6A): Starts and stops free-running counter 6A (TCNT6A).

Bit 0: STR6A	Description	
0	TCNT6A is halted	(Initial value)
1	TCNT6A counts	

Timer Start Register 3 (TSTR3)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	STR11
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

TSTR3 is an 8-bit readable/writable register that starts and stops the free-running counter (TCNT11) in channel 11.

TSTR3 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit 0: STR11	Description	
0	TCNT11 is halted	(Initial value)
1	TCNT11 counts	

11.2.2 Prescaler Registers (PSCR)

The prescaler registers (PSCR) are 8-bit registers. The ATU-II has four PSCR registers.

Channel	Abbreviation	Function
0, 1, 2, 3, 4, 5, 8, 11	PSCR1	Prescaler setting for respective channels
6	PSCR2	
7	PSCR3	
10	PSCR4	

PSCRx is an 8-bit writable register that enables the first-stage counter clock ϕ' input to each channel to be set to any value from Pϕ/1 to Pϕ/32.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	PSCxE	PSCxD	PSCxC	PSCxB	PSCxA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

x = 1 to 4

Input counter clock ϕ' is determined by setting PSCxA to PSCxE: ϕ' is Pϕ/1 when the set value is H'00, and Pϕ/32 when H'1F.

PSCRx is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

The internal clock ϕ' set with this register can undergo further second-stage scaling to create clock ϕ" for channels 1 to 8 and 11, the setting being made in the timer control register (TCR).

- Bits 7 to 5—Reserved: These bits cannot be modified.
- Bits 4 to 0—Prescaler (PSCxE, PSCxD, PSCxC, PSCxB, PSCxA): These bits specify frequency division of first-stage counter clock ø' input to the corresponding channel.

The timer control registers (TCR) are 8-bit registers. The ATU-II has 16 TCR registers: two each for channels 1 and 2, one each for channels 3, 4, 5, 8, and 11, two each for channels 6 and 7, and three for channel 9. For details of channel 10, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
1	TCR1A, TCR1B	Internal clock/external clock/TI10 input clock selection
2	TCR2A, TCR2B	
3	TCR3	
4	TCR4	
5	TCR5	
6	TCR6A, TCR6B	Internal clock selection
7	TCR7A, TCR7B	
8	TCR8	
9	TCR9A, TCR9B, TCR9C	External clock selection/setting of channel 3 trigger in event of compare-match
11	TCR11	Internal clock/external clock selection

Each TCR is an 8-bit readable/writable register that selects whether an internal clock or external clock is used for channels 1 to 5 and 11. For channels 6 to 8, TCR selects an internal clock, and for channel 9, an external clock.

When an internal clock is selected, TCR selects the value of ϕ" further scaled from clock ϕ' scaled with prescaler register (PSCR). Scaled clock ϕ" can be selected, for channels 1 to 8 and 11 only, from ϕ', ϕ'/2, ϕ'/4, ϕ'/8, ϕ'/16, and ϕ'/32 (only ϕ' is available for channel 0). Edge detection is performed on the rising edge.

When an external clock is selected, TCR selects whether TCLKA, TCLKB (channels 1 to 5 and 11 only), TI10 pin input (channels 1 to 5 only), or a TI10 pin input multiplied clock (channels 1 to 5 only) is used, and also performs edge selection.

Each TCR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	CKEGA1	CKEGA0	CKSELA3	CKSELA2	CKSELA1	CKSELA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

TCR1B, TCR2B

Bit:	7	6	5	4	3	2	1	0
	—	—	CKEGB1	CKEGB0	CKSELB3	CKSELB2	CKSELB1	CKSELB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 7 and 6—Reserved: These bits always read 0. The write value should always be 0.
- Bits 5 and 4—Clock Edge 1 and 0 (CKEGx1, CKEGx0): These bits select the count edge(s) for external clock TCLKA and TCLKB input.

Bit 5: CKEGx1	Bit 4: CKEGx0	Description	
0	0	Rising edges counted	(Initial value)
	1	Falling edges counted	
1	0	Both rising and falling edges counted	
	1	Count disabled	

x = A or B

- Bits 3 to 0—Clock Select A3 to A0, B3 to B0 (CKSELA3 to CKSELA0, CKSELB3 to CKSELB0): These bits select whether an internal clock or external clock is used.

  When an internal clock is selected, scaled clock ϕ" is selected from ϕ', ϕ'/2, ϕ'/4, ϕ'/8, ϕ'/16, and ϕ'/32.

  When an external clock is selected, TCLKA, TCLKB, TI10 pin input, or a TI10 pin input multiplied clock is selected.

  When TI10 pin input and TI10 pin input clock multiplication are selected, set CKEG1 and CKEG0 in TCR10 so that TI10 input is possible.

0	0	0	0	Internal clock φ": counting on φ (Initial value)
			1	Internal clock φ": counting on φ'/2
		1	0	Internal clock φ": counting on φ'/4
			1	Internal clock φ": counting on φ'/8
	1	0	0	Internal clock φ": counting on φ'/16
			1	Internal clock φ": counting on φ'/32
		1	0	External clock: counting on TCLKA pin input
			1	External clock: counting on TCLKB pin input
1	0	0	0	Counting on TI10 pin input (AGCK)
			1	Counting on multiplied (corrected)(AGCKM) TI10 pin input clock
		1	*	Setting prohibited
	1	*	*	Setting prohibited

x = A or B

*: Donít care

Timer Control Registers 3 to 5 (TCR3, TCR4, TCR5)

Bit:	7	6	5	4	3	2	1	0
	—	—	CKEG1	CKEG0	CKSEL3	CKSEL2	CKSEL1	CKSEL0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 7 and 6—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 5 and 4—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select the count edge(s) for external clock TCLKA and TCLKB input.

Bit 5: CKEG1	Bit 4: CKEG0	Description	
0	0	Rising edges counted	(Initial value)
	1	Falling edges counted	
1	0	Both rising and falling edges counted	
	1	Count disabled	

When an internal clock is selected, scaled clock φ" is selected from φ', φ'/2, φ'/4, φ'/8, φ'/16, and φ'/32.

When an external clock is selected, TCLKA, TCLKB, TI10 pin input, or a TI10 pin input multiplied clock is selected.

When TI10 pin input and TI10 pin input clock multiplication are selected, set CKEG1 and CKEG0 in TCR10 so that TI10 input is possible.

Bit 3: CKSEL3	Bit 2: CKSEL2	Bit 1: CKSEL1	Bit 0: CKSEL0	Description
0	0	0	0	Internal clock φ": counting on φ' (Initial value)
			1	Internal clock φ": counting on φ'/2
		1	0	Internal clock φ": counting on φ'/4
			1	Internal clock φ": counting on φ'/8
	1	0	0	Internal clock φ": counting on φ'/16
			1	Internal clock φ": counting on φ'/32
		1	0	External clock: counting on TCLKA pin input
			1	External clock: counting on TCLKB pin input
1	0	0	0	Counting on TI10 pin input (AGCK)
			1	Counting on multiplied (corrected)(AGCKM) TI10 pin input clock
		1	*	Setting prohibited
	1	*	*	Setting prohibited

*: Donít care

Bit:	7	6	5	4	3	2	1	0
	—	CKSELB2	CKSELB1	CKSELB0	—	CKSELA2	CKSELA1	CKSELA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TCR6B, TCR7B

Bit:	7	6	5	4	3	2	1	0
	—	CKSELD2	CKSELD1	CKSELD0	—	CKSELC2	CKSELC1	CKSELC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bits 6 to 4—Clock Select B2 to B0, D2 to D0 (CKSELB2 to CKSELB0, CKSELD2 to CKSELD0): These bits select clock φ", scaled from the internal clock source, from φ', φ'/2, φ'/4, φ'/8, φ'/16, and φ'/32.

Bit 6: CKSELx2	**Bit 5: CKSELx1**	**Bit 4: CKSELx0**	**Description**	
0	0	0	Internal clock φ": counting on φ'	(Initial value)
		1	Internal clock φ": counting on φ'/2	
	1	0	Internal clock φ": counting on φ'/4	
		1	Internal clock φ": counting on φ'/8	
1	0	0	Internal clock φ": counting on φ'/16	
		1	Internal clock φ": counting on φ'/32	
	1	0	Setting prohibited	
		1	Setting prohibited	

x = B or D

CKSELC0): These bits select clock φ", scaled from the internal clock source, from φ', φ'/2, φ'/4, φ'/8, φ'/16, and φ'/32.

Bit 2: CKSELx2	Bit 1 CKSELx1	Bit 0 CKSELx0	Description	
0	0	0	Internal clock φ": counting on φ'	(Initial value)
		1	Internal clock φ": counting on φ'/2	
	1	0	Internal clock φ": counting on φ'/4	
		1	Internal clock φ": counting on φ'/8	
1	0	0	Internal clock φ": counting on φ'/16	
		1	Internal clock φ": counting on φ'/32	
	1	0	Setting prohibited	
		1	Setting prohibited	

x = A or B

Timer Control Register 8 (TCR8)

Bit:	7	6	5	4	3	2	1	0
	—	CKSELB2	CKSELB1	CKSELB0	—	CKSELA2	CKSELA1	CKSELA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

The CKSELAx bits relate to DCNT8A to DCNT8H, and the CKSELBx bits relate to DCNT8I to DCNT8P.

DCNT8I to DCNT8P, select clock ϕ", scaled from the internal clock source, from ϕ', ϕ'/2, ϕ'/4, ϕ'/8, ϕ'/16, and ϕ'/32.

Bit 6: CKSELB2	Bit 5: CKSELB1	Bit 4: CKSELB0	Description	
0	0	0	Internal clock ϕ": counting on ϕ'	(Initial value)
		1	Internal clock ϕ": counting on ϕ'/2	
	1	0	Internal clock ϕ": counting on ϕ'/4	
		1	Internal clock ϕ": counting on ϕ'/8	
1	0	0	Internal clock ϕ": counting on ϕ'/16	
		1	Internal clock ϕ": counting on ϕ'/32	
	1	0	Setting prohibited	
		1	Setting prohibited	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bits 2 to 0—Clock Select A2 to A0 (CKSELA2 to CKSELA0): These bits, relating to counters DCNT8A to DCNT8H, select clock ϕ", scaled from the internal clock source, from ϕ', ϕ'/2, ϕ'/4, ϕ'/8, ϕ'/16, and ϕ'/32.

Bit 2: CKSELA2	Bit 1: CKSELA1	Bit 0: CKSELA0	Description	
0	0	0	Internal clock ϕ": counting on ϕ'	(Initial value)
		1	Internal clock ϕ": counting on ϕ'/2	
	1	0	Internal clock ϕ": counting on ϕ'/4	
		1	Internal clock ϕ": counting on ϕ'/8	
1	0	0	Internal clock ϕ": counting on ϕ'/16	
		1	Internal clock ϕ": counting on ϕ'/32	
	1	0	Setting prohibited	
		1	Setting prohibited	

Bit:	7	6	5	4	3	2	1	0
	—	TRG3BEN	EGSELB1	EGSELB0	—	TRG3AEN	EGSELA1	EGSELA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TCR9B

Bit:	7	6	5	4	3	2	1	0
	—	TRG3DEN	EGSELD1	EGSELD0	—	TRG3CEN	EGSELC1	EGSELC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TCR9C

Bit:	7	6	5	4	3	2	1	0
	—	—	EGSELF1	EGSELF0	—	—	EGSELE1	EGSELE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—Trigger Channel 3BEN, 3DEN (TRG3BEN, TRG3DEN): These bits select the channel 9 event counter compare-match signal channel 3 input capture trigger.

Bit 6: TRG3xEN	Description
0	Channel 3 input capture trigger in event of channel 9 compare-match (ECNT9x = GR9x) is disabled (Initial value)
1	Channel 3 input capture trigger in event of channel 9 compare-match (ECNT9x = GR9x) is enabled

x = B or D

Bit 5: EGSELx1	Bit 4: EGSELx0	Description	
0	0	Count disabled	(Initial value)
	1	Rising edges counted	
1	0	Falling edges counted	
	1	Both rising and falling edges counted	

x = B, D, or F

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—Trigger Channel 3AEN, 3CEN (TRG3AEN, TRG3CEN): These bits select the channel 9 event counter compare-match signal channel 3 input capture trigger.

Bit 2: TRG3xEN	Description	
0	Channel 3 input capture trigger in event of channel 9 compare-match (ECNT9x = GR9x) is disabled	(Initial value)
1	Channel 3 input capture trigger in event of channel 9 compare-match (ECNT9x = GR9x) is enabled	

x = A or C

- Bits 1 and 0—Edge Select A1, A0, C1, C0, E1, E0 (EGSELA1, EGSELA0, EGSELC1, EGSELC0, EGSELE1, EGSELE0): These bits select the event counter counted edge(s).

Bit 1: EGSELx1	Bit 0: EGSELx0	Description	
0	0	Count disabled	(Initial value)
	1	Rising edges counted	
1	0	Falling edges counted	
	1	Both rising and falling edges counted	

x = A, C, or E

	—	—	CKEG1	CKEG0	—	CKSELA2	CKSELA1	CKSELA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W

- Bits 7, 6, and 3—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 5 and 4—Edge Select: These bits select the event counter counted edge(s).

Bit 5: CKEG1	Bit 4: CKEG0	Description	
0	0	Rising edges counted	(Initial value)
	1	Falling edges counted	
1	0	Both rising and falling edges counted	
	1	Count disabled	

- Bits 2 to 0—Clock Select A2 to A0 (CKSELA2 to CKSELA0): These bits select clock φ", scaled from the internal clock source, from φ', φ'/2, φ'/4, φ'/8, φ'/16, and φ'/32.

Bit 2: CKSELA2	Bit 1: CKSELA1	Bit 0: CKSELA0	Description	
0	0	0	Internal clock φ": counting on φ'	(Initial value)
		1	Internal clock φ": counting on φ'/2	
	1	0	Internal clock φ": counting on φ'/4	
		1	Internal clock φ": counting on φ'/8	
1	0	0	Internal clock φ": counting on φ'/16	
		1	Internal clock φ": counting on φ'/32	
	1	0	External clock: counting on TCLKA pin input	
		1	External clock: counting on TCLKB pin input	

The timer I/O control registers (TIOR) are 8-bit registers. The ATU-II has 16 TIOR registers: one for channel 0, four each for channels 1 and 2, two each for channels 3 to 5, and one for channel 11. For details of channel 10, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
0	TIOR0	ICR0 edge detection setting
1	TIOR1A–1D	GR input capture/compare-match switching, edge detection/ output value setting
2	TIOR2A–2D	
3	TIOR3A, TIOR3B	GR input capture/compare-match switching, edge detection/output value setting, TCNT3 to TCNT5 clear enable/disable setting
4	TIOR4A, TIOR4B	
5	TIOR5A, TIOR5B	
11	TIOR11	GR input capture/compare-match switching, edge detection/output value setting

Each TIOR is an 8-bit readable/writable register used to select the functions of dedicated input capture registers and general registers.

For dedicated input capture registers (ICR), TIOR performs edge detection setting.

For general registers (GR), TIOR selects use as an input capture register or output compare register, and performs edge detection setting. For channels 3 to 5, TIOR also selects enabling or disabling of free-running counter (TCNT) clearing in the event of a compare-match.

Timer I/O Control Register 0 (TIOR0)

Bit:	7	6	5	4	3	2	1	0
	IO0D1	IO0D0	IO0C1	IO0C0	IO0B1	IO0B0	IO0A1	IO0A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TIOR0 specifies edge detection for input capture registers ICR0A to ICR0D.

TIOR0 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit 7: IO0D1	Bit 6: IO0D0	Description
0	0	Input capture disabled (input capture possible in TCNT10B compare-match) (Initial value)
	1	Input capture in ICR0D on rising edge
1	0	Input capture in ICR0D on falling edge
	1	Input capture in ICR0D on both rising and falling edges

- Bits 5 and 4—I/O Control 0C1 and 0C0 (IO0C1, IO0C0): These bits select TI0C pin input capture signal edge detection.

Bit 5: IO0C1	Bit 4: IO0C0	Description
0	0	Input capture disabled (Initial value)
	1	Input capture in ICR0C on rising edge
1	0	Input capture in ICR0C on falling edge
	1	Input capture in ICR0C on both rising and falling edges

- Bits 3 and 2—I/O Control 0B1 and 0B0 (IO0B1, IO0B0): These bits select TI0B pin input capture signal edge detection.

Bit 3: IO0B1	Bit 2: IO0B0	Description
0	0	Input capture disabled (Initial value)
	1	Input capture in ICR0B on rising edge
1	0	Input capture in ICR0B on falling edge
	1	Input capture in ICR0B on both rising and falling edges

- Bits 1 and 0—I/O Control 0A1 and 0A0 (IO0A1, IO0A0): These bits select TI0A pin input capture signal edge detection.

Bit 1: IO0A1	Bit 0: IO0A0	Description
0	0	Input capture disabled (Initial value)
	1	Input capture in ICR0A on rising edge
1	0	Input capture in ICR0A on falling edge
	1	Input capture in ICR0A on both rising and falling edges

Bit:	7	6	5	4	3	2	1	0
	—	IO1B2	IO1B1	IO1B0	—	IO1A2	IO1A1	IO1A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR1B

Bit:	7	6	5	4	3	2	1	0
	—	IO1D2	IO1D1	IO1D0	—	IO1C2	IO1C1	IO1C0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR1C

Bit:	7	6	5	4	3	2	1	0
	—	IO1F2	IO1F1	IO1F0	—	IO1E2	IO1E1	IO1E0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR1D

Bit:	7	6	5	4	3	2	1	0
	—	IO1H2	IO1H1	IO1H0	—	IO1G2	IO1G1	IO1G0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Registers TIOR1A to TIOR1D specify whether general registers GR1A to GR1H are used as input capture or compare-match registers, and also perform edge detection and output value setting.

Each TIOR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

IO1D2 to IO1D0, IOF12 to IO1F0, IO1H2 to IO1H0): These bits select the general register (GR) function.

Bit 6: IO1x2	Bit 5: IO1x1	Bit 4: IO1x0	Description	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register	Input capture disabled (GR cannot be written to)
		1		Input capture in GR on rising edge at TIO1x pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIO1x pin (GR cannot be written to)
		1		Input capture in GR on both rising and falling edges at TIO1x pin (GR cannot be written to)

x = B, D, F, or H

IO1C2 to IO1C0, IO1E2 to IO1E0, IO1G2 to IO1G0): These bits select the general register (GR) function.

Bit 2: IO1x2	Bit 1: IO1x1	Bit 0: IO1x0	Description	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register	Input capture disabled
		1		Input capture in GR on rising edge at TIO1x pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIO1x pin (GR cannot be written to)
		1		Input capture in GR on both rising and falling edges at TIO1x pin (GR cannot be written to)

x = A, C, E, or G

Bit:	7	6	5	4	3	2	1	0
	—	IO2B2	IO2B1	IO2B0	—	IO2A2	IO2A1	IO2A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR2B

Bit:	7	6	5	4	3	2	1	0
	—	IO2D2	IO2D1	IO2D0	—	IO2C2	IO2C1	IO2C0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR2C

Bit:	7	6	5	4	3	2	1	0
	—	IO2F2	IO2F1	IO2F0	—	IO2E2	IO2E1	IO2E0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR2D

Bit:	7	6	5	4	3	2	1	0
	—	IO2H2	IO2H1	IO2H0	—	IO2G2	IO2G1	IO2G0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Registers TIOR2A to TIOR2D specify whether general registers GR2A to GR2H are used as input capture or compare-match registers, and also perform edge detection and output value setting.

Each TIOR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

IO2D2 to IO2D0, IO2F2 to IO2F0, IO2H2 to IO2H0): These bits select the general register (GR) function.

Bit 6: IO2x2	Bit 5: IO2x1	Bit 4: IO2x0	Description	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register	Input capture disabled
		1		Input capture in GR on rising edge at TIO2x pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIO2x pin (GR cannot be written to)
		1		Input capture in GR on both rising and falling edges at TIO2x pin (GR cannot be written to)

x = B, D, F, or H

IO2C2 to IO2C0, IO2E2 to IO2E0, IO2G2 to IO2G0): These bits select the general register (GR) function.

Bit 2: IO2x2	Bit 1: IO2x1	Bit 0: IO2x0	Description	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register	Input capture disabled
		1		Input capture in GR on rising edge at TIO2x pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIO2x pin (GR cannot be written to)
		1		Input capture in GR on both rising and falling edges at TIO2x pin (GR cannot be written to)

x = A, C, E, or G

Timer I/O Control Registers 3A, 3B, 4A, 4B, 5A, 5B (TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B)

TIOR3A, TIOR4A, TIOR5A

Bit:	7	6	5	4	3	2	1	0
	CCIxB	IOxB2	IOxB1	IOxB0	CCIxA	IOxA2	IOxA1	IOxA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

x = 3 to 5

TIOR3B, TIOR4B, TIOR5B

Bit:	7	6	5	4	3	2	1	0
	CCIxD	IOxD2	IOxD1	IOxD0	CCIxC	IOxC2	IOxC1	IOxC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

x = 3 to 5

match registers, and also perform edge detection and output value setting. They also select enabling or disabling of free-running counter (TCNT3 to TCNT5) clearing.

Each TIOR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Clear Counter Enable Flag 3B, 4B, 5B, 3D, 4D, 5D (CCI3B, CCI4B, CCI5B, CCI3D, CCI4D, CCI5D): These bits select enabling or disabling of free-running counter (TCNT) clearing.

Bit 7: CCIxx	Description	
0	TCNT clearing disabled	(Initial value)
1	TCNT cleared on GR compare-match	

xx = 3B, 4B, 5B, 3D, 4D, or 5D

TCNT is cleared on compare-match only when GR is functioning as an output compare register.

- Bits 6 to 4—I/O Control 3B2 to 3B0, 4B2 to 4B0, 5B2 to 5B0, 3D2 to 3D0, 4D2 to 4D0, 5D2 to 5D0 (IO3B2 to IO3B0, IO4B2 to IO4B0, IO5B2 to IO5B0, IO3D2 to IO3D0, IO4D2 to IO4D0, IO5D2 to IO5D0): These bits select the general register (GR) function.

Bit 6: IOxx2	Bit 5: IOxx1	Bit 4: IOxx0	Description	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register (input capture by channel 3 and 9 compare-match enabled)	Input capture disabled (In channel 3 only, GR cannot be written to)
		1		Input capture in GR on rising edge at TIOxx pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIOxx pin (GR cannot be written to)
		1		Input capture in GR on both rising and falling edges at TIOxx pin (GR cannot be written to)

xx = 3B, 4B, 5B, 3D, 4D, or 5D

clearing.

Bit 3: CCIxx	Description	
0	TCNT clearing disabled	(Initial value)
1	TCNT cleared on GR compare-match	

xx = 3A, 4A, 5A, 3C, 4C, or 5C

TCNT is cleared on compare-match only when GR is functioning as an output compare register.

- Bits 2 to 0—I/O Control 3A2 to 3A0, 4A2 to 4A0, 5A2 to 5A0, 3C2 to 3C0, 4C2 to 4C0, 5C2 to 5C0 (IO3A2 to IO3A0, IO4A2 to IO4A0, IO5A2 to IO5A0, IO3C2 to IO3C0, IO4C2 to IO4C0, IO5C2 to IO5C0): These bits select the general register (GR) function.

Bit 2: IOxx2	Bit 1: IOxx1	Bit 0: IOxx0	Description	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register (input capture by channel 3 and 9 compare-match enabled)	Input capture disabled (In channel 3 only, GR cannot be written to)
		1		Input capture in GR on rising edge at TIOxx pin (GR connot be written to)
	1	0		Input capture in GR on falling edge at TIOxx pin (GR connot be written to)
		1		Input capture in GR on both rising and falling edges at TIOxx pin (GR connot be written to)

xx = 3A, 4A, 5A, 3C, 4C, or 5C

Bit:	7	6	5	4	3	2	1	0
	—	IO11B2	IO11B1	IO11B0	—	IO11A2	IO11A1	IO11A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

TIOR11 specifies whether general registers GR11A and GR11B are used as input capture or compare-match registers, and also performs edge detection and output value setting.

TIOR11 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bits 6 to 4—I/O Control 11B2 to 11B0 (IO11B2 to IO11B0): These bits select the general register (GR) function.

Bit 6: IO11B2	**Bit 5: IO11B1**	**Bit 4: IO11B0**	**Description**	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register	Input capture disabled
		1		Input capture in GR on rising edge at TIO11B pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIO11B pin (GR cannot be written to)
		1		Input capture in GR on both rising and falling edges at TIO11B pin (GR cannot be written to)

register (GR) function.

Bit 2: IO11A2	Bit 1: IO11A1	Bit 0: IO11A0	Description	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register	Input capture disabled
		1		Input capture in GR on rising edge at TIO11A pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIO11A pin (GR cannot be written to)
		1		Input capture in GR on both rising and falling edges at TIO11A pin (GR cannot be written to)

11.2.5 Timer Status Registers (TSR)

The timer status registers (TSR) are 16-bit registers. The ATU-II has 11 TSR registers: one each for channels 0, 6 to 9, and 11, two each for channels 1 and 2, and one for channels 3 to 5. For details of channel 10, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
0	TSR0	Indicates input capture, interval interrupt, and overflow status
1	TSR1A, TSR1B	Indicate input capture, compare-match, and overflow status
2	TSR2A, TSR2B	
3	TSR3	Indicates input capture, compare-match, and overflow status
4		
5		
6	TSR6	Indicate cycle register compare-match status
7	TSR7	
8	TSR8	Indicates down-counter output end (low) status
9	TSR9	Indicates event counter compare-match status
11	TSR11	Indicates input capture, compare-match, and overflow status

and 11 general register input capture or compare-match, channel 6 and 7 compare-matches, channel 8 down-counter output end, and channel 9 event counter compare-matches.

Each flag is an interrupt source, and issues an interrupt request to the CPU if the interrupt is enabled by the corresponding bit in the timer interrupt enable register (TIER).

Each TSR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Timer Status Register 0 (TSR0)

TSR0 indicates the status of channel 0 interval interrupts, input capture, and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	IIF2B	IIF2A	IIF1	OVF0	ICF0D	ICF0C	ICF0B	ICF0A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

- Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 7—Interval Interrupt Flag 2B (IIF2B): Status flag that indicates the generation of an interval interrupt.

Bit 7: IIF2B	Description
0	[Clearing condition] (Initial value) When IIF2B is read while set to 1, then 0 is written to IIF2B
1	[Setting condition] When interval interrupt selected by ITVRR2B is generated

Bit 6: IIF2A	Description
0	[Clearing condition] (Initial value) When IIF2A is read while set to 1, then 0 is written to IIF2A
1	[Setting condition] When interval interrupt selected by ITVRR2A is generated

- Bit 5—Interval Interrupt Flag 1 (IIF1): Status flag that indicates the generation of an interval interrupt.

Bit 5: IIF1	Description
0	[Clearing condition] (Initial value) When IIF1 is read while set to 1, then 0 is written to IIF1
1	[Setting condition] When interval interrupt selected by ITVRR1 is generated

- Bit 4—Overflow Flag 0 (OVF0): Status flag that indicates TCNT0 overflow.

Bit 4: OVF0	Description
0	[Clearing condition] (Initial value) When OVF0 is read while set to 1, then 0 is written to OVF0
1	[Setting condition] When the TCNT0 value overflows (from H'FFFFFFFF to H'00000000)

- Bit 3—Input Capture Flag 0D (ICF0D): Status flag that indicates ICR0D input capture.

Bit 3: ICF0D	Description
0	[Clearing condition] (Initial value) When ICF0D is read while set to 1, then 0 is written to ICF0D
1	[Setting condition] When the TCNT0 value is transferred to the input capture register by an input capture signal. Also set by input capture with a channel 10 compare match as the trigger

0	[Clearing condition] (Initial value) When ICF0C is read while set to 1, then 0 is written to ICF0C
1	[Setting condition] When the TCNT0 value is transferred to the input capture register by an input capture signal

- Bit 1—Input Capture Flag 0B (ICF0B): Status flag that indicates ICR0B input capture.

Bit 1: ICF0B	Description
0	[Clearing condition] (Initial value) When ICF0B is read while set to 1, then 0 is written to ICF0B
1	[Setting condition] When the TCNT0 value is transferred to the input capture register by an input capture signal

- Bit 0—Input Capture Flag 0A (ICF0A): Status flag that indicates ICR0A input capture.

Bit 0: ICF0A	Description
0	[Clearing condition] (Initial value) When ICF0A is read while set to 1, then 0 is written to ICF0A
1	[Setting condition] When the TCNT0 value is transferred to the input capture register by an input capture signal

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVF1A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	IMF1H	IMF1G	IMF1F	IMF1E	IMF1D	IMF1C	IMF1B	IMF1A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, to clear the flag.

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Flag 1A (OVF1A): Status flag that indicates TCNT1A overflow.

Bit 8: OVF1A	**Description**
0	[Clearing condition] (Initial value) When OVF1A is read while set to 1, then 0 is written to OVF1A
1	[Setting condition] When the TCNT1A value overflows (from H'FFFF to H'0000)

- Bit 7—Input Capture/Compare-Match Flag 1H (IMF1H): Status flag that indicates GR1H input capture or compare-match.

Bit 7: IMF1H	**Description**
0	[Clearing condition] (Initial value) When IMF1H is read while set to 1, then 0 is written to IMF1H
1	[Setting conditions] • When the TCNT1A value is transferred to GR1H by an input capture signal while GR1H is functioning as an input capture register • When TCNT1A = GR1H while GR1H is functioning as an output compare register

Bit 6: IMF1G	Description
0	[Clearing condition] (Initial value) When IMF1G is read while set to 1, then 0 is written to IMF1G
1	[Setting conditions] • When the TCNT1A value is transferred to GR1G by an input capture signal while GR1G is functioning as an input capture register • When TCNT1A = GR1G while GR1G is functioning as an output compare register

- Bit 5—Input Capture/Compare-Match Flag 1F (IMF1F): Status flag that indicates GR1F input capture or compare-match.

Bit 5: IMF1F	Description
0	[Clearing condition] (Initial value) When IMF1F is read while set to 1, then 0 is written to IMF1F
1	[Setting conditions] • When the TCNT1A value is transferred to GR1F by an input capture signal while GR1F is functioning as an input capture register • When TCNT1A = GR1F while GR1F is functioning as an output compare register

- Bit 4—Input Capture/Compare-Match Flag 1E (IMF1E): Status flag that indicates GR1E input capture or compare-match.

Bit 4: IMF1E	Description
0	[Clearing condition] (Initial value) When IMF1E is read while set to 1, then 0 is written to IMF1E
1	[Setting conditions] • When the TCNT1A value is transferred to GR1E by an input capture signal while GR1E is functioning as an input capture register • When TCNT1A = GR1E while GR1E is functioning as an output compare register

Bit 3: IMF1D	Description
0	[Clearing condition] (Initial value) When IMF1D is read while set to 1, then 0 is written to IMF1D
1	[Setting conditions] • When the TCNT1A value is transferred to GR1D by an input capture signal while GR1D is functioning as an input capture register • When TCNT1A = GR1D while GR1D is functioning as an output compare register

- Bit 2—Input Capture/Compare-Match Flag 1C (IMF1C): Status flag that indicates GR1C input capture or compare-match.

Bit 2: IMF1C	Description
0	[Clearing condition] (Initial value) When IMF1C is read while set to 1, then 0 is written to IMF1C
1	[Setting conditions] • When the TCNT1A value is transferred to GR1C by an input capture signal while GR1C is functioning as an input capture register • When TCNT1A = GR1C while GR1C is functioning as an output compare register

- Bit 1—Input Capture/Compare-Match Flag 1B (IMF1B): Status flag that indicates GR1B input capture or compare-match.

Bit 1: IMF1B	Description
0	[Clearing condition] (Initial value) When IMF1B is read while set to 1, then 0 is written to IMF1B
1	[Setting conditions] • When the TCNT1A value is transferred to GR1B by an input capture signal while GR1B is functioning as an input capture register • When TCNT1A = GR1B while GR1B is functioning as an output compare register

Bit 0: IMF1A	Description
0	[Clearing condition] (Initial value) When IMF1A is read while set to 1, then 0 is written to IMF1A
1	[Setting conditions] • When the TCNT1A value is transferred to GR1A by an input capture signal while GR1A is functioning as an input capture register • When TCNT1A = GR1A while GR1A is functioning as an output compare register

TSR1B: TSR1B indicates the status of channel 1 compare-match and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVF1B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CMF1
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Note: * Only 0 can be written, to clear the flag.

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Flag 1B (OVF1B): Status flag that indicates TCNT1B overflow.

Bit 8: OVF1B	Description
0	[Clearing condition] (Initial value) When OVF1B is read while set to 1, then 0 is written to OVF1B
1	[Setting condition] When the TCNT1B value overflows (from H'FFFF to H'0000)

Bit 0: CMF1	Description
0	[Clearing condition] (Initial value) When CMF1 is read while set to 1, then 0 is written to CMF1
1	[Setting condition] When TCNT1B = OCR1

Timer Status Registers 2A and 2B (TSR2A, TSR2B)

TSR2A: TSR2A indicates the status of channel 2 input capture, compare-match, and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVF2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	IMF2H	IMF2G	IMF2F	IMF2E	IMF2D	IMF2C	IMF2B	IMF2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Flag 2A (OVF2A): Status flag that indicates TCNT2A overflow.

Bit 8: OVF2A	Description
0	[Clearing condition] (Initial value) When OVF2A is read while set to 1, then 0 is written to OVF2A
1	[Setting condition] When the TCNT2A value overflows (from H'FFFF to H'0000)

Bit 7: IMF2H	Description
0	[Clearing condition] (Initial value) When IMF2H is read while set to 1, then 0 is written to IMF2H
1	[Setting conditions] • When the TCNT2A value is transferred to GR2H by an input capture signal while GR2H is functioning as an input capture register • When TCNT2A = GR2H while GR2H is functioning as an output compare register

- Bit 6—Input Capture/Compare-Match Flag 2G (IMF2G): Status flag that indicates GR2G input capture or compare-match.

Bit 6: IMF2G	Description
0	[Clearing condition] (Initial value) When IMF2G is read while set to 1, then 0 is written to IMF2G
1	[Setting conditions] • When the TCNT2A value is transferred to GR2G by an input capture signal while GR2G is functioning as an input capture register • When TCNT2A = GR2G while GR2G is functioning as an output compare register

- Bit 5—Input Capture/Compare-Match Flag 2F (IMF2F): Status flag that indicates GR2F input capture or compare-match.

Bit 5: IMF2F	Description
0	[Clearing condition] (Initial value) When IMF2F is read while set to 1, then 0 is written to IMF2F
1	[Setting conditions] • When the TCNT2A value is transferred to GR2F by an input capture signal while GR2F is functioning as an input capture register • When TCNT2A = GR2F while GR2F is functioning as an output compare register

Bit 4: IMF2E	Description
0	[Clearing condition] (Initial value) When IMF2E is read while set to 1, then 0 is written to IMF2E
1	[Setting conditions] • When the TCNT2A value is transferred to GR2E by an input capture signal while GR2E is functioning as an input capture register • When TCNT2A = GR2E while GR2E is functioning as an output compare register

- Bit 3—Input Capture/Compare-Match Flag 2D (IMF2D): Status flag that indicates GR2D input capture or compare-match.

Bit 3: IMF2D	Description
0	[Clearing condition] (Initial value) When IMF2D is read while set to 1, then 0 is written to IMF2D
1	[Setting conditions] • When the TCNT2A value is transferred to GR2D by an input capture signal while GR2D is functioning as an input capture register • When TCNT2A = GR2D while GR2D is functioning as an output compare register

- Bit 2—Input Capture/Compare-Match Flag 2C (IMF2C): Status flag that indicates GR2C input capture or compare-match.

Bit 2: IMF2C	Description
0	[Clearing condition] (Initial value) When IMF2C is read while set to 1, then 0 is written to IMF2C
1	[Setting conditions] • When the TCNT2A value is transferred to GR2C by an input capture signal while GR2C is functioning as an input capture register • When TCNT2A = GR2C while GR2C is functioning as an output compare register

Bit 1: IMF2B	Description
0	[Clearing condition] (Initial value) When IMF2B is read while set to 1, then 0 is written to IMF2B
1	[Setting conditions] • When the TCNT2A value is transferred to GR2B by an input capture signal while GR2B is functioning as an input capture register • When TCNT2A = GR2B while GR2B is functioning as an output compare register

- Bit 0—Input Capture/Compare-Match Flag 2A (IMF2A): Status flag that indicates GR2A input capture or compare-match.

Bit 0: IMF2A	Description
0	[Clearing condition] (Initial value) When IMF2A is read while set to 1, then 0 is written to IMF2A
1	[Setting conditions] • When the TCNT2A value is transferred to GR2A by an input capture signal while GR2A is functioning as an input capture register • When TCNT2A = GR2A while GR2A is functioning as an output compare register

TSR2B: TSR2B indicates the status of channel 2 compare-match and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVF2B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	CMF2H	CMF2G	CMF2F	CMF2E	CMF2D	CMF2C	CMF2B	CMF2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

Bit 8: OVF2B	Description
0	[Clearing condition] (Initial value) When OVF2B is read while set to 1, then 0 is written to OVF2B
1	[Setting condition] When the TCNT2B value overflows (from H'FFFF to H'0000)

- Bit 7—Compare-Match Flag 2H (CMF2H): Status flag that indicates OCR2H compare-match.

Bit 7: CMF2H	Description
0	[Clearing condition] (Initial value) When CMF2H is read while set to 1, then 0 is written to CMF2H
1	[Setting condition] When TCNT2B = OCR2H

- Bit 6—Compare-Match Flag 2G (CMF2G): Status flag that indicates OCR2G compare-match.

Bit 6: CMF2G	Description
0	[Clearing condition] (Initial value) When CMF2G is read while set to 1, then 0 is written to CMF2G
1	[Setting condition] When TCNT2B = OCR2G

- Bit 5—Compare-Match Flag 2F (CMF2F): Status flag that indicates OCR2F compare-match.

Bit 5: CMF2F	Description
0	[Clearing condition] (Initial value) When CMF2F is read while set to 1, then 0 is written to CMF2F
1	[Setting condition] When TCNT2B = OCR2F

- Bit 4—Compare-Match Flag 2E (CMF2E): Status flag that indicates OCR2E compare-match.

Bit 4: CMF2E	Description
0	[Clearing condition] (Initial value) When CMF2E is read while set to 1, then 0 is written to CMF2E
1	[Setting condition] When TCNT2B = OCR2E

0	[Clearing condition] (Initial value) When CMF2D is read while set to 1, then 0 is written to CMF2D
1	[Setting condition] When TCNT2B = OCR2D

- Bit 2—Compare-Match Flag 2C (CMF2C): Status flag that indicates OCR2C compare-match.

Bit 2: CMF2C	Description
0	[Clearing condition] (Initial value) When CMF2C is read while set to 1, then 0 is written to CMF2C
1	[Setting condition] When TCNT2B = OCR2C

- Bit 1—Compare-Match Flag 2B (CMF2B): Status flag that indicates OCR2B compare-match.

Bit 1: CMF2B	Description
0	[Clearing condition] (Initial value) When CMF2B is read while set to 1, then 0 is written to CMF2B
1	[Setting condition] When TCNT2B = OCR2B

- Bit 0—Compare-Match Flag 2A (CMF2A): Status flag that indicates OCR2A compare-match.

Bit 0: CMF2A	Description
0	[Clearing condition] (Initial value) When CMF2A is read while set to 1, then 0 is written to CMF2A
1	[Setting condition] When TCNT2B = OCR2A

Bit:	15	14	13	12	11	10	9	8
	—	OVF5	IMF5D	IMF5C	IMF5B	IMF5A	OVF4	IMF4D
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	IMF4C	IMF4B	IMF4A	OVF3	IMF3D	IMF3C	IMF3B	IMF3A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—Overflow Flag 5 (OVF5): Status flag that indicates TCNT5 overflow.

Bit 14: OVF5	**Description**
0	[Clearing condition] (Initial value) When OVF5 is read while set to 1, then 0 is written to OVF5
1	[Setting condition] When the TCNT5 value overflows (from H'FFFF to H'0000)

- Bit 13—Input Capture/Compare-Match Flag 5D (IMF5D): Status flag that indicates GR5D input capture or compare-match.

Bit 13: IMF5D	**Description**
0	[Clearing condition] (Initial value) When IMF5D is read while set to 1, then 0 is written to IMF5D
1	[Setting conditions] • When the TCNT5 value is transferred to GR5D by an input capture signal while GR5D is functioning as an input capture register • When TCNT5 = GR5D while GR5D is functioning as an output compare register • When TCNT5 = GR5D while GR5D is functioning as a cycle register in PWM mode

Bit 12: IMF5C	Description
0	[Clearing condition] (Initial value) When IMF5C is read while set to 1, then 0 is written to IMF5C
1	[Setting conditions] • When the TCNT5 value is transferred to GR5C by an input capture signal while GR5C is functioning as an input capture register • When TCNT5 = GR5C while GR5C is functioning as an output compare register

- Bit 11—Input Capture/Compare-Match Flag 5B (IMF5B): Status flag that indicates GR5B input capture or compare-match. The flag is not set in PWM mode.

Bit 11: IMF5B	Description
0	[Clearing condition] (Initial value) When IMF5B is read while set to 1, then 0 is written to IMF5B
1	[Setting conditions] • When the TCNT5 value is transferred to GR5B by an input capture signal while GR5B is functioning as an input capture register • When TCNT5 = GR5B while GR5B is functioning as an output compare register

- Bit 10—Input Capture/Compare-Match Flag 5A (IMF5A): Status flag that indicates GR5A input capture or compare-match. The flag is not set in PWM mode.

Bit 10: IMF5A	Description
0	[Clearing condition] (Initial value) When IMF5A is read while set to 1, then 0 is written to IMF5A
1	[Setting conditions] • When the TCNT5 value is transferred to GR5A by an input capture signal while GR5A is functioning as an input capture register • When TCNT5 = GR5A while GR5A is functioning as an output compare register

0	[Clearing condition] (Initial value) When OVF4 is read while set to 1, then 0 is written to OVF4
1	[Setting condition] When the TCNT4 value overflows (from H'FFFF to H'0000)

- Bit 8—Input Capture/Compare-Match Flag 4D (IMF4D): Status flag that indicates GR4D input capture or compare-match.

Bit 8: IMF4D	Description
0	[Clearing condition] (Initial value) When IMF4D is read while set to 1, then 0 is written to IMF4D
1	[Setting conditions] • When the TCNT4 value is transferred to GR4D by an input capture signal while GR4D is functioning as an input capture register • When TCNT4 = GR4D while GR4D is functioning as an output compare register • When TCNT4 = GR4D while GR4D is functioning as a PWM mode synchronous register

- Bit 7—Input Capture/Compare-Match Flag 4C (IMF4C): Status flag that indicates GR4C input capture or compare-match. The flag is not set in PWM mode.

Bit 7: IMF4C	Description
0	[Clearing condition] (Initial value) When IMF4C is read while set to 1, then 0 is written to IMF4C
1	[Setting conditions] • When the TCNT4 value is transferred to GR4C by an input capture signal while GR4C is functioning as an input capture register • When TCNT4 = GR4C while GR4C is functioning as an output compare register

Bit 6: IMF4B	Description
0	[Clearing condition] (Initial value) When IMF4B is read while set to 1, then 0 is written to IMF4B
1	[Setting conditions] • When the TCNT4 value is transferred to GR4B by an input capture signal while GR4B is functioning as an input capture register • When TCNT4 = GR4B while GR4B is functioning as an output compare register

- Bit 5—Input Capture/Compare-Match Flag 4A (IMF4A): Status flag that indicates GR4A input capture or compare-match. The flag is not set in PWM mode.

Bit 5: IMF4A	Description
0	[Clearing condition] (Initial value) When IMF4A is read while set to 1, then 0 is written to IMF4A
1	[Setting conditions] • When the TCNT4 value is transferred to GR4A by an input capture signal while GR4A is functioning as an input capture register • When TCNT4 = GR4A while GR4A is functioning as an output compare register

- Bit 4—Overflow Flag 3 (OVF3): Status flag that indicates TCNT3 input capture or compare-match.

Bit 4: OVF3	Description
0	[Clearing condition] (Initial value) When OVF3 is read while set to 1, then 0 is written to OVF3
1	[Setting condition] When the TCNT3 value overflows (from H'FFFF to H'0000)

Bit 3: IMF3D	Description
0	[Clearing condition] (Initial value) When IMF3D is read while set to 1, then 0 is written to IMF3D
1	[Setting conditions] • When the TCNT3 value is transferred to GR3D by an input capture signal while GR3D is functioning as an input capture register. However, IMF3D is not set by input capture with a channel 9 compare match as the trigger • When TCNT3 = GR3D while GR3D is functioning as an output compare register • When TCNT3 = GR3D while GR3D is functioning as a synchronous register in PWM mode

- Bit 2—Input Capture/Compare-Match Flag 3C (IMF3C): Status flag that indicates GR3C input capture or compare-match. The flag is not set in PWM mode.

Bit 2: IMF3C	Description
0	[Clearing condition] (Initial value) When IMF3C is read while set to 1, then 0 is written to IMF3C
1	[Setting conditions] • When the TCNT3 value is transferred to GR3C by an input capture signal while GR3C is functioning as an input capture register. However, IMF3C is not set by input capture with a channel 9 compare match as the trigger • When TCNT3 = GR3C while GR3C is functioning as an output compare register

- Bit 1—Input Capture/Compare-Match Flag 3B (IMF3B): Status flag that indicates GR3B input capture or compare-match. The flag is not set in PWM mode.

Bit 1: IMF3B	Description
0	[Clearing condition] (Initial value) When IMF3B is read while set to 1, then 0 is written to IMF3B
1	[Setting conditions] • When the TCNT3 value is transferred to GR3B by an input capture signal while GR3B is functioning as an input capture register. However, IMF3B is not set by input capture with a channel 9 compare match as the trigger • When TCNT3 = GR3B while GR3B is functioning as an output compare register

Bit 0: IMF3A	Description
0	[Clearing condition] (Initial value) When IMF3A is read while set to 1, then 0 is written to IMF3A
1	[Setting conditions] • When the TCNT3 value is transferred to GR3A by an input capture signal while GR3A is functioning as an input capture register. However, IMF3A is not set by input capture with a channel 9 compare match as the trigger • When TCNT3 = GR3A while GR3A is functioning as an output compare register

Timer Status Registers 6 and 7 (TSR6, TSR7)

TSR6 and TRS7 indicate the channel 6 and 7 free-running counter up-count and down-count status, and cycle register compare status.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	UDxD	UDxC	UDxB	UDxA	CMFxD	CMFxC	CMFxB	CMFxA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.
x = 6 or 7

UDxA to UDxD relate to TSR6 only. Bits relating to TSR7 always read 0.

- Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 7—Count-Up/Count-Down Flag 6D (UD6D): Status flag that indicates the TCNT6D count operation.

Bit 7: UD6D	Description
0	Free-running counter TCNT6D operates as an up-counter
1	Free-running counter TCNT6D operates as a down-counter

Bit 6: UD6C	Description
0	Free-running counter TCNT6C operates as an up-counter
1	Free-running counter TCNT6C operates as a down-counter

- Bit 5—Count-Up/Count-Down Flag 6B (UD6B): Status flag that indicates the TCNT6B count operation.

Bit 5: UD6B	Description
0	Free-running counter TCNT6B operates as an up-counter
1	Free-running counter TCNT6B operates as a down-counter

- Bit 4—Count-Up/Count-Down Flag 6A (UD6A): Status flag that indicates the TCNT6A count operation.

Bit 4: UD6A	Description
0	Free-running counter TCNT6A operates as an up-counter
1	Free-running counter TCNT6A operates as a down-counter

- Bit 3—Cycle Register Compare-Match Flag 6D/7D (CMF6D/CMF7D): Status flag that indicates CYLRxD compare-match.

Bit 3: CMFxD	Description
0	[Clearing condition] (Initial value) When CMFxD is read while set to 1, then 0 is written to CMFxD
1	[Setting conditions] • When TCNTxD = CYLRxD (in non-complementary PWM mode) • When TCNT6D = H'0000 in a down-count (in complementary PWM mode)

x = 6 or 7

Bit 2: CMFxC	Description
0	[Clearing condition] (Initial value) When CMFxC is read while set to 1, then 0 is written to CMFxC
1	[Setting conditions] • When TCNTxC = CYLRxC (in non-complementary PWM mode) • When TCNT6C = H'0000 in a down-count (in complementary PWM mode)

x = 6 or 7

- Bit 1—Cycle Register Compare-Match Flag 6B/7B (CMF6B/CMF7B): Status flag that indicates CYLRxB compare-match.

Bit 1: CMFxB	Description
0	[Clearing condition] (Initial value) When CMFxB is read while set to 1, then 0 is written to CMFxB
1	[Setting conditions] • When TCNTxB = CYLRxB (in non-complementary PWM mode) • When TCNT6B = H'0000 in a down-count (in complementary PWM mode)

x = 6 or 7

- Bit 0—Cycle Register Compare-Match Flag 6A/7A (CMF6A/CMF7A): Status flag that indicates CYLRxA compare-match.

Bit 0: CMFxA	Description
0	[Clearing condition] (Initial value) When CMFxA is read while set to 1, then 0 is written to CMFxA
1	[Setting conditions] • When TCNTxA = CYLRxA (in non-complementary PWM mode) • When TCNT6A = H'0000 in a down-count (in complementary PWM mode)

x = 6 or 7

Bit:	15	14	13	12	11	10	9	8
	OSF8P	OSF8O	OSF8N	OSF8M	OSF8L	OSF8K	OSF8J	OSF8I
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	OSF8H	OSF8G	OSF8F	OSF8E	OSF8D	OSF8C	OSF8B	OSF8A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

- Bit 15—One-Shot Pulse Flag 8P (OSF8P): Status flag that indicates a DCNT8P one-shot pulse.

Bit 15: OSF8P	**Description**
0	[Clearing condition] (Initial value) When OSF8P is read while set to 1, then 0 is written to OSF8P
1	[Setting condition] When DCNT8P underflows

- Bit 14—One-Shot Pulse Flag 8O (OSF8O): Status flag that indicates a DCNT8O one-shot pulse.

Bit 14: OSF8O	**Description**
0	[Clearing condition] (Initial value) When OSF8O is read while set to 1, then 0 is written to OSF8O
1	[Setting condition] When DCNT8O underflows

Bit 13: OSF8N	Description
0	[Clearing condition] (Initial value) When OSF8N is read while set to 1, then 0 is written to OSF8N
1	[Setting condition] When DCNT8N underflows

- Bit 12—One-Shot Pulse Flag 8M (OSF8M): Status flag that indicates a DCNT8M one-shot pulse.

Bit 12: OSF8M	Description
0	[Clearing condition] (Initial value) When OSF8M is read while set to 1, then 0 is written to OSF8M
1	[Setting condition] When DCNT8M underflows

- Bit 11—One-Shot Pulse Flag 8L (OSF8L): Status flag that indicates a DCNT8L one-shot pulse.

Bit 11: OSF8L	Description
0	[Clearing condition] (Initial value) When OSF8L is read while set to 1, then 0 is written to OSF8L
1	[Setting condition] When DCNT8L underflows

- Bit 10—One-Shot Pulse Flag 8K (OSF8K): Status flag that indicates a DCNT8K one-shot pulse.

Bit 10: OSF8K	Description
0	[Clearing condition] (Initial value) When OSF8K is read while set to 1, then 0 is written to OSF8K
1	[Setting condition] When DCNT8K underflows

0	[Clearing condition] (Initial value) When OSF8J is read while set to 1, then 0 is written to OSF8J
1	[Setting condition] When DCNT8J underflows

- Bit 8—One-Shot Pulse Flag 8I (OSF8I): Status flag that indicates a DCNT8I one-shot pulse.

Bit 8: OSF8I	Description
0	[Clearing condition] (Initial value) When OSF8I is read while set to 1, then 0 is written to OSF8I
1	[Setting condition] When DCNT8I underflows

- Bit 7—One-Shot Pulse Flag 8H (OSF8H): Status flag that indicates a DCNT8H one-shot pulse.

Bit 7: OSF8H	Description
0	[Clearing condition] (Initial value) When OSF8H is read while set to 1, then 0 is written to OSF8H
1	[Setting condition] When DCNT8H underflows

- Bit 6—One-Shot Pulse Flag 8G (OSF8G): Status flag that indicates a DCNT8G one-shot pulse.

Bit 6: OSF8G	Description
0	[Clearing condition] (Initial value) When OSF8G is read while set to 1, then 0 is written to OSF8G
1	[Setting condition] When DCNT8G underflows

0	[Clearing condition] (Initial value) When OSF8F is read while set to 1, then 0 is written to OSF8F
1	[Setting condition] When DCNT8F underflows

- Bit 4—One-Shot Pulse Flag 8E (OSF8E): Status flag that indicates a DCNT8E one-shot pulse.

Bit 4: OSF8E	Description
0	[Clearing condition] (Initial value) When OSF8E is read while set to 1, then 0 is written to OSF8E
1	[Setting condition] When DCNT8E underflows

- Bit 3—One-Shot Pulse Flag 8D (OSF8D): Status flag that indicates a DCNT8D one-shot pulse.

Bit 3: OSF8D	Description
0	[Clearing condition] (Initial value) When OSF8D is read while set to 1, then 0 is written to OSF8D
1	[Setting condition] When DCNT8D underflows

- Bit 2—One-Shot Pulse Flag 8C (OSF8C): Status flag that indicates a DCNT8C one-shot pulse.

Bit 2: OSF8C	Description
0	[Clearing condition] (Initial value) When OSF8C is read while set to 1, then 0 is written to OSF8C
1	[Setting condition] When DCNT8C underflows

- Bit 1—One-Shot Pulse Flag 8B (OSF8B): Status flag that indicates a DCNT8B one-shot pulse.

Bit 1: OSF8B	Description
0	[Clearing condition] (Initial value) When OSF8B is read while set to 1, then 0 is written to OSF8B
1	[Setting condition] When DCNT8B underflows

Bit 0: OSF8A	Description
0	[Clearing condition] (Initial value) When OSF8A is read while set to 1, then 0 is written to OSF8A
1	[Setting condition] When DCNT8A underflows

Timer Status Register 9 (TSR9)

TSR9 indicates the channel 9 event counter compare-match status.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	CMF9F	CMF9E	CMF9D	CMF9C	CMF9B	CMF9A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

- Bits 15 to 6—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 5—Compare-Match Flag 9F (CMF9F): Status flag that indicates GR9F compare-match.

Bit 5: CMF9F	Description
0	[Clearing condition] (Initial value) When CMF9F is read while set to 1, then 0 is written to CMF9F
1	[Setting condition] When the next edge is input while ECNT9F = GR9F

0	[Clearing condition] (Initial value) When CMF9E is read while set to 1, then 0 is written to CMF9E
1	[Setting condition] When the next edge is input while ECNT9E = GR9E

- Bit 3—Compare-Match Flag 9D (CMF9D): Status flag that indicates GR9D compare-match.

Bit 3: CMF9D	Description
0	[Clearing condition] (Initial value) When CMF9D is read while set to 1, then 0 is written to CMF9D
1	[Setting condition] When the next edge is input while ECNT9D = GR9D

- Bit 2—Compare-Match Flag 9C (CMF9C): Status flag that indicates GR9C compare-match.

Bit 2: CMF9C	Description
0	[Clearing condition] (Initial value) When CMF9C is read while set to 1, then 0 is written to CMF9C
1	[Setting condition] When the next edge is input while ECNT9C = GR9C

- Bit 1—Compare-Match Flag 9B (CMF9B): Status flag that indicates GR9B compare-match.

Bit 1: CMF9B	Description
0	[Clearing condition] (Initial value) When CMF9B is read while set to 1, then 0 is written to CMF9B
1	[Setting condition] When the next edge is input while ECNT9B = GR9B

- Bit 0—Compare-Match Flag 9A (CMF9A): Status flag that indicates GR9A compare-match.

Bit 0: CMF9A	Description
0	[Clearing condition] (Initial value) When CMF9A is read while set to 1, then 0 is written to CMF9A
1	[Setting condition] When the next edge is input while ECNT9A = GR9A

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVF11
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IMF11B	IMF11A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Flag 11 (OVF11): Status flag that indicates TCNT11 overflow.

Bit 8: OVF11	**Description**
0	[Clearing condition] (Initial value) When OVF11 is read while set to 1, then 0 is written to OVF11
1	[Setting condition] When the TCNT11 value overflows (from H'FFFF to H'0000)

- Bits 7 to 2—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 1—Input Capture/Compare-Match Flag 11B (IMF11B): Status flag that indicates GR11B input capture or compare-match.

Bit 1: IMF11B	**Description**
0	[Clearing condition] (Initial value) When IMF11B is read while set to 1, then 0 is written to IMF11B
1	[Setting conditions] • When the TCNT11 value is transferred to GR11B by an input capture signal while GR11B is functioning as an input capture register • When TCNT11 = GR11B while GR11B is functioning as an output compare register

Bit 0: IMF11A	Description
0	[Clearing condition] (Initial value) When IMF11A is read while set to 1, then 0 is written to IMF11A
1	[Setting conditions] • When the TCNT11 value is transferred to GR11A by an input capture signal while GR11A is functioning as an input capture register • When TCNT11 = GR11A while GR11A is functioning as an output compare register

11.2.6 Timer Interrupt Enable Registers (TIER)

The timer interrupt enable registers (TIER) are 16-bit registers. The ATU-II has 11 TIER registers: one each for channels 0, 6 to 9, and 11, two each for channels 1 and 2, and one for channels 3 to 5. For details of channel 10, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
0	TIER0	Controls input capture, and overflow interrupt request enabling/disabling.
1	TIER1A, TIER1B	Control input capture, compare-match, and overflow interrupt request enabling/disabling.
2	TIER2A, TIER2B	
3	TIER3	Controls input capture, compare-match, and overflow interrupt request enabling/disabling.
4		
5		
6	TIER6	Control cycle register compare-match interrupt request enabling/disabling.
7	TIER7	
8	TIER8	Controls down-counter output end (low) interrupt request enabling/disabling.
9	TIER9	Controls event counter compare-match interrupt request enabling/disabling.
11	TIER11	Controls input capture, compare-match, and overflow interrupt request enabling/disabling.

The TIER registers are 16-bit readable/writable registers that control enabling/disabling of free-running counter (TCNT) overflow interrupt requests, channel 0 input capture interrupt requests, channel 1 to 5 and 11 general register input capture/compare-match interrupt requests, channel 6 and 7 compare-match interrupt requests, channel 8 down-counter output end interrupt requests, and channel 9 event counter compare-match interrupt requests.

Timer Interrupt Enable Register 0 (TIER0)

TIER0 controls enabling/disabling of channel 0 input capture and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	OVE0	ICE0D	ICE0C	ICE0B	ICE0A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 5—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 4—Overflow Interrupt Enable 0 (OVE0): Enables or disables interrupt requests by the overflow flag (OVF0) in TSR0 when OVF0 is set to 1.

Bit 4: OVE0	Description	
0	OVI0 interrupt requested by OVF0 is disabled	(Initial value)
1	OVI0 interrupt requested by OVF0 is enabled	

- Bit 3—Input Capture Interrupt Enable 0D (ICE0D): Enables or disables interrupt requests by the input capture flag (ICF0D) in TSR0 when ICF0D is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 3: ICE0D	Description	
0	ICI0D interrupt requested by ICF0D is disabled	(Initial value)
1	ICI0D interrupt requested by ICF0D is enabled	

interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 2: ICE0C	Description	
0	ICI0C interrupt requested by ICF0C is disabled	(Initial value)
1	ICI0C interrupt requested by ICF0C is enabled	

- Bit 1—Input Capture Interrupt Enable 0B (ICE0B): Enables or disables interrupt requests by the input capture flag (ICF0B) in TSR0 when ICF0B is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 1: ICE0B	Description	
0	ICI0B interrupt requested by ICF0B is disabled	(Initial value)
1	ICI0B interrupt requested by ICF0B is enabled	

- Bit 0—Input Capture Interrupt Enable 0A (ICE0A): Enables or disables interrupt requests by the input capture flag (ICF0A) in TSR0 when ICF0A is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 0: ICE0A	Description	
0	ICI0A interrupt requested by ICF0A is disabled	(Initial value)
1	ICI0A interrupt requested by ICF0A is enabled	

Timer Interrupt Enable Registers 1A and 1B (TIER1A, TIER1B)

TIER1A: TIER1A controls enabling/disabling of channel 1 input capture, compare-match, and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	—	—	—	OVE1A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	IME1H	IME1G	IME1F	IME1E	IME1D	IME1C	IME1B	IME1A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

OVF1A in TSR1A when OVF1A is set to 1.

Bit 8: OVE1A	Description	
0	OVI1A interrupt requested by OVF1A is disabled	(Initial value)
1	OVI1A interrupt requested by OVF1A is enabled	

- Bit 7—Input Capture/Compare-Match Interrupt Enable 1H (IME1H): Enables or disables interrupt requests by IMF1H in TSR1A when IMF1H is set to 1.

Bit 7: IME1H	Description	
0	IMI1H interrupt requested by IMF1H is disabled	(Initial value)
1	IMI1H interrupt requested by IMF1H is enabled	

- Bit 6—Input Capture/Compare-Match Interrupt Enable 1G (IME1G): Enables or disables interrupt requests by IMF1G in TSR1A when IMF1G is set to 1.

Bit 6: IME1G	Description	
0	IMI1G interrupt requested by IMF1G is disabled	(Initial value)
1	IMI1G interrupt requested by IMF1G is enabled	

- Bit 5—Input Capture/Compare-Match Interrupt Enable 1F (IME1F): Enables or disables interrupt requests by IMF1F in TSR1A when IMF1F is set to 1.

Bit 5: IME1F	Description	
0	IMI1F interrupt requested by IMF1F is disabled	(Initial value)
1	IMI1F interrupt requested by IMF1F is enabled	

- Bit 4—Input Capture/Compare-Match Interrupt Enable 1E (IME1E): Enables or disables interrupt requests by IMF1E in TSR1A when IMF1E is set to 1.

Bit 4: IME1E	Description	
0	IMI1E interrupt requested by IMF1E is disabled	(Initial value)
1	IMI1E interrupt requested by IMF1E is enabled	

Bit 3: IME1D	Description	
0	IMI1D interrupt requested by IMF1D is disabled	(Initial value)
1	IMI1D interrupt requested by IMF1D is enabled	

- Bit 2—Input Capture/Compare-Match Interrupt Enable 1C (IME1C): Enables or disables interrupt requests by IMF1C in TSR1A when IMF1C is set to 1.

Bit 2: IME1C	Description	
0	IMI1C interrupt requested by IMF1C is disabled	(Initial value)
1	IMI1C interrupt requested by IMF1C is enabled	

- Bit 1—Input Capture/Compare-Match Interrupt Enable 1B (IME1B): Enables or disables interrupt requests by IMF1B in TSR1A when IMF1B is set to 1.

Bit 1: IME1B	Description	
0	IMI1B interrupt requested by IMF1B is disabled	(Initial value)
1	IMI1B interrupt requested by IMF1B is enabled	

- Bit 0—Input Capture/Compare-Match Interrupt Enable 1A (IME1A): Enables or disables interrupt requests by IMF1A in TSR1A when IMF1A is set to 1.

Bit 0: IME1A	Description	
0	IMI1A interrupt requested by IMF1A is disabled	(Initial value)
1	IMI1A interrupt requested by IMF1A is enabled	

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVE1B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CME1
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Interrupt Enable 1B (OVE1B): Enables or disables interrupt requests by OVF1B in TSR1B when OVF1B is set to 1.

Bit 8: OVE1B	Description	
0	OVI1B interrupt requested by OVF1B is disabled	(Initial value)
1	OVI1B interrupt requested by OVF1B is enabled	

- Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 0—Compare-Match Interrupt Enable 1 (CME1): Enables or disables interrupt requests by CMF1 in TSR1B when CMF1 is set to 1.

Bit 0: CME1	Description	
0	CMI1 interrupt requested by CMF1 is disabled	(Initial value)
1	CMI1 interrupt requested by CMF1 is enabled	

overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVE2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	IME2H	IME2G	IME2F	IME2E	IME2D	IME2C	IME2B	IME2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Interrupt Enable 2A (OVE2A): Enables or disables interrupt requests by OVF2A in TSR2A when OVF2A is set to 1.

Bit 8: OVE2A	Description	
0	OVI2A interrupt requested by OVF2A is disabled	(Initial value)
1	OVI2A interrupt requested by OVF2A is enabled	

- Bit 7—Input Capture/Compare-Match Interrupt Enable 2H (IME2H): Enables or disables interrupt requests by IMF2H in TSR2A when IMF2H is set to 1.

Bit 7: IME2H	Description	
0	IMI2H interrupt requested by IMF2H is disabled	(Initial value)
1	IMI2H interrupt requested by IMF2H is enabled	

- Bit 6—Input Capture/Compare-Match Interrupt Enable 2G (IME2G): Enables or disables interrupt requests by IMF2G in TSR2A when IMF2G is set to 1.

Bit 6: IME2G	Description	
0	IMI2G interrupt requested by IMF2G is disabled	(Initial value)
1	IMI2G interrupt requested by IMF2G is enabled	

Bit 5: IME2F	Description	
0	IMI2F interrupt requested by IMF2F is disabled	(Initial value)
1	IMI2F interrupt requested by IMF2F is enabled	

- Bit 4—Input Capture/Compare-Match Interrupt Enable 2E (IME2E): Enables or disables interrupt requests by IMF2E in TSR2A when IMF2E is set to 1.

Bit 4: IME2E	Description	
0	IMI2E interrupt requested by IMF2E is disabled	(Initial value)
1	IMI2E interrupt requested by IMF2E is enabled	

- Bit 3—Input Capture/Compare-Match Interrupt Enable 2D (IME2D): Enables or disables interrupt requests by IMF2D in TSR2A when IMF2D is set to 1.

Bit 3: IME2D	Description	
0	IMI2D interrupt requested by IMF2D is disabled	(Initial value)
1	IMI2D interrupt requested by IMF2D is enabled	

- Bit 2—Input Capture/Compare-Match Interrupt Enable 2C (IME2C): Enables or disables interrupt requests by IMF2C in TSR2A when IMF2C is set to 1.

Bit 2: IME2C	Description	
0	IMI2C interrupt requested by IMF2C is disabled	(Initial value)
1	IMI2C interrupt requested by IMF2C is enabled	

- Bit 1—Input Capture/Compare-Match Interrupt Enable 2B (IME2B): Enables or disables interrupt requests by IMF2B in TSR2A when IMF2B is set to 1.

Bit 1: IME2B	Description	
0	IMI2B interrupt requested by IMF2B is disabled	(Initial value)
1	IMI2B interrupt requested by IMF2B is enabled	

Bit 0: IME2A	Description	
0	IMI2A interrupt requested by IMF2A is disabled	(Initial value)
1	IMI2A interrupt requested by IMF2A is enabled	

TIER2B: TIER2B controls enabling/disabling of channel 2 compare-match and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVE2B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	CME2H	CME2G	CME2F	CME2E	CME2D	CME2C	CME2B	CME2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Interrupt Enable 2B (OVE2B): Enables or disables interrupt requests by OVF2B in TSR2B when OVF2B is set to 1.

Bit 8: OVE2B	Description	
0	OVI2B interrupt requested by OVF2B is disabled	(Initial value)
1	OVI2B interrupt requested by OVF2B is enabled	

- Bit 7—Compare-Match Interrupt Enable 2H (CME2H): Enables or disables interrupt requests by CMF2F in TSR2B when CMF2H is set to 1.

Bit 7: CME2H	Description	
0	CMI2H interrupt requested by CMF2H is disabled	(Initial value)
1	CMI2H interrupt requested by CMF2H is enabled	

Bit 6: CME2G	Description	
0	CMI2G interrupt requested by CMF2G is disabled	(Initial value)
1	CMI2G interrupt requested by CMF2G is enabled	

- Bit 5—Compare-Match Interrupt Enable 2F (CME2F): Enables or disables interrupt requests by CMF2F in TSR2B when CMF2F is set to 1.

Bit 5: CME2F	Description	
0	CMI2F interrupt requested by CMF2F is disabled	(Initial value)
1	CMI2F interrupt requested by CMF2F is enabled	

- Bit 4—Compare-Match Interrupt Enable 2E (CME2E): Enables or disables interrupt requests by CMF2E in TSR2B when CMF2E is set to 1.

Bit 4: CME2E	Description	
0	CMI2E interrupt requested by CMF2E is disabled	(Initial value)
1	CMI2E interrupt requested by CMF2E is enabled	

- Bit 3—Compare-Match Interrupt Enable 2D (CME2D): Enables or disables interrupt requests by CMF2D in TSR2B when CMF2D is set to 1.

Bit 3: CME2D	Description	
0	CMI2D interrupt requested by CMF2D is disabled	(Initial value)
1	CMI2D interrupt requested by CMF2D is enabled	

- Bit 2—Compare-Match Interrupt Enable 2C (CME2C): Enables or disables interrupt requests by CMF2C in TSR2B when CMF2C is set to 1.

Bit 2: CME2C	Description	
0	CMI2C interrupt requested by CMF2C is disabled	(Initial value)
1	CMI2C interrupt requested by CMF2C is enabled	

Bit 1: CME2B	Description	
0	CMI2B interrupt requested by CMF2B is disabled	(Initial value)
1	CMI2B interrupt requested by CMF2B is enabled	

- Bit 0—Compare-Match Interrupt Enable 2A (CME2A): Enables or disables interrupt requests by CMF2A in TSR2B when CMF2A is set to 1.

Bit 0: CME2A	Description	
0	CMI2A interrupt requested by CMF2A is disabled	(Initial value)
1	CMI2A interrupt requested by CMF2A is enabled	

Timer Interrupt Enable Register 3 (TIER3)

TIER3 controls enabling/disabling of channel 3 to 5 input capture, compare-match, and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	OVE5	IME5D	IME5C	IME5B	IME5A	OVE4	IME4D
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	IME4C	IME4B	IME4A	OVE3	IME3D	IME3C	IME3B	IME3A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—Overflow Interrupt Enable 5 (OVE5): Enables or disables interrupt requests by OVF5 in TSR3 when OVF5 is set to 1.

Bit 14: OVE5	Description	
0	OVI5 interrupt requested by OVF5 is disabled	(Initial value)
1	OVI5 interrupt requested by OVF5 is enabled	

Bit 13: IME5D	Description	
0	IMI5D interrupt requested by IMF5D is disabled	(Initial value)
1	IMI5D interrupt requested by IMF5D is enabled	

- Bit 12—Input Capture/Compare-Match Interrupt Enable 5C (IME5C): Enables or disables interrupt requests by IMF5C in TSR3 when IMF5C is set to 1.

Bit 12: IME5C	Description	
0	IMI5C interrupt requested by IMF5C is disabled	(Initial value)
1	IMI5C interrupt requested by IMF5C is enabled	

- Bit 11—Input Capture/Compare-Match Interrupt Enable 5B (IME5B): Enables or disables interrupt requests by IMF5B in TSR3 when IMF5B is set to 1.

Bit 11: IME5B	Description	
0	IMI5B interrupt requested by IMF5B is disabled	(Initial value)
1	IMI5B interrupt requested by IMF5B is enabled	

- Bit 10—Input Capture/Compare-Match Interrupt Enable 5A (IME5A): Enables or disables interrupt requests by IMF5A in TSR3 when IMF5A is set to 1.

Bit 10: IME5A	Description	
0	IMI5A interrupt requested by IMF5A is disabled	(Initial value)
1	IMI5A interrupt requested by IMF5A is enabled	

- Bit 9—Overflow Interrupt Enable 4 (OVE4): Enables or disables interrupt requests by OVF4 in TSR3 when OVF4 is set to 1.

Bit 9: OVE4	Description	
0	OVI4 interrupt requested by OVF4 is disabled	(Initial value)
1	OVI4 interrupt requested by OVF4 is enabled	

Bit 8: IME4D	Description	
0	IMI4D interrupt requested by IMF4D is disabled	(Initial value)
1	IMI4D interrupt requested by IMF4D is enabled	

- Bit 7—Input Capture/Compare-Match Interrupt Enable 4C (IME4C): Enables or disables interrupt requests by IMF4C in TSR3 when IMF4C is set to 1.

Bit 7: IME4C	Description	
0	IMI4C interrupt requested by IMF4C is disabled	(Initial value)
1	IMI4C interrupt requested by IMF4C is enabled	

- Bit 6—Input Capture/Compare-Match Interrupt Enable 4B (IME4B): Enables or disables interrupt requests by IMF4B in TSR3 when IMF4B is set to 1.

Bit 6: IME4B	Description	
0	IMI4B interrupt requested by IMF4B is disabled	(Initial value)
1	IMI4B interrupt requested by IMF4B is enabled	

- Bit 5—Input Capture/Compare-Match Interrupt Enable 4A (IME4A): Enables or disables interrupt requests by IMF4A in TSR3 when IMF4A is set to 1.

Bit 5: IME4A	Description	
0	IMI4A interrupt requested by IMF4A is disabled	(Initial value)
1	IMI4A interrupt requested by IMF4A is enabled	

- Bit 4—Overflow Interrupt Enable 3 (OVE3): Enables or disables interrupt requests by OVF3 in TSR3 when OVF3 is set to 1.

Bit 4: OVE3	Description	
0	OVI3 interrupt requested by OVF3 is disabled	(Initial value)
1	OVI3 interrupt requested by OVF3 is enabled	

Bit 3: IME3D	Description	
0	IMI3D interrupt requested by IMF3D is disabled	(Initial value)
1	IMI3D interrupt requested by IMF3D is enabled	

- Bit 2—Input Capture/Compare-Match Interrupt Enable 3C (IME3C): Enables or disables interrupt requests by IMF3C in TSR3 when IMF3C is set to 1.

Bit 2: IME3C	Description	
0	IMI3C interrupt requested by IMF3C is disabled	(Initial value)
1	IMI3C interrupt requested by IMF3C is enabled	

- Bit 1—Input Capture/Compare-Match Interrupt Enable 3B (IME3B): Enables or disables interrupt requests by IMF3B in TSR3 when IMF3B is set to 1.

Bit 1: IME3B	Description	
0	IMI3B interrupt requested by IMF3B is disabled	(Initial value)
1	IMI3B interrupt requested by IMF3B is enabled	

- Bit 0—Input Capture/Compare-Match Interrupt Enable 3A (IME3A): Enables or disables interrupt requests by IMF3A in TSR3 when IMF3A is set to 1.

Bit 0: IME3A	Description	
0	IMI3A interrupt requested by IMF3A is disabled	(Initial value)
1	IMI3A interrupt requested by IMF3A is enabled	

requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	CMExD	CMExC	CMExB	CMExA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

x = 6 or 7

- Bits 15 to 4—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 3—Cycle Register Compare-Match Interrupt Enable 6D/7D (CME6D/CME7D): Enables or disables interrupt requests by CMFxD in TSR6 or TSR7 when CMFxD is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 3: CMExD	Description	
0	CMIxD interrupt requested by CMFxD is disabled	(Initial value)
1	CMIxD interrupt requested by CMFxD is enabled	

x = 6 or 7

- Bit 2—Cycle Register Compare-Match Interrupt Enable 6C/7C (CME6C/CME7C): Enables or disables interrupt requests by CMFxC in TSR6 or TSR7 when CMFxC is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 2: CMExC	Description	
0	CMIxC interrupt requested by CMFxC is disabled	(Initial value)
1	CMIxC interrupt requested by CMFxC is enabled	

x = 6 or 7

DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 1: CMExB	Description	
0	CMIxB interrupt requested by CMFxB is disabled	(Initial value)
1	CMIxB interrupt requested by CMFxB is enabled	

x = 6 or 7

- Bit 0—Cycle Register Compare-Match Interrupt Enable 6A/7A (CME6A/CME7A): Enables or disables interrupt requests by CMFxA in TSR6 or TSR7 when CMFxA is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 0: CMExA	Description	
0	CMIxA interrupt requested by CMFxA is disabled	(Initial value)
1	CMIxA interrupt requested by CMFxA is enabled	

x = 6 or 7

Timer Interrupt Enable Register 8 (TIER8)

TIER8 controls enabling/disabling of channel 8 one-shot pulse interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	OSE8P	OSE8O	OSE8N	OSE8M	OSE8L	OSE8K	OSE8J	OSE8I
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	OSE8H	OSE8G	OSE8F	OSE8E	OSE8D	OSE8C	OSE8B	OSE8A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15: OSE8P	Description	
0	OSI8P interrupt requested by OSF8P is disabled	(Initial value)
1	OSI8P interrupt requested by OSF8P is enabled	

- Bit 14—One-Shot Pulse Interrupt Enable 8O (OSE8O): Enables or disables interrupt requests by OSF8O in TSR8 when OSF8O is set to 1.

Bit 14: OSE8O	Description	
0	OSI8O interrupt requested by OSF8O is disabled	(Initial value)
1	OSI8O interrupt requested by OSF8O is enabled	

- Bit 13—One-Shot Pulse Interrupt Enable 8N (OSE8N): Enables or disables interrupt requests by OSF8N in TSR8 when OSF8N is set to 1.

Bit 13: OSE8N	Description	
0	OSI8N interrupt requested by OSF8N is disabled	(Initial value)
1	OSI8N interrupt requested by OSF8N is enabled	

- Bit 12—One-Shot Pulse Interrupt Enable 8M (OSE8M): Enables or disables interrupt requests by OSF8M in TSR8 when OSF8M is set to 1.

Bit 12: OSE8M	Description	
0	OSI8M interrupt requested by OSF8M is disabled	(Initial value)
1	OSI8M interrupt requested by OSF8M is enabled	

- Bit 11—One-Shot Pulse Interrupt Enable 8L (OSE8L): Enables or disables interrupt requests by OSF8L in TSR8 when OSF8L is set to 1.

Bit 11: OSE8L	Description	
0	OSI8L interrupt requested by OSF8L is disabled	(Initial value)
1	OSI8L interrupt requested by OSF8L is enabled	

Bit 10: OSE8K	Description	
0	OSI8K interrupt requested by OSF8K is disabled	(Initial value)
1	OSI8K interrupt requested by OSF8K is enabled	

- Bit 9—One-Shot Pulse Interrupt Enable 8J (OSE8J): Enables or disables interrupt requests by OSF8J in TSR8 when OSF8J is set to 1.

Bit 9: OSE8J	Description	
0	OSI8J interrupt requested by OSF8J is disabled	(Initial value)
1	OSI8J interrupt requested by OSF8J is enabled	

- Bit 8—One-Shot Pulse Interrupt Enable 8I (OSE8I): Enables or disables interrupt requests by OSF8I in TSR8 when OSF8I is set to 1.

Bit 8: OSE8I	Description	
0	OSI8I interrupt requested by OSF8I is disabled	(Initial value)
1	OSI8I interrupt requested by OSF8I is enabled	

- Bit 7—One-Shot Pulse Interrupt Enable 8H (OSE8H): Enables or disables interrupt requests by OSF8H in TSR8 when OSF8H is set to 1.

Bit 7: OSE8H	Description	
0	OSI8H interrupt requested by OSF8H is disabled	(Initial value)
1	OSI8H interrupt requested by OSF8H is enabled	

- Bit 6—One-Shot Pulse Interrupt Enable 8G (OSE8G): Enables or disables interrupt requests by OSF8G in TSR8 when OSF8G is set to 1.

Bit 6: OSE8G	Description	
0	OSI8G interrupt requested by OSF8G is disabled	(Initial value)
1	OSI8G interrupt requested by OSF8G is enabled	

Bit 5: OSE8F	Description	
0	OSI8F interrupt requested by OSF8F is disabled	(Initial value)
1	OSI8F interrupt requested by OSF8F is enabled	

- Bit 4—One-Shot Pulse Interrupt Enable 8E (OSE8E): Enables or disables interrupt requests by OSF8E in TSR8 when OSF8E is set to 1.

Bit 4: OSE8E	Description	
0	OSI8E interrupt requested by OSF8E is disabled	(Initial value)
1	OSI8E interrupt requested by OSF8E is enabled	

- Bit 3—One-Shot Pulse Interrupt Enable 8D (OSE8D): Enables or disables interrupt requests by OSF8D in TSR8 when OSF8D is set to 1.

Bit 3: OSE8D	Description	
0	OSI8D interrupt requested by OSF8D is disabled	(Initial value)
1	OSI8D interrupt requested by OSF8D is enabled	

- Bit 2—One-Shot Pulse Interrupt Enable 8C (OSE8C): Enables or disables interrupt requests by OSF8C in TSR8 when OSF8C is set to 1.

Bit 2: OSE8C	Description	
0	OSI8C interrupt requested by OSF8C is disabled	(Initial value)
1	OSI8C interrupt requested by OSF8C is enabled	

- Bit 1—One-Shot Pulse Interrupt Enable 8B (OSE8B): Enables or disables interrupt requests by OSF8B in TSR8 when OSF8B is set to 1.

Bit 1: OSE8B	Description	
0	OSI8B interrupt requested by OSF8B is disabled	(Initial value)
1	OSI8B interrupt requested by OSF8B is enabled	

Bit 0: OSE8A	Description	
0	OSI8A interrupt requested by OSF8A is disabled	(Initial value)
1	OSI8A interrupt requested by OSF8A is enabled	

Timer Interrupt Enable Register 9 (TIER9)

TIER9 controls enabling/disabling of channel 9 event counter compare-match interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	CME9F	CME9E	CME9D	CME9C	CME9B	CME9A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 6—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 5—Compare-Match Interrupt Enable 9F (CME9F): Enables or disables interrupt requests by CMF9F in TSR9 when CMF9F is set to 1.

Bit 5: CME9F	Description	
0	CMI9F interrupt requested by CMF9F is disabled	(Initial value)
1	CMI9F interrupt requested by CMF9F is enabled	

- Bit 4—Compare-Match Interrupt Enable 9E (CME9E): Enables or disables interrupt requests by CMF9E in TSR9 when CMF9E is set to 1.

Bit 4: CME9E	Description	
0	CMI9E interrupt requested by CMF9E is disabled	(Initial value)
1	CMI9E interrupt requested by CMF9E is enabled	

Bit 3: CME9D	Description	
0	CMI9D interrupt requested by CMF9D is disabled	(Initial value)
1	CMI9D interrupt requested by CMF9D is enabled	

- Bit 2—Compare-Match Interrupt Enable 9C (CME9C): Enables or disables interrupt requests by CMF9C in TSR9 when CMF9C is set to 1.

Bit 2: CME9C	Description	
0	CMI9C interrupt requested by CMF9C is disabled	(Initial value)
1	CMI9C interrupt requested by CMF9C is enabled	

- Bit 1—Compare-Match Interrupt Enable 9B (CME9B): Enables or disables interrupt requests by CMF9B in TSR9 when CMF9B is set to 1.

Bit 1: CME9B	Description	
0	CMI9B interrupt requested by CMF9B is disabled	(Initial value)
1	CMI9B interrupt requested by CMF9B is enabled	

- Bit 0—Compare-Match Interrupt Enable 9A (CME9A): Enables or disables interrupt requests by CMF9A in TSR9 when CMF9A is set to 1.

Bit 0: CME9A	Description	
0	CMI9A interrupt requested by CMF9A is disabled	(Initial value)
1	CMI9A interrupt requested by CMF9A is enabled	

interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVE11
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IME11B	IME11A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Interrupt Enable 11 (OVE11): Enables or disables interrupt requests by OVF11 in TSR11 when OVF11 is set to 1.

Bit 8: OVE11	Description	
0	OVI11 interrupt requested by OVF11 is disabled	(Initial value)
1	OVI11 interrupt requested by OVF11 is enabled	

- Bits 7 to 2—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 1—Input Capture/Compare-Match Interrupt Enable 11B (IME11B): Enables or disables interrupt requests by IMF11B in TSR11 when IMF11B is set to 1.

Bit 1: IME11B	Description	
0	IMI11B interrupt requested by IMF11B is disabled	(Initial value)
1	IMI11B interrupt requested by IMF11B is enabled	

- Bit 0—Input Capture/Compare-Match Interrupt Enable 11A (IME11A): Enables or disables interrupt requests by IMF11A in TSR11 when IMF11A is set to 1.

Bit 0: IME11A	Description	
0	IMI11A interrupt requested by IMF11A is disabled	(Initial value)
1	IMI11A interrupt requested by IMF11A is enabled	

The interval interrupt request registers (ITVRR) are 8-bit registers. The ATU-II has three ITVRR registers in channel 0.

Channel	Abbreviation	Function
0	ITVRR1	TCNT0 bit 6 to 9 interval interrupt generation and A/D2 converter activation
	ITVRR2A	TCNT0 bit 10 to 13 interval interrupt generation and A/D0 converter activation
	ITVRR2B	TCNT0 bit 10 to 13 interval interrupt generation and A/D1 converter activation

Interval Interrupt Request Register 1 (ITVRR1)

Bit:	7	6	5	4	3	2	1	0
	ITVA9	ITVA8	ITVA7	ITVA6	ITVE9	ITVE8	ITVE7	ITVE6
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ITVRR1 is an 8-bit readable/writable register that detects the rise of bits corresponding to the channel 0 free-running counter (TCNT0) and controls cyclic interrupt output and A/D2 converter activation.

ITVRR1 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—A/D2 Converter Interval Activation Bit 9 (ITVA9): A/D2 converter activation setting bit corresponding to bit 9 in TCNT0. The rise of bit 9 in TCNT0 is ANDed with ITVA9, and the result is output to the A/D2 converter as an activation signal.

Bit 7: ITVA9	Description	
0	A/D2 converter activation by rise of TCNT0 bit 9 is disabled	(Initial value)
1	A/D2 converter activation by rise of TCNT0 bit 9 is enabled	

the result is output to the A/D2 converter as an activation signal.

Bit 6: ITVA8	Description	
0	A/D2 converter activation by rise of TCNT0 bit 8 is disabled	(Initial value)
1	A/D2 converter activation by rise of TCNT0 bit 8 is enabled	

- Bit 5—A/D2 Converter Interval Activation Bit 7 (ITVA7): A/D2 converter activation setting bit corresponding to bit 7 in TCNT0. The rise of bit 7 in TCNT0 is ANDed with ITVA7, and the result is output to the A/D2 converter as an activation signal.

Bit 5: ITVA7	Description	
0	A/D2 converter activation by rise of TCNT0 bit 7 is disabled	(Initial value)
1	A/D2 converter activation by rise of TCNT0 bit 7 is enabled	

- Bit 4—A/D2 Converter Interval Activation Bit 6 (ITVA6): A/D2 converter activation setting bit corresponding to bit 6 in TCNT0. The rise of bit 6 in TCNT0 is ANDed with ITVA6, and the result is output to the A/D2 converter as an activation signal.

Bit 4: ITVA6	Description	
0	A/D2 converter activation by rise of TCNT0 bit 6 is disabled	(Initial value)
1	A/D2 converter activation by rise of TCNT0 bit 6 is enabled	

- Bit 3—Interval Interrupt Bit 9 (ITVE9): INTC interval interrupt setting bit corresponding to bit 9 in TCNT0. The rise of bit 9 in TCNT0 is ANDed with ITVE9, the result is stored in IIF1 in TSR0, and an interrupt request is sent to the CPU.

Bit 3: ITVE9	Description	
0	Interrupt request (ITV1) by rise of TCNT0 bit 9 is disabled	(Initial value)
1	Interrupt request (ITV1) by rise of TCNT0 bit 9 is enabled	

- Bit 2—Interval Interrupt Bit 8 (ITVE8): INTC interval interrupt setting bit corresponding to bit 8 in TCNT0. The rise of bit 8 in TCNT0 is ANDed with ITVE8, the result is stored in IIF1 in TSR0, and an interrupt request is sent to the CPU.

Bit 2: ITVE8	Description	
0	Interrupt request (ITV1) by rise of TCNT0 bit 8 is disabled	(Initial value)
1	Interrupt request (ITV1) by rise of TCNT0 bit 8 is enabled	

TSR0, and an interrupt request is sent to the CPU.

Bit 1: ITVE7	Description	
0	Interrupt request (ITV1) by rise of TCNT0 bit 7 is disabled	(Initial value)
1	Interrupt request (ITV1) by rise of TCNT0 bit 7 is enabled	

- Bit 0—Interval Interrupt Bit 6 (ITVE6): INTC interval interrupt setting bit corresponding to bit 6 in TCNT0. The rise of bit 6 in TCNT0 is ANDed with ITVE6, the result is stored in IIF1 in TSR0, and an interrupt request is sent to the CPU.

Bit 0: ITVE6	Description	
0	Interrupt request (ITV1) by rise of TCNT0 bit 6 is disabled	(Initial value)
1	Interrupt request (ITV1) by rise of TCNT0 bit 6 is enabled	

Interval Interrupt Request Registers 2A and 2B (ITVRR2A, ITVRR2B)

Bit:	7	6	5	4	3	2	1	0
	ITVA13x	ITVA12x	ITVA11x	ITVA10x	ITVE13x	ITVE12x	ITVE11x	ITVE10x
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

x = A or B

- Bit 7—A/D0 / A/D1 Converter Interval Activation Bit 13A/13B (ITVA13A/ITVA13B): A/D0 or A/D1 (ITVRR2A: A/D0; ITVRR2B: A/D1) converter activation setting bit corresponding to bit 13 in TCNT0. The rise of bit 13 in TCNT0 is ANDed with ITVA13x, and the result is output to the A/D0 or A/D1 converter as an activation signal.

Bit 7: ITVA13x	Description
0	A/D0 or A/D1 converter activation by rise of TCNT0 bit 13 is disabled (Initial value)
1	A/D0 or A/D1 converter activation by rise of TCNT0 bit 13 is enabled

x = A or B

bit 12 in TCNT0. The rise of bit 12 in TCNT0 is ANDed with ITVA12x, and the result is output to the A/D0 or A/D1 converter as an activation signal.

Bit 6: ITVA12x	Description
0	A/D0 or A/D1 converter activation by rise of TCNT0 bit 12 is disabled (Initial value)
1	A/D0 or A/D1 converter activation by rise of TCNT0 bit 12 is enabled

x = A or B

- Bit 5—A/D0 / A/D1 Converter Interval Activation Bit 11A/11B (ITVA11A/ITVA11B): A/D0 or A/D1 (ITVRR2A: A/D0; ITVRR2B: A/D1) converter activation setting bit corresponding to bit 11 in TCNT0. The rise of bit 11 in TCNT0 is ANDed with ITVA11x, and the result is output to the A/D0 or A/D1 converter as an activation signal.

Bit 5: ITVA11x	Description
0	A/D0 or A/D1 converter activation by rise of TCNT0 bit 11 is disabled (Initial value)
1	A/D0 or A/D1 converter activation by rise of TCNT0 bit 11 is enabled

x = A or B

- Bit 4—A/D0 / A/D1 Converter Interval Activation Bit 10A/10B (ITVA10A/ITVA10B): A/D0 or A/D1 (ITVRR2A: A/D0; ITVRR2B: A/D1) converter activation setting bit corresponding to bit 10 in TCNT0. The rise of bit 10 in TCNT0 is ANDed with ITVA10x, and the result is output to the A/D0 or A/D1 converter as an activation signal.

Bit 4: ITVA10x	Description
0	A/D0 or A/D1 converter activation by rise of TCNT0 bit 10 is disabled (Initial value)
1	A/D0 or A/D1 converter activation by rise of TCNT0 bit 10 is enabled

x = A or B

the result is stored in IIF2x in TSR0, and an interrupt request is sent to the CPU.

Bit 3: ITVE13x	Description	
0	Interrupt request (ITV2x) by rise of TCNT0 bit 13 is disabled	(Initial value)
1	Interrupt request (ITV2x) by rise of TCNT0 bit 13 is enabled	

x = A or B

- Bit 2—Interval Interrupt Bit 12A/12B (ITVE12A/ITVE12B): INTC interval interrupt setting bit corresponding to bit 12 in TCNT0. The rise of bit 12 in TCNT0 is ANDed with ITVE12x, the result is stored in IIF2x in TSR0, and an interrupt request is sent to the CPU.

Bit 2: ITVE12x	Description	
0	Interrupt request (ITV2x) by rise of TCNT0 bit 12 is disabled	(Initial value)
1	Interrupt request (ITV2x) by rise of TCNT0 bit 12 is enabled	

x = A or B

- Bit 1—Interval Interrupt Bit 11A/11B (ITVE11A/ITVE11B): INTC interval interrupt setting bit corresponding to bit 11 in TCNT0. The rise of bit 11 in TCNT0 is ANDed with ITVE11x, the result is stored in IIF2x in TSR0, and an interrupt request is sent to the CPU.

Bit 1: ITVE11x	Description	
0	Interrupt request (ITV2x) by rise of TCNT0 bit 11 is disabled	(Initial value)
1	Interrupt request (ITV2x) by rise of TCNT0 bit 11 is enabled	

x = A or B

- Bit 0—Interval Interrupt Bit 10 (ITVE10): INTC interval interrupt setting bit corresponding to bit 10 in TCNT0. The rise of bit 10 in TCNT0 is ANDed with ITVE10x, the result is stored in IIF2x in TSR0, and an interrupt request is sent to the CPU.

Bit 0: ITVE10x	Description	
0	Interrupt request (ITV2x) by rise of TCNT0 bit 10 is disabled	(Initial value)
1	Interrupt request (ITV2x) by rise of TCNT0 bit 10 is enabled	

x = A or B

For details, see section 11.3.7, Interval Timer Operation.

The trigger mode register (TRGMDR) is an 8-bit register. The ATU-II has one TRGMDR register.

Bit:	7	6	5	4	3	2	1	0
	TRGMD	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

TRGMDR is an 8-bit readable/writable register that selects whether a channel 1 compare-match is used as a channel 8 one-shot pulse start trigger or as a one-shot pulse terminate trigger when channel 1 and channel 8 are used in combination.

TRGMDR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Trigger Mode Selection Register (TRGMD): Selects the channel 8 one-shot pulse start trigger/one-shot pulse terminate trigger setting.

Bit 7: TRGMD	Description	
0	One-shot pulse start trigger (TCNT1B = OCR1)	(Initial value)
	One-shot pulse terminate trigger (TCNT1A = GR1A–GR1H)	
1	One-shot pulse start trigger (TCNT1A = GR1A–GR1H)	
	One-shot pulse terminate trigger (TCNT1B = OCR1)	

- Bits 6 to 0—Reserved: These bits are always read as 0. The write value should always be 0.

11.2.9 Timer Mode Register (TMDR)

The timer mode register (TMDR) is an 8-bit register. The ATU-II has one TDR register.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	T5PWM	T4PWM	T3PWM
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

TMDR is an 8-bit readable/writable register that specifies whether channels 3 to 5 are used in input capture/output compare mode or PWM mode.

TMDR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

compare mode or PWM mode.

Bit 2: T5PWM	Description	
0	Channel 5 operates in input capture/output compare mode	(Initial value)
1	Channel 5 operates in PWM mode	

When bit T5PWM is set to 1 to select PWM mode, pins TIO5A to TIO5C become PWM output pins, general register 5D (GR5D) functions as a cycle register, and general registers 5A to 5C (GR5A to GR5C) function as duty registers. Settings in the timer I/O control registers (TIOR5A, TIOR5B) are invalid, and general registers 5A to 5D (GR5A to GR5D) can be written to. Do not use the TIO5D pin as a timer output.

- Bit 1—PWM Mode 4 (T4PWM): Selects whether channel 4 operates in input capture/output compare mode or PWM mode.

Bit 1: T4PWM	Description	
0	Channel 4 operates in input capture/output compare mode	(Initial value)
1	Channel 4 operates in PWM mode	

When bit T4PWM is set to 1 to select PWM mode, pins TIO4A to TIO4C become PWM output pins, general register 4D (GR4D) functions as a cycle register, and general registers 4A to 4C (GR4A to GR4C) function as duty registers. Settings in the timer I/O control registers (TIOR4A, TIOR4B) are invalid, and general registers 4A to 4D (GR4A to GR4D) can be written to. Do not use the TIO4D pin as a timer output.

- Bit 0—PWM Mode 3 (T3PWM): Selects whether channel 3 operates in input capture/output compare mode or PWM mode.

Bit 0: T3PWM	Description	
0	Channel 3 operates in input capture/output compare mode	(Initial value)
1	Channel 3 operates in PWM mode	

When bit T3PWM is set to 1 to select PWM mode, pins TIO3A to TIO3C become PWM output pins, general register 3D (GR3D) functions as a cycle register, and general registers 3A to 3C (GR3A to GR3C) function as duty registers. Settings in the timer I/O control registers (TIOR3A, TIOR3B) are invalid, and general registers 3A to 3D (GR3A to GR3D) can be written to. Do not use the TIO3D pin as a timer output.

The PWM mode register (PMDR) is an 8-bit register. The APC-II has one PMDR register.

Bit:	7	6	5	4	3	2	1	0
	DTSELD	DTSELC	DTSELB	DTSELA	CNTSELD	CNTSELC	CNTSELB	CNTSELA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMDR is an 8-bit readable/writable register that selects whether channel 6 PWM output is set to on-duty/off-duty, or to non-complementary PWM mode/complementary PWM mode.

PMDR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Duty Selection Register D (DTSELD): Selects whether channel 6D TO6D output PWM is set to on-duty or to off-duty.

Bit 7: DTSELD	Description	
0	TO6D PWM output is on-duty	(Initial value)
1	TO6D PWM output is off-duty	

- Bit 6—Duty Selection Register C (DTSELC): Selects whether channel 6C TO6C output PWM is set to on-duty or to off-duty.

Bit 6: DTSELC	Description	
0	TO6C PWM output is on-duty	(Initial value)
1	TO6C PWM output is off-duty	

- Bit 5—Duty Selection Register B (DTSELB): Selects whether channel 6B TO6B output PWM is set to on-duty or to off-duty.

Bit 5: DTSELB	Description	
0	TO6B PWM output is on-duty	(Initial value)
1	TO6B PWM output is off-duty	

Bit 4: DTSELA	Description	
0	TO6A PWM output is on-duty	(Initial value)
1	TO6A PWM output is off-duty	

- Bit 3—Counter Selection Register D (CNTSELD): Selects whether channel 6D PWM is set to non-complementary PWM mode or to complementary PWM mode.

Bit 3: CNTSELD	Description	
0	TCNT6D is set to non-complementary PWM mode	(Initial value)
1	TCNT6D is set to complementary PWM mode	

- Bit 2—Counter Selection Register C (CNTSELC): Selects whether channel 6C PWM is set to non-complementary PWM mode or to complementary PWM mode.

Bit 2: CNTSELC	Description	
0	TCNT6C is set to non-complementary PWM mode	(Initial value)
1	TCNT6C is set to complementary PWM mode	

- Bit 1—Counter Selection Register B (CNTSELB): Selects whether channel 6B PWM is set to non-complementary PWM mode or to complementary PWM mode.

Bit 1: CNTSELB	Description	
0	TCNT6B is set to non-complementary PWM mode	(Initial value)
1	TCNT6B is set to complementary PWM mode	

- Bit 0—Counter Selection Register A (CNTSELA): Selects whether channel 6A PWM is set to non-complementary PWM mode or to complementary PWM mode.

Bit 0: CNTSELA	Description	
0	TCNT6A is set to non-complementary PWM mode	(Initial value)
1	TCNT6A is set to complementary PWM mode	

The down-count start register (DSTR) is a 16-bit register. The ATU-II has one DSTR register in channel 8.

Bit:	15	14	13	12	11	10	9	8
	DST8P	DST8O	DST8N	DST8M	DST8L	DST8K	DST8J	DST8I
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit:	7	6	5	4	3	2	1	0
	DST8H	DST8G	DST8F	DST8E	DST8D	DST8C	DST8B	DST8A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only 1 can be written.

DSTR is a 16-bit readable/writable register that starts the channel 8 down-counter (DCNT).

When the one-shot pulse function is used, a value of 1 can be set in a DST8x bit at any time by the user program, except when the corresponding DCNT8x value is H'0000. The DST8x bits are cleared to 0 automatically when the DCNT value overflows.

When the offset one-shot pulse function is used, DST8x is automatically set to 1 (except when the DCNT8x value is H'0000) when a compare-match occurs between the channel 1 or 2 free-running counter (TCNT) and a general register (GR) or the output compare register (OCR1) while the corresponding timer connection register (TCNR) bit is set to 1. As regards DST8I to DST8P, if the RLDEN bit in the reload enable register (RLDENR) is set to 1 and the reload register (RLDR8) value is not H'0000, a reload is performed into the corresponding DCNT8x, and the DST8x bit is set to 1. DST8x is automatically cleared to 0 when the DCNT8x vaue underflows, or by input of a channel 1 or 2 one-shot terminate trigger signal set in the trigger mode register (TRGMDR) while the corresponding one-shot pulse terminate register (OTR) bit is set to 1, whichever occurs first.

DCNT8x is cleared to H'0000 when underflow occurs.

DSTR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 11.3.5, One-Shot Pulse Function, and 11.3.6, Offset One-Shot Pulse Function and Output Cutoff Function.

0	DCNT8P is halted (Initial value) [Clearing conditions] When the DCNT8P value underflows, or on channel 2 (GR2H) compare-match
1	DCNT8P counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8P ≠ H'0000) • Offset one-shot pulse function: Set on OCR2H compare-match (DCNT8P ≠ H'0000 or reload possible) or by user program (DCNT8P ≠ H'0000)

- Bit 14—Down-Count Start 8O (DST8O): Starts down-counter 8O (DCNT8O).

Bit 14: DST8O	Description
0	DCNT8O is halted (Initial value) [Clearing conditions] When the DCNT8O value underflows, or on channel 2 (GR2G) compare-match
1	DCNT8O counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8O ≠ H'0000) • Offset one-shot pulse function: Set on OCR2G compare-match (DCNT8O ≠ H'0000 or reload possible) or by user program (DCNT8O ≠ H'0000)

- Bit 13—Down-Count Start 8N (DST8N): Starts down-counter 8N (DCNT8N).

Bit 13: DST8N	Description
0	DCNT8N is halted (Initial value) [Clearing conditions] When the DCNT8N value underflows, or on channel 2 (GR2F) compare-match
1	DCNT8N counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8N ≠ H'0000) • Offset one-shot pulse function: Set on OCR2F compare-match (DCNT8N ≠ H'0000 or reload possible) or by user program (DCNT8N ≠ H'0000)

0	DCNT8M is halted (Initial value) [Clearing conditions] When the DCNT8M value underflows, or on channel 2 (GR2E) compare-match
1	DCNT8M counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8M ≠ H'0000) • Offset one-shot pulse function: Set on OCR2E compare-match (DCNT8M ≠ H'0000 or reload possible) or by user program (DCNT8M ≠ H'0000)

- Bit 11—Down-Count Start 8L (DST8L): Starts down-counter 8L (DCNT8L).

Bit 11: DST8L	Description
0	DCNT8L is halted (Initial value) [Clearing conditions] When the DCNT8L value underflows, or on channel 2 (GR2D) compare-match
1	DCNT8L counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8L ≠ H'0000) • Offset one-shot pulse function: Set on OCR2D compare-match (DCNT8L ≠ H'0000 or reload possible) or by user program (DCNT8L ≠ H'0000)

- Bit 10—Down-Count Start 8K (DST8K): Starts down-counter 8K (DCNT8K).

Bit 10: DST8K	Description
0	DCNT8K is halted (Initial value) [Clearing conditions] When the DCNT8K value underflows, or on channel 2 (GR2C) compare-match
1	DCNT8K counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8K ≠ H'0000) • Offset one-shot pulse function: Set on OCR2C compare-match (DCNT8K ≠ H'0000 or reload possible) or by user program (DCNT8K ≠ H'0000)

0	DCNT8J is halted (Initial value) [Clearing conditions] When the DCNT8J value underflows, or on channel 2 (GR2B) compare-match
1	DCNT8J counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8J ≠ H'0000) • Offset one-shot pulse function: Set on OCR2B compare-match (DCNT8J ≠ H'0000 or reload possible) or by user program (DCNT8J ≠ H'0000)

- Bit 8—Down-Count Start 8I (DST8I): Starts down-counter 8I (DCNT8I).

Bit 8: DST8I	Description
0	DCNT8I is halted (Initial value) [Clearing conditions] When the DCNT8I value underflows, or on channel 2 (GR2A) compare-match
1	DCNT8I counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8I ≠ H'0000) • Offset one-shot pulse function: Set on OCR2A compare-match (DCNT8I ≠ H'0000 or reload possible) or by user program (DCNT8I ≠ H'0000)

- Bit 7—Down-Count Start 8H (DST8H): Starts down-counter 8H (DCNT8H).

Bit 7: DST8H	Description
0	DCNT8H is halted (Initial value) [Clearing conditions] When the DCNT8H value underflows, or on channel 1 (GR1H or OCR1) compare-match
1	DCNT8H counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8H ≠ H'0000) • Offset one-shot pulse function: Set on OCR1 compare-match or GR1H compare-match, or by user program (DCNT8H ≠ H'0000)

0	DCNT8G is halted (Initial value) [Clearing conditions] When the DCNT8G value underflows, or on channel 1 (GR1G or OCR1) compare-match
1	DCNT8G counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8G ≠ H'0000) • Offset one-shot pulse function: Set on OCR1 compare-match or GR1G compare-match, or by user program (DCNT8G ≠ H'0000)

- Bit 5—Down-Count Start 8F (DST8F): Starts down-counter 8F (DCNT8F).

Bit 5: DST8F	Description
0	DCNT8F is halted (Initial value) [Clearing conditions] When the DCNT8F value underflows, or on channel 1 (GR1F or OCR1) compare-match
1	DCNT8F counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8F ≠ H'0000) • Offset one-shot pulse function: Set on OCR1 compare-match or GR1F compare-match, or by user program (DCNT8F ≠ H'0000)

- Bit 4—Down-Count Start 8E (DST8E): Starts down-counter 8E (DCNT8E).

Bit 4: DST8E	Description
0	DCNT8E is halted (Initial value) [Clearing conditions] When the DCNT8E value underflows, or on channel 1 (GR1E or OCR1) compare-match
1	DCNT8E counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8E ≠ H'0000) • Offset one-shot pulse function: Set on OCR1 compare-match or GR1E compare-match, or by user program (DCNT8E ≠ H'0000)

0	DCNT8D is halted (Initial value) [Clearing conditions] When the DCNT8D value underflows, or on channel 1 (GR1D or OCR1) compare-match
1	DCNT8D counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8D ≠ H'0000) • Offset one-shot pulse function: Set on OCR1 compare-match or GR1D compare-match, or by user program (DCNT8D ≠ H'0000)

- Bit 2—Down-Count Start 8C (DST8C): Starts down-counter 8C (DCNT8C).

Bit 2: DST8C	Description
0	DCNT8C is halted (Initial value) [Clearing conditions] When the DCNT8C value underflows, or on channel 1 (GR1C or OCR1) compare-match
1	DCNT8C counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8C ≠ H'0000) • Offset one-shot pulse function: Set on OCR1 compare-match or GR1C compare-match, or by user program (DCNT8C ≠ H'0000)

- Bit 1—Down-Count Start 8B (DST8B): Starts down-counter 8B (DCNT8B).

Bit 1: DST8B	Description
0	DCNT8B is halted (Initial value) [Clearing conditions] When the DCNT8B value underflows, or on channel 1 (GR1B or OCR1) compare-match
1	DCNT8B counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8B ≠ H'0000) • Offset one-shot pulse function: Set on OCR1 compare-match or GR1B compare-match, or by user program (DCNT8B ≠ H'0000)

0	DCNT8A is halted [Clearing conditions] When the DCNT8A value underflows, or on channel 1 (GR1A or OCR1) compare-match	(Initial value)
1	DCNT8A counts [Setting conditions] • One-shot pulse function: Set by user program (DCNT8A ≠ H'0000) • Offset one-shot pulse function: Set on OCR1 compare-match or GR1A compare-match, or by user program (DCNT8A ≠ H'0000)	

11.2.12 Timer Connection Register (TCNR)

The timer connection register (TCNR) is a 16-bit register. The ATU-II has one TCNR register in channel 8.

Bit:	15	14	13	12	11	10	9	8
	CN8P	CN8O	CN8N	CN8M	CN8L	CN8K	CN8J	CN8I
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	CN8H	CN8G	CN8F	CN8E	CN8D	CN8C	CN8B	CN8A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNR is a 16-bit readable/writable register that enables or disables connection between the channel 8 down-count start register (DSTR) and channel 1 and 2 compare-match signals (down-count start triggers). Channel 1 down-count start triggers A to H are channel 1 OCR1 compare-match signals or GR1x compare-match signals (set in TRGMDR). Channel 2 down-count start triggers A to H are channel 2 OCR2x compare-match signals. When GR1x compare-matches are used, set TIOR1A to TIOR1D to allow compare-matches.

TCNR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 11.3.5, One-Shot Pulse Function, and 11.3.6, Offset One-Shot Pulse Function and Output Cutoff Function.

Bit 15: CN8P	Description
0	Connection between DST8P and channel 2 down-count start trigger H is disabled (Initial value)
1	Connection between DST8P and channel 2 down-count start trigger H is enabled

- Bit 14—Connection Flag 8O (CN8O): Enables or disables connection between DST8O and the channel 2 down-count start trigger.

Bit 14: CN8O	Description
0	Connection between DST8O and channel 2 down-count start trigger G is disabled (Initial value)
1	Connection between DST8O and channel 2 down-count start trigger G is enabled

- Bit 13—Connection Flag 8N (CN8N): Enables or disables connection between DST8N and the channel 2 down-count start trigger.

Bit 13: CN8N	Description
0	Connection between DST8N and channel 2 down-count start trigger F is disabled (Initial value)
1	Connection between DST8N and channel 2 down-count start trigger F is enabled

- Bit 12—Connection Flag 8M (CN8M): Enables or disables connection between DST8M and the channel 2 down-count start trigger.

Bit 12: CN8M	Description
0	Connection between DST8M and channel 2 down-count start trigger E is disabled (Initial value)
1	Connection between DST8M and channel 2 down-count start trigger E is enabled

Bit 11: CN8L	Description
0	Connection between DST8L and channel 2 down-count start trigger D is disabled (Initial value)
1	Connection between DST8L and channel 2 down-count start trigger D is enabled

- Bit 10—Connection Flag 8K (CN8K): Enables or disables connection between DST8K and the channel 2 down-count start trigger.

Bit 10: CN8K	Description
0	Connection between DST8K and channel 2 down-count start trigger C is disabled (Initial value)
1	Connection between DST8K and channel 2 down-count start trigger C is enabled

- Bit 9—Connection Flag 8J (CN8J): Enables or disables connection between DST8J and the channel 2 down-count start trigger.

Bit 9: CN8J	Description
0	Connection between DST8J and channel 2 down-count start trigger B is disabled (Initial value)
1	Connection between DST8J and channel 2 down-count start trigger B is enabled

- Bit 8—Connection Flag 8I (CN8I): Enables or disables connection between DST8I and the channel 2 down-count start trigger.

Bit 8: CN8I	Description
0	Connection between DST8I and channel 2 down-count start trigger A is disabled (Initial value)
1	Connection between DST8I and channel 2 down-count start trigger A is enabled

Bit 7: CN8H	Description
0	Connection between DST8H and channel 1 down-count start trigger H is disabled (Initial value)
1	Connection between DST8H and channel 1 down-count start trigger H is enabled

- Bit 6—Connection Flag 8G (CN8G): Enables or disables connection between DST8G and the channel 1 down-count start trigger.

Bit 6: CN8G	Description
0	Connection between DST8G and channel 1 down-count start trigger G is disabled (Initial value)
1	Connection between DST8G and channel 1 down-count start trigger G is enabled

- Bit 5—Connection Flag 8F (CN8F): Enables or disables connection between DST8F and the channel 1 down-count start trigger.

Bit 5: CN8F	Description
0	Connection between DST8F and channel 1 down-count start trigger F is disabled (Initial value)
1	Connection between DST8F and channel 1 down-count start trigger F is enabled

- Bit 4—Connection Flag 8E (CN8E): Enables or disables connection between DST8E and the channel 1 down-count start trigger.

Bit 4: CN8E	Description
0	Connection between DST8E and channel 1 down-count start trigger E is disabled (Initial value)
1	Connection between DST8E and channel 1 down-count start trigger E is enabled

Bit 3: CN8D	Description
0	Connection between DST8D and channel 1 down-count start trigger D is disabled (Initial value)
1	Connection between DST8D and channel 1 down-count start trigger D is enabled

- Bit 2—Connection Flag 8C (CN8C): Enables or disables connection between DST8C and the channel 1 down-count start trigger.

Bit 2: CN8C	Description
0	Connection between DST8C and channel 1 down-count start trigger C is disabled (Initial value)
1	Connection between DST8C and channel 1 down-count start trigger C is enabled

- Bit 1—Connection Flag 8B (CN8B): Enables or disables connection between DST8B and the channel 1 down-count start trigger.

Bit 1: CN8B	Description
0	Connection between DST8B and channel 1 down-count start trigger B is disabled (Initial value)
1	Connection between DST8B and channel 1 down-count start trigger B is enabled

- Bit 0—Connection Flag 8A (CN8A): Enables or disables connection between DST8A and the channel 1 down-count start trigger.

Bit 0: CN8A	Description
0	Connection between DST8A and channel 1 down-count start trigger A is disabled (Initial value)
1	Connection between DST8A and channel 1 down-count start trigger A is enabled

The one-shot pulse terminate register (OTR) is a 16-bit register. The ATU-II has one OTR register in channel 8.

Bit:	15	14	13	12	11	10	9	8
	OTEP	OTEO	OTEN	OTEM	OTEL	OTEK	OTEJ	OTEI
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	OTEH	OTEG	OTEF	OTEE	OTED	OTEC	OTEB	OTEA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

OTR is a 16-bit readable/writable register that enables or disables forced termination of channel 8 one-shot pulse output by channel 1 and 2 compare-match signals. When one-shot pulse output is forcibly terminated, the corresponding DSTR bit and down-counter are cleared, and the corresponding TSR8 bit is set. The channel 1 one-shot pulse terminate signal is generated by GR1A to GR1H compare-matches and OCR1 compare-match (see TRGMDR). The channel 2 one-shot pulse terminate signal is generated by GR2A to GR2H compare-matches. To generate the terminate signal with GR1A to GR1H and GR2A to GR2H, select the respective compare-matches in TIOR1A to TIOR1D.

OTR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 15—One-Shot Pulse Terminate Enable P (OTEP): Enables or disables forced termination of output by channel 2 down-counter terminate trigger H.

Bit 15: OTEP	Description
0	Forced termination of TO8P by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8P by down-counter terminate trigger is enabled

Bit 14: OTEO	Description
0	Forced termination of TO8O by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8O by down-counter terminate trigger is enabled

- Bit 13—One-Shot Pulse Terminate Enable N (OTEN): Enables or disables forced termination of output by channel 2 down-counter terminate trigger F.

Bit 13: OTEN	Description
0	Forced termination of TO8N by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8N by down-counter terminate trigger is enabled

- Bit 12—One-Shot Pulse Terminate Enable M (OTEM): Enables or disables forced termination of output by channel 2 down-counter terminate trigger E.

Bit 12: OTEM	Description
0	Forced termination of TO8M by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8M by down-counter terminate trigger is enabled

- Bit 11—One-Shot Pulse Terminate Enable L (OTEL): Enables or disables forced termination of output by channel 2 down-counter terminate trigger D.

Bit 11: OTEL	Description
0	Forced termination of TO8L by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8L by down-counter terminate trigger is enabled

- Bit 10—One-Shot Pulse Terminate Enable K (OTEK): Enables or disables forced termination of output by channel 2 down-counter terminate trigger C.

Bit 10: OTEK	Description
0	Forced termination of TO8K by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8K by down-counter terminate trigger is enabled

Bit 9: OTEJ	Description
0	Forced termination of TO8J by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8J by down-counter terminate trigger is enabled

- Bit 8—One-Shot Pulse Terminate Enable I (OTEI): Enables or disables forced termination of output by channel 2 down-counter terminate trigger A.

Bit 8: OTEI	Description
0	Forced termination of TO8I by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8I by down-counter terminate trigger is enabled

- Bit 7—One-Shot Pulse Terminate Enable H (OTEH): Enables or disables forced termination of output by channel 1 down-counter terminate trigger H.

Bit 7: OTEH	Description
0	Forced termination of TO8H by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8H by down-counter terminate trigger is enabled

- Bit 6—One-Shot Pulse Terminate Enable G (OTEG): Enables or disables forced termination of output by channel 1 down-counter terminate trigger G.

Bit 6: OTEG	Description
0	Forced termination of TO8G by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8G by down-counter terminate trigger is enabled

- Bit 5—One-Shot Pulse Terminate Enable F (OTEF): Enables or disables forced termination of output by channel 1 down-counter terminate trigger F.

Bit 5: OTEF	Description
0	Forced termination of TO8F by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8F by down-counter terminate trigger is enabled

Bit 4: OTEE	Description
0	Forced termination of TO8E by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8E by down-counter terminate trigger is enabled

- Bit 3—One-Shot Pulse Terminate Enable D (OTED): Enables or disables forced termination of output by channel 1 down-counter terminate trigger D.

Bit 3: OTED	Description
0	Forced termination of TO8D by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8D by down-counter terminate trigger is enabled

- Bit 2—One-Shot Pulse Terminate Enable C (OTEC): Enables or disables forced termination of output by channel 1 down-counter terminate trigger C.

Bit 2: OTEC	Description
0	Forced termination of TO8C by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8C by down-counter terminate trigger is enabled

- Bit 1—One-Shot Pulse Terminate Enable B (OTEB): Enables or disables forced termination of output by channel 1 down-counter terminate trigger B.

Bit 1: OTEB	Description
0	Forced termination of TO8B by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8B by down-counter terminate trigger is enabled

- Bit 0—One-Shot Pulse Terminate Enable A (OTEA): Enables or disables forced termination of output by channel 1 down-counter terminate trigger A.

Bit 0: OTEA	Description
0	Forced termination of TO8A by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8A by down-counter terminate trigger is enabled

The reload enable register (RLDENR) is an 8-bit register. The ATU-II has one RLDENR register in channel 8.

Bit:	7	6	5	4	3	2	1	0
	RLDEN	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

RLDENR is an 8-bit readable/writable register that enables or disables loading of the reload register8 (RLDR8) value into the down-counters (DCNT8I to DCNT8P). Loading is performed on generation of a channel 2 compare-match signal one-shot pulse start trigger. Reloading is not performed if there is no linkage with channel 2 (one-shot pulse function), or while the down-counter (DCNT8I to DCNT8P) is running.

RLDENR is initialized to H'00 by a power-on reset and in hardware standby mode and software standby mode.

- Bit 7—Reload Enable (RLDEN): Enables or disables loading of the RLDR value into DCNT8I to DCNT8P.

Bit 7: RLDEN	Description
0	Loading of reload register value into down-counters is disabled (Initial value)
1	Loading of reload register value into down-counters is enabled

- Bits 6 to 0—Reserved: These bits are always read as 0. The write value should always be 0.

The free-running counters (TCNT) are 32- or 16-bit up- or up/down-counters. The ATU-II has 17 TCNT counters: one 32-bit TCNT in channel 0, and sixteen 16-bit TCNTs in each of channels 1 to 7 and 11. For details of the channel 10 free-running counters, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
0	TCNT0H, TCNT0L	32-bit up-counter (initial value Hí00000000)
1	TCNT1A, TCNT1B	16-bit up-counters (initial value H'0000)
2	TCNT2A, TCNT2B	
3	TCNT3	
4	TCNT4	
5	TCNT5	
6	TCNT6A–D	16-bit up/down-counters (initial value H'0001)
7	TCNT7A–D	16-bit up-counters (initial value H'0001)
11	TCNT11	16-bit up-counter (initial value H'0000)

Free-Running Counter 0 (TCNT0H, TCNT0L): Free-running counter 0 (comprising TCNT0H and TCNT0L) is a 32-bit readable/writable register that counts on an input clock. The counter is started when the corresponding bit in the timer start register (TSTR1) is set to 1. The input clock is selected with prescaler register 1 (PSCR1).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When TCNT0 overflows (from H'FFFFFFFF to H'00000000), the OVF0 overflow flag in the timer status register (TSR0) is set to 1.

TCNT0 can only be accessed by a longword read or write. Word reads or writes cannot be used.

Free-Running Counters 1A, 1B, 2A, 2B, 3, 4, 5, 11 (TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, TCNT5, TCNT11): Free-running counters 1A, 1B, 2A, 2B, 3, 4, 5, and 11 (TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, TCNT5, TCNT11) are 16-bit readable/writable registers that count on an input clock. Counting is started when the corresponding bit in the timer start register (TSTR1 or TSTR3) is set to 1. The input clock is selected with prescaler register 1 (PSCR1) and the timer control register (TCR).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:																
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The TCNT1A, TCNT1B, TCNT2A, and TCNT2B counters are cleared if incremented during counter clear trigger input from channel 10.

TCNT3 to TCNT5 counter clearing is performed by a compare-match with the corresponding general register, according to the setting in TIOR.

When one of counters TCNT1A/1B/2A/2B/3/4/5/11 overflows (from H'FFFF to H'0000), the overflow flag (OVF) for the corresponding channel in the timer status register (TSR) is set to 1.

TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, TCNT5, and TCNT11 can only be accessed by a word read or write.

TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, TCNT5, and TCNT11 are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, and TCNT5 can count on external clock (TCLKA or TCLKB) input.

TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, and TCNT5 can count on an external interrupt clock (TI10) (AGCK) generated in channel 10 and on a channel 10 multiplied clock (AGCKM).

TCNT7D) are 16-bit readable/writable registers. Channel 6 and 7 counts are started by the timer start register (TSTR2).

The clock input to channels 6 and 7 is selected with prescaler registers 2 and 3 (PSCR2, PSCR3) and timer control registers 6 and 7 (TCR6, TCR7).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT6A to TCNT6D (in non-complementary PWM mode) and TCNT7A to TCNT7D are cleared by a compare-match with the cycle register (CYLR).

TCNT6A to TCNT6D (in complementary PWM mode) count up and down between zero and the cycle register value.

TCNT6A to TCNT6D and TCNT7A to TCNT7D are connected to the CPU by an internal 16-bit bus, and can only be accessed by a word read or write.

TCNT6A to TCNT6D and TCNT7A to TCNT7D are initialized to H'0001 by a power-on reset, and in hardware standby mode and software standby mode.

11.2.16 Down-Counters (DCNT)

The DCNT registers are 16-bit down-counters. The ATU-II has 16 DCNT counters in channel 8.

Channel	Abbreviation	Function
8	DCNT8A, DCNT8B, DCNT8C, DCNT8D, DCNT8E, DCNT8F, DCNT8G, DCNT8H, DCNT8I, DCNT8J, DCNT8K, DCNT8L, DCNT8M, DCNT8N, DCNT8O, DCNT8P	16-bit down-counters

selected with prescaler register 1 (PSCR1) and the timer control register (TCR).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:																
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When the one-shot pulse function is used, DCNT8x starts counting down when the corresponding DSTR bit is set to 1 by the user program after the DCNT8x value has been set. When the DCNT8x value underflows, DSTR and DCNT8x are automatically cleared to 0, and the count is stopped. At the same time, the corresponding channel 8 timer status register 8 (TSR8) status flag is set to 1.

When the offset one-shot pulse function is used, on compare-match with a channel 1 or 2 general register (GR) or output compare register (OCR) (the compare-match setting being made in the trigger mode register (TRGMDR) (for channel 1 only)) when the corresponding timer connection register (TCNR) bit is 1, the corresponding down-count start register (DSTR) bit is automatically set to 1 and the down-count is started. When the DCNT8x value underflows, the corresponding DSTR bit and DCNT8x are automatically cleared to 0, the count is stopped, and the output is inverted, or, if a one-shot terminate register (OTR) setting has been made to forcibly terminate output by means of a trigger, DSTR is cleared to 0 by a channel 1 or 2 compare-match between GR and OCR, the count is forcibly terminated, and the output is inverted. The output is inverted for whichever is first. When the output is inverted, the corresponding channel 8 TSR8 status flag is set to 1.

The DCNT8x counters can only be accessed by a word read or write.

The DCNT8x counters are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 11.3.5, One-Shot Pulse Function, and 11.3.6, Offset One-Shot Pulse Function and Output Cutoff Function.

The event counters (ECNT) are 8-bit up-counters. The ATU-II has six ECNT counters in channel 9.

Channel	Abbreviation	Function
9	ECNT9A, ECNT9B, ECNT9C, ECNT9D, ECNT9E, ECNT9F	8-bit event counters

The ECNT counters are 8-bit readable/writable registers that count on detection of an input signal from input pins TI9A to TI9F. Rising edge, falling edge, or both rising and falling edges can be selected for edge detection.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When a compare-match with GR9 corresponding to an ECNT9x counter occurs, the compare-match flag (CMF9) in the timer status register (TSR9) is set to 1. When a compare-match with GR occurs, the ECNT9x counter is cleared automatically.

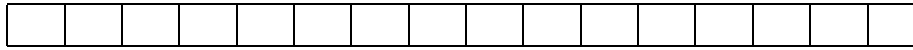
The ECNT9x counters can only be accessed by a byte read or write.

The ECNT9x counters are initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

11.2.18 Output Compare Registers (OCR)

The output compare registers (OCR) are 16-bit registers. The ATU-II has nine OCR registers: one in channel 1 and eight in channel 2. For details of the channel 10 free-running counters, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
1	OCR1	Output compare registers
2	OCR2A, OCR2B, OCR2C, OCR2D, OCR2E, OCR2F, OCR2G, OCR2H	



Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The OCR registers are 16-bit readable/writable registers that have an output compare register function.

The OCR and free-running counter (TCNT1B, TCNT2B) values are constantly compared, and if the two values match, the CMF bit in the timer status register (TSR) is set to 1. If channels 1 and 2 and channel 8 are linked by the timer connection register (TCNR), the corresponding channel 8 down-counter (DCNT) is started at the same time.

The OCR registers can only be accessed by a word read.

The OCR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

11.2.19 Input Capture Registers (ICR)

The input capture registers (ICR) are 32-bit registers. The ATU-II has four 32-bit ICR registers in channel 0. For details of the channel 10 free-running counters, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
0	ICR0AH, ICR0AL, ICR0BH, ICR0BL, ICR0CH, ICR0CL, ICR0DH, ICR0DL	Dedicated input capture registers

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The ICR registers are 32-bit read-only registers used exclusively for input capture.

These dedicated input capture registers store the TCNT0 value on detection of an input capture signal from an external source. The corresponding TSR0 bit is set to 1 at this time. The input capture signal edge to be detected is specified by timer I/O control register TIOR0. By setting the TRG0DEN bit in TCR10, ICR0DH and ICR0DL can also be used for input capture in a compare match between TCNT10B and OCR10B.

The ICR registers can only be accessed by a longword read. Word reads cannot be used.

The ICR registers are initialized to H'00000000 by a power-on reset, and in hardware standby mode and software standby mode.

11.2.20 General Registers (GR)

The general registers (GR) are 16-bit registers. The ATU-II has 36 general registers: eight each in channels 1 and 2, four each in channels 3 to 5, six in channel 9, and two in channel 11. For details of the channel 10 free-running counters, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
1	GR1A–GR1H	Dual-purpose input capture and output compare registers
2	GR2A–GR2H	
3	GR3A–GR3D	
4	GR4A–GR4D	
5	GR5A–GR5D	
9	GR9A–GR9F	Dedicated output compare registers
11	GR11A, GR11B	Dual-purpose input capture and output compare registers

Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

These GR registers are 16-bit readable/writable registers with both input capture and output compare functions. Function switching is performed by means of the timer I/O control registers (TIOR).

When a general register is used for input capture, it stores the TCNT1A or TCNT2A value on detection of an input capture signal from an external source. The corresponding IMF bit in TSR is set to 1 at this time. The input capture signal edge to be detected is specified by the corresponding TIOR.

When a general register is used for output compare, the GR value and free-running counter (TCNT1A, TCNT2A) value are constantly compared, and when both values match, the IMF bit in the timer status register (TSR) is set to 1. If connection of channels 1 and 2 and channel 8 is specified in the timer connection register (TCNR), the corresponding channel 8 down-counter (DCNT) is started. Compare-match output is specified by the corresponding TIOR.

The GR registers can only be accessed by a word read or write.

The GR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

General Registers 3A to 3D, 4A to 4D, 5A to 5D, 11A and 11B
(GR3A to GR3D, GR4A to GR4D, GR5A to GR5D, GR11A and GR11B)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

These GR registers are 16-bit readable/writable registers with both input capture and output compare functions. Function switching is performed by means of the timer I/O control registers (TIOR).

When a general register is used for input capture, it stores the corresponding TCNT value on detection of an input capture signal from an external source. The corresponding IMF bit in TSR is set to 1 at this time. The input capture signal edge to be detected is specified by the corresponding

When a general register is used for output compare, the GR value and free-running counter (TCNT) value are constantly compared, and when both values match, the IMF bit in the timer status register (TSR) is set to 1. Compare-match output is specified by the corresponding TIOR.

GRIIA and GR11B compare-match signals are transmitted to the advanced pulse controller (APC). For details, see section 12, Advanced Pulse Controller (APC).

The GR registers can only be accessed by a word read or write.

The GR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

General Registers 9A to 9F (GR9A to GR9F)

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

These GR registers are 8-bit readable/writable registers with a compare-match function.

The GR value and event counter (ECNT) value are constantly compared, and when both values match a compare-match signal is generated and the next edge is input, the corresponding CMF bit in TSR is set to 1.

In addition, channel 3 (GR3A to GR3D) input capture can be generated by GR9A to GR9D compare-matches. This function is set by TRG3xEN in the timer control register (TCR).

The GR registers can be accessed by a byte read or write.

The GR registers are initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode.

The offset base registers (OSBR) are 16-bit registers. The ATU-II has two OSBR registers, one each in channels 1 and 2.

Channel	Abbreviation	Function
1	OSBR1	Dedicated input capture registers with signal from channel 0
2	OSBR2	ICR0A as input trigger

Offset Base Registers 1 and 2 (OSBR1, OSBR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

OSBR1 and OSBR2 are 16-bit read-only registers used exclusively for input capture. OSBR1 and OSBR2 use the channel 0 ICR0A input capture register input as their trigger signal, and store the TCNT1A or TCNT2A value on detection of an edge.

The OSBR registers can only be accessed by a word read.

The OSBR registers are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 11.3.8, Twin Capture Function.

11.2.22 Cycle Registers (CYLR)

The cycle registers (CYLR) are 16-bit registers. The ATU-II has eight cycle registers, four each in channels 6 and 7.

Channel	Abbreviation	Function
6	CYLR6A– CYLR6D	16-bit PWM cycle registers
7	CYLR7A– CYLR7D	

Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The CYLR registers are 16-bit readable/writable registers used for PWM cycle storage.

The CYLR value is constantly compared with the corresponding free-running counter (TCNT6A to TCNT6D, TCNT7A to TCNT7D) value, and when the two values match, the corresponding timer start register (TSR) bit (CMF6A to CMF6D, CMF7A to CMF7D) is set to 1, and the free-running counter (TCNT6A to TCNT6D, TCNT7A to TCNT7D) is cleared. At the same time, the buffer register (BFR) value is transferred to the duty register (DTR). Output pin (TO6A to TO6D, TO7A to TO7D) of corresponding channnel will be 0 when H'0000 of BFR is 0 output and otherwise will be 1.

The CYLR registers can only be accessed by a word read or write.

The CYLR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

For details of the CYLR, BFR, and DTR registers, see section 11.3.9, PWM Timer Function.

11.2.23 Buffer Registers (BFR)

The buffer registers (BFR) are 16-bit registers. The ATU-II has eight buffer registers, four each in channels 6 and 7.

Channel	Abbreviation	Function
6	BFR6A–BFR6D	16-bit PWM buffer registers
7	BFR7A–BFR7D	Buffer register (BFR) value is transferred to duty register (DTR) on compare-match of corresponding cycle register (CYLR)

Buffer Registers (BFR6A to BFR6D, BFR7A to BFR7D)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The BFR registers can only be accessed by a word read or write.

The BFR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

11.2.24 Duty Registers (DTR)

The duty registers (DTR) are 16-bit registers. The ATU-II has eight duty registers, four each in channels 6 and 7.

Channel	Abbreviation	Function
6	DTR6A–DTR6D	16-bit PWM duty registers
7	DTR7A–DTR7D	

Duty Registers (DTR6A to DTR6D, DTR7A to DTR7D)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The DTR registers are 16-bit readable/writable registers used for PWM duty storage.

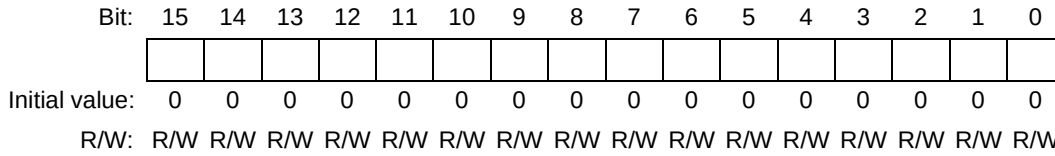
The DTR value is constantly compared with the corresponding free-running counter (TCNT6A to TCNT6D, TCNT7A to TCNT7D) value, and when the two values match, the corresponding channel output pin (TO6A to TO6D, TO7A to TO7D) goes to 0 output. Also, when CYLR and the corresponding the free-running counter match, the corresponding BFR value is loaded. Set a value in the range 0 to CYLR for DTR; do not set a value greater than CYLR.

The DTR registers can only be accessed by a word read or write.

The DTR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

The reload register is a 16-bit register. The ATU-II has one RLDR register in channel 8.

Reload Register 8 (RLDR8)



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RLDR8 is a 16-bit readable/writable register. When reload is enabled (by a setting in RLDENR) and DSTR8I to DSTR8P are set to 1 by the channel 2 compare-match signal one-shot pulse start trigger, the reload register value is transferred to DCNT8I to DCNT8P before the down-count is started. The reload register value is not transferred when the one-shot pulse function is used independently, without linkage to channel 2, or when down-counters DCNT8I to DCNT8P are running.

RLDR8 can only be accessed by a word read or write.

RLDR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

11.2.26 Channel 10 Registers

Counters (TCNT)

Channel 10 has seven TCNT counters: one 32-bit TCNT, four 16-bit TCNTs, and two 8-bit TCNTs.

The input clock is selected with prescaler register 4 (PSCR4). Count operations are performed by setting STR10 to 1 in timer start register 1 (TSTR1).

Channel	Abbreviation	Function
10	TCNT10AH, AL	32-bit free-running counter (initial value H'00000001)
	TCNT10B	8-bit event counter (initial value H'00)
	TCNT10C	16-bit reload counter (initial value H'0001)
	TCNT10D	8-bit correction counter (initial value H'00)
	TCNT10E	16-bit correction counter (initial value H'0000)
	TCNT10F	16-bit correction counter (initial value H'0001)
	TCNT10G	16-bit free-running counter (initial value H'0000)

an input clock and is cleared to the initial value by input capture input (TI10) (AGCK).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT10A can only be accessed by a longword read or write. Word reads or writes cannot be used.

TCNT10A is initialized to H'00000001 by a power-on reset, and in hardware standby mode and software standby mode.

Event Counter 10B (TCNT10B): Event counter 10B (TCNT10B) is an 8-bit readable/writable register that counts on external clock input (TI10) (AGCK). For this operation, TI10 input must be set with bits CKEG1 and CKEG0 in TCR10. TI10 input will be counted even if halting of the count operation is specified by bit STR10 in TSTR1.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT10B can only be accessed by a byte read or write.

TCNT10B is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When TCNT10C = H'0001 in the down-count operation, the value in the reload register (RLD10C) is transferred to TCNT10C, and a multiplied clock (AGCK1) is generated.

TCNT10C is connected to the CPU via an internal 16-bit bus, and can only be accessed by a word read or write.

TCNT10C is initialized to H'0001 by a power-on reset, and in hardware standby mode and software standby mode.

Correction Counter 10D (TCNT10D): Correction counter 10D (TCNT10D) is an 8-bit readable/writable register that counts on external clock input (TI10) after transfer of the counter value to correction counter E (TCNT10E). Set TI10 input with bits CKEG1 and CKEG0 in TCR10. Transfer and counting will not be performed on TI10 input unless the count operation is enabled by bit STR10 in TSTR1.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

At the external clock input (TI10) (AGCK) timing, the value in this counter is shifted according to the multiplication factor set by bits PIM1 and PIM0 in timer I/O control register 10 (TIOR10) and transferred to correction counter E (TCNT10E).

TCNT10D can only be accessed by a byte read or write.

TCNT10D is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

and counts on the multiplied clock (AGCKM) output by reload counter 10C (TCNT10C). However, if CCS in timer I/O control register 10 (TIOR10) is set to 1, when the TCNT10D shifted value is reached the count is halted.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT10E can only be accessed by a word read or write.

TCNT10E is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Correction Counter 10F (TCNT10F): Correction counter 10F (TCNT10F) is a 16-bit readable/writable register that counts up on Pϕ clock cycles if the counter value is smaller than the correction counter 10E (TCNT10E) value when the STR10 bit in TSTR1 has been set for counter operation. The count is halted by a match with the correction counter clear register (TCCLR10). If TI10 is input when TCNT10D = H'00, TCNT10F is initialized and correction is carried out. When TCNT10F = TCCLR10, TCNT10F is cleared to H'0001. While TCNT10F ≠ TCCLR10, TCNT10F is incremented automatically until it reaches the TCCLR10 value, and is then cleared to H'0001.

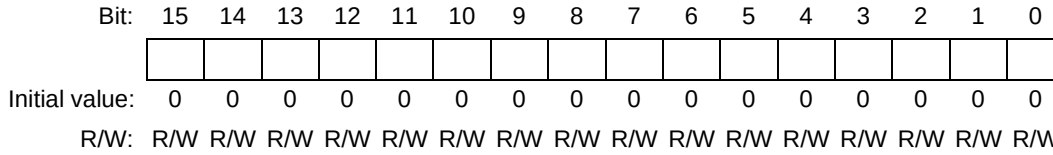
A corrected clock (AGCKM) is output following correction each time this counter is incremented.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT10F is can only be accessed by a word read or write.

TCNT10F is initialized to H'0001 by a power-on reset, and in hardware standby mode and software standby mode.

initialized to H'0000 by input from external input (TI10) (AGCR).



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT10G can only be accessed by a word read or write.

TCNT10G is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Registers

There are six registers in channel 10: a 32-bit ICR, 32-bit OCR, 16-bit GR, 16-bit RLD, 16-bit TCCLR, and 8-bit OCR.

Channel	Abbreviation	Function
10	ICR10AH, AL	32-bit input capture register (initial value H'00000000)
	OCR10AH, AL	32-bit output compare register (initial value H'FFFFFFFF)
	OCR10B	8-bit output compare register (initial value H'FF)
	RLD10C	16-bit reload register (initial value H'0000)
	GR10G	16-bit general register (initial value H'FFFF)
	TCCLR10	16-bit correction counter clear register (initial value H'0000)

value is transferred on external input (FTI0) (AGCR). At the same time, ICF10A in timer status register 10 (TSR10) is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ICR10A is initialized to H'00000000 by a power-on reset, and in hardware standby mode and software standby mode.

Output Compare Register 10AH, AL (OCR10AH, OCR10AL): Output compare register 10AH, AL (comprising OCR10AH and OCR10AL) is a 32-bit readable/writable register that is constantly compared with free-running counter 10AH, AL (TCNT10AH, TCNT10AL). When both values match, CMF10A in timer status register 10 (TSR10) is set to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

OCR10A is initialized to H'FFFFFFFF by a power-on reset, and in hardware standby mode and software standby mode.

which AGCK is input with both values matching, CMF10B in timer status register 10 (TSR10) is set to 1.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

OCR10B is initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode.

Reload Register 10C (RLD10C): Reload register 10C (RLD10C) is a 16-bit readable/writable register. When STR10 in timer start register 1 (TSTR1) is 1 and RLDEN in the timer I/O control register (TIOR10) is 0, and the value of TCNT10A is captured into input capture register 10A (ICR10A), the ICR10A capture value is shifted according to the multiplication factor set by bits PIM1 and PIM0 in TIOR10 before being transferred to RLD10C. The contents of reload register 10C (RLD10C) are loaded when reload counter 10C (TCNT10C) reaches H'0001.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RLD10C is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

General Register 10G (GR10G): General register 10G (GR10G) is a 16-bit readable/writable register with an output compare function. Function switching is performed by means of timer I/O control register 10 (TIOR10). The GR10G value and free-running counter 10G (TCNT10G) value are constantly compared, and when AGCK is input with both values matching, CMF10G in timer status register 10 (TSR10) is set to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

GR10G is initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

TCCLR10 is constantly compared with TCNT10F, and when the two values match, TCNT10F halts. TCNTxx can be cleared at this time by setting TRGxxEN (xx = 1A, 1B, 2A, 2B) in TCR10. Then, when TCNT10D is H'00 and TI10 is input, TCNT10F is cleared to H'0001.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCCLR10 is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Noise Canceler Registers

There are two 8-bit noise canceler registers in channel 10: TCNT10H and NCR10.

Channel	Abbreviation	Function	
10	TCNT10H	Noise canceler counter	(Initial value H'00)
	NCR10	Noise canceler compare-match register	(Initial value H'FF)

Noise Canceler Counter 10H (TCNT10H): Noise canceler counter 10H (TCNT10H) is an 8-bit readable/writable register. When the noise canceler function is enabled, TCNT10H starts counting up on $P\phi \times 10$, with the signal from external input (TI10) (AGCK) as a trigger. The counter operates even if STR10 is cleared to 0 in the timer start register (TSTR1). TI10 input is masked while the counter is running. When the count matches the noise canceler register (NCR10) value, the counter is cleared and TI10 input masking is released.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT10H is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

(TCNT10H). TCNT10H is constantly compared with NCR10 during the count, and when a compare-match occurs the TCNT10H counter is halted and input signal masking is released.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NCR10 is initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode.

Channel 10 Control Registers

There are four control registers in channel 10.

Channel	Abbreviation	Function
10	TIOR10	Reload setting, counter correction setting, external input (TI10) edge interval multiplier setting
		GR compare-match setting (Initial value H'00)
	TCR10	TCCLR10 counter clear source
		Noise canceler function enabling/disabling selection
		External input (TI10) edge selection (Initial value H'00)
	TSR10	Input capture/compare-match status (Initial value H'0000)
	TIER10	Input capture/compare-match interrupt request enabling/disabling selection (Initial value H'0000)

setting for using the general register (GR10G) for output compare, and makes the edge detection setting.

TIOR10 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	RLDEN	CCS	PIM1	PIM0	—	IO10G2	IO10G1	IO10G0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

- Bit 7—Reload Enable (RLDEN): Enables or disables transfer of the input capture register 10A (ICR10A) value to reload register 10C (RLD10C).

Bit 7: RLDEN	Description
0	Transfer of ICR10A value to RLD10C on input capture is enabled (Initial value)
1	Transfer of ICR10A value to RLD10C on input capture is disabled

- Bit 6—Counter Clock Select (CCS): Selects the operation of correction counter 10E (TCNT10E). Set the multiplication factor with bits PIM1 and PIM0.

Bit 6: CCS	Description
0	TCNT10E count is not halted when TCNT10D x multiplication factor = TCNT10E* (Initial value)
1	TCNT10E count is halted when TCNT10D x multiplication factor = TCNT10E*

Note: * When [TCNT10D × multiplication factor] matches the value of TCNT10E with bits 8 to 0 masked

- Bits 5 and 4—Pulse Interval Multiplier (PIM1, PIM0): These bits select the external input (TI10) cycle multiplier.

Bit 5: PIM1	Bit 4: PIM0	Description
0	0	Counting on external input cycle × 32 (Initial value)
	1	Counting on external input cycle × 64
1	0	Counting on external input cycle × 128
	1	Counting on external input cycle × 256

general register 10G (GR10G).

Bit 2: IO10G2	Bit 1: IO10G1	Bit 0: IO10G0	Description	
0	0	0	GR is an output compare register	Compare-match disabled (Initial value)
		1		GR10G = TCNT10G compare-match
	1	*		Cannot be used
1	*	*	Cannot be used	

*: Donít care

Timer Control Register 10 (TCR10): TCR10 is an 8-bit readable/writable register that selects the correction counter clear register (TCCLR10) compare-match counter clear source, enables or disables the noise canceler function, and selects the external input (TI10) edge.

TCR10 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	TRG2BEN	TRG1BEN	TRG2AEN	TRG1AEN	TRG0DEN	NCE	CKEG1	CKEG0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 7—Trigger 2B Enable (TRG2BEN): Enables or disables counter clearing for channel 2 TCNT2B. When clearing is enabled, set the correction angle clock (AGCKM) as the TCNT2B count clock. If TCNT2B counts while clearing is enabled, TCNT2B will be cleared.

Bit 7: TRG2BEN	Description
0	Channel 2 counter B (TCNT2B) clearing when correction counter clear register (TCCLR10) = correction counter (TNT10F) is disabled (Initial value)
1	Channel 2 counter B (TCNT2B) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is enabled

count clock. If TCNT1B counts while clearing is enabled, TCNT1B will be cleared.

Bit 6: TRG1BEN	Description
0	Channel 1 counter B (TCNT1B) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is disabled (Initial value)
1	Channel 1 counter B (TCNT1B) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is enabled

- Bit 5—Trigger 2A Enable (TRG2AEN): Enables or disables counter clearing for channel 2 TCNT2A. When clearing is enabled, set the correction angle clock (AGCKM) as the TCNT2A count clock. If TCNT2A counts while clearing is enabled, TCNT2A will be cleared.

Bit 5: TRG2AEN	Description
0	Channel 2 counter 2A (TCNT2A) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is disabled (Initial value)
1	Channel 2 counter 2A (TCNT2A) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is enabled

- Bit 4—Trigger 1A Enable (TRG1AEN): Enables or disables counter clearing for channel 1 TCNT1A. When clearing is enabled, set the correction angle clock (AGCKM) as the TCNT1A count clock. If TCNT1A counts while clearing is enabled, TCNT1A will be cleared.

Bit 4: TRG1AEN	Description
0	Channel 1 counter 1A (TCNT1A) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is disabled (Initial value)
1	Channel 1 counter 1A (TCNT1A) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is enabled

- Bit 3—Trigger 0D Enable (TRG0DEN): Enables or disables channel 0 ICR0D input capture signal requests.

Bit 3: TRG0DEN	Description
0	Capture requests for channel 0 input capture register (ICR0D) on event counter (TCNT10B) compare-match are disabled (Initial value)
1	Capture requests for channel 0 input capture register (ICR0D) on event counter (TCNT10B) compare-match are enabled

0	Noise canceler function is disabled	(Initial value)
1	Noise canceler function is enabled	

- Bits 1 and 0—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select the channel 10 external input (TI10) edge(s). The clock (AGCK) is generated by the detected edge(s).

Bit 1: CKEG1	Bit 0: CKEG0	Description	
0	0	TI10 input disabled	(Initial value)
	1	TI10 input rising edges detected	
1	0	TI10 input falling edges detected	
	1	TI10 input rising and falling edges both detected	

Timer Status Register 10 (TSR10): TSR10 is a 16-bit readable/writable register that indicates the occurrence of channel 10 input capture or compare-match.

Each flag is an interrupt source, and issues an interrupt request to the CPU if the interrupt is enabled by the corresponding bit in timer interrupt enable register 10 (TIER10).

TSR10 is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	CMF10G	CMF10B	ICF10A	CMF10A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag.

match.

Bit 3: CMF10G	Description
0	[Clearing condition] (Initial value) When CMF10G is read while set to 1, then 0 is written to IMF10G
1	[Setting condition] When TCNT10G = GR10G

- Bit 2—Compare-Match Flag 10B (CMF10B): Status flag that indicates OCR10B compare-match.

Bit 2: CMF10B	Description
0	[Clearing condition] (Initial value) When CMF10B is read while set to 1, then 0 is written to CMF10B
1	[Setting condition] When TCNT10B is incremented while TCNT10B = OCR10B

- Bit 1—Input Capture Flag 10A (ICF10A): Status flag that indicates ICR10A input capture.

Bit 1: ICF10A	Description
0	[Clearing condition] (Initial value) When ICR10A is read while set to 1, then 0 is written to ICR10A
1	[Setting condition] When the TCNT10A value is transferred to ICR10A by an input capture signal

- Bit 0—Compare-Match Flag 10A (CMF10A): Status flag that indicates OCR10A compare-match.

Bit 0: CMF10A	Description
0	[Clearing condition] (Initial value) When CMF10A is read while set to 1, then 0 is written to CMF10A
1	[Setting condition] When TCNT10A = OCR10A

TIER10 is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	IREG	CME10G	CME10B	ICE10A	CME10A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 5—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 4—Interrupt Enable Edge G (IREG): Specifies TSR10 CMF10G interrupt request timing.

Bit 4: IREG	Description	
0	Interrupt is requested when CMF10G becomes 1	(Initial value)
1	Interrupt is requested by next external input (TI10) (AGCK) after CMF10G becomes 1	

- Bit 3—Compare-Match Interrupt Enable 10G (CME10G): Enables or disables interrupt requests by CMF10G in TSR10 when CMF10G is set to 1.

Bit 3: CME10G	Description	
0	CMI10G interrupt requested by CMF10G is disabled	(Initial value)
1	CMI10G interrupt requested by CMF10G is enabled	

- Bit 2—Compare-Match Interrupt Enable 10B (CME10B): Enables or disables interrupt requests by CMF10B in TSR10 when CMF10B is set to 1.

Bit 2: CME10B	Description	
0	CMI10B interrupt requested by CMF10B is disabled	(Initial value)
1	CMI10B interrupt requested by CMF10B is enabled	

Bit 1: ICE10A	Description	
0	ICI10A interrupt requested by ICF10A is disabled	(Initial value)
1	ICI10A interrupt requested by ICF10A is enabled	

- Bit 0—Compare-Match Interrupt Enable 10A (CME10A): Enables or disables interrupt requests by CMF10A in TSR10 when CMF10A is set to 1.

Bit 0: CME10A	Description	
0	CMI10A interrupt requested by CMF10A is disabled	(Initial value)
1	CMI10A interrupt requested by CMF10A is enabled	

11.3 Operation

11.3.1 Overview

The ATU-II has twelve timers of eight kinds in channels 0 to 11. It also has a built-in prescaler that generates input clocks, and it is possible to generate or select internal clocks of the required frequency independently of circuitry outside the ATU-II.

The operation of each channel and the prescaler is outlined below.

Channel 0: Channel 0 has a 32-bit free-running counter (TCNT0) and four 32-bit input capture registers (ICR0A to ICR0D). TCNT0 is an up-counter that performs free-running operation. An interrupt request can be generated on counter overflow. The four input capture registers (ICR0A to ICR0D) capture the free-running counter (TCNT0) value by means of input from the corresponding external signal input pin (TI0A to TI0D). For capture by means of input from an external signal input pin, rising edge, falling edge, or both edges can be selected in the timer I/O control register (TIOR0). In the case of input capture register 0D (ICR0D) only, capture can be performed by means of a compare-match between free-running counter 10B (TCNT10B) and compare-match register 10B (OCR10B), by making a setting in timer control register 10 (TCR10). In this case, capture is performed even if an input capture disable setting has been made for TIOR0. In each case, the DMAC can be activated or an interrupt requested when capture occurs.

Channel 0 also has three interval interrupt request registers (ITVRR1, ITVRR2A, and ITVRR2B). A/D converter (AD0 to AD2) activation can be selected by setting 1 in ITVA6 to ITVA13 in ITVRR, and an interrupt request to the CPU by setting 1 in ITVE6 to ITVE13. These operations are performed when the corresponding bit of bits 6 to 13 in TCNT0 changes to 1, enabling use as an interval timer function.

TCNT1A and TCNT1B are up-counters that perform free-running operation. When the clock generated in channel 10 (described below) is selected, these counters can be cleared at the count specified in channel 10. Each counter can generate an interrupt request when it overflows.

The eight general registers (GR1A to GR1H) can be used as input capture or output compare registers using the corresponding external signal I/O pin (TIO1A to TIO1H). When used for input capture, the free-running counter (TCNT1A) value is captured by means of input from the corresponding external signal I/O pin (TIO1A to TIO1H). Rising edge, falling edge, or both edges can be selected for the input capture signal in the timer I/O control registers (TIOR1A to TIOR1D). When used for output compare, compare-match with the free-running counter (TCNT1A) is performed. For the output from the external signal I/O pins by compare-match, 0 output, 1 output, or toggle output can be selected in the timer I/O control registers (TIOR1A to TIOR1D). When used as output compare registers, a compare-match can be used as a one-shot pulse start/terminate trigger by setting the channel 8 timer connection register (TCNR) and one-shot pulse terminate register (OTR), and using these in combination with the down-counters (DCNT8A to DCNT8H). Start/terminate trigger selection is performed by means of the trigger mode register (TRGMDR).

The output compare register (OCR1) can be used as a one-shot pulse offset function, in the same way as the general registers, in combination with channel 8 down-counters DCNT8A to DCNT8H. An interrupt can be requested on the occurrence of the respective input capture or compare-match.

In addition, channel 1 has a 16-bit dedicated input capture register (OSBR1). The channel 0 TI0A input pin can also be used as the OSBR1 trigger input, enabling use of a twin-capture function.

Channel 2: Channel 2 has two 16-bit free-running counters (TCNT2A and TCNT2B), eight 16-bit general registers (GR2A to GR2H), and eight 16-bit output compare registers (OCR2A to OCR2H).

TCNT2A and TCNT2B are up-counters that perform free-running operation. When the clock generated in channel 10 (described below) is selected, these counters can be cleared at the count specified in channel 10. Each counter can generate an interrupt request when it overflows.

The eight general registers (GR2A to GR2H) can be used as input capture or output compare registers using the corresponding external signal I/O pin (TIO2A to TIO2H). When used for input capture, the free-running counter (TCNT2A) value is captured by means of input from the corresponding external signal I/O pin (TIO2A to TIO2H). Rising edge, falling edge, or both edges can be selected for the input capture signal in the timer I/O control registers (TIOR2A to TIOR2D). When used for output compare, compare-match with the free-running counter (TCNT2A) is performed. For the output from the external signal I/O pins by compare-match, 0 output, 1 output, or toggle output can be selected in the timer I/O control registers (TIOR2A to

this in combination with the down-counters (DCNT8I to DCNT8P).

In the case of the output compare registers (OCR2A to OCR2H), a TCNT2B compare-match can be used as a one-shot pulse start trigger by setting the channel 8 timer connection register (TCNR), and using this in combination with the down-counters (DCNT8I to DCNT8P). An interrupt can be requested on the occurrence of the respective input capture or compare-match.

In addition, channel 2 has a 16-bit dedicated input capture register (OSBR2). The channel 0 TI0A input pin can also be used as the OSBR2 trigger input, enabling use of a twin-capture function.

Channels 3 to 5: Channels 3 to 5 each have a 16-bit free-running counter (TCNT3 to TCNT5) and four 16-bit general registers (GR3A to GR3D, GR4A to GR4D, GR5A to GR5D). TCNT3 to TCNT5 are up-counters that perform free-running operation. Channels 3 to 5 each have a 16-bit free-running counter (TCNT3 to TCNT5) and four 16-bit general registers (GR3A to GR3D, GR4A to GR4D, GR5A to GR5D). TCNT3 to TCNT5 are up-counters that perform free-running operation. In addition, counter clearing can be performed by compare-match by making a setting in the timer I/O control register (TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B). Each counter can generate an interrupt request when it overflows.

The four general registers (GR3A to GR3D, GR4A to GR4D, GR5A to GR5D) each have corresponding external signal I/O pins (TIO3A to TIO3D, TIO4A to TIO4D, TIO5A to TIO5D), and can be used as input capture or output compare registers. When used for input capture, the free-running counter (TCNT3 to TCNT5) value is captured by means of input from the corresponding external signal I/O pin (TIO3A to TIO3D, TIO4A to TIO4D, TIO5A to TIO5D). Rising edge, falling edge, or both edges can be selected for the input capture signal in the timer I/O control registers (TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B). Also, in use for input capture, input capture can be performed using a compare-match between a channel 9 event counter (ECNT9A to ECNT9D), described later, and a general register (GR9A to GR9D) as the trigger (channel 3 only). In this case, capture is performed even if an input capture disable setting has been made for TIOR3A to TIOR3D. When used for output compare, compare-match with the free-running counter (TCNT3 to TCNT5) is performed. For the output from the external signal I/O pins by compare-match, 0 output, 1 output, or toggle output can be selected in the timer I/O control registers (TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B). An interrupt can be requested on the occurrence of the respective input capture or compare-match. However, in the case of input capture using channel 9 as a trigger, an interrupt request from channel 3 cannot be used.

By selecting PWM mode in the timer mode register (TMDR), PWM output can be obtained, with three outputs for each. In this case, GR3D, GR4D, and GR5D are automatically used as cycle registers, and GR3A to GR3C, GR4A to GR4C, GR5A to GR5C, as duty registers. TCNT3 to TCNT5 are cleared by the corresponding GR3D, GR4D, or GR5D compare-match.

