EPSON
EXCEED YOUR VISION

S2S65A00
Technical Manual

SEIKO EPSON CORPORATION

Rev.1.3

NOTICE

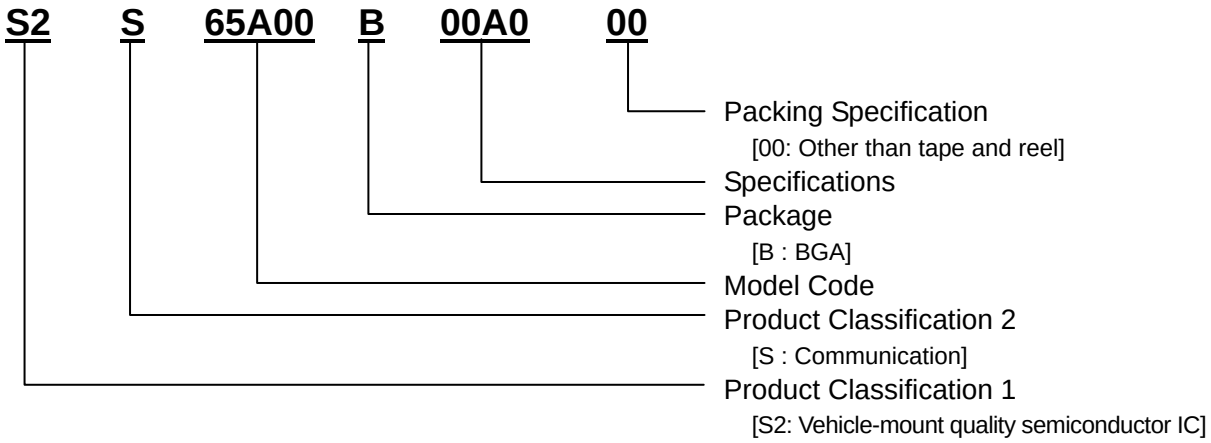
No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of Economy, Trade and Industry or other approval from another government agency.

is a registered trademark of ARM.
CompactFlash is a registered trademark of Sandisk.

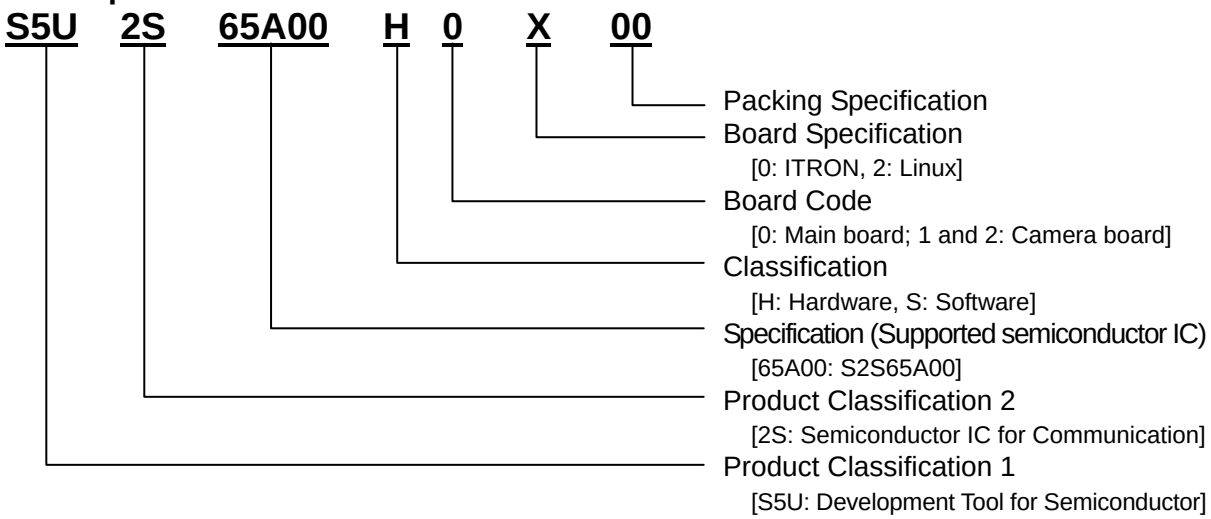
All other product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

Configuration of product number

● **Device**



● **Development tool**



Precautions in Use

For descriptions of the registers in this document, be careful with the following:

For descriptions of the registers in this document, the following abbreviations may be used.

R/W:	Read and Write
RO:	Read Only
WO:	Write Only
RSV:	Reserved bit/register (write down “0”, if not otherwise specified)
n/a:	not available (write down “0”, if not otherwise specified)

If not otherwise specified, set "0" in the reserved bits for the registers. If a write operation performed on a reserved bit, unexpected results may occur. The bits specified as "n/a" have no impact on the hardware.

Some of the resisters can be accessed only under certain conditions. Read/write to the non-accessible registers is ineffective.

Table of Contents

Appendix 2 USB Device Controller

< Table of Contents >

<List of Figures>

<List of Tables>

1. DESCRIPTION

This product, S2S65A00, is an IC specifically for a drive recorder. It has camera interfaces and JPEG encoder functions, as well as embedded CF, SD memory, USB 2.0 device (High-Speed supported) interfaces, and 8-ch ADC. A drive recorder can easily be configured by connecting S2S65A00 with camera modules, SDRAM, an external storage (CF or SD memory card), and Flash ROM containing firmware. For example, JPEG data generated by the embedded encoder may constantly be accumulated into the SDRAM. In response to an external trigger, such as detection of a rapid change in speed by the acceleration sensor IC, the accumulated image data can be transferred to and saved in an external storage (CF or SD memory card). In addition, S2S65A00 is equipped with GPIO and I^2C bus, which enable you to set cameras and control external devices.

1.1 Features

- One-chip solution, which can reduce system cost.
- Provides JPEG encoding by using 30 fps @VGA hardware (ISO 10918 compliant).
- Up to two camera modules can be connected.
- Each camera module has two hardware JPEG encoders.
- Provides moving-object detection function to support motion detection.
- Supports I^2S for voice data.
- Has a CompactFlash interface for a CF memory card or a wireless LAN interface (802.11b/g).
- An SD memory interface for SD memory card connection.
- ARM720T 50MHz operation.
- USB 2.0 device (High-Speed) function support, which enables connection to a PC.
- Supports 8-ch ADC for connection with various analog sensors.
- Contains event counter timers.
- Memory bus: 2 ports (6bit-Bus: FROM/SRAM, 16/32bit-Bus: SDRAM).

1.2 Built-In Functions

CPU:

- 32-bit RISC ARM720T (maximum of 56MHz).
- 32-bit long command codes and 16-bit long command codes called Efficient Thumb Code can be used by switching them.
- 32-bit general purpose register (×31).
- A multiplier is included in the CPU.

RAM:

- 56 KB Built-in RAM for CPU/JPEG Work (CPU Work: 32KB Max.).

Standby Function

- A HALT function to stop the CPU clock when any CPU operation is not required.
- An I/O clock stop function to stop each clock of the main I/O blocks.

Camera Input/JPEG Encoder:

- 8-bit parallel interface × 2 ports
- 2 camera modules can be connected.
- Up to 640×480 resolution (VGA, QVGA, CIF, QCIF).
- Hardware JPEG encoder × 2
- Throughput greater than 30 fps @VGA (when 1 camera module connected).
- YUV4-2-2 progressive (both ports)
- Pixel clock frequency for inputting camera data is less than 2/3 of CPU clock frequency.

1. DESCRIPTION

JPEG:
- Hardware JPEG encoder
- Resize function (screen can be cut off)
- Dedicated line buffer
- Variable volume FIFO built in the JPEG encoder output.
- An enhanced DMA is included in the network.

USB2.0 Device:
- Supports HS (480Mbps) and FS (12Mbps) transfer.
- Has built-in FS/HS Termination function (external circuit not required).
- VBUS 5V Interface (external protection circuit required).
- Supports control bulk and interrupt transfer.
- Supports 8 end points shared by control (End Point 0) and bulk/interrupt
- Has a 16-bit or 8-bit width general purpose CPU interface.
- Littler Endian is supported.
- Addition to and deletion form the register table is performed based on the HS-Device section of S1R72V05.
- 12MHz or 24MHz crystal transducer input is supported as clock input for USB.
- As internal clock use, the following frequencies are available based on the clock for the input USB.
- Clock for input USB: 12MHz or 24MHz.
- Clock for internal USB: 60MHz (via PLL for built-in USB).

Memory Controller:
- AHB bus interface memory controller.
- Supports up to four SRAM timing devices.
- Supports up to eight SDRAMs.
- Refresh interval of SDRAM auto-refresh can be adjusted to the device.
- SDRAM burst refresh support.
- Supports SDRAM self-refresh.

CF Card Interface:
- Complies with CF+ Specification Rev.1.4.
- Can be used as the interface of wireless LAN, PHS card, etc.
- Supports the True IDE mode.

SD Memory Interface:
- Complies with SD Memory Card Physical Layer Spec. ver.1.0.
- 1-bit/4-bit interface support

Interrupt Controller:
- Supports 32 IRQs and 2 FIQs.

Serial interface:
- UART: Compatible with 16550 software × 3 channel
- SPI: Clock synchronous type × 1 channel
- I^2C master interface (camera interface and multipurpose use)
- I^2S interface (voice/audio data supported, I^2S compliant)

Timer A:
- 16-bit timer × 3-channel timer
- Re-load/cyclic or one shot operation mode
- Supports toggle outputs resulted from underflow output or port outputs.

Timer B: for Event Counter
- 16-bit upcount timer
- Four COMMON registers are implemented, each of which can be configured as an output register or an input capture register.

Watchdog Timer
- Interrupt output or re-settable watchdog timer.

Real Time Clock
- Supports year, month, day, hour, minute and second.
- The internal timer tap from 1/128 to 1/2 can be used as the interrupt source as well.
- Supports alarm function and interrupt.

GPIO:
- General-purpose I/O port (up to 82)
- Programmable setting of directions is possible for all ports.
- Partly selects other I/O functions.

AD Converter:
- 8-channel analog signal inputs.
- 10-bits resolution AD converter.
- AD conversion time is 20μsec or less.

Power Supply:
- 3.3V (I/O power supply)
- 3.3V (USB)
- 1.8V (core power supply)
- 2.4 to 3.6V (Camera 1/2 I/O power supply)
- 2.7 to 3.6V (SDRAM I/O power supply)
- 3.3V (A/D power supply)
- 1.8V (USB/PLL/RTC)

Package:
- PFBGA 280 Pin (PFBGA16UX280) 16 × 16 × 1.2 mm, 0.8 mm ball pitch

2. BLOCK DIAGRAM

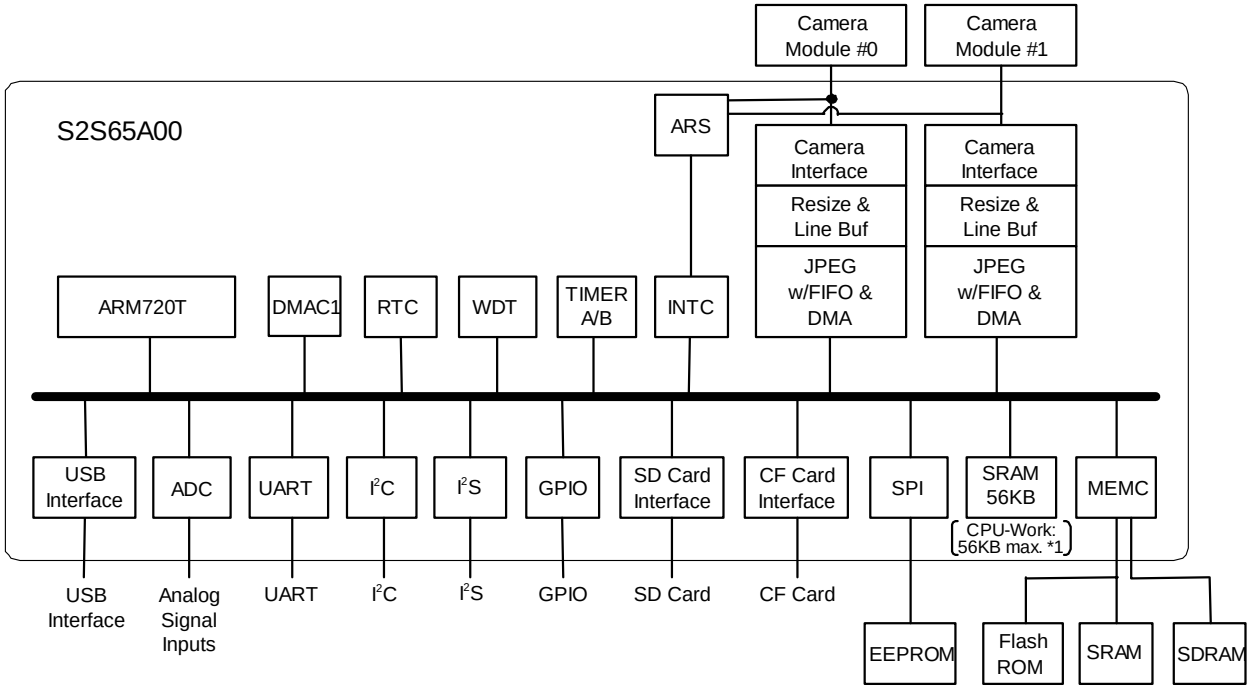


Fig.2.1 S2S65A00 Internal Block Diagram

Note (*1) : Internal SRAM is shared with Line Buffer of JPG[2:1].
When JPG[2:1] is used, CPU-Work cannot be used.

3. PIN

3.1 Pin Assignment

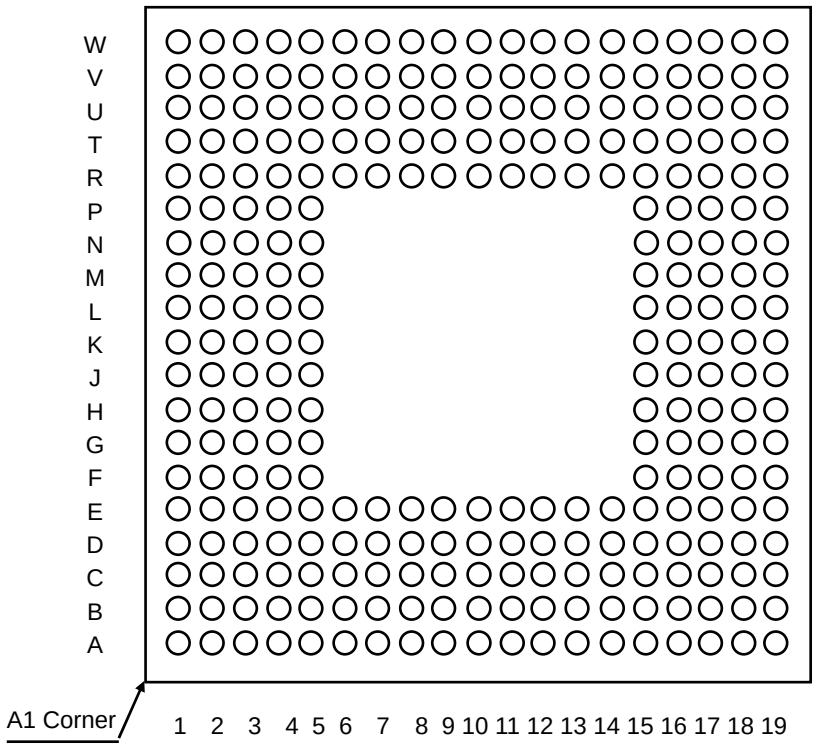


Fig.3.1 Pin Assignment (Bottom View)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
A1	NC	D14	HVDD	K15	LVDD	T7	CM2DATA2
A2	CM1CLKIN	D15	GPIOA2	K16	CFDEN#	T8	GPIOD2
A3	CM1CLKOUT	D16	GPIOA4	K17	CFDDIR	T9	GPIOD1
A4	C1VDD	D17	GPIOA0	K18	LVDD	T10	MA17
A5	SYS_OSCO	D18	LVDD	K19	CFSTSCHG#	T11	MA15
A6	SYS_OSCI	D19	GPIOA1	L1	Vss	T12	MA11
A7	RTCVDD	E1	LVDD	L2	GPIOJ6	T13	MA7
A8	PLLVSS	E2	Vss	L3	GPIOJ3	T14	Vss
A9	SYSVCP	E3	SDA6	L4	GPIOJ5	T15	MA2
A10	SYSCLKI	E4	VSS	L5	LVDD	T16	MCS1#
A11	LVDD	E5	SDA0	L15	HVDD	T17	MD13
A12	Vss	E6	GPIOK5	L16	CFWAIT#	T18	MD12
A13	UXVDD	E7	GPIOK4	L17	CFRST	T19	MD10
A14	DP	E8	Vss	L18	Vss	U1	SDD0
A15	DM	E9	CM1DATA0	L19	CFIRQ	U2	SDDQM1#
A16	UVDD3	E10	TESTEN1	M1	GPIOJ0	U3	CM2HREF
A17	R1	E11	Vss	M2	GPIOJ2	U4	Vss
A18	UPVDD	E12	LVDD	M3	GPIOJ1	U5	CM2DATA4
A19	NC	E13	TDI	M4	SDD15	U6	AVSS
B1	SDA13	E14	GPIOB3	M5	Vss	U7	CM2DATA5
B2	SDA14	E15	GPIOB1	M15	CFCE2#	U8	CM2DATA7
B3	CM1HREF	E16	GPIOB0	M16	MD3	U9	GPIOD3

3. PIN

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
B4	Vss	E17	GPIOA3	M17	CFCE1#	U10	GPIOD0
B5	CM1DATA3	E18	HVDD	M18	CFIORD#	U11	MA16
B6	CM1DATA6	E19	GPIOA5	M19	CFIOWR#	U12	MA12
B7	Vss	F1	SDA1	N1	SDD11	U13	MA8
B8	BUP#	F2	SDVDD	N2	Vss	U14	MA4
B9	PLLVDD	F3	SDA5	N3	SDD13	U15	MA1
B10	SYSCKSEL	F4	SDA2	N4	SDD14	U16	Vss
B11	TRST#	F5	GPIOK1	N5	SDD12	U17	MBEL#
B12	TESTCK	F15	LVDD	N15	MD5	U18	MD14
B13	XVSS	F16	GPIOB2	N16	MD1	U19	MD15
B14	UVDD3	F17	GPIOA6	N17	Vss	V1	GPIOD6
B15	UVSS	F18	LVDD	N18	MD2	V2	Vss
B16	Vss	F19	GPIOA7	N19	MD0	V3	CM2VREF
B17	UVSS	G1	GPIOK7	P1	SDVDD	V4	CM2DATA0
B18	PVSS	G2	GPIOK6	P2	SDD10	V5	AVDD
B19	USBCK_OSCO	G3	GPIOK3	P3	SDD9	V6	ADIN7
C1	SDA11	G4	GPIOK0	P4	SDD8	V7	ADIN5
C2	Vss	G5	Vss	P5	Vss	V8	ADIN3
C3	SDVDD	G15	GPIOC1	P15	MD9	V9	ADIN1
C4	SDA12	G16	HVDD	P16	MD8	V10	AVSS
C5	CM1VREF	G17	GPIOB5	P17	MD6	V11	MA19
C6	CM1DATA5	G18	Vss	P18	MD4	V12	MA14
C7	CM1DATA7	G19	GPIOB4	P19	LVDD	V13	Vss
C8	Vss	H1	LVDD	R1	SDD6	V14	MA3
C9	HVDD	H2	Vss	R2	SDD5	V15	GPIOD5
C10	Vss	H3	GPIOK2	R3	SDD4	V16	MCS0#
C11	TESTEN0	H4	SDWE#	R4	SDD7	V17	MWE#
C12	TDO	H5	SDCAS#	R5	SDD3	V18	LVDD
C13	Vss	H15	GPIOC6	R6	SDD2	V19	Vss
C14	VBUS	H16	GPIOC0	R7	CM2DATA1	W1	NC
C15	Vss	H17	GPIOB6	R8	CM2DATA6	W2	CM2CLKIN
C16	HVDD	H18	HVDD	R9	Vss	W3	CM2CLKOUT
C17	Vss	H19	GPIOB7	R10	HVDD	W4	C2VDD
C18	Vss	J1	SDCS1#	R11	MA13	W5	AVDD
C19	USBCK_OSCI	J2	SDCS0#	R12	MA9	W6	ADIN6
D1	SDA8	J3	SDRAS#	R13	MA6	W7	ADIN4
D2	SDA9	J4	GPIOJ7	R14	MA5	W8	ADIN2
D3	SDA7	J5	GPIOJ4	R15	HVDD	W9	ADIN0
D4	SDA10	J15	GPIOC7	R16	GPIOD4	W10	AVDD
D5	SDA4	J16	GPIOC5	R17	MD11	W11	MA18
D6	SDA3	J17	GPIOC4	R18	Vss	W12	HVDD
D7	CM1DATA1	J18	GPIOC3	R19	MD7	W13	MA10
D8	CM1DATA2	J19	GPIOC2	T1	Vss	W14	LVDD
D9	CM1DATA4	K1	SDCLK	T2	LVDD	W15	HVDD
D10	LVDD	K2	SDCLKEN	T3	SDD1	W16	MA0
D11	TCK	K3	SDVDD	T4	GPIOD7	W17	MOE#
D12	TMS	K4	SDVDD	T5	SDDQM0#	W18	MBEH#
D13	RESET#	K5	Vss	T6	CM2DATA3	W19	NC

Note: # at the right end of pin name indicates to be an active low signal.

3.2 Pin Description

＃ : # at the right end of pin name indicates to be an active low signal.
I : Input pin
O : Output pin
IO : Bi-directional pin
P : Power supply

Table 3.1 Cell Type Description

Cell Type	Description	Example of Pin being Used	
		Pin Name	Power Supply
ILS	Low Voltage LVCMOS Schmitt input	BUP#	RTCVDD
ICD1	LVCMOS input with pull-down resistor (50kΩ@3.3V)	TESTEN[1:0],TESTCK	HVDD
ICU1	LVCMOS input with pull-up resistor (50kΩ@3.3V)	TMS, TDI	HVDD
ICSU1	LVCMOS Schmitt input with pull-up resistor (50kΩ@3.3V)	TRST#,TCK,RESET#	HVDD
ICSD1	LVCMOS Schmitt input with pull-down resistor (50kΩ@3.3V)	SYSCKSEL	HVDD
ILTR	Low Voltage Transparent Input	SYS_OSCI	RTCVDD
		USBCK_OSCI	UPVDD
		R1	UVDD3
IHTR	High Voltage Transparent Input	ADIN[7:0]	ADVDD
OLTR	Low Voltage Transpatent Output	SYS_OSCO,SYSVCP	PLLVDD
		USBCK_OSCO	UPVDD
BLNC4U1	Low noise LVCMOS IO buffer with pull-up resistor (50kΩ@3.3V) (±4mA)	CF I/F	HVDD
BLNC4D2	Low noise LVCMOS IO buffer with pull-down resistor (100kΩ@3.3V) (±4mA)	MD [15:0]	HVDD
		SDD[15:0],GPIOJ,GPIOK	SDVDD
BLNS4	Low noise LVCMOS Schmitt IO buffer (±4mA)	GPIOA, GPIOB, GPIOC,GPIOD, SYSCLKI	HVDD
BLNS4D1	Low noise LVCMOS Schmitt IO buffer with pull-down resistor (50kΩ@3.3V) (±4mA)	Camera I/F	C1VDD, C2VDD
OLN4	Low noise output buffer (±4mA)	SRAM Device I/F (excluding MD)	HVDD
		SDRAM I/F (excluding SDD[15:0])	SDVDD
OTLN4	Low noise Tri-state output buffer (±4mA)	TDO	HVDD
USBDM	USB DM buffer	DM	UVDD3
USBDP	USB DP buffer	DP	UVDD3
USBVBUS	USB VBUS output buffer	VBUS	UVDD3

Table 3.2 Pin Description

Pin Name	Type	Cell Type	Pin No.	Description
(MA [23:20])	(I/O)	(BLNS4)		For information on these pins, see the description of **GPIOD [3:0]**.
MA [19:12]	O	OLN4	V11,W11, T10,U11, T11,V12, R11,U12	**Address Output Signal for Flash-ROM/SRAM [19:12]**
MA 11	O	OLN4	T12	This pin has the following functions: • **MA11: Address output signal for Flash-ROM/SRAM [11]** (Pin function right after reset) • **CFREG# Output Signal** When the compact flash (CF) interface is in operation, this signal functions as the REG signal selecting attribute of the CF interface and I/O space.
MA [10:0]	O	OLN4	W13,R12, U13,T13, R13,R14, U14,V14, T15,U15, W16	These pins have the following functions: • **MA [10:0]: Address output signal for Flash-ROM/SRAM [10:0]** (Pin function right after reset) • **CFADDR [10:0] Output signal** When the CF interface is in operation, this signal becomes the CF interface address signal [10:0].
MBEL#	I/O	OLN4	U17	Data bus low byte enable output for **Flash-ROM/SRAM**
MBEH#	I/O	OLN4	W18	Data bus high byte enable output for **Flash-ROM/SRAM**
MD [15:0]	I/O	BLNC4D2	U19,U18, T17,T18, R17,T19, P15,P16, R19,P17, N15,P18, M16,N18, N16,N19	These pins have the following functions: • **16-bit data bus for Flash-ROM/SRAM** (Pin function right after reset) • **When the CF interface is in operation, this pin becomes good for 16-bit data.** • **MODESEL[15:0]** Sampled to determine the internal operation mode, at power-on resetting (RESET# transition from Low to High). For details, see section “4.1 System Configuration”. Here, to determine the operation mode, a pull-up resistance may be required externally. (Resistance in the range from 4.7 to 10kΩ)
MCS [3:2]#	O	(BLNS4)	V15,R16	For information on these pins, see the description of **GPIOD [5:4]**.
MCS [1:0]#	O	OLN4	T16,V16	**Chip select signal for Flash-ROM/SRAM [1:0]** (Active low signal)
MOE#	O	OLN4	W17	This pin has the following functions: (Active low signal) • **MOE#: Strobe signal for Flash-ROM/SRAM** (Pin function right after reset) • **CFOE# output signal** When the CF interface is in operation, this signal becomes the output enable signal of CF interface memory and attribute spaces.