CYLR7D), 16-bit duty registers (DTR6A to DTR6D, DTR7A to DTR7D), and buffer registers (BFR6A to BFR6D, BFR7A to BFR7D). Channels 6 and 7 also each have external output pins (TO6A to TO6D, TO7A to TO7D), and can be used as buffered PWM timers. The TCNT registers are up-counters, and 0 is output to the corresponding external output pin when the TCNT value matches the DTR value (when DTR ≠ CYLR). When the TCNT value matches the CYLR value (when DTR ≠ H'0000), 1 is output to the external output pin, TCNT is initialized to H'0001, and the BFR value is transferred to DTR. Thus, the configuration of channels 6 and 7 enables them to perform waveform output with the CYLR value as the cycle and the DTR value as the duty, and to use BFR to absorb the time lag between setting of data in DTR and compare-match occurrence.

When DTR = CYLR, 1 is output continuously to the external output pin, giving a duty of 100%. When DTR = H'0000, 0 is output continuously to the external output pin, giving a duty of 0%. Do not set a value in DTR that will result in the condition DTR > CYLR. When H'0000 is set to DTR, do not have DTR directly read H'0000. Set BFR to H'0000 and set H'0000 by forwarding from BFR to DTR. If H'0000 is directly set to DTR, duty may not be 0%.

In channel 6, TCNT can also be designated for complementary PWM output by means of the PWM mode register (PMDR). When the corresponding TSTR is set to 1, TCNT starts counting up, then switches to a down-count when the count matches the CYLR value. When TCNT reaches H'0000, it starts counting up again. When TCNT = DTR, the corresponding TO6A to TO6D output changes. Whether TCNT is counting up or down can be ascertained from the timer status register (TSR6).

DMAC activation and interrupt request generation, respectively, are possible when TCNT = CYLR in asynchronous PWM mode, and when TCNT = H'0000 in complementary PWM mode.

Channel 8: Channel 8 has sixteen 16-bit down-counters (DCNT8A to DCNT8P). The down-counters have corresponding external signal output pins, and can generate one-shot pulses. Setting a value in DCNT and setting the corresponding bit to 1 in the down-count start register (DSTR) starts DCNT operation and simultaneously outputs 1 to the external output pin. When DCNT counts down to H'0000, it stops and outputs 0 to the external output pin. An interrupt can be requested when DCNT underflows.

Down-counter operation can be coupled with the channel 1 or channel 2 output compare function by means of settings in the timer connection register (TCNR) and one-shot pulse terminate register (OTR), respectively, so that DCNT8I to DCNT8H count operations are started and stopped from channel 1, and DCNT8I to DCNT8P count operations from channel 2.

DCNT8I to DCNT8P have a reload register (RLDR), and a setting in the reload enable register (RLDEN) enables count operations to be started after reading the value from this register.

external input pin (ECNT9A to ECNT9F). The event counter value is incremented by input from the corresponding external input pin. Incrementing on the rising edge, falling edge, or both edges can be selected by means of settings in the timer control registers (TCR9A to TCR9C). An event counter is cleared by edge input after a match with the corresponding general register. An interrupt can requested when an event counter is cleared.

Timer control register (TCR9A, TCR9B) settings can be made to enable event counters ECNT9A to ECNT9D to send a compare-match signal to channel 3 when the count matches the corresponding general register (GR9A to GR9D), allowing input capture to be performed on channel 3. This enables the pulse input interval to be measured.

Channel 10: Channel 10 generates a multiplied clock based on external input, and supplies this to channels 1 to 5. Channel 10 is divided into three blocks: (1) an inter-edge measurement block, (2) a multiplied clock generation block, and (3) a multiplied clock correction block.

(1) Inter-edge measurement block

This block has a 32-bit free-running counter (TCNT10A), 32-bit input capture register (ICR10A), 32-bit output compare register (OCR10A), 8-bit event counter (TCNT10B), 8-bit output compare register (OCR10B), 8-bit noise canceler counter (TCNT10H), and 8-bit noise canceler compare-match register (NCR10).

The 32-bit free-running counter (TCNT10A) is an up-counter that performs free-running operations. When input capture is performed by means of TI10 input, this counter is cleared to H'00000001. When free-running counter (TCNT10A) reaches the value set in the output compare register (OCR10A), a compare-match interrupt can be requested.

The input capture register (ICR10A) has an external signal input pin (TI10), and the free-running counter (TCNT10A) value can be captured by means of input from TI10. Rising edge, falling edge, or both edges can be selected by making a setting in bits CKEG1 and CKEG0 in the timer control register (TCR10). The TI10 input has a noise canceler function, which can be enabled by setting the NCE bit in the timer control register (TCR10). When the counter value is captured, TCNT10A is cleared to 0 and an interrupt can be requested. The captured value can be transferred to the multiplied clock generation block reload register (RLD10C).

The 8-bit event counter (TCNT10B) is an up-counter that is incremented by TI10 input. When the event counter (TCNT10B) value reaches the value set in the output compare register (OCR10B), a compare-match interrupt can be requested. By setting the TRG0DEN bit in the timer control register (TCR10), a capture request can also be issued for the channel 0 input capture register 0D (ICR0D) when compare-match occurs.

The 16-bit noise canceler counter (TCNT10H) and 16-bit noise canceler compare-match register (NCR10) are used to set the period for which the noise canceler functions. By setting a

When the noise canceler counter (TCNT10H) value matches the noise canceler compare-match register (NCR10) value, the noise canceler counter (TCNT10H) is cleared to H'0000 and TI10 input masking is cleared.

(2) Multiplied clock generation block

This block has 16-bit reload counters (TCNT10C, RLD10C), a 16-bit register free-running counter (TCNT10G), and a 16-bit general register (GR10G).

16-bit reload counter 10C (RLD10C) is captured by 32-bit input capture register 10A (ICR10A), and when RLDEN in the timer I/O control register (TIOR10) is 0, the value captured in input capture register 10A is transferred to the multiplied clock generation block reload register (RLD10C). The value transferred can be selected from 1/32, 1/64, 1/128, or 1/256 the original value, according to the setting of bits PIM1 and PIM0 in TIOR10.

16-bit reload counter 10C (TCNT10C) performs down-count operations. When TCNT10C reaches H'0001, the value is read automatically from the reload buffer (RLD10C), internal clock AGCK1 is generated, and the down-count operation is repeated. Internally generated AGCK1 is input as a clock to the multiplied clock correction block 16-bit correction counter (TCNT10E) and 16-bit free-running counter 10G (TCNT10G).

16-bit register free-running counter 10G (TCNT10G) counts on AGCK1 generated by TCNT10C. It is initialized to H'0000 by external input from TI10.

The 16-bit general register (GR10G) can be used in a compare-match with free-running counter 10G (TCNT10G) by setting bits IO10G2 to IO10G0 in the timer I/O control register (TIOR10). An interrupt can be requested when a compare-match occurs. Also, by setting timer interrupt enable register 10 (TIER10), an interrupt can be request in the event of TI10 input after a compare-match.

(3) Multiplied clock correction block

This block has three 16-bit correction counters (TCNT10D, TCNT10E, TCNT10F) and a 16-bit correction counter clear register (TCCLR10). When 32-bit input capture register 10A (ICR10A) performs a capture operation due to input from external input pin TI10, the value in correction counter 10D (TCNT10D) is transferred to TCNT10E and TCNT10D is incremented. The value transferred to TCNT10E is 32, 64, 128, or 256 times the TCNT10D value, according to the setting of bits PIM1 and PIM0 in the timer I/O control register (TIOR10).

CCS bit in the timer I/O control register (TIOR10), it is possible to stop free-running counter 10E (TCNT10E) when the free-running counter 10D (TCNT10D) multiplication value specified by PIM1 and PIM0 and the free-running counter 10E (TCNT10E) value match. The multiplied TCNT10D value is transferred when input capture register 10A (ICR10A) performs a capture operation due to TI10 input.

16-bit correction counter 10F (TCNT10F) has Pϕ as its input and is constantly compared with 16-bit correction counter 10E (TCNT10E). When the 16-bit correction counter 10F (TCNT10F) value is smaller than that in 16-bit correction counter 10E (TCNT10E), it is incremented and generates count-up AGCKM. When the 16-bit correction counter 10F (TCNT10F) value exceeds that in 16-bit correction counter 10E (TCNT10E) (for example, when TCNT10F reloads TCNT10D), no count-up operation is performed. The TI10 multiplied signal (AGCKM) generated when TCNT10F is incremented is output to the channel 1 to 5 free-running counters (TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, TCNT5), and an up-count can be performed on AGCKM by setting this as the counter clock on each channel. TCNT10F is constantly compared with the 16-bit correction counter clear register (TCCLR10), and when the free-running counter 10F (TCNT10F) and correction counter clear register (TCCLR10) values match, the TCNT10F up-count stops. Setting TRG1AEN, TRG1BEN, TRG2AEN, and TRG2BEN in the timer control register (TCR10) enables the channel 1 and 2 free-running counters (TCNT1A, TCNT1B, TCNT2A, TCNT2B) to be cleared at this time. If TI10 is input when TCNT10D = H'0000, initialization and correction operations are performed. When TCNT10F = TCCLR10, TCNT10F is cleared to H'0001. When TCNT10F ≠ TCCLR10, TCNT10F automatically counts up to the TCCLR10 value, and is cleared to H'0001.

Channel 11: Channel 11 has a 16-bit free-running counter (TCNT11) and two 16-bit general registers (GR11A and GR11B). TCNT11 is an up-counter that performs free-running operation. The counter can generate an interrupt request when it overflows. The two general registers (GR11A and GR11B) each have a corresponding external signal I/O pin (TIO11A, TIO11B), and can be used as input capture or output compare registers.

When used for input capture, the free-running counter (TCNT11) value is captured by means of input from the corresponding external signal I/O pin (TIO11A, TIO11B). Rising edge, falling edge, or both edges can be selected for the input capture signal in the timer I/O control register (TIOR11). When used for output compare, compare-match with the free-running counter (TCNT11) is performed. For the output from the external signal I/O pins by compare-match, 0 output, 1 output, or toggle output can be selected in the timer I/O control register (TIOR11). An interrupt can be requested on the occurrence of the respective input capture or compare-match. When the two general registers (GR11A and GR11B) are designated for compare-match use, a compare-match signal can be output to the APC.

respect to clock Pφ. The second prescaler stage allows selection of a clock obtained by further scaling the clock from the first stage by 2^n (where n = 0 to 5) according to the timer control registers for the respective channels (TCR1A, TCR1B, TCR2A, TCR2B, TCR3 to TCR5, TCR6A, TCR6B, TCR7A, TCR7B, TCR8, TCR11).

The prescalers of channels 1 to 8 and 11 have a 2-stage configuration, while the channel 0 and 10 prescalers only have a first stage. The first-stage prescaler is common to channels 0 to 5, 8, and 11, and it is not possible to set different first-stage division ratios for each. Channels 6, 7, and 10 each have a first-stage prescaler, and different first-stage division ratios can be set for each.

11.3.2 Free-Running Counter Operation and Cyclic Counter Operation

The free-running counters (TCNT) in ATU-II channels 0 to 5 and 11 start counting up as free-running counters when the corresponding timer start register (TSTR) bit is set to 1. When TCNT overflows (channel 0: from H'FFFFFFFF to H'00000000; channels 1 to 5 and 11: from H'FFFF to H'0000), the OVF bit in the timer status register (TSR) is set to 1. If the OVE bit in the corresponding timer interrupt enable register (TIER) is set to 1 at this time, an interrupt request is sent to the CPU. After overflowing, TCNT starts counting up again from H'00000000 or H'0000.

If the TSTR value is cleared to 0 during TCNT operation, the corresponding TCNT halts. In this case, TCNT is not reset. If external output is being performed from the GR for the corresponding TCNT, the output value does not change.

Channel 0 free-running counter operation is shown in figure 11.13.

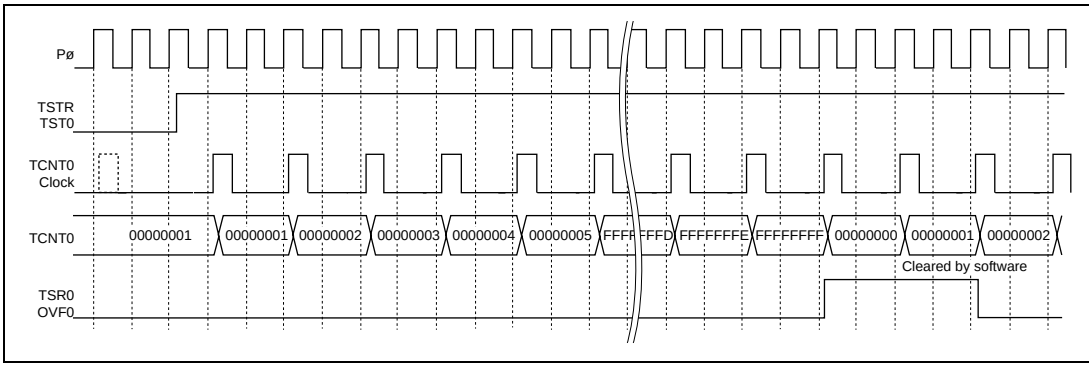


Figure 11.13 Free-Running Counter Operation and Overflow Timing

The free-running counters (TCNT) in ATU-II channels 6 and 7 perform cyclic count operations unconditionally. With channel 3 to 5 free-running counters (TCNT), when the corresponding T3PWM to T5PWM bit in the timer mode register (TMDR) is set to 1, or the corresponding CCI bit in the timer I/O control register (TIOR) is set to 1 when bits T3PWM to T5PWM are 0, the

corresponding STR bit in TSTR is set to 1 after the TMDR setting is made. When the count value matches the GR3D, GR4D, GR5D, or CYLR value, the corresponding IMF3D, IMF4D, or IMF5D bit in the timer status register (TSR) (or the CMF bit in TSR6 or TSR7 for channels 6 and 7) is set to 1, and TCNT is cleared to H'0000 (H'0001 in channels 6 and 7).

If the corresponding TIER bit is set to 1 at this time, an interrupt request is sent to the CPU. After the compare-match, TCNT starts counting up again from H'0000 (H'0001 in channels 6 and 7).

Figure 11.14 shows the operation when channel 3 is used as a cyclic counter (with a cycle setting of H'0008).

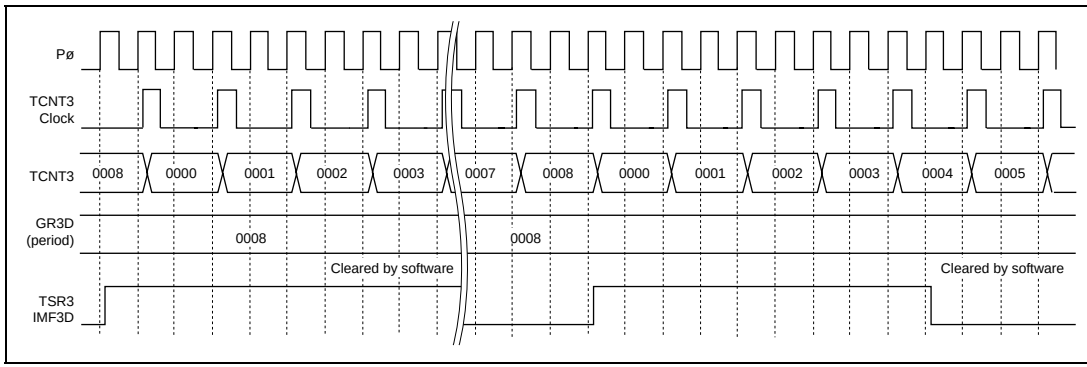


Figure 11.14 Example of Cyclic Counter Operation

11.3.3 Compare-Match Function

Designating general registers in channels 1 to 5 and 11 (GR1A to GR1H, GR2A to GR2H, GR3A to GR3D, GR4A to GR4D, GR5A to GR5D, GR11A, GR11B) for compare-match operation in the timer I/O control registers (TIOR1 to TIOR5, TIOR11) enables compare-match output to be performed at the corresponding external pins (TIO1A to TIO1H, TIO2A to TIO2H, TIO3A to TIO3D, TIO4A to TIO4D, TIO5A to TIO5D, TIO11A, TIO11B).

A free-running counter (TCNT) starts counting up when 1 is set in the timer status register (TSTR). When the desired number is set beforehand in GR, and the TCNT value matches the GR value, the timer status register (TSR) bit corresponding to GR is set and a waveform is output from the corresponding external pin.

1 output, 0 output, or toggle output can be selected by means of a setting in TIOR. If the appropriate interrupt enable register (TIER) setting is made, an interrupt request will be sent to the CPU when a compare-match occurs.

pin as a general I/O pin and select 1 output, 0 output, or toggle output on compare-match in TIOR.

Channel 1 and 2 compare-match registers (OCR1, OCR2A to OCR2H) perform compare-match operations unconditionally. However, there are no corresponding output pins. If the appropriate TIER setting is made, an interrupt request will be sent to the CPU when a compare-match occurs.

Channel 1 and 2 GR and OCR registers can send a trigger/terminate signal to channel 8 when a compare-match occurs. In this case, settings should be made in the trigger mode register (TRGMDR), timer connection register (TCNR), and one-shot pulse terminate register (OTR).

An example of compare-match operation is shown in figure 11.15.

In the example in figure 11.15, channel 1 is activated, and external output is performed with toggle output specified for GR1A, 1 output for GR1B, and 0 output for GR1C.

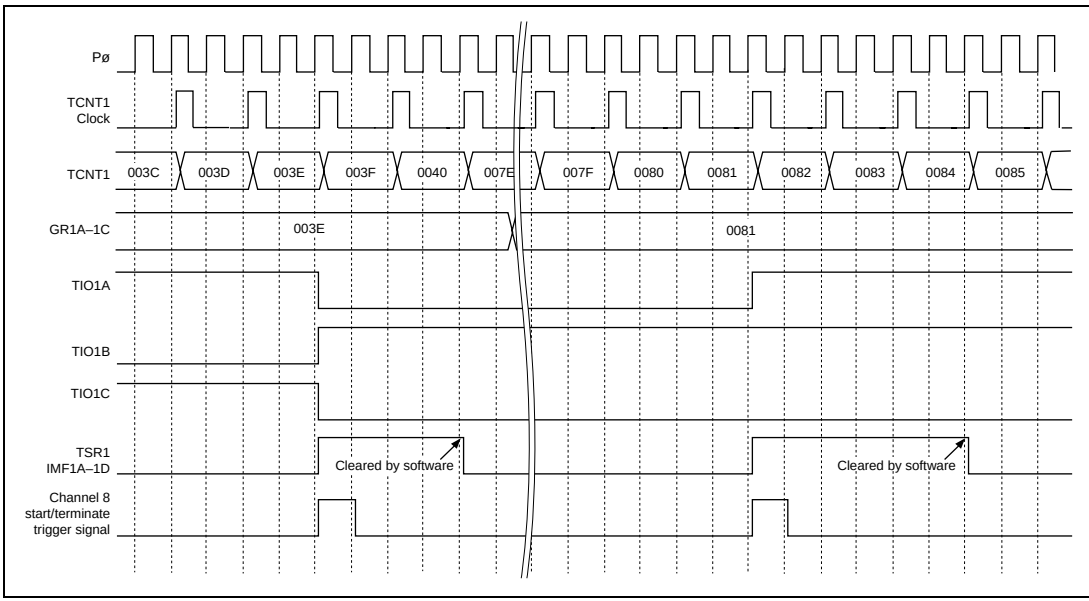


Figure 11.15 Compare-Match Operation

11.3.4 Input Capture Function

If input capture registers (ICR0A to ICR0D) and general registers (GR1A to GR1H, GR2A to GR2H, GR3A to GR3D, GR4A to GR4D, GR5A to GR5D, GR11A, GR11B) in channels 1 to 5 and 11 are designated for input capture operation in the timer I/O control registers (TIOR0 to TIOR5, TIOR11), input capture is performed when an edge is input at the corresponding external

A free-running counter (TCNT) starts counting up when a setting is made in the timer start register (TSTR). When an edge is input at an external pin corresponding to ICR or GR, the corresponding timer status register (TSR) bit is set and the TCNT value is transferred to ICR or GR. Rising-edge, falling-edge, or both-edge detection can be selected. By making the appropriate setting in the interrupt enable register (TIER), an interrupt request can be sent to the CPU.

An example of input capture operation is shown in figure 11.16.

In the example in figure 11.16, channel 1 is activated, and input capture operation is performed with both-edge detection specified for TIO1A, rising-edge detection for TIO1B, and falling-edge detection for TIO1C.

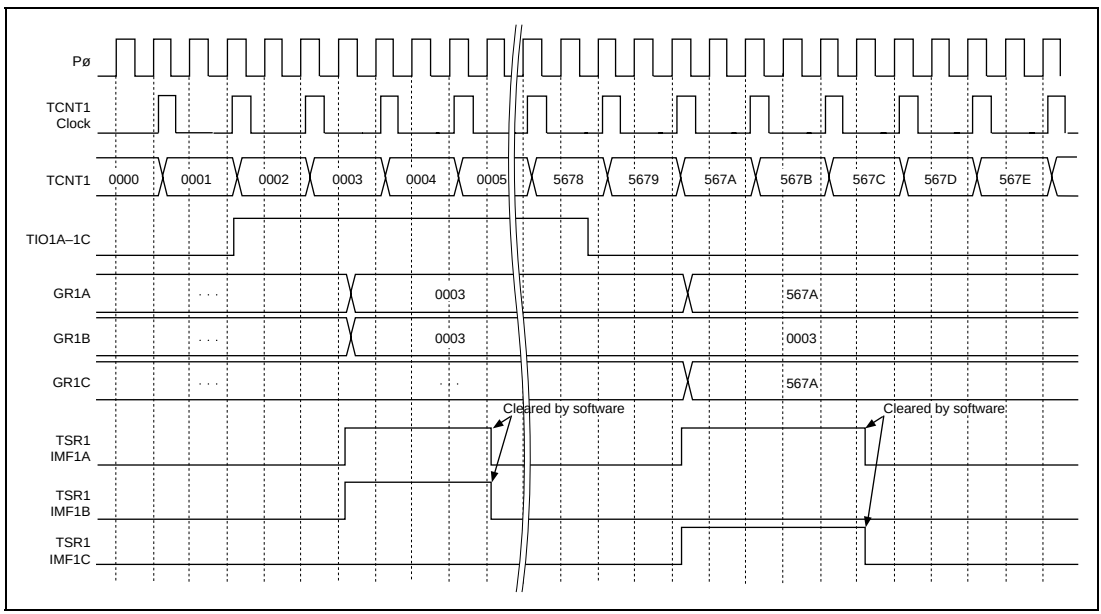


Figure 11.16 Input Capture Operation

11.3.5 One-Shot Pulse Function

Channel 8 has sixteen down-counters (DCNT8A to DCNT8P) and corresponding external pins (TO8A to TO8P) which can be used as one-shot pulse output pins.

When a value is set beforehand in DCNT and the corresponding bit in the down-counter start register (DSTR) is set, DCNT starts counting down, and at the same time 1 is output from the corresponding external pin. When DCNT reaches H'0000 the down-count stops, the corresponding bit in the timer status register (TSR) is set, and 0 is output from the external pin. The

An example of one-shot pulse operation is shown in figure 11.17.

In the example in figure 11.17, H'0005 is set in DCNT and a down-count is started.

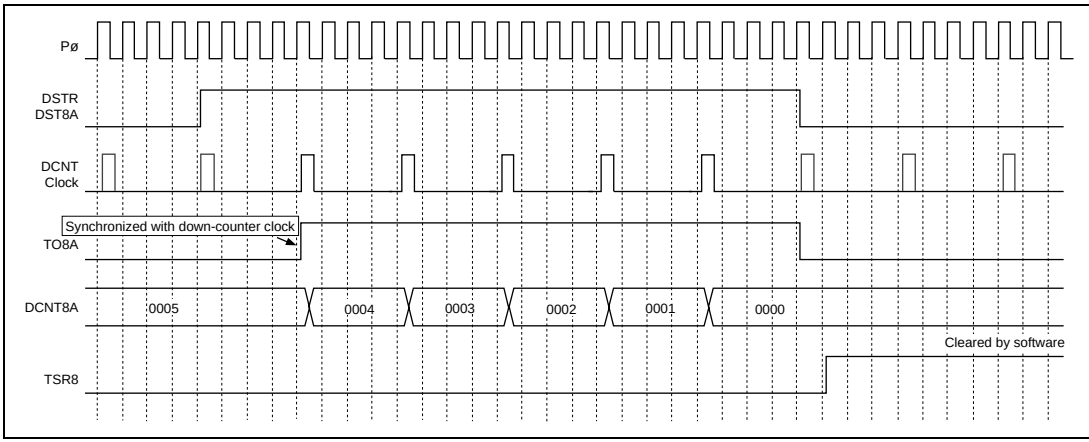


Figure 11.17 One-Shot Pulse Output Operation

11.3.6 Offset One-Shot Pulse Function and Output Cutoff Function

By making an appropriate setting in the timer connection register (TCNR), down-counting by channel 8 down-counters (DCNT8A to DCNT8P) can be started using compare-match signals from channel 1 general registers (GR1A to GR1H) or channel 1 and 2 compare-match registers (OCR1, OCR2A to OCR2H). DCNT8A to DCNT8H are connected to channel 1 OCR1 or GR1A to GR1H, and DCNT8I to DCNT8P are connected to channel 2 OCR2A to OCR2H or GR2A to GR2H. This enables one-shot pulse output from the external pin (TO8A to TO8P) corresponding to DCNT. The down-count can be forcibly stopped by making a setting in the one-shot pulse terminate register (OTR). On channel 1, down-count start or termination by a GR or OCR compare-match can be selected with the trigger mode register (TRGMDR).

Making a setting in the timer start register (TSTR) starts an up-count by a free-running counter (TCNT) in channel 1 or 2. When TCNT matches GR or OCR while connection is enabled by TCNR, the corresponding DSTR is automatically set and DCNT starts counting down. At the same time, 1 is output from the corresponding external pin (TO8A to TO8P). By making the appropriate setting in the interrupt enable register (TIER), an interrupt request can be sent to the CPU.

When TCNT1 matches GR or OCR, or TCNT2 matches GR, while channel 8 one-shot pulse termination by a channel 1 or 2 compare-match signal is enabled by OTR, the corresponding

DCNT8I to DCNT8P are connected to the reload register (RLDR8), and when the DSTR corresponding to DCNT8I to DCNT8P is set, the DCNT8I to DCNT8P counter loads RLDR8 before starting the down-count.

An example of the offset one-shot pulse output function and output cutoff function is shown in figure 11.18.

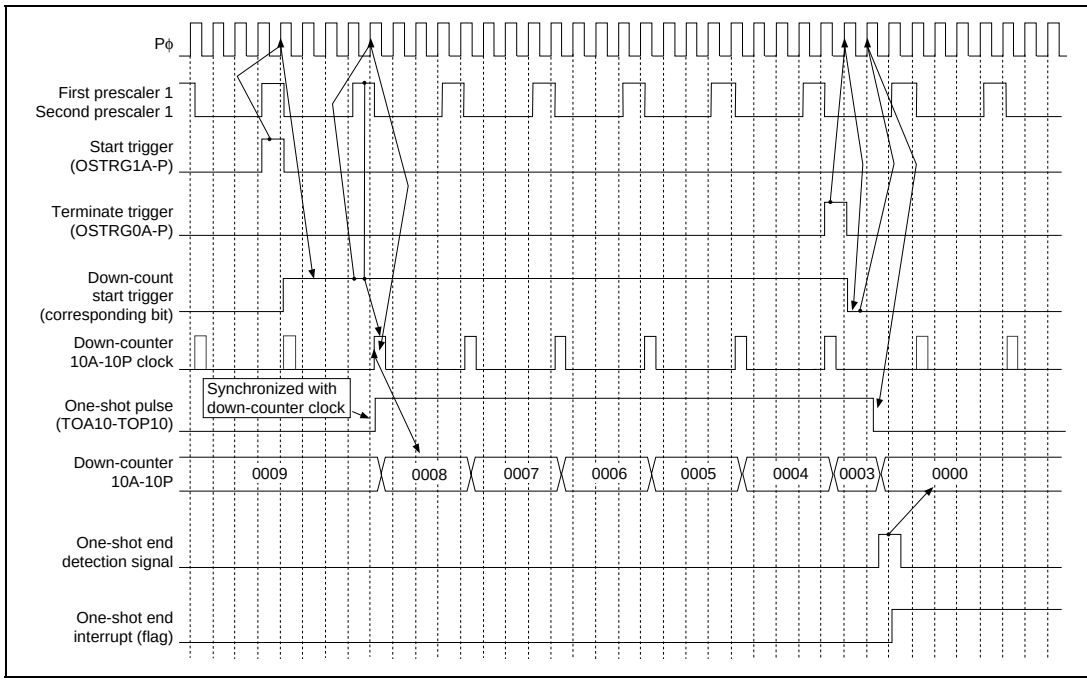


Figure 11.18 Offset One-Shot Pulse Output Function and Output Cutoff Function Operation

11.3.7 Interval Timer Operation

The interval interrupt request registers (ITVRR1, ITVRR2A, ITVRR2B) are connected to bits 6 to 9 and 10 to 13 of the channel 0 free-running counter (TCNT0). The ITVRR registers are 8-bit registers; the upper 4 bits (ITVA) are used for A/D converter activation, and the lower 4 bits (ITVE) are used for interrupt requests. ITVRR1 is connected to A/D converter 2 (AD2), ITVRR2A to A/D converter 0 (AD0), and ITVRR2B to A/D converter 1 (AD1).

When the ITVA bit for the desired timing is set, the A/D converter is activated when the corresponding bit of TCNT0 changes to 1.

register (TSR0) is set. There are four interrupt sources for the respective ITVRR registers, but there is only one interrupt vector.

To suppress interrupts and A/D converter activation, ITVRR bits should be cleared to 0.

An example of interval timer function operation is shown in figure 11.19.

In the example in figure 11.19, TCNT0 is started by setting ITVE to 1 in ITVRR1.

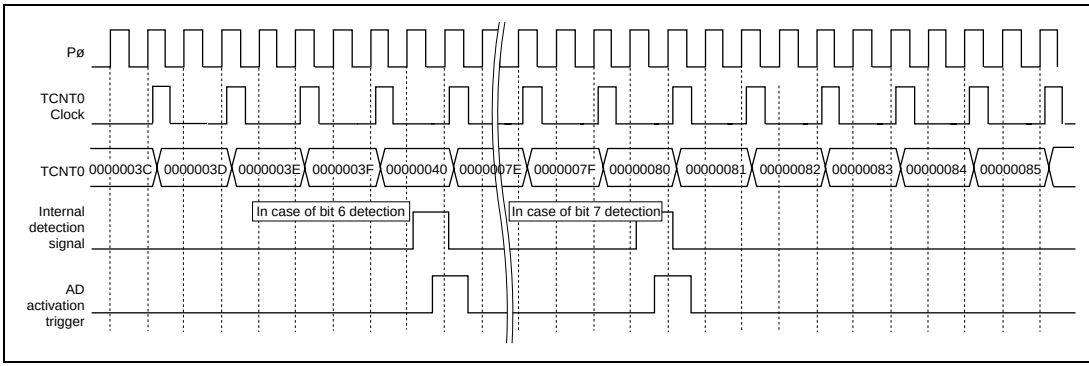


Figure 11.19 Interval Timer Function

11.3.8 Twin-Capture Function

Channel 0 input capture register ICR0A, channel 1 offset base register 1 (OSBR1), and channel 2 offset base register 2 (OSBR2) can be made to perform input capture in response to the same trigger by means of a setting in timer I/O control register 0 (TIOR0).

When TCNT0, TCNT1A, and TCNT2A in channel 0, channel 1, and channel 2 are started by a setting in the timer status register (TSR), and an edge detection is carried out by the ICR0A input as a trigger signal, the TCNT1A value is transferred to OSBR1, and the TCNT2A value to OSBR2. Edge detection is as described in section 11.3.4, Input Capture Function.

An example of twin-capture operation is shown in figure 11.20.

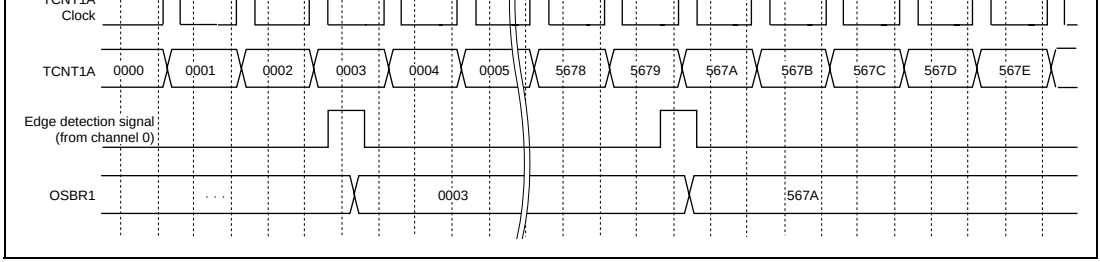


Figure 11.20 Twin-Capture Operation

11.3.9 PWM Timer Function

Channels 6 and 7 can be used unconditionally as PWM timers using external pins (TO6A to TO6D, TO7A to TO7D).

In channels 6 and 7, when the corresponding bit is set in the timer start register (TSTR) and the free-running counter (TCNT) is started, the counter counts up until its value matches the corresponding cycle register (CYLR). When TCNT matches CYLR, it is cleared to H'0001 and starts counting up again from that value. At this time, 1 is output from the corresponding external pin. An interrupt request can be sent to the CPU by setting the corresponding bit in the timer interrupt enable register (TIER). If a value has been set in the duty register (DTR), when TCNT matches DTR, 0 is output to the corresponding external pin. If the DTR value is H'0000, the output does not change (0% duty). However, when H'0000 is set to DTR, do not directly write H'0000 to DTR. Set H'0000 to BFR and forward it from BFR to DTR. If H'0000 is directly set to DTR, duty may not be 0%. A duty of 100% is specified by setting DTR = CYLR. Do not set a value in DTR that will result in the condition DTR > CYLR.

Channels 6 and 7 have buffers (BFR); the BFR value is transferred to DTR when TCNT matches CYLR. The duty value written into BFR is reflected in the output value in the cycle following that in which BFR is written to.

An example of PWM timer operation is shown in figure 11.21.

In the example in figure 11.21, H'0004 is set in channel 6 CYLR6A, and H'0002, H'0000 (0%), H'0004 (100%), and H'0001 in BFR6A.

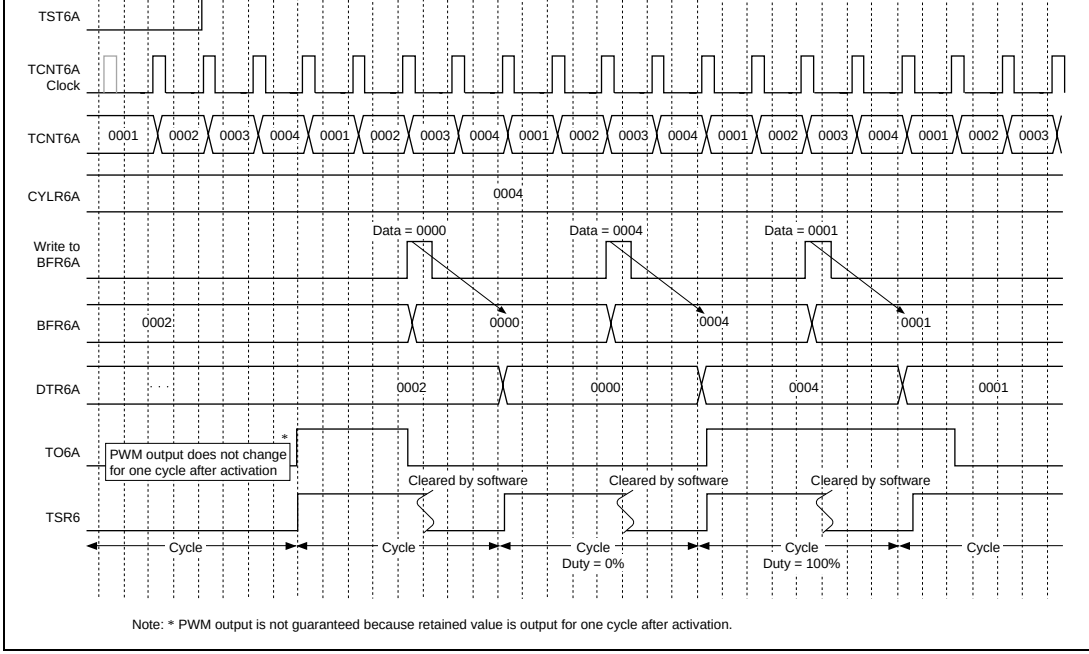


Figure 11.21 PWM Timer Operation

Channel 6 can be used in complementary PWM mode by making a setting in the PWM mode control register (PMDR). On-duty or off-duty can also be selected with a setting in PMDR.

When TCNT6 is started by a setting in TSTR, it starts counting up. When TCNT6 reaches the CYLR6 value, it starts counting down, and on reaching H'000, starts counting up again. The counter status is shown by TSR6. When TCNT6 underflows, an interrupt request can be sent to the CPU by setting the corresponding bit in TIER. When TCNT6 matches the duty register (DTR6) value, the output is inverted. The output prior to the match depends on the PMDR setting. When a value including dead time is set in DTR6, a maximum of 4-phase PWM output is possible. Data transfer from BFR6 to DTR6 is performed when TCNT6 underflows.

An example of channel 6 complementary PWM mode operation is shown in figure 11.22.

In the example in figure 11.22, H'0004 is set in channel 6 CYLR6A, and H'0002, H'0003, H'0004 (100%), and H'0000 (0%) in BFR6A.

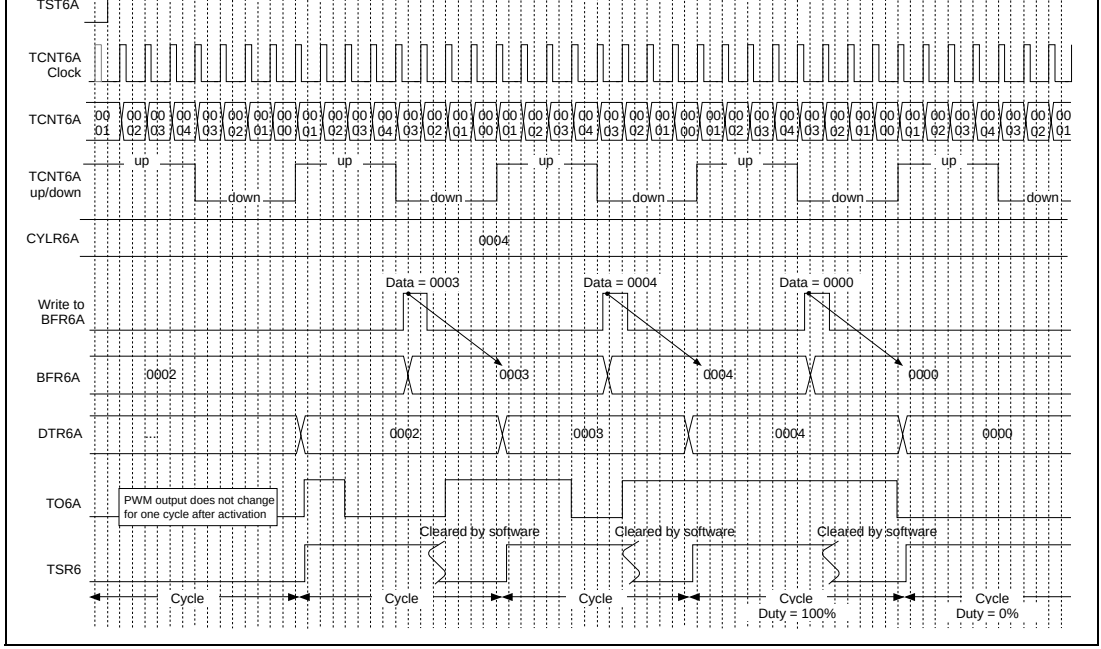


Figure 11.22 Complementary PWM Mode Operation

11.3.10 Channel 3 to 5 PWM Function

PWM mode is selected for channels 3 to 5 by setting the corresponding bits to 1 in the timer mode register (TMDR), enabling the channels to operate as PWM timers with the same cycle.

In PWM mode, general registers D (GR3D, GR4D, GR5D) are used as cycle registers, and general registers A to C (GR3A to GR3C, GR4A to GR4C, GR5A to GR5C) as duty registers. The external pins (TIO3A to TIO3C, TIO4A to TIO4C, TIO5A to TIO5C) corresponding to the GRs used as duty registers are used as PWM outputs. External pins TIO3D, TIO4D, and TIO5D should not be used as timer outputs.

The free-running counter (TCNT) is started by making a setting in the timer start register (TSTR), and when TCNT reaches the cycle register (GR3D, GR4D, GR5D) value, a compare-match is generated and TCNT starts counting up again from H'0000. At the same time, the corresponding bit is set in the timer status register (TSR) and 1 is output from the corresponding external pin. When TCNT reaches the duty register (GR3A to GR3C, GR4A to GR4C, GR5A to GR5C) value, 0 is output to the external pin. The corresponding status flag is not set. When PWM operation is performed by starting the free-running counter from its initial value of H'0000, PWM output is not performed for one cycle. To perform immediate PWM output, the value in the cycle register must be set in the free-running counter before the counter is started. If PWM operation is performed

Note that 0% or 100% duty output is not possible in channel 3 to 5 PWM mode.

An example of channel 3 to 5 PWM mode operation is shown in figure 11.23.

In the example in figure 11.23, H'0008 is set in GR3D, H'0002 is set in GR3A, GR3B, and GR3C, and channel 3 is activated; then, during operation, H'0000 is set in GR3A, GR3B, and GR3C, and output is performed to external pins TIOA3 to TIOC3. Note that 0% duty output is not possible even though H'0000 is set.

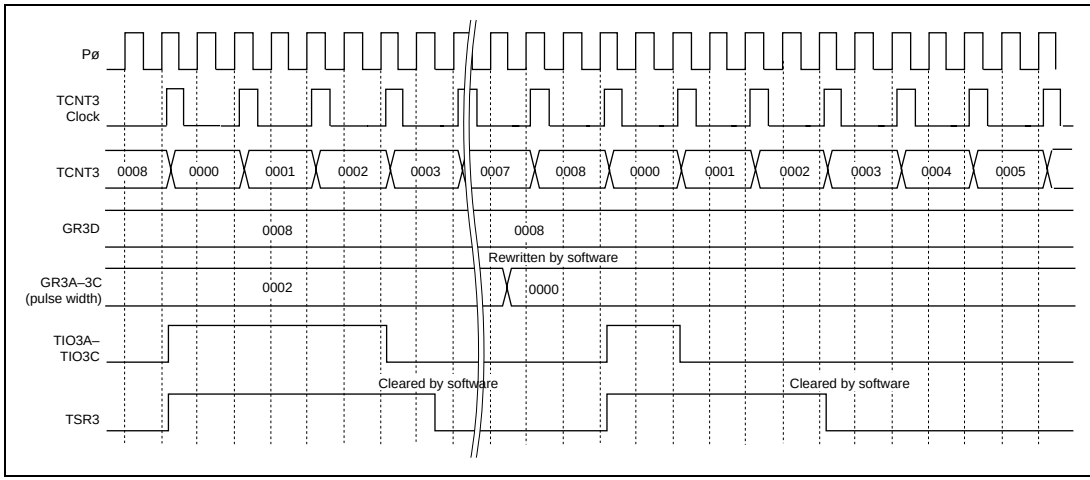


Figure 11.23 Channel 3 to 5 PWM Mode Operation

11.3.11 Event Count Function and Event Cycle Measurement

Channel 9 has six 8-bit event counters (ECNT9A to ECNT9F) and corresponding general registers (GR9A to GR9F). Each event counter has an external pin (TI9A to TI9F).

Each ECNT9 operates unconditionally as an event counter. When an edge is input from the external pin, ECNT9 is incremented. When ECNT9 matches the value set in GR9, it is cleared, and then counts up when an edge is again input at the external pin. By making the appropriate setting in the interrupt enable register (TIER) beforehand, an interrupt request can be sent to the CPU on compare-match.

For ECNT9A to ECNT9D, a trigger can be transmitted to channel 3 when a compare-match occurs. In channel 3, if the channel 9 trigger input is set in the timer I/O control register (TIOR) and the corresponding bit is set to 1 in the timer start register (TSTR), the TCNT3 value is captured in the corresponding general register (GR3A to GR3D) when an ECNT9A to ECNT9D compare-match occurs. This enables the event cycle to be measured.

match is generated.

An example of event cycle measurement operation is shown in figure 11.25. In this example, GR3A in channel 3 captures TCNT3 in response to a trigger from channel 9.

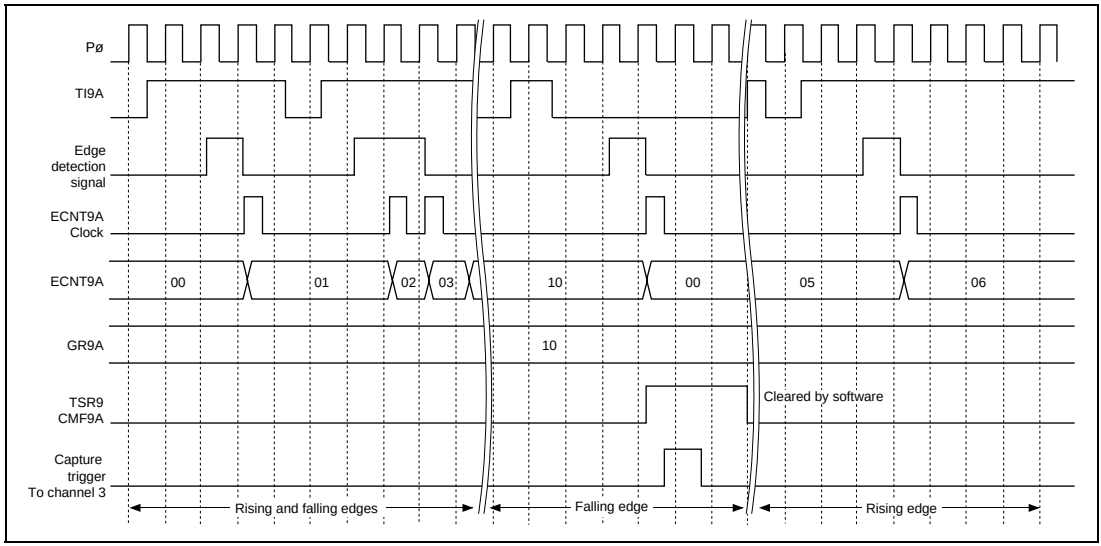


Figure 11.24 Event Count Operation

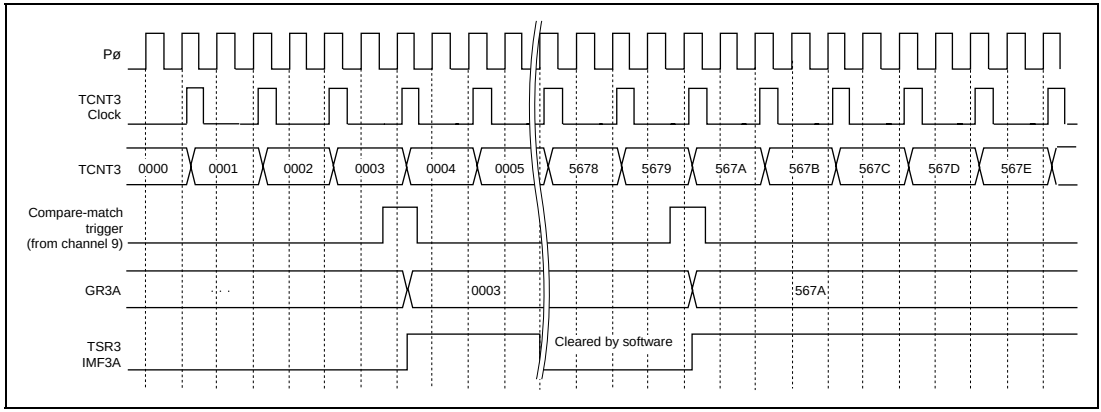


Figure 11.25 Event Cycle Measurement Operation

Inter-Edge Measurement Function and Edge Input Cessation Detection Function: 32-bit input capture register 10A (ICR10A) and 32-bit output compare register 10A (OCR10A) in channel 10 unconditionally perform input capture and compare-match operations, respectively. These registers are connected to 32-bit free-running counter TCNT10A.

When the corresponding bit is set in the timer start register (TSTR), the entire channel 10 starts operating. ICR10A has an external input pin (TI10), and when an edge is input at this input pin, ICR10A captures the TCNT10A value. At this time, TCNT10A is cleared to H'00000001. The captured value is transferred to the read register (RLD10C) in the multiplied clock generation block. By making the appropriate setting in the interrupt enable register (TIER), an interrupt request can be sent to the CPU. This allows inter-edge measurement to be carried out.

When TCNT10A reaches the value set in OCR10A, a compare-match interrupt can be requested. In this way it is possible to detect the cessation of edge input beyond the time set in OCR10A.

The input edge from TI10 is synchronized internally; the internal signal is AGCK. Noise cancellation is possible for edges input at TI10 using the timer 10H (TCNT10H) input cancellation function by setting the NCE bit in timer control register TCR10. When an edge is input at TI10, TCNT10H starts and input is disabled until it reaches compare-match register NCR10.

Edge input operation without noise cancellation is shown in figure 11.26, edge input operation with noise cancellation in figure 11.27, and TCNT10A capture operation and compare-match operation in figure 11.28.

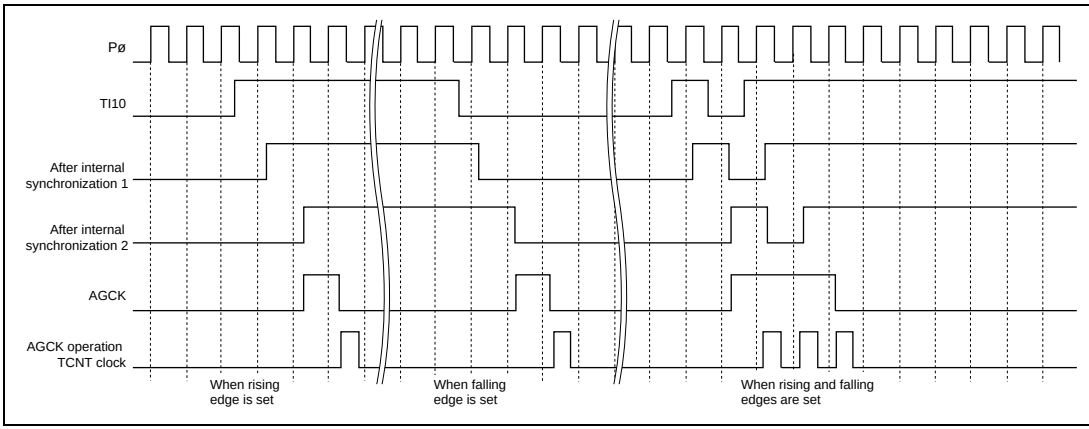


Figure 11.26 Edge Input Operation (Without Noise Cancellation)

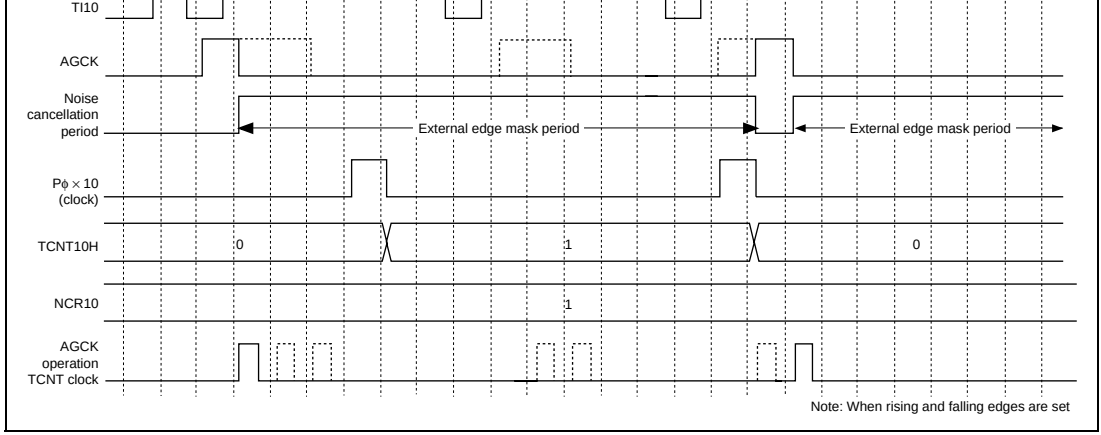


Figure 11.27 Edge Input Operation (With Noise Cancellation)

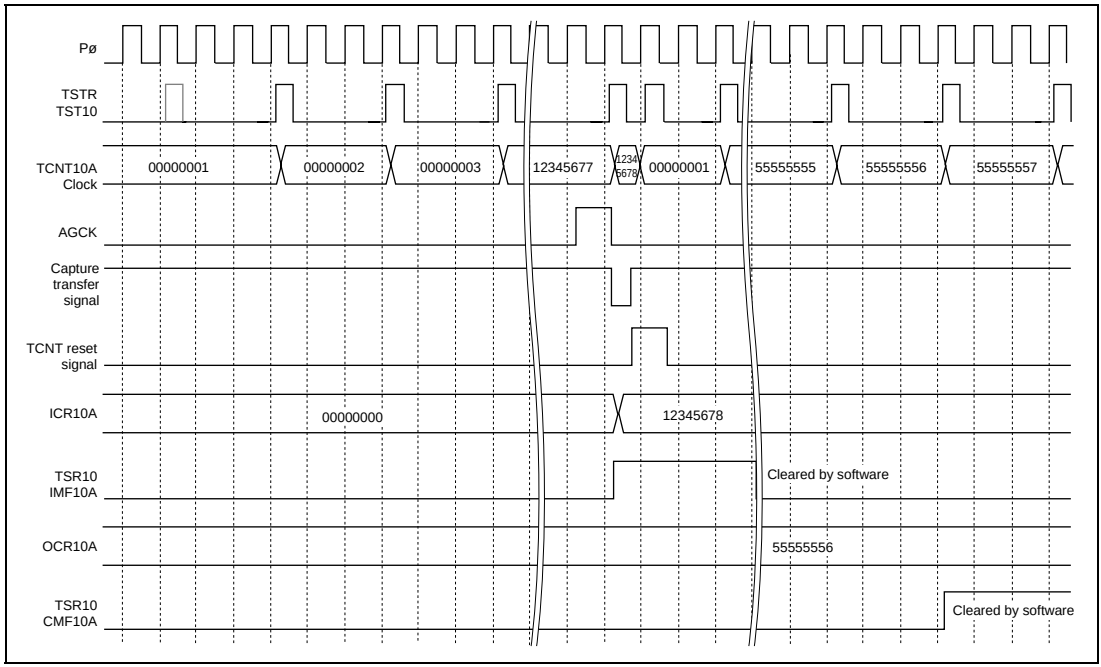


Figure 11.28 TCNT10A Capture Operation and Compare-Match Operation

Internally synchronized AGCK is counted by event count 10B (TCNT10B), and when TCNT10B reaches the value set beforehand in compare-match register 10B (OCR10B), a compare-match occurs, and the compare-match trigger signal is transmitted to channel 0. By setting the corresponding bit in TIER, an interrupt request can be sent to the CPU.

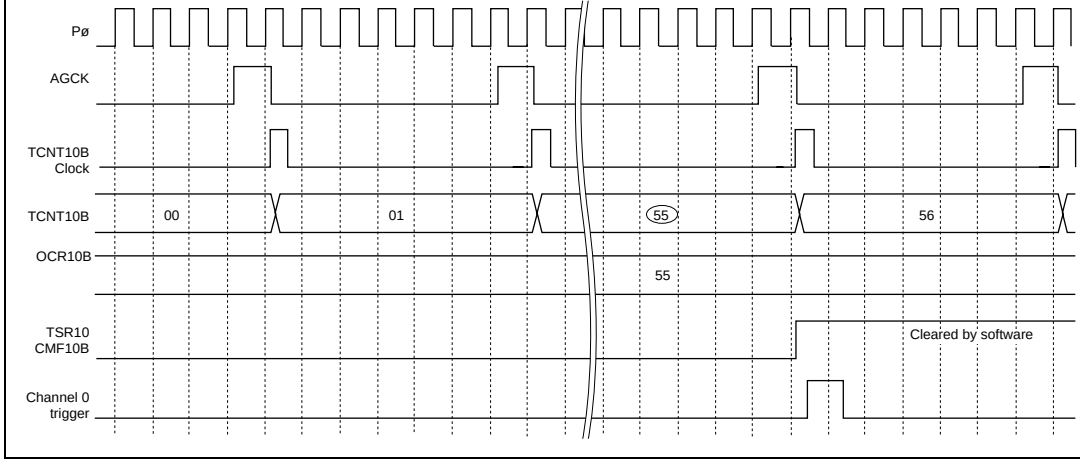


Figure 11.29 TCNT10B Compare-Match Operation

Multiplied Clock Generation Function: The channel 10 16-bit reload counter (TCNT10C, RLD10C) and 16-bit free-running counter 10G (TCNT10G) can be used to multiply the interval between edges input from external pin TI10 by 32, 64, 128, or 256.

The value captured in ICR10A above is multiplied by 1/32, 1/64, 1/128, or 1/256 according to the value set in the timer I/O control register (TIOR10), and transferred to the reload buffer (RLD10C). At the same time, the same value is transferred to 16-bit reload counter 10C (TCNT10C) and a down-count operation is started. When this counter reaches H'0001, the value is read automatically from RLD10C and the down-count operation is repeated. When this reload occurs, a multiplied clock signal (AGCK1) is generated. AGCK1 is converted to a corrected clock (AGCKM) by the multiplied clock correction function described in the following section.

Channel 10 can also perform compare-match operation by means of the multiplied clock (AGCK1) using general register 10G (GR10G) and 16-bit free-running counter 10G (TCNT10G). TCNT10G is incremented unconditionally by AGCK1. By making the appropriate setting in the interrupt enable register (TIER), an interrupt request can be sent to the CPU when TCNT10G and GR10G match. The timing of this interrupt can be selected with the IREG bit in TIER as either on occurrence of the compare-match or on input of the first TI10 edge after the compare-match.

TCNT10C operation is shown in figure 11.30, and TCNT10G compare-match operation in figure 11.31.

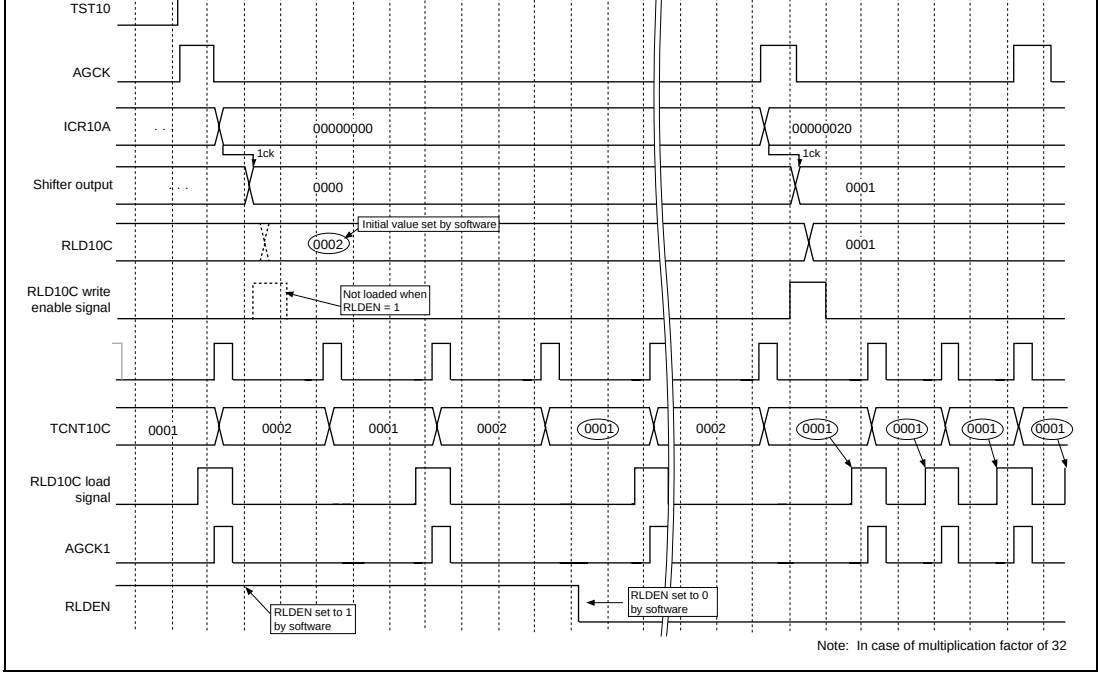


Figure 11.30 TCNT10C Operation

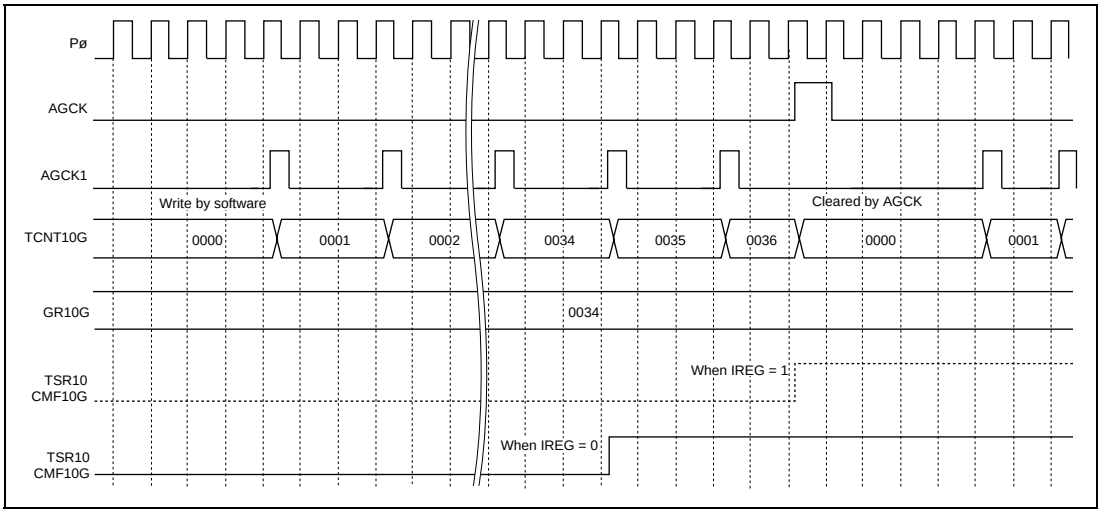


Figure 11.31 TCNT10G Compare-Match Operation

correction function that makes the interval between edges input from TI10 the frequency multiplication value set in TIOR10.

When AGCK is input, the value in TCNT10D multiplied by the multiplication factor set in TIOR10 is transferred to TCNT10E. At the same time, TCNT10D is incremented.

TCNT10E counts up on AGCK1. TCNT10E loads TCNT10D on AGCK, and counts up again on AGCK1. Using the counter correction select bit (CCS) in TIOR10, it is possible to select whether or not TCNT10E is halted when TCNT10D = TCNT10E.

TCNT10F has the peripheral clock (Pϕ) as its input and is constantly compared with TCNT10E. When the TCNT10F value is smaller than that in TCNT10E, TCNT10F is incremented and outputs a corrected multiplied clock signal (AGCKM).

When the TCNT10E value exceeds the TCNT10F value (when TCNT10E loads TCNT10D), no count-up operation is performed. AGCKM is output to the channel 1 to 5 free-running counters (TCNT1 to TCNT5).

Channel 10 also has a correction counter clear register (TCCLR10). The correction counters (TCNT10D, TCNT10E, TCNT10F) and channel 1 and 2 free-running counters (TCNT1 and TCNT2) can be cleared when TCNT10F reaches the value set in TCCLR10.

TCNT10D operation is shown in figure 11.32, TCNT10E operation in figure 11.33, TCNT10F operation (at startup) in figure 11.34, TCNT10F operation (end of cycle, acceleration, deceleration) in figure 11.35, and TCNT10F operation (end of cycle, steady-state) in figure 11.36.

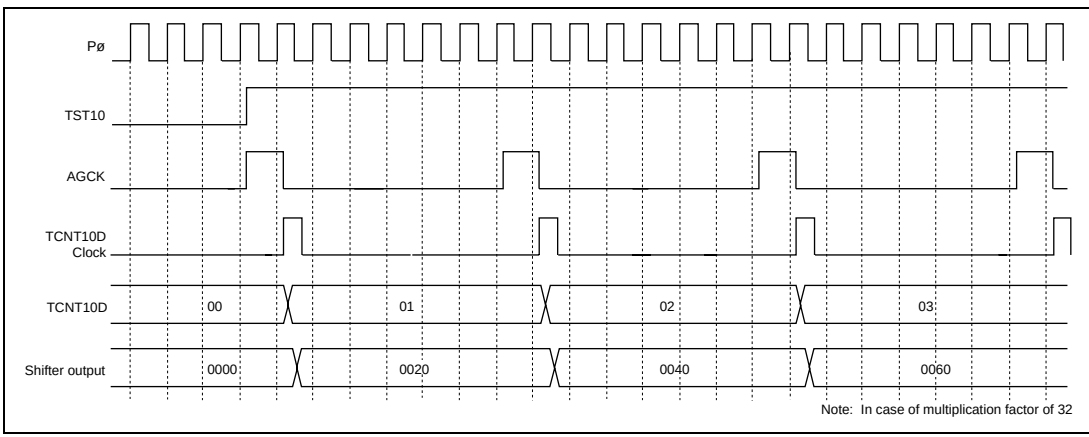


Figure 11.32 TCNT10D Operation

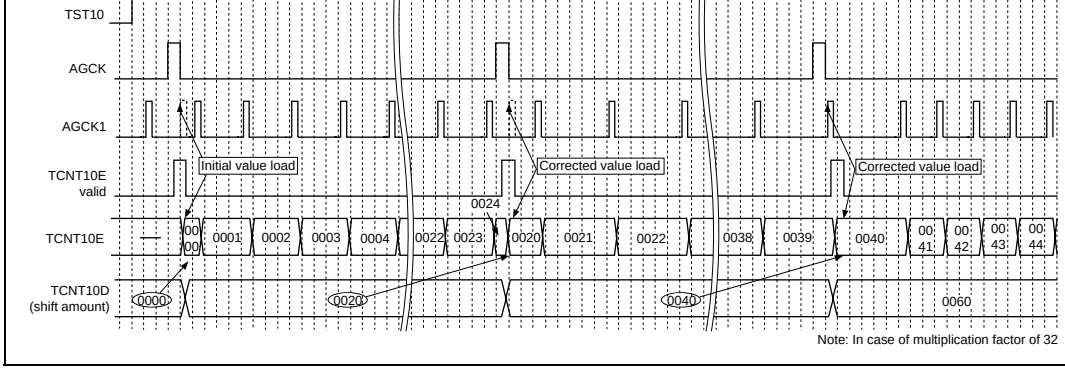


Figure 11.33 TCNT10E Operation

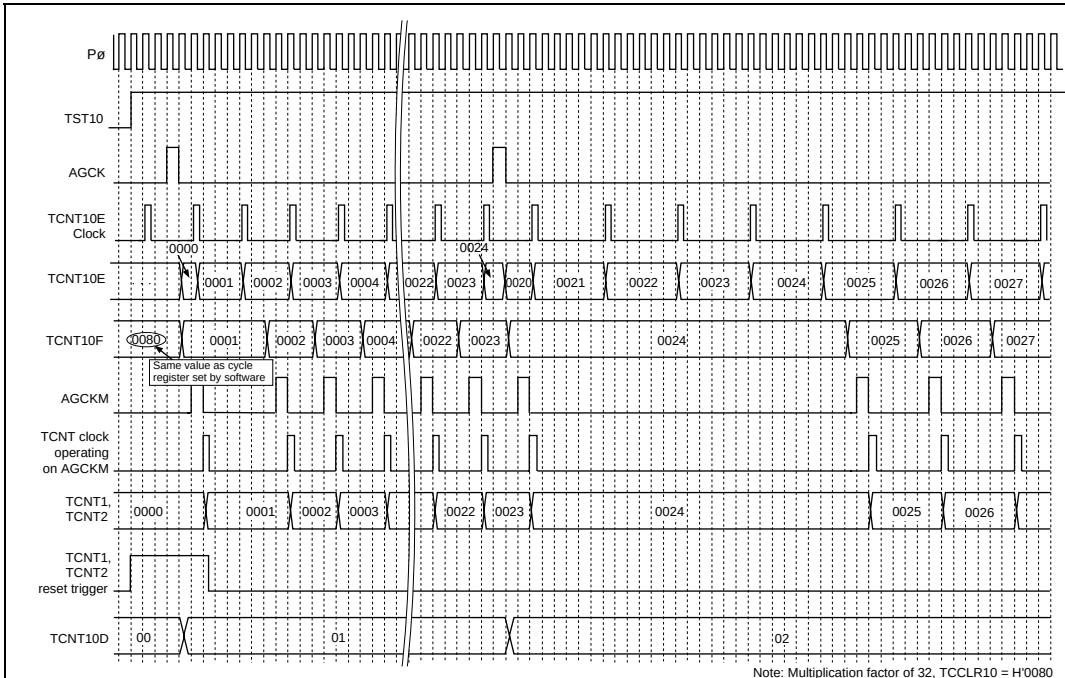


Figure 11.34 TCNT10F Operation (At Startup)

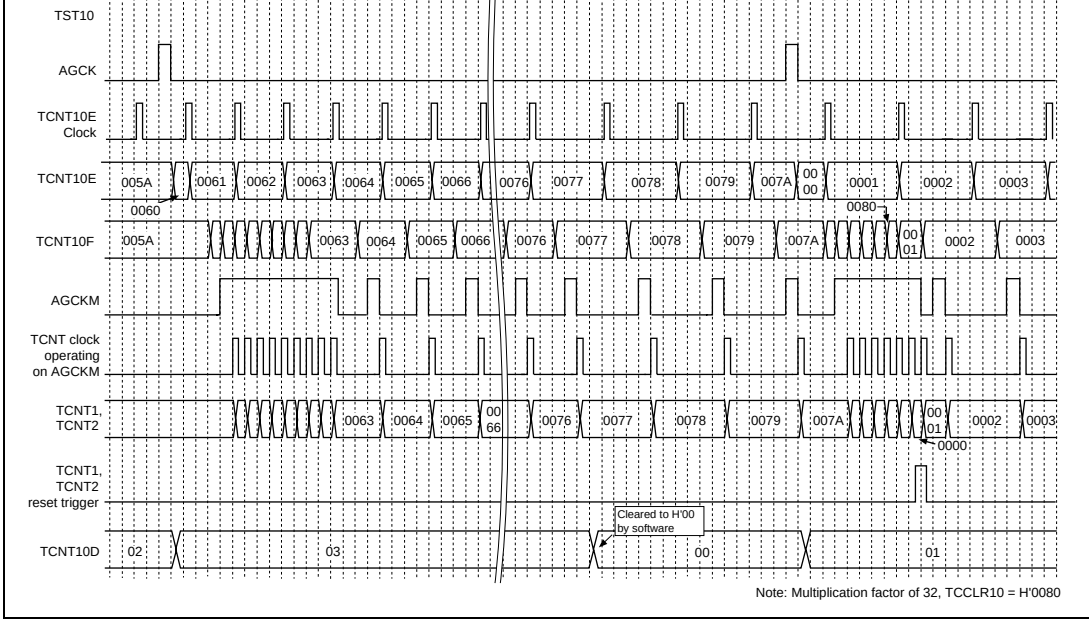


Figure 11.35 TCNT10F Operation (End of Cycle, Acceleration, Deceleration)

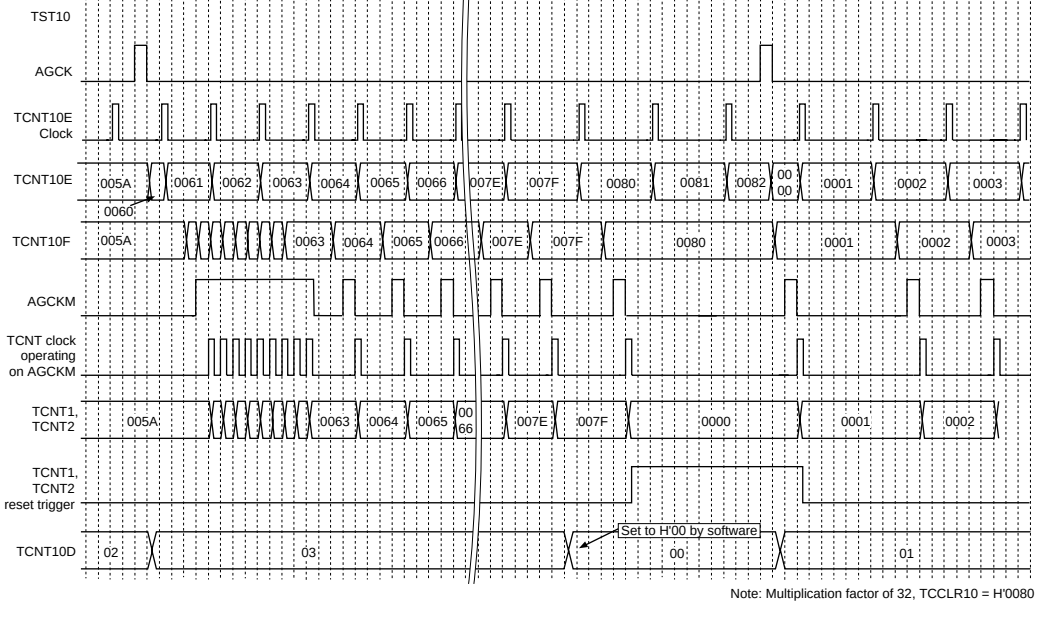


Figure 11.36 TCNT10F Operation (End of Cycle, Steady-State)

The ATU has 75 interrupt sources of five kinds: input capture interrupts, compare-match interrupts, overflow interrupts, underflow interrupts, and interval interrupts.

11.4.1 Status Flag Setting Timing

IMF (ICF) Setting Timing in Input Capture: When an input capture signal is generated, the IMF bit and ICF bit are set to 1 in the timer status register (TSR), and the TCNT value is simultaneously transferred to the corresponding GR, ICR, and OSBR.

The timing in this case is shown in figure 11.37.

In the example in figure 11.37, a signal is input from an external pin, and input capture is performed on detection of a rising edge.

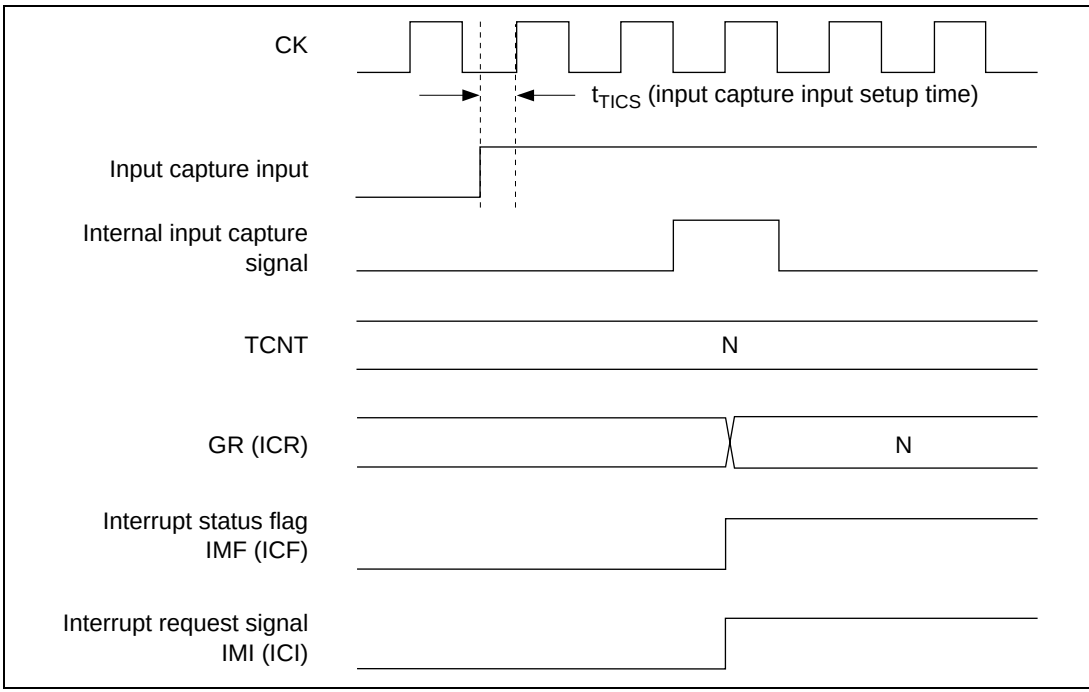


Figure 11.37 IMF (ICF) Setting Timing in Input Capture

output compare register (OCR), or cycle register (CYLR) value matches the timer counter (TCNT) value. The compare-match signal is generated in the last state of the match (when the matched TCNT count value is updated).

The timing in this case is shown in figure 11.38.

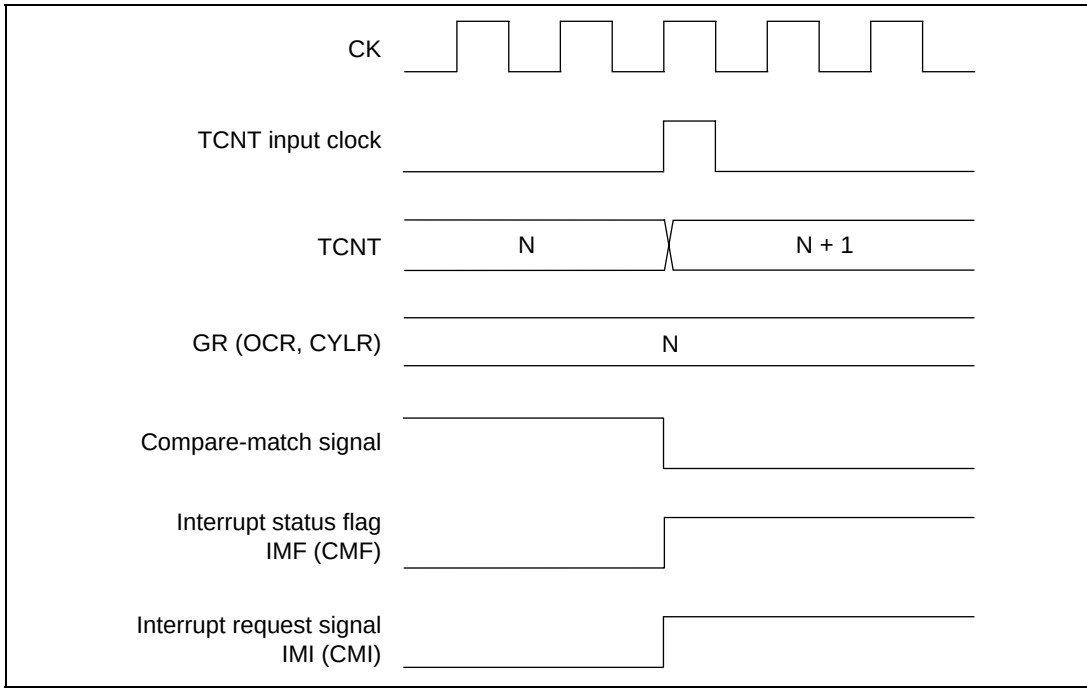


Figure 11.38 IMF (CMF) Setting Timing in Compare-Match

The timing in this case is shown in figure 11.39.

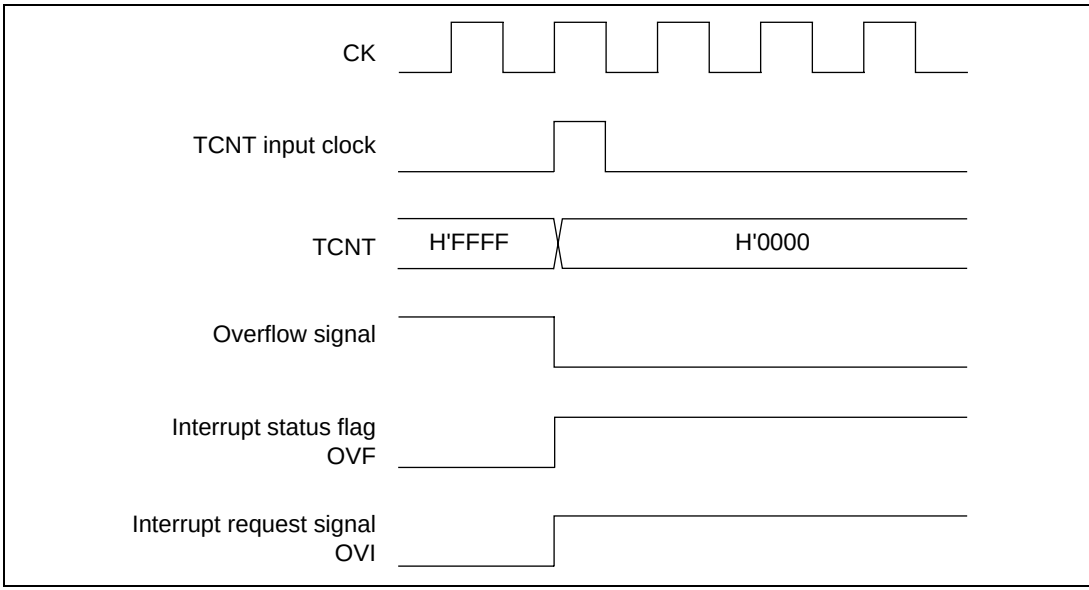


Figure 11.39 OVF Setting Timing in Overflow

the next DCNT input clock pulse is input (which underflow occurs). However, when DCNT is H'0000, it remains unchanged at H'0000 no matter how many DCNT input clock pulses are input.

When DCNT is cleared by means of the one-shot pulse function, the OSF bit is cleared when the next DCNT input clock is input.

The timing in this case is shown in figure 11.40.

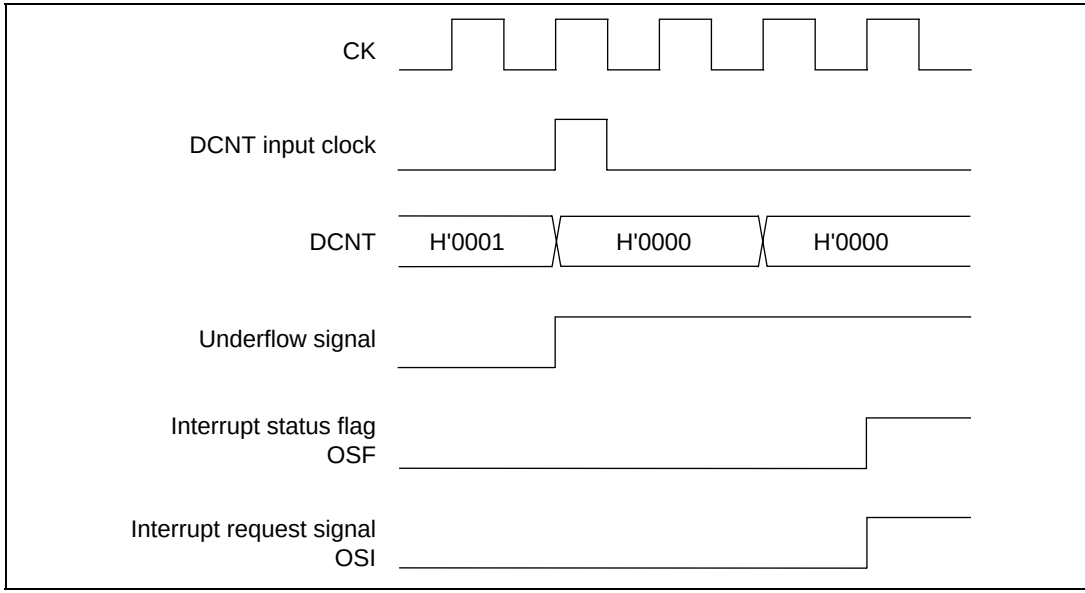


Figure 11.40 OSF Setting Timing in Underflow

(ITVRR), the IIF bit is set to 1 in the timer status register (TSR).

The timing in this case is shown in figure 11.41. TCNT0 value N in the figure is the counter value when TCNT0L bit 6–13 changes to 1. (For example, N = H'00000400 in the case of bit 10, H'00000800 in the case of bit 11, etc.)

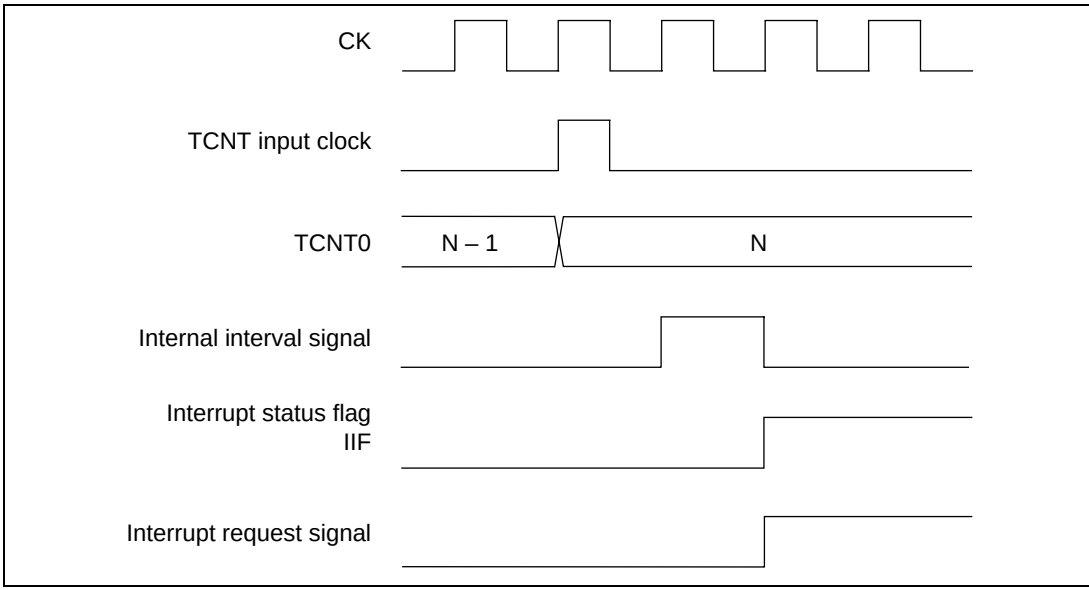


Figure 11.41 Timing of IIF Setting Timing by Interval Timer

Clearing by CPU Program: The interrupt status flag is cleared when the CPU writes 0 to the flag after reading it while set to 1.

The procedure and timing in this case are shown in figure 11.42.

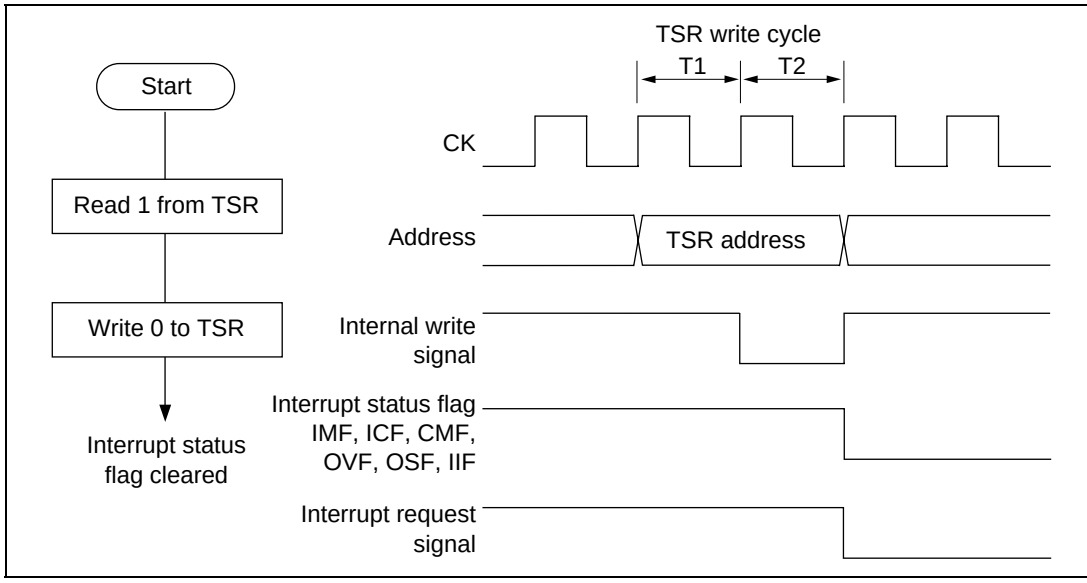


Figure 11.42 Procedure and Timing for Clearing by CPU Program

capture or compare-match.

The procedure and timing in this case are shown in figure 11.43.

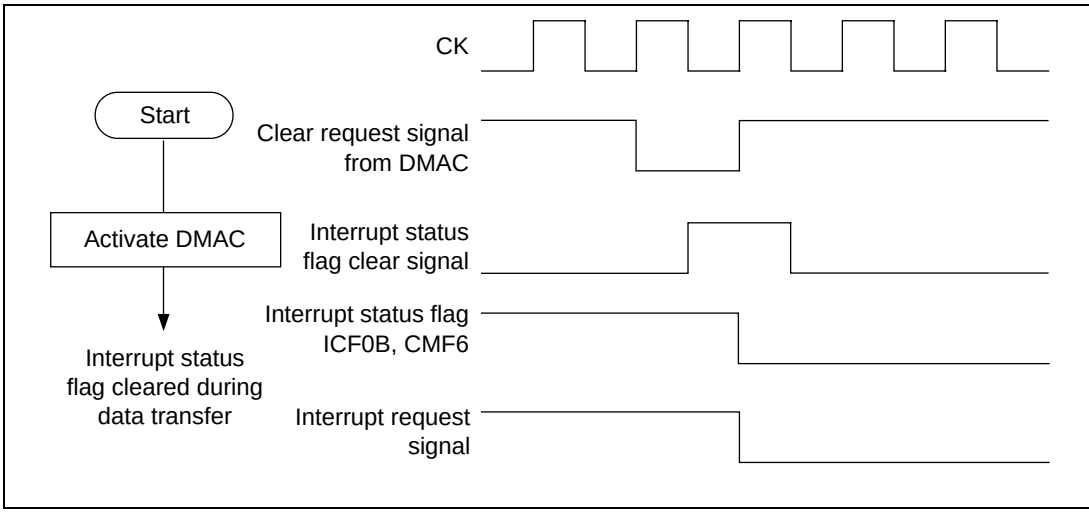


Figure 11.43 Procedure and Timing for Clearing by DMAC

11.5.1 Registers Requiring 32-Bit Access

Free-running counters 0 and 10A (TCNT0, TCNT10A), input capture registers 0A to 0D and 10A (ICR0A to ICR0D, ICR10A), and output compare register 10A (OCR10A) are 32-bit registers. As these registers are connected to the CPU via an internal 16-bit data bus, a read or write (read only, in the case of ICR0A to ICR0D and ICR10A) is automatically divided into two 16-bit accesses.

Figure 11.44 shows a read from TCNT0, and figure 11.45 a write to TCNT0.

When reading TCNT0, in the first read the TCNT0H (upper 16-bit) value is output to the internal data bus, and at the same time, the TCNT0L (lower 16-bit) value is output to an internal buffer register. Then, in the second read, the TCNT0L (lower 16-bit) value held in the internal buffer register is output to the internal data bus.

When writing to TCNT0, in the first write the upper 16 bits are output to an internal buffer register. Then, in the second write, the lower 16 bits are output to TCNT0L, and at the same time, the upper 16 bits held in the internal buffer register are output to TCNT0H to complete the write. The above method performs simultaneous reading and simultaneous writing of 32-bit data, preventing contention with an up-count.

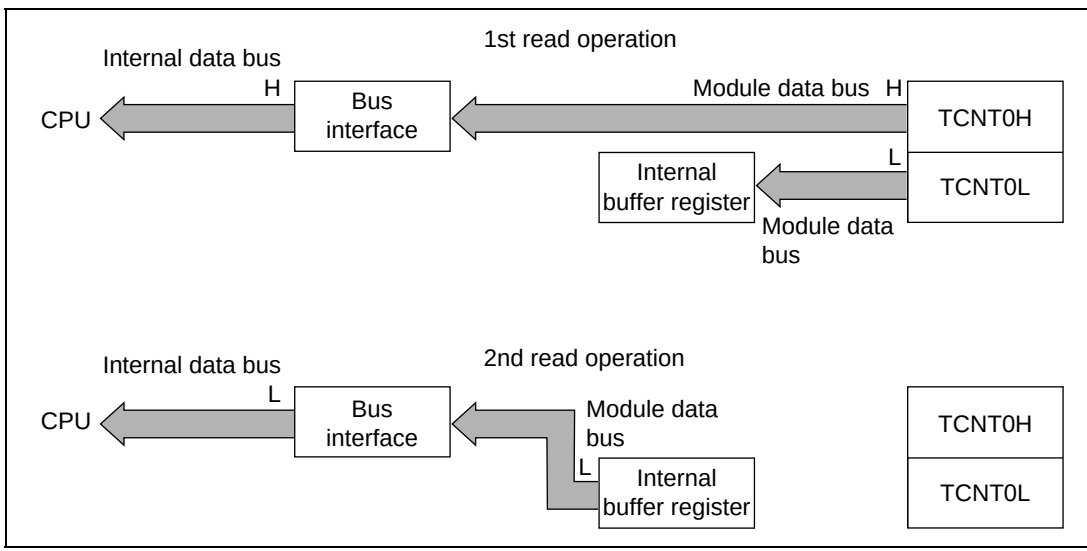


Figure 11.44 Read from TCNT0

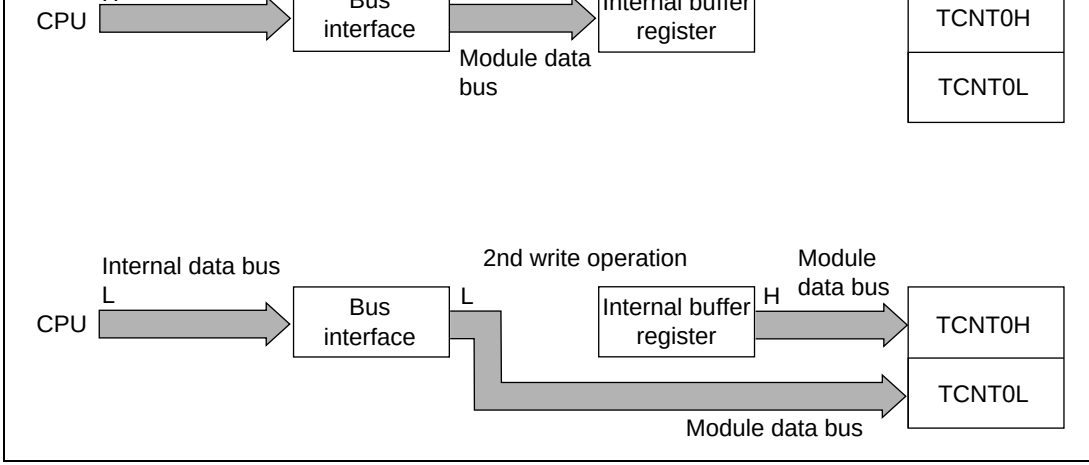


Figure 11.45 Write to TCNT0

Timer registers 1, 2, and 3 (TSTR1, TSTR2, TSTR3) are 8-bit registers. As these registers are connected to the CPU via an internal 16-bit data bus, a simultaneous 32-bit read or write access to TSTR1, TSTR2, and TSTR3 is automatically divided into two 16-bit accesses.

Figure 11.46 shows a read from TSTR, and figure 11.47 a write to TSTR.

When reading TSTR, in the first read the TSTR1 and TSTR2 (upper 16-bit) value is output to the internal data bus. Then, in the second read, the TSTR3 (lower 16-bit) value is output to the internal data bus.

When writing to TSTR, in the first write the upper 16 bits are written to TSTR1 and TSTR2. Then, in the second write, the lower 16 bits are written to TSTR3. Note that, with the above method, in a 32-bit write the write timing is not the same for TSTR1/TSTR2 and TSTR3.

For information on 8-bit and 16-bit access, see section 11.5.4, 8-Bit or 16-Bit Accessible Registers.

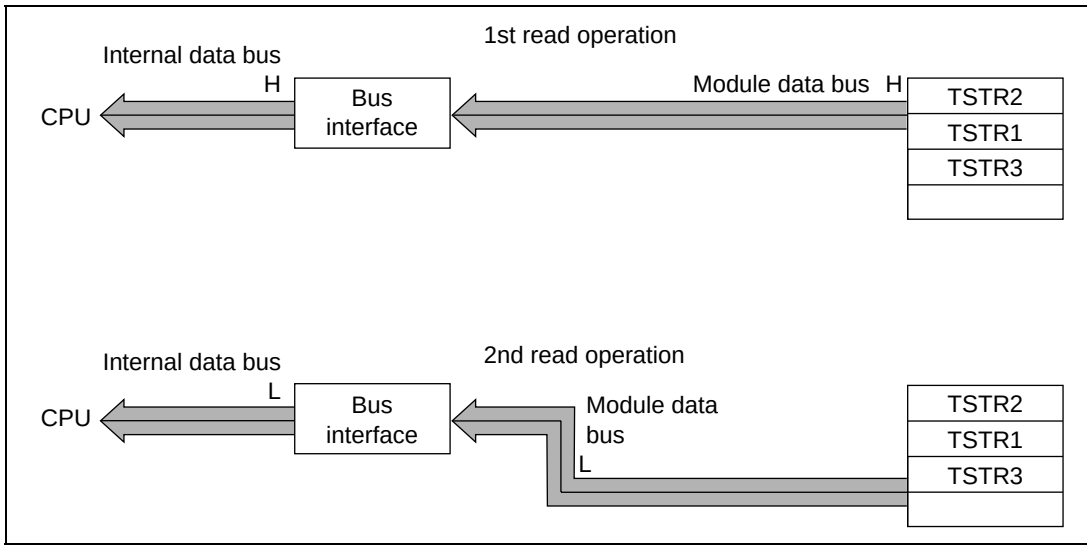


Figure 11.46 Read from TSTR1, TSTR2, and TSTR3

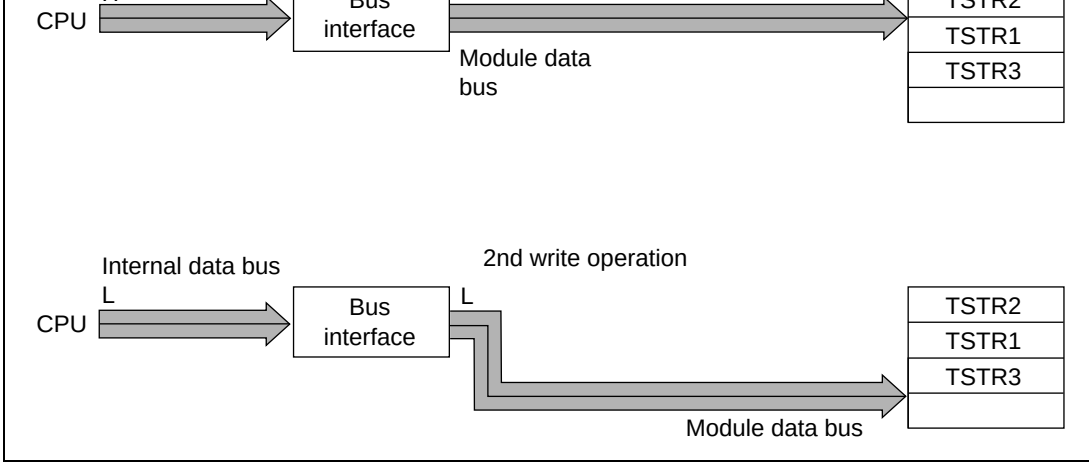


Figure 11.47 Write to TSTR1, TSTR2, and TSTR3

11.5.3 Registers Requiring 16-Bit Access

The free-running counters (TCNT; but excluding TCNT0, TCNT10A, TCNT10B, TCNT10D, and TCNT10H), the general registers (GR; but excluding GR9A to GR9D), down-counters (DCNT), offset base register (OSBR), cycle registers (CYLR), buffer registers (BFR), duty registers (DTR), timer connection register (TCNR), one-shot pulse terminate register (OTR), down-count start register (DSTR), output compare registers (OCR: but excluding OCR10B), reload registers (RLDR8, RLD10C), correction counter clear register (TCCLR10), timer interrupt enable register (TIER), and timer status register (TSR) are 16-bit registers. These registers are connected to the CPU via an internal 16-bit data bus, and can be read or written (read only, in the case of OSBR) a word at a time.

Figure 11.48 shows the operation when performing a word read or write access to TCNT1A.

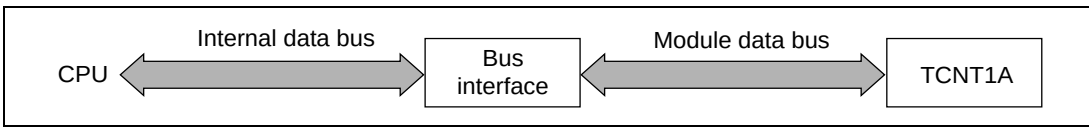


Figure 11.48 TCNT1A Read/Write Operation

The timer control registers (TCR1A, TCR1B, TCR2A, TCR2B, TCR6A, TCR6B, TCR7A, TCR7B), timer I/O control registers (TIOR1A to TIOR1D, TIOR2A to TIOR2D, TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B), and the timer start register (TSTR1, TSTR2, TSTR3) are 8-bit registers. These registers are connected to the upper 8 bits or lower 8 bits of the internal 16-bit data bus, and can be read or written a byte at a time.

In addition, a pair of 8-bit registers for which only the least significant bit of the address is different, such as timer I/O control register 1A (TIOR1A) and timer I/O control register 1B (TIOR1B), can be read or written in combination a word at a time.

Figures 11.49 and 11.50 show the operation when performing individual byte read or write accesses to TIOR1A and TIOR1B. Figure 11.51 shows the operation when performing a word read or write access to TIOR1A and TIOR1B simultaneously.

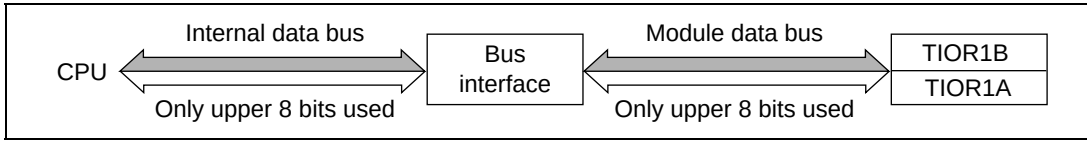


Figure 11.49 Byte Read/Write Access to TIOR1B

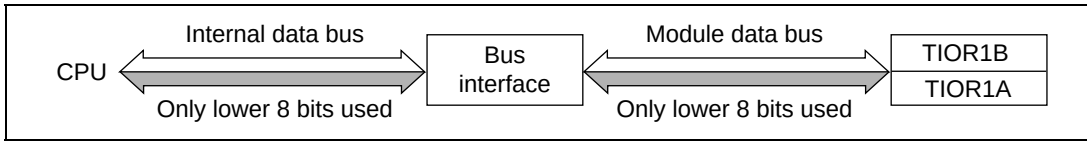


Figure 11.50 Byte Read/Write Access to TIOR1A

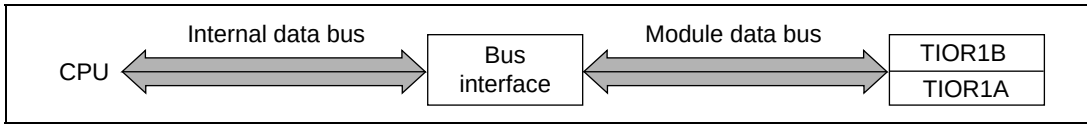


Figure 11.51 Word Read/Write Access to TIOR1A and TIOR1B

The timer mode register (TMDR), prescaler register (PSCR), timer I/O control registers (TIOR0, TIOR10, TIOR11), trigger mode register (TRGMDR), interval interrupt request register (ITVRR), timer control registers (TCR3, TCR4, TCR5, TCR8, TCR9A to TCR9C, TCR10, TCR11), PWM mode register (PMDR), reload enable register (RLDENR), free-running counters (TCNT10B, TCNT10D, TCNT10H), event counter (ECNT), general registers (GR9A to GR9F), output compare register (OCR10B), and noise canceler register (NCR) are 8-bit registers. These registers are connected to the upper 8 bits of the internal 16-bit data bus, and can be read or written a byte at a time.

Figure 11.52 shows the operation when performing individual byte read or write accesses to ITVRR1.

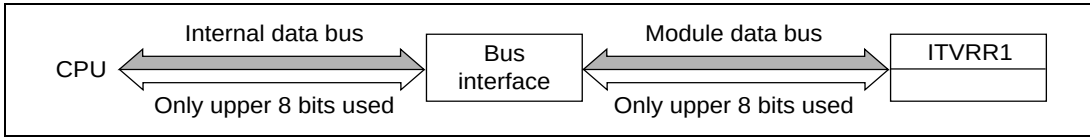


Figure 11.52 Byte Read/Write Access to ITVRR1

11.6 Sample Setup Procedures

Sample setup procedures for activating the various ATU-II functions are shown below.

Sample Setup Procedure for Input Capture: An example of the setup procedure for input capture is shown in figure 11.53.

1. Select the first-stage counter clock ø' in prescaler register (PSCR) and the second-stage counter clock ø" with the CKSEL bit in the timer control register (TCR). When selecting an external clock, also select the external clock edge type with the CKEG bit in TCR.
2. Set the port control register, corresponding to the port for signal input as the input capture trigger, to ATU input capture input.
3. Select rising edge, falling edge, or both edges as the input capture signal input edge(s) with the timer I/O control register (TIOR).

   If necessary, a timer interrupt request can be sent to the CPU on input capture by making the appropriate setting in the interrupt enable register (TIER). In channel 0, setting the DMAC allows DMAC activation to be performed.
4. Set the corresponding bit to 1 in the timer start register (TSTR) to start the free-running counter (TCNT) for the relevant channel.

Note: When input capture occurs, the counter value is always captured, irrespective of free-running counter (TCNT) activation.

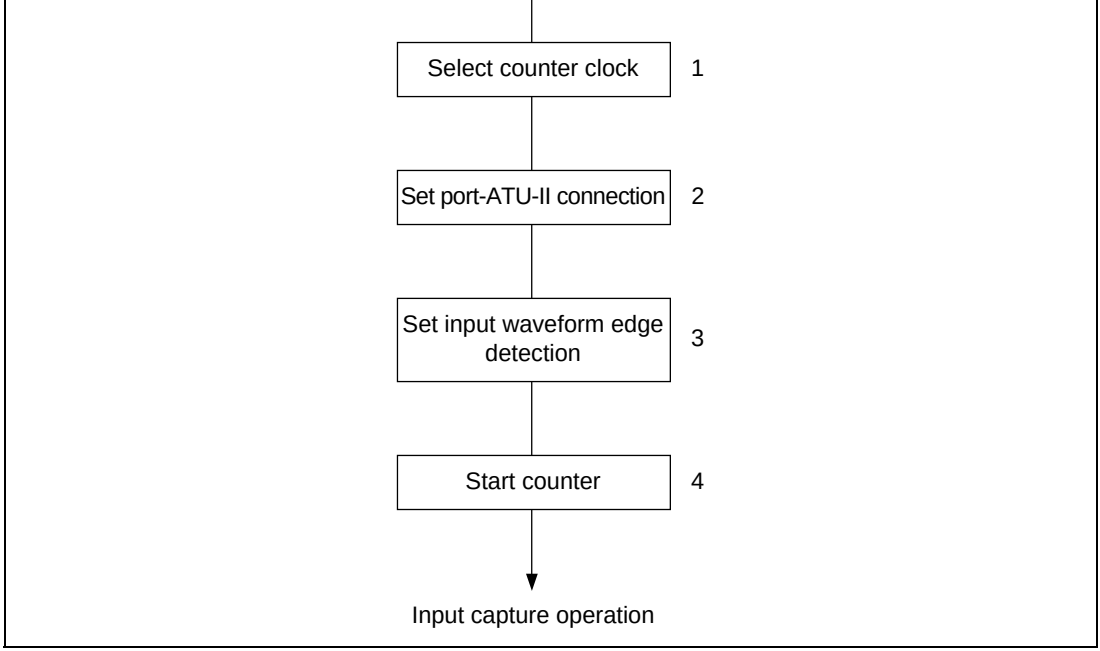


Figure 11.53 Sample Setup Procedure for Input Capture

Sample Setup Procedure for Waveform Output by Output Compare-Match: An example of the setup procedure for waveform output by output compare-match is shown in figure 11.54.

1. Select the first-stage counter clock ø' in prescaler register (PSCR), and the second-stage counter clock ø" with the CKSEL bit in the timer control register (TCR). When selecting an external clock, also select the external clock edge type with the CKEG bit in TCR.
2. Set the port control register corresponding to the waveform output port to ATU output compare-match output. Also set the corresponding bit to 1 in the port IO register to specify the output attribute for the port.
3. Select 0, 1, or toggle output for output compare-match output with the timer I/O control register (TIOR). If necessary, a timer interrupt request can be sent to the CPU on output compare-match by making the appropriate setting in the interrupt enable register (TIER).
4. Set the timing for compare-match generation in the ATU general register (GR) corresponding to the port set in 2.
5. Set the corresponding bit to 1 in the timer start register (TSTR) to start the free-running counter (TCNT). Waveform output is performed from the relevant port when the TCNT value and GR value match.

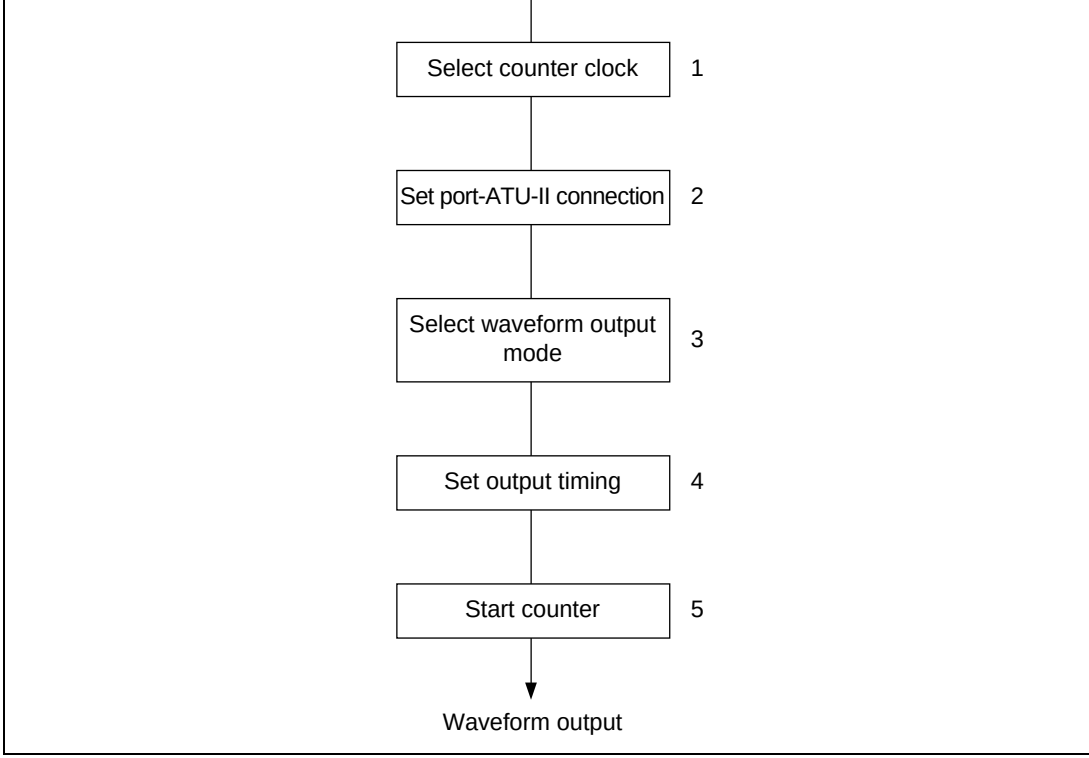


Figure 11.54 Sample Setup Procedure for Waveform Output by Output Compare-Match

figure 11.55.

1. Set the timing for compare-match generation in the channel 10 output compare register (OCR10B).
2. Set the TRG0DEN bit to 1 in the channel 10 timer control register (TCR10).
3. Set the corresponding bit to 1 in the timer start register (TSTR) to start the channel 10 free-running counter (TCNT10B). On compare-match between TCNT10 and OCR10B, the compare-match signal is transmitted to channel 0 as the channel 0 ICR0D input capture signal.

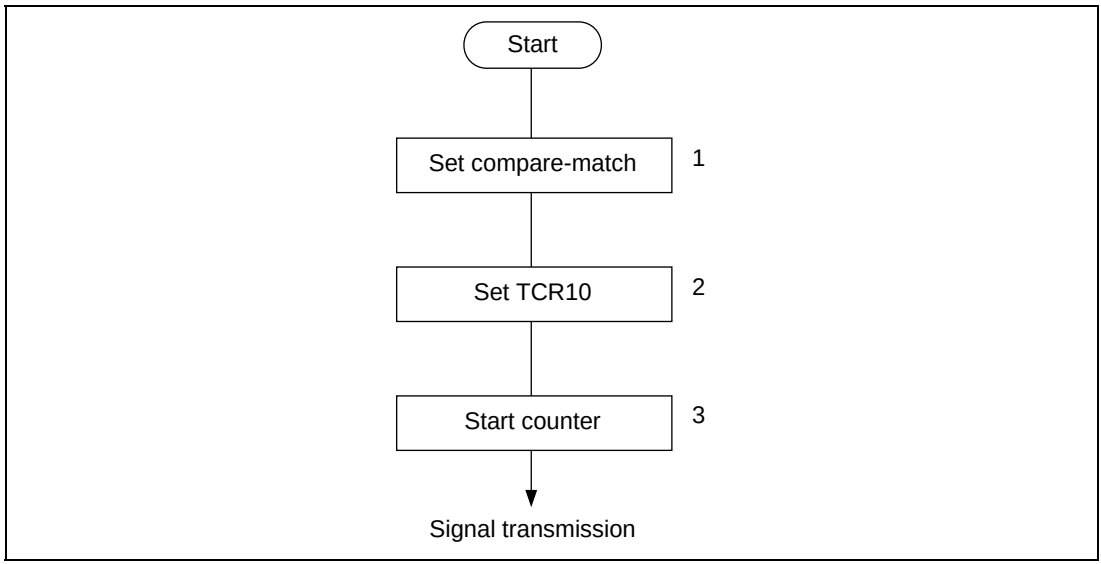


Figure 11.55 Sample Setup Procedure for Compare-Match Signal Transmission

1. Set the first-stage counter clock ø' in prescaler register 1 (PSCR1), and select the second-stage counter clock ø" with the CKSEL bit in timer control register8 TCR8.
2. Set port K control registers H and L (PKCRH, PKCRL) corresponding to the waveform output port to ATU one-shot pulse output. Also set the corresponding bit to 1 in the port K IO register (PKIOR) to specify the output attribute.
3. Set the one-shot pulse width in the down-counter (DCNT) corresponding to the port set in (2). If necessary, a timer interrupt request can be sent to the CPU when the down-counter underflows by making the appropriate setting in the interrupt enable register (TIER8).
4. Set the corresponding bit (DST8A to DST8P) to 1 in the down-count start register (DSTR) to start the down-counter (DCNT).

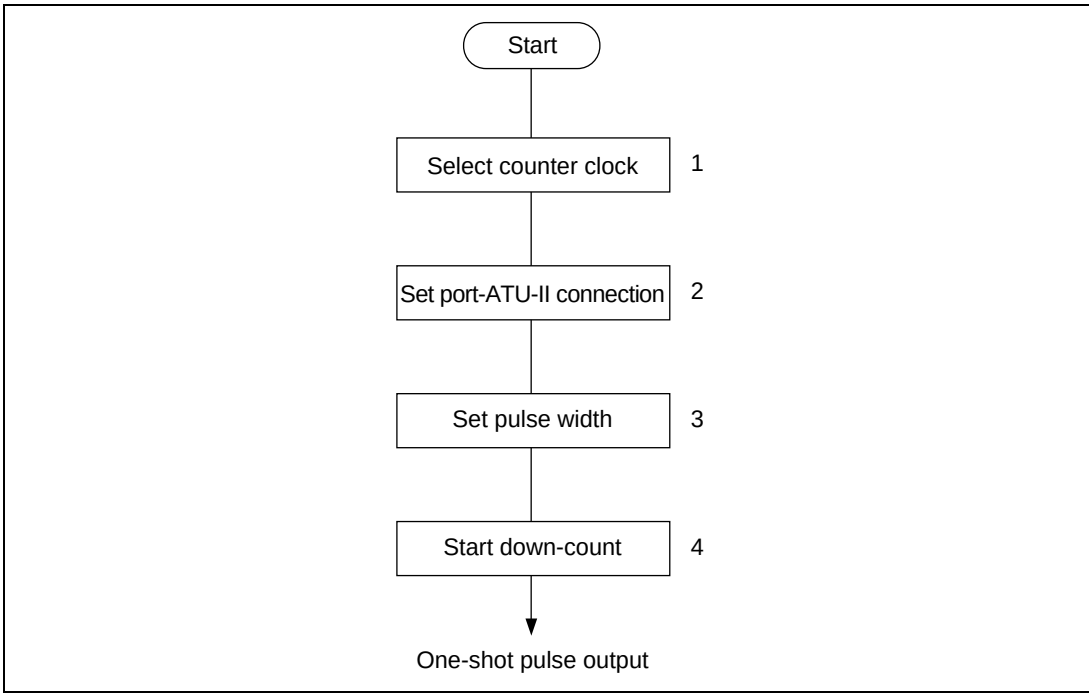


Figure 11.56 Sample Setup Procedure for One-Shot Pulse Output

1. Set the first-stage counter clock ø' in prescaler register 1 (PSCR1), and select the second-stage counter clock ø" with the CKSEL bit in the timer control register (TCR1, TCR2, TCR8).
2. Set port K control registers H and L (PKCRH, PKCRL) corresponding to the waveform output port to ATU one-shot pulse output. Also set the corresponding bit to 1 in the port K IO register (PKIOR) to specify the output attribute
3. Set the one-shot pulse width in the down-counter (DCNT) corresponding to the port set in (2). If necessary, a timer interrupt request can be sent to the CPU when the down-counter underflows by making the appropriate setting in the interrupt enable register (TIER8).
4. Set the offset width in the channel 1 or 2 general register (GR1A—GR1H, GR2A—GR2H) connected to the down-counter (DCNT) corresponding to the port set in (2), and in the output compare register (OCR1, OCR2A—OCR2H). Set the timer I/O control register (TIOR1A—TIOR1D, TIOR2A—TIOR2D) to the compare-match enabled state.
5. Set the start/terminate trigger by means of the trigger mode register (TRGMDR), timer connection register (TCNR), and one-shot pulse terminate register (OTR), so that it corresponds to the port set in step 2 above.
6. Set the corresponding bit to 1 in the timer start register (TSTR) to start the channel 1 or 2 free-running counter (TCNT1, TCNT2). When the TCNT value and GR value or OCR value match, the corresponding DCNT starts counting down or is forcibly cleared, and one-shot pulse output is performed.

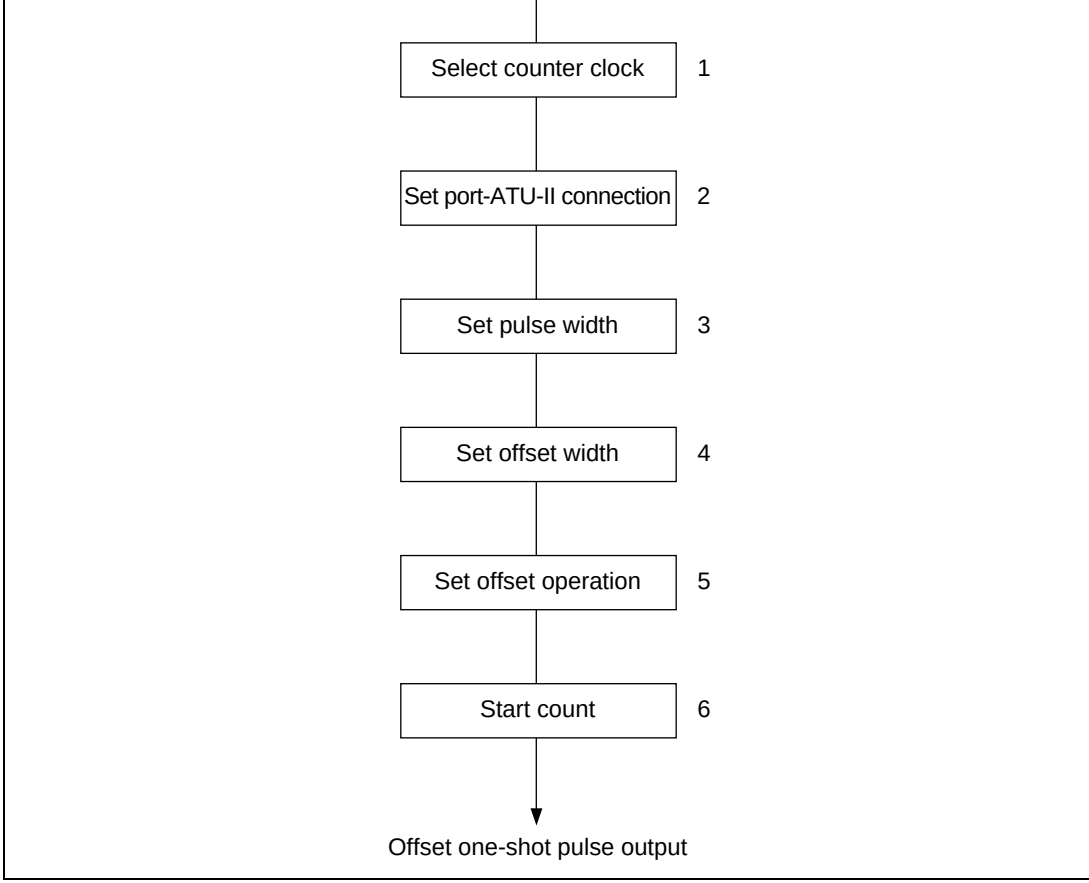


Figure 11.57 Sample Setup Procedure for Offset One-Shot Pulse Output

1. Set the first-stage counter clock ø' in prescaler register 1 (PSCR1).
2. Set the ITVE bit to be used in the interval interrupt request register (ITVRR) to 1. An interrupt request can be sent to the CPU when the corresponding bit changes to 1 in the channel 0 free-running counter (TCNT0).

   To start A/D converter sampling, set the ITVA bit to be used in ITVRR to 1.
3. Set bit 0 to 1 in the timer start register (TSTR) to start TCNT0.

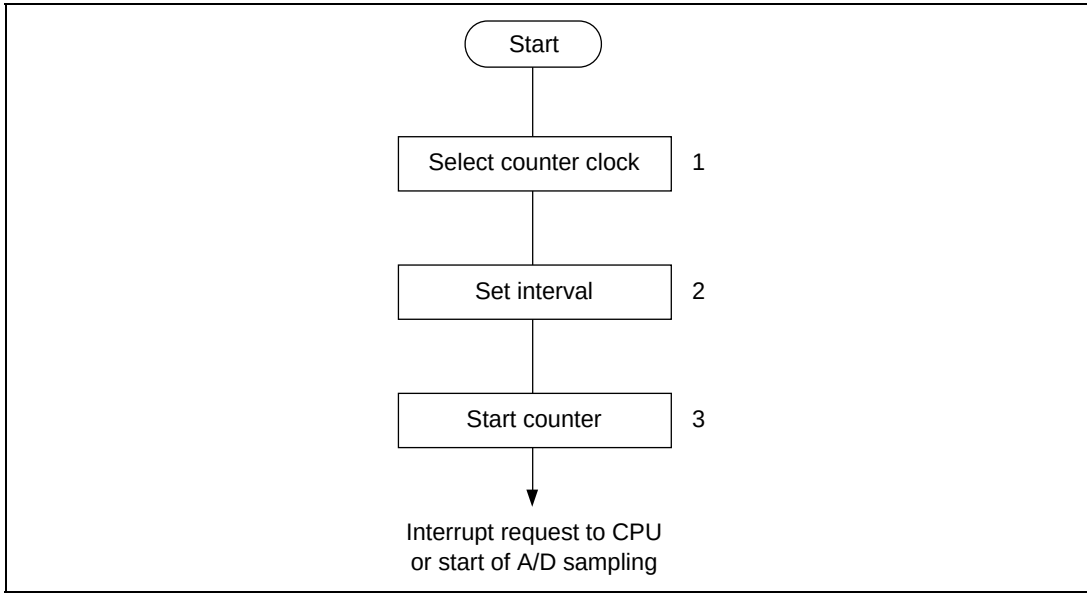


Figure 11.58 Sample Setup Procedure for Interval Timer Operation

1. Set the first-stage counter clock ø' in prescaler register 1 (PSCR1), and select the second-stage counter clock ø" with the CKSEL bit in the timer control register (TCR). When selecting an external clock, at the same time select the external clock edge type with the CKEG bit in TCR.
2. Set the port control registers (PxCRH, PxCRL) corresponding to the waveform output port to ATU output compare-match output. Also set the corresponding bit to 1 in the port IO register (PxIOR) to specify the output attribute.
3. Set bit T3PWM–T5PWM in the timer mode register (TMDR) to PWM mode. When PWM mode is set, the timer operates in PWM mode irrespective of the timer I/O control register (TIOR) contents, and general registers (GR3A to GR3D, GR4A to GR4D, GR5A to GR5D) can be written to.
4. The GR3A–GR3C, GR4A–GR4C, and GR5A–GR5C ATU general registers are used as duty registers (DTR), and the GR3D, GR4D, and GR5D ATU general registers as cycle registers (CYLR). Set the PWM waveform output 0 output timing in DTR, and the PWM waveform output 1 output timing in CYLR. Also, if necessary, interrupt requests can be sent to the CPU at the 0/1 output timing by making a setting in the timer interrupt enable register (TIER).
5. Set the corresponding bit to 1 in the timer start register (TSTR) to start the free-running counter (TCNT) for the relevant channel.

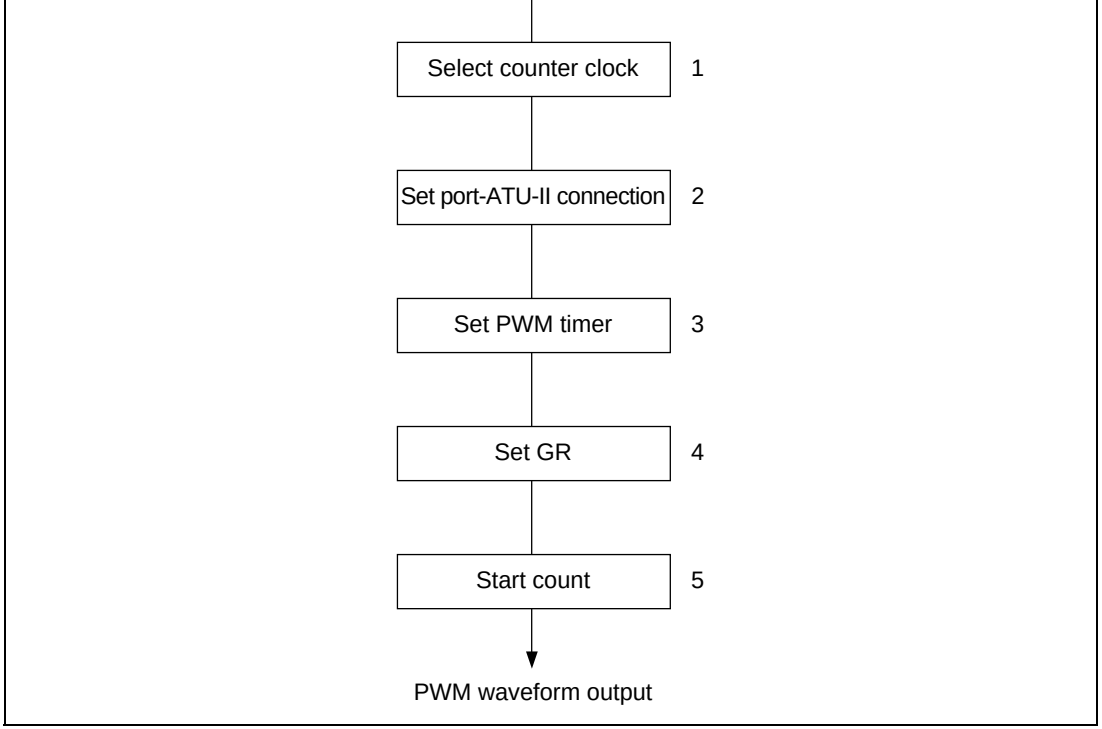


Figure 11.59 Sample Setup Procedure for PWM Timer Operation (Channels 3 to 5)

1. Set the first-stage counter clock ø' in prescaler register 2 and 3 (PSCR2, PSCR3), and select the second-stage counter clock ø" with the CKSEL bit in the timer control register (TCR6A, TCR6B, TCR7A, TCR7B).
2. Set the port B control register L (PBCRL) corresponding to the waveform output port to ATU PWM output. Also set the corresponding bit to 1 in the port B IO register (PBIOR) to specify the output attribute.
3. Set PWM waveform output 1 output timing in the cycle register (CYLR6A to CYLR6D, CYLR7A to CYLR7D), and set the PWM waveform output 0 output timing in the buffer register (BFR6A to BFR6D, BFR7A to BFR7D) and duty register (DTR6A to DTR6D, DTR7A to DTR7D). If necessary, an interrupt request can be sent to the CPU on a compare-match between the CYLR value and the free-running counter (TCNT) value by making the appropriate setting in the interrupt enable register (TIERE). In addition, setting the DMAC allows DMAC activation to be performed.
4. Set the corresponding bit to 1 in the timer start register (TSTR) to start the TCNT counter for the relevant channel.

Notes: 1. Do not make a setting in DTR after the counter is started. Use BFR to make a DTR setting.
2. 0% duty is specified by setting H'0000 in the duty register (DTR), and 100% duty is specified by setting buffer register (BFR) = cycle register (CYLR). Do not set BFR > CYLR.

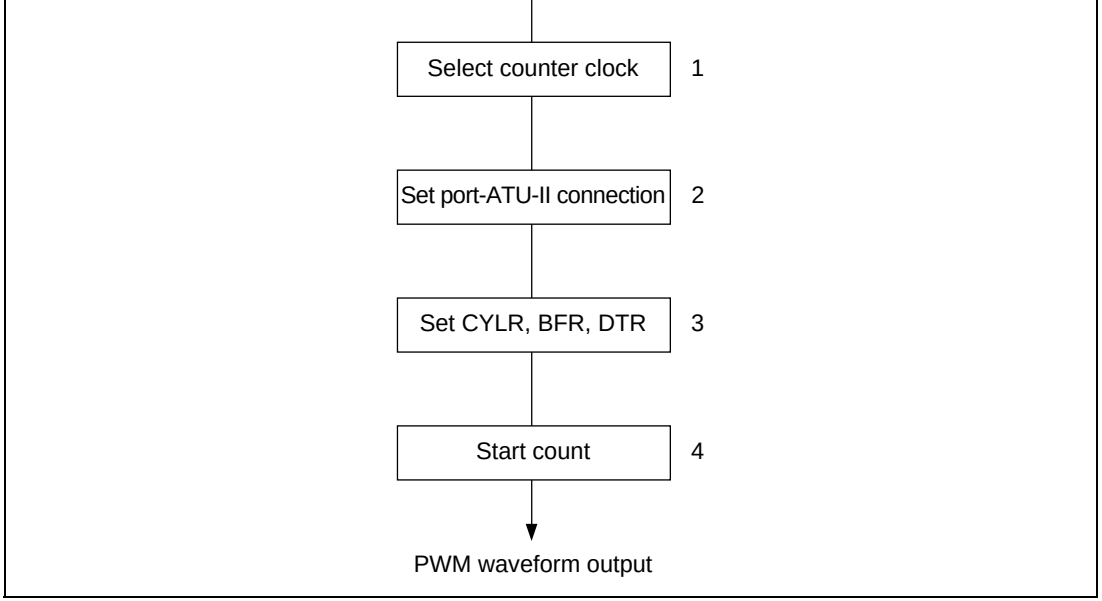


Figure 11.60 Sample Setup Procedure for PWM Timer Operation (Channels 6 and 7)

Sample Setup Procedure for Event Counter Operation: An example of the setup procedure for event counter operation is shown in figure 11.61.

1. Set the number of events to be counted in a general register (GR9A to GR9D). Also, if necessary, an interrupt request can be sent to the CPU upon compare-match by making a setting in the timer interrupt enable register (TIER).
2. Set the port control register, corresponding to the port for signal input to the event counter, to ATU event counter input.
3. Select the event counter count edge with the EGSEL bits in the channel 9 timer control register (TCR9A to TCR9C).
4. Input a signal to the event counter input pin.

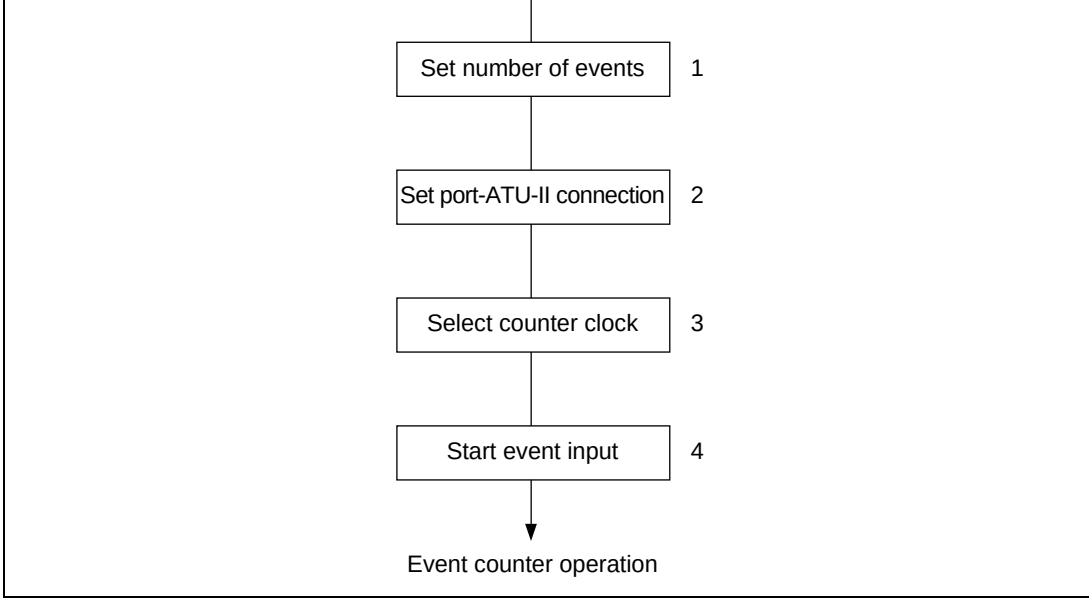


Figure 11.61 Sample Setup Procedure for Event Counter Operation

figure 11.62.

1. Set the port control register, corresponding to the port for signal input to the event counter, to ATU event counter input.
2. Set the channel 3 timer I/O control register (TIOR3A, TIOR3B), and select the input capture disable setting for the general registers (GR3A to GR3D). Input from pins TIO3A to TIO3D is masked.
3. Select the event counter count edge with the EGSEL bits in the channel 9 timer control register (TCR9A, TCR9B), and set the TRG3xEN bit to 1. Set the timing for capture in the general register (GR9A to GR9D).
4. Set bit STR3 to 1 in the timer start register (TSTR) to start the channel 3 free-running counter (TCNT3).
5. Input a signal to the event counter input pin.

Note: An interrupt request can be sent to the CPU upon channel 9 compare-match by making a setting in the timer interrupt enable register (TIER), but an interrupt request cannot be sent to the CPU upon channel 3 input capture.

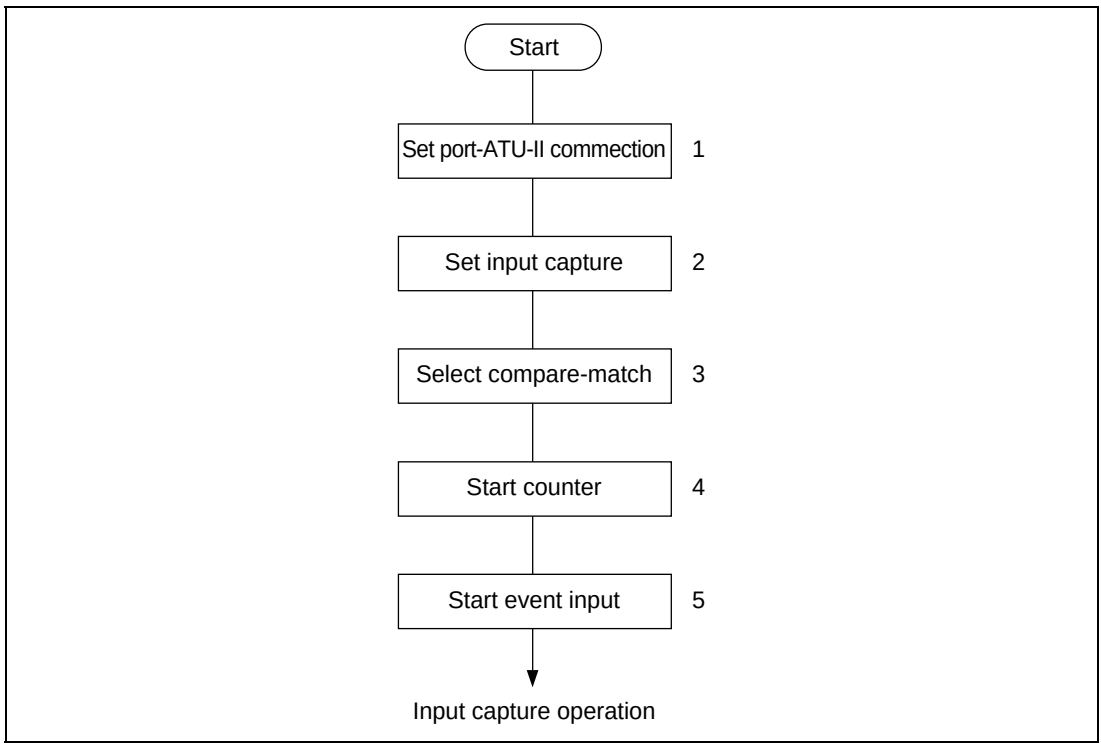


Figure 11.62 Sample Setup Procedure for Compare-Match Signal Transmission

1. Set port B control register H (PBCRH) or port L control register L (PLCRL), corresponding to the port for input of the external signal (missing-teeth signal), to ATU edge input (TI10).
2. Set 1st-stage counter clock ø' in prescaler register 4 (PSCR4). Set the external input (TI10) cycle multiplication factor with the PIM bits in timer I/O control register 10 (TIOR10), and enable reload register 10C (RLD10C) updating with the RLDEN bit. Select the external input edge type with the CKEG bits in timer control register 10 (TCR10).
3. Set general register 10G (GR10G) to the compare-match function with bit IO10G in TIOR10. Also, an interrupt request can be sent to the CPU upon compare-match by making a setting in interrupt enable register 10 (TIER10).
4. Set the timing for compare-match generation in GR10G according to the multiplication factor and number of missing-teeths in the missing-teeth interval set in step 1.
5. Set the corresponding bit to 1 in timer start register 1 (TSTR1) to start the channel 10 count. A compare-match occurs when the values in free-running counter 10G (TCNT10G) and GR10G match.

Note: The TCNT10G counter clock is generated according to the external input edge interval and multiplication factor selected in step 1, and the counter is cleared to H'0000 by an external input edge.

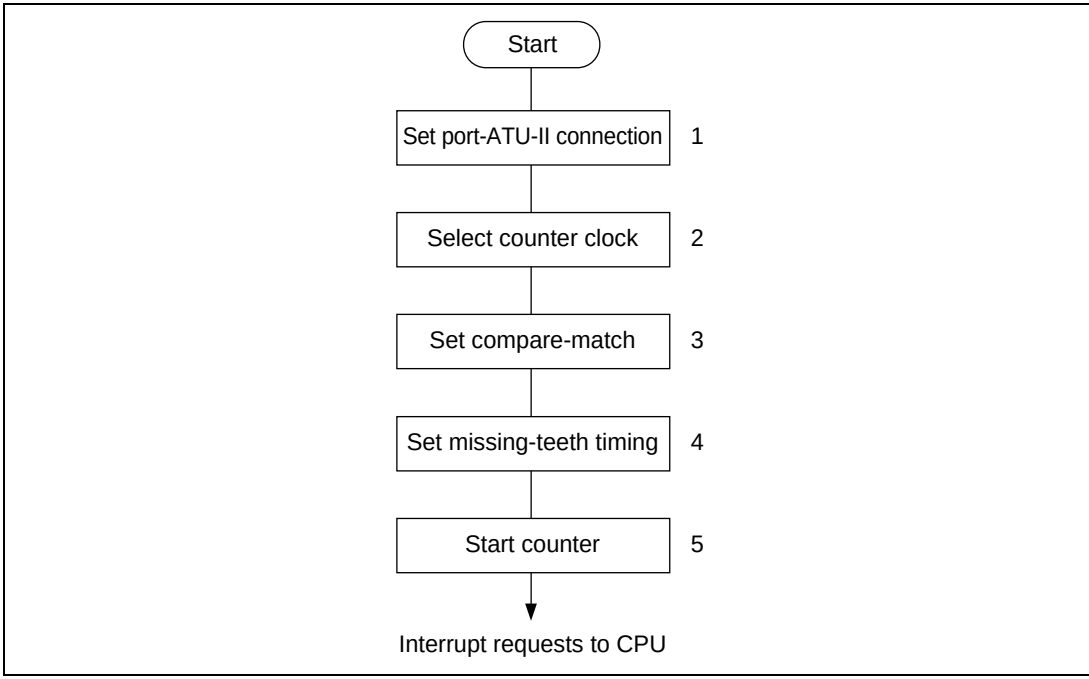


Figure 11.63 Sample Setup Procedure for Missing-Teeth Detection

Note that the kinds of operation and contention described below occur during ATU operation.

Contention between TCNT Write and Clearing by Compare-Match: With channel 3 to 7 free-running counters (TCNT3 to TCNT5, TCNT6A to TCNT6D, TCNT7A to TCNT7D), if a compare-match occurs in the T2 state of a CPU write cycle when counter clearing by compare-match has been set, or when PWM mode is used, the write to TCNT has priority and TCNT clearing is not performed.

The compare-match remains valid, and writing of 1 to the interrupt status flag and waveform output to an external destination are performed in the same way as for a normal compare-match.

The timing in this case is shown in figure 11.64.

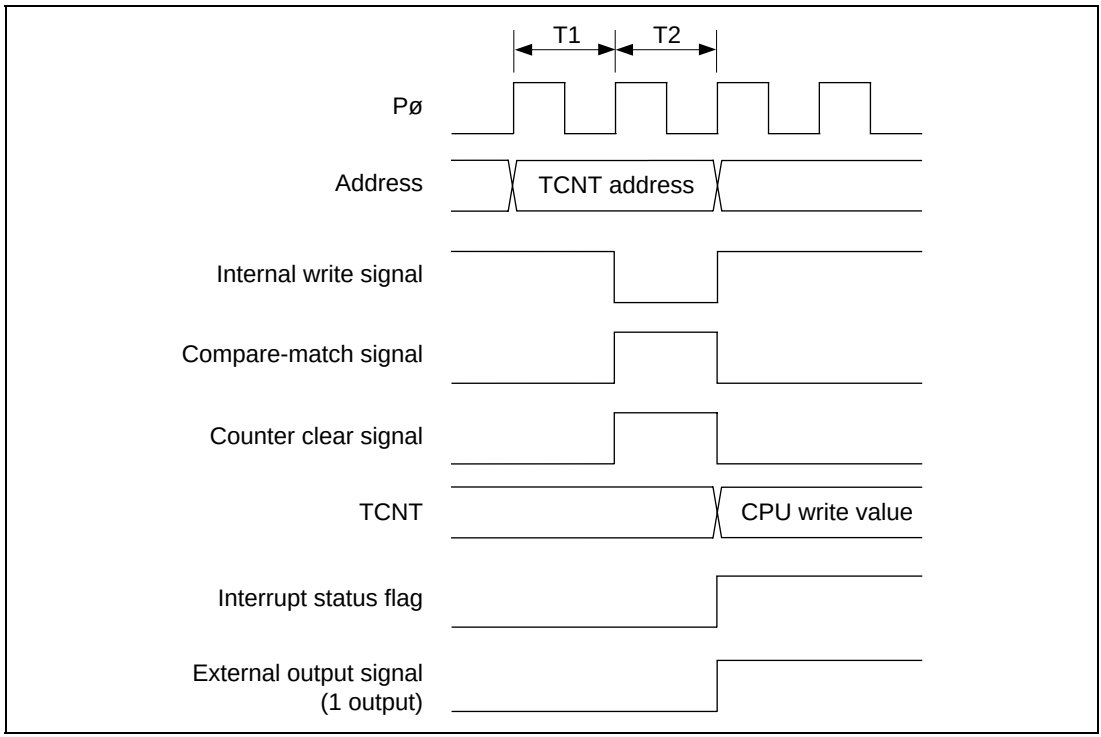


Figure 11.64 Contention between TCNT Write and Clear

TCNT6B, TCNT7A to TCNT7D, TCNT10A to TCNT10H, TCNT11), down-counter (DCNT8A to DCNT8P), or event counter 9 (ECNT9A to ECNT9F) is performed while that counter is counting up or down, the write to the counter has priority and the counter is not incremented or decremented.

The timing in this case is shown in figure 11.65. In this example, the CPU writes H'5555 at the point at which TCNT is to be incremented from H'1001 to H'1002.

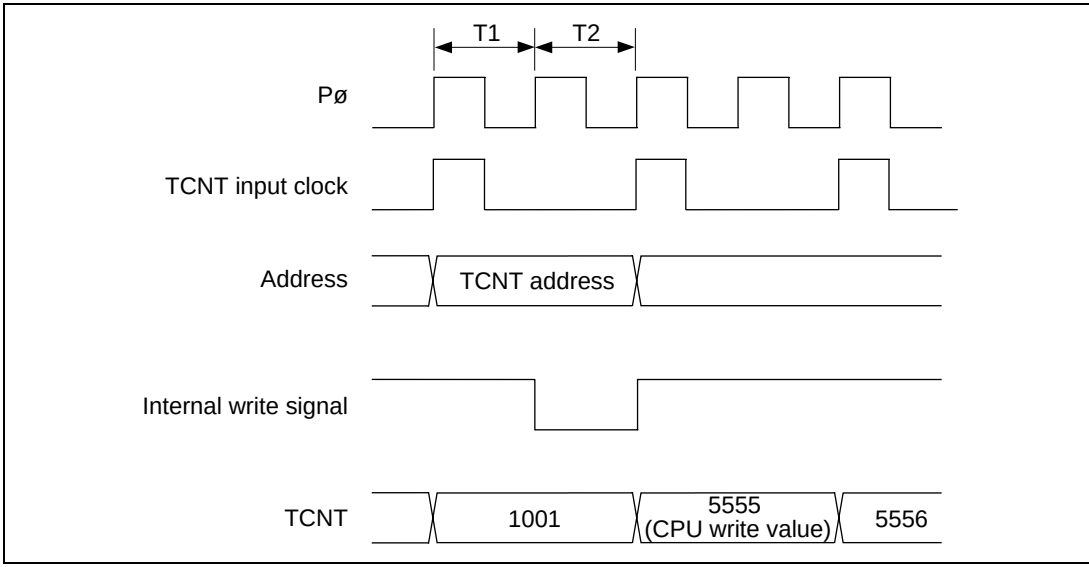


Figure 11.65 Contention between TCNT Write and Increment

TCNT3, TCNT11), if overflow occurs in the T2 state of a CPU write cycle, the write to TCNT has priority and TCNT is not cleared.

Writing of 1 to the interrupt status flag (OVF) due to the overflow is performed in the same way as for normal overflow.

The timing in this case is shown in figure 11.66. In this example, H'5555 is written at the point at which TCNT overflows.

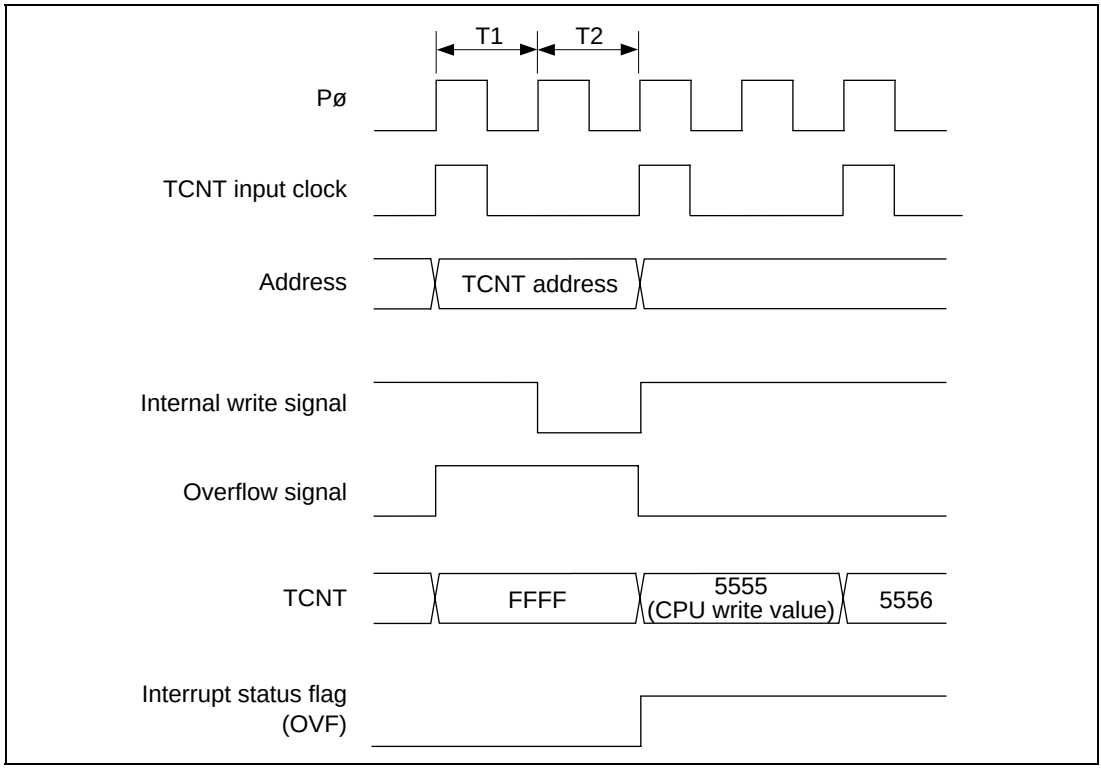


Figure 11.66 Contention between TCNT Write and Overflow

interrupt status flag 0 write cycle by the CPU, clearing by the 0 write has priority and the interrupt status flag is cleared.

The timing in this case is shown in figure 11.67.

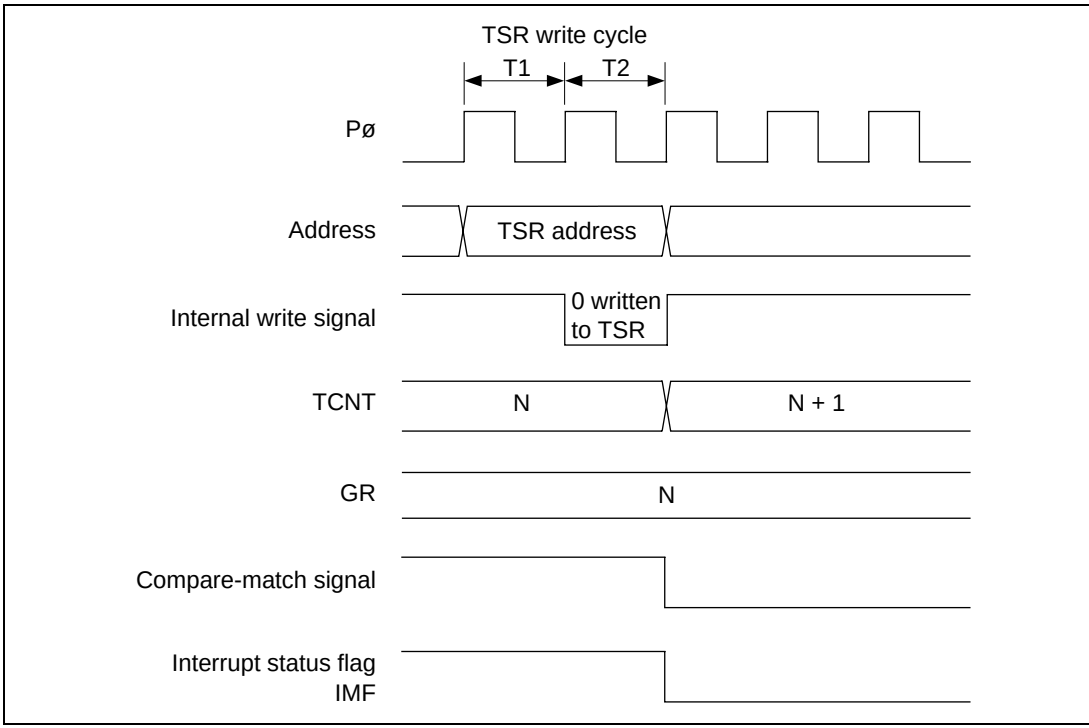


Figure 11.67 Contention between Interrupt Status Flag Setting by Compare-Match and Clearing

corresponding duty register (DTR) due to a cycle register (CYLR) compare-match, and a write to DTR by the CPU, the CPU write value is written to DTR.

Figure 11.68 shows an example in which contention arises when the BFR value is H'AAAA and the value to be written to DTR is H'5555.

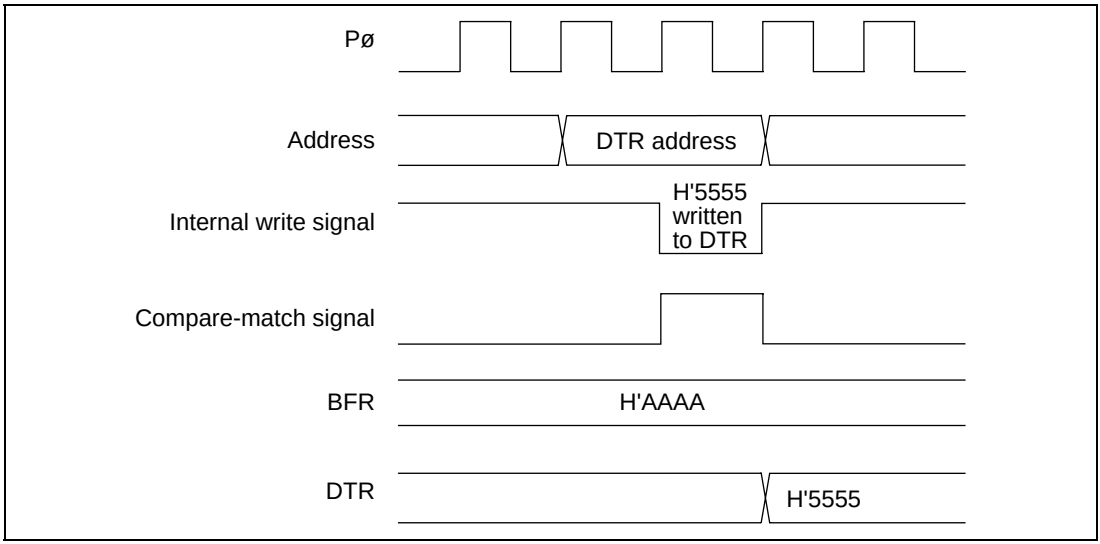


Figure 11.68 Contention between DTR Write and BFR Value Transfer by Buffer Function

status flag (ICF0A to ICF0D, CMF6A to CMF6D, CMF7A to CMF7D) is set by input capture (ICR0A to ICR0D) or compare-match (CYLR6A to CYLR6D, CYLR7A to CYLR7D), clearing by the DMAC has priority and the interrupt status flag is not set.

The timing in this case is shown in figure 11.69.

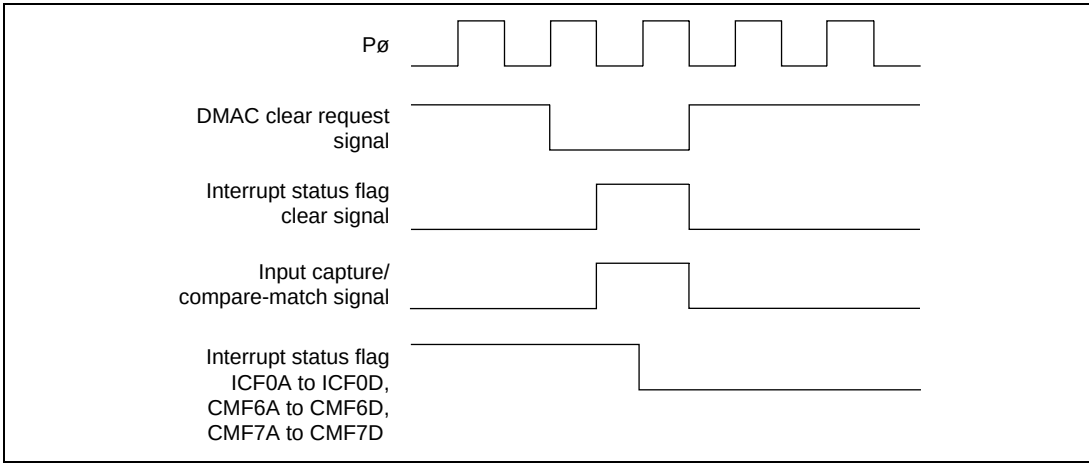


Figure 11.69 Contention between Interrupt Status Flag Clearing by DMAC and Setting by Input Capture/Compare-Match

setting DCNT to H'0000, the corresponding DSTR bit is cleared to 0 and the count is stopped. However, the OSF bit in the timer status register (TSR) is set when DCNT underflows.