Pin Name	Type	Cell Type	Pin No.	Description
MWE#	O	OLN4	V17	This pin has the following functions: (Active low signal) • **MWE#: Write enable signal for Flash-ROM/SRAM** (for static memory) (Pin function right after reset) • **CFWE# output signal** When the CF interface is in operation, this signal becomes the write enable signal of CF interface memory and attribute spaces.
SDA [14:0]	O	OLN4	B2,B1, C4,C1, D4,D2, D1,D3, E3,F3, D5,D6, F4,F1, E5	**Address output for SDRAM [14:0]**
SDD[31:16]	(I/O)	(BLNC4D2)		For information on these pins, see the descriptions of **GPIOK [7:0] and GPIOJ [7:0]**.
SDD [15:0]	I/O	BLNC4D2	M4,N4, N3,N5, N1,P2, P3,P4, R4,R1, R2,R3, R5,R6, T3,U1	**Data I/O for SDRAM [15:0]**
SDWE#	O	OLN4	H4	**Write enable signal for SDRAM**
SDCLK	O	OLN4	K1	**Outputting clock for SDRAM** The same frequency as internal operation frequency (CPUCLK) is output.
SDCLKEN	O	OLN4	K2	**Clock enable signal for SDRAM**
SDRAS#	O	OLN4	J3	**RAS signal for SDRAM** (Active low signal)
SDCAS#	O	OLN4	H5	**CAS signal for SDRAM** (Active low signal)
SDCS[1:0]#	O	OLN4	J1,J2	**Chip select signal for SDRAM** (Active low signal)
SDDQM[3:2]#	(I/O)	(BLNS4)		For information on these pins, see the description of **GPIOD [7:6]**.
SDDQM[1:0]#	O	OLN4	U2,T5	**DQM signal for SDRAM** (Active low signal) SDDQM0# corresponds to the lower bytes; SDDQM1#, to the higher bytes.
CM1DATA[7:0]	I/O	BLNS4D1	C7,B6, C6,D9, B5,D8, D7,E9	These pins have the following functions: • **CM1DATA [7:0]: Camera 1YUV data input** This pin, when reset, becomes good for inputting GPIOE [7:0]. To use as the CM1DATA [7:0] pin, set bits [15:0] of the GPIOE pin function register to “Function 1 other than GPIO.” • **GPIOE [7:0] I/O** (Pin function right after reset)
CM1VREF	I/O	BLNS4D1	C5	This pin has the following functions: • **CM1VREF: Vertical sync input at camera 1 data input** This pin, when reset, becomes good for inputting GPIOF0. To use as the CM1VREF pin, set bits [1:0] of the GPIOF pin function register to “Function 1 of other than GPIO.” • **GPIOF0 I/O** (Pin function right after reset)

Pin Name	Type	Cell Type	Pin No.	Description
CM1HREF	I/O	BLNS4D1	B3	This pin has the following functions: • **CM1HREF: Horizontal sync input at camera 1 data input** This pin, when reset, becomes good for inputting GPIOF1. To use as the CM1HREF pin, set bits [3:2] of the GPIOF pin function register to "Function 1 of other than GPIO." • **GPIOF1 I/O** (Pin function right after reset)
CM1CLKOUT	I/O	BLNS4D1	A3	This pin has the following functions: • **CM1CLKOUT: Basic clock output for camera 1** This pin, when reset, becomes good for inputting GPIOF2. To use as the CM1CLKOUT pin, set bits [5:4] of the GPIOF pin function register to "Function 1 of other than GPIO." • **GPIOF2 I/O** (Pin function right after reset)
CM1CLKIN	I/O	BLNS4D1	A2	This pin has the following functions: • **CM1CLKIN: Pixel clock for inputting camera 1 data** This pin, when reset, becomes good for inputting GPIOF3. To use as the CM1CLKIN pin, set bits [7:6] of the GPIOF pin function register to "Function 1 of other than GPIO." • **GPIOF3 I/O** (Pin function right after reset)
CM2DATA[7:0]	I/O	BLNS4D1	U8,R8 U7,U5 T6,T7, R7,V4	These pins have the following functions: • **CM2DATA [7:0]: Camera 2YUV data input** This pin, when reset, becomes good for inputting GPIOG [7:0]. To use as the CM2DATA [7:0] pin, set bits [15:0] of the GPIOG pin function register to "Function 1 other than GPIO." • **GPIOG [7:0] I/O** (Pin function right after reset)
CM2VREF	I/O	BLNS4D1	V3	This pin has the following functions: • **CM2VREF: Vertical sync input at camera 2 data input** This pin, when reset, becomes good for inputting GPIOF4. To use as the CM2VREF pin, set bits [9:8] of the GPIOF pin function register to "Function 1 of other than GPIO." • **GPIOF4 I/O** (Pin function right after reset)
CM2HREF	I/O	BLNS4D1	U3	This pin has the following functions: • **CM2HREF: Horizontal sync input at camera 2 data input** This pin, when reset, becomes good for inputting GPIOF5. To use as the CM2HREF pin, set bits [11:10] of the GPIOF pin function register to "Function 1 other than GPIO." • **GPIOF5 I/O** (Pin function right after reset)
CM2CLKOUT	I/O	BLNS4D1	W3	This pin has the following functions: • **CM2CLKOUT: Basic clock output for camera 2** This pin, when reset, becomes good for inputting GPIOF6. To use as the CM2CLKOUT pin, set bits [13:12] of the GPIOF pin function register to "Function 1 other than GPIO." • **GPIOF6 I/O** (Pin function right after reset)

Pin Name	Type	Cell Type	Pin No.	Description
CM2CLKIN	I/O	BLNS4D1	W2	This pin has the following functions: • **CM2CLKIN: Pixel clock for inputting camera 2 data** This pin, when reset, becomes good for inputting GPIOF7. To use as the CM2CLKIN pin, set bits [15:14] of the GPIOF pin function register to "Function 1 other than GPIO." • **GPIOF7 I/O** (Pin function right after reset)
CFCE2#	I/O	BLNC4U1	M15	This pin has the following functions: • **CFCE2#: Card enable 2 (CE2#) output** (active low signal) **for Compact Flash Memory Interface (hereafter referred to as CF)** This pin, when reset, becomes good for inputting GPIOH0. To use as the CFCE2# pin, set bits [1:0] of the GPIOH pin function register to "Function 1 of other than GPIO." • **GPIOH0 I/O** (Pin function right after reset) • **SDMDATA0: Data I/O 0 for SD card** (Pin Function 2 of other than GPIO)
CFCE1#	I/O	BLNC4U1	M17	This pin has the following functions: • **CFCE1#: Card Enable 1 (CE1#) Output for CF** (Active low signal) This pin, when reset, becomes good for inputting GPIOH1. To use as the CFCE1# pin, set bits [3:2] of the GPIOH pin function register to "Function 1 of other than GPIO." • **GPIOH1 I/O** (Pin function right after reset) • **SDMDATA1: Data I/O 1 for SD card** (Pin Function 2 of other than GPIO)
CFIORD#	I/O	BLNC4U1	M18	This pin has the following functions: • **CFIORD#: IO read strobe output for CF** (Active low signal) This pin, when reset, becomes good for inputting GPIOH2. To use as the CFIORD# pin, set bits [5:4] of the GPIOH pin function register to "Function 1 other than GPIO." • **GPIOH2 I/O** (Pin function right after reset) • **SDMDATA2: Data I/O 2 for SD card** (Pin Function 2 of other than GPIO)
CFIOWR#	I/O	BLNC4U1	M19	This pin has the following functions: (Active low signal) • **CFIOWR#: IO write strobe output for CF** This pin, when reset, becomes good for inputting GPIOH3. To use as the CFIOWR# pin, set bits [7:6] of the GPIOH pin function register to "Function 1 other than GPIO." • **GPIOH3 I/O** (Pin function right after reset) • **SDMDATA3: Data I/O 3 for SD card** (Pin Function 2 of other than GPIO)

Pin Name	Type	Cell Type	Pin No.	Description
CFWAIT#	I/O	BLNC4U1	L16	This pin has the following functions: • **CFWAIT#: Wait request input for CF** (active low signal) This pin, when reset, becomes good for inputting GPIOH4. To use as the CFWAIT# pin, set bits [9:8] of the GPIOH pin function register to "Function 1 other than GPIO." • **GPIOH4 I/O** (Pin function right after reset) • **SDMCMD: Command I/O for SD card** (Function 2 of other than GPIO)
CFRST	I/O	BLNC4U1	L17	This pin has the following functions: • **CFRST: Reset signal to CF card** The signal is HIGH when the card is reset and LOW when the card is under normal operation. This pin, when reset, becomes good for inputting GPIOH5. To use as the CFRST pin, set bits [11:10] of the GPIOH pin function register to "Function 1 other than GPIO." • **GPIOH5 I/O** (Pin function right after reset) • **SDMCLK: Clock output for SD card** (Function 2 of other than GPIO)
CFIRQ	I/O	BLNC4U1	L19	This pin has the following functions: • **CFIRQ: Interrupt request signal from CF card** This pin, when reset, becomes good for inputting GPIOH6. To use as the CFIRQ pin, set bits [13:12] of the GPIOH pin function register to "Function 1 other than GPIO." • **GPIOH6 I/O** (Pin function right after reset) • **SDMCD#: Card detect input for SD card** (Function 2 of other than GPIO)
CFSTSCHG#	I/O	BLNC4U1	K19	This pin has the following functions: • **CFSTSCHG#: Status change signal from CF card** (active low signal) This pin, when reset, becomes good for inputting GPIOH7. To use as the CFSTSCHG# pin, set bits [15:14] of the GPIOH pin function register to "Function 1 other than GPIO." • **GPIOH7 I/O** (Pin function right after reset) • **SDMWP: Write protect input for SD card** (Function 2 of other than GPIO)
CFDEN#	I/O	BLNC4U1	K16	This pin has the following functions: • **CFDEN#: Data bus enable signal for external buffer of CF card** (active low signal) This pin, when reset, becomes good for inputting GPIOI0. To use as the CFDEN# pin, set bits [1:0] of the GPIOI pin function register to "Function 1 other than GPIO." • **GPIOI0 I/O pin** (Pin function right after reset) • **SDMGPO: General-purpose output for SD card** (Function 2 of other than GPIO)

Pin Name	Type	Cell Type	Pin No.	Description
CFDDIR	I/O	BLNC4U1	K17	This pin has the following functions: • **CFDDIR: Data bus directional instruction output for CF** When CF data is read, this pin becomes Low. Also, this pin, when reset, becomes good for inputting GPIOI1. To use as the CFDDIR pin, set bits [3:2] of the GPIOI pin function register to "Function 1 other than GPIO." • **GPIOI1 I/O** (Pin function right after reset)
R1	I	ILTR	A17	USB Device internal operation setting pin
DM	I/O	USBDM	A15	USB Device D- I/O
DP	I/O	USBDP	A14	USB Device D+ I/O
VBUS	I	USBVBUS	C14	USB Device VBUS input
ADIN[7:0]	I	IHTR	V6,W6, V7,W7, V8,W8, V9,W9	Analog signal input
GPIOA0	I/O	BLNS4	D17	This pin has the following functions: • **GPIOA0 I/O** (Pin function right after reset) • **TXD1: UART1 transmit data output** (Function 1 of other than GPIO)
GPIOA1	I/O	BLNS4	D19	This pin has the following functions: • **GPIOA1 I/O** (Pin function right after reset) • **RXD1: UART1 receive data input** (Function 1 of other than GPIO)
GPIOA2	I/O	BLNS4	D15	This pin has the following functions: • **GPIOA2 I/O** (Pin function right after reset) • **RTS1: UART1 request to send output** (Function 1 of other than GPIO) • **I^2S1_WS: Word select for I^2S1** (Function 2 of other than GPIO)
GPIOA3	I/O	BLNS4	E17	This pin has the following functions: • **GPIOA3 I/O** (Pin function right after reset) • **CTS1: UART1 clear to send input** (Function 1 of other than GPIO) • **I^2S1_SCK: Serial clock for I^2S1** (Function 2 of other than GPIO)
GPIOA4	I/O	BLNS4	D16	This pin has the following functions: • **GPIOA4 I/O** (Pin function right after reset) • **TXD2: UART2 transmit data output** (Function 1 of other than GPIO)
GPIOA5	I/O	BLNS4	E19	This pin has the following functions: • **GPIOA5 I/O** (Pin function right after reset) • **RXD2: UART2 receive data input** (Function 1 of other than GPIO)
GPIOA6	I/O	BLNS4	F17	This pin has the following functions: • **GPIOA6 I/O** (Pin function right after reset) • **RTS2: UART2 request to send output** (Function 1 of other than GPIO) • **SCL: I^2C clock I/O** (Function 2 of other than GPIO)
GPIOA7	I/O	BLNS4	F19	This pin has the following functions: • **GPIOA7 I/O** (Pin function right after reset) • **CTS2: UART2 clear to receive input** (Function 1 of other than GPIO) • **SDA: I^2C data I/O** (Function 2 of other than GPIO)

Pin Name	Type	Cell Type	Pin No.	Description
GPIOB0	I/O	BLNS4	E16	This pin has the following functions: • **GPIOB0 I/O** (Pin function right after reset) • **INT0 input** • **I^2S0_WS: Word select for I^2S0** (Function 1 of other than GPIO)
GPIOB1	I/O	BLNS4	E15	This pin has the following functions: • **GPIOB1 I/O** (Pin function right after reset) • **INT1 input** • **I^2S0_SCK: Serial clock for I^2S0** (Function 1 of other than GPIO)
GPIOB2	I/O	BLNS4	F16	This pin has the following functions: • **GPIOB2 I/O** (Pin function right after reset) • **INT2 input** • **I^2S0_SD: Serial data for I^2S0** (Function 1 of other than GPIO)
GPIOB3	I/O	BLNS4	E14	This pin has the following functions: • **GPIOB3 I/O** (Pin function right after reset) • **INT3 input** • **I^2S1_SD: Serial data for I^2S1** (Function 1 of other than GPIO)
GPIOB4	I/O	BLNS4	G19	This pin has the following functions: • **GPIOB4 I/O** (Pin function right after reset) • **INT4 input** • **TimerA0Out** (Function 1 other than GPIO)
GPIOB5	I/O	BLNS4	G17	This pin has the following functions: • **GPIOB5 I/O** (Pin function right after reset) • **INT5 input** • **TimerA1Out** (Function 1 other than GPIO)
GPIOB6	I/O	BLNS4	H17	This pin has the following functions: • **GPIOB6 I/O** (Pin function right after reset) • **INT6 input** • **TimerA2Out** (Function 1 other than GPIO)
GPIOB7	I/O	BLNS4	H19	This pin has the following functions: • **GPIOB7 I/O** (Pin function right after reset) • **INT7 input** • **TimerBIn** (Function 1 other than GPIO)
GPIOC0	I/O	BLNS4	H16	This pin has the following functions: • **GPIOC0 I/O** (Pin function right after reset) • **TimerB0IO** (Function 1 other than GPIO)
GPIOC1	I/O	BLNS4	G15	This pin has the following functions: • **GPIOC1 I/O** (Pin function right after reset) • **TimerB1IO** (Function 1 other than GPIO)
GPIOC2	I/O	BLNS4	J19	This pin has the following functions: • **GPIOC2 I/O** (Pin function right after reset) • **TimerB2IO** (Function 1 other than GPIO)
GPIOC3	I/O	BLNS4	J18	This pin has the following functions: • **GPIOC3 I/O** (Pin function right after reset) • **TimerB3IO** (Function 1 other than GPIO) • **UART3_CLK** (Function 2 of other than GPIO)
GPIOC4	I/O	BLNS4	J17	This pin has the following functions: • **GPIOC4 I/O** (Pin function right after reset) • **SPI_SS: Chip select for SPI** (Function 1 of other than GPIO) • **TXD3: UART3 transmit data output** (Function 2 of other than GPIO)

Pin Name	Type	Cell Type	Pin No.	Description
GPIOC5	I/O	BLNS4	J16	This pin has the following functions: • **GPIOC5 I/O** (Pin function right after reset) • **SPI_SCLK: Serial clock for SPI** (Function 1 of other than GPIO) • **RXD3: UART3 receive data input** (Function 2 of other than GPIO)
GPIOC6	I/O	BLNS4	H15	This pin has the following functions: • **GPIOC6 I/O** (Pin function right after reset) • **SPI_MISO: Serial data master input/slave output for SPI** (Function 1 of other than GPIO) • **RTS3: UART3 request to send output** (Function 2 of other than GPIO)
GPIOC7	I/O	BLNS4	J15	This pin has the following functions: • **GPIOC6 I/O** (Pin function right after reset) • **SPI_MOSI: Serial data master output/slave input for SPI** (Function 1 of other than GPIO) • **CTS3: UART3 clear to receive input** (Function 2 of other than GPIO)
GPIOD[3:0]	I/O	BLNS4	U9,T8, T9,U10	This pin has the following functions: • **GPIOD [3:0] I/O** (Pin function right after reset) • **MA [23:20]: Address output signal [23:20]** (Function 1 of other than GPIO)
GPIOD[5:4]	I/O	BLNS4	V15,R16	This pin has the following functions: • **GPIOD [5:4] I/O** (Pin function right after reset) • **MCS [3:2]#: Chip select output signal for memory** (Function 1 of other than GPIO)
GPIOD6	I/O	BLNS4	V1	This pin has the following functions: • **GPIOD6 I/O** (Pin function right after reset) • **DQM2# signal for SDRAM** (active low signal) • **SDDQM2#: Corresponds to the lower byte of the higher 16 bits of SDRAM 32-bit data width** (Function 1 of other than GPIO)
GPIOD7	I/O	BLNS4	T4	This pin has the following functions: • **GPIOD7 I/O** (Pin function right after reset) • **DQM3# signal for SDRAM** (active low signal) • **SDDQM3##: Corresponds to the higher byte of the higher 16 bits of SDRAM 32-bit data width** (Function 1 of other than GPIO)
GPIOJ[7:0]	I/O	BLNC4D2	J4,L2, L4,J5, L3,M2, M3,M1	This pin has the following functions: • **GPIOJ [7:0] I/O** (Pin function right after reset) • **SDD [23:16]: Data I/O for SDRAM** (Function 1 of other than GPIO)
GPIOK[7:0]	I/O	BLNC4D2	G1,G2, E6,E7, G3,H3, F5,G4	This pin has the following functions: • **GPIOK [7:0] I/O** (Pin function right after reset) • **SDD [31:24]: Data I/O for SDRAM** (Function 1 of other than GPIO)
SYSCLKI	I/O	BLNS4	A10	**32KHz system clock input** Basic clock input when SYSCKSEL is “HIGH”. 1/4 of system clock or 32KHz becomes output by setting it when SYSCKSEL is “LOW”.
SYS_OSCI	I	ILTR	A6	**Connection pin for crystal transducer** This is an operation clock oscillator pin. It connects a 32KHz crystal transducer.
SYS_OSCO	O	OLTR	A5	**Connection pin for crystal transducer** This is an operation clock oscillator pin. It connects a 32KHz crystal transducer.

3. PIN

Pin Name	Type	Cell Type	Pin No.	Description
SYSVCP	O	OLTR	A9	**System test pin for built-in PLL** This pin is used to monitor outputs of the PLL at the time of system test. Make the pin open at the time of normal operation.
SYSCKSEL	I	ICSD1	B10	**32KHz system clock input crystal transducer/oscillator select signal** Crystal oscillator is used when SYSCKSEL is "HIGH". Crystal transducer is used when SYSCKSEL is "LOW".
USBCK_OSCI	I	ILTR	C19	**Connection pin for crystal transducer** This is an operation clock oscillator pin specifically for USB. It connects a 12/24MHz crystal transducer.
USBCK_OSCO	O	OLTR	B19	**Connection pin for crystal transducer** This is an operation clock oscillator pin specifically for USB. It connects a 12/24MHz crystal transducer.
TRST#	I	ICSU1	B11	**Resetting for JTAG Interface** (active low signal) This signal is to be input with a Schmitt trigger with pull-up resistor.
TCK	I	ICSU1	D11	**Clock Input Pin for JTAG Interface** This clock is to be input with a Schmitt trigger.
TMS	I	ICU1	D12	**TMS Pin for JTAG Interface** This pin has a built-in pull-up resistor.
TDI	I	ICU1	E13	**Serial Data Input Pin for JTAG Interface** This pin has a built-in pull-up resistor.
TDO	O	OTLN4	C12	**Serial Data Output Pin for JTAG Interface**
TESTEN0	I	ICD1	C11	**Test enable 0 (active high signal)** This pin has a built-in pull-down resistor. Connect this pin to Vss or make it open at the time of normal operation.
TESTEN1	I	ICD1	E10	**Test enable 1** (active high signal) This pin has a built-in pull-down resistor. Connect this pin to Vss or make it open at the time of normal operation.
TESTCK	I	ICD1	B12	**Test clock** This pin has a built-in pull-down resistor. Connect this pin to Vss or make it open at the time of normal operation.
RESET#	I	ICSU1	D13	**System reset signal** (active low signal) Even after HVDD and LVDD become stable, keep RESET# active (LOW) for 100ms.
BUP#	I	ILS	B8	**Backup signal** (active low signal)
HVDD	P	P	C9,C16, D14,E18, G16,H18, L15,R10, R15,W12, W15	**Power supply for I/O cell :** 3.3V (Typical) 3.0V (Min.) - 3.6V (Max.)
C1VDD	P	P	A4	**Power supply for camera 1 interface:** 3.0 (Typical) 2.4V (Min.) - 3.6V (Max.)
C2VDD	P	P	W4	**Power supply for camera 2 interface:** 3.0 (Typical) 2.4V (Min.) - 3.6V (Max.)

Pin Name	Type	Cell Type	Pin No.	Description
SDVDD	P	P	C3,F2, K3,K4, P1	**Power supply for SDRAM:** 3.3V (Typical) 2.7V (Min.) - 3.6V (Max.)
AVDD	P	P	V5,W5, W10	**Power supply for A/D C:** 3.3V (Typical) 3.0V (Min.) - 3.6V (Max.)
UVDD3	P	P	A16,B14	**Power supply for USB:** 3.3V (Typical) 3.0V (Min.) - 3.6V (Max.)
LVDD	P	P	A11,D10, D18,E1, E12,F15, F18,H1, K15,K18, L5,P19, T2,V18, W14	**Power supply for core (internal):** 1.8V (Typical) 1.65V (Min.) - 1.95V (Max.)
UPVDD	P	P	A18	**Power supply for USB:** 1.8V (Typical) 1.65V (Min.) - 1.95V (Max.)
UXVDD	P	P	A13	**Power supply for USB:** 1.8V (Typical) 1.65V (Min.) - 1.95V (Max.)
PLLVDD	P	P	B9	**Power supply for analog (PLL):** 1.8V (Typical) 1.65V (Min.) - 1.95V (Max.) Handling as an analog power supply is required. Supply stable power that generates less noise.
RTCVDD	P	P	A7	**Power supply for RTC:** 1.8V (Typical) 1.65V (Min) – 1.95V (Max)
UVSS	P	P	B15,B17	**Ground for USB**
PVSS	P	P	B18	**Ground for USB**
XVSS	P	P	B13	**Ground for USB**
AVSS	P	P	U6,V10	**Ground for A/D C**
PLLVSS	P	P	A8	**Ground for analog (PLL)** Handling as an analog power supply is required. Supply stable ground that generates less noise.
Vss	P	P	A12,B4, B7,B16, C2,C8, C10,C13, C15,C17, C18,E2, E4,E8, E11,G5, G18,H2, K5,L1, L18,M5, N2,N17, P5,R9, R18,T1, T14,U4, U16,V2, V13,V19	**Ground common to I/O cell, camera interface, and core power supplies**

3.3 Multiplex Pin Function of GPIO Pins, Pin Function Right after Reset

S2S65A00 Pin Name	Pin Function Right after Reset	GPIO	INT	FlashROM/ SAM Extension	SDRAM Extension	UART	I^2C/ I^2S/ SPI	Timer	Camera Interface	CF Card	SD Card
GPIOA0	GPIOA0	GPIOA0				TXD1					
GPIOA1	GPIOA1	GPIOA1				RXD1					
GPIOA2	GPIOA2	GPIOA2				RTS1	I^2S1_WS				
GPIOA3	GPIOA3	GPIOA3				CTS1	I^2S1_SCK				
GPIOA4	GPIOA4	GPIOA4				TXD2					
GPIOA5	GPIOA5	GPIOA5				RXD2					
GPIOA6	GPIOA6	GPIOA6				RTS2	SCL				
GPIOA7	GPIOA7	GPIOA7				CTS2	SDA				
GPIOB0	GPIOB0	GPIOB0	INT0				I^2S0_WS				
GPIOB1	GPIOB1	GPIOB1	INT1				I^2S0_SCK				
GPIOB2	GPIOB2	GPIOB2	INT2				I^2S0_SD				
GPIOB3	GPIOB3	GPIOB3	INT3				I^2S1_SD				
GPIOB4	GPIOB4	GPIOB4	INT4					TimerA0out			
GPIOB5	GPIOB5	GPIOB5	INT5					TimerA1out			
GPIOB6	GPIOB6	GPIOB6	INT6					TimerA2out			
GPIOB7	GPIOB7	GPIOB7	INT7					TimerBIN			
GPIOC0	GPIOC0	GPIOC0						TimerB0IO			
GPIOC1	GPIOC1	GPIOC1						TimerB1IO			
GPIOC2	GPIOC2	GPIOC2						TimerB2IO			
GPIOC3	GPIOC3	GPIOC3				UART3_CLK		TimerB3IO			
GPIOC4	GPIOC4	GPIOC4				TXD3	SPI_SS				
GPIOC5	GPIOC5	GPIOC5				RXD3	SPI_SCLK				
GPIOC6	GPIOC6	GPIOC6				RTS3	SPI_MISO				
GPIOC7	GPIOC7	GPIOC7				CTS3	SPI_MOSI				
GPIOD0	GPIOD0	GPIOD0		MA20							
GPIOD1	GPIOD1	GPIOD1		MA21							
GPIOD2	GPIOD2	GPIOD2		MA22							
GPIOD3	GPIOD3	GPIOD3		MA23							
GPIOD4	GPIOD4	GPIOD4		MCS2#							
GPIOD5	GPIOD5	GPIOD5		MCS3#							
GPIOD6	GPIOD6	GPIOD6			SDDQM2#						
GPIOD7	GPIOD7	GPIOD7			SDDQM3#						
CM1DATA0	GPIOE0	GPIOE0							CM1DATA0		
CM1DATA1	GPIOE1	GPIOE1							CM1DATA1		
CM1DATA2	GPIOE2	GPIOE2							CM1DATA2		
CM1DATA3	GPIOE3	GPIOE3							CM1DATA3		
CM1DATA4	GPIOE4	GPIOE4							CM1DATA4		
CM1DATA5	GPIOE5	GPIOE5							CM1DATA5		
CM1DATA6	GPIOE6	GPIOE6							CM1DATA6		
CM1DATA7	GPIOE7	GPIOE7							CM1DATA7		
CM1VREF	GPIOF0	GPIOF0							CM1VREF		
CM1HREF	GPIOF1	GPIOF1							CM1HREF		
CM1CLKOUT	GPIOF2	GPIOF2							CM1CLKOUT		
CM1CLKIN	GPIOF3	GPIOF3							CM1CLKIN		
CM2VREF	GPIOF4	GPIOF4							CM2VREF		
CM2HREF	GPIOF5	GPIOF5							CM2HREF		
CM2CLKOUT	GPIOF6	GPIOF6							CM2CLKOUT		
CM2CLKIN	GPIOF7	GPIOF7							CM2CLKIN		
CM2DATA0	GPIOG0	GPIOG0							CM2DATA0		
CM2DATA1	GPIOG1	GPIOG1							CM2DATA1		
CM2DATA2	GPIOG2	GPIOG2							CM2DATA2		
CM2DATA3	GPIOG3	GPIOG3							CM2DATA3		
CM2DATA4	GPIOG4	GPIOG4							CM2DATA4		
CM2DATA5	GPIOG5	GPIOG5							CM2DATA5		
CM2DATA6	GPIOG6	GPIOG6							CM2DATA6		
CM2DATA7	GPIOG7	GPIOG7							CM2DATA7		

S2S65A00 Pin Name	Pin Function Right after Reset	GPIO	INT	FlashROM/ SAM Extension	SDRAM Extension	UART/ SPI	I^2C/ I^2S	Timer	Camera Interface	CF Card	SD Card
CFCE2#	GPIOH0	GPIOH0								CFCE2#	SDMDATA0
CFCE1#	GPIOH1	GPIOH1								CFCE1#	SDMDATA1
CFIORD#	GPIOH2	GPIOH2								CFIORD#	SDMDATA2
CFIOWR#	GPIOH3	GPIOH3								CFIOWR#	SDMDATA3
CFWAIT#	GPIOH4	GPIOH4								CFWAIT#	SDMCMD
CFRST	GPIOH5	GPIOH5								CFRST	SDMCLK
CFIRQ	GPIOH6	GPIOH6								CFIRQ	SDMCD#
CFSTSCHG#	GPIOH7	GPIOH7								CFSTSCHG#	SDMWP
CFDEN#	GPIOI0	GPIOI0								CFDEN#	SDMGPO
CFDDIR	GPIOI1	GPIOI1								CFDDIR	
GPIOJ0	GPIOJ0	GPIOJ0			SDD16						
GPIOJ1	GPIOJ1	GPIOJ1			SDD17						
GPIOJ2	GPIOJ2	GPIOJ2			SDD18						
GPIOJ3	GPIOJ3	GPIOJ3			SDD19						
GPIOJ4	GPIOJ4	GPIOJ4			SDD20						
GPIOJ5	GPIOJ5	GPIOJ5			SDD21						
GPIOJ6	GPIOJ6	GPIOJ6			SDD22						
GPIOJ7	GPIOJ7	GPIOJ7			SDD23						
GPIOK0	GPIOK0	GPIOK0			SDD24						
GPIOK1	GPIOK1	GPIOK1			SDD25						
GPIOK2	GPIOK2	GPIOK2			SDD26						
GPIOK3	GPIOK3	GPIOK3			SDD27						
GPIOK4	GPIOK4	GPIOK4			SDD28						
GPIOK5	GPIOK5	GPIOK5			SDD29						
GPIOK6	GPIOK6	GPIOK6			SDD30						
GPIOK7	GPIOK7	GPIOK7			SDD31						

Function 1 : Function 1 Function 2 : Function 2

3.4 Pin Status during Reset

Pin Name	Direction during RESET	Value during RESET	Presence of Internal Resistor	Description
MA[19:0]	Output	Low (However, only bit11 is High.)	No	
MD[15:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	100kΩ
MCS[1:0]#	Output	High	No	
MCS[3:2]#	Input	High-Z	No	Depends on the external circuit. GPIOD[5:4]
MBEL#	Output	Low	No	
MBEH#	Output	Low	No	
MOE#	Output	High	No	
MWE#	Output	High	No	
SDA[14:0]	Output	Low	No	
SDD[15:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	100kΩ
SDCS[1:0]#	Output	High	No	
SDWE#	Output	High	No	
SDCLK	Output	SDCLK(32KHz)	No	
SDLKEN	Output	Low	No	
SDRAS#	Output	High	No	
SDCAS#	Output	High	No	
SDDQM[1:0]#	Output	Low	No	
CM1DATA[7:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM1VREF	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM1HREF	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM1CLKOUT	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM1CLKIN	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM2DATA[7:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM2VREF	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM2HREF	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM2CLKOUT	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM2CLKIN	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CFCE2#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFCE1#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFIORD#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFIOWR#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFWAIT#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFRST	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFIRQ	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFSTSCHG#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFDEN#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFDDIR	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
GPIOA[7:0]	Input	High-Z	No	Depends on the external circuit.
GPIOB[7:0]	Input	High-Z	No	Depends on the external circuit.
GPIOC[7:0]	Input	High-Z	No	Depends on the external circuit.
GPIOD[7:0]	Input	High-Z	No	Depends on the external circuit.
GPIOJ[7:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	100kΩ
GPIOK[7:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	100kΩ
SYSCLKI	–	High-Z	No	Depends on the SYSCKSEL Pin
SYSVCP	Output	High-Z	No	Leave this pin Open when it is used
SYSCKSEL	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
TRST#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
TCK	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
TMS	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
TDI	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
TDO	Output	High-Z	No	
TESTEN0	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
TESTEN1	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
TESTCK	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
RESET#	Input	Low	No	

4. FUNCTIONAL DESCRIPTION

4.1 System Configuration

S2S65A00 determines internal chip operation based on the MD bus.

Specifically, the IC's operation mode is determined by sampling internally and establishing the values of the MD [15:0] bus while it resets for the period during which RESET# is active and which it is starting up. Usually, you do not have to configure anything because the MD bus have a built-in pull-down resistor (Typ: 100kΩ). Depending on your system, you can add an external pull-up resistor and change the operation mode.

If an external pull-up resistor is used, steady current through the external pull-up resistor and the internal pull-down resistor should be present. In S2S65A00, however, this steady current can be turned off by the software that disconnects the internal pull-down resistor.

For more information, see "13. SYSTEM CONTROLLER."

Table 4.1 System Configuration Pins MODESEL [15:0]

Pin Name	Pin Functions	Value at Resetting	
		Low	High
MD0	MODESEL0	32KHz Mode (normal operation)	Reserved (For test)*
MD1	MODESEL1	Normal operation	Reserved (For test)*
MD2	MODESEL2	Normal operation	Reserved (For test)*
MD3	MODESEL3	For user setting	For user setting
MD4	MODESEL4	For user setting	For user setting
MD5	MODESEL5	For user setting	For user setting
MD6	MODESEL6	For user setting	For user setting
MD7	MODESEL7	For user setting	For user setting
MD8	MODESEL8	For user setting	For user setting
MD9	MODESEL9	For user setting	For user setting
MD10	MODESEL10	For user setting	For user setting
MD11	MODESEL11	For user setting	For user setting
MD12	MODESEL12	For user setting	For user setting
MD13	MODESEL13	For user setting	For user setting
MD14	MODESEL14	For user setting	For user setting
MD15	MODESEL15	For user setting	For user setting

Note *: Do not use the setting of where Reserved (for test) is specified. If the setting of Reserved (for test) is used, this IC may be broken.

MD0: Clock input setting
- Low: 32KHz (PLL input: normal operation)
- High: Reserved (for test; cannot be used by users)

MD[2:1]: Clock related settings
- Low: Specifies normal operation
- High: Reserved (for test; cannot be used by users)

MD [15:3]: Provides 12-bit portion for users.
Set values are reflected to the chip configuration register inside the system controller. Users can use this for their purpose.

4.2 Memory Map

There are two AHB buses in the IC; the one that is connected to ARM720T is hereafter referred as AHB1 bus.

4.2.1 Memory Map (AHB1)

Table 4.2 AHB1 Memory Map

Start Address	End Address	Size (Mega Byte)	Device	External Chip Select	Device Bus size (bit)
0x0000_0000	0x01FF_FFFF	32 MB	External ROM	MCS0#	16
0x0200_0000	0x03FF_FFFF	32 MB	Reverved		
0x0400_0000	0x05FF_FFFF	32 MB	External SRAM	MCS1#	16
0x0600_0000	0x07FF_FFFF	32 MB	Reverved		
0x0800_0000	0x09FF_FFFF	32 MB	External SRAM	MCS2#	16
0x0A00_0000	0x0BFF_FFFF	32 MB	Reverved		
0x0C00_0000	0x0DFF_FFFF	32 MB	External SRAM	MCS3#	16
0x0E00_0000	0x0FFF_FFFF	32 MB	Reverved		
0x1000_0000	0x1FFF_FFFF	256 MB	Reserved		
0x2000_0000	0x2FFF_FFFF	256 MB	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256 MB	Reserved		
0x4000_0000	0x4FFF_FFFF	256 MB	External SDRAM	SDCS0#	16/32
0x5000_0000	0x5FFF_FFFF	256 MB	External SDRAM	SDCS1#	16/32
0x6000_0000	0x6FFF_FFFF	256 MB	Reserved		
0x7000_0000	0x7FFF_FFFF	256 MB	Reserved		
0x8000_0000	0x8FFF_FFFF	256 MB	Reserved		
0x9000_0000	0x9FFF_FFFF	256 MB	Reserved		
0xA000_0000	0xAFFF_FFFF	256 MB	Reserved		
0xB000_0000	0xBFFF_FFFF	256 MB	Reserved		
0xC000_0000	0xC1FF_FFFF	32 MB	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32 MB	Reserved		
0xC400_0000	0xC5FF_FFFF	32 MB	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32 MB	Reserved		
0xC800_0000	0xC9FF_FFFF	32 MB	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32 MB	Reserved		
0xCC00_0000	0xCDFF_FFFF	32 MB	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32 MB	Reserved		
0xD000_0000	0xDFFF_FFFF	256 MB	Reserved		
0xE000_0000	0xEFFF_FFFF	256 MB	USB		8
0xF000_0000	0xFFFF_FFFF	256 MB	Built-in I/O area		32/16/8

The device connected to CS0 becomes the boot device.

The area from 0xC000_0000 to 0xCFFF_FFFF is used as the alias space area for the device assigned to 0x0000_0000. This means that you can see the same information by accessing this space.

The portion shown at the bottom of the memory map is used by S2S65A00's built-in I/O device. The layout of this built-in I/O is shown in “Specifications: IO Map.”

4.2.2 Memory Map (AHB2)

The second AHB bus in S2S65A00 is referred as AHB2. The memory map is shown below.

Table 4.3 Memory Map (AHB2)

Start Address	End Address	Size (Mega Byte)	Device	External Chip Select	Device Bus size (bit)
0x0000_0000	0x01FF_FFFF	32 MB	External ROM	MCS0#	16
0x0200_0000	0x03FF_FFFF	32 MB	Reverved		
0x0400_0000	0x05FF_FFFF	32 MB	External SRAM	MCS1#	16
0x0600_0000	0x07FF_FFFF	32 MB	Reverved		
0x0800_0000	0x09FF_FFFF	32 MB	External SRAM	MCS2#	16
0x0A00_0000	0x0BFF_FFFF	32 MB	Reverved		
0x0C00_0000	0x0DFF_FFFF	32 MB	External SRAM	MCS3#	16
0x0E00_0000	0x0FFF_FFFF	32 MB	Reverved		
0x1000_0000	0x1FFF_FFFF	256 MB	Reserved		
0x2000_0000	0x2FFF_FFFF	256 MB	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256 MB	Reserved		
0x4000_0000	0x4FFF_FFFF	256 MB	External SDRAM	SDCS0#	16/32
0x5000_0000	0x5FFF_FFFF	256 MB	External SDRAM	SDCS1#	16/32
0x6000_0000	0x6FFF_FFFF	256 MB	Reserved		
0x7000_0000	0x7FFF_FFFF	256 MB	Reserved		
0x8000_0000	0x8FFF_FFFF	256 MB	Reserved		
0x9000_0000	0x9FFF_FFFF	256 MB	Reserved		
0xA000_0000	0xAFFF_FFFF	256 MB	Reserved		
0xB000_0000	0xBFFF_FFFF	256 MB	Reserved		
0xC000_0000	0xC1FF_FFFF	32 MB	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32 MB	Reserved		
0xC400_0000	0xC5FF_FFFF	32 MB	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32 MB	Reserved		
0xC800_0000	0xC9FF_FFFF	32 MB	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32 MB	Reserved		
0xCC00_0000	0xCDFF_FFFF	32 MB	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32 MB	Reserved		
0xD000_0000	0xDFFF_FFFF	256 MB	JPEG2 DMA Port		32
0xE000_0000	0xEFFF_FFFF	256 MB	JPEG1 DMA Port		32
0xF000_0000	0xFFFF_FFFF	256 MB	Reserved		

As shown in the memory maps for AHB1 and AHB2, memories connected to S2S65A00's (external) memory controller (external ROM, external SRAM, and external SDRAM) and the built-in SRAM can be accessed as shared resource through both AHB buses (Master).

4.3 I/O Map

The internal I/O area uses the following portion of the 256 MB area from 0xF000_0000 to 0xFFFF_FFFF. No device exists for the blank (Reserved) sections. If they are accessed, only indefinite data will be read.

Table 4.4 Built-in I/O Map

Base Address	Space Size	S2S65A00
0xFFFD_0000	4KB	Reserved
0xFFFD_1000	4KB	Reserved
0xFFFD_2000	4KB	Reserved
0xFFFD_3000	4KB	Timer B
0xFFFD_4000	4KB	UART2
0xFFFD_5000	4KB	UART3
0xFFFD_6000	4KB	Reserved
0xFFFD_7000	4KB	Reserved
0xFFFD_8000	4KB	Camera Interface_2
0xFFFD_9000	4KB	JPEG resize_2
0xFFFD_A000	4KB	JPEG module/FIFO_2
0xFFFD_B000	4KB	JPEG codec_2
0xFFFD_C000	4KB	ADC
0xFFFD_D000	4KB	SD memory
0xFFFD_E000	4KB	Reserved
0xFFFD_F000	4KB	USB_2.0_Device
0xFFFE_0000	4KB	APB bridge
0xFFFE_1000	4KB	Reserved
0xFFFE_2000	4KB	Reserved
0xFFFE_3000	4KB	DMAC1
0xFFFE_4000	2KB	CF attribute
0xFFFE_4800	2KB	CF common
0xFFFE_5000	2KB	CF I/O
0xFFFE_5800	1KB	CF ture IDE CS1#
0xFFFE_5C00	1KB	CF ture IDE CS2#
0xFFFE_6000	4KB	CF control
0xFFFE_7000	4KB	ARS
0xFFFE_8000	4KB	Camera Interface_1
0xFFFE_9000	4KB	JPEG resize_1
0xFFFE_A000	4KB	JPEG module/FIFO_1
0xFFFE_B000	4KB	JPEG codec_1
0xFFFE_C000	4KB	JPEG DMAC
0xFFFE_D000	4KB	I^2C
0xFFFE_E000	4KB	I^2S
0xFFFE_F000	4KB	(Interrupt controller)
0xFFFF_0000	4KB	Reserved
0xFFFF_1000	4KB	GPIO/pin function
0xFFFF_2000	4KB	SPI
0xFFFF_3000	4KB	Reserved
0xFFFF_4000	4KB	Reserved
0xFFFF_5000	4KB	UART1
0xFFFF_6000	1KB	Reserved
0xFFFF_7000	4KB	Reserved
0xFFFF_8000	4KB	RTC
0xFFFF_9000	4KB	DMAC2
0xFFFF_A000	4KB	Memory controller
0xFFFF_B000	4KB	Timer A
0xFFFF_C000	4KB	WDT
0xFFFF_D000	4KB	System controller
0xFFFF_E000	4KB	Reserved
0xFFFF_F000	4KB	Interrupt controller

4.4 Interrupt Controller

S2S65A00 can handle two sources for FIQ and up to 32 interrupt sources for IRQ.

Based on the IRQ mapping, internal interrupt requests are connected to the interrupt controller as shown below. For more information, see “15. INTERRUPT CONTROLLER.”

Type	Level	S2S65A00
FIQ	FIQ0	Watchdog timer
	FIQ1	GPIOB0
IRQ	IRQ0	Watchdog timer
	IRQ1	Interrupt controller
	IRQ2	ARM720T COMMRx
	IRQ3	ARM721T COMMTx
	IRQ4	Timer A Ch.0
	IRQ5	Timer A Ch.1
	IRQ6	Timer A Ch.2
	IRQ7	Reserved
	IRQ8	JPEG control_1
	IRQ9	DMA1C
	IRQ10	JPEG DMAC_1
	IRQ11	Camera Interface_1
	IRQ12	ARS
	IRQ13	DMA2C
	IRQ14	GPIOA or GPIOB
	IRQ15	SPI
	IRQ16	I^2C
	IRQ17	UART_1
	IRQ18	RTC
	IRQ19	CF card Interface
	IRQ20	INT_GPIOB (GPIOB)
	IRQ21	Timer B
	IRQ22	Reserved
	IRQ23	JPEG control_2
	IRQ24	JPEG DMAC_2
	IRQ25	Camera Interface_2
	IRQ26	UART2
	IRQ27	UART3
	IRQ28	SD memory
	IRQ29	USB_DEV
	IRQ30	ADC
	IRQ31	I^2S

4.5 Built-in Functions of S2S65A00

S2S65A00 has many built-in functional blocks to realize drive camera controller function. These functions are listed below.

Chapter Number	Functional Block Name	Abbr. for Function Name
5.	CPU	CPU
6.	DMA Controller 1	DMAC1
7.	Camera Interface [1:0]	CAM[1:0]
8.	JPEG Controller [1:0]	JPG[1:0]
9.	JPEG_DMAC	JDMA
10.	DMA Controller 2	DMAC2
11.	USB2.0 Device	USB2d
12.	APB Bridge	APB
13.	System Controller	SYS
14.	Memory Controller	MEMC
15.	Interrupt Controller	INT
16.	UART[2:0]	UART[2:0]
17.	I^2C Single Master Core Module	I^2C
18.	I^2S Interface	I^2S
19.	Serial Peripheral Interface	SPI
20.	Compact Flash Card Interface	CF
21.	SD Memory Controller Interface	SD
22.	Timer A	TIMA
23.	Timer B	TIMB
24.	Real Time Clock	RTC
25.	Watchdog Timer	WDT
26.	GPIO	GPIO
27.	A/D Converter	ADC
28.	Area Sensor	ARS

5. CPU

5.1 Description

The ARM720T core is used as the S2S65A00 CPU module. ARM720T, whose core is ARM7TDMI, is equipped with a unified 8k-byte cache, a memory management unit, and an extended write buffer.

For more information on the ARM720T core, see *"ARM720T Revision 4 (AMBA AHB Bus Interface Version) core CPU Manual."*

5.2 ARM720T Block Diagram

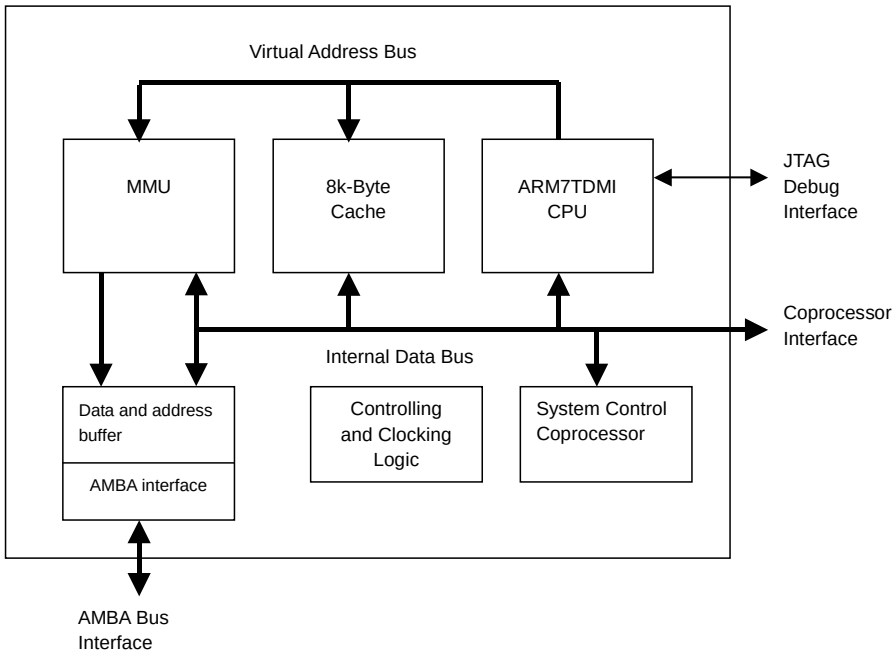


Fig.5.1 ARM720T Block Diagram

6. DMA CONTROLLER 1 (DMAC1)

6.1 Description

DMAC1, which is placed as the bus master on AHB1, is a DMA controller that transfers data between an APB device and a memory (internal or external memory) or between memories (internal and external memories) without using the CPU.

This DMA controller employs a dual-address transfer method where two address phases are used. In this method, data from the transfer source address is read into the temporary register within DMAC1 and then written to the destination address, each time a DMA request arises. This operation is repeated until the number of transfers reaches "0." You can specify 8, 16, or 32 bits as the transfer size.

6.2 Block Diagram

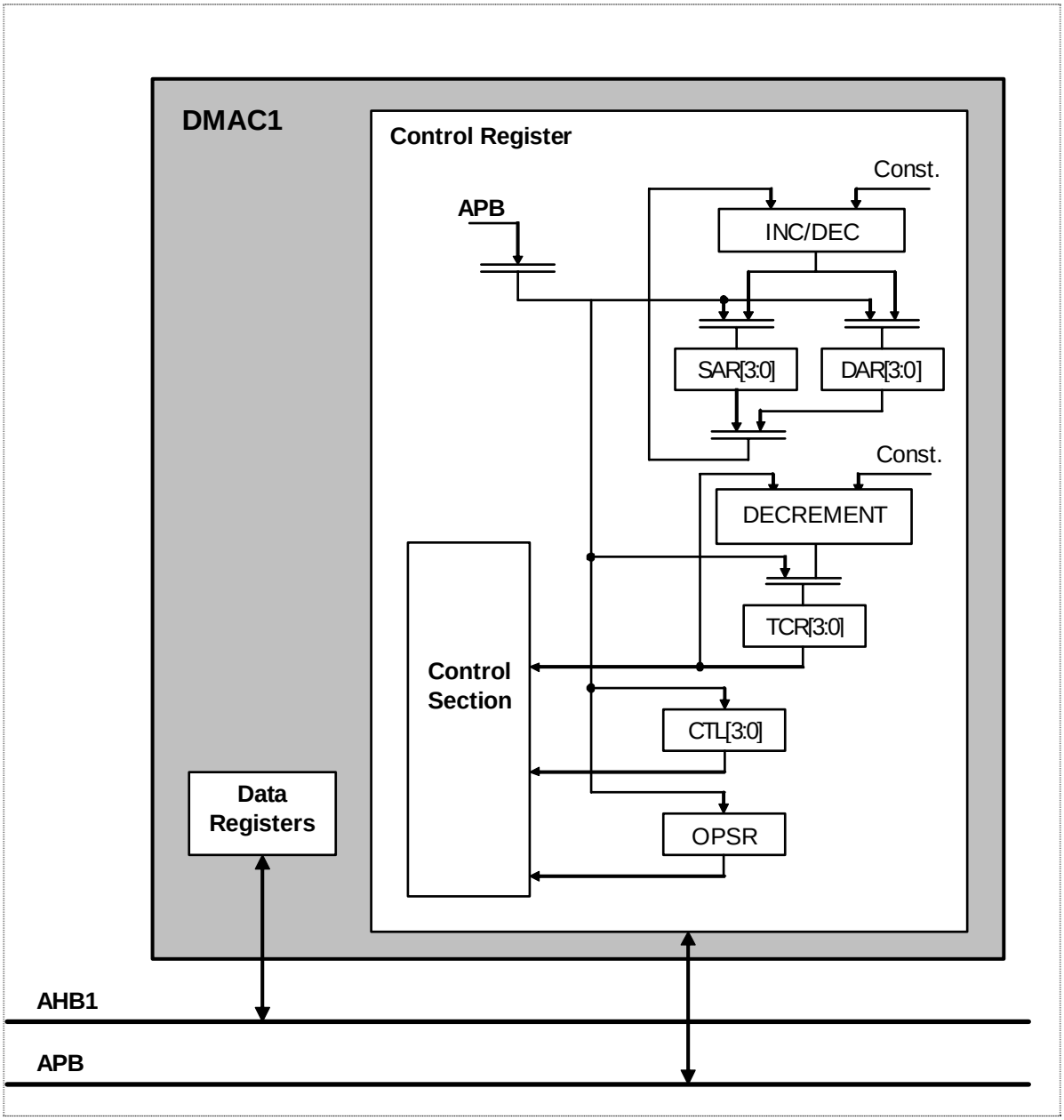


Fig.6.1 DMA Controller 1 (DMAC1) Block Diagram

6.3 External Pins

There is no external pin that relates to DMA Controller 1.

6.4 Registers

6.4.1 List of Registers

The base address of a DMAC1-related register is 0xFFFE_3000.
For the registers described in this chapter and latter chapters, the following abbreviations may be used.

R/W: Read and Write
RO: Read Only
WO: Write Only
RSV: Reserved bit/register (write down "0", if not otherwise specified)
n/a: not available (write down "0", if not otherwise specified)

Table 6.1 List of Registers (Base Address: 0xFFFE_3000)

Address Offset	Register Name	Register Abbreviation	Default Value	R/W	Data Access Size
0x00	DMA Channel 0 Source Address Register	SAR0	0xXXXX_XXXX	R/W	32
0x04	DMA Channel 0 Destination Address Register	DAR0	0xXXXX_XXXX	R/W	32
0x08	DMA Channel 0 Transfer Count Register	TCR0	0x00XX_XXXX	R/W	32
0x0C	DMA Channel 0 Control Register	CTL0	0x00XX_XXXX	R/W	32
0x10	DMA Channel 1 Source Address Register	SAR1	0xXXXX_XXXX	R/W	32
0x14	DMA Channel 1 Destination Address Register	DAR1	0xXXXX_XXXX	R/W	32
0x18	DMA Channel 1 Transfer Count Register	TCR1	0x00XX_XXXX	R/W	32
0x1C	DMA Channel 1 Control Register	CTL1	0x00XX_XXXX	R/W	32
0x20	DMA Channel 2 Source Address Register	SAR2	0xXXXX_XXXX	R/W	32
0x24	DMA Channel 2 Destination Address Register	DAR2	0xXXXX_XXXX	R/W	32
0x28	DMA Channel 2 Transfer Count Register	TCR2	0x00XX_XXXX	R/W	32
0x2C	DMA Channel 2 Control Register	CTL2	0x00XX_XXXX	R/W	32
0x30	DMA Channel 3 Source Address Register	SAR3	0xXXXX_XXXX	R/W	32
0x34	DMA Channel 3 Destination Address Register	DAR3	0xXXXX_XXXX	R/W	32
0x38	DMA Channel 3 Transfer Count Register	TCR3	0x00XX_XXXX	R/W	32
0x3C	DMA Channel 3 Control Register	CTL3	0x00XX_XXXX	R/W	32
0x60	DMA Channel Operating Select Register	OPSR	0x0000_0000	R/W	32

6.4.2 Details of Registers

If not otherwise specified, set "0" in the reserved bits. If a write operation performed on a reserved bit, unexpected results may occur. The bits specified as "n/a" have no impact on the hardware.

Some of the resisters can be accessed only under certain conditions. Read/write to the non-accessible registers is ineffective.

DMA Channel [3:0] Source Address Register (SAR [3:0])

DMAC1[0x00] [0x10] [0x20] [0x30] Default value = 0xXXXX_XXXX Read/Write

DMA Channel [3:0] Source Address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA Channel [3:0] Source Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **DMA Channel [3:0] Source Address [31:0]**

A transfer source address for DMA transfer on channel [3:0] is set by the software. A transfer source address must be an boundary address appropriate for the transferred data size. For example, the bits [1:0] of this resister must be 00b for 32-bit transfer. Once DMA transfer begins, the transfer source address automatically updates to the next address every time when a transfer ends, according to the transferred data size (TS: channel [3:0] control register bits [4:3]) and the source address mode (SAM: channel [3:0] control register bits [13:12]).

DMA Channel [3:0] Destination Address Register (DAR [3:0])

DMAC1[0x04] [0x14] [0x24] [0x34] Default value = 0xXXXX_XXXX Read/Write

DMA Channel [3:0] Destination Address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA Channel [3:0] Destination Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **DMA Channel [3:0] Destination Address [31:0]**

A transfer destination address for DMA transfer on channel [3:0] is set by the software. A transfer destination address must be an boundary address appropriate for the transferred data size. For example, the bits [1:0] of this resister must be 00b for 32-bit transfer. Once DMA transfer begins, the transfer destination address automatically updates to the next address every time when a transfer ends, according to the transferred data size (TS: channel [3:0] control register bits [4:3]) and the destination address mode (DAM: channel [3:0] control register bits [15:14]).

DMA Channel [3:0] Transfer Count Register (TCR [3:0])

DMAC1[0x08] [0x18] [0x28] [0x38] Default value = 0x00XX_XXXX Read/Write

n/a								DMA Channel [3:0] Transfer Count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA Channel [3:0] Transfer Count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [23:0]: **DMA Channel [3:0] Transfer Count [23:0]**

In these bits, the number of transfers for DMA transfer is set by the software. Once DMA transfer begins, the number decrements every time when a transfer ends. If "0" is set here, the number of transfers will be 2^{24} = 16777216. A DMA interruption occurs when the number counts down to "0."

If this register is read, "0" are appended to bits [31:24].

DMA Channel [1:0] Control Resister (CTL [1:0])

DMAC1[0x0C] [0x1C] Default value = 0x00XX_XXXX Read/Write

n/a								RSV				IDLE	RSV	AM	RSV
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV	RIM	TM	TS		IE	TE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [23:20]: **RSV Reserved (0)**

Bit 19: **IDLE Idle Delay Enable**

Some DMA target devices may require this setting to be enabled.
It is recommended to set this to "1" for write transfer to a device (transfer from memory to device).

0: Normal operation
1: Delays the timing for the next request from a device to be received.

Bit 18: **RSV Reserved**

Bit 17: **AM Acknowledge Mode**

Selects an output timing of DACK signal.

0: Active during DMA read cycle
1: Active during DMA write cycle

Bit 16: **RSV Reserved (0)**

Bits [15:14]: **DAM Destination Address Mode [1:0]**

Selects an update mode for the destination address register for the update that is taken place after one transfer end.

00: Fixes the transfer destination address (no update)
01: Increments the transfer destination address according to the transferred data size
(8-bit address increments by 1; 16-bit, by 2; 32-bit, by 4)
10: Decrements the transfer destination address according to the transferred data size
(8-bit address decrements by 1; 16-bit, by 2; 32-bit, by 4)
11: Reserved

Bits [13:12]: **SAM Source Address Mode [1:0]**

Selects an update mode for the source address register for the update that is taken place after one transfer ends.

00: Fixes the source destination address (no update)
01: Increments the transfer source address according to the transferred data size
(8-bit address increments by 1; 16-bit, by 2; 32-bit, by 4)
10: Decrements the transfer source address according to the transferred data size
(8-bit address decrements by 1; 16-bit, by 2; 32-bit, by 4)
11: Reserved

Bits [11:8]: **RS Resource Select [3:0]**

Selects a trigger to start DMA transfer.

0000: USB_Device I/O (USB-REQ0)
0001: USB_Device I/O (USB-REQ1)
0010: Reserved
0011: Reserved
0100: UART#1 output (TX)
0101: UART#1 input (RX)
0110: UART#2 output (TX)
0111: UART#2 input (RX)
1000: UART#3 output (TX)
1001: UART#3 input (RX)
1010: SPI I/O (SPI-TX/RX)

1011: Reserved
1100: SD_MEM I/O (SDM-REQ)
1101: Reserved
1110: Reserved
1111: SW-Request Software Request
Set bits [11:8] to "1111" to select software DMA transfer.

Bit 7: **RSV Reserved (0)**

Bit 6: **RIM Request Input Mode**

Selects an input mode for DMA request signal.
0: Active LOW (level trigger)
1: Falling edge (edge trigger)

Bit 5: **TM Transfer Mode**

Selects a transfer mode for DMA transfer.
0: Single transfer (one transfer per DMA request)
1: Demand transfer (continues the transfer while the DMA request exists)

Bits [4:3]: **TS Transfer Size [1:0]**

Selects data size to be transferred in a single transfer.
00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2: **IE Interrupt Enable**

Permits/inhibits transfer end interrupt on the DMA channel [1:0].
0: Interrupt inhibited.
1: Interrupt permitted.

Bit 1: **TE DMA Transfer End**

0 (Read): Transferring or waiting
1 (Read): DMA transfer ended
0 (Write): Clears this bit
1 (Write): Ineffective

This bit is set when the value of the DMA channel 0 transfer count register becomes "0" as a result of DMA transfer. Once this bit is set, it preserves "1" until "0" is written to clear it. DMA transfer on this channel is inhibited until this bit is cleared. This bit also works as an interrupt source flag.

Bit 0: **DE DMA Enable**

DMA transfer on channel [1:0] is permitted based on this bit.
0: DMA transfer inhibited
1: DMA transfer permitted

DMA Channel [3:2] Control Resister (CTL [3:2])

DMAC1[0x2C] [0x3C] Default value = 0x00XX_XXXX Read/Write

n/a								RSV				IDLE	RSV	AM	RSV
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV	RIM	TM	TS		IE	TE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [23:20]: **RSV Reserved (0)**

Bit 19: **IDLE Idle Delay Enable**

Some DMA target devices may require this setting to be enabled.
It is recommended to set this to "1" for write transfer to a device (transfer from memory to device).

0: Normal operation
1: Delays the timing for the next request from a device to be received.

Bit 18: **RSV Reserved (0)**

Bit 17: **AM Acknowledge Mode**

Selects an output timing of DACK signal.

0: Active during DMA read cycle
1: Active during DMA write cycle

Bit 16: **RSV Reserved (0)**

Bits [15:14]: **DAM Destination Address Mode [1:0]**

Selects an update mode for the destination address register for the update that is taken place after one transfer end.

00: Fixes the transfer destination address (no update)
01: Increments the transfer destination address according to the transferred data size
(8-bit address increments by 1; 16-bit, by 2; 32-bit, by 4)
10: Decrements the transfer destination address according to the transferred data size
(8-bit address decrements by 1; 16-bit, by 2; 32-bit, by 4)
11: Reserved

Bits [13:12]: **SAM Source Address Mode [1:0]**

Selects an update mode for the source address register for the update that is taken place after one transfer ends.

00: Fixes the source destination address (no update)
01: Increments the transfer source address according to the transferred data size
(8-bit address increments by 1; 16-bit, by 2; 32-bit, by 4)
10: Decrements the transfer source address according to the transferred data size
(8-bit address decrements by 1; 16-bit, by 2; 32-bit, by 4)
11: Reserved

Bits [11:8]: **RS Resource Select [3:0]**

Selects a trigger to start DMA transfer.

0000: USB_Device#0 I/O (USB-REQ0)
0001: USB_Device#1 I/O (USB-REQ1)
0010: Reserved
0011: Reserved
0100: I^2C output (I^2C-TX)
0101: I^2C input (I^2C-RX)
0110: I^2S I/O (I^2S-CH0)
0111: I^2S I/O (I^2S-CH1)
1000: Reserved
1001: Reserved
1010: SPI I/O (SPI-TX/RX)

1011: Reserved
1100: SD_MEM I/O (SDM-REQ)
1101: Reserved
1110: Reserved
1111: SW-Request Software Request
Set bits [11:8] to "1111" to select software DMA transfer.

Bit 7: **RSV Reserved (0)**

Bit 6: **RIM Request Input Mode**

Selects an input mode for DMA request signal.
0: Active LOW (level trigger)
1: Falling edge (edge trigger)

Bit 5: **TM Transfer Mode**

Selects a transfer mode for DMA transfer.
0: Single transfer (one transfer per DMA request)
1: Demand transfer (continues the transfer while the DMA request exists)

Bits [4:3]: **TS Transfer Size [1:0]**

Selects data size to be transferred in a single transfer.
00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2: **IE Interrupt Enable**

Permits/inhibits transfer end interrupt on the DMA channel [3:2].
0: Interrupt inhibited.
1: Interrupt permitted.

Bit 1: **TE DMA Transfer End**

0 (Read): Transferring or waiting
1 (Read): DMA transfer ended
0 (Write): Clears this bit
1 (Write): Ineffective

This bit is set when the value of the DMA channel 3 transfer count register becomes "0" as a result of DMA transfer. Once this bit is set, it preserves "1" until "0" is written to clear it. DMA transfer on this channel is inhibited until this bit is cleared. This bit also works as an interrupt source flag.

Bit 0: **DE DMA Enable**

DMA transfer on channel [3:2] is permitted based on this bit.
0: DMA transfer inhibited.
1: DMA transfer permitted.

DMA Channel Operating Select Resister (OPSR)

DMAC1[0x60] Default value = 0x0000_0000 Read/Write

na															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a						DPM		n/a							DGE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit [9:8]: **DPM Priority Mode**

Determines the priority when there are transfer requests for multiple channels simultaneously.

00: CH0>CH1>CH2>CH3
01: CH1>CH0>CH3>CH2
10: CH2>CH3>CH0>CH1
11: Simple round robin mode

Bit 0: **DGE DMA Global Enable**

Enables/disables all channels for the DMA.