Note that when H'0000 is written to DCNT, the corresponding DSTR bit is not cleared to 0 immediately; it is cleared to 0, and the down-counter is stopped, when underflow occurs following the H'0000 write.

The timing in this case is shown in figure 11.70.

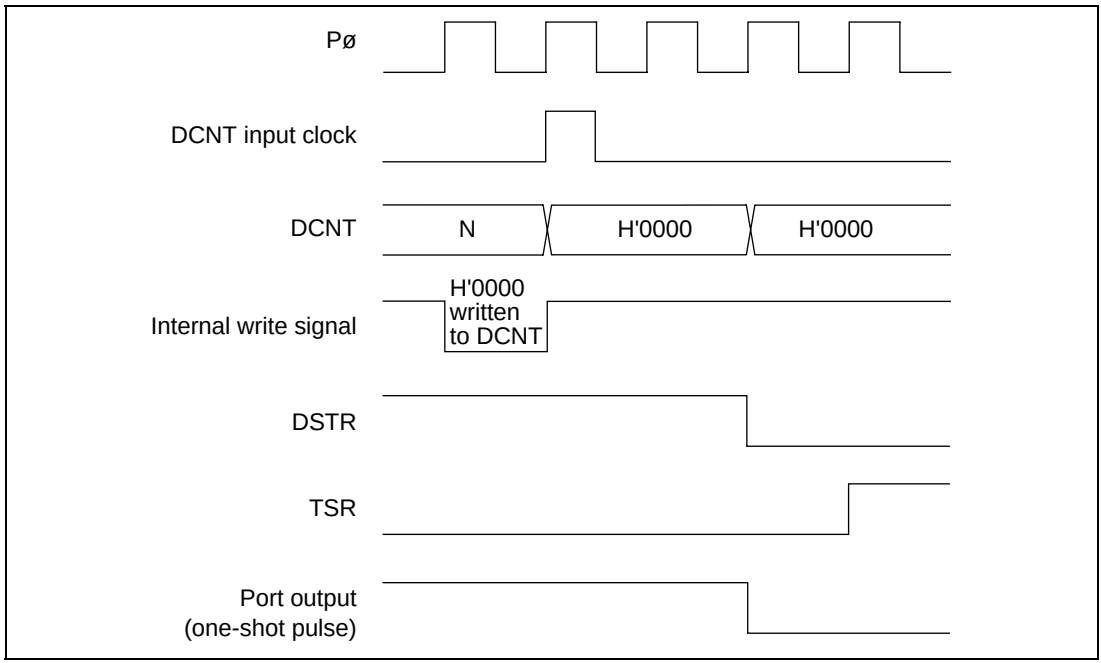


Figure 11.70 Halting of a Down-Counter by the CPU

pin, the TCNT value will be transferred to the corresponding general register (GR) or input capture register (ICR) irrespective of whether the free-running counter (TCNT) is running or halted, and the IMF or ICF bit will be set in the timer status register (TSR).

The timing in this case is shown in figure 11.71.

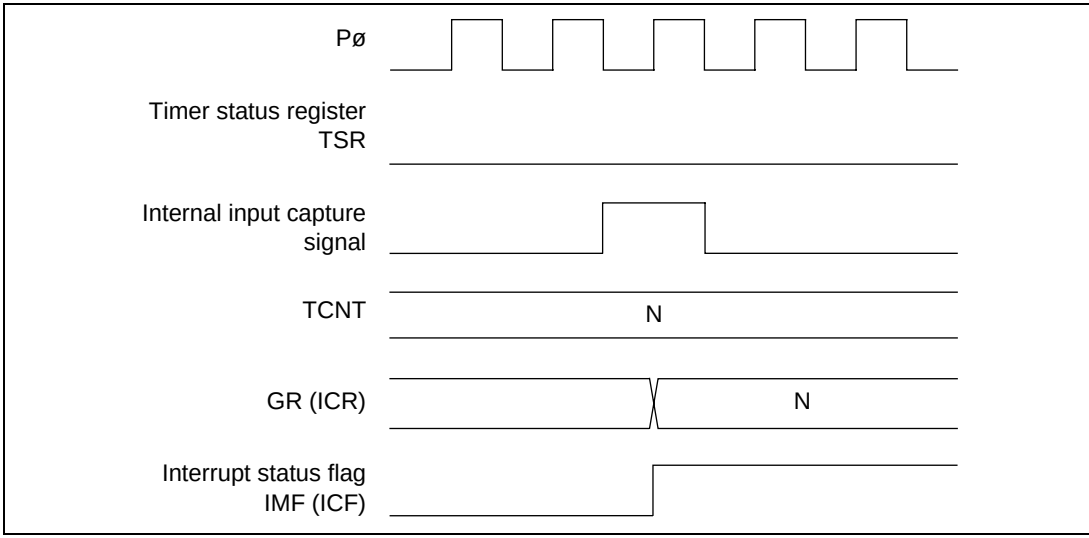


Figure 11.71 Input Capture Operation before Free-Running Counter is Started

CPU, the DCNT continues counting down because the write to the DCNT by the CPU has priority.

The timing in this case is shown in figure 11.72. In this example, a write of H'5555 to DCNT is attempted at the same time as DCNT underflows.

Note: In the SH7055F, the retention of the H’0000 value has priority and the write to the DCNT by the CPU is not performed. Note that the operation of the channel 8 down-counters differs between SH7055SF and SH7055F.

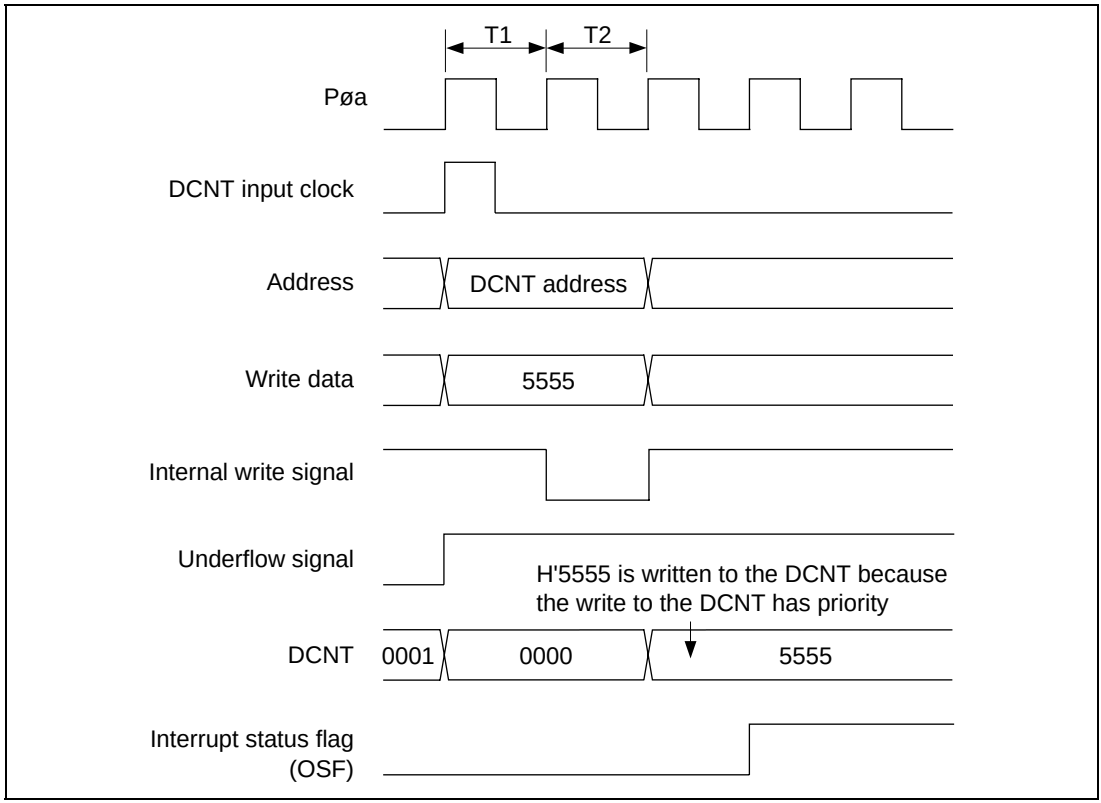


Figure 11.72 Contention between DCNT Write and Underflow

clearing to 0 by the underflow has priority, and the corresponding bit of DSTR is not set to 1.

The timing in this case is shown in figure 11.73.

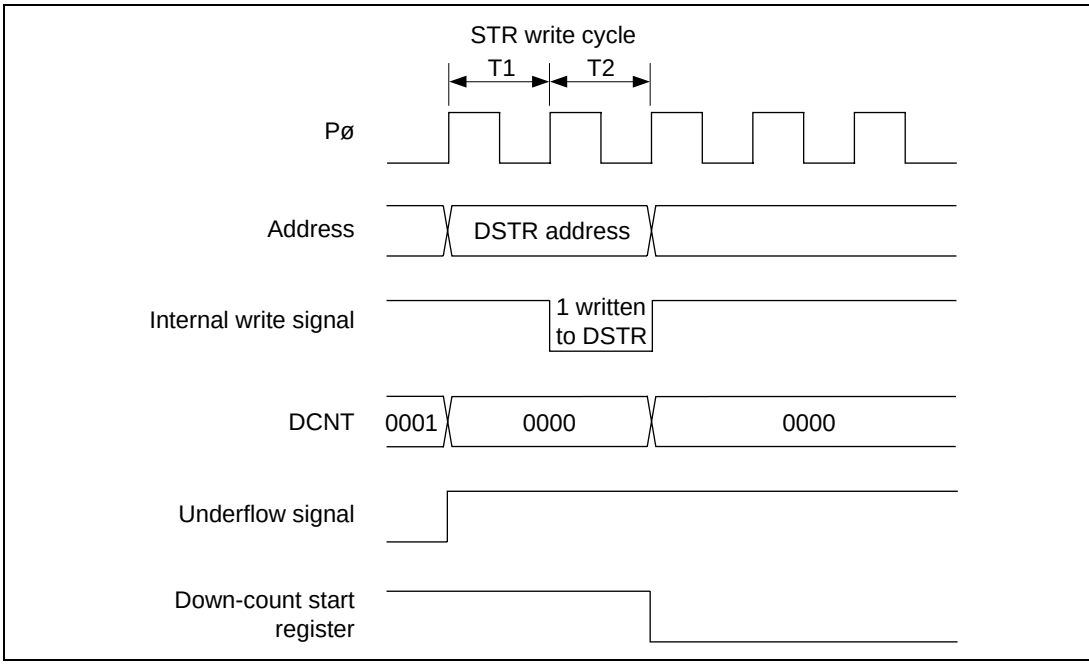


Figure 11.73 Contention between DSTR Bit Setting by CPU and Clearing by Underflow

(TCR), and timer mode register (TMDR) should be made before the counter is started. Operation is not guaranteed if these registers are modified while the counter is running.

Also, the counter must not be started until Pø has been input 32 times after setting PSCR1 to PSCR4.

Interrupt Status Flag Clearing Procedure: When an interrupt status flag is cleared to 0 by the CPU, it must first be read before 0 is written to it. Correct operation cannot be guaranteed if 0 is written without first reading the flag.

Setting H'0000 in Free-Running Counters 6A to 6D, 7A to 7D (TCNT6A to TCNT6D, TCNT7A to TCNT7D): If H'0000 is written to a channel 6 and 7 free-running counter (TCNT6A to TCNT6D, TCNT7A to TCNT7D), and the counter is started, the interval up to the first compare-match with the cycle register (CYLR) and duty register (DTR) will be a maximum of one TCNT input clock cycle longer than the set value. With subsequent compare-matches, the correct waveform will be output for the CYLR and DTR values.

Register Values when a Free-Running Counter (TCNT) Halts: If the timer start register (TSTR) value is set to 0 during counter operation, only incrementing of the corresponding free-running counter (TCNT) is stopped, and neither the free-running counter (TCNT) nor any other ATU registers are initialized. The external output value at the time TSTR is cleared to 0 will continue to be output.

TCNT0 Writing and Interval Timer Operation: If the CPU program writes 1 to a bit in free-running counter 0 (TCNT0) corresponding to a bit set to 1 in the interval interrupt request register (ITVRR) when that TCNT0 bit is 0, TCNT0 bit 6, 7, 8, 9, 10, 11, 12, or 13 will be detected as having changed from 0 to 1, and an interrupt request will be sent to INTC and A/D sampling will be started. While the count is halted with the STR0 bit cleared to 0 in timer start register 1 (TSTR1), the bit transition from 0 to 1 will still be detected.

Automatic TSR Clearing by DMAC Activation by the ATU: Automatic clearing of TSR is performed after completion of the transfer when the DMAC is in burst mode, and each time the DMAC returns the bus in cycle steal mode.

Interrupt Status Flag Setting/Resetting: With TSR, a 0 write to a bit is possible even if overlapping events occur for the same bit before writing 0 after reading 1 to clear that bit. (The duplicate events are not accepted.)

TIO1A to TIO1H, TIO2A to TIO2H, TIO11A, and TIO11B external output values are cleared to 0 immediately after software standby mode is exited, other external output values and all registers are cleared to 0 immediately after a transition to software standby mode.

Also, when pin output is inverted by the pin function controller's port B invert register (PBIR) or port K invert register (PKIR), the corresponding pins are set to 1.

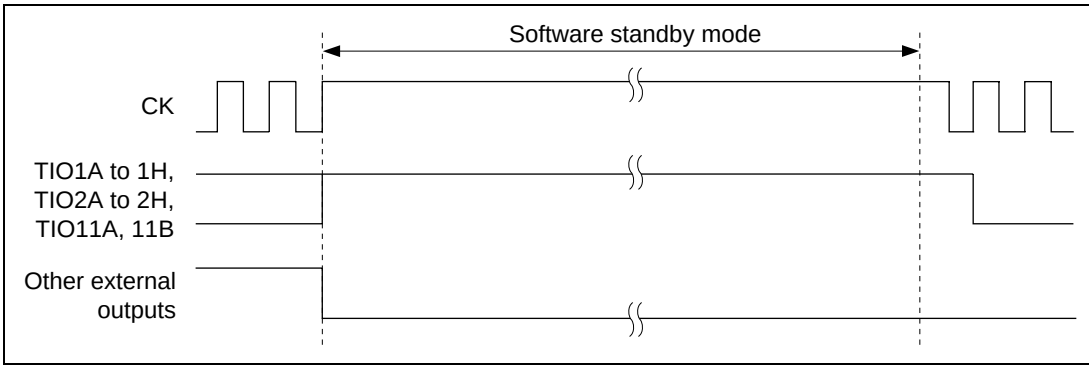


Figure 11.74 External Output Value Transition Points in Relation to Software Standby Mode

Contention between TCNT Clearing from Channel 10 and TCNT Overflow: When a channel 1 or 2 free-running counter (TCNT1A, TCNT1B, TCNT2A, TCNT2B) overflows, it is cleared to H'0000. If a clear signal from the channel 10 correction counter clear register (TCCLR) is input at the same time, setting 1 to the overflow interrupt status flag (OVF) due to the overflow is still performed in the same way as for a normal overflow.

Contention between Channel 10 Reload Register Transfer Timing and Write: If there is contention between a multiplied-output transfer from the input capture register (ICR10A) to the channel 10 reload register (RLDR10C), and the timing of a CPU write to that register, the CPU write has priority and the multiplied output is ignored.

Contention between Channel 10 Reload Timing and Write to TCNT10C: If there is contention between a multiplied-output transfer from the input capture register (ICR10A) to the channel 10 reload register (RLDR10C), and a CPU write to the reload counter (TCNT10C), the CPU write has priority and the multiplied output is ignored.

When using a port for input capture input, the corresponding TIOR register must be in the input capture disabled state when the port is set. Regarding channel 10 TI10 input, TCR10 must be in the TI10 input disabled state when the port is set. When using a port for external clock input, the STR bit for the corresponding channel must be in the count operation disabled state when the port is set. When using a port for event input, the corresponding TCR register must be in the count operation disabled state when the port is set.

Regarding TCLKB and TI10 input, although input is assigned to a number of pins, when using TCLKB and TI10 input, only one pin should be enabled.

Writing to ROM Area Immediately after ATU Register Write: If a write cycle for a ROM address for which address bit 11 = 0 and address bit 12 = 1 (H'00001000 to H'000017FF, H'00003000 to H'000037FF, H'00005000 to H'000057FF, ..., H'0007F000 to H'0007F7FF, ..., H'000FF000 to H'000FF7FF) occurs immediately after an ATU register write cycle, the value, or part of the value, written to ROM will be written to the ATU register. The following measures should be taken to prevent this.

- Do not perform a CPU write to a ROM address immediately after an ATU register write cycle. For example, an instruction arrangement in which an MOV instruction that writes to the ATU is located at an even-word address (4n address), and is immediately followed by an MOV instruction that writes to a ROM area, will meet the bug conditions.
- Do not perform an AUD write to any of the above ROM addresses immediately after an ATU register write cycle. For example, in the case of a write to overlap RAM when using the RAM emulation function, the write should be performed to the on-chip RAM area address, not the overlapping ROM area address.
- Do not perform a DMAC write to an ATU register when a ROM address write operation occurs.

Table 11.4 ATU-II Registers and Pins

Register Name*[1]	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11
TSTR (3)	TSTR1	TSTR1	TSTR1	TSTR1	TSTR1	TSTR1	TSTR2	TSTR2	—	—	TSTR1	TSTR3
PSCR (4)	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1	PSCR2	PSCR3	PSCR1	—	PSCR4	PSCR1
TCNT (25)	TCNT0H, TCNT0L	TCNT1A, TCNT1B	TCNT2A, TCNT2B	TCNT3	TCNT4	TCNT5	TCNT6A to TCNT6D	TCNT7A to TCNT7D	—	—	TCNT10AH, TCNT10AL, TCNT10B to TCNT10H	TCNT11
DCNT (16)	—	—	—	—	—	—	—	—	DCNT8A to DCNT8P	—	—	—
ECNT (6)	—	—	—	—	—	—	—	—	—	ECNT9A to ECNT9F	—	—
TCR (17)	—	TCR1A, TCR1B	TCR2A, TCR2B	TCR3	TCR4	TCR5	TCR6A, TCR6B	TCR7A, TCR7B	TCR8	TCR9A to TCR9C	TCR10	TCR11
TIOR (17)	TIOR0	TIOR1A to TIOR1D	TIOR2A to TIOR2D	TIOR3A, TIOR3B	TIOR4A, TIOR4B	TIOR5A, TIOR5B	—	—	—	—	TIOR10	TIOR11
TSR (12)	TSR0	TSR1A, TSR1B	TSR2A, TSR2B	TSR3	TSR3	TSR3	TSR6	TSR7	TSR8	TSR9	TSR10	TSR11
TIER (12)	TIER0	TIER1A, TIER1B	TIER2A, TIER2B	TIER3	TIER3	TIER3	TIER6	TIER7	TIER8	TIER9	TIER10	TIER11
ITVRR (3)	ITVRR1, ITVRR2A, ITVRR2B	—	—	—	—	—	—	—	—	—	—	—
GR (37)	—	GR1A to GR1H	GR2A to GR2H	GR3A to GR3D	GR4A to GR4D	GR5A to GR5D	—	—	—	GR9A to GR9F	GR10G	GR11A, GR11B
ICR (5)	ICR0AH, ICR0AL to ICR0DH, ICR0DL	—	—	—	—	—	—	—	—	—	ICR10AH, ICR10AL	—
OCR (11)	—	OCR1	OCR2A to OCR2H	—	—	—	—	—	—	—	OCR10AH, OCR10AL, OCR10B	—
OSBR (2)	—	OSBR1	OSBR2	—	—	—	—	—	—	—	—	—
TRGMDR (1)	—	TRGMDR	—	—	—	—	—	—	—	—	—	—
TMDR (1)	—	—	—	TMDR	TMDR	TMDR	—	—	—	—	—	—

Register Name*[1]	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11
CYLR (8)	—	—	—	—	—	—	CYLR6A to CYLR6D	CYLR7A to CYLR7D	—	—	—	—
BFR (8)	—	—	—	—	—	—	BFR6A to BFR6D	BFR7A to BFR7D	—	—	—	—
DTR (8)	—	—	—	—	—	—	DTR6A to DTR6D	DTR7A to DTR7D	—	—	—	—
PMDR (1)	—	—	—	—	—	—	PMDR	—	—	—	—	—
RLDR (1)	—	—	—	—	—	—	—	—	RLDR	—	—	—
TCNR (1)	—	—	—	—	—	—	—	—	TCNR	—	—	—
OTR (1)	—	—	—	—	—	—	—	—	OTR	—	—	—
DSTR (1)	—	—	—	—	—	—	—	—	DSTR	—	—	—
RLDENR (1)	—	—	—	—	—	—	—	—	RLDENR	—	—	—
RLD (1)	—	—	—	—	—	—	—	—	—	—	RLD10C	—
NCR (1)	—	—	—	—	—	—	—	—	—	—	NCR10	—
TCCLR (1)	—	—	—	—	—	—	—	—	—	—	TCCLR10	—
Pins*[2]	TI0A to D	TIO1A to H, TCLKA, TCLKB	TIO2A to H, TCLKA, TCLKB	TIO3A to D, TCLKA, TCLKB	TIO4A to D, TCLKA, TCLKB	TIO5A to D, TCLKA, TCLKB	TO6A to D	TO7A to D	TO8A to P	TI9A to F	T10	TIO11A, TIO11B, TCLKA, TCLKB

Notes: *1 Figures in parentheses show the number of registers. A 32-bit register is shown as a single register.

*2 Pin functions should be set as described in section 20, Pin Function Controller (PFC).

12.1 Overview

The SH7055SF has an on-chip advanced pulse controller (APC) that can generate a maximum of eight pulse outputs, using the advanced timer unit II (ATU-II) as the time base.

12.1.1 Features

The features of the APC are summarized below.

- Maximum eight pulse outputs
  The pulse output pins can be selected from among eight pins. Multiple settings are possible.
- Output trigger provided by advanced timer unit II (ATU-II) channel 2
  Pulse 0 output and 1 output is performed using the compare-match signal generated by the ATU-II channel II compare-match register as the trigger.

Figure 12.1 shows a block diagram of the advanced pulse controller.

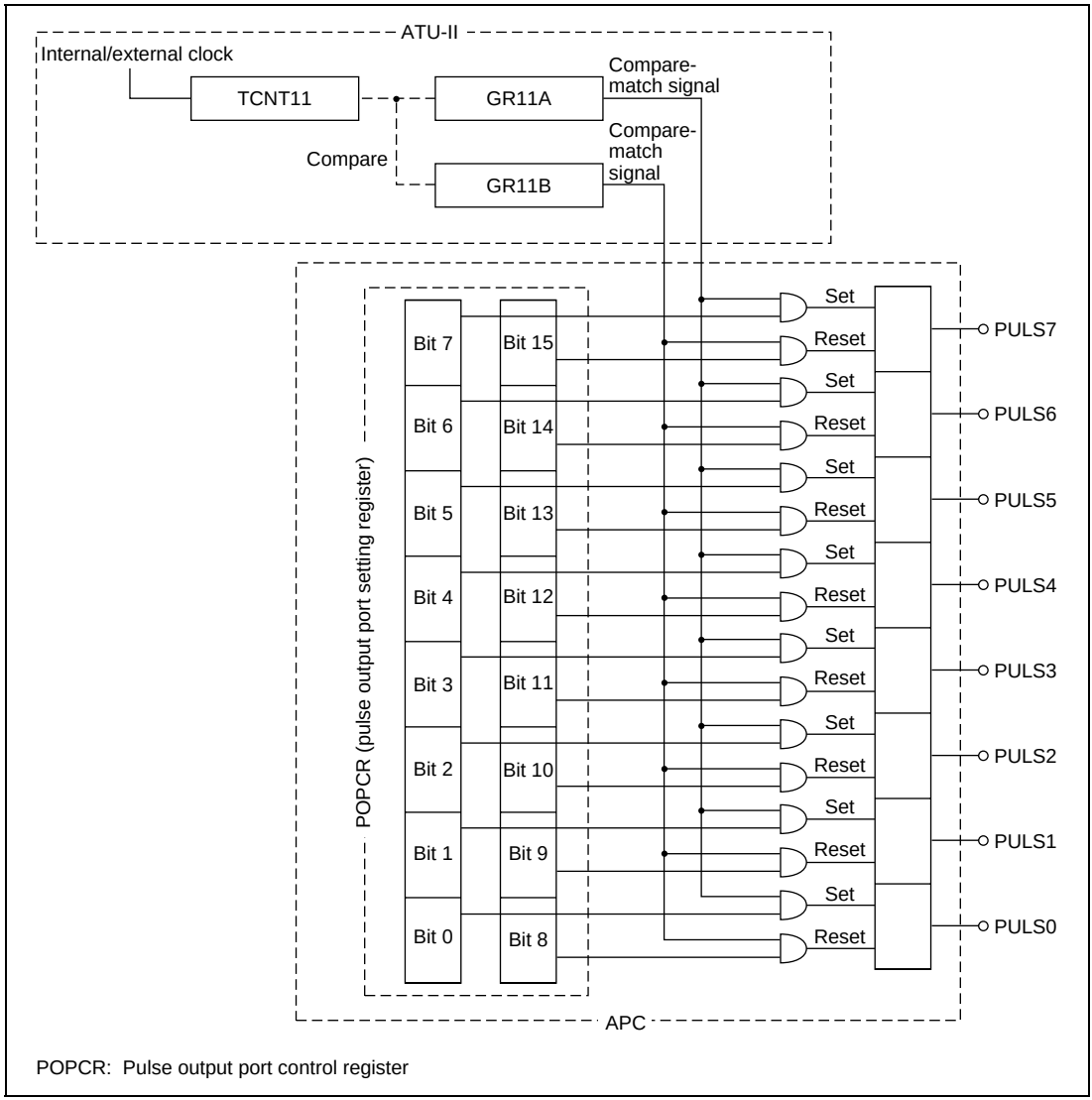


Figure 12.1 Advanced Pulse Controller Block Diagram

Table 12.1 summarizes the advanced pulse controller's output pins.

Table 12.1 Advanced Pulse Controller Pins

Pin Name	I/O	Function
PULS0	Output	APC pulse output 0
PULS1	Output	APC pulse output 1
PULS2	Output	APC pulse output 2
PULS3	Output	APC pulse output 3
PULS4	Output	APC pulse output 4
PULS5	Output	APC pulse output 5
PULS6	Output	APC pulse output 6
PULS7	Output	APC pulse output 7

12.1.4 Register Configuration

Table 12.2 summarizes the advanced pulse controller's register.

Table 12.2 Advanced Pulse Controller Register

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Pulse output port control register	POPCR	R/W	H'0000	H'FFFFF700	8, 16

Note: Register access requires 4 or 5 cycles.

12.2.1 Pulse Output Port Control Register (POPCR)

The pulse output port control register (POPCR) is a 16-bit readable/writable register.

POPCR is initialized to H'0000 by a power-on reset and in hardware standby mode. It is not initialized in software standby mode.

Bit:	15	14	13	12	11	10	9	8
	PULS7 ROE	PULS6 ROE	PULS5 ROE	PULS4 ROE	PULS3 ROE	PULS2 ROE	PULS1 ROE	PULS0 ROE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PULS7 SOE	PULS6 SOE	PULS5 SOE	PULS4 SOE	PULS3 SOE	PULS2 SOE	PULS1 SOE	PULS0 SOE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 8—PULS7 to PULS0 Reset Output Enable (PULS7ROE to PULS0ROE): These bits enable or disable 0 output to the APC pulse output pins (PULS7 to PULS0) bit by bit.

Bits 15 to 8:

PULS7ROE to PULS0ROE	Description
0	0 output to APC pulse output pin (PULS7–PULS0) is disabled (Initial value)
1	0 output to APC pulse output pin (PULS7–PULS0) is enabled

When one of these bits is set to 1, 0 is output from the corresponding pin on a compare-match between the GR11B and TCNT11 values.

Bits 7 to 0: PULS7SOE to PULS0SOE	Description
0	1 output to APC pulse output pin (PULS7–PULS0) is disabled (Initial value)
1	1 output to APC pulse output pin (PULS7–PULS0) is enabled

When one of these bits is set to 1, 1 is output from the corresponding pin on a compare-match between the GR11A and TCNT11 values.

12.3 Operation

12.3.1 Overview

APC pulse output is enabled by designating multiplex pins for APC pulse output with the pin function controller (PFC), and setting the corresponding bits to 1 in the pulse output port control register (POPCR).

When general register IIA (GRIIA) in the advanced timer unit II (ATU-II) subsequently generates a compare-match signal, 1 is output from the pins set to 1 by bits 7 to 0 in POPCR. When general register 11B (GR11B) generates a compare-match signal, 0 is output from the pins set to 1 by bits 15 to 8 in POPCR.

0 is output from the output-enabled state until the first compare-match occurs.

The advanced pulse controller output operation is shown in figure 12.2.

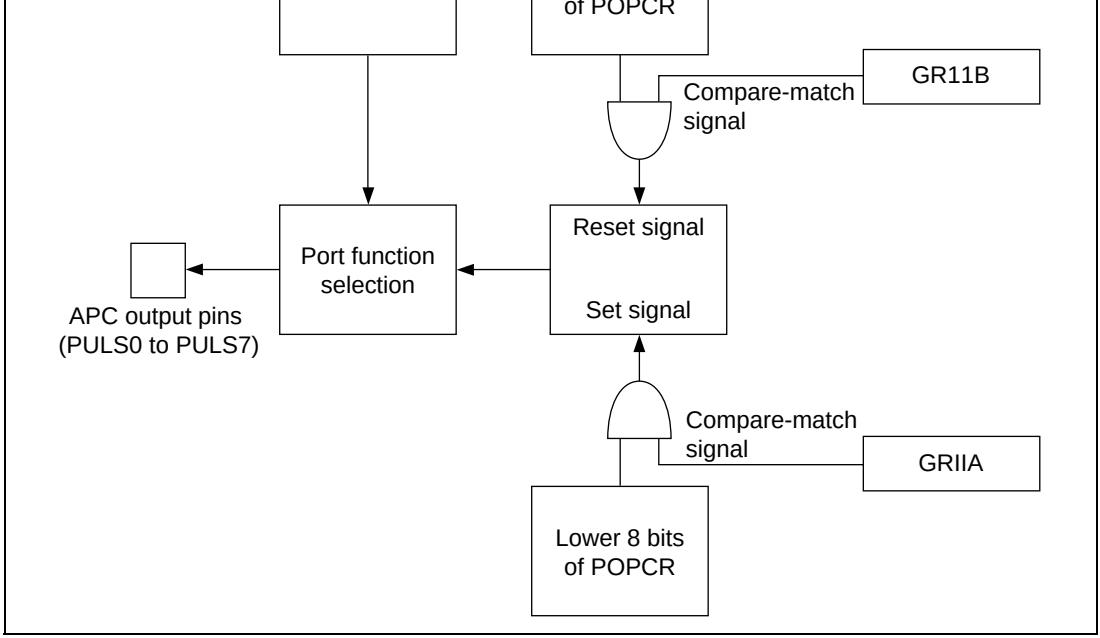


Figure 12.2 Advanced Pulse Controller Output Operation

12.3.2 Advanced Pulse Controller Output Operation

Example of Setting Procedure for Advanced Pulse Controller Output Operation: Figure 12.3 shows an example of the setting procedure for advanced pulse controller output operation.

1. Set general registers GR11A and GR11B as output compare registers with the timer I/O control register (TIOR).
2. Set the pulse rise point with GR11A and the pulse fall point with GR11B.
3. Select the timer counter 11 (TCNT11) counter clock with the timer prescale register (PSCR). TCNT11 can only be cleared by an overflow.
4. Enable the respective interrupts with the timer interrupt enable register (TIER).
5. Set the pins for 1 output and 0 output with POPCR.
6. Set the control register for the port to be used by the APC to the APC output pin function.
7. Set the STR bit to 1 in the timer start register (TSTR) to start timer counter 11 (TCNT11).
8. Each time a compare-match interrupt is generated, update the GR value and set the next pulse output time.
9. Each time a compare-match interrupt is generated, update the POPCR value and set the next pin for pulse output.

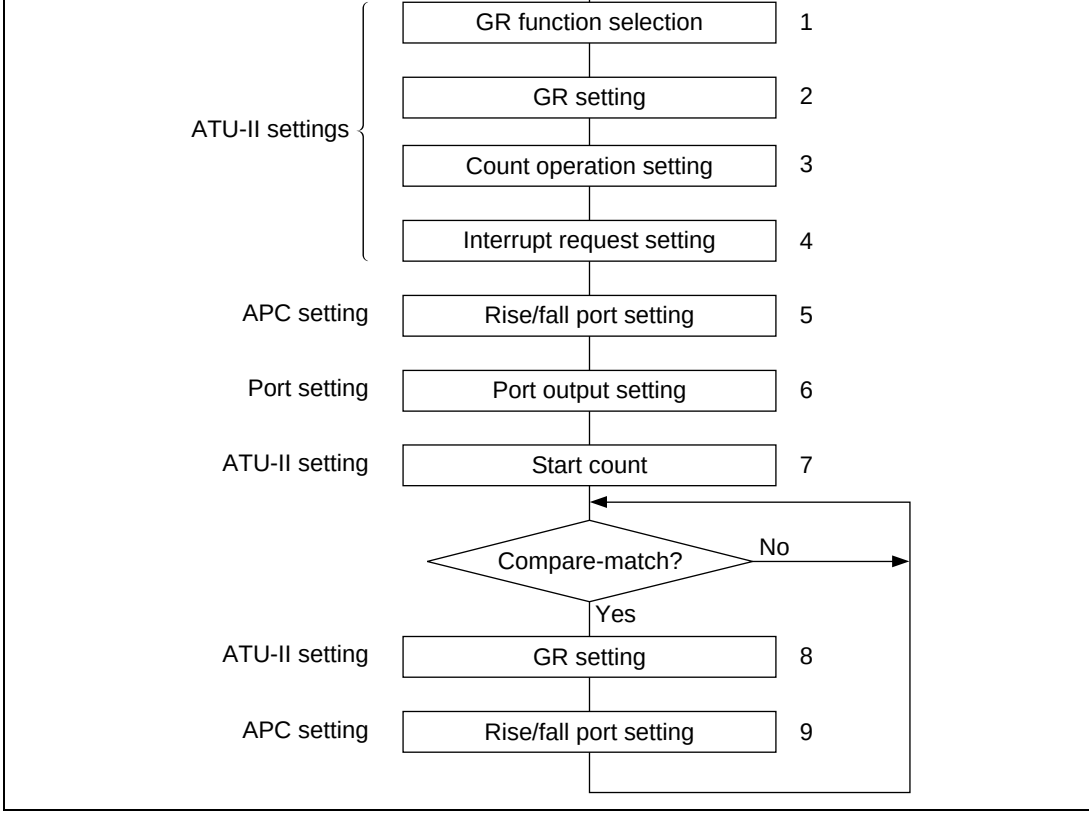


Figure 12.3 Example of Setting Procedure for Advanced Pulse Controller Output Operation

1. Set ATU-11 registers GR11A and GR11B (to be used for output trigger generation) as output compare registers. Set the rise point in GR11A and the fall point in GR11B, and enable the respective compare-match interrupts.
2. Write H'0101 to POPCR.
3. Start the TCNT11 count, when a GR11A compare-match occurs, 1 is output from the PULS0 pin. When a GR11B compare-match occurs, 0 is output from the PULS0 pin.
4. Pulse output widths and output pins can be continually changed by successively rewriting GR11A, GR11B, and POPCR in response to compare-match interrupts.
5. By setting POPCR to a value such as H'E0E0, pulses can be output from up to 8 pins in response to a single compare-match.

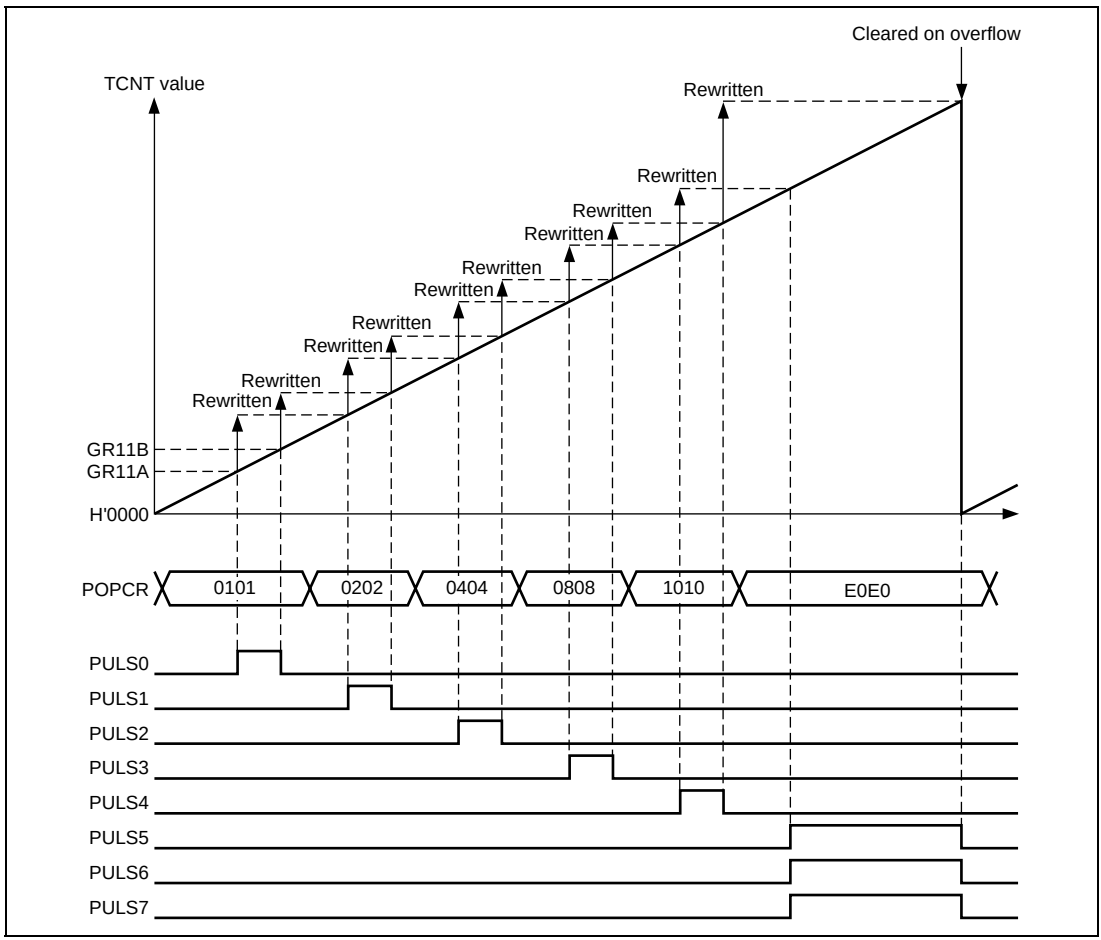


Figure 12.4 Example of Advanced Pulse Controller Output Operation

Contention between Compare-Match Signals: If the same value is set for both GR11A and GR11B, and 0 output and 1 output are both enabled for the same pin by the POPCR settings, 0 output has priority on pins PULS0 to PULS7 when compare-matches occur.

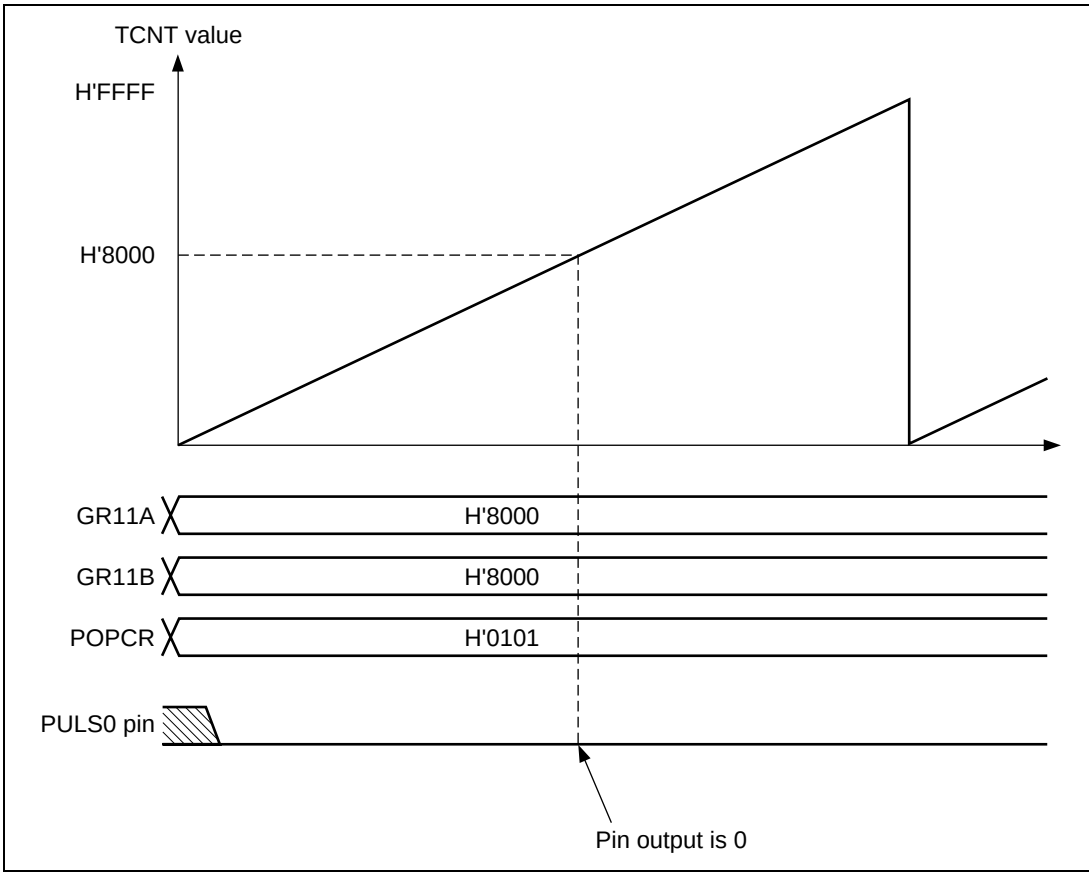


Figure 12.5 Example of Compare-Match Contention

13.1 Overview

The watchdog timer (WDT) is a 1-channel timer for monitoring system operations. If a system encounters a problem (crashes, for example) and the timer counter overflows without being rewritten correctly by the CPU, an overflow signal ($\overline{\text{WDTOVF}}$) is output externally. The WDT can simultaneously generate an internal reset signal for the entire chip.

When the watchdog function is not needed, the WDT can be used as an interval timer. In the interval timer operation, an interval timer interrupt is generated at each counter overflow.

13.1.1 Features

The WDT has the following features:

- Works in watchdog timer mode or interval timer mode
- Outputs $\overline{\text{WDTOVF}}$ in watchdog timer mode
  When the counter overflows in watchdog timer mode, overflow signal $\overline{\text{WDTOVF}}$ is output externally. It is possible to select whether to reset the chip internally when this happens. Either the power-on reset or manual reset signal can be selected as the internal reset signal.
- Generates interrupts in interval timer mode
  When the counter overflows, it generates an interval timer interrupt.
- Works with eight counter input clocks

Figure 13.1 is the block diagram of the WDT.

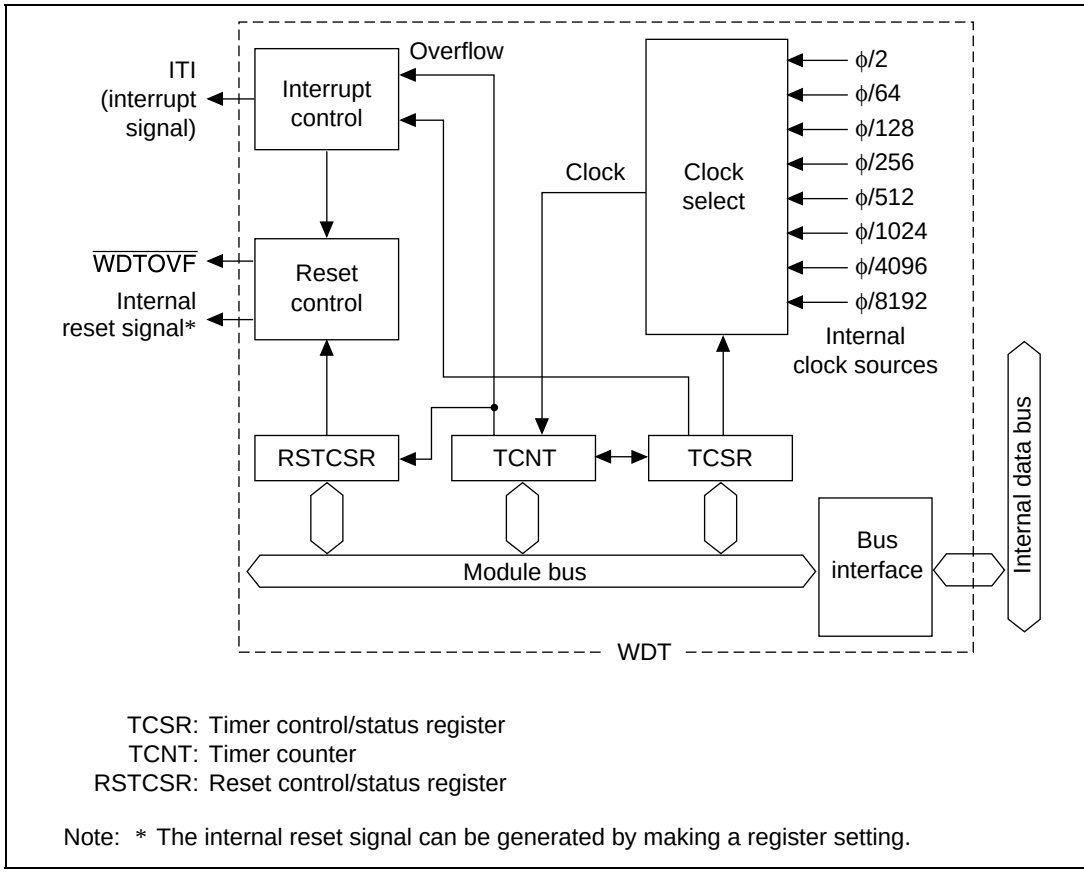


Figure 13.1 WDT Block Diagram

13.1.3 Pin Configuration

Table 13.1 shows the pin configuration.

Table 13.1 Pin Configuration

Pin	Abbreviation	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	O	Outputs the counter overflow signal in watchdog timer mode

Table 13.2 summarizes the three WDT registers. They are used to select the clock, switch the WDT mode, and control the reset signal.

Table 13.2 WDT Registers

Name	Abbreviation	R/W	Initial Value	Address	
				Write*[1]	Read*[2]
Timer control/status register	TCSR	R/(W)*[3]	H'18	H'FFFFEC10	H'FFFFEC10
Timer counter	TCNT	R/W	H'00		H'FFFFEC11
Reset control/status register	RSTCSR	R/(W) *[3]	H'1F	H'FFFFEC12	H'FFFFEC13

Notes: In register access, three cycles are required for both byte access and word access.
*1 Write by word transfer. These registers cannot be written in byte or longword.
*2 Read by byte transfer. These registers cannot be read in word or longword.
*3 Only 0 can be written to bit 7 to clear the flag.

13.2 Register Descriptions

13.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable upcounter. (TCNT differs from other registers in that it is more difficult to write to. See section 13.2.4, Register Access, for details.) When the timer enable bit (TME) in the timer control/status register (TCSR) is set to 1, the watchdog timer counter starts counting pulses of an internal clock selected by clock select bits 2 to 0 (CKS2 to CKS0) in TCSR. When the value of TCNT overflows (changes from H'FF to H'00), a watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) or interval timer interrupt (ITI) is generated, depending on the mode selected in the WT/$\overline{\text{IT}}$ bit of TCSR.

TCNT is initialized to H'00 by a power-on reset, in hardware and software standby modes, and when the TME bit is cleared to 0.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The timer control/status register (TCSR) is an 8-bit readable/writable register. (TCSR differs from other registers in that it is more difficult to write to. See section 13.2.4, Register Access, for details.) TCSR performs selection of the timer counter (TCNT) input clock and mode.

TCSR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)*	R/W	R/W	R	R	R/W	R/W	R/W

Note: * The only operation permitted on the OVF bit is a write of 0 after reading 1.

- Bit 7—Overflow Flag (OVF): Indicates that TCNT has overflowed from H'FF to H'00 in interval timer mode. This flag is not set in the watchdog timer mode.

Bit 7: OVF	**Description**
0	No overflow of TCNT in interval timer mode (Initial value) [Clearing condition] When 0 is written to OVF after reading OVF
1	TCNT overflow in interval timer mode

- Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether to use the WDT as a watchdog timer or interval timer. When TCNT overflows, the WDT either generates an interval timer interrupt (ITI) or generates a $\overline{WDTOVF}$ signal, depending on the mode selected.

Bit 6: WT/$\overline{IT}$	**Description**
0	Interval timer mode: interval timer interrupt (ITI) request to the CPU when TCNT overflows (Initial value)
1	Watchdog timer mode: $\overline{WDTOVF}$ signal output externally when TCNT overflows. (Section 13.2.3, Reset Control/Status Register (RSTCSR), describes in detail what happens when TCNT overflows in watchdog timer mode.)

0	Timer disabled: TCNT is initialized to H'00 and count-up stops (Initial value)
1	Timer enabled: TCNT starts counting. A $\overline{\text{WDTOVF}}$ signal or interrupt is generated when TCNT overflows.

- Bits 4 and 3—Reserved: These bits always read 1. The write value should always be 1.
- Bits 2 to 0: Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight internal clock sources for input to TCNT. The clock signals are obtained by dividing the frequency of the system clock (ϕ).

			Description	
Bit 2: CKS2	**Bit 1: CKS1**	**Bit 0: CKS0**	**Clock Source**	**Overflow Interval* (ϕ = 40 MHz)**
0	0	0	ϕ/2 (Initial value)	12.8 µs
0	0	1	ϕ/64	409.6 µs
0	1	0	ϕ/128	0.8 ms
0	1	1	ϕ/256	1.6 ms
1	0	0	ϕ/512	3.3 ms
1	0	1	ϕ/1024	6.6 ms
1	1	0	ϕ/4096	26.2 ms
1	1	1	ϕ/8192	52.4 ms

Note: * The overflow interval listed is the time from when the TCNT begins counting at H'00 until an overflow occurs.

RSTCSR is an 8-bit readable/writable register. (RSTCSR differs from other registers in that it is more difficult to write. See section 13.2.4, Register Access, for details.) It controls output of the internal reset signal generated by timer counter (TCNT) overflow. RSTCR is initialized to H'1F by input of a reset signal from the $\overline{\text{RES}}$ pin, but is not initialized by the internal reset signal generated by overflow of the WDT. It is initialized to H'1F in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	RSTS	—	—	—	—	—
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/(W)*	R/W	R/W	R	R	R	R	R

Note: * Only 0 can be written to bit 7 to clear the flag.

- Bit 7—Watchdog Timer Overflow Flag (WOVF): Indicates that TCNT has overflowed (H'FF to H'00) in watchdog timer mode. This flag is not set in interval timer mode.

Bit 7: WOVF	Description	
0	No TCNT overflow in watchdog timer mode	(Initial value)
	[Clearing condition]	
	When 0 is written to WOVF after reading WOVF	
1	Set by TCNT overflow in watchdog timer mode	

- Bit 6—Reset Enable (RSTE): Selects whether to reset the chip internally if TCNT overflows in watchdog timer mode.

Bit 6: RSTE	Description	
0	Not reset when TCNT overflows	(Initial value)
	LSI not reset internally, but TCNT and TCSR reset within WDT.	
1	Reset when TCNT overflows	

- Bit 5—Reset Select (RSTS): Selects the kind of internal reset to be generated when TCNT overflows in watchdog timer mode.

Bit 5: RSTS	Description	
0	Power-on reset	(Initial value)
1	Manual reset	

- Bits 4 to 0—Reserved: These bits are always read as 1. The write value should always be 1.

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in that they are more difficult to write to. The procedures for writing and reading these registers are given below.

Writing to TCNT and TCSR: These registers must be written by a word transfer instruction. They cannot be written by byte transfer instructions.

TCNT and TCSR both have the same write address. The write data must be contained in the lower byte of the written word. The upper byte must be H'5A (for TCNT) or H'A5 (for TCSR) (figure 13.2). This transfers the write data from the lower byte to TCNT or TCSR.

Figure 13.2 Writing to TCNT and TCSR

Writing to RSTCSR: RSTCSR must be written by a word access to address H'FFFFEC12. It cannot be written by byte transfer instructions.

Procedures for writing 0 to WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 13.3.

To write 0 to the WOVF bit, the write data must be H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write to the RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data. The values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respectively. The WOVF bit is not affected.

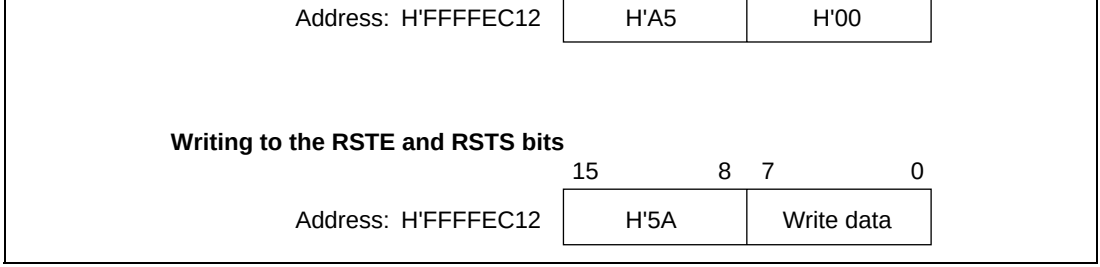


Figure 13.3 Writing to RSTCSR

Reading from TCNT, TCSR, and RSTCSR: TCNT, TCSR, and RSTCSR are read like other registers. Use byte transfer instructions. The read addresses are H'FFFFEC10 for TCSR, H'FFFFEC11 for TCNT, and H'FFFFEC13 for RSTCSR.

13.3 Operation

13.3.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ and TME bits of TCSR to 1. Software must prevent TCNT overflow by rewriting the TCNT value (normally by writing H'00) before overflow occurs. No TCNT overflows will occur while the system is operating normally, but if TCNT fails to be rewritten and overflows occur due to a system crash or the like, a $\overline{WDTOVF}$ signal is output externally (figure 13.4). The $\overline{WDTOVF}$ signal can be used to reset the system. The $\overline{WDTOVF}$ signal is output for 128 ϕ clock cycles.

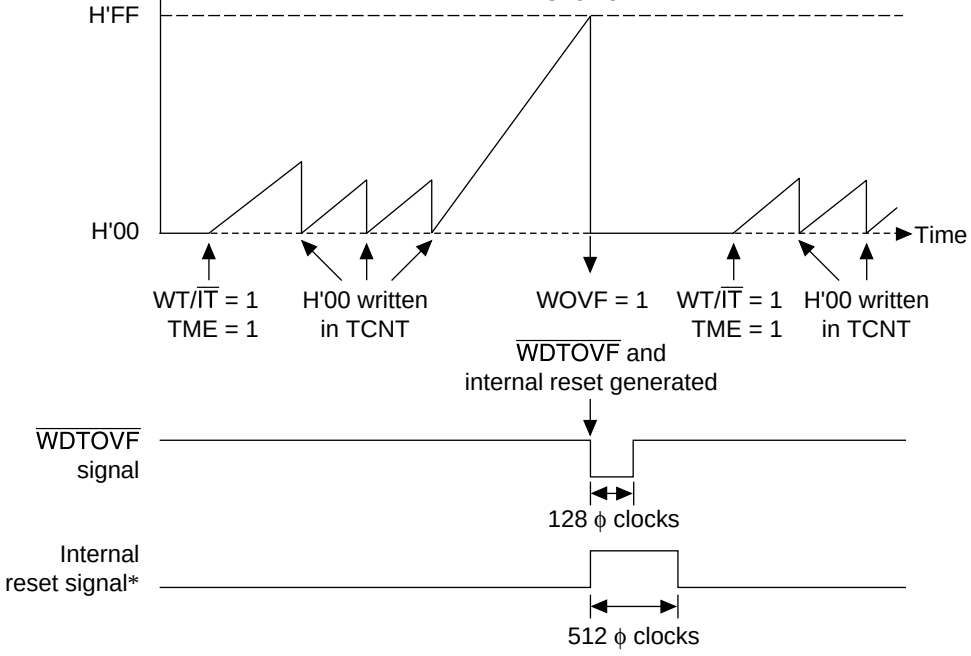
If the RSTE bit in RSTCSR is set to 1, a signal to reset the chip will be generated internally simultaneous to the $\overline{WDTOVF}$ signal when TCNT overflows. Either a power-on reset or a manual reset can be selected by the RSTS bit in RSTCSR. The internal reset signal is output for 512 ϕ clock cycles.

When a WDT overflow reset is generated simultaneously with a reset input at the $\overline{RES}$ pin, the $\overline{RES}$ reset takes priority, and the WOVF bit in RSTCSR is cleared to 0.

The following are not initialized by a WDT reset signal:

- PFC (pin function controller) registers
- I/O port registers

These registers are initialized only by an external power-on reset.



WT/IT: Timer mode select bit
TME: Timer enable bit

Note: * Internal reset signal occurs only when the RSTE bit is set to 1.

Figure 13.4 Operation in Watchdog Timer Mode

To use the WDT as an interval timer, clear WT/IT to 0 and set TME to 1 in TCSR. An interval timer interrupt (ITI) is generated each time the timer counter overflows. This function can be used to generate interval timer interrupts at regular intervals (figure 13.5).

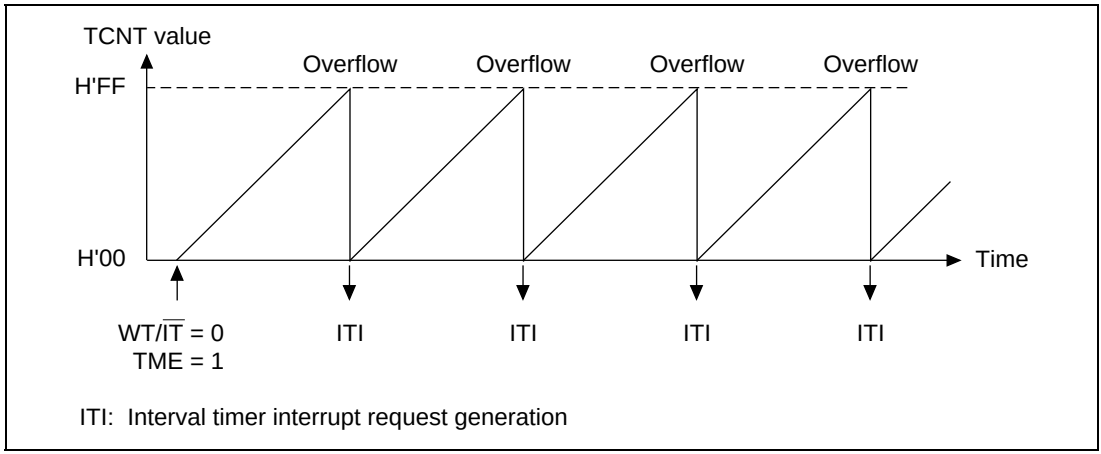


Figure 13.5 Operation in Interval Timer Mode

13.3.3 Timing of Setting the Overflow Flag (OVF)

In interval timer mode, when TCNT overflows, the OVF flag of TCSR is set to 1 and an interval timer interrupt (ITI) is simultaneously requested (figure 13.6).

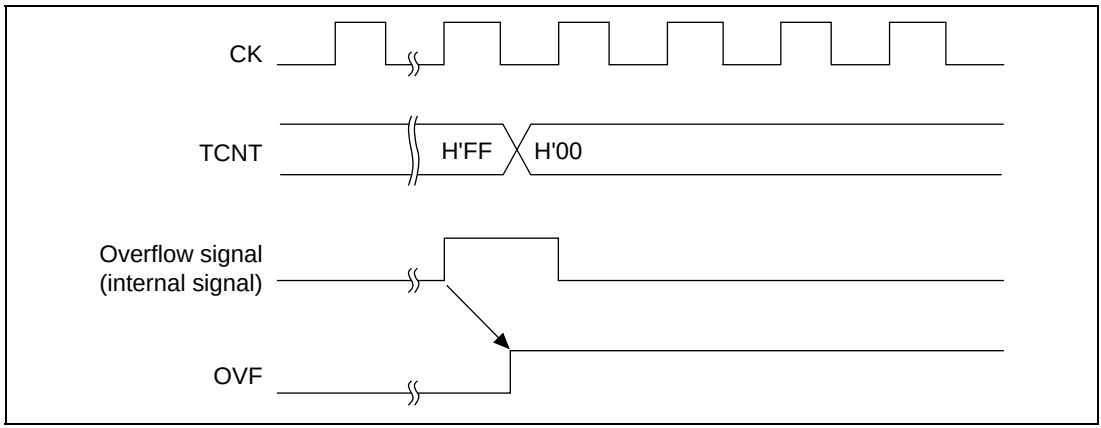


Figure 13.6 Timing of Setting OVF

which TCNT overflows in watchdog timer mode, the WOVF bit of RSTCSR is set to 1 and a $\overline{\text{WDTOVF}}$ signal is output. When the RSTE bit in RSTCSR is set to 1, TCNT overflow enables an internal reset signal to be generated for the entire chip (figure 13.7).

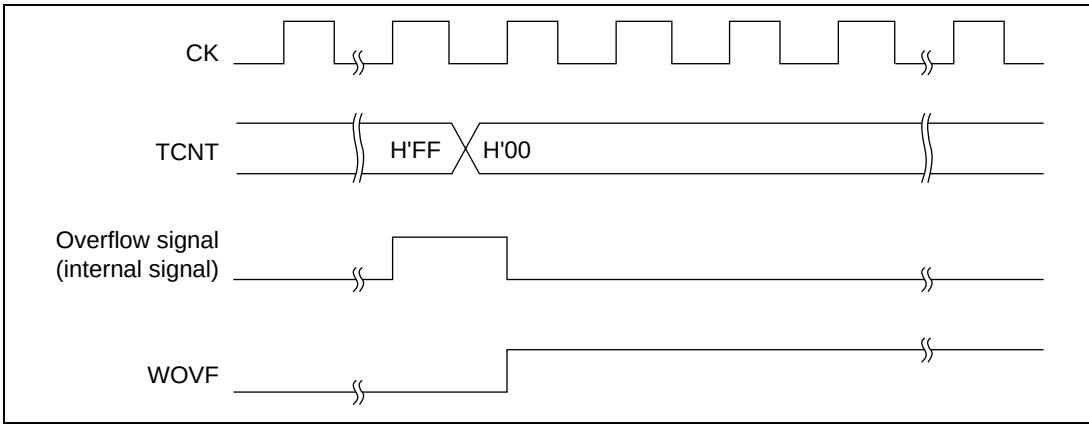


Figure 13.7 Timing of Setting WOVF

13.4.1 TCNT Write and Increment Contention

If a timer counter increment clock pulse is generated during the T3 state of a write cycle to TCNT, the write takes priority and the timer counter is not incremented (figure 13.8).

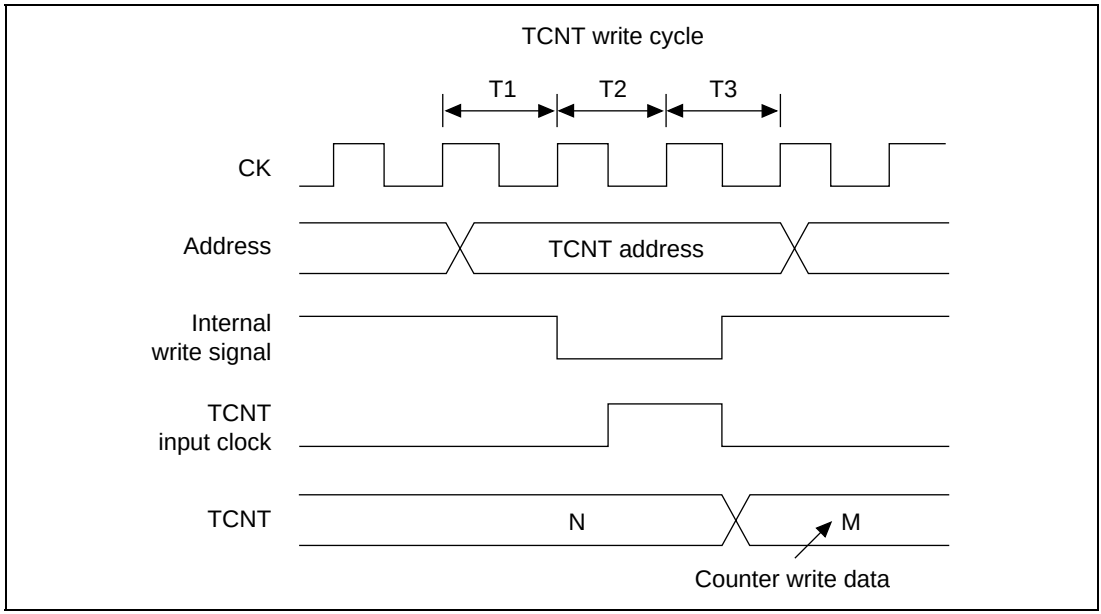


Figure 13.8 Contention between TCNT Write and Increment

13.4.2 Changing CKS2 to CKS0 Bit Values

If the values of bits CKS2 to CKS0 in the timer control/status register (TCSR) are rewritten while the WDT is running, the count may not increment correctly. Always stop the watchdog timer (by clearing the TME bit to 0) before changing the values of bits CKS2 to CKS0.

13.4.3 Changing between Watchdog Timer/Interval Timer Modes

To prevent incorrect operation, always stop the watchdog timer (by clearing the TME bit to 0) before switching between interval timer mode and watchdog timer mode.

If a WDTOVF signal is input to the RES pin, the chip cannot initialize correctly.

Avoid logical input of the $\overline{\text{WDTOVF}}$ output signal to the $\overline{\text{RES}}$ input pin. To reset the entire system with the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 13.9.

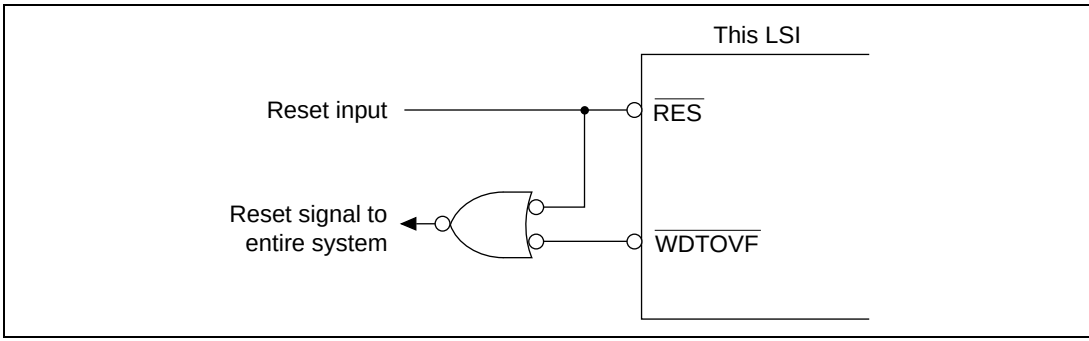


Figure 13.9 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

13.4.5 Internal Reset in Watchdog Timer Mode

If the RSTE bit is cleared to 0 in watchdog timer mode, the chip will not be reset internally when a TCNT overflow occurs, but TCNT and TCSR in the WDT will be reset.

Because the internal clock obtained by dividing the system clock(ϕ) is also reset at this time, the SCI, A/D converter, and CMT that use the internal clock may not operate correctly from hereafter. To continue using these modules, initialize them before use.

13.4.6 Manual Reset in Watchdog Timer

When an internal reset is effected by TCNT overflow in watchdog timer mode, the processor waits until the end of the bus cycle at the time of manual reset generation before making the transition to manual reset exception processing. Therefore, the bus cycle is retained in a manual reset, but if a manual reset occurs while the bus is released or during DMAC burst transfer, manual reset exception processing will be deferred until the CPU acquires the bus. However, if the interval from generation of the manual reset until the end of the bus cycle is equal to or longer than the internal manual reset interval of 512 cycles, the internal manual reset source is ignored instead of being deferred, and manual reset exception processing is not executed.

14.1 Overview

The SH7055SF has an on-chip compare match timer (CMT) comprising two 16-bit timer channels. The CMT has 16-bit counters and can generate interrupts at set intervals.

14.1.1 Features

The CMT has the following features:

- Four types of counter input clock can be selected
  - One of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, Pϕ/512) can be selected independently for each channel.
- Interrupt sources
  - A compare match interrupt can be requested independently for each channel.

Figure 14.1 shows a block diagram of the CMT.

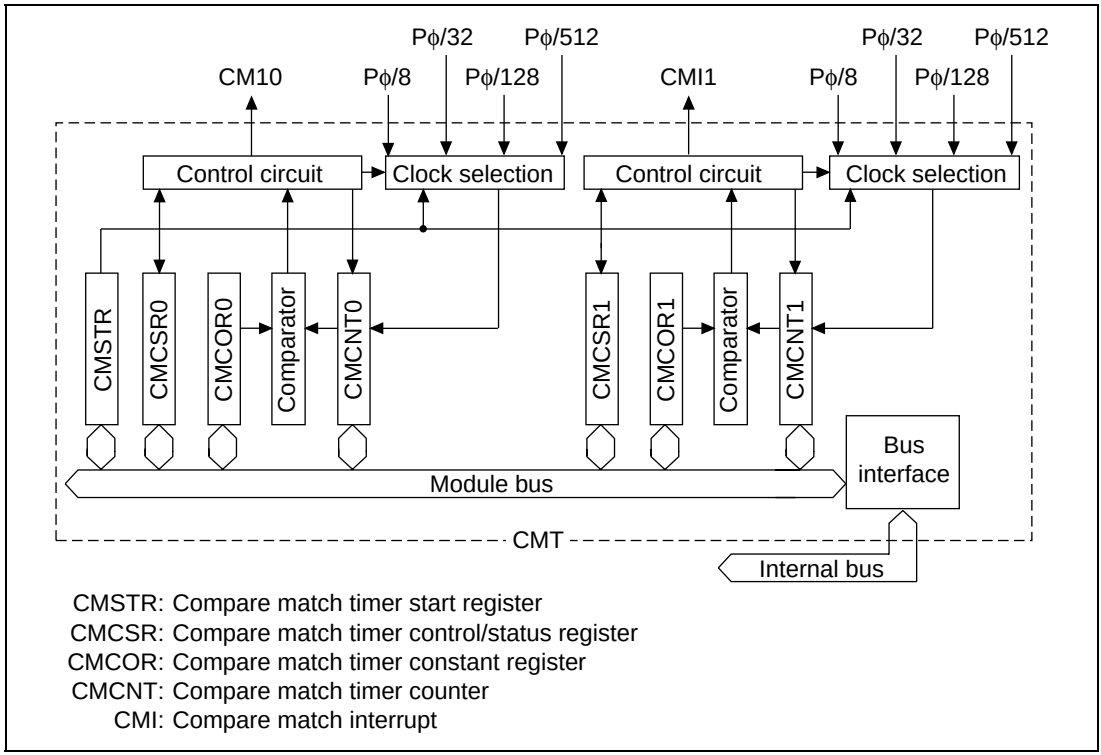


Figure 14.1 CMT Block Diagram

Table 14.1 summarizes the CMT register configuration.

Table 14.1 Register Configuration

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits)
Shared	Compare match timer start register	CMSTR	R/W	H'0000	H'FFFFF710	8, 16, 32
0	Compare match timer control/status register 0	CMCSR0	R/(W)*	H'0000	H'FFFFF712	8, 16, 32
	Compare match timer counter 0	CMCNT0	R/W	H'0000	H'FFFFF714	8, 16, 32
	Compare match timer constant register 0	CMCOR0	R/W	H'FFFF	H'FFFFF716	8, 16, 32
1	Compare match timer control/status register 1	CMCSR1	R/(W)*	H'0000	H'FFFFF718	8, 16, 32
	Compare match timer counter 1	CMCNT1	R/W	H'0000	H'FFFFF71A	8, 16, 32
	Compare match timer constant register 1	CMCOR1	R/W	H'FFFF	H'FFFFF71C	8, 16, 32

Notes: With regard to access size, four of five cycles are required for byte access and word access, and eight or nine cycles for longword access.

* Only 0 can be written to the CMCSR0 and CMCSR1 CMF bits to clear the flags.

14.2.1 Compare Match Timer Start Register (CMSTR)

The compare match timer start register (CMSTR) is a 16-bit register that selects whether to operate or halt the channel 0 and channel 1 counters (CMCNT). It is initialized to H'0000 by a power-on reset and in the standby modes.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

- Bits 15–2—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 1—Count Start 1 (STR1): Selects whether to operate or halt compare match timer counter 1.

Bit 1: STR1	Description	
0	CMCNT1 count operation halted	(Initial value)
1	CMCNT1 count operation	

- Bit 0—Count Start 0 (STR0): Selects whether to operate or halt compare match timer counter 0.

Bit 0: STR0	Description	
0	CMCNT0 count operation halted	(Initial value)
1	CMCNT0 count operation	

The compare match timer control/status register (CMCSR) is a 16-bit register that indicates the occurrence of compare matches, sets the enable/disable status of interrupts, and establishes the clock used for incrementation. It is initialized to H'0000 by a power-on reset and in the standby modes.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	CMF	CMIE	—	—	—	—	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R	R	R	R	R/W	R/W

Note: * Only 0 can be written to clear the flag.

- Bits 15–8 and 5–2—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 7—Compare Match Flag (CMF): This flag indicates whether or not the CMCNT and CMCOR values have matched.

Bit 7: CMF	Description	
0	CMCNT and CMCOR values have not matched	(Initial value)
	[Clearing condition]	
	Write 0 to CMF after reading 1 from it	
1	CMCNT and CMCOR values have matched	

- Bit 6—Compare Match Interrupt Enable (CMIE): Selects whether to enable or disable a compare match interrupt (CMI) when the CMCNT and CMCOR values have matched (CMF = 1).

Bit 6: CMIE	Description	
0	Compare match interrupt (CMI) disabled	(Initial value)
1	Compare match interrupt (CMI) enabled	

When the STR bit of CMSTR is set to 1, CMCNT begins incrementing with the clock selected by CKS1 and CKS0.

Bit 1: CKS1	Bit 0: CKS0	Description	
0	0	Pϕ/8	(Initial value)
	1	Pϕ/32	
1	0	Pϕ/128	
	1	Pϕ/512	

14.2.3 Compare Match Timer Counter (CMCNT)

The compare match timer counter (CMCNT) is a 16-bit register used as an up-counter for generating interrupt requests.

When an internal clock is selected with the CKS1 and CKS0 bits of the CMCSR register and the STR bit of CMSTR is set to 1, CMCNT begins incrementing with that clock. When the CMCNT value matches that of the compare match timer constant register (CMCOR), CMCNT is cleared to H'0000 and the CMF flag of CMCSR is set to 1. If the CMIE bit of CMCSR is set to 1 at this time, a compare match interrupt (CMI) is requested.

CMCNT is initialized to H'0000 by a power-on reset and in the standby modes. It is not initialized by a manual reset.

Bit:	15	14	13	12	11	10	9	8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The compare match timer constant register (CMCOR) is a 16-bit register that sets the period for compare match with CMCNT.

CMCOR is initialized to H'FFFF by a power-on reset and in the standby modes. It is not initialized by a manual reset.

Bit:	15	14	13	12	11	10	9	8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.3 Operation

14.3.1 Cyclic Count Operation

When an internal clock is selected with the CKS1, CKS0 bits of the CMCSR register and the STR bit of CMSTR is set to 1, CMCNT begins incrementing with the selected clock. When the CMCNT counter value matches that of the compare match constant register (CMCOR), the CMCNT counter is cleared to H'0000 and the CMF flag of the CMCSR register is set to 1. If the CMIE bit of the CMCSR register is set to 1 at this time, a compare match interrupt (CMI) is requested. The CMCNT counter begins counting up again from H'0000.

Figure 14.2 shows the compare match counter operation.

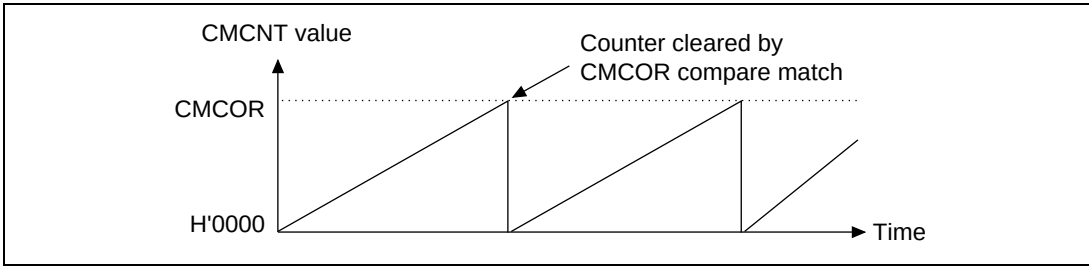


Figure 14.2 Counter Operation

One of four clocks (Pϕ/8, Pϕ/32, Pϕ/128, Pϕ/512) obtained by dividing the peripheral clock (Pϕ) can be selected by the CKS1 and CKS0 bits of CMCSR. Figure 14.3 shows the timing.

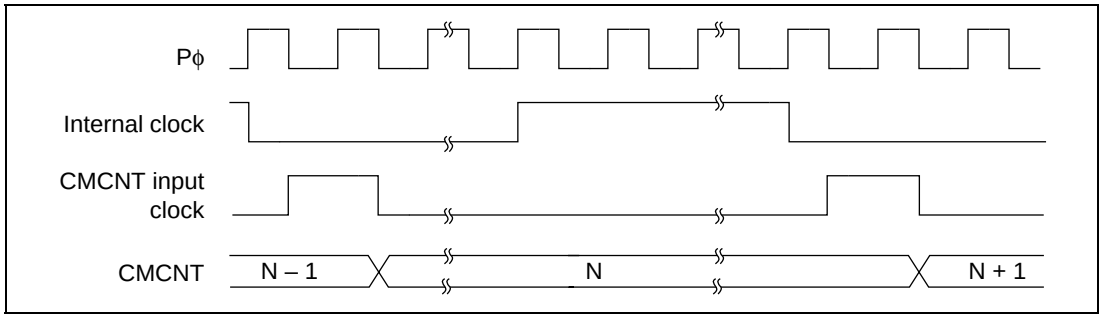


Figure 14.3 Count Timing

14.4 Interrupts

14.4.1 Interrupt Sources and DTC Activation

The CMT has a compare match interrupt for each channel, with independent vector addresses allocated to each of them. The corresponding interrupt request is output when interrupt request flag CMF is set to 1 and interrupt enable bit CMIE has also been set to 1.

When activating CPU interrupts by interrupt request, the priority between the channels can be changed by means of interrupt controller settings. See section 7, Interrupt Controller (INTC), for details.

14.4.2 Compare Match Flag Set Timing

The CMF bit of the CMCSR register is set to 1 by the compare match signal generated when the CMCOR register and the CMCNT counter match. The compare match signal is generated upon the final state of the match (timing at which the CMCNT counter matching count value is updated). Consequently, after the CMCOR register and the CMCNT counter match, a compare match signal will not be generated until a CMCNT counter input clock occurs. Figure 14.4 shows the CMF bit set timing.

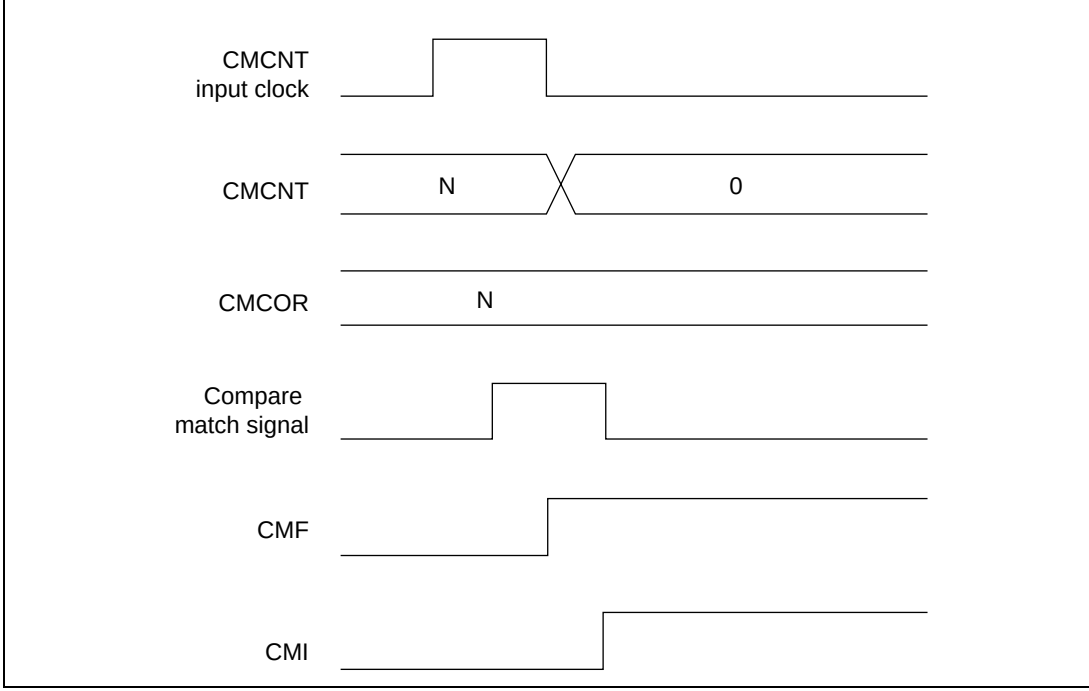


Figure 14.4 CMF Set Timing

14.4.3 Compare Match Flag Clear Timing

The CMF bit of the CMCSR register is cleared by writing a 0 to it after reading a 1. Figure 14.5 shows the timing when the CMF bit is cleared by the CPU.

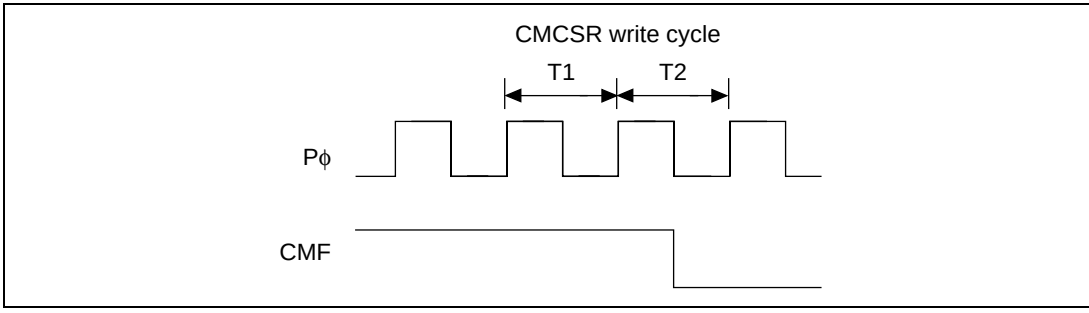


Figure 14.5 Timing of CMF Clear by the CPU

Take care that the contentions described in sections 14.5.1 to 14.5.3 do not arise during CMT operation.

14.5.1 Contention between CMCNT Write and Compare Match

If a compare match signal is generated during the T2 state of the CMCNT counter write cycle, the CMCNT counter clear has priority, so the write to the CMCNT counter is not performed. Figure 14.6 shows the timing.

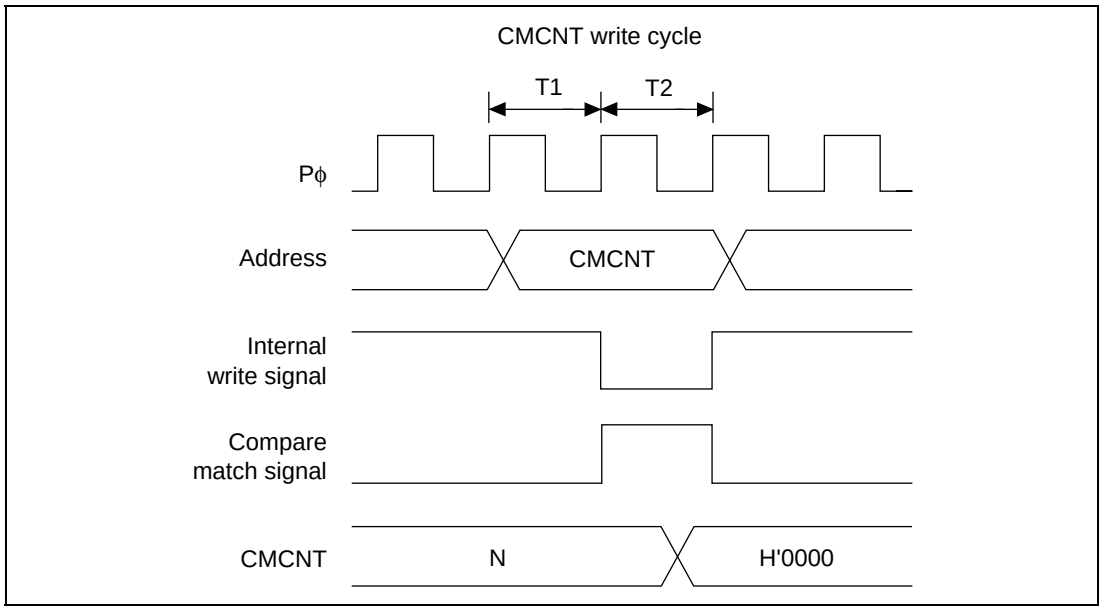


Figure 14.6 CMCNT Write and Compare Match Contention

If an increment occurs during the T2 state of the CMCNT counter word write cycle, the counter write has priority, so no increment occurs. Figure 14.7 shows the timing.

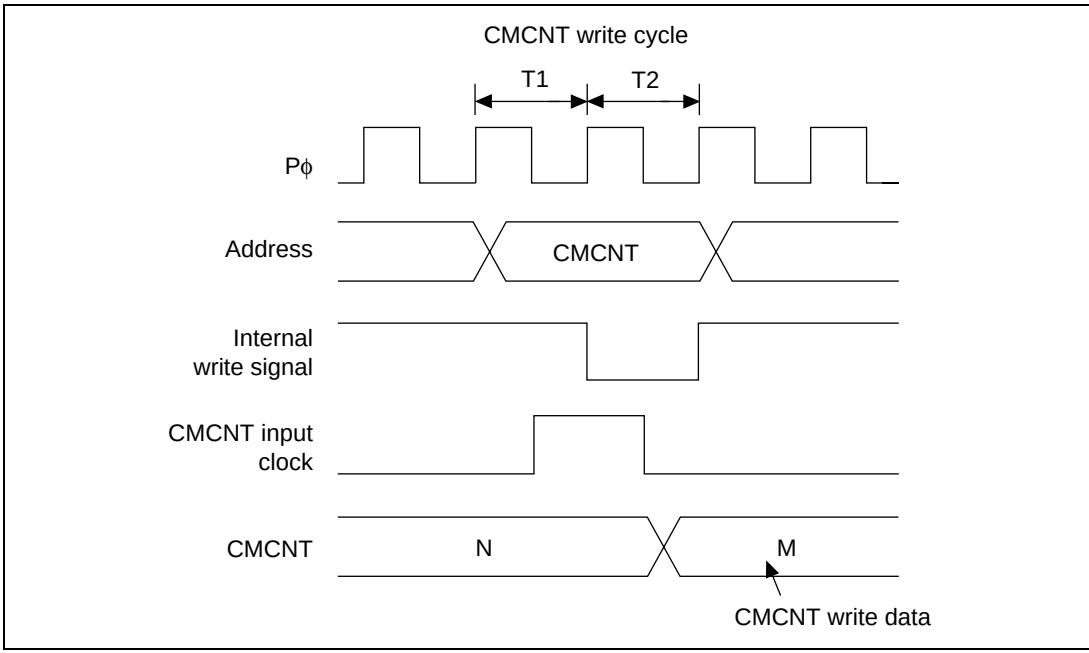


Figure 14.7 CMCNT Word Write and Increment Contention

If an increment occurs during the T2 state of the CMCNT byte write cycle, the counter write has priority, so no increment of the write data results on the side on which the write was performed. The byte data on the side on which writing was not performed is also not incremented, so the contents are those before the write.

Figure 14.8 shows the timing when an increment occurs during the T2 state of the CMCNTH write cycle.

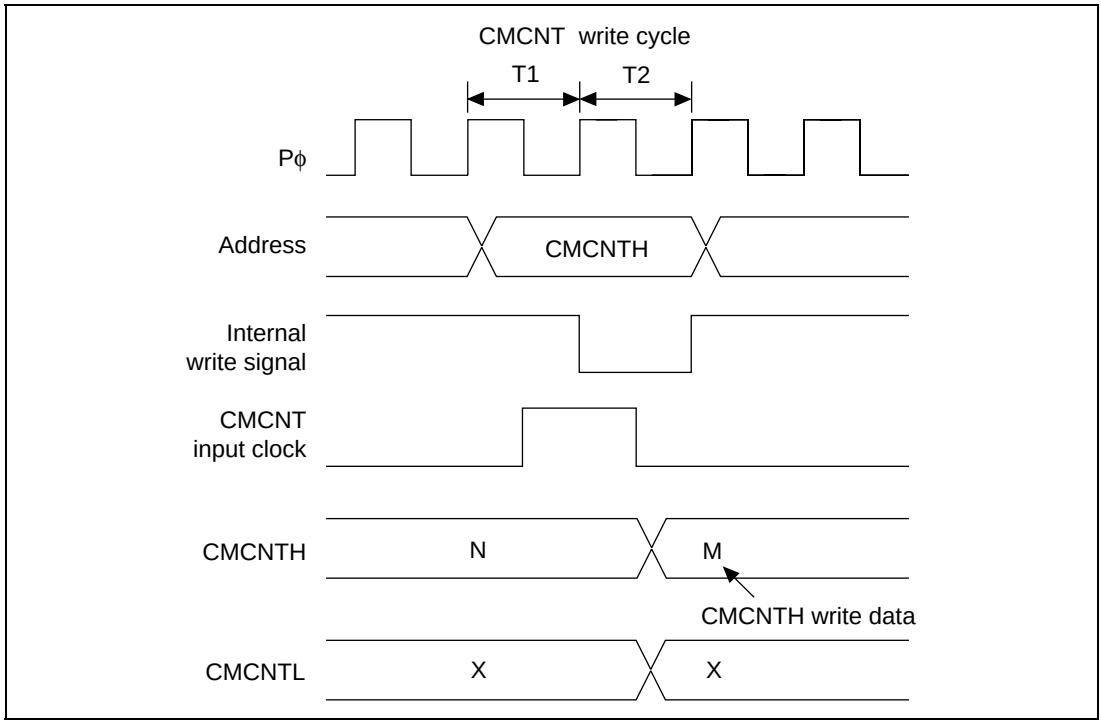


Figure 14.8 CMCNT Byte Write and Increment Contention

15.1 Overview

The SH7055SF has a serial communication interface (SCI) with five independent channels.

The SCI supports both asynchronous and synchronous serial communication. It also has a multiprocessor communication function for serial communication between two or more processors, and a clock inverted input/output function.

15.1.1 Features

The SCI has the following features:

- Selection of asynchronous or synchronous as the serial communication mode
  - — Asynchronous mode

    Serial data communication is synchronized in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other chip that employs standard asynchronous serial communication. It can also communicate with two or more other processors using the multiprocessor communication function. There are twelve selectable serial data communication formats.
    - Data length: seven or eight bits
    - Stop bit length: one or two bits
    - Parity: even, odd, or none
    - Multiprocessor bit: one or none
    - Receive error detection: parity, overrun, and framing errors
    - Break detection: by reading the RxD level directly when a framing error occurs
  - — Synchronous mode

    Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a synchronous communication function. There is one serial data communication format.
    - Data length: eight bits
    - Receive error detection: overrun errors
    - Serial clock inverted input/output
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates

- Four types of interrupts: Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently. The transmit-data-empty and receive-data-full interrupts can start the direct memory access controller (DMAC) to transfer data.
- Selection of LSB-first or MSB-first transfer (8-bit length)

  This selection is available regardless of the communication mode. (The descriptions in this section are based on LSB-first transfer.)

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the SCI.

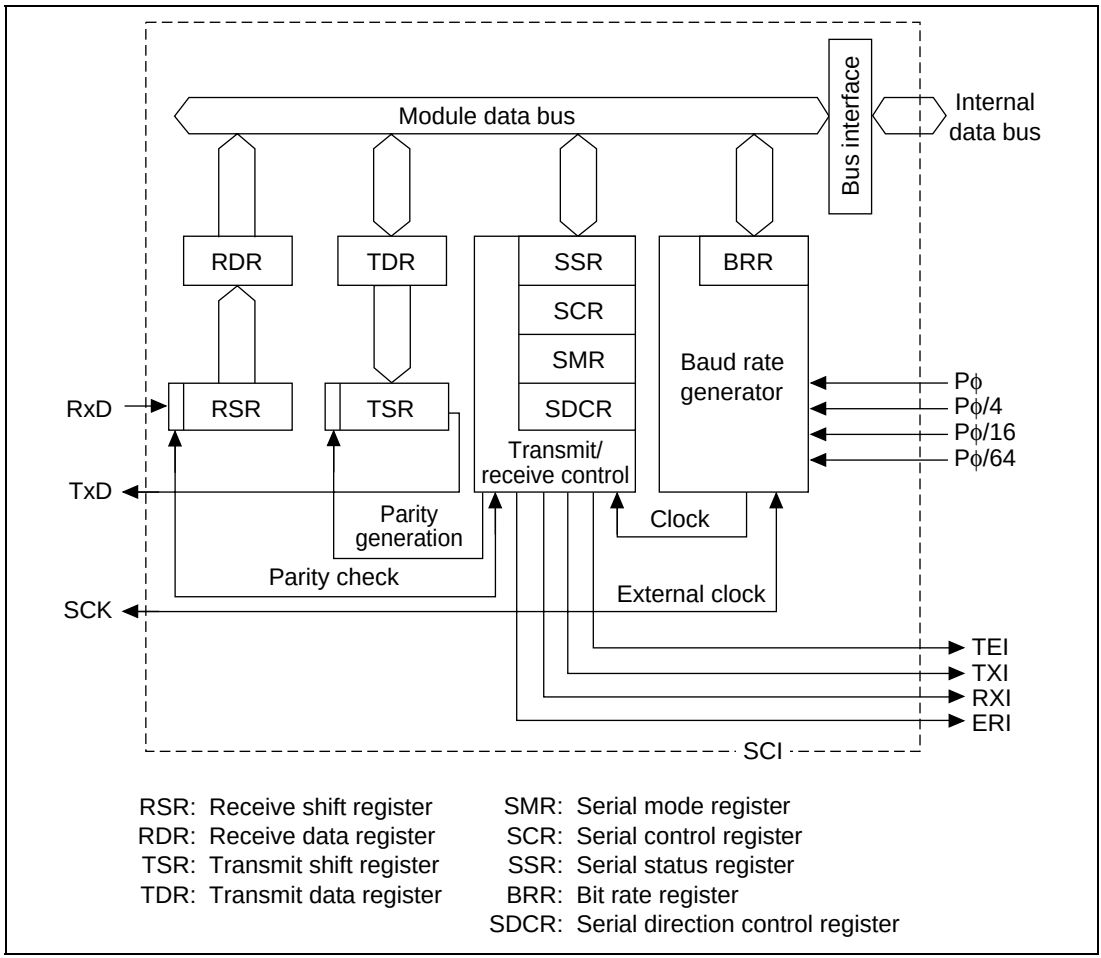


Figure 15.1 SCI Block Diagram

Table 15.1 summarizes the SCI pins by channel.

Table 15.1 SCI Pins

Channel	Pin Name	Abbreviation	Input/Output	Function
0	Serial clock pin	SCK0	Input/output	SCI0 clock input/output
	Receive data pin	RxD0	Input	SCI0 receive data input
	Transmit data pin	TxD0	Output	SCI0 transmit data output
1	Serial clock pin	SCK1	Input/output	SCI1 clock input/output
	Receive data pin	RxD1	Input	SCI1 receive data input
	Transmit data pin	TxD1	Output	SCI1 transmit data output
2	Serial clock pin	SCK2	Input/output	SCI2 clock input/output
	Receive data pin	RxD2	Input	SCI2 receive data input
	Transmit data pin	TxD2	Output	SCI2 transmit data output
3	Serial clock pin	SCK3	Input/output	SCI3 clock input/output
	Receive data pin	RxD3	Input	SCI3 receive data input
	Transmit data pin	TxD3	Output	SCI3 transmit data output
4	Serial clock pin	SCK4	Input/output	SCI4 clock input/output
	Receive data pin	RxD4	Input	SCI4 receive data input
	Transmit data pin	TxD4	Output	SCI4 transmit data output

Note: In the text the pins are referred to as SCK, RxD, and TxD, omitting the channel number.

Table 15.2 summarizes the SCI internal registers. These registers select the communication mode (asynchronous or synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

Table 15.2 Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*[2]	Access Size
0	Serial mode register 0	SMR0	R/W	H'00	H'FFFFF000	8, 16
	Bit rate register 0	BRR0	R/W	H'FF	H'FFFFF001	
	Serial control register 0	SCR0	R/W	H'00	H'FFFFF002	
	Transmit data register 0	TDR0	R/W	H'FF	H'FFFFF003	
	Serial status register 0	SSR0	R/(W) *[1]	H'84	H'FFFFF004	
	Receive data register 0	RDR0	R	H'00	H'FFFFF005	
	Serial direction control register 0	SDCR0	R/W	H'F2	H'FFFFF006	8
1	Serial mode register 1	SMR1	R/W	H'00	H'FFFFF008	8, 16
	Bit rate register 1	BRR1	R/W	H'FF	H'FFFFF009	
	Serial control register 1	SCR1	R/W	H'00	H'FFFFF00A	
	Transmit data register 1	TDR1	R/W	H'FF	H'FFFFF00B	
	Serial status register 1	SSR1	R/(W) *[1]	H'84	H'FFFFF00C	
	Receive data register 1	RDR1	R	H'00	H'FFFFF00D	
	Serial direction control register 1	SDCR1	R/W	H'F2	H'FFFFF00E	8
2	Serial mode register 2	SMR2	R/W	H'00	H'FFFFF010	8, 16
	Bit rate register 2	BRR2	R/W	H'FF	H'FFFFF011	
	Serial control register 2	SCR2	R/W	H'00	H'FFFFF012	
	Transmit data register 2	TDR2	R/W	H'FF	H'FFFFF013	
	Serial status register 2	SSR2	R/(W) *[1]	H'84	H'FFFFF014	
	Receive data register 2	RDR2	R	H'00	H'FFFFF015	
	Serial direction control register 2	SDCR2	R/W	H'F2	H'FFFFF016	8

Channel	Name	Abbreviation	R/W	Value	Address*[2]	Size
3	Serial mode register 3	SMR3	R/W	H'00	H'FFFFF018	8, 16
	Bit rate register 3	BRR3	R/W	H'FF	H'FFFFF019	
	Serial control register 3	SCR3	R/W	H'00	H'FFFFF01A	
	Transmit data register 3	TDR3	R/W	H'FF	H'FFFFF01B	
	Serial status register 3	SSR3	R/(W) *[1]	H'84	H'FFFFF01C	
	Receive data register 3	RDR3	R	H'00	H'FFFFF01D	
	Serial direction control register 3	SDCR3	R/W	H'F2	H'FFFFF01E	8
4	Serial mode register 4	SMR4	R/W	H'00	H'FFFFF020	8, 16
	Bit rate register 4	BRR4	R/W	H'FF	H'FFFFF021	
	Serial control register 4	SCR4	R/W	H'00	H'FFFFF022	
	Transmit data register 4	TDR4	R/W	H'FF	H'FFFFF023	
	Serial status register 4	SSR4	R/(W) *[1]	H'84	H'FFFFF024	
	Receive data register 4	RDR4	R	H'00	H'FFFFF025	
	Serial direction control register 4	SDCR4	R/W	H'F2	H'FFFFF026	8

Notes: In register access, four or five cycles are required for byte access, and eight or nine cycles for word access.

*1 Only 0 can be written to clear the flags.

*2 Do not access empty addresses.

15.2 Register Descriptions

15.2.1 Receive Shift Register (RSR)

The receive shift register (RSR) receives serial data. Data input at the RxD pin is loaded into RSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to RDR.

The CPU cannot read or write to RSR directly.

Bit:	7	6	5	4	3	2	1	0
R/W:	—	—	—	—	—	—	—	—

byte of serial data by moving the received data from the receive shift register (RSR) into RDR for storage. RSR is then ready to receive the next data. This double buffering allows the SCI to receive data continuously.

The CPU can read but not write to RDR. RDR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode. It is not initialized by a manual reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

15.2.3 Transmit Shift Register (TSR)

The transmit shift register (TSR) transmits serial data. The SCI loads transmit data from the transmit data register (TDR) into TSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from TDR into TSR and starts transmitting again. If the TDRE bit of SSR is 1, however, the SCI does not load the TDR contents into TSR.

The CPU cannot read or write to TSR directly.

Bit:	7	6	5	4	3	2	1	0
R/W:	—	—	—	—	—	—	—	—

The transmit data register (TDR) is an 8-bit register that stores data for serial transmission. When the SCI detects that the transmit shift register (TSR) is empty, it moves transmit data written in TDR into TSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in TDR during serial transmission from TSR.

The CPU can always read and write to TDR. TDR is initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode. It is not initialized by a manual reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.2.5 Serial Mode Register (SMR)

The serial mode register (SMR) is an 8-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SMR. SMR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode. It is not initialized by a manual reset.

Bit:	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 7—Communication Mode (C/$\overline{A}$): Selects whether the SCI operates in asynchronous or synchronous mode.

Bit 7: C/$\overline{A}$	Description	
0	Asynchronous mode	(Initial value)
1	Synchronous mode	

Bit 6: CHR	Description	
0	Eight-bit data	(Initial value)
1	Seven-bit data When 7-bit data is selected, the MSB (bit 7) of the transmit data register is not transmitted. LSB-first/MSB-first selection is not available.	

- Bit 5—Parity Enable (PE): Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In synchronous mode and when using a multiprocessor format, a parity bit is neither added nor checked, regardless of the PE bit setting.

Bit 5: PE	Description	
0	Parity bit not added or checked	(Initial value)
1	Parity bit added and checked When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$ bit) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$ bit) setting.	

- Bit 4—Parity Mode (O/$\overline{E}$): Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{E}$ setting is invalid in synchronous mode, in asynchronous mode when parity bit addition and checking is disabled, and when using a multiprocessor format.

Bit 4: O/$\overline{E}$	Description	
0	Even parity If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined.	(Initial value)
1	Odd parity If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.	

because no stop bits are added.

In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character.

Bit 3: STOP	Description	
0	One stop bit	(Initial value)
	In transmitting, a single bit of 1 is added at the end of each transmitted character.	
1	Two stop bits	
	In transmitting, two 1-bits are added at the end of each transmitted character.	

- Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, settings of the parity enable (PE) and parity mode (O/$\overline{E}$) bits are ignored. The MP bit setting is used only in asynchronous mode; it is ignored in synchronous mode. For the multiprocessor communication function, see section 15.3.3, Multiprocessor Communication.

Bit 2: MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

- Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available: Pϕ, Pϕ/4, Pϕ/16, or Pϕ/64 (Pϕ is the peripheral clock). For further information on the clock source, bit rate register settings, and baud rate, see section 15.2.8, Bit Rate Register (BRR).

Bit 1: CKS1	Bit 0: CKS0	Description	
0	0	Pϕ	(Initial value)
	1	Pϕ/4	
1	0	Pϕ/16	
	1	Pϕ/64	

The serial control register (SCR) operates the SCI transmitter/receiver, selects the serial clock output in asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCR. SCR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode. It is not initialized by a manual reset.

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 7—Transmit Interrupt Enable (TIE): Enables or disables the transmit-data-empty interrupt (TXI) requested when the transmit data register empty bit (TDRE) in the serial status register (SSR) is set to 1 by transfer of serial transmit data from TDR to TSR.

Bit 7: TIE	Description
0	Transmit-data-empty interrupt request (TXI) is disabled (Initial value) The TXI interrupt request can be cleared by reading TDRE after it has been set to 1, then clearing TDRE to 0, or by clearing TIE to 0.
1	Transmit-data-empty interrupt request (TXI) is enabled

- Bit 6—Receive Interrupt Enable (RIE): Enables or disables the receive-data-full interrupt (RXI) requested when the receive data register full bit (RDRF) in the serial status register (SSR) is set to 1 by transfer of serial receive data from RSR to RDR. It also enables or disables receive-error interrupt (ERI) requests.

Bit 6: RIE	Description
0	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled (Initial value) RXI and ERI interrupt requests can be cleared by reading the RDRF flag or error flag (FER, PER, or ORER) after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0.
1	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled

0	Transmitter disabled (Initial value) The transmit data register empty bit (TDRE) in the serial status register (SSR) is locked at 1.
1	Transmitter enabled Serial transmission starts when the transmit data register empty (TDRE) bit in the serial status register (SSR) is cleared to 0 after writing of transmit data into TDR. Select the transmit format in SMR before setting TE to 1.

- Bit 4—Receive Enable (RE): Enables or disables the SCI serial receiver.

Bit 4: RE	Description
0	Receiver disabled (Initial value) Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, ORER). These flags retain their previous values.
1	Receiver enabled Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in synchronous mode. Select the receive format in SMR before setting RE to 1.

(MP) in the serial mode register (SMR) is set to 1 during reception. The MPIE setting is ignored in synchronous mode or when the MP bit is cleared to 0.

Bit 3: MPIE	Description
0	Multiprocessor interrupts are disabled (normal receive operation) (Initial value) [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts are enabled. Receive-data-full interrupt requests (RXI), receive-error interrupt requests (ERI), and setting of the RDRF, FER, and ORER status flags in the serial status register (SSR) are disabled until data with the multiprocessor bit set to 1 is received. The SCI does not transfer receive data from RSR to RDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in the serial status register (SSR). When it receives data that includes MPB = 1, MPB is set to 1, and the SCI automatically clears MPIE to 0, generates RXI and ERI interrupts (if the TIE and RIE bits in SCR are set to 1), and allows the FER and ORER bits to be set.

- Bit 2—Transmit-End Interrupt Enable (TEIE): Enables or disables the transmit-end interrupt (TEI) requested if TDR does not contain valid transmit data when the MSB is transmitted.

Bit 2: TEIE	Description
0	Transmit-end interrupt (TEI) requests are disabled* (Initial value)
1	Transmit-end interrupt (TEI) requests are enabled*

Note: * The TEI request can be cleared by reading the TDRE bit in the serial status register (SSR) after it has been set to 1, then clearing TDRE to 0 and clearing the transmit end (TEND) bit to 0; or by clearing the TEIE bit to 0.

and CKE0, the SCK pin can be used for serial clock output, or serial clock input. Select the SCK pin function by using the pin function controller (PFC).

The CKE0 setting is valid only in asynchronous mode, and only when the SCI is internally clocked (CKE1 = 0). The CKE0 setting is ignored in synchronous mode, or when an external clock source is selected (CKE1 = 1). For further details on selection of the SCI clock source, see table 15.9 in section 15.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description*[1]	
0	0	Asynchronous mode	Internal clock, SCK pin used for input pin (input signal is ignored) or output pin (output level is undefined)*[2]
		Synchronous mode	Internal clock, SCK pin used for synchronous clock output*[2]
0	1	Asynchronous mode	Internal clock, SCK pin used for clock output*[3]
		Synchronous mode	Internal clock, SCK pin used for synchronous clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input*[4]
		Synchronous mode	External clock, SCK pin used for synchronous clock input
1	1	Asynchronous mode	External clock, SCK pin used for clock input*[4]
		Synchronous mode	External clock, SCK pin used for synchronous clock input

Notes: *1 The SCK pin is multiplexed with other functions. Use the pin function controller (PFC) to select the SCK function for this pin, as well as the I/O direction.

*2 Initial value.

*3 The output clock frequency is the same as the bit rate.

*4 The input clock frequency is 16 times the bit rate.

The serial status register (SSR) is an 8-bit register containing multiprocessor bit values, and status flags that indicate the SCI operating status.

The CPU can always read and write to SSR, but cannot write 1 in the status flags (TDRE, RDRF, ORER, PER, and FER). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 2 (TEND) and 1 (MPB) are read-only bits that cannot be written. SSR is initialized to H'84 by a power-on reset, and in hardware standby mode and software standby mode. It is not initialized by a manual reset.

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * The only value that can be written is a 0 to clear the flag.

- Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from TDR into TSR and new serial transmit data can be written in TDR.

Bit 7: TDRE	Description
0	TDR contains valid transmit data [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC writes data in TDR
1	TDR does not contain valid transmit data (Initial value) [Setting conditions] • Power-on reset, hardware standby mode, or software standby mode • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR, enabling new data to be written in TDR

0	RDR does not contain valid receive data (Initial value) [Clearing conditions] • Power-on reset, hardware standby mode, or software standby mode • When 0 is written to RDRF after reading RDRF = 1 • When the DMAC reads data from RDR
1	RDR contains valid received data [Setting condition] RDRF is set to 1 when serial data is received normally and transferred from RSR to RDR

Note: RDR and RDRF are not affected by detection of receive errors or by clearing of the RE bit to 0 in the serial control register. They retain their previous contents. If RDRF is still set to 1 when reception of the next data ends, an overrun error (ORER) occurs and the receive data is lost.

- Bit 5—Overrun Error (ORER): Indicates that data reception ended abnormally due to an overrun error.

Bit 5: ORER	Description
0	Receiving is in progress or has ended normally (Initial value) Clearing the RE bit to 0 in the serial control register does not affect the ORER bit, which retains its previous value. [Clearing conditions] • Power-on reset, hardware standby mode, or software standby mode • When 0 is written to ORER after reading ORER = 1
1	A receive overrun error occurred RDR continues to hold the data received before the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while ORER is set to 1. In synchronous mode, serial transmitting is disabled. [Setting condition] ORER is set to 1 if reception of the next serial data ends when RDRF is set to 1

Bit 4: FER	Description
0	Receiving is in progress or has ended normally (Initial value) Clearing the RE bit to 0 in the serial control register does not affect the FER bit, which retains its previous value. [Clearing conditions] • Power-on reset, hardware standby mode, or software standby mode • When 0 is written to FER after reading FER = 1
1	A receive framing error occurred When the stop bit length is two bits, only the first bit is checked to see if it is a 1. The second stop bit is not checked. When a framing error occurs, the SCI transfers the receive data into RDR but does not set RDRF. Serial receiving cannot continue while FER is set to 1. In synchronous mode, serial transmitting is also disabled. [Setting condition] FER is set to 1 if the stop bit at the end of receive data is checked and found to be 0

- Bit 3—Parity Error (PER): Indicates that data reception (with parity) ended abnormally due to a parity error in asynchronous mode.

Bit 3: PER	Description
0	Receiving is in progress or has ended normally (Initial value) Clearing the RE bit to 0 in the serial control register does not affect the PER bit, which retains its previous value. [Clearing conditions] • Power-on reset, hardware standby mode, or software standby mode • When 0 is written to PER after reading PER = 1
1	A receive parity error occurred When a parity error occurs, the SCI transfers the receive data into RDR but does not set RDRF. Serial receiving cannot continue while PER is set to 1. [Setting condition] PER is set to 1 if the number of 1s in receive data, including the parity bit, does not match the even or odd parity setting of the parity mode bit (O/$\overline{E}$) in the serial mode register (SMR)

bit and cannot be written.

Bit 2: TEND	Description
0	Transmission is in progress [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC writes data in TDR
1	End of transmission (Initial value) [Setting conditions] • Power-on reset, hardware standby mode, or software standby mode • When the TE bit in SCR is 0 • If TDRE = 1 when the last bit of a one-byte serial transmit character is transmitted

- Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in receive data when a multiprocessor format is selected for receiving in asynchronous mode. MPB is a read-only bit and cannot be written.

Bit 1: MPB	Description
0	Multiprocessor bit value in receive data is 0 (Initial value) If RE is cleared to 0 when a multiprocessor format is selected, the MPB retains its previous†value.
1	Multiprocessor bit value in receive data is 1

- Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in asynchronous mode. The MPBT setting is ignored in synchronous mode, when a multiprocessor format is not selected, or when the SCI is not transmitting.

Bit 0: MPBT	Description
0	Multiprocessor bit value in transmit data is 0 (Initial value)
1	Multiprocessor bit value in transmit data is 1

The bit rate register (BRR) is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to BRR. BRR is initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode. It is not initialized by a manual reset. Each channel has independent baud rate generator control, so different values can be set for each channel.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.3 lists examples of BRR settings in the asynchronous mode; table 15.4 lists examples of BBR settings in the clock synchronous mode.

The BRR setting is calculated as follows:

Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Synchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: Baud rate generator BRR setting ($0 \leq N \leq 255$)
Pϕ: Peripheral module operating frequency (MHz) (1/2 of system clock)
n: Baud rate generator input clock (n = 0 to 3)
(See the following table for the clock sources and value of n.)

		SMR Settings	
n	**Clock Source**	**CKS1**	**CKS2**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 15.3 Bit Rates and BRR Settings in Asynchronous Mode

Bit Rate (Bits/s)	Pϕ (MHz)								
	10			11.0592			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	–0.25	2	195	0.19	2	212	0.03
150	2	129	0.16	2	143	0.00	2	155	0.16
300	2	64	0.16	2	71	0.00	2	77	0.16
600	1	129	0.16	1	143	0.00	1	155	0.16
1200	1	64	0.16	1	71	0.00	1	77	0.16
2400	0	129	0.16	0	143	0.00	0	155	0.16
4800	0	64	0.16	0	71	0.00	0	77	0.16
9600	0	32	–1.36	0	35	0.00	0	28	0.16
14400	0	21	–1.36	0	23	0.00	0	25	0.16
19200	0	15	1.73	0	19	0.00	0	19	–2.34
28800	0	10	–1.36	0	11	0.00	0	12	0.16
31250	0	9	0.00	0	10	0.54	0	11	0.00
38400	0	7	1.73	0	8	0.00	0	9	–2.34

Bit Rate (Bits/s)	12.288			14			14.7456		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	–0.17	3	64	0.70
150	2	159	0.00	2	181	0.16	2	191	0.00
300	2	79	0.00	2	90	0.16	2	95	0.00
600	1	159	0.00	1	181	0.16	1	191	0.00
1200	1	79	0.00	1	90	0.16	1	95	0.00
2400	0	159	0.00	0	181	0.16	0	191	0.00
4800	0	79	0.00	0	90	0.16	0	95	0.00
9600	0	39	0.00	0	45	–0.93	0	47	0.00
14400	0	26	–1.23	0	29	1.27	0	31	0.00
19200	0	19	0.00	0	22	–0.93	0	23	0.00
28800	0	12	2.56	0	14	1.27	0	15	0.00
31250	0	11	2.40	0	13	0.00	0	14	–1.70
38400	0	9	0.00	0	10	3.57	0	11	0.00

Bit Rate (Bits/s)	16			17.2032			18		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	70	0.03	3	75	0.48	3	79	–0.12
150	2	207	0.16	2	223	0.00	2	233	0.16
300	2	103	0.16	2	111	0.00	2	116	0.16
600	1	207	0.16	1	223	0.00	1	233	0.16
1200	1	103	0.16	1	111	0.00	1	116	0.16
2400	0	207	0.16	0	223	0.00	0	233	0.16
4800	0	103	0.16	0	111	0.00	0	116	0.16
9600	0	51	0.16	0	55	0.00	0	58	–0.69
14400	0	34	–0.79	0	36	0.90	0	38	0.16
19200	0	25	0.16	0	27	0.00	0	28	1.02
28800	0	16	2.12	0	18	–1.75	0	19	–2.34
31250	0	15	0.00	0	16	1.20	0	17	0.00
38400	0	12	0.16	0	13	0.00	0	14	–2.34

Bit Rate (Bits/s)	18.432			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	81	–0.22	3	86	0.31	3	88	–0.25
150	2	239	0.00	2	255	0.00	3	64	0.16
300	2	119	0.00	2	127	0.00	2	129	0.16
600	1	239	0.00	1	255	0.00	2	64	0.16
1200	1	119	0.00	1	127	0.00	1	129	0.16
2400	0	239	0.00	0	255	0.00	1	64	0.16
4800	0	119	0.00	0	127	0.00	0	129	0.16
9600	0	59	0.00	0	63	0.00	0	64	0.16
14400	0	39	0.00	0	42	–0.78	0	42	0.94
19200	0	29	0.00	0	31	0.00	0	32	–1.36
28800	0	19	0.00	0	20	1.59	0	21	–1.36
31250	0	17	2.40	0	19	–1.70	0	19	0.00
38400	0	14	0.00	0	15	0.00	0	15	1.73

Bit Rate (Bits/s)	10		12		16		20	
	n	N	n	N	n	N	n	N
250	—	—	3	187	3	249		
500	—	—	3	93	3	124	—	—
1 k	—	—	2	187	2	249	—	—
2.5 k	1	249	2	74	2	99	2	124
5 k	1	124	1	149	1	199	2	249
10 k	0	249	1	74	1	99	1	124
25 k	0	99	0	119	0	159	1	199
50 k	0	49	0	59	0	79	0	99
100 k	0	24	0	29	0	39	0	49
250 k	0	9	0	11	0	15	0	19
500 k	0	4	0	5	0	7	0	9
1 M			0	2	0	3	0	4
2.5 M	0	0*	0	0*	—	—	0	1
5 M							0	0*

Note: Settings with an error of 1% or less are recommended.

Legend

Blank: No setting available

—: Setting possible, but error occurs

*: Continuous transmission/reception not possible

Table 15.5 indicates the maximum bit rates in asynchronous mode when the baud rate generator is being used for various frequencies. Tables 15.6 and 15.7 show the maximum rates for external clock input.

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (Bits/s)**	**n**	**N**
10	312500	0	0
11.0592	345600	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
18.432	576000	0	0
19.6608	614400	0	0
20	625000	0	0

Table 15.6 Maximum Bit Rates during External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bits/s)
10	2.5000	156250
11.0592	2.7648	172800
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
18.432	4.6080	288000
19.6608	4.9152	307200
20	5.0000	312500

10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3

15.2.9 Serial Direction Control Register (SDCR)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	DIR	—	—	—
Initial value:	1	1	1	1	0	0	1	0
R/W:	R	R	R	R	R/W	R	R	R

The DIR bit in the serial direction control register (SDCR) selects LSB-first or MSB-first transfer. With an 8-bit data length, LSB-first/MSB-first selection is available regardless of the communication mode. With a 7-bit data length, LSB-first transfer must be selected. The description in this section assumes LSB-first transfer.

SDCR is initialized to H'F2 by a power-on reset, and in the hardware standby mode and software standby mode. It is not initialized by a manual reset.

- Bits 7–4—Reserved: The write value should always be 1. If 0 is written to these bits, correct operation cannot be guaranteed.
- Bit 3—Data Transfer Direction (DIR): Selects the serial/parallel conversion format. Valid for an 8-bit transmit/receive format.

Bit 3: DIR	Description	
0	TDR contents are transmitted in LSB-first order	(Initial value)
	Receive data is stored in RDR in LSB-first order	
1	TDR contents are transmitted in MSB-first order	
	Receive data is stored in RDR in MSB-first order	

- Bit 2—Reserved: The write value should always be 0. If 1 is written to this bit, correct operation cannot be guaranteed.
- Bit 1—Reserved: This bit is always read as 1, and cannot be modified.

15.2.10 Inversion of SCK Pin Signal

The signal input from the SCK pin and the signal output from the SCK pin can be inverted by means of a port control register setting. See section 20, Pin function Controller (PFC), for details.

15.3 Operation

15.3.1 Overview

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a synchronous mode in which communication is synchronized with clock pulses. Asynchronous synchronous mode and the transmission format are selected in the serial mode register (SMR), as shown in table 15.8. The SCI clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SMR) and the CKE1 and CKE0 bits in the serial control register (SCR), as shown in table 15.9.

Asynchronous Mode:

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable, as well as the stop bit length (one or two bits). These selections determine the transmit/receive format and character length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), overrun errors (ORER), and the break state.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator clock, and can output a clock with a frequency matching the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

Synchronous Mode:

- The communication format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator clock, and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

Mode	Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 2 MP	Bit 3 STOP	Data Length	Parity Bit	Multipro-cessor Bit	Stop Bit Length
Asynchronous	0	0	0	0	0	8-bit	Absent	Absent	1 bit
					1				2 bits
			1		0		Present		1 bit
					1				2 bits
		1	0		0	7-bit	Absent		1 bit
					1				2 bits
			1		0		Present		1 bit
					1				2 bits
Asynchronous (multiprocessor format)		0	*	1	0	8-bit	Absent	Present	1 bit
			*		1				2 bits
		1	*		0	7-bit			1 bit
			*		1				2 bits
Synchronous	1	*	*	*	*	8-bit		Absent	None

Note: Asterisks (*) in the table indicate donít-care bits.

Table 15.9 SMR and SCR Settings and SCI Clock Source Selection

	SMR	SCR Settings		SCI Transmit/Receive Clock	
Mode	Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Clock Source	SCK Pin Function*
Asynchronous	0	0	0	Internal	SCI does not use the SCK pin
			1		Outputs a clock with frequency matching the bit rate
		1	0	External	Inputs a clock with frequency 16 times the bit rate
			1		
Synchronous	1	0	0	Internal	Outputs the serial clock or the inverted serial clock
			1		
		1	0	External	Inputs the serial clock or the inverted serial clock
			1		

Note: * Select the function in combination with the pin function controller (PFC).

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 15.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the marking (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes on the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

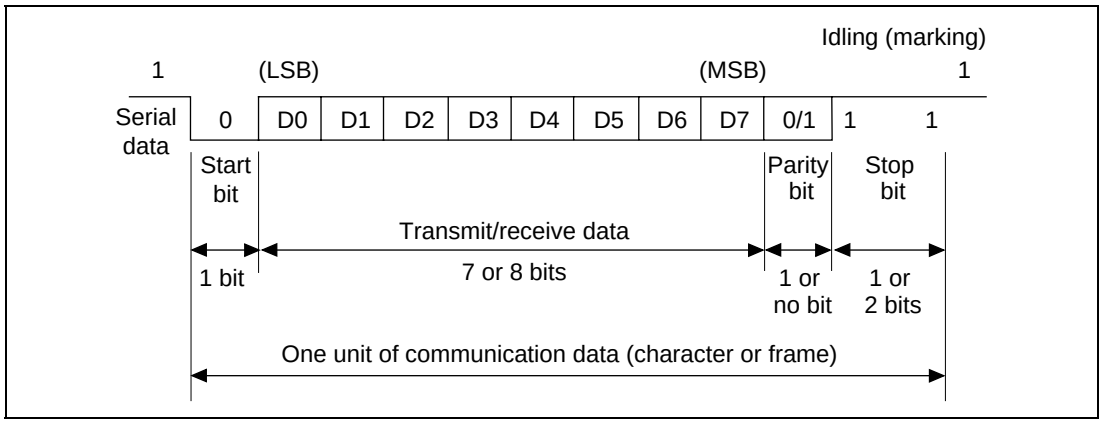


Figure 15.2 Data Format in Asynchronous Communication (Example: 8-bit Data with Parity and Two Stop Bits)

(SMR).

Table 15.10 Serial Communication Formats (Asynchronous Mode)

SMR Bits				Serial Transmit/Receive Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	START	8-bit data								STOP		
0	0	0	1	START	8-bit data								STOP	STOP	
0	1	0	0	START	8-bit data								P	STOP	
0	1	0	1	START	8-bit data								P	STOP	STOP
1	0	0	0	START	7-bit data							STOP			
1	0	0	1	START	7-bit data							STOP	STOP		
1	1	0	0	START	7-bit data							P	STOP		
1	1	0	1	START	7-bit data							P	STOP	STOP	
0	—	1	0	START	8-bit data								MPB	STOP	
0	—	1	1	START	8-bit data								MPB	STOP	STOP
1	—	1	0	START	7-bit data							MPB	STOP		
1	—	1	1	START	7-bit data							MPB	STOP	STOP	

—: Don't care bits.

Notes: START: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SMR) and bits CKE1 and CKE0 in the serial control register (SCR) (table 15.9).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

the rising edge of the clock occurs at the center of each transmit data bit.

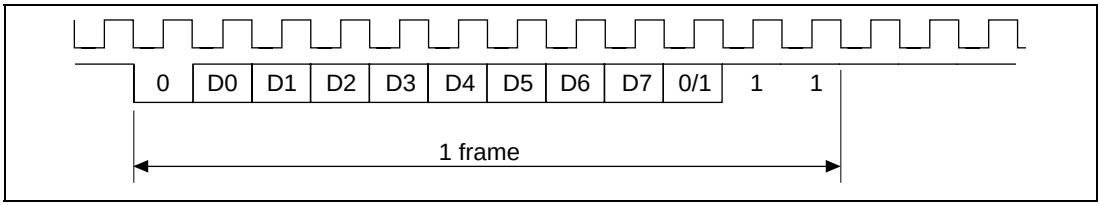


Figure 15.3 Output Clock and Communication Data Phase Relationship (Asynchronous Mode)

Data Transmit/Receive Operation

SCI Initialization (Asynchronous Mode): Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as follows.

When changing the operation mode or communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

Figure 15.4 is a sample flowchart for initializing the SCI. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. Select the clock source in the serial control register (SCR). Leave RIE, TIE, TEIE, MPIE, TE, and RE cleared to 0. If clock output is selected in asynchronous mode, clock output starts immediately after the setting is made in SCR.
2. Select the communication format in the serial mode register (SMR) and serial direction control register (SDCR).
3. Write the value corresponding to the bit rate in the bit rate register (BRR) (unless an external clock is used).
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCR) to 1.* Also set RIE, TIE, TEIE and MPIE as necessary. Setting TE or RE enables the SCI to use the TxD or RxD pin.

Note: * In simultaneous transmit/receive operation, the TE bit and RE bit must be cleared to 0 or set to 1 simultaneously.

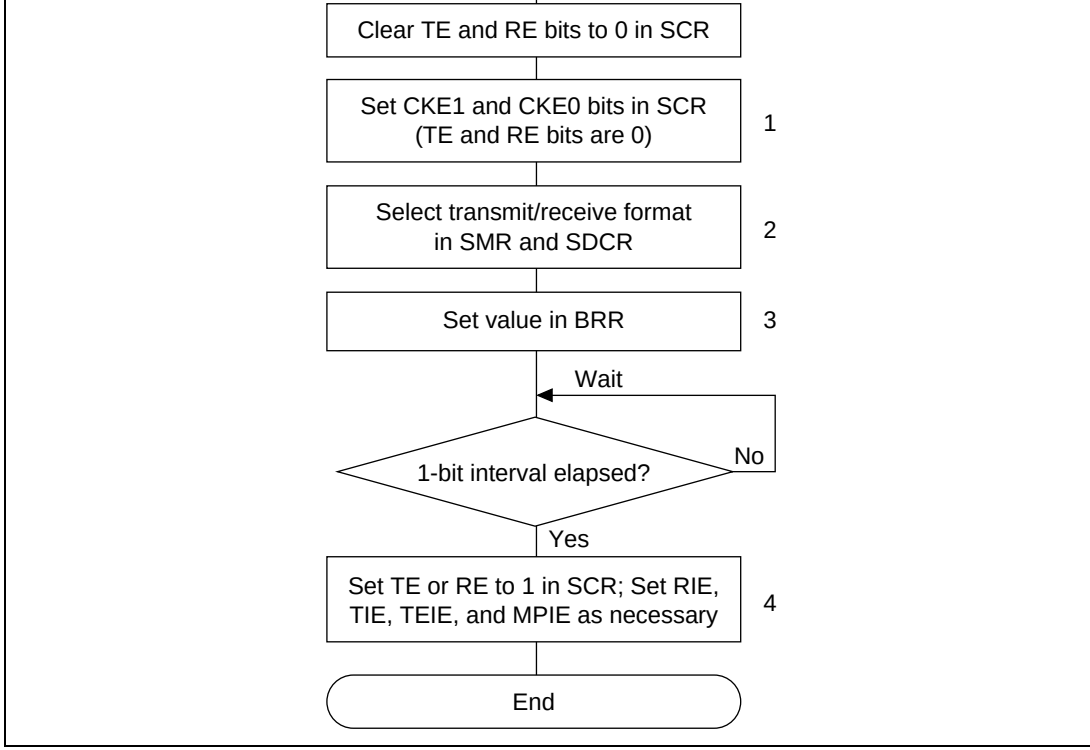


Figure 15.4 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode): Figure 15.5 shows a sample flowchart for transmitting serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the TxD pin using the PFC.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0.
3. Continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in TDR, then clear TDRE to 0. When the DMAC is started by a transmit-data-empty interrupt request (TXI) in order to write data in TDR, the TDRE bit is checked and cleared automatically.
4. To output a break at the end of serial transmission, first clear the port data register (DR) to 0, then clear the TE bit to 0 in SCR and use the PFC to establish the TxD pin as an output port.

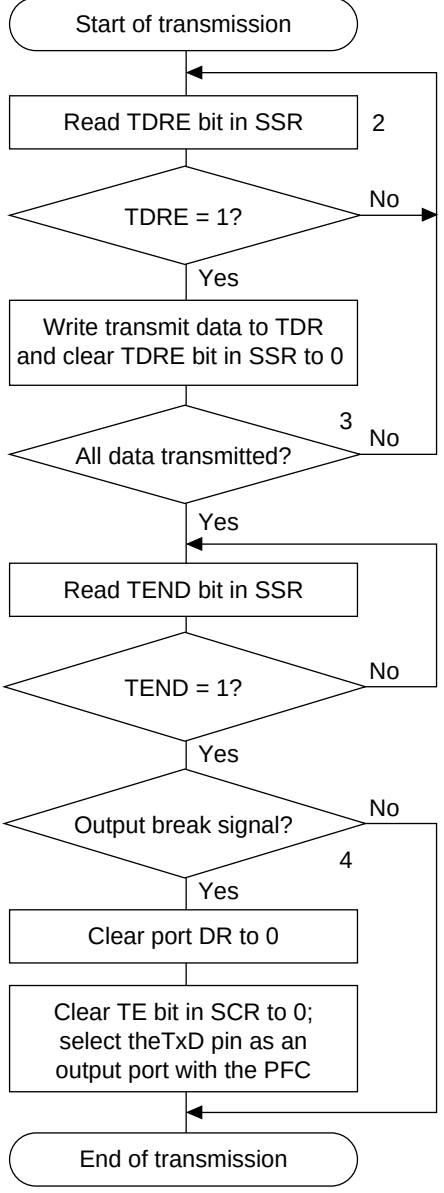


Figure 15.5 Sample Flowchart for Transmitting Serial Data

the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).

2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

   Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: one 0-bit is output.
   b. Transmit data: seven or eight bits of data are output, LSB first.
   c. Parity bit or multiprocessor bit: one parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.
   d. Stop bit: one or two 1-bits (stop bits) are output.
   e. Marking: output of 1-bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads new data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit to 1 in SSR, outputs the stop bit, then continues output of 1-bits (marking). If the transmit-end interrupt enable bit (TEIE) in SCR is set to 1, a transmit-end interrupt (TEI) is requested.

Figure 15.6 shows an example of SCI transmit operation in asynchronous mode.

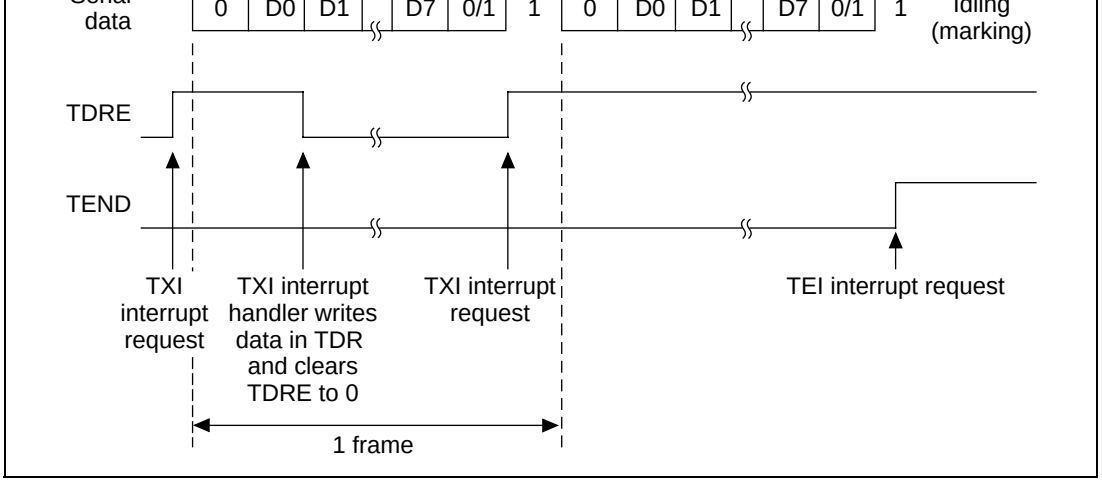


Figure 15.6 SCI Transmit Operation in Asynchronous Mode (Example: 8-Bit Data with Parity and One Stop Bit)

Receiving Serial Data (Asynchronous Mode): Figures 15.7 and 15.8 show a sample flowchart for receiving serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart).

1. SCI initialization: Set the RxD pin using the PFC.
2. Receive error handling and break detection: If a receive error occurs, read the ORER, PER, and FER bits of SSR to identify the error. After executing the necessary error handling, clear ORER, PER, and FER all to 0. Receiving cannot resume if ORER, PER or FER remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.
3. SCI status check and receive-data read: Read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
4. Continue receiving serial data: Read RDR and the RDRF bit and clear RDRF to 0 before the stop bit of the current frame is received. If the DMAC is started by a receive-data-full interrupt (RXI) to read RDR, the RDRF bit is cleared automatically so this step is unnecessary.

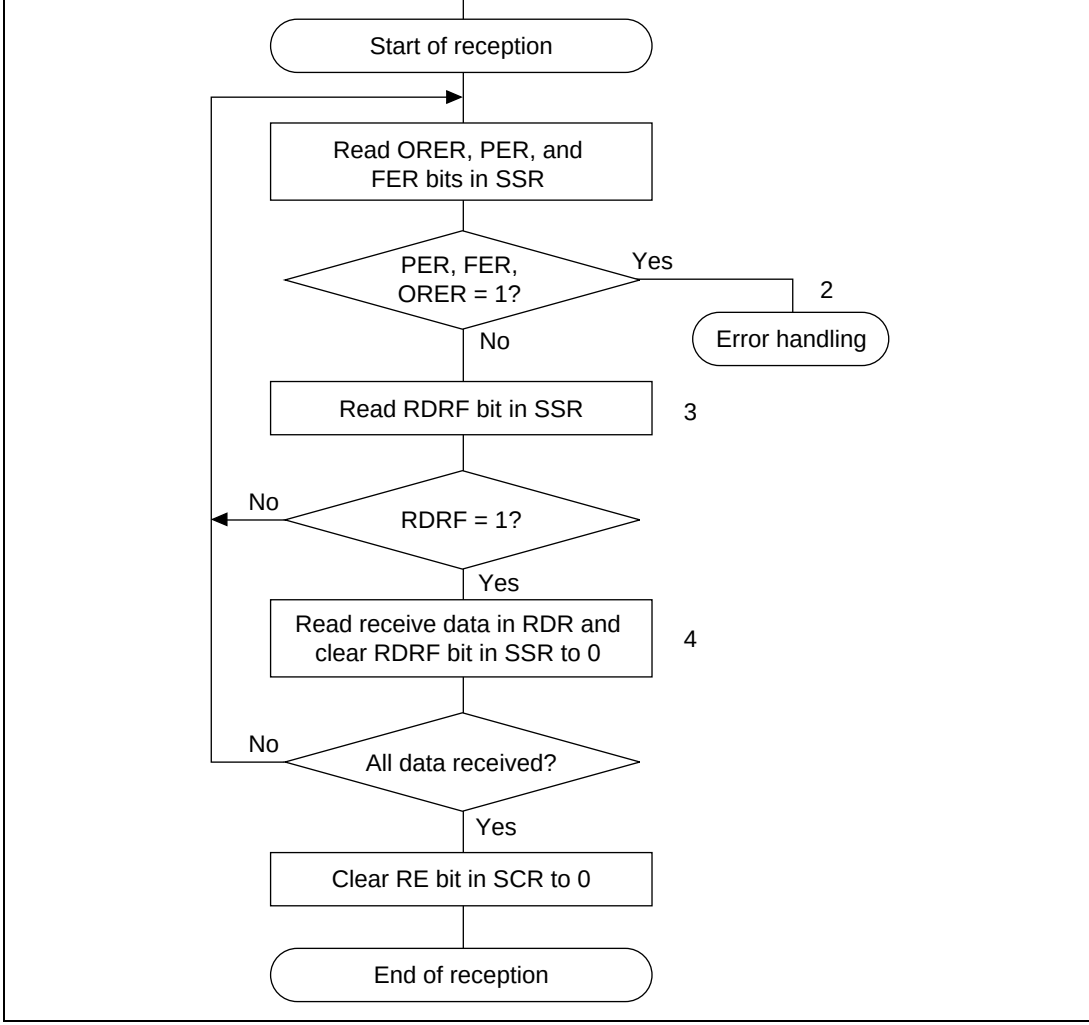


Figure 15.7 Sample Flowchart for Receiving Serial Data (1)

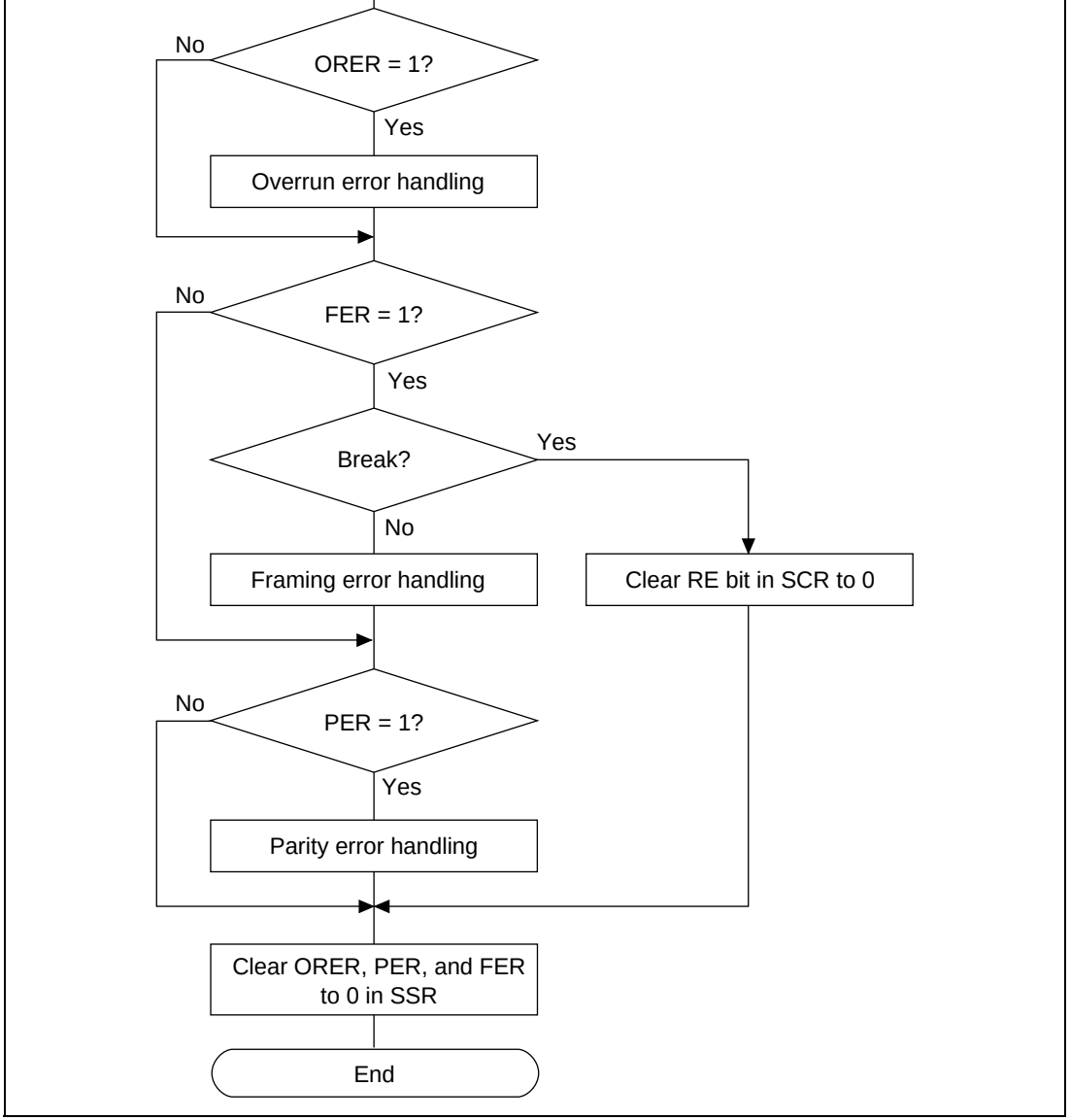


Figure 15.8 Sample Flowchart for Receiving Serial Data (2)

internally and starts receiving.

2. Receive data is shifted into RSR in order from the LSB to the MSB.
3. The parity bit and stop bit are received. After receiving these bits, the SCI makes the following checks:
   a. Parity check. The number of 1s in the receive data must match the even or odd parity setting of the O/$\overline{E}$ bit in SMR.
   b. Stop bit check. The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.
   c. Status check. RDRF must be 0 so that receive data can be loaded from RSR into RDR.

   If the data passes these checks, the SCI sets RDRF to 1 and stores the receive data in RDR. If one of the checks fails (receive error), the SCI operates as indicated in table 15.11.

   Note: When a receive error occurs, further receiving is disabled. While receiving, the RDRF bit is not set to 1, so be sure to clear the error flags.
4. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCR, the SCI requests a receive-data-full interrupt (RXI). If one of the error flags (ORER, PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Table 15.11 Receive Error Conditions and SCI Operation

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SSR	Receive data not loaded from RSR into RDR
Framing error	FER	Stop bit is 0	Receive data loaded from RSR into RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data loaded from RSR into RDR

Figure 15.9 shows an example of SCI receive operation in asynchronous mode.

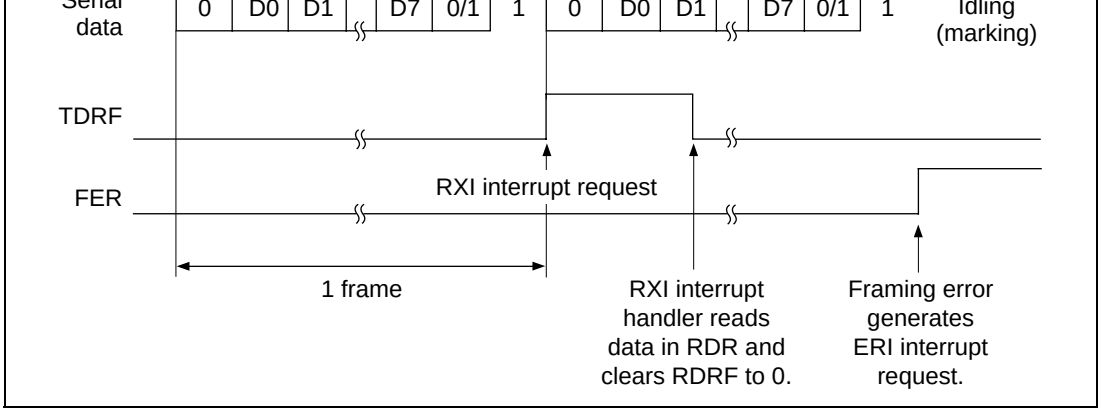


Figure 15.9 SCI Receive Operation
(Example: 8-Bit Data with Parity and One Stop Bit)

15.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line for sending and receiving data. The processors communicate in the asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by a unique ID. A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles. The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1. When they receive data with the multiprocessor bit set to 1, receiving processors compare the data with their IDs. The receiving processor with a matching ID continues to receive further incoming data. Processors with IDs not matching the received data skip further incoming data until they again receive data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 15.10 shows an example of communication among processors using the multiprocessor format.

Clock: See the description in the asynchronous mode section.

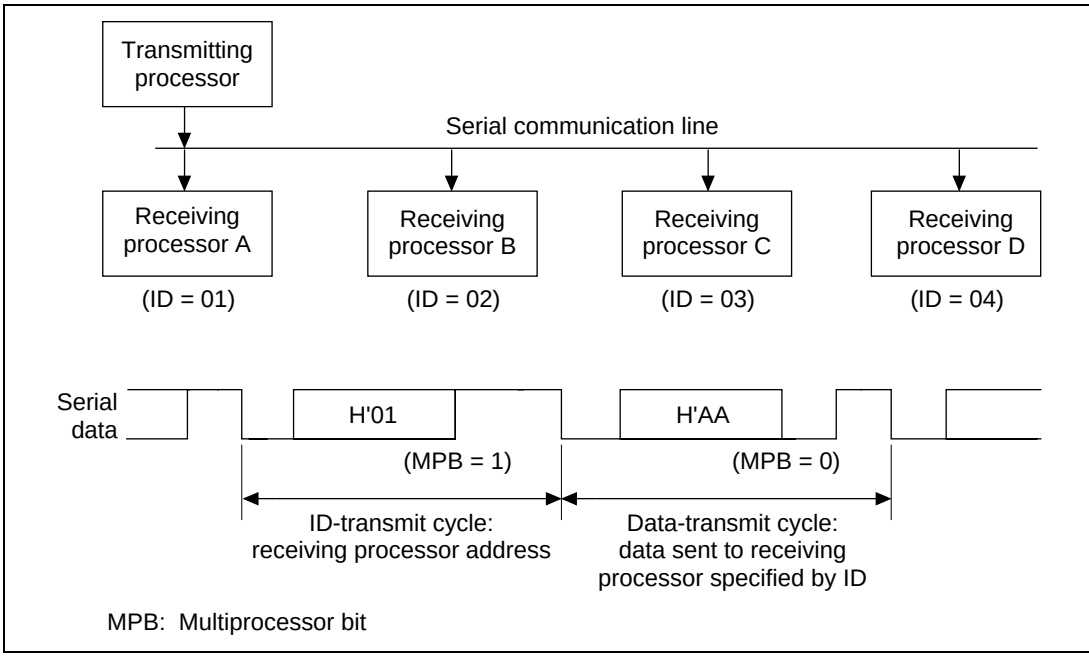


Figure 15.10 Communication among Processors Using Multiprocessor Format (Example: Sending Data H'AA to Receiving Processor A)

Data Transmit/Receive Operation

Transmitting Multiprocessor Serial Data: Figure 15.11 shows a sample flowchart for transmitting multiprocessor serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the TxD pin using the PFC.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR). Also set MPBT (multiprocessor bit transfer) to 0 or 1 in SSR. Finally, clear TDRE to 0.
3. Continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in TDR, then clear TDRE to 0. When the DMAC is started by a transmit-data-empty interrupt request (TXI) to write data in TDR, the TDRE bit is checked and cleared automatically.
4. Output a break at the end of serial transmission: Set the data register (DR) of the port to 0, then clear TE to 0 in SCR and set the TxD pin function as output port with the PFC.

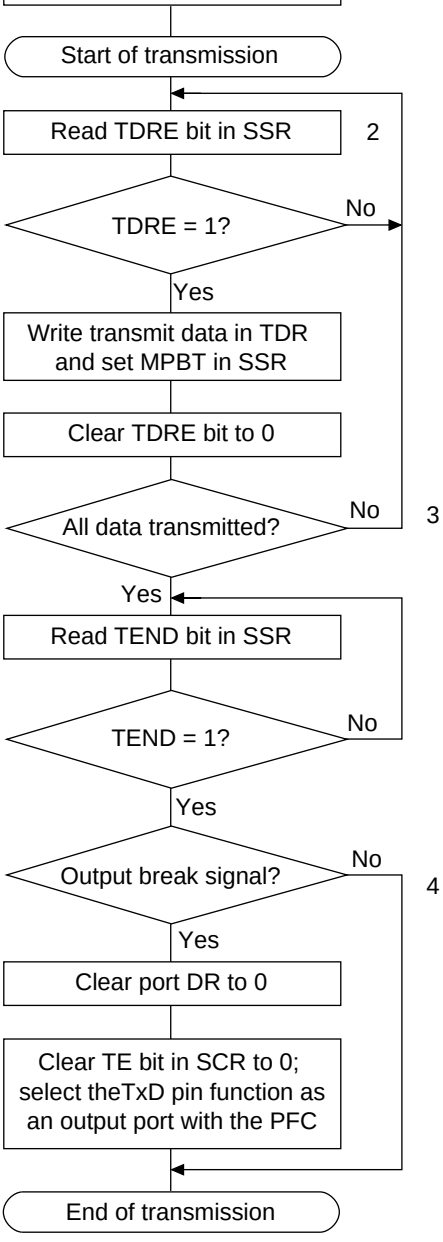


Figure 15.11 Sample Flowchart for Transmitting Multiprocessor Serial Data

the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).

2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

   Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: one 0-bit is output.
   b. Transmit data: seven or eight bits are output, LSB first.
   c. Multiprocessor bit: one multiprocessor bit (MPBT value) is output.
   d. Stop bit: one or two 1-bits (stop bits) are output.
   e. Marking: output of 1-bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in SSR to 1, outputs the stop bit, then continues output of 1-bits in the marking state. If the transmit-end interrupt enable bit (TEIE) in SCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

Figure 15.12 shows an example of SCI receive operation in the multiprocessor format.

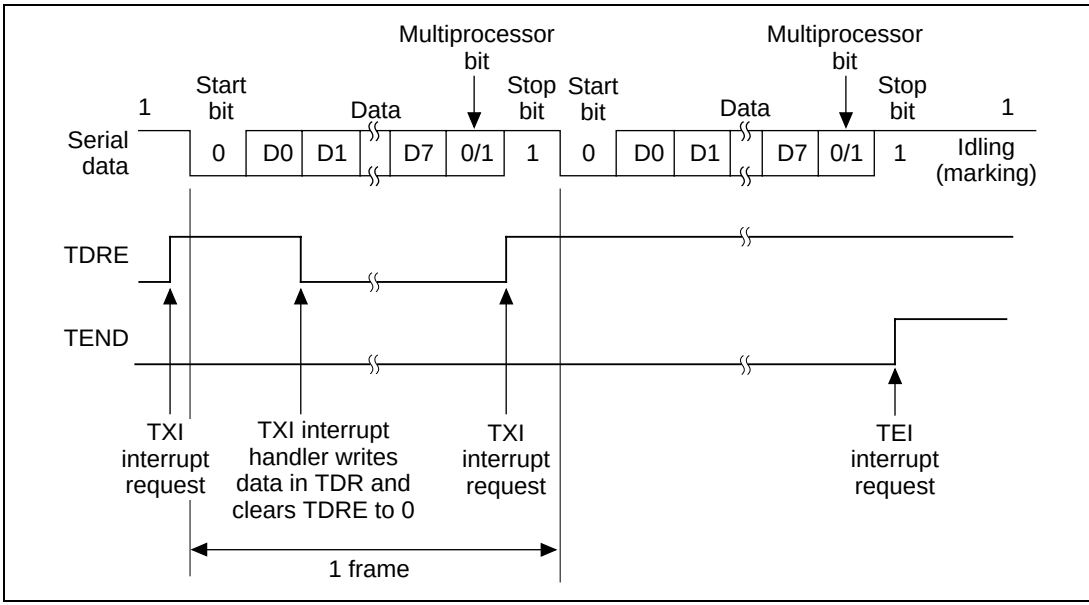


Figure 15.12 SCI Multiprocessor Transmit Operation (Example: 8-Bit Data with Multiprocessor Bit and One Stop Bit)

steps correspond to the numbers in the flowchart).

1. SCI initialization: Set the RxD pin using the PFC.
2. ID receive cycle: Set the MPIE bit in the serial control register (SCR) to 1.
3. SCI status check and compare to ID reception: Read the serial status register (SSR), check that RDRF is set to 1, then read data from the receive data register (RDR) and compare with the processor's own ID. If the ID does not match the receive data, set MPIE to 1 again and clear RDRF to 0. If the ID matches the receive data, clear RDRF to 0.
4. Receive error handling and break detection: If a receive error occurs, read the ORER and FER bits in SSR to identify the error. After executing the necessary error handling, clear both ORER and FER to 0. Receiving cannot resume if ORER or FER remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.
5. SCI status check and data receiving: Read SSR, check that RDRF is set to 1, then read data from the receive data register (RDR).