0: Disable
1: Enable

7. CAMERA INTERFACE [2:1] (CAM[2:1])

7.1 Description

The camera interface has the following features.

- 8-bit parallel interface × 2 ports
- 2 camera modules can be connected
- Can support cameras whose maximum resolution size is VGA (640×480) (depending on AC characteristics)
- 8-bit data bus interface (YUV4:2:2 format)
- Supports ITU-R BT.656 format
- Capture frames can be configured.
- Pixel clock frequency for inputting camera data is less than 2/3 of CPU clock.

7.2 Block Diagram

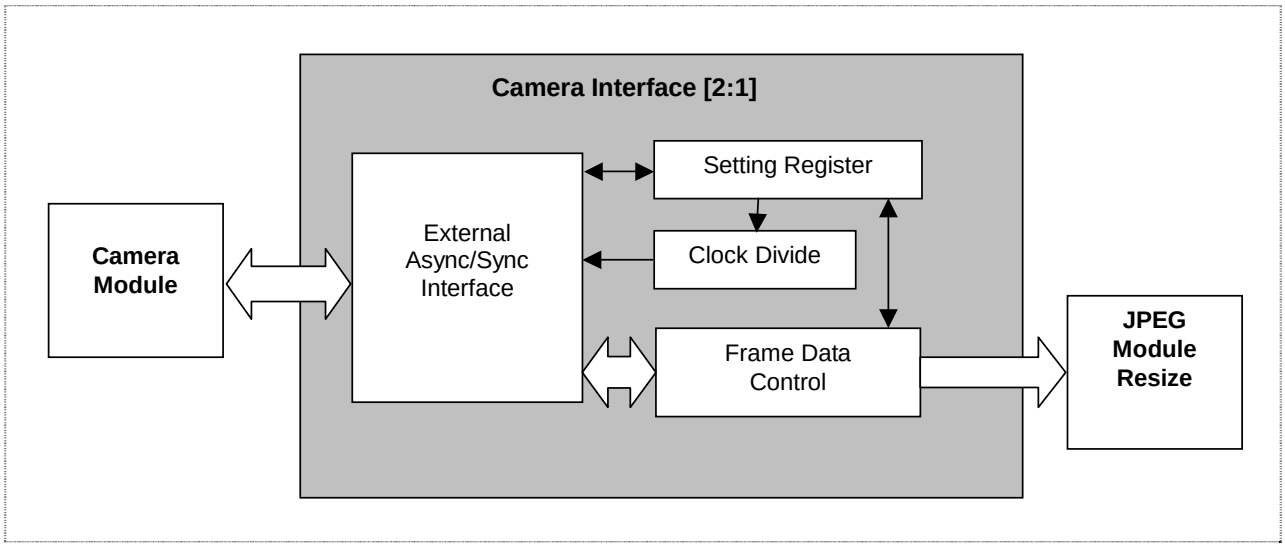


Fig.7.1 Camera Interface [2:1] Block Diagram

7.3 External Pins

The Camera Interface [2:1]-related external pins are listed below.

Table 7.1 External Pins

Pin Name	Input/Output	Pin Functions	Multiplex Pin*/Remarks
CM1DATA [7:0]	I	Camera 1 data (YUV) input	GPIOE [7:0]
CM1VREF	I	Vertical sync input at Camera 1 data input	GPIOF0
CM1HREF	I	Horizontal sync input at Camera 1 data input	GPIOF1
CM1CLKOUT	O	Basic clock output for Camera 1	GPIOF2
CM1CLKIN	I	Pixel clock for inputting Camera 1 data	GPIOF3
CM2DATA [7:0]	I	Camera 2 data (YUV) input	GPIOG [7:0]
CM2VREF	I	Vertical sync input at Camera 2 data input	GPIOF4
CM2HREF	I	Horizontal sync input at Camera 2 data input	GPIOF5
CM2CLKOUT	O	Basic clock output for Camera 2	GPIOF6
CM2CLKIN	I	Pixel clock for inputting Camera 2 data	GPIOF7

Note(*): External pins for the camera interface (CAM) are multiplexed with GPIO pins. To use as an external pin for CAM, set the value to “Function 1 of other than GPIO” by using the GPIO pin function register.

7.4 Registers

7.4.1 List of Registers

The registers for camera interface [2:1] are listed below. The base addresses of these registers are as follows.
Camera interface 1: 0xFFFE_8000
Camera interface 2: 0xFFFD_8000

Table 7.2 List of Registers

Address Offset	Register Name	Default value	R/W	Data Access Size
Camera Interface : Base Address 0xFFFE_8000(CAM1) 0xFFFD_8000(CAM2)				
0x00	Camera 1/2 clock cycle setting register	0x0000	R/W	16bit
0x04	Camera 1/2 signal setting register	0x0000	R/W	16bit
0x08 - 0x1C	Reserved	—	—	—
0x20	Camera 1/2 mode setting register	0x0000	R/W	16bit
0x24	Camera 1/2 frame control register	0x0000	R/W	16bit
0x28	Camera 1/2 control register	0x0000	WO	16bit
0x2C	Camera 1/2 status register	0x0004	RO	16bit
0x30 - 0x5C	Reserved	—	—	—

7.4.2 Detailed Description of Registers

Camera [2:1] Clock Cycle Setting Register

CAM2 [0x00], CAM1 [0x00] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a			Clock Frequency Select bits [4:0]				
7	6	5	4	3	2	1	0

Bits [4:0]: **Clock Frequency Select bits [4:0]**

Sets the cycle for the output clock (CMCLKOUT).

Table 7.3 Camera Clock Cycle Settings

Bits [4:0]	Clock Frequency
00000	Internal clock 1/1
00001	Internal clock 1/2 (Recommended value)
00010	Internal clock 1/3
00011	Internal clock 1/4
00100	Internal clock 1/5
00101	Internal clock 1/6
00110	Internal clock 1/7
00111	Internal clock 1/8
01000	Internal clock 1/9
01001	Internal clock 1/10
01010	Internal clock 1/11
01011	Internal clock 1/12
01100	Internal clock 1/13
01101	Internal clock 1/14
01110	Internal clock 1/15
01111	Internal clock 1/16
10000	Internal clock 1/17
10001	Internal clock 1/18
10010	Internal clock 1/19
10011	Internal clock 1/20
10100	Internal clock 1/21
10101	Internal clock 1/22
10110	Internal clock 1/23
10111	Internal clock 1/24
11000	Internal clock 1/25
11001	Internal clock 1/26
11010	Internal clock 1/27
11011	Internal clock 1/28
11100	Internal clock 1/29
11101	Internal clock 1/30
11110	Internal clock 1/31
11111	Internal clock 1/32

Camera [2:1] Signal Setting Register							
CAM2 [0x04], CAM1 [0x04] Default value = 0x0000							Read/Write
n/a							
15	14	13	12	11	10	9	8
n/a	Reserved (0)	Clock Mode Select	YUV Data Format Select bits [1:0]		HREF Active Select	VREF Active Select	Valid Input Clock Edge
7	6	5	4	3	2	1	0

Bit 6: **Reserved bit (0)**

If necessary, write "0" to this bit.

Bit 5: **Clock Mode Select**

Selects a clock to use for the internal operation. If an internal divided clock is used, the cycle must be set to greater than or equal to 1/2. Unlike external clock input, using an internal divided click makes it difficult to synchronize output data with the internal divided clock because of delays occur on the board and within the camera (image sensor). For stable operation, the dividing ratio must be large enough for these delays to be ignored.

0: External clock input (CMCLKIN input is used)
1: Internal divided clock (The clock set by the camera clock cycle setting register (the same clock as =CMCLKOUT) is internally used.)

Bits [4:3]: **YUV Data Format Select bits[1:0]**

Selects the order of YUV data that is input by byte.

Table 7.4 YUV Format Select

Bits [4:3]	YUV Format (8bit)
00	(1st) CbYCrY (last)
01	(1st) CrYCbY (last)
10	(1st) YCbYCr (last)
11	(1st) YCrYCb (last)

Bit 2: **HREF Active Select**

Sets the HREF data active level.

0: Recognizes at High level while HREF data is effective.
1: Recognizes at Low level while HREF data is effective.

Bit 1: **VREF Active Select**

Sets the VREF data active level.

0: Recognizes at High level while VREF data is effective.
1: Recognizes at Low level while VREF data is effective.

Bit 0: **Valid Input Clock Edge**

Selects the valid edge for input clock.
This is effective on both external and internal clocks selected by setting bit 5.

0: Data is read when the clock changes from Low to High.
1: Data is read when the clock changes from High to Low.

Camera [2:1] Mode Setting Register

CAM2 [0x20], CAM1 [0x20] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
RSV (0)			RSV Camera Active Pull-down Disable	n/a	Fast Sampling Mode	RSV (0)	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ITU-R BT656 Enable	RSV (0)			Clock Output Disable	RSV (0)		Camera Module Enable

Bit [15:13]: **RSV Reserved (0)**

Bit 12: **RSV Reserved (Camera Active Pull-down Disable)**

Disables active pull-down of the camera pins.
0: Active pull-down enabled
1: Active pull-down disabled

Note: For S2S65A00, this bit function does not work. The GPIOE/GPIOF/GPIOG resistor control registers in the system controller provide similar control.

Bit 10: **Fast Sampling Mode**

Sets the operation that samples input data from a camera at twice as fast as normal.

0: Normal sampling
1: Fast sampling

Bit [9:8]: **RSV Reserved (0)**

Bit 7: **ITU-R BT656 Enable**

Selects input from ITU-R BT656 compliant cameras.
This mode is available only in YUV422-8-bit interface mode.

0: Normal camera mode
1: ITU-R BT656 compliant camera mode

Bits [6:4]: **RSV Reserved (0)**

Bit 3: **Clock Output Disable**

Sets the operation of clock output (CMCLKOUT).

0: Clock output (CMCLKOUT output is used. The cycle is set by the camera cycle setting register.)
1: Clock output inhibited (CMCLKOUT is not output; the pin is fixed to Low level.)

Bits [2:1]: **RSV Reserved (0)**

Bit 0: **Camera Module Enable**

Sets an operation of camera module.
External clock output and other operations will be enabled when the setting of this bit is “enabled”.
Setting this bit to “0” stops clock supply to the camera module, and thus is effective to reduce power consumption. Read/write operation on the camera interface registers is possible even if this bit is set to “0.”

0: Camera module disabled
1: Camera module enabled

Camera [2:1] Frame Control Register

CAM2 [0x24], CAM1 [0x24] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
n/a							Jpeg Raw Data Capture Mode
Frame Capture Interrupt Control	Single Frame Capture Enable	Shutter Sync. Disable	Frame Sample Control bits [2:0]			Frame Capture Interrupt Polarity	Frame Capture Interrupt Enable
7	6	5	4	3	2	1	0

Bit 8: **Jpeg Raw Data Capture Mode**

Specifies the JPEG data capture mode.

0: YUV data capture setting

1: JPEG data capture setting

Bit 7: **Frame Capture Interrupt Control**

Controls frame interrupts.

If frame capture end interrupt is specified, it does not affected by the setting in bit 5.

If frame capture end interrupt is specified, it does not affected by the setting in bit 0 and interrupt is enabled.

Table 7.5 Frame Interrupt Control

Image Capture Interrupt Polarity Bit	Interrupt
0	Interrupt is generated when an effective frame is captured.
1	Interrupt is generated when a frame capture ends.

Bit 6: **Single Frame Capture Enable**

Specifies the frame capture mode.

0: Repeats capturing.

1: Stops capturing after one frame is captured when Image frame capture operation (CAM[0x28] bit 2 = 1) has been set.

Note: This bit must not be changed when Camera Module Enable (CAM[0x20] bit 0) is set to “1.”

Bit 5: **Shutter Synchronization Disable**

Inhibits shutter synchronization of the frame interrupt flag.

0: After the shutter is pressed, outputs frame interrupt status for each effective frame.

1: Always outputs frame interrupt status for each effective frame.

Bits [4:2]: **Frame Sample Control Bits [2:0]**

Controls frame thinning-out of camera input.

Table 7.6 Frame Thinning-out Control

Frame Sample Control Bits[2:0]	Mode
000	No thinning-out
001	Thins out to 1/2
010	Thins out to 1/3
011	Thins out to 1/4
100	Thins out to 1/5
101	Thins out to 1/6
110	Reserved
111	Reserved (Thins out all frames)

Bit 1: **Image Capture Interrupt Polarity**

Controls interrupt generation during image capture.

Table 7.7 Frame Interrupt Control

Image Capture Interrupt Polarity Bit	Interrupt
0	When the VREF data active level changes to the VREF data inactive level
1	When the VREF data inactive level changes to the VREF data active level

Bit 0: **Image Capture Interrupt Enable**

Enables interrupt during image capture.

0: Disabled

1: Enabled

Camera [2:1] Control Register

CAM2 [0x28], CAM1 [0x28] Default value = 0x0000 Write Only

n/a						ITU-R BT656 Error Flag 1 Clear	ITU-R BT656 Error Flag 0 Clear
15	14	13	12	11	10	9	8
n/a				Frame Capture Stop	Frame Capture Start	Frame Interrupt Status Clear	Camera Module Soft Reset
7	6	5	4	3	2	1	0

Bit 9: **ITU-R BT656 Error Flag 1 Clear**

The state can be cleared by writing "1" to this bit.

0: No operation

1: Clears the value of error flag 1.

Bit 8: **ITU-R BT656 Error Flag 0 Clear**

The state can be cleared by writing "1" to this bit.

0: No operation

1: Clears the value of error flag 0.

Bit 3: **Frame Capture Stop**

Specifies the setting for stopping image frame capture.

Although the default after reset is the capture operation state, a camera module can be started with the capture stopped state by setting this bit along with bit 0 Camera Module Soft Reset.

0: No operation

1: Specifies the setting for stopping capture operation

Bit 2: **Frame Capture Start**

Specifies the setting for executing image frame capture.

Stops the operation after one frame is captured if Single Frame Capture has been enabled.

0: No operation

1: Specifies the setting for executing capture operation

Bit 1: **Frame Capture Interrupt Status Clear**

The frame interrupt status from a camera is cleared by writing "1" to this bit.

0: Does not clear the frame interrupt status.

1: Clears the frame interrupt status.

Bit 0: **Camera Module Soft Reset**

Initializes the circuit section of the camera.

0: Does not initialize the circuit section of the camera.

1: Initializes the circuit section of the camera.

Note: When this bit is set to "1", only CAM [0x20] bit 0 (Camera Module Enable) is initialized to "0" and the register is not initialized.

Camera [2:1] Status Register

CAM2 [0x2C], CAM1 [0x2C] Default value = 0x0004 Read Only

15	14	13	12	11	10	9	8
n/a						ITU-R BT656 Error Flag 1	ITU-R BT656 Error Flag 0
n/a	Camera VSYNC	RSV (1)	Effective Frame Status	Frame Capture Busy Status	Frame Capture Start/Stop Flag	Frame Interrupt Status	n/a
7	6	5	4	3	2	1	0

Bit 9: **ITU-R BT656 Error Flag 1**

In the ITU-R BT656 mode, the state of the reference command can be determined.

0: Normal operation

1: Detects 2-bit error during reference decoding.

Bit 8: **ITU-R BT656 Error Flag 0**

In the ITU-R BT656 mode, the state of the reference command can be determined.

0: Normal operation

1: Detects 1-bit error correction during reference decoding.

Bit 6: **Camera VSYNC**

The VSYNC value from a camera module can be determined.

In the ITU-R BT656 mode, the value after decoding the reference code can be determined.

The VSYNC value is constant irrespective of the polarity setting.

0: In the vertical blank period

1: In the valid data period

Bit 5: **RSV Reserved (1)**

Bit 4: **Effective Frame Status**

By reading this bit, whether a frame is effective (sampled) or ineffective (thinned out) can be determined.

0: The frame is ineffective (thinned out)

1: The frame is effective

Bit 3: **Frame Capture Busy Status**

The image frame capture status can be determined.

0: Frame capture has stopped.

1: Frame capture is in progress.

Bit 2: **Frame Capture Start/Stop Flag**

The setting status of image frame capture can be determined.

After the setting is set to Start, the setting automatically changes to Stop when frame capture ends, if Single Frame Capture has been enabled.

0: Frame Capture Stop setting

1: Frame Capture Start setting

Bit 1: **Frame Capture Interrupt Status**

By reading this bit, the frame interrupt status can be determined.

0: Frame interrupt has not been generated.

1: Frame interrupt has been generated.

Note: Frame interrupt is generated only when bit 7 = 1 or bit 0 = 1 in the camera frame control register (CAM[0x24]).

7.5 Explanation of Operation

Camera modules (image sensors) whose maximum resolution size is UXGA can be connected to this device. The camera interface uses an 8 bit-data bus to receive image data in YUV4:2:2 format using clock.

Make sure to check the AC characteristics to determine if a camera can be used or not.

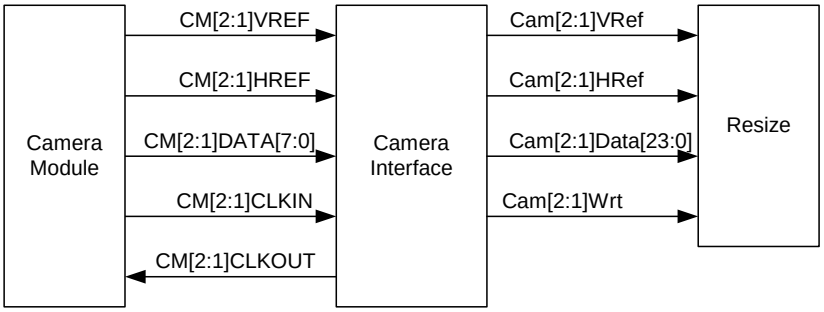


Fig.7.2 Camera Interface Connection Diagram

Table 7.8 Internal Signal

Internal Signal Name	Description
CM[2:1]VREF	Incoming vertical data enable signal from the camera module
CM[2:1]HREF	Incoming horizontal data enable signal from a camera module
CM[2:1]DATA[7:0]	Incoming 8-bit data signal from a camera module The ITU-R BT.601 format is also supported.
CM[2:1]CLKIN	Incoming pixel clock from a camera module
CM[2:1]CLKOUT	Clock that makes a camera module operate
Cam[2:1]VREF	Outgoing vertical data enable signal to the resizing circuit
Cam[2:1]HREF	Outgoing horizontal data enable signal to the resizing circuit
Cam[2:1]Data[23:0]	Signal that is converted to 24-bit after converting YUV4:2:2 to YUV4:4:4.
Cam[2:1]Wrt	Outgoing data enable signal to the resizing circuit

The camera interface synchronizes the incoming internal image processing clock and synchronous signal from the camera module with the internal image processing clock and outputs camera image data to the resizing circuit. The figure below shows a circuit that samples image data output from a camera module using CM[2:1]DATA, CM[2:1]HREF, and CM[2:1]VREF, on the rising edge of CM[2:1]CLKIN.

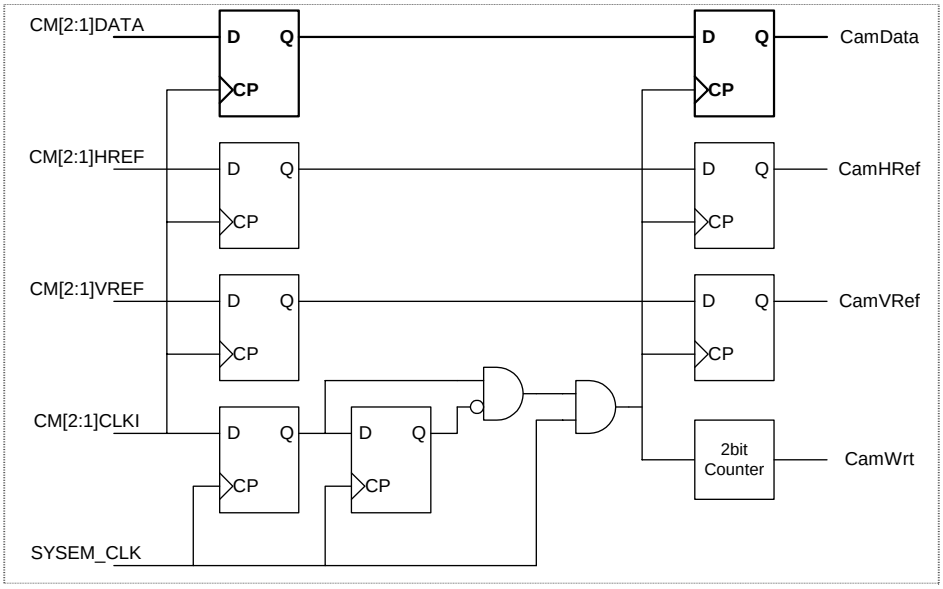


Fig.7.3 Data Sampling Circuit

In normal sampling mode (camera mode setting register CAMx[0x20] bit10=0), where an internal system clock samples the incoming clock from a camera clock module and detects clock edges, the system clock frequency must be at least twice as high as the incoming pixel clock from the camera module, in theory. In reality, whether actual operation is performed at doubled frequency or not depends on the system because there are factors such as clock duty ratio. Therefore, the frequency should at least be tripled to ensure the operation.

In fast sampling mode (camera mode setting register CAMx[0x20] bit10=1), the sampling performance is doubled because two systems of the circuit are used (just one system is used in normal sampling mode) and toggled after each cycle.

7.5.1 Frame Capture Interrupt

Interrupt can be generated from a VREF signal of camera image data. This interrupt is necessary for the image processing routine involving camera images, such as JPEG encoding.

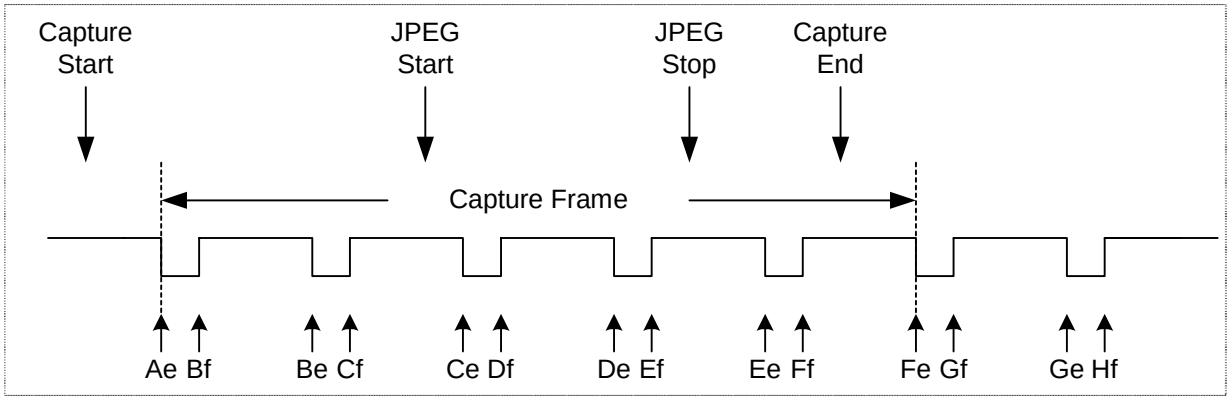


Fig.7.4 Timing of Interrupt Generation

Fig.7.4 is a timing chart that starts with enabling camera interface capture and ends with disabling it, showing actions related to a camera frame interrupt. Capture Start means that bit 0 in the camera mode setting register of the camera interface is set to “1”. Similarly, Capture End means that bit 0 in the camera mode setting register is set to “0”. JPEG Start means that “1” is written to the JPEG start/stop control register; JPEG End means that “0” is written to it. Capture Frame represents a frame where incoming data from the camera module is sent to the resizing circuit.

There are four register settings in total that determine the timing of the frame capture interrupt. Table 7.9 shows where in the figure above an interrupt is generated based on the settings of these four register bits. Usually, the fourth register setting from the top (indicated by hatching) is used.

Table 7.9 Timing of Interrupt Generation

Camera Frame Control Register CAM[0x24]				Camera Frame Interrupt Generation Timing
Bit 7	Bit 5	Bit 1	Bit 0	
0	x	x	0	No Interrupt Generation
0	0	0	1	Ce, De
0	0	1	1	Df, Ef
0	1	0	1	Be, Ce, De, Ee, Fe
0	1	1	1	Cf, Df, Ef, Ff, Gf
1	x	0	x	Fe
1	x	1	x	Gf

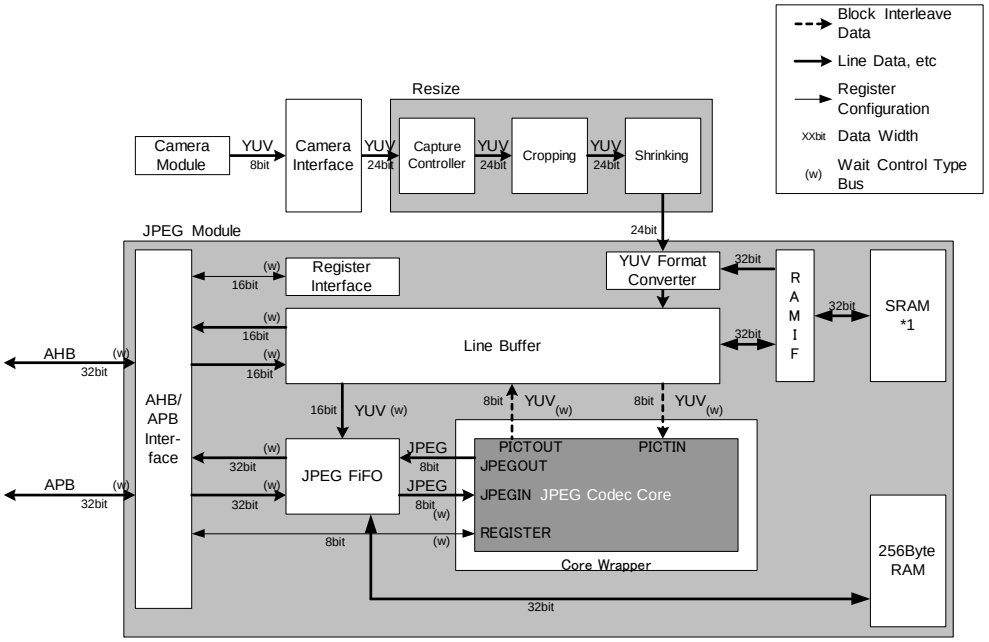
8. JPEG CONTROLLER (JPG[2:1])

8.1 Description

This controller provides JPEG encoding for camera input images and a capture function that captures camera input images as YUV data. The JPEG encoding almost complies with JPEG's Baseline method. The calculation accuracy complies with JPEG Part2 (ISO/IEC10918-2) and thus enough accuracy is provided. Maximum image size is VGA. Images that can be JPEG-encoded are minimum unit images of MCU size or larger in YUV format. Two quantization tables can be configured when compressed and up to four tables are supported when expanded. The Huffman table can handle two DC tables and two AC tables. Optional markers up to 36 bytes can be inserted during encoding. Processed makers are SOI, SOF0, SOS, DQT, DHT, DRI, RSTm, and EOI, which are automatically decoded when expanded. DNL makers are not supported. Supported YUV formats are YUV4:4:4, YUV4:2:2, YUV4:1:1, and YUV4:2:0 for encoding camera images and YUV4:2:2 and YUV:4:2:0 for capturing YUV data. Gray images and RGB images are not supported. The processing speed for a VGA size image is 1/30 second at minimum. However, it cannot be guaranteed that this minimum speed is always achieved because processing is greatly affected by the image data, Huffman table, and quantization tables.

To support two camera modules, there are two channels of JPEG controllers [2:1] that have exactly equivalent functions.

8.2 Block Diagram



*1 : Refer to bit[5:4](EMBRAMSEL[1:0]) of SYS[0x68](EMBMEMCTL) for the allocation of SRAM.

Fig.8.1 JPEG Controller Block Diagram

8.3 External Pins

There is no external pin that relates to the JPEG controller.

8.4 Registers

The following table lists the registers. There are two identical channels of JPEG controllers (JPEG1 and JPEG2). Although the registers for these two channels have different base addresses, their functions are exactly same.