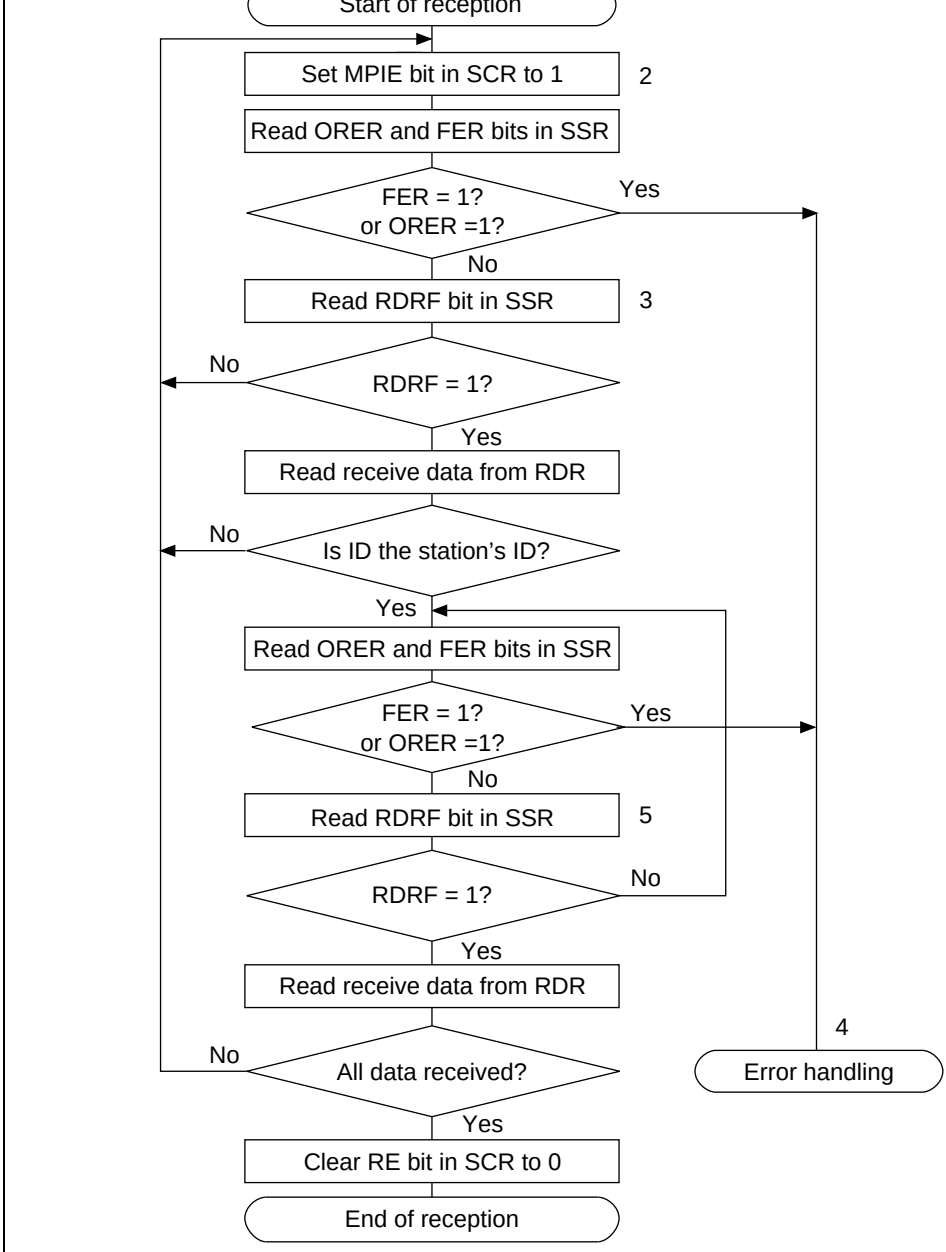


Figure 15.13 Sample Flowchart for Receiving Multiprocessor Serial Data (1)

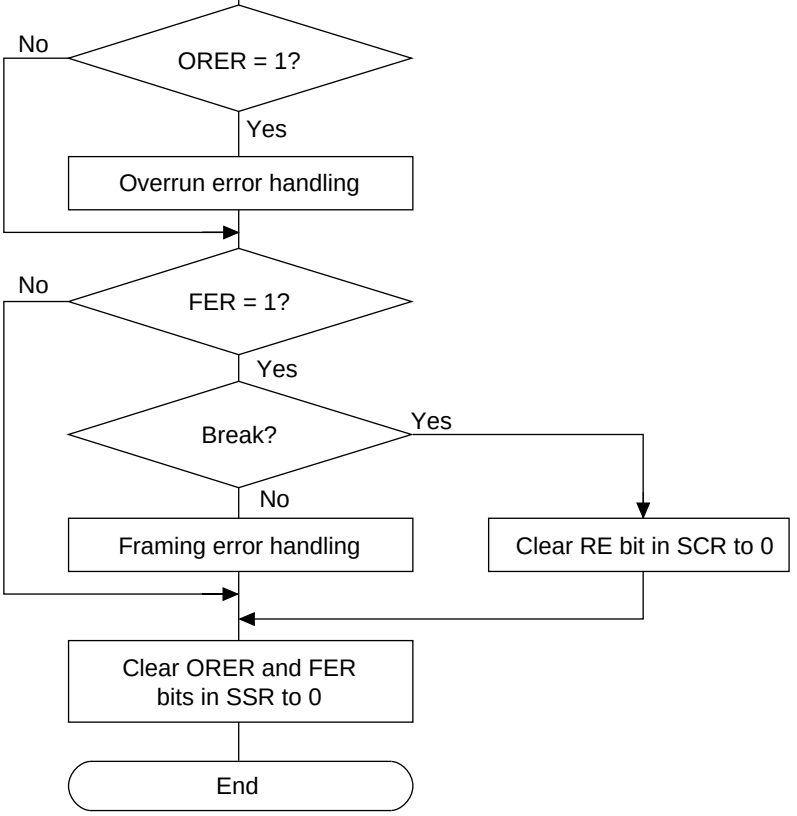


Figure 15.14 Sample Flowchart for Receiving Multiprocessor Serial Data (2)

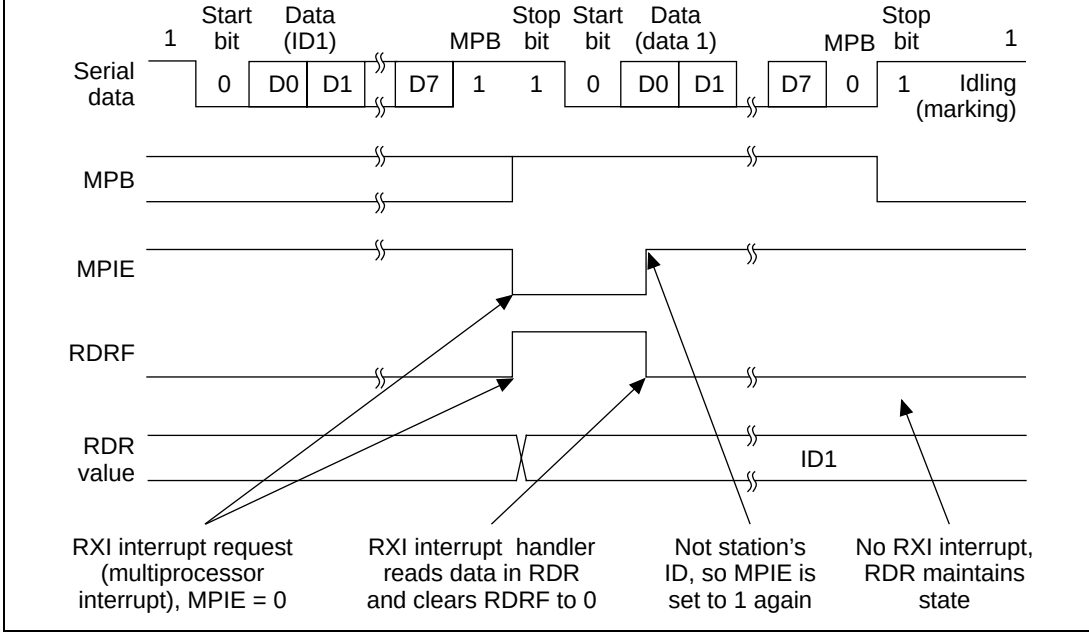


Figure 15.15 SCI Receive Operation (ID Does Not Match)
(Example: 8-Bit Data with Multiprocessor Bit and One Stop Bit)

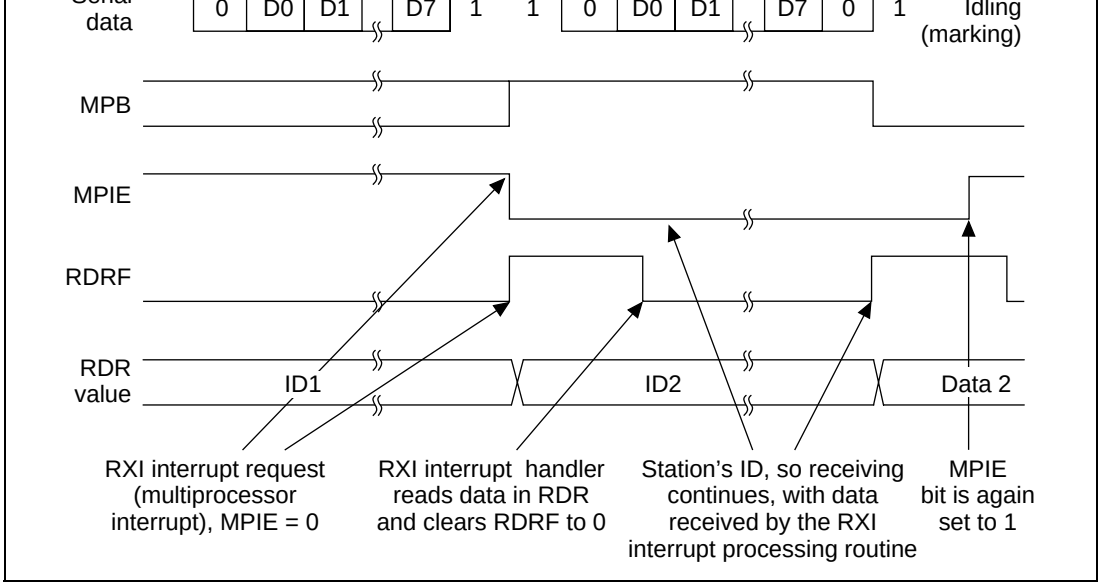


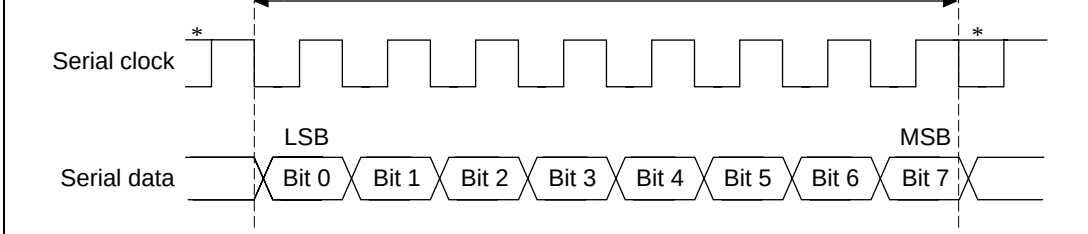
Figure 15.16 Example of SCI Receive Operation (ID Matches)
(Example: 8-Bit Data with Multiprocessor Bit and One Stop Bit)

15.3.4 Synchronous Operation

In synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full duplex communication is possible while sharing the same clock. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 15.17 shows the general format in synchronous serial communication.



Note: * High except in continuous transmitting or receiving.

Figure 15.17 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In synchronous mode, the SCI transmits or receives data by synchronizing with the rise of the serial clock.

Communication Format: The data length is fixed at eight bits. No parity bit or multiprocessor bit can be added.

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SMR) and bits CKE1 and CKE0 in the serial control register (SCR). See table 15.9.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state. An overrun error occurs only during the receive operation, and the serial clock is output until the RE bit is cleared to 0. To perform a receive operation in one-character units, select an external clock for the clock source.

the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as follows.

When changing the mode or communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

Figure 15.18 is a sample flowchart for initializing the SCI.

1. Select the clock source in the serial control register (SCR). Leave RIE, TIE, TEIE, MPIE, TE, and RE cleared to 0.
2. Select the communication format in the serial mode register (SMR) and serial direction control register (SDCR).
3. Write the value corresponding to the bit rate in the bit rate register (BRR) (unless an external clock is used).
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCR) to 1.* Also set RIE, TIE, TEIE, and MPIE. The TxD, RxD pins becomes usable in response to the PFC corresponding bits and the TE, RE bit settings.

Note: * In simultaneous transmit/receive operation, the TE bit and RE bit must be cleared to 0 or set to 1 simultaneously.

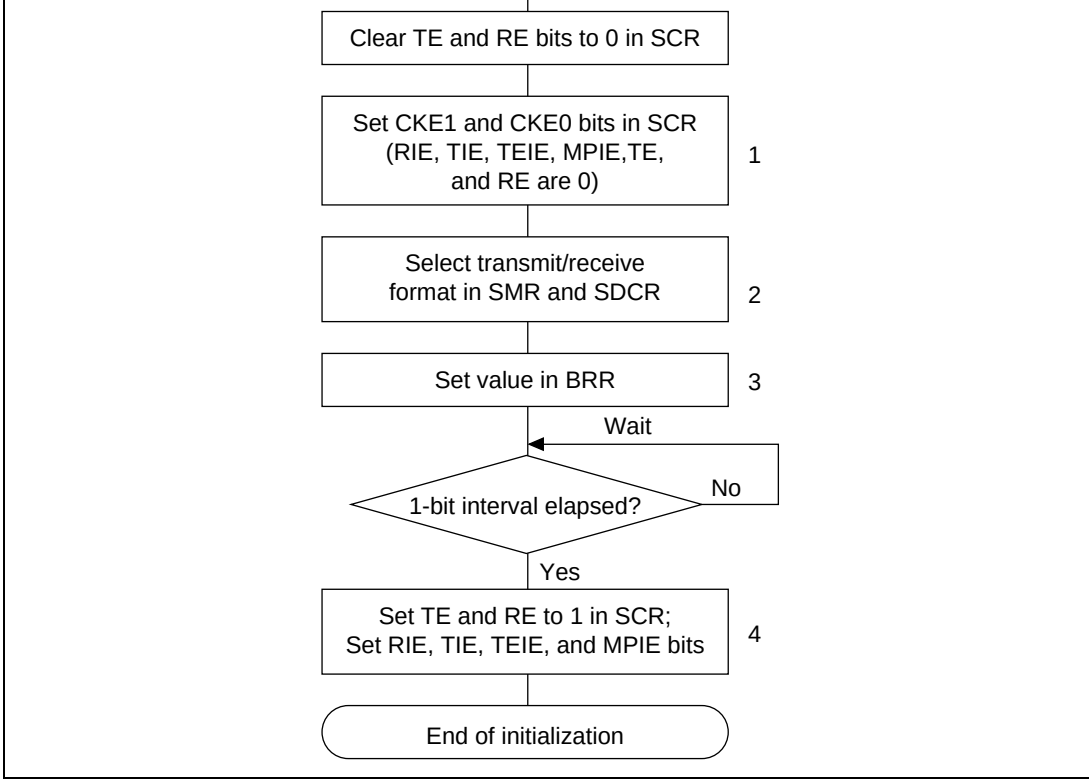


Figure 15.18 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Synchronous Mode): Figure 15.19 shows a sample flowchart for transmitting serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the TxD pin function with the PFC.
2. SCI status check and transmit data write: Read SSR, check that the TDRE flag is 1, then write transmit data in TDR and clear the TDRE flag to 0.
3. To continue transmitting serial data: After checking that the TDRE flag is 1, indicating that data can be written, write data in TDR, then clear the TDRE flag to 0. When the DMAC is activated by a transmit-data-empty interrupt request (TXI) to write data in TDR, the TDRE flag is checked and cleared automatically.

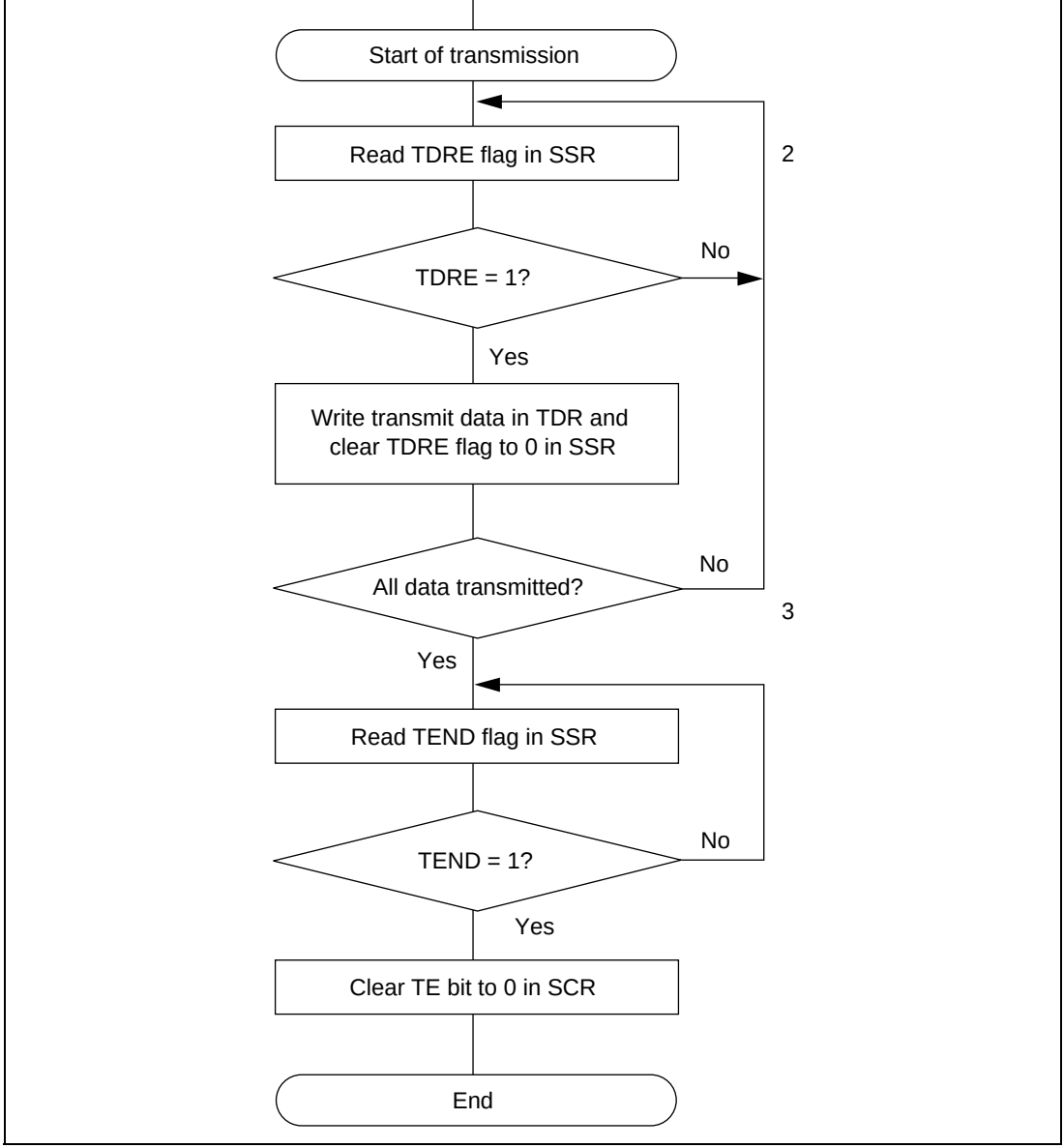


Figure 15.19 Sample Flowchart for Serial Transmitting

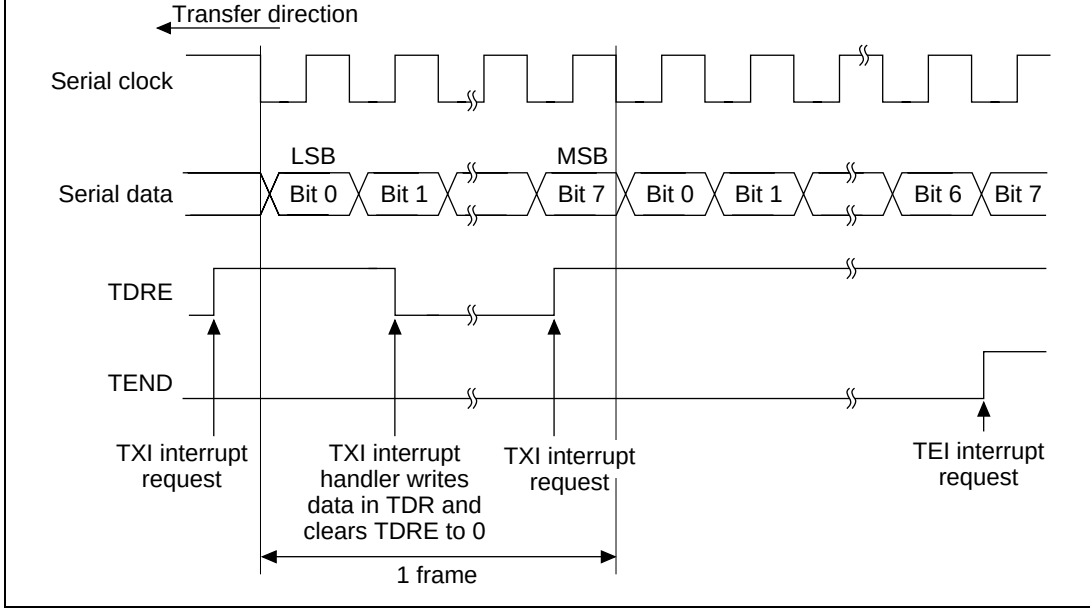


Figure 15.20 Example of SCI Transmit Operation

SCI serial transmission operates as follows.

1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data and loads this data from TDR into the transmit shift register (TSR).
2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

   If clock output mode is selected, the SCI outputs eight serial clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data is output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the SCI loads data from TDR into TSR, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in SSR to 1, transmits the MSB, then holds the transmit data pin (TxD) in the MSB state. If the transmit-end interrupt enable bit (TEIE) in SCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.
4. After the end of serial transmission, the SCK pin is held in the high state.

sure that ORER, PER, and FER are cleared to 0. If PER or FER is set to 1, the RDRF bit will not be set and both transmitting and receiving will be disabled.

The procedure for receiving serial data is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the RxD pin using the PFC.
2. Receive error handling: If a receive error occurs, read the ORER bit in SSR to identify the error. After executing the necessary error handling, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
3. SCI status check and receive data read: Read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
4. Continue receiving serial data: Read RDR, and clear RDRF to 0 before the MSB (bit 7) of the current frame is received. If the DMAC is started by a receive-data-full interrupt (RXI) to read RDR, the RDRF bit is cleared automatically so this step is unnecessary.

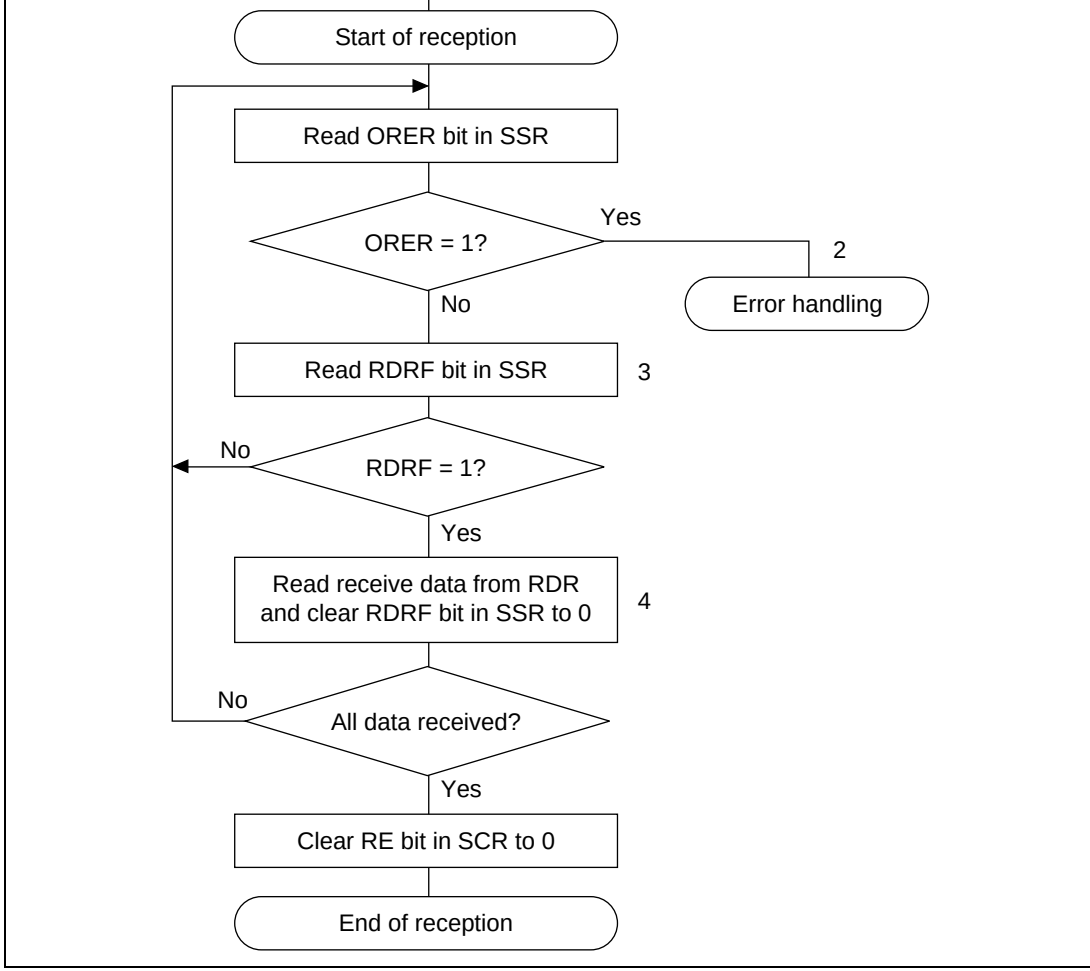


Figure 15.21 Sample Flowchart for Serial Receiving (1)

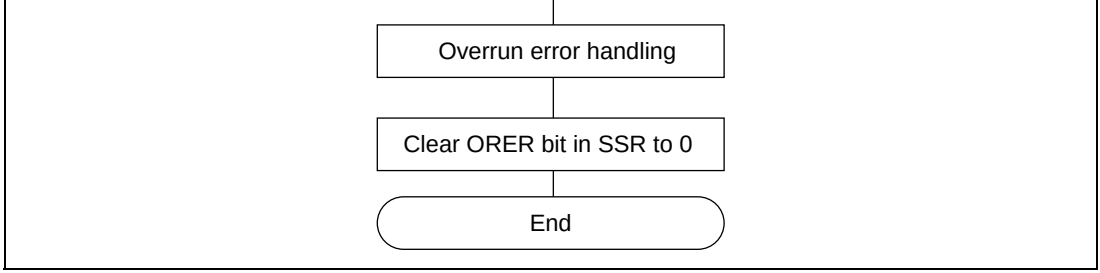


Figure 15.22 Sample Flowchart for Serial Receiving (2)

Figure 15.23 shows an example of the SCI receive operation.

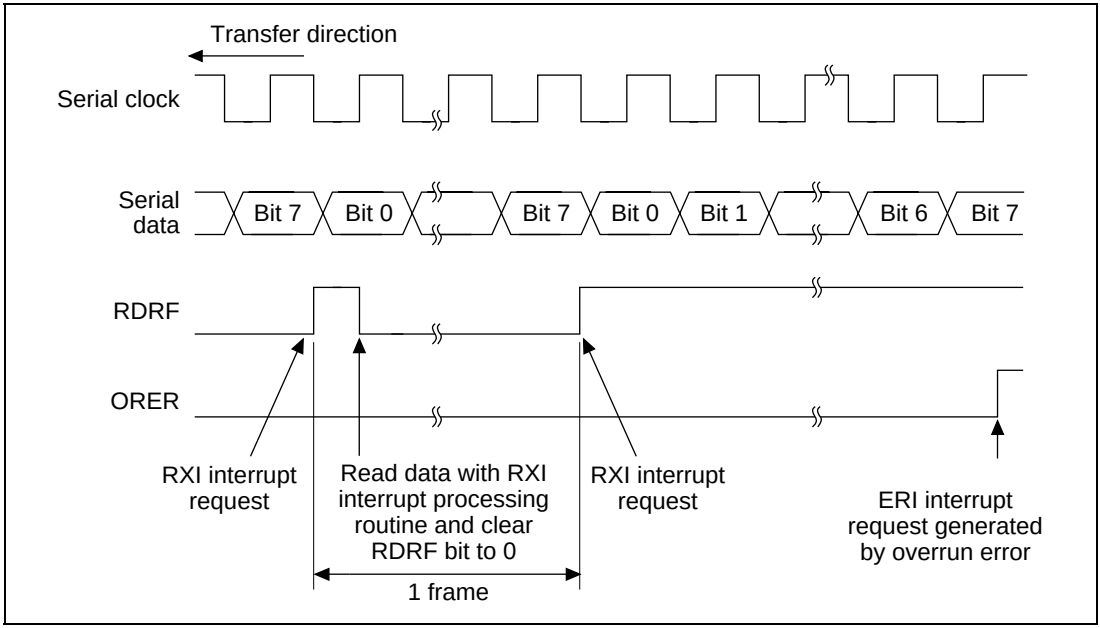


Figure 15.23 Example of SCI Receive Operation

In receiving, the SCI operates as follows:

1. The SCI synchronizes with serial clock input or output and initializes internally.
2. Receive data is shifted into RSR in order from the LSB to the MSB. After receiving the data, the SCI checks that RDRF is 0 so that receive data can be loaded from RSR into RDR. If this check passes, the SCI sets RDRF to 1 and stores the receive data in RDR. If the check does not pass (receive error), the SCI operates as indicated in table 15.11 and no further transmission or

3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) in SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode): Figure 15.24 shows a sample flowchart for transmitting and receiving serial data simultaneously. The procedure is as follows (the steps correspond to the numbers in the flowchart):

1. SCI initialization: Set the TxD and RxD pins using the PFC.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0. The TXI interrupt can also be used to determine if the TDRE bit has changed from 0 to 1.
3. Receive error handling: If a receive error occurs, read the ORER bit in SSR to identify the error. After executing the necessary error handling, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
4. SCI status check and receive data read: Read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
5. Continue transmitting and receiving serial data: Read the RDRF bit and RDR, and clear RDRF to 0 before the MSB (bit 7) of the current frame is received. Also read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in TDR, then clear TDRE to 0 before the MSB (bit 7) of the current frame is transmitted. When the DMAC is started by a transmit-data-empty interrupt request (TXI) to write data in TDR, the TDRE bit is checked and cleared automatically. When the DMAC is started by a receive-data-full interrupt (RXI) to read RDR, the RDRF bit is cleared automatically.

Note: In switching from transmitting or receiving to simultaneous transmitting and receiving, clear both TE and RE to 0, then set both TE and RE to 1 simultaneously.

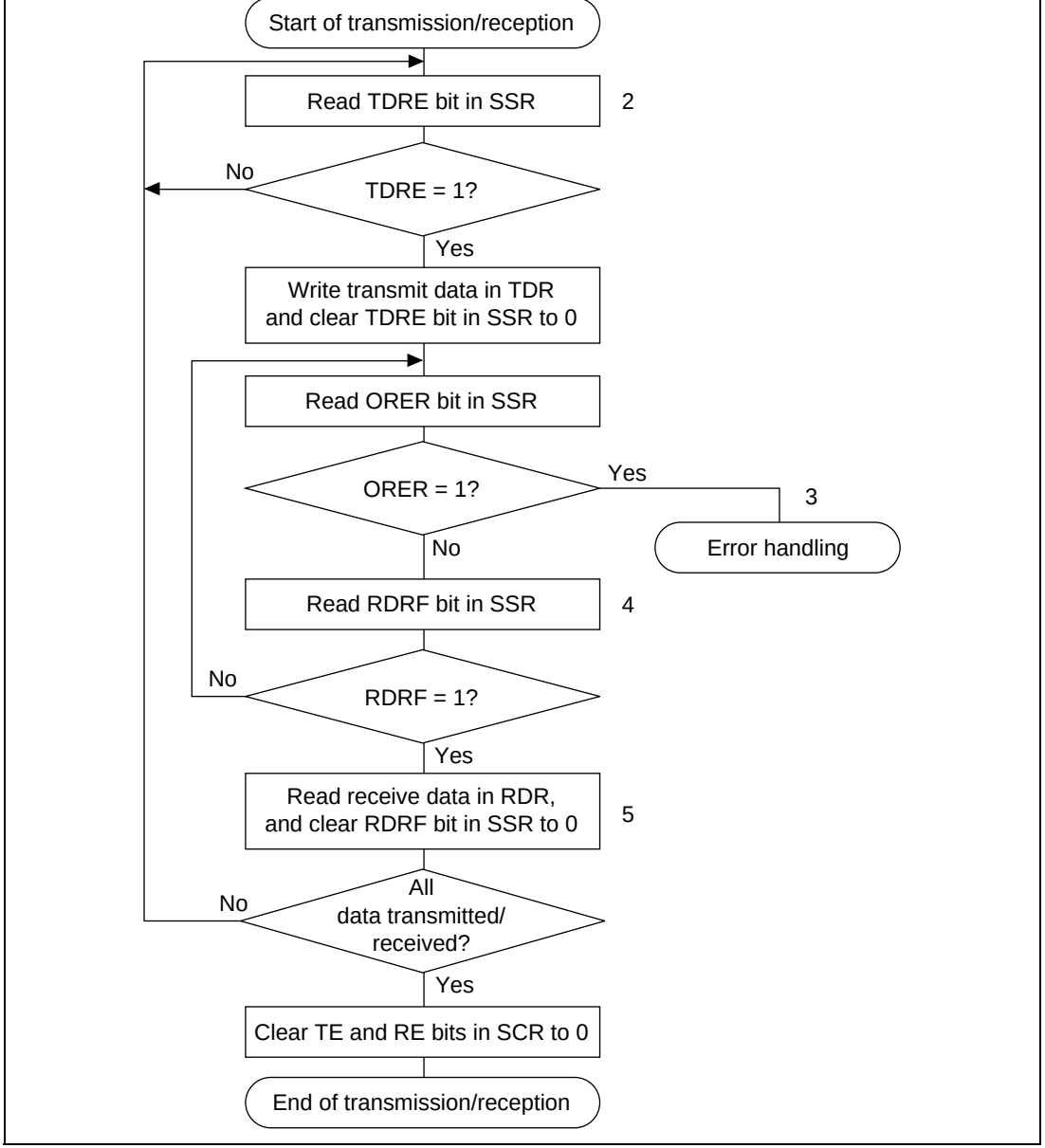


Figure 15.24 Sample Flowchart for Serial Transmission and Reception

The SCI has four interrupt sources: transmit-end (TEI), receive-error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI). Table 15.12 lists the interrupt sources and indicates their priority. These interrupts can be enabled and disabled by the TIE, RIE, and TEIE bits in the serial control register (SCR). Each interrupt request is sent separately to the interrupt controller.

TXI is requested when the TDRE bit in SSR is set to 1. TXI can start the direct memory access controller (DMAC) to transfer data. TDRE is automatically cleared to 0 when the DMAC writes data in the transmit data register (TDR).

RXI is requested when the RDRF bit in SSR is set to 1. RXI can start the DMAC to transfer data. RDRF is automatically cleared to 0 when the DMAC reads the receive data register (RDR).

ERI is requested when the ORER, PER, or FER bit in SSR is set to 1. ERI cannot start the DMAC.

TEI is requested when the TEND bit in SSR is set to 1. TEI cannot start the DMAC. Where the TXI interrupt indicates that transmit data writing is enabled, the TEI interrupt indicates that the transmit operation is complete.

Table 15.12 SCI Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority
ERI	Receive error (ORER, PER, or FER)	No	High
RXI	Receive data full (RDRF)	Yes	
TXI	Transmit data empty (TDRE)	Yes	
TEI	Transmit end (TEND)	No	Low

Sections 15.5.1 through 15.5.3 provide information concerning use of the SCI.

15.5.1 TDR Write and TDRE Flag

The TDRE bit in the serial status register (SSR) is a status flag indicating loading of transmit data from TDR into TSR. The SCI sets TDRE to 1 when it transfers data from TDR to TSR. Data can be written to TDR regardless of the TDRE bit status. If new data is written in TDR when TDRE is 0, however, the old data stored in TDR will be lost because the data has not yet been transferred to TSR. Before writing transmit data to TDR, be sure to check that TDRE is set to 1.

15.5.2 Simultaneous Multiple Receive Errors

Table 15.13 indicates the state of the SSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs, the RSR contents cannot be transferred to RDR, so receive data is lost.

Table 15.13 SSR Status Flags and Transfer of Receive Data

	SSR Status Flags				Receive Data Trans
Receive Error Status	**RDRF**	**ORER**	**FER**	**PER**	**RSR → RDR**
Overrun error	1	1	0	0	X
Framing error	0	0	1	0	O
Parity error	0	0	0	1	O
Overrun error + framing error	1	1	1	0	X
Overrun error + parity error	1	1	0	1	X
Framing error + parity error	0	0	1	1	O
Overrun error + framing error + parity error	1	1	1	1	X

Notes: O: Receive data is transferred from RSR to RDR.
X: Receive data is not transferred from RSR to RDR.

Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state, the input from the RxD pin consists of all 0s, so FER is set and the parity error flag (PER) may also be set. In the break state, the SCI receiver continues to operate, so if the FER bit is cleared to 0, it will be set to 1 again.

15.5.4 Sending a Break Signal

The TxD pin becomes a general I/O pin with the I/O direction and level determined by the I/O port data register (DR) and pin function controller (PFC) control register (CR). These conditions allow break signals to be sent. The DR value is substituted for the marking status until the PFC is set. Consequently, the output port is set to initially output a 1. To send a break in serial transmission, first clear the DR to 0, then establish the TxD pin as an output port using the PFC. When TE is cleared to 0, the transmission section is initialized regardless of the present transmission status.

15.5.5 Receive Error Flags and Transmitter Operation (Synchronous Mode Only)

When a receive error flag (ORER, PER, or FER) is set to 1, the SCI will not start transmitting even if TDRE is set to 1. Be sure to clear the receive error flags to 0 before starting to transmit. Note that clearing RE to 0 does not clear the receive error flags.

15.5.6 Receive Data Sampling Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times the transfer rate. In receiving, the SCI synchronizes internally with the falling edge of the start bit, which it samples on the base clock. Receive data is latched on the rising edge of the eighth base clock pulse (figure 15.25).

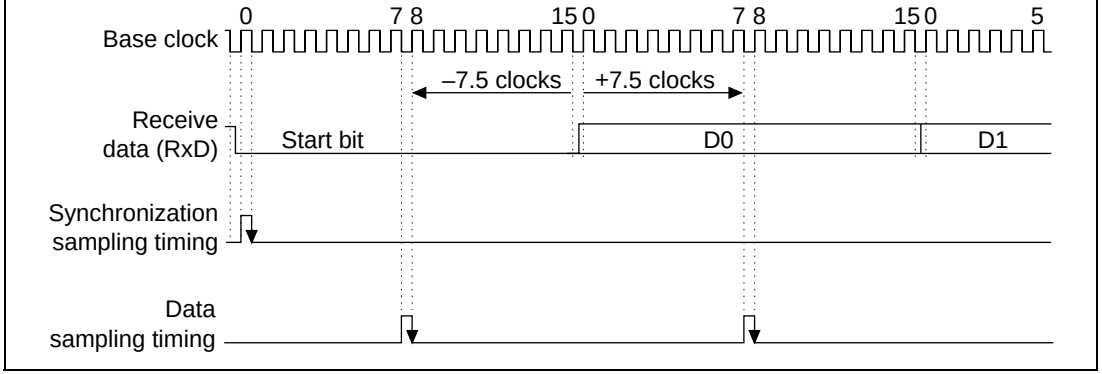


Figure 15.25 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as:

$$M = \left|\left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\%$$

M : Receive margin (%)
N : Ratio of clock frequency to bit rate (N = 16)
D : Clock duty cycle (D = 0–1.0)
L : Frame length (L = 9–12)
F : Absolute deviation of clock frequency

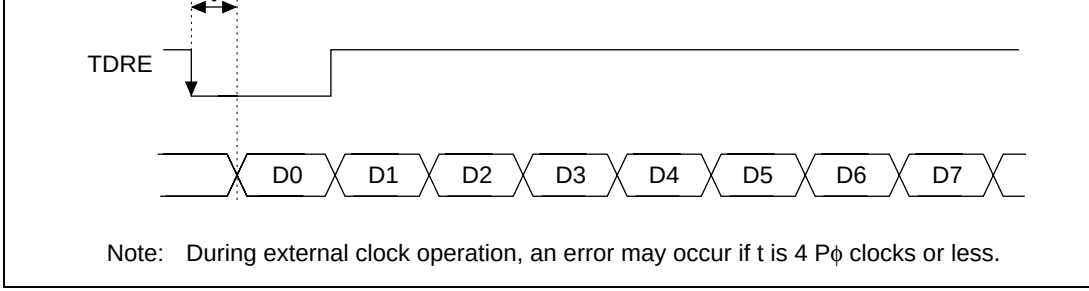
From the equation above, if F = 0 and D = 0.5 the receive margin is 46.875%:

D = 0.5, F = 0
M = $(0.5 - 1/(2 \times 16)) \times 100\%$
= 46.875%

This is a theoretical value. A reasonable margin to allow in system designs is 20–30%.

15.5.7 Constraints on DMAC Use

- When using an external clock source for the serial clock, update TDR with the DMAC, and then after the elapse of five peripheral clocks (Pϕ) or more, input a transmit clock. If a transmit clock is input in the first four Pϕ clocks after TDR is written, an error may occur (figure 15.26).
- Before reading the receive data register (RDR) with the DMAC, select the receive-data-full (RXI) interrupt of the SCI as a start-up source.



Note: During external clock operation, an error may occur if t is 4 Pϕ clocks or less.

Figure 15.26 Example of Synchronous Transmission with DMAC

15.5.8 Cautions on Synchronous External Clock Mode

- Set TE = RE = 1 only when external clock SCK is 1.
- Do not set TE = RE = 1 until at least four Pϕ clocks after external clock SCK has changed from 0 to 1.
- When receiving, RDRF is 1 when RE is cleared to zero 2.5–3.5 Pϕ clocks after the rising edge of the RxD D7 bit SCK input, but copying to RDR is not possible.

15.5.9 Caution on Synchronous Internal Clock Mode

When receiving, RDRF is 1 when RE is cleared to zero 1.5 Pϕ clocks after the rising edge of the RxD D7 bit SCK output, but copying to RDR is not possible.

16.1 Overview

The HCAN is a module for controlling a controller area network (CAN) for realtime communication in vehicular and industrial equipment systems, etc. The SH7055SF has a 2-channel on-chip HCAN module.

Reference: Bosch CAN Specification Version 2.0 1991, Robert Bosch GmbH

16.1.1 Features

The HCAN has the following features:

- CAN version: Bosch 2.0B active compatible
  - Communication systems:
    NRZ (Non-Return to Zero) system (with bit-stuffing function)
    Broadcast communication system
  - Transmission path: Bidirectional 2-wire serial communication
  - Communication speed: Max. 1 Mbps (at 40 MHz operation)
  - Data length: 0 to 8 bytes
- Number of channels: 2 (HCAN0, HCAN1)
- Data buffers: 16 per channel (one receive-only buffer and 15 buffers settable for transmission/reception)
- Data transmission: Choice of two methods:
  - Mailbox (buffer) number order (low-to-high)
  - Message priority (identifier) high-to-low order
- Data reception: Two methods:
  - Message identifier match (transmit/receive-setting buffers)
  - Reception with message identifier masked (receive-only)
- CPU interrupts: Four independent interrupt vectors per channel:
  - Error interrupt
  - Reset processing interrupt
  - Message reception interrupt
  - Message transmission interrupt
- HCAN operating modes: Support for various modes:
  - Hardware reset
  - Software reset
  - Normal status (error-active, error-passive)

— HCAN sleep mode
— HCAN halt mode

- HCAN connection methods: Choice of two methods of use:
  — Two-channel 16-buffer HCANs (two transmit pins, two receive pins)
  — One-channel 32-buffer HCAN (wired-AND) (one transmit pin, one receive pin)
- Other features: DMAC can be activated by message reception mailbox (HCAN0 mailbox 0 only)

Figure 16.1 shows a block diagram of the HCAN.

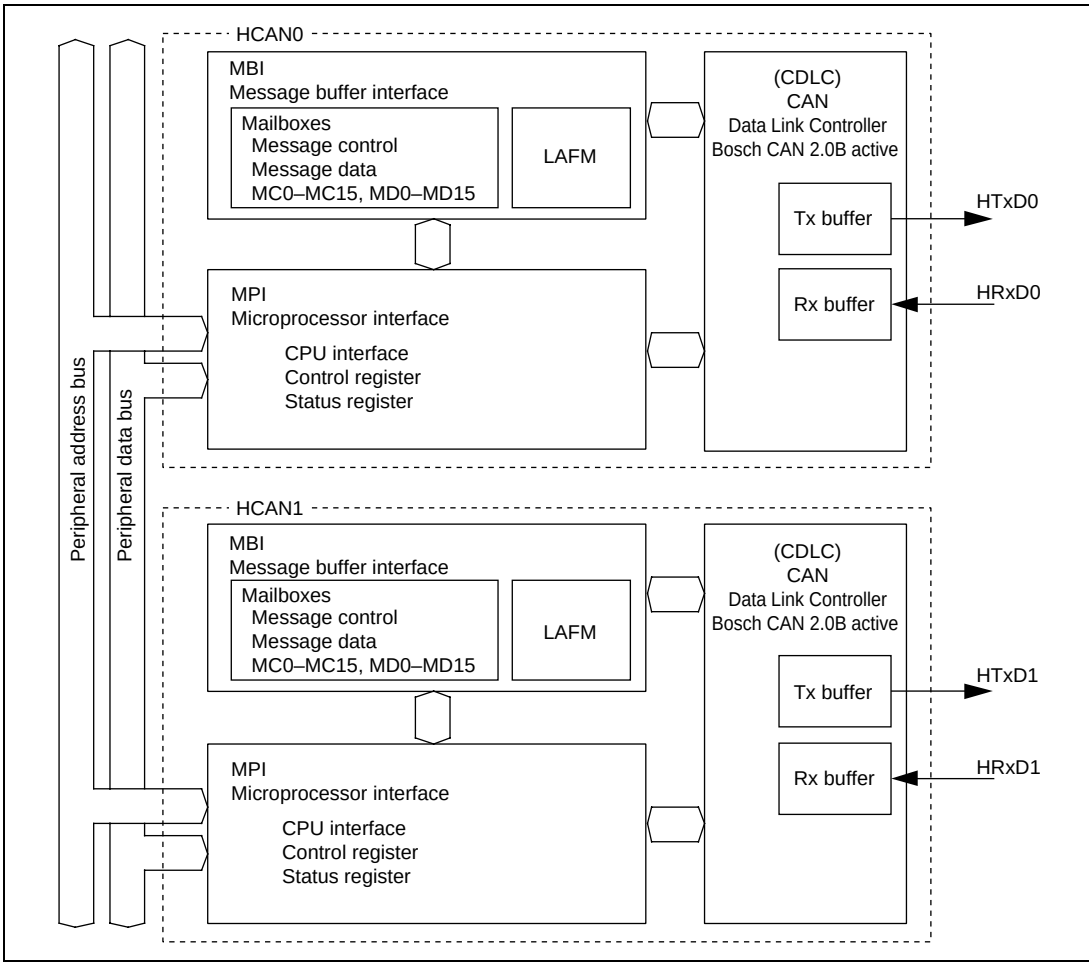


Figure 16.1 HCAN Block Diagram

Message Buffer Interface (MBI): The MBI, consisting of mailboxes and a local acceptance filter mask (LAFM), stores CAN transmit/receive messages (identifiers, data, etc.). Transmit messages are written by the CPU. For receive messages, the data received by the CDLC is stored automatically.

Microprocessor Interface (MPI): The MPI, consisting of a bus interface, control register, status register, etc., controls HCAN internal data, statuses, and so forth.

CAN Data Link Controller (CDLC): The CDLC performs transmission and reception of messages conforming to the Bosch CAN Ver. 2.0B active standard (data frames, remote frames,

16.1.3 Pin Configuration

Table 16.1 shows the HCAN's pins. When using the functions of these external pins, the pin function controller (PFC) must also be set in line with the HCAN settings.

When using HCAN pins, settings must be made in the HCAN configuration mode (during initialization: MCR0 = 1 and GSR3 = 1).

Table 16.1 HCAN Pins

Channel	Name	Abbreviation	Input/Output	Function
0	HCAN transmit data pin 0	HTxD0	Output	Channel 0 CAN bus transmission pin
	HCAN receive data pin 0	HRxD0	Input	Channel 0 CAN bus reception pin
1	HCAN transmit data pin 1	HTxD1	Output	Channel 1 CAN bus transmission pin
	HCAN receive data pin 1	HRxD1	Input	Channel 1 CAN bus reception pin

A bus transceiver IC is necessary between the pins and the CAN bus. A Philips PCA82C250 compatible model is recommended.

These pins are multiplexed, and can be set in either of the following ways.

- Setting each channel as an independent 16-message-buffer HCAN (two HCAN channels: two transmit pins and two receive pins)
- Setting one HCAN channel, using wired-AND connection of the pins for the two channels (one 32-message-buffer HCAN channel: one transmit pin and one receive pin)

See section 16.3, Operation, for details.

The pin numbers of the pins that can be set for each channel are shown in table 16.2.

Table 16.2 Pin Numbers of Pins Settable as HCAN Pins

	HCAN0	HCAN1	HCAN0,1 (Wired AND)
	16 Message Buffers	16 Message Buffers	32 Message Buffers
HTxD	6,157,228	6,228	228
HRxD	158,170,229	170,229	229

Table 16.3 lists the HCAN's registers.

Table 16.3 HCAN Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Master control register	MCR	R/W	H'01	H'FFFF E400	8 bits 16 bits
	General status register	GSR	R	H'0C	H'FFFF E401	8 bits
	Bit configuration register	BCR	R/W	H'0000	H'FFFF E402	8/16 bits
	Mailbox configuration register	MBCR	R/W	H'0100	H'FFFF E404	8/16 bits
	Transmit wait register	TXPR	R/W	H'0000	H'FFFF E406	8/16 bits
	Transmit wait cancel register	TXCR	R/W	H'0000	H'FFFF E408	8/16 bits
	Transmit acknowledge register	TXACK	R/W	H'0000	H'FFFF E40A	8/16 bits
	Abort acknowledge register	ABACK	R/W	H'0000	H'FFFF E40C	8/16 bits
	Receive complete register	RXPR	R/W	H'0000	H'FFFF E40E	8/16 bits
	Remote request register	RFPR	R/W	H'0000	H'FFFF E410	8/16 bits
	Interrupt register	IRR	R/W	H'0100	H'FFFF E412	8/16 bits
	Mailbox interrupt mask register	MBIMR	R/W	H'FFFF	H'FFFF E414	8/16 bits
	Interrupt mask register	IMR	R/W	H'FEFF	H'FFFF E416	8/16 bits
	Receive error counter	REC	R	H'00	H'FFFF E418	8 bits 16 bits
	Transmit error counter	TEC	R	H'00	H'FFFF E419	8 bits
	Unread message status register	UMSR	R/W	H'0000	H'FFFF E41A	8/16 bits
	Local acceptance filter mask L	LAFML	R/W	H'0000	H'FFFF E41C	8/16 bits
	Local acceptance filter mask H	LAFMH	R/W	H'0000	H'FFFF E41E	8/16 bits

nel	Name	viation	R/W	Value	Address	Access Size
0	Message control 0 [1:8]	MC0 [1:8]	R/W	Undefined	H'FFFF E420	8/16 bits
	Message control 1 [1:8]	MC1 [1:8]	R/W	Undefined	H'FFFF E428	8/16 bits
	Message control 2 [1:8]	MC2 [1:8]	R/W	Undefined	H'FFFF E430	8/16 bits
	Message control 3 [1:8]	MC3 [1:8]	R/W	Undefined	H'FFFF E438	8/16 bits
	Message control 4 [1:8]	MC4 [1:8]	R/W	Undefined	H'FFFF E440	8/16 bits
	Message control 5 [1:8]	MC5 [1:8]	R/W	Undefined	H'FFFF E448	8/16 bits
	Message control 6 [1:8]	MC6 [1:8]	R/W	Undefined	H'FFFF E450	8/16 bits
	Message control 7 [1:8]	MC7 [1:8]	R/W	Undefined	H'FFFF E458	8/16 bits
	Message control 8 [1:8]	MC8 [1:8]	R/W	Undefined	H'FFFF E460	8/16 bits
	Message control 9 [1:8]	MC9 [1:8]	R/W	Undefined	H'FFFF E468	8/16 bits
	Message control 10 [1:8]	MC10 [1:8]	R/W	Undefined	H'FFFF E470	8/16 bits
	Message control 11 [1:8]	MC11 [1:8]	R/W	Undefined	H'FFFF E478	8/16 bits
	Message control 12 [1:8]	MC12 [1:8]	R/W	Undefined	H'FFFF E480	8/16 bits
	Message control 13 [1:8]	MC13 [1:8]	R/W	Undefined	H'FFFF E488	8/16 bits
	Message control 14 [1:8]	MC14 [1:8]	R/W	Undefined	H'FFFF E490	8/16 bits
	Message control 15 [1:8]	MC15 [1:8]	R/W	Undefined	H'FFFF E498	8/16 bits
	Message data 0 [1:8]	MD0 [1:8]	R/W	Undefined	H'FFFF E4B0	8/16 bits
	Message data 1 [1:8]	MD1 [1:8]	R/W	Undefined	H'FFFF E4B8	8/16 bits
	Message data 2 [1:8]	MD2 [1:8]	R/W	Undefined	H'FFFF E4C0	8/16 bits
	Message data 3 [1:8]	MD3 [1:8]	R/W	Undefined	H'FFFF E4C8	8/16 bits
	Message data 4 [1:8]	MD4 [1:8]	R/W	Undefined	H'FFFF E4D0	8/16 bits
	Message data 5 [1:8]	MD5 [1:8]	R/W	Undefined	H'FFFF E4D8	8/16 bits
	Message data 6 [1:8]	MD6 [1:8]	R/W	Undefined	H'FFFF E4E0	8/16 bits
	Message data 7 [1:8]	MD7 [1:8]	R/W	Undefined	H'FFFF E4E8	8/16 bits
	Message data 8 [1:8]	MD8 [1:8]	R/W	Undefined	H'FFFF E4F0	8/16 bits
	Message data 9 [1:8]	MD9 [1:8]	R/W	Undefined	H'FFFF E4F8	8/16 bits
	Message data 10 [1:8]	MD10 [1:8]	R/W	Undefined	H'FFFF E500	8/16 bits
	Message data 11 [1:8]	MD11 [1:8]	R/W	Undefined	H'FFFF E508	8/16 bits
	Message data 12 [1:8]	MD12 [1:8]	R/W	Undefined	H'FFFF E510	8/16 bits
	Message data 13 [1:8]	MD13 [1:8]	R/W	Undefined	H'FFFF E518	8/16 bits
	Message data 14 [1:8]	MD14 [1:8]	R/W	Undefined	H'FFFF E520	8/16 bits
	Message data 15 [1:8]	MD15 [1:8]	R/W	Undefined	H'FFFF E528	8/16 bits

nel	Name	viation	R/W	Value	Address	Access Size
1	Master control register	MCR	R/W	H'01	H'FFFF E600	8 bits 16 bits
	General status register	GSR	R	H'0C	H'FFFF E601	8 bits
	Bit configuration register	BCR	R/W	H'0000	H'FFFF E602	8/16 bits
	Mailbox configuration register	MBCR	R/W	H'0100	H'FFFF E604	8/16 bits
	Transmit wait register	TXPR	R/W	H'0000	H'FFFF E606	8/16 bits
	Transmit wait cancel register	TXCR	R/W	H'0000	H'FFFF E608	8/16 bits
	Transmit acknowledge register	TXACK	R/W	H'0000	H'FFFF E60A	8/16 bits
	Abort acknowledge register	ABACK	R/W	H'0000	H'FFFF E60C	8/16 bits
	Receive complete register	RXPR	R/W	H'0000	H'FFFF E60E	8/16 bits
	Remote request register	RFPR	R/W	H'0000	H'FFFF E610	8/16 bits
	Interrupt register	IRR	R/W	H'0100	H'FFFF E612	8/16 bits
	Mailbox interrupt mask register	MBIMR	R/W	H'FFFF	H'FFFF E614	8/16 bits
	Interrupt mask register	IMR	R/W	H'FEFF	H'FFFF E616	8/16 bits
	Receive error counter	REC	R	H'00	H'FFFF E618	8 bits 16 bits
	Transmit error counter	TEC	R	H'00	H'FFFF E619	8 bits
	Unread message status register	UMSR	R/W	H'0000	H'FFFF E61A	8/16 bits
	Local acceptance filter mask L	LAFML	R/W	H'0000	H'FFFF E61C	8/16 bits
	Local acceptance filter mask H	LAFMH	R/W	H'0000	H'FFFF E61E	8/16 bits

nel	Name	viation	R/W	Value	Address	Access Size
1	Message control 0 [1:8]	MC0 [1:8]	R/W	Undefined	H'FFFF E620	8/16 bits
	Message control 1 [1:8]	MC1 [1:8]	R/W	Undefined	H'FFFF E628	8/16 bits
	Message control 2 [1:8]	MC2 [1:8]	R/W	Undefined	H'FFFF E630	8/16 bits
	Message control 3 [1:8]	MC3 [1:8]	R/W	Undefined	H'FFFF E638	8/16 bits
	Message control 4 [1:8]	MC4 [1:8]	R/W	Undefined	H'FFFF E640	8/16 bits
	Message control 5 [1:8]	MC5 [1:8]	R/W	Undefined	H'FFFF E648	8/16 bits
	Message control 6 [1:8]	MC6 [1:8]	R/W	Undefined	H'FFFF E650	8/16 bits
	Message control 7 [1:8]	MC7 [1:8]	R/W	Undefined	H'FFFF E658	8/16 bits
	Message control 8 [1:8]	MC8 [1:8]	R/W	Undefined	H'FFFF E660	8/16 bits
	Message control 9 [1:8]	MC9 [1:8]	R/W	Undefined	H'FFFF E668	8/16 bits
	Message control 10 [1:8]	MC10 [1:8]	R/W	Undefined	H'FFFF E670	8/16 bits
	Message control 11 [1:8]	MC11 [1:8]	R/W	Undefined	H'FFFF E678	8/16 bits
	Message control 12 [1:8]	MC12 [1:8]	R/W	Undefined	H'FFFF E680	8/16 bits
	Message control 13 [1:8]	MC13 [1:8]	R/W	Undefined	H'FFFF E688	8/16 bits
	Message control 14 [1:8]	MC14 [1:8]	R/W	Undefined	H'FFFF E690	8/16 bits
	Message control 15 [1:8]	MC15 [1:8]	R/W	Undefined	H'FFFF E698	8/16 bits
	Message data 0 [1:8]	MD0 [1:8]	R/W	Undefined	H'FFFF E6B0	8/16 bits
	Message data 1 [1:8]	MD1 [1:8]	R/W	Undefined	H'FFFF E6B8	8/16 bits
	Message data 2 [1:8]	MD2 [1:8]	R/W	Undefined	H'FFFF E6C0	8/16 bits
	Message data 3 [1:8]	MD3 [1:8]	R/W	Undefined	H'FFFF E6C8	8/16 bits
	Message data 4 [1:8]	MD4 [1:8]	R/W	Undefined	H'FFFF E6D0	8/16 bits
	Message data 5 [1:8]	MD5 [1:8]	R/W	Undefined	H'FFFF E6D8	8/16 bits
	Message data 6 [1:8]	MD6 [1:8]	R/W	Undefined	H'FFFF E6E0	8/16 bits
	Message data 7 [1:8]	MD7 [1:8]	R/W	Undefined	H'FFFF E6E8	8/16 bits
	Message data 8 [1:8]	MD8 [1:8]	R/W	Undefined	H'FFFF E6F0	8/16 bits
	Message data 9 [1:8]	MD9 [1:8]	R/W	Undefined	H'FFFF E6F8	8/16 bits
	Message data 10 [1:8]	MD10 [1:8]	R/W	Undefined	H'FFFF E700	8/16 bits
	Message data 11 [1:8]	MD11 [1:8]	R/W	Undefined	H'FFFF E708	8/16 bits
	Message data 12 [1:8]	MD12 [1:8]	R/W	Undefined	H'FFFF E710	8/16 bits
	Message data 13 [1:8]	MD13 [1:8]	R/W	Undefined	H'FFFF E718	8/16 bits
	Message data 14 [1:8]	MD14 [1:8]	R/W	Undefined	H'FFFF E720	8/16 bits
	Message data 15 [1:8]	MD15 [1:8]	R/W	Undefined	H'FFFF E728	8/16 bits

16.2.1 Master Control Register (MCR)

The master control register (MCR) is an 8-bit readable/writable register that controls the CAN interface.

Bit:	7	6	5	4	3	2	1	0
	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R	R/W	R	R	R/W	R/W	R/W

- Bit 7—HCAN Sleep Mode Release (MCR7): Enables or disables HCAN sleep mode release by bus operation.

Bit 7: MCR7	Description	
0	HCAN sleep mode release by CAN bus operation disabled	(Initial value)
1	HCAN sleep mode release by CAN bus operation enabled	

- Bit 6—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 5—HCAN Sleep Mode (MCR5): Enables or disables HCAN sleep mode transition.

Bit 5: MCR5	Description	
0	HCAN sleep mode released	(Initial value)
1	Transition to HCAN sleep mode enabled	

- Bits 4 and 3—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 2—Message Transmission Method (MCR2): Selects the transmission method for transmit messages.

Bit 2: MCR2	Description	
0	Transmission order determined by message identifier priority	(Initial value)
1	Transmission order determined by mailbox (buffer) number priority (TXPR1 > TXPR15)	

0	Normal operating mode	(Initial value)
1	Halt mode transition request	

- Bit 0—Reset Request (MCR0): Controls resetting of the HCAN module.

Bit 0: MCR0	Description	
0	Normal operating mode (MCR0 = 0 and GSR3 = 0) [Setting condition] When 0 is written after an HCAN reset	
1	Reset mode transition request	(Initial value)

In order for GSR3 to change from 1 to 0 after 0 is written to MCR0, time is required before the HCAN is internally reset. There is consequently a delay before GSR3 is cleared to 0 after MCR0 is cleared to 0.

16.2.2 General Status Register (GSR)

The general status register (GSR) is an 8-bit readable register that indicates the status of the CAN bus.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	GSR3	GSR2	GSR1	GSR0
Initial value:	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R

- Bits 7 to 4—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 3—Reset Status Bit (GSR3): Indicates whether the HCAN module is in the normal operating state or the reset state. This bit cannot be modified.

Bit 3: GSR3	Description	
0	Normal operating state [Setting condition] After an HCAN internal reset	
1	Configuration mode [Reset condition] MCR0-initiated reset state or sleep mode	(Initial value)

from the start of message transmission (SOF) until the end of a 3-bit intermission interval after EOF (End of Frame). This bit cannot be modified.

Bit 2: GSR2	Description
0	Transmission in progress
1	[Reset condition] Idle period (Initial value)

- Bit 1—Transmit/Receive Warning Flag (GSR1): Flag that indicates an error warning. This bit cannot be modified.

it 1: GSR1	Description
0	[Reset condition] When TEC < 96 and REC < 96 or TEC ≥ 256 (Initial value)
1	When TEC ≥ 96 or REC ≥ 96

- Bit 0—Bus Off Flag (GSR0): Flag that indicates the bus off state. This bit cannot be modified.

Bit 0: GSR0	Description
0	[Reset condition] Recovery from bus off state (Initial value)
1	When TEC ≥ 256 (bus off state)

16.2.3 Bit Configuration Register (BCR)

The bit configuration register (BCR) is a 16-bit readable/writable register that is used to set CAN bit timing parameters and the baud rate prescaler.

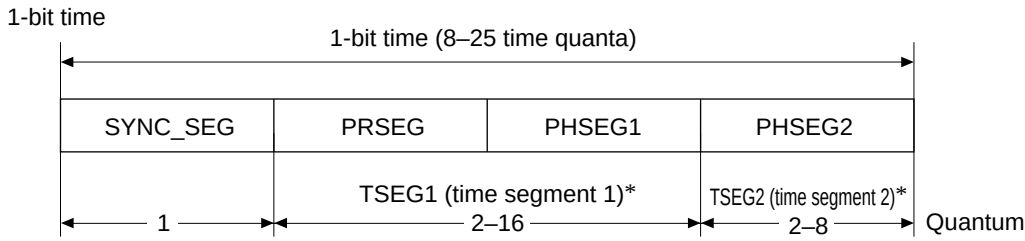
Bit:	15	14	13	12	11	10	9	8
	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15: BCR7	Bit 14: BCR6	Description
0	0	Maximum bit synchronization width = 1 time quantum
	1	Maximum bit synchronization width = 2 time quanta
1	0	Maximum bit synchronization width = 3 time quanta
	1	Maximum bit synchronization width = 4 time quanta

- Bits 13 to 8—Baud Rate Prescaler (BRP): These bits are used to set the CAN bus baud rate.

Bit 13: BCR5	Bit 12: BCR4	Bit 11: BCR3	Bit 10: BCR2	Bit 9: BCR1	Bit 8: BCR0	Description	
0	0	0	0	0	0	2 × system clock	(Initial value)
0	0	0	0	0	1	4 × system clock	
0	0	0	0	1	0	6 × system clock	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
1	1	1	1	1	1	128 × system clock	



SYNC_SEG: Segment for establishing synchronization of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)
PRSEG: Segment for compensating for physical delay between networks.
PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended when synchronization (re-synchronization) is established.)
PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronization (re-synchronization) is established.)

Note: * The time quanta value for TSEG1 and TSEG2 is the TSEG value + 1.

Figure 16.2 Detailed Description of One Bit

$$\text{Bit rate [b/s]} = \frac{f_{CLK}}{2 \times (BRP + 1) \times (3 + TSEG1 + TSEG2)}$$

Note: f_{CLK} = Pϕ (peripheral clock (ϕ/2))
The BCR values are used for BRP, TSEG1, and TSEG2.

BCR Setting Constraints

TSEG1 > TSEG2 = SJW (SJW = 1 to 4)

3 + TSEG1 + TSEG2 = 8 to 25 time quanta

TSEG2 > B'001 (BRP = B'000000)

TSEG2 > B'000 (BRP > B'000000)

These constraints allow the setting range shown in table 16.4 for TSEG1 and TSEG2 in BCR.

Table 16.4 Setting Range for TSEG1 and TSEG2 in BCR

		TSEG2 (BCR [14:12])						
		001	**010**	**011**	**100**	**101**	**110**	**111**
TSEG1 (BCR [11:8])	**0011**	No	Yes	No	No	No	No	No
	0100	Yes*	Yes	Yes	No	No	No	No
	0101	Yes*	Yes	Yes	Yes	No	No	No
	0110	Yes*	Yes	Yes	Yes	Yes	No	No
	0111	Yes*	Yes	Yes	Yes	Yes	Yes	No
	1000	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1001	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1010	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1011	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1100	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1101	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1110	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1111	Yes*	Yes	Yes	Yes	Yes	Yes	Yes

Notes: The time quanta value for TSEG1 and TSEG2 is the TSEG value + 1.
* Setting is enabled except when BRP[13:8] = B'000000.

0	Bit sampling at one point (end of time segment 1 (TSEG1))	(Initial value)
1	Bit sampling at three points (end of time segment 1 (TSEG1), and 1 time quantum before and after)	

- Bits 6 to 4—Time Segment 2 (TSEG2): These bits are used to set the segment for correcting 1-bit time error. A value from 2 to 8 can be set.

Bit 6: BCR14	Bit 5: BCR13	Bit 4: BCR12	Description	
0	0	0	Setting prohibited	(Initial value)
		1	TSEG2 (PHSEG2) = 2 time quanta	
	1	0	TSEG2 (PHSEG2) = 3 time quanta	
		1	TSEG2 (PHSEG2) = 4 time quanta	
1	0	0	TSEG2 (PHSEG2) = 5 time quanta	
		1	TSEG2 (PHSEG2) = 6 time quanta	
	1	0	TSEG2 (PHSEG2) = 7 time quanta	
		1	TSEG2 (PHSEG2) = 8 time quanta	

- Bits 3 to 0—Time Segment 1 (TSEG1): These bits are used to set the segment for absorbing output buffer, CAN bus, and input buffer delay. A value from 4 to 16 can be set.

Bit 3: BCR11	Bit 2: BCR10	Bit 1: BCR9	Bit 0: BCR8	Description	
0	0	0	0	Setting prohibited	(Initial value)
0	0	0	1	Setting prohibited	
0	0	1	0	Setting prohibited	
0	0	1	1	TSEG1 (PRSEG + PHSEG1) = 4 time quanta	
0	1	0	0	TSEG1 (PRSEG + PHSEG1) = 5 time quanta	
.	.	.	.	.	
.	.	.	.	.	
.	.	.	.	.	
1	1	1	1	TSEG1 (PRSEG + PHSEG1) = 16 time quanta	

The mailbox configuration register (MBCR) is a 16-bit readable/writable register that is used to set mailbox (buffer) transmission/reception.

Bit:	15	14	13	12	11	10	9	8
	MBCR7	MBCR6	MBCR5	MBCR4	MBCR3	MBCR2	MBCR1	—
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	7	6	5	4	3	2	1	0
	MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9	MBCR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 9 and 7 to 0—Mailbox Setting Register (MBCR7 to 1, MBCR15 to 8): These bits set the polarity of the corresponding mailboxes.

Bit x: MBCRx	Description	
0	Corresponding mailbox is set for transmission	(Initial value)
1	Corresponding mailbox is set for reception	

- Bit 8—Reserved: This bit always reads 1. The write value should always be 1.

16.2.5 Transmit Wait Register (TXPR)

The transmit wait register (TXPR) is a 16-bit readable/writable register that is used to set a transmit wait after a transmit message is stored in a mailbox (buffer) (CAN bus arbitration wait).

Bit:	15	14	13	12	11	10	9	8
	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	7	6	5	4	3	2	1	0
	TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit x: TXPRx	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing condition] Message transmission completion and cancellation completion
1	Transmit message transmit wait in corresponding mailbox (CAN bus arbitration)

x = 1 to 15

- Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

16.2.6 Transmit Wait Cancel Register (TXCR)

The transmit wait cancel register (TXCR) is a 16-bit readable/writable register that controls cancellation of transmit wait messages in mailboxes (buffers).

Bit:	15	14	13	12	11	10	9	8
	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	7	6	5	4	3	2	1	0
	TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 9 and 7 to 0—Transmit Wait Cancel Register (TXCR7 to 1, TXCR15 to 8): These bits control cancellation of transmit wait messages in the corresponding HCAN mailboxes.

Bit x: TXCRx	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing condition] Completion of TXPR clearing (when transmit message is canceled normally)
1	TXPR cleared for corresponding mailbox (transmit message cancellation)

- Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

The transmit acknowledge register (TXACKR) is a 16-bit readable/writable register containing status flags that indicate normal completion of mailbox (buffer) message transmission.

Bit:	15	14	13	12	11	10	9	8
	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	7	6	5	4	3	2	1	0
	TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 9 and 7 to 0—Transmit Acknowledge Register (TXACK7 to 1, TXACK15 to 8): These bits indicate that transmission of a message in the corresponding HCAN mailbox has been completed normally.

Bit x: TXACKx	**Description**
0	[Clearing condition] Writing 1 (Initial value)
1	Completion of message transmission for corresponding mailbox

- Bit 8—Reserved: This bit is always read as 0. The write value should always be 0.

The abort acknowledge register (ABACKR) is a 16-bit readable/writable register containing status flags that indicate normal cancellation (aborting) of a mailbox (buffer) transmit messages.

Bit:	15	14	13	12	11	10	9	8
	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	7	6	5	4	3	2	1	0
	ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 9 and 7 to 0—Abort Acknowledge Register (ABACK7 to 1, ABACK15 to 8): These bits indicate that a transmit message in the corresponding mailbox has been canceled (aborted) normally.

Bit x: ABACKx	**Description**
0	[Clearing condition] Writing 1 (Initial value)
1	Completion of transmit message cancellation for corresponding mailbox

- Bit 8—Reserved: This bit is always read as 0. The write value should always be 0.

The receive complete register (RXPR) is a 16-bit readable/writable register containing status flags that indicate normal reception of messages (data frames or remote frames) in mailboxes (buffers). In the case of remote frame reception, the corresponding bit in the remote request register (RFPR) is also set.

Bit:	15	14	13	12	11	10	9	8
	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 0—Receive Complete Register (RXPR7 to 0, RXPR15 to 8): These bits indicate that a receive message has been received normally in the corresponding mailbox.

Bit x: RXPRx	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Completion of message (data frame or remote frame) reception in corresponding mailbox

The remote request register (RFPR) is a 16-bit readable/writable register containing status flags that indicate normal reception of remote frames in mailboxes (buffers). When a bit in this register is set, the corresponding bit in the receive complete register (RXPR) is also set.

Bit:	15	14	13	12	11	10	9	8
	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 0—Remote Request Register (RFPR7 to 0, RFPR15 to 8): These bits indicate that a remote frame has been received normally in the corresponding mailbox.

Bit x: RFPRx	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Completion of remote frame reception in corresponding mailbox

16.2.11 Interrupt Register (IRR)

The interrupt register (IRR) is a 16-bit readable/writable register containing status flags for the various interrupt sources.

Bit:	15	14	13	12	11	10	9	8
	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	IRR12	—	—	IRR9	IRR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W

Bit 15: IRR7	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Overload frame transmission [Setting conditions] Error active/error passive state • When overload frame is transmitted

- Bit 14—Bus Off Interrupt Flag (IRR6): Status flag indicating the bus off state caused by the transmit error counter.

Bit 14: IRR6	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Bus off state caused by transmit error [Setting condition] When TEC ≥ 256

- Bit 13—Error Passive Interrupt Flag (IRR5): Status flag indicating the error passive state caused by the transmit/receive error counter.

Bit 13: IRR5	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error passive state caused by transmit/receive error [Setting condition] When TEC ≥ 128 or REC ≥ 128

- Bit 12—Receive Overload Warning Interrupt Flag (IRR4): Status flag indicating the error warning state caused by the receive error counter.

Bit 12: IRR4	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by receive error [Setting condition] When REC ≥ 96

Bit 11: IRR3	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by transmit error [Setting condition] When TEC ≥ 96

- Bit 10—Remote Frame Request Interrupt Flag (IRR2): Status flag indicating that a remote frame has been received in a mailbox.

Bit 10: IRR2	Description
0	[Clearing condition] Clearing of all bits in RFPR (remote request wait register) (Initial value)
1	Remote frame received and stored in mailbox [Setting conditions] When remote frame reception is completed. When corresponding MBIMR = 0.

- Bit 9—Receive Message Interrupt Flag (IRR1): Status flag indicating that a mailbox receive message has been received normally.

Bit 9: IRR1	Description
0	[Clearing condition] Clearing of all bits in RXPR (receive complete register) when MBIMR is 0 (Initial value)
1	Data frame or remote frame received and stored in mailbox [Setting conditions] When data frame or remote frame reception is completed. When corresponding MBIMR = 0.

after a power-on reset or recovery from software standby mode, interrupt processing will be executed immediately when interrupts are enabled by the interrupt controller.

Bit 8: IRR0	Description
0	[Clearing condition] Writing 1
1	Interrupt request (OVR) due to power-on reset or transition to software standby mode (Initial value) [Setting condition] When reset processing is completed after power-on reset or software standby mode transition

- Bits 7 to 5, 3, and 2—Reserved: These bits always read 0. The write value should always be 0.
- Bit 4—Bus Operation Interrupt Flag (IRR12): Status flag indicating detection of a dominant bit due to bus operation when the HCAN module is in HCAN sleep mode.

Bit 4: IRR12	Description
0	CAN bus idle state (Initial value) [Clearing condition] Writing 1
1	CAN bus operation in HCAN sleep mode [Setting condition] Bus operation (dominant bit detection) in HCAN sleep mode

- Bit 1—Unread Interrupt Flag (IRR9): Status flag indicating that a receive message has been overwritten while still unread.

Bit 1: IRR9	Description
0	[Clearing condition] Clearing of all bits in UMSR (unread message status register) (Initial value)
1	Unread message overwrite [Setting condition] When UMSR (unread message status register) is set

Bit 0: IRR8	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Transmit message has been transmitted or aborted, and new message can be stored [Setting condition] When TXPR (transmit wait register) is cleared by completion of transmission or completion of transmission abort

16.2.12 Mailbox Interrupt Mask Register (MBIMR)

The mailbox interrupt mask register (MBIMR) is a 16-bit readable/writable register containing flags that enable or disable individual mailbox (buffer) interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 0—Mailbox Interrupt Mask (MBIMR7 to 0, MBIMR15 to 8): Flags that enable or disable individual mailbox interrupt requests.

Bit x: MBIMRx	Description
0	[Transmitting] Interrupt request to CPU due to TXPR clearing [Receiving] Interrupt request to CPU due to RXPR setting
1	Interrupt requests to CPU disabled (Initial value)

The interrupt mask register (IMR) is a 16-bit readable/writable register containing flags that enable or disable requests by individual interrupt sources.

Bit:	15	14	13	12	11	10	9	8
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	—
Initial value:	1	1	1	1	1	1	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

Bit:	7	6	5	4	3	2	1	0
	—	—	—	IMR12	—	—	IMR9	IMR8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R/W	R	R	R/W	R/W

- Bit 15—Overload Frame Interrupt Mask (IMR7): Enables or disables overload frame interrupt requests.

Bit 15: IMR7	Description
0	Overload frame interrupt request (OVR) to CPU by IRR7 enabled
1	Overload frame interrupt request (OVR) to CPU by IRR7 disabled (Initial value)

- Bit 14—Bus Off Interrupt Mask (IMR6): Enables or disables bus off interrupt requests caused by the transmit error counter.

Bit 14: IMR6	Description
0	Bus off interrupt request (ERS) to CPU by IRR6 enabled
1	Bus off interrupt request (ERS) to CPU by IRR6 disabled (Initial value)

- Bit 13—Error Passive Interrupt Mask (IMR5): Enables or disables error passive interrupt requests caused by the transmit/receive error counter.

Bit 13: IMR5	Description
0	Error passive interrupt request (ERS) to CPU by IRR5 enabled
1	Error passive interrupt request (ERS) to CPU by IRR5 disabled (Initial value)

Bit 12: IMR4	Description
0	REC error warning interrupt request (OVR) to CPU by IRR4 enabled
1	REC error warning interrupt request (OVR) to CPU by IRR4 disabled (Initial value)

- Bit 11—Transmit Overload Warning Interrupt Mask (IMR3): Enables or disables error warning interrupt requests caused by the transmit error counter.

Bit 11: IMR3	Description
0	TEC error warning interrupt request (OVR) to CPU by IRR3 enabled
1	TEC error warning interrupt request (OVR) to CPU by IRR3 disabled (Initial value)

- Bit 10—Remote Frame Request Interrupt Mask (IMR2): Enables or disables remote frame reception interrupt requests.

Bit 10: IMR2	Description
0	Remote frame reception interrupt request (OVR) to CPU by IRR2 enabled
1	Remote frame reception interrupt request (OVR) to CPU by IRR2 disabled (Initial value)

- Bit 9—Receive Message Interrupt Mask (IMR1): Enables or disables message reception interrupt requests.

Bit 9: IMR1	Description
0	Message reception interrupt request (RM) to CPU by IRR1 enabled
1	Message reception interrupt request (RM) to CPU by IRR1 disabled (Initial value)

- Bit 8—Reserved: This bit is always read as 0. The write value should always be 0.
- Bits 7 to 5, 3, and 2—Reserved: These bits are always read as 1. The write value should always be 1.

Bit 4: IMR12	Description
0	Bus operation interrupt request (OVR) to CPU by IRR12 enabled
1	Bus operation interrupt request (OVR) to CPU by IRR12 disabled (Initial value)

- Bit 1—Unread Interrupt Mask (IMR9): Enables or disables unread receive message overwrite interrupt requests.

Bit 1: IMR9	Description
0	Unread message overwrite interrupt request (OVR) to CPU by IRR9 enabled
1	Unread message overwrite interrupt request (OVR) to CPU by IRR9 disabled (Initial value)

- Bit 0—Mailbox Empty Interrupt Mask (IMR8): Enables or disables mailbox empty interrupt requests.

Bit 0: IMR8	Description
0	Mailbox empty interrupt request (SLE) to CPU by IRR8 enabled
1	Mailbox empty interrupt request (SLE) to CPU by IRR8 disabled (Initial value)

16.2.14 Receive Error Counter (REC)

The receive error counter (REC) is an 8-bit read-only register that functions as a counter indicating the number of receive message errors on the CAN bus. The count value is stipulated in the CAN protocol. This register cannot be modified.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

The transmit error counter (TEC) is an 8-bit read-only register that functions as a counter indicating the number of transmit message errors on the CAN bus. The count value is stipulated in the CAN protocol. This register cannot be modified.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

16.2.16 Unread Message Status Register (UMSR)

The unread message status register (UMSR) is a 16-bit readable/writable register containing status flags that indicate, for individual mailboxes (buffers), that a received message has been overwritten by a new receive message before being read.

Bit:	15	14	13	12	11	10	9	8
	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 to 0—Unread Message Status Flags (UMSR7 to 0, UMSR15 to 8): Status flags indicating that an unread receive message has been overwritten. When an unread receive message is overwritten by a new receive message, the old data is lost.

Bit x: UMSRx	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Unread receive message is overwritten by a new message [Setting condition] When a new message is received before RXPR is cleared

x = 0 to 15

The local acceptance filter masks (LAFML, LAFMH) are 16-bit readable/writable registers that filter receive messages to be stored in the receive-only mailbox (MC0, MD0) according to the identifier. In these registers, consist of LAFMH15: MSB to LAFMH5: LSB are 11 standard/extended identifier bits, and LAFMH1: MSB to LAFML0: LSB are 18 extended identifier bits.

LAFML

Bit:	15	14	13	12	11	10	9	8
	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LAFMH

Bit:	15	14	13	12	11	10	9	8
	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1	LAFMH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9	LAFMH8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

extended identifiers).

Bit x: LAFMHx	Description
0	Stored in MC0, MD0 (receive-only mailbox) depending on bit match between MC0 message identifier and receive message identifier (Initial value)
1	Stored in MC0, MD0 (receive-only mailbox) regardless of bit match between MC0 message identifier and receive message identifier

- LAFMH Bits 12 to 10—Reserved: These bits are always read as 0. The write value should always be 0.
- LAFMH Bits 9 and 8, LAFML bits 15 to 0–18-Bit Identifier Filter (LAFMH1, 0, LAFML7 to 0, LAFML15 to 8): Filter mask bits for the 18 bits of the receive message identifier (extended).

Bit x: LAFMHx LAFMLx	Description
0	Stored in MC0 (receive-only mailbox) depending on bit match between MC0 message identifier and receive message identifier (Initial value)
1	Stored in MC0 (receive-only mailbox) regardless of bit match between MC0 message identifier and receive message identifier

The message control register sets (MC0 to MC15) consist of eight 8-bit readable/writable registers (MCx[1] to MCx[8]). The HCAN has 16 sets of these registers (MC0 to MC15).

The initial value of these registers is undefined, so they must be initialized (by writing 0 or 1).

MCx [1]

Bit:	7	6	5	4	3	2	1	0
					DLC3	DLC2	DLC1	DLC0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [2]

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [3]

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [4]

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [5]

Bit:	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [7]

Bit:	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [8]

Bit:	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- MCx[1] Bits 7 to 4—Reserved: The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).
- MCx[1] Bits 3 to 0: Data Length Code (DLC3 to 0): These bits indicate the required length of data frames and remote frames.

Bit 3: DLC3	Bit 2: DLC2	Bit 1: DLC1	Bit 0: DLC0	Description
0	0	0	0	Data length = 0 bytes
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	*	*	*	Data length = 8 bytes

*: Don't care

- MCx[2] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).

- MCx[4] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).
- MCx[6] Bits 7 to 0: Standard Identifier (STD_ID10 to STD_ID3)
  MCx[5] Bits 7 to 5: Standard Identifier (STD_ID2 to STD_ID0)
  These bits set the identifier (standard identifier) of data frames and remote frames.

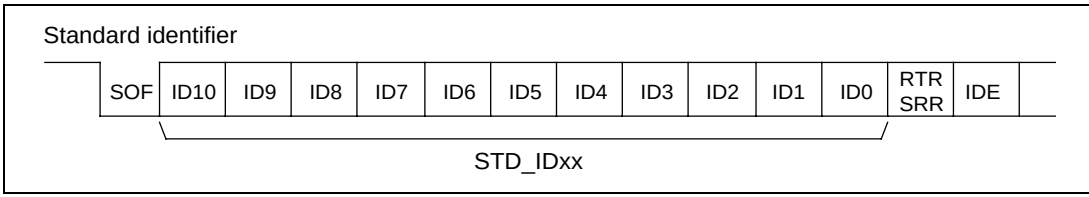


Figure 16.3 Standard Identifier

- MCx[5] Bit 4: Remote Transmission Request (RTR): Used to distinguish between data frames and remote frames.

Bit 4: RTR	Description
0	Data frame
1	Remote frame

- MCx[5] Bit 3: Identifier Extension (IDE): Used to distinguish between the standard format and extended format of data frames and remote frames.

Bit 3: IDE	Description
0	Standard format
1	Extended format

- MCx[5] Bit 2—Reserved: The initial value of this bit is undefined; it must be initialized (by writing 0 or 1).
- MCx[5] Bits 1 and 0: Extended Identifier (EXD_ID17, EXD_ID16)
  MCx[8] Bits 7 to 0: Extended Identifier (EXD_ID15 to EXD_ID8)
  MCx[7] Bits 7 to 0: Extended Identifier (EXD_ID7 to EXD_ID0)
  These bits set the identifier (extended identifier) of data frames and remote frames.

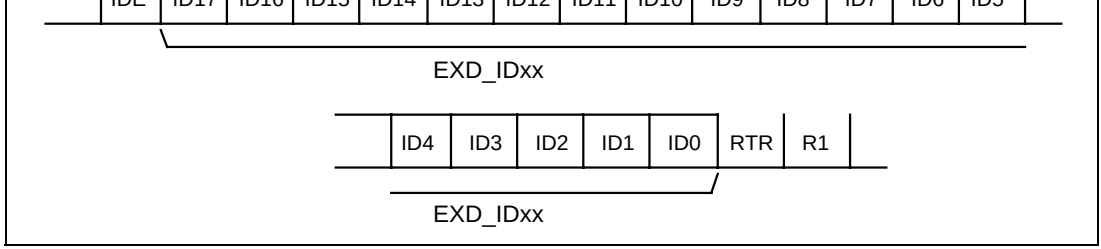


Figure 16.4 Extended Identifier

16.2.19 Message Data (MD0 to MD15)

The message data register sets (MD0 to MD15) consist of eight 8-bit readable/writable registers (MDx[1] to MDx[8]). The HCAN has 16 sets of these registers (MD0 to MD15).

The initial value of these registers is undefined, so they must be initialized (by writing 0 or 1).

MDx[1] Message Data 1

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[2] Message Data 2

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[3] Message Data 3

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[5] Message Data 5

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[6] Message Data 6

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[7] Message Data 7

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[8] Message Data 8

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The SH7055SF has an on-chip HCAN module with two channels, each of which can be controlled independently. Except for pin states, both channels have identical specifications, and so control should be carried out in the same way for both.

16.3.1 Hardware Reset and Software Reset

There are two ways of resetting the HCAN: Hardware reset and software reset.

Hardware Reset (Power-on Reset or Hardware/Software Standby): The MCR reset request bit (MCR0) in MCR and the reset state bit (GSR3) in GSR within the HCAN are automatically set and initialized (hardware reset). At the same time, all internal registers are initialized. However mailbox (RAM) contents are not initialized. A flowchart of this reset is shown in figure 16.5.

Software Reset (Write to MCR0): In normal operation HCAN is initialized by setting the MCR reset request bit (MCR0) in MCR (software reset). With this kind of reset, if the CAN controller is performing a communication operation (transmission or reception), the initialization state is not entered until the message has been completed. During initialization, the reset state bit (GSR3) in GSR is set. In this kind of initialization, the error counters (TEC and REC) are initialized but other registers and RAM are not. A flowchart of this reset is shown in figure 16.6.

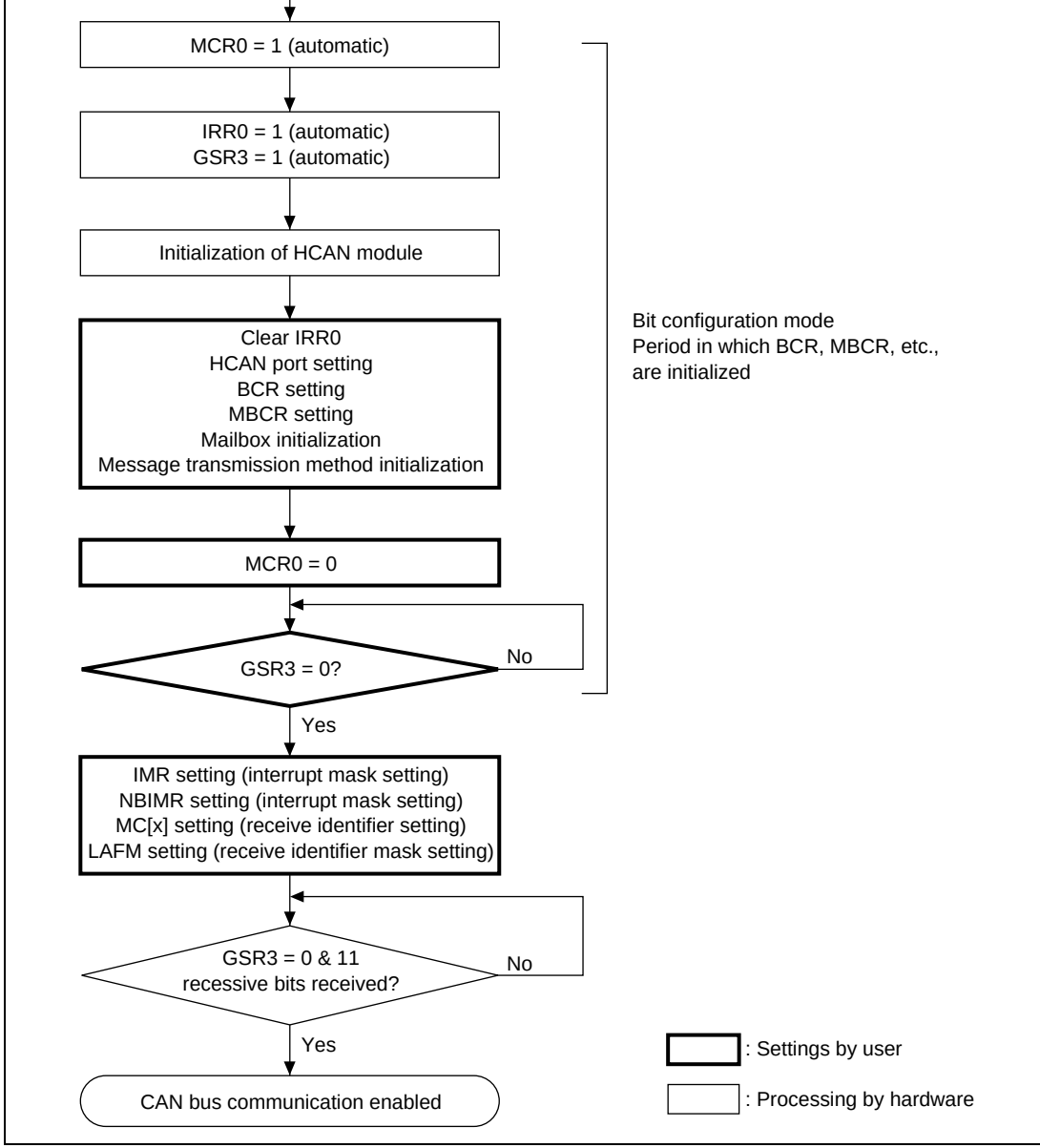


Figure 16.5 Hardware Reset Flowchart

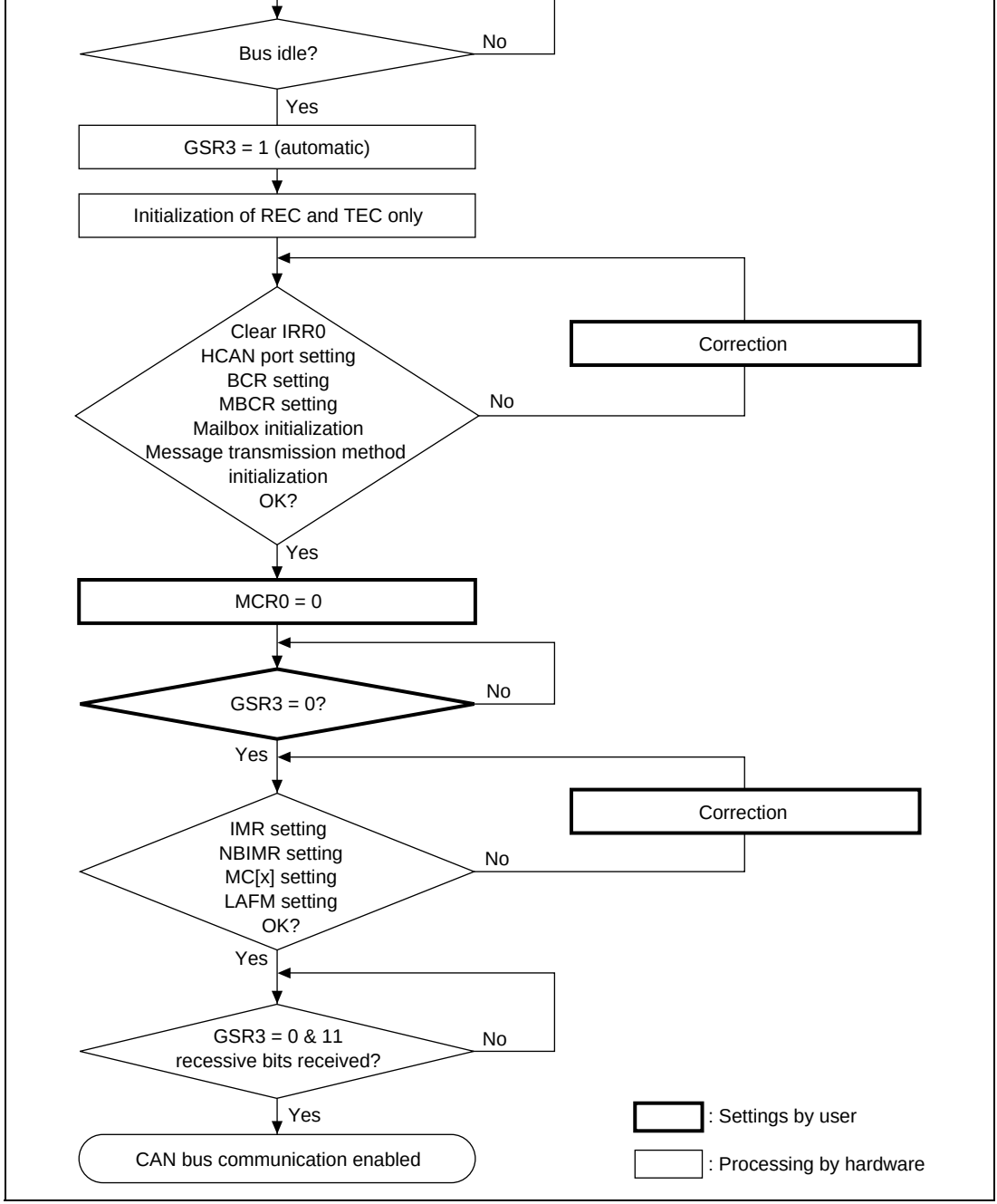


Figure 16.6 Software Reset Flowchart

After a hardware reset, the following initialization processing should be carried out.

1. Clearing of IRR0 bit in interrupt register (IRR)
2. HCAN pin port settings
3. Bit rate setting
4. Mailbox transmit/receive settings
5. Mailbox (buffer) initialization
6. Message transmission method setting

These initial settings must be made while the HCAN is in bit configuration mode. Configuration mode is a state in which the reset request bit (MCR0) in the master control register (MCR) is 1 and the reset status bit in the general status register (GSR) is also 1 (GSR3 = 1). Configuration mode is exited by clearing the reset request bit in MCR to 0; when MCR0 is cleared to 0, the HCAN automatically clears the reset state bit (GSR3) in the general status register (GSR). The power-up sequence then begins, and communication with the CAN bus is possible as soon as the sequence ends. The power-up sequence consists of the detection of 11 consecutive recessive bits.

IRR0 Clearing: The reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software standby mode. As an HCAN interrupt is initiated immediately when interrupts are enabled, IRR0 should be cleared.

HCAN Pin Port Settings: HCAN pin port settings must be made during or before bit configuration. Refer to the section 20, Pin Function Controller (PFC), for details of the setting method.

The SH7055SF has two on-chip HCAN channels, which can be used in either of the following ways:

1. Two-channel 16-buffer HCAN
2. One-channel 32-buffer HCAN

An example of 2-channel/16-buffer independent use is shown in figure 16.7, and an example of 2-channel/32-buffer use in figure 16.8.

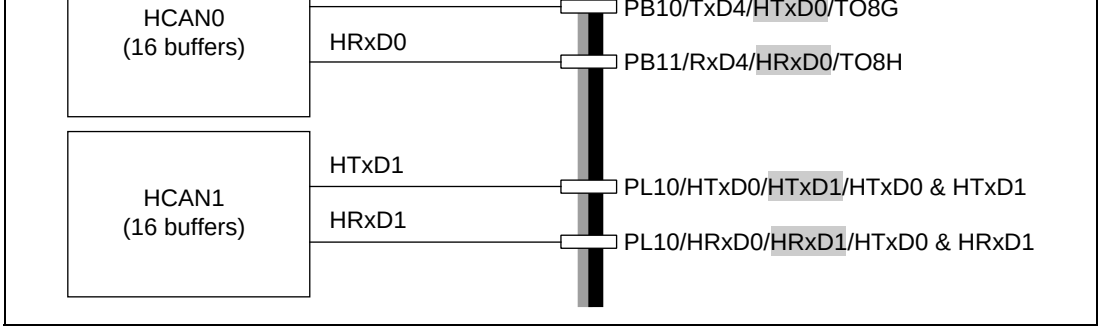


Figure 16.7 Example of 2-Channel/16-Buffer Independent Use

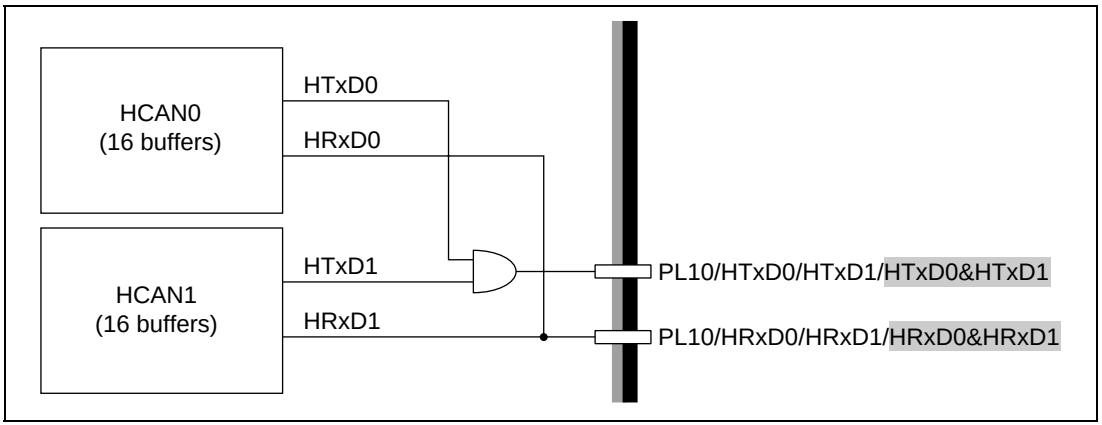


Figure 16.8 Example of 1-Channel/32-Buffer Use

the bit configuration register (BCR).

Notes: 1. BCR can be written to at all times, but should only be modified in configuration mode.
2. Settings should be made so that all CAN controllers connected to the CAN bus have the same baud rate and bit width.
3. Limits for the settable variables (TSEG1, TSEG2, BRP, sample point, and SJW) are shown in table 16.5.

Table 16.5 BCR Setting Limits

Name	Abbreviation	Min. Value	Max. Value	Unit
Time segment 1	TSEG1	4	16	TQ
Time segment 2	TSEG2	2	8	TQ
Baud rate prescaler	BRP	2	128	System clock
Sample point	SAM	1	3	Point
Re-synchronization jump width	SJW	1	4	TQ

Settable Variable Limits

- The bit width consists of the total of the settable time quanta (TQ). TQ (number of system clocks) is determined by the baud rate prescaler (BRP).

  $TQ = (2 \times (BRP + 1))/(f_{CLK/2})$

  $f_{CLK} = P\phi$
- The minimum value of SJW is stipulated in the CAN specifications.

  $4 \geq SJW \geq 1$
- The minimum value of TSEG1 is stipulated in the CAN specifications.

  $TSEG1 \geq TSEG2$
- The minimum value of TSEG2 is stipulated in the CAN specifications.

  $TSEG2 \geq (1 + SJW)$

The following formula is used to calculate the baud rate.

$$\text{Bit rate [b/s]} = \frac{f_{CLK}}{2 \times (BRP + 1) \times (3 + TSEG1 + TSEG2)}$$

Note: $f_{CLK} = P\phi$ (peripheral clock: $\phi/2$)
The BCR values are used for BRP, TSEG1, and TSEG2.

Example: With a 1 Mb/s baud rate and a 40 MHz input clock:

	Set Values	Actual Values
f_{CLK}	40 MHz/2	—
BRP	0 (B'000000)	System clock × 2
TSEG1	4 (B'0100)	5TQ
TSEG2	3 (B'011)	4TQ

Mailbox Transmit/Receive Settings: HCAN0 and HCAN1 each have 16 mailboxes. Mailbox 0 is receive-only, while mailboxes 1 to 15 can be set for transmission or reception. Mailboxes that can be set for transmission or reception must be designated either for transmission use or for reception use before communication begins. The Initial status of mailboxes 1 to 15 is for transmission (while mailbox 0 is for reception only). Mailbox transmit/receive settings are not initialized by a software reset.

- Setting for transmission

  Transmit mailbox setting (mailboxes 1 to 15)

  Clearing a bit to 0 in the mailbox configuration register (MBCR) designates the corresponding mailbox for transmission use. After a reset, mailboxes are initialized for transmission use, so this setting is not necessary.
- Setting for reception

  Transmit/receive mailbox setting (mailboxes 1 to 15)

  Setting a bit to 1 in the mailbox configuration register (MBCR) designates the corresponding mailbox for reception use. When setting mailboxes for reception, to improve message transmission efficiency, high-priority messages should be set in low-to-high mailbox order (priority order: mailbox 1 (MCx[1]) > mailbox 15 (MCx[15])).
- Receive-only mailbox (mailbox 0)

  No setting is necessary, as this mailbox is always used for reception.

Mailbox (Message Control/Data (MCx[x], MDx[x]) Initial Settings: After power is supplied, all registers and RAM (message control/data, control registers, status registers, etc.) are initialized. Message control/data (MCx[x], MDx[x]) only are in RAM, and so their values are undefined. Initial values must therefore be set in all the mailboxes (by writing 0s or 1s).

Setting the Message Transmission Method: Either of the following message transmission methods can be selected with the message transmission method bit (MCR2) in the master control register (MCR):

1. Transmission order determined by message identifier priority
2. Transmission order determined by mailbox number priority

messages in the transmit buffer, and message transmission is performed when the bus is acquired.

When the mailbox number priority method is selected, if a number of messages are designated as waiting for transmission (TXPR = 1), the message with the highest priority set in the message identifier (MCx[5]–MCx[8]) is stored in the transmit buffer. CAN bus arbitration is then carried out for the message in the transmit buffer, and message transmission is performed when the transmission right is acquired. When the TXPR bit is set, internal arbitration is performed again, and the highest-priority message is found and stored in the transmit buffer.

16.3.3 Transmit Mode

Message transmission is performed using mailboxes 1 to 15. The transmission procedure is described below, and a transmission flowchart is shown in figure 16.9.

1. Initialization (after hardware reset only)
   a. Clearing of IRR0 bit in interrupt register (IRR)
   b. HCAN pin port settings
   c. Bit rate settings
   d. Mailbox transmit/receive settings
   e. Mailbox initialization
   f. Message transmission method setting
2. Interrupt and transmit data settings
   a. Interrupt setting
   b. Arbitration field setting
   c. Control field setting
   d. Data field setting
3. Message transmission and interrupts
   a. Message transmission wait
   b. Message transmission completion and interrupt
   c. Message transmission abort
   d. Message retransmission

1. IRR0 clearing

   The reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software standby mode. As an HCAN interrupt is initiated immediately when interrupts are enabled, IRR0 should be cleared.

2. HCAN pin port settings

   To prevent erroneous identification of CAN bus data, HCAN pin port settings should be made first. See HCAN Pin Port Settings in section 16.3.2, Initialization after a Hardware Reset, and section 20, Pin Function Controller, for details.

3. Bit rate settings

   Set values relating to the CAN bus communication speed and re-synchronization. See Bit Rate Settings in section 16.3.2, Initialization after a Hardware Reset, for details.

4. Mailbox transmit/receive settings

   Mailbox transmit/receive settings should be made in advance. A total of 30 mailbox can be set for transmission or reception (mailboxes 1 to 15 in HCAN0 and HCAN1). To set a mailbox for transmission, clear the corresponding bit to 0 in the mailbox configuration register (MBCR). See Mailbox Transmit/Receive Settings in section 16.3.2, Initialization after a Hardware Reset, for details.

5. Mailbox initialization

   As message control/data registers (MCx[x], MDx[x]) are configured in RAM, their initial values after powering on are undefined, and so bit initialization is necessary. Write 0s or 1s to the mailboxes. See Mailbox (Message Control/Data (MCx[x], MDx[x]) Initial Settings in section 16.3.2, Initialization after a Hardware Reset, for details.

6. Message transmission method setting

   Set the transmission method for mailboxes designated for transmission. The following two transmission methods can be used. See Setting the Message Transmission Method in section 16.3.2, Initialization after a Hardware Reset, for details.

   a. Transmission order determined by message identifier priority
   b. Transmission order determined by mailbox number priority

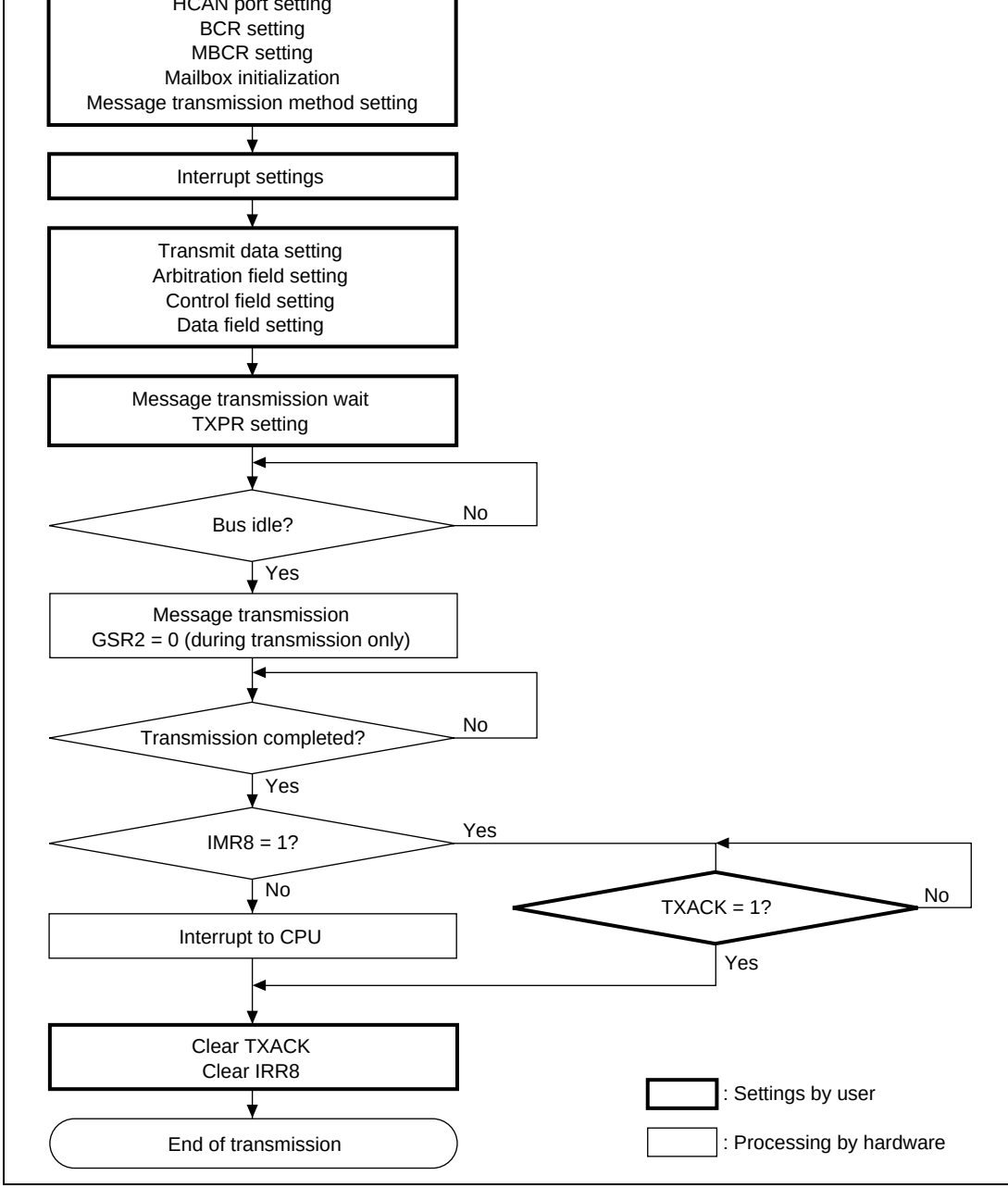


Figure 16.9 Transmission Flowchart

interrupt mask register (MBIMR) and interrupt mask register (IMR), while transmit data settings are made by writing the necessary data in 2, 3, and 4 below to the message control registers (MCx[1] to MCx[8]) and message data registers (MDx[1] to MDx[8]).

1. CPU interrupt source settings

   Transmission acknowledge and transmission abort acknowledge interrupts can be masked for individual mailboxes in the mailbox interrupt mask register (MBIMR). Interrupt register (IRR) interrupts can be masked in the interrupt mask register (IMR).
2. Arbitration field

   In the arbitration field, the 11-bit identifier (STD_ID0 to STD_ID10) and RTR bit (standard format) or 29-bit identifier (STD_ID0 to STD_ID10, EXT_ID0 to EXT_ID17) and IDE.RTR bit (extended format) are set. The registers to be set are MCx[5] to MCx[8].
3. Control field

   In the control field, the byte length of the data to be transmitted is set in DLC0 to DLC3. The register to be set is MCx[1].
4. Data field

   In the data field, the data to be transmitted is set in byte units in the range of 0 to 8 bytes. The registers to be set are MDx[1] to MDx[8].

The number of bytes in the data actually transmitted depends on the data length code (DLC) in the control field. If a value exceeding the value set in DLC is set in the data field, only the number of bytes set in DLC will actually be transmitted.

Message Transmission and Interrupts

1. Message transmission wait

   If message transmission is to be performed after completion of the message control (MCx[1] to MCx[8]) and message data (MDx[1] to MDx[8]).settings, transmission is started by setting the corresponding mailbox transmit wait bit (TXPR1 to TXPR15) to 1 in the transmit wait register (TXPR). The following two transmission methods can be used:

   a. Transmission order determined by message identifier priority
   b. Transmission order determined by mailbox number priority

   When the message identifier priority method is selected, if a number of messages are designated as waiting for transmission (TXPR = 1), the message with the highest priority set in the message identifier (MCx[5] to MCx[8]) is stored in the transmit buffer. CAN bus arbitration is then carried out for the message in the transmit buffer, and message transmission is performed when the transmission right is acquired. When the TXPR bit is set, internal arbitration is performed again, the highest-priority message is found and stored in the transmit buffer, CAN bus arbitration is carried out in the same way, and message transmission is performed when the transmission right is acquired.

high mailbox order (priority order: mailbox 1 > mailbox 15). CAN bus arbitration is then carried out for the messages in the transmit buffer, and message transmission is performed when the bus is acquired.

2. Message transmission completion and interrupt

   When a message is transmitted normally using the above procedure, the corresponding acknowledge bit (TXACK1 to TXACK15) in the transmit acknowledge register (TXACK) and transmit wait bit (TXPR1 to TXPR15) in the transmit wait register (TXPR) are automatically initialized. If the corresponding bit (MBIMR1 to MBIMR15) in the mailbox interrupt mask register (MBIMR) and the mailbox empty interrupt bit (IRR8) in the interrupt mask register (IMR) are set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

3. Message transmission cancellation

   Transmission cancellation can be specified for a message stored in a mailbox as a transmit wait message. A transmit wait message is canceled by setting the bit for the corresponding mailbox (TXCR1 to TXCR15) to 1 in the transmit cancel register (TXCR). When cancellation is executed, the transmit wait register (TXPR) is automatically reset, and the corresponding bit is set to 1 in the abort acknowledge register (ABACK). An interrupt to the CPU can be requested. If the corresponding bit (MBIMR1 to MBIMR15) in the mailbox interrupt mask register (MBIMR) and the mailbox empty interrupt bit (IRR8) in the interrupt mask register (IMR) are set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

   However, a transmit wait message cannot be canceled at the following times:

   a. During internal arbitration or CAN bus arbitration
   b. During data frame or remote frame transmission

   Also, transmission cannot be canceled by clearing the transmit wait register (TXPR). Figure 16.10 shows a flowchart of transmit message cancellation.

4. Message retransmission

   If transmission of a transmit message is aborted in the following cases, the message is retransmitted automatically:

   a. CAN bus arbitration failure (failure to acquire the bus)
   b. Error during transmission (bit error, stuff error, CRC error, frame error, ACK error)

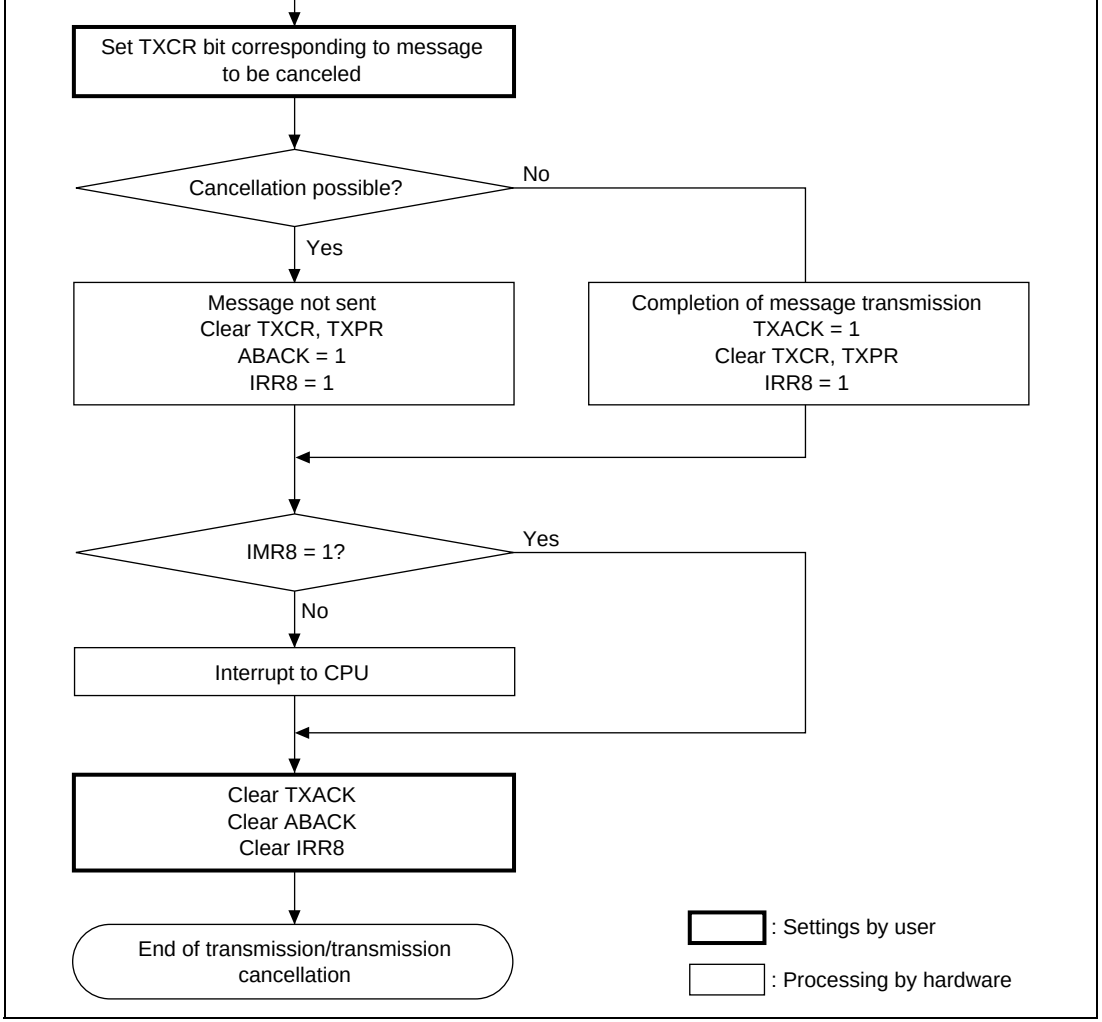


Figure 16.10 Transmit Message Cancellation Flowchart

Message reception is performed using mailboxes 0 and 1 to 15. The reception procedure is described below, and a reception flowchart is shown in figure 16.11.

1. Initialization (after hardware reset only)
   a. Clearing of IRR0 bit in interrupt register (IRR)
   b. HCAN pin port settings
   c. Bit rate settings
   d. Mailbox transmit/receive settings
   e. Mailbox initialization
2. Interrupt and receive message settings
   a. Interrupt setting
   b. Arbitration field setting
   c. Local acceptance filter mask (LAFM) settings
3. Message reception and interrupts
   a. Message reception CRC check
   b. Data frame reception
   c. Remote frame reception
   d. Unread message reception

Initialization (after Hardware Reset Only): These settings should be made while the HCAN is in bit configuration mode.

1. IRR0 clearing

   The reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software standby mode. As an HCAN interrupt is initiated immediately when interrupts are enabled, IRR0 should be cleared.
2. HCAN pin port settings

   To prevent erroneous identification of CAN bus data, HCAN pin port settings should be made first. See HCAN Pin Port Settings in section 16.3.2, Initialization after a Hardware Reset, and section 20, Pin Function Controller (PFC), for details.
3. Bit rate settings

   Set values relating to the CAN bus communication speed and re-synchronization. See Bit Rate Settings in section 16.3.2, Initialization after a Hardware Reset, for details.
4. Mailbox transmit/receive settings

   Each channel has one receive-only mailbox (mailbox 0) and 15 mailboxes that can be set for reception. Thus a total of 32 mailboxes can be used for reception. To set a mailbox for reception, set the corresponding bit to 1 in the mailbox configuration register (MBCR). The

5. Mailbox (RAM) initialization

   As message control/data registers (MCx[x], MDx[x]) are configured in RAM, their initial values after powering on are undefined, and so bit initialization is necessary. Write 0s or 1s to the mailboxes. See Mailbox (Message Control/Data (MCx[x], MDx[x]) Initial Settings in section 16.3.2, Initialization after a Hardware Reset, for details.

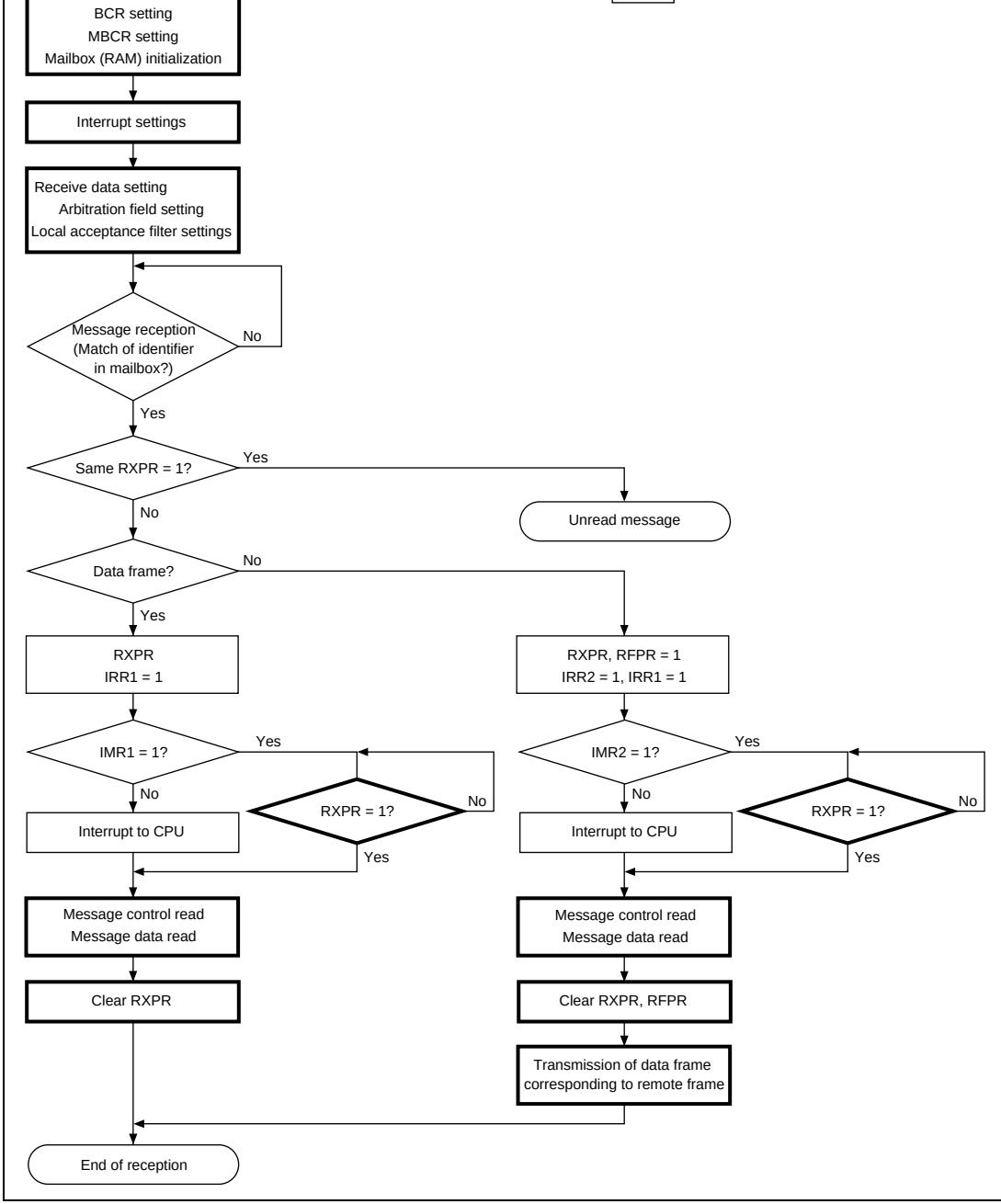


Figure 16.11 Reception Flowchart

mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR). To receive a message, the identifier must be set in advance in the message control (MCx[1] to MCx[8]) for the receiving mailbox. When a message is received, all the bits in the receive message identifier are compared, and if a 100% match is found, the message is stored in the matching mailbox. Mailbox 0 (MC0[x], MD0[x]) has a local acceptance filter mask (LAFM) that allows Don't Care settings to be made.

1. CPU interrupt source settings

   When transmitting, transmission acknowledge and transmission abort acknowledge interrupts can be masked for individual mailboxes in the mailbox interrupt mask register (MBIMR). When receiving, data frame and remote frame receive wait interrupts can be masked. Interrupt register (IRR) interrupts can be masked in the interrupt mask register (IMR).

2. Arbitration field setting

   In the arbitration field, the identifier (STD_ID0 to STD_ID10, EXT_ID0 to EXT_ID17) of the message to be received is set. If all the bits in the set identifier do not match, the message is not stored in a mailbox.

   Example: Mailbox 1 010_1010_1010 (standard identifier)
   Only one kind of message identifier can be received by MB1
   Identifier 1: 010_1010_1010

3. Local acceptance filter mask (LAFM) setting

   The local acceptance filter mask is provided for mailbox 0 (MC0[x], MD0[x]) only, enabling a Don't Care specification to be made for all bits in the received identifier. This allows various kinds of messages to be received.

   Example: Mailbox 0 010_1010_1010 (standard identifier)
   LAFM 000_0000_0011 (0: Care, 1: Don't Care)
   A total of four kinds of message identifiers can be received by MB0
   Identifier 1: 010_1010_1000
   Identifier 2: 010_1010_1001
   Identifier 3: 010_1010_1010
   Identifier 4: 010_1010_1011

When a message is received, a CRC check is performed automatically (by hardware). If the result of the CRC check is normal, ACK is transmitted in the ACK field irrespective of whether or not the message can be received.

2. Data frame reception

   If the received message is confirmed to be error-free by the CRC check, etc., the identifier in the mailbox (and also LAFM in the case of mailbox 0 only) and the identifier of the receive message are compared, and if a complete match is found, the message is stored in the mailbox. The message identifier comparison is carried out on each mailbox in turn, starting with mailbox 0 and ending with mailbox 15. If a complete match is found, the comparison ends at that point, the message is stored in the matching mailbox, and the corresponding receive complete bit (RXPR0 to RXPR15) is set in the receive complete register (RXPR). However, if the identifier matches when a comparison with the mailbox 0 LAFM is carried out, the mailbox comparison sequence does not end at that point, but continues with mailbox 1 and then the remaining mailboxes. It is therefore possible for a message matching mailbox 0 to be received by another mailbox (however, the same message cannot be stored in more than one of mailboxes 1 to 15). If the corresponding bit (MBIMR0 to MBIMR15) in the mailbox interrupt mask register (MBIMR) and the receive message interrupt mask (IMR1) in the interrupt mask register (IMR) are set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

3. Remote frame reception

   Two kinds of messages—data frames and remote frames—can be stored in mailboxes. A remote frame differs from a data frame in that the remote reception request bit (RTR) in the message control register (MC[x]5) and the data field are 0 bytes. The data length to be returned in a data frame must be stored in the data length code (DLC) in the control field.

   When a remote frame (RTR = recessive) is received, the corresponding bit is set in the remote request wait register (RFPR). If the corresponding bit (MBIMR0 to MBIMR15) in the mailbox interrupt mask register (MBIMR) and the remote frame request interrupt mask (IRR2) in the interrupt mask register (IMR) are set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

4. Unread message reception

   When a received message matches the identifier in a mailbox, the message is stored in the mailbox. If a message overwrite occurs before the CPU reads the message, the corresponding bit (UMSR0 to UMSR15) is set in the unread message register (UMSR). In overwriting of an unread message, when a new message is received before the corresponding bit in the receive complete register (RXPR) has been cleared, the unread message register (UMSR) is set. If the unread interrupt flag (IRR9) in the interrupt mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be sent to the CPU. Figure 16.12 shows a flowchart of unread message overwriting.

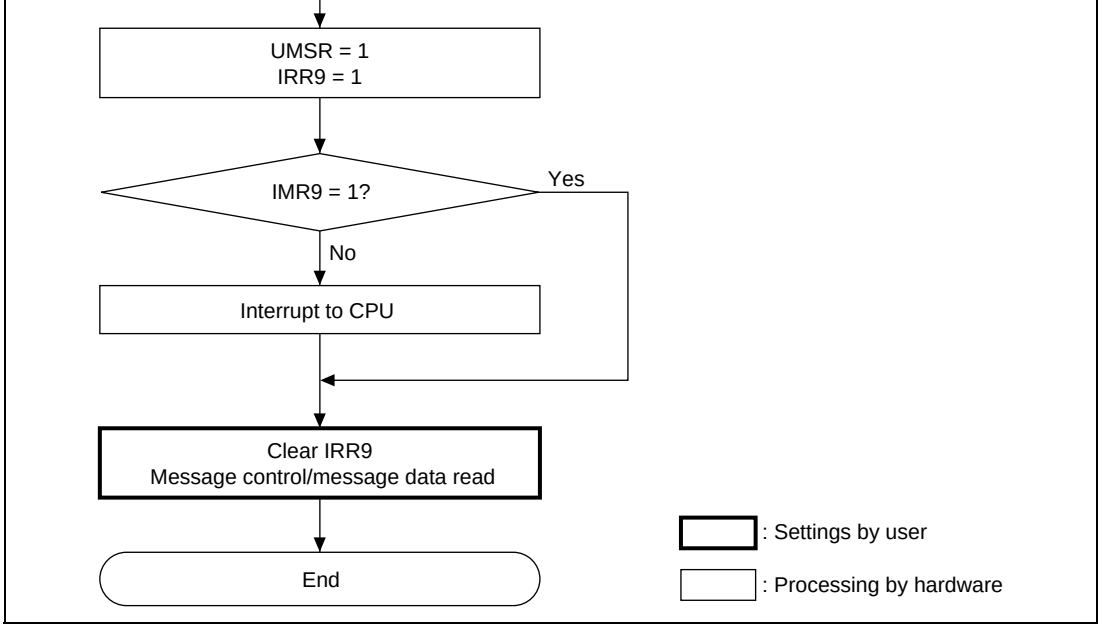


Figure 16.12 Unread Message Overwrite Flowchart

The HCAN is provided with an HCAN sleep mode that places the HCAN module in the sleep state to reduce current dissipation. Figure 16.13 shows a flowchart of the HCAN sleep mode.

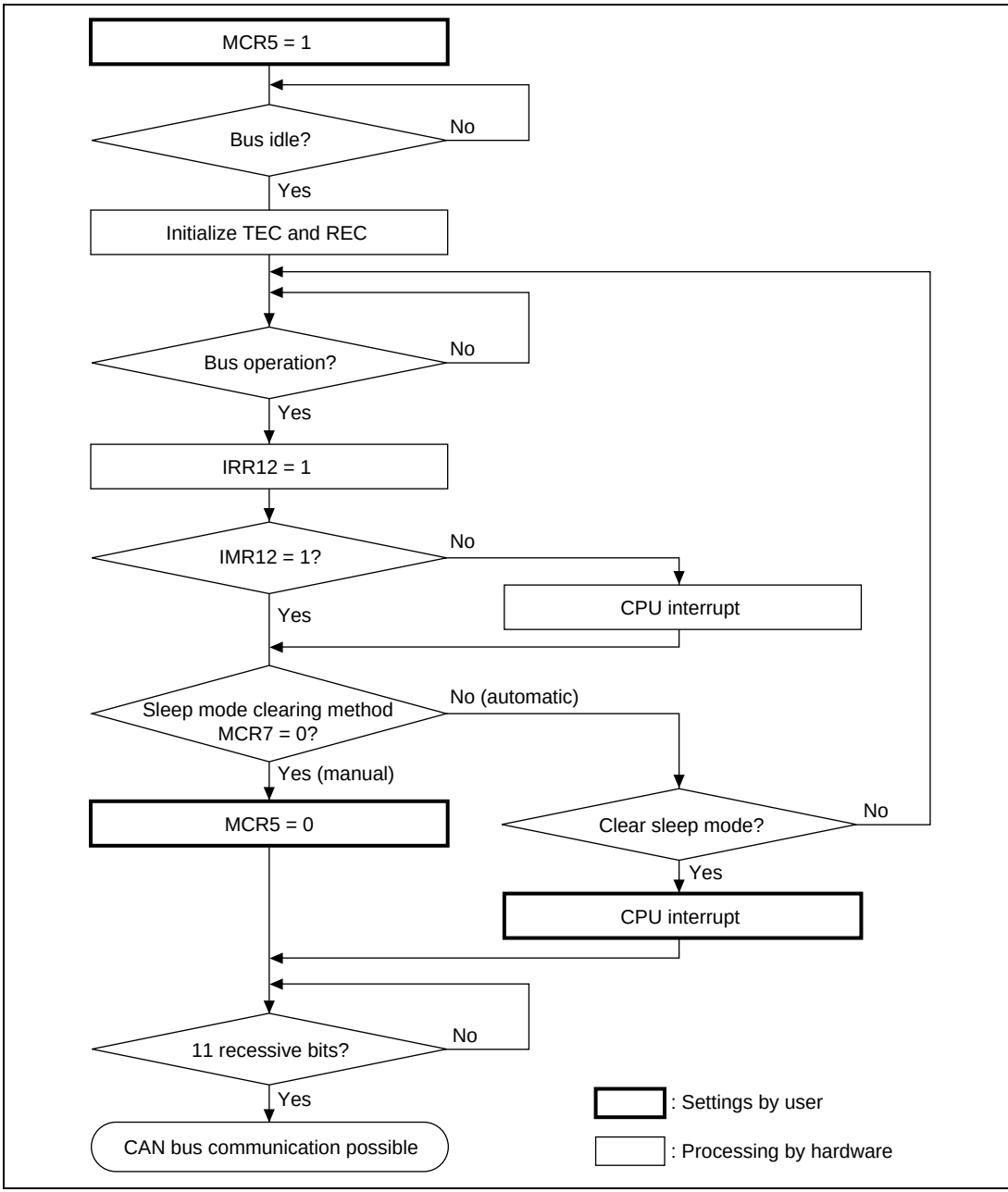


Figure 16.13 HCAN Sleep Mode Flowchart

delayed until the bus becomes idle.

Either of the following methods of clearing HCAN sleep mode can be selected by making a setting in the MCR7 bit.

1. Clearing by software
2. Clearing by CAN bus operation

Eleven recessive bits must be received after HCAN sleep mode is cleared before CAN bus communication is enabled again.

Clearing by Software: Clearing by software is performed by having the CPU write 0 to MCR5.

Clearing by CAN Bus Operation: Clearing by CAN bus operation occurs automatically when the CAN bus performs an operation and this change is detected. In this case, the first message is not received in the message box; normal reception starts with the second message. When a change is detected on the CAN bus in HCAN sleep mode, the bus operation interrupt flag (IRR12) is set in the interrupt register (IRR). If the bus interrupt mask (IMR12) in the interrupt mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

The HCAN halt mode is provided to enable mailbox settings to be changed without performing an HCAN hardware or software reset. Figure 16.14 shows a flowchart of the HCAN halt mode.

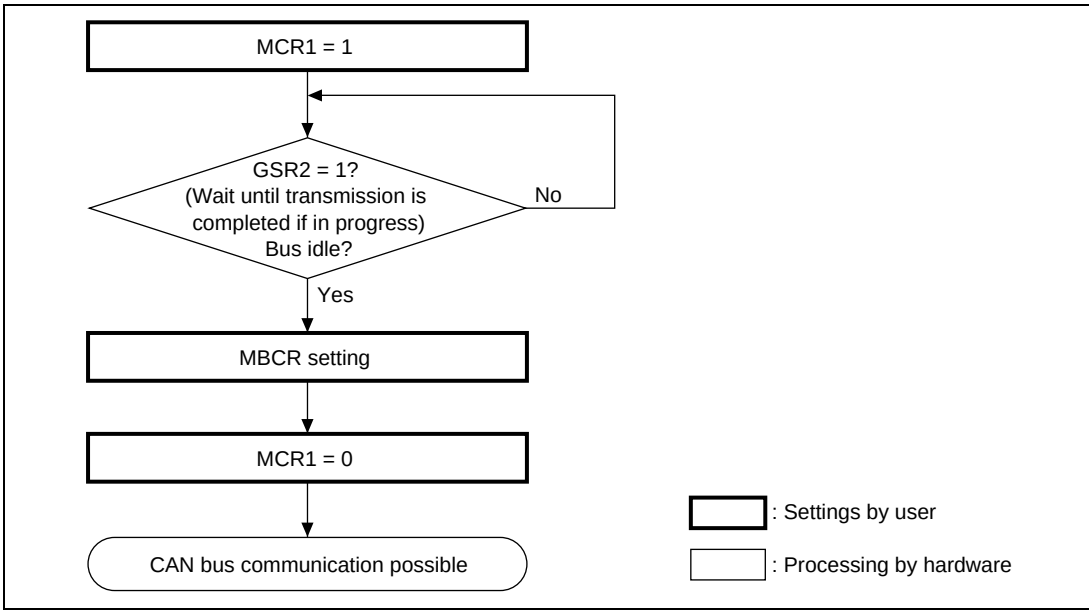


Figure 16.14 HCAN Halt Mode Flowchart

HCAN halt mode is entered by setting the halt request bit (MCR1) to 1 in the master control register (MCR). If the CAN bus is operating, the transition to HCAN halt mode is delayed until the bus becomes idle.

HCAN halt mode is cleared by clearing MCR1 to 0.

16.3.7 Interrupt Interface

There are 12 interrupt sources for each HCAN channel. Four independent interrupt vectors are assigned to each channel. Table 16.6 lists the HCAN interrupt sources.

With the exception of the power-on reset processing vector (IRR0), these sources can be masked. Masking is implemented using the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR).

Channel	IPR Bits	Vector	Number	IRR Bit	Description
HCAN0	IPRL (11–8) Interrupt priority level 0 to 15 (Initial value: 0)	ERS0	220	IRR5	Error passive interrupt (TEC ≥ 128 or REC ≥ 128)
				IRR6	Bus off interrupt (TEC ≥ 256)
		OVR0	221	IRR0	Power-on reset processing interrupt
				IRR2	Remote frame reception interrupt
				IRR3	Error warning interrupt (TEC ≥ 96)
				IRR4	Error warning interrupt (REC ≥ 96)
				IRR7	Overload frame transmission interrupt
				IRR9	Unread message overwrite interrupt
				IRR12	HCAN sleep mode CAN bus operation interrupt
		RM0	222	IRR1	Mailbox 0 message reception interrupt
				IRR1	Mailbox 1 to 15 message reception interrupt
		SLE0	223	IRR8	Message transmission/cancellation interrupt
HCAN1	IPRL (3–0) Interrupt priority level 0 to 15 (Initial value: 0)	ERS1	228	IRR5	Error passive interrupt (TEC ≥ 128 or REC ≥ 128)
				IRR6	Bus off interrupt (TEC ≥ 256)
		OVR1	229	IRR0	Power-on reset processing interrupt
				IRR2	Remote frame reception interrupt
				IRR3	Error warning interrupt (TEC ≥ 96)
				IRR4	Error warning interrupt (REC ≥ 96)
				IRR7	Overload frame transmission interrupt
				IRR9	Unread message overwrite interrupt
				IRR12	HCAN sleep mode CAN bus operation interrupt
		RM1	230	IRR1	Mailbox 0 message reception interrupt
				IRR1	Mailbox 1 to 15 message reception interrupt
		SLE1	231	IRR8	Message transmission/cancellation interrupt

16.3.8 DMAC Interface

The DMAC can be activated by reception of a message in HCAN0's mailbox 0. When DMAC transfer ends after DMAC activation has been set, the RXPR0 and RFPR0 flags are acknowledge signal automatically. An interrupt request due to a receive interrupt from the HCAN cannot be sent to the CPU in this case. Figure 16.15 shows a DMAC transfer flowchart.

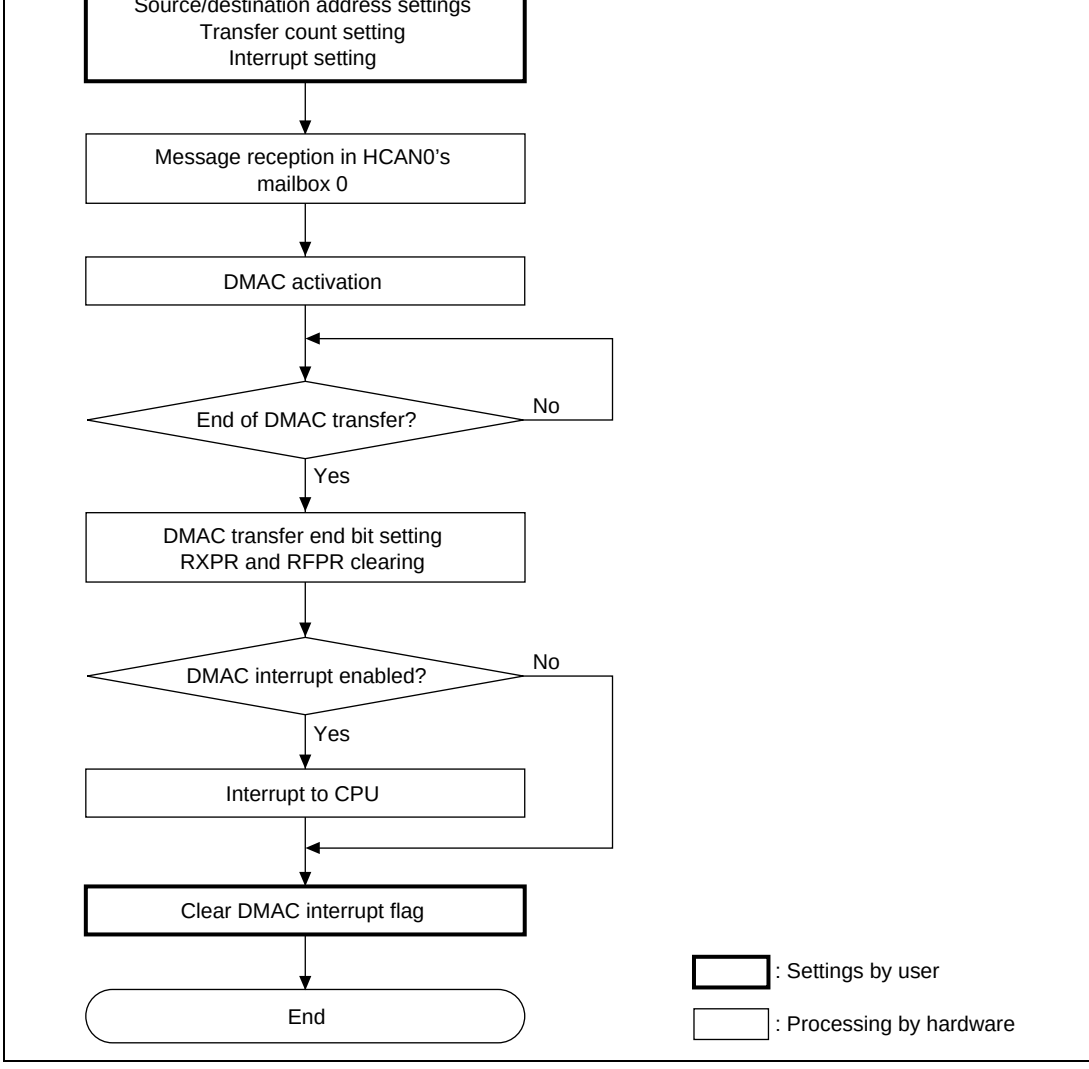


Figure 16.15 DMAC Transfer Flowchart

A bus transceiver IC is necessary to connect the SH7055SF chip to a CAN bus. A Philips PCA82C250 transceiver IC, or compatible device, is recommended. Figure 16.16 shows a sample connection diagram.

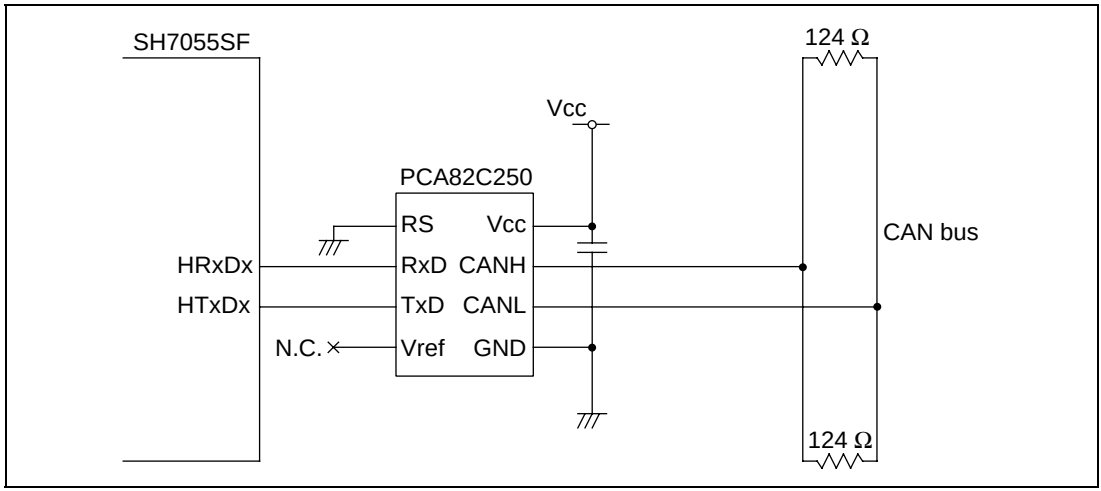


Figure 16.16 Example of High-Speed Interface Using PCA82C250

Reset: The HCAN is reset by a power-on reset, and in hardware standby mode and software standby mode. All the registers are initialized in a reset, but mailboxes (message control (MCx[x])/message data (MDx[x]) are not. However, after powering on, mailboxes (message control (MCx[x])/message data (MDx[x]) are initialized, and their values are undefined. Therefore, mailbox initialization must always be carried out after a power-on reset or a transition to hardware standby mode or software standby mode. The reset interrupt flag (IRR0) is always set after a power-on reset or recovery from software standby mode. As this bit cannot be masked in the interrupt mask register (IMR), if HCAN interrupt enabling is set in the interrupt controller without clearing the flag, an HCAN interrupt will be initiated immediately. IRR0 should therefore be cleared during initialization.

HCAN Sleep Mode: The bus operation interrupt flag (IRR12) in the interrupt register (IRR) is set by bus operation in HCAN sleep mode. Therefore, this flag is not used by the HCAN to indicate sleep mode release. Also note that the reset status bit (GSR3) in the general status register (GSR) is set in sleep mode.

Port Settings: Port settings must be made with the PFC before the HCAN begins CAN bus communication.

When using the two HCAN pins in a 2-channel/32-buffer configuration (wired-AND), set the other two HCAN pin locations as non-HCAN.

DMAC Activation: When the DMAC is activated automatically by reception of a message in HCAN0's mailbox 0 (receive-only mailbox), an interrupt request signal is not sent to the INTC.

Interrupts: When the mailbox interrupt mask register (MBIMR) is set, the interrupt register (IRR8, 2, 1) is not set by reception completion, transmission completion, or transmission cancellation for the set mailboxes.

Error Counters: In the case of error active and error passive, REC and TEC normally count up and down. In the bus off state, 11-bit recessive sequences are counted (REC + 1) using REC. If REC reaches 96 during the count, IRR4 and GSR1 are set, and if REC reaches 128, IRR7 is set.

Register Access: Byte or word access can be used on all HCAN registers. Longword access cannot be used.

Register Initialization in Standby Modes: All HCAN registers are initialized in hardware standby mode and software standby mode.

Differences from the HD64F7005:

(HRxD) is fixed to 1 as a result of faults such as the CAN bus short during message transmission or reception in the HCAN error active state. The HD64F7055S operation always meets the CAN specifications.

(1) When the CAN bus is shorted (the CAN bus is fixed to 1) during transmission

If the CAN bus is shorted while the HCAN is transmitting messages in the error active state, the conventional HD64F7055 outputs 0s consecutively during the error passive state until the transition to bus off state. In the same case, however, the HD64F7055S outputs 1s consecutively. For details, see figure 16.17.

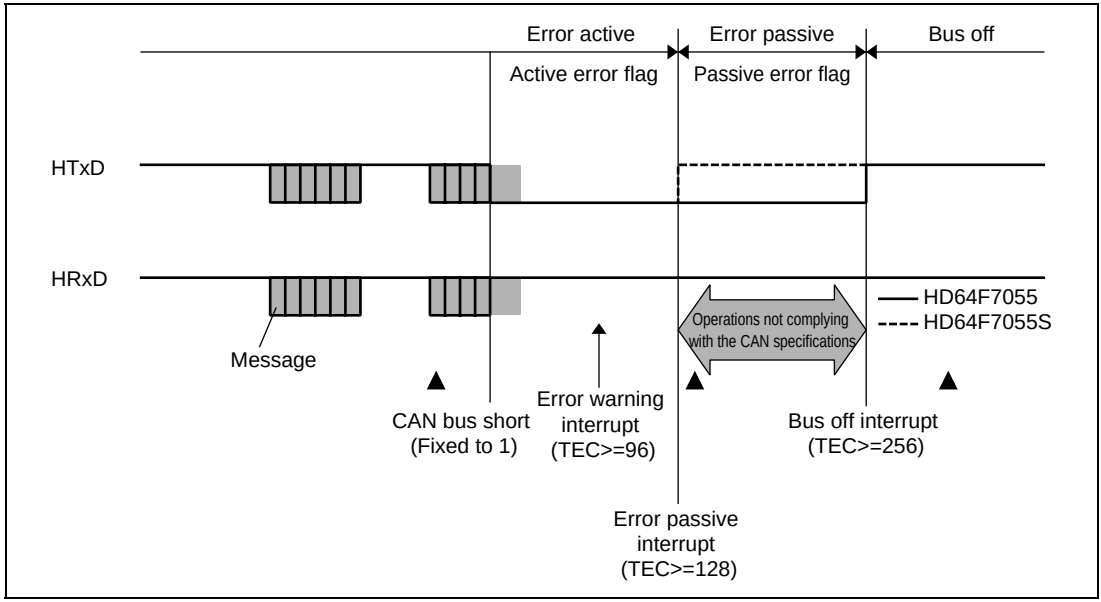


Figure 16.17 HCAN Operation while the CAN Bus is Fixed to 1 during Transmission

(2) When the CAN bus is shorted (the CAN bus is fixed to 1) during reception

If the CAN bus is shorted while the HCAN is receiving messages in the error active state, the conventional HD64F7055 outputs 0s consecutively during the error passive state. In the same case, however, the HD64F7055S outputs 1s consecutively. For details, see figure 16.18.

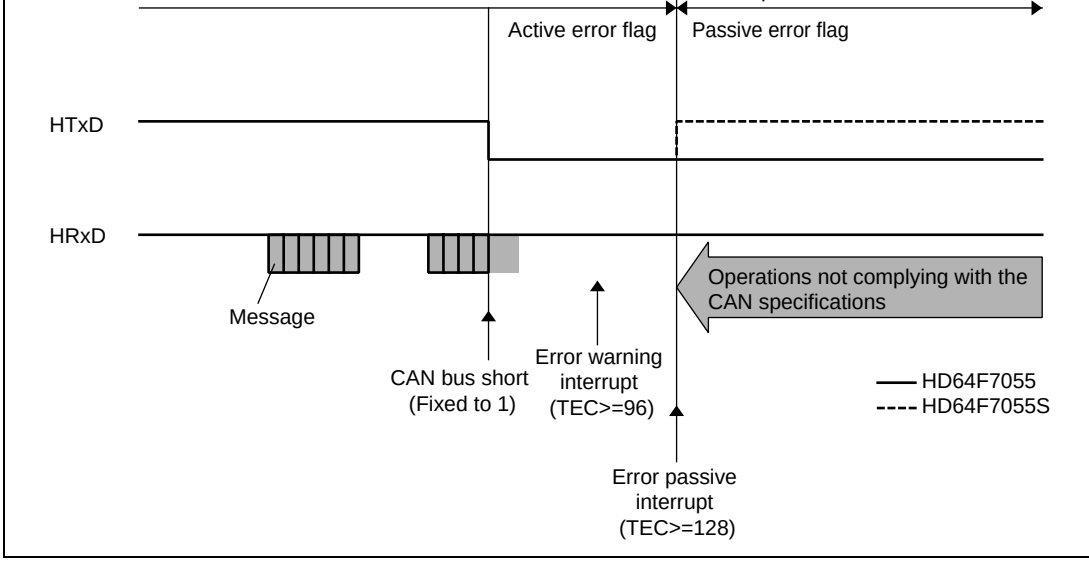


Figure 16.18 HCAN Operation while the CAN Bus is Fixed to 1 during Reception

(b) The contents of the interrupt register after recovery from the bus off state

The conventional HD64F7055 sets the interrupt register (IRR7) at the recovery of the HCAN from bus off state, while the HD64F7055S does not set the interrupt register (IRR7).

17.1 Overview

The SH7055SF includes a 10-bit successive-approximation A/D converter, with software selection of up to 32 analog input channels.

The A/D converter is composed of three independent modules, A/D, A/D1, and A/D2. A/D0 and A/D1 each comprise three groups, while A/D2 comprises two groups.

Module	Analog Groups	Channels
A/D0	Analog group 0	AN0–AN3
	Analog group 1	AN4–AN7
	Analog group 2	AN8–AN11
A/D1	Analog group 3	AN12–AN15
	Analog group 4	AN16–AN19
	Analog group 5	AN20–AN23
A/D2	Analog group 6	AN24–AN27
	Analog group 7	AN28–AN31

17.1.1 Features

The features of the A/D converter are summarized below.

- 10-bit resolution
  32 input channels (A/D0: 12 channels, A/D1: 12 channels, A/D2: 8 channels)
- High-speed conversion
  Conversion time: minimum 13.4 µs per channel (when ϕ = 40 MHz)
- Two conversion modes
  — Single mode: A/D conversion on one channel
  — Scan mode: cotinuous scan mode, single-cycle scan mode (AN0–AN3, AN4–AN7, AN8–AN11, AN12–AN15, AN16–AN19, AN20–AN23, AN24–AN27, AN28–AN31)
  Continuous conversion on 1 to 12 channels (A/D0)
  Continuous conversion on 1 to 12 channels (A/D1)
  Continuous conversion on 1 to 8 channels (A/D2)
- Thirty-two 10-bit A/D data registers
  A/D conversion results are transferred for storage into data registers corresponding to the channels.

simplifying the configuration of external analog input circuitry.

- A/D conversion interrupts and DMA function supported

  An A/D conversion interrupt request (ADI) can be sent to the CPU at the end of A/D conversion (ADI0: A/D0 interrupt request; ADI1: A/D1 interrupt request; ADI2: A/D2 interrupt request). Also, the DMAC can be activated by an ADI interrupt request.

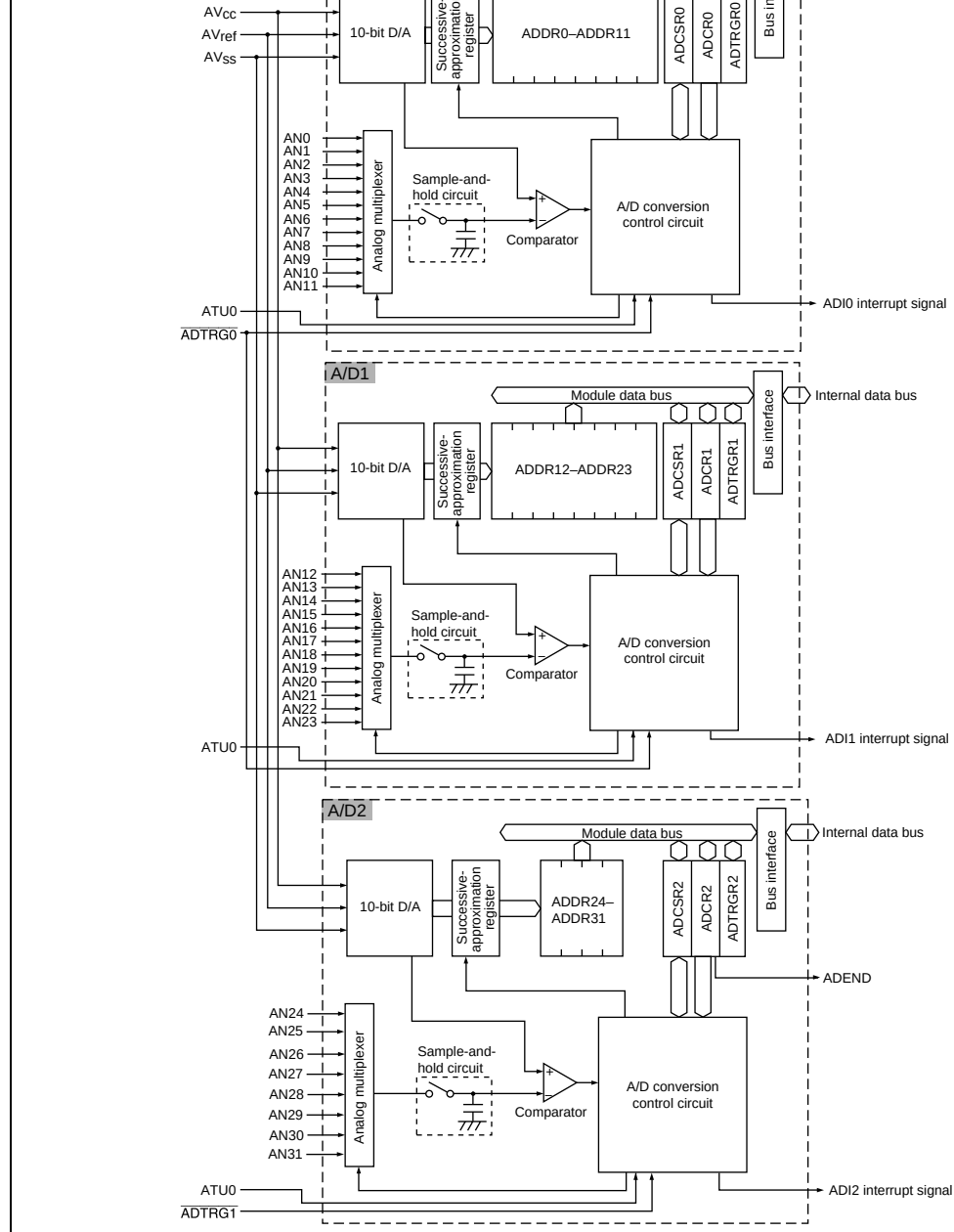
- Two kinds of conversion activation
  - Software or external trigger ($\overline{\text{ADTER0}}$, ATU-II (ITVRR2A)) can be selected (A/D0)
  - Software or external trigger ($\overline{\text{ADTGR0}}$, ATU-II (ITVRR2B)) can be selected (A/D1)
  - Software or external trigger ($\overline{\text{ADTGR1}}$, ATU-II (ITVRR1)) can be selected (A/D2)

- ADEND output

  Conversion timing can be monitored with the ADEND output pin when using channel 31 in scan mode.

17.1.2 Block Diagram

Figure 17.1 shows a block diagram of the A/D converter.



ADCR0, ADCR1, ADCR2: A/D control registers 0 to 2
ADCSR0, ADCSR1, ADCSR2: A/D control/status registers 0 to 2
ADDR0 to ADDR31: A/D data registers 0 to 31
ADTRGR0, ADTRGR1, ADTRGR2: A/D trigger registers 0 to 2

Figure 17.1 A/D Converter Block Diagram

Table 17.1 summarizes the A/D converter's input pins. There are 32 analog input pins, AN0 to AN31. The 12 pins AN0 to AN11 are A/D0 analog inputs, divided into three groups: AN0 to AN3 (group 0), AN4 to AN7 (group 1), and AN8 to AN11 (group 2). The 12 pins AN12 to AN23 are A/D1 analog inputs, divided into three groups: AN12 to AN15 (group 3), AN16 to AN19 (group 4), and AN20 to AN23 (group 5). The 8 pins AN24 to AN31 are A/D2 analog inputs, divided into two groups: AN24 to AN27 (group 6), and AN28 to AN31 (group 7).

The $\overline{\text{ADTRG0}}$ and $\overline{\text{ADTRG1}}$ pins are used to provide A/D conversion start timing from off-chip. When a low level is applied to one of these pins, A/D0, A/D1, or A/D2 starts conversion.

The ADEND pin is an output used to monitor conversion timing when channel 31 is used in scan mode.

The AV_{CC} and AV_{SS} pins are power supply voltage pins for the analog section in A/D converter modules A/D0 to A/D2. The AV_{ref} pin is the A/D converter module A/D0 to A/D2 reference voltage pin.

To maintain chip reliability, ensure that AV_{CC} = 5 V ±0.5 V and AV_{SS} = V_{SS} during normal operation, and never leave the AV_{CC} and AV_{SS} pins open, even when the A/D converter is not being used.

The voltage applied to the analog input pins should be in the range AV_{SS} = ANn = AV_{ref}.

Analog power supply pin	AV_{CC}	Input	A/D0–A/D2 analog section power supply
Analog ground pin	AV_{SS}	Input	A/D0–A/D2 analog section ground and reference†voltage
Analog reference power supply pin	AV_{ref}	Input	A/D0–A/D2 analog section reference voltage
Analog input pin 0	AN0	Input	A/D0 analog inputs 0 to 3 (analog group 0)
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	A/D0 analog inputs 4 to 7 (analog group 1)
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	A/D0 analog inputs 8 to 11 (analog group 2)
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	A/D1 analog inputs 12 to 15 (analog group 3)
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
Analog input pin 16	AN16	Input	A/D1 analog inputs 16 to 19 (analog group 4)
Analog input pin 17	AN17	Input	
Analog input pin 18	AN18	Input	
Analog input pin 19	AN19	Input	
Analog input pin 20	AN20	Input	A/D1 analog inputs 20 to 23 (analog group 5)
Analog input pin 21	AN21	Input	
Analog input pin 22	AN22	Input	
Analog input pin 23	AN23	Input	

Analog input pin 24	AN24	Input	A/D2 analog inputs 24 to 27 (analog group 6)
Analog input pin 25	AN25	Input	
Analog input pin 26	AN26	Input	
Analog input pin 27	AN27	Input	
Analog input pin 28	AN28	Input	A/D2 analog inputs 28 to 31 (analog group 7)
Analog input pin 29	AN29	Input	
Analog input pin 30	AN30	Input	
Analog input pin 31	AN31	Input	
A/D conversion trigger input pin 0	$\overline{\text{ADTRG0}}$	Input	A/D0 and A/D1 A/D conversion trigger input
A/D conversion trigger input pin 1	$\overline{\text{ADTRG1}}$	Input	A/D2 A/D conversion trigger input
ADEND output pin	ADEND	Output	A/D2 channel 31 conversion timing monitor output

Table 17.2 summarizes the A/D converter's registers.

Table 17.2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size*[1]
A/D data register 0 (H/L)	ADDR0 (H/L)	R	H'0000	H'FFFFF800	8, 16
A/D data register 1 (H/L)	ADDR1 (H/L)	R	H'0000	H'FFFFF802	8, 16
A/D data register 2 (H/L)	ADDR2 (H/L)	R	H'0000	H'FFFFF804	8, 16
A/D data register 3 (H/L)	ADDR3 (H/L)	R	H'0000	H'FFFFF806	8, 16
A/D data register 4 (H/L)	ADDR4 (H/L)	R	H'0000	H'FFFFF808	8, 16
A/D data register 5 (H/L)	ADDR5 (H/L)	R	H'0000	H'FFFFF80A	8, 16
A/D data register 6 (H/L)	ADDR6 (H/L)	R	H'0000	H'FFFFF80C	8, 16
A/D data register 7 (H/L)	ADDR7 (H/L)	R	H'0000	H'FFFFF80E	8, 16
A/D data register 8 (H/L)	ADDR8 (H/L)	R	H'0000	H'FFFFF810	8, 16
A/D data register 9 (H/L)	ADDR9 (H/L)	R	H'0000	H'FFFFF812	8, 16
A/D data register 10 (H/L)	ADDR10 (H/L)	R	H'0000	H'FFFFF814	8, 16
A/D data register 11 (H/L)	ADDR11 (H/L)	R	H'0000	H'FFFFF816	8, 16
A/D data register 12 (H/L)	ADDR12 (H/L)	R	H'0000	H'FFFFF820	8, 16
A/D data register 13 (H/L)	ADDR13 (H/L)	R	H'0000	H'FFFFF822	8, 16
A/D data register 14 (H/L)	ADDR14 (H/L)	R	H'0000	H'FFFFF824	8, 16
A/D data register 15 (H/L)	ADDR15 (H/L)	R	H'0000	H'FFFFF826	8, 16
A/D data register 16 (H/L)	ADDR16 (H/L)	R	H'0000	H'FFFFF828	8, 16
A/D data register 17 (H/L)	ADDR17 (H/L)	R	H'0000	H'FFFFF82A	8, 16
A/D data register 18 (H/L)	ADDR18 (H/L)	R	H'0000	H'FFFFF82C	8, 16
A/D data register 19 (H/L)	ADDR19 (H/L)	R	H'0000	H'FFFFF82E	8, 16
A/D data register 20 (H/L)	ADDR20 (H/L)	R	H'0000	H'FFFFF830	8, 16
A/D data register 21 (H/L)	ADDR21 (H/L)	R	H'0000	H'FFFFF832	8, 16
A/D data register 22 (H/L)	ADDR22 (H/L)	R	H'0000	H'FFFFF834	8, 16
A/D data register 23 (H/L)	ADDR23 (H/L)	R	H'0000	H'FFFFF836	8, 16
A/D data register 24 (H/L)	ADDR24 (H/L)	R	H'0000	H'FFFFF840	8, 16
A/D data register 25 (H/L)	ADDR25 (H/L)	R	H'0000	H'FFFFF842	8, 16
A/D data register 26 (H/L)	ADDR26 (H/L)	R	H'0000	H'FFFFF844	8, 16

Name	Abbreviation	R/W	Value	Address	Size*[1]
A/D data register 27 (H/L)	ADDR27 (H/L)	R	H'0000	H'FFFFF846	8, 16
A/D data register 28 (H/L)	ADDR28 (H/L)	R	H'0000	H'FFFFF848	8, 16
A/D data register 29 (H/L)	ADDR29 (H/L)	R	H'0000	H'FFFFF84A	8, 16
A/D data register 30 (H/L)	ADDR30 (H/L)	R	H'0000	H'FFFFF84C	8, 16
A/D data register 31 (H/L)	ADDR31 (H/L)	R	H'0000	H'FFFFF84E	8, 16
A/D control/status register 0	ADCSR0	R/(W)*[2]	H'00	H'FFFFF818	8, 16
A/D control register 0	ADCR0	R/W	H'0F	H'FFFFF819	8, 16
A/D trigger register 0	ADTRGR0	R/W	H'FF	H'FFFFF76E	8
A/D control/status register 1	ADCSR1	R/(W)*[2]	H'00	H'FFFFF838	8, 16
A/D control register 1	ADCR1	R/W	H'0F	H'FFFFF839	8, 16
A/D trigger register 1	ADTRGR1	R/W	H'FF	H'FFFFF72E	8
A/D control/status register 2	ADCSR2	R/(W)*[2]	H'08	H'FFFFF858	8, 16
A/D control register 2	ADCR2	R/W	H'0F	H'FFFFF859	8, 16
A/D trigger register 2	ADTRGR2	R/W	H'FF	H'FFFFF72F	8

Notes: Register accesses consist of 6 or 7 cycles for byte access and 12 or 13 cycles for word access.

*1 A 16-bit access must be made on a word boundary.

*2 Only 0 can be written to bit 7 to clear the flag.

17.2.1 A/D Data Registers 0 to 31 (ADDR0 to ADDR31)

A/D data registers 0 to 31 (ADDR0 to ADDR31) are 16-bit read-only registers that store the results of A/D conversion. There are 31 registers, corresponding to analog inputs 0 to 31 (AN0 to AN31).

The ADDR registers are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
ADDRnH (upper byte)	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
ADDRnL (lower byte)	AD1	AD0	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

(n = 0 to 31)

The A/D converter converts analog input to a 10-bit digital value. The upper 8 bits of this data are stored in the upper byte of the ADDR corresponding to the selected channel, and the lower 2 bits in the lower byte of that ADDR. Only the most significant 2 bits of the ADDR lower byte data are valid.

Table 17.3 shows correspondence between the analog input channels and A/D data registers.

Input Channel	A/D Data Register	Input Channel	A/D Data Register	Input Channel	A/D Data Register	Input Channel	A/D Data Register
AN0	ADDR0	AN8	ADDR8	AN16	ADDR16	AN24	ADDR24
AN1	ADDR1	AN9	ADDR9	AN17	ADDR17	AN25	ADDR25
AN2	ADDR2	AN10	ADDR10	AN18	ADDR18	AN26	ADDR26
AN3	ADDR3	AN11	ADDR11	AN19	ADDR19	AN27	ADDR27
AN4	ADDR4	AN12	ADDR12	AN20	ADDR20	AN28	ADDR28
AN5	ADDR5	AN13	ADDR13	AN21	ADDR21	AN29	ADDR29
AN6	ADDR6	AN14	ADDR14	AN22	ADDR22	AN30	ADDR30
AN7	ADDR7	AN15	ADDR15	AN23	ADDR23	AN31	ADDR31

17.2.2 A/D Control/Status Registers 0 and 1 (ADCSR0, ADCSR1)

A/D control/status registers 0 and 1 (ADCSR0, ADCSR1) are 8-bit readable/writable registers whose functions include selection of the A/D conversion mode for A/D0 and A/D1.

ADCSR0 and ADCSR1 are initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	ADF	ADIE	ADM1	ADM0	CH3	CH2	CH1	CH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written to clear the flag.

ADF	Description
0	Indicates that A/D0 or A/D1 is performing A/D conversion, or is in the idle state (Initial value) [Clearing conditions] • When ADF is read while set to 1, then 0 is written to ADF • When the DMAC is activated by ADI0 or ADI1
1	Indicates that A/D0 or A/D1 has finished A/D conversion, and the digital value has been transferred to ADDR [Setting conditions] • Single mode: When A/D conversion ends • Scan mode: When all set A/D conversions end

The operation of the A/D converter after ADF is set to 1 differs between single mode and scan mode.

In single mode, after the A/D converter transfers the digit value to ADDR, ADF is set to 1 and the A/D converter enters the idle state. In scan mode, ADF is set to 1 after all the set conversions end. For example, in the case of 12-channel scanning, ADF is set to 1 immediately after the end of conversion for AN8 to AN11 (group 2) or AN20 to AN23 (group 5). After ADF is set to 1, conversion continues in the case of continuous scanning, and ends in the case of single-cycle scanning.

Note that 1 cannot be written to ADF.

- Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the A/D interrupt (ADI).

  To prevent incorrect operation, ensure that the ADST bit in A/D control registers 0 and 1 (ADCR0, ADCR1) is cleared to 0 before switching the operating mode.

Bit 6: ADIE	Description
0	A/D interrupt (ADI0, ADI1) is disabled (Initial value)
1	A/D interrupt (ADI0, ADI1) is enabled

When A/D conversion ends and the ADF bit is set to 1, an A/D0 or A/D1 A/D interrupt (ADI0, ADI1) will be generated If the ADIE bit is 1. ADI0 and ADI1 are cleared by clearing ADF or ADIE to 0.

To prevent incorrect operation, ensure that the ADST bit in A/D control registers 1 and 0 (ADCR1, ADCR0) is cleared to 0 before switching the operating mode.

Bit 5: ADM1	Bit 4: ADM0	Description	
0	0	Single mode	(Initial value)
	1	4-channel scan mode (analog groups 0, 1, 2, 3, 4, 5)	
1	0	8-channel scan mode (analog groups 0, 1, 3, 4)	
	1	12-channel scan mode (analog groups 0, 1, 2, 3, 4, 5)	

When ADM1 and ADM0 are set to 00, single mode is set. In single mode, operation ends after A/D conversion has been performed once on the analog channels selected with bits CH3 to CH0 in ADCSR.

When ADM1 and ADM0 are set to 01, 4-channel scan mode is set. In scan mode, A/D conversion is performed continuously on a number of channels. The channels on which A/D conversion is to be performed in scan mode are set with bits CH3 to CH0 in ADCSR1 and ADCSR0. In 4-channel scan mode, conversion is performed continuously on the channels in one of analog groups 0 (AN0 to AN3), 1 (AN4 to AN7), 2 (AN8 to AN11), 3 (AN12 to AN15, 4 (AN16 to AN19), or 5 (AN20 to AN23).

When the ADCS bit is cleared to 0, selecting scanning of all channels within the group (AN0 to AN3, AN4 to AN7, AN8 to AN11, or AN12 to AN15, AN16 to AN19, AN20 to AN23), conversion is performed continuously, once only for each channel within the group, and operation stops on completion of conversion for the last (highest-numbered) channel.

When ADM1 and ADM0 are set to 10, 8-channel scan mode is set. In 8-channel scan mode, conversion is performed continuously on the 8 channels in analog groups 0 (AN0 to AN3) and 1 (AN4 to AN7) or analog groups 3 (AN12 to AN15) and 4 (AN16 to AN19). When the ADCS bit is cleared to 0, selecting scanning of all channels within the groups (AN0 to AN7 or AN12 to AN19), conversion is performed continuously, once only for each channel within the groups, and operation stops on completion of conversion for the last (highest-numbered) channel.

When ADM1 and ADM0 are set to 11, 12-channel scan mode is set. In 12-channel scan mode, conversion is performed continuously on the 12 channels in analog groups 0 (AN0 to AN3), 1 (AN4 to AN7), and 2 (AN8 to AN11) or analog groups 3 (AN12 to AN15), 4 (AN16 to AN19), and 5 (AN20 to AN23). When the ADCS bit is cleared to 0, selecting scanning of all channels within the groups (AN0 to AN11 or AN12 to AN19), conversion is performed continuously, once only for each channel within the groups, and operation stops on completion of conversion for the last (highest-numbered) channel.

For details of the operation in single mode and scan mode, see section 17.4, Operation.

To prevent incorrect operation, ensure that the ADST bit in A/D control registers 1 and 0 (ADCR1, ADCR0) is cleared to 0 before changing the analog input channel selection.

				Analog Input Channels			
				Single Mode		4-Channel Scan Mode	
Bit 3: CH3	Bit 2: CH2	Bit 1: CH1	Bit 0: CH0	A/D0	A/D1	A/D0	A/D1
0	0	0	0	AN0 (Initial value)	AN12 (Initial value)	AN0	AN12
			1	AN1	AN13	AN0, AN1	AN12, AN13
		1	0	AN2	AN14	AN0–AN2	AN12–AN14
			1	AN3	AN15	AN0–AN3	AN12–AN15
	1	0	0	AN4	AN16	AN4	AN16
			1	AN5	AN17	AN4, AN5	AN16, AN17
		1	0	AN6	AN18	AN4–AN6	AN16–AN18
			1	AN7	AN19	AN4–AN7	AN16–AN19
1	0*	0	0	AN8	AN20	AN8	AN20
			1	AN9	AN21	AN8, AN9	AN20, AN21
		1	0	AN10	AN22	AN8–AN10	AN20–AN22
			1	AN11	AN23	AN8–AN11	AN20–AN23

Note: * Must be cleared to 0.

CH3	CH2	CH1	CH0	A/D0	A/D1	A/D0	A/D1
0	0	0	0	AN0, AN4	AN12, AN16	AN0, AN4, AN8	ANAN12, AN16, AN20
			1	AN0, AN1, AN4, AN5	AN12, AN13, AN16, AN17	AN0, AN1, AN4, AN5, AN8, AN9	AN12, AN13, AN16, AN17, AN20, AN21
		1	0	AN0–AN2, AN4–AN6	AN12–AN14, AN16–AN18	AN0–AN2, AN4–AN6, AN8–AN10	AN12–AN14, AN16–AN18, AN20–AN22
			1	AN0–AN7	AN12–AN19	AN0–AN11	AN12–AN23
	1	0	0	AN0, AN4	AN12, AN16	AN0, AN4, AN8	AN12, AN16, AN20
			1	AN0, AN1, AN4, AN5	AN12, AN13, AN16, AN17	AN0, AN1, AN4, AN5, AN8, AN9	AN12, AN13, AN16, AN17, AN20, AN21
		1	0	AN0–AN2, AN4–AN6	AN12–AN14, AN16–AN18	AN0–AN2, AN4–AN6, AN8–AN10	AN12–AN14, AN16–AN18, AN20–AN22
			1	AN0–AN7	AN12–AN19	AN0–AN11	AN12–AN23
1	0*[1]	0	0	Reserved*[2]	Reserved*[2]	AN0, AN4, AN8	AN12, AN16, AN20
			1			AN0, AN1, AN4, AN5, AN8, AN9	AN12, AN13, AN16, AN17, AN20, AN21
		1	0			AN0–AN2, AN4–AN6, AN8–AN10	AN12–AN14, AN16–AN18, AN20–AN22
			1			AN0–AN11	AN12–AN23

Notes: *1 Must be cleared to 0.

*2 These modes are provided for future expansion, and cannot be used at present.

A/D control registers 0 to 2 (ADCR0 to ADCR2) are 8-bit readable/writable registers that control the start of A/D conversion and selects the operating clock for A/D0 to A/D2.

ADCR0 to ADCR2 are initialized to H'0F by a power-on reset, and in hardware standby mode and software standby mode.

Bits 3 to 0 of ADCR0 to ADCR2 are reserved. These bits cannot be modified. These bits are always read as 1.

Bit:	7	6	5	4	3	2	1	0
	TRGE	CKS	ADST	ADCS	—	—	—	—
Initial value:	0	0	0	0	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

- Bit 7—Trigger Enable (TRGE): Enables or disables triggering of A/D conversion by external input or the ATU-II.

Bit 7: TRGE	**Description**	
0	A/D conversion triggering by external input or ATU-II is disabled	(Initial value)
1	A/D conversion triggering by external input or ATU-II is enabled	

For details of external or ATU-II trigger selection, see section 17.2.5, A/D Trigger Registers 0 to 2 (ADTRGR0 to ADTRGR2).

When ATU triggering is selected, clear bit 7 of registers ADTRGR0 to ADTRGR2 to 0.

When external triggering is selected, upon input of a low level to the $\overline{\text{ADTRG0}}$ or $\overline{\text{ADTRG1}}$ pin after TRGE has been set to 1, the A/D converter detects the low level and sets the ADST bit to 1 in ADCR. The same operation is subsequently performed when 1 is written in the ADST bit by software. External triggering of A/D conversion is only enabled when the ADST bit is cleared to 0.

When external triggering is used, the low level input to the $\overline{\text{ADTRG0}}$ or $\overline{\text{ADTRG1}}$ pin must be at least 1.5 Pϕ clock cycles in width. For details, see section 17.4.4, External Triggering of A/D Conversion.

is cleared to 0 before changing the A/D conversion time. For details, see section 17.4.3, Analog Input Sampling and A/D Conversion Time.

Bit 6: CKS	Description	
0	Conversion time = 532 states (maximum)	(Initial value)
1	Conversion time = 268 states (maximum)	

- Bit 5—A/D Start (ADST): Starts or stops A/D conversion. A/D conversion is started when ADST is set to 1, and stopped when ADST is cleared to 0.

Bit 5: ADST	Description	
0	A/D conversion is stopped	(Initial value)
1	A/D conversion is being executed [Clearing conditions] • Single mode: Automatically cleared to 0 when A/D conversion ends • Scan mode: Automatically cleared to 0 on completion of one round of conversion on all set channels (single-cycle scan)	

Note that the operation of the ADST bit differs between single mode and scan mode.

In single mode, ADST is automatically cleared to 0 when A/D conversion ends on one channel. In scan mode (continuous scan), when all conversions have ended for the selected analog inputs, ADST remains set to 1 in order to start A/D conversion again for all the channels. Therefore, in scan mode (continuous scan), the ADST bit must be cleared to 0, stopping A/D conversion, before changing the conversion time or the analog input channel selection. However, in scan mode (single-cycle scan), the ADST bit is automatically cleared to 0, stopping A/D conversion, when one round of conversion ends on all the set channels.

Ensure that the ADST bit in ADCR0 to ADCR2 is cleared to 0 before switching the operating mode.

Also, make sure that A/D conversion is stopped (ADST is cleared to 0) before changing A/D interrupt enabling (bit ADIE in ADCSR0 to ADCSR2), the A/D conversion time (bit CKS in ADCR0 to ADCR2), the operating mode (bits ADM1 and ADM0 in ADSCR0 to ADCSR2), or the analog input channel selection (bits CH3 to CH0 in ADCSR0 to ADCSR2). The A/D data register contents will not be guaranteed if these changes are made while the A/D converter is operating (ADST is set to 1).

for details.

Bit 4: ADCS	Description	
0	Single-cycle scan	(Initial value)
1	Continuous scan	

- Bits 3 to 0—Reserved: These bits are always read as 1. The write value should always be 1.

17.2.4 A/D Control/Status Register 2 (ADCSR2)

A/D control/status register 2 (ADCSR2) is an 8-bit readable/writable register whose functions include selection of the A/D conversion mode for A/D2.

ADCSR2 is initialized to H'08 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	ADF	ADIE	ADM1	ADM0	—	CH2	CH1	CH0
Initial value:	0	0	0	0	1	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * Only 0 can be written to clear the flag.

- Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7: ADF	Description	
0	Indicates that A/D2 is performing A/D conversion, or is in the idle state [Clearing conditions] • When ADF is read while set to 1, then 0 is written to ADF • When the DMAC is activated by ADI2	(Initial value)
1	Indicates that A/D2 has finished A/D conversion, and the digital value has been transferred to ADDR [Setting conditions] • Single mode: When A/D conversion ends • Scan mode: When all set A/D conversions end	

The operation of the A/D converter after ADF is set to 1 differs between single mode and scan mode.

after the end of conversion for AN28 to AN31 (group 7). After ADF is set to 1, conversion continues in the case of continuous scanning, and ends in the case of single-cycle scanning.

Note that 1 cannot be written to ADF.

- Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the A/D interrupt (ADI).

  To prevent incorrect operation, ensure that the ADST bit in A/D control register 2 (ADCR2) is cleared to 0 before switching the operating mode.

Bit 6: ADIE	Description	
0	A/D interrupt (ADI2) is disabled	(Initial value)
1	A/D interrupt (ADI2) is enabled	

When A/D conversion ends and the ADF bit in ADCSR2 is set to 1, an A/D2 A/D interrupt (ADI2) will be generated If the ADIE bit is 1. ADI2 is cleared by clearing ADF or ADIE to 0.

- Bits 5 and 4: A/D Mode 1 and 0 (ADM1, ADM0): These bits select the A/D conversion mode from single mode, 4-channel scan mode,and 8-channel scan mode.

  To prevent incorrect operation, ensure that the ADST bit in A/D control register 2 (ADCR2) is cleared to 0 before switching the operating mode.

Bit 5: ADM1	Bit 4: ADM0	Description	
0	0	Single mode	(Initial value)
	1	4-channel scan mode (analog groups 6 and 7)	
1	0	8-channel scan mode (analog groups 6 and 7)	
	1	Reserved	

When ADM1 and ADM0 are set to 00, single mode is set. In single mode, operation ends after A/D conversion has been performed once on the analog channels selected with bits CH2 to CH0 in ADCSR.

When ADM1 and ADM0 are set to 01, 4-channel scan mode is set. In scan mode, A/D conversion is performed continuously on a number of channels. The channels on which A/D conversion is to be performed in scan mode are set with bits CH2 to CH0 in ADCSR2. In 4-channel scan mode, conversion is performed continuously on the channels in one of analog groups 6 (AN24 to AN27) or 7 (AN28 to AN31).

When the ADCS bit is cleared to 0, selecting scanning of all channels within the group (AN24 to AN27, AN28 to AN31), conversion is performed continuously, once only for each channel within the group, and operation stops on completion of conversion for the last (highest-numbered) channel.

and 7 (AN28 to AN31). When the ADCS bit is cleared to 0, selecting scanning of all channels within the groups (AN24 to AN31), conversion is performed continuously, once only for each channel within the groups, and operation stops on completion of conversion for the last (highest-numbered) channel.

For details of the operation in single mode and scan mode, see section 17.4, Operation.

- Bit 3—Reserved: This bit is always read as 1. The write value should always be 0.
- Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits, together with the ADM1 and ADM0 bits, select the analog input channels.

  To prevent incorrect operation, ensure that the ADST bit in A/D control register 2 (ADCR2) is cleared to 0 before changing the analog input channel selection.

			Analog Input Channels		
Bit: CH2	**Bit: CH1**	**Bit: CH0**	**Single Mode**	**4-Channel Scan Mode**	**8-Channel Scan Mode**
0	0	0	AN24 (Initial value)	AN24	AN24, AN28
		1	AN25	AN24, AN25	AN24, AN25, AN28, AN29
	1	0	AN26	AN24–AN26	AN24–AN26, AN28–AN30
		1	AN27	AN24–AN27	AN24–AN31
1	0	0	AN28	AN28	AN24, AN28
		1	AN29	AN28, AN29	AN24, AN25, AN28, AN29
	1	0	AN30	AN28–AN30	AN24–AN26, AN28–AN30
		1	AN31	AN28–AN31	AN24–AN31

The A/D trigger registers (ADTRGR0 to ADTRGR2) are 8-bit readable/writable registers that select the A/D0, A/D1, and A/D2 triggers. Either external pin ($\overline{\text{ADTRG0}}$, $\overline{\text{ADTRG1}}$) or ATU-II (ATU-II interval timer A/D conversion request) triggering can be selected.

ADTRGR0 to ADTRGR2 are initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	EXTRG	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R	R	R	R	R	R	R

- Bit 7—Trigger Enable (EXTRG): Selects external pin input ($\overline{\text{ADTRG0}}$, $\overline{\text{ADTRG1}}$) or the ATU-II interval timer A/D conversion request.

Bit 7: EXTRG	**Description**
0	A/D conversion is triggered by the ATU-II channel 0 interval timer A/D conversion request
1	A/D conversion is triggered by external pin input ($\overline{\text{ADTRG}}$) (Initial value)

In order to select external triggering or ATU-II triggering, the TGRE bit in ADCR0 to ADCR2 must be set to 1. For details, see section 17.2.3, A/D Control Registers 0 to 2 (ADCR0 to ADCR2).

- Bits 6 to 0—Reserved: These bits are always read as 1. The write value should always be 1.

A/D data registers 0 to 31 (ADDR0 to ADDR31) are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, the upper and lower bytes must be read separately.

To prevent the data being changed between the reads of the upper and lower bytes of an A/D data register, the lower byte is read via a temporary register (TEMP). The upper byte can be read directly.

Data is read from an A/D data register as follows. When the upper byte is read, the upper-byte value is transferred directly to the CPU and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When performing byte-size reads on an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained. If a word-size read is performed on an A/D data register, reading is performed in upper byte, lower byte order automatically.

Figure 17.2 shows the data flow for access to an A/D data register.

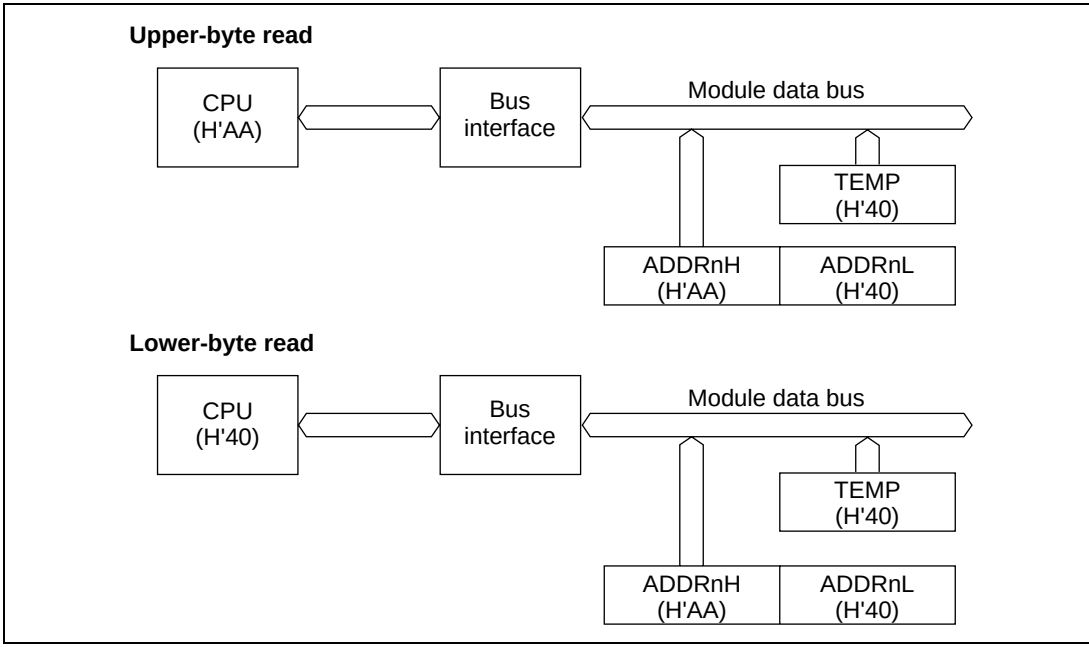


Figure 17.2 A/D Data Register Access Operation (Reading H'AA40)

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode. There are two kinds of scan mode: continuous and single-cycle. In single mode, conversion is performed once on one specified channel, then ends. In continuous scan mode, A/D conversion continues on one or more specified channels until the ADST bit is cleared to 0. In single-cycle scan mode, A/D conversion ends after being performed once on one or more channels.

17.4.1 Single Mode

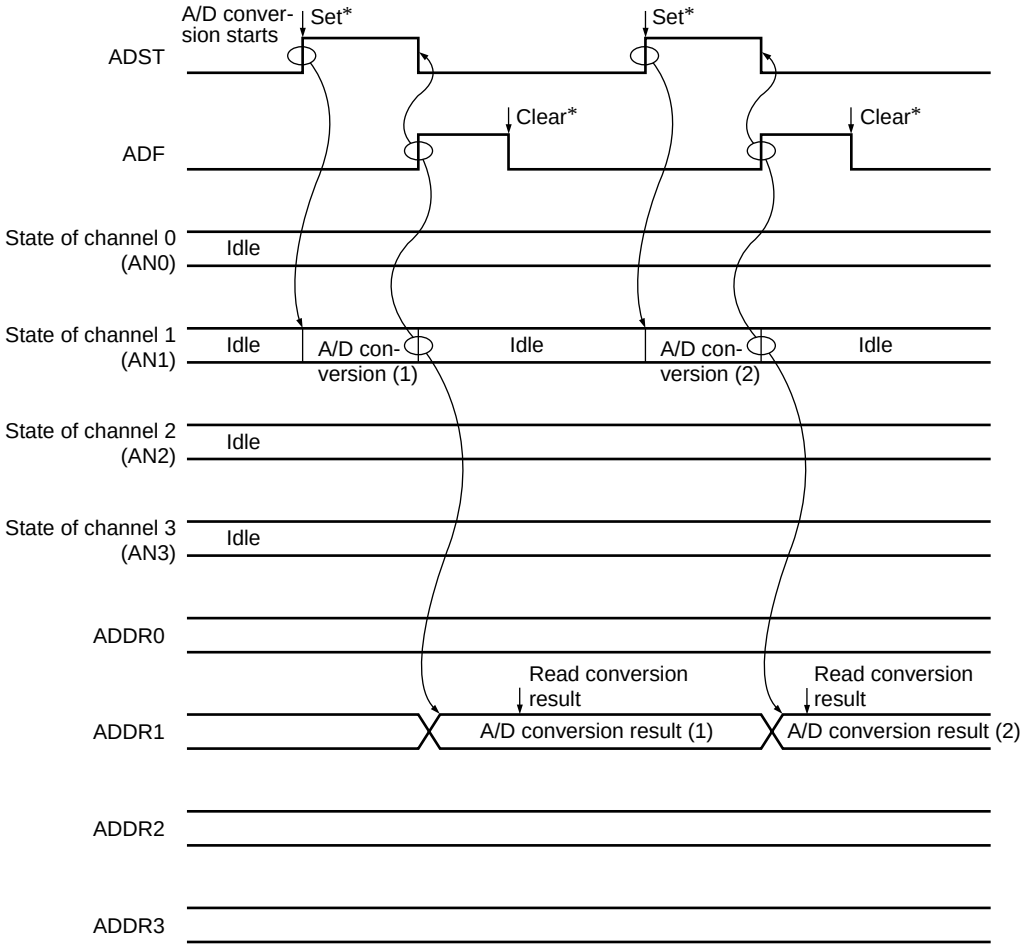
Single mode, should be selected when only one A/D conversion on one channel is required. Single mode is selected by setting the ADM1 and ADM0 bits in the A/D control/status register (ADSCR) to 00. When the ADST bit in the A/D control register (ADCR) is set to 1, A/D conversion is started in single mode.

The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends.

When conversion ends, the ADF flag in ADCSR is set to 1. If the ADIE bit in ADCSR is also 1, an ADI interrupt is requested. To clear the ADF flag, first read ADF when set to 1, then write 0 to ADF. If the DMAC is activated by the ADI interrupt, ADF is cleared automatically.

An example of the operation when analog input channel 1 (AN1) is selected and A/D conversion is performed in single mode is described next. Figure 17.3 shows a timing diagram for this example.

1. Single mode is selected (ADM1 = ADM0 = 0), input channel AN1 is selected (CH3 = CH2 = CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred to ADDR1. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine is started.
5. The routine reads ADF set to 1, then writes 0 to ADF.
6. The routine reads and processes the conversion result (ADDR1).
7. Execution of the A/D interrupt handling routine ends. After this, if the ADST bit is set to 1, A/D conversion starts again and steps 2 to 7 are repeated.



Note: * Vertical arrows (↓) indicate instructions executed by software.

Figure 17.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. Scan mode is selected for A/D0 or A/D1 by setting the ADM1 and ADM0 bits in A/D control/status register 0 or 1 (ADSCR0 or ADSCR1) to 01 (4-channel scan mode), 10 (8-channel scan mode), or 11 (12-channel scan mode).

For A/D2, scan mode is selected by setting the ADM1 and ADM0 bits in A/D control/status register 2 (ADCSR2) to 01 (4-channel scan mode) or 10 (8-channel scan mode). When the ADCS bit is cleared to 0 and the ADST bit is set to 1 in the A/D control register (ADCR), single-cycle scanning is performed. When the ADCS bit is set to 1 and the ADST bit is set to 1, continuous scanning is performed.

In scan mode, A/D conversion is performed in low-to-high analog input channel number order (AN0, AN1 ... AN11, AN12, AN13 ... AN23, AN24, AN25 ... AN31).

In single-cycle scanning, the ADF bit in ADCSR is set to 1 when conversion has been performed once on all the set channels, and the ADST bit is automatically cleared to 0.

In continuous scanning, the ADF bit in ADCSR is set to 1 when conversion ends on all the set channels. To stop A/D conversion, write 0 to the ADST bit.

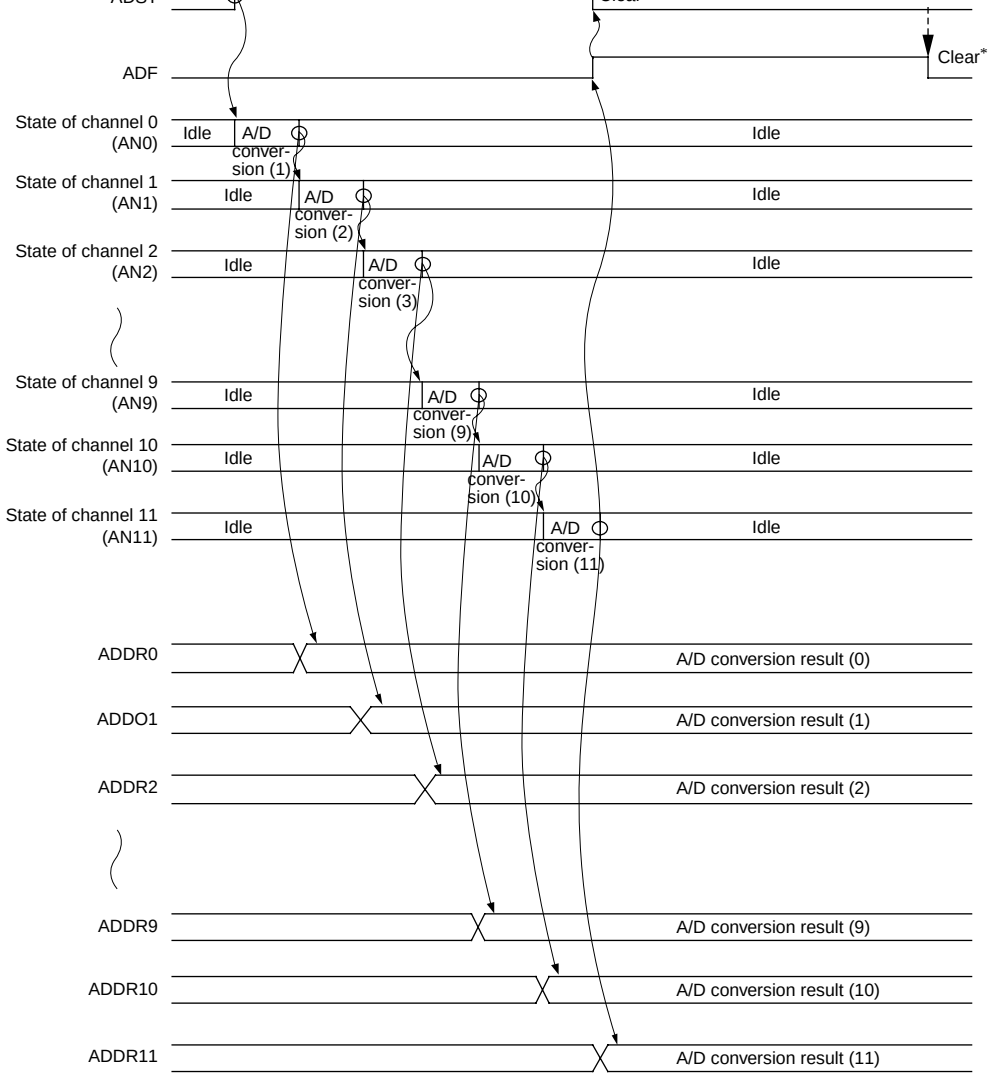
If the ADIE bit in ADCSR is set to 1 when ADF is set to 1, an ADI interrupt (ADI0, ADI1, or ADI2) is requested. To clear the ADF flag, first read ADF when set to 1, then write 0 to ADF. If the DMAC is activated by the ADI interrupt, ADF is cleared to 0 automatically.

An example of the operation when analog inputs 0 to 11 (AN0 to AN11) are selected and A/D conversion is performed in single-cycle scan mode is described below. Figure 17.4 shows the operation timing for this example.

1. 12-channel scan mode is selected (ADM1 = 1, ADM0 = 1), single-cycle scan mode is selected (ADCS = 0), analog input channels AN0 to AN11 are selected (CH3 = 0, CH2 = 0, CH1 = 1, CH0 = 1), and A/D conversion is started.
2. When conversion of the first channel (AN0) is completed, the result is transferred to ADDR0. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the 12th channel (AN11).
4. When conversion is completed for all the selected channels (AN0 to AN11), the ADF flag is set to 1, the ADST bit is cleared to 0 automatically, and A/D conversion stops. If the ADIE bit is 1, an ADI interrupt is requested after A/D conversion ends.

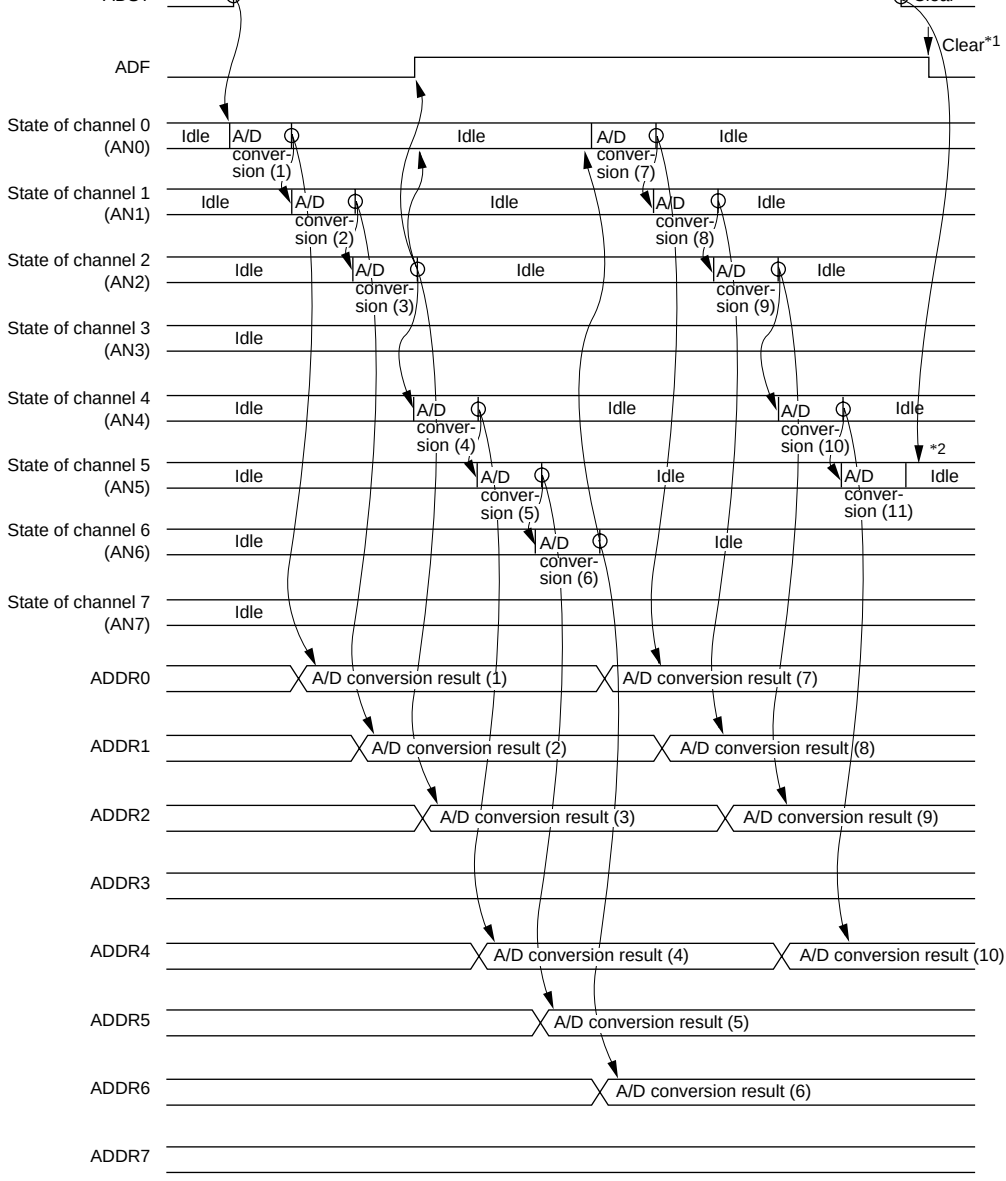
17.3 shows the operation timing.

1. 8-channel scan mode is selected (ADM1 = 1, ADM0 = 0) continuous scan mode is selected (ADCS = 1), analog input channels AN0 to AN2 and AN4 to AN6 are selected (CH3 = 0, CH2 = 0, CH1 = 1, CH0 = 0), and A/D conversion is started.
2. When conversion of the first channel (AN0) is completed, the result is transferred to ADDR0. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. Conversion of the fourth channel (AN4) starts automatically.
5. Conversion proceeds in the same way through the sixth channel (AN6)
6. When conversion is completed for all the selected channels (AN0 to AN2 and AN4 to AN6), the ADF flag is set to 1. If the ADIE bit is also 1, an ADI interrupt is requested.
7. Steps 2 to 6 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After this, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).



Note: * Vertical arrows (↓) indicate instructions executed by software.

Figure 17.4 Example of A/D Converter Operation (Scan Mode (Single-Cycle Scan), Channels AN0 to AN11 Selected)



Notes: *1 Vertical arrows (↓) indicate instructions executed by software.
*2 Data currently being converted is ignored.

Figure 17.5 Example of A/D Converter Operation (Scan Mode (Continuous Scan), Channels AN0 to AN2 and AN4 to AN6 Selected)

The A/D converter has a built-in sample-and-hold circuit in A/D0, A/D1, and A/D2. The A/D converter samples the analog input at time t_D (A/D conversion start delay time) after the ADST bit is set to 1, then starts conversion. Figure 17.6 shows the A/D conversion timing.

The A/D conversion time (t_{CONV}) includes t_D and the analog input sampling time (t_{SPL}). The length of t_D is not fixed, since it includes the time required for synchronization of the A/D conversion operation. The total conversion time therefore varies within the ranges shown in table 17.4.

In scan mode, the t_{CONV} values given in table 17.4 apply to the first conversion. In the second and subsequent conversions, t_{CONV} is fixed at 512 states when CKS = 0 or 256 states when CKS = 1.

Table 17.4 A/D Conversion Time (Single Mode)

		CKS = 0: φ = 20 to 40 MHz			CKS = 1: φ = 20 MHz			
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
A/D conversion start delay time	t_D	20	—	34	12	—	18	States (φ base)
Input sampling time	t_{SPL}	—	128	—	—	64	—	
A/D conversion time	t_{CONV}	518	—	532	262	—	268	

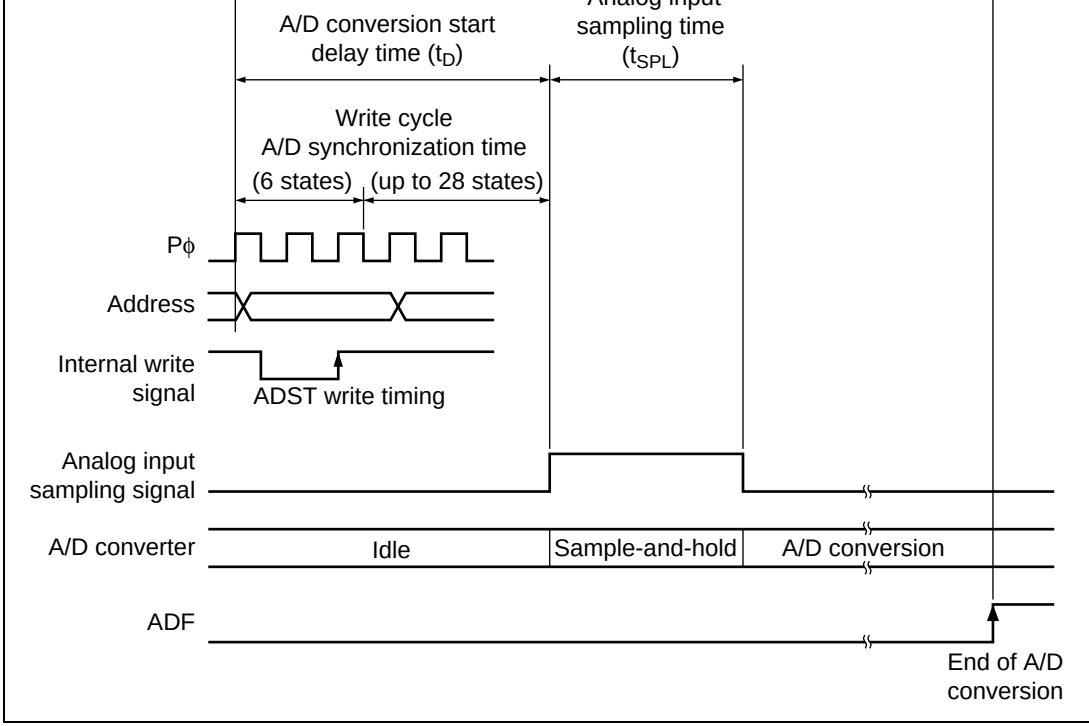


Figure 17.6 A/D Conversion Timing

A/D conversion can be externally triggered. To activate the A/D converter with an external trigger, first set the pin functions with the PFC (pin function controller) and input a high level to the $\overline{\text{ADTRG}}$ pin, then set the TRGE bit to 1 and clear the ADST bit to 0 in the A/D control register (ADCR), and set the EXTRG bit to 1 in the A/D trigger register (ADTRGR). When a low level is input to the $\overline{\text{ADTRG}}$ pin after these settings have been made, the A/D converter detects the low level and sets the ADST bit to 1. If a low level is being input to the $\overline{\text{ADTRG}}$ pin when A/D conversion ends, the ADST bit is set to 1 again, and A/D conversion is started. Figure 17.7 shows the timing for external trigger input.

The ADST bit is set to 1 two states after the A/D converter samples the low level on the $\overline{\text{ADTRG}}$ pin. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written into the ADST bit by software.

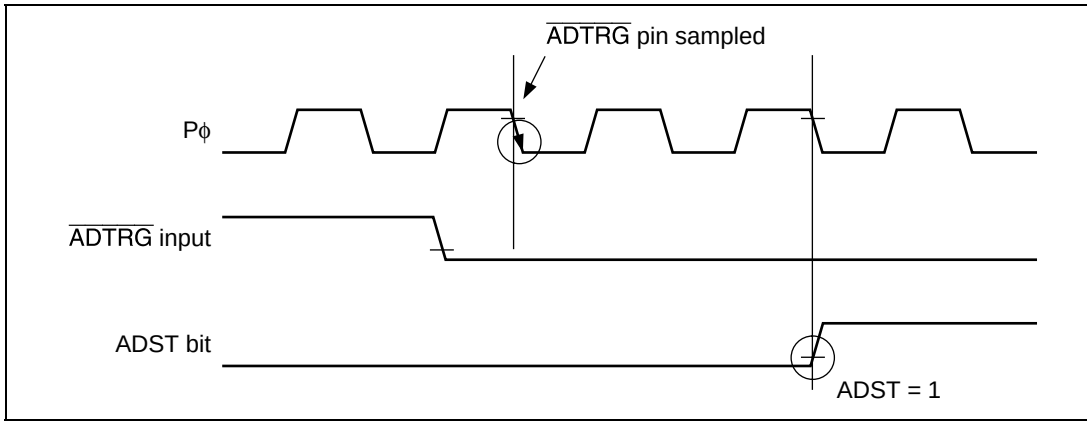


Figure 17.7 External Trigger Input Timing

The A/D0, A/D1, and A/D2 converter modules can be activated by an A/D conversion request from the ATU-II's channel 0 interval timer.

To activate the A/D converter by means of the ATU-II, set the TRGE bit to 1 in the A/D control register (ADCR) and clear the EXTRG bit to 0 in the A/D trigger register (ADTRGR). When an ATU-II channel 0 interval timer A/D conversion request is generated after these settings have been made, the ADST bit set to 1. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written into the ADST bit by software.

17.4.6 ADEND Output Pin

When channel 31 is used in scan mode, the conversion timing can be monitored with the ADEND output pin.

After the channel 31 analog voltage has been latched in scan mode, and conversion has started, the ADEND pin goes high. The ADEND pin subsequently goes low when channel 31 conversion ends.

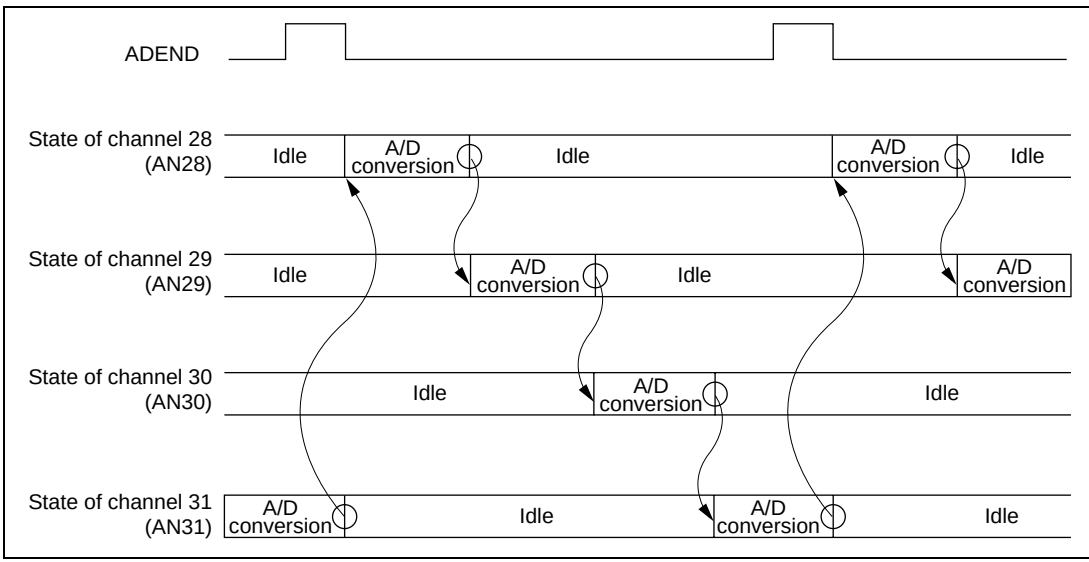


Figure 17.8 ADEND Output Timing

The A/D converter can generate an A/D conversion end interrupt request (ADI0, ADI1, or ADI2) upon completion of A/D conversions. The ADI interrupt can be enabled by setting the ADIE bit in the A/D control/status register (ADCSR) to 1, or disabled by clearing the ADIE bit to 0.

The DMAC can be activated by an ADI interrupt. In this case an interrupt request is not sent to the CPU.

When the DMAC is activated by an ADI interrupt, the ADF bit in ADCSR is automatically cleared when data is transferred by the DMAC.

See section 10.4.2, Example of DMA Transfer between A/D Converter and On-Chip Memory (Address Reload On), for an example of this operation.

17.6 Usage Notes

The following points should be noted when using the A/D converter.

1. Analog input voltage range

   The voltage applied to analog input pins during A/D conversion should be in the range $AV_{SS} \leq AN_n \leq AV_{ref}$.
2. Relation between, AV_{SS}, AV_{CC}, and V_{SS}, V_{CC}

   When using the A/D converter, set AV_{CC} = 5.0 V ±0.5 V, and $AV_{SS} = V_{SS}$. When the A/D converter is not used, set $AV_{SS} = V_{SS}$, and do not leave the AV_{CC} pin open.
3. AV_{ref} input range

   Set AV_{ref} = 4.5 V to AV_{CC} when the A/D converter is used, and $AV_{ref} \leq AV_{CC}$ when not used.

   If conditions above are not met, the reliability of the device may be adversely affected.
4. Notes on board design

   In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

   Also, digital circuitry must be isolated from the analog input signals (ANn), analog reference voltage (AV_{ref}), and analog power supply (AV_{CC}) by the analog ground (AV_{SS}). AV_{SS} should be connected at one point to a stable digital ground (V_{SS}) on the board.
5. Notes on noise countermeasures

   A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (ANn) and analog reference voltage (AV_{ref}) should be connected between AV_{CC} and AV_{SS} as shown in figure 17.9.

input currents at the analog input pins (ANn) are averaged, and so an error may arise. Careful consideration is therefore required when deciding the circuit constants.

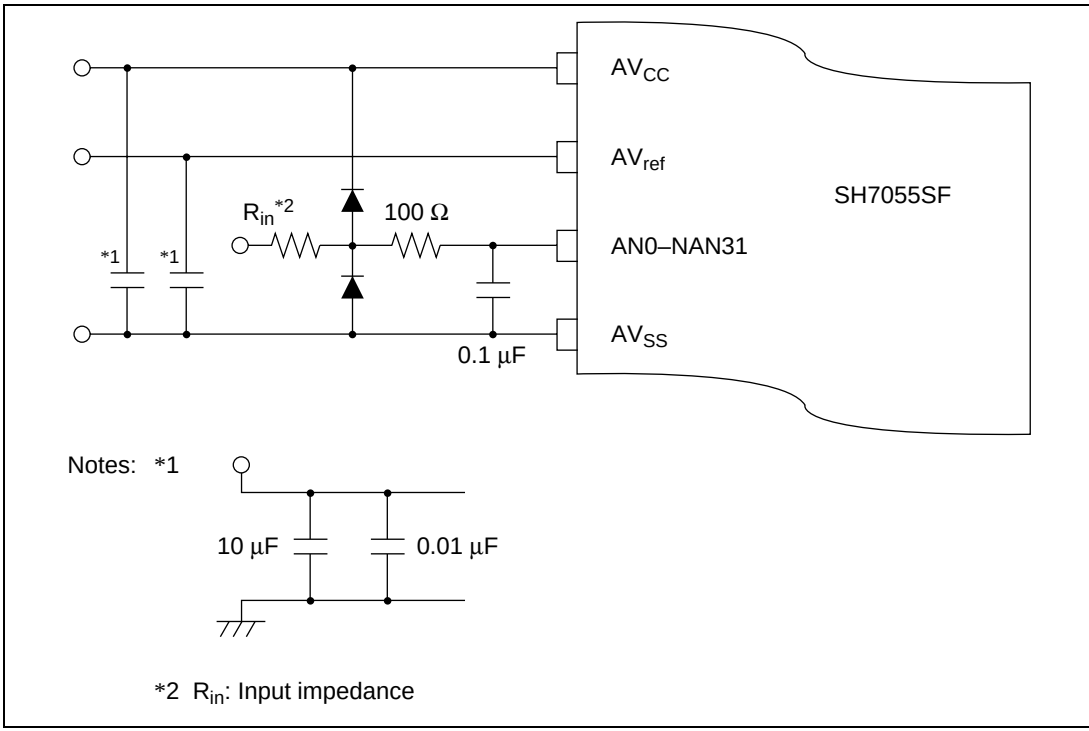


Figure 17.9 Example of Analog Input Pin Protection Circuit

Table 17.5 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	3	kΩ

A/D conversion accuracy definitions are given below.

1. Resolution

   The number of A/D converter digital conversion output codes

2. Offset error

   The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 0000000000 to 0000000001 (does not include quantization error) (see figure 17.10).

3. Full-scale error

   The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 1111111110 to 111111111 (does not include quantization error) (see figure 17.10).

4. Quantization error

   The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 17.10).

5. Nonlinearity error

   The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error.

6. Absolute accuracy

   The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

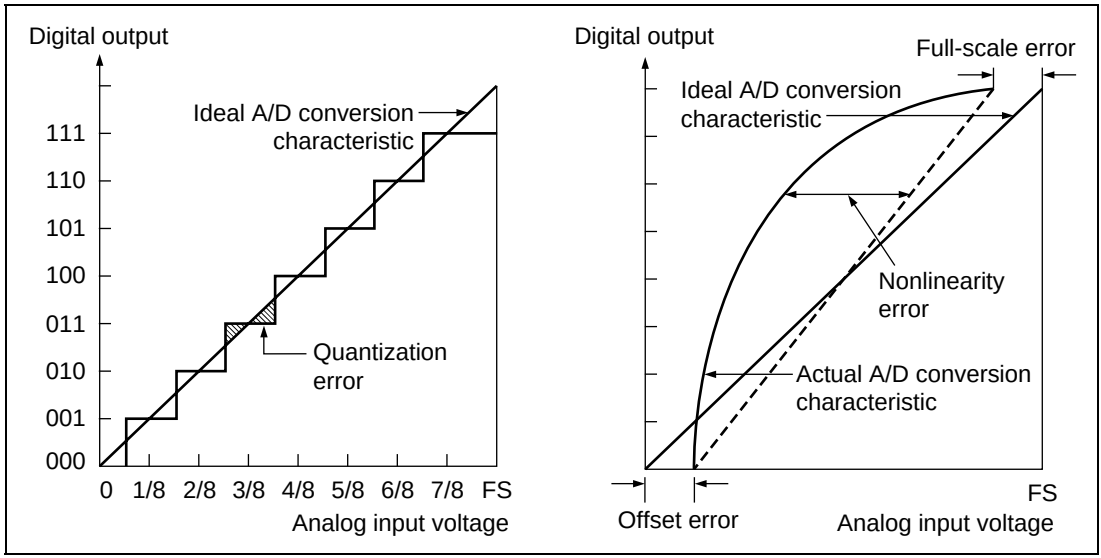


Figure 17.10 A/D Conversion Accuracy Definitions

18.1 Overview

The high-performance user debug interface (H-UDI) provides data transfer and interrupt request functions. The H-UDI performs serial transfer by means of external signal control.

18.1.1 Features

The H-UDI has the following features conforming to the IEEE 1149.1 standard:

- Five test signals (TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$)
- TAP controller
- Instruction register
- Data register
- Bypass register

The H-UDI has two instructions:

- Bypass mode
  Test mode conforming to IEEE 1149.1
- H-UDI interrupt
  H-UDI interrupt request to INTC

The SH7055SF does not support test modes other than the bypass mode.

Figure 18.1 shows a block diagram of the H-UDI.

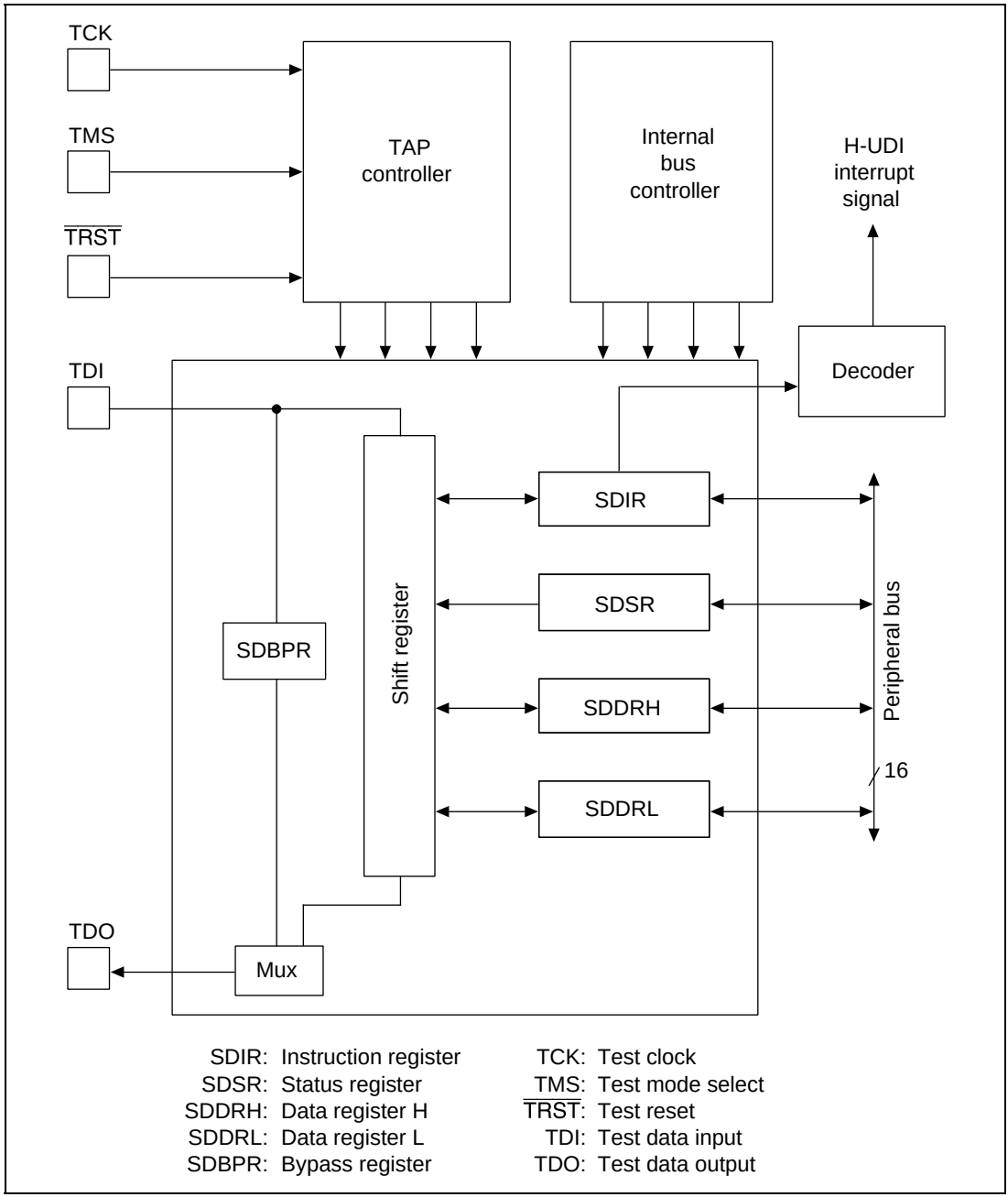


Figure 18.1 H-UDI Block Diagram

Table 18.1 shows the H-UDI pin configuration.

Table 18.1 H-UDI Pins

Name	Abbreviation	I/O	Function
Test clock	TCK	Input	Test clock input
Test mode select	TMS	Input	Test mode select input signal
Test data input	TDI	Input	Serial data input
Test data output	TDO	Output	Serial data output
Test reset	$\overline{\text{TRST}}$	Input	Test reset input signal

18.1.4 Register Configuration

Table 18.2 shows the H-UDI registers.

Table 18.2 H-UDI Registers

Register	Abbreviation	R/W*[1]	Initial Value*[2]	Address	Access Size (Bits)
Instruction register	SDIR	R	H'F000	H'FFFFF7C0	8/16/32
Status register	SDSR	R/W	H'0201	H'FFFFF7C2	8/16/32
Data register H	SDDRH	R/W	Undefined	H'FFFFF7C4	8/16/32
Data register L	SDDRL	R/W	Undefined	H'FFFFF7C6	8/16/32
Bypass register	SDBPR	—	—	—	—

Notes: *1 Indicates whether the register can be read and written to by the CPU.
*2 Initial value when the $\overline{\text{TRST}}$ signal is input. Not initialized by a reset (power-on or manual) or in software standby mode.

Instructions and data can be input to the instruction register (SDIR) and data register (SDDR) by serial transfer from the test data input pin (TDI). Data from SDIR, the status register (SDSR), and SDDR can be output via the test data output pin (TDO). The bypass register (SDBPR) is a one-bit register that is connected to TDI and TDO in bypass mode. Except for SDBPR, all the registers can be accessed by the CPU.

Table 18.3 shows the kinds of serial transfer that can be used with each of the H-UDI's registers.

SDIR	Possible	Possible
SDSR	Not possible	Possible
SDDRH	Possible	Possible
SDDRL	Possible	Possible
SDBPR	Possible	Possible

18.2 External Signals

18.2.1 Test Clock (TCK)

The test clock pin (TCK) supplies an independent clock to the H-UDI. As the clock input to TCK is supplied directly to the H-UDI, a clock waveform with a duty ratio close to 50% should be input (see section 25, Electrical Characteristics, for details). If no signal is input, TCK is fixed at 1 by internal pull-up.

18.2.2 Test Mode Select (TMS)

The test mode select pin (TMS) is sampled at the rise of TCK. TMS controls the internal status of the TAP controller. If no signal is input, TMS is fixed at 1 by internal pull-up.

18.2.3 Test Data Input (TDI)

The test data input pin (TDI) performs serial input of instructions and data to H-UDI registers. TDI is sampled at the rise of TCK. If no signal is input, TDI is fixed at 1 by internal pull-up.

18.2.4 Test Data Output (TDO)

The test data input pin (TDO) performs serial output of instructions and data from H-UDI registers. Transfer is synchronized with TCK. When no signal is being output, TDO goes to the high-impedance state.

18.2.5 Test Reset ($\overline{\text{TRST}}$)

The test reset pin ($\overline{\text{TRST}}$) is used to initialize the H-UDI asynchronously. If no signal is input, $\overline{\text{TRST}}$ is fixed at 1 by internal pull-up.

18.3.1 Instruction Register (SDIR)

Bit:	15	14	13	12	11	10	9	8
	TS3	TS2	TS1	TS0	—	—	—	—
Initial value:	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

The instruction register (SDIR) is a 16-bit register that can be read, but not written to, by the CPU. H-UDI instructions can be transferred to SDIR from TDI by serial input. SDIR can be initialized by the $\overline{\text{TRST}}$ signal, but is not initialized by a reset or in software standby mode.

Instructions transferred to SDIR must be 4 bits in length. If an instruction exceeding 4 bits is input, the last 4 bits of the serial data will be stored in SDIR.

Table 18.4 Instruction Configuration

TS3	TS2	TS1	TS0	Instruction	
0	0	0	0	Reserved	
			1	Reserved	
		1	0	Reserved	
			1	Reserved	
	1	0	0	Reserved	
			1	Reserved	
		1	0	Reserved	
			1	Reserved	
1	0	0	0	Reserved	
			1	Reserved	
		1	0	H-UDI interrupt	
			1	Reserved	
	1	0	0	Reserved	
			1	Reserved	
		1	0	Reserved	
			1	Bypass mode	(Initial value)

- Bits 11 to 0—Reserved: These bits always read 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SDTRF
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W

The status register (SDSR) is a 16-bit register that can be read and written to by the CPU. The SDSR value can be output from TDO, but serial data cannot be written to SDSR via TDI. The SDTRF bit is output by means of a one-bit shift. In a two-bit shift, the SDTRF bit is output first, followed by a reserved bit.

SDSR is initialized by $\overline{\text{TRST}}$ signal input, but is not initialized by a reset or in software standby mode.

- Bits 15 to 1—Reserved: Bits 15 to 10 and 8 to 1 always read 0, and the write value should always be 0. Bit 9 always reads 1, and the write value should always be 1.
- Bit 0—Serial Data Transfer Control Flag (SDTRF): Indicates whether H-UDI registers can be accessed by the CPU. The SDTRF bit is initialized by the $\overline{\text{TRST}}$ signal, but is not initialized by a reset or in software standby mode.

Bit 0: SDTRF	**Description**
0	Serial transfer to SDDR has ended, and SDDR can be accessed (Initial value)
1	Serial transfer to SDDR is in progress

The data register (SDDR) comprises data register H (SDDRH) and data register L (SDDRL), each of which has the following configuration.

Bit:	15	14	13	12	11	10	9	8
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SDDRH and SDDRL are 16-bit registers that can be read and written to by the CPU. SDDR is connected to TDO and TDI for serial data transfer to and from an external device.

32-bit data is input and output in serial data transfer. If data exceeding 32 bits is input, only the last 32 bits will be stored in SDDR. Serial data is input starting with the MSB of SDDR (bit 15 of SDDRH), and output starting with the LSB (bit 0 of SDDRL).

SDDR is not initialized by a reset, in hardware or software standby mode, or by the $\overline{\text{TRST}}$ signal.

18.3.4 Bypass Register (SDBPR)

The bypass register (SDBPR) is a one-bit shift register. In bypass mode, SDBPR is connected to TDI and TDO, and the SH7055SF chip is bypassed in a board test. SDBPR cannot be read or written to by the CPU.

18.4.1 H-UDI Interrupt

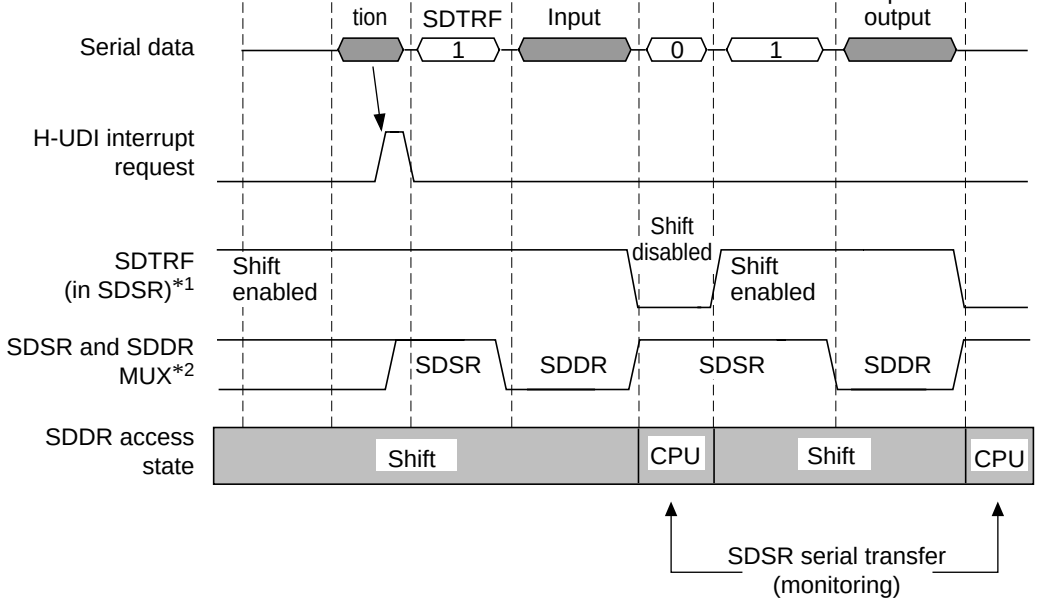
When an H-UDI interrupt instruction is transferred to SDIR via TDI, an interrupt is generated. Data transfer can be controlled by means of the H-UDI interrupt service routine. Transfer can be performed by means of SDDR.

Control of data input/output between an external device and the H-UDI is performed by monitoring the SDTRF bit in SDSR externally and internally. Internal SDTRF bit monitoring is carried out by having SDSR read by the CPU.

The H-UDI interrupt and serial transfer procedure is as follows.

1. An instruction is input to SDIR by serial transfer, and an H-UDI interrupt request is generated.
2. After the H-UDI interrupt request is issued, the SDTRF bit in SDSR is monitored externally. After output of SDTRF = 1 from TDO is observed, serial data is transferred to SDDR.
3. On completion of the serial transfer to SDDR, the SDTRF bit is cleared to 0, and SDDR can be accessed by the CPU. After SDDR has been accessed, SDDR serial transfer is enabled by setting the SDTRF bit to 1 in SDSR.
4. Serial data transfer between an external device and the H-UDI can be carried out by constantly monitoring the SDTRF bit in SDSR externally and internally.

Figures 18.2, 18.3, and 18.4 show the timing of data transfer between an external device and the H-UDI.



Notes: *1 SDTRF flag (in SDSR): Indicates whether SDDR access by the CPU or serial transfer data input/output to SDDR is possible.

1	SDDR is shift-enabled. Do not access SDDR until SDTRF = 0.
0	SDDR is shift-disabled. SDDR access by the CPU is enabled.

Conditions: •o SDTRF = 1
— When $\overline{\text{TRST}}$ = 0
— When the CPU writes 1
— In bypass mode
• SDTRF = 0
— End of SDDR shift access in serial transfer

*2 SDSR/SDDR (Update-DR state) internal MUX switchover timing
• Switchover from SDSR to SDDR: On completion of serial transfer in which SDTRF = 1 is output from TDO
• Switchover from SDDR to SDSR: On completion of serial transfer to SDDR

Figure 18.2 Data Input/Output Timing Chart (1)

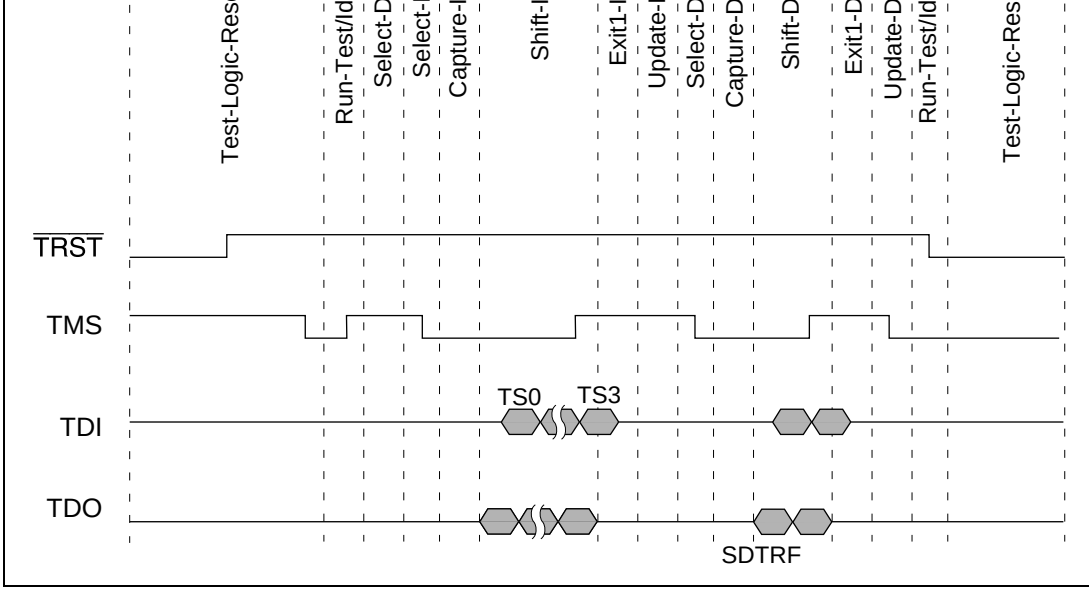


Figure 18.3 Data Input/Output Timing Chart (2)

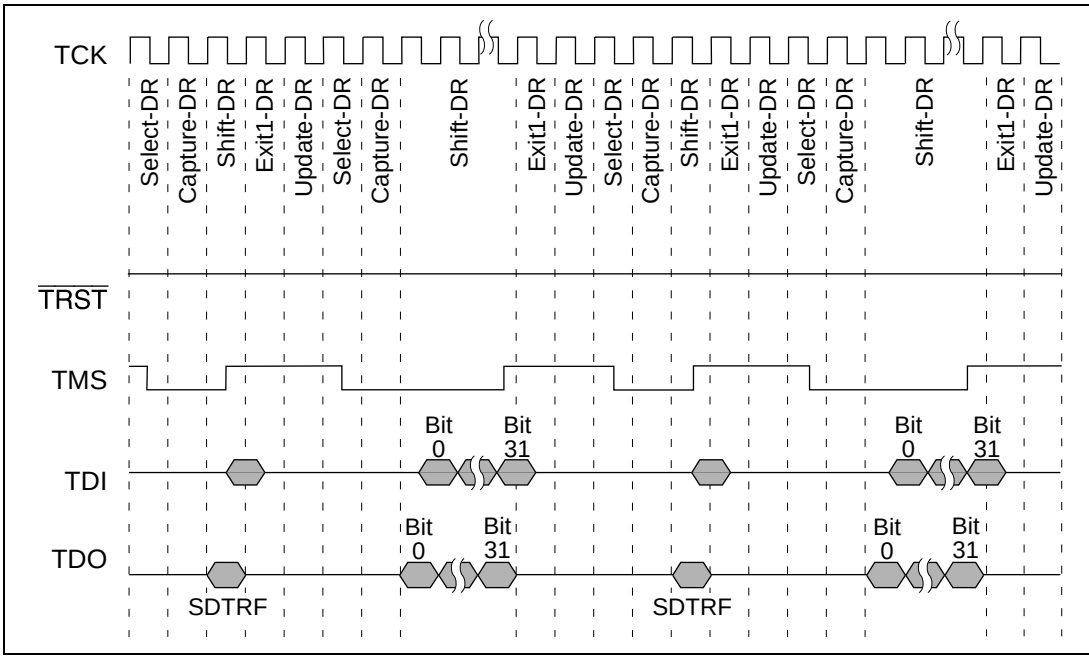


Figure 18.4 Data Input/Output Timing Chart (3)

Bypass mode can be used to bypass the SH7055F chip in a boundary scan test. Bypass mode is entered by transferring B'1111 to SDIR. In bypass mode, SDBPR is connected to TDI and TDO.

18.4.3 H-UDI Reset

The H-UDI can be reset as follows.

- By holding the $\overline{\text{TRST}}$ signal at 0
- When $\overline{\text{TRST}}$ = 1, by inputting at least five TCK clock cycles while TMS = 1
- By setting the MSTOP2 bit to 1 in the MSTCR register (see section 24.2.3)
- By entering hardware standby mode

18.5 Usage Notes

- A reset must always be executed by driving the $\overline{\text{TRST}}$ signal to 0, regardless of whether or not the H-UDI is to be activated. $\overline{\text{TRST}}$ must be held low for 20 TCK clock cycles. For details, see section 26, Electrical Characteristics.
- The registers are not initialized in software standby mode. If $\overline{\text{TRST}}$ is set to 0 in software standby mode, the correct operation is not guaranteed. Note that the operation is different from that of SH7055F.
- The frequency of TCK must be lower than that of the peripheral module clock (Pø). For details, see section 26, Electrical Characteristics.
- In data transfer, data input/output starts with the LSB. Figure 18.5 shows serial data input/output.
- If the H-UDI serial transfer sequence is disrupted, a $\overline{\text{TRST}}$ reset must be executed. Transfer should then be retried, regardless of the transfer operation.
- The TDO output timing is from the rise of TCK.
- In the Shift-IR state, the lower 2 bits of the output data from TDO (the IR status word) may not always be 01.
- If more than 32 bits are serially transferred, serial data exceeding 32 bits output from TDO should be ignored.

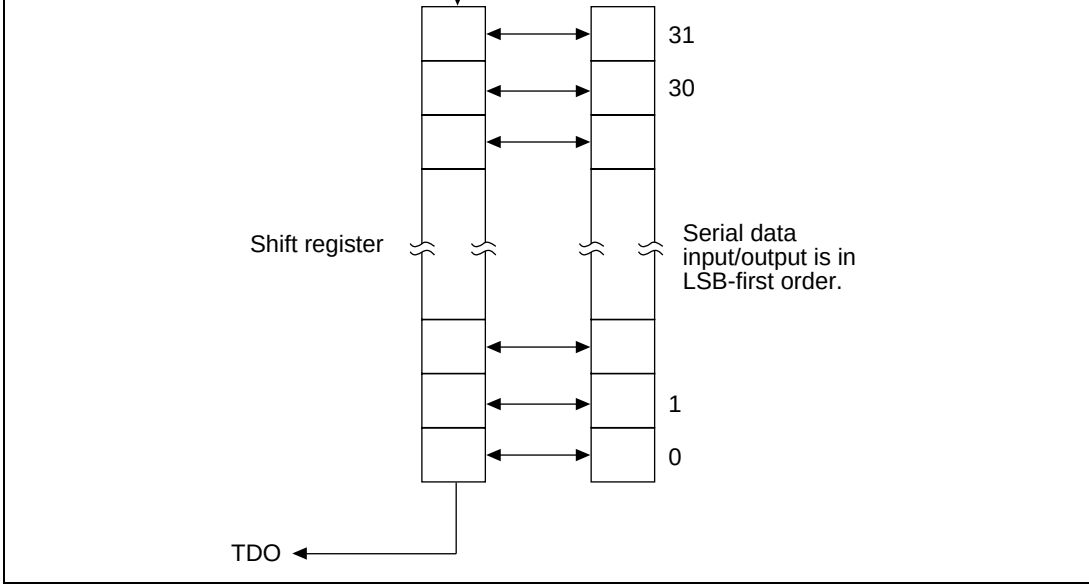


Figure 18.5 Serial Data Input/Output

19.1 Overview

The SH7055SF has an on-chip advanced user debugger (AUD). Use of the AUD simplifies the construction of a simple emulator, with functions such as acquisition of branch trace data and monitoring/tuning of on-chip RAM data.

19.1.1 Features

The AUD has the following features:

- Eight input/output pins
  - Data bus (AUDATA3–AUDATA0)
  - AUD reset ($\overline{\text{AUDRST}}$)
  - AUD sync signal ($\overline{\text{AUDSYNC}}$)
  - AUD clock (AUDCK)
  - AUD mode (AUDMD)
- Two modes

  Branch trace mode or RAM monitor mode can be selected by switching AUDMD.
  - Branch trace mode

    When the PC branches on execution of a branch instruction or generation of an interrupt in the user program , the branch is detected by the AUD and the branch destination address is output from AUDATA. The address is compared with the previously output address, and 4-, 8-, 16-, or 32-bit output is selected automatically according to the upper address matching status.
  - RAM monitor mode

    When an address is written to AUDATA from off-chip, the data corresponding to that address is output. If an address and data are written to AUDATA, the data is transferred to that address.

Figure 19.1 shows a block diagram of the AUD.

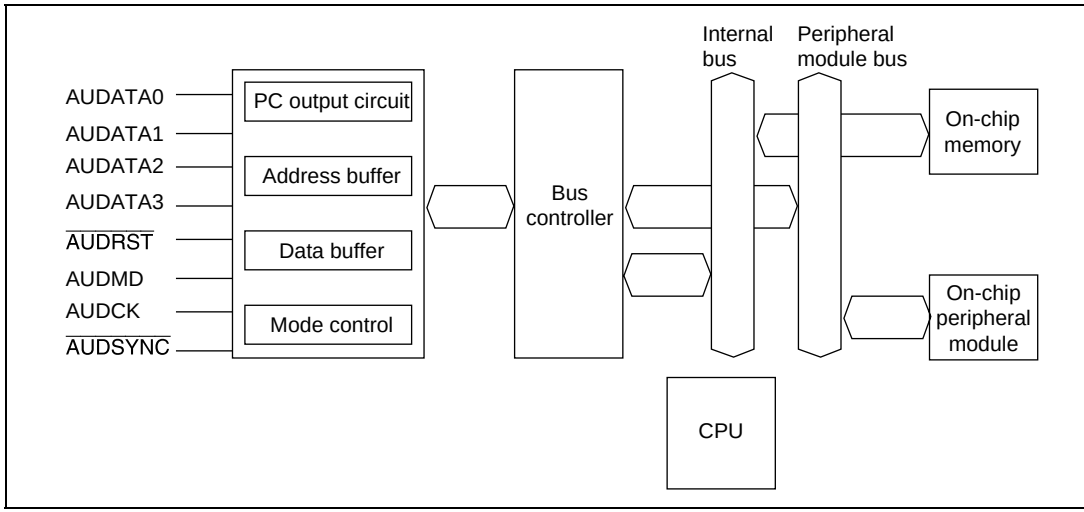


Figure 19.1 AUD Block Diagram

19.2 Pin Configuration

Table 19.1 shows the AUD's input/output pins.

Table 19.1 AUD Pins

		Function	
Name	**Abbreviation**	**Branch Trace Mode**	**RAM Monitor Mode**
AUD data	AUDATA3–AUDATA0	Branch destination address output	Monitor address/data input/output
AUD reset	$\overline{\text{AUDRST}}$	AUD reset input	AUD reset input
AUD mode	AUDMD	Mode select input (L)	Mode select input (H)
AUD clock	AUDCK	Serial clock (ϕ/2) output	Serial clock input
AUD sync signal	$\overline{\text{AUDSYNC}}$	Data start position identification signal output	Data start position identification signal input

Pins Used in Both Modes

Pin	Description
AUDMD	The mode is selected by changing the input level at this pin. Low: Branch trace mode High: RAM monitor mode The input at this pin should be changed when $\overline{\text{AUDRST}}$ is low. When no connection is made, this pin is pulled up internally.
$\overline{\text{AUDRST}}$	The AUDís internal buffers and logic are initialized by inputting a low level to this pin. When this signal goes low, the AUD enters the reset state and the AUDís internal buffers and logic are reset. When $\overline{\text{AUDRST}}$ goes high again after the AUDMD level settles, the AUD starts operating in the selected mode. When no connection is made, this pin is pulled down internally.

AUDCK	This pin outputs 1/2 the operating frequency (φ/2). This is the clock for AUDATA synchronization.
$\overline{\text{AUDSYNC}}$	This pin indicates whether output from AUDATA is valid. High: Valid data is not being output Low: An address is being output
AUDATA3 to AUDATA0	1. When $\overline{\text{AUDSYNC}}$ is low When a program branch or interrupt branch occurs, the AUD asserts $\overline{\text{AUDSYNC}}$ and outputs the branch destination address. The output order is A3–A0, A7–A4, A11–A8, A15–A12, A19–A16, A23–A20, A27–A24, A31–A28. 2. When $\overline{\text{AUDSYNC}}$ is high When waiting for branch destination address output, these pins constantly output 0011. When an branch occurs, AUDATA3–AUDATA2 output 10, and AUDATA1–AUDATA0 indicate whether a 4-, 8-, 16-, or 32-bit address is to be output by comparing the previous fully output address with the address output this time (see table below).

AUDATA1, AUDATA0	
00	Address bits A31–A4 match; 4 address bits A3–A0 are to be output (i.e. output is performed once).
01	Address bits A31–A8 match; 8 address bits A3–A0 and A7–A4 are to be output (i.e. output is performed twice).
10	Address bits A31–A16 match; 16 address bits A3–A0, A7–A4, A11–A8, and A15–A12 are to be output (i.e. output is performed four times).
11	None of the above cases applies; 31 address bits A3–A0, A7–A4, A11–A8, and A15–A12, A19–A16, A23–A20, A27–A24, and A31–A28 are to be output (i.e. output is performed eight times).

AUDCK	The external clock input pin. Input the clock to be used for debugging to this pin. The input frequency must not exceed 1/4 the operating frequency. When no connection is made, this pin is pulled up internally.
$\overline{\text{AUDSYNC}}$	Do not assert this pin until a command is input to AUDATA from off-chip and the necessary data can be prepared. See the protocol description for details. When no connection is made, this pin is pulled up internally.
AUDATA3 to AUDATA0	When a command is input from off-chip, data is output after Ready reception. Output starts when $\overline{\text{AUDSYNC}}$ is negated. See the protocol description for details. When no connections are made, these pins are pulled up internally.

19.3 Branch Trace Mode

19.3.1 Overview

In this mode, the branch destination address is output when a branch occurs in the user program. Branches may be caused by branch instruction execution or interrupt/exception processing, but no distinction is made between the two in this mode.

19.3.2 Operation

Operation starts in branch trace mode when $\overline{\text{AUDRST}}$ is asserted, AUDMD is driven low, then $\overline{\text{AUDRST}}$ is negated.

Figure 19.2 shows an example of data output.

While the user program is being executed without branches, the AUDATA pins constantly output 0011 in synchronization with AUDCK.

When a branch occurs, after execution starts at the branch destination address in the PC, the previous fully output address (i.e. for which output was not interrupted by the occurrence of another branch) is compared with the current branch address, and depending on the result, $\overline{\text{AUDSYNC}}$ is asserted and the branch destination address output after 1-clock output of 1000 (in the case of 4-bit output), 1001 (8-bit output), 1010 (16-bit output), or 1011 (32-bit output). The initial value of the compared address is H'00000000.

On completion of the cycle in which the address is output, $\overline{\text{AUDSYNC}}$ is negated and 0011 is output from the AUDATA pins.

outputting 10xx again (figure 19.3 shows an example of the output when consecutive branches occur). Note that the compared address is the previous fully output address, and not an interrupted address (since the upper address of an interrupted address will be unknown).

The interval from the start of execution at the branch destination address in the PC until the AUDATA pins output 10xx is 1.5 or 2 AUDCK cycles.

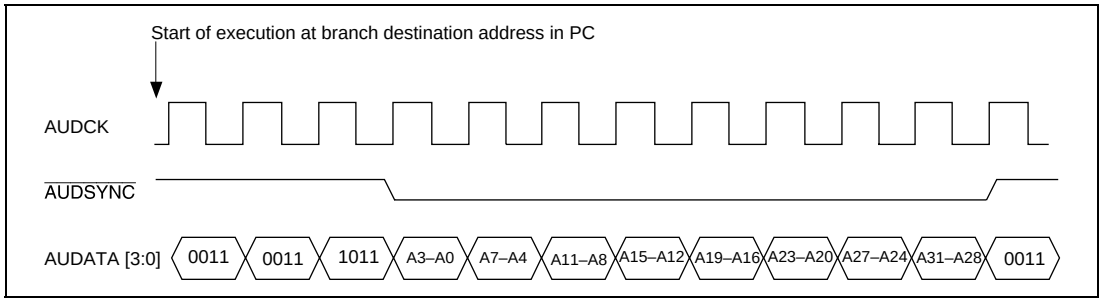


Figure 19.2 Example of Data Output (32-Bit Output)

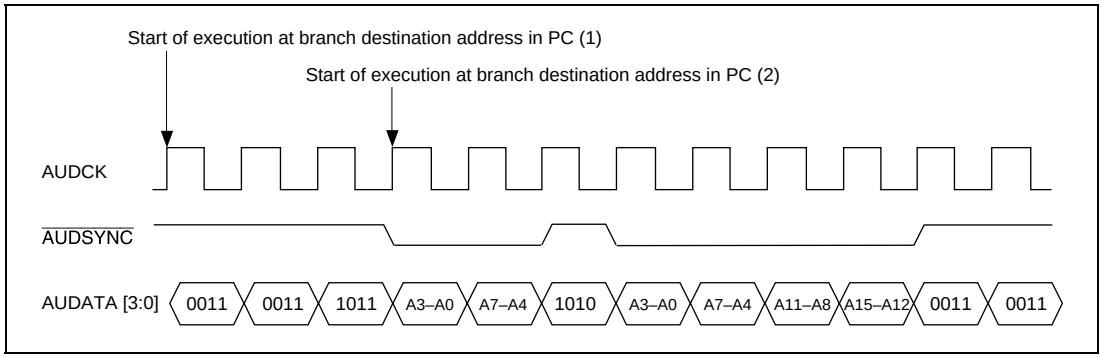


Figure 19.3 Example of Output in Case of Successive Branches

19.4.1 Overview

In this mode, all the modules connected to the SH7055SF's internal or external bus can be read and written to, allowing RAM monitoring and tuning to be carried out.

19.4.2 Communication Protocol

The AUD latches the AUDATA input when $\overline{\text{AUDSYNC}}$ is asserted. The following AUDATA input format should be used.

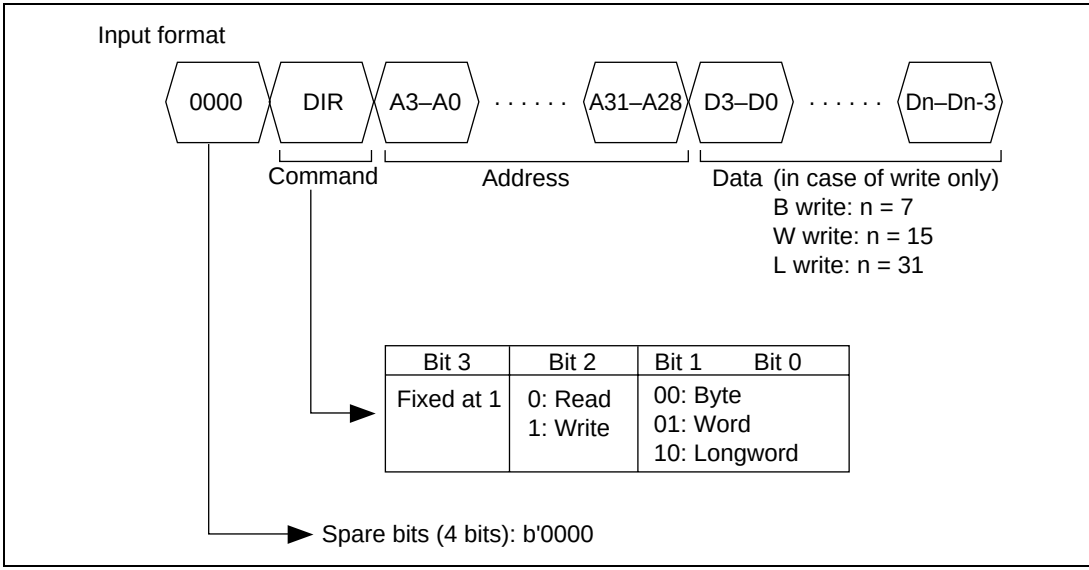


Figure 19.4 AUDATA Input Format

Operation starts in RAM monitor mode AUDMD is driven high after AUDRST has been asserted, then $\overline{\text{AUDRST}}$ is negated.

Figure 19.5 shows an example of a read operation, and figure 19.6 an example of a write operation.

When $\overline{\text{AUDSYNC}}$ is asserted, input from the AUDATA pins begins. When a command, address, or data (writing only) is input in the format shown in figure 19.2, execution of read/write access to the specified address is started. During internal execution, the AUD returns Not Ready (0000). When execution is completed, the Ready flag (0001) is returned (figures 19.5 and 19.6). Table 19.2 shows the Ready flag format.

In a read, data of the specified size is output when $\overline{\text{AUDSYNC}}$ is negated following detection of this flag (figure 19.7).

If a command other than the above is input in DIR, the AUD treats this as a command error, disables processing, and sets bit 1 in the Ready flag to 1. If a read/write operation initiated by the command specified in DIR causes a bus error, the AUD disables processing and sets bit 2 in the Ready flag to 1 (figure 19.7).

Table 19.2 Ready Flag Format

Bit 3	Bit 2	Bit 1	Bit 0
Fixed at 0	0: Normal status	0: Normal status	0: Not ready
	1: Bus error	1: Bus error	1: Ready

Bus error conditions are shown below.

1. Word access to address 4n+1 or 4n+3
2. Longword access to address 4n+1, 4n+2, or 4n+3
3. Longword access to on-chip I/O 8-bit space
4. Access to external space in single-chip mode

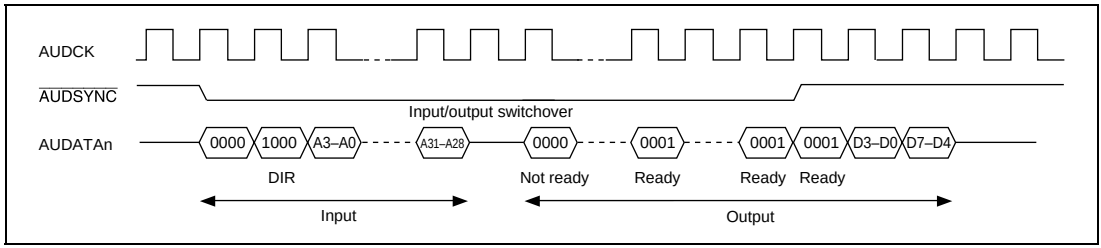


Figure 19.5 Example of Read Operation (Byte Read)

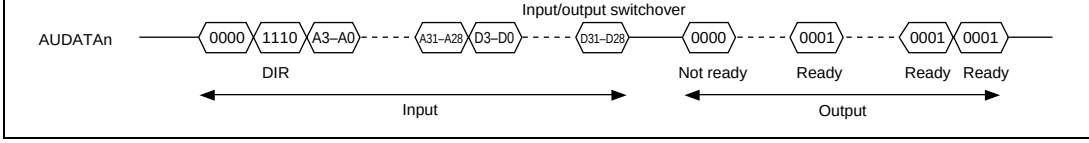


Figure 19.6 Example of Write Operation (Longword Write)

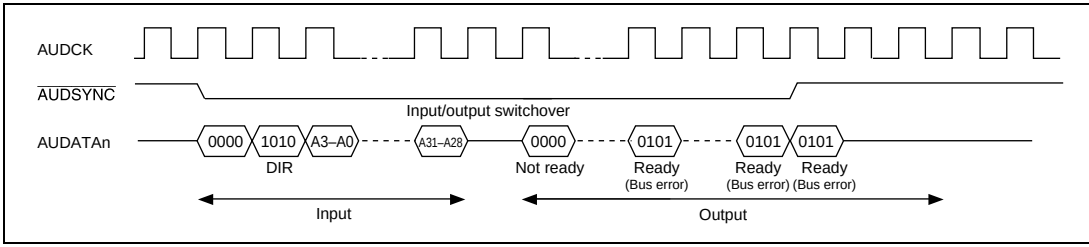


Figure 19.7 Example of Error Occurrence (Longword Read)

19.5 Usage Notes

19.5.1 Initialization

The debugger's internal buffers and processing states are initialized in the following cases:

1. In a power-on reset
2. In hardware standby mode
3. When $\overline{\text{AUDRST}}$ is driven low
4. When the AUDSRST bit is set to 1 in the SYSCR register (see section 24.2.2)
5. When the MSTOP3 bit is set to 1 in the MSTCR register (see section 24.2.3)

19.5.2 Operation in Software Standby Mode

The debugger is not initialized in software standby mode. However, since the SH7055SF's internal operation halts in software standby mode:

1. When AUDMD is high (RAM monitor mode): Ready is not returned. Since the operation after the software standby mode cancellation is not guaranteed, input $\overline{\text{AUDRST}}$, and re-execute. However, when operating on an external clock, the protocol continues.
2. When AUDMD is low (PC trace): Operation stops. However, operation continues when software standby is released.

The AUD operation cannot be provided in the boot mode operation and user boot mode initial states. For details on the boot mode and user boot mode, see section 22, ROM.

19.5.4 AUD Input Signal in Software Standby/Hardware Standby Mode

If the AUD interface input is changed during software standby/hardware standby mode operation, the reliability may be lowered significantly. Be careful not to change the AUD input signal during software standby/hardware standby mode operation.

20.1 Overview

The pin function controller (PFC) consists of registers for selecting multiplex pin functions and their input/output direction. Table 20.1 shows the SH7055SF's multiplex pins.

Table 20.1 SH7055SF Multiplex Pins

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
A	PA0 input/output (port)	TI0A input (ATU-II)		
A	PA1 input/output (port)	TI0B input (ATU-II)		
A	PA2 input/output (port)	TI0C input (ATU-II)		
A	PA3 input/output (port)	TI0D input (ATU-II)		
A	PA4 input/output (port)	TIO3A input/output (ATU-II)		
A	PA5 input/output (port)	TIO3B input/output (ATU-II)		
A	PA6 input/output (port)	TIO3C input/output (ATU-II)		
A	PA7 input/output (port)	TIO3D input/output (ATU-II)		
A	PA8 input/output (port)	TIO4A input/output (ATU-II)		
A	PA9 input/output (port)	TIO4B input/output (ATU-II)		
A	PA10 input/output (port)	TIO4C input/output (ATU-II)		
A	PA11 input/output (port)	TIO4D input/output (ATU-II)		
A	PA12 input/output (port)	TIO5A input/output (ATU-II)		
A	PA13 input/output (port)	TIO5B input/output (ATU-II)		
A	PA14 input/output (port)	TxD0 output (SCI)		
A	PA15 input/output (port)	RxD0 input (SCI)		
B	PB0 input/output (port)	TO6A output (ATU-II)		
B	PB1 input/output (port)	TO6B output (ATU-II)		
B	PB2 input/output (port)	TO6C output (ATU-II)		
B	PB3 input/output (port)	TO6D output (ATU-II)		
B	PB4 input/output (port)	TO7A output (ATU-II)	TO8A output (ATU-II)	
B	PB5 input/output (port)	TO7B output (ATU-II)	TO8B output (ATU-II)	
B	PB6 input/output (port)	TO7C output (ATU-II)	TO8C output (ATU-II)	
B	PB7 input/output (port)	TO7D output (ATU-II)	TO8D output (ATU-II)	
B	PB8 input/output (port)	TxD3 output (SCI)	TO8E output (ATU-II)	
B	PB9 input/output (port)	RxD3 input (SCI)	TO8F output (ATU-II)	

Port	(Related Module)	(Related Module)	(Related Module)	(Related Module)
B	PB10 input/output (port)	TxD4 output (SCI)	HTxD0 output (HCAN)	TO8G output (ATU-II)
B	PB11 input/output (port)	RxD4 input (SCI)	HRxD0 input (HCAN)	TO8H output (ATU-II)
B	PB12 input/output (port)	TCLKA input (ATU-II)	$\overline{\text{UBCTRG}}$ output (UBC)	
B	PB13 input/output (port)	SCK0 input/output (SCI)		
B	PB14 input/output (port)	SCK1 input/output (SCI)	TCLKB input (ATU-II)	TI10 input (ATU-II)
B	PB15 input/output (port)	PULS5 output (APC)	SCK2 input/output (SCI)	
C	PC0 input/output (port)	TxD1 output (SCI)		
C	PC1 input/output (port)	RxD1 input (SCI)		
C	PC2 input/output (port)	TxD2 output (SCI)		
C	PC3 input/output (port)	RxD2 input (SCI)		
C	PC4 input/output (port)	$\overline{\text{IRQ0}}$ input (INTC)		
D	PD0 input/output (port)	TIO1A input/output (ATU-II)		
D	PD1 input/output (port)	TIO1B input/output (ATU-II)		
D	PD2 input/output (port)	TIO1C input/output (ATU-II)		
D	PD3 input/output (port)	TIO1D input/output (ATU-II)		
D	PD4 input/output (port)	TIO1E input/output (ATU-II)		
D	PD5 input/output (port)	TIO1F input/output (ATU-II)		
D	PD6 input/output (port)	TIO1G input/output (ATU-II)		
D	PD7 input/output (port)	TIO1H input/output (ATU-II)		
D	PD8 input/output (port)	PULS0 output (APC)		
D	PD9 input/output (port)	PULS1 output (APC)		
D	PD10 input/output (port)	PULS2 output (APC)		
D	PD11 input/output (port)	PULS3 output (APC)		
D	PD12 input/output (port)	PULS4 output (APC)		
D	PD13 input/output (port)	PULS6 output (APC)	HTxD0 output (HCAN)	HTxD1 output (HCAN)
E	PE0 input/output (port)	A0 output (BSC)		
E	PE1 input/output (port)	A1 output (BSC)		
E	PE2 input/output (port)	A2 output (BSC)		
E	PE3 input/output (port)	A3 output (BSC)		
E	PE4 input/output (port)	A4 output (BSC)		
E	PE5 input/output (port)	A5 output (BSC)		
E	PE6 input/output (port)	A6 output (BSC)		
E	PE7 input/output (port)	A7 output (BSC)		

Port	(Related Module)	(Related Module)	(Related Module)	(Related Module)
E	PE8 input/output (port)	A8 output (BSC)		
E	PE9 input/output (port)	A9 output (BSC)		
E	PE10 input/output (port)	A10 output (BSC)		
E	PE11 input/output (port)	A11 output (BSC)		
E	PE12 input/output (port)	A12 output (BSC)		
E	PE13 input/output (port)	A13 output (BSC)		
E	PE14 input/output (port)	A14 output (BSC)		
E	PE15 input/output (port)	A15 output (BSC)		
F	PF0 input/output (port)	A16 output (BSC)		
F	PF1 input/output (port)	A17 output (BSC)		
F	PF2 input/output (port)	A18 output (BSC)		
F	PF3 input/output (port)	A19 output (BSC)		
F	PF4 input/output (port)	A20 output (BSC)		
F	PF5 input/output (port)	A21 output (BSC)	$\overline{\text{POD}}$ input (port)	
F	PF6 input/output (port)	$\overline{\text{WRL}}$ output (BSC)		
F	PF7 input/output (port)	$\overline{\text{WRH}}$ output (BSC)		
F	PF8 input/output (port)	$\overline{\text{WAIT}}$ input (BSC)		
F	PF9 input/output (port)	$\overline{\text{RD}}$ output (BSC)		
F	PF10 input/output (port)	$\overline{\text{CS0}}$ output (BSC)		
F	PF11 input/output (port)	$\overline{\text{CS1}}$ output (BSC)		
F	PF12 input/output (port)	$\overline{\text{CS2}}$ output (BSC)		
F	PF13 input/output (port)	$\overline{\text{CS3}}$ output (BSC)		
F	PF14 input/output (port)	$\overline{\text{BACK}}$ output (BSC)		
F	PF15 input/output (port)	$\overline{\text{BREQ}}$ input (BSC)		
G	PG0 input/output (port)	PULS7 output (APC)	HRxD0 input (HCAN)	HRxD1 input (HCAN)
G	PG1 input/output (port)	$\overline{\text{IRQ1}}$ input (INTC)		
G	PG2 input/output (port)	$\overline{\text{IRQ2}}$ input (INTC)	ADEND output (A/D)	
G	PG3 input/output (port)	$\overline{\text{IRQ3}}$ input (INTC)	$\overline{\text{ADTRG0}}$ input (A/D)	
H	PH0 input/output (port)	D0 input/output (BSC)		
H	PH1 input/output (port)	D1 input/output (BSC)		
H	PH2 input/output (port)	D2 input/output (BSC)		
H	PH3 input/output (port)	D3 input/output (BSC)		
H	PH4 input/output (port)	D4 input/output (BSC)		

Port	(Related Module)	(Related Module)	(Related Module)	(Related Module)
H	PH5 input/output (port)	D5 input/output (BSC)		
H	PH6 input/output (port)	D6 input/output (BSC)		
H	PH7 input/output (port)	D7 input/output (BSC)		
H	PH8 input/output (port)	D8 input/output (BSC)		
H	PH9 input/output (port)	D9 input/output (BSC)		
H	PH10 input/output (port)	D10 input/output (BSC)		
H	PH11 input/output (port)	D11 input/output (BSC)		
H	PH12 input/output (port)	D12 input/output (BSC)		
H	PH13 input/output (port)	D13 input/output (BSC)		
H	PH14 input/output (port)	D14 input/output (BSC)		
H	PH15 input/output (port)	D15 input/output (BSC)		
J	PJ0 input/output (port)	TIO2A input/output (ATU-II)		
J	PJ1 input/output (port)	TIO2B input/output (ATU-II)		
J	PJ2 input/output (port)	TIO2C input/output (ATU-II)		
J	PJ3 input/output (port)	TIO2D input/output (ATU-II)		
J	PJ4 input/output (port)	TIO2E input/output (ATU-II)		
J	PJ5 input/output (port)	TIO2F input/output (ATU-II)		
J	PJ6 input/output (port)	TIO2G input/output (ATU-II)		
J	PJ7 input/output (port)	TIO2H input/output (ATU-II)		
J	PJ8 input/output (port)	TIO5C input/output (ATU-II)		
J	PJ9 input/output (port)	TIO5D input/output (ATU-II)		
J	PJ10 input/output (port)	TI9A input (ATU-II)		
J	PJ11 input/output (port)	TI9B input (ATU-II)		
J	PJ12 input/output (port)	TI9C input (ATU-II)		
J	PJ13 input/output (port)	TI9D input (ATU-II)		
J	PJ14 input/output (port)	TI9E input (ATU-II)		
J	PJ15 input/output (port)	TI9F input (ATU-II)		
K	PK0 input/output (port)	TO8A output (ATU-II)		
K	PK1 input/output (port)	TO8B output (ATU-II)		
K	PK2 input/output (port)	TO8C output (ATU-II)		
K	PK3 input/output (port)	TO8D output (ATU-II)		
K	PK4 input/output (port)	TO8E output (ATU-II)		
K	PK5 input/output (port)	TO8F output (ATU-II)		

Port	(Related Module)	(Related Module)	(Related Module)	(Related Module)
K	PK6 input/output (port)	TO8G output (ATU-II)		
K	PK7 input/output (port)	TO8H output (ATU-II)		
K	PK8 input/output (port)	TO8I output (ATU-II)		
K	PK9 input/output (port)	TO8J output (ATU-II)		
K	PK10 input/output (port)	TO8K output (ATU-II)		
K	PK11 input/output (port)	TO8L output (ATU-II)		
K	PK12 input/output (port)	TO8M output (ATU-II)		
K	PK13 input/output (port)	TO8N output (ATU-II)		
K	PK14 input/output (port)	TO8O output (ATU-II)		
K	PK15 input/output (port)	TO8P output (ATU-II)		
L	PL0 input/output (port)	TI10 input (ATU-II)		
L	PL1 input/output (port)	TIO11A input/output (ATU-II)	$\overline{\text{IRQ6}}$ input (INTC)	
L	PL2 input/output (port)	TIO11B input/output (ATU-II)	$\overline{\text{IRQ7}}$ input (INTC)	
L	PL3 input/output (port)	TCLKB input (ATU-II)		
L	PL4 input/output (port)	$\overline{\text{ADTRG0}}$ input (A/D)		
L	PL5 input/output (port)	$\overline{\text{ADTRG1}}$ input (A/D)		
L	PL6 input/output (port)	ADEND output (A/D)		
L	PL7 input/output (port)	SCK2 input/output (SCI)		
L	PL8 input/output (port)	SCK3 input/output (SCI)		
L	PL9 input/output (port)	SCK4 input/output (SCI)	$\overline{\text{IRQ5}}$ input (INTC)	
L	PL10 input/output (port)	HTxD0 output (HCAN)	HTxD1 output (HCAN)	HTxD0 & HTxD1 (HCAN)
L	PL11 input/output (port)	HRxD0 input (HCAN)	HRxD1 input (HCAN)	HRxD0 & HRxD1 (HCAN)
L	PL12 input/output (port)	$\overline{\text{IRQ4}}$ input (INTC)		
L	PL13 input/output (port)	$\overline{\text{IRQOUT}}$ output (INTC)	$\overline{\text{IRQOUT}}$ output (INTC)	

PFC registers are listed in table 20.2.

Table 20.2 PFC Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A IO register	PAIOR	R/W	H'0000	H'FFFFF720	8, 16
Port A control register H	PACRH	R/W	H'0000	H'FFFFF722	8, 16
Port A control register L	PACRL	R/W	H'0000	H'FFFFF724	8, 16
Port B IO register	PBIOR	R/W	H'0000	H'FFFFF730	8, 16
Port B control register H	PBCRH	R/W	H'0000	H'FFFFF732	8, 16
Port B control register L	PBCRL	R/W	H'0000	H'FFFFF734	8, 16
Port B invert register	PBIR	R/W	H'0000	H'FFFFF736	8, 16
Port C IO register	PCIOR	R/W	H'0000	H'FFFFF73A	8, 16
Port C control register	PCCR	R/W	H'0000	H'FFFFF73C	8, 16
Port D IO register	PDIOR	R/W	H'0000	H'FFFFF740	8, 16
Port D control register H	PDCRH	R/W	H'0000	H'FFFFF742	8, 16
Port D control register L	PDCRL	R/W	H'0000	H'FFFFF744	8, 16
Port E IO register	PEIOR	R/W	H'0000	H'FFFFF750	8, 16
Port E control register	PECR	R/W	H'0000	H'FFFFF752	8, 16
Port F IO register	PFIOR	R/W	H'0000	H'FFFFF748	8, 16
Port F control register H	PFCRH	R/W	H'0015	H'FFFFF74A	8, 16
Port F control register L	PFCRL	R/W	H'5000	H'FFFFF74C	8, 16
Port G IO register	PGIOR	R/W	H'0000	H'FFFFF760	8, 16
Port G control register	PGCR	R/W	H'0000	H'FFFFF762	8, 16
Port H IO register	PHIOR	R/W	H'0000	H'FFFFF728	8, 16
Port H control register	PHCR	R/W	H'0000	H'FFFFF72A	8, 16
Port J IO register	PJIOR	R/W	H'0000	H'FFFFF766	8, 16
Port J control register H	PJCRH	R/W	H'0000	H'FFFFF768	8, 16
Port J control register L	PJCRL	R/W	H'0000	H'FFFFF76A	8, 16
Port K IO register	PKIOR	R/W	H'0000	H'FFFFF770	8, 16
Port K control register H	PKCRH	R/W	H'0000	H'FFFFF772	8, 16
Port K control register L	PKCRL	R/W	H'0000	H'FFFFF774	8, 16
Port K invert register	PKIR	R/W	H'0000	H'FFFFF776	8, 16

Name	Abbreviation	R/W	Value	Address	Size
Port L IO register	PLIOR	R/W	H'0000	H'FFFFF756	8, 16
Port L control register H	PLCRH	R/W	H'0000	H'FFFFF758	8, 16
Port L control register L	PLCRL	R/W	H'0000	H'FFFFF75A	8, 16
Port L invert register	PLIR	R/W	H'0000	H'FFFFF75C	8, 16

20.3 Register Descriptions

20.3.1 Port A IO Register (PAIOR)

Bit:	15	14	13	12	11	10	9	8
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port A IO register (PAIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port A. Bits PA15IOR to PA0IOR correspond to pins PA15/RxD0 to PA0/TI0A. PAIOR is enabled when port A pins function as general input/output pins (PA15 to PA0) or ATU-II input/output pins, and disabled otherwise. For bits 3 to 0, when ATU-II input capture input is selected, the PAIOR bits should be cleared to 0.

When port A pins function as PA15 to PA0 or ATU-II input/output pins, a pin becomes an output when the corresponding bit in PAIOR is set to 1, and an input when the bit is cleared to 0.

PAIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Port A control registers H and L (PACRH, PACRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port A. PACRH selects the functions of the pins for the upper 8 bits of port A, and PACRL selects the functions of the pins for the lower 8 bits.

PACRH and PACRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

Port A Control Register H (PACRH)

Bit:	15	14	13	12	11	10	9	8
	—	PA15MD	—	PA14MD	—	PA13MD	—	PA12MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PA11MD	—	PA10MD	—	PA9MD	—	PA8MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PA15 Mode Bit (PA15MD): Selects the function of pin PA15/RxD0.

Bit 14: PA15MD	Description	
0	General input/output (PA15)	(Initial value)
1	Receive data input (RxD0)	

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PA14 Mode Bit (PA14MD): Selects the function of pin PA14/TxD0.

Bit 12: PA14MD	Description	
0	General input/output (PA14)	(Initial value)
1	Transmit data output (TxD0)	

Bit 10: PA13MD	Description	
0	General input/output (PA13)	(Initial value)
1	ATU-II input capture input/output compare output (TIO5B)	

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PA12 Mode Bit (PA12MD): Selects the function of pin PA12/TIO5A.

Bit 8: PA12MD	Description	
0	General input/output (PA12)	(Initial value)
1	ATU-II input capture input/output compare output (TIO5A)	

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PA11 Mode Bit (PA11MD): Selects the function of pin PA11/TIO4D.

Bit 6: PA11MD	Description	
0	General input/output (PA11)	(Initial value)
1	ATU-II input capture input/output compare output (TIO4D)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PA10 Mode Bit (PA10MD): Selects the function of pin PA10/TIO4C.

Bit 4: PA10MD	Description	
0	General input/output (PA10)	(Initial value)
1	ATU-II input capture input/output compare output (TIO4C)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PA9 Mode Bit (PA9MD): Selects the function of pin PA9/TIO4B.

Bit 2: PA9MD	Description	
0	General input/output (PA9)	(Initial value)
1	ATU-II input capture input/output compare output (TIO4B)	

Bit 0: PA8MD	Description	
0	General input/output (PA8)	(Initial value)
1	ATU-II input capture input/output compare output (TIO4A)	

Port A Control Register L (PACRL)

Bit:	15	14	13	12	11	10	9	8
	—	PA7MD	—	PA6MD	—	PA5MD	—	PA4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PA3MD	—	PA2MD	—	PA1MD	—	PA0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PA7 Mode Bit (PA7MD): Selects the function of pin PA7/TIO3D.

Bit 14: PA7MD	Description	
0	General input/output (PA7)	(Initial value)
1	ATU-II input capture input/output compare output (TIO3D)	

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PA6 Mode Bit (PA6MD): Selects the function of pin PA6/TIO3C.

Bit 12: PA6MD	Description	
0	General input/output (PA6)	(Initial value)
1	ATU-II input capture input/output compare output (TIO3C)	

Bit 10: PA5MD	Description	
0	General input/output (PA5)	(Initial value)
1	ATU-II input capture input/output compare output (TIO3B)	

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PA4 Mode Bit (PA4MD): Selects the function of pin PA4/TIO3A.

Bit 8: PA4MD	Description	
0	General input/output (PA4)	(Initial value)
1	ATU-II input capture input/output compare output (TIO3A)	

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0. If 1 is written to this bit, correct operation cannot be guaranteed.
- Bit 6—PA3 Mode Bit (PA3MD): Selects the function of pin PA3/TI0D.

Bit 6: PA3MD	Description	
0	General input/output (PA3)	(Initial value)
1	ATU-II input capture input (TI0D)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0. If 1 is written to this bit, correct operation cannot be guaranteed.
- Bit 4—PA2 Mode Bit (PA2MD): Selects the function of pin PA2/TI0C.

Bit 4: PA2MD	Description	
0	General input/output (PA2)	(Initial value)
1	ATU-II input capture input (TI0C)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0. If 1 is written to this bit, correct operation cannot be guaranteed.
- Bit 2—PA1 Mode Bit (PA1MD): Selects the function of pin PA1/TI0B.

Bit 2: PA1MD	Description	
0	General input/output (PA1)	(Initial value)
1	ATU-II input capture input (TI0B)	

- Bit 0—PA0 Mode Bit (PA0MD): Selects the function of pin PA0/TI0A.

Bit 0: PA0MD	Description	
0	General input/output (PA0)	(Initial value)
1	ATU-II input capture input (TI0A)	

20.3.3 Port B IO Register (PBIOR)

Bit:	15	14	13	12	11	10	9	8
	PB15 IOR	PB14 IOR	PB13 IOR	PB12 IOR	PB11 IOR	PB10 IOR	PB9 IOR	PB8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port B IO register (PBIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port B. Bits PB15IOR to PB0IOR correspond to pins PB15/PULS5/SCK2 to PB0/TO6A. PBIOR is enabled when port B pins function as general input/output pins (PB15 to PB0) or serial clock pins (SCK0, SCK1, SCK2), and disabled otherwise.

When port B pins function as PB15 to PB0 or SCK0, SCK1, and SCK2, a pin becomes an output when the corresponding bit in PBIOR is set to 1, and an input when the bit is cleared to 0.

PBIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Port B control registers H and L (PBCRH, PBCRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port B. PBCRH selects the functions of the pins for the upper 8 bits of port B, and PBCRL selects the functions of the pins for the lower 8 bits.

PBCRH and PBCRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

Port B Control Register H (PBCRH)

Bit:	15	14	13	12	11	10	9	8
	PB15 MD1	PB15 MD0	PB14 MD1	PB14 MD0	—	PB13 MD	PB12 MD1	PB12 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PB11 MD1	PB11 MD0	PB10 MD1	PB10 MD0	PB9 MD1	PB9 MD0	PB8 MD1	PB8 MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bits 15 and 14—PB15 Mode Bits 1 and 0 (PB15MD1, PB15MD0): These bits select the function of pin PB15/PULS5/SCK2.

Bit 15: PB15MD1	Bit 14: PB15MD0	Description	
0	0	General input/output (PB15)	(Initial value)
	1	APC pulse output (PULS5)	
1	0	Serial clock input/output (SCK2)	
	1	Reserved (Do not set)	

Bit 13: PB14MD1	Bit 12: PB14MD0	Description	
0	0	General input/output (PB14)	(Initial value)
	1	Serial clock input/output (SCK1)	
1	0	ATU-II clock input (TCLKB)	
	1	ATU-II edge input (TI10)	

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PB13 Mode Bit (PB13MD): Selects the function of pin PB13/SCK0.

Bit 10: PB13MD	Description	
0	General input/output (PB13)	(Initial value)
1	Serial clock input/output (SCK0)	

- Bits 9 and 8—PB12 Mode Bits 1 and 0 (PB12MD1, PB12MD0): These bits select the function of pin PB12/TCLKA/$\overline{\text{UBCTRG}}$.

Bit 9: PB12MD1	Bit 8: PB12MD0	Description	
0	0	General input/output (PB12)	(Initial value)
	1	ATU-II clock input (TCLKA)	
1	0	Trigger pulse output ($\overline{\text{UBCTRG}}$)	
	1	Reserved (Do not set)	

- Bits 7 and 6—PB11 Mode Bits 1 and 0 (PB11MD1, PB11MD0): These bits select the function of pin PB11/RxD4/HRxD0/TO8H.

Bit 7: PB11MD1	Bit 6: PB11MD0	Description	
0	0	General input/output (PB11)	(Initial value)
	1	Receive data input (RxD4)	
1	0	HCAN receive data input (HRxD0)	
	1	ATU-II one-shot pulse output (TO8H)	

Bit 5: PB10MD1	Bit 4: PB10MD0	Description	
0	0	General input/output (PB10)	(Initial value)
	1	Transmit data output (TxD4)	
1	0	HCAN transmit data output (HTxD0)	
	1	ATU-II one-shot pulse output (TO8G)	

- Bits 3 and 2—PB9 Mode Bits 1 and 0 (PB9MD1, PB9MD0): These bits select the function of pin PB9/RxD3/TO8F.

Bit 3: PB9MD1	Bit 2: PB9MD0	Description	
0	0	General input/output (PB9)	(Initial value)
	1	Receive data input (RxD3)	
1	0	ATU-II one-shot pulse output (TO8F)	
	1	Reserved (Do not set)	

- Bits 1 and 0—PB8 Mode Bits 1 and 0 (PB8MD1, PB8MD0): These bits select the function of pin PB8/TxD3/TO8E.

Bit 1: PB8MD1	Bit 0: PB8MD0	Description	
0	0	General input/output (PB8)	(Initial value)
	1	Transmit data output (TxD3)	
1	0	ATU-II one-shot pulse output (TO8E)	
	1	Reserved (Do not set)	

Port B Control Register L (PBCRL)

Bit:	15	14	13	12	11	10	9	8
	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PB3MD	—	PB2MD	—	PB1MD	—	PB0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit 15: PB7MD1	Bit 14: PB7MD0	Description	
0	0	General input/output (PB7)	(Initial value)
	1	ATU-II PWM output (TO7D)	
1	0	ATU-II one-shot pulse output (TO8D)	
	1	Reserved (Do not set)	

- Bits 13 and 12—PB6 Mode Bits 1 and 0 (PB6MD1, PB6MD0): These bits select the function of pin PB6/TO7C/TO8C.

Bit 13: PB6MD1	Bit 12: PB6MD0	Description	
0	0	General input/output (PB6)	(Initial value)
	1	ATU-II PWM output (TO7C)	
1	0	ATU-II one-shot pulse output (TO8C)	
	1	Reserved (Do not set)	

- Bits 11 and 10—PB5 Mode Bits 1 and 0 (PB5MD1, PB5MD0): These bits select the function of pin PB5/TO7B/TO8B.

Bit 11: PB5MD1	Bit 10: PB5MD0	Description	
0	0	General input/output (PB5)	(Initial value)
	1	ATU-II PWM output (TO7B)	
1	0	ATU-II one-shot pulse output (TO8B)	
	1	Reserved (Do not set)	

- Bits 9 and 8—PB4 Mode Bits 1 and 0 (PB4MD1, PB4MD0): These bits select the function of pin PB4/TO7A/TO8A.

Bit 9: PB4MD1	Bit 8: PB4MD0	Description	
0	0	General input/output (PB4)	(Initial value)
	1	ATU-II PWM output (TO7A)	
1	0	ATU-II one-shot pulse output (TO8A)	
	1	Reserved (Do not set)	

Bit 6: PB3MD	Description	
0	General input/output (PB3)	(Initial value)
1	ATU-II PWM output (TO6D)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PB2 Mode Bit (PB2MD): Selects the function of pin PB2/TO6C.

Bit 4: PB2MD	Description	
0	General input/output (PB2)	(Initial value)
1	ATU-II PWM output (TO6C)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PB1 Mode Bit (PB1MD): Selects the function of pin PB1/TO6B.

Bit 2: PB1MD	Description	
0	General input/output (PB1)	(Initial value)
1	ATU-II PWM output (TO6B)	

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PB0 Mode Bit (PB0MD): Selects the function of pin PB0/TO6A.

Bit 0: PB0MD	Description	
0	General input/output (PB0)	(Initial value)
1	ATU-II PWM output (TO6A)	

Bit:	15	14	13	12	11	10	9	8
	PB15IR	PB14IR	PB13IR	—	PB11IR	PB10IR	PB9IR	PB8IR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PB7IR	PB6IR	PB5IR	PB4IR	PB3IR	PB2IR	PB1IR	PB0IR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port B invert register (PBIR) is a 16-bit readable/writable register that sets the port B inversion function. Bits PB15IR to PB13IR and PB11IR to PB0IR correspond to pins PB15/PULS5/SCK2 to PB13/SCK0 and PB11/RxD4/HRxD0/TO8H to PB0/TO6A. PBIR is enabled when port B pins function as ATU-II outputs or serial clock pins, and disabled otherwise.

When port B pins function as ATU-II outputs or serial clock pins, the value of a pin is inverted when the corresponding bit in PBIR is set to 1.

PBIR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PBnIR	Description	
0	Value is not inverted	(Initial value)
1	Value is inverted	

n = 15 to 13, 11 to 0

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

The port C IO register (PCIOR) is a 16-bit readable/writable register that selects the input/output direction of the 5 pins in port C. Bits PC4IOR to PC0IOR correspond to pins PC4/$\overline{\text{IRQ0}}$ to PC0/TxD1. PCIOR is enabled when port C pins function as general input/output pins (PC4 to PC0), and disabled otherwise.

When port C pins function as PC4 to PC0, a pin becomes an output when the corresponding bit in PCIOR is set to 1, and an input when the bit is cleared to 0.

PCIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

20.3.7 Port C Control Register (PCCR)

The port C control register (PCCR) is a 16-bit readable/writable register that selects the functions of the 5 multiplex pins in port C.

PCCR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	PC4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PC3MD	—	PC2MD	—	PC1MD	—	PC0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit 8: PC4MD	Description	
0	General input/output (PC4)	(Initial value)
1	Interrupt request input ($\overline{\text{IRQ0}}$)	

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PC3 Mode Bit (PC3MD): Selects the function of pin PC3/RxD2.

Bit 6: PC3MD	Description	
0	General input/output (PC3)	(Initial value)
1	Receive data input (RxD2)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PC2 Mode Bit (PC2MD): Selects the function of pin PC2/TxD2.

Bit 4: PC2MD	Description	
0	General input/output (PC2)	(Initial value)
1	Transmit data output (TxD2)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PC1 Mode Bit (PC1MD): Selects the function of pin PC1/RxD1.

Bit 2: PC1MD	Description	
0	General input/output (PC1)	(Initial value)
1	Receive data input (RxD1)	

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PC0 Mode Bit (PC0MD): Selects the function of pin PC0/TxD1.

Bit 0: PC0MD	Description	
0	General input/output (PC0)	(Initial value)
1	Transmit data output (TxD1)	

Bit:	15	14	13	12	11	10	9	8
	—	—	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR	PC8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	PD1 IOR	PD0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port D IO register (PDIOR) is a 16-bit readable/writable register that selects the input/output direction of the 14 pins in port D. Bits PD13IOR to PD0IOR correspond to pins PD13/PULS6/HTxD0/HTxD1 to PD0/TIO1A. PDIOR is enabled when port D pins function as general input/output pins (PD13 to PD0) or timer input/output pins, and disabled otherwise.

When port D pins function as PD13 to PD0 or timer input/output pins, a pin becomes an output when the corresponding bit in PDIOR is set to 1, and an input when the bit is cleared to 0.

PDIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

20.3.9 Port D Control Registers H and L (PDCRH, PDCRL)

Port D control registers H and L (PDCRH, PDCRL) are 16-bit readable/writable registers that select the functions of the 14 multiplex pins in port D. PDCRH selects the functions of the pins for the upper 6 bits of port D, and PDCRL selects the functions of the pins for the lower 8 bits.

PDCRH and PDCRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

	—	—	—	—	PD13 MD1	PD13 MD0	—	PD12 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PD11 MD	—	PD10 MD	—	PD9 MD	—	PD8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bits 15 to 12—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 11 and 10—PD13 Mode Bits 1 and 0 (PD13MD1, PD13MD0): These bits select the function of pin PD13/PULS6/HTxD0/HTxD1.

Bit 11: PD13MD1	Bit 10: PD13MD0	Description	
0	0	General input/output (PD13)	(Initial value)
	1	APC pulse output (PULS6)	
1	0	HCAN transmit data output (HTxD0)	
	1	HCAN transmit data output (HTxD1)	

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PD12 Mode Bit (PD12MD): Selects the function of pin PD12/PULS4.

Bit 8: PD12MD	Description	
0	General input/output (PD12)	(Initial value)
1	APC pulse output (PULS4)	

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PD11 Mode Bit (PD11MD): Selects the function of pin PD11/PULS3.

Bit 6: PD12MD	Description	
0	General input/output (PD11)	(Initial value)
1	APC pulse output (PULS3)	

Bit 4: PD10MD	Description	
0	General input/output (PD10)	(Initial value)
1	APC pulse output (PULS2)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PD9 Mode Bit (PD9MD): Selects the function of pin PD9/PULS1.

Bit 2: PD9MD	Description	
0	General input/output (PD9)	(Initial value)
1	APC pulse output (PULS1)	

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PD8 Mode Bit (PD8MD): Selects the function of pin PD8/PULS0.

Bit 0: PD8MD	Description	
0	General input/output (PD8)	(Initial value)
1	APC pulse output (PULS0)	

Port D Control Register L (PDCRL)

Bit:	15	14	13	12	11	10	9	8
	—	PD7MD	—	PD6MD	—	PD5MD	—	PD4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PD3MD	—	PD2MD	—	PD1MD	—	PD0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit 14: PD7MD	Description	
0	General input/output (PD7)	(Initial value)
1	ATU-II input capture input/output compare output (TIO1H)	

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PD6 Mode Bit (PD6MD): Selects the function of pin PD6/TIO1G.

Bit 12: PD6MD	Description	
0	General input/output (PD6)	(Initial value)
1	ATU-II input capture input/output compare output (TIO1G)	

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PD5 Mode Bit (PD5MD): Selects the function of pin PD5/TIO1F.

Bit 10: PD5MD	Description	
0	General input/output (PD5)	(Initial value)
1	ATU-II input capture input/output compare output (TIO1F)	

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PD4 Mode Bit (PD4MD): Selects the function of pin PD4/TIO1E.

Bit 8: PD4MD	Description	
0	General input/output (PD4)	(Initial value)
1	ATU-II input capture input/output compare output (TIO1E)	

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PD3 Mode Bit (PD3MD): Selects the function of pin PD3/TIO1D.

Bit 6: PD3MD	Description	
0	General input/output (PD3)	(Initial value)
1	ATU-II input capture input/output compare output (TIO1D)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PD2 Mode Bit (PD2MD): Selects the function of pin PD2/TIO1C.

1	ATU-II input capture input/output compare output (TIO1C)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PD1 Mode Bit (PD1MD): Selects the function of pin PD1/TIO1B.

Bit 2: PD1MD	Description	
0	General input/output (PD1)	(Initial value)
1	ATU-II input capture input/output compare output (TIO1B)	

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PD0 Mode Bit (PD0MD): Selects the function of pin PD0/TIO1A.

Bit 0: PD0MD	Description	
0	General input/output (PD0)	(Initial value)
1	ATU-II input capture input/output compare output (TIO1A)	

20.3.10 Port E IO Register (PEIOR)

Bit:	15	14	13	12	11	10	9	8
	PE15 IOR	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port E IO register (PEIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port E. Bits PE15IOR to PE0IOR correspond to pins PE15/A15 to PE0/A0. PEIOR is enabled when port E pins function as general input/output pins (PE15 to PE0), and disabled otherwise.

When port E pins function as PE15 to PE0, a pin becomes an output when the corresponding bit in PEIOR is set to 1, and an input when the bit is cleared to 0.

20.3.11 Port E Control Register (PECR)

Bit:	15	14	13	12	11	10	9	8
	PE15 MD	PE14 MD	PE13 MD	PE12 MD	PE11 MD	PE10 MD	PE9 MD	PE8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PE7 MD	PE6 MD	PE5 MD	PE4 MD	PE3 MD	PE2 MD	PE1 MD	PE0 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port E control register (PECR) is a 16-bit readable/writable register that selects the functions of the 16 multiplex pins in port E. PECR settings are not valid in all operating modes.

1. Expanded mode with on-chip ROM disabled

   Port E pins function as address output pins, and PECR settings are invalid.
2. Expanded mode with on-chip ROM enabled

   Port E pins are multiplexed as address output pins and general input/output pins. PECR settings are valid.
3. Single-chip mode

   Port E pins function as general input/output pins, and PECR settings are invalid.

PECR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bit 15—PE15 Mode Bit (PE15MD): Selects the function of pin PE15/A15.

	Description		
Bit 15: PE15MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A15) (Initial value)	General input/output (PE15) (Initial value)	General input/output (PE15) (Initial value)
1	Address output (A15)	Address output (A15)	General input/output (PE15)

Bit 14: PE14MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A14) (Initial value)	General input/output (PE14) (Initial value)	General input/output (PE14) (Initial value)
1	Address output (A14)	Address output (A14)	General input/output (PE14)

- Bit 13—PE13 Mode Bit (PE13MD): Selects the function of pin PE13/A13.

Bit 13: PE13MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A13) (Initial value)	General input/output (PE13) (Initial value)	General input/output (PE13) (Initial value)
1	Address output (A13)	Address output (A13)	General input/output (PE13)

- Bit 12—PE12 Mode Bit (PE12MD): Selects the function of pin PE12/A12.

Bit 12: PE12MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A12) (Initial value)	General input/output (PE12) (Initial value)	General input/output (PE12) (Initial value)
1	Address output (A12)	Address output (A12)	General input/output (PE12)

- Bit 11—PE11 Mode Bit (PE11MD): Selects the function of pin PE11/A11.

Bit 11: PE11MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A11) (Initial value)	General input/output (PE11) (Initial value)	General input/output (PE11) (Initial value)
1	Address output (A11)	Address output (A11)	General input/output (PE11)

Bit 10: PE10MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A10) (Initial value)	General input/output (PE10) (Initial value)	General input/output (PE10) (Initial value)
1	Address output (A10)	Address output (A10)	General input/output (PE10)

- Bit 9—PE9 Mode Bit (PE9MD): Selects the function of pin PE9/A9.

	Description		
Bit 9: PE9MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A9) (Initial value)	General input/output (PE9) (Initial value)	General input/output (PE9) (Initial value)
1	Address output (A9)	Address output (A9)	General input/output (PE9)

- Bit 8—PE8 Mode Bit (PE8MD): Selects the function of pin PE8/A8.

	Description		
Bit 8: PE8MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A8) (Initial value)	General input/output (PE8) (Initial value)	General input/output (PE8) (Initial value)
1	Address output (A8)	Address output (A8)	General input/output (PE8)

- Bit 7—PE7 Mode Bit (PE7MD): Selects the function of pin PE7/A7.

	Description		
Bit 7: PE7MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A7) (Initial value)	General input/output (PE7) (Initial value)	General input/output (PE7) (Initial value)
1	Address output (A7)	Address output (A7)	General input/output (PE7)

Bit 6: PE6MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A6) (Initial value)	General input/output (PE6) (Initial value)	General input/output (PE6) (Initial value)
1	Address output (A6)	Address output (A6)	General input/output (PE6)

- Bit 5—PE5 Mode Bit (PE5MD): Selects the function of pin PE5/A5.

Bit 5: PE5MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A5) (Initial value)	General input/output (PE5) (Initial value)	General input/output (PE5) (Initial value)
1	Address output (A5)	Address output (A5)	General input/output (PE5)

- Bit 4—PE4 Mode Bit (PE4MD): Selects the function of pin PE4/A4.

Bit 4: PE4MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A4) (Initial value)	General input/output (PE4) (Initial value)	General input/output (PE4) (Initial value)
1	Address output (A4)	Address output (A4)	General input/output (PE4)

- Bit 3—PE3 Mode Bit (PE3MD): Selects the function of pin PE3/A3.

Bit 3: PE3MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A3) (Initial value)	General input/output (PE3) (Initial value)	General input/output (PE3) (Initial value)
1	Address output (A3)	Address output (A3)	General input/output (PE3)

Bit 2: PE2MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A2) (Initial value)	General input/output (PE2) (Initial value)	General input/output (PE2) (Initial value)
1	Address output (A2)	Address output (A2)	General input/output (PE2)

- Bit 1—PE1 Mode Bit (PE1MD): Selects the function of pin PE1/A1.

Bit 1: PE1MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A1) (Initial value)	General input/output (PE1) (Initial value)	General input/output (PE1) (Initial value)
1	Address output (A1)	Address output (A1)	General input/output (PE1)

- Bit 0—PE0 Mode Bit (PE0MD): Selects the function of pin PE0/A0.

Bit 0: PE0MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A0) (Initial value)	General input/output (PE0) (Initial value)	General input/output (PE0) (Initial value)
1	Address output (A0)	Address output (A0)	General input/output (PE0)

Bit:	15	14	13	12	11	10	9	8
	PF15 IOR	PF14 IOR	PF13 IOR	PF12 IOR	PF11 IOR	PF10 IOR	PF9 IOR	PF8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PF7 IOR	PF6 IOR	PF5 IOR	PF4 IOR	PF3 IOR	PF2 IOR	PF1 IOR	PF0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port F IO register (PFIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port F. Bits PF15IOR to PF0IOR correspond to pins PF15/$\overline{\text{BREQ}}$ to PF0/A16. PFIOR is enabled when port F pins function as general input/output pins (PF15 to PF0), and disabled otherwise.

When port F pins function as PF15 to PF0, a pin becomes an output when the corresponding bit in PFIOR is set to 1, and an input when the bit is cleared to 0.

PFIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Port F control registers H and L (PFCRH, PFCRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port F and the function of the CK pin. PFCRH selects the functions of the pins for the upper 8 bits of port F, and PFCRL selects the functions of the pins for the lower 8 bits.

PFCRH and PFCRL are initialized to H'0015 and H'5000, respectively, by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

Port F Control Register H (PFCRH)

Bit:	15	14	13	12	11	10	9	8
	CKHIZ	PF15MD	—	PF14MD	—	PF13MD	—	PF12MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PF11MD	—	PF10MD	—	PF9MD	—	PF8MD
Initial value:	0	0	0	1	0	1	0	1
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—CKHIZ Bit: Selects the function of pin CK.

Bit: CKHIZ	Description	
0	CK pin output	(Initial value)
1	CK pin Hi-Z	

- Bit 14—PF15 Mode Bit (PF15MD): Selects the function of pin PF15/$\overline{\text{BREQ}}$.

	Description	
Bit 14: PF15MD	**Expanded Mode**	**Single-Chip Mode**
0	General input/output (PF15) (Initial value)	General input/output (PF15) (Initial value)
1	Bus request input ($\overline{\text{BREQ}}$)	General input/output (PF15)

	Description	
Bit 12: PF14MD	**Expanded Mode**	**Single-Chip Mode**
0	General input/output (PF14) (Initial value)	General input/output (PF14) (Initial value)
1	Bus acknowledge output ($\overline{\text{BACK}}$)	General input/output (PF14)

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PF13 Mode Bit (PF13MD): Selects the function of pin PF13/$\overline{\text{CS3}}$.

	Description	
Bit 10: PF13MD	**Expanded Mode**	**Single-Chip Mode**
0	General input/output (PF13) (Initial value)	General input/output (PF13) (Initial value)
1	Chip select output ($\overline{\text{CS3}}$)	General input/output (PF13)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PF12 Mode Bit (PF12MD): Selects the function of pin PF12/$\overline{\text{CS2}}$.

	Description	
Bit 8: PF12MD	**Expanded Mode**	**Single-Chip Mode**
0	General input/output (PF12) (Initial value)	General input/output (PF12) (Initial value)
1	Chip select output ($\overline{\text{CS2}}$)	General input/output (PF12)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PF11 Mode Bit (PF11MD): Selects the function of pin PF11/$\overline{\text{CS1}}$.

	Description	
Bit 6: PF11MD	**Expanded Mode**	**Single-Chip Mode**
0	General input/output (PF11) (Initial value)	General input/output (PF11) (Initial value)
1	Chip select output ($\overline{\text{CS1}}$)	General input/output (PF11)

	Description	
Bit 4: PF10MD	**Expanded Mode**	**Single-Chip Mode**
0	General input/output (PF10)	General input/output (PF10)
1	Chip select output ($\overline{CS0}$) (Initial value)	General input/output (PF10) (Initial value)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PF9 Mode Bit (PF9MD): Selects the function of pin PF9/$\overline{RD}$.

	Description	
Bit 2: PF9MD	**Expanded Mode**	**Single-Chip Mode**
0	General input/output (PF9)	General input/output (PF9)
1	Read output ($\overline{RD}$) (Initial value)	General input/output (PF9) (Initial value)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PF8 Mode Bit (PF8MD): Selects the function of pin PF8/$\overline{WAIT}$.

	Description	
Bit 0: PF8MD	**Expanded Mode**	**Single-Chip Mode**
0	General input/output (PF8)	General input/output (PF8)
1	Wait state input ($\overline{WAIT}$) (Initial value)	General input/output (PF8) (Initial value)

Port F Control Register L (PFCRL)

Bit:	15	14	13	12	11	10	9	8
	—	PF7MD	—	PF6MD	PF5MD1	PF5MD0	—	PF4MD
Initial value:	0	1	0	1	0	0	0	0
R/W:	R	R/W	R	R/W	R/W	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PF3MD	—	PF2MD	—	PF1MD	—	PF0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

	Description	
Bit 14: PF7MD	**Expanded Mode**	**Single-Chip Mode**
0	General input/output (PF7)	General input/output (PF7)
1	Upper write ($\overline{\text{WRH}}$) (Initial value)	General input/output (PF7) (Initial value)

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PF6 Mode Bit (PF6MD): Selects the function of pin PF6/$\overline{\text{WRL}}$.

	Description	
Bit 12: PF6MD	**Expanded Mode**	**Single-Chip Mode**
0	General input/output (PF6)	General input/output (PF6)
1	Lower write ($\overline{\text{WRL}}$) (Initial value)	General input/output (PF6) (Initial value)

- Bits 11 and 10—PF5 Mode Bits 1 and 0 (PF5MD1, PF5MD0): These bits select the function of pin PF5/A21/$\overline{\text{POD}}$.

		Description		
Bit 11: PF5MD1	**Bit 10: PF5MD0**	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	0	Address output (A21) (Initial value)	General input/output (PF5) (Initial value)	General input/output (PF5) (Initial value)
	1	Address output (A21)	Address output (A21)	General input/output (PF5)
1	0	Address output (A21)	Port output disable input ($\overline{\text{POD}}$)	Port output disable input ($\overline{\text{POD}}$)
	1	Reserved (Do not set)	Reserved (Do not set)	Reserved (Do not set)

	Description		
Bit 8: PF4MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A20) (Initial value)	General input/output (PF4) (Initial value)	General input/output (PF4) (Initial value)
1	Address output (A20)	Address output (A20)	General input/output (PF4)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PF3 Mode Bit (PF3MD): Selects the function of pin PF3/A19.

	Description		
Bit 6: PF3MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A19) (Initial value)	General input/output (PF3) (Initial value)	General input/output (PF3) (Initial value)
1	Address output (A19)	Address output (A19)	General input/output (PF3)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PF2 Mode Bit (PF2MD): Selects the function of pin PF2/A18.

	Description		
Bit 4: PF2MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A18) (Initial value)	General input/output (PF2) (Initial value)	General input/output (PF2) (Initial value)
1	Address output (A18)	Address output (A18)	General input/output (PF2)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PF1 Mode Bit (PF1MD): Selects the function of pin PF1/A17.

	Description		
Bit 2: PF1MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A17) (Initial value)	General input/output (PF1) (Initial value)	General input/output (PF1) (Initial value)
1	Address output (A17)	Address output (A17)	General input/output (PF1)

	Description		
Bit 0: PF0MD	**Expanded Mode with ROM Disabled**	**Expanded Mode with ROM Enabled**	**Single-Chip Mode**
0	Address output (A16) (Initial value)	General input/output (PF0) (Initial value)	General input/output (PF0) (Initial value)
1	Address output (A16)	Address output (A16)	General input/output (PF0)

20.3.14 Port G IO Register (PGIOR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	PG3IOR	PG2IOR	PG1IOR	PG0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

The port G IO register (PGIOR) is a 16-bit readable/writable register that selects the input/output direction of the 4 pins in port G. Bits PG3IOR to PG0IOR correspond to pins PG3/$\overline{\text{IRQ3}}$/$\overline{\text{ADTRG0}}$ to PG0/PULS7/HRxD0/HRxD1.

When port G pins function as PG3 to PG0, a pin becomes an output when the corresponding bit in PGIOR is set to 1, and an input when the bit is cleared to 0.

PGIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

The port G control register (PGCR) is a 16-bit readable/writable register that selects the functions of the 4 multiplex pins in port G.

PGCR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	PG3MD1	PG3MD0	PG2MD1	PG2MD0	—	PG1MD	PG0MD1	PG0MD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

- Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 7 and 6—PG3 Mode Bits 1 and 0 (PG3MD1, PG3MD0): These bits select the function of pin PG3/$\overline{\text{IRQ3}}$/$\overline{\text{ADTRG0}}$.

Bit 7: PG3MD1	Bit 6: PG3MD0	Description	
0	0	General input/output (PG3)	(Initial value)
	1	Interrupt request input ($\overline{\text{IRQ3}}$)	
1	0	A/D conversion trigger input ($\overline{\text{ADTRG0}}$)	
	1	Reserved (Do not set)	

- Bits 5 and 4—PG2 Mode Bits 1 and 0 (PG2MD1, PG2MD0): These bits select the function of pin PG2/$\overline{\text{IRQ2}}$/ADEND.

Bit 5: PG2MD1	Bit 4: PG2MD0	Description	
0	0	General input/output (PG2)	(Initial value)
	1	Interrupt request input ($\overline{\text{IRQ2}}$)	
1	0	A/D conversion end output (ADEND)	
	1	Reserved (Do not set)	

Bit 2: PG1MD	Description	
0	General input/output (PG1)	(Initial value)
1	Interrupt request input ($\overline{\text{IRQ1}}$)	

- Bits 1 and 0—PG0 Mode Bits 1 and 0 (PG0MD1, PG2MD0): These bits select the function of pin PG0/PULS7/HRxD0/HRxD1.

Bit 1: PG0MD1	Bit 0: PG0MD0	Description	
0	0	General input/output (PG0)	(Initial value)
	1	APC pulse output (PULS7)	
1	0	HCAN receive data input (HRxD0)	
	1	HCAN receive data input (HRxD1)	

20.3.16 Port H IO Register (PHIOR)

Bit:	15	14	13	12	11	10	9	8
	PH15 IOR	PH14 IOR	PH13 IOR	PH12 IOR	PH11 IOR	PH10 IOR	PH9 IOR	PH8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PH7 IOR	PH6 IOR	PH5 IOR	PH4 IOR	PH3 IOR	PH2 IOR	PH1 IOR	PH0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port H IO register (PHIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port H. Bits PH15IOR to PH0IOR correspond to pins PH15/D15 to PH0/D0. PHIOR is enabled when port H pins function as general input/output pins (PH15 to PH0), and disabled otherwise.

When port H pins function as PH15 to PH0, a pin becomes an output when the corresponding bit in PHIOR is set to 1, and an input when the bit is cleared to 0.

PHIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Bit:	15	14	13	12	11	10	9	8
	PH15 MD	PH14 MD	PH13 MD	PH12 MD	PH11 MD	PH10 MD	PH9 MD	PH8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PH7 MD	PH6 MD	PH5 MD	PH4 MD	PH3 MD	PH2 MD	PH1 MD	PH0 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port H control register (PHCR) is a 16-bit readable/writable register that selects the functions of the 16 multiplex pins in port H. PHCR settings are not valid in all operating modes.

1. Expanded mode with on-chip ROM disabled (area 0: 8-bit bus)
   Port H pins D0 to D7 function as data input/output pins, and PHCR settings are invalid.
2. Expanded mode with on-chip ROM disabled (area 0: 16-bit bus)
   Port H pins function as data input/output pins, and PHCR settings are invalid.
3. Expanded mode with on-chip ROM enabled
   Port H pins are multiplexed as data input/output pins and general input/output pins. PHCR settings are valid.
4. Single-chip mode
   Port H pins function as general input/output pins, and PHCR settings are invalid.

PHCR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Bit 15: PH15MD	Description: Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH15) (Initial value)	Data input/output (D15) (Initial value)	General input/output (PH15) (Initial value)	General input/output (PH15) (Initial value)
1	Data input/output (D15)	Data input/output (D15)	Data input/output (D15)	General input/output (PH15)

- Bit 14—PH14 Mode Bit (PH14MD): Selects the function of pin PH14/D14.

Bit 14: PH14MD	Description: Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH14) (Initial value)	Data input/output (D14) (Initial value)	General input/output (PH14) (Initial value)	General input/output (PH14) (Initial value)
1	Data input/output (D14)	Data input/output (D14)	Data input/output (D14)	General input/output (PH14)

- Bit 13—PH13 Mode Bit (PH13MD): Selects the function of pin PH13/D13.

Bit 13: PH13MD	Description: Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH13) (Initial value)	Data input/output (D13) (Initial value)	General input/output (PH13) (Initial value)	General input/output (PH13) (Initial value)
1	Data input/output (D13)	Data input/output (D13)	Data input/output (D13)	General input/output (PH13)

Bit 12: PH12MD	Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH12) (Initial value)	Data input/output (D12) (Initial value)	General input/output (PH12) (Initial value)	General input/output (PH12) (Initial value)
1	Data input/output (D12)	Data input/output (D12)	Data input/output (D12)	General input/output (PH12)

- Bit 11—PH11 Mode Bit (PH11MD): Selects the function of pin PH11/D11.

	Description			
Bit 11: PH11MD	Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH11) (Initial value)	Data input/output (D11) (Initial value)	General input/output (PH11) (Initial value)	General input/output (PH11) (Initial value)
1	Data input/output (D11)	Data input/output (D11)	Data input/output (D11)	General input/output (PH11)

- Bit 10—PH10 Mode Bit (PH10MD): Selects the function of pin PH10/D10.

	Description			
Bit 10: PH10MD	Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH10) (Initial value)	Data input/output (D10) (Initial value)	General input/output (PH10) (Initial value)	General input/output (PH10) (Initial value)
1	Data input/output (D10)	Data input/output (D10)	Data input/output (D10)	General input/output (PH10)

Bit 9: PH9MD	Description: Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH9) (Initial value)	Data input/output (D9) (Initial value)	General input/output (PH9) (Initial value)	General input/output (PH9) (Initial value)
1	Data input/output (D9)	Data input/output (D9)	Data input/output (D9)	General input/output (PH9)

- Bit 8—PH8 Mode Bit (PH8MD): Selects the function of pin PH8/D8.

Bit 8: PH8MD	Description: Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH8) (Initial value)	Data input/output (D8) (Initial value)	General input/output (PH8) (Initial value)	General input/output (PH8) (Initial value)
1	Data input/output (D8)	Data input/output (D8)	Data input/output (D8)	General input/output (PH8)

- Bit 7—PH7 Mode Bit (PH7MD): Selects the function of pin PH7/D7.

Bit 7: PH7MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D7) (Initial value)	General input/output (PH7) (Initial value)	General input/output (PH7) (Initial value)
1	Data input/output (D7)	Data input/output (D7)	General input/output (PH7)

Bit 6: PH6MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D6) (Initial value)	General input/output (PH6) (Initial value)	General input/output (PH6) (Initial value)
1	Data input/output (D6)	Data input/output (D6)	General input/output (PH6)

- Bit 5—PH5 Mode Bit (PH5MD): Selects the function of pin PH5/D5.

	Description		
Bit 5: PH5MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D5) (Initial value)	General input/output (PH5) (Initial value)	General input/output (PH5) (Initial value)
1	Data input/output (D5)	Data input/output (D5)	General input/output (PH5)

- Bit 4—PH4 Mode Bit (PH4MD): Selects the function of pin PH4/D4.

	Description		
Bit 4: PH4MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D4) (Initial value)	General input/output (PH4) (Initial value)	General input/output (PH4) (Initial value)
1	Data input/output (D4)	Data input/output (D4)	General input/output (PH4)

- Bit 3—PH3 Mode Bit (PH3MD): Selects the function of pin PH3/D3.

	Description		
Bit 3: PH3MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D3) (Initial value)	General input/output (PH3) (Initial value)	General input/output (PH3) (Initial value)
1	Data input/output (D3)	Data input/output (D3)	General input/output (PH3)

Bit 2: PH2MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D2) (Initial value)	General input/output (PH2) (Initial value)	General input/output (PH2) (Initial value)
1	Data input/output (D2)	Data input/output (D2)	General input/output (PH2)

- Bit 1—PH1 Mode Bit (PH1MD): Selects the function of pin PH1/D1.

Bit 1: PH1MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D1) (Initial value)	General input/output (PH1) (Initial value)	General input/output (PH1) (Initial value)
1	Data input/output (D1)	Data input/output (D1)	General input/output (PH1)

- Bit 0—PH0 Mode Bit (PH0MD): Selects the function of pin PH0/D0.

Bit 0: PH0MD	Description: Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D0) (Initial value)	General input/output (PH0) (Initial value)	General input/output (PH0) (Initial value)
1	Data input/output (D0)	Data input/output (D0)	General input/output (PH0)

Bit:	15	14	13	12	11	10	9	8
	PJ15 IOR	PJ14 IOR	PJ13 IOR	PJ12 IOR	PJ11 IOR	PJ10 IOR	PJ9 IOR	PJ8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PJ7 IOR	PJ6 IOR	PJ5 IOR	PJ4 IOR	PJ3 IOR	PJ2 IOR	PJ1 IOR	PJ0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port J IO register (PJIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port J. Bits PJ15IOR to PJ0IOR correspond to pins PJ15/TI9F to PJ0/TIO2A. PJIOR is enabled when port J pins function as general input/output pins (PJ15 to PJ0) or ATU-II input/output pins, and disabled otherwise. When ATU-II event counter input is selected, however, the bits 10 to 15 of the PJIOR should be cleared to 0.

When port J pins function as PJ15 to PJ0 or ATU-II input/output pins, a pin becomes an output when the corresponding bit in PJIOR is set to 1, and an input when the bit is cleared to 0.

PJIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Port J control registers H and L (PJCRH, PJCRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port J. PJCRH selects the functions of the pins for the upper 8 bits of port J, and PJCRL selects the functions of the pins for the lower 8 bits.

PJCRH and PJCRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

Port J Control Register H (PJCRH)

Bit:	15	14	13	12	11	10	9	8
	—	PJ15MD	—	PJ14MD	—	PJ13MD	—	PJ12MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PJ11MD	—	PJ10MD	—	PJ9MD	—	PJ8MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PJ15 Mode Bit (PJ15MD): Selects the function of pin PJ15/TI9F.

Bit 14: PJ15MD	Description	
0	General input/output (PJ15)	(Initial value)
1	ATU-II event counter input (TI9F)	

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PJ14 Mode Bit (PJ14MD): Selects the function of pin PJ14/TI9E.

Bit 12: PJ14MD	Description	
0	General input/output (PJ14)	(Initial value)
1	ATU-II event counter input (TI9E)	

Bit 10: PJ13MD	Description	
0	General input/output (PJ13)	(Initial value)
1	ATU-II event counter input (TI9D)	

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PJ12 Mode Bit (PJ12MD): Selects the function of pin PJ12/TI9C.

Bit 8: PJ12MD	Description	
0	General input/output (PJ12)	(Initial value)
1	ATU-II event counter input (TI9C)	

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PJ11 Mode Bit (PJ11MD): Selects the function of pin PJ11/TI9B.

Bit 6: PJ11MD	Description	
0	General input/output (PJ11)	(Initial value)
1	ATU-II event counter input (TI9B)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PJ10 Mode Bit (PJ10MD): Selects the function of pin PJ10/TI9A.