8.4.1 List of Registers

Table 8.1 List of Registers

Address Offset	Register Name	Default Value	R/W	Data Access Size
Resizer Operation Registers (RSZ[2:1]) : Base Address = 0xFFFE_9000(JPG1) 0xFFFD_9000(JPG2)				
0x60	Global resizer control register	0x0000	WO	16bit
0x64	Capture control state register	0x0000	RO	16bit
0x68	Capture data setting register	0x0000	R/W	16bit
0x70-0x7C	Reserved register	—	—	—
0xC0	Capture resizer control register	0x0000	R/W	16bit
0xC8	Capture resizer start X position register	0x0000	R/W	16bit
0xCC	Capture resizer start Y position register	0x0000	R/W	16bit
0xD0	Capture resizer end X position register	0x027F	R/W	16bit
0xD4	Capture resizer end Y position register	0x01DF	R/W	16bit
0xD8	Capture resizer scaling rate register	0x8080	R/W	16bit
0xDC	Capture resizer scaling mode register	0x0000	R/W	16bit
JPEG Module Registers (JCTL[2:1]) : Base Address = 0xFFFE_A000(JPG1) 0xFFFD_A000(JPG2)				
0x00	JPEG control register	0x0000	R/W	16bit
0x04	JPEG status flag refresh register	0x8080	R/W	16bit
0x08	JPEG raw status flag register	0x8080	RO	16bit
0x0C	JPEG interrupt control register	0x0000	R/W	16bit
0x10	Reserved register	—	—	—
0x14	JPEG codec start stop control register	0x0000	WO	16bit
0x18 - 0x1C	Reserved register	—	—	—
0x20	Huffman table auto setting register	0x0000	R/W	16bit
JPEG FIFO Setting Registers (JFIFO[2:1]) : Base Address = 0xFFFE_A000(JPG1) 0xFFFD_A000(JPG2)				
0x40	JPEG FIFO control resister	0x0000	R/W	16bit
0x44	JPEG FIFO status register	0x8001	RO	16bit
0x48	JPEG FIFO size register	0x003F	R/W	16bit
0x4C	JPEG FIFO read/write port resister	0x0000 0000	R/W	32bit
0x50 - 0x58	Reserved register	—	—	—
0x60	Encode size limit register 0	0x0000	R/W	16bit
0x64	Encode size limit register 1	0x0000	R/W	16bit
0x68	Encode size result register 0	0x0000	RO	16bit
0x6C	Encode size result register 1	0x0000	RO	16bit
0x70 - 0x78	Reserved register	—	—	—
JPEG Line Buffer Setting Registers (JLB[2:1]) : Base Address = 0xFFFE_A000(JPG1) 0xFFFD_A000(JPG2)				
0x80	JPEG line buffer status flag register	0x0000	R/W	16bit
0x84	JPEG line buffer raw status flag register	0x0000	RO	16bit
0x88	JPEG line buffer current status flag register	0xX009	RO	16bit
0x8C	JPEG line buffer interrupt control register	0x0000	R/W	16bit
0x90 - 0x9C	Reserved register	—	—	—

Address Offset	Register Name	Default Value	R/W	Data Access Size
0xA0	JPEG line buffer horizontal pixel acceptable size register	0x2800	R/W	16bit
0xA4	JPEG line buffer memory address offset register	0x0020	R/W	16bit
0xA8 - 0xBC	Reserved register	—	—	—
0xC0	JPEG line buffer read/write port resister	0x0000 0000	R/W	32bit
JPEG Codec Registers (JCODEC[2:1]) : Base Address = 0xFFFE_B000(JPG0) 0xFFFD_B000(JPG1)				
0x00	Operation mode setting register	0x0000	R/W	16bit
0x04	Command setting register	Not applicable	WO	16bit
0x08	JPEG operation status register	0x0000	RO	16bit
0x0C	Quantization table number register	0x0000	R/W	16bit
0x10	Huffman table number register	0x0000	R/W	16bit
0x14	DRI setting register 0	0x0000	R/W	16bit
0x18	DRI setting register 1	0x0000	R/W	16bit
0x1C	Vertical pixel size register 0	0x0000	R/W	16bit
0x20	Vertical pixel size register 1	0x0000	R/W	16bit
0x24	Horizontal pixel size register 0	0x0000	R/W	16bit
0x28	Horizontal pixel size register 1	0x0000	R/W	16bit
0x2C - 0x34	Reserved register	—	—	—
0x38	RST marker operation setting register	0x0000	R/W	16bit
0x3C	RST marker operation status register	0x0000	RO	16bit
0x40 - 0xCC	Inserted marker data register	0x00FF	R/W	16bit
0x400 - 0x4FC	Quantization table No.0 register	Not applicable	R/W	16bit
0x500 - 0x5FC	Quantization table No.1 register	Not applicable	R/W	16bit
0x800 - 0x83C	DC Huffman table No.0 register 0	Not applicable	WO	16bit
0x840 - 0x86C	DC Huffman table No.0 register 1	Not applicable	WO	16bit
0x880 - 0x8BC	AC Huffman table No.0 register 0	Not applicable	WO	16bit
0x8C0 - 0xB44	AC Huffman table No.0 register 1	Not applicable	WO	16bit
0xC00 - 0xC3C	DC Huffman table No.1 register 0	Not applicable	WO	16bit
0xC40 - 0xC6C	DC Huffman table No.1 register 1	Not applicable	WO	16bit
0xC80 - 0xCBC	AC Huffman table No.1 register 0	Not applicable	WO	16bit
0xCC0 - 0xF44	AC Huffman table No.1 register 1	Not applicable	WO	16bit

Details of these registers are described in the following sections.

8.4.2 Resizer Operation Registers (RSZ[2:1])

Note: Most of the resizer registers cannot be changed while receiving data from a camera interface.

Global Resizer Control Register

RSZ[0x60] Default value = 0x0000 Write Only

15	14	13	12	11	10	9	8
n/a					Reserved		ACTAGAIN
7	6	5	4	3	2	1	0
n/a			Reserved		n/a	Reserved	

Bit [10:9]: **Reserved**
Be sure to set each bit to “0”.

Bit 8: **ACTAGAIN (Write only)**
This bit is used when continuous encoding is performed. If you write “1” to this bit, the frame that follows the current frame is also sent to the JPEG codec circuit.
Whether continuous encoding is possible or not depends on the system and software specifications. For details, see Explanation of Operation.

Bit [4:3]: **Reserved**
Be sure to set each bit to “0”.

Bits [1:0]: **Reserved**
Be sure to set each bit to “0”.

Capture Control State Register

RSZ[0x64] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
n/a				State Value			

Bit [3:0]: **State Value**
Indicates the value for the current state of the capture control sequence state machine. For the meanings of state values, see Explanation of Operation, “8.5.1 Capture Control.”

Capture Data Setting Register

RSZ[0x68] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
n/a							Data Format Select

Bit 0: **Captured Image Data Format Select**
Selects a captured image data format. If this bit is set to “1”, all settings in RSZ[*] registers except RSZ[0xC0] are disabled. (The statuses of some Read-Only registers may change but the values are not effective.)

0: YUV data
1: JPEG data

Reserved Register

RSZ[0x70-7C] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bit [15:0]: **Reserved**

Be sure to set each bit to “0”.

Capture Resizer Control Register

RSZ[0xC0] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
n/a							
Capture Resizer Software Reset (WO)	n/a			Reserved (0)			Capture Resizer Enable
7	6	5	4	3	2	1	0

Bit 7: **Capture Resizer Software Reset (Write only)**

By writing “1” to this bit, the capture resizer is software-reset. Nothing happens when “0” is written to it.

Bits [3:1]: **Reserved**

Be sure to set each bit to “0”.

Bit 0: **Capture Resizer Enable**

Setting this bit to “0” stops clock supply to the resize module, and thus is effective to reduce power consumption. Read/write operation on the resize module registers is possible even if this bit is set to “0.”

Write:

0: Capture resizer is disabled.

1: Capture resizer is enabled.

Read:

0: Capture resizer has been disabled.

1: Capture resizer has been enabled.

Capture Resizer Start X Position Register

RSZ[0xC8] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
n/a					Capture Resizer Start X Position bits [10:8]		
Capture Resizer Start X Position bits [7:0]							
7	6	5	4	3	2	1	0

Bits [10:0]: **Capture Resizer Start X Position [10:0]**

These bits specify the X position where the capturing of the capture resizer starts.

Capture Resizer Start Y Position Register

RSZ[0xCC] Default value = 0x0000 Read/Write

n/a					Capture Resizer Start Y Position bits [10:8]		
15	14	13	12	11	10	9	8
Capture Resizer Start Y Position bits [7:0]							
7	6	5	4	3	2	1	0

Bits [10:0]: **Capture Resizer Start Y Position [10:0]**

These bits specify the Y position where the capturing of the capture resizer starts.

Capture Resizer End X Position Register

RSZ[0xD0] Default value = 0x027F Read/Write

n/a					Capture Resizer End X Position bits [10:8]		
15	14	13	12	11	10	9	8
Capture Resizer End X Position bits [7:0]							
7	6	5	4	3	2	1	0

Bits [10:0]: **Capture Resizer End X Position [10:0]**

These bits specify the X position where the capturing of the capture resizer ends.

Capture Resizer End Y Position Register

RSZ[0xD4] Default value = 0x01DF Read/Write

n/a					Capture Resizer End Y Position bits [10:8]		
15	14	13	12	11	10	9	8
Capture Resizer End Y Position bits [7:0]							
7	6	5	4	3	2	1	0

Bits [10:0]: **Capture Resizer End Y Position [10:0]**

These bits specify the Y position where the capturing of the capture resizer ends.

Capture Resizer Scaling Rate Register

RSZ[0xD8] Default value = 0x8080 Read/Write

15	14	13	12	11	10	9	8
Reserved (0)							

7	6	5	4	3	2	1	0
Reserved (0)				Capture Resizer Scaling Rate bits [3:0]			

Bits [15:4]: **Reserved**

Be sure to set each bit to "0".

Bits [3:0]: **Capture Resizer Scaling Rate [3:0]**

These bits specify the capture resizer scaling rate. Some of the scaling rates cannot be used depending on the setting in the capture resizer scaling mode register. For details, see *"Table 8.4 Capture Resizer Scaling Rate/Mode Selection"*, which is a part of the description on the capture resizer scaling mode register.

Table 8.2 Capture Resizer Scaling Rate Settings

bits [3:0]	Capture Resizer Scaling Rate Settings
0000	Reserved
0001	1/1
0010	1/2
0011	1/3 (Only in vertical thinning-out mode, capture resizer scaling mode register bit 1-0=01)
0100	1/4
0101	1/5 (Only in vertical thinning-out mode, capture resizer scaling mode register bit 1-0=01)
0110	1/6 (Only in vertical thinning-out mode, capture resizer scaling mode register bit 1-0=01)
0111	1/7 (Only in vertical thinning-out mode, capture resizer scaling mode register bit 1-0=01)
1000	1/8
1001-1111	Reserved

Capture Resizer Scaling Mode Register

RSZ[0xDC] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a				Reserved (0)		Capture Resizer Scaling Mode bits [1:0]	
7	6	5	4	3	2	1	0

Bits [3:2]: **Reserved**

Be sure to set each bit to "0".

Bits [1:0]: **Capture Resizer Scaling Mode [1:0]**

These bits specify the scaling mode of the capture resizer. Depending on the scaling mode, some of the scale rates cannot be set in the capture resizer scaling rate register. For details, see *"Table 8.4 Capture Resizer Scaling Rate/Mode Selection."*

Table 8.3 Capture Resizer Scaling Mode Selection

Bits [1:0]	Capture Resizer Scaling Mode
00	No scaling
01	Thinning-out in both vertical and horizontal directions
10	Thinning-out in vertical direction, averaging in horizontal direction
11	Reserved

Table 8.4 Capture Resizer Scaling Rate/Mode Selection

		RSZ[0xDC] Bits [1:0]			
		00	**01**	**10**	**11**
RSZ[0xD8] Bits [3:0]	**0000**	1/1	Reserved	Reserved	Reserved
	0001	1/1	Reserved	Reserved	Reserved
	0010	1/1	1/2	1/2	Reserved
	0011	1/1	1/3	Reserved	Reserved
	0100	1/1	1/4	1/4	Reserved
	0101	1/1	1/5	Reserved	Reserved
	0110	1/1	1/6	Reserved	Reserved
	0111	1/1	1/7	Reserved	Reserved
	1000	1/1	1/8	1/8	Reserved
	other	Reserved	Reserved	Reserved	Reserved

8.4.3 JPEG Module Registers (JCTL[2:1])

JPEG Control Register

JCTL[0x00] Default value = 0x0000 Read/Write

JPEG Encode Fast Mode	JPEG Marker Fast Output Mode	Reserved (0)					JPEG 180˚ Rotation Enable
15	14	13	12	11	10	9	8

JPEG Module SW Reset (WO)	Reserved (0)		UV Data Type Conversion	Operation Mode bits [2:0]			JPEG Module Enable
7	6	5	4	3	2	1	0

Bit15: **JPEG Encoding Fast Mode**

0: No impact.

1: Expedites JPEG compression process by using the fixed Huffman table.

Note: When JPEG compression is performed with this bit set to “1”, the JPEG Huffman table value (effective address in JCODEC[0x800]-[0xF44]) must be changed to one listed in “ISO/IEC 10918-1 Annex K” or the JCTL[0x20] Huffman auto setting register must be set to “1.”

Bit14: **JPEG Marker Fast Output Mode**

0: No impact.

1: Expedites JPEG maker output by using the fixed Huffman table.

Note: The setting of this bit is effective only when the encode fast mode (Bit15) is set to “1.” When JPEG compression is performed with this bit set to “1”, you do not have to write the JPEG Huffman table value (effective address in JCODEC[0x800]-[0xF44]). The one listed in “ISO/IEC 10918-1 Annex K” will be used.)

Bits [13:9]: **Reserved**

Be sure to set each bit to “0”.

Bit 8: **JPEG 180° Rotation Enable**

Specifies whether the JPEG encode data should be rotated. Because the hardware rotates the data on a specific line basis, you may need to rearrange the data in the JPEG file by using the software. For details, see “8.5.5.5 JPEG 180° Rotation Enable.”

0: No rotation

1: Rotates by 180°

Bit 7: **JPEG Module Software Reset (Write only)**

Resets JPEG modules except JPEG codec. The registers are not reset. Be sure to reset the modules before starting JPEG encoding. To execute this reset, JPEG Module Enable must be set to “1.” If the JPEG modules have not been enabled, the software reset may not be executed even if “1” is written to this bit.

0: No impact

1: Reset

Bit [6:5]: **Reserved**

Be sure to set each bit to “0”.

Bit 4: **UV Data Type Conversion**

Converts the UV type input from a camera.

Table 8.5 UV Data Type

Bit 4	Camera Data Type	Internal Data Type
0 (Converts)	0 ≤ U ≤ 255 0 ≤ V ≤ 255	-128 ≤ U ≤ 127 -128 ≤ V ≤ 127
	16 ≤ Cb ≤ 240 16 ≤ Cr ≤ 240	-112 ≤ Cb ≤ 112 -112 ≤ Cr ≤ 112
	-128 ≤ U ≤ 127 -128 ≤ V ≤ 127	0 ≤ U ≤ 255 0 ≤ V ≤ 255
	-112 ≤ Cb ≤ 112 -112 ≤ Cr ≤ 112	16 ≤ Cb ≤ 240 16 ≤ Cr ≤ 240
1 (Not Convert)	0 ≤ U ≤ 255 0 ≤ V ≤ 255	0 ≤ U ≤ 255 0 ≤ V ≤ 255
	16 ≤ Cb ≤ 240 16 ≤ Cr ≤ 240	16 ≤ Cb ≤ 240 16 ≤ Cr ≤ 240
	-128 ≤ U ≤ 127 -128 ≤ V ≤ 127	-128 ≤ U ≤ 127 -128 ≤ V ≤ 127
	-112 ≤ Cb ≤ 112 -112 ≤ Cr ≤ 112	-112 ≤ Cb ≤ 112 -112 ≤ Cr ≤ 112

Bits [3:1] : **Operation Mode Select**

Specifies the JPEG operation mode. When YUV data capture is selected, clock supply to the JPEG codec stops and the JPEG codec register cannot be accessed. Therefore, if you change the setting of the JPEG codec register, camera image JPEG encode (other than YUV data capture) should be set.

While camera image JPEG encode YUV format is set in JCODEC[0x00]bit1-0YUV format select, YUV format for YUV data capture is set in this resister bit.

Table 8.6 JPEG Operation Mode

bits [3:1]	JPEG Operation Mode
000	Camera image JPEG encode (YUV4:4:4, YUV4:2:2, YUV4:1:1, and YUV4:2:0. YUV4:4:4 supports images resized to less than 1/2 only.) (Default value)
001	Reserved
010	Reserved
011	YUV data capture (YUV 4:2:2)
100	Reserved
101	Reserved
110	Reserved
111	YUV data capture (YUV 4:2:0)

Bit 0: **JPEG Module Enable**

Enables the JPEG modules. When the modules are disabled, clock supply to the JPEG modules stops and the JPEG codec register cannot be accessed. Be sure to disable the JPEG modules before disabling the resizer.

0: Disables (Default value)

1: Enables

JPEG Status Flag Register

JCTL[0x04] Default value = 0x8080 Read/Write

Reserved (1) -	JPEG Codec File Out Status (RO)	JPEG FIFO Threshold Status bits [1:0] (RO)		Encode Size Limit Violation Flag (R/W)	JPEG FIFO Threshold Trigger Flag (R/W)	JPEG FIFO Full Flag (R/W)	JPEG FIFO Empty Flag (R/W)
15	14	13	12	11	10	9	8
Reserved (1) —			Reserved (R/W)	Reserved —	JPEG Line Buffer Overflow Flag (RO)	JPEG Codec Interrupt Flag (RO)	JPEG Line Buffer Interrupt Flag (RO)
7	6	5	4	3	2	1	0

Bit 15: **Reserved**

Be sure to set "1".

Bit 14: **JPEG Codec File Out Status (Read only)**

Shows the JPEG codec output state during encoding.

0: JPEG file output has been stopped

1: JPEG encoding or JPEG file output is in progress.

Be sure to set "1" when a write operation is performed.

Bits [13:12]: **JPEG FIFO Threshold Status (Read only)**

Shows the current data status of the JPEG FIFO.

Be sure to set "1" when a write operation is performed.

Table 8.7 JPEG FIFO Threshold Status

bits [13:12]	JPEG FIFO Threshold Status
00	Empty
01	Greater than or equal to 4 bytes and less than 1/4 of the FIFO size
10	Greater than or equal to 1/4 and less than 1/2 of the FIFO size
11	Greater than or equal to 1/2 of the FIFO size

Bit 11: **Encode Size Limit Violation Flag**

This is an interrupt flag that indicates that, during JPEG encoding, the JPEG file size has exceeded the encode size limits set in the encode size limit register. This flag can be disabled by using the JPEG interrupt control register (JCTL[0x0C] bit 11). This function just reports that the file size has exceeded the limits and the JPEG encoding will continue.

Read:

0: No Interrupt

1: Interrupt exists (encode was oversized)

Write:

0: No impact

1: Clear

Bit 10: **JPEG FIFO Threshold Trigger Flag**

This is an interrupt flag that indicates that the data size of the JPEG FIFO has exceeded the JPEG FIFO threshold (JPEG FIFO control register bits 5-4) at least once. This flag can be disabled by using the JPEG interrupt control register (bit 10).

Read:

0: No Interrupt

1: Interrupt exists (JPEG FIFO threshold was violated)

Write:

0: No impact

1: Clear

Bit 9: **JPEG FIFO Full Flag**

This is an interrupt flag that indicates that the JPEG FIFO has been full at least once. This flag can be disabled by using the JPEG interrupt control register (bit 9).

Read:

0: No Interrupt
1: Interrupt exists (JPEG FIFO full occurred)

Write:

0: No impact
1: Clear

Bit 8: **JPEG FIFO Empty Flag**

This is an interrupt flag that indicates that the JPEG FIFO has been empty at least once. This flag can be disabled by using the JPEG interrupt control register (bit 8).

Read:

0: No Interrupt
1: Interrupt exists (JPEG FIFO empty occurred)

Write:

0: No impact
1: Clear

Bit [7:5]: **Reserved**

Be sure to set each bit to “1”.

Bit 4: **Reserved**

Be sure to set “0”.

Bit 3: **Reserved**

Be sure to set “1”.

Bit 2: **JPEG Buffer Interrupt Flag (Read only)**

This is an interrupt flag that indicates that the JPEG line buffer has overflowed. This flag can be disabled by using the JPEG interrupt control register (bit 2).
This flag is cleared by the setting of JPEG Module Software Reset (bit 7).

0: No Interrupt
1: Interrupt exists (JPEG line buffer overflow occurred)

Be sure to set “1” when a write operation is performed.
This function just reports that the line buffer has overflowed and the JPEG modules is continuing the operation. However, the data has been destroyed by the overflow.

Bit 1: **JPEG Codec Interrupt Flag (Read only)**

This is an interrupt flag that indicates that the JPEG codec has generated an interrupt. This flag can be disabled by using the JPEG interrupt control register (bit 1). This flag is cleared by reading the JPEG operation status (bit 0).

0: No Interrupt
1: Interrupt exists (JPEG codec interrupt occurred)

Bit 0: **Reserved (Read only)**

JPEG raw status flag register

JCTL[0x08] Default value = 0x8080 Read Only

Reserved	Raw JPEG Codec File Out Status	Raw JPEG FIFO Threshold Status bits [1:0]		Raw Encode Size Limit Violation Flag	Raw JPEG FIFO Threshold Trigger Flag	Raw JPEG FIFO Full Flag	Raw JPEG FIFO Empty Flag
15	14	13	12	11	10	9	8
Reserved			Reserved	Reserved	Raw JPEG Line Buffer Overflow Flag	Raw JPEG Codec Interrupt Flag	Raw JPEG Line Buffer Interrupt Flag
7	6	5	4	3	2	1	0

Bit 15: **Reserved**

Bit 14: **Raw JPEG Codec File Out Status**

Shows the JPEG codec output state during encoding.

0: JPEG file output has been stopped

1: JPEG encoding or JPEG file output is in progress.

Bits [13:12]: **Raw JPEG FIFO Threshold Status**

Shows the current data status of the JPEG FIFO.

Table 8.8 JPEG FIFO Threshold Status

Bits [13:12]	JPEG FIFO Threshold Status
00	Empty
01	Greater than or equal to 4 bytes, less than FIFO size
10	Greater than or equal to 1/4 and less than 1/2 of the FIFO size
11	Greater than or equal to 1/2 of the FIFO size

Bit 11: **Raw Encode Size Limit Violation Flag**

This is a flag that indicates that, during JPEG encoding, the JPEG file size has exceeded the encode size limits (encode size limit register 0, 1). This flag is not affected by the JPEG interrupt control register (bit 11). Write "1" to the JPEG status flag register bit 11 to clear this flag. This function just reports that the file size has exceeded the limits and the JPEG encoding will continue.

0: No oversized encoding occurred

1: Oversized encoding occurred

Bit 10: **Raw JPEG FIFO Threshold Trigger Flag**

This is an flag that indicates that the data size of the JPEG FIFO has exceeded the JPEG FIFO threshold (JPEG FIFO control register bits 5-4) at least once. This flag is not affected by the JPEG interrupt control register (bit 10). Write "1" to the JPEG status flag register bit 10 to clear this flag.

0: No JPEG FIFO threshold violation occurred

1: JPEG FIFO threshold violation occurred

Bit 9: **Raw JPEG FIFO Full Flag**

This is a flag that indicates that the JPEG FIFO has been full at least once. This flag is not affected by the JPEG interrupt control register (bit 9). Write "1" to the JPEG status flag register bit 9 to clear this flag.

0: No JPEG FIFO full occurred

1: JPEG FIFO full occurred

Bit 8: **Raw JPEG FIFO Empty Flag**

This is a flag that indicates that the JPEG FIFO has been empty at least once. This flag is not affected by the JPEG interrupt control register (bit 8). Write "1" to the JPEG status flag register bit 8 to clear this flag.

0: No JPEG FIFO empty occurred
1: JPEG FIFO empty occurred

Bit [7:5]: **Reserved**

Bit 4: **Reserved**

Bit 3: **Reserved**

Bit 2: **Raw JPEG Line Buffer Overflow Flag**

This is a flag that indicates that the JPEG line buffer has overflowed. This flag is not affected by the JPEG interrupt control register (JCTL[0x0C] bit 2). This flag is cleared by the setting of JPEG Module Software Reset (JCTL[0x00] bit 7).

0: No JPEG line buffer overflow occurred
1: JPEG line buffer overflow occurred

Bit 1: **Raw JPEG Codec Interrupt Flag**

This is a flag that indicates that the JPEG codec has generated an interrupt. This flag is not affected by the JPEG interrupt control register (bit 1). This flag is cleared by reading the JPEG operation status register (bit 0).

0: No JPEG codec interrupt generated
1: JPEG codec interrupt generated

Bit 0: **Raw JPEG Line Buffer Interrupt Flag**

This is a flag is used in YUV data capture and indicates that a JPEG line buffer interrupt has been generated. This flag is not affected by the JPEG interrupt control register (bit 0). This flag is cleared by clearing the JPEG line buffer interrupt flag register.

0: No JPEG line buffer interrupt generated
1: JPEG line buffer interrupt generated

JPEG Interrupt Control Register

JCTL[0x0C] Default value = 0x0000 Read/Write

Reserved (0)				Encode Size Limit Violation Interrupt Enable	JPEG FIFO Threshold Trigger Interrupt Enable	JPEG FIFO Full Interrupt Enable	JPEG FIFO Empty Interrupt Enable
15	14	13	12	11	10	9	8
Reserved (0)			Reserved	Reserved (0)	JPEG Line Buffer Overflow Interrupt Enable	JPEG Codec Interrupt Enable	JPEG Line Buffer Interrupt Enable
7	6	5	4	3	2	1	0

Bits [15:12]: **Reserved**

Be sure to set each bit to “0”.

Bit 11: **Encode Size Limit Violation Interrupt Enable**

Enables encode size limit violation interrupts.

0: Disabled (default value)
1: Enabled

Bit 10: **JPEG FIFO Threshold-Triggered Interrupt Enable**

Enables JPEG FIFO threshold-triggered interrupts.

0: Disabled (default value)
1: Enabled

Bit 9: **JPEG FIFO-Full Interrupt Enable**

Enables JPEG FIFO-full interrupts.

0: Disabled (default value)
1: Enabled

Bit 8: **JPEG FIFO Empty Interrupt Enable**

Enables JPEG FIFO empty interrupts.

0: Disabled (default value)
1: Enabled

Bits [7:5]: **Reserved**

Be sure to set each bit to “0”.

Bit 4: **Reserved**

Be sure to set “0”.

Bit 3: **Reserved**

Be sure to set “0”.

Bit 2: **JPEG Line Buffer Overflow Interrupt Enable**

Enables JPEG line buffer overflow interrupts.

0: Disabled (default value)
1: Enabled

Bit 1: **JPEG Codec Interrupt Enable**

Enables JPEG Codec interrupts.

0: Disabled (default value)
1: Enabled

Bit 0: **JPEG Line Buffer Interrupt Enable**

Enables JPEG line buffer interrupts. This bit serves as a basis for controlling enable-conditions of the JPEG line buffer interrupt control register. If this bit is disabled, enabling any bit of the JPEG line buffer interrupt control register does not cause JPEG line buffer interrupts that are controlled by the JPEG line buffer interrupt control register to be enabled. This control bit is not related to the JPEG line buffer overflow interrupt.

0: Disabled (default value)
1: Enabled

JPEG Codec Start/Stop Control Register

JCTL[0x14] Default value = 0x0000 Write Only

15	14	13	12	11	10	9	8
n/a							

7	6	5	4	3	2	1	0
n/a							JPEG Start/Stop Control

Bit 0: **JPEG Start/Stop Control**

Controls the operation of the JPEG module (including YUV data capture).

For JPEG encoding

0: Cancel encoding (cancel the initiation of encoding if encoding process has not been applied)
1: Initiate encoding (encode the next frame)

On YUV data capture

0: Stops capture (stops capture after current frame)
1: Starts capture (starts capture on the subsequent frames)

Huffman Table Auto Setting Register

JCTL[0x20] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
n/a							

7	6	5	4	3	2	1	0
n/a						Huffman Table Auto Setting Non-Standby Mode	Huffman Table Auto Setting

Bit 1: **Non-Standby Mode that is triggered when automatically setting Huffman tables**

By writing a “1” in both of Bit 0 and Bit 1 with the JPEG Codec Core set to Encode, it is possible to access to registers other then the JCODEC Registers.

By writing a “1” in Bit 0 and a “0” at the same time, it is possible to access to other registers after setting up Huffman tables.

This bit is valid only when it is simultaneously written on write of “1” to Bit 0.

“0” is read.

Bit 0: **Huffman Table Auto Setting**

With the JPEG Codec core being configured for encoding, writing a 1 to this bit causes the value defined in Annex K of ISO/IEC 10918-1 to be automatically loaded into the JPEG Huffman table value (valid addresses between JCODEC[0x800] and [0xF44).

Nothing happens at times except when setting encoding operations and when “0” is written to it.

When this bit is set to “1”, JCODEC Register accesses are prohibited. (It is regarded as a dummy write.)

When the bit is active, a write access to the register address is prohibited.

This bit will be automatically reset to “0” upon completion of Huffman table setting after a “1” is written.

8.4.4 JPEG FIFO Setting Register (JFIFO[2:1])

JPEG FIFO Control Resister

JFIFO[0x40] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
Reserved (0)							
7	6	5	4	3	2	1	0
Reserved (0)		JPEG FIFO Trigger Threshold bits [1:0]		Reserved (0)	JPEG FIFO Clear	JPEG FIFO Direction (RO)	Reserved (0)

Bits [15:6]: **Reserved**

Be sure to set each bit to “0”.

Bits [5:4]: **JPEG FIFO trigger threshold**

Set the threshold that triggers JPEG FIFO.

Table 8.9 JPEG FIFO Trigger Threshold Selection

Bits [5:4]	JPEG FIFO Trigger Threshold
00	Not triggered
01	Triggered when the size exceeds 4 bytes.
10	Triggered when the value exceeds 1/4 of the FIFO size.
11	Triggered when the value exceeds 1/2 of the FIFO size.

Bit 3: **Reserved**

Be sure to set “0”.

Bit 2: **JPEG FIFO Clear**

Clears JPEG FIFOs. IF a JPEG FIFO is cleared, be sure to reset the JPEG module (bit 7 of JPEG Control Register).

0: No effect.

1: Clear JPEG FIFO

Bit 1: **JPEG FIFO Codec Direction (Read only)**

Indicates the direction of JPEG FIFO.

0: Receive (set JPEG encoding)

1: Reserved

Bit 0: **Reserved**

Be sure to set “0”.

JPEG FIFO Status Register

JFIFO[0x44] Default value = 0x8001 Read Only

Reserved							
15	14	13	12	11	10	9	8
Reserved				JPEG FIFO Threshold Status bits [1:0]		JPEG FIFO Full Status	JPEG FIFO Empty Status
7	6	5	4	3	2	1	0

Bits [15:4]: **Reserved**

Bits [3:2]: **JPEG FIFO Threshold Status**

Indicates the current data status of the JPEG FIFO.

Table 8.10 JPEG FIFO Threshold Status

Bits [3:2]	JPEG FIFO Threshold Status
00	Empty
01	Greater than or equal to 4 bytes and less than 1/4 of the FIFO size
10	Greater than or equal to 1/4 and less than 1/2 of the FIFO size
11	Greater than or equal to 1/2 of the FIFO size

Bit 1: **JPEG FIFO Full Status**

Indicates that the JPEG FIFO is full.

0: Not full.

1: Full.

Bit 0: **JPEG FIFO Empty Status**

Indicates that the JPEG FIFO is empty.

0: Not empty.

1: Empty.

JPEG FIFO Size Register

JFIFO[0x48] Default value = 0x003F Read/Write

Reserved (0)	JPEG FIFO Size bits [14:8]						
15	14	13	12	11	10	9	8
JPEG FIFO Size bits [7:0]							
7	6	5	4	3	2	1	0

Bit 15: **Reserved**

Be sure to set “0”.

Bits [14:0]: **JPEG FIFO Size**

Specify the size of JPEG FIFO in words. The maximum size of JPEG FIFO is 64 words. Since the JPEG FIFO has a dedicated RAM, the user usually specify 64 words, or the maximum size.

JPEG FIFO Size (words) = JPEG FIFO Size bits[14:0]+1

Note: The valid addresses are 0x003F (default), 0x001F, 0x000F, 0x0007, 0x0003, 0x0001, and 0x000. Any other addresses should not be set. It is recommended that the user choose the register be set to the default address (0x003F).

JPEG FIFO Read/Write Port Register

JFIFO[0x4C] Default value = 0x0000 0000 Read/Write

JPEG FIFO Read/Write Port bits [31:24]							
31	30	29	28	27	26	25	24
JPEG FIFO Read/Write Port bits [23:16]							
23	22	21	20	19	18	17	16
JPEG FIFO Read/Write Port bits [15:8]							
15	14	13	12	11	10	9	8
JPEG FIFO Read/Write Port bits [7:0]							
7	6	5	4	3	2	1	0

Bits [31:0]: **JPEG FIFO Read/Write Port**

These bits are the JPEG FIFO Read Port for JPEG encoding and YUV Data Capture.

Reserved Register

JFIFO[0x50, 0x54, 0x58] Default value = — —/—

Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0

Encode Size Limit Register 0

JFIFO[0x60] Default value = 0x0000 Read/Write

Encode Size Limit bits [15:8]							
15	14	13	12	11	10	9	8
Encode Size Limit bits [7:0]							
7	6	5	4	3	2	1	0

Encode Size Limit Register 1

JFIFO[0x64] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
Encode Size Limit bits [23:16]							
7	6	5	4	3	2	1	0

Encode Size Limit Register 1 bits [7:0]
Encode Size Limit Register 0 bits [15:0]:

Encode Size Limit bits[23:0]

Specify the limit on the size of data in a JPEG file for JPEG encoding in bytes. When a JPEG file whose size is larger than the setting in this register is encoded, an encoding size limit violation interrupt occurs. This will not affect the JPEG encoding process itself.

Encode Size Result Register 0

JFIFO[0x68] Default value = 0x0000 Read Only

Encode Size Result bits [15:8]							
15	14	13	12	11	10	9	8
Encode Size Result bits [7:0]							
7	6	5	4	3	2	1	0

Encode Size Result Register 1

JFIFO[0x6C] Default value = 0x0000 Read Only

n/a							
15	14	13	12	11	10	9	8
Encode Size Result bits [23:16]							
7	6	5	4	3	2	1	0

Encode Size Result Register 1 bits [7:0]
Encode Size Result Register 0 bits [15:0]:

Encode Size Result bits[23:0]

Indicate the size of a JPEG file in JPEG encoding. It gives the correct size only after the JPEG encoding process is completed.

Reserved Register

JFIFO[0x70, 0x74, 0x78] Default value = — —/—

Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0

8.4.5 JPEG Line Buffer Setting Register (JLB[2:1])

JPEG Line Buffer Status Flag Register

JLB[0x80] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a			Reserved	JPEG Line Buffer Empty Flag	JPEG Line Buffer Full Flag	JPEG Line Buffer Half Flag	Reserved
7	6	5	4	3	2	1	0

Bit 4: **Reserved**

Be sure to set “0”.

Bit 3: **JPEG Line Buffer Empty Flag**

This is an interrupt flag that indicates that the JPEG line buffer has reached an empty state at least once. This flag can be disabled by using the JPEG line buffer interrupt control register (bit 3).

Read:

0: Interrupt disabled

1: Interrupt enabled (JPEG line buffer empty state occurred)

Write:

0: No effect.

1: Clear.

Bit 2: **JPEG Line Buffer Full Flag**

This is an interrupt flag that indicates that the JPEG line buffer has been full at least once. This flag can be disabled by using the JPEG line buffer interrupt control register (bit 2).

Read:

0: Interrupt disabled

1: Interrupt enabled (JPEG line buffer full state occurred)

Write:

0: No effect.

1: Clear.

Bit 1: **JPEG Line Buffer Half Full Flag**

This is an interrupt flag that indicates that the JPEG line buffer has been half full at least once. This flag can be disabled by using the JPEG Line Buffer Interrupt Control Register (bit 1).

Read:

0: Interrupt disabled

1: Interrupt enabled (JPEG line buffer half full state occurred)

Write:

0: No effect.

1: Clear.

Bit 0: **Reserved**

Be sure to set “0”.

JPEG Line Buffer Row Status Flag Register

JLB[0x84] Default value = 0x0000 Read Only

n/a							
15	14	13	12	11	10	9	8
n/a			Reserved	Raw JPEG Line Buffer Empty Flag	Raw JPEG Line Buffer Full Flag	Raw JPEG Line Buffer Half Flag	Reserved
7	6	5	4	3	2	1	0

Bit 4: **Reserved**

Bit 3: **JPEG Line Buffer Empty Row Flag**

This is a flag that indicates that the JPEG line buffer has been empty at least once. This flag is not affected by the JPEG line buffer interrupt control register (bit 3).

0: No JPEG line buffer empty state occurred.
1: JPEG line buffer empty state occurred.

Bit 2: **JPEG Line Buffer Full Row Flag**

This is a flag that indicates that the JPEG line buffer has been full at least once. This flag is not affected by the JPEG line buffer interrupt control register (bit 2).

0: No JPEG line buffer full state occurred.
1: JPEG line buffer full state occurred

Bit 1: **JPEG Line Buffer Half Full Row Flag**

This is a flag that indicates that the JPEG FIFO has been half full at least once. This flag is not affected by the JPEG line buffer interrupt control register (bit 1).

0: No JPEG line buffer half full state occurred.
1: JPEG line buffer half full state occurred

Bit 0: **Reserved**

JPEG Line Buffer Current Status Flag Register

JLB[0x88] Default value = 0xX009 Read Only

n/a							
15	14	13	12	11	10	9	8
n/a			Reserved	JPEG Line Buffer Empty Current Status	JPEG Line Buffer Full Current Status	JPEG Line Buffer Half Full Current Status	Reserved
7	6	5	4	3	2	1	0

Bit 4: **Reserved**

Bit 3: **JPEG Line Buffer Empty Current Status**

Indicates whether or not the JPEG line buffer is empty.

0: Not empty.
1: Empty.

Bit 2: **JPEG Line Buffer Full Current Status**

Indicates whether or not the JPEG line buffer is full.

0: Not full.
1: Full.

Bit 1: **JPEG Line Buffer Half Full Status**

Indicates whether or not the JPEG line buffer is half full.

0: Not half full.

1: Half full.

Bit 0: **Reserved**

JPEG Line Buffer Interrupt Control Register							
JLB[0x8C] Default value = 0x0000							Read/Write
n/a							
15	14	13	12	11	10	9	8
n/a			Reserved	JPEG Line Buffer Empty Interrupt Enable	JPEG Line Buffer Full Interrupt Enable	JPEG Line Buffer Half Full Interrupt Enable	Reserved
7	6	5	4	3	2	1	0

Bit4: **Reserved**
Be sure to set “0”.

Bit 3: **JPEG Line Buffer Empty Interrupt Enable**
Enables JPEG line buffer empty interrupts.
0: Disabled (Default value)
1: Enabled

Bit 2: **JPEG Line Buffer Full Interrupt Enable**
Enables JPEG line buffer full interrupts.
0: Disabled (Default value)
1: Enabled

Bit 1: **JPEG Line Buffer Half Full Interrupt Enable**
Enables JPEG line buffer half full interrupts.
0: Disabled (Default value)
1: Enabled

Bit 0: **Reserved**
Be sure to set “0”.

JPEG Line Buffer Horizontal Pixel Allowable Size Register							
JLB[0xA0] Default value = 0x2800							Read/Write
Horizontal Support Size bits [10:4]							
15	14	13	12	11	10	9	8
Horizontal Support Size bits [3:0]				n/a	Horizontal Support Size Setting bits [2:0]		
7	6	5	4	3	2	1	0

Bits [15:4]: **JPEG Line Buffer Horizontal Pixel Allowable Size (read only)**
Represent the horizontal allowable size specified by bits [2:0].

Bits [2:0]: **JPEG Line Buffer Horizontal Pixel Allowable Size Setting**
Specify the horizontal pixel size that is accepted by the JPEG line buffer. When using this register with a value other than its default value, it is required to modify the setting in JLB[0xA4] and in the appropriate memory allocation register in the system controller.
000: Allows up to 640 horizontal pixels (default value).
001: Allows up to 800 horizontal pixels.
010: Allows up to 1024 horizontal pixels.
011: Allows up to 1280 horizontal pixels.
100: Allows up to 1600 horizontal pixels.
101-111: Not available for setting.

JPEG Line Buffer Memory Address Offset Register

JLB[0xA4] Default value = 0x0020 Read/Write

15	14	13	12	11	10	9	8
Reserved (0)							
7	6	5	4	3	2	1	0
Reserved (0)	JPG-LB Memory Address Offset bits [6:0]						

Bits [15:7]: **Reserved**

Be sure to set “0”.

Bits [6:0]: **JPEG Line Buffer Memory Address Offset**

Specify the address offset of the internal memory that is used by the JPEG line buffer in multiples of 1 KB. When using this register with a value other than its default value, it is required to modify the appropriate memory allocation register in the system controller.

JPEG Line Buffer Read/Write Port Register

JLB[0xC0] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
JPEG Line Buffer Read/Write Port bits [31:24]							
23	22	21	20	19	18	17	16
JPEG Line Buffer Read/Write Port bits [23:16]							
15	14	13	12	11	10	9	8
JPEG Line Buffer Read/Write Port bits [15:8]							
7	6	5	4	3	2	1	0
JPEG Line Buffer Read/Write Port bits [7:0]							

Bits [31:0]: **JPEG Line Buffer Read/Write Port**

Reserved

8.4.6 JPEG Codec Registers (JCODEC[2:1])

Operation Mode Setting Register

JCODEC[0x00] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a			Reserved (0)	Marker Insert Enable	JPEG Operation Select	YUV Format Select bits [1:0]	
7	6	5	4	3	2	1	0

Bit 4: **Reserved**

Be sure to set “0”.

Bit 3: **Marker Insert Enable**

This bit enables insertion of data written in the Insert Marker Data Register in a JPEG file.

0: Disabled (marker not inserted) (default value)

1: Enabled (marker inserted)

Bit 2: **JPEG Operation Selection**

Specifies the JPEG operation mode.

Table 8.11 JPEG Operation Selection

Bit 2	JPEG Operation
0 (Default value)	Encode
1	Reserved

Bits [1:0]: **YUV Format Selection**

Specify the YUV data format for JPEG encoding. JPEG encoding in YUV 4:4:4 format is not available if scaling has not been performed by the resizer.

Table 8.12 YUV Format Selection

Bits [1:0]	YUV Format
00 (default value)	4:4:4
01	4:2:2
10	4:2:0
11	4:1:1

Command Setting Register

JCODEC[0x04] Default value = not applicable Write Only

15	14	13	12	11	10	9	8
n/a							

7	6	5	4	3	2	1	0
JPEG Codec SW Reset	n/a						JPEG Operation Start

The user should not attempt to read this register. Also, do not attempt to write it when the JPEG controller is in operation. (A reset operation is allowed.)

Bit 7: **JPEG Codec Software Reset**

This bit resets the JPEG Codec through software. It does not reset the JPEG Codec registers.

0: No effect.
1: Reset

Bit 0: **JPEG Operation Start**

Initiates the JPEG operation (including YUV Data Capture).

0: No effect.
1: Start JPEG operation.

JPEG Operation Status Register

JCODEC[0x08] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
n/a							

7	6	5	4	3	2	1	0
n/a							JPEG Operation Status (RO)

Bit 1 of JPEG Status Flag Register and bit 1 of JPEG Raw Status Flag Register are cleared by reading this register.

Bit 0: **JPEG Operation Status.**

Indicates the operating state of the JPEG Codec.

0: Stopped.
1: JPEG encoding.

Quantization Table Number Register

JCODEC[0x0C] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
n/a							

7	6	5	4	3	2	1	0
n/a					V Table Select	U Table Select	Y Table Select

Bit 2: **V Table Select**

Specifies the quantization table number used for V component in JPEG Encoding Mode.

0: Use quantization table 0 (default value).
1: Use quantization table 1.

Bit 1: **U Table Select**

Specifies the quantization table number used for U component in JPEG Encoding Mode.

0: Use quantization table 0 (default value).
1: Use quantization table 1.

Bit 0: **Y-Component Table Select**

Specifies the quantization table number used for Y component in JPEG Encoding Mode.

0: Use quantization table 0 (default value).
1: Use quantization table 1.

Huffman Table Number Register

JCODEC[0x10] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a		V ACTable Select	V DCTable Select	U ACTable Select	U DCTable Select	Y ACTable Select	Y DCTable Select
7	6	5	4	3	2	1	0

Bit 5: **V AC Table Select**

Specifies the AC Huffman table number used for V component in JPEG Encoding Mode. Set this bit to “1” for the JPEG Fast Encoding Mode.

0: Use AC Huffman table 0 (default value).
1: Use AC Huffman table 1.

Bit 4: **V DC Table Select**

Specifies the DC Huffman table number used for V component in JPEG Encoding Mode. Set this bit to “1” for the JPEG Fast Encoding Mode.

0: Use DC Huffman table 0 (default value).
1: Use DC Huffman table 1.

Bit 3: **U AC Table Select**

Specifies the AC Huffman table number used for U component in JPEG Encoding Mode. Set this bit to “1” for the JPEG Fast Encoding Mode.

0: Use AC Huffman table 0 (default value).
1: Use AC Huffman table 1.

Bit 2: **U DC Table Select**

Specifies the DC Huffman table number used for U component in JPEG Encoding Mode. Set this bit to “1” for the JPEG Fast Encoding Mode.

0: Use DC Huffman table 0 (default value).
1: Use DC Huffman table 1.

Bit 1: **Y AC Table Select**

Specifies the AC Huffman table number used for Y component in JPEG Encoding Mode. Set this bit to “0” for the JPEG Fast Encoding Mode.

0: Use AC Huffman table 0 (default value).
1: Use AC Huffman table 1.

Bit 0: **Y DC Table Select**

Specifies the DC Huffman table number used for Y component in JPEG Encoding Mode. Set this bit to “0” for the JPEG Fast Encoding Mode.

0: Use DC Huffman table 0 (default value).
1: Use DC Huffman table 1.

Set DRI Register 0

JCODEC[0x14] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
DRI Value bits [15:8]							
7	6	5	4	3	2	1	0

Set DRI Register 1

JCODEC[0x18] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
DRI Value bits [7:0]							
7	6	5	4	3	2	1	0

Set DRI Register 0 bits [7:0]
Set DRI Register 1 bits [7:0]:

DRI Value bits [15:0]

Specify the number of MCUs into which a RST marker is inserted in JPEG encoding. (If these bits are set to “0”, a RST marker is not inserted, but a “0” is inserted into the Define RST Interval Marker as the value defining the intervals.

Vertical Pixel Size Register 0

JCODEC[0x1C] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
Y Pixel Size bits [15:8]							
7	6	5	4	3	2	1	0

Vertical Pixel Size Register 1

JCODEC[0x20] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
Y Pixel Size bits [7:0]							
7	6	5	4	3	2	1	0

Vertical Pixel Size Register 0 bits [7:0]
Vertical Pixel Size Register 1 bits [7:0] :

Y Pixel Size bits[15:0]
These bits specify the image size in the vertical direction during JPEG encoding and YUV Data Capture.

Note: These bits are write-only bits when the YUV Data Capture mode is specified (JCTL [0x00], Bits 3-1 = 011 or 111). A read on the register during that time yields an unknown value.

Horizontal Pixel Size Register 0

JCODEC[0x24] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
X Pixel Size bits [15:8]							
7	6	5	4	3	2	1	0

Horizontal Pixel Size Register 1

JCODEC[0x28] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
X Pixel Size bits [7:0]							
7	6	5	4	3	2	1	0

Horizontal Pixel Size Register 0 bits [7:0]
Horizontal Pixel Size Register 1 bits [7:0] :

X Pixel Size bits[15:0]
These bits specify the image size in the horizontal direction during JPEG encoding and YUV Data Capture.

Note: These bits are write-only bits when the YUV Data Capture mode is specified (JCTL [0x00], Bits 3-1 = 011 or 111). A read on the register during that time yields an unknown value.

Reserved Register
JCODEC[0x2C-34]

Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0

RST Marker Operation Setting Register
JCODEC[0x38] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a						RST Marker Operation Select bits [1:0]	
7	6	5	4	3	2	1	0

Bits [1:0]: **RST Marker Operation Select**
Reserved

RST Marker Operation Status Register
JCODEC[0x3C] Default value = 0x0000 Read Only

n/a							
15	14	13	12	11	10	9	8
Revise Code	JPEG Error Status bits [3:0]				n/a		
7	6	5	4	3	2	1	0

Bit 7: **Revise Code**
Reserved

Bits [6:3]: **JPEG Decode Error Status**
Reserved

Insert Marker Data Register
JCODEC[0x40-0xCC] Default value = 0x00FF Read/Write

n/a							
15	14	13	12	11	10	9	8
Insert marker Data bits [7:0]							
7	6	5	4	3	2	1	0

A 36-byte register that inserts a marker in JPEG Encoding Mode. All of 36 bytes are inserted regardless of the length to the marker.

Address Offset [40h-44h]: Specify the marker code to be inserted.

Address Offset [48h-4Ch]: Specify the length of the marker within a range from 0002h to 0022h.

Address Offset [50h-CCh]: Specify the marker data (up to 32 bytes). The data portion exceeding the length of marker should be padded with FFh.

Quantization Table No.0 Register

JCODEC[0x400-0x4FC] Default value = not applicable Read/Write

n/a							
15	14	13	12	11	10	9	8
Quantization Table No. 0 bits [7:0]							
7	6	5	4	3	2	1	0

Quantization Table No.0

Specifies the quantization table value that corresponds to table number 0 used for JPEG encoding.

Quantization Table No.1 Register

JCODEC[0x500-0x5FC] Default value = not applicable Read/Write

n/a							
15	14	13	12	11	10	9	8
Quantization Table No. 1 bits [7:0]							
7	6	5	4	3	2	1	0

Quantization Table No.1

Specifies the quantization table value that corresponds to table number 1 used for JPEG encoding.

DC Huffman Table No.0 Register 0

JCODEC[0x800-0x83C] Default value = not applicable Write Only

n/a							
15	14	13	12	11	10	9	8
DC Huffman Table No. 0 Register 0 bits [7:0]							
7	6	5	4	3	2	1	0

DC Huffman Table No.0

Specifies the DC Huffman table value that corresponds to table number 0 used for JPEG encoding. It specifies the number of symbols for each length in the Huffman table. This bit is not used when Huffman tables are automatically setup.

DC Huffman Table No. 0 Register 1

JCODEC[0x840-0x86C] Default value = not applicable Write Only

n/a							
15	14	13	12	11	10	9	8
Reserved (must be all 0)				DC Huffman Table No. 0 Register 1 bits [3:0]			
7	6	5	4	3	2	1	0

DC Huffman Table No.0

Specifies the DC Huffman table value that corresponds to table number 0 used for JPEG encoding. It specifies a group number of all symbols in order of probability of occurrence. Use the low-order 4 bits to specify the value. The high-order 4 bits must be set to “0000”. This bit is not used when Huffman tables are automatically setup.

AC Huffman Table No. 0 Register 0

JCODEC[0x880-0x8BC] Default value = not applicable Write Only

n/a							
15	14	13	12	11	10	9	8
AC Huffman Table No. 0 Register 0 bits [7:0]							
7	6	5	4	3	2	1	0

AC Huffman Table No.0

Specifies the AC Huffman table value that corresponds to table number 0 used for JPEG encoding. It specifies the number of symbols for each length in the Huffman table. This bit is not used when Huffman tables are automatically setup.

AC Huffman Table No.0 Register 1

JCODEC[0x8C0-0xB44] Default value = not applicable Write Only

n/a							
15	14	13	12	11	10	9	8
AC Huffman Table No. 0 Register 0 bits [7:0]							
7	6	5	4	3	2	1	0

AC Huffman Table No.0

Specifies the AC Huffman table value that corresponds to table number 0 used for JPEG encoding. It specifies a run-length of zeros and group number of all symbols in order of probability of occurrence. This bit is not used when Huffman tables are automatically setup.

DC Huffman Table No. 1 Register 0

JCODEC[0xC00-0xC3C] Default value = not applicable Write Only

n/a							
15	14	13	12	11	10	9	8
DC Huffman Table 1 Register No. 0 bits [7:0]							
7	6	5	4	3	2	1	0

DC Huffman Table No.1

Specifies the DC Huffman table value that corresponds to table number 1 used for JPEG encoding. It specifies the number of symbols for each length in the Huffman table. This bit is not used when Huffman tables are automatically setup.

DC Huffman Table No.1 Register 1

JCODEC[0xC40-0xC6C] Default value = not applicable Write Only

n/a							
15	14	13	12	11	10	9	8
Reserved (must be all 0)				DC Huffman Table No. 1 Register 1 bits [3:0]			
7	6	5	4	3	2	1	0

DC Huffman Table No.1

Specifies the DC Huffman table value that corresponds to table number 1 used for JPEG encoding. It specifies a group number of all symbols in order of probability of occurrence. Use the low-order 4 bits to specify the value. The high-order 4 bits must be set to “0000”. This bit is not used when Huffman tables are automatically setup.

AC Huffman Table No.1 Register 0

JCODEC[0xC80-0xCBC] Default value = not applicable Write Only

n/a							
15	14	13	12	11	10	9	8
AC Huffman Table No. 1 Register 0 bits [7:0]							
7	6	5	4	3	2	1	0

AC Huffman Table No.1

Specifies the AC Huffman table value that corresponds to table number 1 used for JPEG encoding. It specifies the number of symbols for each length in the Huffman table. This bit is not used when Huffman tables are automatically setup.

AC Huffman Table No.1 Register 1

JCODEC[0xCC0-0xF44] Default value = not applicable Write Only

n/a							
15	14	13	12	11	10	9	8
AC Huffman Table No. 1 Register 0 bits [7:0]							
7	6	5	4	3	2	1	0

AC Huffman Table No.1

Specifies the AC Huffman table value that corresponds to table number 1 used for JPEG encoding. It specifies a run-length of zeros and group number of all symbols in order of probability of occurrence. This bit is not used when Huffman tables are automatically setup.

8.5 Explanation of Operations

8.5.1 Capture Control

The device controls data capture using state machines. This is because JPEG encoding of camera images or YUV capture requires frame by frame capture and it is expected that restrictions on time will be relaxed by controlling it through hardware.

For state machines, conditions for state transitions and states have different meaning depending upon the JPEG encoding of camera images and YUV capture.

The current state of a state machine can be read from the relevant register. Since it is a signal that changes in real time, it is desired to read it for debugging purposes.

8.5.1.1 State Machine for JPEG Encoding of Camera Image

Fig.8.2 illustrates a state machine for controlling capture of camera images in JPEG encoding mode.

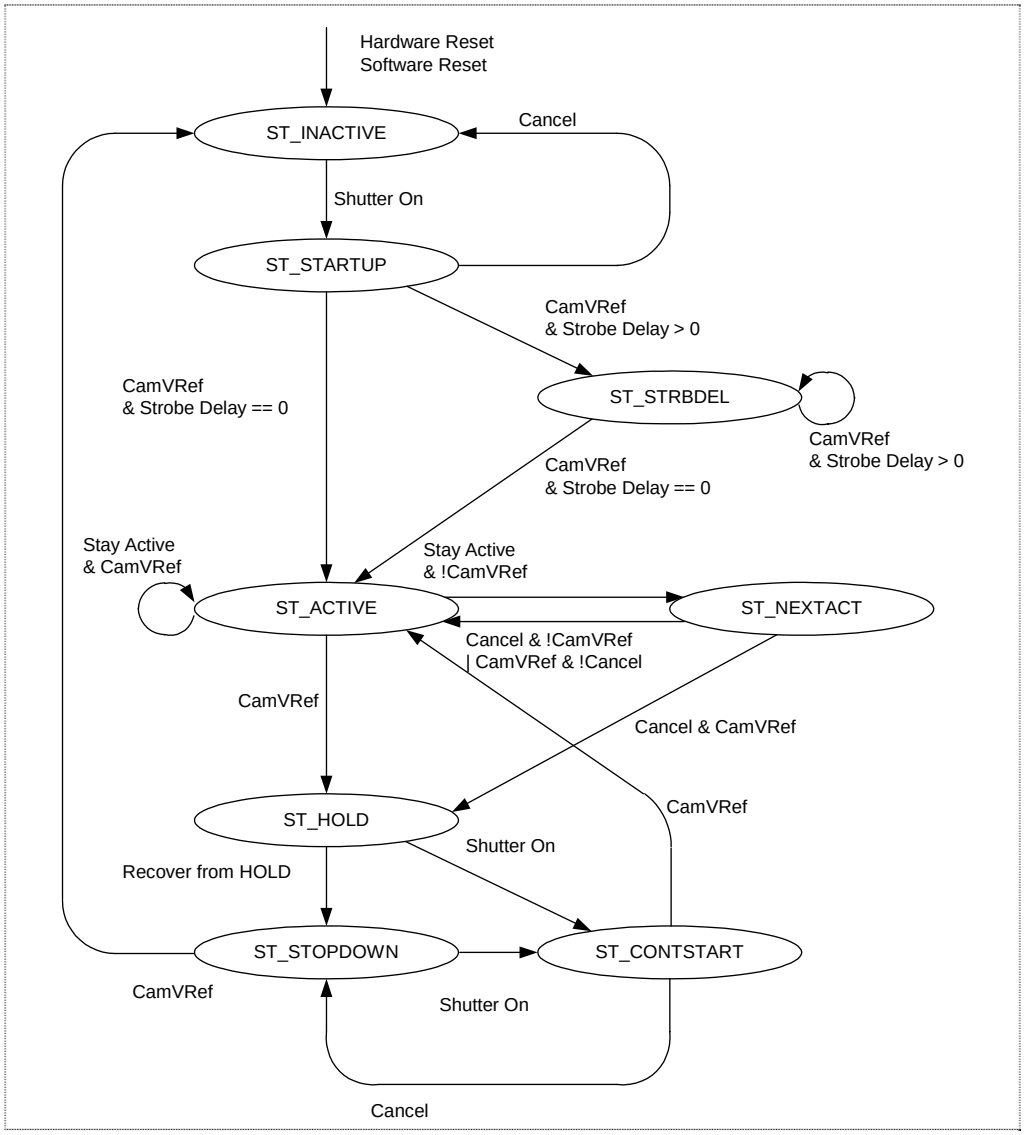


Fig.8.2 State Machine for Camera Image JPEG Encode

The meaning of each state is described in the table below. State IDs are values that are read by reading the appropriate registers.

Table 8.13 State Description in Camera Image JPEG Encode Mode

State	Description
ST_INACTIVE State ID (0x0)	An initial state at which the state machine has not captured camera data yet.
ST_STARTUP State ID (0x1)	A state changes from ST_INACTIVE to a state at which a shutter button is pressed, but the state machine has not captured camera data yet.
ST_STRBDEL State ID (0x3)	The state machine waits for Strobe Frame Delay. If Strobe Delay is detected, the state machine causes Capture to be delayed by the number of frames specified before going into ST_ACTIVE. The state machine has not captured camera data yet.
ST_ACTIVE State ID (0x7)	A state at which the state machine captures camera data. In ST_ACTIVE, a state transition only occurs if a VREF sent from the camera is detected. This ensures that frame data is captured as long as trimming and scaling parameters are correctly specified.
ST_NEXTACT State ID (0xB)	The state machine goes to this state by asserting Request Next Frame Capture while ST_ACTIVE is triggered. It goes back to ST_ACTIVE as soon as a VREF from the camera is detected. This state is provided to encode frames in succession. The state machine has captured camera data.
ST_HOLD State ID (0x6)	The state machine goes to this state for the next frame which has been captured. The state machine has not captured camera data yet.
ST_STOPDOWN State ID (0x4)	The state machine receives the Reset Hold signal in response to ST_HOLD. It transitions to ST-INACTIVE on the subsequent frames, whereby the HOLD state is cleared. The state machine has not captured camera data yet.
ST_CONTSTART State ID (0x5)	The state machine initiates capturing of frames when “Shutter press” is detected in HOLD state. The state machine has not captured camera data yet.

The association between events and actions are shown below.

Table 8.14 Event Description in Camera Image JPEG Encode Mode

Event	Action
Software Reset	Reset the resizer through software.
Shutter On	Write a “1” to the JPEG Start/Stop Register.
Cancel, Recover from HOLD	Write a “0” to the JPEG Start/Stop Register.
CamVRef	The timing of transition of camera image data's VREF data level from active to inactive. The term “VREF” used alone in this section refers to this timing.
Stay Active	Write a “1” to the Request Next Frame bit of JPEG Control Register.
Strobe Delay	Strobe Frame Delay Count.

Fig.8.3 shows a timing chart for a single JPEG encoding sequence of a camera image.

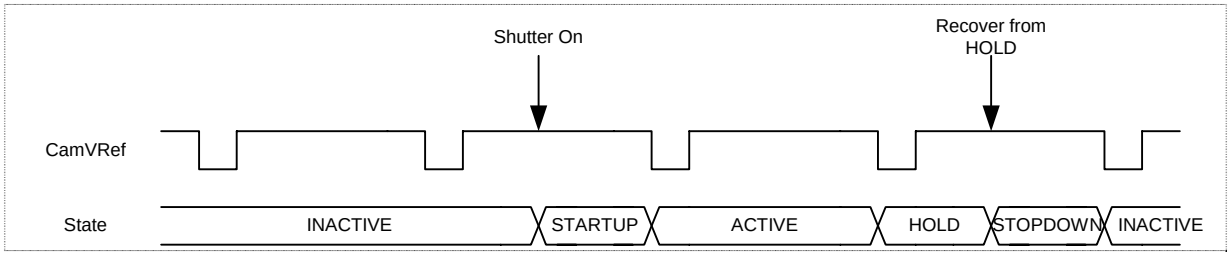


Fig.8.3 Timing Chart for Camera Image JPEG Encode (Single)

8.5.1.2 State Machine for YUV Capture

Fig.8.4 illustrates a state machine for controlling the capture in the YUV capture mode.