Bit 4: PJ10MD	Description	
0	General input/output (PJ10)	(Initial value)
1	ATU-II event counter input (TI9A)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PJ9 Mode Bit (PJ9MD): Selects the function of pin PJ9/TIO5D.

Bit 2: PJ9MD	Description	
0	General input/output (PJ9)	(Initial value)
1	ATU-II input capture input/output compare output (TIO5D)	

Bit 0: PJ8MD	Description	
0	General input/output (PJ8)	(Initial value)
1	ATU-II input capture input/output compare output (TIO5C)	

Port J Control Register L (PJCRL)

Bit:	15	14	13	12	11	10	9	8
	—	PJ7MD	—	PJ6MD	—	PJ5MD	—	PJ4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PJ3MD	—	PJ2MD	—	PJ1MD	—	PJ0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PJ7 Mode Bit (PJ7MD): Selects the function of pin PJ7/TIO2H.

Bit 14: PJ7MD	Description	
0	General input/output (PJ7)	(Initial value)
1	ATU-II input capture input/output compare output (TIO2H)	

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PJ6 Mode Bit (PJ6MD): Selects the function of pin PJ6/TIO2G.

Bit 12: PJ6MD	Description	
0	General input/output (PJ6)	(Initial value)
1	ATU-II input capture input/output compare output (TIO2G)	

Bit 10: PJ5MD	Description	
0	General input/output (PJ5)	(Initial value)
1	ATU-II input capture input/output compare output (TIO2F)	

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PJ4 Mode Bit (PJ4MD): Selects the function of pin PJ4/TIO2E.

Bit 8: PJ4MD	Description	
0	General input/output (PJ4)	(Initial value)
1	ATU-II input capture input/output compare output (TIO2E)	

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PJ3 Mode Bit (PJ3MD): Selects the function of pin PJ3/TIO2D.

Bit 6: PJ3MD	Description	
0	General input/output (PJ3)	(Initial value)
1	ATU-II input capture input/output compare output (TIO2D)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PJ2 Mode Bit (PJ2MD): Selects the function of pin PJ2/TIO2C.

Bit 4: PJ2MD	Description	
0	General input/output (PJ2)	(Initial value)
1	ATU-II input capture input/output compare output (TIO2C)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PJ1 Mode Bit (PJ1MD): Selects the function of pin PJ1/TIO2B.

Bit 2: PJ1MD	Description	
0	General input/output (PJ1)	(Initial value)
1	ATU-II input capture input/output compare output (TIO2B)	

Bit 0: PJ0MD	Description	
0	General input/output (PJ0)	(Initial value)
1	ATU-II input capture input/output compare output (TIO2A)	

20.3.20 Port K IO Register (PKIOR)

Bit:	15	14	13	12	11	10	9	8
	PK15 IOR	PK14 IOR	PK13 IOR	PK12 IOR	PK11 IOR	PK10 IOR	PK9 IOR	PK8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PK7 IOR	PK6 IOR	PK5 IOR	PK4 IOR	PK3 IOR	PK2 IOR	PK1 IOR	PK0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port K IO register (PKIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port K. Bits PK15IOR to PK0IOR correspond to pins PK15/TO8P to PK0/TO8A. PKIOR is enabled when port K pins function as general input/output pins (PK15 to PK0), and disabled otherwise.

When port K pins function as PK15 to PK0, a pin becomes an output when the corresponding bit in PKIOR is set to 1, and an input when the bit is cleared to 0.

PKIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

20.3.21 Port K Control Registers H and L (PKCRH, PKCRL)

Port K control registers H and L (PKCRH, PKCRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port K. PKCRH selects the functions of the pins for the upper 8 bits of port K, and PKCRL selects the functions of the pins for the lower 8 bits.

PKCRH and PKCRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

	—	PK15 MD	—	PK14 MD	—	PK13 MD	—	PK12 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PK11 MD	—	PK10 MD	—	PK9 MD	—	PK8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PK15 Mode Bit (PK15MD): Selects the function of pin PK15/TO8P.

Bit 14: PK15MD	Description	
0	General input/output (PK15)	(Initial value)
1	ATU-II one-shot pulse output (TO8P)	

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PK14 Mode Bit (PK14MD): Selects the function of pin PK14/TO8O.

Bit 12: PK14MD	Description	
0	General input/output (PK14)	(Initial value)
1	ATU-II one-shot pulse output (TO8O)	

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PK13 Mode Bit (PK13MD): Selects the function of pin PK13/TO8N.

Bit 10: PK13MD	Description	
0	General input/output (PK13)	(Initial value)
1	ATU-II one-shot pulse output (TO8N)	

Bit 8: PK12MD	Description	
0	General input/output (PK12)	(Initial value)
1	ATU-II one-shot pulse output (TO8M)	

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PK11 Mode Bit (PK11MD): Selects the function of pin PK11/TO8L.

Bit 6: PK11MD	Description	
0	General input/output (PK11)	(Initial value)
1	ATU-II one-shot pulse output (TO8L)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PK10 Mode Bit (PK10MD): Selects the function of pin PK10/TO8K.

Bit 4: PK10MD	Description	
0	General input/output (PK10)	(Initial value)
1	ATU-II one-shot pulse output (TO8K)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PK9 Mode Bit (PK9MD): Selects the function of pin PK9/TO8J.

Bit 2: PK9MD	Description	
0	General input/output (PK9)	(Initial value)
1	ATU-II one-shot pulse output (TO8J)	

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PK8 Mode Bit (PK8MD): Selects the function of pin PK8/TO8I.

Bit 0: PK8MD	Description	
0	General input/output (PK8)	(Initial value)
1	ATU-II one-shot pulse output (TO8I)	

Bit:	15	14	13	12	11	10	9	8
	—	PK7MD	—	PK6MD	—	PK5MD	—	PK4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PK3MD	—	PK2MD	—	PK1MD	—	PK0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PK7 Mode Bit (PK7MD): Selects the function of pin PK7/TO8H.

Bit 14: PK7MD	Description	
0	General input/output (PK7)	(Initial value)
1	ATU-II one-shot pulse output (TO8H)	

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PK6 Mode Bit (PK6MD): Selects the function of pin PK6/TO8G.

Bit 12: PK6MD	Description	
0	General input/output (PK6)	(Initial value)
1	ATU-II one-shot pulse output (TO8G)	

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PK5 Mode Bit (PK5MD): Selects the function of pin PK5/TO8F.

Bit 10: PK5MD	Description	
0	General input/output (PK5)	(Initial value)
1	ATU-II one-shot pulse output (TO8F)	

Bit 8: PK4MD	Description	
0	General input/output (PK4)	(Initial value)
1	ATU-II one-shot pulse output (TO8E)	

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PK3 Mode Bit (PK3MD): Selects the function of pin PK3/TO8D.

Bit 6: PK3MD	Description	
0	General input/output (PK3)	(Initial value)
1	ATU-II one-shot pulse output (TO8D)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PK2 Mode Bit (PK2MD): Selects the function of pin PK2/TO8C.

Bit 4: PK2MD	Description	
0	General input/output (PK2)	(Initial value)
1	ATU-II one-shot pulse output (TO8C)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PK1 Mode Bit (PK1MD): Selects the function of pin PK1/TO8B.

Bit 2: PK1MD	Description	
0	General input/output (PK1)	(Initial value)
1	ATU-II one-shot pulse output (TO8B)	

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PK0 Mode Bit (PK0MD): Selects the function of pin PK0/TO8A.

Bit 0: PK0MD	Description	
0	General input/output (PK0)	(Initial value)
1	ATU-II one-shot pulse output (TO8A)	

Bit:	15	14	13	12	11	10	9	8
	PK15IR	PK14IR	PK13IR	PK12IR	PK11IR	PK10IR	PK9IR	PK8IR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PK7IR	PK6IR	PK5IR	PK4IR	PK3IR	PK2IR	PK1IR	PK0IR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port K invert register (PKIR) is a 16-bit readable/writable register that sets the port K inversion function. Bits PK15IR to PK0IR correspond to pins PK15/TO8P to PK0/TO8A. PKIR is enabled when port K pins function as ATU-II outputs, and disabled otherwise.

When port K pins function as ATU-II outputs, the value of a pin is inverted when the corresponding bit in PKIR is set to 1.

PKIR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PKnIR	**Description**	
0	Value is not inverted	(Initial value)
1	Value is inverted	

n = 15 to 0

Bit:	15	14	13	12	11	10	9	8
	—	—	PL13 IOR	PL12 IOR	PL11 IOR	PL10 IOR	PL9 IOR	PL8 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PL7 IOR	PL6 IOR	PL5 IOR	PL4 IOR	PL3 IOR	PL2 IOR	PL1 IOR	PL0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port L IO register (PLIOR) is a 16-bit readable/writable register that selects the input/output direction of the 14 pins in port L. Bits PL13IOR to PL0IOR correspond to pins PL13/$\overline{\text{IRQOUT}}$ to PL0/TI10. PLIOR is enabled when port L pins function as general input/output pins (PL13 to PL0), timer input/output pins (TIO11A, TIO11B), or serial clock pins (SCK2, SCK3, SCK4), and disabled otherwise.

When port L pins function as PL13 to PL0, TIO11A and TIO11B, or SCK2, SCK3, and SCK4, a pin becomes an output when the corresponding bit in PLIOR is set to 1, and an input when the bit is cleared to 0.

PLIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Port L control registers H and L (PLCRH, PLCRL) are 16-bit readable/writable registers that select the functions of the 14 multiplex pins in port L. PLCRH selects the functions of the pins for the upper 6 bits of port L, and PLCRL selects the functions of the pins for the lower 8 bits.

PLCRH and PLCRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

Port L Control Register H (PLCRH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	PL13 MD1	PL13 MD0	—	PL12 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	PL11 MD1	PL11 MD0	PL10 MD1	PL10 MD0	PL9 MD1	PL9 MD0	—	PL8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

- Bits 15 to 12—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 11 and 10—PL13 Mode Bits 1 and 0 (PL13MD1, PL13MD0): These bits select the function of pin PL13/$\overline{\text{IRQOUT}}$.

Bit 11: PL13MD1	Bit 10: PL13MD0	Description	
0	0	General input/output (PL13)	(Initial value)
	1	$\overline{\text{IRQOUT}}$ is fixed high ($\overline{\text{IRQOUT}}$)	
1	0	$\overline{\text{IRQOUT}}$ is output by INTC interrupt request ($\overline{\text{IRQOUT}}$)	
	1	Reserved (Do not set)	

Bit 8: PL12MD	Description	
0	General input/output (PL12)	(Initial value)
1	Interrupt request input ($\overline{\text{IRQ4}}$)	

- Bits 7 and 6—PL11 Mode Bits 1 and 0 (PL11MD1, PL11MD0): These bits select the function of pin PL11/HRxD0/HRxD1.

Bit 7: PL11MD1	Bit 6: PL11MD0	Description	
0	0	General input/output (PL11)	(Initial value)
	1	HCAN receive data input (HRxD0)	
1	0	HCAN receive data input (HRxD1)	
	1	HCAN receive data input (both HRxD0 and HRxD1 input)	

- Bits 5 and 4—PL10 Mode Bits 1 and 0 (PL10MD1, PL10MD0): These bits select the function of pin PL10/HTxD0/HTxD1.

Bit 5: PL10MD1	Bit 4: PL10MD0	Description	
0	0	General input/output (PL10)	(Initial value)
	1	HCAN transmit data output (HTxD0)	
1	0	HCAN transmit data output (HTxD1)	
	1	HCAN transmit data output (AND of HTxD0 and HTxD1)	

- Bits 3 and 2—PL9 Mode Bits 1 and 0 (PL9MD1, PL9MD0): These bits select the function of pin PL9/SCK4/$\overline{\text{IRQ5}}$.

Bit 3: PL9MD1	Bit 2: PL9MD0	Description	
0	0	General input/output (PL9)	(Initial value)
	1	Serial clock input/output (SCK4)	
1	0	Interrupt request input ($\overline{\text{IRQ5}}$)	
	1	Reserved (Do not set)	

Bit 0: PL8MD	Description	
0	General input/output (PL8)	(Initial value)
1	Serial clock input/output (SCK3)	

Port L Control Register L (PLCRL)

Bit:	15	14	13	12	11	10	9	8
	—	PL7MD	—	PL6MD	—	PL5MD	—	PL4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PL3MD	PL2MD1	PL2MD0	PL1MD1	PL1MD0	—	PL0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PL7 Mode Bit (PL7MD): Selects the function of pin PL7/SCK2.

Bit 14: PL7MD	Description	
0	General input/output (PL7)	(Initial value)
1	Serial clock input/output (SCK2)	

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PL6 Mode Bit (PL6MD): Selects the function of pin PL6/ADEND.

Bit 12: PL6MD	Description	
0	General input/output (PL6)	(Initial value)
1	A/D conversion end output (ADEND)	

Bit 10: PL5MD	Description	
0	General input/output (PL5)	(Initial value)
1	A/D conversion trigger input ($\overline{\text{ADTRG1}}$)	

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PL4 Mode Bit (PL4MD): Selects the function of pin PL4/$\overline{\text{ADTRG0}}$.

Bit 8: PL4MD	Description	
0	General input/output (PL4)	(Initial value)
1	A/D conversion trigger input ($\overline{\text{ADTRG0}}$)	

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PL3 Mode Bit (PL3MD): Selects the function of pin PL3/TCLKB.

Bit 6: PL3MD	Description	
0	General input/output (PL3)	(Initial value)
1	ATU-II clock input (TCLKB)	

- Bits 5 and 4—PL2 Mode Bits 1 and 0 (PL2MD1, PL2MD0): These bits select the function of pin PL2/TIO11B/$\overline{\text{IRQ7}}$.

Bit 5: PL2MD1	Bit 4: PL2MD0	Description	
0	0	General input/output (PL2)	(Initial value)
	1	ATU-II input capture input/output compare output (TIO11B)	
1	0	Interrupt request input ($\overline{\text{IRQ7}}$)	
	1	Reserved (Do not set)	

Bit 3: PL1MD1	Bit 2: PL1MD0	Description	
0	0	General input/output (PL1)	(Initial value)
	1	ATU-II input capture input/output compare output (TIO11A)	
1	0	Interrupt request input ($\overline{\text{IRQ6}}$)	
	1	Reserved (Do not set)	

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PL0 Mode Bit (PL0MD): Selects the function of pin PL0/TI10.

Bit 0: PL0MD	Description	
0	General input/output (PL0)	(Initial value)
1	ATU-II edge input (TI10)	

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PL9IR	PL8IR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PL7IR	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

The port L invert register (PLIR) is a 16-bit readable/writable register that sets the port L inversion function. Bits PL9IR to PL7IR correspond to pins PL9/SCK4/$\overline{\text{IRQ5}}$ to PL7/SCK2. PLIR is enabled when port L pins function as serial clock pins, and disabled otherwise.

When port L pins function as serial clock pins, the value of a pin is inverted when the corresponding bit in PLIR is set to 1.

PLIR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PLnIR	Description	
0	Value is not inverted	(Initial value)
1	Value is inverted	

n = 9 to 7

21.1 Overview

The SH7055SF has 11 ports: A, B, C, D, E, F, G, H, I, J, K and L, all supporting both input and output.

Ports A B, E, F, H, J and K are 16-bit ports, port C is a 5-bit port, ports D and L are 14-bit ports, and port G is a 4-bit port.

All the port pins are multiplexed as general input/output pins and special function pins. The functions of the multiplex pins are selected by means of the pin function controller (PFC). Each port is provided with a data register for storing the pin data.

Each of the ports A, B, D, J and L is provided with a port register to read the pin values.

21.2 Port A

Port A is an input/output port with the 16 pins shown in figure 21.1.

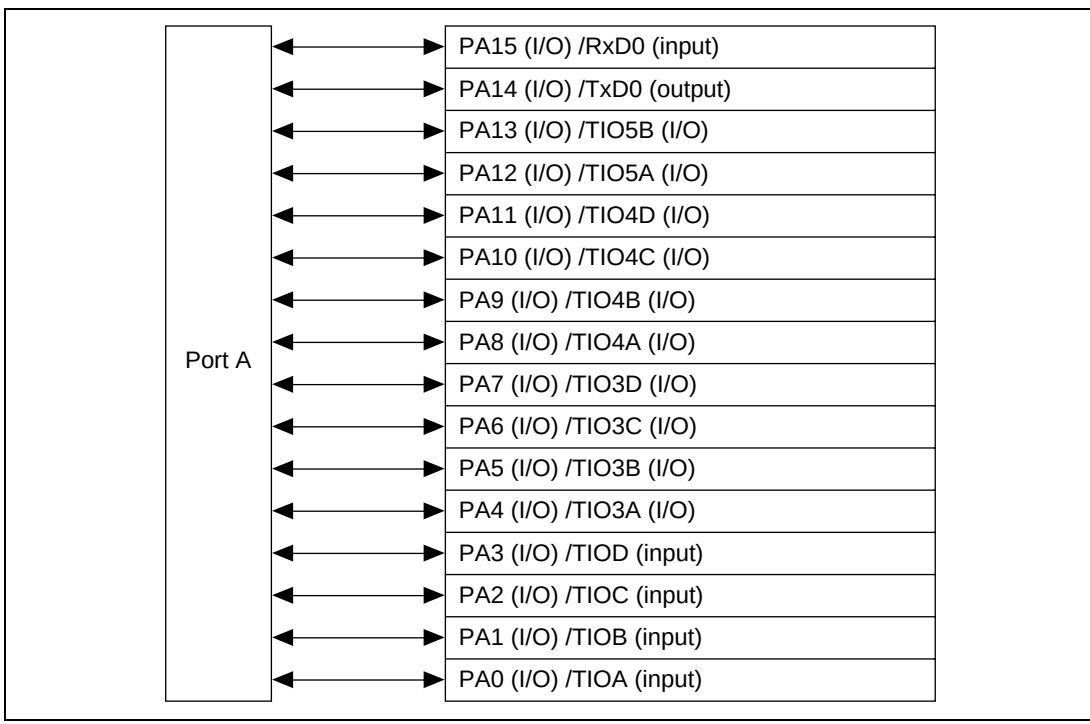


Figure 21.1 Port A

The port A register configuration is shown in table 21.1.

Table 21.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register	PADR	R/W	H'0000	H'FFFFF726	8, 16
Port A port register	PAPR	R	port A pin values	H'FFFFF780	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

21.2.2 Port A Data Register (PADR)

Bit:	15	14	13	12	11	10	9	8
	PA15 DR	PA14 DR	PA13 DR	PA12 DR	PA11 DR	PA10 DR	PA9 DR	PA8 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PA7 DR	PA6 DR	PA5 DR	PA4 DR	PA3 DR	PA2 DR	PA1 DR	PA0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port A data register (PADR) is a 16-bit readable/writable register that stores port A data. Bits PA15DR to PA0DR correspond to pins PA15/RxD0 to PA0/TI0A.

When a pin functions as a general output, if a value is written to PADR, that value is output directly from the pin, and if PADR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PADR is read the pin state, not the register value, is returned directly. If a value is written to PADR, although that value is written into PADR it does not affect the pin state. Table 21.2 summarizes port A data register read/write operations.

PADR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PAIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PADR, but does not affect pin state
	Other than general input	Pin state	Value is written to PADR, but does not affect pin state
1	General output	PADR value	Write value is output from pin
	Other than general output	PADR value	Value is written to PADR, but does not affect pin state

21.2.3 Port A Port Register (PAPR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR	PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * The initial value is 1 when the PA15 to PA0 pins are high, and it is 0 when the pins are low.

The port A port register (PAPR) is a 16-bit read-only register that always stores the value of the port A pins. The CPU cannot write data to this register. Bits PA15PR to PA0PR correspond to pins PA15/RxD0 to PA0/TI0A. If PAPR is read, the corresponding pin values are returned.

- Bits 15 to 0: Port A15 to A0 Port Register (PA15PR to PA0PR)

PA15PR to PA0PR	Description
0	Low-level signals are output from or input to the PA15 to PA0 pins.
1	High-level signals are output from or input to the PA15 to PA0 pins.

Port B is an input/output port with the 16 pins shown in figure 21.2.

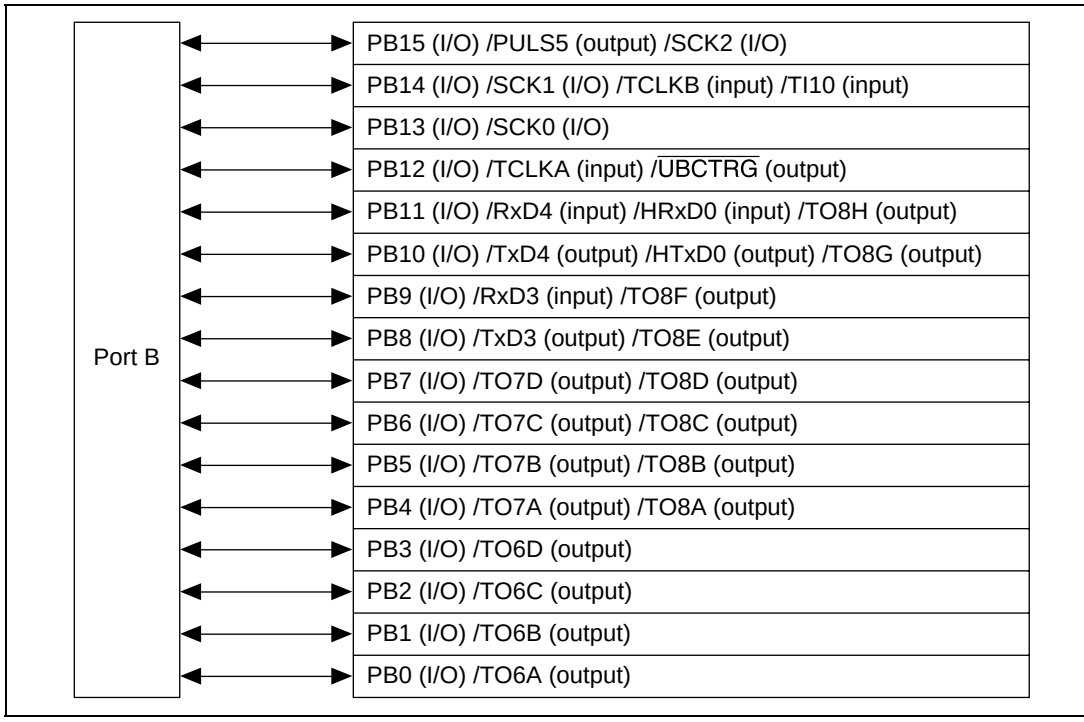


Figure 21.2 Port B

21.3.1 Register Configuration

The port B register configuration is shown in table 21.3.

Table 21.3 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B data register	PBDR	R/W	H'0000	H'FFFFF738	8, 16
Port B port register	PBPR	R	port B pin values	H'FFFFF782	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

Bit:	15	14	13	12	11	10	9	8
	PB15 DR	PB14 DR	PB13 DR	PB12 DR	PB11 DR	PB10 DR	PB9 DR	PB8 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PB7 DR	PB6 DR	PB5 DR	PB4 DR	PB3 DR	PB2 DR	PB1 DR	PB0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port B data register (PBDR) is a 16-bit readable/writable register that stores port B data. Bits PB15DR to PB0DR correspond to pins PB15/PULS5/SCK2 to PB0/TO6A.

When a pin functions as a general output, if a value is written to PBDR, that value is output directly from the pin, and if PBDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PBDR is read the pin state, not the register value, is returned directly. If a value is written to PBDR, although that value is written into PBDR it does not affect the pin state. Table 21.4 summarizes port B data register read/write operations.

PBDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Table 21.4 Port B Data Register (PBDR) Read/Write Operations

Bits 15 to 0:

PBIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PBDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PBDR, but does not affect pin state
1	General output	PBDR value	Write value is output from pin
	Other than general output	PBDR value	Value is written to PBDR, but does not affect pin state

	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * The initial value is 1 when the PB15 to PB0 pins are high, and it is 0 when the pins are low.

The port B port register (PBPR) is a 16-bit read-only register that always stores the value of the port B pins. The CPU cannot write data to this register. Bits PB15PR to PB0PR correspond to pins PB15/PULS5/SCK2 to PB0/TO6A. If PBPR is read, the corresponding pin values are returned.

- Bits 15 to 0: Port B15 to B0 Port Register (PB15PR to PB0PR)

PB15PR to PB0PR	Description
0	Low-level signals are output from or input to the PB15 to PB0 pins.
1	High-level signals are output from or input to the PB15 to PB0 pins.

21.4 Port C

Port C is an input/output port with the 5 pins shown in figure 21.3.

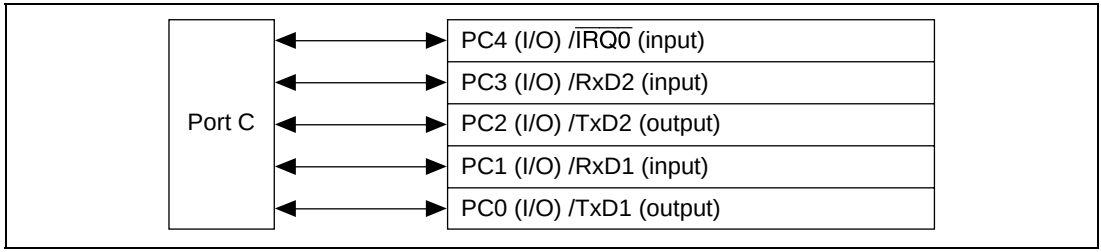


Figure 21.3 Port C

21.4.1 Register Configuration

The port C register configuration is shown in table 21.5.

Table 21.5 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port C data register	PCDR	R/W	H'0000	H'FFFFF73E	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	PC4 DR	PC3 DR	PC2 DR	PC1 DR	PC0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

The port C data register (PCDR) is a 16-bit readable/writable register that stores port C data. Bits PC4DR to PC0DR correspond to pins PC4/$\overline{\text{IRQ0}}$ to PC0/TxD1.

When a pin functions as a general output, if a value is written to PCDR, that value is output directly from the pin, and if PCDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PCDR is read the pin state, not the register value, is returned directly. If a value is written to PCDR, although that value is written into PCDR it does not affect the pin state. Table 21.6 summarizes port C data register read/write operations.

PCDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bits 15 to 5—Reserved: These bits always read 0. The write value should always be 0.

Table 21.6 Port C Data Register (PCDR) Read/Write Operations

Bits 4 to 0:

PCIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PCDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PCDR, but does not affect pin state
1	General output	PCDR value	Write value is output from pin
	Other than general output	PCDR value	Value is written to PCDR, but does not affect pin state

Port D is an input/output port with the 14 pins shown in figure 21.4.

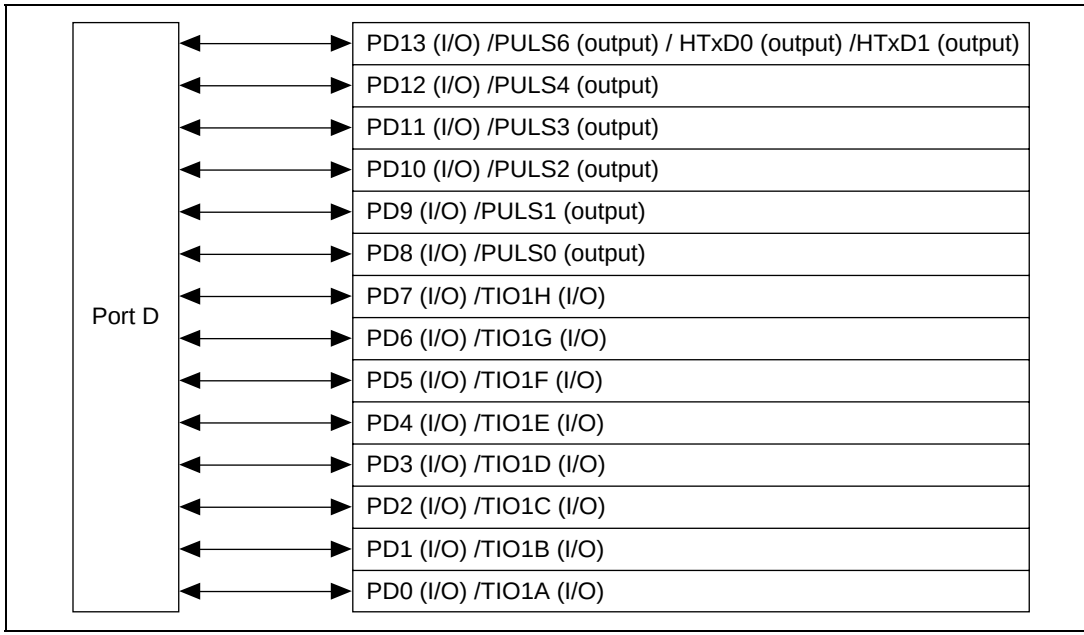


Figure 21.4 Port D

21.5.1 Register Configuration

The port D register configuration is shown in table 21.7.

Table 21.7 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D data register	PDDR	R/W	H'0000	H'FFFFF746	8, 16
Port D port register	PDPR	R	port D pin values	H'FFFFF784	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

Bit:	15	14	13	12	11	10	9	8
	—	—	PD13 DR	PD12 DR	PD11 DR	PD10 DR	PD9 DR	PD8 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PD7 DR	PD6 DR	PD5 DR	PD4 DR	PD3 DR	PD2 DR	PD1 DR	PD0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port D data register (PDDR) is a 16-bit readable/writable register that stores port D data. Bits PD13DR to PD0DR correspond to pins PD13/PULS6/HTxD0/HTxD1 to PD0/TIO1A.

When a pin functions as a general output, if a value is written to PDDR, that value is output directly from the pin, and if PDDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PDDR is read the pin state, not the register value, is returned directly. If a value is written to PDDR, although that value is written into PDDR it does not affect the pin state. Table 21.8 summarizes port D data register read/write operations.

PDDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bits 15 and 14— Reserved: These bits always read 0. The write value should always be 0.

Table 21.8 Port D Data Register (PDDR) Read/Write Operations

Bits 13 to 0:

PDIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PDDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PDDR, but does not affect pin state
1	General output	PDDR value	Write value is output from pin
	Other than general output	PDDR value	Value is written to PDDR, but does not affect pin state

	–	–	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR	PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
Initial value:	0	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

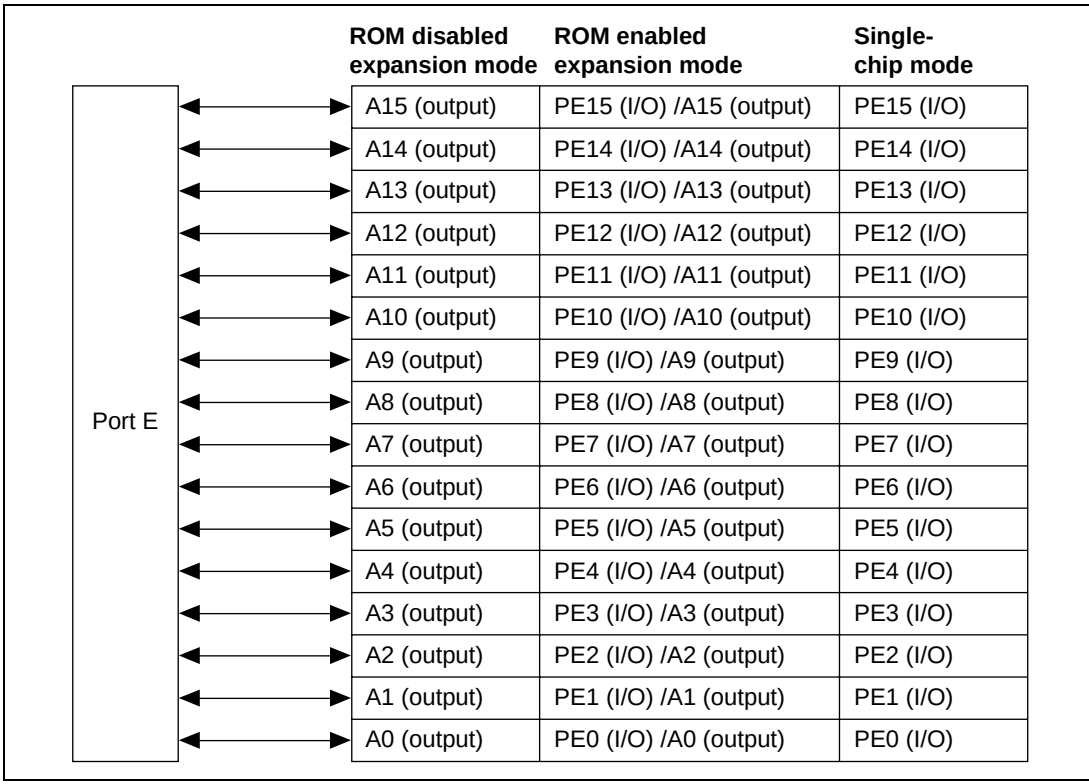
Note: * The initial value is 1 when the PD13 to PD0 pins are high, and it is 0 when the pins are low.

The port D port register (PDPR) is a 16-bit read-only register that always stores the value of the port D pins. The CPU cannot write data to this register. Bits PD13PR to PD0PR correspond to pins PD13/PULS6/HTxD0/HTxD1 to PD0/TIO1A. If PDPR is read, the corresponding pin values are returned.

- Bits 15 and 14: Reserved: These bits are always read as 0.
- Bits 13 to 0: Port D13 to D0 Port Register (PD13PR to PD0PR)

PD13PR to PD0PR	Description
0	Low-level signals are output from or input to the PD13 to PD0 pins.
1	High-level signals are output from or input to the PD13 to PD0 pins.

Port E is an input/output port with the 16 pins shown in figure 21.5.



ROM disabled expansion mode	ROM enabled expansion mode	Single-chip mode
A15 (output)	PE15 (I/O) /A15 (output)	PE15 (I/O)
A14 (output)	PE14 (I/O) /A14 (output)	PE14 (I/O)
A13 (output)	PE13 (I/O) /A13 (output)	PE13 (I/O)
A12 (output)	PE12 (I/O) /A12 (output)	PE12 (I/O)
A11 (output)	PE11 (I/O) /A11 (output)	PE11 (I/O)
A10 (output)	PE10 (I/O) /A10 (output)	PE10 (I/O)
A9 (output)	PE9 (I/O) /A9 (output)	PE9 (I/O)
A8 (output)	PE8 (I/O) /A8 (output)	PE8 (I/O)
A7 (output)	PE7 (I/O) /A7 (output)	PE7 (I/O)
A6 (output)	PE6 (I/O) /A6 (output)	PE6 (I/O)
A5 (output)	PE5 (I/O) /A5 (output)	PE5 (I/O)
A4 (output)	PE4 (I/O) /A4 (output)	PE4 (I/O)
A3 (output)	PE3 (I/O) /A3 (output)	PE3 (I/O)
A2 (output)	PE2 (I/O) /A2 (output)	PE2 (I/O)
A1 (output)	PE1 (I/O) /A1 (output)	PE1 (I/O)
A0 (output)	PE0 (I/O) /A0 (output)	PE0 (I/O)

Figure 21.5 Port E

21.6.1 Register Configuration

The port E register configuration is shown in table 21.9.

Table 21.9 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E data register	PEDR	R/W	H'0000	H'FFFFF754	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

Bit:	15	14	13	12	11	10	9	8
	PE15 DR	PE14 DR	PE13 DR	PE12 DR	PE11 DR	PE10 DR	PE9 DR	PE8 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PE7 DR	PE6 DR	PE5 DR	PE4 DR	PE3 DR	PE2 DR	PE1 DR	PE0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port E data register (PEDR) is a 16-bit readable/writable register that stores port E data. Bits PE15DR to PE0DR correspond to pins PE15/A15 to PE0/A0.

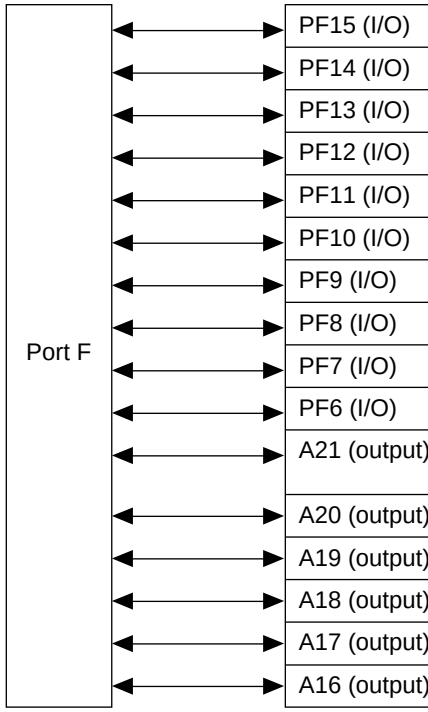
When a pin functions as a general output, if a value is written to PEDR, that value is output directly from the pin, and if PEDR is read, the register value is returned directly regardless of the pin state. When the $\overline{\text{POD}}$ pin is driven low, general outputs go to the high-impedance state regardless of the PEDR value. When the $\overline{\text{POD}}$ pin is driven high, the written value is output from the pin.

When a pin functions as a general input, if PEDR is read the pin state, not the register value, is returned directly. If a value is written to PEDR, although that value is written into PEDR it does not affect the pin state. Table 21.10 summarizes port E data register read/write operations.

PEDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PEIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PEDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PEDR, but does not affect pin state
1	General output	PEDR value	Write value is output from pin ($\overline{\text{POD}}$ pin = high)
			High impedance regardless of PEDR value ($\overline{\text{POD}}$ pin = low)
	Other than general output	PEDR value	Value is written to PEDR, but does not affect pin state

Port F is an input/output port with the 16 pins shown in figure 21.6.



	ROM disabled expansion mode	ROM enabled expansion mode	Single-chip mode
	PF15 (I/O)	$\overline{\text{BREQ}}$ (input)	PF15 (I/O)
	PF14 (I/O)	$\overline{\text{BACK}}$ (output)	PF14 (I/O)
	PF13 (I/O)	$\overline{\text{CS3}}$ (output)	PF13 (I/O)
	PF12 (I/O)	$\overline{\text{CS2}}$ (output)	PF12 (I/O)
	PF11 (I/O)	$\overline{\text{CS1}}$ (output)	PF11 (I/O)
	PF10 (I/O)	$\overline{\text{CS0}}$ (output)	PF10 (I/O)
	PF9 (I/O)	$\overline{\text{RD}}$ (output)	PF9 (I/O)
	PF8 (I/O)	$\overline{\text{WAIT}}$ (input)	PF8 (I/O)
	PF7 (I/O)	$\overline{\text{WRH}}$ (output)	PF7 (I/O)
	PF6 (I/O)	$\overline{\text{WRL}}$ (output)	PF6 (I/O)
	A21 (output)	PF5 (I/O) /A21 (output) / $\overline{\text{POD}}$ (input)	PF5 (I/O) / $\overline{\text{POD}}$ (input)
	A20 (output)	PF4 (I/O) /A20 (output)	PF4 (I/O)
	A19 (output)	PF3 (I/O) /A19 (output)	PF3 (I/O)
	A18 (output)	PF2 (I/O) /A18 (output)	PF2 (I/O)
	A17 (output)	PF1 (I/O) /A17 (output)	PF1 (I/O)
	A16 (output)	PF0 (I/O) /A16 (output)	PF0 (I/O)

Figure 21.6 Port F

21.7.1 Register Configuration

The port F register configuration is shown in table 21.11.

Table 21.11 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register	PFDR	R/W	H'0000	H'FFFFF74E	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

Bit:	15	14	13	12	11	10	9	8
	PF15 DR	PF14 DR	PF13 DR	PF12 DR	PF11 DR	PF10 DR	PF9 DR	PF8 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PF7 DR	PF6 DR	PF5 DR	PF4 DR	PF3 DR	PF2 DR	PF1 DR	PF0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port F data register (PFDR) is a 16-bit readable/writable register that stores port F data. Bits PF15DR to PF0DR correspond to pins PF15/$\overline{\text{BREQ}}$ to PF0/A16.

When a pin functions as a general output, if a value is written to PFDR, that value is output directly from the pin, and if PFDR is read, the register value is returned directly regardless of the pin state. For pins PF0 to PF4, when the $\overline{\text{POD}}$ pin is driven low, general outputs go to the high-impedance state regardless of the PFDR value. When the $\overline{\text{POD}}$ pin is driven high, the written value is output from the pin.

When a pin functions as a general input, if PFDR is read the pin state, not the register value, is returned directly. If a value is written to PFDR, although that value is written into PFDR it does not affect the pin state. Table 21.12 summarizes port F data register read/write operations.

PFDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PFIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PFDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PFDR, but does not affect pin state
1	General output	PFDR value	Write value is output from pin
	Other than general output	PFDR value	Value is written to PFDR, but does not affect pin state

Bits 4–0:

PFIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PFDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PFDR, but does not affect pin state
1	General output	PFDR value	Write value is output from pin ($\overline{\text{POD}}$ pin = high)
			High impedance regardless of PFDR value ($\overline{\text{POD}}$ pin = low)
	Other than general output	PFDR value	Value is written to PFDR, but does not affect pin state

21.8 Port G

Port G is an input/output port with the 4 pins shown in figure 21.7.

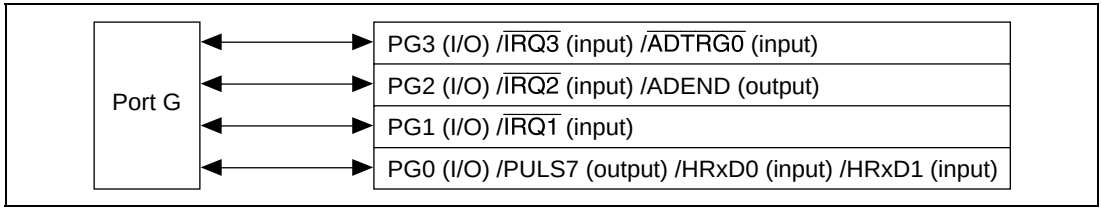


Figure 21.7 Port G

The port G register configuration is shown in table 21.13.

Table 21.13 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port G data register	PGDR	R/W	H'0000	H'FFFFF764	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

21.8.2 Port G Data Register (PGDR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	PG3 DR	PG2 DR	PG1 DR	PG0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

The port G data register (PGDR) is a 16-bit readable/writable register that stores port G data. Bits PG3DR to PG0DR correspond to pins PG3/$\overline{\text{IRQ3}}$/$\overline{\text{ADTRG0}}$ to PG0/PULS7/HRxD0/HRxD1.

When a pin functions as a general output, if a value is written to PGDR, that value is output directly from the pin, and if PGDR is read, the register value is returned directly regardless of the pin state.

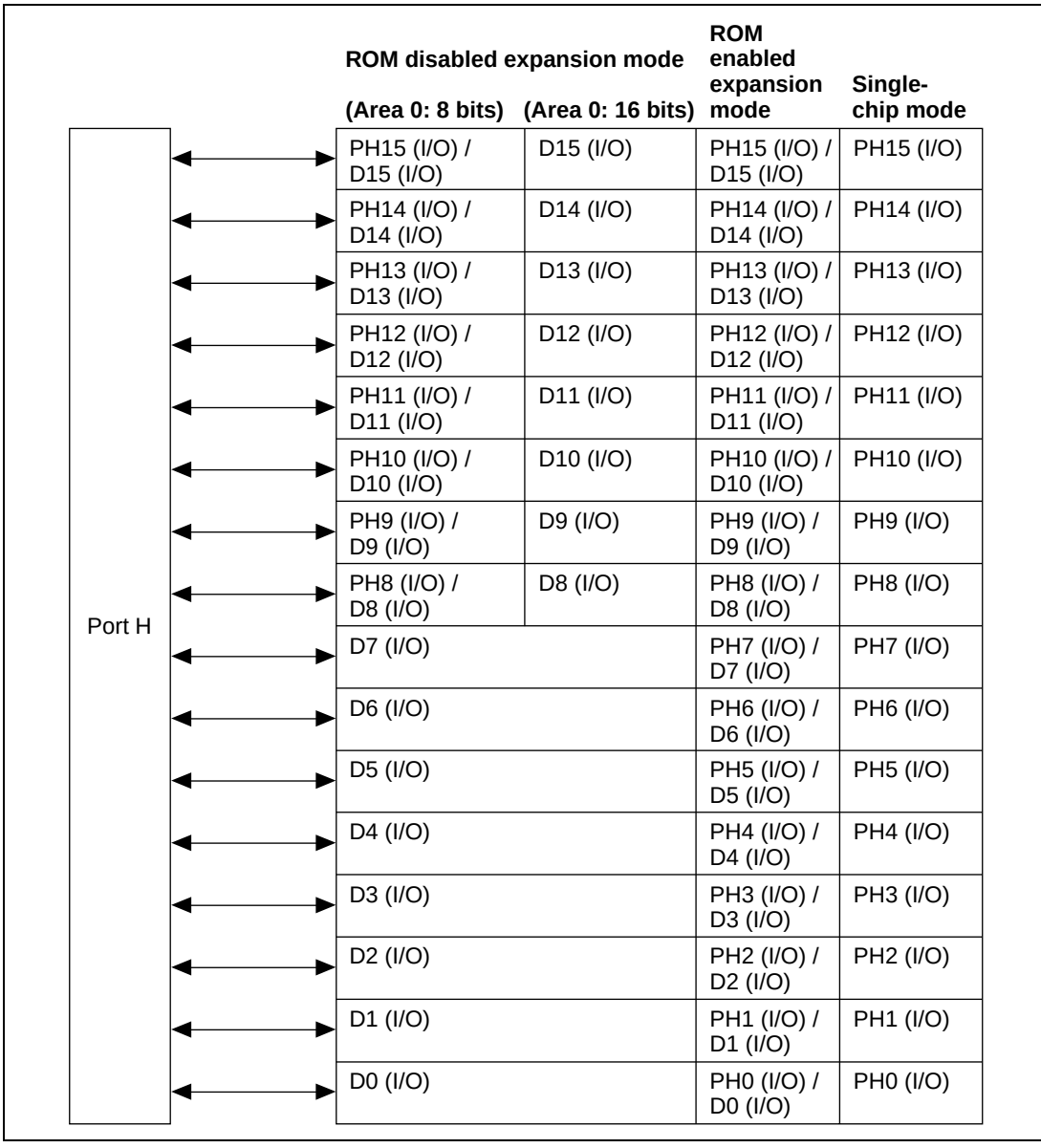
When a pin functions as a general input, if PGDR is read the pin state, not the register value, is returned directly. If a value is written to PGDR, although that value is written into PGDR it does not affect the pin state. Table 21.14 summarizes port G data register read/write operations.

PGDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bits 15 to 4—Reserved: These bits always read 0. The write value should always be 0.

PGIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PGDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PGDR, but does not affect pin state
1	General output	PGDR value	Write value is output from pin
	Other than general output	PGDR value	Value is written to PGDR, but does not affect pin state

Port H is an input/output port with the 16 pins shown in figure 21.8.



ROM disabled expansion mode (Area 0: 8 bits)	ROM disabled expansion mode (Area 0: 16 bits)	ROM enabled expansion mode	Single-chip mode
PH15 (I/O) / D15 (I/O)	D15 (I/O)	PH15 (I/O) / D15 (I/O)	PH15 (I/O)
PH14 (I/O) / D14 (I/O)	D14 (I/O)	PH14 (I/O) / D14 (I/O)	PH14 (I/O)
PH13 (I/O) / D13 (I/O)	D13 (I/O)	PH13 (I/O) / D13 (I/O)	PH13 (I/O)
PH12 (I/O) / D12 (I/O)	D12 (I/O)	PH12 (I/O) / D12 (I/O)	PH12 (I/O)
PH11 (I/O) / D11 (I/O)	D11 (I/O)	PH11 (I/O) / D11 (I/O)	PH11 (I/O)
PH10 (I/O) / D10 (I/O)	D10 (I/O)	PH10 (I/O) / D10 (I/O)	PH10 (I/O)
PH9 (I/O) / D9 (I/O)	D9 (I/O)	PH9 (I/O) / D9 (I/O)	PH9 (I/O)
PH8 (I/O) / D8 (I/O)	D8 (I/O)	PH8 (I/O) / D8 (I/O)	PH8 (I/O)
D7 (I/O)		PH7 (I/O) / D7 (I/O)	PH7 (I/O)
D6 (I/O)		PH6 (I/O) / D6 (I/O)	PH6 (I/O)
D5 (I/O)		PH5 (I/O) / D5 (I/O)	PH5 (I/O)
D4 (I/O)		PH4 (I/O) / D4 (I/O)	PH4 (I/O)
D3 (I/O)		PH3 (I/O) / D3 (I/O)	PH3 (I/O)
D2 (I/O)		PH2 (I/O) / D2 (I/O)	PH2 (I/O)
D1 (I/O)		PH1 (I/O) / D1 (I/O)	PH1 (I/O)
D0 (I/O)		PH0 (I/O) / D0 (I/O)	PH0 (I/O)

Figure 21.8 Port H

The port H register configuration is shown in table 21.15.

Table 21.15 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port H data register	PHDR	R/W	H'0000	H'FFFFF72C	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

21.9.2 Port H Data Register (PHDR)

Bit:	15	14	13	12	11	10	9	8
	PH15 DR	PH14 DR	PH13 DR	PH12 DR	PH11 DR	PH10 DR	PH9 DR	PH8 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PH7 DR	PH6 DR	PH5 DR	PH4 DR	PH3 DR	PH2 DR	PH1 DR	PH0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port H data register (PHDR) is a 16-bit readable/writable register that stores port H data. Bits PH15DR to PH0DR correspond to pins PH15/D15 to PH0/D0.

When a pin functions as a general output, if a value is written to PHDR, that value is output directly from the pin, and if PHDR is read, the register value is returned directly regardless of the pin state. When the $\overline{\text{POD}}$ pin is driven low, general outputs go to the high-impedance state regardless of the PHDR value. When the $\overline{\text{POD}}$ pin is driven high, the written value is output from the pin.

When a pin functions as a general input, if PHDR is read the pin state, not the register value, is returned directly. If a value is written to PHDR, although that value is written into PHDR it does not affect the pin state. Table 21.16 summarizes port H data register read/write operations.

PHDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PHIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PHDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PHDR, but does not affect pin state
1	General output	PHDR value	Write value is output from pin ($\overline{\text{POD}}$ pin = high)
			High impedance regardless of PHDR value ($\overline{\text{POD}}$ pin = low)
	Other than general output	PHDR value	Value is written to PHDR, but does not affect pin state

21.10 Port J

Port J is an input/output port with the 16 pins shown in figure 21.9.

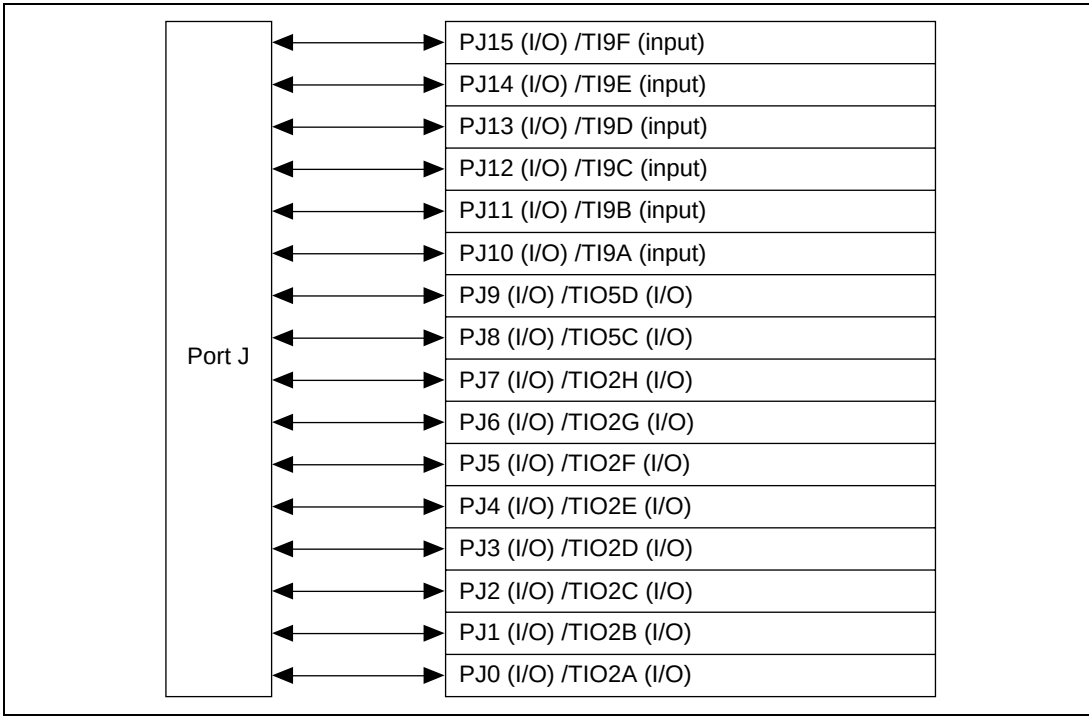


Figure 21.9 Port J

The port J register configuration is shown in table 21.17.

Table 21.17 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port J data register	PJDR	R/W	H'0000	H'FFFFF76C	8, 16
Port J port register	PJPR	R	port J pin values	H'FFFFF786	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

21.10.2 Port J Data Register (PJDR)

Bit:	15	14	13	12	11	10	9	8
	PJ15 DR	PJ14 DR	PJ13 DR	PJ12 DR	PJ11 DR	PJ10 DR	PJ9 DR	PJ8 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PJ7 DR	PJ6 DR	PJ5 DR	PJ4 DR	PJ3 DR	PJ2 DR	PJ1 DR	PJ0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port J data register (PJDR) is a 16-bit readable/writable register that stores port J data. Bits PJ15DR to PJ0DR correspond to pins PJ15/TI9F to PJ0/TIO2A.

When a pin functions as a general output, if a value is written to PJDR, that value is output directly from the pin, and if PJDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PJDR is read the pin state, not the register value, is returned directly. If a value is written to PJDR, although that value is written into PJDR it does not affect the pin state. Table 21.18 summarizes port J data register read/write operations.

PJDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PJIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PJDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PJDR, but does not affect pin state
1	General output	PJDR value	Write value is output from pin
	Other than general output	PJDR value	Value is written to PJDR, but does not affect pin state

21.10.3 Port J Port Register (PJPR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ15PR	PJ14PR	PJ13PR	PJ12PR	PJ11PR	PJ10PR	PJ9PR	PJ8PR	PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * The initial value is 1 when the PJ15 to PJ0 pins are high, and it is 0 when the pins are low.

The port J port register (PJPR) is a 16-bit read-only register that always stores the value of the port J pins. The CPU cannot write data to this register. Bits PJ15PR to PJ0PR correspond to pins PJ15/TI9F to PJ0/TIO2A. If PJPR is read, the corresponding pin values are returned.

- Bits 15 to 0: Port J15 to J0 Port Register (PJ15PR to PJ0PR)

PJ15PR to PJ0PR	Description
0	Low-level signals are output from or input to the PJ15 to PJ0 pins.
1	High-level signals are output from or input to the PJ15 to PJ0 pins.

Port K is an input/output port with the 16 pins shown in figure 21.10.

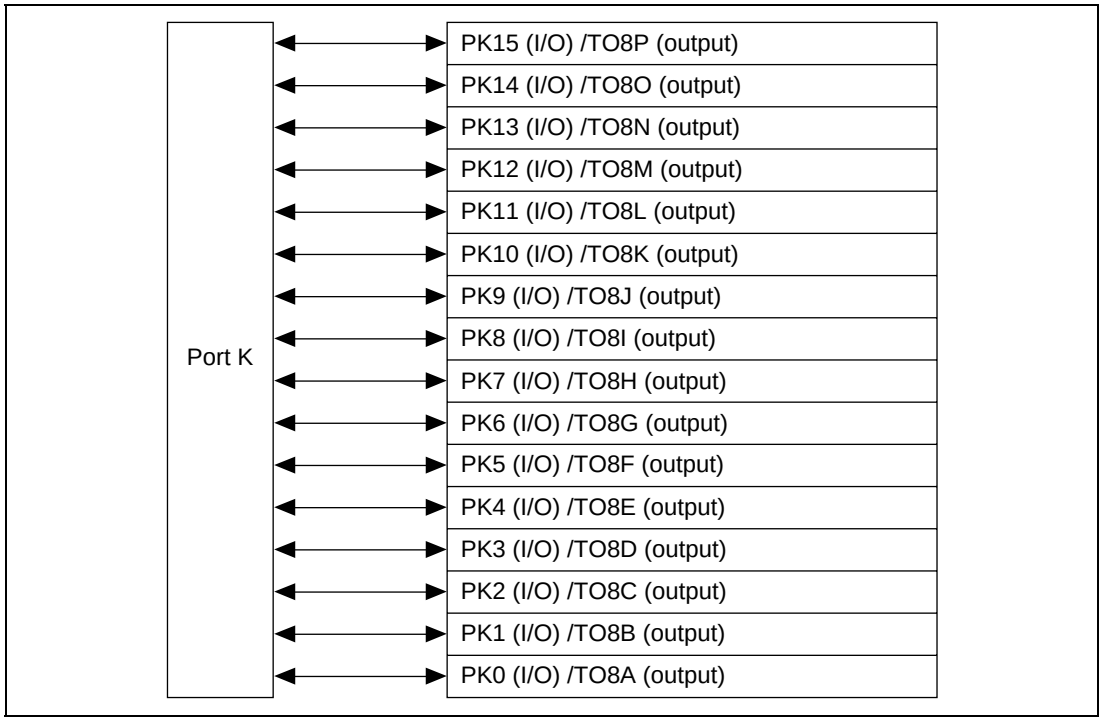


Figure 21.10 Port K

21.11.1 Register Configuration

The port K register configuration is shown in table 21.19.

Table 21.19 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port K data register	PKDR	R/W	H'0000	H'FFFFF778	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

Bit:	15	14	13	12	11	10	9	8
	PK15 DR	PK14 DR	PK13 DR	PK12 DR	PK11 DR	PK10 DR	PK9 DR	PK8 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PK7 DR	PK6 DR	PK5 DR	PK4 DR	PK3 DR	PK2 DR	PK1 DR	PK0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port K data register (PKDR) is a 16-bit readable/writable register that stores port K data. Bits PK15DR to PK0DR correspond to pins PK15/TO8P to PK0/TO8A.

When a pin functions as a general output, if a value is written to PKDR, that value is output directly from the pin, and if PKDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PKDR is read the pin state, not the register value, is returned directly. If a value is written to PKDR, although that value is written into PKDR it does not affect the pin state. Table 21.20 summarizes port K data register read/write operations.

PKDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Table 21.20 Port K Data Register (PKDR) Read/Write Operations

Bits 15 to 0:

PKIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PKDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PKDR, but does not affect pin state
1	General output	PKDR value	Write value is output from pin
	Other than general output	PKDR value	Value is written to PKDR, but does not affect pin state

Port L is an input/output port with the 14 pins shown in figure 21.11.

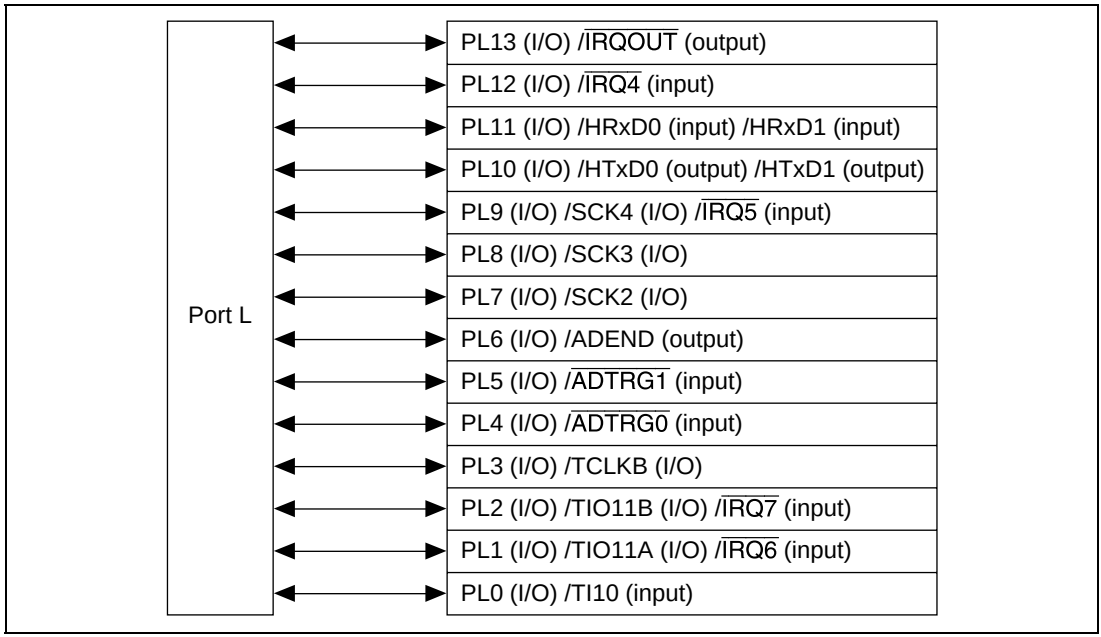


Figure 21.11 Port L

21.12.1 Register Configuration

The port L register configuration is shown in table 21.21.

Table 21.21 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port L data register	PLDR	R/W	H'0000	H'FFFFF75E	8, 16
Port L port register	PLPR	R	port L pin values	H'FFFFF788	8, 16

Note: A register access is performed in four or five cycles regardless of the access size.

Bit:	15	14	13	12	11	10	9	8
	—	—	PL13 DR	PL12 DR	PL11 DR	PL10 DR	PL9 DR	PL8 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PL7 DR	PL6 DR	PL5 DR	PL4 DR	PL3 DR	PL2 DR	PL1 DR	PL0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port L data register (PLDR) is a 16-bit readable/writable register that stores port L data. Bits PL13DR to PL0DR correspond to pins PL13/$\overline{\text{IRQOUT}}$ to PL0/TI10.

When a pin functions as a general output, if a value is written to PLDR, that value is output directly from the pin, and if PLDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PLDR is read the pin state, not the register value, is returned directly. If a value is written to PLDR, although that value is written into PLDR it does not affect the pin state. Table 21.22 summarizes port L data register read/write operations.

PLDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bits 15 and 14—Reserved: These bits always read 0. The write value should always be 0.

Table 21.22 Port L Data Register (PLDR) Read/Write Operations

Bits 13 to 0:

PLIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PLDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PLDR, but does not affect pin state
1	General output	PLDR value	Write value is output from pin
	Other than general output	PLDR value	Value is written to PLDR, but does not affect pin state

	–	–	PL13 PR	PL12 PR	PL11 PR	PL10 PR	PL9 PR	PL8 PR	PL7 PR	PL6 PR	PL5 PR	PL4 PR	PL3 PR	PL2 PR	PL1 PR	PL0 PR
Initial value:	0	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * The initial value is 1 when the PL13 to PL0 pins are high, and it is 0 when the pins are low.

The port L port register (PLPR) is a 16-bit read-only register that always stores the value of the port L pins. The CPU cannot write data to this register. Bits PL13PR to PL0PR correspond to pins PL13/$\overline{\text{IRQOUT}}$ to PL0/TI10. If PLPR is read, the corresponding pin values are returned.

- Bits 15 and 14: Reserved: These bits are always read as 0.
- Bits 13 to 0: Port L13 to L0 Port Register (PL13PR to PL0PR)

PL13PR to PL0PR	Description
0	Low-level signals are output from or input to the PL13 to PL0 pins.
1	High-level signals are output from or input to the PL13 to PL0 pins.

21.13 POD (Port Output Disable) Control

The output port drive buffers for the address bus pins (A20 to A0) and data bus pins (D15 to D0) can be controlled by the $\overline{\text{POD}}$ (port output disable) pin input level. However, this function is enabled only when the address bus pins (A20 to A0) and data bus pins (D15 to D0) are designated as general output ports.

Output buffer control by means of $\overline{\text{POD}}$ is performed asynchronously from bus cycles.

$\overline{\text{POD}}$	Address Bus Pins (A20 to A0) and Data Bus Pins (D15 to D0) (when designated as output ports)
0	Enabled (high-impedance)
1	Disabled (general output)

(1) Table 21.23 lists the differences between the SH7055F and the SH7055SF.

Table 21.23 Differences between the SH7055F and the SH7055SF

	Item	SH7055F	SH7055SF	Notes
PFC	Bits 7, 5, 3 and 1 of PACRL: Reserved	These bits are read as 0 after 1 is written.	These bits are read as 1 after 1 is written. See 20.3.2(2).	Only 0 should be written to these bits.
I/O ports	PAPR PBPR PDPR PJPR PLPR	–	If a port registers is read, the pin values are always returned regardless of the setting of other registers. See sections 21.2.3, 21.3.3, 21.5.3, 21.10.3, 21.12.3.	The pin values can be read from the port registers when the I/O pins of the ATU-II and SCI are designated as outputs.*
Electrical Characteristics	DC Characteristics PG0,PL11	V_{IH}: 2.2V(min) V_{IL}: 0.8V(max)	V_{IH}: PVcc2×0.7V(min) V_{IL}: PVcc2×0.3V(max) See section 25.4.	HCAN port characteristics

Note: * The pin values cannot be read from the data registers.

(2) When port pins do not function as I/O pins, the input/output direction of the pins is selected by the port control registers. (For the PJ15 to 10 pins and PA4 to PA0 pins, the IO register must also be specified.)

22.1 Features

This LSI has 512-kbyte on-chip flash memory. The flash memory has the following features.

- Two flash-memory MATs according to LSI initiation mode

  The on-chip flash memory has two memory spaces in the same address space (hereafter referred to as memory MATs). The mode setting in the initiation determines which memory MAT is initiated first. The MAT can be switched by using the bank-switching method after initiation.

  — The user MAT is initiated at a power-on reset in user mode: 512 kbytes

  — The user boot MAT is initiated at a power-on reset in user boot mode: 8 kbytes
- Three on-board programming modes and one off-board programming mode

  — On-board programming modes

  Boot Mode: This mode is a program mode that uses an on-chip SCI interface. The user MAT and user boot MAT can be programmed. This mode can automatically adjust the bit rate between the host and this LSI.

  User Program Mode: The user MAT can be programmed by using the optional interface.

  User Boot Mode: The user boot program of the optional interface can be made and the user MAT can be programmed.

  — Off-board programming mode

  Programmer Mode: This mode uses the PROM programmer. The user MAT and user boot MAT can be programmed.
- Programming/erasing interface by the download of on-chip program

  This LSI has a dedicated programming/erasing program. After downloading this program to the on-chip RAM, programming/erasing can be performed by setting the argument parameter. The user branch is also supported.

  — User branch

  The program processing is performed in 128-byte units. It consists the program pulse application, verify read, and several other steps. Erasing is performed in one divided-block units and consists of several steps. The user processing routine can be executed between the steps, this setting for which is called the user branch addition.
- Emulation function of flash memory by using the on-chip RAM

  As flash memory is overlapped with part of the on-chip RAM, the flash memory programming can be emulated in real time.
- Protection modes

  There are two protection modes. Software protection by the register setting and hardware protection by the FWE pin. The protection state for flash memory programming/erasing can be set.

- Programming/erasing time
  The flash memory programming time is t_p ms (typ) in 128-byte simultaneous programming and t_p/128 ms per byte. The erasing time is t_E s (typ) per block.
- Number of programming
  The number of flash memory programming can be up to N_{WEC} times.

22.2.1 Block Diagram

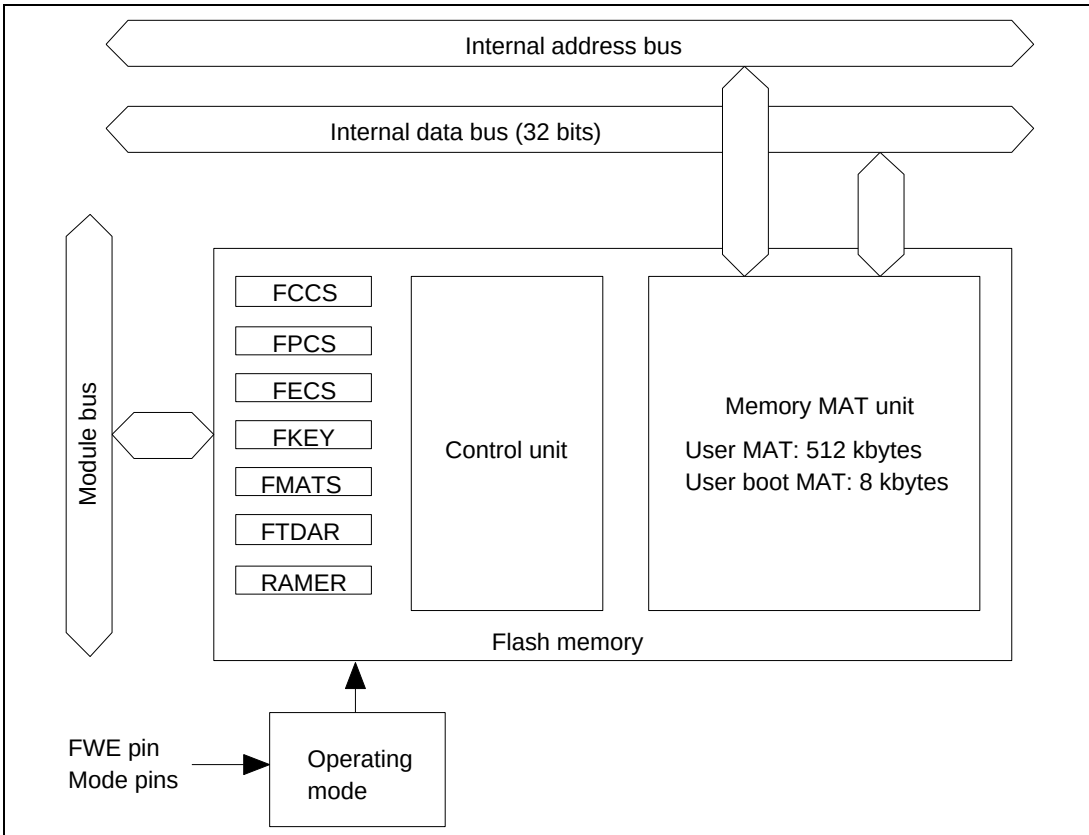


Figure 22.1 Block Diagram of Flash Memory

When each mode pin and the FWE pin are set in the reset state and the reset signal is released, the microcomputer enters each operating mode as shown in figure 22.2. For the setting of each mode pin and the FWE pin, see table 22.1.

- Flash memory cannot be read, programmed, or erased in ROM invalid mode. The programming/erasing interface registers cannot be written to. When these registers are read, H'00 is always read.
- Flash memory can be read in user mode, but cannot be programmed or erased.
- Flash memory can be read, programmed, or erased on the board only in user program mode, user boot mode, and boot mode.
- Flash memory can be read, programmed, or erased by means of the PROM programmer in programmer mode.

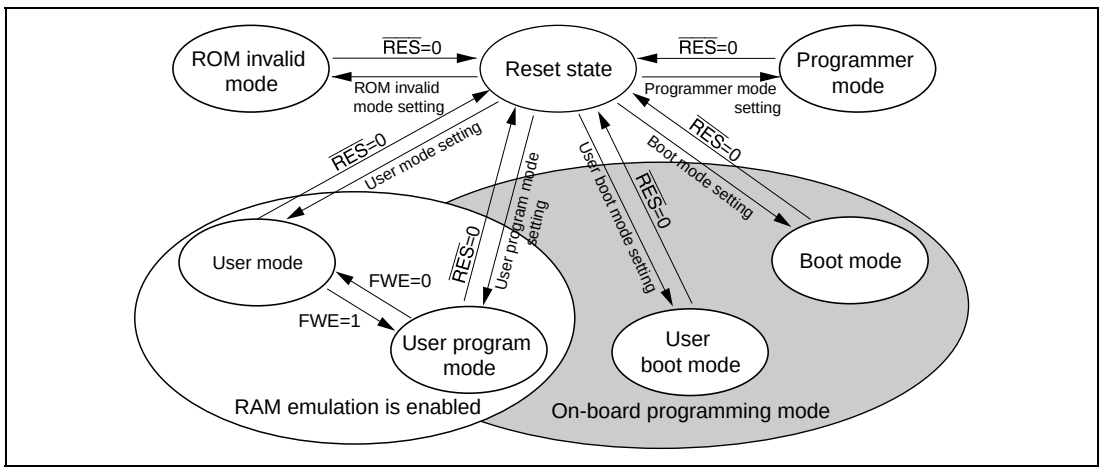


Figure 22.2 Mode Transition of Flash Memory

Pin	Reset State	ROM Invalid Mode	ROM Valid Mode	User Program Mode	User Boot Mode	Boot Mode	Programmer Mode
$\overline{\text{RES}}$	0	1	1	1	1	1	1
FWE	0/1	0	0	1	1	1	0/1
MD0	0/1	0/1*[1]	0/1*[2]	0/1*[2]	0/1*[2]	0/1*[2]	1
MD1	0/1	0	1	1	0	0	1
MD2	0/1	1	1	1	0	1	0

Notes: *1 MD0 = 0: 8-bit external bus, MD0 = 1: 16-bit external bus

*2 MD0 = 0: External bus can be used, MD0 = 1: Single-chip mode (external bus cannot be used)

22.2.3 Mode Comparison

The comparison table of programming and erasing related items about boot mode, user program mode, user boot mode, and programmer mode is shown in table 22.2.

	Boot Mode	Mode	User Boot Mode	Mode
Programming/ erasing environment	On-board programming	On-board programming	On-board programming	Off-board programming
Programming/ erasing enable MAT	User MAT User boot MAT	User MAT	User MAT	User MAT User boot MAT
Programming/ erasing control	Command method	Programming/ erasing interface	Programming/ erasing interface	Command method
All erasure	O (Automatic)	O	O	O (Automatic)
Block division erasure	O*[1]	O	O	X
Program data transfer	From host via SCI	From optional device via RAM	From optional device via RAM	Via programmer
User branch function	X	O	O	X
RAM emulation	X	O	X	X
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT*[2]	Embedded program storage MAT
Transition to user mode	Mode setting change and reset	FWE setting change	Mode setting change and reset	—

Notes: *1 All-erasure is performed. After that, the specified block can be erased.
*2 Initiation starts from the embedded program storage MAT. After checking the flash-memory related registers, initiation starts from the reset vector of the user MAT.

- The user boot MAT can be programmed or erased only in boot mode and programmer mode.
- The user MAT and user boot MAT are all erased in boot mode. Then, the user MAT and user boot MAT can be programmed by means of the command method. However, the contents of the MAT cannot be read until this state.
  Only user boot MAT is programmed and the user MAT is programmed in user boot mode or only user MAT is programmed because user boot mode is not used.
- In user boot mode, the boot operation of the optional interface can be performed by a mode pin setting different from user program mode.

This LSI's flash memory is configured by the 512-kbyte user MAT and 8-kbyte user boot MAT.

The start address is allocated to the same address in the user MAT and user boot MAT. Therefore, when the program execution or data access is performed between the two MATs, the MAT must be switched by using FMATS. The user MAT is divided into two 512-kbyte banks (bank 0 and bank 1).

The user MAT or user boot MAT can be read in all modes if it is in ROM valid mode. However, the user boot MAT can be programmed only in boot mode and programmer mode.

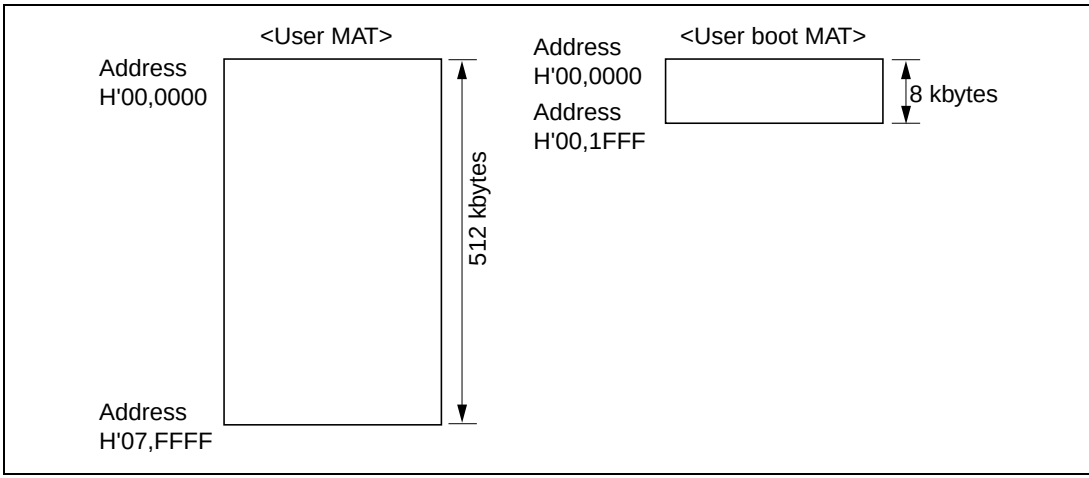
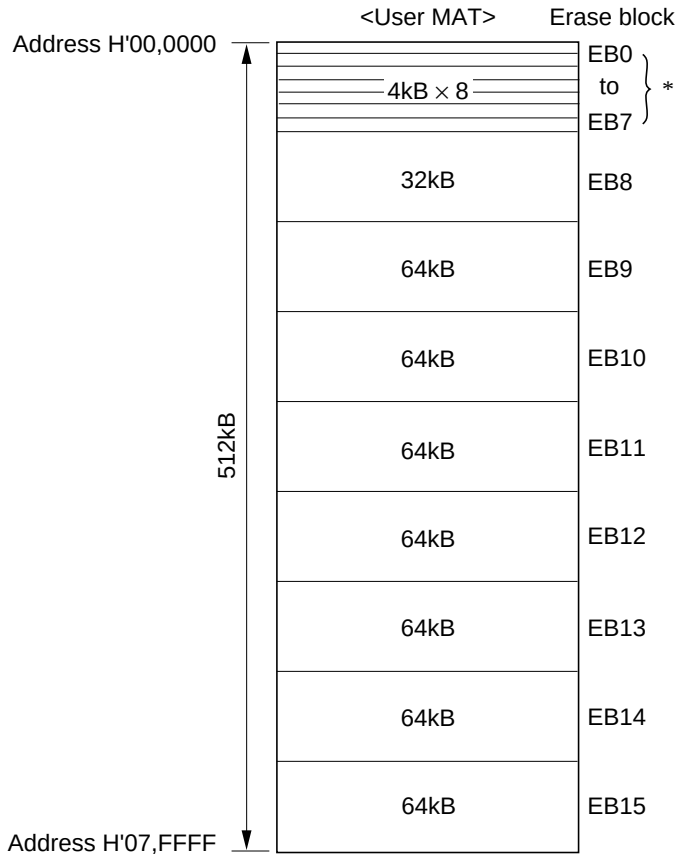


Figure 22.3 Flash Memory Configuration

The user MAT and user boot MAT have different memory sizes. Do not access a user boot MAT that is 8 kbytes or more. When a user boot MAT exceeding 8 kbytes is read from, an undefined value is read.

The user MAT is divided into 64 kbytes (seven blocks), 32 kbytes (one block), and 4 kbytes (eight blocks) as shown in figure 22.4. The user MAT can be erased in this divided-block units and the erase-block number of EB0 to EB15 is specified when erasing.

The RAM emulation can be performed in the eight blocks of 4 kbytes.



Note: * RAM emulation can be performed in the eight blocks of 4 kbytes.

Figure 22.4 Block Division of User MAT

Programming/erasing is executed by downloading the on-chip program to the on-chip RAM and specifying the program address/data and erase block by using the interface registers/parameters.

The procedure program is made by the user in user program mode and user boot mode. The overview of the procedure is as follows. For details, see section 22.5.2, User Program Mode.

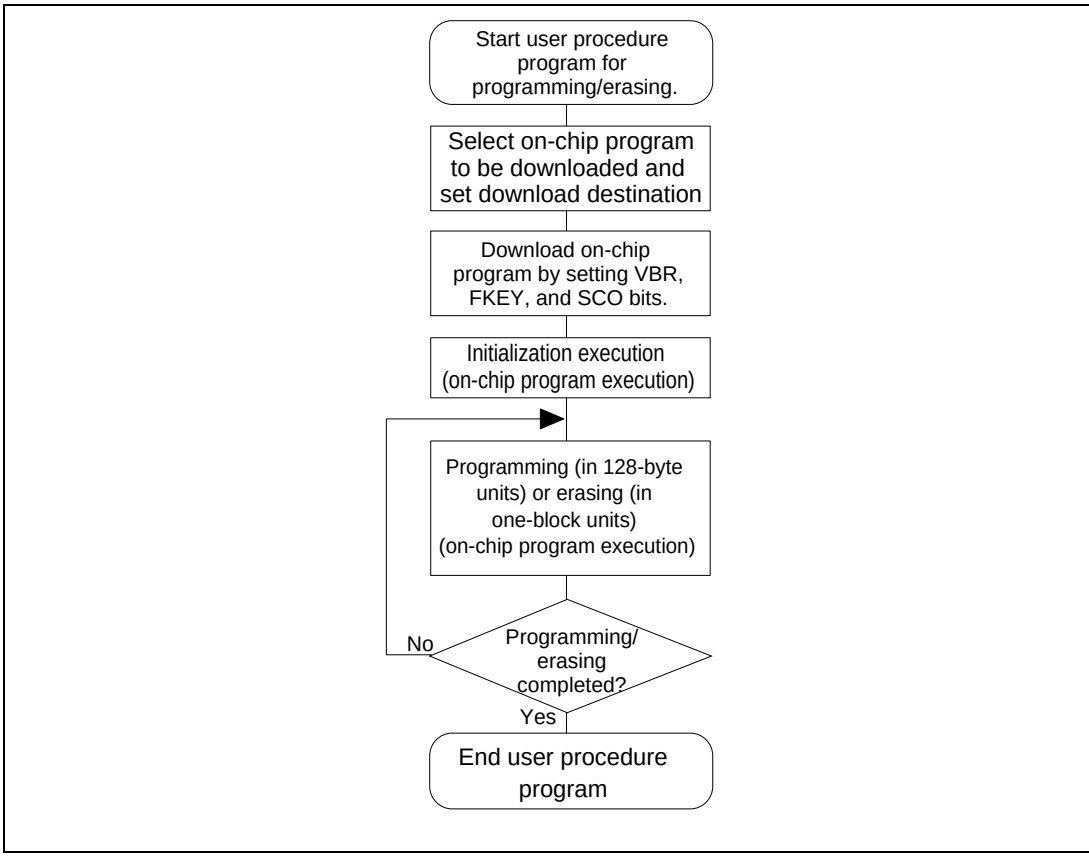


Figure 22.5 Overview of User Procedure Program

(1) Selection of On-Chip Program to be Downloaded and Setting of Download Destination

This LSI has programming/erasing programs and they can be downloaded to the on-chip RAM. The on-chip program to be downloaded is selected by setting the corresponding bits in the programming/erasing interface registers. The download destination can be specified by FTDAR.

H'00000000 and then setting the SCO bit in the flash key code register (FKEY) and the flash code control and status register (FCCS), which are programming/erasing interface registers.

The user MAT is replaced to the embedded program storage area when downloading. Since the flash memory cannot be read when programming/erasing, the procedure program, which is working from download to completion of programming/erasing, must be executed in a space other than the flash memory to be programmed/erased (for example, on-chip RAM).

Since the result of download is returned to the programming/erasing interface parameters, whether the normal download is executed or not can be confirmed.

Note that VBR can be changed after download is completed.

(3) Initialization of Programming/Erasing

The operating frequency and user branch are set before execution of programming/erasing. The user branch destination must be in an area other than the on-chip flash memory area and the area where the on-chip program is downloaded. These settings are performed by using the programming/erasing interface parameters.

(4) Programming/Erasing Execution

To program or erase, the FWE pin must be brought high and user program mode must be entered.

The program data/programming destination address is specified in 128-byte units when programming.

The block to be erased is specified in a erase-block unit when erasing.

These specifications are set by using the programming/erasing interface parameters and the on-chip program is initiated. The on-chip program is executed by using the JSR or BSR instruction to perform the subroutine call of the specified address in the on-chip RAM. The execution result is returned to the programming/erasing interface parameters.

The area to be programmed must be erased in advance when programming flash memory.

There are limitations and notes on the interrupt processing during programming/erasing. For details, see section 22.8.2, Interrupts during Programming/Erasing.

(5) When Programming/Erasing is Executed Consecutively

When the processing is not ended by the 128-byte programming or one-block erasure, the program address/data and erase-block number must be updated and consecutive programming/erasing is required.

Since the downloaded on-chip program is left in the on-chip RAM after the processing, download and initialization are not required when the same processing is executed consecutively.

Flash memory is controlled by the pins as shown in table 22.3.

Table 22.3 Pin Configuration

Pin Name	Abbreviation	Input/Output	Function
Power-on reset	$\overline{\text{RES}}$	Input	Reset
Flash programming enable	FWE	Input	Hardware protection when programming flash memory
Mode 2	MD2	Input	Sets operating mode of this LSI
Mode 1	MD1	Input	Sets operating mode of this LSI
Mode 0	MD0	Input	Sets operating mode of this LSI
Transmit data	TxD1	Output	Serial transmit data output (used in boot mode)
Receive data	RxD1	Input	Serial receive data input (used in boot mode)

Note: For the pin configuration in programmer mode, see section 22.9, Programmer Mode.

22.4.1 Registers

The registers/parameters which control flash memory when the on-chip flash memory is valid are shown in table 22.4.

There are several operating modes for accessing flash memory, for example, read mode/program mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/parameters are allocated for each operating mode and MAT selection. The correspondence of operating modes and registers/parameters for use is shown in table 22.5.

Table 22.4 (1) Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Flash code control status register	FCCS	R, W*[1]	H'00*[2] H'80*[2]	H'FFFFE800	8
Flash program code select register	FPCS	R/W	H'00	H'FFFFE801	8
Flash erase code select register	FECS	R/W	H'00	H'FFFFE802	8
Flash key code register	FKEY	R/W	H'00	H'FFFFE804	8
Flash MAT select register	FMATS	R/W	H'00*[3] H'AA*[3]	H'FFFFE805	8
Flash transfer destination address register	FTDAR	R/W	H'00	H'FFFFE806	8
RAM emulation register	RAMER	R/W	H'0000	H'FFFFEC26	8, 16

Notes: All registers except for RAMER can be accessed only in bytes, and the access requires three cycles.
RAMER can be accessed in bytes or words, and the access requires three cycles.

*1 The bits except the SCO bit are read-only bits. The SCO bit is a programming-only bit. (The value which can be read is always 0.)

*2 The initial value is H'00 when the FWE pin goes low.
The initial value is H'80 when the FWE pin goes high.

*3 The initial value at initiation in user mode or user program mode is H'00.
The initial value at initiation in user boot mode is H'AA.

Name	Abbreviation	R/W	Value	Address	Size
Download pass/fail result	DPFR	R/W	Undefined	On-chip RAM*	8, 16, 32
Flash pass/fail result	FPFR	R/W	Undefined	R0 of CPU	8, 16, 32
Flash multipurpose address area	FMPAR	R/W	Undefined	R5 of CPU	8, 16, 32
Flash multipurpose data destination area	FMPDR	R/W	Undefined	R4 of CPU	8, 16, 32
Flash erase block select	FEBS	R/W	Undefined	R4 of CPU	8, 16, 32
Flash program and erase frequency control	FPEFEQ	R/W	Undefined	R4 of CPU	8, 16, 32
Flash user branch address set parameter	FUBRA	R/W	Undefined	R5 of CPU	8, 16, 32

Note: * One byte of the start address in the on-chip RAM area specified by FTDAR is valid.

		Download	zation	ming	Erasure	Read	Emulation
Programming/ erasing interface registers	FCCS	O	—	—	—	—	—
	FPCS	O	—	—	—	—	—
	PECS	O	—	—	—	—	—
	FKEY	O	—	O	O	—	—
	FMATS	—	—	O*[1]	O*[1]	O*[2]	—
	FTDAR	O	—	—	—	—	—
Programming/ erasing interface parameters	DPFR	O	—	—	—	—	—
	FPFR	O	O	O	O	—	—
	FPEFEQ	—	O	—	—	—	—
	FUBRA	—	O	—	—	—	—
	FMPAR	—	—	O	—	—	—
	FMPDR	—	—	O	—	—	—
	FEBS	—	—	—	O	—	—
RAM emulation	RAMER	—	—	—	—	—	O

Notes: *1 The setting is required when programming or erasing user MAT in user boot mode.
*2 The setting may be required according to the combination of initiation mode and read target MAT.

22.4.2 Programming/Erasing Interface Registers

The programming/erasing interface registers are as described below. They are all 8-bit registers that can be accessed in bytes. Except for the FLER bit in FCCS and FMATS, these registers are initialized at a power-on reset, in hardware standby mode, or in software standby mode. The FLER bit or FMATS is not initialized in software standby mode.

(1) Flash Code Control and Status Register (FCCS)

FCCS is configured by bits which request the monitor of the FWE pin state and error occurrence during programming or erasing flash memory and the download of the on-chip program.

Bit	:	7	6	5	4	3	2	1	0
		FWE	—	—	FLER	—	—	—	SCO
Initial value	:	1/0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	(R)/W

is 0 or 1 according to the FWE pin state.

Bit 7	
FWE	**Description**
0	When the FWE pin goes low (in hardware protection state)
1	When the FWE pin goes high

Bits 6 and 5—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 4—Flash Memory Error (FLER): Indicates an error occurs during programming and erasing flash memory.

When FLER is set to 1, flash memory enters the error protection state.

This bit is initialized at a power-on reset or in hardware standby mode.

When FLER is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to flash memory, the reset signal must be released after the reset period of 100 μs which is longer than normal.

Bit 4	
FLER	**Description**
0	Flash memory operates normally (Initial value) Programming/erasing protection for flash memory (error protection) is invalid. [Clearing condition] At a power-on reset or in hardware standby mode
1	Indicates an error occurs during programming/erasing flash memory. Programming/erasing protection for flash memory (error protection) is valid. [Setting condition] See section 22.6.3, Error Protection.

Bits 3 to 1—Reserved: These bits should always be cleared to 0.

Bit 0—Source Program Copy Operation (SCO): Requests the on-chip programming/erasing program to be downloaded to the on-chip RAM.

When this bit is set to 1, the on-chip program which is selected by FPCS/FECS is automatically downloaded in the on-chip RAM area specified by FTDAR.

In order to set this bit to 1, RAM emulation state must be canceled, H'A5 must be written to FKEY, and this operation must be in the on-chip RAM.

Four NOP instructions must be executed immediately after setting this bit to 1.

Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1.

Download by setting the SCO bit to 1 requires a special interrupt processing that performs bank switching to the on-chip program storage area. Therefore, before issuing a download request (SCO = 1), set VBR to H'00000000. Otherwise, the CPU gets out of control. Once download end is confirmed, VBR can be changed to any other value.

Bit 0

SCO	Description
0	Download of the on-chip programming/erasing program to the on-chip RAM is not executed (Initial value) [Clear condition] When download is completed
1	Request that the on-chip programming/erasing program is downloaded to the on-chip RAM is generated [Set conditions] When all of the following conditions are satisfied and 1 is written to this bit • H'A5 is written to FKEY • During execution in the on-chip RAM • Not in RAM emulation mode (RAMS in RAMCR = 0)

(2) Flash Program Code Select Register (FPCS)

FPCS selects the on-chip programming program to be downloaded.

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	—	—	PPVS
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R/W

Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 0—Program Pulse Single (PPVS): Selects the programming program.

Bit 0

PPVS	Description
0	On-chip programming program is not selected (Initial value) [Clear condition] When transfer is completed
1	On-chip programming program is selected

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	—	—	EPVB
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R/W

Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 0—Erase Pulse Verify Block (EPVB): Selects the erasing program.

Bit 0

EPVB	Description	
0	On-chip erasing program is not selected	(Initial value)
	[Clear condition] When transfer is completed	
1	On-chip erasing program is selected	

(4) Flash Key Code Register (FKEY)

FKEY is a register for software protection that enables download of the on-chip program and programming/erasing of flash memory. Before setting the SCO bit to 1 in order to download the on-chip program or executing the downloaded programming/erasing program, these processings cannot be executed if the key code is not written.

Bit	:	7	6	5	4	3	2	1	0
		K7	K6	K5	K4	K3	K2	K1	K0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 to 0—Key Code (K7 to K0): Only when H'A5 is written, writing to the SCO bit is valid. When a value other than H'A5 is written to FKEY, 1 cannot be written to the SCO bit. Therefore downloading to the on-chip RAM cannot be executed.

Only when H'5A is written, programming/erasing of flash memory can be executed. Even if the on-chip programming/erasing program is executed, flash memory cannot be programmed or erased when a value other than H'5A is written to FKEY.

H'A5	Writing to the SCO bit is enabled (The SCO bit cannot be set by a value other than H'A5.)
H'5A	Programming/erasing is enabled (A value other than H'A5 enables software protection state.)
H'00	Initial value

(5) Flash MAT Select Register (FMATS)

FMATS specifies whether user MAT or user boot MAT is selected.

Bit :	7	6	5	4	3	2	1	0	
	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
Initial value :	0	0	0	0	0	0	0	0	(When not in user boot mode)
Initial value :	1	0	1	0	1	0	1	0	(When in user boot mode)
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bits 7 to 0—MAT Select (MS7 to MS0): These bits are in user-MAT selection state when a value other than H'AA is written and in user-boot-MAT selection state when H'AA is written.

The MAT is switched by writing a value in FMATS.

When the MAT is switched, follow section 22.8.1, Switching between User MAT and User Boot MAT. (The user boot MAT cannot be programmed in user program mode if user boot MAT is selected by FMATS. The user boot MAT must be programmed in boot mode or in programmer mode.)

Bits 7 to 0

MS7 to MS0	Description
H'AA	The user boot MAT is selected (in user-MAT selection state when the value of these bits are other than H'AA) Initial value when these bits are initiated in user boot mode.
H'00	Initial value when these bits are initiated in a mode except for user boot mode (in user-MAT selection state)

[Programmable condition] These bits are in the execution state in the on-chip RAM.

Make settings for FTDAR before writing 1 to the SCO bit in FCCS. The initial value is H'00 which points to the start address (H'FFFF6000) in on-chip RAM.

Bit	:	7	6	5	4	3	2	1	0
		TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—Transfer Destination Address Setting Error: This bit is set to 1 when there is an error in the download start address set by bits 6 to 0 (TDA6 to TDA0). Whether the address setting is erroneous or not is judged by checking whether the setting of TDA6 to TDA0 is between the range of H'00 and H'05 after setting the SCO bit in FCCS to 1 and performing download. Before setting the SCO bit to 1 be sure to set the FTDAR value between H'00 to H'05 as well as clearing this bit to 0.

Bit 7

TDER	Description (Return Value after Download)
0	Setting of TDA6 to TDA0 is normal (Initial value)
1	Setting of TDER and TDA6 to TDA0 is H'06 to H'FF and download has been aborted

Bits 6 to 0—Transfer Destination Address (TDA6 to TDA0): These bits specify the download start address. A value from H'00 to H'05 can be set to specify the download start address in on-chip RAM in 2-kbyte units.

A value from H'06 to H'7F cannot be set. If such a value is set, the TDER bit (bit 7) in this register is set to 1 to prevent download from being executed.

TDA0	Description
H'00	Download start address is set to H'FFFF6000
H'01	Download start address is set to H'FFFF6800
H'02	Download start address is set to H'FFFF7000
H'03	Download start address is set to H'FFFF7800
H'04	Download start address is set to H'FFFF8000
H'05	Download start address is set to H'FFFF8800
H'06 to H'7F	Setting prohibited. If this value is set, the TDER bit (bit 7) is set to 1 to abort the download processing.

22.4.3 Programming/Erasing Interface Parameters

The programming/erasing interface parameters specify the operating frequency, user branch destination address, storage place for program data, programming destination address, and erase block and exchanges the processing result for the downloaded on-chip program. This parameter uses the general registers of the CPU (R4, R5, and R0) or the on-chip RAM area. The initial value is undefined at a power-on reset or in hardware standby mode.

At download all CPU registers are stored, and at initialization or when the on-chip program is executed, CPU registers except for R0 are stored. The return value of the processing result is written in R0. Since the stack area is used for storing the registers or as a work area, the stack area must be saved at the processing start. (The maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameters are used in the following four items.

(1) Download control
(2) Initialization before programming or erasing
(3) Programming
(4) Erasing

These items use different parameters. The correspondence table is shown in table 22.6.

The processing results of initialization, programming, and erasing are returned, but the bit contents have different meanings according to the processing program. See the description of FPFR for each processing.

	ation	load	zation	ming			Value	
Download pass/fail result	DPFR	O	—	—	—	R/W	Undefined	On-chip RAM*
Flash pass/fail result	FPFR	—	O	O	O	R/W	Undefined	R0 of CF
Flash programming/ erasing frequency control	FPEFEQ	—	O	—	—	R/W	Undefined	R4 of CF
Flash user branch address set parameter	FUBRA	—	O	—	—	R/W	Undefined	R5 of CF
Flash multipurpose address area	FMPAR	—	—	O	—	R/W	Undefined	R5 of CF
Flash multipurpose data destination area	FMPDR	—	—	O	—	R/W	Undefined	R4 of CF
Flash erase block select	FEBS	—	—	—	O	R/W	Undefined	R4 of CF

Note: * One byte of start address of download destination specified by FTDAR

(1) Download Control

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-chip RAM area to be downloaded is the area as much as 2 kbytes starting from the start address specified by FTDAR. For the address map of the on-chip RAM, see figure 22.10.

The download control is set by using the programming/erasing interface registers. The return value is given by the DPFR parameter.

(a) Download pass/fail result parameter (DPFR: one byte of start address of on-chip RAM specified by FTDAR)

This parameter indicates the return value of the download result. The value of this parameter can be used to determine if downloading is executed or not. Since the confirmation whether the SCO bit is set to 1 is difficult, the certain determination must be performed by setting one byte of the start address of the on-chip RAM area specified by FTDAR to a value other than the return value of download (for example, H'FF) before the download start (before setting the SCO bit to 1). For the checking method of download results, see section 22.5.2, User Program Mode.

Bit :	7	6	5	4	3	2	1	0
	0	0	0	0	0	SS	FK	SF

Bits 7 to 3—Unused: Return 0.

program is not selected, or the program is selected without mapping, an error occurs.

Bit 2

SS	Description
0	Download program can be selected normally
1	Download error occurs (Multi-selection or program which is not mapped is selected)

Bit 1—Flash Key Register Error Detect (FK): Returns the check result whether the value of FKEY is set to H'A5.

Bit 1

FK	Description
0	FKEY setting is normal (FKEY = H'A5)
1	FKEY setting is abnormal (FKEY = value other than H'A5)

Bit 0—Success/Fail (SF): Returns the result whether download has ended normally or not.

Bit 0

SF	Description
0	Downloading on-chip program has ended normally (no error)
1	Downloading on-chip program has ended abnormally (error occurs)

(2) Programming/Erasing Initialization

The on-chip programming/erasing program to be downloaded includes the initialization program.

The specified period pulse must be applied when programming or erasing. The specified pulse width is made by the method in which wait loop is configured by the CPU instruction. The operating frequency of the CPU must be set. Since the user branch function is supported, the user branch destination address must be set.

The initial program is set as a parameter of the programming/erasing program which has downloaded these settings.

For the range of the operating frequency of this LSI, see section 25.3.2, Clock Timing.

Bit :	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0

Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0

Bit :	15	14	13	12	11	10	9	8
	F15	F14	F13	F12	F11	F10	F9	F8

Bit :	7	6	5	4	3	2	1	0
	F7	F6	F5	F4	F3	F2	F1	F0

Bits 31 to 16—Unused: Return 0.

Bits 15 to 0—Frequency Set (F15 to F0): Set the operating frequency of the CPU. The setting value must be calculated as the following methods.

1. The operating frequency which is shown in MHz units must be rounded in a number to three decimal places and be shown in a number of two decimal places.
2. The centuplicated value is converted to the binary digit and is written to the FPEFEQ parameter (general register R4). For example, when the operating frequency of the CPU is 28.882 MHz, the value is as follows.
1. The number to three decimal places of 28.882 is rounded and the value is thus 28.88.
2. The formula that $28.88 \times 100 = 2888$ is converted to the binary digit and b'0000,1011,0100,1000 (H'0B48) is set to R4.

can be executed in specified processing units when programming and erasing.

Bit :	31	30	29	28	27	26	25	24
	UA31	UA30	UA29	UA28	UA27	UA26	UA25	UA24

Bit :	23	22	21	20	19	18	17	16
	UA23	UA22	UA21	UA20	UA19	UA18	UA17	UA16

Bit :	15	14	13	12	11	10	9	8
	UA15	UA14	UA13	UA12	UA11	UA10	UA9	UA8

Bit :	7	6	5	4	3	2	1	0
	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0

Bits 31 to 0—User Branch Destination Address (UA31 to UA0): When the user branch is not required, address 0 (H'00000000) must be set.

The user branch destination must be an area other than the flash memory, an area other than the RAM area in which on-chip program has been transferred, or the external bus space.

Note that the CPU must not branch to an area without the execution code and get out of control. The on-chip program download area and stack area must not be overwritten. If CPU runaway occurs or the download area or stack area is overwritten, the value of flash memory cannot be guaranteed.

The download of the on-chip program, initialization, initiation of the programming/erasing program must not be executed in the processing of the user branch destination. Programming or erasing cannot be guaranteed when returning from the user branch destination. The program data which has already been prepared must not be programmed.

The general registers R8 to R15 are stored. The general registers R0 to R7 can be used without being stored.

Moreover, the programming/erasing interface registers must not be written to or RAM emulation mode must not be entered in the processing of the user branch destination.

After the processing of the user branch has ended, the programming/erasing program must be returned to by using the RTS instruction.

For the execution intervals of the user branch processing, see note 2 (User branch processing intervals) in section 22.8.3, Other Notes.

Bit :	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0
Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
Bit :	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
Bit :	7	6	5	4	3	2	1	0
	0	0	0	0	0	BR	FQ	SF

Bits 31 to 3—Unused: Return 0.

Bit 2—User Branch Error Detect (BR): Returns the check result whether the specified user branch destination address is in the area other than the storage area of the programming/erasing program which has been downloaded .

Bit 2

BR	Description
0	User branch address setting is normal
1	User branch address setting is abnormal

Bit 1—Frequency Error Detect (FQ): Returns the check result whether the specified operating frequency of the CPU is in the range of the supported operating frequency.

Bit 1

FQ	Description
0	Setting of operating frequency is normal
1	Setting of operating frequency is abnormal

Bit 0—Success/Fail (SF): Indicates whether initialization is completed normally.

Bit 0

SF	Description
0	Initialization has ended normally (no error)
1	Initialization has ended abnormally (error occurs)

must be passed to the programming program in which the program data is downloaded.

1. The start address of the programming destination on the user MAT is set in general register R5 of the CPU. This parameter is called FMPAR (flash multipurpose address area parameter).
   Since the program data is always in 128-byte units, the lower eight bits (MOA7 to MOA0) must be H'00 or H'80 as the boundary of the programming start address on the user MAT.
2. The program data for the user MAT must be prepared in the consecutive area. The program data must be in the consecutive space which can be accessed by using the MOV.B instruction of the CPU and is not the flash memory space.
   When data to be programmed does not satisfy 128 bytes, the 128-byte program data must be prepared by embedding the dummy code (H'FF).
   The start address of the area in which the prepared program data is stored must be set in general register R4. This parameter is called FMPDR (flash multipurpose data destination area parameter).
   For details on the programming procedure, see section 22.5.2, User Program Mode.

(3.1) Flash multipurpose address area parameter (FMPAR: general register R5 of CPU)

This parameter indicates the start address of the programming destination on the user MAT.

When an address in an area other than the flash memory space is set, an error occurs.

The start address of the programming destination must be at the 128-byte boundary. If this boundary condition is not satisfied, an error occurs. The error occurrence is indicated by the WA bit (bit 1) in FPFR.

Bit :	31	30	29	28	27	26	25	24
	MOA31	MOA30	MOA29	MOA28	MOA27	MOA26	MOA25	MOA24

Bit :	23	22	21	20	19	18	17	16
	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17	MOA16

Bit :	15	14	13	12	11	10	9	8
	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9	MOA8

Bit :	7	6	5	4	3	2	1	0
	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1	MOA0

Bits 31 to 0—MOA31 to MOA0: Store the start address of the programming destination on the user MAT. The consecutive 128-byte programming is executed starting from the specified start address of the user MAT. The MOA6 to MOA0 bits are always 0 because the start address of the programming destination is at the 128-byte boundary.

in the user MAT. When the storage destination of the program data is in flash memory, an error occurs. The error occurrence is indicated by the WD bit (bit 2) in FPFR.

Bit :	31	30	29	28	27	26	25	24
	MOD31	MOD30	MOD29	MOD28	MOD27	MOD26	MOD25	MOD24

Bit :	23	22	21	20	19	18	17	16
	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MOD16

Bit :	15	14	13	12	11	10	9	8
	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	MOD8

Bit :	7	6	5	4	3	2	1	0
	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0

Bits 31 to 0—MOD31 to MOD0: Store the start address of the area which stores the program data for the user MAT. The consecutive 128-byte data is programmed to the user MAT starting from the specified start address.

(3.3) Flash pass/fail parameter (FPFR: general register R0 of CPU)

This parameter indicates the return value of the program processing result.

Bit :	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0

Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0

Bit :	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0

Bit :	7	6	5	4	3	2	1	0
	0	MD	EE	FK	0	WD	WA	SF

Bits 31 to 7—Unused: Return 0.

Bit 6—Programming Mode Related Setting Error Detect (MD): Returns the check result of whether the signal input to the FWE pin is high and whether the error protection state is entered.

When a low-level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The input level to the FWE pin and the error protection state can be confirmed

Bit 6

MD	Description
0	FWE and FLER settings are normal (FWE = 1, FLER = 0)
1	FWE = 0 or FLER = 1, and programming cannot be performed

Bit 5—Programming Execution Error Detect (EE): 1 is returned to this bit when the specified data could not be written because the user MAT was not erased or when flash-memory related register settings are partially changed on returning from the user branch processing.

If this bit is set to 1, there is a high possibility that the user MAT is partially rewritten. In this case, after removing the error factor, erase the user MAT.

If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when programming is performed. In this case, both the user MAT and user boot MAT are not rewritten.

Programming of the user boot MAT must be executed in boot mode or programmer mode.

Bit 5

EE	Description
0	Programming has ended normally
1	Programming has ended abnormally (programming result is not guaranteed)

Bit 4—Flash Key Register Error Detect (FK): Returns the check result of the value of FKEY before the start of the programming processing.

Bit 4

FK	Description
0	FKEY setting is normal (FKEY = H'A5)
1	FKEY setting is error (FKEY = value other than H'A5)

Bit 3—Unused: Returns 0.

Bit 2	
WD	**Description**
0	Setting of write data address is normal
1	Setting of write data address is abnormal

Bit 1—Write Address Error Detect (WA): When the following items are specified as the start address of the programming destination, an error occurs.

1. The programming destination address is an area other than flash memory
2. The specified address is not at the 128-byte boundary (A6 to A0 are not 0)

Bit 1	
WA	**Description**
0	Setting of programming destination address is normal
1	Setting of programming destination address is abnormal

Bit 0—Success/Fail (SF): Indicates whether the program processing has ended normally or not.

Bit 0	
SF	**Description**
0	Programming has ended normally (no error)
1	Programming has ended abnormally (error occurs)

(4) Erasure Execution

When flash memory is erased, the erase-block number on the user MAT must be passed to the erasing program which is downloaded. This is set to the FEBS parameter (general register R4).

One block is specified from the block number 0 to 15.

For details on the erasing procedure, see section 22.5.2, User Program Mode.

Bit :	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0

Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0

Bit :	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0

Bit :	7	6	5	4	3	2	1	0
	EBS7	EBS6	EBS5	EBS4	EBS3	EBS2	EBS1	EBS0

Bits 31 to 8—Unused: Return 0.

Bits 7 to 0—Erase Block (EB7 to EB0): Set the erase-block number in the range from 0 to 15. 0 corresponds to the EB0 block and 15 corresponds to the EB15 block. An error occurs when a number other than 0 to 15 (H'00 to H'0F) is set.

(4.2) Flash pass/fail result parameter (FPFR: general register R0 of CPU)

This parameter returns the value of the erasing processing result.

Bit :	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0

Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0

Bit :	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0

Bit :	7	6	5	4	3	2	1	0
	0	MD	EE	FK	EB	0	0	SF

Bits 31 to 7—Unused: Return 0.

Bit 6—Erasure Mode Related Setting Error Detect (MD): Returns the check result of whether the signal input to the FWE pin is high and whether the error protection state is entered.

When a low-level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The input level to the FWE pin and the error protection state can be confirmed

Bit 6	
MD	**Description**
0	FWE and FLER settings are normal (FWE = 1, FLER = 0)
1	FWE = 0 or FLER = 1, and erasure cannot be performed

Bit 5—Erasure Execution Error Detect (EE): 1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed on returning from the user branch processing.

If this bit is set to 1, there is a high possibility that the user MAT is partially erased. In this case, after removing the error factor, erase the user MAT.

If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when erasure is performed. In this case, both the user MAT and user boot MAT are not erased.

Erasure of the user boot MAT must be executed in boot mode or programmer mode.

Bit 5	
EE	**Description**
0	Erasure has ended normally
1	Erasure has ended abnormally (erasure result is not guaranteed)

Bit 4—Flash Key Register Error Detect (FK): Returns the check result of FKEY value before start of the erasing processing.

Bit 4	
FK	**Description**
0	FKEY setting is normal (FKEY = H'5A)
1	FKEY setting is error (FKEY = value other than H'5A)

Bit 3—Erase Block Select Error Detect (EB): Returns the check result whether the specified erase-block number is in the block range of the user MAT.

Bit 3	
EB	**Description**
0	Setting of erase-block number is normal
1	Setting of erase-block number is abnormal

Bit 0

SF	Description
0	Erasure has ended normally (no error)
1	Erasure has ended abnormally (error occurs)

22.4.4 RAM Emulation Register (RAMER)

When the realtime programming of the user MAT is emulated, RAMER sets the area of the user MAT which is overlapped with a part of the on-chip RAM. RAMER is initialized to H'0000 at a power-on reset or in hardware standby mode and is not initialized in software standby mode. The RAMER setting must be executed in user mode or in user program mode.

For the division method of the user-MAT area, see table 22.7. In order to operate the emulation function certainly, the target MAT of the RAM emulation must not be accessed immediately after RAMER is programmed. If it is accessed, the normal access is not guaranteed.

Bit	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bits 15 to 4—Reserved: These bits are always read as 0. The write value should always be 0.

Bit 3 RAMS	Description	
0	Emulation is not selected Programming/erasing protection of all user-MAT blocks is invalid	(Initial value)
1	Emulation is selected Programming/erasing protection of all user-MAT blocks is valid	

Bits 2 to 0—User MAT Area Select: These bits are used with bit 3 to select the user-MAT area to be overlapped with the on-chip RAM. (See table 22.7.)

Table 22.7 Overlapping of RAM Area and User MAT Area

RAM Area	Block Name	RAMS	RAM2	RAM1	RAM0
H'FFFF6000 to H'FFFF6FFF	RAM area (4 kbytes)	0	*	*	*
H'00000000 to H'00000FFF	EB0 (4 kbytes)	1	0	0	0
H'00001000 to H'00001FFF	EB1 (4 kbytes)	1	0	0	1
H'00002000 to H'00002FFF	EB2 (4 kbytes)	1	0	1	0
H'00003000 to H'00003FFF	EB3 (4 kbytes)	1	0	1	1
H'00004000 to H'00004FFF	EB4 (4 kbytes)	1	1	0	0
H'00005000 to H'00005FFF	EB5 (4 kbytes)	1	1	0	1
H'00006000 to H'00006FFF	EB6 (4 kbytes)	1	1	1	0
H'00007000 to H'00007FFF	EB7 (4 kbytes)	1	1	1	1

Note: * Don't care.

When the pin is set in on-board programming mode and the reset start is executed, the on-board programming state that can program/erase the on-chip flash memory is entered. On-board programming mode has three operating modes: user programming mode, user boot mode, and boot mode.

For details on the pin setting for entering each mode, see table 22.1. For details on the state transition of each mode for flash memory, see figure 22.2.

22.5.1 Boot Mode

Boot mode executes programming/erasing user MAT and user boot MAT by means of the control command and program data transmitted from the host using the on-chip SCI. The tool for transmitting the control command and program data must be prepared in the host. The SCI communication mode is set to asynchronous mode. When reset start is executed after this LSI's pin is set in boot mode, the boot program in the microcomputer is initiated. After the SCI bit rate is automatically adjusted, the communication with the host is executed by means of the control command method.

The system configuration diagram in boot mode is shown in figure 22.6. For details on the pin setting in boot mode, see table 22.1. Interrupts are ignored in boot mode, so do not generate them. Note that the AUD cannot be used during boot mode operation.

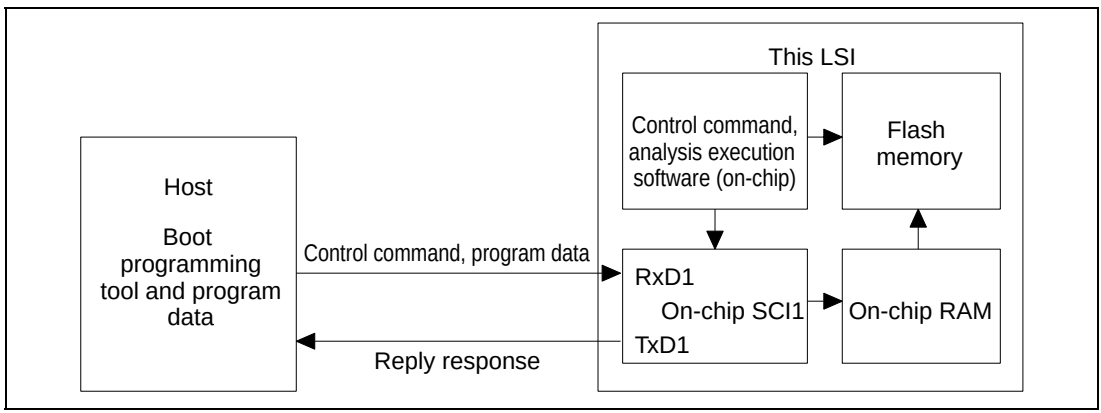


Figure 22.6 System Configuration in Boot Mode

communication data (H'00), which is transmitted consecutively by the host. The SCI transmit/receive format is set to 8-bit data, 1 stop bit, and no parity. This LSI calculates the bit rate of transmission by the host by means of the measured low period and transmits the bit adjustment end sign (1 byte of H'00) to the host. The host must confirm that this bit adjustment end sign (H'00) has been received normally and transmits 1 byte of H'55 to this LSI. When reception is not executed normally, boot mode is initiated again (reset) and the operation described above must be executed. The bit rate between the host and this LSI is not matched because of the bit rate of transmission by the host and system clock frequency of this LSI. To operate the SCI normally, the transfer bit rate of the host must be set to 9,600 bps or 19,200 bps.

The system clock frequency which can automatically adjust the transfer bit rate of the host and the bit rate of this LSI is shown in table 22.8. Boot mode must be initiated in the range of this system clock.

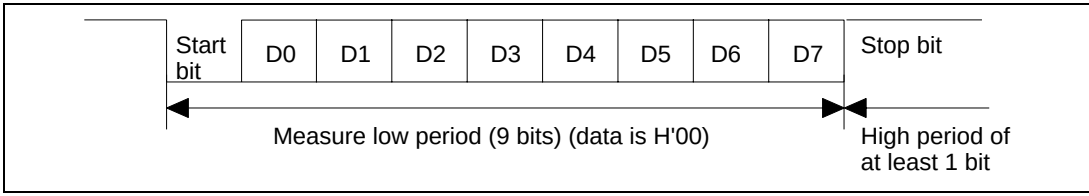


Figure 22.7 Automatic Adjustment Operation of SCI Bit Rate

Table 22.8 System Clock Frequency that Can Automatically Adjust Bit Rate of This LSI

Host Bit Rate	System Clock Frequency Which Can Automatically Adjust LSI's Bit Rate
9,600 bps	20 to 40 MHz (input frequency of 5 to 10 MHz)
19,200 bps	20 to 40 MHz (input frequency of 5 to 10 MHz)

(2) State Transition

The overview of the state transition after boot mode is initiated is shown in figure 22.8. For details on boot mode, see section 22.10.1, Serial Communications Interface Specification for Boot Mode.

1. Bit rate adjustment

   After boot mode is initiated, the bit rate of the SCI interface is adjusted with that of the host.
2. Waiting for inquiry set command

   For inquiries about the user-MAT size and configuration, MAT start address, and support state, the required information is transmitted to the host.
3. Automatic erasure of all user MAT and user boot MAT

   After inquiries have finished and a programming/erasing status transition command has been sent, all of the user MAT and user boot MAT are automatically erased.

entered. The programming start address and program data must be transmitted following the programming command. When programming is finished, the programming start address must be set to H'FFFFFFFF and transmitted. Then the state for waiting program data is returned to the state of programming/erasing command wait.

- When the erasure selection command is received, the state for waiting erase-block data is entered. The erase-block number must be transmitted following the erasing command.
  When the erasure is finished, the erase-block number must be set to H'FF and transmitted.
  Then the state for waiting erase-block data is returned to the state for waiting programming/erasing command. The erasure must be executed when reset start is not executed and the specified block is programmed after programming is executed in boot mode. When programming can be executed by only one operation, all blocks are erased before the state for waiting programming/erasing/other command is entered. The erasing operation is not required.
- There are many commands other than programming/erasing. Examples are sum check, blank check (erasure check), and memory read of the user MAT/user boot MAT and acquisition of current status information.

Note that memory read of the user MAT/user boot MAT can only read the program data after all user MAT/user boot MAT has automatically been erased.

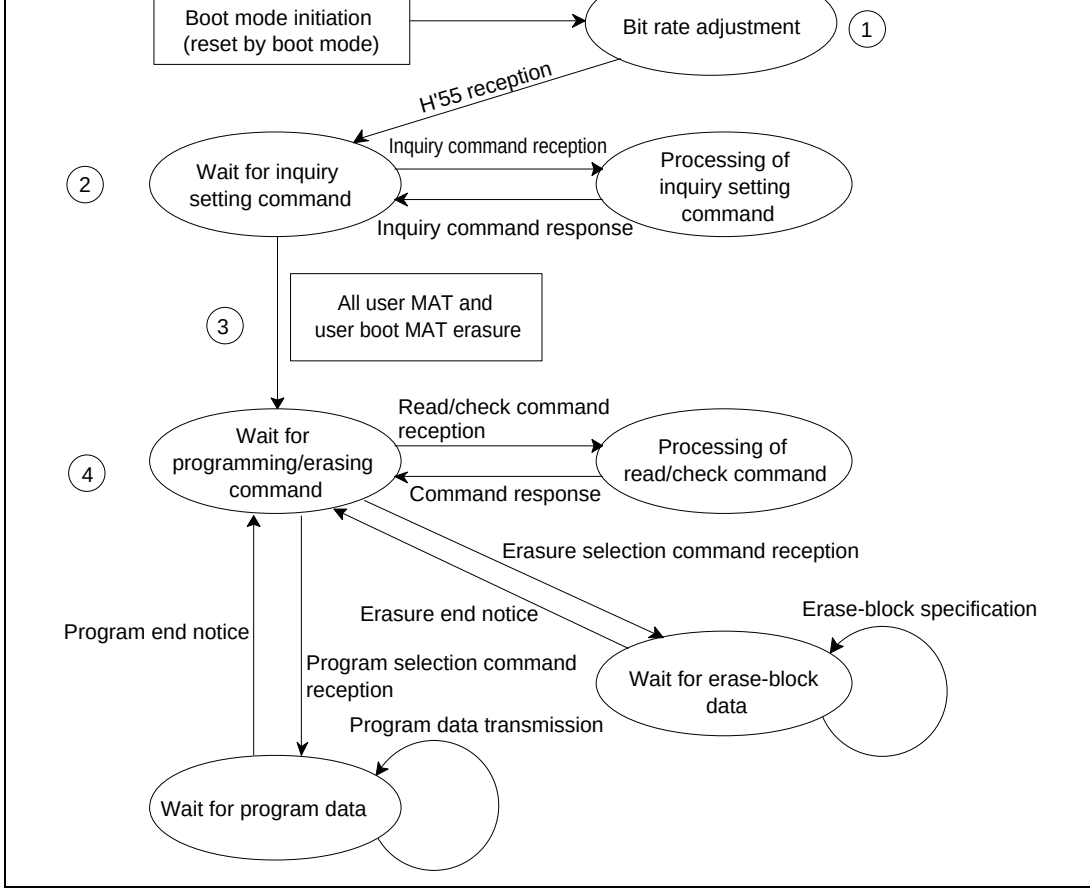


Figure 22.8 Overview of Boot Mode State Transition

22.5.2 User Program Mode

The user MAT can be programmed/erased in user program mode. (The user boot MAT cannot be programmed/erased.)

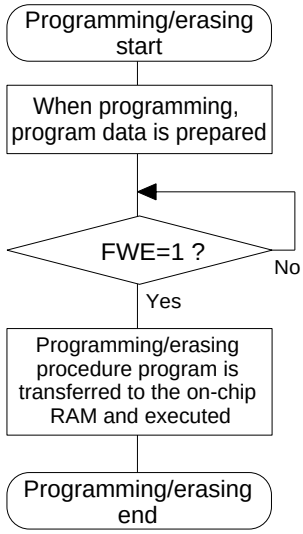
Programming/erasing is executed by downloading the program in the microcomputer.

The overview flow is shown in figure 22.9.

High voltage is applied to internal flash memory during the programming/erasing processing. Therefore, transition to reset or hardware standby mode must not be executed. Doing so may cause damage or destroy flash memory. If reset is executed accidentally, the reset signal must be released after the reset input period, which is longer than the normal 100 µs.

22.5.2 (5) Erasing Procedure in User Program Mode.

For the overview of a processing that repeats erasing and programming by downloading the programming program and the erasing program in separate on-chip ROM areas using FTDAR, see the description in 22.5.2 (4), Erasing and Programming Procedure in User Program Mode.



1. RAM emulation mode must be canceled in advance. Download cannot be executed in emulation mode.
2. When the program data is made by means of emulation, the download destination must be changed by FTDAR. With the initial setting of FTDAR (H'00), the download area is overlapped with the emulation area.
3. Inputting high level to the FWE pin sets the FWE bit to 1.
4. Programming/erasing is executed only in the on-chip RAM. However, if the program data is in a consecutive area and can be accessed by the MOV.B instruction of the CPU like SRAM/ROM, the program data can be in an external space.
5. After programming/erasing is finished, low level must be input to the FWE pin for protection.

Figure 22.9 Programming/Erasing Overview Flow

programming/erasing procedure, and judgement of the result, must be executed in the on-chip RAM. All of the on-chip program that is to be downloaded is in on-chip RAM. Note that on-chip RAM must be controlled so that these parts do not overlap.

Figure 22.10 shows the program area to be downloaded.

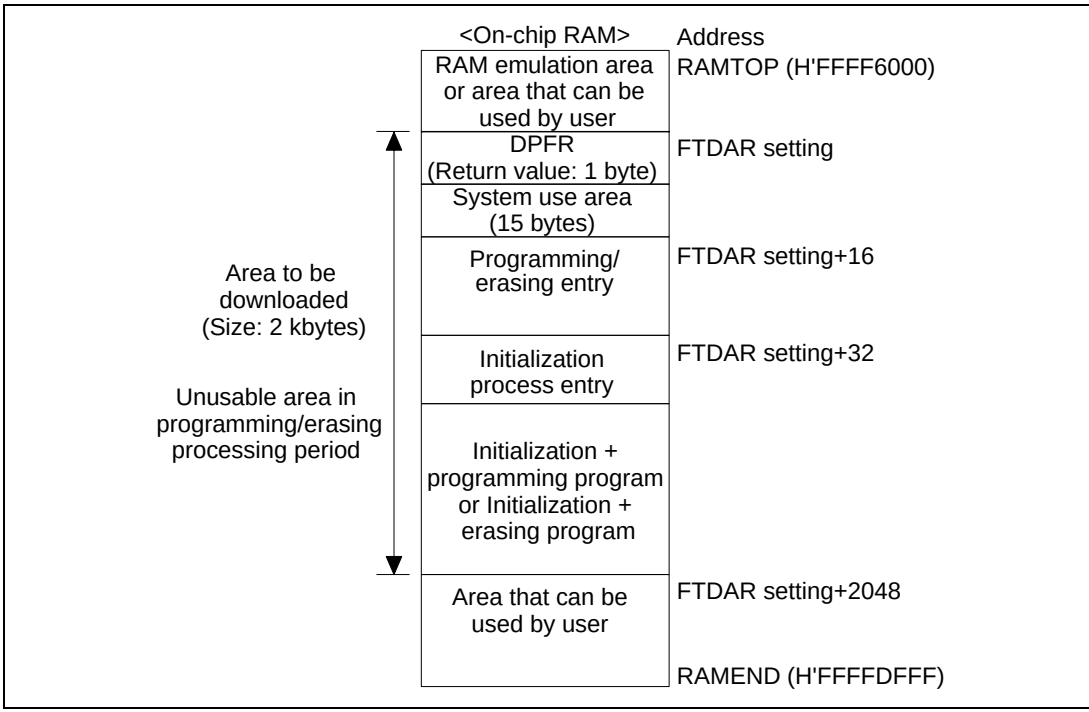


Figure 22.10 RAM Map after Download

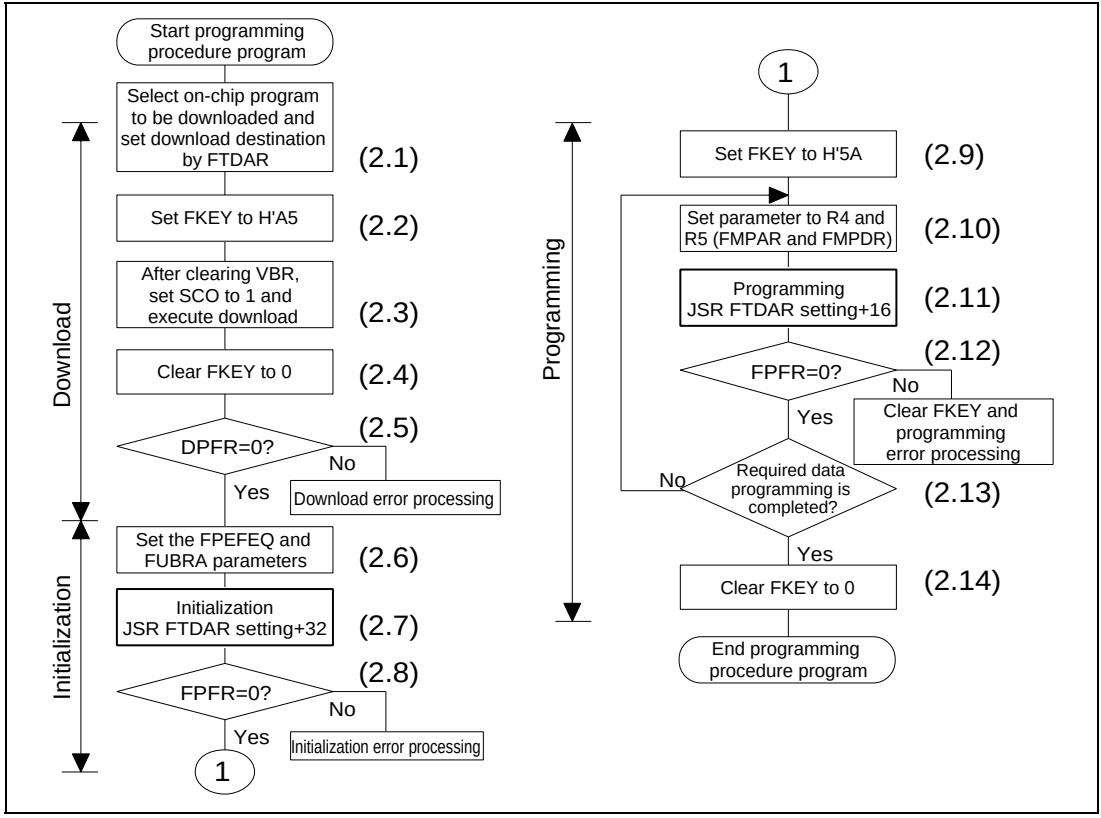


Figure 22.11 Programming Procedure

The details of the programming procedure are described below. The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 22.10.3, Storable Area for Procedure Program and Programming Data.

The following description assumes the area to be programmed on the user MAT is erased and program data is prepared in the consecutive area. When erasing has not been executed, carry out erasing before writing.

128-byte programming is performed in one program processing. When more than 128-byte programming is performed, programming destination address/program data parameter is updated in 128-byte units and programming is repeated.

shortened.

(2.1) Select the on-chip program to be downloaded

When the PPVS bit of FPCS is set to 1, the programming program is selected.
Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the source select error detect (SS) bit in the DPFR parameter.
Specify the start address of the download destination by FTDAR.

(2.2) Write H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be written to the SCO bit for a download request.

(2.3) VBR is cleared to 0 and 1 is written to the SCO bit of FCCS, and then download is executed.

VBR must always be cleared to H'00000000 before setting the SCO bit to 1.
To write 1 to the SCO bit, the following conditions must be satisfied.

- RAM emulation mode is canceled.
- H'A5 is written to FKEY.
- The SCO bit writing is executed in the on-chip RAM.

When the SCO bit is set to 1, download is started automatically. When execution returns to the user procedure program, the SCO bit is cleared to 0. Therefore, the SCO bit cannot be confirmed to be 1 in the user procedure program.
The download result can be confirmed only by the return value of the DPFR parameter. Before the SCO bit is set to 1, incorrect judgement must be prevented by setting the DPFR parameter, that is one byte of the start address of the on-chip RAM area specified by FTDAR, to a value other than the return value (H'FF).
When download is executed, particular interrupt processing, which is accompanied by the bank switch as described below, is performed as an internal microcomputer processing, so VBR need to be cleared to 0. Four NOP instructions are executed immediately after the instructions that set the SCO bit to 1.

- The user MAT space is switched to the on-chip program storage area.
- After the selection condition of the download program and the address set in FTDAR are checked, the transfer processing is executed starting from the on-chip RAM address specified by FTDAR.
- The SCO bits in FPCS, FECS, and FCCS are cleared to 0.
- The return value is set to the DPFR parameter.
- After the on-chip program storage area is returned to the user MAT space, execution returns to the user procedure program.

After download is completed and the user procedure program is running, the VBR setting can be changed.

The notes on download are as follows.

NMI, UBC, and H-UDI interrupt requests are retained, so that on returning to the user procedure program, the interrupt processing starts. For details on the relationship between download and interrupts, see section 22.8.2, Interrupts during Programming/Erasing.

Since a stack area of maximum 128 bytes is used, an area of at least 128 bytes must be saved before setting the SCO bit to 1.

If flash memory is accessed by the DMAC or AUD during downloading, operation cannot be guaranteed. Therefore, access by the DMAC or AUD must not be executed.

(2.4) FKEY is cleared to H'00 for protection.

(2.5) The value of the DPFR parameter must be checked to confirm the download result.

A recommended procedure for confirming the download result is shown below.

- Check the value of the DPFR parameter (one byte of start address of the download destination specified by FTDAR). If the value is H'00, download has been performed normally. If the value is not H'00, the source that caused download to fail can be investigated by the description below.
- If the value of the DPFR parameter is the same as before downloading (e.g. H'FF), the address setting of the download destination in FTDAR may be abnormal. In this case, confirm the setting of the TDER bit (bit 7) in FTDAR.
- If the value of the DPFR parameter is different from before downloading, check the SS bit (bit 2) and the FK bit (bit 1) in the DPFR parameter to ensure that the download program selection and FKEY register setting were normal, respectively.

(2.6) The operating frequency is set to the FPEFEQ parameter and the user branch destination is set to the FUBRA parameter for initialization.

- The current frequency of the CPU clock is set to the FPEFEQ parameter (general register R4). For the settable range of the FPEFEQ parameter, see section 25.3.2, Clock Timing.

  For the settable range of the FPEFEQ parameter, see section 25.3.2, Clock Timing. When the frequency is set out of this range, an error is returned to the FPFR parameter of the initialization program and initialization is not performed. For details on the frequency setting, see the description in 22.4.3 (2.1) Flash programming/erasing frequency parameter (FPEFEQ).
- The start address in the user branch destination is set to the FUBRA parameter (general register R5).

  When the user branch processing is not required, 0 must be set to FUBRA.
  When the user branch is executed, the branch destination is executed in flash memory other than the one that is to be programmed. The area of the on-chip program that is downloaded cannot be set.

  The program processing must be returned from the user branch processing by the RTS instruction.

(2.7) Initialization

When a programming program is downloaded, the initialization program is also downloaded to on-chip RAM. There is an entry point of the initialization program in the area from (download start address set by FTDAR) + 32 bytes. The subroutine is called and initialization is executed by using the following steps.

```
MOV.L  #DLTOP+32,R1        ; Set entry address to R1
JSR    @R1                 ; Call initialization routine
NOP
```

— The general registers other than R0 are saved in the initialization program.

— R0 is a return value of the FPFR parameter.

— Since the stack area is used in the initialization program, a stack area of maximum 128 bytes must be reserved in RAM.

— Interrupts can be accepted during the execution of the initialization program. However, the program storage area and stack area in on-chip RAM and register values must not be destroyed.

(2.8) The return value of the initialization program, FPFR (general register R0) is judged.

(2.9) FKEY must be set to H'5A and the user MAT must be prepared for programming.

(2.10) The parameter which is required for programming is set.

The start address of the programming destination of the user MAT (FMPAR) is set to general register R5. The start address of the program data storage area (FMPDR) is set to general register R4.

- FMPAR setting

  FMPAR specifies the programming destination start address. When an address other than one in the user MAT area is specified, even if the programming program is executed, programming is not executed and an error is returned to the return value parameter FPFR. Since the unit is 128 bytes, the lower eight bits (MOA7 to MOA0) must be in the 128-byte boundary of H'00 or H'80.

- FMPDR setting

  If the storage destination of the program data is flash memory, even when the program execution routine is executed, programming is not executed and an error is returned to the FPFR parameter. In this case, the program data must be transferred to on-chip RAM and then programming must be executed.

(2.11) Programming

There is an entry point of the programming program in the area from (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and programming is executed by using the following steps.

```
NOP
```

— The general registers other than R0 are saved in the programming program.

— R0 is a return value of the FPFR parameter.

— Since the stack area is used in the programming program, a stack area of maximum 128 bytes must be reserved in RAM.

(2.12) The return value in the programming program, FPFR (general register R0) is judged.

(2.13) Determine whether programming of the necessary data has finished.

If more than 128 bytes of data are to be programmed, specify FMPAR and FMPDR in 128-byte units, and repeat steps (2.10) to (2.13). Increment the programming destination address by 128 bytes and update the programming data pointer correctly. If an address which has already been programmed is written to again, not only will a programming error occur, but also flash memory will be damaged.

(2.14) After programming finishes, clear FKEY and specify software protection.

If this LSI is restarted by a power-on reset immediately after user MAT programming has finished, secure a reset period (period of $\overline{RES}$ = 0) that is at least as long as the normal 100 µs.

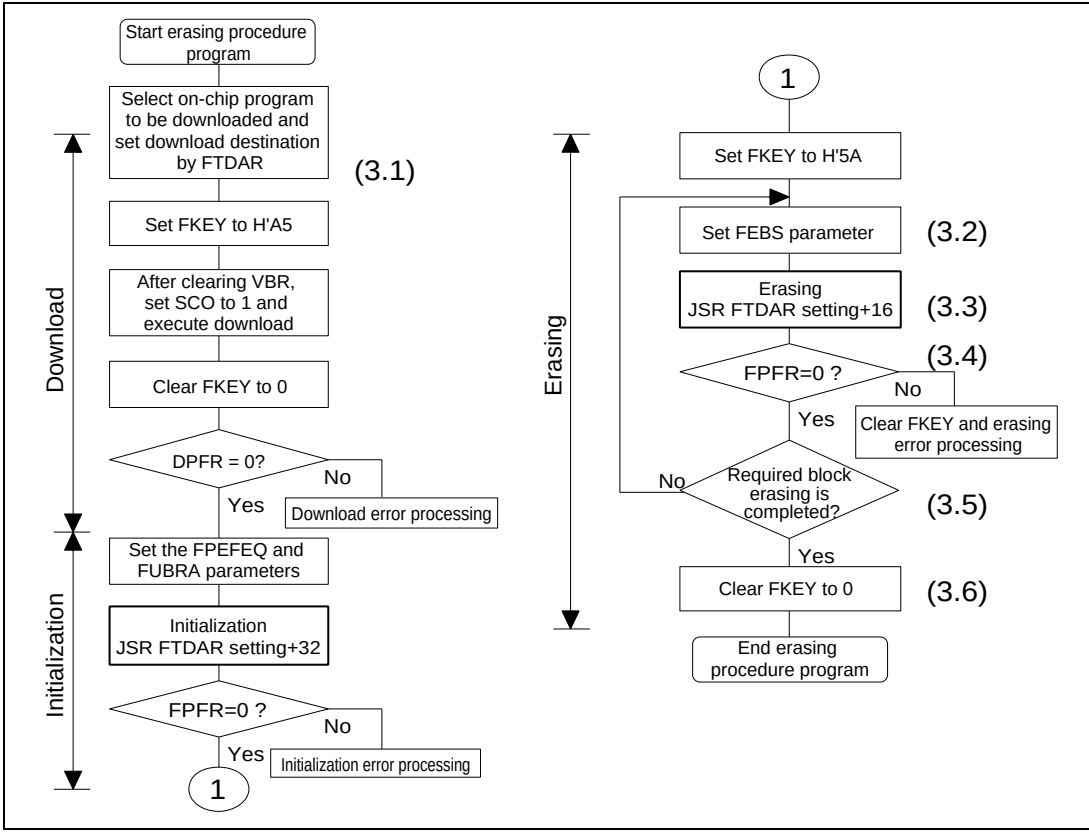


Figure 22.12 Erasing Procedure

The details of the erasing procedure are described below. The procedure program must be executed in an area other than the user MAT to be erased.

Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 22.10.3, Storable Area for Procedure Program and Programming Data.

For the downloaded on-chip program area, see the RAM map for programming/erasing in figure 22.10.

for each block.

(3.1) Select the on-chip program to be downloaded

Set the EPVB bit in FECS to 1.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the source select error detect (SS) bit in the DPFR parameter.

Specify the start address of the download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization, are the same as those in the programming procedure. For details, see the description in 22.5.2 (2) Programming Procedure in User Program Mode.

(3.2) Set the FEBS parameter necessary for erasure

Set the erase block number of the user MAT in the flash erase block select parameter (FEBS: general register R4). If a value other than an erase block number of the user MAT is set, no block is erased even though the erasing program is executed, and an error is returned to the return value parameter FPFR.

(3.3) Erasure

Similar to as in programming, there is an entry point of the erasing program in the area from (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and erasing is executed by using the following steps.

```
MOV.L  #DLTOP+16,R1        ; Set entry address to R1
JSR    @R1                 ; Call erasing routine
NOP
```

- The general registers other than R0L are saved in the erasing program.
- R0 is a return value of the FPFR parameter.
- Since the stack area is used in the erasing program, a stack area of maximum 128 bytes must be reserved in RAM.

(3.4) The return value in the erasing program, FPFR (general register R0) is judged.

(3.5) Determine whether erasure of the necessary blocks has finished.

If more than one block is to be erased, update the FEBS parameter and repeat steps (3.2) to (3.5). Blocks that have already been erased can be erased again.

(3.6) After erasure finishes, clear FKEY and specify software protection.

If this LSI is restarted by a power-on reset immediately after user MAT programming has finished, secure a reset period (period of $\overline{RES}$ = 0) that is at least as long as the normal 100 µs.

program and programming program can be downloaded to separate on-chip RAM areas. Figure 22.13 shows an example of repetitively executing RAM emulation, erasing, and programming.

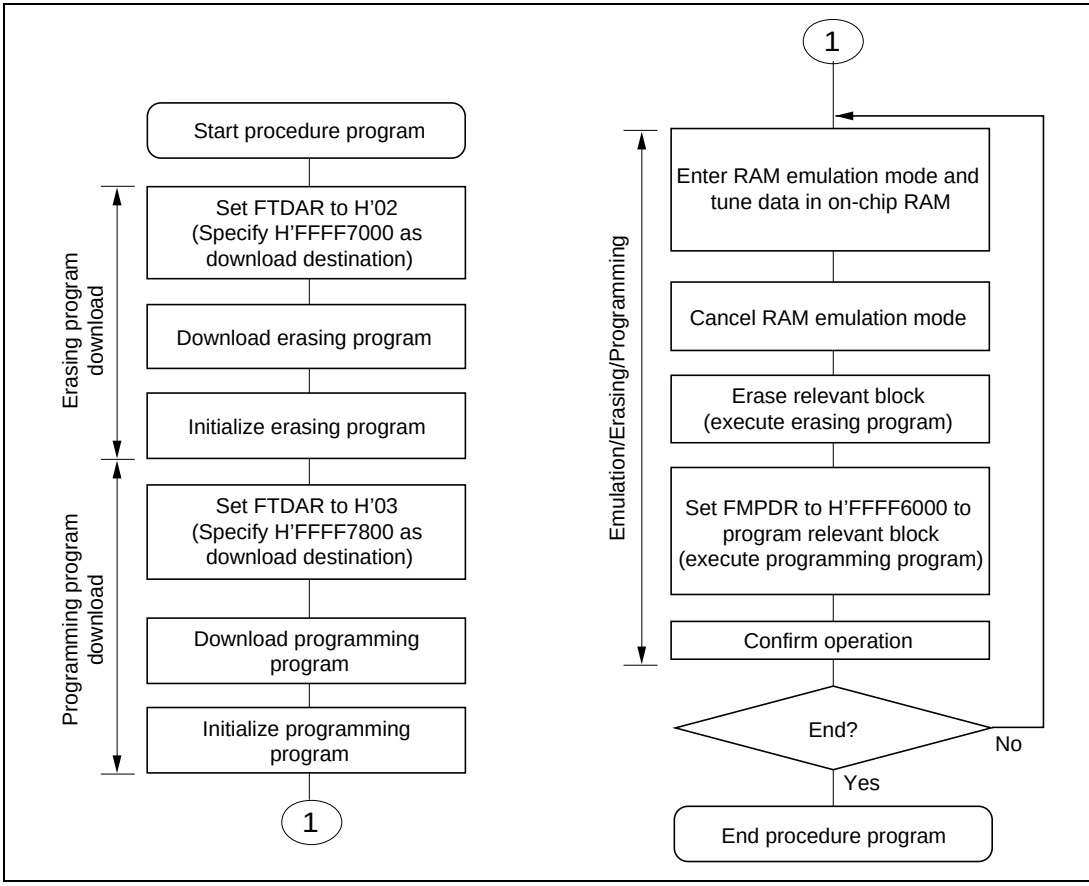


Figure 22.13 Sample Procedure of Repeating RAM Emulation, Erasing, and Programming (Overview)

In the above example, the erasing program and programming program are downloaded to areas excluding the 4 kbytes (H'FFFF6000 to H'FFFF6FFF) from the start of on-chip ROM.

Download and initialization are performed only once at the beginning.

In this kind of operation, note the following:

- Be careful not to damage on-chip RAM with overlapped settings.
  In addition to the RAM emulation area, erasing program area, and programming program area,

- Be sure to initialize both the erasing program and programming program.
  Initialization by setting the FPEFEQ and FUBRA parameters must be performed for both the erasing program and the programming program. Initialization must be executed for both entry addresses: (download start address for erasing program) + 32 bytes (H'FFFF7020 in this example) and (download start address for programming program) + 32 bytes (H'FFFF7820 in this example).

22.5.3 User Boot Mode

This LSI has user boot mode which is initiated with different mode pin settings than those in user program mode or boot mode. User boot mode is a user-arbitrary boot mode, unlike boot mode that uses the on-chip SCI.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing of the user boot MAT is only enabled in boot mode or programmer mode.

(1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 22.1, Relationship between FWE and MD Pins and Operating Modes.

When the reset start is executed in user boot mode, the check routine for flash-memory related registers runs. While the check routine is running, the RAM area about 1.2 kbytes from H'FFFF6800 is used by the routine and 4 bytes from H'FFFFDFFC is used as a stack area. NMI and all other interrupts cannot be accepted. Neither can the AUD be used in this period. This period is approximately 100 µs while operating at an internal frequency of 40 MHz.

Next, processing starts from the execution start address of the reset vector in the user boot MAT. At this point, H'AA is set to the flash MAT select register (FMATS) because the execution MAT is the user boot MAT.

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processings made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after programming completes.

Figure 22.14 shows the procedure for programming the user MAT in user boot mode.

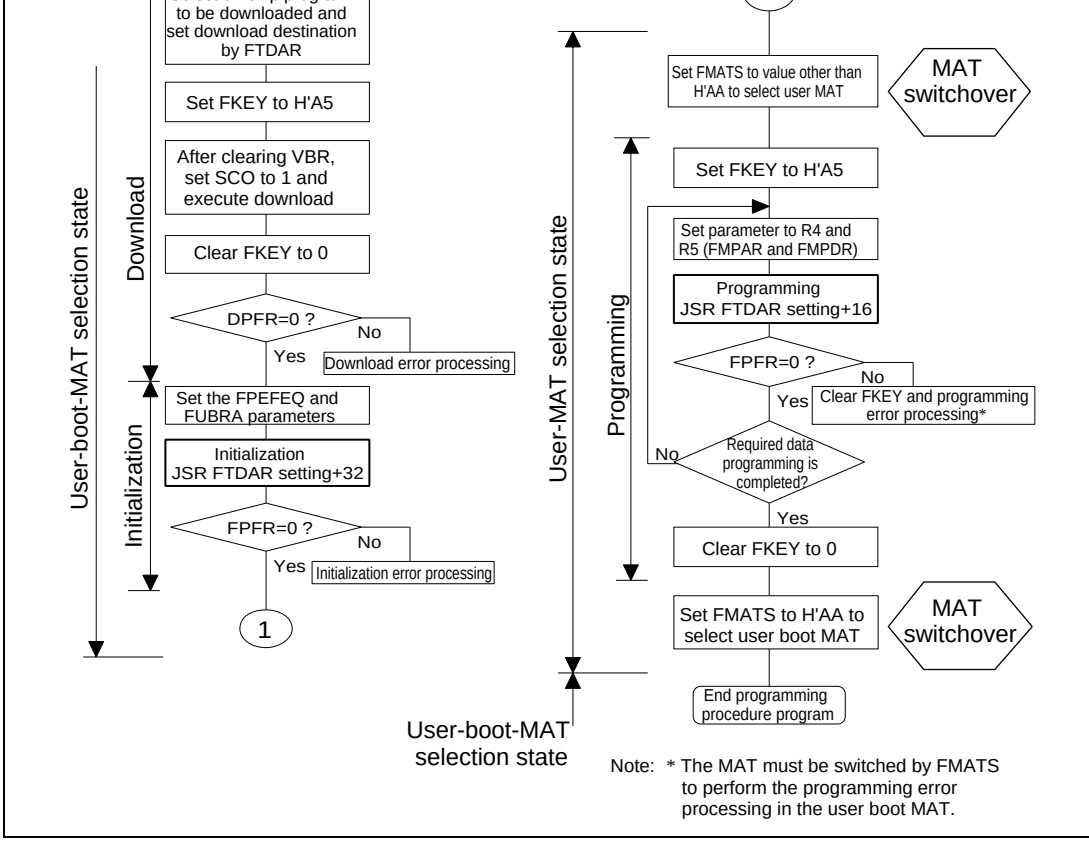


Figure 22.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user boot mode is whether the MAT is switched or not as shown in figure 22.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the user MAT hidden in the background. The user MAT and user boot MAT are switched only while the user MAT is being programmed. Because the user boot MAT is hidden while the user MAT is being programmed, the procedure program must be located in an area other than flash memory. After programming finishes, switch the MATs again to return to the first state.

MAT switchover is enabled by writing a specific value to FMATS. However note that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completely finished, and if an interrupt occurs, from which MAT the interrupt vector is read from is undetermined. Perform MAT switching in accordance with the description in section 22.8.1, Switching between User MAT and User Boot MAT.

MAT, and external space) is shown in section 22.10.3, Storable Area for Procedure Program and Programming Data.

(3) User MAT Erasing in User Boot Mode

For erasing the user MAT in user boot mode, additional processings made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after erasing completes.

Figure 22.15 shows the procedure for erasing the user MAT in user boot mode.

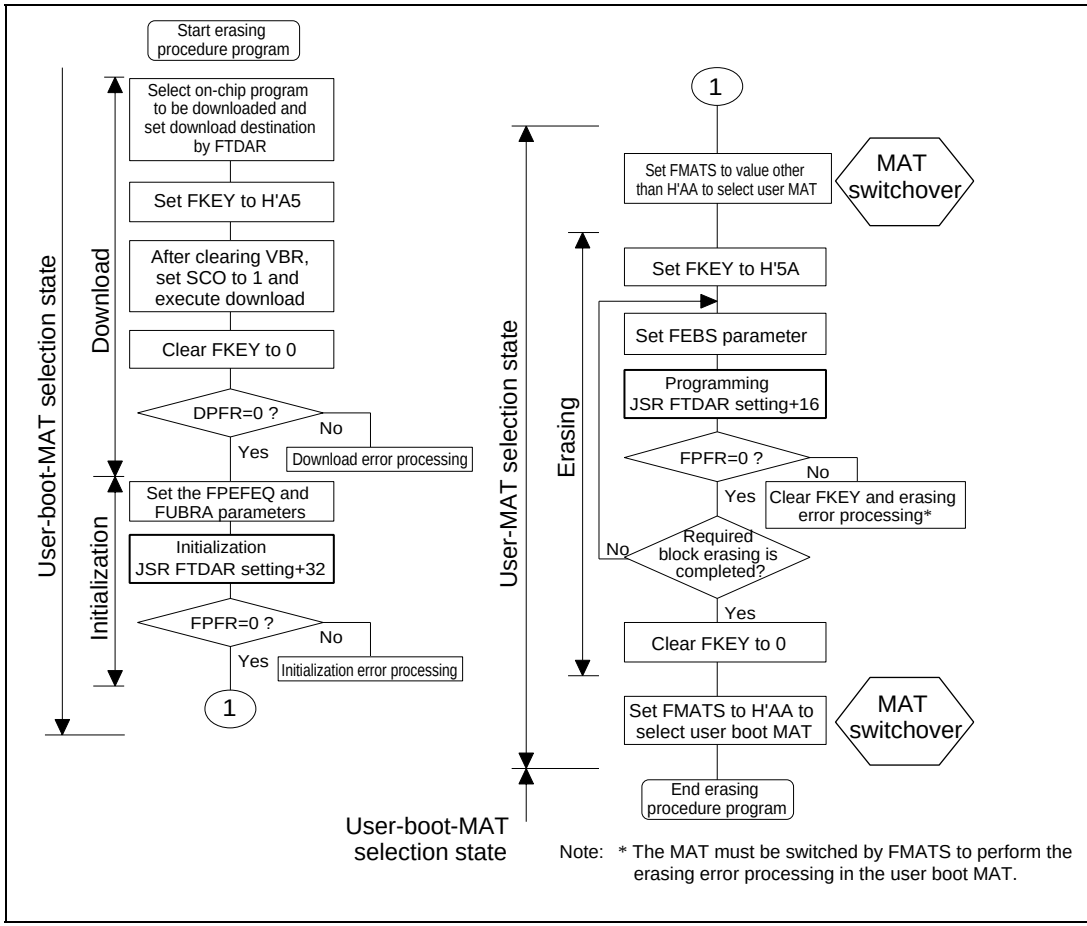


Figure 22.15 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mode depends on whether the MAT is switched or not as shown in figure 22.15.

MAT switching is completed finished, and if an interrupt occurs, from which MAT the interrupt vector is read from is undetermined. Perform MAT switching in accordance with the description in section 22.8.1, Switching between User MAT and User Boot MAT.

Except for MAT switching, the erasing procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 22.10.3, Storable Area for Procedure Program and Programming Data.

22.6 Protection

There are three kinds of flash memory program/erase protection: hardware, software, and error protection.

22.6.1 Hardware Protection

Programming and erasing of flash memory is forcibly disabled or suspended by hardware protection. In this state, the downloading of an on-chip program and initialization of the flash memory are possible. However, an activated program for programming or erasure cannot program or erase locations in a user MAT, and the error in programming/erasing is reported in the FPFR parameter.

Item	Description	Download	Programming/ Erasure
FWE-pin protection	The input of a low-level signal on the FWE pin clears the FWE bit of FCCS and the LSI enters a programming/erasing-protected state.	—	O
Reset/standby protection	• A power-on reset (including a power-on reset by the WDT) and entry to standby mode initializes the programming/erasing interface registers and the LSI enters a programming/erasing-protected state. • Resetting by means of the $\overline{\text{RES}}$ pin after power is initially supplied will not make the LSI enter the reset state unless the $\overline{\text{RES}}$ pin is held low until oscillation has stabilized. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width that is specified in the section on AC characteristics. If the LSI is reset during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again.	O	O

22.6.2 Software Protection

Software protection is set up in any of three ways: by disabling the downloading of on-chip programs for programming and erasing, by means of a key code, and by the RAM emulation register (RAMER).

Item	Description	Download	Programming/ Erasure
Protection by the SCO bit	Clearing the SCO bit in FCCS disables downloading of the programming/erasing program, thus making the LSI enter a programming/erasing-protected state.	O	O
Protection by FKEY	Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and for programming/erasing.	O	O
Emulation protection	Setting the RAMS bit in RAMER to 1 makes the LSI enter a programming/erasing-protected state.	O	O

22.6.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occurs, in the form of the microcomputer getting out of control during programming/erasing of the flash memory or operations that are not in accordance with the established procedures for programming/erasing. Aborting programming or erasure in such cases prevents damage to the flash memory due to excessive programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, the FLER bit in FCCS is set to 1 and the LSI enters the error protection state, thus aborting programming or erasure.

The FLER bit is set to 1 in the following conditions:

- When the relevant block area of flash memory is read during programming/erasing (including a vector read or an instruction fetch)
- When a SLEEP instruction (including software standby mode) is executed during programming/erasing

Error protection is cancelled (FLER bit is cleared) only by a power-on reset or in hardware-standby mode.

Note that the reset signal should only be released after providing a reset input over a period longer than the normal 100 µs. Since high voltages are applied during programming/erasing of the flash memory, some voltage may still remain even after the error protection state has been entered. For

The state-transition diagram in figure 22.16 shows transitions to and from the error protection state.

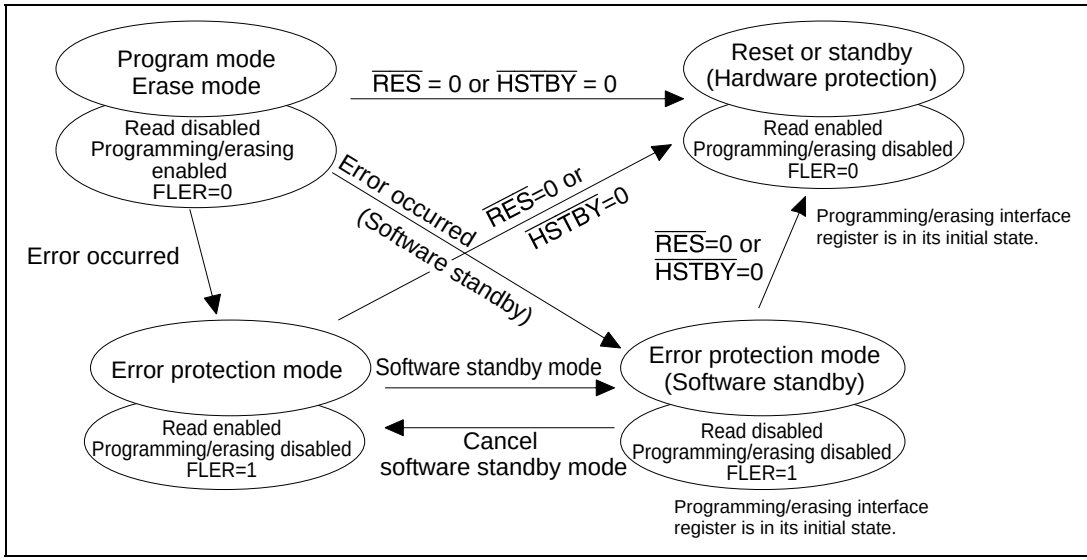


Figure 22.16 Transitions to and from Error Protection State

To provide real-time emulation in RAM of data that is to be written to the flash memory, a part of the RAM can be overlaid on an area of flash memory (user MAT) that has been specified by the RAM emulation register (RAMER). After the RAMER setting is made, the RAM is accessible in both the user MAT area and as the RAM area that has been overlaid on the user MAT area. Such emulation is possible in user mode and user program mode.

Figure 22.17 shows an example of the emulation of realtime programming of the user MAT area.