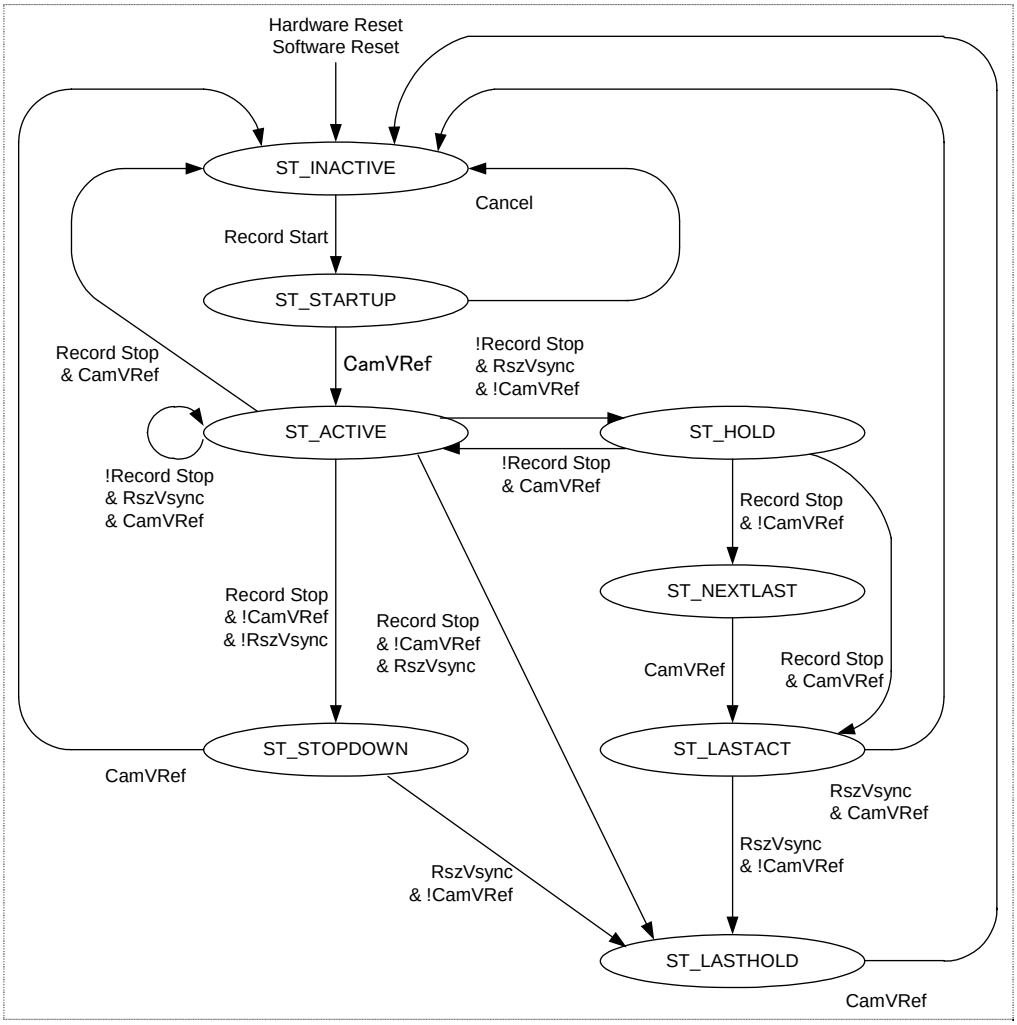


Fig.8.4 State Machine for YUV Capture

Table 8.15 State Description in YUV Capture Mode

State	Description
ST_INACTIVE State ID (0x0)	An initial state at which the state machine has not captured camera data yet.
ST_STARTUP State ID (0x1)	The state machine transitions from ST_INACTIVE to a state at which it receives a Record Start signal. The state machine has not captured camera data yet.
ST_ACTIVE State ID (0x7)	A state at which the state machine captures camera data. The state machine has captured camera data.
ST_HOLD State ID (0x6)	The state machine transitions from ST_ACTIVE to ST_HOLD when a Capture Data VREF is generated in the frame. The state machine has not captured camera data yet.
ST_STOPDOWN State ID (0x4)	The state machine transitions from ST_INACTIVE to a state at which it receives a Record Stop signal. Capture process completes with the current frame. The state machine has captured camera data.
ST_NEXTLAST State ID (0xE)	The state machine goes to this state upon reception of a Record Stop signal while ST_HOLD is triggered. It is not possible for the JPEG module to recognize the completion of motion image capture between the moment a Capture Data VREF is generated and the moment a Camera VREF is generated. To resolve this problem, this state is provided to let the Record Stop signal capture another one frame after receiving a Capture Data VREF. The state machine has not captured camera data yet.
ST_LASTACT State ID (0xF)	The state machine goes to a state at which it captures another frame as described above. The state machine has captured camera data.
ST_LASTHOLD State ID (0xD)	The state machine transitions to ST_LASTHOLD when a Capture Data VREF is generated in the last recording frame. The state machine has not captured camera data yet.

The association between events and actions are shown below.

Table 8.16 Event Description in YUV Capture Mode

Event	Action
Software Reset	Reset the resizer through software.
Record Start	Write a “1” to the JPEG Start/Stop Register.
Cancel, Record Stop	Write a “0” to the JPEG Start/Stop Register.
CamVRef	The timing of transition of camera image data's VREF data level from active to inactive. The term “VREF” used alone in this section refers to this timing.
RszVsync	An end-of-frame signal that is output by the resizer.

8.5.2 Resizing

The resizer performs trimming and downsizing for image data sent from a camera interface. It has two stages – trimming and downsizing. Since image data sent from a camera module in YUV 4:2:2 format is converted to YUV 4:4:4 format by a camera interface, trimming can be performed on a single pixel basis (1x1) by the resizer.

8.5.2.1 Trimming

Trimming is performed as the preprocessing stage for scaling to trim the edges of an image to a size required for scaling process. The trimming process uses the register that contains starting X and Y coordinates and ending X and Y coordinates. The upper left corner of the original image is defined as coordinates (0, 0). The starting coordinates correspond to the upper left corner of a trimmed image and the ending coordinates correspond to the upper right corner of a trimmed image. These settings can be specified on a single pixel basis (a pixel matrix of 1 x 1), they cannot be greater than the maximum coordinate value of an image from a camera. If a specified ending coordinate value is less than a starting coordinate value, the resizer fails to operate correctly, outputting some piece of data. For this reason, be sure to specify ending coordinates greater than starting coordinates. To forcefully stop data output from the resizer, disable the circuit by writing a 0 to RSZ[0xC0], bit 0.

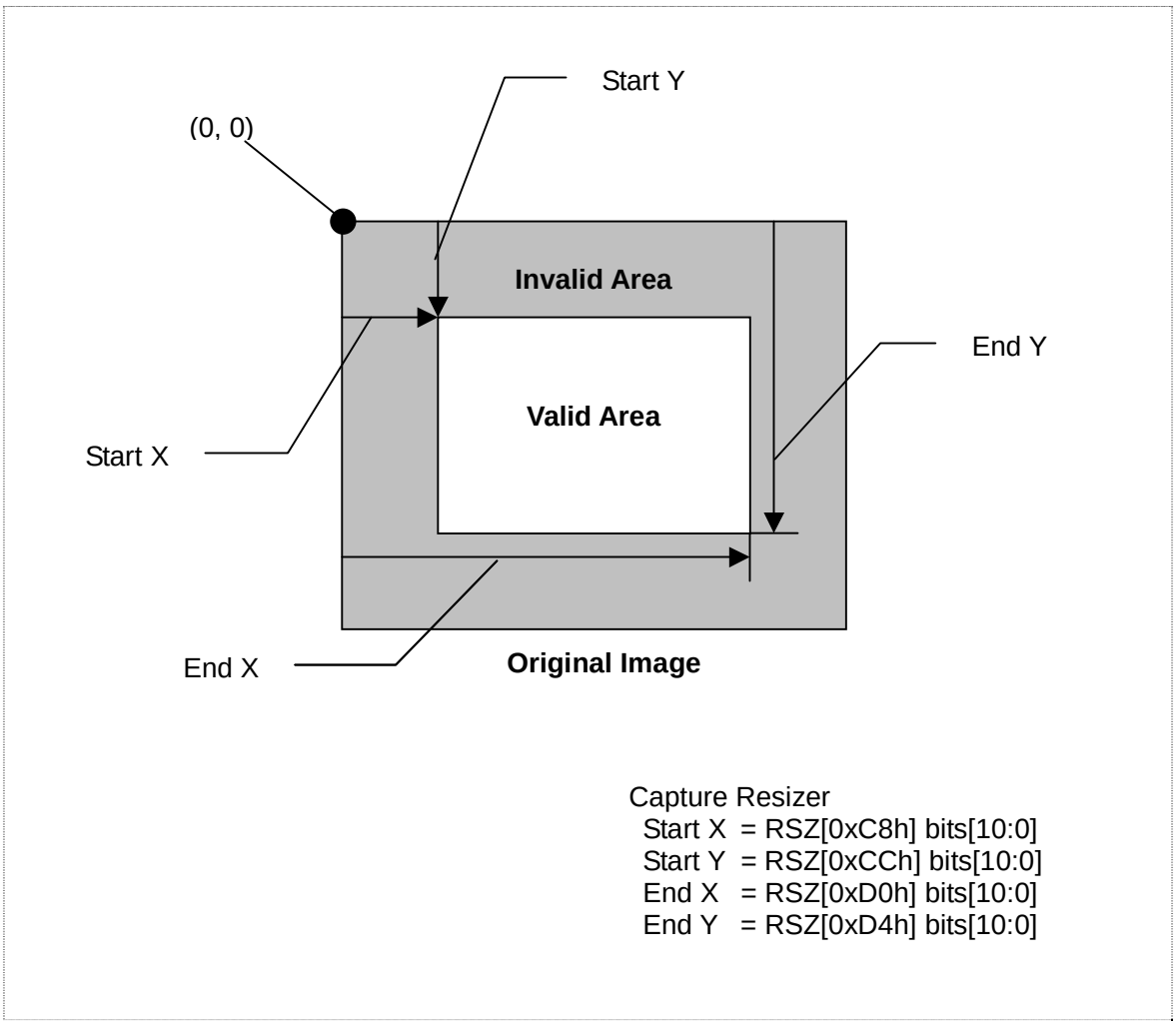


Fig.8.5 Trimming Function

8.5.2.2 Scaling

Scaling is performed as the postprocessing stage of trimming to reduce the image by a factor specified for the trimmed image. The size of the image that was scaled down must be identical to the one to be specified in JPEG Encoding Mode.

The scaling process has three scaling modes – “Downsizing“, “Averaging” and “No Scaling”. In Downsizing mode, select a representative point from an image block and use it for a single pixel of a scaled down image. The scaling can be set by selecting any scaling factor between 1/2 and 1/8. In Averaging mode, the scaling in the vertical direction is achieved by downsizing and the scaling in the horizontal direction is achieved by averaging. For an image that is selected for downsizing, all pixels in each column of an image block are added and their average value is used for a single pixel in the scaled down image block. The scaling factors available in Averaging mode are 1/2, 1/4 and 1/8. Specify No Scaling mode when images are not to be scaled down. In No Scaling mode, the value set in the Set Scaling Factor Register becomes invalid. If a “1” is written in the above register in any other modes, operations are not guaranteed. Be sure to select the No Scaling mode when images are not to be scaled down.

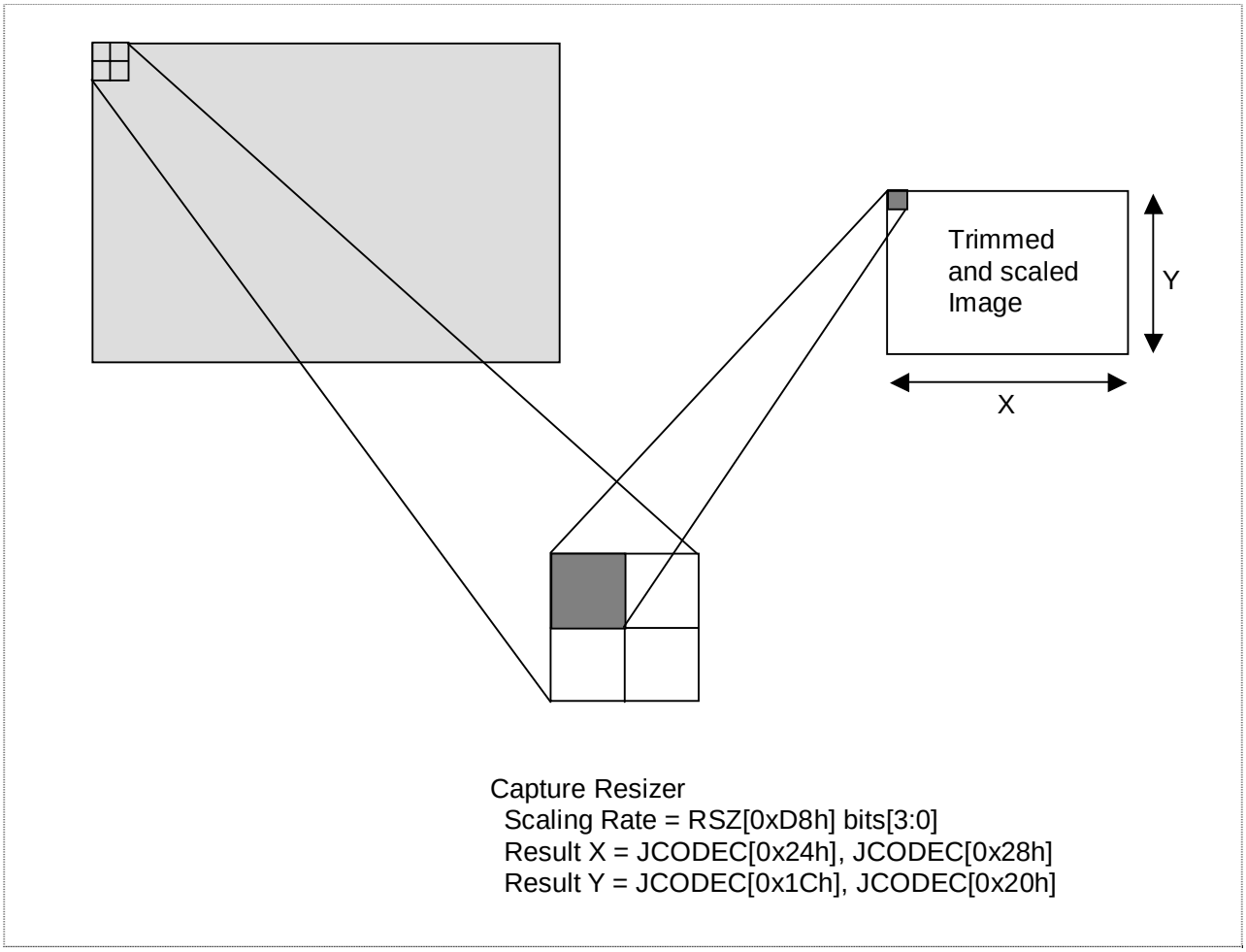


Fig.8.6 Scaling Example (1/2 Scaling)

Scaling Down by a Factor of 1/2

Scaling down by a factor of 1/2 reduces every 2 x 2 pixel block to one pixel. For this scaling factor, both of two scaling modes - Downsizing and Averaging - are available. Note that, for both Downsizing and Averaging modes, the scaling in the vertical direction is fixed to Downsizing mode.

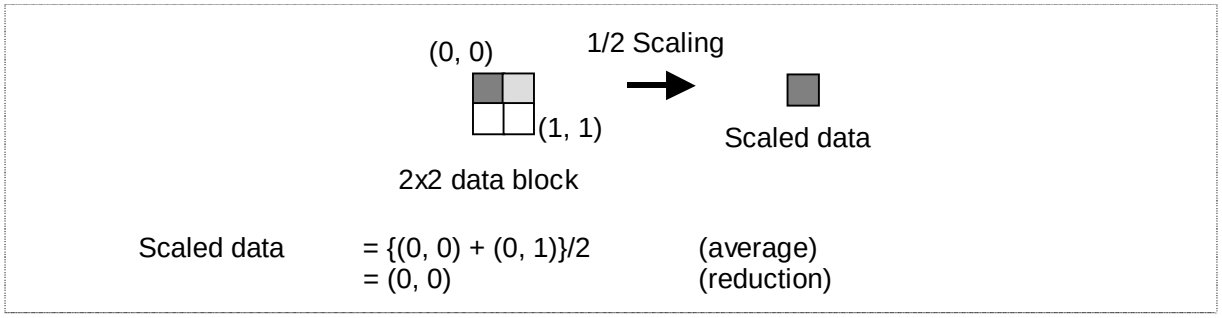


Fig.8.7 1/2 Compression

Scaling Down by a Factor of 1/3

Scaling down by a factor of 1/3 reduces every 3 x 3 pixel block to one pixel. For this scaling factor, only Downsizing mode is available.

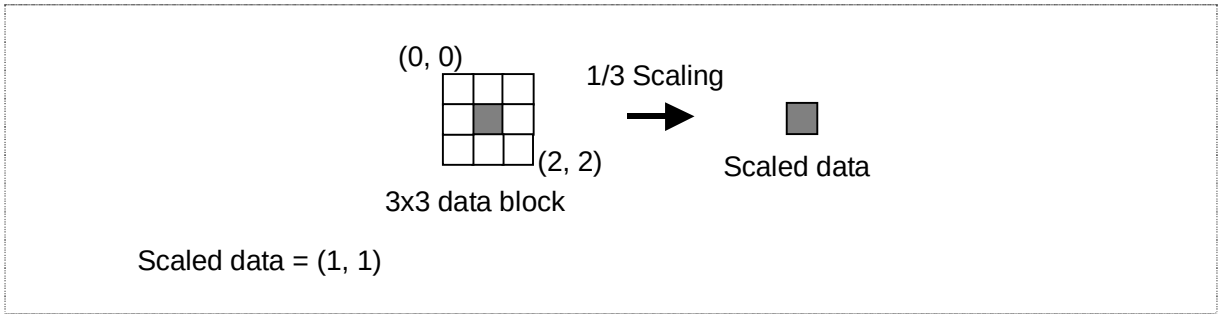


Fig.8.8 1/3 Scaling

Scaling Down by a Factor of 1/4

Scaling down by a factor of 1/4 reduces every 4 x 4 pixel block to one pixel. For this scaling factor, both of two scaling modes - downsizing and averaging - are available. Note that, for both Downsizing and Averaging modes, the scaling in the vertical direction is fixed to Downsizing mode.

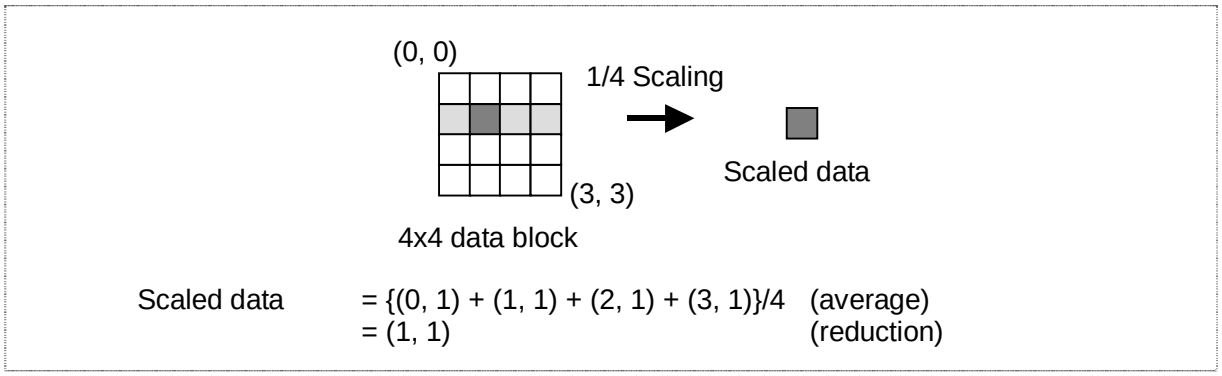


Fig.8.9 1/4 Scaling

Scaling Down by a Factor of 1/5

Scaling down by a factor of 1/5 reduces every 5 x 5 pixel block to one pixel. For this scaling factor, only Downsizing mode is available.

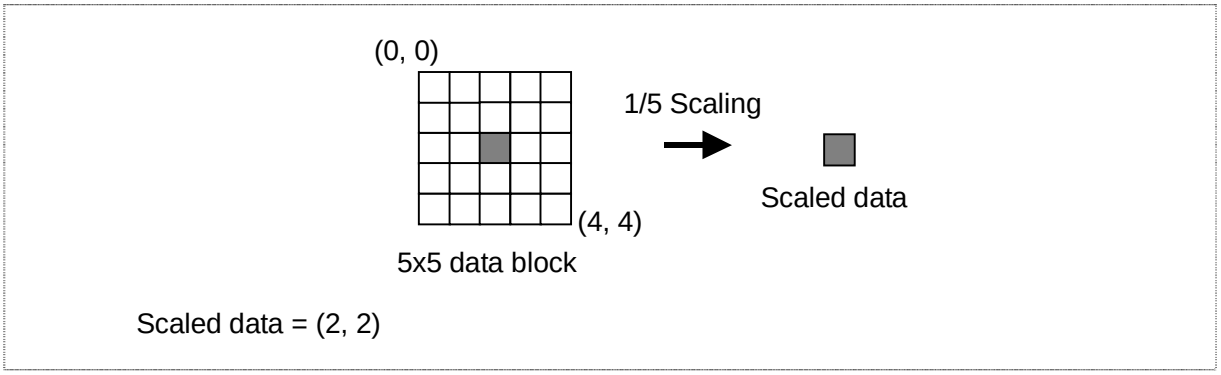


Fig.8.10 1/5 Scaling

Scaling Down by a Factor of 1/6

Scaling down by a factor of 1/6 reduces every 6 x 6 pixel block to one pixel. For this scaling factor, only Downsizing mode is available.

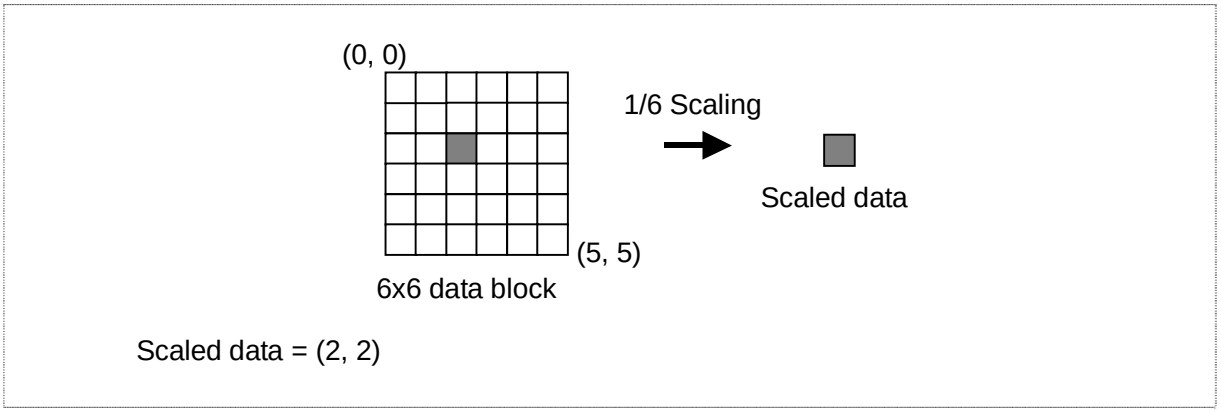


Fig.8.11 1/6 Scaling

Scaling Down by a Factor of 1/7

Scaling down by a factor of 1/7 reduces every 7 x 7 pixel block to one pixel. For this scaling factor, only Downsizing mode is available.

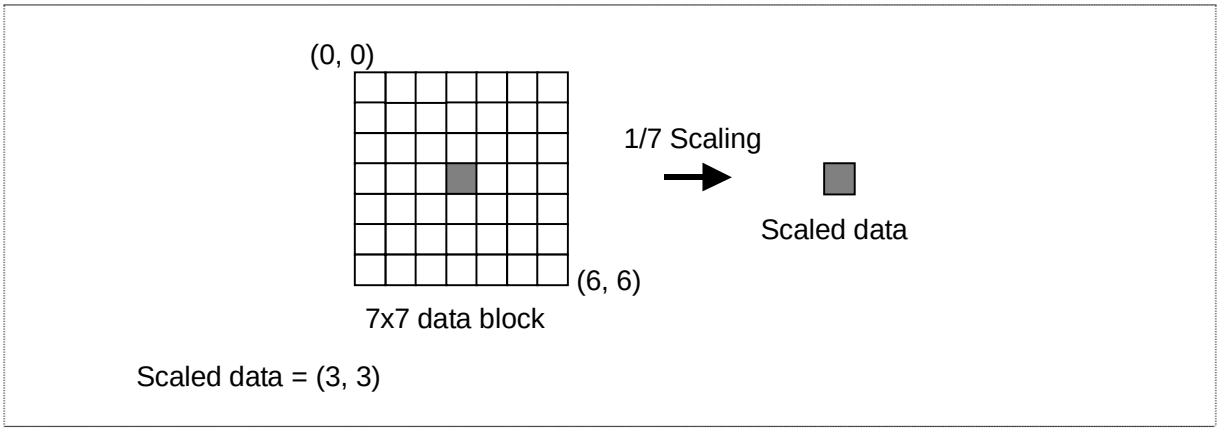


Fig.8.12 1/7 Scaling

Scaling Down by a Factor of 1/8

Scaling down by a factor of 1/8 reduces every 8 x 8 pixel block to one pixel. For this scaling factor, both of two scaling modes - Downsizing and Averaging - are available. Note that, for both Downsizing and Averaging modes, the scaling in the vertical direction is fixed to Downsizing mode.

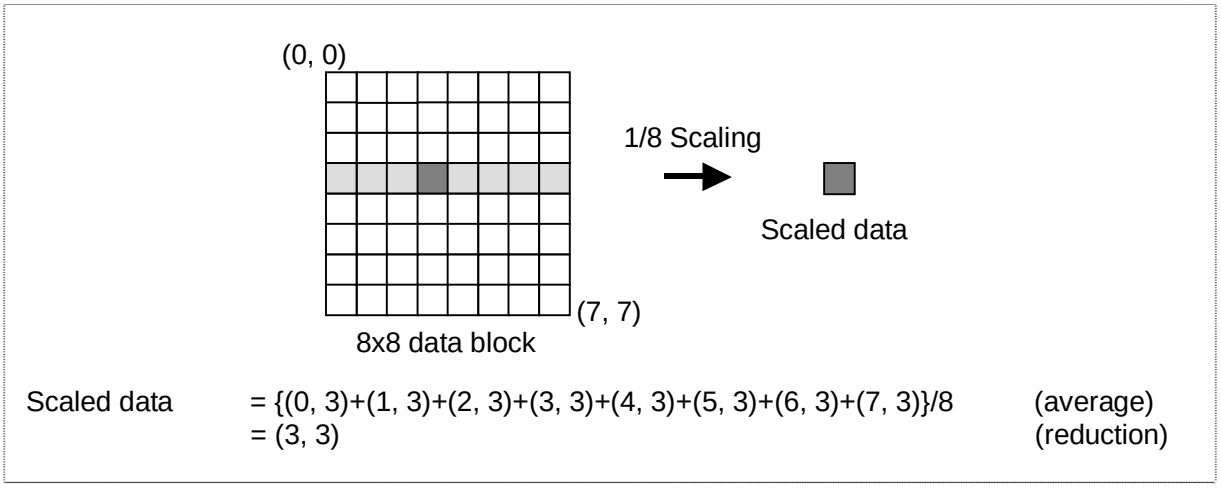


Fig.8.13 1/8 Scaling

8.5.2.3 Restrictions on the Use of Resizer

It is not possible to change the registers in the resizer, except Software Rest or Enable, when data is being transmitted from a camera interface. Any attempt to do so results in data corruption. To change settings of registers in the resizer when a camera is operating, it is common to change them in a period of time during which the camera interface is not receiving data from the image sensor between frames (VREF data bank period) using a VREF interrupt from the camera interface.

It is not possible to specify any values greater than those values for a camera image in any of the Resizer Start X, Start Y, End X and End Y Coordinates Registers. If such a value is specified, the resizer will not operate correctly.

An image that is trimmed by specifying the Resizer Start X, Start Y, End X and End Y Coordinates Registers must be divisible by a value specified in the Set Scaling Factor Register.

8.5.3 Data Flow of Image Processing

This section provides a diagram showing the data flow in various image processing modes.