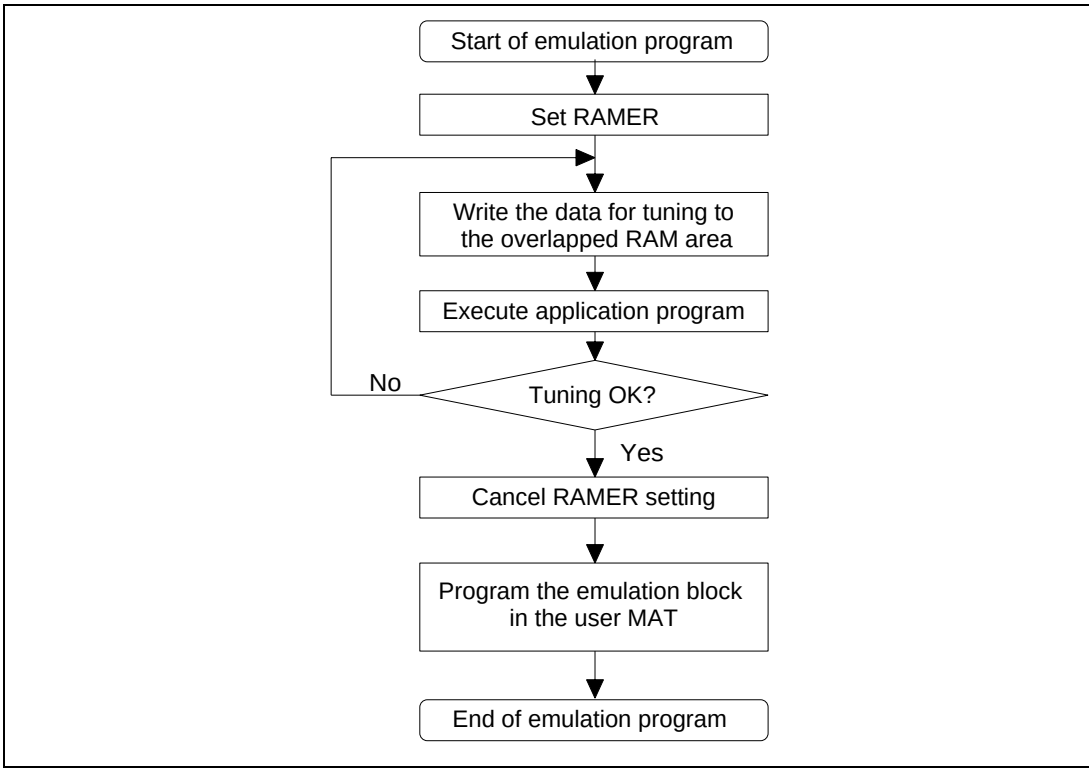


Figure 22.17 Emulation of Flash Memory in RAM

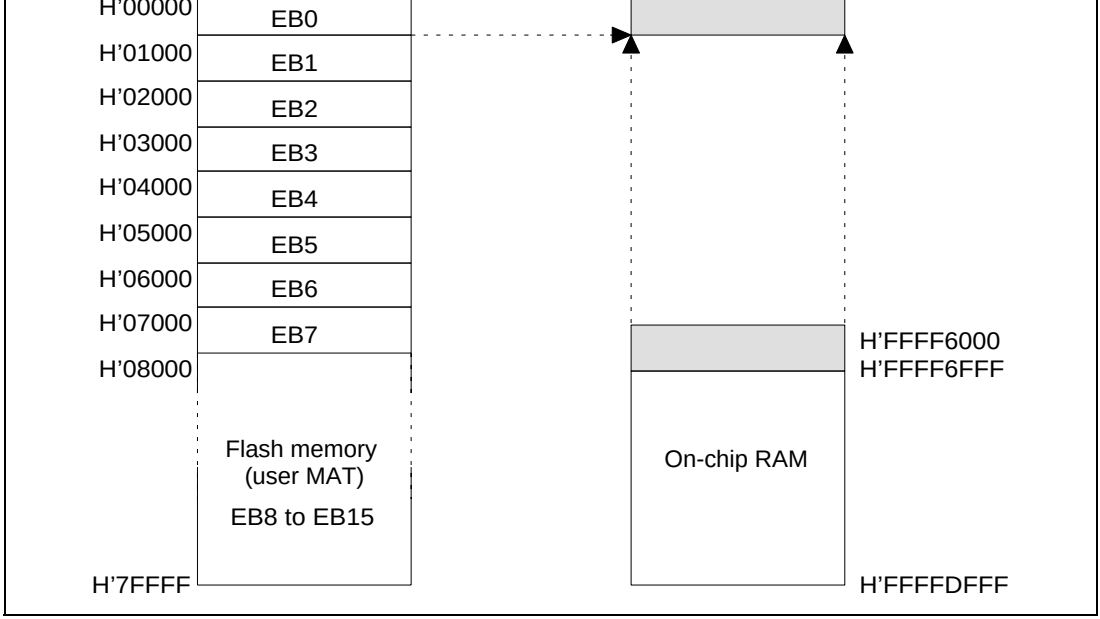


Figure 22.18 Example of Overlapped RAM Operation

Figure 22.18 shows an example of an overlap on block area EB0 of the flash memory.

Emulation is possible for a single area selected from among the eight areas, from EB0 to EB7, of the user MAT. The area is selected by the setting of the RAM2 to RAM0 bits in RAMER.

(1) To overlap a part of the RAM on area EB0, to allow realtime programming of the data for this area, set the RAMS bit in RAMER to 1, and each of the RAM2 to RAM0 bits to 0.

(2) Realtime programming is carried out using the overlaid area of RAM.

In programming or erasing the user MAT, it is necessary to run a program that implements a series of procedural steps, including the downloading of an on-chip program. In this process, set the download area with FTDAR so that the overlaid RAM area and the area where the on-chip program is to be downloaded do not overlap. The initial setting (H'00) of FTDAR causes the tuned data area to overlap with the download area. When using the initial setting of FTDAR, the data that is to be programmed must be saved beforehand in an area that is not used by the system.

Figure 22.19 shows an example of programming data that has been emulated to the EB0 area in the user MAT.

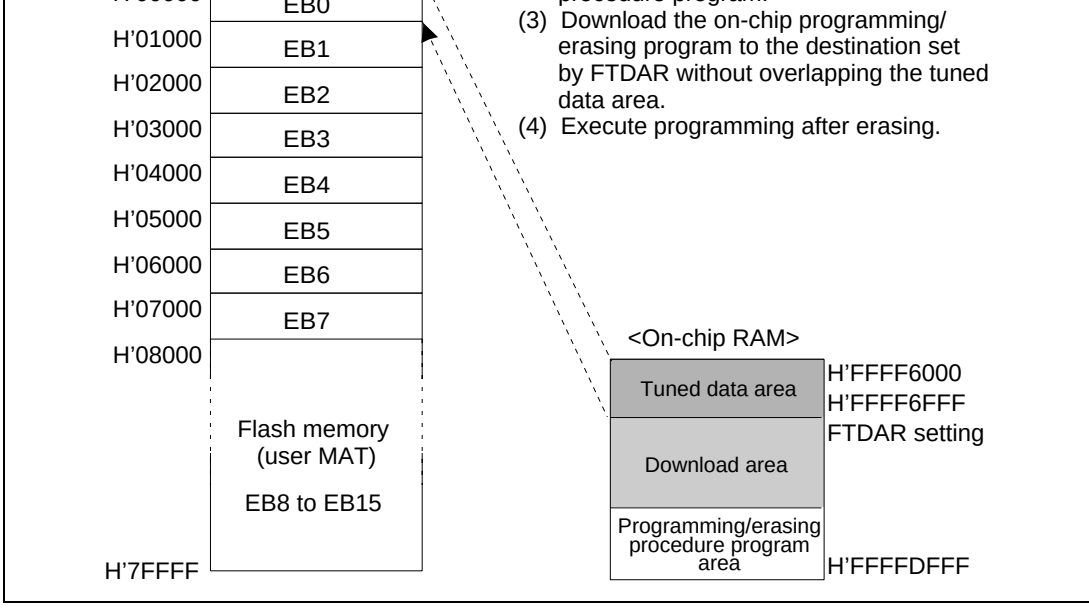


Figure 22.19 Programming of Tuned Data

1. After the data to be programmed has fixed values, clear the RAMS bit to 0 to cancel the overlap of RAM. Emulation mode is canceled and emulation protection is also cleared.
2. Transfer the user programming/erasing procedure program to RAM.
3. Run the programming/erasing procedure program in RAM and download the on-chip programming/erasing program.
   Specify the download start address with FTDAR so that the tuned data area does not overlap with the download area.
4. When the EB0 area of the user MAT has not been erased, erasing must be performed before programming. Set the parameters FMPAR and FMPDR so that the tuned data is designated, and execute programming.

Note: Setting the RAMS bit to 1 puts all the blocks in flash memory in the programming/erasing-protected state regardless of the values of the RAM2 to RAM0 bits (emulation protection). Clear the RAMS bit to 0 before actual programming or erasure. RAM emulation can be performed when the user boot MAT is selected. However, programming/erasing user boot MAT can be performed only in boot mode or program mode.

22.8.1 Switching between User MAT and User Boot MAT

It is possible to switch between the user MAT and user boot MAT. However, the following procedure is required because these MATs are allocated to address 0.
(Switching to the user boot MAT disables programming and erasing. Programming of the user boot MAT must take place in boot mode or programmer mode.)

(1) MAT switching by FMATS should always be executed from the on-chip RAM. The SH microcomputer prefetches execution instructions. Therefore, a switchover during program execution in the user MAT causes an instruction code in the user MAT to be prefetched or an instruction in the newly selected user boot MAT to be prefetched, thus resulting in unstable operation.

(2) To ensure that the MAT that has been switched to is accessible, execute four NOP instructions in on-chip RAM immediately after writing to FMATS of on-chip RAM (this prevents access to the flash memory during MAT switching).

(3) If an interrupt occurs during switching, there is no guarantee of which memory MAT is being accessed.
Always mask the maskable interrupts before switching MATs. In addition, configuring the system so that NMI interrupts do not occur during MAT switching is recommended.

(4) After the MATs have been switched, take care because the interrupt vector table will also have been switched.
If the same interrupt processings are to be executed before and after MAT switching or interrupt requests cannot be disabled, transfer the interrupt processing routine to on-chip RAM, and use the VBR setting to place the interrupt vector table in on chip RAM. In this case, make sure the VBR setting change does not conflict with the interrupt occurrence.

(5) Memory sizes of the user MAT and user boot MAT are different. When accessing the user boot MAT, do not access addresses exceeding the 8-kbyte memory space. If access goes beyond the 8-kbyte space, the values read are undefined.

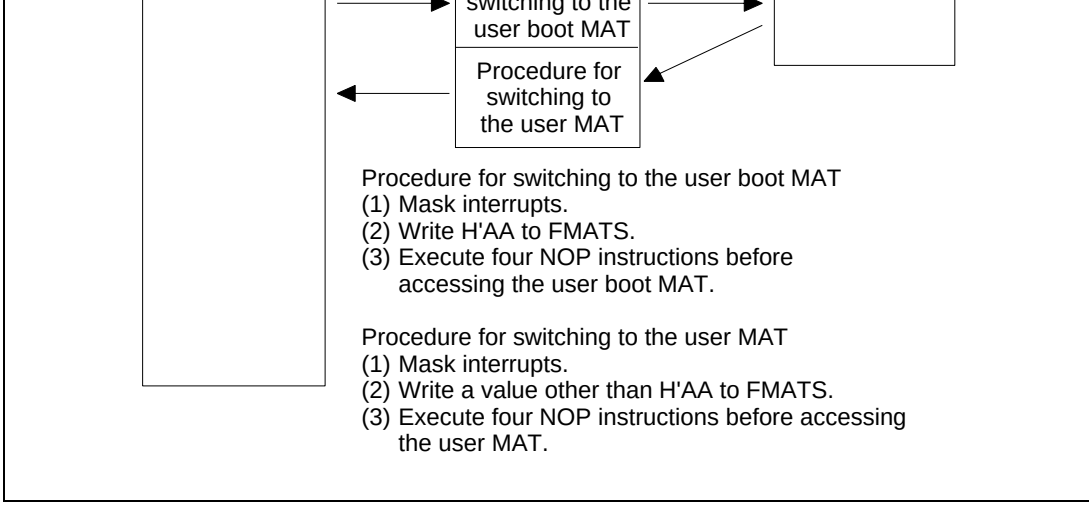


Figure 22.20 Switching between User MAT and User Boot MAT

22.8.2 Interrupts during Programming/Erasing

(1) Download of On-Chip Program

(1.1) VBR setting change

Before downloading the on-chip program, VBR must be set to H'00000000 (initial value). If VBR is set to a value other than the initial value, the interrupt vector table is placed in the user MAT (FMATS is not H'AA) or the user boot MAT (FMATS is H'AA) on initialization of VBR.

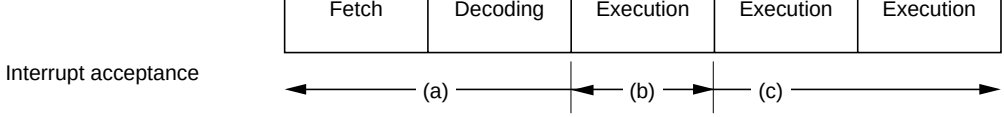
When VBR setting change conflicts with interrupt occurrence, whether the vector table before or after VBR is changed is referenced may cause an error.

Therefore, for cases where VBR setting change may conflict with interrupt occurrence, prepare a vector table to be referenced when VBR is H'00000000 at the start of the user MAT or user boot MAT.

(1.2) SCO download request and interrupt request

Download of the on-chip programming/erasing program that is initiated by setting the SCO bit in FCCS to 1 generates a particular interrupt processing accompanied by MAT switchover. Operation when the SCO download request and interrupt request conflicts is described below.

1. Contention between SCO download request and interrupt request
   Figure 22.21 shows the timing of contention between execution of the instruction that sets the SCO bit in FCCS to 1 and interrupt acceptance.



(a) When the interrupt is accepted at or before the (n + 1) cycle
After the interrupt processing completes, the SCO bit is set to 1 and download is executed.

(b) When the interrupt is accepted at the (n + 2) cycle
The interrupt conflicts with the SCO download request. For details on operation in this case, see 2. Operation when contention occurs.

(c) When the interrupt is accepted at or after the (N + 3) cycle
The SCO download request occurs prior to the interrupt request, and download is executed. During download, no other interrupt processing can be handled. If an interrupt is still being requested after download completes, the interrupt processing starts. For details on interrupt requests during download, see 3. Interrupt requests generated during download.

Figure 22.21 Timing of Contention between SCO Download Request and Interrupt Request

2. Operation when contention occurs
Operation differs according to the type of interrupt with which the SCO download request has conflicted.

— NMI, UBC, and H-UDI interrupt requests
Operation for when these interrupts conflict with the SCO download request is described below.

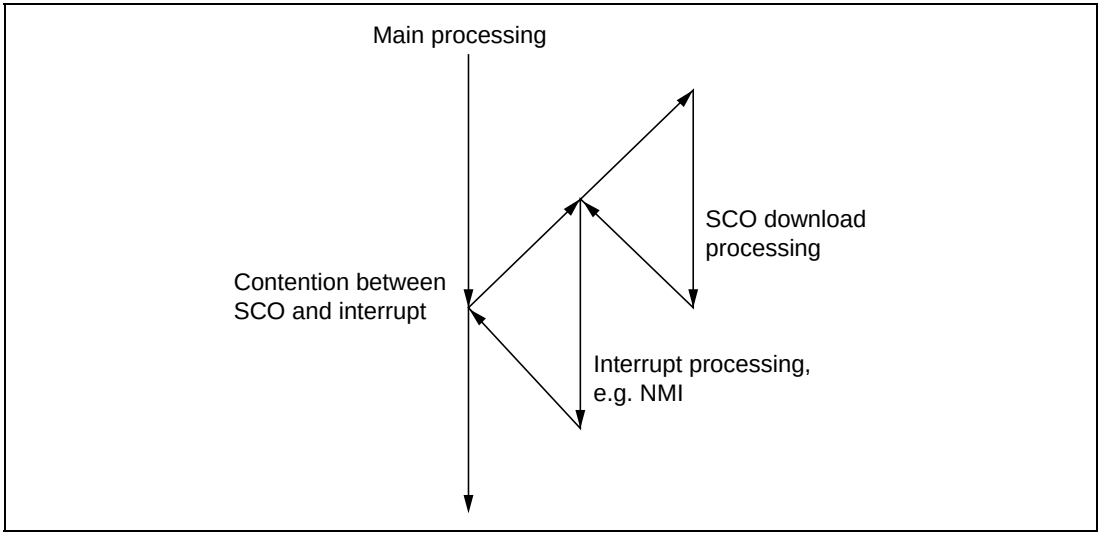


Figure 22.22 Contention between Interrupts (e.g. NMI)

interrupt processing routine is fetched.

- At this point, the SCO download request with a higher priority occurs. The SCO download processing is started.
- After the download processing has ended, the interrupt processing routine (e.g. NMI) that was in the middle of execution resumes from the point of fetching the start instruction of the interrupt processing routine.
- The interrupt processing routine is ended, and execution returns to the main processing.

— IRQ and on-chip peripheral module interrupt requests
Operation for when these interrupts conflict with the SCO download request is described below.

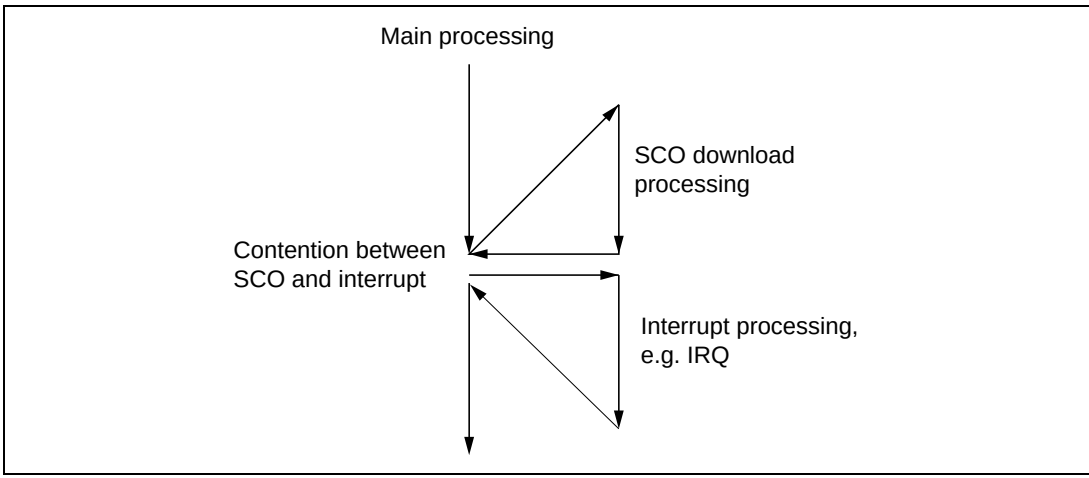


Figure 22.23 Contention between Interrupts (e.g. IRQ)

- An IRQ interrupt or interrupt from an on-chip peripheral module is replaced with the SCO download request and download is executed.
- If the IRQ or on-chip peripheral module interrupt is still being requested when the download processing has ended, the interrupt processing is executed. If these interrupt requests have been canceled, execution returns to the main processing.
- An interrupt request is canceled when the IRQ signal, for which low-level detection is set, has been driven high before download ends. Also refer to the description below (3. Interrupt requests generated during download).

3. Interrupt requests generated during download

completion of download, the interrupt processing starts. When more than one type of interrupts are requested, their priorities are judged by the interrupt controller (INTC), and execution starts from the interrupt processing with higher priority.

— NMI, UBC, and H-UDI interrupt requests
When these interrupt requests occur during SCO download, their interrupt sources are retained.

— IRQ interrupt request
Falling-edge detection or low-level detection can be specified for an IRQ interrupt.

- Falling-edge detection is selected: When the falling-edge of IRQ is detected during SCO download, the interrupt source is retained.
- Low-level detection is selected: When the low-level of IRQ is detected during SCO download, if the IRQ remains low when download ends, the interrupt processing starts. If the IRQ is high when download ends, the interrupt source will be canceled.

— On-chip peripheral module interrupt request
An interrupt from an on-chip peripheral module is requested by input of the specified level. Since the interrupt signal continues to be output unless the interrupt flag is cleared, the interrupt source is retained.

(2) Interrupts during programming/erasing

Though an interrupt processing can be executed at realtime during programming/erasing of the downloaded on-chip program, the following limitations and notes are applied.

1. When flash memory is being programmed or erased, both the user MAT and user boot MAT cannot be accessed. Prepare the interrupt vector table and interrupt processing routine in on-chip RAM or external memory. Make sure the flash memory being programmed or erased is not accessed by the interrupt processing routine. If flash memory is read, the read values are not guaranteed. If the relevant bank in flash memory that is being programmed or erased is accessed, the error protection state is entered, and programming or erasing is aborted. If a bank other than the relevant bank is accessed, the error protection state is not entered but the read values are not guaranteed.
2. Do not rewrite the program data specified by the FMPDR parameter. If new program data is to provided by the interrupt processing, temporarily save the new program data in another area. After confirming the completion of programming, save the new program data in the area specified by FMPDR or change the setting in FMPDR to indicated the other area in which the new program data was temporarily saved.
3. Make sure the interrupt processing routine does not rewrite the contents of the flash-memory related registers or data in the downloaded on-chip program area. During the interrupt processing, do not simultaneously perform RAM emulation, download of the on-chip program by an SCO request, or programming/erasing.
4. At the beginning of the interrupt processing routine, save the CPU register contents. Before returning from the interrupt processing, write the saved contents in the CPU registers again.

If a transition is made to the reset state, the reset signal should only be released after providing a reset input over a period longer than the normal 100 µs to reduce the damage to flash memory.

22.8.3 Other Notes

1. Download time of on-chip program
   The programming program that includes the initialization routine and the erasing program that includes the initialization routine are each 2 kbytes or less. Accordingly, when the CPU clock frequency is 40 MHz, the download for each program takes approximately 75 µs at maximum.
2. User branch processing intervals
   The intervals for executing the user branch processing differs in programming and erasing. The processing phase also differs. Table 22.11 lists the maximum and minimum intervals for initiating the user branch processing when the CPU clock frequency is 40 MHz.

Table 22.11 Initiation Intervals of User Branch Processing

Processing Name	Maximum Interval	Minimum Interval
Programming	Approximately 1 ms	Approximately 19 µs
Erasing	Approximately 5 ms	Approximately 19 µs

Table 22.12 lists the maximum and minimum periods until the user branch processing is initiated when the CPU clock frequency is 40 MHz.

Table 22.12 Required Period for Initiating User Branch Processing

Processing	Max.	Min.
Programming	Approximately 113 µs	Approximately 113 µs
Erasing	Approximately 85 µs	Approximately 45 µs

3. Write to flash-memory related registers by AUD or DMAC
   While an instruction in on-chip RAM is being executed, the AUD or DMAC can write to the SCO bit in FCCS that is used for a download request or FMATS that is used for MAT switching. Make sure that these registers are not accidentally written to, otherwise an on-chip program may be downloaded and damage RAM or a MAT switchover may occur and the CPU get out of control.
4. State in which AUD operation is disabled and interrupts are ignored
   In the following modes or period, the AUD is in module standby mode and cannot operate. The NMI or maskable interrupt requests are ignored; they are not executed and the interrupt sources are not retained.

— Checking the flash-memory related registers immediately after user boot mode is initiated (Approximately 100 µs when operation with internal frequency of 40 MHz is carried out after the reset signal is released.)

5. Compatibility with programming/erasing program of conventional F-ZTAT SH microcomputer
   A programming/erasing program for flash memory used in the conventional F-ZTAT SH microcomputer which does not support download of the on-chip program by a SCO transfer request cannot run in this LSI.
   Be sure to download the on-chip program to execute programming/erasing of flash memory in this LSI.
6. Monitoring runaway by WDT
   Unlike the conventional F-ZTAT SH microcomputer, no countermeasures are available for a runaway by WDT during programming/erasing by the downloaded on-chip program.
   Prepare countermeasures (e.g. use of the user branch routine and periodic timer interrupts) for WDT while taking the programming/erasing time into consideration as required.

22.9 Programmer Mode

Along with its on-board programming mode, this LSI also has programmer mode as another mode for writing and erasing of programs and data. Programmer mode supports memory-read mode, auto-program mode, auto-erase mode, and status-read mode. Programming/erasing is possible on the user MAT and user boot MAT.

A status-polling system is adopted for operation in auto-program mode, auto-erase mode, and status-read mode. In status-read mode, details of the system's internal state are output after execution of automatic programming or automatic erasure.

In programmer mode, set the mode pins as shown in table 22.13, and provide a 6-MHz input-clock signal.

Mode pins: MD2, MD1, and MD0	0, 1, 1
FWE	High-level input (automatic programming and automatic erasure)
$\overline{RES}$	Power-on reset circuit
EXTAL, XTAL, $PLLV_{CC}$, $PLLV_{SS}$, PLLCAP	Oscillation circuit and PLL circuit
V_{CL}	Internal stepdown stabilization capacitor

22.9.1 Pin Arrangement of Socket Adapter

Attach the socket adapter to the LSI in the way shown in figure 22.25. This allows conversion to 40 pins. Figure 22.24 shows the memory mapping of on-chip ROM, and figure 22.25 shows the arrangement of the socket adapter's pins.

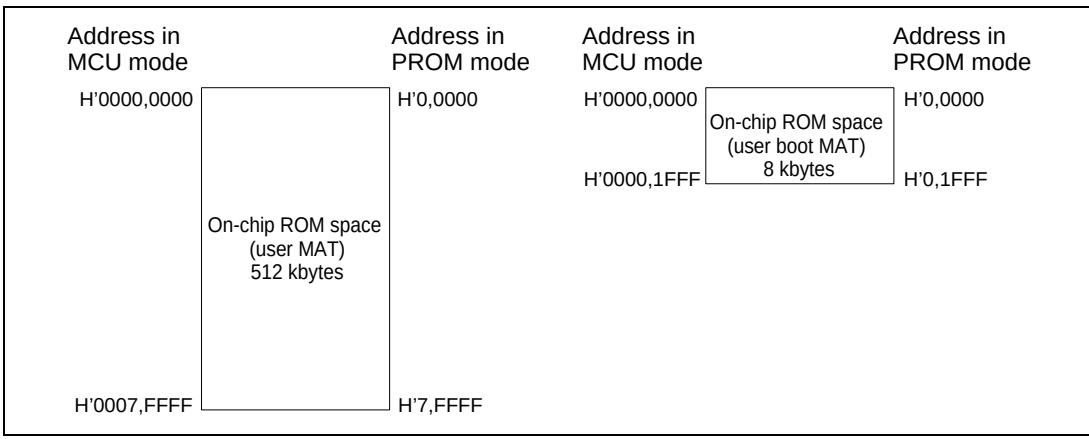


Figure 22.24 Mapping of On-Chip Flash Memory

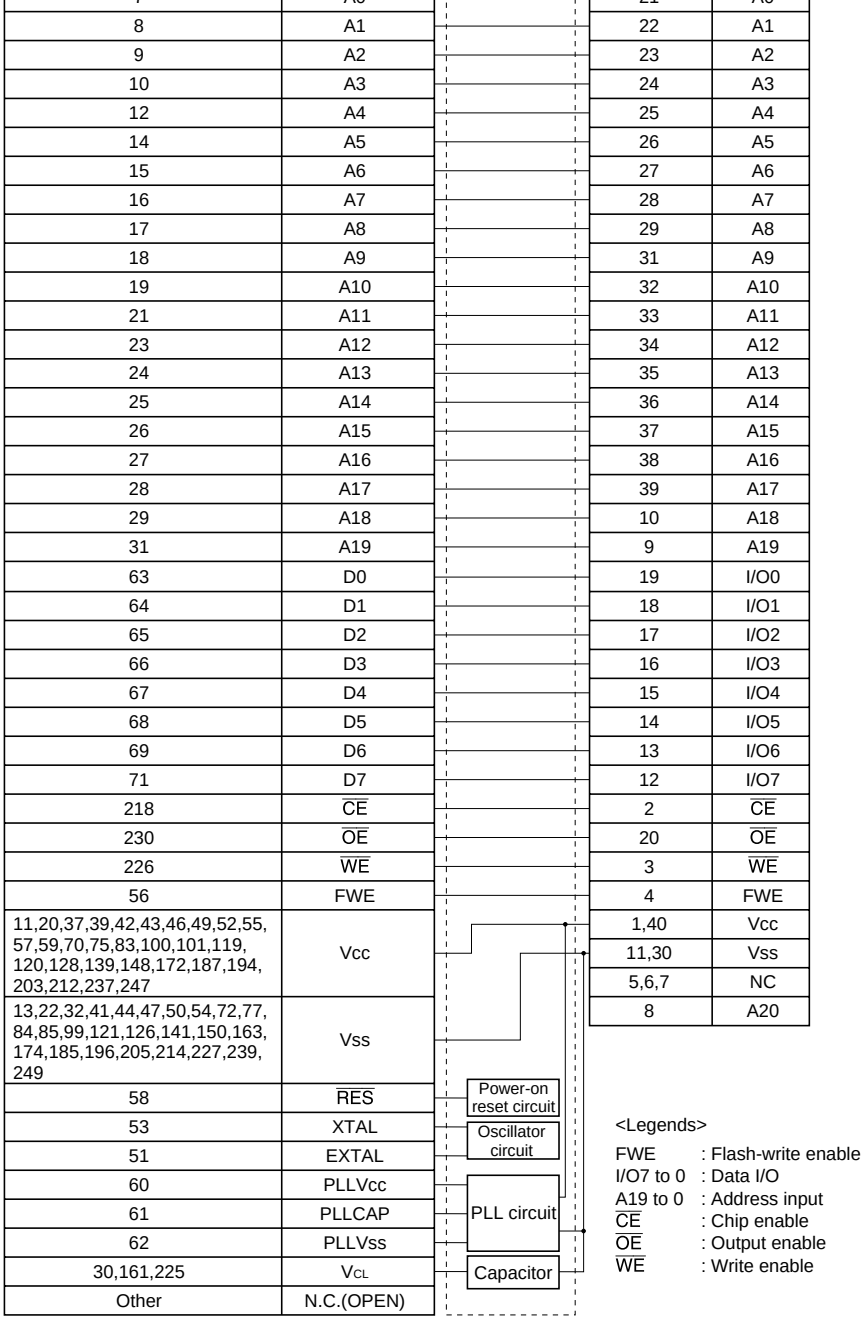


Figure 22.25 Pin Arrangement of Socket Adapter

Table 22.14 shows the settings for the operating modes of programmer mode, and table 22.15 lists the commands used in programmer mode. The following sections provide detailed information on each mode.

- Memory-read mode
  Supports reading from the user MAT or user boot MAT in bytes.
- Auto-program mode
  Supports the simultaneous programming of the user MAT and user boot MAT in 128-byte units. Status polling is used to confirm the end of automatic programming.
- Auto-erase mode
  Supports only automatic erasure of the entire user MAT or user boot MAT. Status polling is used to confirm the end of automatic erasure.
- Status-read mode
  Status polling is used with automatic programming and automatic erasure. Normal completion can be detected by reading the signal on the I/O6 pin. In status-read mode, error information is output when an error has occurred.

Table 22.14 Settings for Each Operating Mode of Programmer Mode

	Pin Name					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**I/O7 to I/O0**	**A19 to A0**
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-Z	X
Command write	H or L	L	H	L	Data input	Ain*
Chip disable	H or L	H	X	X	Hi-Z	X

Notes: 1. The chip-disable mode is not a standby state; internally, it is an operational state.
2. To write commands when making a transition to auto-program or auto-erase mode, input a high-level signal on the FWE pin.

* Ain indicates that there is also an address input in auto-program mode.

Command	Number of Cycles	Memory MAT to be Accessed	Mode	Address	Command	Mode	Address	Data
Memory-read mode	1+n	User MAT	Write	X	H'00	Read	RA	Dout
		User boot MAT	Write	X	H'05			
Auto-program mode	129	User MAT	Write	X	H'40	Write	WA	Din
		User boot MAT	Write	X	H'45			
Auto-erase mode	2	User MAT	Write	X	H'20	Write	X	H'20
		User boot MAT	Write	X	H'25			H'25
Status-read mode	2	Common to both MATs	Write	X	H'71	Write	X	H'71

Notes 1. In auto-program mode, 129 cycles are required in command writing because of the simultaneous 128-byte write.
2. In memory read mode, the number of cycles varies with the number of address writing cycles (n).
3. In an automatic erasure command, input the same command code for the 1st and 2nd cycles (for erasing of the user boot MAT, input H'25 for the 1st and 2nd cycles).

22.9.3 Memory-Read Mode

(1) On completion of automatic programming, automatic erasure, or status read, the LSI enters a command input wait state. So, to read the contents of memory after these operations, issue the command to transit to memory-read mode before reading from the memory.

(2) In memory-read mode, the writing of commands is possible in the same way as in command input wait state.

(3) After entering memory-read mode, continuous reading is possible.

(4) After power has first been supplied, the LSI enters memory-read mode of the user MAP.

For the AC characteristics in memory read mode, see section 22.10.2, AC Characteristics and Timing in programmer Mode.

(1) In auto-program mode, programming is in 128-byte units. That is, 128 bytes of data are transferred in succession.

(2) Even in the programming of less than 128 bytes, 128 bytes of data must be transferred. H'FF should be written to those addresses that are unnecessarily written to.

(3) Set the lower seven bits of the address to be transferred to low level. Inputting an invalid address will result in a programming error, although processing will proceed to the memory-programming operation.

(4) The memory address is transferred in the 2nd cycle. Do not transfer addresses in the 3rd or later cycles.

(5) Do not issue commands while programming is in progress.

(6) When programming, execute automatic programming once for each 128-byte block of addresses. Programming the block at an address where programming has already been performed is not possible.

(7) To confirm the end of automatic programming, check the signal on the I/O6 pin. Confirmation in status-read mode is also possible (status polling of the I/O7 pin is used to check the end status of automatic programming).

(8) Status-polling information on the I/O6 and I/O7 pins is retained until the next command is written. As long as no command is written, the information is made readable by enabling $\overline{CE}$ and $\overline{OE}$.

For the AC characteristics in auto-program mode, see section 22.10.2, AC Characteristics and Timing in programmer Mode.

22.9.5 Auto-Erase Mode

(1) Auto-erase mode only supports erasing of the entire memory.

(2) Do not perform command writing while auto erasing is in progress.

(3) To confirm the end of automatic erasure, check the signal on the I/O6 pin. Confirmation in the status-read mode is also possible (status polling of the I/O7 pin is used to check the end status of automatic erasure).

(4) Status polling information on the I/O6 and I/O7 pins is retained until the next command writing. As long as no command is written, the information is made readable by enabling $\overline{CE}$ and $\overline{OE}$.

For the AC characteristics in auto-erase mode, see section 22.10.2, AC Characteristics and Timing in programmer Mode.

(1) Status-read mode is used to determine the type of an abnormal termination. Use this mode when automatic programming or automatic erasure ends abnormally.

(2) The return code is retained until writing of a command that selects a mode other than status-read mode.

Table 22.16 lists the return codes of status-read mode.

For the AC characteristics in status-read mode, see section 22.10.2, AC Characteristics and Timing in programmer Mode.

Table 22.16 Return Codes of Status-Read Mode

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end indicator	Command error	Programming error	Erasure error	—	—	Programming or erase count exceeded	Invalid address error
Initial value	0	0	0	0	0	0	0	0
Indication	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Programming error: 1 Otherwise: 0	Erasure error:1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Invalid address error: 1 Otherwise: 0

Note: I/O2 and I/O3 are undefined pins.

22.9.7 Status Polling

(1) The I/O7 status-polling output is a flag that indicates the operating status in auto-program or auto-erase mode.

(2) The I/O6 status-polling output is a flag that indicates normal/abnormal end of auto-program or auto-erase mode.

Table 22.17 Truth Table of Status-Polling Output

Pin Name	In Progress	Abnormal End	—	Normal End
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O0 to I/O5	0	0	0	0

Until oscillation has stabilized and while programmer mode is being set up, the LSI is unable to accept commands. After the programmer-mode setup time has elapsed, the LSI enters memory-read mode. For details, see section 22.10.2, AC Characteristics and Timing in Programmer Mode.

22.9.9 Notes on Programming in Programmer Mode

(1) When programming addresses which have previously been programmed, apply auto-erasing before auto-programming.

(2) When using programmer mode to program a chip that has been programmed/erased in an on-board programming mode, auto-erasing before auto-programming is recommended.

(3) Do not take the chip out of the PROM programmer or reset the chip during programming or erasure. Flash memory is susceptible to permanent damage since a high voltage is being applied during the programming/erasing. When the reset signal is accidentally input to the chip, the period in the reset state until the reset signal is released should be longer than the normal 100 µs.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Renesas. For other chips for which the history of erasure is unknown, auto-erasing as a check and supplement for the initialization (erase) level is recommended.

2. Automatic programming to a single address block can only be performed once. Additional programming to an address block that has already been programmed is not allowed.

22.10.1 Serial Communication Interface Specification for Boot Mode

Initiating boot mode enables the boot program to communicate with the host by using the on-chip SCI. The serial communication interface specifications are shown below.

- Status

  The boot program has three states.

(1) Bit-rate-adjustment state

In this state, the boot program adjusts the bit rate to communicate with the host. Initiating boot mode enables starting of the boot program and entry to the bit-rate-adjustment state. The program receives the command from the host to adjust the bit rate. After adjusting the bit rate, the program enters the inquiry/selection state.

(2) Inquiry/Selection state

In this state, the boot program responds to inquiry commands from the host. The device name, clock mode, and bit rate are selected. After selection of these settings, the program is made to enter the programming/erasing state by the command for a transition to the programming/erasing state. The boot program transfers the erasure program to RAM and erases the user MATs and user boot MATs before the transition.

(3) Programming/erasing state

Programming and erasure by the boot program take place in this state. The boot program is made to transfer the programming/erasing program to RAM by commands from the host. Sum checks and blank checks are executed by sending these commands from the host.

These boot program states are shown in figure 22.26.

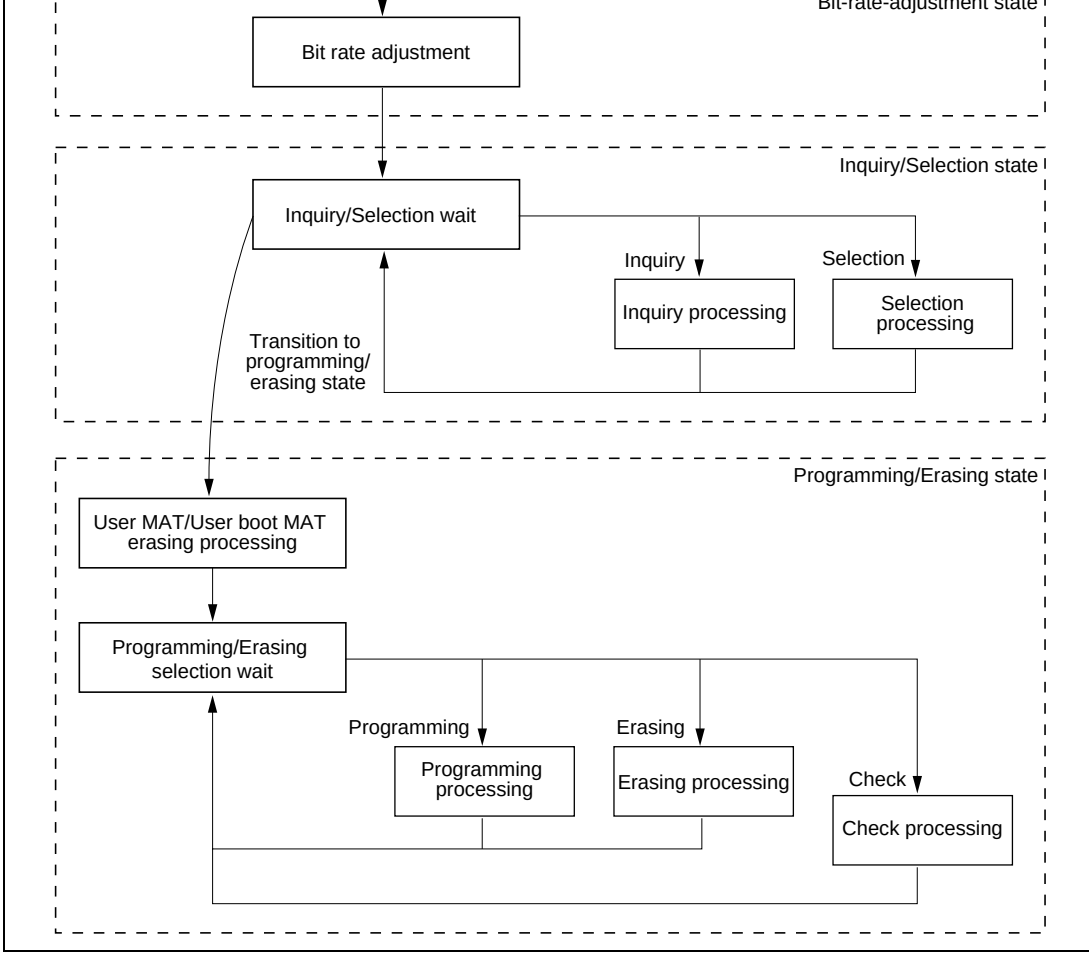


Figure 22.26 Boot Program Processing Flow

- Bit-rate-adjustment state

  The bit rate is calculated by measuring the period of transfer of a low-level byte (H'00) from the host. The bit rate can be changed by the command for a new bit rate selection. After the bit rate has been adjusted, the boot program enters the inquiry/selection state. The bit-rate-adjustment sequence is shown in figure 22.27.

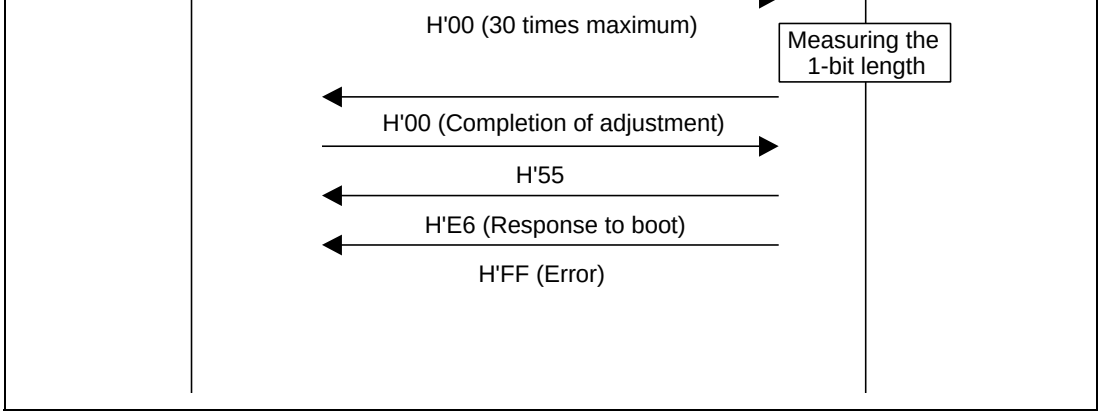


Figure 22.27 Bit-Rate-Adjustment Sequence

- Communications protocol

  After adjustment of the bit rate, the protocol for serial communications between the host and the boot program is as shown below.

(1) One-byte commands and one-byte responses

These commands and responses are comprised of a single byte. These consists of the inquiries and ACK for successful completion.

(2) n-byte commands or n-byte responses

These commands and responses are comprised of n bytes of data. These are selections and responses to inquiries.

The amount of programming data is not included under this heading because it is determined in another command.

(3) Error response

The error response is a response to inquiries. It consists of an error response and an error code and which take up two bytes.

(4) Programming of 128 bytes

The size is not specified in commands. The data size is indicated in response to the programming unit inquiry.

(5) Memory read response

This response consists of four bytes of data.

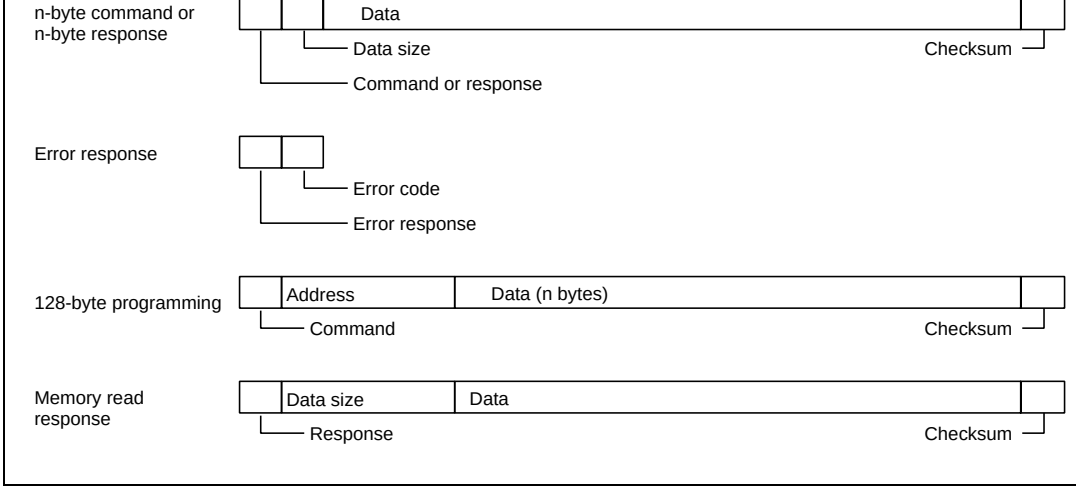


Figure 22.28 Communications Protocol Format

— Command (one byte): Commands including inquiries, selection, programming, erasing, and checking
— Response (one byte): Response to an inquiry
— Size (one or two bytes): The amount of data for transmission excluding the command, amount of data, and checksum
— Data (n bytes): Detailed data of a command or response
— Checksum (one byte): The checksum is calculated so that the total of all values from the command byte to the SUM byte becomes H'00.
— Error Response (one byte): Error response to a command
— Error Code (one byte): Type of the error
— Address (four bytes): Address for programming
— Data (n bytes): Data to be programmed. n is indicated in the response to the programming unit inquiry.
— Data Size (four bytes): Four-byte response to a memory read

- Inquiry/Selection State

  The boot program returns information from the flash memory in response to the host's inquiry commands and sets the device code, clock mode, and bit rate in response to the host's selection command.

  Table 22.18 lists the inquiry and selection commands.

H'20	Supported Device Inquiry	Inquiry regarding device codes and product names of F-ZTAT
H'10	Device Selection	Selection of device code
H'21	Clock Mode Inquiry	Inquiry regarding numbers of clock modes and values of each mode
H'11	Clock Mode Selection	Indication of the selected clock mode
H'22	Multiplication Ratio Inquiry	Inquiry regarding the number of clock types, the number of multiplication/division ratios, and the multiplication/division ratios
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and minimum values of the main clock and peripheral clocks
H'24	User Boot MAT Information Inquiry	Inquiry regarding the number of user boot MATs and the start and last addresses of each MAT
H'25	User MAT Information Inquiry	Inquiry regarding the a number of user MATs and the start and last addresses of each MAT
H'26	Block for Erasing Information Inquiry	Inquiry regarding the number of blocks and the start and last addresses of each block
H'27	Programming Unit Inquiry	Inquiry regarding the unit of programming data
H'28	Two-MAT Simultaneous Programming Information Inquiry	Inquiry into whether or not simultaneous two-MAT programming is allowed
H'3F	New Bit Rate Selection	Selection of new bit rate
H'40	Transition to Programming/Erasing State	Erasing of user MAT and user boot MAT, and entry to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry into the operation status of the boot program

The selection commands, which are device selection (H'10), clock mode selection (H'11), and new bit rate selection (H'3F), should be sent from the host in this order. These commands are certainly required. When two or more selection commands are sent at once, the last command will be valid.

All of these commands, except for the boot program status inquiry command (H'4F), will be valid until the boot program receives the programming/erasing transition (H'40). The host can choose the needed commands out of the commands and inquiries listed above. The boot program status

(1) Supported device inquiry

The boot program will return the device codes of supported devices in response to the supported device inquiry.

Command

H'20

— Command: H'20 (one byte): Inquiry regarding supported devices

Response

H'30	Size	Number of devices	
Number of characters	Device code		Product name
...			
SUM			

— Response: H'30 (one byte): Response to the supported device inquiry

— Size (one byte): Number of bytes to be transmitted, excluding the command, amount of data, and checksum, that is, the amount of data consists of the product names, the number of devices, characters, and device codes

— Number of devices (one byte): Number of device types supported by the boot program

— Number of characters (one byte): Number of characters in the device code and boot program's name

— Device code (four bytes): Supporting product (ASCII code)

— Product name (n bytes): Type name of the boot program (ASCII code)

— SUM (one byte): Checksum
The checksum is calculated so that the total number of all values from the command byte to the SUM byte becomes H'00.

(2) Device Selection

The boot program will set the supported device to the specified device code. The program will return the selected device code in response to the inquiry after this setting has been made.

Command

H'10	Size	Device code	SUM

— Command: H'10 (one byte): Device selection

— Size (one byte): Number of characters in the device code (fixed at 2)

— Device code (four bytes): Device code returned in response to the supported device inquiry (ASCII code)

— SUM (one byte): Checksum

Response

H'06

— Response: H'06, (one byte): Response to the device selection command
ACK will be returned when the device code matches.

— ERROR: (one byte): Error code
H'11: Sum check error
H'21: Device code mismatch error

(3) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode inquiry.

Command

H'21

— Command: H'21 (one byte): Inquiry regarding clock mode

Response

H'31	Size	Number of modes	Mode	SUM

— Response: H'31 (one byte): Response to the clock-mode inquiry

— Size (one byte): Amount of data that represents the number of modes and modes

— Number of modes (one byte): Number of supported clock modes
H'00 indicates no clock mode or the device allows the clock mode to be read.

— Mode (one byte): Supported clock modes (i.e. H'01 means clock mode 1.)

— SUM (one byte): Checksum

(4) Clock Mode Selection

The boot program will set the specified clock mode. The program will return the selected clock-mode information after this setting has been made.

The clock-mode selection command should be sent after the device selection command.

Command

H'11	Size	Mode	SUM

— Command: H'11 (one byte): Selection of clock mode

— Size (one byte): Number of characters that represents the mode (fixed at 1)

— Mode (one byte): Clock mode returned in reply to the supported clock mode inquiry.

— SUM (one byte): Checksum

Response

H'06

— Response: H'06 (one byte): Response to the clock-mode selection command
ACK will be returned when the clock mode matches.

Error response

H'91	ERROR

— Error response: H'91 (one byte): Error response to the clock-mode selection command

— ERROR (one byte): Error code
H'11: Sum check error
H'22: Clock mode mismatch error

(5) Multiplication Ratio Inquiry

The boot program will return the supported multiplication/division ratios.

Response

H'32	Size	Number of clock types					
Number of multiplication ratios	Multiplica-tion ratio	???					
...							
SUM							

— Response: H'32 (one byte): Response to the multiplication ratio inquiry

— Size (one byte): Amount of data that represents the number of clock types, the number of multiplication ratios, and the multiplication ratios

— Number of clock types (one byte): Number of supported multiplied clock types
(e.g. when there are two multiplied clock types, which are the main operating frequency and the peripheral module operating frequency, the number of types will be H'02)

— Number of multiplication ratios (one byte): Number of multiplication ratios for each operating frequency
(e.g. the number of multiplication ratios to which the main operating frequency can be set and the peripheral module operating frequency can be set)

— Multiplication ratio (one byte)
Multiplication ratio: Value of the multiplication ratio (e.g. when the clock-frequency multiplier is four, the value of multiplication ratio will be H'04)
Division ratio: Value of the division ratio, inverted to be a negative number (e.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = –2)
The number of multiplication ratios returned is the same as the number of multiplication ratios and as many groups of data are returned as there are types.

— SUM (one byte): Checksum

(6) Operating Clock Frequency Inquiry

The boot program will return the number of operating clock frequencies, and the maximum and minimum values.

Command

H'23

— Command: H'23, (one byte): Inquiry regarding operating clock frequencies

Response

H'33	Size	Number of operating clock frequencies
Minimum value of operating clock frequency		Maximum value of operating clock frequency
...		
SUM		

— Response: H'33 (one byte): Response to operating clock frequency inquiry

— Number of types (one byte): Number of supported operating clock frequency types (e.g. when there are two operating clock frequency types, which are the main and peripheral clocks, the number of types will be H'02)

— Minimum value of operating clock frequency (two bytes): Minimum value for each multiplied or divided clock frequency.
The minimum and maximum values represent the values in MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the value is 20.00 MHz, it will be multiplied by 100 to be 2000 which is H'07D0)

— Maximum value of operating clock frequency (two bytes): Maximum value for each multiplied or divided clock frequency.
There are as many pairs of minimum and maximum values as there are operating clock frequencies.

— SUM (one byte): Checksum

(7) User Boot MAT Information Inquiry

The boot program will return the number of user boot MATs and their addresses.

Command

H'24

— Command: H'24 (one byte): Inquiry regarding user boot MAT information

Response

H'34	Size	Number of areas	
Start address of area			Last address of area
...			
SUM			

— Response: H'34 (one byte): Response to user boot MAT information inquiry

— Size (one byte): Amount of data that represents the number of areas, the start address of each area, and the last address of each area

— Number of areas (one byte): Number of non-consecutive user boot MAT areas
When user boot MAT areas are consecutive, the number of areas returned is H'01.

— Start address of area (four bytes): Start address of the area

— Last address of area (four bytes): Last address of the area
There are as many groups of data representing the start and last addresses as there are areas.

— SUM (one byte): Checksum

(8) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command

H'25

— Command: H'25 (one byte): Inquiry regarding user MAT information

...	

SUM

— Response: H'35 (one byte): Response to the user MAT information inquiry

— Size (one byte): Amount of data that represents the number of areas, the start address of each area, and the last address of each area

— Number of areas (one byte): Number of non-consecutive user MAT areas
When user MAT areas are consecutive, the number of areas returned is H'01.

— Start address of area (four bytes): Start address of the area

— Last address of area (four bytes): Last address of the area
There are as many groups of data representing the start and last addresses as there are areas.

— SUM (one byte): Checksum

(9) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command

H'26

— Command: H'26 (one byte): Inquiry regarding erased block information

Response

H'36	Size	Number of blocks
Start address of block		Last address of block
...		

SUM

— Response: H'36 (one byte): Response to the number of erased blocks and addresses

— Size (two bytes): Amount of data that represents the number of blocks, the start address of each block, and the last address of each block

— Number of blocks (one byte): Number of erased blocks in flash memory

— Start address of block (four bytes): Start address of the block

— Last address of block (four bytes): Last address of the block
There are as many groups of data representing the start and last addresses as there are blocks.

— SUM (one byte): Checksum

(10) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command

H'27

— Command: H'27 (one byte): Inquiry regarding programming unit

— Size (one byte): Number of characters that indicate the programming unit (fixed at 2)
— Programming unit (two bytes): Unit for programming
This is the unit for reception of program data.
— SUM (one byte): Checksum

(11) Two-MAT Simultaneous Programming Information Inquiry

The boot program will return an indication whether or not two-MAT simultaneous programming is allowed and the start address.

Command

H'28

— Command: H'28 (one byte): Inquiry regarding two-MAT simultaneous programming information

Response

H'38	Size	Programming method	
Start address of 1st MAT			Start address of 2nd MAT
SUM			

— Response: H'38 (one byte): Response to 2-MAT simultaneous programming information inquiry
— Size (one byte): Amount of data that represents the programming method and start addresses, which is fixed at five bytes for one-MAT programming and at nine bytes for two-MAT simultaneous programming.
— Programming method (one byte): H'01 = one-MAT programming
H'02 = two-MAT simultaneous programming
— Start address of 1st MAT (four bytes): Start address of the first MAT
— Start address of 2nd MAT (four bytes): Start address of the second MAT
The start address of the second MAT is included only when the two-MAT simultaneous programming method is allowed.
— SUM (one byte): Checksum

(12) New Bit Rate Selection

The boot program will set a new bit rate and return the new bit rate.

This selection should be sent after sending the clock-mode selection command.

Command

H'3F	Size	Bit rate	Input frequency
Number of multiplication ratios	Multiplication ratio 1	Multiplication ratio 2	
SUM			

— Command: H'3F (one byte): Selection of new bit rate
— Size (one byte): Amount of data that represents the bit rate, input frequency, number of multiplication ratios, and multiplication ratios

H'00C0)

— Input frequency (two bytes): Frequency of the clock input to the boot program
This value is valid to the hundredths place and represents the value in MHz multiplied by 100. (e.g. when the value is 28.882 MHz, it will be multiplied by 100 to be 2888 which is H'0B48.

— Number of multiplication ratios (one byte): Number of multiplication ratios to which the device can be set.

— Multiplication ratio 1 (one byte): Value of the multiplication or division ratio for the main operating frequency
Multiplication ratio: Value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)
Division ratio: Value of the division ratio, inverted to be a negative number (e.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = –2)

— Multiplication ratio 2 (one byte): Value of the multiplication or division ratio for the peripheral operating frequency
Multiplication ratio: Value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)
Division ratio: Value of the division ratio, inverted to be a negative number (e.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = –2)

— SUM (one byte): Checksum

Response

H'06

— Response: H'06 (one byte): Response to selection of a new bit rate
When it is possible to set the bit rate, the response will be ACK.

Error response

H'BF	ERROR

— Error response: H'BF (one byte): Error response to selection of new bit rate

— ERROR: (one byte): Error code
H'11: Sum check error
H'24: Bit-rate selection error
This bit rate is not available.
H'25: Input frequency error
This input frequency is not within the range set by the minimum and maximum values.
H'26: Multiplication ratio error
This ratio does not match an available ratio.
H'27: Operating frequency error
This operating frequency is not within the range set by the minimum and maximum values.

The methods for checking of received data are listed below.

minimum to maximum frequencies which matches the clock modes of the specified device. When the value is out of this range, an input frequency error is generated.

- Multiplication ratio

  The received value of the multiplication ratio or division ratio is checked to ensure that it matches the clock modes of the specified device. When the value is out of this range, a multiplication error is generated.

- Operating frequency error

  The operating frequency is calculated from the received value of the input frequency and the multiplication or division ratio. The input frequency is input to the LSI and the LSI is actually operated at the operating frequency. The expression is given below.

  Operating frequency = Input frequency Multiplication ratio, or

  Operating frequency = Input frequency/Division ratio

  The calculated operating frequency should be checked to ensure that it is within the range of minimum to maximum frequencies which are available with the clock modes of the specified device. When it is out of this range, an operating frequency error is generated.

- Bit rate

  From peripheral operating clock (ϕ) and bit rate (B), the clock select (CKS) value (n) in the serial mode register (SMR) and the bit rate register (BRR) value (N) are obtained. The error between n and N that is calculated by the method below is checked to ensure that it is less than 4%. When it is 4% or more, a bit-rate selection error is generated.

$$\text{Error (\%)} = \{[\ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{(2 \times n-1)}}\ ] - 1\} \times 100$$

  When the new bit rate is selectable, the new bit rate will be set in the register after sending ACK in response. The host will send ACK with the new bit rate for confirmation and the boot program will response with that rate.

Confirmation | H'06 |

— Confirmation: H'06 (one byte): Confirmation of a new bit rate

Response | H'06 |

— Response: H'06 (one byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 22.29.

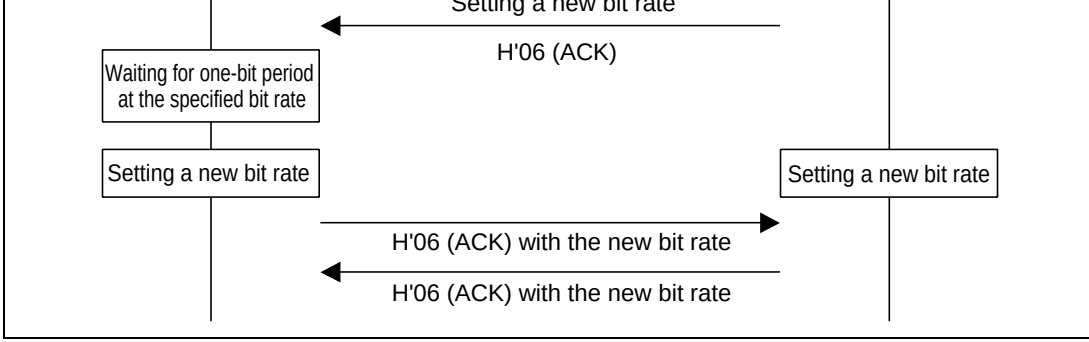


Figure 22.29 New Bit-Rate Selection Sequence

(13) Transition to Programming/Erasing State

To enter the programming/erasing state, the boot program will transfer the erasing program, and erase the user MATs and user boot MATs in that order. On completion of this erasure, ACK will be returned and a transition is made to the programming/erasing state.

The host should select the device code, clock mode, and new bit rate with device selection, clock-mode selection, and new bit-rate selection commands, and then send the command for the transition to programming/erasing state. This procedure should be carried out before transferring the programming selection command or program data.

Command	H'40

— Command: H'40 (one byte): Transition to programming/erasing state

Response	H'06

— Response: H'06 (one byte): Response to transition to programming/erasing state
The boot program will send ACK when the user MATs and user boot MATs have been erased by the transferred erasing program.

Error response	H'C0	H'51

— Error response: H'C0 (one byte): Error response to transition to programming/erasing state

— Error code: H'51 (one byte): Erasing error
An error occurred and erasure was not completed.

Command Error: A command error will occur when a command is undefined, the order of commands is incorrect, or a command is unacceptable. Issuing a clock-mode selection command before a device selection or issuing an inquiry command after the command for transition to the programming/erasing state, are examples.

Error response	H'80	H'xx

— Error response: H'80 (one byte): Command error

— Command: H'xx (one byte): Received command

(1) A supported device inquiry (H'20) should be made to inquire about the supported devices.

(2) The device should be selected from among those described by the returned information and set with a device selection (H'10) command.

(3) A clock-mode inquiry (H'21) should be made to inquire about the supported clock modes.

(4) The clock mode should be selected from among those described by the returned information and set with a clock-mode selection (H'11) command.

(5) After selection of the device and clock mode, inquiries for other required information should be made, such as the multiplication ratio inquiry (H'22) or operating frequency inquiry (H'23).

(6) A new bit rate should be selected with the new bit-rate selection (H'3F) command, according to the returned information on multiplication ratios and operating frequencies.

(7) After selection of the device and clock mode, the information of the user boot MAT and user MAT should be made to inquire about the user boot MAT information inquiry (H'24), user MAT information inquiry (H'25), erased block information inquiry (H'26), programming unit inquiry (H'27), and two-MAT simultaneous programming information inquiry (H'28).

(8) After making inquiries and selecting a new bit rate, issue the command for transition to the programming/erasing state (H'40). The boot program will then enter the programming/erasing state.

Programming/Erasing State: In the programming/erasing state, a programming selection command makes the boot program select the programming method, a 128-byte programming command makes it program the memory with data, and an erasing selection command and block erasing command make it erase the block. Table 22.19 lists the programming/erasing commands.

H'42	User boot MAT programming selection	Transfers the user boot MAT programming program
H'43	User MAT programming selection	Transfers the user MAT programming program
H'44	Two-user-MAT simultaneous programming selection	Transfers the two-user-MAT simultaneous programming program
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the checksum of the user boot MAT
H'4B	User MAT sum check	Checks the checksum of the user MAT
H'4C	User boot MAT blank check	Checks whether the contents of the user boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of the user MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's state

Programming: Programming is executed by a programming selection command and a 128-byte programming command.

First, the host should send the programming selection command and select the programming method and programming MATs. There are three programming selection commands used according to the area and method for programming.

(1) User boot MAT programming selection

(2) User MAT programming selection

(3) Two-user-MAT simultaneous programming selection

After issuing the programming selection command, the host should send the 128-byte programming command. The 128-byte programming command that follows the selection command represents the data programmed according to the method specified by the selection command. When more than 128-byte data is programmed, 128-byte commands should repeatedly be executed. Sending a 128-byte programming command with H'FFFFFFFF as the address will stop the programming. On completion of programming, the boot program will wait for selection of programming or erasing.

The programming selection command and sequence for the 128-byte programming commands are shown in figure 22.30.

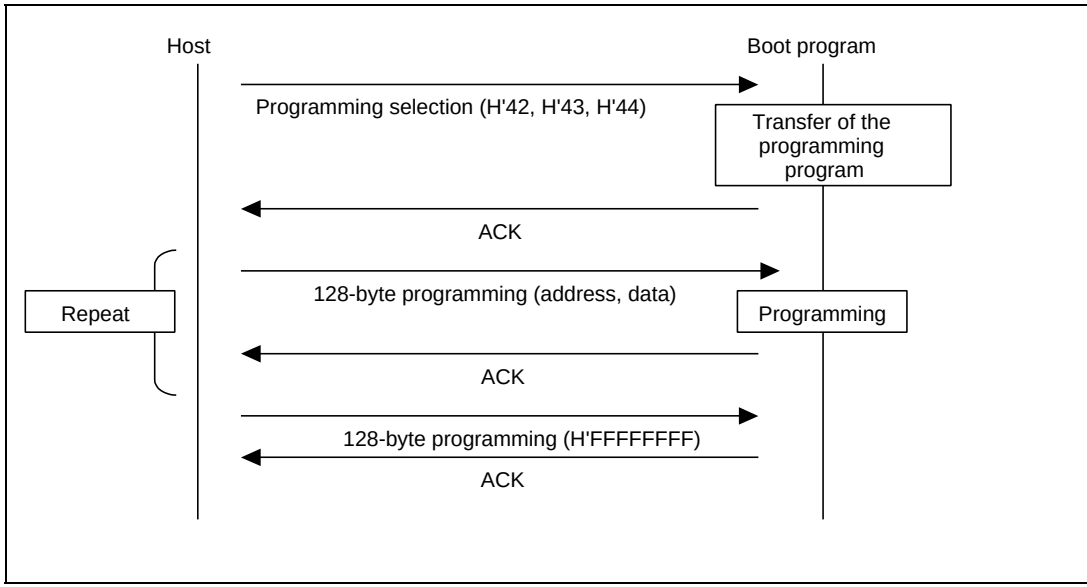


Figure 22.30 Programming Sequence

(1) User boot MAT programming selection

The boot program will transfer a programming program. The data is programmed to the user boot MATs by the transferred programming program.

Command | H'42 |

— Command: H'42 (one byte): User boot MAT programming selection

Response | H'06 |

— Response: H'06 (one byte): Response to user boot MAT programming selection
When the programming program has been transferred, the boot program will return ACK.

Error response | H'C2 | ERROR |

— Error response: H'C2 (one byte): Error response to user boot MAT programming selection

— ERROR: (one byte): Error code
H'54: Selection processing error (transfer error occurs and processing is not completed)

(2) User MAT programming selection

The boot program will transfer a programming program. The data is programmed to the user MATs by the transferred programming program.

Response

H'06

— Response: H'06 (one byte): Response to user MAT programming selection
When the programming program has been transferred, the boot program will return ACK.

Error response

H'C3	ERROR

— Error response: H'C3 (one byte): Error response to user MAT programming selection

— ERROR: (one byte): Error code
H'54: Selection processing error (transfer error occurs and processing is not completed)

(3) Two-user-MAT simultaneous programming selection

The boot program will transfer the two-user-MAT simultaneous programming program. Data is simultaneously programmed to the two user MATs by the transferred two-user-MAT simultaneous programming program. The host must alternately send addresses and data that correspond to each MAT for simultaneous programming to two user MATs. The boot program will return one ACK for one 128-byte programming command, however, programming of the data will start when the boot program has received data for both MATs.

Command

H'44

— Command: H'44 (one byte): Two-user-MAT simultaneous programming selection

Response

H'06

— Response: H'06 (one byte): Response to two-user-MAT simultaneous programming selection
After the two-user-MAT simultaneous programming program has been transferred, the boot program will return ACK.

Error response

H'C4	ERROR

— Error response: H'C4 (one byte): Error response to two-user-MAT simultaneous programming selection

— ERROR: (one byte): Error code
H'54: Selection processing error (transfer error occurs and processing is not completed)

(4) 128-byte programming

The boot program will use the programming program transferred by the programming selection command for programming the user boot MATs or user MATs. When two-user-MAT simultaneous programming command is selected, programming will start after the boot program has received data for both MATs.

Command

H'50	Programming address						
Data	...						
...							
SUM							

— Command: H'50 (one byte): 128-byte programming

boundary (e.g. H'00, H'01, H'00, H'01000000)

— Data (n bytes): Data to be programmed
The size is specified in response to the programming unit inquiry.

— SUM (one byte): Checksum

Response

H'06

— Response: H'06 (one byte): Response to 128-byte programming
On completion of programming, the boot program will return ACK. In two-MAT programming, when all data for the first MAT has been received, the boot program will return ACK.

Error response

H'D0	ERROR

— Error response: H'D0 (one byte): Error response to 128-byte programming

— ERROR: (one byte): Error code
H'11: Sum check error
H'2A: Address error (address is not within the specified range)
H'53: Programming error (a programming error has occurred and programming cannot be continued)

The specified address should match the unit for programming of data. For example, when the programming is in 128-byte units, the lower byte of the address should be H'00 or H'80.

When there are less than 128 bytes of data to be programmed, the host should fill the rest with H'FF.

In two-user-MAT simultaneous programming, the host should alternately send the data for each MAT address.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop the programming operation. The boot program will interpret this as the end of programming and wait for selection of programming or erasing. When the most recently received data has not been programmed in two-user-MAT simultaneous programming, the most recent data is programmed before programming is stopped.

Command

H'50	Programming address	SUM

— Command: H'50 (one byte): 128-byte programming

— Programming address (four bytes): End code is H'FF, H'FF, H'FF, H'FF.

— SUM (one byte): Checksum

Error response

H'D0	ERROR

— Error response: H'D0 (one byte): Error response to 128-byte programming

H'53: Programming error

An error has occurred in programming, and programming cannot be continued (in two-user-MAT simultaneous programming, when programming to the last MAT has not been completed.)

Erasure: Erasure is performed with the erasing selection and block erasing command.

First, erasure is selected by the erasing selection command and the boot program then erases the block specified by the block erasing command. The command should be repeatedly executed if two or more blocks are to be erased. Sending a block erasing command from the host with the block number H'FF will stop erasure. On completion of erasing, the boot program will wait for selection of programming or erasing.

The erasing selection command and sequence for erasing data are shown in figure 22.31.

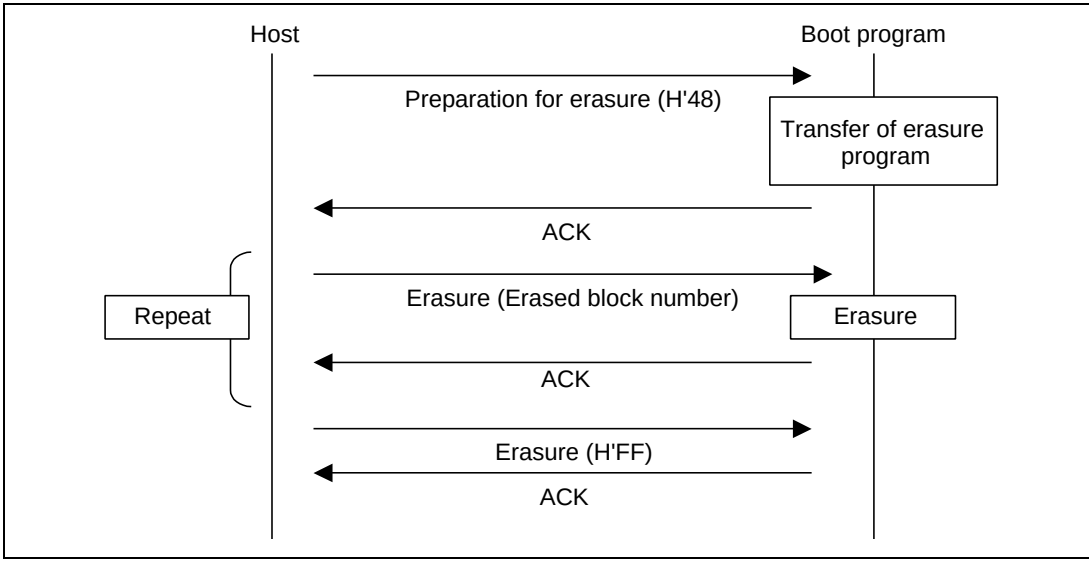


Figure 22.31 Erasing Sequence

(1) Erasing selection

The boot program will transfer the erasing program. User MAT data is erased by the transferred erasing program.

Command	H'48

— Command: H'48 (one byte): Erasing selection

After the erasing program has been transferred, the boot program will return ACK.

Error response

H'C8	ERROR

— Error response: H'C8 (one byte): Error response to erasing selection

— ERROR: (one byte): Error code
H'54: Selection processing error (transfer error occurs and processing is not completed)

(2) Block erasing

The boot program will erase the contents of the specified block.

Command

H'58	Size	Block number	SUM

— Command: H'58 (one byte): Erasing

— Size (one byte): Number of characters that represents the erasure block number (fixed at 1)

— Block number (one byte): Number of the block whose data is to be erased

— SUM (one byte): Checksum

Response

H'06

— Response: H'06 (one byte): Response to erasing
After erasure has been completed, the boot program will return ACK.

Error response

H'D8	ERROR

— Error response: H'D8 (one byte): Error response to erasing
H'11: Sum check error
H'29: Block number error
Block number is incorrect.
H'51: Erasure error
An error has occurred during erasure.
On receiving block number H'FF, the boot program will stop erasure and wait for a selection command.

Command

H'58	Size	Block number	SUM

— Command: H'58 (one byte): Erasure

— Size (one byte): Number of characters that represents the block number (fixed at 1)

— Block number (one byte): H'FF (stop code for erasure)

— SUM (one byte): Checksum

Response

H'06

— Response: H'06 (one byte): Response to end of erasure (ACK)
When erasure is to be performed again after the block number H'FF has been sent, the procedure should be executed from the erasure selection command.

(3) Memory read

The boot program will return the data in the specified address.

— Command: H'52 (one byte): Memory read

— Size (one byte): Amount of data that represents the area, read address, and read size (fixed at 9)

— Area (one byte)
H'00: User boot MAT
H'01: User MAT
An address error occurs when the area setting is incorrect.

— Read start address (four bytes): Start address to be read from

— Read size (four bytes): Size of data to be read

— SUM (one byte): Checksum

Response

H'52	Read size						
Data	...						
SUM							

— Response: H'52 (one byte): Response to memory read

— Read size (four bytes): Size of data to be read

— Data (n bytes): Data for the read size from the read address

— SUM (one byte): Checksum

Error response

H'D2	ERROR

— Error response: H'D2 (one byte): Error response to memory read

— ERROR: (one byte): Error code
H'11: Sum check error
H'2A: Address error
The start address for reading is not in the MAT.
H'2B: Size error
The read size exceeds the MAT, the last address for reading calculated from the start address for reading and the read size is not in the MAT, or read size is 0.

(4) User boot MAT sum check

The boot program will add the amount of data in user boot MATs and return the result.

Command

H'4A

— Command: H'4A (one byte): Sum check of user boot MATs

Response

H'5A	Size	MAT checksum	SUM

— Response: H'5A (one byte): Response to sum check of user boot MATs

— Size (one byte): Number of characters in checksum data (fixed at 4)

— MAT checksum (four bytes): Checksum of user boot MATs
The total amount of data is obtained in byte units.

— SUM (one byte): Checksum (for transmit data)

Command | H'4B |

— Command: H'4B (one byte): Sum check of user MATs

Response	H'5B	Size	MAT checksum	SUM

— Response: H'5B (one byte): Response to sum check of user MATs

— Size (one byte): Number of characters in checksum data (fixed at 4)

— MAT checksum (four bytes): Checksum of user MATs
The total amount of data is obtained in byte units.

— SUM (one byte): Checksum (for transmit data)

(6) User boot MAT blank check

The boot program will check whether or not all user boot MATs are blank and return the result.

Command | H'4C |

— Command: H'4C (one byte): Blank check of user boot MATs

Response | H'06 |

— Response: H'06 (one byte): Response to blank check of user boot MATs
If all user boot MATs are blank (H'FF), the boot program will return ACK.

Error response	H'CC	H'52

— Error response: H'CC (one byte): Error response to blank check of user boot MATs

— Error code: H'52 (one byte): Erasure has not been completed

(7) User MAT blank check

The boot program will check whether or not all user MATs are blank and return the result.

Command | H'4D |

— Command: H'4D (one byte): Blank check of user MATs

Response | H'06 |

— Response: H'06 (one byte): Response to blank check of user MATs
If all user MATs are blank (H'FF), the boot program will return ACK.

Error response	H'CD	H'52

— Error response: H'CD (one byte): Error response to blank check of user MATs

— Error code: H'52 (one byte): Erasure has not been completed.

(8) Boot program status inquiry

The boot program will return indications of its present state and error condition. This inquiry can be made in the inquiry/selection state or the programming/erasing state.

Command | H'4F |

— Command: H'4F (one byte): Inquiry regarding boot program status

— Size (one byte): Number of characters in data (fixed at 2)

— STATUS (one byte): Standard boot program status
For details, see table 22.20, Status Code.

— ERROR (one byte): Error state
ERROR = 0 indicates normal operation.
ERROR = 1 indicates error has occurred.
For details, see table 22.21, Error Code.

— SUM (one byte): Checksum

Table 22.20 Status Code

Code	Description
H'11	Device Selection Wait
H'12	Clock Mode Selection Wait
H'13	Bit Rate Selection Wait
H'1F	Programming/Erasing State Transition Wait (bit rate selection is completed)
H'31	Programming State for Erasing User MAT and User Boot MAT
H'3F	Programming/Erasing Selection Wait (Erasure is completed)
H'4F	Programming Data Receive Wait
H'5F	Erasure Block Specification Wait (erasure is completed)

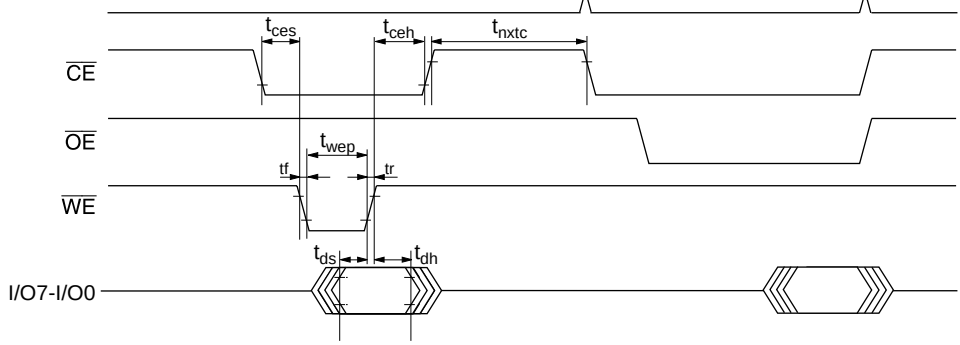
H'00	No Error
H'11	Sum Check Error
H'21	Device Code Mismatch Error
H'22	Clock Mode Mismatch Error
H'24	Bit Rate Selection Error
H'25	Input Frequency Error
H'26	Multiplication Ratio Error
H'27	Operating Frequency Error
H'29	Block Number Error
H'2A	Address Error
H'2B	Data Length Error
H'51	Erasure Error
H'52	Erasure Incompletion Error
H'53	Programming Error
H'54	Selection Error
H'80	Command Error
H'FF	Bit-Rate-Adjustment Confirmation Error

22.10.2 AC Characteristics and Timing in Programmer Mode

Table 22.22 AC Characteristics in Memory Read Mode

Condition: V_{CC} = 3.3 V ± 0.3 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Code	Symbol	Min	Max	Unit	Note
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Programming pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



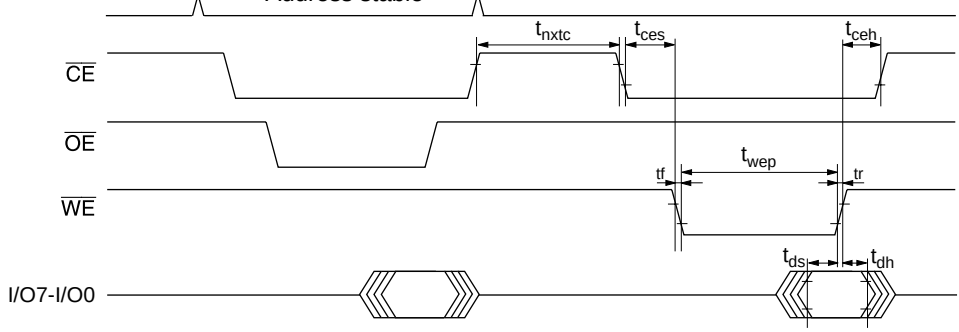
Note : Data is latched at the rising edge of $\overline{WE}$.

Figure 22.32 Memory Read Timing after Command Write

Table 22.23 AC Characteristics in Transition from Memory Read Mode to Others

Condition: V_{CC} = 3.3 V ± 0.3 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Code	Symbol	Min	Max	Unit	Note
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Programming pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



Note : $\overline{WE}$ and $\overline{OE}$ should not be enabled simultaneously.

Figure 22.33 Timing at Transition from Memory Read Mode to Other Modes

Table 22.24 AC Characteristics in Memory Read Mode

Condition: V_{CC} = 3.3 V ± 0.3 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Code	Symbol	Min	Max	Unit	Note
Access time	t_{acc}		20	µs	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Output disable delay time	t_{df}		100	ns	
Data output hold time	t_{oh}	5		ns	

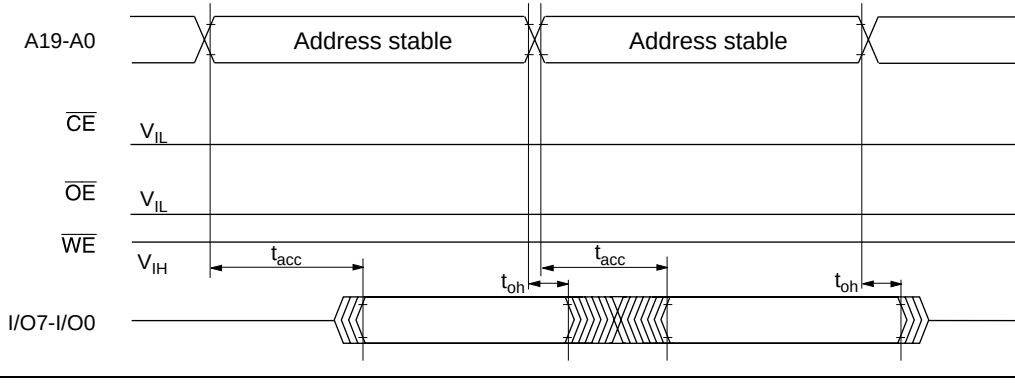


Figure 22.34 $\overline{CE}$/$\overline{OE}$ Enable State Read

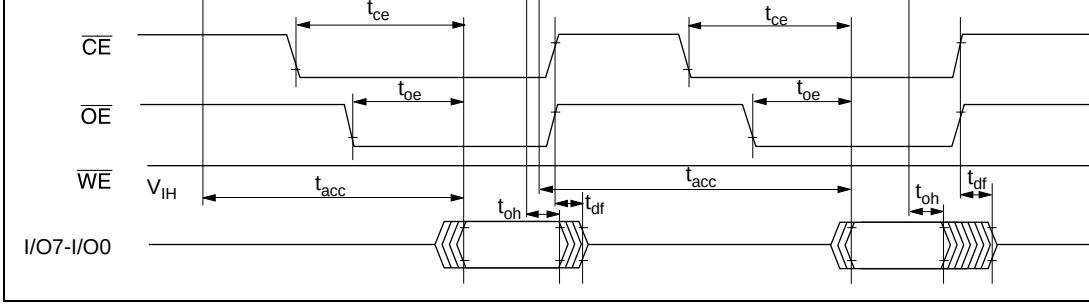


Figure 22.35 $\overline{CE}$/$\overline{OE}$ Clock Read

Table 22.25 AC Characteristics in Auto-Program Mode

Condition: V_{CC} = 3.3 V ± 0.3 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Code	Symbol	Min	Max	Unit	Note
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Programming pulse width	t_{wep}	70		ns	
Status polling start time	t_{wsts}	1		ms	
Status polling access time	t_{spa}		150	ns	
Address setup time	t_{as}	0		ns	
Address hold time	t_{ah}	60		ns	
Memory programming time	t_{write}	1	3000	ms	
Programming setup time	t_{pns}	100		ns	
Programming end setup time	t_{pnh}	100		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

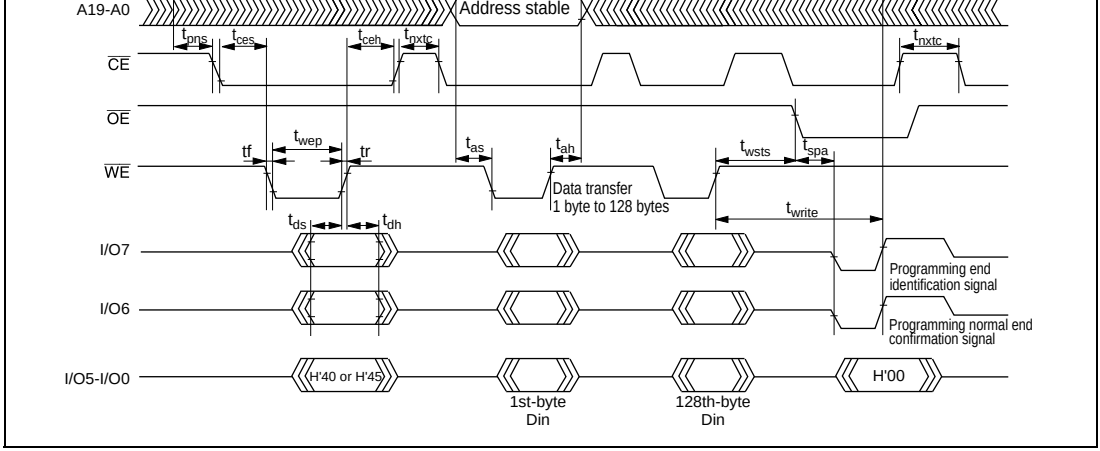


Figure 22.36 Timing in Auto-Program Mode

Table 22.26 AC Characteristics in Auto-Erase Mode

Condition: V_{CC} = 3.3 V ± 0.3 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Code	Symbol	Min	Max	Unit	Note
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Programming pulse width	t_{wep}	70		ns	
Status polling start time	t_{ests}	1		ms	
Status polling access time	t_{spa}		150	ns	
Memory erase time	t_{erase}	100	40000	ms	
Erase setup time	t_{ens}	100		ns	
Erase end setup time	t_{enh}	100		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

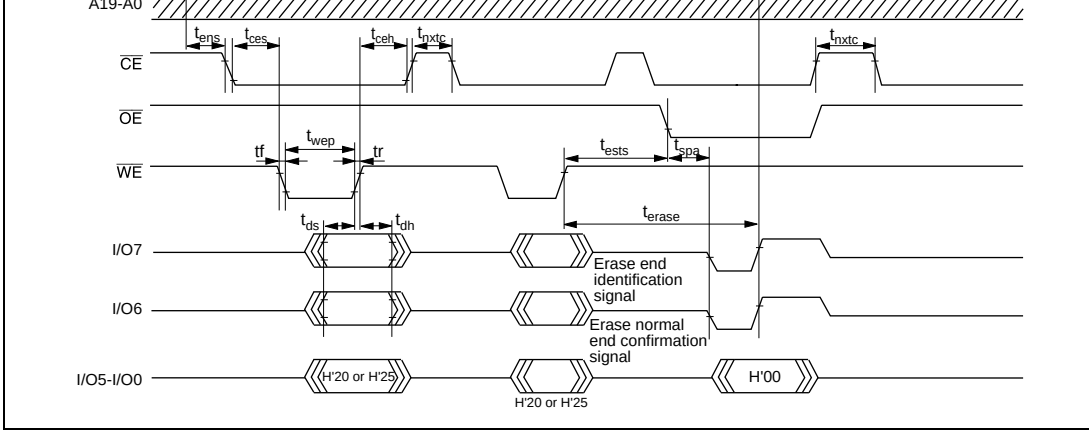
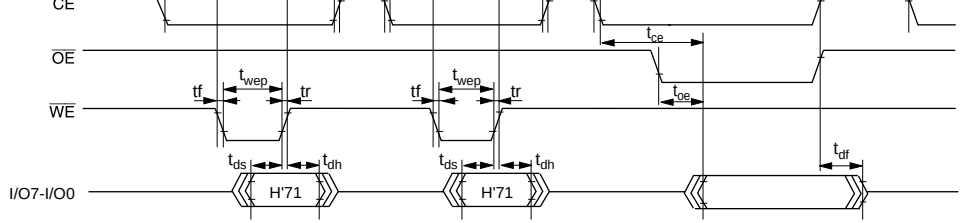


Figure 22.37 Timing in Auto-Erase Mode

Table 22.27 AC Characteristics Status Read Mode

Condition: V_{CC} = 3.3 V ± 0.3 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Code	Symbol	Min	Max	Unit	Note
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Programming pulse width	t_{wep}	70		ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Disable delay time	t_{df}		100	ns	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



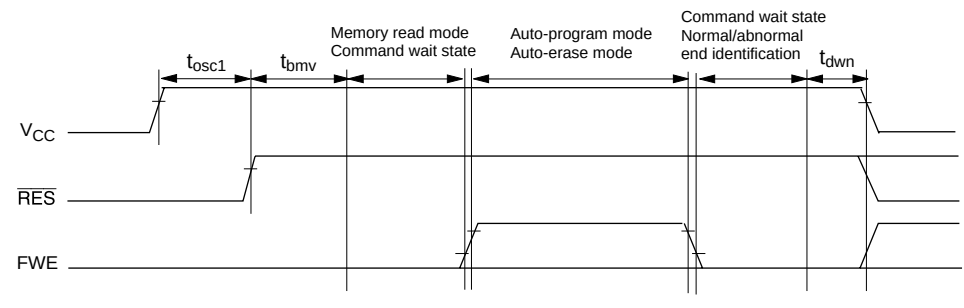
Note: I/O3 and I/O2 are undefined.

Figure 22.38 Timing in Status Read Mode

Table 22.28 Stipulated Transition Times to Command Wait State

Condition: V_{CC} = 3.3 V ± 0.3 V, V_{SS} = 0 V, T_a = 25°C ± 5°C

Code	Symbol	Min	Max	Unit	Note
Standby release (oscillation stabilization time)	t_{osc1}	30		ms	
Programmer mode setup time	t_{bmv}	10		ms	
V_{CC} hold time	t_{dwn}	0		ms	



Note: Set the FWE input pin to low level, except in the auto-program and auto-erase modes.

Figure 22.39 Oscillation Stabilization Time, Programmer Mode Setup Time, and Power-Down Sequence

In the descriptions in the previous section, storable areas for the programming/erasing procedure programs and program data are assumed to be in on-chip RAM. However, the procedure programs and data can be stored in and executed from other areas (e.g. external address space) as long as the following conditions are satisfied.

(1) The on-chip programming/erasing program is downloaded from the address set by FTDAR in on-chip RAM, therefore, this area is not available for use.

(2) The on-chip programming/erasing program will use 128 bytes or more as a stack. Make sure this area is reserved.

(3) Since download by setting the SCO bit to 1 will cause the MATs to be switched, it should be executed in on-chip RAM.

(4) The flash memory is accessible until the start of programming or erasing, that is, until the result of downloading has been judged. When in a mode in which the external address space is not accessible, such as single-chip mode, the required procedure programs, interrupt vector table, interrupt processing routine, and user branch program should be transferred to on-chip RAM before programming/erasing of the flash memory starts.

(5) The flash memory is not accessible during programming/erasing operations. Therefore, the programming/erasing program must be downloaded to on-chip RAM in advance. Areas for executing each procedure program for initiating programming/erasing, the user program at the user branch destination for programming/erasing, the interrupt vector table, and the interrupt processing routine must be located in on-chip memory other than flash memory or the external address space.

(6) After programming/erasing, access to flash memory is inhibited until FKEY is cleared.

A reset state ($\overline{RES}$ = 0) for more than at least 100 µs must be taken when the LSI mode is changed to reset on completion of a programming/erasing operation.

Transitions to the reset state or hardware standby mode during programming/erasing are inhibited. When the reset signal is accidentally input to the LSI, a longer period in the reset state than usual (100 µs) is needed before the reset signal is released.

(7) Switching of the MATs by FMATS is needed for programming/erasing of the user MAT in user boot mode. The program which switches the MATs should be executed from the on-chip RAM. For details, see section 22.8.1, Switching between User MAT and User Boot MAT. Please make sure you know which MAT is selected when switching the MATs.

(8) When the program data storage area indicated by the FMPDR parameter in the programming processing is within the flash memory area, an error will occur. Therefore, temporarily transfer the program data to on-chip RAM to change the address set in FMPDR to an address other than flash memory.

Based on these conditions, tables 22.29 and 22.30 show the areas in which the program data can be stored and executed according to the operation type and mode.

Operation	User Program Mode	User Boot Mode*
Programming	Table 22.30 (1)	Table 22.30 (3)
Erasing	Table 22.30 (2)	Table 22.30 (4)

Note: * Programming/Erasing is possible to user MATs.

	Item	On-Chip RAM	User MAT	External Space (Expanded Mode with MD0 = 0)	User MAT	Embedded Program Storage MAT
Pro-gram-ming proce-dure	Program data storage area	O	X*	O	—	—
	Selecting on-chip program to be downloaded	O	O	O	O	
	Writing H'A5 to key register	O	O	O	O	
	Writing 1 to SCO in FCCS (download)	O	X	X		O
	Key register clearing	O	O	O	O	
	Judging download result	O	O	O	O	
	Download error processing	O	O	O	O	
	Setting initialization parameters	O	O	O	O	
	Initialization	O	X	X	O	
	Judging initialization result	O	O	O	O	
	Initialization error processing	O	O	O	O	
	Interrupt processing routine	O	X	O	O	
	Writing H'5A to key register	O	O	O	O	
	Setting programming parameters	O	X	O	O	
	Programming	O	X	X	O	
	Judging programming result	O	X	O	O	
	Programming error processing	O	X	O	O	
	Key register clearing	O	X	O	O	

Note: * If the data has been transferred to on-chip RAM in advance, this area can be used.

		Storable /Executable Area			Selected MAT	
	Item	On-Chip RAM	User MAT	External Space (Expanded Mode with MD0 = 0)	User MAT	Embedded Program Storage MAT
Erasing procedure	Selecting on-chip program to be downloaded	O	O	O	O	
	Writing H'A5 to key register	O	O	O	O	
	Writing 1 to SCO in FCCS (download)	O	X	X		O
	Key register clearing	O	O	O	O	
	Judging download result	O	O	O	O	
	Download error processing	O	O	O	O	
	Setting initialization parameters	O	O	O	O	
	Initialization	O	X	X	O	
	Judging initialization result	O	O	O	O	
	Initialization error processing	O	O	O	O	
	Interrupt processing routine	O	X	O	O	
	Writing H'5A to key register	O	O	O	O	
	Setting erasure parameters	O	X	O	O	
	Erasure	O	X	X	O	
	Judging erasure result	O	X	O	O	
	Erasing error processing	O	X	O	O	
	Key register clearing	O	X	O	O	

	Item	On-Chip RAM	User MAT	External Space (Expanded Mode with MD0 = 0)	User MAT	User Boot Mat	Embedded Program Storage Area
Pro-gram-ming proce-dure	Program data storage area	O	X*[1]	O	—	—	—
	Selecting on-chip program to be downloaded	O	O	O		O	
	Writing H'A5 to key register	O	O	O		O	
	Writing 1 to SCO in FCCS (download)	O	X	X			O
	Key register clearing	O	O	O		O	
	Judging download result	O	O	O		O	
	Download error processing	O	O	O		O	
	Setting initialization parameters	O	O	O		O	
	Initialization	O	X	X		O	
	Judging initialization result	O	O	O		O	
	Initialization error processing	O	O	O		O	
	Interrupt processing routine	O	X	O		O	
	Switching MATs by FMATS	O	X	X	O		
	Writing H'5A to Key Register	O	X	O	O		

	Item	On-Chip RAM	User MAT	User Space (Expanded Mode with MD0 = 0)	User MAT	User Boot Mat	Embedded Program Storage Area
Programming procedure	Setting programming parameters	O	X	O	O		
	Programming	O	X	X	O		
	Judging programming result	O	X	O	O		
	Programming error processing	O	X*2	O	O		
	Key register clearing	O	X	O	O		
	Switching MATs by FMATS	O	X	X		O	

Notes *1 If the data has been transferred to on-chip RAM in advance, this area can be used.

*2 If the MATs have been switched by FMATS in on-chip RAM, this MAT can be used.

	Item	On-Chip RAM	User MAT	External Space (Expanded Mode with MD0 = 0)	User MAT	User Boot Mat	Embedded Program Storage Area
	Selecting on-chip program to be downloaded	O	O	O		O	
	Writing H'A5 to key register	O	O	O		O	
	Writing 1 to SCO in FCCS (download)	O	X	X			O
	Key register clearing	O	O	O		O	
	Judging download result	O	O	O		O	
	Download error processing	O	O	O		O	
Erasing procedure	Setting initialization parameters	O	O	O		O	
	Initialization	O	X	X		O	
	Judging initialization result	O	O	O		O	
	Initialization error processing	O	O	O		O	
	Interrupt processing routine	O	X	O		O	
	Switching MATs by FMATS	O	X	X		O	
	Writing H'5A to key register	O	X	O	O		
	Setting erasure parameters	O	X	O	O		

	Item	On-Chip RAM	User MAT	Space (Expanded Mode with MD0 = 0)	User MAT	User Boot Mat	Embedded Program Storage Area
Erasing procedure	Erasure	O	X	X	O		
	Judging erasure result	O	X	O	O		
	Erasing error processing	O	X*	O	O		
	Key register clearing	O	X	O	O		
	Switching MATs by FMATS	O	X	X		O	

Note: * If the MATs have been switched by FMATS in on-chip RAM, this MAT can be used.

23.1 Overview

The SH7055SF has 32 kbytes of on-chip RAM. The on-chip RAM is linked to the CPU, direct memory access controller (DMAC), and advanced user debugger (AUD) with a 32-bit data bus (figure 23.1).

The CPU, DMAC, and AUD can access data in the on-chip RAM in 8, 16, or 32 bit widths. On-chip RAM data can always be accessed in one state, making the RAM ideal for use as a program area, stack area, or data area, which require high-speed access. The contents of the on-chip RAM are held in both the sleep and software standby modes. When the RAME bit (see below) is cleared to 0, the on-chip RAM contents are also held in hardware standby mode.

The on-chip RAM is allocated to addresses H'FFFF6000 to H'FFFFDFFF.

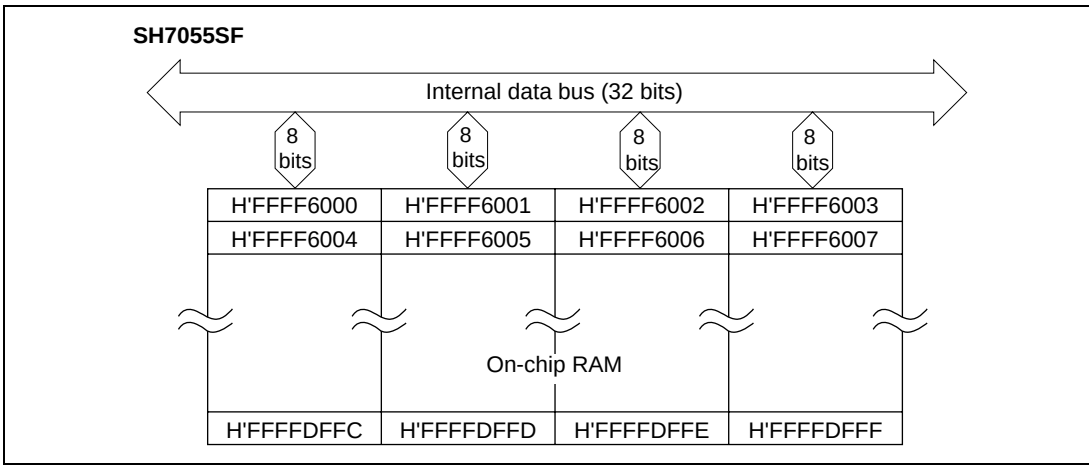


Figure 23.1 Block Diagram of RAM

The on-chip RAM is controlled by means of the system control register (SYSCR).

When the RAME bit in SYSCR is set to 1, the on-chip RAM is enabled. Accesses to addresses H'FFFF6000–H'FFFFDFFF are then directed to the on-chip RAM.

When the RAME bit in SYSCR is cleared to 0, the on-chip RAM is not accessed. A read will return an undefined value, and a write is invalid. If a transition is made to hardware standby mode after the RAME bit in SYSCR is cleared to 0, the contents of the on-chip RAM are held.

For details of SYSCR, see 24.2.2, System Control Register (SYSCR), in section 24, Power-Down State.

24.1 Overview

Three modes are provided as power-save modes, namely, the hardware standby, software standby and sleep modes. Also, a module stop function is available to stop some modules. These standby modes can be selected depending on applications to reduce the power consumption of the SH7055SF.

24.1.1 Power-Down States

The power-down state is effected by the following modes:

1. Hardware standby mode

   A transition to hardware standby mode is made according to the input level of the $\overline{\text{RES}}$ and $\overline{\text{HSTBY}}$ pins.

   In hardware standby mode, all SH7055SF functions are halted.

   This state is exited by means of a power-on reset.
2. Software standby mode

   A transition to software standby mode is made by means of software (a CPU instruction).

   In software standby mode, all SH7055SF functions are halted.

   This state is exited by means of a power-on reset or an NMI interrupt.
3. Sleep mode

   A transition to sleep mode is made by means of a CPU instruction.

   In software standby mode, basically only the CPU is halted, and all on-chip peripheral modules operate.

   This state is exited by means of a power-on reset, a manual reset, interrupt, or DMA address error.
4. Module standby mode

   Operation of the on-chip peripheral modules* which can be placed in a standby mode can be stopped by stopping the clock supply. Clock supply to the individual modules can be controlled by setting bits in the module standby control register (MSTCR).

Note: * AUD, H-UDI, FPU, and UBC

canceling each mode.

Table 24.1 Power-Down State Conditions

Mode	Entering Procedure	State						Canceling Procedure
		Clock	CPU	CPU Registers	On-Chip Peripheral Modules	RAM	Pins	
Hardware standby	Low-level input at $\overline{\text{HSTBY}}$ pin	Halted	Halted	Halted	Undefined	Held*2	Initialized	High-level input at $\overline{\text{HSTBY}}$ pin, executing power-on reset
Software standby	Execute SLEEP instruction with SSBY bit set to 1 in SBYCR	Halted	Halted	Held	Halted*1	Held	Held or high impedance*3	• NMI interrupt • Power-on reset
Sleep	Execute SLEEP instruction with SSBY bit cleared to 0 in SBYCR	Runs	Halted	Held	Run	Held	Held	• Interrupt • DMA address error • Power-on reset • Manual reset

Notes: SBYCR: Standby control register

SSBY: Software standby bit

*1 Some bits within on-chip peripheral module registers are initialized in software standby mode, and some are not. See table A.2, Register States in Reset and Power-Down States. Also refer to the register descriptions for each peripheral module.

*2 Clear the RAME bit of the SYSCR to 0 in advance when changing the state from the program execution state to the hardware standby state.

*3 The state of the I/O ports in standby mode is set by the port high impedance bit (HIZ) in SBYCR. See section 24.2.1, Standby Control Register (SBYCR). For details of other pin states, refer to appendix B, Pin States.

Pins related to power-down modes are shown in table 24.2.

Table 24.2 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Hardware standby input pin	$\overline{\text{HSTBY}}$	Input	Input level determines transition to hardware standby mode
Power-on reset input pin	$\overline{\text{RES}}$	Input	Power-on reset signal input pin

24.1.3 Related Registers

Table 24.3 shows the registers used for power-down state control.

Table 24.3 Related Registers

Name	Abbreviation	R/W	Initial Value	Address		Access Size
				Write	Read	
Standby control register	SBYCR*1	R/W	H'1F	H'FFFFEC14		8
System control register	SYSCR*1	R/W	H'01*4	H'FFFFF708		8
Module standby control register	MSTCR*1	R/W	H'01	H'FFFFF70A*2	H'FFFFF70B*3	8, 16

Notes: *1 SBYCR is accessed in three cycles, SYSCR and MSTCR in four or five cycles.
*2 Write data in word units. Data cannot be written in byte or longword units.
*3 Read data in byte units. Values cannot be read correctly if data is read in word or longword units.
*4 The initial value of bit 7 in SYSCR is not defined.

24.2.1 Standby Control Register (SBYCR)

The standby control register (SBYCR) is an 8-bit readable/writable register that sets the transition to standby mode, and the port state in standby mode. SBYCR is initialized to H'1F by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	SSBY	HIZ	—	—	—	—	—	—
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R	R	R	R	R	R

- Bit 7—Software Standby (SSBY): Specifies transition to software standby mode. The SSBY bit cannot be set to 1 while the watchdog timer is running (when the timer enable bit (TME) in the WDT timer control/status register (TCSR) is set to 1). To enter software standby mode, always halt the WDT by clearing the TME bit to 0, then set the SSBY bit.

Bit 7: SSBY	Description	
0	Executing SLEEP instruction puts the SH7055SF into sleep mode	(Initial value)
1	Executing SLEEP instruction puts the SH7055SF into standby mode	

- Bit 6—Port High Impedance (HIZ): In software standby mode, this bit selects whether to set I/O port pins to high impedance or hold the pin state. The HIZ bit cannot be set to 1 when the TME bit of the WDT timer control/status register (TCSR) is set to 1. When making the I/O port pin state high impedance, always clear the TME bit to 0 before setting the HIZ bit.

Bit 6: HIZ	Description	
0	Pin states held in software standby mode	(Initial value)
1	Pins go to high impedance in software standby mode	

- Bit 5—Reserved: This bit always reads 0. The write value should always be 0.
- Bits 4 to 0—Reserved: These bits always read 1. The write value should always be 1.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	AUDSRST	RAME
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/W	R/W

The system control register (SYSCR) is an 8-bit readable/writable register that performs AUD software reset control and enables or disables access to the on-chip RAM.

SYSCR is initialized to H'01 by a power-on reset.

- Bit 7—Reserved: The read value is not defined. The write value should always be 0.
- Bits 6 to 2—Reserved: These bits always read 0. The write value should always be 0.
- Bit1— AUD Software Reset (AUDSRST): This bit controls AUD reset using software. Setting AUDSRST bit to 1 places, the AUD module in the power-on reset state.

Bit 1: AUDSRST	**Description**
0	AUD reset state cleared
1	AUD reset state entered (Initial value)

- Bit 0—RAME Enable (RAME): Selects enabling or disabling of the on-chip RAM. When RAME is set to 1, on-chip RAM is enabled. When RAME is cleared to 0, on-chip RAM cannot be accessed. In this case, a read or instruction fetch from on-chip RAM will return an undefined value, and a write to on-chip RAM will be ignored. The initial value of RAME is 1.

  When on-chip RAM is disabled by clearing RAME to 0, do not place an instruction that attempts to access on-chip RAM immediately after the SYSCR write instruction, as normal access cannot be guaranteed in this case.

  When on-chip RAM is enabled by setting RAME to 1, place an SYSCR read instruction immediately after the SYSCR write instruction. Normal access cannot be guaranteed if an on-chip RAM access instruction is placed immediately after the SYSCR write instruction.

Bit 0: RAME	**Description**
0	On-chip RAM disabled
1	On-chip RAM enabled (Initial value)

Bit:	—	—	—	—	MSTOP3	MSTOP2	MSTOP1	MSTOP0
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

The module standby control register (MSTCR) is an 8-bit readable/writable register that controls the standby state of the AUD, H-UDI, FPU, and UBC on-chip modules.

MSTCR is initialized to H'01 by a power-on reset.

Note: The method of writing to MSTCR is different from that of ordinary registers to prevent inadvertent rewriting. See section 24.2.4, Notes on Register Access, for more information.

- Bits 7 to 4—Reserved: These bits always read 0. The write value should always be 0.
- Bit 3—Module Stop 3 (MSTOP3): Specifies halting of the clock supply to the AUD on-chip peripheral module. Setting the MSTOP3 bit to 1 stops the clock supply to the AUD. To cancel halting of the clock supply to the AUD, first set the AUD software reset bit (AUDSRST) in the system control register (SYSCR) to the AUD reset state value. Use of the AUD will then be enabled by clearing the AUD reset.

Bit 3: MSTOP3	Description	
0	AUD operates	(Initial value)
1	Clock supply to AUD stopped	

- Bit 2—Module Stop 2 (MSTOP2): Specifies halting of the clock supply to the H-UDI on-chip peripheral module. Setting the MSTOP2 bit to 1 stops the clock supply to the H-UDI.

Bit 2: MSTOP2	Description	
0	H-UDI operates	
1	Clock supply to H-UDI stopped	(Initial value)

The MSTOP1 bit cannot be cleared by writing 0 after it has been set to 1. In other words, once the MSTOP1 bit has been set to 1 and the clock supply to the FPU has been stopped, the clock supply to the FPU cannot be resumed by clearing the MSTOP1 bit to 0.

An SH7055SF power-on reset is necessary to restart the FPU clock supply after it has been stopped.

Bit 1: MSTOP1	Description	
0	FPU operates	(Initial value)
1	Clock supply to FPU stopped	

- Bit 0—Module Stop 0 (MSTOP0): Specifies halting of the clock supply stop to the UBC on-chip peripheral module.

  Clearing the MSTOP0 bit to 0 starts the clock supply to the UBC.

  Stopping clock supply to the UBC will reset the internal state of the UBC including its registers.

Bit 0: MSTOP0	Description	
0	UBC operates	(Initial value)
1	Clock supply to UBC stopped	

24.2.4 Notes on Register Access

The method of writing to the module standby control register (MSTCR) is different from that of ordinary registers to prevent inadvertent rewriting.

Be certain to use a word transfer instruction when writing data to MSTCR. Data cannot be written by a byte transfer instruction. As shown in figure 24.1, set the upper byte to H'3C and transfer data using the lower byte as write data.

Data can be read by the same method as for ordinary registers.

MSTCR is allocated to address H'FFFFF70A. Always use a byte transfer instruction to read data.

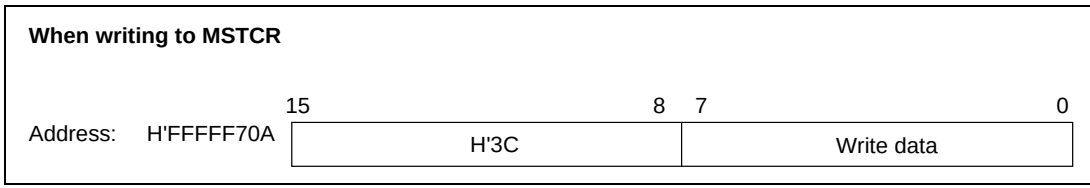


Figure 24.1 Writing to MSTCR

24.3.1 Transition to Hardware Standby Mode

The chip enters hardware standby mode when the $\overline{\text{HSTBY}}$ and $\overline{\text{RES}}$ pins go low. Set the pins following to mode setup pin shown in section 4, Operating modes. Operation with other pin set up are not guaranteed.

Hardware standby mode reduces power consumption drastically by halting all SH7055SF functions. As the transition to hardware standby mode is made by means of external pin input, the transition is made asynchronously, regardless of the current state of the SH7055SF, and therefore the chip state prior to the transition is not preserved. However, on-chip RAM data is retained as long as the specified voltage is supplied. To retain on-chip RAM data, clear the RAM enable bit (RAME) to 0 in the system control register (SYSCR) before driving the $\overline{\text{HSTBY}}$ pin low. See appendix B, Pin States, for the pin states in hardware standby mode.

24.3.2 Canceling Hardware Standby Mode

Hardware standby mode is canceled by means of the $\overline{\text{HSTBY}}$ pin and $\overline{\text{RES}}$ pin. When $\overline{\text{HSTBY}}$ is driven high while $\overline{\text{RES}}$ is low, the clock oscillator starts running. The $\overline{\text{RES}}$ pin should be held low long enough for clock oscillation to stabilize. When $\overline{\text{RES}}$ is driven high, power-on reset exception processing is started and a transition is made to the program execution state.

24.3.3 Hardware Standby Mode Timing

Figure 24.2 shows sample pin timings for hardware standby mode. A transition to hardware standby mode is made by driving the $\overline{\text{HSTBY}}$ pin low after driving the $\overline{\text{RES}}$ pin low. Hardware standby mode is canceled by driving $\overline{\text{HSTBY}}$ high, waiting for clock oscillation to stabilize, then switching $\overline{\text{RES}}$ from low to high.

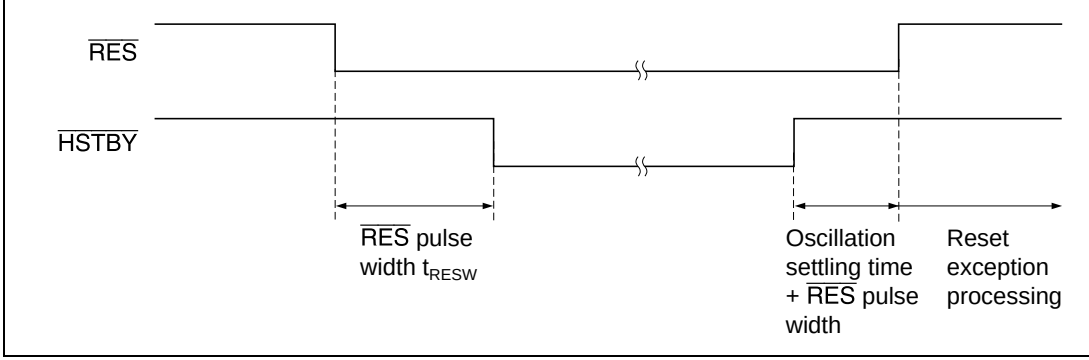


Figure 24.2 Hardware Standby Mode Timing

24.4.1 Transition to Software Standby Mode

To enter software standby mode, set the software standby bit (SSBY) to 1 in SBYCR, then execute the SLEEP instruction. The SH7055SF switches from the program execution state to software standby mode. In software standby mode, power consumption is greatly reduced by halting not only the CPU, but the clock and on-chip peripheral modules as well. CPU register contents and on-chip RAM data are held as long as the prescribed voltages are applied (when the RAME bit in SYSCR is 0). The register contents of some on-chip peripheral modules are initialized, but some are not. For details on the register states, refer to appendix A.2, Register States in Reset and Power-Down States. The I/O port state can be selected as held or high impedance by the port high impedance bit (HIZ) in SBYCR. For other pin states, refer to appendix B, Pin States.

24.4.2 Canceling Software Standby Mode

Software standby mode is canceled by an NMI interrupt or a power-on reset.

Cancellation by NMI: Clock oscillation starts when a rising edge or falling edge (selected by the NMI edge select bit (NMIE) in the interrupt control register (ICR) of the INTC) is detected in the NMI signal. This clock is supplied only to the oscillation settling counter which counts the oscillation stablizing time.

The oscillation settling counter overflows when it counts 2^{16}=65536 with the input clock frequency. Since the frequency of this counting clock is unstable until the PLL multiply curcuit is locked in the absolute time is not fixed, and the CK pin signal output is in the high level for the meantime.

Counting the oscillation settling time by the oscillation settling counter is used to indicate that the clock has stabilized, so the clock is supplied to the entire chip, software standby mode is canceled, and NMI exception processing begins.

When canceling standby mode with an NMI pin set for falling edge, be sure that the NMI pin level upon entering software standby (when the clock is halted) is high, and that the NMI pin level upon returning from software standby (when the clock starts after oscillation stabilization) is low. When canceling software standby mode with an NMI pin set for rising edge, be sure that the NMI pin level upon entering software standby (when the clock is halted) is low, and that the NMI pin level upon returning from software standby (when the clock starts after oscillation stabilization) is high.

Cancellation by Power-On Reset: A power-on reset of the SH7055SF caused by driving the $\overline{\text{RES}}$ pin low cancels software standby mode.

This example describes a transition to software standby mode on the falling edge of the NMI signal, and cancellation on the rising edge of the NMI signal. The timing is shown in figure 24.3.

When the NMI pin is changed from high to low level while the NMI edge select bit (NMIE) in ICR is set to 0 (falling edge detection), the NMI interrupt is accepted. When the NMIE bit is set to 1 (rising edge detection) by the NMI exception service routine, the software standby bit (SSBY) in SBYCR is set to 1, and a SLEEP instruction is executed, software standby mode is entered. Thereafter, software standby mode is canceled when the NMI pin is changed from low to high level.

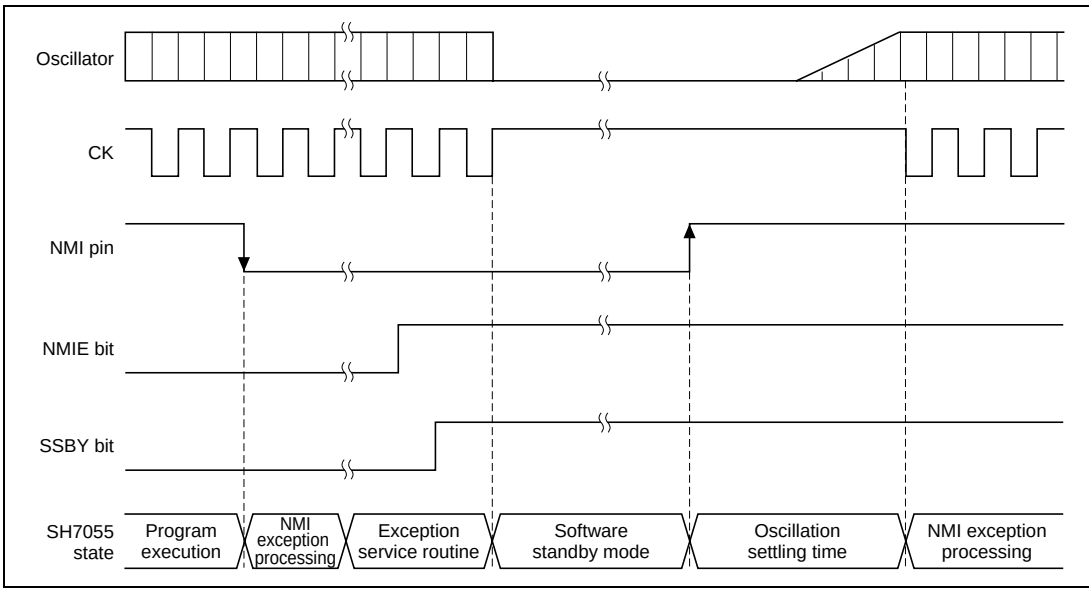


Figure 24.3 Software Standby Mode NMI Timing (Application Example)

24.5.1 Transition to Sleep Mode

Executing the SLEEP instruction after the software standby bit (SSBY) in SBYCR has been cleared to 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to run during sleep mode.

24.5.2 Canceling Sleep Mode

Cancellation by Interrupt: When an interrupt occurs, sleep mode is canceled and interrupt exception processing is executed. The sleep mode is not canceled if the interrupt cannot be accepted because its priority level is equal to or less than the mask level set in the CPU's status register (SR) or if an interrupt by an on-chip peripheral module is disabled at the peripheral module.

Cancellation by DMA Address Error: If a DMA address error occurs, sleep mode is canceled and DMA address error exception processing is executed.

Cancellation by Manual Reset: When an internal manual reset is triggered by the WDT and the CPU acquires the bus during the internal manual reset period, the state of the SH7055SF changes to the manual reset state and sleep mode will be released.

Cancellation by Power-On Reset: A power-on reset of the SH7055SF resulting from driving the $\overline{\text{RES}}$ pin low, or caused by the WDT, cancels sleep mode.

25.1 Reliability

A failure rate curve represents an index of the reliability of a semiconductor device. The failure rate curve traces a bathtub shape over the course of time, as is shown in figure 25.1. The curve is divided into three periods according to the type of failure phenomena: an initial failure period, a random failure period (functional lifetime), and a wear-out failure period. Initial failures, which occur during the initial failure period, are caused by contamination with foreign matter and localized chemical pollution; these can be eliminated by screening. Wear-out failures in the final period are caused by the deterioration of materials that make up semiconductor devices during long periods of usage. Random failures, which occur during the random failure period, are thought to occur in cases where a device with a minor failure is not removed by screening, and so is shipped, and then fails during the customer's production process or in the field, and in cases where a failure which should normally not have occurred until the wear-out period occurs earlier because of variations in production. Therefore, the reliability of semiconductor device is secured by appropriate screening to reduce the presence of initial failures and high reliability design to prevent the occurrence of wear-out failures. The reliability of a product is confirmed by producing a large quantity of prototypes for checking of the initial failure rate and executing accelerated life testing to identify the wear-out failure time in a realistic environment.

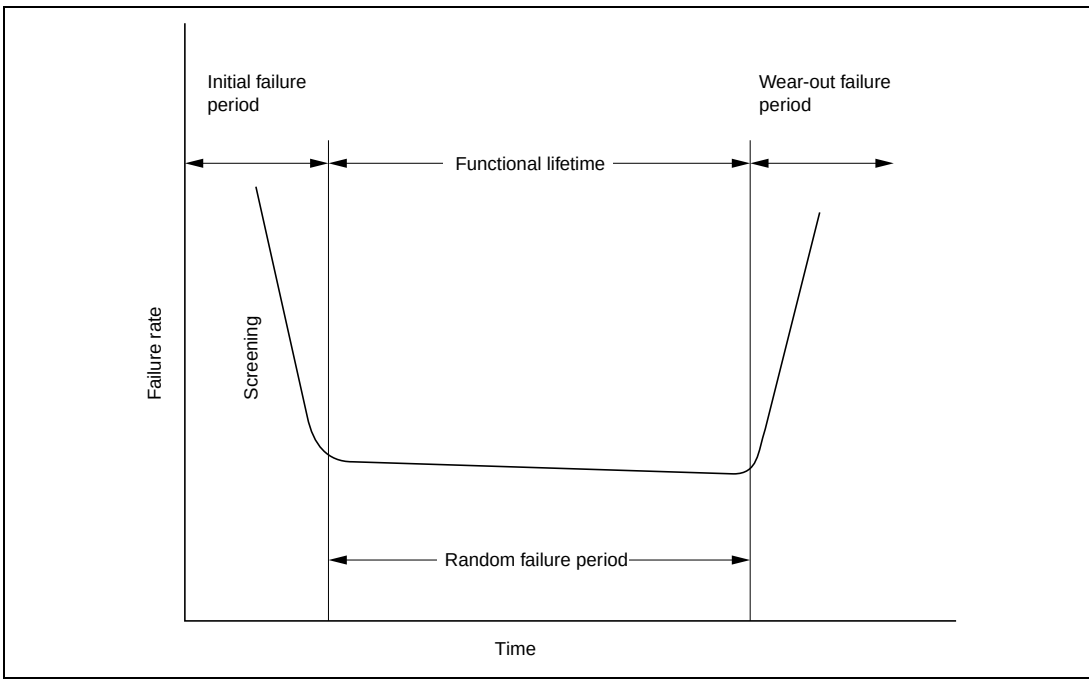


Figure 25.1 Failure Rate Curve (Bathtub Curve)

sectors. The representative failure phenomena of semiconductor devices, such as the dielectric breakdown of oxide films and electromigration in wiring, constitute wear-out failures. The stress factors in such failures are the voltage, current, and temperature applied to devices while they are in use. Since the temperature range for the guaranteed operation of products for use in automobiles is conventionally –40°C to 85°C, their reliability in terms of the above failure phenomena has to be confirmed by accelerated life testing at all temperatures in this range. Operation at temperatures in excess of 85°C leads to failure within a short time, since high temperatures induce failures in semiconductor devices. Figure 25.2 shows the temperature dependence of semiconductor device lifetimes. The type of failure in this figure is a wear-out failure, i.e. the dielectric breakdown of oxide film. According to figure 25.2, the life at 125°C is 1/10 of life at 85°C, and operation at the higher temperature leads to a correspondingly higher probability of a failure in the field. Therefore, the reliability of operation at a temperature in excess of 85°C is checked on the assumption that the period of operation at the upper-limit temperature of the range for guaranteed operation is 3000 hours.

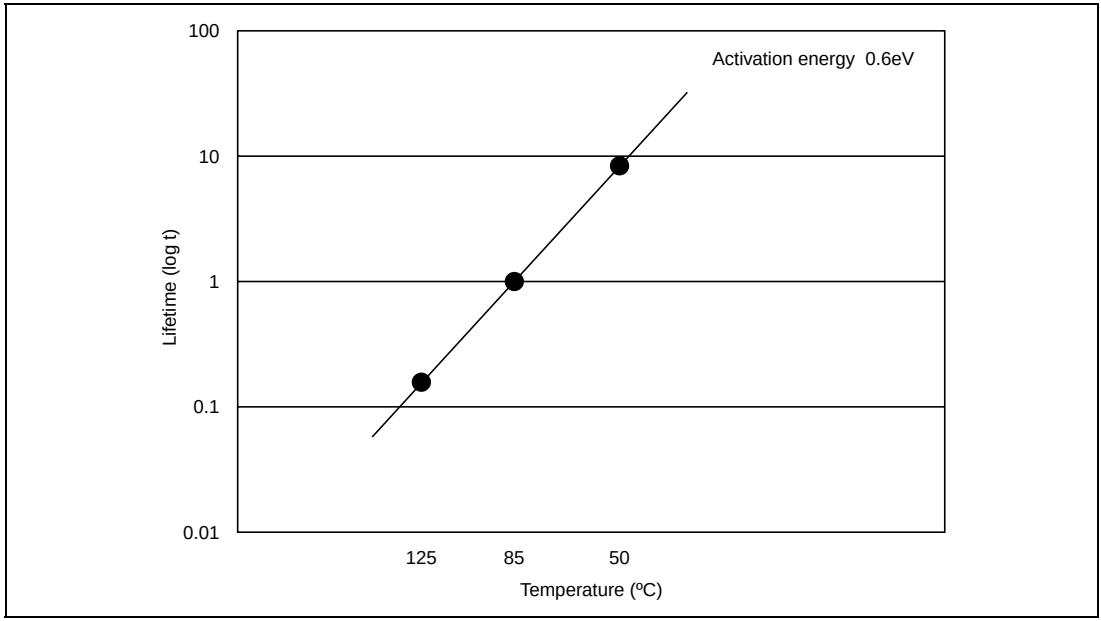


Figure 25.2 Temperature Reliability of Dielectric Breakdown of Oxide Film

26.1 Absolute Maximum Ratings

Table 26.1 shows the absolute maximum ratings.

Table 26.1 Absolute Maximum Ratings

Item		Symbol	Rating	Unit	Remarks
Power supply voltage*	V_{CC} and $PLLV_{CC}$ pins	V_{CC}	–0.3 to +4.3	V	The PLLCAP, EXTAL, XTAL, CK, and H-UDI pins are concerned. (V_{CC} and $PLLV_{CC}$ are the same voltage)
	$PV_{CC}1$ and $PV_{CC}2$ pins	PV_{CC}	–0.3 to +6.5	V	Except for the PLLCAP, EXTAL, XTAL, CK, and H-UDI pins and the analog input pin
Input voltage	EXTAL and H-UDI pins	Vin	–0.3 to V_{CC} + 0.3	V	
	All pins other than analog input, EXTAL, and H-UDI pins	Vin	–0.3 to PV_{CC} + 0.3	V	Refer to table 26.2, Correspondence between Power Supply Names and Pins
Analog supply voltage		AV_{CC}	–0.3 to +7.0	V	
Analog reference voltage		AVref	–0.3 to AV_{CC} + 0.3	V	
Analog input voltage		V_{AN}	–0.3 to AV_{CC} + 0.3	V	
Operating temperature ** (except writing or erasing on-chip flash memory)		Topr	–40 to +125	°C	
Operating temperature (writing or erasing on-chip flash memory)		TWEopr	–40 to +85	°C	
Storage temperature		Tstg	–55 to +125	°C	

[Operating precautions]

Operating the LSI in excess of the absolute maximum ratings may result in permanent damage. The two power supply voltages of PV_{CC} of 5V and V_{CC} of 3V may be used simultaneously with the LSI. Be sure to use the LSI in compliance with the connection of power pins, combination conditions of applicable power supply voltages, voltage applicable to each pin, and conditions of output voltage, as specified in the manual. Connecting a non-specified power supply or using the LSI at an incorrect voltage may result in permanent damage of the LSI or the system that contains the LSI.

When this LSI is used at temperatures in excess of the range from −40 to +85 °C, it can be operated within following accumulated hours.

Temperature Range for Operation	Accumulated Time
85 to 105 °C	3000 hours

Table 26.2 shows the correspondence between power supply names and pins.

Table 26.4 shows DC characteristics.

Table 26.2 Correspondence between Power Supply Names and Pins

Pin No.	Power Supply Pin		User Pin				Output Circuit	Input Voltage	
	Power Supply Name	Dedicated Pin	Function 1	Function 2	Function 3	Function 4	Power Supply Name	Upper Limit (V)	Notes
1			PD8	PULS0			$PV_{cc}2$	$PV_{cc}2+0.3$	
2			PD9	PULS1			$PV_{cc}2$	$PV_{cc}2+0.3$	
3			PD10	PULS2			$PV_{cc}2$	$PV_{cc}2+0.3$	
4			PD11	PULS3			$PV_{cc}2$	$PV_{cc}2+0.3$	
5			PD12	PULS4			$PV_{cc}2$	$PV_{cc}2+0.3$	
6			PD13	PULS6	HTxD0	HTxD1	$PV_{cc}2$	$PV_{cc}2+0.3$	
7			PE0	A0			$PV_{cc}1$	$PV_{cc}1+0.3$	
8			PE1	A1			$PV_{cc}1$	$PV_{cc}1+0.3$	
9			PE2	A2			$PV_{cc}1$	$PV_{cc}1+0.3$	
10			PE3	A3			$PV_{cc}1$	$PV_{cc}1+0.3$	
11	V_{cc}								
12			PE4	A4			$PV_{cc}1$	$PV_{cc}1+0.3$	
13	V_{ss}								
14			PE5	A5			$PV_{cc}1$	$PV_{cc}1+0.3$	
15			PE6	A6			$PV_{cc}1$	$PV_{cc}1+0.3$	
16			PE7	A7			$PV_{cc}1$	$PV_{cc}1+0.3$	
17			PE8	A8			$PV_{cc}1$	$PV_{cc}1+0.3$	
18			PE9	A9			$PV_{cc}1$	$PV_{cc}1+0.3$	
19			PE10	A10			$PV_{cc}1$	$PV_{cc}1+0.3$	
20	$PV_{cc}1$								
21			PE11	A11			$PV_{cc}1$	$PV_{cc}1+0.3$	
22	V_{ss}								
23			PE12	A12			$PV_{cc}1$	$PV_{cc}1+0.3$	

Pin No.	Supply Pin Power Supply Name	Dedicated Pin	User Pin Function 1	Function 2	Function 3	Function 4	Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
24			PE13	A13			$PV_{cc}1$	$PV_{cc}1+0.3$	
25			PE14	A14			$PV_{cc}1$	$PV_{cc}1+0.3$	
26			PE15	A15			$PV_{cc}1$	$PV_{cc}1+0.3$	
27			PF0	A16			$PV_{cc}1$	$PV_{cc}1+0.3$	
28			PF1	A17			$PV_{cc}1$	$PV_{cc}1+0.3$	
29			PF2	A18			$PV_{cc}1$	$PV_{cc}1+0.3$	
30	V_{CL}								
31			PF3	A19			$PV_{cc}1$	$PV_{cc}1+0.3$	
32	V_{SS}								
33			PF4	A20			$PV_{cc}1$	$PV_{cc}1+0.3$	
34			PF5	A21	$\overline{POD}$		$PV_{cc}1$	$PV_{cc}1+0.3$	
35			PF6	$\overline{WRL}$			$PV_{cc}1$	$PV_{cc}1+0.3$	
36			PF7	$\overline{WRH}$			$PV_{cc}1$	$PV_{cc}1+0.3$	
37			PF8	$\overline{WAIT}$			$PV_{cc}1$	$PV_{cc}1+0.3$	
38			PF9	$\overline{RD}$			$PV_{cc}1$	$PV_{cc}1+0.3$	
39	$PV_{cc}1$								
40			PF10	$\overline{CS0}$			$PV_{cc}1$	$PV_{cc}1+0.3$	
41	V_{SS}								
42			PF11	$\overline{CS1}$			$PV_{cc}1$	$PV_{cc}1+0.3$	
43			PF12	$\overline{CS2}$			$PV_{cc}1$	$PV_{cc}1+0.3$	
44			PF13	$\overline{CS3}$			$PV_{cc}1$	$PV_{cc}1+0.3$	
45			PF14	$\overline{BACK}$			$PV_{cc}1$	$PV_{cc}1+0.3$	
46			PF15	$\overline{BREQ}$			$PV_{cc}1$	$PV_{cc}1+0.3$	
47	V_{SS}								
48			CK				V_{cc}		
49	V_{cc}								
50		MD2						5.5+0.3	
51		EXTAL						$V_{cc}+0.3$	
52	V_{cc}								
53		XTAL					V_{cc}		

Pin No.	Supply Pin Power Supply Name	Dedicated Pin	User Pin Function 1	Function 2	Function 3	Function 4	Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
54	V_{ss}								
55		MD1						5.5+0.3	
56		FWE						5.5+0.3	
57		$\overline{\text{HSTBY}}$						5.5+0.3	
58		$\overline{\text{RES}}$						5.5+0.3	
59		MD0						5.5+0.3	
60	$PLLV_{cc}$								
61		PLLCAP							
62	$PLLV_{ss}$								
63			PH0	D0			$PV_{cc}1$	$PV_{cc}1$+0.3	
64			PH1	D1			$PV_{cc}1$	$PV_{cc}1$+0.3	
65			PH2	D2			$PV_{cc}1$	$PV_{cc}1$+0.3	
66			PH3	D3			$PV_{cc}1$	$PV_{cc}1$+0.3	
67			PH4	D4			$PV_{cc}1$	$PV_{cc}1$+0.3	
68			PH5	D5			$PV_{cc}1$	$PV_{cc}1$+0.3	
69			PH6	D6			$PV_{cc}1$	$PV_{cc}1$+0.3	
70	$PV_{cc}1$								
71			PH7	D7			$PV_{cc}1$	$PV_{cc}1$+0.3	
72	V_{ss}								
73			PH8	D8			$PV_{cc}1$	$PV_{cc}1$+0.3	
74			PH9	D9			$PV_{cc}1$	$PV_{cc}1$+0.3	
75	V_{cc}								
76			PH10	D10			$PV_{cc}1$	$PV_{cc}1$+0.3	
77	V_{ss}								
78			PH11	D11			$PV_{cc}1$	$PV_{cc}1$+0.3	
79			PH12	D12			$PV_{cc}1$	$PV_{cc}1$+0.3	
80			PH13	D13			$PV_{cc}1$	$PV_{cc}1$+0.3	
81			PH14	D14			$PV_{cc}1$	$PV_{cc}1$+0.3	
82			PH15	D15			$PV_{cc}1$	$PV_{cc}1$+0.3	
83	$PV_{cc}1$								

Pin No.	Supply Pin Power Supply Name	Dedicated Pin	User Pin Function 1	Function 2	Function 3	Function 4	Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
84		NMI						5.5+0.3	
85	V_{SS}								
86			AN0					AV_{CC}+0.3	
87			AN1					AV_{CC}+0.3	
88			AN2					AV_{CC}+0.3	
89			AN3					AV_{CC}+0.3	
90			AN4					AV_{CC}+0.3	
91			AN5					AV_{CC}+0.3	
92			AN6					AV_{CC}+0.3	
93			AN7					AV_{CC}+0.3	
94			AN8					AV_{CC}+0.3	
95			AN9					AV_{CC}+0.3	
96			AN10					AV_{CC}+0.3	
97			AN11					AV_{CC}+0.3	
98			AN12					AV_{CC}+0.3	
99	AV_{SS}								
100		AVref							
101	AV_{CC}								
102			AN13					AV_{CC}+0.3	
103			AN14					AV_{CC}+0.3	
104			AN15					AV_{CC}+0.3	
105			AN16					AV_{CC}+0.3	
106			AN17					AV_{CC}+0.3	
107			AN18					AV_{CC}+0.3	
108			AN19					AV_{CC}+0.3	
109			AN20					AV_{CC}+0.3	
110			AN21					AV_{CC}+0.3	
111			AN22					AV_{CC}+0.3	
112			AN23					AV_{CC}+0.3	
113			AN24					AV_{CC}+0.3	

Pin No.	Supply Pin Power Supply Name	Dedicated Pin	User Pin Function 1	Function 2	Function 3	Function 4	Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
114			AN25					AV_{cc}+0.3	
115			AN26					AV_{cc}+0.3	
116			AN27					AV_{cc}+0.3	
117			AN28					AV_{cc}+0.3	
118			AN29					AV_{cc}+0.3	
119	AV_{cc}								
120		AVref							
121	AV_{ss}								
122			AN30					AV_{cc}+0.3	
123			AN31					AV_{cc}+0.3	
124		$\overline{\text{WDTOVF}}$					$PV_{cc}2$		
125			PA0	TI0A			$PV_{cc}2$	$PV_{cc}2$+0.3	Schmitt-trigger input pin
126	V_{ss}								
127			PA1	TI0B			$PV_{cc}2$	$PV_{cc}2$+0.3	Schmitt-trigger input pin
128	$PV_{cc}2$								
129			PA2	TI0C			$PV_{cc}2$	$PV_{cc}2$+0.3	Schmitt-trigger input pin
130			PA3	TI0D			$PV_{cc}2$	$PV_{cc}2$+0.3	
131			PA4	TIO3A			$PV_{cc}2$	$PV_{cc}2$+0.3	
132			PA5	TIO3B			$PV_{cc}2$	$PV_{cc}2$+0.3	
133			PA6	TIO3C			$PV_{cc}2$	$PV_{cc}2$+0.3	
134			PA7	TIO3D			$PV_{cc}2$	$PV_{cc}2$+0.3	
135			PA8	TIO4A			$PV_{cc}2$	$PV_{cc}2$+0.3	
136			PA9	TIO4B			$PV_{cc}2$	$PV_{cc}2$+0.3	
137			PA10	TIO4C			$PV_{cc}2$	$PV_{cc}2$+0.3	
138			PA11	TIO4D			$PV_{cc}2$	$PV_{cc}2$+0.3	
139	V_{cc}								

Pin No.	Supply Pin Power Supply Name	Dedicated Pin	User Pin Function 1	Function 2	Function 3	Function 4	Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
140			PA12	TIO5A			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
141	V_{ss}								
142			PA13	TIO5B			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
143			PA14	TxD0			$PV_{cc}2$	$PV_{cc}2+0.3$	
144			PA15	RxD0			$PV_{cc}2$	$PV_{cc}2+0.3$	
145			PB0	TO6A			$PV_{cc}2$	$PV_{cc}2+0.3$	
146			PB1	TO6B			$PV_{cc}2$	$PV_{cc}2+0.3$	
147			PB2	TO6C			$PV_{cc}2$	$PV_{cc}2+0.3$	
148	$PV_{cc}2$								
149			PB3	TO6D			$PV_{cc}2$	$PV_{cc}2+0.3$	
150	V_{ss}								
151			PB4	TO7A	TO8A		$PV_{cc}2$	$PV_{cc}2+0.3$	
152			PB5	TO7B	TO8B		$PV_{cc}2$	$PV_{cc}2+0.3$	
153			PB6	TO7C	TO8C		$PV_{cc}2$	$PV_{cc}2+0.3$	
154			PB7	TO7D	TO8D		$PV_{cc}2$	$PV_{cc}2+0.3$	
155			PB8	TxD3	TO8E		$PV_{cc}2$	$PV_{cc}2+0.3$	
156			PB9	RxD3	TO8F		$PV_{cc}2$	$PV_{cc}2+0.3$	
157			PB10	TxD4	HTxD0	TO8G	$PV_{cc}2$	$PV_{cc}2+0.3$	
158			PB11	RxD4	HRxD0	TO8H	$PV_{cc}2$	$PV_{cc}2+0.3$	
159			PB12	TCLKA	$\overline{\text{UBCTRG}}$		$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
160			PB13	SCK0			$PV_{cc}2$	$PV_{cc}2+0.3$	
161	V_{CL}								
162			PB14	SCK1	TCLKB	TI10	$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
163	V_{ss}								

Pin No.	Supply Pin Power Supply Name	Dedicated Pin	User Pin Function 1	Function 2	Function 3	Function 4	Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
164			PB15	PULS5	SCK2		$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
165			PC0	TxD1			$PV_{cc}2$	$PV_{cc}2+0.3$	
166			PC1	RxD1			$PV_{cc}2$	$PV_{cc}2+0.3$	
167			PC2	TxD2			$PV_{cc}2$	$PV_{cc}2+0.3$	
168			PC3	RxD2			$PV_{cc}2$	$PV_{cc}2+0.3$	
169			PC4	$\overline{IRQ0}$			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
170			PG0	PULS7	HRxD0	HRxD1	$PV_{cc}2$	$PV_{cc}2+0.3$	
171			PG1	$\overline{IRQ1}$			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
172	$PV_{cc}2$								
173			PG2	$\overline{IRQ2}$	ADEND		$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
174	V_{ss}								
175			PG3	$\overline{IRQ3}$	ADTRG0		$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
176			PJ0	TIO2A			$PV_{cc}2$	$PV_{cc}2+0.3$	
177			PJ1	TIO2B			$PV_{cc}2$	$PV_{cc}2+0.3$	
178			PJ2	TIO2C			$PV_{cc}2$	$PV_{cc}2+0.3$	
179			PJ3	TIO2D			$PV_{cc}2$	$PV_{cc}2+0.3$	
180			PJ4	TIO2E			$PV_{cc}2$	$PV_{cc}2+0.3$	
181			PJ5	TIO2F			$PV_{cc}2$	$PV_{cc}2+0.3$	
182			PJ6	TIO2G			$PV_{cc}2$	$PV_{cc}2+0.3$	
183			PJ7	TIO2H			$PV_{cc}2$	$PV_{cc}2+0.3$	
184			PJ8	TIO5C			$PV_{cc}2$	$PV_{cc}2+0.3$	
185	V_{ss}								
186			PJ9	TIO5D			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin

Pin No.	Supply Pin Power Supply Name	Dedicated Pin	User Pin Function 1	Function 2	Function 3	Function 4	Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
187	V_{cc}								
188			PJ10	TI9A			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
189			PJ11	TI9B			$PV_{cc}2$	$PV_{cc}2+0.3$	
190			PJ12	TI9C			$PV_{cc}2$	$PV_{cc}2+0.3$	
191			PJ13	TI9D			$PV_{cc}2$	$PV_{cc}2+0.3$	
192			PJ14	TI9E			$PV_{cc}2$	$PV_{cc}2+0.3$	
193			PJ15	TI9F			$PV_{cc}2$	$PV_{cc}2+0.3$	
194	$PV_{cc}2$								
195			PK0	TO8A			$PV_{cc}2$	$PV_{cc}2+0.3$	
196	V_{ss}								
197			PK1	TO8B			$PV_{cc}2$	$PV_{cc}2+0.3$	
198			PK2	TO8C			$PV_{cc}2$	$PV_{cc}2+0.3$	
199			PK3	TO8D			$PV_{cc}2$	$PV_{cc}2+0.3$	
200			PK4	TO8E			$PV_{cc}2$	$PV_{cc}2+0.3$	
201			PK5	TO8F			$PV_{cc}2$	$PV_{cc}2+0.3$	
202			PK6	TO8G			$PV_{cc}2$	$PV_{cc}2+0.3$	
203	V_{cc}								
204			PK7	TO8H			$PV_{cc}2$	$PV_{cc}2+0.3$	
205	V_{ss}								
206			PK8	TO8I			$PV_{cc}2$	$PV_{cc}2+0.3$	
207			PK9	TO8J			$PV_{cc}2$	$PV_{cc}2+0.3$	
208			PK10	TO8K			$PV_{cc}2$	$PV_{cc}2+0.3$	
209			PK11	TO8L			$PV_{cc}2$	$PV_{cc}2+0.3$	
210			PK12	TO8M			$PV_{cc}2$	$PV_{cc}2+0.3$	
211			PK13	TO8N			$PV_{cc}2$	$PV_{cc}2+0.3$	
212	$PV_{cc}2$								
213			PK14	TO8O			$PV_{cc}2$	$PV_{cc}2+0.3$	
214	V_{ss}								
215			PK15	TO8P			$PV_{cc}2$	$PV_{cc}2+0.3$	

Pin No.	Supply Pin Power Supply Name	Dedicated Pin	User Pin Function 1	Function 2	Function 3	Function 4	Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
216			PL0	TI10			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
217			PL1	TIO11A	$\overline{IRQ6}$		$PV_{cc}2$	$PV_{cc}2+0.3$	
218			PL2	TIO11B	$\overline{IRQ7}$		$PV_{cc}2$	$PV_{cc}2+0.3$	
219			PL3	TCLKB			$PV_{cc}2$	$PV_{cc}2+0.3$	
220			PL4	$\overline{ADTRG0}$			$PV_{cc}2$	$PV_{cc}2+0.3$	
221			PL5	$\overline{ADTRG1}$			$PV_{cc}2$	$PV_{cc}2+0.3$	
222			PL6	ADEND			$PV_{cc}2$	$PV_{cc}2+0.3$	
223			PL7	SCK2			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
224			PL8	SCK3			$PV_{cc}2$	$PV_{cc}2+0.3$	
225	V_{CL}								
226			PL9	SCK4	$\overline{IRQ5}$		$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
227	V_{SS}								
228			PL10	HTxD0	HTxD1	HTxD0 and 1	$PV_{cc}2$	$PV_{cc}2+0.3$	
229			PL11	HRxD0	HRxD1	HRxD0, 1	$PV_{cc}2$	$PV_{cc}2+0.3$	
230			PL12	$\overline{IRQ4}$			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
231			PL13	$\overline{IRQOUT}$	$\overline{IRQOUT}$		$PV_{cc}2$	$PV_{cc}2+0.3$	
232			TMS					$V_{cc}+0.3$	
233			$\overline{TRST}$					$V_{cc}+0.3$	
234			TDI					$V_{cc}+0.3$	
235			TDO				V_{cc}		
236			TCK					$V_{cc}+0.3$	
237	V_{cc}								
238			$\overline{AUDRST}$					$PV_{cc}2+0.3$	
239	V_{SS}								

Pin No.	Supply Pin: Power Supply Name	Supply Pin: Dedicated Pin	User Pin: Function 1	User Pin: Function 2	User Pin: Function 3	User Pin: Function 4	Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
240			AUDMD					$PV_{cc}2+0.3$	
241			AUDATA0				$PV_{cc}2$	$PV_{cc}2+0.3$	
242			AUDATA1				$PV_{cc}2$	$PV_{cc}2+0.3$	
243			AUDATA2				$PV_{cc}2$	$PV_{cc}2+0.3$	
244			AUDATA3				$PV_{cc}2$	$PV_{cc}2+0.3$	
245			AUDCK				$PV_{cc}2$	$PV_{cc}2+0.3$	
246			$\overline{\text{AUDSYNC}}$				$PV_{cc}2$	$PV_{cc}2+0.3$	
247	$PV_{cc}2$								
248			PD0	TIO1A			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
249	V_{ss}								
250			PD1	TIO1B			$PV_{cc}2$	$PV_{cc}2+0.3$	Schmitt-trigger input pin
251			PD2	TIO1C			$PV_{cc}2$	$PV_{cc}2+0.3$	
252			PD3	TIO1D			$PV_{cc}2$	$PV_{cc}2+0.3$	
253			PD4	TIO1E			$PV_{cc}2$	$PV_{cc}2+0.3$	
254			PD5	TIO1F			$PV_{cc}2$	$PV_{cc}2+0.3$	
255			PD6	TIO1G			$PV_{cc}2$	$PV_{cc}2+0.3$	
256			PD7	TIO1H			$PV_{cc}2$	$PV_{cc}2+0.3$	

V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V

When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$

The $PV_{CC}1$ power supply voltage depends on the operating mode as shown below. Operation cannot be guaranteed with other $PV_{CC}1$ power supply voltages.

Table 26.3 $PV_{CC}1$ Voltage in Each Operating Mode

Operating Mode No.	Pin Setting				Mode Name	$PV_{CC}1$ Voltage
	FEW	MD2	MD1	MD0		
Mode 0	0	1	0	0	MCU expanded mode	3.3 V ±0.3 V
Mode 1	0	1	0	1		
Mode 2	0	1	1	0		
Mode 3	0	1	1	1	MCU Single-chip mode	5.0 V ±0.5 V
Mode 4	1	1	0	0	Boot mode	3.3 V ±0.3 V
Mode 5	1	1	0	1		5.0 V ±0.5 V
Mode 6	1	1	1	0	User program mode	3.3 V ±0.3 V
Mode 7	1	1	1	1		5.0 V ±0.5 V
Mode 8	1	0	0	0	User boot mode	3.3 V ±0.3 V
Mode 9	1	0	0	1		5.0 V ±0.5 V

$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input high-level voltage (except Schmitt trigger input voltage)	$\overline{RES}$, NMI, FWE, MD2–0, $\overline{HSTBY}$	V_{IH}	V_{CC} – 0.5	—	5.8	V	2.7 V ≤ V_{CC} ≤ 3.6V
	EXTAL		V_{CC} × 0.7	—	V_{CC} + 0.3	V	
	D15–D0, $\overline{WAIT}$, $\overline{BREQ}$ (When in MCU expansion mode)		2.2	—	$PV_{CC}1$ + 0.3	V	$PV_{CC}1$ = 3.3 V ± 0.3 V
	PE15–PE0, PF15–PF0, PH15–PH0 (When in MCU expansion mode)		2.2	—	$PV_{CC}1$ + 0.3	V	$PV_{CC}1$ = 3.3 V ± 0.3 V
	$\overline{TRST}$		V_{CC} – 0.5	—	V_{CC} + 0.3	V	
	TMS, TDI, TCK		2.2	—	V_{CC} + 0.3	V	
	$\overline{AUDRST}$, AUDMD		V_{CC} – 0.5	—	$PV_{CC}2$ + 0.3	V	
	PG0, PL11		PV_{CC} × 0.7	—	$PV_{CC}2$ + 0.3	V	
	Other input pins		2.2	—	PVcc + 0.3	V	
Input low-level voltage (except Schmitt trigger input voltage)	$\overline{RES}$, NMI, FWE, MD2–0, $\overline{HSTBY}$, $\overline{TRST}$, $\overline{AUDRST}$, AUDMD	V_{IL}	–0.3	—	0.5	V	2.7 V ≤ V_{CC} ≤ 3.6V
	PG0, PL11		–0.3	—	$PV_{CC}2$ × 0.3	V	
	Other input pins		–0.3	—	0.8	V	

$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Schmitt trigger input voltage	TI0A–TI0D, TIO1A–TIO1H, TIO2A–TIO2H, TIO3A–TIO3D, TIO4A–TIO4D, TIO5A–TIO5D, TI9A–TI9F, TI10, TIO11A–TIO11B, TCLKA, TCLKB, $\overline{\text{ADTRG0}}$, $\overline{\text{ADTRG1}}$, SCK0–SCK4, $\overline{\text{IRQ0}}$–$\overline{\text{IRQ7}}$ and when thses pins are selected as I/O ports	(V_{IH}) V_T^+	4.0	—	($PV_{CC}2$ + 0.3)	V	Refer to table 26.2, Correspondence between Power Supply Names and Pins
		(V_{IL}) V_T^-	(–0.3)	—	1.0	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input leak current	$\overline{\text{RES}}$, NMI, FWE, MD2–0, $\overline{\text{HSTBY}}$,	| Iin |	—	—	3.0*[1]	µA	Vin = 0.5 V to 5.8 V
					6.0*[2]		
	EXTAL (Standby)		—	—	3.0*[1]	µA	Vin = 0.5 V to V_{CC} – 0.5 V
					6.0*[2]		
	TMS, $\overline{\text{TRST}}$, TDI, TCK (Standby)		—	—	3.0*[1]	µA	Vin = 0.5 V to V_{CC} – 0.5 V
					6.0*[2]		
	AUDMD, AUDCK, $\overline{\text{AUDSYNC}}$, AUDATA3–0 (Standby)		—	—	3.0*[1]	µA	Vin = 0.5 V to $PV_{CC}2$ – 0.5 V
					6.0*[2]		
	$\overline{\text{AUDRST}}$ (Standby)		—	—	3.0*[1]	µA	Vin = 0.5 V to $PV_{CC}2$ – 0.5 V
					6.0*[2]		
	A/D port		—	—	0.2*[1]	µA	Vin = 0 to AV_{CC}
					0.4*[2]		

$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input leak current	D15–D0, $\overline{WAIT}$, $\overline{BREQ}$	| Iin |	—	—	3.0*[1] 6.0*[2]	µA	Vin = 0.5 V to $PV_{CC}1$ – 0.5 V $PV_{CC}1$ = 3.3 V ± 0.3 V
	PE15–PE0, PF15–PF0, PH15–PH0 (When in MCU expansion mode)		—	—	3.0*[1] 6.0*[2]	µA	Vin = 0.5 V to $PV_{CC}1$ – 0.5 V $PV_{CC}1$ = 3.3 V ± 0.3 V
	Other input pins		—	—	3.0*[1] 6.0*[2]	µA	Vin = 0.5 V to $PV_{CC}2$ – 0.5 V
Input pull-up MOS current	TMS, $\overline{TRST}$, TDI, TCK (pull-up characteristic)	–Ipu	—	—	350	µA	Vin = 0 V
	AUDMD, AUDCK, $\overline{AUDSYNC}$, AUDATA3-0 (pull-up characteristic)		—	—	800	µA	Vin = 0 V
Input pull-down MOS current	$\overline{AUDRST}$ (pull-down characteristic)	Ipd	—	—	500	µA	Vin = $PV_{CC}2$
Three-state leak current (while OFF)	A21–A0, D15–D0, $\overline{CS3}$–$\overline{CS0}$, $\overline{WRH}$, $\overline{WRL}$, $\overline{RD}$, $\overline{BACK}$ (When in MCU expansion mode)	| Its |	—	—	3.0*[1] 6.0*[2]	µA	Vin = 0.5 to $PV_{CC}1$ – 0.5 V $PV_{CC}1$ = 3.3 V ± 0.3 V
Output high-level voltage	A21–A0, D15–D0, $\overline{CS3}$–$\overline{CS0}$, $\overline{WRH}$, $\overline{WRL}$, $\overline{RD}$, $\overline{BACK}$ (When in MCU expansion mode)	V_{OH}	PVcc1–0.5	—	—	V	I_{OH} = 200 µA $PV_{CC}1$ = 3.3 V ± 0.3 V
	PE15–PE0, PF15–PF0, PH15–PH0 (When in MCU expansion mode)		PVcc1–0.5	—	—	V	I_{OH} = 200 µA $PV_{CC}1$ = 3.3 V ± 0.3 V
	CK, TDO		V_{CC} – 0.5	—	—	V	I_{OH} = 200 µA

$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Output high-level voltage	Other output pins	V_{OH}	PV_{CC} – 0.5	—	—	V	I_{OH} = 200 µA
			PV_{CC} – 1.0	—	—	V	I_{OH} = 1 mA
Output low-level voltage	A21–A0, D15–D0, $\overline{CS3}$–$\overline{CS0}$, $\overline{WRH}$, $\overline{WRL}$, $\overline{RD}$, $\overline{BACK}$ (When in MCU expansion mode)	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA $PV_{CC}1$ = 3.3 V ± 0.3 V
	PE15–PE0, PF15–PF0, PH15–PH0 (When in MCU expansion mode)		—	—	0.4	V	I_{OL} = 1.6 mA $PV_{CC}1$ = 3.3 V ± 0.3 V
	Other output pins (except XTAL)		—	—	0.4	V	I_{OL} = 1.6 mA
			—	—	1.2	V	I_{OL} = 6 mA
Input capacitance	$\overline{RES}$	Cin	—	—	60	pF	Vin = 0 V
	NMI		—	—	30	pF	f = 1 MHz
	All other input pins		—	—	20	pF	T_a = 25°C
Current consumption	Normal operation	I_{CC}	—	50	80	mA	f = 40 MHz
	Sleep		—	40	60	mA	
	Standby		—	50	200	µA	$T_a \leq 50°C$
			—	—	500	µA	50°C < Ta≤105°C
			—	—	1000	µA	Ta > 105°C
	Write operation		—	60	90	mA	V_{CC} = 3.3 V f = 40 MHz
Analog supply current	During A/D conversion	AI_{CC}	—	1.2	5	mA	
	Awaiting A/D conversion		—	1	30	µA	

$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Reference power supply current	During A/D conversions	Alref	—	1.3	5	mA	AV_{ref} = 5 V
	Awaiting A/D conversion		—	1.1	10	µA	
RAM standby voltage		V_{RAM}	2.7	—	—	V	V_{CC}

Notes: *1 Ta≤105°C
*2 Ta>105°C

[Operating precautions]

1. When the A/D converter is not used (including during standby), do not leave the AV_{CC}, AV_{ref}, and AV_{SS} pins open.
2. The current consumption is measured when V_{IH}min = V_{CC} – 0.3 V/PV_{CC} – 0.3 V, V_{IL} = 0.3 V, with all output pins unloaded.
3. The guaranteed operating range of power supply $PV_{CC}1$ in the MCU expanded modes is only $PV_{CC}1$ = 3.3 V ±0.3 V. Do not use a voltage outside this range.
4. The guaranteed operating range of power supply $PV_{CC}1$ in MCU single-chip mode is only $PV_{CC}1$ = 5.0 V ±0.5 V. Do not use a voltage outside this range.

$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, Ta = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	6	mA
Output low-level permissible current (total)	ΣI_{OL}	—	—	80	mA
Output high-level permissible current (per pin)	I_{OH}	—	—	2	mA
Output high-level permissible current (total)	ΣI_{OL}	—	—	25	mA

[Operating precautions]

To assure LSI reliability, do not exceed the output values listed in this table.

26.3.1 Timing for swicthing the power supply on/off

Table 26.6 Timing for swicthing the power supply on/off

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
Time taken to switch V_{CC} on	t_{VCCS}	0	—	ms	Figure 26.1
V_{CC} hold-time when PV_{CC} is swtched off	t_{VCCH}	0	—	ms	

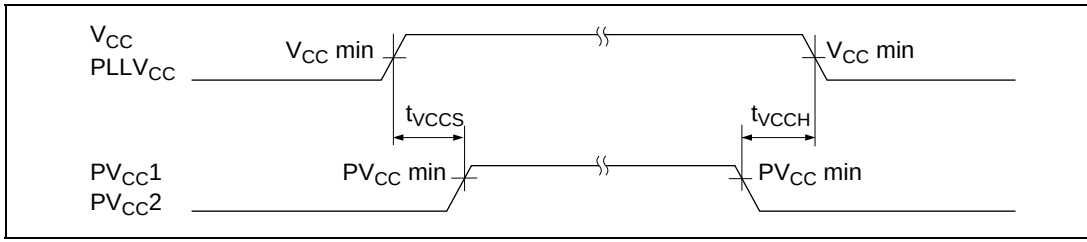


Figure 26.1 Power-On/Off Timing

Table 26.7 shows the clock timing.

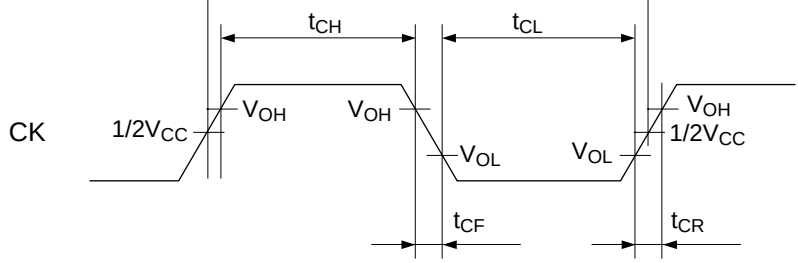
Table 26.7 Clock Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
Operating frequency	f_{op}	20	40	MHz	Figure 26.2
Clock cycle time	t_{cyc}	25	50	ns	
Clock low-level pulse width	t_{CL}	4	—	ns	
Clock high-level pulse width	t_{CH}	4	—	ns	
Clock rise time	t_{CR}	—	8	ns	
Clock fall time	t_{CF}	—	8	ns	
EXTAL clock input frequency	f_{EX}	5	10	MHz	Figure 26.3
EXTAL clock input cycle time	t_{Excyc}	100	200	ns	
EXTAL clock input low-level pulse width	t_{EXL}	30	—	ns	
EXTAL clock input low-level pulse width	t_{EXH}	30	—	ns	
EXTAL clock input rise time	t_{EXR}	—	8	ns	
EXTAL clock input fall time	t_{EXF}	—	8	ns	
Reset oscillation settling time	t_{osc1}	30	—	ms	Figure 26.4
Standby return clock settling time	t_{osc2}	30	—	ms	

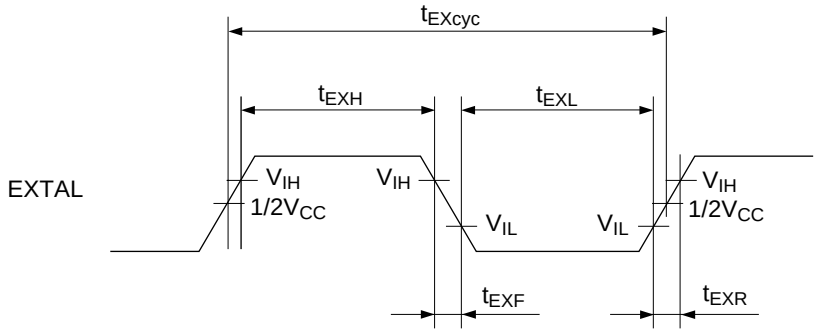
[Operating precautions]

The EXTAL, XTAL, and CK pins constitute a circuit requiring a power supply voltage of V_{CC} = 3.3 V ±0.3 V. Comply with the input and output voltages specified in the DC characteristics.



Note: CK pin is V_{CC} = 3.3 V ±0.3 V power supply circuit.

Figure 26.2 System Clock Timing



Note: EXTAL pin is V_{CC} = 3.3 V ±0.3 V power supply circuit.

Figure 26.3 EXTAL Clock Input Timing

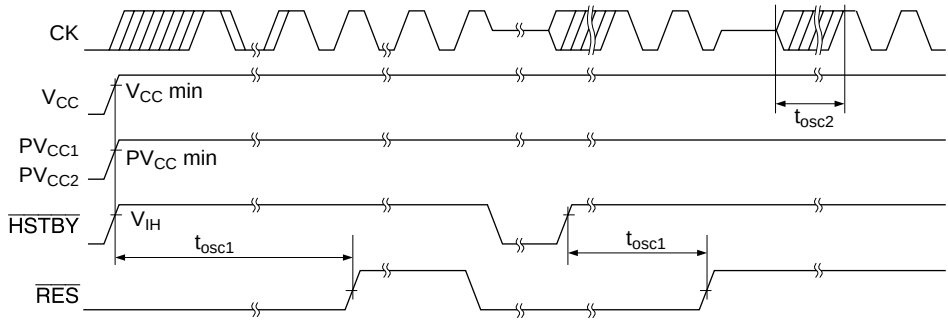


Figure 26.4 Oscillation Settling Time

Table 26.8 shows control signal timing.

Table 26.8 Control Signal Timing

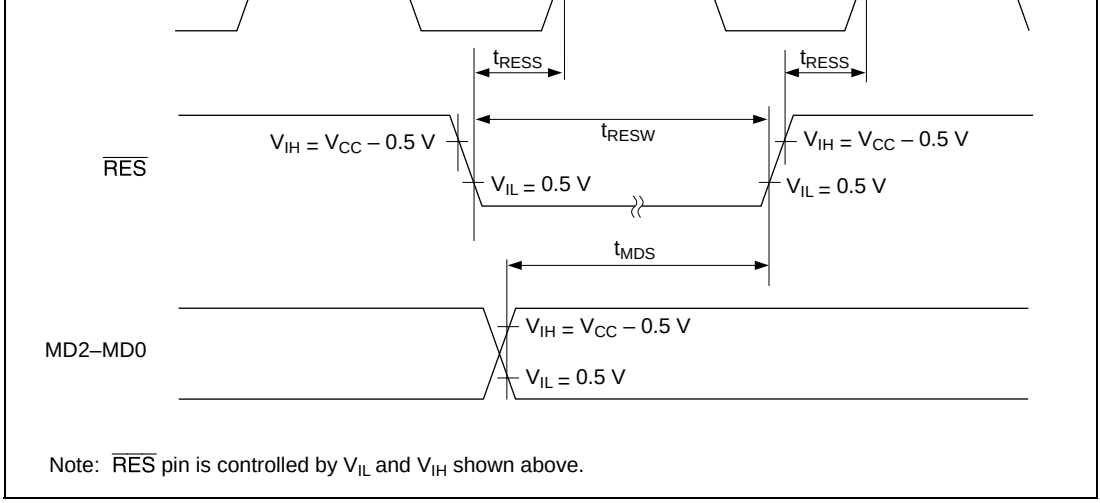
Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	Figure 26.5
$\overline{\text{RES}}$ setup time	t_{RESS}	40	—	ns	
MD2–MD0 setup time	t_{MDS}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	24	—	ns	Figure 26.6
$\overline{\text{IRQ7}}$–$\overline{\text{IRQ0}}$ setup time*[1] (edge detection)	t_{IRQES}	24	—	ns	
$\overline{\text{IRQ7}}$–$\overline{\text{IRQ0}}$ setup time*[1] (level detection)	t_{IRQLS}	24	—	ns	
NMI hold time	t_{NMIH}	24	—	ns	
$\overline{\text{IRQ7}}$–$\overline{\text{IRQ0}}$ hold time	t_{IRQEH}	24	—	ns	
$\overline{\text{IRQOUT}}$ output delay time	t_{IRQOD}	—	100	ns	Figure 26.7
Bus request setup time	t_{BRQS}	24	—	ns	Figure 26.8*[2]
Bus acknowledge delay time 1	t_{BACKD1}	—	30	ns	
Bus acknowledge delay time 2	t_{BACKD2}	—	30	ns	
Bus three-state delay time	t_{BZD}	—	30	ns	

[Operating precautions]

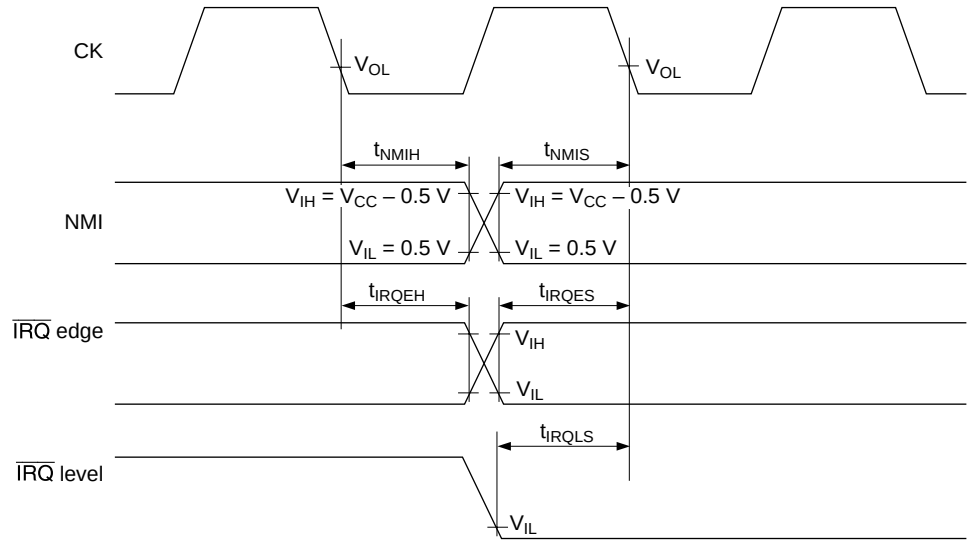
*1 The $\overline{\text{RES}}$, NMI, and $\overline{\text{IRQ7}}$–$\overline{\text{IRQ0}}$ signals are asynchronous inputs, but when the setup times shown here are provided, the signals are considered to have been changed at clock fall. If the setup times are not provided, recognition is delayed until the next clock rise or fall.

*2 The guaranteed operating range of power supply $PV_{CC}1$ in the MCU expanded modes is only $PV_{CC}1$ = 3.3 V ±0.3 V. Do not use a voltage outside this range.



Note: $\overline{RES}$ pin is controlled by V_{IL} and V_{IH} shown above.

Figure 26.5 Reset Input Timing



Note: NMI pin is controlled by V_{IL} and V_{IH} shown above.

Figure 26.6 Interrupt Signal Input Timing

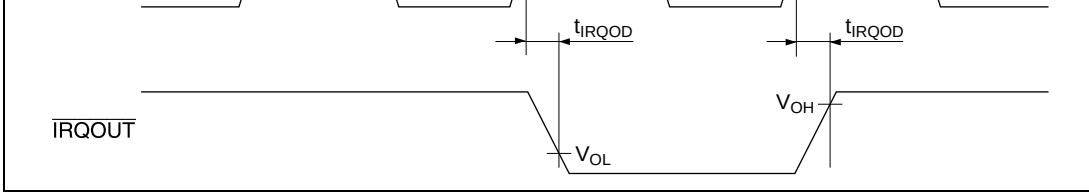


Figure 26.7 Interrupt Signal Output Timing

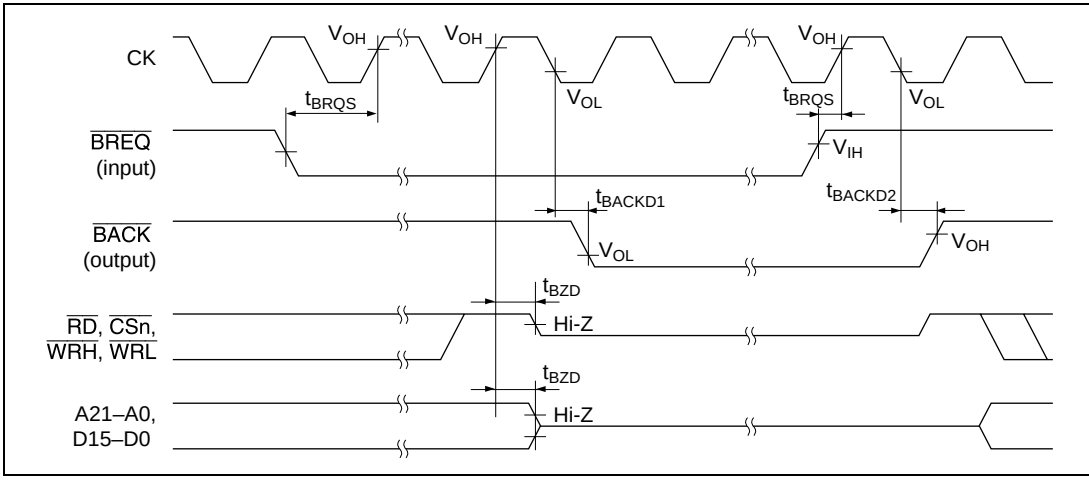


Figure 26.8 Bus Right Release Timing

Table 26.9 shows bus timing.

Table 26.9 Bus Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
Address delay time	t_{AD}	—	35	ns	Figure 26.9, 26.10
CS delay time 1	t_{CSD1}	—	30	ns	
CS delay time 2	t_{CSD2}	—	30	ns	
Read strobe delay time 1	t_{RSD1}	—	30	ns	
Read strobe delay time 2	t_{RSD2}	—	30	ns	
Read data setup time	t_{RDS}	15	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Write strobe delay time 1	t_{WSD1}	—	30	ns	
Write strobe delay time 2	t_{WSD2}	—	30	ns	
Write data delay time	t_{WDD}	—	30	ns	
Write data hold time	t_{WDH}	$t_{cyc} \times m$	—	ns	
WAIT setup time	t_{WTS}	15	—	ns	Figure 26.11
WAIT hold time	t_{WTH}	0	—	ns	
Read data access time	t_{ACC}	$t_{cyc} \times (n+1.5) – 39$	—	ns	Figure 26.9, 26.10
Access time from read strobe	t_{OE}	$t_{cyc} \times (n+1.0) – 39$	—	ns	
Write address setup time	t_{AS}	0	—	ns	
Write address hold time	t_{WR}	5	—	ns	

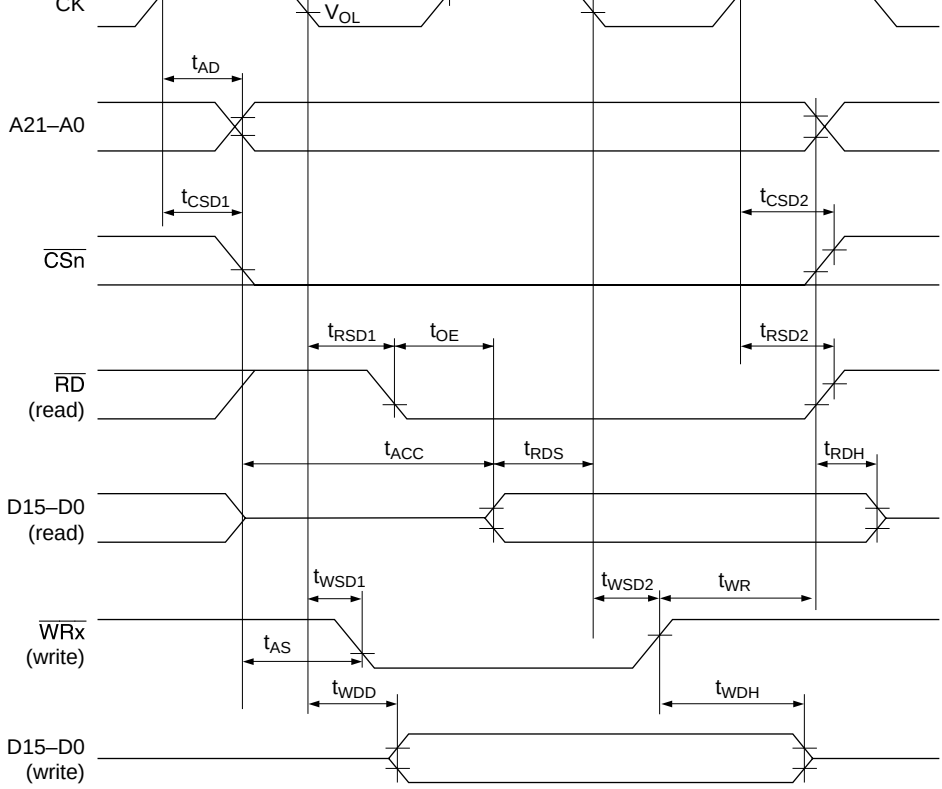
n: Number of waits

m = 1: CS assertion extension cycle

m = 0: Normal cycle (CS assertion non-extension cycle)

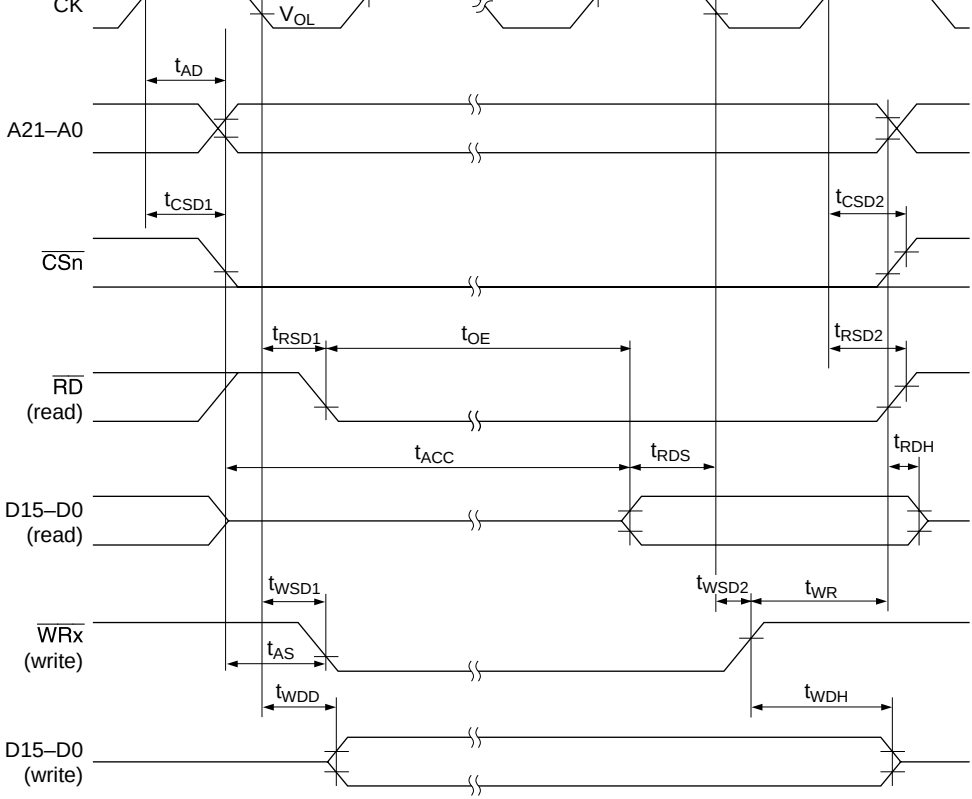
[Operating precautions]

The guaranteed operating range of power supply $PV_{CC}1$ in the MCU expanded modes is only $PV_{CC}1$ = 3.3 V ±0.3 V. Do not use a voltage outside this range.



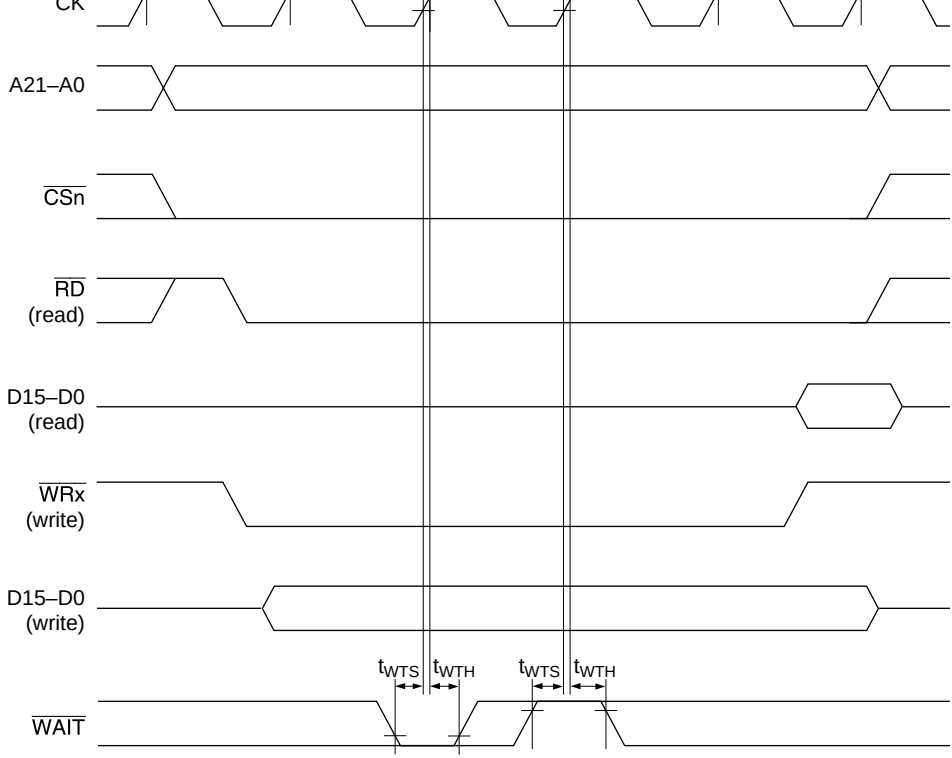
Note: t_{RDH}: Specified from the negate timing of A21–A0, CSn, or RD, whichever is first.

Figure 26.9 Basic Cycle (No Waits)



Note: t_{RDH}: Specified from the negate timing of A21–A0, $\overline{CSn}$, or $\overline{RD}$, whichever is first.

Figure 26.10 Basic Cycle (One Software Wait)



Note: t_{RDH}: Specified from the negate timing of A21–A0, $\overline{CSn}$, or $\overline{RD}$, whichever is first.

Figure 26.11 Basic Cycle (Two Software Waits + Waits by $\overline{WAIT}$ Signal)

Table 26.10 shows advanced timer unit timing and advanced pulse controller timing.

Table 26.10 Advanced Timer Unit Timing and Advanced Pulse Controller Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
Output compare output delay time	t_{TOCD}	—	100	ns	Figure 26.12
Input capture input setup time	t_{TICS}	24* 24 + t_{cyc}	—	ns	
PULS output delay time	t_{PLSD}	–	100	ns	
Timer clock input setup time	t_{TCKS}	24* 24 + t_{cyc}	—	ns	Figure 26.13
Timer clock pulse width (single edge specified)	$t_{TCKWH/L}$	3.0	—	t_{cyc}	
Timer clock pulse width (both edges specified)	$t_{TCKWH/L}$	5.0	—	t_{cyc}	

[Operating precautions]

* The timer input signals and timer clock input signals are asynchronous, but judged to have been changed at clock rise with two-state intervals shown in figures 26.12 and 26.13. If the setup times shown here are not provided, recognition is delayed until the clock rise two states after that timing.

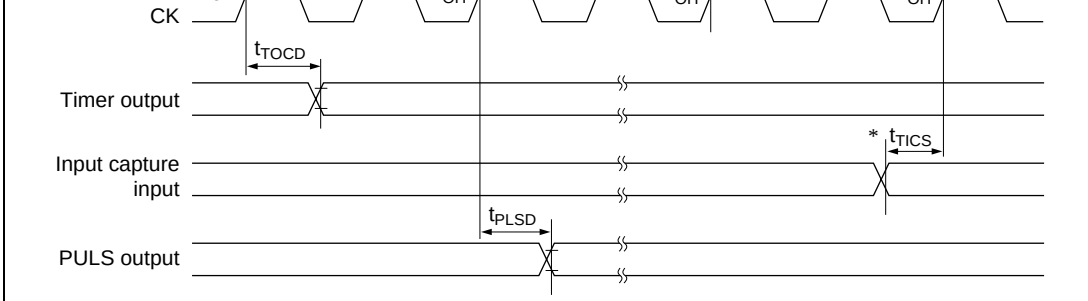


Figure 26.12 ATU Input/Output timing and APC Output timing

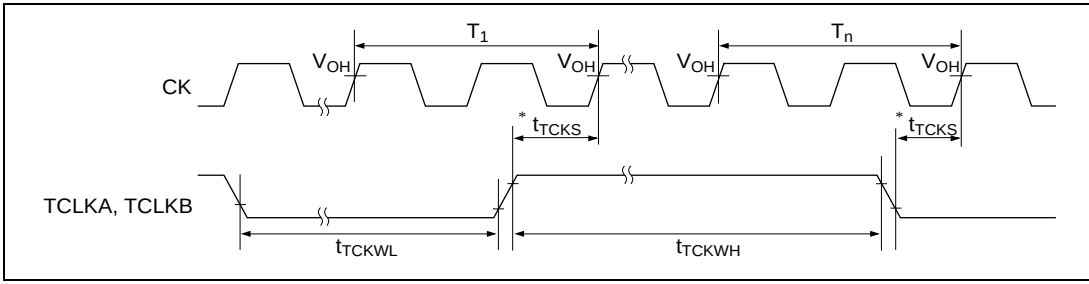


Figure 26.13 ATU Clock Input Timing

Table 26.11 shows I/O port timing.

Table 26.11 I/O Port Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ± 0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
Port output data delay time	t_{PWD}	—	100	ns	Figure 26.14
Port input hold time	t_{PRH}	24* 24+tcyc	—	ns	
Port input setup time	t_{PRS}	24* 24+tcyc	—	ns	

[Operating precautions]

The port input signals are asynchronous, but judged to have been changed at CK clock rise with two-state intervals shown in figure 26.14. If the setup times shown here are not provided, recognition is delayed until the clock rise two states after that timing.

* The guaranteed operating range of power supply $PV_{CC}1$ in MCU single-chip mode is only $PV_{CC}1$ = 5.0 V ±0.5 V. Do not use a voltage outside this range.

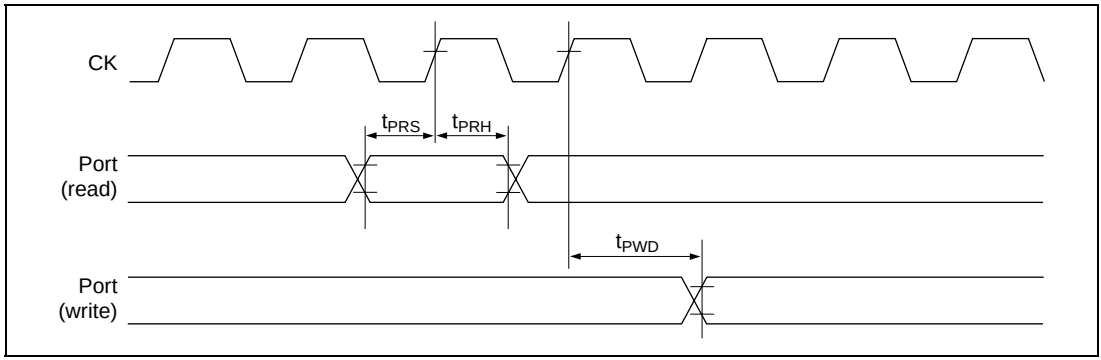


Figure 26.14 I/O Port Input/Output timing

Table 26.12 shows watchdog timer timing.

Table 26.12 Watchdog Timer Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	Figure 26.15

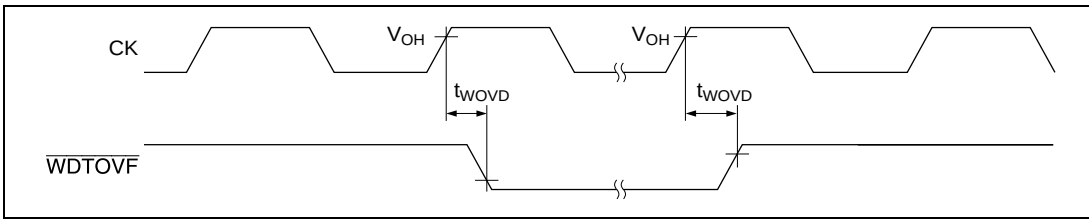


Figure 26.15 Watchdog Timer Timing

Table 26.13 shows serial communication interface timing.

Table 26.13 Serial Communication Interface Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
Clock cycle	t_{scyc}	8	—	t_{cyc}	Figure 26.16
Clock cycle (clock sync)	t_{scyc}	12	—	t_{cyc}	
Clock pulse width	t_{sckw}	0.4	0.6	t_{scyc}	
Input clock rise time	t_{sckr}	—	3.0	t_{cyc}	
Input clock fall time	t_{sckf}	—	3.0	t_{cyc}	
Transmit data delay time	t_{TxD}	—	100	ns	Figure 26.17
Transmit data setup time	t_{RxS}	100	—	ns	
Transmit data hold time	t_{RxH}	100	—	ns	

[Operating precautions]

The inputs and outputs are asynchronous in start-stop synchronous mode, but as shown in figure 26.17, the receive data are judged to have been changed at CK clock rise (two-clock intervals). The transmit signals change with a reference of CK clock rise (two-clock intervals).

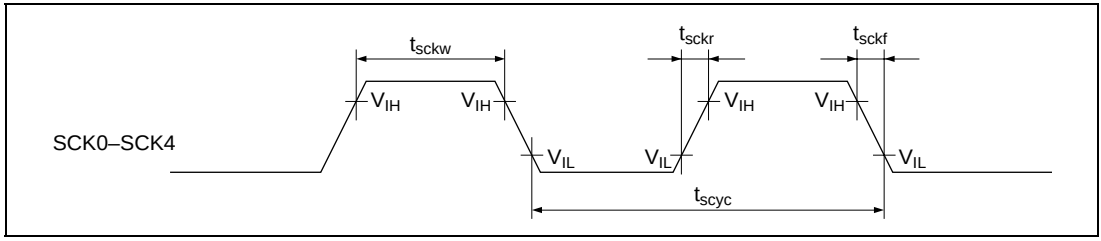
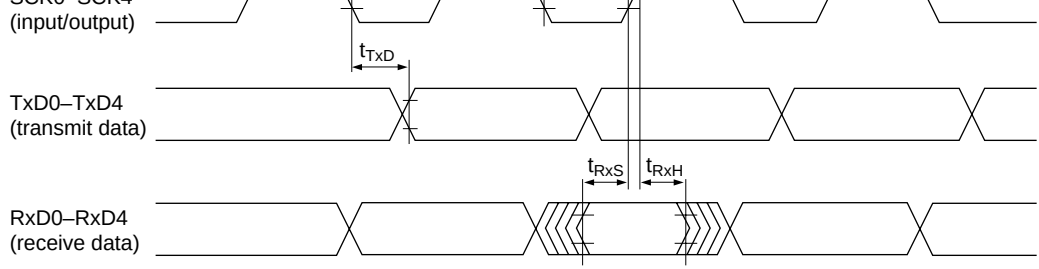
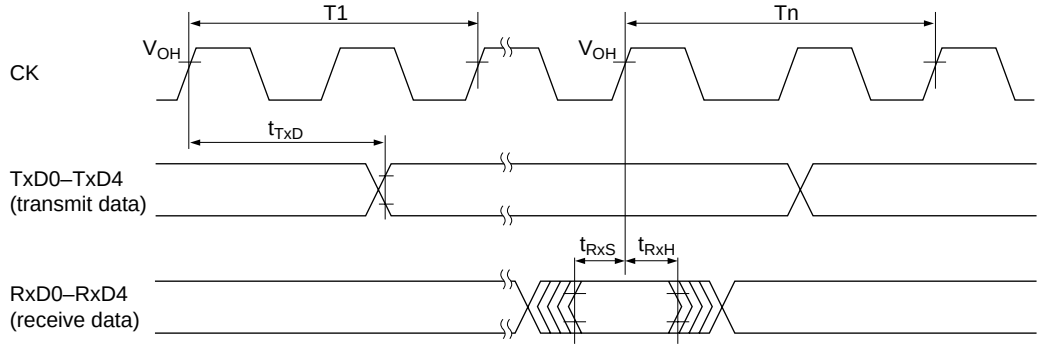


Figure 26.16 SCI Input/Output Timing



SCI input/output timing (clock synchronous mode)



SCI input/output timing (start-stop synchronous mode)

Figure 26.17 SCI Input/Output Timing

Table 26.14 shows HCAN timing.

Table 26.14 HCAN Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
Transmit data delay time	t_{HTxD}	—	100	ns	Figure 26.18
Transmit data setup time	t_{HRxS}	100	—	ns	
Transmit data hold time	t_{HRxH}	100	—	ns	

[Operating precautions]

The HCAN input signals are asynchronous, but judged to have been changed at CK clock rise (two-clock intervals) shown in figure 26.18. The HCAN output signals are asynchronous, but they change with a reference of CK clock rise (two-clock intervals) shown in figure 26.18.

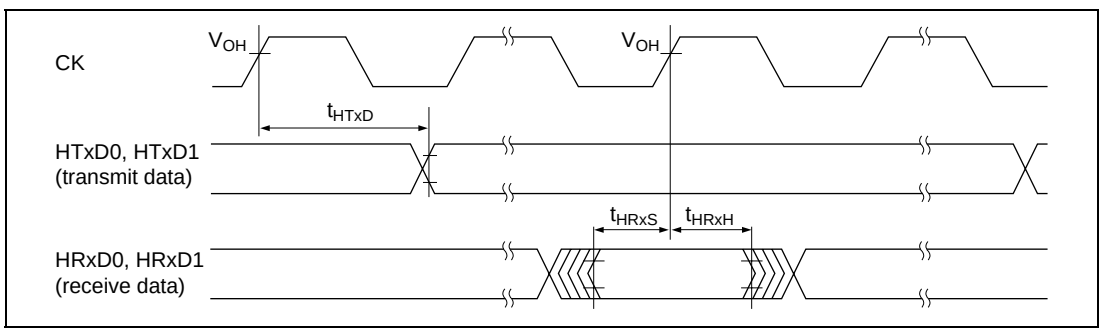


Figure 26.18 HCAN Input/Output timing

Table 26.15 shows A/D converter timing.

Table 26.15 A/D Converter Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

		CSK = 0: fop = 20–40 MHz			CSK = 1: fop = 20 MHz				
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**	**Figure**
External trigger input start delay time	t_{TRGS}	50	—	—	50	—	—	ns	Figure 26.19
A/D conversion time	t_{CONV}	518	—	532	262	—	268	t_{cyc}	Figure 26.20
A/D conversion start delay time	t_D	20	—	34	12	—	18	t_{cyc}	
Input sampling time	t_{SPL}	—	128	—	—	64		t_{cyc}	
ADEND output delay time	t_{ADENDD}	—	—	100	—	—	100	ns	

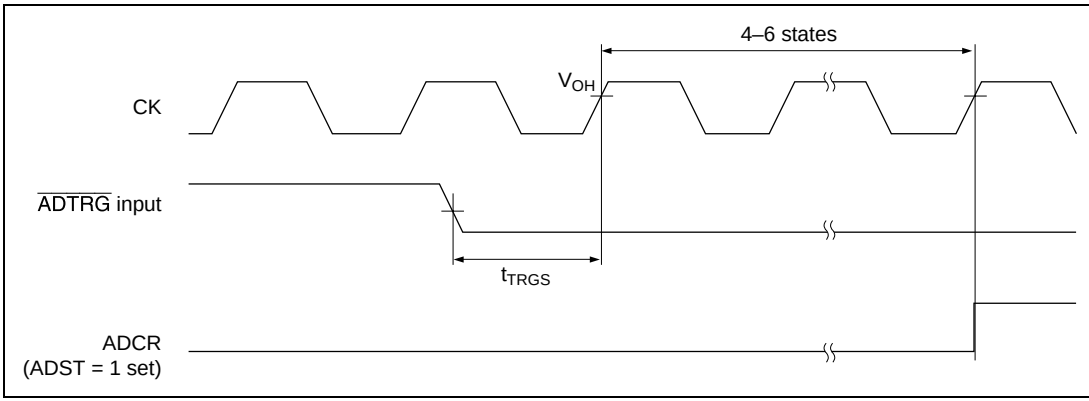


Figure 26.19 External Trigger Input Timing

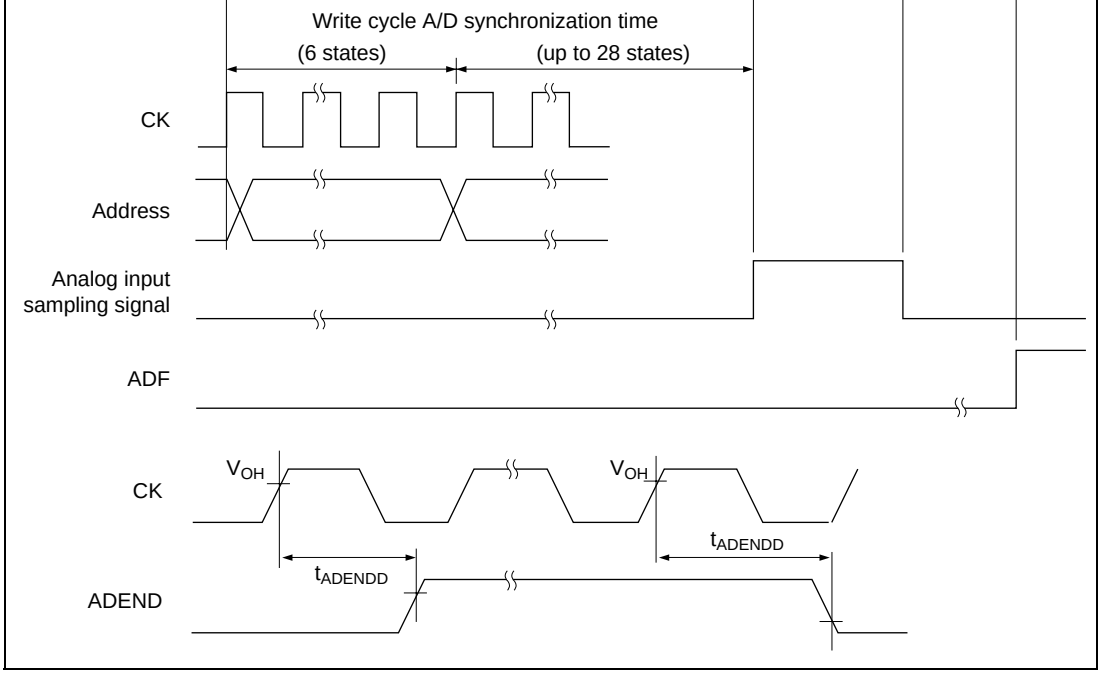


Figure 26.20 Analog Conversion Timing

Table 26.16 shows H-UDI timing.

Table 26.16 H-UDI Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
TCK clock cycle	t_{tcyc}	4	—	t_{tcyc}	Figure 26.21
TCK clock high-level width	t_{TCKH}	0.4	0.6	t_{tcyc}	
TCK clock low-level width	t_{TCKL}	0.4	0.6	t_{tcyc}	
$\overline{\text{TRST}}$ pulse width	t_{TRSW}	20	—	t_{cyc}	Figure 26.22
$\overline{\text{TRST}}$ setup time	t_{TRSS}	30	—	ns	
TMS setup time	t_{TMSS}	30	—	ns	Figure 26.23
TMS hold time	t_{TMSH}	10	—	ns	
TDI setup time	t_{TDIS}	30	—	ns	
TDI hold time	t_{TDIH}	10	—	ns	
TDO delay time	t_{TDOD}	—	30	ns	

[Operating precautions]

The H-UDI pins constitute a circuit requiring the voltage of V_{CC} = 3.3 V ±0.3 V. Comply with the input and output voltages specified in the DC characteristics, for operation.

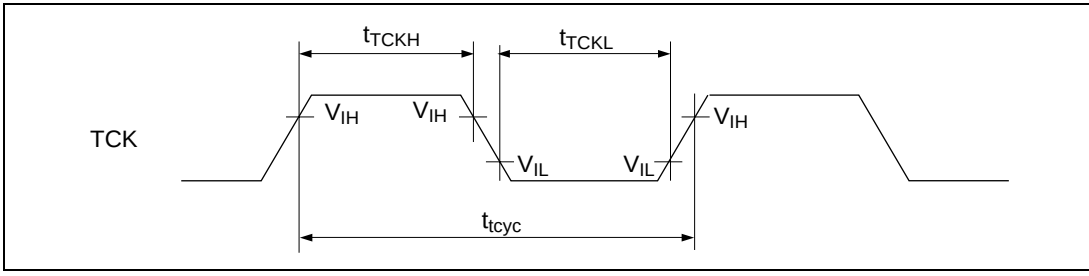


Table 26.21 H-UDI Clock Timing

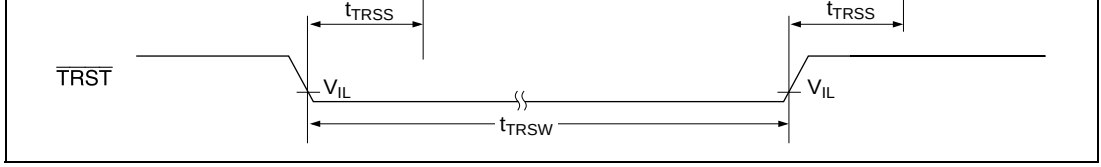


Table 26.22 H-UDI $\overline{\text{TRST}}$ Timing

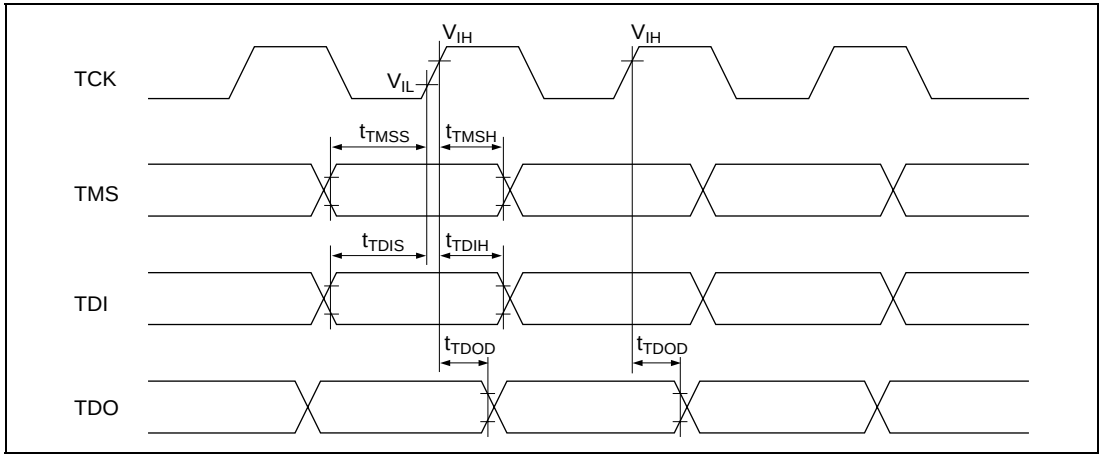


Table 26.23 H-UDI Input/Output Timing

Table 26.17 shows AUD timing.

Table 26.17 AUD Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
$\overline{\text{AUDRST}}$ pulse width (Branch trace)	$t_{AUDRSTW}$	20	—	t_{cyc}	Figure 26.24
$\overline{\text{AUDRST}}$ pulse width (RAM monitor)	$t_{AUDRSTW}$	5	—	t_{RMCYC}	
AUDMD setup time (Branch trace)	t_{AUDMDS}	20	—	t_{cyc}	
AUDMD setup time (RAM monitor)	t_{AUDMDS}	5	—	t_{RMCYC}	
Branch trace clock cycle	t_{BTCYC}	2	2	t_{cyc}	Figure 26.25
Branch trace clock duty	t_{BTCKW}	40	60	%	
Branch trace data delay time	t_{BTDD}	—	40	ns	
Branch trace data hold time	t_{BTDH}	0	—	ns	
Branch trace SYNC delay time	t_{BTSD}	—	40	ns	
Branch trace SYNC hold time	t_{BTSH}	0	—	ns	
RAM monitor clock cycle	t_{RMCYC}	100	—	ns	Figure 26.26
RAM monitor clock low pulse width	t_{RMCKW}	45	—	ns	
RAM monitor output data delay time	t_{RMDD}	7	t_{RMCYC} – 20	ns	
RAM monitor output data hold time	t_{RMDHD}	5	—	ns	
RAM monitor input data setup time	t_{RMDS}	20	—	ns	
RAM monitor input data hold time	t_{RMDH}	5	—	ns	
RAM monitor SYNC setup time	t_{RMSS}	20	—	ns	
RAM monitor SYNC hold time	t_{RMSH}	5	—	ns	

Load conditions: AUDCK (branch trace): CL = 30 pF: otherwise CL = 100 pF
AUDSYNC: CL = 100 pF
AUDATA3 to AUDATA0: CL = 100 pF

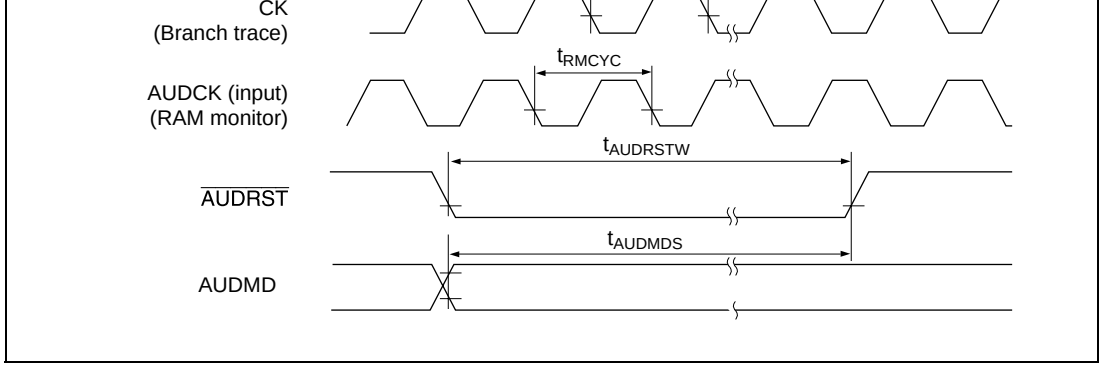


Figure 26.24 AUD Reset Timing

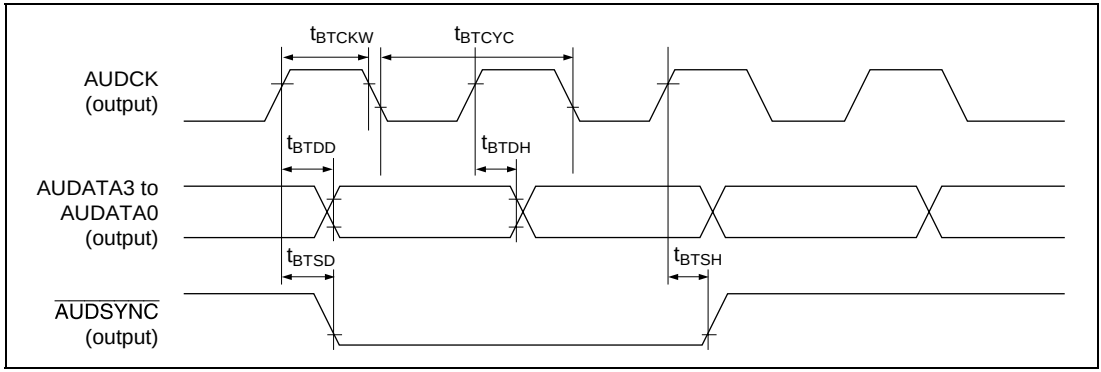


Figure 26.25 Branch Trace Timing

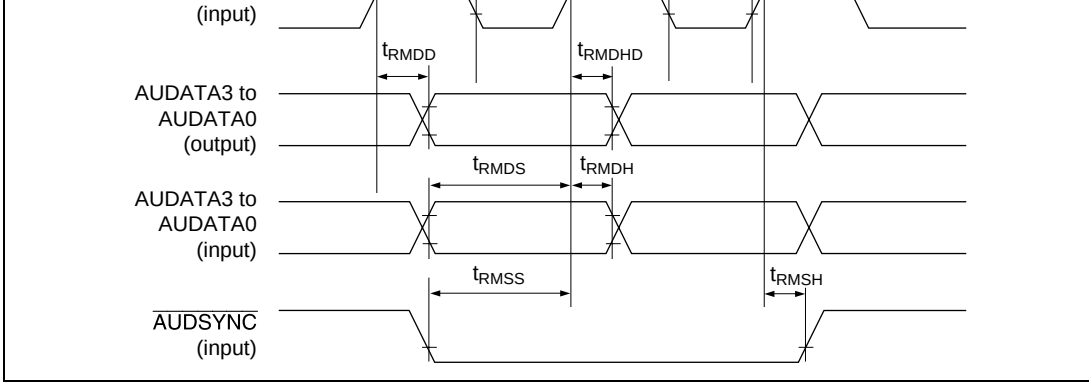


Figure 26.26 RAM Monitor Timing

26.3.13 UBC Trigger Timing

Table 26.18 shows UBC trigger timing.

Table 26.18 UBC Trigger Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Max	Unit	Figures
$\overline{\text{UBCTRG}}$ delay time	t_{UBCTGD}	—	35	ns	Figure 26.27

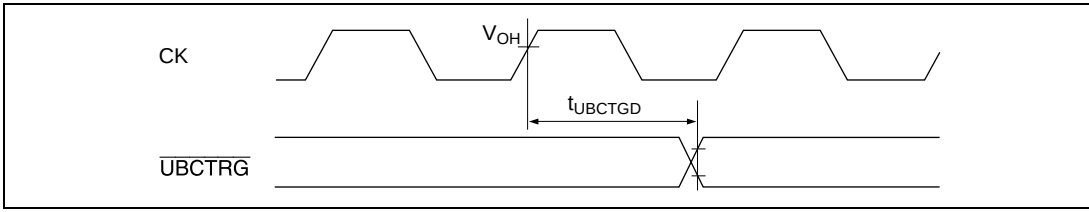
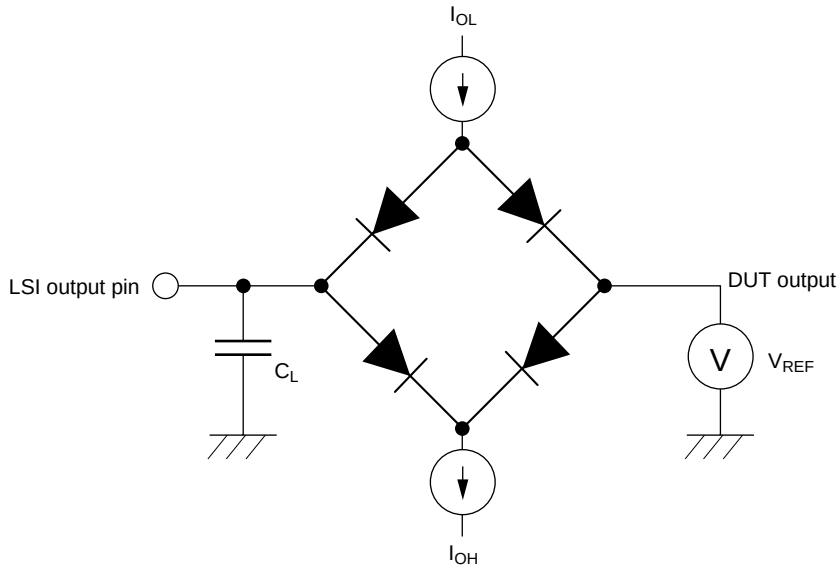


Figure 26.27 UBC Trigger Timing

Input reference levels	High level: V_{IH} min. value, low level: V_{IL} max. value
Output reference level	High level: 2.0 V, Low level: 0.8 V



C_L is a total value that includes the measuring instrument capacitance.
The following C_L values are used:

30 pF: CK, $\overline{CS3}$–$\overline{CS0}$, $\overline{BREQ}$, $\overline{BACK}$, $\overline{IRQOUT}$, AUDCK
50 pF: A21–A0, D15–D0, $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$, TDO
100 pF: AUDATA3–0, AUDSYNC
30 pF: All port pins other than the above, and peripheral module output pins.

I_{OL} and I_{OH} are the condition for the I_{OL} = 1.6 mA, I_{OH} = 200 µA.

Figure 26.28 Output Test Circuit

Table 26.19 shows A/D converter characteristics.

Table 26.19 A/D Converter Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 125°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

	CSK = 0: fop = 10–20 MHz			CSK = 1: fop = 10 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	bit
A/D conversion time	—	—	13.3	—	—	13.4	µs
Analog input capacitance	—	—	20	—	—	20	pF
Permitted analog signal source impedance	—	—	3	—	—	3	kΩ
Non-linear error	—	—	±1.5*[1]	—	—	±1.5*[1]	LSB
			±2.5*[2]			±2.5*[2]	
Offset error	—	—	±1.5*[1]	—	—	±1.5*[1]	LSB
			±2.5*[2]			±2.5*[2]	
Full-scale error	—	—	±1.5*[1]	—	—	±1.5*[1]	LSB
			±2.5*[2]			±2.5*[2]	
Quantization error	—	—	±0.5	—	—	±0.5	LSB
Absolute error	—	—	±2.0*[1]	—	—	±2.0*[1]	LSB
			±2.5*[2]			±2.5*[2]	

Note: *1 Ta≤105°C
*2 Ta>105°C

Table 26.20 shows the flash memory characteristics.

Table 26.20 Flash Memory Characteristics

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V, $PV_{CC}2$ = 5.0 V ±0.5 V, AV_{CC} = 5.0 V ±0.5 V, AV_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –40°C to 105°C.
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
When writing or erasing on-chip flash memory, T_a = –40°C to 85°C.

Item	Symbol	Min	Typ	Max	Unit
Programming time*[1]*[2]	t_P	—	20	200	ms/128 bytes
Erase time*[1]*[3]	t_E	—	1	10	s/block
Reprogramming count	N_{WEC}	—	—	100	Times

Note: *1 Use the on-chip programming/erasing routine for programming/erasure.
*2 When all 0 are programmed.
*3 64 kbytes of block

26.6.1 Notes on Connecting External Capacitor for Current Stabilization

The SH7055SF includes an internal step-down curcuit to automatically reduce the microporocessor power supply voltage to an appropriate level. Between this internal stepped-down power supply (V_{CL} pin) and the V_{SS} pin, an capacitor (0.33 to 0.47 µF) for stabilizing the internal voltage. Connection of the external capacitor is shown in figure 26.29. The external capacitor should be located near the pin. Do not apply any power supply voltage to the V_{CL} pin.

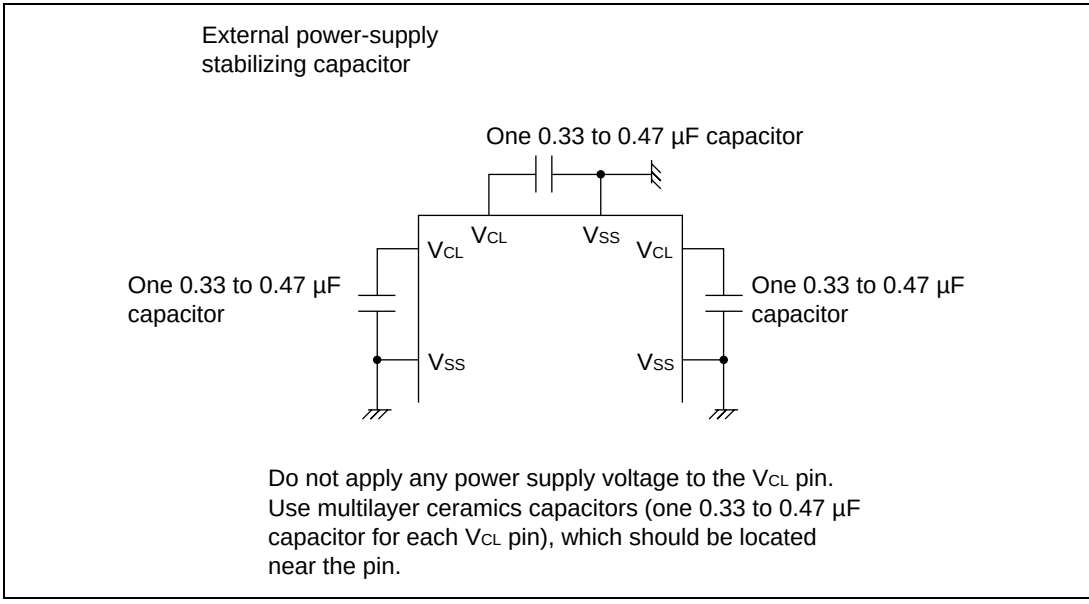


Figure 26.29 Connection of V_{CL} Capacitor

26.6.2 Notes on Mode Pin Input

When power is supplied and in hardware standby mode, mode setup time is determined by tMDS1. When power-on reset is performed only by the $\overline{RES}$ pin, mode setup time is differs according to the combination of input to the FWE and MD2 to MD0. When low is input to the $\overline{RES}$ pin with the pins FWE and MD2 to MD0 operated in mode specified in table 26.3, the mode setup time is determined by tMDS2. When combination which is not specified in table 26.3 is input, the mode setup time is determined by tMDS1.

Mode setup time 1	t_{MDS1}	30	—	—	ms	Figure 26.30
Mode setup time 2	t_{MDS2}	10	—	—	t_{cyc}	

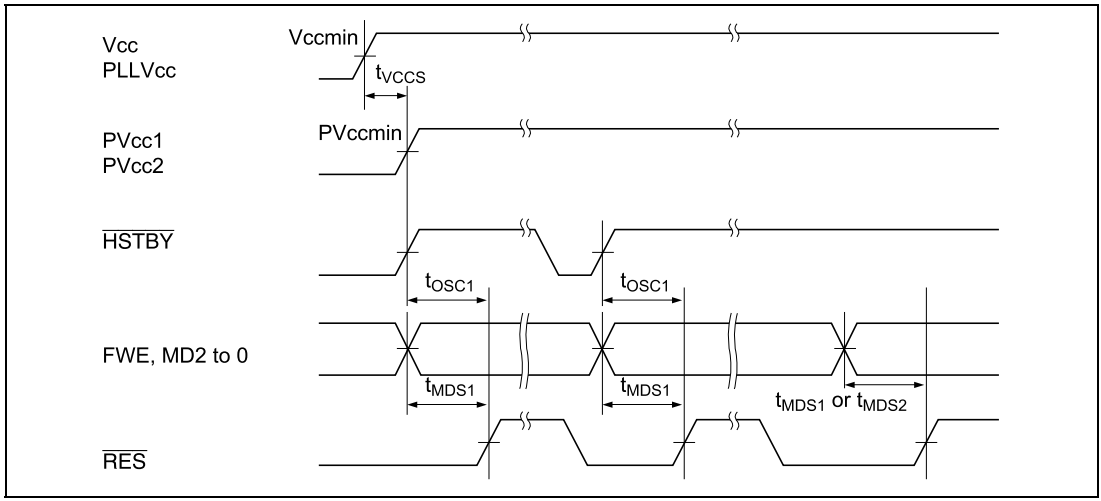


Figure 26.30 Mode Pin Input Timing

A.1 Address

On-chip peripheral module register addresses and bit names are shown in the following table. 16-bit and 32-bit registers are shown in two and four rows of 8 bits, respectively.

Table A.1 Address

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFE400	MCR	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	HCAN (channel 0)
H'FFFFE401	GSR	—	—	—	—	GSR3	GSR2	GSR1	GSR0	
H'FFFFE402	BCR	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0	
H'FFFFE403		BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8	
H'FFFFE404	MBCR	MBCR7	MBCR6	MBCR5	MBCR4	MBCR3	MBCR2	MBCR1	—	
H'FFFFE405		MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9	MBCR8	
H'FFFFE406	TXPR	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—	
H'FFFFE407		TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8	
H'FFFFE408	TXCR	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—	
H'FFFFE409		TXCR15	TXCR14	TCR13	TXCR12	TXCR11	TSCR10	TXCR9	TXCR8	
H'FFFFE40A	TXACK	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—	
H'FFFFE40B		TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8	
H'FFFFE40C	ABACK	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—	
H'FFFFE40D		ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8	
H'FFFFE40E	RXPR	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0	
H'FFFFE40F		RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8	
H'FFFFE410	RFPR	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0	
H'FFFFE411		RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8	
H'FFFFE412	IRR	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
H'FFFFE413		—	—	—	IRR12	—	—	IRR9	IRR8	
H'FFFFE414	MBIMR	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0	
H'FFFFE415		MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8	
H'FFFFE416	IMR	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	—	
H'FFFFE417		—	—	—	IMR12	—	—	IMR9	IMR8	
H'FFFFE418	REC									
H'FFFFE419	TEC									
H'FFFFE41A	UMSR	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0	
H'FFFFE41B		UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE41C	LAFML	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0	HCAN (channel 0)
H'FFFFE41D		LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8	
H'FFFFE41E	LAFMH	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1	LAFMH0	
H'FFFFE41F		LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9	LAFMH8	
H'FFFFE420	MC0[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE421	MC0[2]									
H'FFFFE422	MC0[3]									
H'FFFFE423	MC0[4]									
H'FFFFE424	MC0[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE425	MC0[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE426	MC0[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE427	MC0[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE428	MC1[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE429	MC1[2]									
H'FFFFE42A	MC1[3]									
H'FFFFE42B	MC1[4]									
H'FFFFE42C	MC1[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE42D	MC1[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE42E	MC1[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE42F	MC1[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE430	MC2[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE431	MC2[2]									
H'FFFFE432	MC2[3]									
H'FFFFE433	MC2[4]									
H'FFFFE434	MC2[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE435	MC2[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE436	MC2[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE437	MC2[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE438	MC3[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE439	MC3[2]									
H'FFFFE43A	MC3[3]									
H'FFFFE43B	MC3[4]									
H'FFFFE43C	MC3[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE43D	MC3[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE43E	MC3[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE43F	MC3[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE440	MC4[1]					DLC3	DLC2	DLC1	DLC0	HCAN (channel 0)
H'FFFFE441	MC4[2]									
H'FFFFE442	MC4[3]									
H'FFFFE443	MC4[4]									
H'FFFFE444	MC4[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE445	MC4[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE446	MC4[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE447	MC4[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE448	MC5[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE449	MC5[2]									
H'FFFFE44A	MC5[3]									
H'FFFFE44B	MC5[4]									
H'FFFFE44C	MC5[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE44D	MC5[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE44E	MC5[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE44F	MC5[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE450	MC6[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE451	MC6[2]									
H'FFFFE452	MC6[3]									
H'FFFFE453	MC6[4]									
H'FFFFE454	MC6[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE455	MC6[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE456	MC6[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE457	MC6[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE458	MC7[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE459	MC7[2]									
H'FFFFE45A	MC7[3]									
H'FFFFE45B	MC7[4]									
H'FFFFE45C	MC7[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE45D	MC7[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE45E	MC7[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE45F	MC7[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE460	MC8[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE461	MC8[2]									
H'FFFFE462	MC8[3]									
H'FFFFE463	MC8[4]									

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE464	MC8[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	HCAN (channel 0)
H'FFFFE465	MC8[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE466	MC8[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE467	MC8[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE468	MC9[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE469	MC9[2]									
H'FFFFE46A	MC9[3]									
H'FFFFE46B	MC9[4]									
H'FFFFE46C	MC9[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE46D	MC9[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE46E	MC9[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE46F	MC9[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE470	MC10[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE471	MC10[2]									
H'FFFFE472	MC10[3]									
H'FFFFE473	MC10[4]									
H'FFFFE474	MC10[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE475	MC10[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE476	MC10[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE477	MC10[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE478	MC11[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE479	MC11[2]									
H'FFFFE47A	MC11[3]									
H'FFFFE47B	MC11[4]									
H'FFFFE47C	MC11[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE47D	MC11[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE47E	MC11[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE47F	MC11[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE480	MC12[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE481	MC12[2]									
H'FFFFE482	MC12[3]									
H'FFFFE483	MC12[4]									
H'FFFFE484	MC12[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE485	MC12[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE486	MC12[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE487	MC12[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE488	MC13[1]					DLC3	DLC2	DLC1	DLC0	HCAN (channel 0)
H'FFFFE489	MC13[2]									
H'FFFFE48A	MC13[3]									
H'FFFFE48B	MC13[4]									
H'FFFFE48C	MC13[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE48D	MC13[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE48E	MC13[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE48F	MC13[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE490	MC14[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE491	MC14[2]									
H'FFFFE492	MC14[3]									
H'FFFFE493	MC14[4]									
H'FFFFE494	MC14[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE495	MC14[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE496	MC14[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE497	MC14[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE498	MC15[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE499	MC15[2]									
H'FFFFE49A	MC15[3]									
H'FFFFE49B	MC15[4]									
H'FFFFE49C	MC15[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE49D	MC15[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE49E	MC15[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE49F	MC15[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE4A0 to H'FFFFE4AF	—	—	—	—	—	—	—	—	—	
H'FFFFE4B0	MD0[1]	MSG_DATA_1								
H'FFFFE4B1	MD0[2]	MSG_DATA_2								
H'FFFFE4B2	MD0[3]	MSG_DATA_3								
H'FFFFE4B3	MD0[4]	MSG_DATA_4								
H'FFFFE4B4	MD0[5]	MSG_DATA_5								
H'FFFFE4B5	MD0[6]	MSG_DATA_6								
H'FFFFE4B6	MD0[7]	MSG_DATA_7								
H'FFFFE4B7	MD0[8]	MSG_DATA_8								

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE4B8	MD1[1]	MSG_DATA_1								HCAN (channel 0)
H'FFFFE4B9	MD1[2]	MSG_DATA_2								
H'FFFFE4BA	MD1[3]	MSG_DATA_3								
H'FFFFE4BB	MD1[4]	MSG_DATA_4								
H'FFFFE4BC	MD1[5]	MSG_DATA_5								
H'FFFFE4BD	MD1[6]	MSG_DATA_6								
H'FFFFE4BE	MD1[7]	MSG_DATA_7								
H'FFFFE4BF	MD1[8]	MSG_DATA_8								
H'FFFFE4C0	MD2[1]	MSG_DATA_1								
H'FFFFE4C1	MD2[2]	MSG_DATA_2								
H'FFFFE4C2	MD2[3]	MSG_DATA_3								
H'FFFFE4C3	MD2[4]	MSG_DATA_4								
H'FFFFE4C4	MD2[5]	MSG_DATA_5								
H'FFFFE4C5	MD2[6]	MSG_DATA_6								
H'FFFFE4C6	MD2[7]	MSG_DATA_7								
H'FFFFE4C7	MD2[8]	MSG_DATA_8								
H'FFFFE4C8	MD3[1]	MSG_DATA_1								
H'FFFFE4C9	MD3[2]	MSG_DATA_2								
H'FFFFE4CA	MD3[3]	MSG_DATA_3								
H'FFFFE4CB	MD3[4]	MSG_DATA_4								
H'FFFFE4CC	MD3[5]	MSG_DATA_5								
H'FFFFE4CD	MD3[6]	MSG_DATA_6								
H'FFFFE4CE	MD3[7]	MSG_DATA_7								
H'FFFFE4CF	MD3[8]	MSG_DATA_8								
H'FFFFE4D0	MD4[1]	MSG_DATA_1								
H'FFFFE4D1	MD4[2]	MSG_DATA_2								
H'FFFFE4D2	MD4[3]	MSG_DATA_3								
H'FFFFE4D3	MD4[4]	MSG_DATA_4								
H'FFFFE4D4	MD4[5]	MSG_DATA_5								
H'FFFFE4D5	MD4[6]	MSG_DATA_6								
H'FFFFE4D6	MD4[7]	MSG_DATA_7								
H'FFFFE4D7	MD4[8]	MSG_DATA_8								
H'FFFFE4D8	MD5[1]	MSG_DATA_1								
H'FFFFE4D9	MD5[2]	MSG_DATA_2								
H'FFFFE4DA	MD5[3]	MSG_DATA_3								

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE4DB	MD5[4]	MSG_DATA_4								HCAN (channel 0)
H'FFFFE4DC	MD5[5]	MSG_DATA_5								
H'FFFFE4DD	MD5[6]	MSG_DATA_6								
H'FFFFE4DE	MD5[7]	MSG_DATA_7								
H'FFFFE4DF	MD5[8]	MSG_DATA_8								
H'FFFFE4E0	MD6[1]	MSG_DATA_1								
H'FFFFE4E1	MD6[2]	MSG_DATA_2								
H'FFFFE4E2	MD6[3]	MSG_DATA_3								
H'FFFFE4E3	MD6[4]	MSG_DATA_4								
H'FFFFE4E4	MD6[5]	MSG_DATA_5								
H'FFFFE4E5	MD6[6]	MSG_DATA_6								
H'FFFFE4E6	MD6[7]	MSG_DATA_7								
H'FFFFE4E7	MD6[8]	MSG_DATA_8								
H'FFFFE4E8	MD7[1]	MSG_DATA_1								
H'FFFFE4E9	MD7[2]	MSG_DATA_2								
H'FFFFE4EA	MD7[3]	MSG_DATA_3								
H'FFFFE4EB	MD7[4]	MSG_DATA_4								
H'FFFFE4EC	MD7[5]	MSG_DATA_5								
H'FFFFE4ED	MD7[6]	MSG_DATA_6								
H'FFFFE4EE	MD7[7]	MSG_DATA_7								
H'FFFFE4EF	MD7[8]	MSG_DATA_8								
H'FFFFE4F0	MD8[1]	MSG_DATA_1								
H'FFFFE4F1	MD8[2]	MSG_DATA_2								
H'FFFFE4F2	MD8[3]	MSG_DATA_3								
H'FFFFE4F3	MD8[4]	MSG_DATA_4								
H'FFFFE4F4	MD8[5]	MSG_DATA_5								
H'FFFFE4F5	MD8[6]	MSG_DATA_6								
H'FFFFE4F6	MD8[7]	MSG_DATA_7								
H'FFFFE4F7	MD8[8]	MSG_DATA_8								
H'FFFFE4F8	MD9[1]	MSG_DATA_1								
H'FFFFE4F9	MD9[2]	MSG_DATA_2								
H'FFFFE4FA	MD9[3]	MSG_DATA_3								
H'FFFFE4FB	MD9[4]	MSG_DATA_4								
H'FFFFE4FC	MD9[5]	MSG_DATA_5								
H'FFFFE4FD	MD9[6]	MSG_DATA_6								

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE4FE	MD9[7]	MSG_DATA_7								HCAN (channel 0)
H'FFFFE4FF	MD9[8]	MSG_DATA_8								
H'FFFFE500	MD10[1]	MSG_DATA_1								
H'FFFFE501	MD10[2]	MSG_DATA_2								
H'FFFFE502	MD10[3]	MSG_DATA_3								
H'FFFFE503	MD10[4]	MSG_DATA_4								
H'FFFFE504	MD10[5]	MSG_DATA_5								
H'FFFFE505	MD10[6]	MSG_DATA_6								
H'FFFFE506	MD10[7]	MSG_DATA_7								
H'FFFFE507	MD10[8]	MSG_DATA_8								
H'FFFFE508	MD11[1]	MSG_DATA_1								
H'FFFFE509	MD11[2]	MSG_DATA_2								
H'FFFFE50A	MD11[3]	MSG_DATA_3								
H'FFFFE50B	MD11[4]	MSG_DATA_4								
H'FFFFE50C	MD11[5]	MSG_DATA_5								
H'FFFFE50D	MD11[6]	MSG_DATA_6								
H'FFFFE50E	MD11[7]	MSG_DATA_7								
H'FFFFE50F	MD11[8]	MSG_DATA_8								
H'FFFFE510	MD12[1]	MSG_DATA_1								
H'FFFFE511	MD12[2]	MSG_DATA_2								
H'FFFFE512	MD12[3]	MSG_DATA_3								
H'FFFFE513	MD12[4]	MSG_DATA_4								
H'FFFFE514	MD12[5]	MSG_DATA_5								
H'FFFFE515	MD12[6]	MSG_DATA_6								
H'FFFFE516	MD12[7]	MSG_DATA_7								
H'FFFFE517	MD12[8]	MSG_DATA_8								
H'FFFFE518	MD13[1]	MSG_DATA_1								
H'FFFFE519	MD13[2]	MSG_DATA_2								
H'FFFFE51A	MD13[3]	MSG_DATA_3								
H'FFFFE51B	MD13[4]	MSG_DATA_4								
H'FFFFE51C	MD13[5]	MSG_DATA_5								
H'FFFFE51D	MD13[6]	MSG_DATA_6								
H'FFFFE51E	MD13[7]	MSG_DATA_7								
H'FFFFE51F	MD13[8]	MSG_DATA_8								

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE520	MD14[1]	MSG_DATA_1								HCAN (channel 0)
H'FFFFE521	MD14[2]	MSG_DATA_2								
H'FFFFE522	MD14[3]	MSG_DATA_3								
H'FFFFE523	MD14[4]	MSG_DATA_4								
H'FFFFE524	MD14[5]	MSG_DATA_5								
H'FFFFE525	MD14[6]	MSG_DATA_6								
H'FFFFE526	MD14[7]	MSG_DATA_7								
H'FFFFE527	MD14[8]	MSG_DATA_8								
H'FFFFE528	MD15[1]	MSG_DATA_1								
H'FFFFE529	MD15[2]	MSG_DATA_2								
H'FFFFE52A	MD15[3]	MSG_DATA_3								
H'FFFFE52B	MD15[4]	MSG_DATA_4								
H'FFFFE52C	MD15[5]	MSG_DATA_5								
H'FFFFE52D	MD15[6]	MSG_DATA_6								
H'FFFFE52E	MD15[7]	MSG_DATA_7								
H'FFFFE52F	MD15[8]	MSG_DATA_8								
H'FFFFE530 to H'FFFFE5FF	—	—	—	—	—	—	—	—	—	—
H'FFFFE600	MCR	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	HCAN (channel 1)
H'FFFFE601	GSR	—	—	—	—	GSR3	GSR2	GSR1	GSR0	
H'FFFFE602	BCR	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0	
H'FFFFE603		BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8	
H'FFFFE604	MBCR	MBCR7	MBCR6	MBCR5	MBCR4	MBCR3	MBCR2	MBCR1	—	
H'FFFFE605		MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9	MBCR8	
H'FFFFE606	TXPR	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—	
H'FFFFE607		TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8	
H'FFFFE608	TXCR	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—	
H'FFFFE609		TXCR15	TXCR14	TCR13	TXCR12	TXCR11	TSCR10	TXCR9	TXCR8	
H'FFFFE60A	TXACK	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—	
H'FFFFE60B		TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8	
H'FFFFE60C	ABACK	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—	
H'FFFFE60D		ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8	
H'FFFFE60E	RXPR	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0	
H'FFFFE60F		RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE610	RFPR	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0	HCAN (channel 1)
H'FFFFE611		RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8	
H'FFFFE612	IRR	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
H'FFFFE613		—	—	—	IRR12	—	—	IRR9	IRR8	
H'FFFFE614	MBIMR	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0	
H'FFFFE615		MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8	
H'FFFFE616	IMR	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	—	
H'FFFFE617		—	—	—	IMR12	—	—	IMR9	IMR8	
H'FFFFE618	REC									
H'FFFFE619	TEC									
H'FFFFE61A	UMSR	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0	
H'FFFFE61B		UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8	
H'FFFFE61C	LAFML	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0	
H'FFFFE61D		LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8	
H'FFFFE61E	LAFMH	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1	LAFMH0	
H'FFFFE61F		LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9	LAFMH8	
H'FFFFE620	MC0[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE621	MC0[2]									
H'FFFFE622	MC0[3]									
H'FFFFE623	MC0[4]									
H'FFFFE624	MC0[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE625	MC0[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE626	MC0[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE627	MC0[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE628	MC1[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE629	MC1[2]									
H'FFFFE62A	MC1[3]									
H'FFFFE62B	MC1[4]									
H'FFFFE62C	MC1[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE62D	MC1[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE62E	MC1[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE62F	MC1[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE630	MC2[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE631	MC2[2]									
H'FFFFE632	MC2[3]									

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE633	MC2[4]									HCAN (channel 1)
H'FFFFE634	MC2[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE635	MC2[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE636	MC2[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE637	MC2[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE638	MC3[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE639	MC3[2]									
H'FFFFE63A	MC3[3]									
H'FFFFE63B	MC3[4]									
H'FFFFE63C	MC3[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE63D	MC3[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE63E	MC3[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE63F	MC3[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE640	MC4[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE641	MC4[2]									
H'FFFFE642	MC4[3]									
H'FFFFE643	MC4[4]									
H'FFFFE644	MC4[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE645	MC4[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE646	MC4[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE647	MC4[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE648	MC5[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE649	MC5[2]									
H'FFFFE64A	MC5[3]									
H'FFFFE64B	MC5[4]									
H'FFFFE64C	MC5[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE64D	MC5[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE64E	MC5[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE64F	MC5[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE650	MC6[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE651	MC6[2]									
H'FFFFE652	MC6[3]									
H'FFFFE653	MC6[4]									
H'FFFFE654	MC6[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE655	MC6[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE656	MC6[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	HCAN (channel 1)
H'FFFFE657	MC6[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE658	MC7[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE659	MC7[2]									
H'FFFFE65A	MC7[3]									
H'FFFFE65B	MC7[4]									
H'FFFFE65C	MC7[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE65D	MC7[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE65E	MC7[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE65F	MC7[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE660	MC8[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE661	MC8[2]									
H'FFFFE662	MC8[3]									
H'FFFFE663	MC8[4]									
H'FFFFE664	MC8[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE665	MC8[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE666	MC8[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE667	MC8[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE668	MC9[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE669	MC9[2]									
H'FFFFE66A	MC9[3]									
H'FFFFE66B	MC9[4]									
H'FFFFE66C	MC9[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE66D	MC9[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE66E	MC9[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE66F	MC9[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE670	MC10[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE671	MC10[2]									
H'FFFFE672	MC10[3]									
H'FFFFE673	MC10[4]									
H'FFFFE674	MC10[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE675	MC10[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE676	MC10[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE677	MC10[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE678	MC11[1]					DLC3	DLC2	DLC1	DLC0	HCAN (channel 1)
H'FFFFE679	MC11[2]									
H'FFFFE67A	MC11[3]									
H'FFFFE67B	MC11[4]									
H'FFFFE67C	MC11[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE67D	MC11[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE67E	MC11[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE67F	MC11[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE680	MC12[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE681	MC12[2]									
H'FFFFE682	MC12[3]									
H'FFFFE683	MC12[4]									
H'FFFFE684	MC12[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE685	MC12[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE686	MC12[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE687	MC12[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE688	MC13[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE689	MC13[2]									
H'FFFFE68A	MC13[3]									
H'FFFFE68B	MC13[4]									
H'FFFFE68C	MC13[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE68D	MC13[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE68E	MC13[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE68F	MC13[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE690	MC14[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE691	MC14[2]									
H'FFFFE692	MC14[3]									
H'FFFFE693	MC14[4]									
H'FFFFE694	MC14[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE695	MC14[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE696	MC14[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE697	MC14[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE698	MC15[1]					DLC3	DLC2	DLC1	DLC0	
H'FFFFE699	MC15[2]									
H'FFFFE69A	MC15[3]									

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE69B	MC15[4]									HCAN (channel 1)
H'FFFFE69C	MC15[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE		EXD_ID17	EXD_ID16	
H'FFFFE69D	MC15[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3	
H'FFFFE69E	MC15[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0	
H'FFFFE69F	MC15[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8	
H'FFFFE6A0 to H'FFFFE6AF	—	—	—	—	—	—	—	—	—	
H'FFFFE6B0	MD0[1]	MSG_DATA_1								
H'FFFFE6B1	MD0[2]	MSG_DATA_2								
H'FFFFE6B2	MD0[3]	MSG_DATA_3								
H'FFFFE6B3	MD0[4]	MSG_DATA_4								
H'FFFFE6B4	MD0[5]	MSG_DATA_5								
H'FFFFE6B5	MD0[6]	MSG_DATA_6								
H'FFFFE6B6	MD0[7]	MSG_DATA_7								
H'FFFFE6B7	MD0[8]	MSG_DATA_8								
H'FFFFE6B8	MD1[1]	MSG_DATA_1								
H'FFFFE6B9	MD1[2]	MSG_DATA_2								
H'FFFFE6BA	MD1[3]	MSG_DATA_3								
H'FFFFE6BB	MD1[4]	MSG_DATA_4								
H'FFFFE6BC	MD1[5]	MSG_DATA_5								
H'FFFFE6BD	MD1[6]	MSG_DATA_6								
H'FFFFE6BE	MD1[7]	MSG_DATA_7								
H'FFFFE6BF	MD1[8]	MSG_DATA_8								
H'FFFFE6C0	MD2[1]	MSG_DATA_1								
H'FFFFE6C1	MD2[2]	MSG_DATA_2								
H'FFFFE6C2	MD2[3]	MSG_DATA_3								
H'FFFFE6C3	MD2[4]	MSG_DATA_4								
H'FFFFE6C4	MD2[5]	MSG_DATA_5								
H'FFFFE6C5	MD2[6]	MSG_DATA_6								
H'FFFFE6C6	MD2[7]	MSG_DATA_7								
H'FFFFE6C7	MD2[8]	MSG_DATA_8								
H'FFFFE6C8	MD3[1]	MSG_DATA_1								
H'FFFFE6C9	MD3[2]	MSG_DATA_2								
H'FFFFE6CA	MD3[3]	MSG_DATA_3								
H'FFFFE6CB	MD3[4]	MSG_DATA_4								

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE6CC	MD3[5]	MSG_DATA_5								HCAN (channel 1)
H'FFFFE6CD	MD3[6]	MSG_DATA_6								
H'FFFFE6CE	MD3[7]	MSG_DATA_7								
H'FFFFE6CF	MD3[8]	MSG_DATA_8								
H'FFFFE6D0	MD4[1]	MSG_DATA_1								
H'FFFFE6D1	MD4[2]	MSG_DATA_2								
H'FFFFE6D2	MD4[3]	MSG_DATA_3								
H'FFFFE6D3	MD4[4]	MSG_DATA_4								
H'FFFFE6D4	MD4[5]	MSG_DATA_5								
H'FFFFE6D5	MD4[6]	MSG_DATA_6								
H'FFFFE6D6	MD4[7]	MSG_DATA_7								
H'FFFFE6D7	MD4[8]	MSG_DATA_8								
H'FFFFE6D8	MD5[1]	MSG_DATA_1								
H'FFFFE6D9	MD5[2]	MSG_DATA_2								
H'FFFFE6DA	MD5[3]	MSG_DATA_3								
H'FFFFE6DB	MD5[4]	MSG_DATA_4								
H'FFFFE6DC	MD5[5]	MSG_DATA_5								
H'FFFFE6DD	MD5[6]	MSG_DATA_6								
H'FFFFE6DE	MD5[7]	MSG_DATA_7								
H'FFFFE6DF	MD5[8]	MSG_DATA_8								
H'FFFFE6E0	MD6[1]	MSG_DATA_1								
H'FFFFE6E1	MD6[2]	MSG_DATA_2								
H'FFFFE6E2	MD6[3]	MSG_DATA_3								
H'FFFFE6E3	MD6[4]	MSG_DATA_4								
H'FFFFE6E4	MD6[5]	MSG_DATA_5								
H'FFFFE6E5	MD6[6]	MSG_DATA_6								
H'FFFFE6E6	MD6[7]	MSG_DATA_7								
H'FFFFE6E7	MD6[8]	MSG_DATA_8								
H'FFFFE6E8	MD7[1]	MSG_DATA_1								
H'FFFFE6E9	MD7[2]	MSG_DATA_2								
H'FFFFE6EA	MD7[3]	MSG_DATA_3								
H'FFFFE6EB	MD7[4]	MSG_DATA_4								
H'FFFFE6EC	MD7[5]	MSG_DATA_5								
H'FFFFE6ED	MD7[6]	MSG_DATA_6								
H'FFFFE6EE	MD7[7]	MSG_DATA_7								

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE6EF	MD7[8]	MSG_DATA_8								HCAN (channel 1)
H'FFFFE6F0	MD8[1]	MSG_DATA_1								
H'FFFFE6F1	MD8[2]	MSG_DATA_2								
H'FFFFE6F2	MD8[3]	MSG_DATA_3								
H'FFFFE6F3	MD8[4]	MSG_DATA_4								
H'FFFFE6F4	MD8[5]	MSG_DATA_5								
H'FFFFE6F5	MD8[6]	MSG_DATA_6								
H'FFFFE6F6	MD8[7]	MSG_DATA_7								
H'FFFFE6F7	MD8[8]	MSG_DATA_8								
H'FFFFE6F8	MD9[1]	MSG_DATA_1								
H'FFFFE6F9	MD9[2]	MSG_DATA_2								
H'FFFFE6FA	MD9[3]	MSG_DATA_3								
H'FFFFE6FB	MD9[4]	MSG_DATA_4								
H'FFFFE6FC	MD9[5]	MSG_DATA_5								
H'FFFFE6FD	MD9[6]	MSG_DATA_6								
H'FFFFE6FE	MD9[7]	MSG_DATA_7								
H'FFFFE6FF	MD9[8]	MSG_DATA_8								
H'FFFFE700	MD10[1]	MSG_DATA_1								
H'FFFFE701	MD10[2]	MSG_DATA_2								
H'FFFFE702	MD10[3]	MSG_DATA_3								
H'FFFFE703	MD10[4]	MSG_DATA_4								
H'FFFFE704	MD10[5]	MSG_DATA_5								
H'FFFFE705	MD10[6]	MSG_DATA_6								
H'FFFFE706	MD10[7]	MSG_DATA_7								
H'FFFFE707	MD10[8]	MSG_DATA_8								
H'FFFFE708	MD11[1]	MSG_DATA_1								
H'FFFFE709	MD11[2]	MSG_DATA_2								
H'FFFFE70A	MD11[3]	MSG_DATA_3								
H'FFFFE70B	MD11[4]	MSG_DATA_4								
H'FFFFE70C	MD11[5]	MSG_DATA_5								
H'FFFFE70D	MD11[6]	MSG_DATA_6								
H'FFFFE70E	MD11[7]	MSG_DATA_7								
H'FFFFE70F	MD11[8]	MSG_DATA_8								
H'FFFFE710	MD12[1]	MSG_DATA_1								
H'FFFFE711	MD12[2]	MSG_DATA_2								

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE712	MD12[3]	MSG_DATA_3								HCAN (channel 1)
H'FFFFE713	MD12[4]	MSG_DATA_4								
H'FFFFE714	MD12[5]	MSG_DATA_5								
H'FFFFE715	MD12[6]	MSG_DATA_6								
H'FFFFE716	MD12[7]	MSG_DATA_7								
H'FFFFE717	MD12[8]	MSG_DATA_8								
H'FFFFE718	MD13[1]	MSG_DATA_1								
H'FFFFE719	MD13[2]	MSG_DATA_2								
H'FFFFE71A	MD13[3]	MSG_DATA_3								
H'FFFFE71B	MD13[4]	MSG_DATA_4								
H'FFFFE71C	MD13[5]	MSG_DATA_5								
H'FFFFE71D	MD13[6]	MSG_DATA_6								
H'FFFFE71E	MD13[7]	MSG_DATA_7								
H'FFFFE71F	MD13[8]	MSG_DATA_8								
H'FFFFE720	MD14[1]	MSG_DATA_1								
H'FFFFE721	MD14[2]	MSG_DATA_2								
H'FFFFE722	MD14[3]	MSG_DATA_3								
H'FFFFE723	MD14[4]	MSG_DATA_4								
H'FFFFE724	MD14[5]	MSG_DATA_5								
H'FFFFE725	MD14[6]	MSG_DATA_6								
H'FFFFE726	MD14[7]	MSG_DATA_7								
H'FFFFE727	MD14[8]	MSG_DATA_8								
H'FFFFE728	MD15[1]	MSG_DATA_1								
H'FFFFE729	MD15[2]	MSG_DATA_2								
H'FFFFE72A	MD15[3]	MSG_DATA_3								
H'FFFFE72B	MD15[4]	MSG_DATA_4								
H'FFFFE72C	MD15[5]	MSG_DATA_5								
H'FFFFE72D	MD15[6]	MSG_DATA_6								
H'FFFFE72E	MD15[7]	MSG_DATA_7								
H'FFFFE72F	MD15[8]	MSG_DATA_8								
H'FFFFE730 to H'FFFFE7FF	—	—	—	—	—	—	—	—	—	—

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFE800	FCCS	FWE	—	—	FLER	—	—	—	SCO	FLASH
H'FFFFE801	FPCS	—	—	—	—	—	—	—	PPVS	
H'FFFFE802	FECS	—	—	—	—	—	—	—	EPVB	
H'FFFFE803	—	System area. Do not access this area.								
H'FFFFE804	FKEY	K7	K6	K5	K4	K3	K2	K1	K0	
H'FFFFE805	FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
H'FFFFE806	FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0	
H'FFFFE807 to H'FFFFEBFF	—	System area. Do not access this area.								
H'FFFFEC00	UBARH	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24	UBC
H'FFFFEC01		UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16	
H'FFFFEC02	UBARL	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8	
H'FFFFEC03		UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0	
H'FFFFEC04	UBAMRH	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24	
H'FFFFEC05		UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16	
H'FFFFEC06	UBAMRL	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8	
H'FFFFEC07		UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0	
H'FFFFEC08	UBBR	—	—	—	—	—	—	—	—	
H'FFFFEC09		CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0	
H'FFFFEC0A	UBCR	—	—	—	—	—	—	—	—	
H'FFFFEC0B		—	—	—	—	—	CKS1	CKS0	UBID	
H'FFFFEC0C to H'FFFFEC0F	—	—	—	—	—	—	—	—	—	—
H'FFFFEC10	TCSR *	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	WDT
H'FFFFEC11	TCNT *									
H'FFFFEC12	—	—	—	—	—	—	—	—	—	
H'FFFFEC13	RSTCSR *	WOVF	RSTE	RSTS	—	—	—	—	—	
H'FFFFEC14	SBYCR	SSBY	HIZ	—	—	—	—	—	—	Power-down state
H'FFFFEC15 to H'FFFFEC1F	—	—	—	—	—	—	—	—	—	—

Note: * This is the read address. The write address is H'FFFEC10 for TCSR and TCNT, and H'FFFEC12 for RSTCSR. For details, see section 13.2.4, Register Access.

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFEC20	BCR1	—	—	—	—	—	—	—	—	BSC
H'FFFFEC21		—	—	—	—	A3SZ	A2SZ	A1SZ	A0SZ	
H'FFFFEC22	BCR2	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00	
H'FFFFEC23		CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0	
H'FFFFEC24	WCR	W33	W32	W31	W30	W23	W22	W21	W20	
H'FFFFEC25		W13	W12	W11	W10	W03	W02	W01	W00	
H'FFFFEC26	RAMER	—	—	—	—	—	—	—	—	
H'FFFFEC27		—	—	—	—	RAMS	RAM2	RAM1	RAM0	
H'FFFFEC28 to H'FFFFECAF	—	—	—	—	—	—	—	—	—	—
H'FFFFECB0	DMAOR	—	—	—	—	—	—	—	—	DMAC (all channels)
H'FFFFECB1		—	—	—	—	—	AE	NMIF	DME	
H'FFFFECB2 to H'FFFFECBF	—	—	—	—	—	—	—	—	—	—
H'FFFFECC0	SAR0									DMAC (channel 0)
H'FFFFECC1										
H'FFFFECC2										
H'FFFFECC3										
H'FFFFECC4	DAR0									
H'FFFFECC5										
H'FFFFECC6										
H'FFFFECC7										
H'FFFFECC8	DMATCR0	—	—	—	—	—	—	—	—	
H'FFFFECC9										
H'FFFFECCA										
H'FFFFECCB										
H'FFFFECCC	CHCR0	—	—	—	—	—	—	—	—	
H'FFFFECCD		—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFFECCE		—	—	SM1	SM0	—	—	DM1	DM0	
H'FFFFECCF		—	—	TS1	TS0	TM	IE	TE	DE	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFECD0	SAR1									DMAC (channel 1)
H'FFFFECD1										
H'FFFFECD2										
H'FFFFECD3										
H'FFFFECD4	DAR1									
H'FFFFECD5										
H'FFFFECD6										
H'FFFFECD7										
H'FFFFECD8	DMATCR1	—	—	—	—	—	—	—	—	
H'FFFFECD9										
H'FFFFECDA										
H'FFFFECDB										
H'FFFFECDC	CHCR1	—	—	—	—	—	—	—	—	
H'FFFFECDD		—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFFECDE		—	—	SM1	SM0	—	—	DM1	DM0	
H'FFFFECDF		—	—	TS1	TS0	TM	IE	TE	DE	
H'FFFFECE0	SAR2									DMAC (channel 2)
H'FFFFECE1										
H'FFFFECE2										
H'FFFFECE3										
H'FFFFECE4	DAR2									
H'FFFFECE5										
H'FFFFECE6										
H'FFFFECE7										
H'FFFFECE8	DMATCR2	—	—	—	—	—	—	—	—	
H'FFFFECE9										
H'FFFFECEA										
H'FFFFECEB										
H'FFFFECEC	CHCR2	—	—	—	—	—	—	—	RO	
H'FFFFECED		—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFFECEE		—	—	SM1	SM0	—	—	DM1	DM0	
H'FFFFECEF		—	—	TS1	TS0	TM	IE	TE	DE	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFECF0	SAR3									DMAC (channel 3)
H'FFFFECF1										
H'FFFFECF2										
H'FFFFECF3										
H'FFFFECF4	DAR3									
H'FFFFECF5										
H'FFFFECF6										
H'FFFFECF7										
H'FFFFECF8	DMATCR3	—	—	—	—	—	—	—	—	
H'FFFFECF9										
H'FFFFECFA										
H'FFFFECFB										
H'FFFFECFC	CHCR3	—	—	—	DI	—	—	—	—	
H'FFFFECFD		—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFFECFE		—	—	SM1	SM0	—	—	DM1	DM0	
H'FFFFECFF		—	—	TS1	TS0	TM	IE	TE	DE	
H'FFFFED00	IPRA									INTC
H'FFFFED01										
H'FFFFED02	IPRB									
H'FFFFED03										
H'FFFFED04	IPRC									
H'FFFFED05										
H'FFFFED06	IPRD									
H'FFFFED07										
H'FFFFED08	IPRE									
H'FFFFED09										
H'FFFFED0A	IPRF									
H'FFFFED0B										
H'FFFFED0C	IPRG									
H'FFFFED0D										
H'FFFFED0E	IPRH									
H'FFFFED0F										
H'FFFFED10	IPRI									
H'FFFFED11										
H'FFFFED12	IPRJ									
H'FFFFED13										

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFED14	IPRK									INTC
H'FFFFED15										
H'FFFFED16	IPRL									
H'FFFFED17										
H'FFFFED18	ICR	NMIL	—	—	—	—	—	—	NMIE	
H'FFFFED19		IRQ0S	IRQ1S	IRQ2S	IRQ3S	IRQ4S	IRQ5S	IRQ6S	IRQ7S	
H'FFFFED1A	ISR	—	—	—	—	—	—	—	—	
H'FFFFED1B		IRQ0F	IRQ1F	IRQ2F	IRQ3F	IRQ4F	IRQ5F	IRQ6F	IRQ7F	
H'FFFFED1C to H'FFFFEFFF	—	—	—	—	—	—	—	—	—	—
H'FFFFF000	SMR0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 0)
H'FFFFF001	BRR0									
H'FFFFF002	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFFF003	TDR0									
H'FFFFF004	SSR0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFFF005	RDR0									
H'FFFFF006	SDCR0	—	—	—	—	DIR	—	—	—	
H'FFFFF007	—	—	—	—	—	—	—	—	—	
H'FFFFF008	SMR1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 1)
H'FFFFF009	BRR1									
H'FFFFF00A	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFFF00B	TDR1									
H'FFFFF00C	SSR1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFFF00D	RDR1									
H'FFFFF00E	SDCR1	—	—	—	—	DIR	—	—	—	
H'FFFFF00F	—	—	—	—	—	—	—	—	—	
H'FFFFF010	SMR2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 2)
H'FFFFF011	BRR2									
H'FFFFF012	SCR2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFFF013	TDR2									
H'FFFFF014	SSR2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFFF015	RDR2									
H'FFFFF016	SDCR2	—	—	—	—	DIR	—	—	—	
H'FFFFF017	—	—	—	—	—	—	—	—	—	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF018	SMR3	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 3)
H'FFFFF019	BRR3									
H'FFFFF01A	SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFFF01B	TDR3									
H'FFFFF01C	SSR3	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFFF01D	RDR3									
H'FFFFF01E	SDCR3	—	—	—	—	DIR	—	—	—	
H'FFFFF01F	—	—	—	—	—	—	—	—	—	
H'FFFFF020	SMR4	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI (channel 4)
H'FFFFF021	BRR4									
H'FFFFF022	SCR4	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFFF023	TDR4									
H'FFFFF024	SSR4	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFFF025	RDR4									
H'FFFFF026	SDCR4	—	—	—	—	DIR	—	—	—	
H'FFFFF027 to H'FFFFF3FF	—	—	—	—	—	—	—	—	—	—
H'FFFFF400	TSTR2	STR7D	STR7C	STR7B	STR7A	STR6D	STR6C	STR6B	STR6A	ATU-II (all channels)
H'FFFFF401	TSTR1	STR10	STR5	STR4	STR3	STR1B,2B	STR2A	STR1A	STR0	
H'FFFFF402	TSTR3	—	—	—	—	—	—	—	STR11	
H'FFFFF403	—	—	—	—	—	—	—	—	—	
H'FFFFF404	PSCR1	—	—	—	PSC1E	PSC1D	PSC1C	PSC1B	PSC1A	
H'FFFFF405	—	—	—	—	—	—	—	—	—	
H'FFFFF406	PSCR2				PSC2E	PSC2D	PSC2C	PSC2B	PSC2A	
H'FFFFF407	—	—	—	—	—	—	—	—	—	
H'FFFFF408	PSCR3	—	—	—	PSC3E	PSC3D	PSC3C	PSC3B	PSC3A	
H'FFFFF409	—	—	—	—	—	—	—	—	—	
H'FFFFF40A	PSCR4	—	—	—	PSC4E	PSC4D	PSC4C	PSC4B	PSC4A	
H'FFFFF40B	—	—	—	—	—	—	—	—	—	
H'FFFFF40C to H'FFFFF41F	—	—	—	—	—	—	—	—	—	—
H'FFFFF420	ICR0DH									ATU-II (channel 0)
H'FFFFF421										
H'FFFFF422	ICR0DL									
H'FFFFF423										

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF424	ITVRR1	ITVA9	ITVA8	ITVA7	ITVA6	ITVE9	ITVE8	ITVE7	TIVE6	ATU-II (channel 1)
H'FFFFF425	—	—	—	—	—	—	—	—	—	
H'FFFFF426	ITVRR2A	ITVA13A	ITVA12A	ITVA11A	ITVA10A	ITVE13A	ITVE12A	ITVE11A	ITVE10A	ATU-II (channel 2)
H'FFFFF427	—	—	—	—	—	—	—	—	—	
H'FFFFF428	ITVRR2B	ITVA13B	ITVA12B	ITVA11B	ITVA10B	ITVE13B	ITVE12B	ITVE11B	ITVE10B	
H'FFFFF429	—	—	—	—	—	—	—	—	—	
H'FFFFF42A	TIOR0	IO0D1	IO0D0	IO0C1	IO0C0	IO0B1	IO0B0	IO0A1	IO0A0	ATU-II (channel 0)
H'FFFFF42B	—	—	—	—	—	—	—	—	—	
H'FFFFF42C	TSR0	—	—	—	—	—	—	—	—	
H'FFFFF42D		IIF2B	IIF2A	IIF1	OVF0	ICF0D	ICF0C	ICF0B	ICF0A	
H'FFFFF42E	TIER0	—	—	—	—	—	—	—	—	
H'FFFFF42F		—	—	—	OVE0	ICE0D	ICE0C	ICE0B	ICE0A	
H'FFFFF430	TCNT0H									
H'FFFFF431										
H'FFFFF432	TCNT0L									
H'FFFFF433										
H'FFFFF434	ICR0AH									
H'FFFFF435										
H'FFFFF436	ICR0AL									
H'FFFFF437										
H'FFFFF438	ICR0BH									
H'FFFFF439										
H'FFFFF43A	ICR0BL									
H'FFFFF43B										
H'FFFFF43C	ICR0CH									
H'FFFFF43D										
H'FFFFF43E	ICR0CL									
H'FFFFF43F										

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF440	TCNT1A									ATU-II (channel 1)
H'FFFFF441										
H'FFFFF442	TCNT1B									
H'FFFFF443										
H'FFFFF444	GR1A									
H'FFFFF445										
H'FFFFF446	GR1B									
H'FFFFF447										
H'FFFFF448	GR1C									
H'FFFFF449										
H'FFFFF44A	GR1D									
H'FFFFF44B										
H'FFFFF44C	GR1E									
H'FFFFF44D										
H'FFFFF44E	GR1F									
H'FFFFF44F										
H'FFFFF450	GR1G									
H'FFFFF451										
H'FFFFF452	GR1H									
H'FFFFF453										
H'FFFFF454	OCR1									
H'FFFFF455										
H'FFFFF456	OSBR1									
H'FFFFF457										
H'FFFFF458	TIOR1B	—	IO1D2	IO1D1	IO1D0	—	IO1C2	IO1C1	IO1C0	
H'FFFFF459	TIOR1A	—	IO1B2	IO1B1	IO1B0	—	IO1A2	IO1A1	IO1A0	
H'FFFFF45A	TIOR1D	—	IO1H2	IO1H1	IO1H0	—	IO1G2	IO1G1	IO1G0	
H'FFFFF45B	TIOR1C	—	IO1F2	IO1F1	IO1F0	—	IO1E2	IO1E1	IO1E0	
H'FFFFF45C	TCR1B	—	—	CKEGB1	CKEGB0	CKSELB3	CKSELB2	CKSELB1	CKSELB0	
H'FFFFF45D	TCR1A	—	—	CKEGA1	CKEGA0	CKSELA3	CKSELA2	CKSELA1	CKSELA0	
H'FFFFF45E	TSR1A	—	—	—	—	—	—	—	OVF1A	
H'FFFFF45F		IMF1H	IMF1G	IMF1F	IMF1E	IMF1D	IMF1C	IMF1B	IMF1A	
H'FFFFF460	TSR1B	—	—	—	—	—	—	—	OVF1B	
H'FFFFF461		—	—	—	—	—	—	—	CMF1	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF462	TIER1A	—	—	—	—	—	—	—	OVE1A	ATU-II (channel 1)
H'FFFFF463		IME1H	IME1G	IME1F	IME1E	IME1D	IME1C	IME1B	IME1A	
H'FFFFF464	TIER1B	—	—	—	—	—	—	—	OVE1B	
H'FFFFF465		—	—	—	—	—	—	—	CME1	
H'FFFFF466	TRGMDR	TRGMD	—	—	—	—	—	—	—	
H'FFFFF467 to H'FFFFF47F	—	—	—	—	—	—	—	—	—	—
H'FFFFF480	TSR3	—	OVF5	IMF5D	IMF5C	IMF5B	IMF5A	OVF4	IMF4D	ATU-II (channels 3 to 5)
H'FFFFF481		IMF4C	IMF4B	IMF4A	OVF3	IMF3D	IMF3C	IMF3B	IMF3A	
H'FFFFF482	TIER3	—	OVE5	IME5D	IME5C	IME5B	IME5A	OVE4	IME4D	
H'FFFFF483		IME4C	IME4B	IME4A	OVE3	IME3D	IME3C	IME3B	IME3A	
H'FFFFF484	TMDR	—	—	—	—	—	T5PWM	T4PWM	T3PWM	
H'FFFFF485 to H'FFFFF49F	—	—	—	—	—	—	—	—	—	—
H'FFFFF4A0	TCNT3									ATU-II (channel 3)
H'FFFFF4A1										
H'FFFFF4A2	TGR3A									
H'FFFFF4A3										
H'FFFFF4A4	GR3B									
H'FFFFF4A5										
H'FFFFF4A6	GR3C									
H'FFFFF4A7										
H'FFFFF4A8	GR3D									
H'FFFFF4A9										
H'FFFFF4AA	TIOR3B	CCI3D	IO3D2	IO3D1	IO3D0	CCI3C	IO3C2	IO3C1	IO3C0	
H'FFFFF4AB	TIOR3A	CCI3B	IO3B2	IO3B1	IO3B0	CCI3A	IO3A2	IO3A1	IO3A0	
H'FFFFF4AC	TCR3	—	—	CKEG1	CKEG0	CKSEL3	CKSEL2	CKSEL1	CKSEL0	
H'FFFFF4AD to H'FFFFF4BF	—	—	—	—	—	—	—	—	—	—

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF4C0	TCNT4									ATU-II (channel 4)
H'FFFFF4C1										
H'FFFFF4C2	GR4A									
H'FFFFF4C3										
H'FFFFF4C4	GR4B									
H'FFFFF4C5										
H'FFFFF4C6	GR4C									
H'FFFFF4C7										
H'FFFFF4C8	GR4D									
H'FFFFF4C9										
H'FFFFF4CA	TIOR4B	CCI4D	IO4D2	IO4D1	IO4D0	CCI4C	IO4C2	IO4C1	IO4C0	
H'FFFFF4CB	TIOR4A	CCI4B	IO4B2	IO4B1	IO4B0	CCI4A	IO4A2	IO4A1	IO4A0	
H'FFFFF4CC	TCR4	—	—	CKEG1	CKEG0	CKSEL3	CKSEL2	CKSEL1	CKSEL0	
H'FFFFF4CD to H'FFFFF4DF	—	—	—	—	—	—	—	—	—	—
H'FFFFF4E0	TCNT5									ATU-II (channel 5)
H'FFFFF4E1										
H'FFFFF4E2	GR5A									
H'FFFFF4E3										
H'FFFFF4E4	GR5B									
H'FFFFF4E5										
H'FFFFF4E6	GR5C									
H'FFFFF4E7										
H'FFFFF4E8	GR5D									
H'FFFFF4E9										
H'FFFFF4EA	TIOR5B	CCI5D	IO5D2	IO5D1	IO5D0	CCI5C	IO5C2	IO5C1	IO5C0	
H'FFFFF4EB	TIOR5A	CCI5B	IO5B2	IO5B1	IO5B0	CCI5A	IO5A2	IO5A1	IO5A0	
H'FFFFF4EC	TCR5	—	—	CKEG1	CKEG0	CKSEL3	CKSEL2	CKSEL1	CKSEL0	
H'FFFFF4ED to H'FFFFF4EF	—	—	—	—	—	—	—	—	—	—

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF500	TCNT6A									ATU-II (channel 6)
H'FFFFF501										
H'FFFFF502	TCNT6B									
H'FFFFF503										
H'FFFFF504	TCNT6C									
H'FFFFF505										
H'FFFFF506	TCNT6D									
H'FFFFF507										
H'FFFFF508	CYLR6A									
H'FFFFF509										
H'FFFFF50A	CYLR6B									
H'FFFFF50B										
H'FFFFF50C	CYLR6C									
H'FFFFF50D										
H'FFFFF50E	CYLR6D									
H'FFFFF50F										
H'FFFFF510	BFR6A									
H'FFFFF511										
H'FFFFF512	BFR6B									
H'FFFFF513										
H'FFFFF514	BFR6C									
H'FFFFF515										
H'FFFFF516	BFR6D									
H'FFFFF517										
H'FFFFF518	DTR6A									
H'FFFFF519										
H'FFFFF51A	DTR6B									
H'FFFFF51B										
H'FFFFF51C	DTR6C									
H'FFFFF51D										

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF51E	DTR6D									ATU-II (channel 6)
H'FFFFF51F										
H'FFFFF520	TCR6B	—	CKSELD2	CKSELD1	CKSELD0	—	CKSELC2	CKSELC1	CKSELC0	
H'FFFFF521	TCR6A	—	CKSELB2	CKSELB1	CKSELB0	—	CKSELA2	CKSELA1	CKSELA0	
H'FFFFF522	TSR6	—	—	—	—	—	—	—	—	
H'FFFFF523		UD6D	UD6C	UD6B	UD6A	CMF6D	CMF6C	CMF6B	CMF6A	
H'FFFFF524	TIER6	—	—	—	—	—	—	—	—	
H'FFFFF525		—	—	—	—	CME6D	CME6C	CME6B	CME6A	
H'FFFFF526	PMDR6	DTSELD	DTSELC	DTSELB	DTSELA	CNTSELD	CNTSELC	CNTSELB	CNTSELA	
H'FFFFF527 to H'FFFFF57F	—	—	—	—	—	—	—	—	—	—
H'FFFFF580	TCNT7A									ATU-II (channel 7)
H'FFFFF581										
H'FFFFF582	TCNT7B									
H'FFFFF583										
H'FFFFF584	TCNT7C									
H'FFFFF585										
H'FFFFF586	TCNT7D									
H'FFFFF587										
H'FFFFF588	CYLR7A									
H'FFFFF589										
H'FFFFF58A	CYLR7B									
H'FFFFF58B										
H'FFFFF58C	CYLR7C									
H'FFFFF58D										
H'FFFFF58E	CYLR7D									
H'FFFFF58F										
H'FFFFF590	BFR7A									
H'FFFFF591										
H'FFFFF592	BFR7B									
H'FFFFF593										
H'FFFFF594	BFR7C									
H'FFFFF595										
H'FFFFF596	BFR7D									
H'FFFFF597										

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF598	DTR7A									ATU-II
H'FFFFF599										(channel 7)
H'FFFFF59A	DTR7B									
H'FFFFF59B										
H'FFFFF59C	DTR7C									
H'FFFFF59D										
H'FFFFF59E	DTR7D									
H'FFFFF59F										
H'FFFFF5A0	TCR7B	—	CKSELD2	CKSELD1	CKSELD0	—	CKSELC2	CKSELC1	CKSELC0	
H'FFFFF5A1	TCR7A	—	CKSELB2	CKSELB1	CKSELB0	—	CKSELA2	CKSELA1	CKSELA0	
H'FFFFF5A2	TSR7	—	—	—	—	—	—	—	—	
H'FFFFF5A3		—	—	—	—	CMF7D	CMF7C	C<F7B	CMF7A	
H'FFFFF5A4	TIER7	—	—	—	—	—	—	—	—	
H'FFFFF5A5		—	—	—	—	CME7D	CME7C	CME7B	CME7A	
H'FFFFF5A6 to H'FFFFF5BF	—	—	—	—	—	—	—	—	—	—
H'FFFFF5C0	TCNT11									ATU-II
H'FFFFF5C1										(channel 11)
H'FFFFF5C2	GR11A									
H'FFFFF5C3										
H'FFFFF5C4	GR11B									
H'FFFFF5C5										
H'FFFFF5C6	TIOR11	—	IO11B2	IO11B1	IO11B0	—	IO11A2	IO11A1	IO11A0	
H'FFFFF5C7	—	—	—	—	—	—	—	—	—	
H'FFFFF5C8	TCR11	—	—	CKEG1	CKEG0	—	CKSELA2	CKSELA1	CKSELA0	
H'FFFFF5C9	—	—	—	—	—	—	—	—	—	
H'FFFFF5CA	TSR11	—	—	—	—	—	—	—	OVF11	
H'FFFFF5CB		—	—	—	—	—	—	IMF11B	IMF11A	
H'FFFFF5CC	TIER11	—	—	—	—	—	—	—	OVE11	
H'FFFFF5CD		—	—	—	—	—	—	IME11B	IME11A	
H'FFFFF5CE to H'FFFFF5FF	—	—	—	—	—	—	—	—	—	—
H'FFFFF600	TCNT2A									ATU-II
H'FFFFF601										(channel 2)

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF602	TCNT2B									ATU-II (channel 2)
H'FFFFF603										
H'FFFFF604	GR2A									
H'FFFFF605										
H'FFFFF606	GR2B									
H'FFFFF607										
H'FFFFF608	GR2C									
H'FFFFF609										
H'FFFFF60A	GR2D									
H'FFFFF60B										
H'FFFFF60C	GR2E									
H'FFFFF60D										
H'FFFFF60E	GR2F									
H'FFFFF60F										
H'FFFFF610	GR2G									
H'FFFFF611										
H'FFFFF612	GR2H									
H'FFFFF613										
H'FFFFF614	OCR2A									
H'FFFFF615										
H'FFFFF616	OCR2B									
H'FFFFF617										
H'FFFFF618	OCR2C									
H'FFFFF619										
H'FFFFF61A	OCR2D									
H'FFFFF61B										
H'FFFFF61C	OCR2E									
H'FFFFF61D										
H'FFFFF61E	OCR2F									
H'FFFFF61F										
H'FFFFF620	OCR2G									
H'FFFFF621										
H'FFFFF622	OCR2H									
H'FFFFF623										
H'FFFFF624	OSBR2									
H'FFFFF625										

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF626	TIOR2B	—	IO2D2	IO2D1	IO2D0	—	IO2C2	IO2C1	IO2C0	ATU-II (channel 2)
H'FFFFF627	TIOR2A	—	IO2B2	IO2B1	IO2B0	—	IO2A2	IO2A1	IO2A0	
H'FFFFF628	TIOR2D	—	IO2H2	IO2H1	IO2H0	—	IO2G2	IO2G1	IO2G0	
H'FFFFF629	TIOR2C	—	IO2F2	IO2F1	IO2F0	—	IO2E2	IO2E1	IO2E0	
H'FFFFF62A	TCR2B	—	—	CKEGB1	CKEGB0	CKSELB3	CKSELB2	CKSELB1	CKSELB0	
H'FFFFF62B	TCR2A	—	—	CKEGA1	CKEGA0	CKSELA3	CKSELA2	CKSELA1	CKSELA0	
H'FFFFF62C	TSR2A	—	—	—	—	—	—	—	OVF2A	
H'FFFFF62D		IMF2H	IMF2G	IMF2F	IMF2E	IMF2D	IMF2C	IMF2B	IMF2A	
H'FFFFF62E	TSR2B	—	—	—	—	—	—	—	OVF2B	
H'FFFFF62F		CMF2H	CMF2G	CMF2F	CMF2E	CMF2D	CMF2C	CMF2B	CMF2A	
H'FFFFF630	TIER2A	—	—	—	—	—	—	—	OVE1A	
H'FFFFF631		IME2H	IME2G	IME2F	IME2E	IME2D	IME2C	IME2B	IME2A	
H'FFFFF632	TIER2B	—	—	—	—	—	—	—	OVE2B	
H'FFFFF633		CME2H	CME2G	CME2F	CME2E	CME2D	CME2C	CME2B	CME2A	
H'FFFFF634 to H'FFFFF63F	—	—	—	—	—	—	—	—	—	—
H'FFFFF640	DCNT8A									ATU-II (channel 8)
H'FFFFF641										
H'FFFFF642	DNCT8B									
H'FFFFF643										
H'FFFFF644	DNCT8C									
H'FFFFF645										
H'FFFFF646	DCNT8D									
H'FFFFF647										
H'FFFFF648	DCNT8E									
H'FFFFF649										
H'FFFFF64A	DCNT8F									
H'FFFFF64B										
H'FFFFF64C	DCNT8G									
H'FFFFF64D										
H'FFFFF64E	DCNT8H									
H'FFFFF64F										
H'FFFFF650	DCNT8I									
H'FFFFF651										

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF652	DCNT8J									ATU-II (channel 8)
H'FFFFF653										
H'FFFFF654	DCNT8K									
H'FFFFF655										
H'FFFFF656	DCNT8L									
H'FFFFF657										
H'FFFFF658	DCNT8M									
H'FFFFF659										
H'FFFFF65A	DCNT8N									
H'FFFFF65B										
H'FFFFF65C	DCNT8O									
H'FFFFF65D										
H'FFFFF65E	DCNT8P									
H'FFFFF65F										
H'FFFFF660	RLDR8									
H'FFFFF661										
H'FFFFF662	TCNR	CN8P	CN8O	CN8N	CN8M	CN8L	CN8K	CN8J	CN8I	
H'FFFFF663		CN8H	CN8G	CN8F	CN8E	CN8D	CN8C	CN8B	CN8A	
H'FFFFF664	OTR	OTEP	OTEO	OTEN	OTEM	OTEL	OTEK	OTEJ	OTEI	
H'FFFFF665		OTEH	OTEG	OTEF	OTEE	OTED	OTEC	OTEB	OTEA	
H'FFFFF666	DSTR	DST8P	DST8O	DST8N	DST8M	DST8L	DST8K	DST8J	DST8I	
H'FFFFF667		DST8H	DST8G	DST8F	DST8E	DST8D	DST8C	DST8B	DST8A	
H'FFFFF668	TCR8	—	CKSELB2	CKSELB1	CKSELB0	—	CKSELA2	CKSELA1	CKSELA0	
H'FFFFF669	—	—	—	—	—	—	—	—	—	
H'FFFFF66A	TSR8	OSF8P	OSF8O	OSF8N	OSF8M	OSF8L	OSF8K	OSF8J	OSF8I	
H'FFFFF66B		OSF8H	OSF8G	OSF8F	OSF8E	OSF8D	OSF8C	OSF8B	OSF8A	
H'FFFFF66C	TIER8	OSE8P	OSE8O	OSE8N	OSE8M	OSE8L	OSE8K	OSE8J	OSE8I	
H'FFFFF66D		OSE8H	OSE8G	OSE8F	OSE8E	OSE8D	OSE8C	OSE8B	OSE8A	
H'FFFFF66E	RLDENR	RLDEN	—	—	—	—	—	—	—	
H'FFFFF66F to H'FFFFF67F	—	—	—	—	—	—	—	—	—	—
H'FFFFF680	ECNT9A									ATU-II (channel 9)
H'FFFFF681	—	—	—	—	—	—	—	—	—	
H'FFFFF682	ECNT9B									
H'FFFFF683	—	—	—	—	—	—	—	—	—	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF684	ECNT9C									ATU-II (channel 9)
H'FFFFF685	—	—	—	—	—	—	—	—	—	
H'FFFFF686	ECNT9D									
H'FFFFF687	—	—	—	—	—	—	—	—	—	
H'FFFFF688	ECNT9E									
H'FFFFF689	—	—	—	—	—	—	—	—	—	
H'FFFFF68A	ECNT9F									
H'FFFFF68B	—	—	—	—	—	—	—	—	—	
H'FFFFF68C	GR9A									
H'FFFFF68D	—	—	—	—	—	—	—	—	—	
H'FFFFF68E	GR9B									
H'FFFFF68F	—	—	—	—	—	—	—	—	—	
H'FFFFF690	GR9C									
H'FFFFF691	—	—	—	—	—	—	—	—	—	
H'FFFFF692	GR9D									
H'FFFFF693	—	—	—	—	—	—	—	—	—	
H'FFFFF694	GR9E									
H'FFFFF695	—	—	—	—	—	—	—	—	—	
H'FFFFF696	GR9F									
H'FFFFF697	—	—	—	—	—	—	—	—	—	
H'FFFFF698	TCR9A	—	TRG3BEN	EGSELB1	EGSELB0	—	TRG3AEN	EGSELA1	EGSELA0	
H'FFFFF699	—	—	—	—	—	—	—	—	—	
H'FFFFF69A	TCR9B	—	TRG3DEN	EGSELD1	EGSELD0	—	TRG3CEN	EGSELC1	EGSELC0	
H'FFFFF69B	—	—	—	—	—	—	—	—	—	
H'FFFFF69C	TCR9C	—	—	EGSELF1	EGSELF0	—	—	EGSELE1	EGSELE0	
H'FFFFF69D	—	—	—	—	—	—	—	—	—	
H'FFFFF69E	TSR9	—	—	—	—	—	—	—	—	
H'FFFFF69F		—	—	CMF9F	CMF9E	CMF9D	CMF9C	CMF9B	CMF9A	
H'FFFFF6A0	TIER9	—	—	—	—	—	—	—	—	
H'FFFFF6A1		—	—	CME9F	CME9E	CME9D	CME9C	CME9B	CME9A	
H'FFFFF6A2 to H'FFFFF6BF	—	—	—	—	—	—	—	—	—	—
H'FFFFF6C0	TCNT10AH									ATU-II (channel 10)
H'FFFFF6C1										
H'FFFFF6C2	TCNT10AL									
H'FFFFF6C3										

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF6C4	TCNT10B									ATU-II (channel 10)
H'FFFFF6C5	—	—	—	—	—	—	—	—	—	
H'FFFFF6C6	TCNT10C									
H'FFFFF6C7										
H'FFFFF6C8	TCNT10D									
H'FFFFF6C9	—	—	—	—	—	—	—	—	—	
H'FFFFF6CA	TCNT10E									
H'FFFFF6CB										
H'FFFFF6CC	TCNT10F									
H'FFFFF6CD										
H'FFFFF6CE	TCNT10G									
H'FFFFF6CF										
H'FFFFF6D0	ICR10AH									
H'FFFFF6D1										
H'FFFFF6D2	ICR10AL									
H'FFFFF6D3										
H'FFFFF6D4	OCR10AH									
H'FFFFF6D5										
H'FFFFF6D6	OCR10AL									
H'FFFFF6D7										
H'FFFFF6D8	OCR10B									
H'FFFFF6D9	—	—	—	—	—	—	—	—	—	
H'FFFFF6DA	RLD10C									
H'FFFFF6DB										
H'FFFFF6DC	GR10G									
H'FFFFF6DD										
H'FFFFF6DE	TCNT10H									
H'FFFFF6DF	—	—	—	—	—	—	—	—	—	
H'FFFFF6E0	NCR10									
H'FFFFF6E1	—	—	—	—	—	—	—	—	—	
H'FFFFF6E2	TIOR10	RLDEN	CCS	PIM1	PIM0	—	IO10G2	IO10G1	IO10G0	
H'FFFFF6E3	—	—	—	—	—	—	—	—	—	
H'FFFFF6E4	TCR10	TRG2BEN	TRG1BEN	TRG2AEN	TRG1AEN	TRG0DEN	NCE	CKEG1	CKEG0	
H'FFFFF6E5	—	—	—	—	—	—	—	—	—	
H'FFFFF6E6	TCCLR10									
H'FFFFF6E7										

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF6E8	TSR10	—	—	—	—	—	—	—	—	ATU-II (channel 10)
H'FFFFF6E9		—	—	—	—	CMF10G	CMF10B	ICF10A	CMF10A	
H'FFFFF6EA	TIER10	—	—	—	—	—	—	—	—	
H'FFFFF6EB		—	—	—	IREG	CME10G	CME10B	ICE10A	CME10A	
H'FFFFF6EC to H'FFFFF6FF	—	—	—	—	—	—	—	—	—	—
H'FFFFF700	POPCR	PULS7 ROE	PULS6 ROE	PULS5 ROE	PULS4 ROE	PULS3 ROE	PULS2 ROE	PULS1 ROE	PULS0 ROE	APC
H'FFFFF701		PULS7 SOE	PULS6 SOE	PULS5 SOE	PULS4 SOE	PULS3 SOE	PULS2 SOE	PULS1 SOE	PULS0 SOE	
H'FFFFF702 to H'FFFFF707	—	—	—	—	—	—	—	—	—	—
H'FFFFF708	SYSCR	—	—	—	—	—	—	AUDSRST	RAME	Power-down state
H'FFFFF709	—	—	—	—	—	—	—	—	—	
H'FFFFF70A	—	—	—	—	—	—	—	—	—	
H'FFFFF70B	MSTCR *	—	—	—	—	MSTOP3	MSTOP2	MSTOP1	MSTOP0	
H'FFFFF70C to H'FFFFF70F	—	—	—	—	—	—	—	—	—	—
H'FFFFF710	CMSTR	—	—	—	—	—	—	—	—	CMT
H'FFFFF711		—	—	—	—	—	—	STR1	STR0	
H'FFFFF712	CMCSR0	—	—	—	—	—	—	—	—	
H'FFFFF713		CMF	CMIE	—	—	—	—	CKS1	CKS0	
H'FFFFF714	CMCNT0									
H'FFFFF715										
H'FFFFF716	CMCOR0									
H'FFFFF717										
H'FFFFF718	CMCSR1	—	—	—	—	—	—	—	—	
H'FFFFF719		CMF	CMIE	—	—	—	—	CKS1	CKS0	
H'FFFFF71A	CMCNT1									
H'FFFFF71B										
H'FFFFF71C	CMCOR1									
H'FFFFF71D										
H'FFFFF71E	—	—	—	—	—	—	—	—	—	
H'FFFFF71F	—	—	—	—	—	—	—	—	—	

Note: * This is the read address. The write address is H'FFFFF70A. For details, see section 24.2.4, Register Access.

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF720	PAIOR	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR	Port A
H'FFFFF721		PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR	
H'FFFFF722	PACRH	—	PA15MD	—	PA14MD	—	PA13MD	—	PA12MD	
H'FFFFF723		—	PA11MD	—	PA10MD	—	PA9MD	—	PA8MD	
H'FFFFF724	PACRL	—	PA7MD	—	PA6MD	—	PA5MD	—	PA4MD	
H'FFFFF725		—	PA3MD	—	PA2MD	—	PA1MD	—	PA0MD	
H'FFFFF726	PADR	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	
H'FFFFF727		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
H'FFFFF728	PHIOR	PH15IOR	PH14IOR	PH13IOR	PH12IOR	PH11IOR	PH10IOR	PH9IOR	PH8IOR	Port H
H'FFFFF729		PH7IOR	PH6IOR	PH5IOR	PH4IOR	PH3IOR	PH2IOR	PH1IOR	PH0IOR	
H'FFFFF72A	PHCR	PH15MD	PH14MD	PH13MD	PH12MD	PH11MD	PH10MD	PH9MD	PH8MD	
H'FFFFF72B		PH7MD	PH6MD	PH5MD	PH4MD	PH3MD	PH2MD	PH1MD	PH0MD	
H'FFFFF72C	PHDR	PH15DR	PH14DR	PH13DR	PH12DR	PH11DR	PH10DR	PH9DR	PH8DR	
H'FFFFF72D		PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR	
H'FFFFF72E	ADTRGR1	EXTRG	—	—	—	—	—	—	—	A/D
H'FFFFF72F	ADTRGR2	EXTRG	—	—	—	—	—	—	—	
H'FFFFF730	PBIOR	PB15IOR	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR	Port B
H'FFFFF731		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR	
H'FFFFF732	PBCRH	PB15MD1	PB15MD0	PB14MD1	PB14MD0	—	PB13MD	PB12MD1	PB12MD0	
H'FFFFF733		PB11MD1	PB11MD0	PB10MD1	PB10MD0	PB9MD1	PB9MD0	PB8MD1	PB8MD0	
H'FFFFF734	PBCRL	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0	
H'FFFFF735		—	PB3MD	—	PB2MD	—	PB1MD	—	PB0MD	
H'FFFFF736	PBIR	PB15IR	PB14IR	PB13IR	—	PB11IR	PB10IR	PB9IR	PB8IR	
H'FFFFF737		PB7IR	PB6IR	PB5IR	PB4IR	PB3IR	PB2IR	PB1IR	PB0IR	
H'FFFFF738	PBDR	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR	
H'FFFFF739		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
H'FFFFF73A	PCIOR	—	—	—	—	—	—	—	—	Port C
H'FFFFF73B		—	—	—	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR	
H'FFFFF73C	PCCR	—	—	—	—	—	—	—	PC4MC	
H'FFFFF73D		—	PC3MD	—	PC2MC	—	PC1MC	—	PC0MD	
H'FFFFF73E	PCDR	—	—	—	—	—	—	—	—	
H'FFFFF73F		—	—	—	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
H'FFFFF740	PDIOR	—	—	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR	Port D
H'FFFFF741		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF742	PDCRH	—	—	—	—	PD13MD1	PD13MD0	—	PD12MD	Port D
H'FFFFF743		—	PD11MD	—	PD10MD	—	PD9MD	—	PD8MD	
H'FFFFF744	PDCRL	—	PD7MD	—	PD6MD	—	PD5MD	—	PD4MD	
H'FFFFF745		—	PD3MD	—	PD2MD	—	PD1MD	—	PD0MD	
H'FFFFF746	PDDR	—	—	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	
H'FFFFF747		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
H'FFFFF748	PFIOR	PF15IOR	PF14IOR	PF13IOR	PF12IOR	PF11IOR	PF10IOR	PF9IOR	PF8IOR	Port F
H'FFFFF749		PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR	
H'FFFFF74A	PFCRH	CKHIZ	PF15MD	—	PF14MD	—	PF13MD	—	PF12MD	
H'FFFFF74B		—	PF11MD	—	PF10MD	—	PF9MD	—	PF8MD	
H'FFFFF74C	PFCRL	—	PF7MD	—	PF6MD	PF5MD1	PF5MD0	—	PF4MD	
H'FFFFF74D		—	PF3MD	—	PF2MD	—	PF1MD	—	PF0MD	
H'FFFFF74E	PFDR	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR	
H'FFFFF74F		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
H'FFFFF750	PEIOR	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR	Port E
H'FFFFF751		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
H'FFFFF752	PECR	PE15MD	PE14MD	PE13MD	PE12MD	PE11MD	PE10MD	PE9MD	PE8MD	
H'FFFFF753		PE7MD	PE6MD	PE5MD	PE4MD	PE3MD	PE2MD	PE1MD	PE0MD	
H'FFFFF754	PEDR	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	
H'FFFFF755		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
H'FFFFF756	PLIOR	—	—	PL13IOR	PL12IOR	PL11IOR	PL10IOR	PL9IOR	PL8IOR	Port L
H'FFFFF757		PL7IOR	PL6IOR	PL5IOR	PL4IOR	PL3IOR	PL2IOR	PL1IOR	PL0IOR	
H'FFFFF758	PLCRH	—	—	—	—	PL13MD1	PL13MD0	—	PL12MD	
H'FFFFF759		PL11MD1	PL11MD0	PL10MD1	PL10MD0	PL9MD1	PL9MD0	—	PL8MD	
H'FFFFF75A	PLCRL	—	PL7MD	—	PL6MD	—	PL5MD	—	PL4MD	
H'FFFFF75B		—	PL3MD	PL2MD1	PL2MD0	PL1MD1	PL1MD0	—	PL0MD0	
H'FFFFF75C	PLIR	—	—	—	—	—	—	PL9IR	PL8IR	
H'FFFFF75D		PL7IR	—	—	—	—	—	—	—	
H'FFFFF75E	PLDR	—	—	PL13DR	PL12DR	PL11DR	PL10DR	PL9DR	PL8DR	
H'FFFFF75F		PL7DR	PL6DR	PL5DR	PL4DR	PL3DR	PL2DR	PL1DR	PL0DR	
H'FFFFF760	PGIOR	—	—	—	—	—	—	—	—	Port G
H'FFFFF761		—	—	—	—	PG3IOR	PG2IOR	PG1IOR	PG0IOR	
H'FFFFF762	PGCR	—	—	—	—	—	—	—	—	
H'FFFFF763		PG3MD1	PG3MD0	PG2MD1	PG2MD0	—	PG1MD	PG0MD1	PG0MD0	
H'FFFFF764	PGDR	—	—	—	—	—	—	—	—	
H'FFFFF765		—	—	—	—	PG3DR	PG2DR	PG1DR	PG0DR	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF766	PJIOR	PJ15IOR	PJ14IOR	PJ13IOR	PJ12IOR	PJ11IOR	PJ10IOR	PJ9IOR	PJ8IOR	Port J
H'FFFFF767		PJ7IOR	PJ6IOR	PJ5IOR	PJ4IOR	PJ3IOR	PJ2IOR	PJ1IOR	PJ0IOR	
H'FFFFF768	PJCRH	—	PJ15MD	—	PJ14MD	—	PJ13MD	—	PJ12MD	
H'FFFFF769		—	PJ11MD	—	PJ10MD	—	PJ9MD	—	PJ8MD	
H'FFFFF76A	PJCRL	—	PJ7MD	—	PJ6MD	—	PJ5MD	—	PJ4MD	
H'FFFFF76B		—	PJ3MD	—	PJ2MD	—	PJ1MD	—	PJ0MD	
H'FFFFF76C	PJDR	PJ15DR	PJ14DR	PJ13DR	PJ12DR	PJ11DR	PJ10DR	PJ9DR	PJ8DR	
H'FFFFF76D		PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR	
H'FFFFF76E	ADTRG0	EXTRG	—	—	—	—	—	—	—	A/D
H'FFFFF76F	—	—	—	—	—	—	—	—	—	
H'FFFFF770	PKIOR	PK15IOR	PK14IOR	PK13IOR	PK12IOR	PK11IOR	PK10IOR	PK9IOR	PK8IOR	Port K
H'FFFFF771		PK7IOR	PK6IOR	PK5IOR	PK4IOR	PK3IOR	PK2IOR	PK1IOR	PK0IOR	
H'FFFFF772	PKCRH	—	PK15MD	—	PK14MD	—	PK13MD	—	PK12MD	
H'FFFFF773		—	PK11MD	—	PK10MD	—	PK9MD	—	PK8MD	
H'FFFFF774	PKCRL	—	PK7MD	—	PK6MD	—	PK5MD	—	PK4MD	
H'FFFFF775		—	PK3MD	—	PK2MD	—	PK1MD	—	PK0MD	
H'FFFFF776	PKIR	PK15IR	PK14IR	PK13IR	PK12IR	PK11IR	PK10IR	PK9IR	PK8IR	
H'FFFFF777		PK7IR	PK6IR	PK5IR	PK4IR	PK3IR	PK2IR	PK1IR	PK0IR	
H'FFFFF778	PKDR	PK15DR	PK14DR	PK13DR	PK12DR	PK11DR	PK10DR	PK9DR	PK8DR	
H'FFFFF779		PK7DR	PK6DR	PK5DR	PK4DR	PK3DR	PK2DR	PK1DR	PK0DR	
H'FFFFF77A to H'FFFFF77F	—	—	—	—	—	—	—	—	—	—
H'FFFFF780	PAPR	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	Port A
H'FFFFF781		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
H'FFFFF782	PBPR	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR	Port B
H'FFFFF783		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
H'FFFFF784	PDPR	—	—	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR	Port D
H'FFFFF785		PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR	
H'FFFFF786	PJPR	PJ15PR	PJ14PR	PJ13PR	PJ12PR	PJ11PR	PJ10PR	PJ9PR	PJ8PR	Port J
H'FFFFF787		PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR	
H'FFFFF788	PLPR	—	—	PL13PR	PL12PR	PL11PR	PL10PR	PL9PR	PL8PR	Port L
H'FFFFF789		PL7PR	PL6PR	PL5PR	PL4PR	PL3PR	PL2PR	PL1PR	PL0PR	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF78A to H'FFFFF7BF	—	—	—	—	—	—	—	—	—	—
H'FFFFF7C0	SDIR	TS3	TS2	TS1	TS0	—	—	—	—	H-UDI
H'FFFFF7C1		—	—	—	—	—	—	—	—	
H'FFFFF7C2	SDSR	—	—	—	—	—	—	—	—	
H'FFFFF7C3		—	—	—	—	—	—	—	SDTRF	
H'FFFFF7C4	SDDRH									
H'FFFFF7C5										
H'FFFFF7C6	SDDRL									
H'FFFFF7C7										
H'FFFFF7C8 to H'FFFFF7FF	—	—	—	—	—	—	—	—	—	—
H'FFFFF800	ADDR0H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
H'FFFFF801	ADDR0L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF802	ADDR1H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF803	ADDR1L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF804	ADDR2H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF805	ADDR2L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF806	ADDR3H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF807	ADDR3L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF808	ADDR4H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF809	ADDR4L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF80A	ADDR5H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF80B	ADDR5L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF80C	ADDR6H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF80D	ADDR6L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF80E	ADDR7H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF80F	ADDR7L	AD1	AD0	—	—	—	—	—	—	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF810	ADDR8H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
H'FFFFF811	ADDR8L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF812	ADDR9H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF813	ADDR9L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF814	ADDR10H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF815	ADDR10L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF816	ADDR11H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF817	ADDR11L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF818	ADCSR0	ADF	ADIE	ADM1	ADM0	CH3	CH2	CH1	CH0	
H'FFFFF819	ADCR0	TRGE	CKS	ADST	ADCS	—	—	—	—	
H'FFFFF81A to H'FFFFF81F	—	—	—	—	—	—	—	—	—	
H'FFFFF820	ADDR12H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF821	ADDR12L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF822	ADDR13H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF823	ADDR13L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF824	ADDR14H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF825	ADDR14L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF826	ADDR15H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF827	ADDR15L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF828	ADDR16H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF829	ADDR16L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF82A	ADDR17H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF82B	ADDR17L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF82C	ADDR18H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF82D	ADDR18L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF82E	ADDR19H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF82F	ADDR19L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF830	ADDR20H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF831	ADDR20L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF832	ADDR21H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF833	ADDR21L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF834	ADDR22H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF835	ADDR22L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF836	ADDR23H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	

Address	Abbr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'FFFFF837	ADDR23L	AD1	AD0	—	—	—	—	—	—	A/D
H'FFFFF838	ADCSR1	ADF	ADIE	ADM1	ADM0	CH3	CH2	CH1	CH0	
H'FFFFF839	ADCR1	TRGE	CKS	ADST	ADCS	—	—	—	—	
H'FFFFF840	ADDR24H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF841	ADDR24L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF842	ADDR25H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF843	ADDR25L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF844	ADDR26H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF845	ADDR26L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF846	ADDR27H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF847	ADDR27L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF848	ADDR28H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF849	ADDR28L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF84A	ADDR29H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF84B	ADDR29L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF84C	ADDR30H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF84D	ADDR30L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF84E	ADDR31H	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FFFFF84F	ADDR31L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF850 to H'FFFFF857	—	—	—	—	—	—	—	—	—	
H'FFFFF858	ADCSR2	ADF	ADIE	ADM1	ADM0	—	CH2	CH1	CH0	
H'FFFFF859	ADCR2	TRGE	CKS	ADST	ADCS	—	—	—	—	
H'FFFFF85A to H'FFFFF85F	—	—	—	—	—	—	—	—	—	
H'FFFFF83A to H'FFFFF83F	—	—	—	—	—	—	—	—	—	

Table A.2 Register States in Reset and Power-Down States

		Reset State	Power-Down State		
Type	**Name**	**Power-On**	**Hardware Standby**	**Software Standby**	**Sleep**
CPU	R0 to R15	Initialized	Initialized	Held	Held
	SR				
	GBR				
	VBR				
	MACH, MACL				
	PR				
	PC				
FPU	FR0 to FR15	Initialized	Initialized	Held	Held
	FPUL				
	FPSCR				
Interrupt controller (INTC)	IPRA to IPRL	Initialized	Initialized	Held	Held
	IOR				
	ISR				
User break controller (UBC)	UBARH, UBARL	Initialized	Initialized	Held	Held
	UBAMRH, UBAMRL				
	UBBR				
	UBCR				
Bus state controller (BSC)	BCR1, BCR2	Initialized	Initialized	Held	Held
	WCR				
Direct memory access controller (DMAC)	SAR0 to SAR3	Initialized	Initialized	Initialized	Held
	DAR0 to DAR3				
	DMATCR0 to DMATCR3				
	CHCR0 to CHCR3				
	DMAOR				
Advanced timer unit-II (ATU-II)	BFR6A-D, BFR7A-D	Initialized	Initialized	Initialized	Held
	CYLR6A-D, CYLR7A-D				
	DCNT8A-P				
	DSTR				

Type	Name	Power-On	Hardware Standby	Software Standby	Sleep
Advanced timer unit-II (ATU-II)	DTR6A-D, DTR7A-D	Initialized	Initialized	Initialized	Held
	ECNT9A-F				
	GR1A-H, GR2A-H GR3A-D, GR4A-D GR5A-D, GR9A-F GR10G, GR11A, 11B				
	ICR0A-D, ICR10A				
	ITVRR1, ITVRR2A, 2B				
	NCR10				
	OCR1, OCR2A-H OCR10AH, 10AL OCR10B				
	OSBR1, OSBR2				
	OTR				
	PMDR				
	PSCR1-4				
	PSTR				
	RLD10C				
	RLDENR				
	RLDR8				
	TCCLR10				
	TCNR				
	TCNT0H, L, TCNT1A 1B, TCNT2A, 2B TCNT3-5, TCNT6A-D TCNT7A-D TCNT10AH, 10AL TCNT10B-H, TCNT11				
	TCR1A, 1B TCR2A, 2B, TCR3-5 TCR6A, 6B, TCR7A 7B, TCR8, TCR9A-C TCR10, TCR11				

Type	Name	Power-On	Hardware Standby	Software Standby	Sleep
Advanced timer unit-II (ATU-II)	TIER0, TIER1A, 1B TIER2A, 2B, TIER3 TIER6-11	Initialized	Initialized	Initialized	Held
	TIOR0, TIOR1A-D TIOR2A-D, TIOR3A 3B, TIOR4A, 4B TIOR5A, 5B TIOR10,11				
	TMDR				
	TNCT10E				
	TRGMDR				
	TSR0, TSR1A, 1B TSR2A, 2B, TSR3 TSR6-11				
	TSTR1-3				
Advanced pulse controller (APC)	POPCR	Initialized	Initialized	Held	Held
Watchdog timer (WDT)	TCNT	Initialized	Initialized	Initialized	Held
	TCSR				
	RSTCSR				
Serial communication interface (SCI)	SMR0 to SMR4	Initialized	Initialized	Held	Held
	BRR0 to BRR4				
	SCR0 to SCR4				
	TDR0 to TDR4			Intialized	
	SSR0 to SSR4				
	RDR0 to RDR4				
	SDCR0 to SDCR4			Held	
A/D converter	ADDR0 (H/L) to ADDR31 (H/L)	Initialized	Initialized	Initialized	Held
	ADSCR0, ADCSR1 ADCSR2				
	ADCR0, ADCR1 ADCR2				

Type	Name	Power-On	Hardware Standby	Software Standby	Sleep
A/D converter	ADTRGR0, ADTRGR1 ADTRGR2	Initialized	Initialized	Initialized	Held
Compare match timer (CMT)	CMSTR	Initialized	Initialized	Initialized	Held
	CMCSR0, CMCSR1				
	CMCNT0, CMCNT1	Initialized	Initialized	Initialized	Held
	CMCOR0, CMCOR1				
Pin function controller (PFC)	PAIOR, PBIOR PCIOR, PDIOR PEIOR, PFIOR PGIOR, PHIOR PJIOR, PKIOR, PLIOR	Initialized	Initialized	Held	Held
	PACRH, PACRL PBCRH, PBCRL PBIR, PCCR, PDCRH PDCRL, PECR PFCRH, PFCRL PGCR, PHCR, PJCRH PJCRL PKCRH PKCRL PKIR, PLCRH PLCRL,PLIR				
I/O ports	PADR, PBDR, PCDR PDDR, PEDR, PFDR PGDE, PHDR, PJDR PKDR, PLDR	Initialized	Initialized	Held	Held
	PAPR, PBPR, PDPR, PJPR, PLPR	Pin value	Held	Held	Pin value
Flash ROM	RAMER	Initialized	Initialized	Held	Held
	FCCS			Initialized/ Held*	
	FPCS			Initialized	
	FECS				
	FKEY				
	FMATS			Held	
	FTDAR			Initialized	

Type	Name	Power-On	Hardware Standby	Software Standby	Sleep
Power-down state related	SBYCR	Initialized	Initialized	Held	Held
	SYSCR				
	MSTCR				
Controller area network (HCAN)	MCR	Initialized	Initialized	Initialized	Held
	GSR				
	BCR				
	MBCR				
	TXPR				
	TXCR				
	TXACK				
	ABACK				
	RXPR				
	RFPR				
	IRR				
	MBIMR				
	IMR				
	REC				
	TEC				
	UMSR				
	LAFML				
	LAFMH				
	MC0 [1:8] to MC15 [1:8]	Underfined	Underfined	Underfined	
	MD0 [1:8] to MD15 [1:8]				
High-performance user debug interface (H-UDI)	SDIR	Held	Held	Held	Held
	SDSR				
	SDDRH, SDDRL				

Note: * Bit 7 (FLER) is held, and bit 0 (SCO) is initialized.

Tables B.1, B.2, and B.3 show the SH7055SF pin states.

Table B.1 Pin States

		Pin State								
		Reset State				Power-Down State				
		Power-On								
		ROMless Expanded Mode		Expanded Mode with ROM	Single-Chip Mode	Hardware Standby	Software Standby	H-UDI Module Standby	AUD Module Standby	Bus-Released State
Type	Pin Name	8 Bits	16 Bits							
Clock	CK*[2]	O				Z	H*[1]	O	O	O
	XTAL	O				L	L	O	O	O
	EXTAL	I				Z	I	I	I	I
	PLLCAP	I				I	I	I	I	I
System control	$\overline{\text{RES}}$	I				Z	I	I	I	I
	FWE	I				I	I	I	I	I
	$\overline{\text{HSTBY}}$	I				I	I	I	I	I
	MD0	I				I	I	I	I	I
	MD1	I				I	I	I	I	I
	MD2	I				I	I	I	I	I
	$\overline{\text{WDTOVF}}$	O				Z	O*[1]	O	O	O
	$\overline{\text{BREQ}}$	—				Z	Z	I	I	I
	$\overline{\text{BACK}}$	—				Z	Z	O	O	L
Interrupt	NMI	I				Z	I	I	I	I
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$	—				Z	Z	I	I	I
	$\overline{\text{IRQOUT}}$	—				Z	O*[1]	O	O	O
Address bus	A0 to A21	O		—		Z	Z	O	O	Z
Data bus	D0 to D7	Z		—		Z	Z	I/O	I/O	Z
	D8 to D15	—	Z	—		Z	Z	I/O	I/O	Z
Bus control	$\overline{\text{WAIT}}$	I			—	Z	Z	I	I	I
	$\overline{\text{WRH}}$, $\overline{\text{WRL}}$	H			—	Z	Z	O	O	Z
	$\overline{\text{RD}}$	H			—	Z	Z	O	O	Z
	$\overline{\text{CS0}}$	H			—	Z	Z	O	O	Z
	$\overline{\text{CS1}}$ to $\overline{\text{CS3}}$	—				Z	Z	O	O	Z
Port	$\overline{\text{POD}}$	—				Z	Z	I	I	I
ATU-II	TI0A to TI0D	—				Z	Z	I	I	I
	TIO1A to TIO1H	—				Z	K*[1]	I/O	I/O	I/O
	TIO2A to TIO2H	—				Z	K*[1]	I/O	I/O	I/O
	TIO3A to TIO3D	—				Z	K*[1]	I/O	I/O	I/O

		Reset State				Power-Down State				
		Power-On								
		ROMless Expanded Mode		Expanded Mode with ROM	Single-Chip Mode	Hardware Standby	Software Standby	H-UDI Module Standby	AUD Module Standby	Bus-Released State
Type	Pin Name	8 Bits	16 Bits							
ATU-II	TIO4A to TIO4D	—				Z	K*[1]	I/O	I/O	I/O
	TIO5A to TIO5D	—				Z	K*[1]	I/O	I/O	I/O
	TO6A to TO6D	—				Z	O*[1]	O	O	O
	TO7A to TO7D	—				Z	O*[1]	O	O	O
	TO8A to TO8P	—				Z	O*[1]	O	O	O
	TI9A to TI9F	—				Z	Z	I	I	I
	TI10	—				Z	Z	I	I	I
	TIO11A, TIO11B	—				Z	K*[1]	I/O	I/O	I/O
	TCLKA, TCLKB	—				Z	Z	I	I	I
SCI	SCK0 to SCK4	—				Z	K*[1]	I/O	I/O	I/O
	TxD0 to TxD4	—				Z	O*[1]	O	O	O
	RxD0 to RxD4	—				Z	Z	I	I	I
A/D converter	AN0 to AN31	Z				Z	Z	I	I	I
	$\overline{\text{ADTRG0}}$, $\overline{\text{ADTRG1}}$	—				Z	Z	I	I	I
	ADEND	—				Z	O*[1]	O	O	O
	AVref	I				I	I	I	I	I
APC	PULS0 to PULS7	—				Z	O*[1]	O	O	O
HCAN	HTxD0, HTxD1	—				Z	O*[1]	O	O	O
	HRxD0, HRxD1	—				Z	Z	I	I	I
UBC	$\overline{\text{UBCTRG}}$	—				Z	O*[1]	O	Z	O
I/O port	PA0 to PA15	Z				Z	K*[1]	I/O	I/O	I/O
	PB0 to PB15	Z				Z	K*[1]	I/O	I/O	I/O
	PC0 to PC4	Z				Z	K*[1]	I/O	I/O	I/O
	PD0 to PD13	Z				Z	K*[1]	I/O	I/O	I/O
	PE0 to PE15	—		Z		Z	K*[1]	I/O	I/O	I/O
	PF0 to PF5	—		Z		Z	K*[1]	I/O	I/O	I/O
	PF6 to PF10	—			Z	Z	K*[1]	I/O	I/O	I/O
	PH11 to PF15	Z				Z	K*[1]	I/O	I/O	I/O
	PG0 to PG3	Z				Z	K*[1]	I/O	I/O	I/O
	PH0 to PH7	—		Z		Z	K*[1]	I/O	I/O	I/O
	PH8 to PH15	Z	—	Z		Z	K*[1]	I/O	I/O	I/O
	PJ0 to PJ15	Z				Z	K*[1]	I/O	I/O	I/O
	PK0 to PK15	Z				Z	K*[1]	I/O	I/O	I/O
	PL0 to PL13	Z				Z	K*[1]	I/O	I/O	I/O

		Reset State				Power-Down State					
		Power-On									
		ROMless Expanded Mode		Expanded Mode with ROM	Single-Chip Mode	Hardware Standby	Software Standby	H-UDI Module Standby	AUD Module Standby	Bus-Released State	No Connection
Type	Pin Name	8 Bits	16 Bits								
H-UDI	TMS	I				Z	I	Z	I	I	Pulled up internally
	$\overline{\text{TRST}}$	I				Z	I	Z	I	I	Pulled up internally
	TDI	I				Z	I	Z	I	I	Pulled up internally
	TDO	O				Z	O	Z	O	O	O/Z
	TCK	I				Z	I	Z	I	I	Pulled up internally

Table B.3 Pin States

		Pin State			
Type	Pin Name	Hardware Standby $\overline{\text{AUD}}$ Module Standby	AUD Reset ($\overline{\text{AUDRST}}$ = L)	Software Standby $\overline{\text{AUDSRST}}$ = 1/ Normal Operation	No Connection
AUD	$\overline{\text{AUDRST}}$	Z	L input	H input	Pulled down internally
	AUDMD	Z	I	I	Pulled up internally
	AUDATA0 to AUDATA3	Z	When AUDMD = H: I When AUDMD = L: K (pulled up internally)	When AUDMD = H: I/O When AUDMD = L: O	Pulled up internally
	AUDCK	Z	When AUDMD = H: I When AUDMD = L: K (pulled up internally)	When AUDMD = H: I When AUDMD = L: O	Pulled up internally
	$\overline{\text{AUDSYNC}}$	Z	When AUDMD = H: I When AUDMD = L: K (pulled up internally)	When AUDMD = H: I When AUDMD = L: O	Pulled up internally

— : Not initial value

I : Input

O : Output

H : High-level output

L : Low-level output

Z : High impedance

K : Input pins become high-impedance, output pins retain their state.

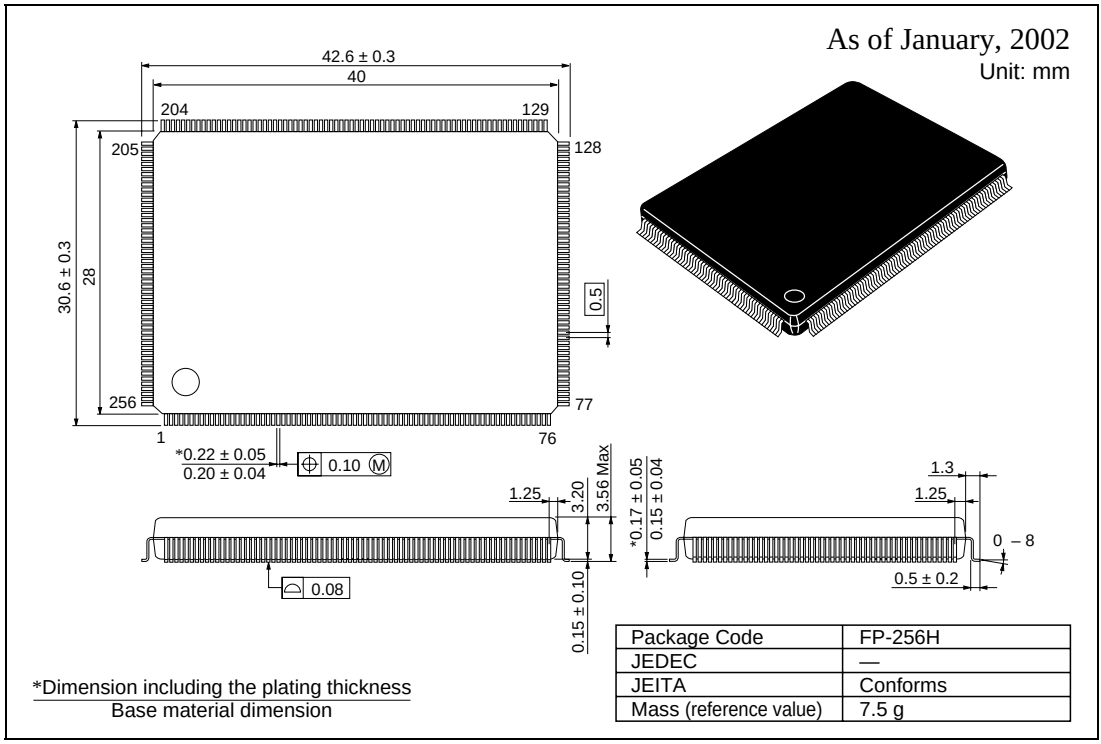
Notes: *1 When the port impedance bit (HIZ) in the standby control register (SBYCR) is set to 1, output pins become high-impedance.

*2 When the CKHIZ bit in PFCRH is set to 1, becomes high-impedance unconditionally.

Table C.1 SH7055S F-ZTAT Product Lineup

Product Type		Model Name	Mark Model Name	Package
SH7055SF	F-ZTAT	HD64F7055S	64F7055F40	256-pin (FP-256H)

Figure D.1 shows the FP-256H package dimensions of the SH7055SF.



Package Code	FP-256H
JEDEC	—
JEITA	Conforms
Mass (reference value)	7.5 g

Figure D.1 Package Dimensions (FP-256H)

SH-2E SH7055S F-ZTAT™ Hardware Manual

Publication Date: 1st Edition, May, 2002
Rev.2.00, July 17, 2003
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Technical Documentation & Information Department
Renesas Kodaira Semiconductor Co., Ltd.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

Colophon 6.0

SH-2E SH7055S F-ZTAT™ Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0045-0200H

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 8-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

CY8C20524-12PVXIT CY8C28433-24PVXIT MB95F012KPFT-G-SNE2 MB95F013KPMC-G-SNE2 MB95F263KPF-G-SNE2 MB95F264KPFT-G-SNE2 MB95F398KPMC-G-SNE2 MB95F478KPMC2-G-SNE2 MB95F562KPF-G-SNE2 MB95F564KPF-G-SNE2 MB95F634KPMC-G-SNE2 MB95F636KWQN-G-SNE1 MB95F696KPMC-G-SNE2 MB95F698KPMC1-G-SNE2 MB95F698KPMC2-G-SNE2 MB95F698KPMC-G-SNE2 MB95F818KPMC1-G-SNE2 MC908JK1ECDWER MC9S08PA32AVLD MC9S08PT60AVLD R5F1076CMSPV0 R5F5631ECDFBV0 C8051F389-B-GQ C8051F392-A-GMR ISD-ES1600_USB_PROG 901015X SC705C8AE0VFBE STM8TL53G4U6 PIC16F877-04/P-B R5F10Y17ASP#30 CY8C3MFIDOCK-125 403708R MB95F354EPF-G-SNE2 MB95F564KPFT-G-SNE2 MB95F564KWQN-G-SNE1 MB95F636KP-G-SH-SNE2 MB95F636KPMC-G-SNE2 MB95F694KPMC-G-SNE2 MB95F778JPMC1-G-SNE2 MB95F818KPMC-G-SNE2 MC908QY8CDWER MC9S08PT16AVLD MC9S08PT32AVLH MC9S08PT60AVLC MC9S08PT60AVLH C8051F500-IQR LC87F0G08AUJA-AH CP8361BT STM8S207C6T3 CG8421AF