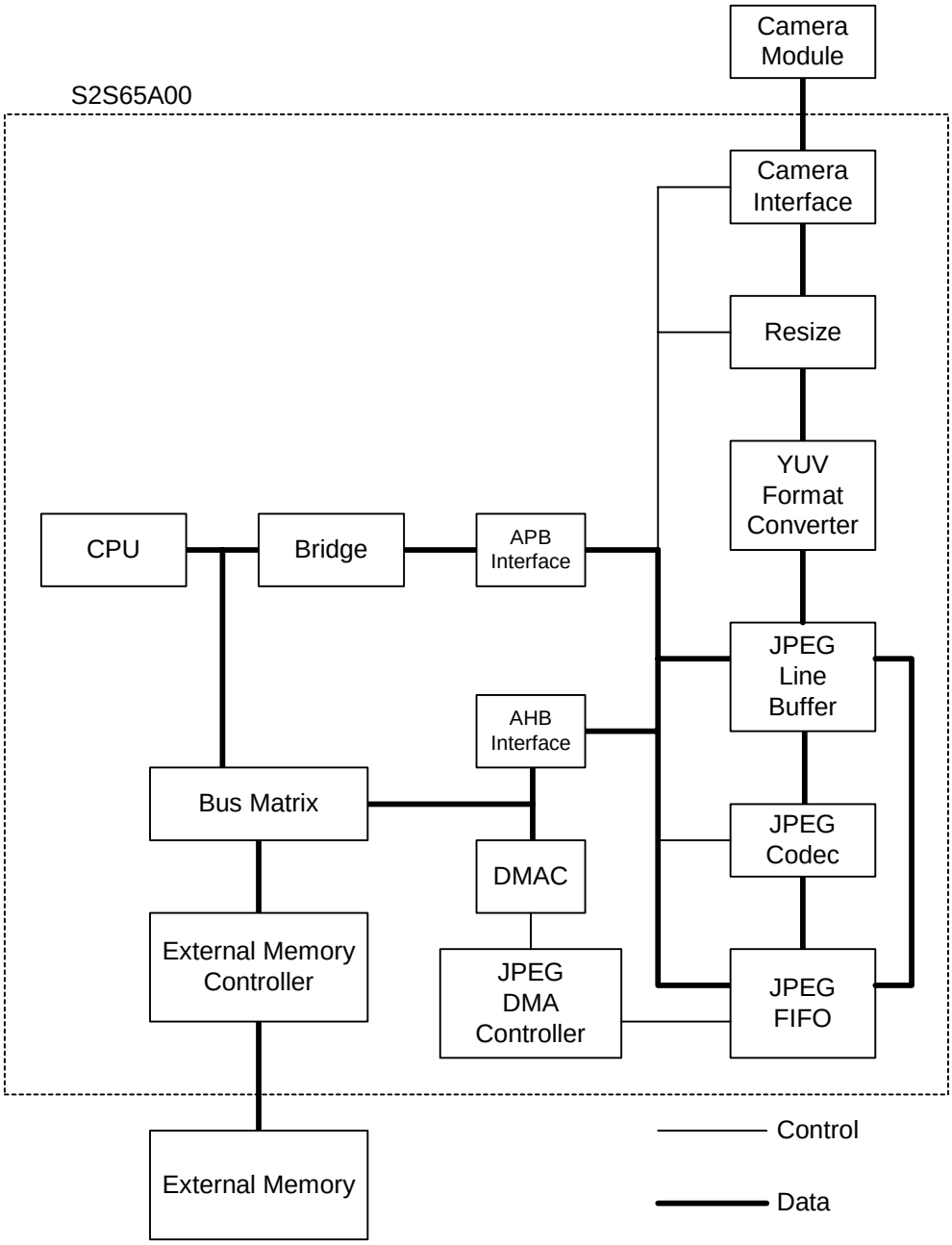


Fig.8.14 Data Flow of Image Processing

8.5.3.1 Camera Image JPEG Encoding

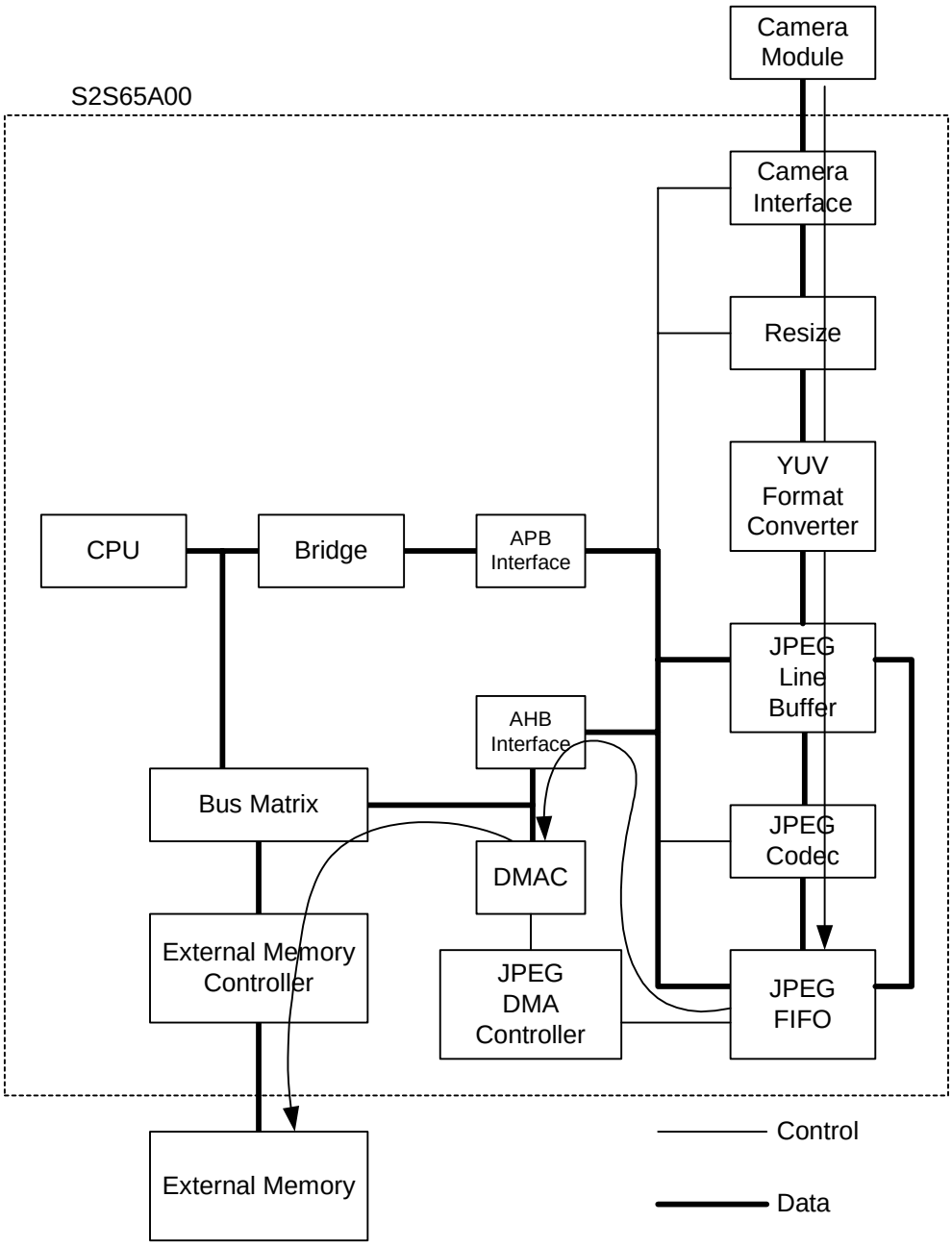


Fig.8.15 Data Flow of Camera Image JPEG Encoding

8.5.3.2 YUV Data Capture

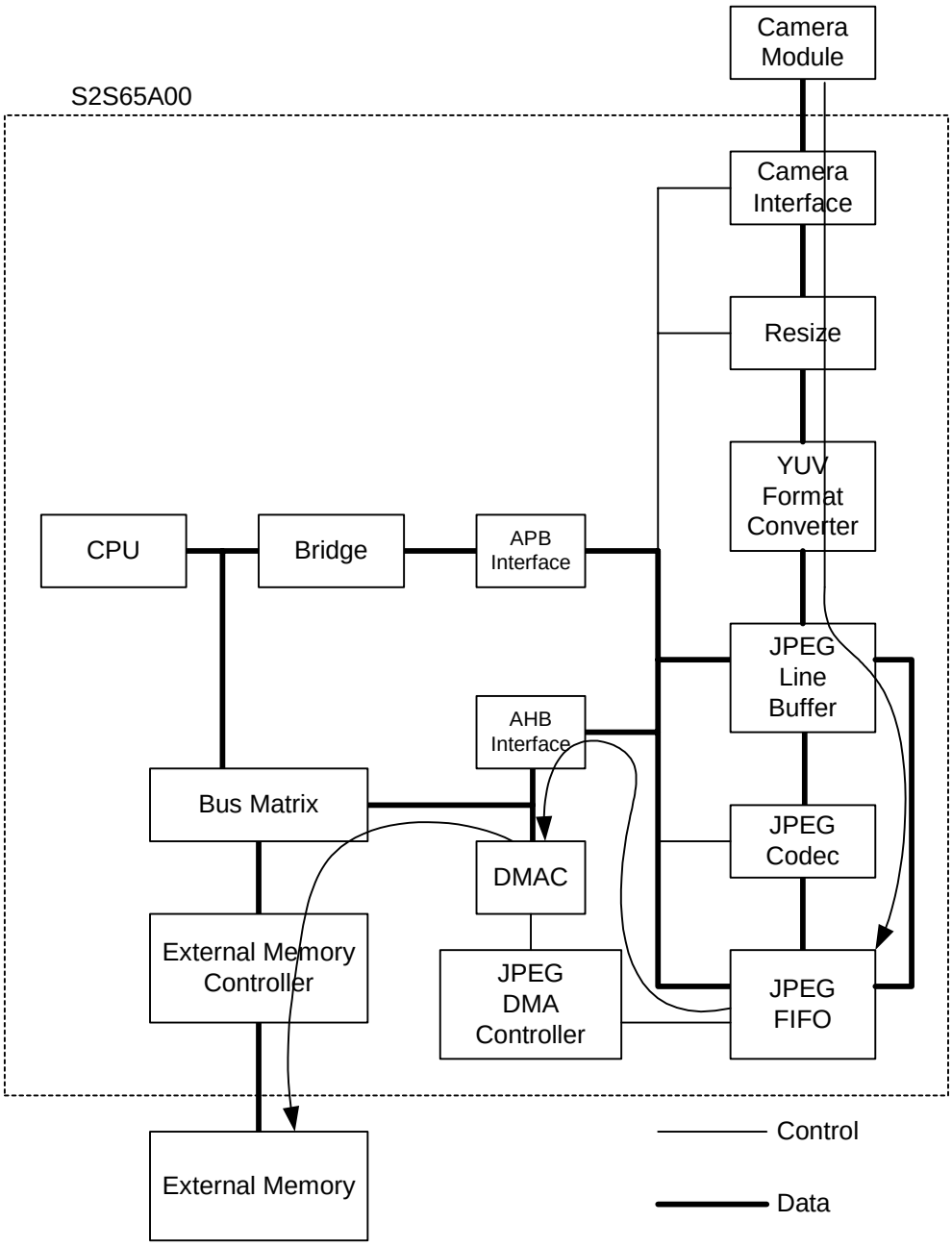


Fig.8.16 Data Flow of YUV Data Capture

8.5.4 JPEG Codec

The JPEG Codec circuit contained in the device almost complies with the baseline JPEG standard and satisfies the conformance tests that are included in Part 2 of JPEG standard (ISO/IEC 10918-2). The JPEG Codec circuit supports up to 640 (width) × 480 (height) pixels. It also supports VGA-size JPEG encoding up to VGA.

YUV 4:4:4 data that is sent from the resizer is converted into a YUV format specified by the JCODEC [0x00] bits [1:0] by the YUV format converter. For JPEG encoding of camera images and YUV capture, there are restrictions on the minimum resolution for the sizes of images processed.

Table 8.17 Minimum Resolution Restrictions

YUV Format	Minimum Resolution
4:4:4	1×1
4:2:2	2×1
4:2:0	2×2
4:1:1	4×1

There are also restrictions on the minimum size of images: operations of images smaller than that size is not guaranteed.

Table 8.18 Minimum Size

YUV Format	MCU Size (Horizontal x Vertical)
4:4:4	(8 × 8)
4:2:2	16 × 8
4:2:0	16 × 16
4:1:1	32 × 8

Two quantization tables can be used for encoding process. The Huffman tables can be used, two each for AC and DC components. For the JPEG encoding process, a marker of up to 36 bytes, including a marker identifier, can be inserted. The JPEG encoding process for camera images supports YUV 4:4:4, YUV 4:2:2, YUV 4:2:0, and YUV 4:1:1. For YUV 4:4:4, the amount of data is larger than that of a camera image. It is, therefore, not supported when encoding camera images without scaling. Encoding from YUV data stored in a RAM to JPEG data supports YUV 4:2:2 and YUV 4:2:0 only. The image data processing capability is less than one-thirtieth of a second for an image size of 640 × 480 pixels. This performance may not be achieved due to such factors as values of quantization and/or Huffman tables, and resolution of input images.

An approach for performing JPEG encoding successively in a consistent manner would be to alternatively activate and deactivate encoding of one frame. The controller is capable of performing JPEG encoding successively on a frame by frame basis. Whether or not this capability can be achieved depends upon the type of camera modules employed in your system or the performance of software. Specifically, the following inequality must be satisfied.

Time period during which a camera VREF is held inactive
> Interrupt response time
+ Total amount of time a process is interrupted by higher priority tasks
+ Time to process settings for encoding the next frame
+ Time required to output markers from JPEG Codec circuit

The JPEG Codec circuits outputs a marker at a rate of 36 microseconds at 50 MHz system clock if the marker fast output mode added to the S2S65A00 is used, whereas it outputs it at a rate of approximately 2 milliseconds if that mode is not used. To perform JPEG encoding successively on a VGA sized frame basis without using the marker fast output mode, it is very likely to generate an overflow unless there are alt least 15 lines in the period of time during which the VREF remains inactive.

8.5.4.1 Restrictions on JPEG Codec Registers

Do not modify any JPEG Codec register while it is in operation (between the start of the JPEG Codec and the end of the JPEG Codec). A response to an access during that period should cause a malfunction or should become invalid.

Accessing the JPEG Codec register with no clock supplied to the JPEG Codec circuit will be invalidated. The conditions that cause the JPEG Codec clock to stop are:

- The JPEG modules are disabled.
- The JPEG modules are enabled and the JPEG processing mode is neither 000 nor 100.

Registers that must be configured for YUV capture are designed so that they can be configured even if no clock is supplied to the JPEG Codec circuit.

If a read access is attempted on a register reserved for the JPEG Codec circuit or a write-only register, a read value as well as the operation of the JPEG Codec circuit are not guaranteed.

Reads from JPEG Codec Status Registers and JPEG Codec Marker Status Registers cause their internal state to change. Therefore, they must be read them only when needed. **Upon completion of JPEG encoding, read the JPEG Codec Marker Status Register before reading the JPEG Codec Status Register. This completes the operations of the JPEG Codec circuit. Subsequent operations should result in failure if the JPEG Codec Marker Status Register is not read.**

For both of the quantization and Huffman tables, a write to table can be omitted only when the same operation is performed in succession. The tables must be rewritten if settings are modified, even for encoding.

8.5.5 Functions Other Than JPEG Codec

8.5.5.1 JPEG FIFO

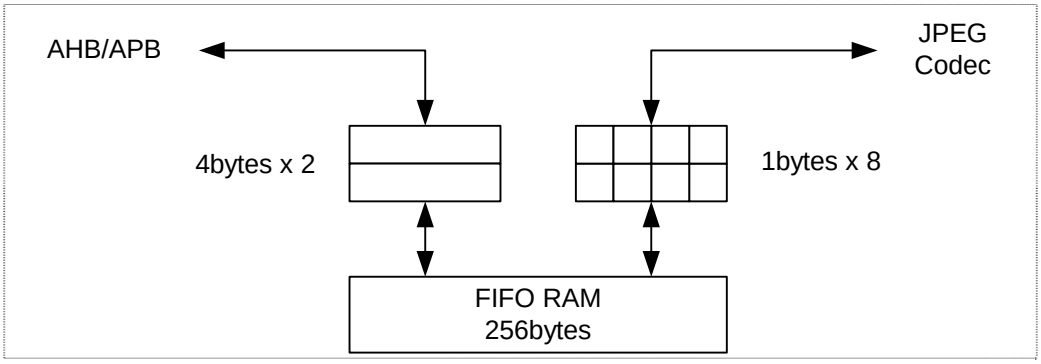


Fig.8.17 JPEG FIFO Overview

The JPEG FIFO is capable of holding 272 bytes of data using a 256-byte RAM and four 4-byte RAM buffers (two for each of read and write). To perform JPEG FIFO read/write operations by the CPU by checking statuses of the JPEG FIFO, it is recommended that the maximum value that can be read from and written into a 256-byte RAM be applied.

The JPEG FIFO Status Register can be used to check JPEG FIFO statuses. Further, most of the JPEG FIFO statuses can also be checked by the JPEG Interrupt Status Register. The JPEG FIFO Status Register indicates the status of the JPEG FIFO when it is read. The Empty Status of the JPEG Interrupt Register indicates that the FIFO has been empty at least once. The Full Status of the JPEG Interrupt Register indicates that the FIFO has been full at least once. The JPEG Interrupt Control Register generates an interrupt based on the above status information and retains it until an interrupt is cleared.

For JPEG encoding of camera images and of YUV data, there are two ways to retrieve data from the JPEG FIFO. One method is to let the CPU read the JPEG FIFO Read/Write Port Register. The other method is to use a JPEG DAM controller.

1. Low performance - An approach in which the JPEG FIFO Empty Flag is used to determine whether or not the FIFO is empty to let the CPU read the JPEG FIFO Read/Write Port Register. This is an impractical approach since the FIFO is not large enough, using interrupts that are triggered by FIFO level (full) or threshold level are inefficient. Since the CPU polls the JPEG FIFO Empty Status Flag to read data until the JPEG Codec End Interrupt Flag is set, this approach consumes large amounts of CPU. It is, therefore, not suitable for multitasking or real-time processing.

2. High performance - An approach in which a JPEG DMA controller is used to transfer data to memory areas. The JPEG DMA controller completes a data transfer upon reception of Frame End sent from the JPEG FIFO. The user can implement required transfer simply by setting the memory size circuit reserved as memory area with DAM setting to the maximum amount of data to be transferred. The user requires to set only a small number of registers for each frame and transfer from the FIFO causes virtually no CPU load.

8.5.5.2 JPEG Line Buffer

The JPEG line buffer contains a 30 KB RAM and enables data interchange between line-base and block interleaved methods. It can handle up to 640 pixels in width. When creating a JPEG file, image data whose size is represented in MCUs is required. The line buffer complements for deficit in the data. For example, to encode a 100 × 100 pixel image in YUV4:2:2, the line buffer compensates for the image to 112 × 104 pixels before sending data to the JPEG Codec circuit. The line buffer simply compensates for data whose MCUs do not contain much details and does not serve to compensate for, for example, a 60 × 60 pixel data sent from the resizer to a 112 × 104.

It is capable of detecting its own overflow state and generating an interrupt. In JPEG encoding, if the JPEG FIFO or JPEG Codec circuit is unable to continue processing, it waits for input data. If the JPEG Codec circuit

frequently keeps incoming inputs from the resizer waiting, the buffer overflows. Whenever real-time encoding cannot be performed due to JPEG FIFO reads or JPEG encoding workloads, line buffer overflow interrupts are detected.

The line buffer handles data containing the lines for width in MCUs as a bank to implement data interchange between line-base and block interleaved methods. It is capable of varying the number of banks that can be stored in the 30 KB RAM based on the widths of input images to efficiently use the RAM.

For YUV 4:2:0, 16 lines make up one bank. For YUV 4:2:2, 8 lines makes up one bank.

Table 8.19 Amount of RAM Data (Number of Banks) Based on Size of Input Image

Width of input image	Amount of data
≤32	Width of input image × 32banks
≤64	Width of input image × 16banks
≤128	Width of input image × 8banks
≤256	Width of input image × 4banks
>256	Width of input image × 2banks

8.5.5.3 YUV Format Converter

The YUV format converter converts YUV 4:4:4 image sent form the resizer into four types of YUV formats. As illustrated below, averaging is adopted for the YUV conversion. Fig.8.18 shows an example of YUV format conversion of U component. The V component has the same formula for the averaging. Y component is the same as the original image for the YUV format.

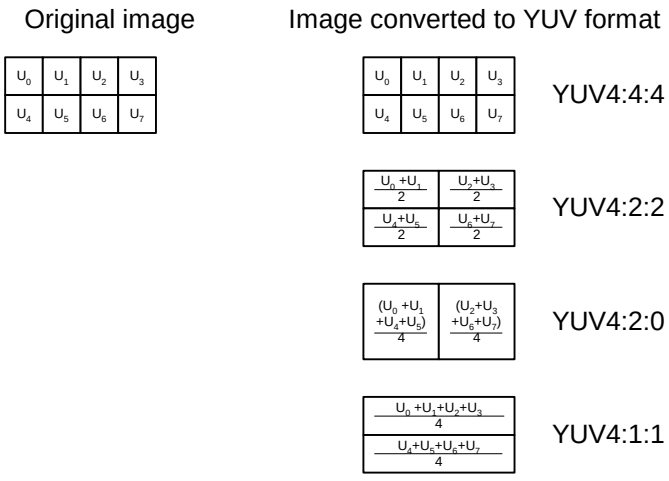


Fig.8.18 YUV Format Conversion

8.5.5.4 JPEG Module Interrupts

This section describes how to use interrupt flags.

1. JPEG Codec Interrupt Flag
   For JPEG-based camera image encoding, the JPEG Codec Interrupt Flag is used to access to a FIFO through a CPU. When this flag is turned on, the size of the JPEG file is determined. The amount of remaining data to be read last corresponds to the value obtained by subtracting the amount of data that has been read from the size of encoding result. This flag is not turned on for YUV Capture.

2. JPEG Line Buffer Overflow Interrupt Flag
   The JPEG Line Buffer Overflow Interrupt Flag is used for encoding JPEG camera images. Since the JPEG Codec circuit does not guarantee that JPEG encoding process always completes successfully, it is required to enable this interrupt when encoding JPEG camera images.

3. JPEG FIFO Empty Flag
   In JPEG encoding, there is a way to use the JPEG FIFO Empty Flag to determine whether or not the FIFO is empty. It has strong implications of using it for just in case. It is, therefore, not necessary to perform encoding process by checking the state of this flag.

4. JPEG FIFO Full Flag
   The JPEG FIFO Full Flag may be possibly used to let CPU read the JPEG FIFO for JPEG-based camera image encoding, in which case, the number of bytes corresponding to the FIFO size is read from the FIFO when it is full. While the FIFO is full, the JPEG Codec circuit aborts encoding process, data accumulating in the line buffer. The user will actually adopt an approach to read the value from the FIFO Effective Size Register by using the JPEG FIFO Threshold Status Register to determine whether or not the value is greater than or equal to 1/2 or 1/4 of the FIFO size.

5. JPEG FIFO Threshold Trigger Flag
   Because the size of FIFO is small, it is virtually impractical to use FIFO access that is driven by threshold-triggered interrupts should increase the overhead involved with the interrupt response time. For that reason, this flag does not server any special purpose.

6. Encode Size Limit Over Flag
   The Encode Limit Over Flag is used for encoding JPEG camera images. When the JPEG DMA controller is used, interrupts that are triggered by the maximum number of DMA transfers are used to prevent memory corruption. It is, therefore, the flag is not used to directly limit the maximum size of the JPEG file. Instead, it is used to detect whether or not the size of the JPEG file is close to reaching the limit. This allow, for example, the user to use it to change the value of a quantization table to a larger value.

8.5.5.5 JPEG 180° Rotation Encode

In JPEG rotation by 180°, rotation is not performed on one frame but performed in increments of lines that corresponds to MCU width of frame. It is, therefore, necessary to sort the data that is rotated by lines via software. For instance, the data in lines is written at the beginning of image data area in the JPEG file and, when rotation is performed on one frame, it is written at the end of the image area. It is, therefore, necessary to move the location to the end of the image area using software.

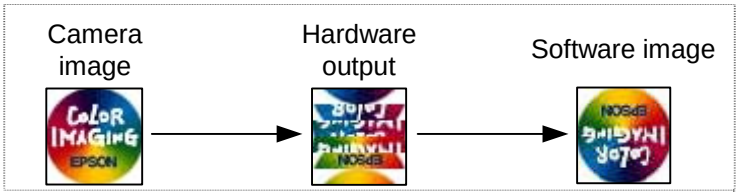


Fig.8.19 JPEG 180° Rotation Encode

Since it is difficult for software to recognize the end of line simply by detecting encoded image data, insert a RST marker at the end of each line to make the software recognize it. Thus, it is mandatory to insert RST markers. For the interval for an RST marker to be inserted, divide the width of an image by horizontal MCUs and then round the result to 0 places. For example, the interval for RST marker to be inserted when 100 pixel wide image is rotated by 180° and encoded in YUV 4:2:2 is 100/16 = 6.25 → 7.

8.5.5.6 Software Reset

In the JPEG Codec circuit, a single encoding sequence is performed on a frame-by-frame basis. Successive encoding is achieved by repeating the single encoding sequence. It is recommended that the JPEG Codec circuit be reset through software before it performs an operation. This initialization ensures stable operations. Since software reset only resets functional blocks, the values set in the registers remain intact.

In addition to the software reset on the JPEG Codec circuit, JPEG module software reset is provided to reset the YUV format converter, JPEG line buffer and JPEG FIFO components. Note that software reset recommended for camera images when performing JPEG encode process differs. This is attributable to time lag caused by processing time for each component.

To perform JPEG encoding successively, we recommend that the JPEG module be not reset through software. This is because the end of processing of current frame operates in parallel with the start of processing of the next frame and no timing at which software reset is performed is most likely to occur.

On the other hand, when the user provides an interval for one or more frames after encoding one frame, there is no possibility of overlapping of encoding process between the current and next frames. We, therefore, recommend that software reset be performed before initiating the JPEG encoding.

It is also necessary to reset the JPEG modules through software after encoding images grouped in MCUs. This means that successive JPEG encoding must be performed on images grouped in MCUs.

8.5.5.7 Marker Fast Output Mode

The types of JPEG data are identified by markers called JPEG markers. In the JPEG Codec circuit, most of the markers are located in front of the compression data part. With earlier versions of the products, these markers are output at a clock rate of approximately 2 ms at 50 MHz system clock. For S2S65A00, new Marker Fast Output Mode is added and it is possible to complete marker outputs at a clock rate of approximately 36 μs at 50 MHz system clock by using this mode together with the JPEG Fast Encoding Mode.

Since processing is accelerated by making the Huffman tables fixed values, the Marker Fast Output Mode will be disabled if the JPEG Fast Encoding Mode is disabled regardless of the setting of the relevant register.

8.5.6 Example of Sequence

8.5.6.1 JPEG Encoding of Camera Image (Single Frame)

This section describes a sequence for encoding a single camera image. The DMA is used to read data from the FIFO.

1. Set up the camera interface. For details, see Explanation of Operation on the camera interface.
2. Enable the JPEG module (JCTL[0x00], Bit 0 = 1) and set the JPEG Operation Mode Setting Register (bits [3:1]) to “000”.
3. Reset the JPEG module through software (JCTL[0x00], bit 7 = 1).
4. Initialize the JPEG Codec Registers in ascending order. There is no dependency on the order except for the Command Register.
   (a) Reset the JPEG circuit through software (JCODEC[0x04], bit 7 = 1).
   (b) Set the JPEG encoding mode (JCODEC[0x00], bit 2 = 0).
   (c) Enable appropriate Insert Marker (JCODEC[0x00], bit 3 = 1)) to insert a desired marker.
   (d) Specify the quantization table number (JCODEC[0x0C]) and Huffman table number (JCODEC[0x10]).
   (e) Set an interval for inserting RST markers, if they are to be inserted (JCODEC[0x14], [0x18]).
   (f) Enter the size of image (JCODEC[0x1C], [0x20], [0x24], [0x28]).
   (g) Enter the marker to be inserted (JCODEC[0x04 - 0xCC]).
   (h) Set the quantization tables in the order of JCODEC[0x400 - 0x4FC] and [0x500 - 0x5FC].

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

   (i) Set the Huffman tables, for example, using data given in the ISO/IEC 10918 Annex K.

   Set A to the DC Huffman Table No.1 Register 0 (JCODEC[0x800 - 0x83C]).
   Set B to the DC Huffman Table No.0 Register 1 (JCODEC[0x840 - 0x86C]).
   Set E to the DC Huffman Table No.0 Register 0 (JCODEC[0x880 - 0x8BC]).
   Set F to the DC Huffman Table No.0 Register 1 (JCODEC[0x8C0 - 0xB44]).
   Set C to the DC Huffman Table No.1 Register 0 (JCODEC[0xC00 - 0xC3C]).
   Set D to the DC Huffman Table No.1 Register 1 (JCODEC[0xC40 - 0xC6C]).
   Set G to the DC Huffman Table No.1 Register 0 (JCODEC[0xC80 - 0xCBC]).
   Set H to the DC Huffman Table No.1 Register 1 (JCODEC[0xCC0 - 0xF44]).

A:	00h, 01h, 05h, ..., 00h, 00h	16 byte
B:	00h, 01h, 02h, ..., 0Ah, 0Bh	12 byte
C:	00h, 03h, 01h, ..., 00h, 00h	16 byte
D:	00h, 01h, 02h, ..., 0Ah, 0Bh	12 byte
E:	00h, 02h, 01h, 03h, ...1h, 7Dh	16 byte
F:	01h, 02h, 03h, ..., F9h, FAh	162 byte
G:	00h, 02h, 01h, 02h, ..., 02h, 77h	16 byte
H:	00h, 01h, 02h, ..., F9h, FAh	162 byte

5. Initialize the JPEG module.
   (a) Specify the size of the JPEG FIFO. Since the FIFO is a dedicated RAM, enter 0x3F. (JFIFO[0x48] = 0x3F)
   (b) Specify the limit on the size for encoding in bytes (JFIFO[0x60], [0x64]).
   (c) Clear the JPEG FIFO (JCODEC[0x04], bit 2 = 1).

6. Initialize the resizer (RSZ[0xC0], bit 7 = 1). Set it so that the size of resized image is the same value as the one specified in step 4 (f). (RSZ[0xC8], [0xCC], [0xD0], [0xD4])
7. Specify interrupts. Clear the interrupt by writing 0x000FFF in the JPEG Status Flag Register (JCTL[0x04]) and then enable the Line Buffer Overflow Interrupt and Encode Size Limit Violation Interrupt (JCTL[0x0C], bit 2 = bit 11 = 1). For the JPEG Status Flag Register, reserved bits can be written with a "1". Enable the Interrupt Enable for the JPEG module in the interrupt controller (INT[0x008]).
8. Configure the DMA controller (JDMA[0x00] to [0x40]).
9. Starts the JPEG encoding operation.
   (a) Start the JPEG circuit (JCODEC[0x04], bit 0 = 1).
   (b) Start the JPEG module (JCODEC[0x14], bit 0 = 1).
   To output the JPEG marker after starting the JPEG Codec circuit, approximately 2 ms are required at 50 MHz system clock. When the JPEG module is started before 2 ms have elapsed, the capture process actually starts after 2 ms have elapsed.
10. Wait for Frame Capture End Interrupt of the JPEG controller. If a line buffer overflow interrupt occurs during the process or the JPEG encoding process does not complete within a few minutes, perform the termination processing.

8.5.6.2 Termination

The user can use the same termination processing for all processing modes. A failed operation can be initialized by performing termination processing. Write access to the JPEG Codec circuit must be followed by a dummy read of another register, which is not included in the sequence described below.

1. Write 0x0000 in the Global Control Register (RSZ[0x60]) for the resizer in case a wrong value is written in a reserved register.
2. Enable the resizer (RSZ[0xC0], bit 0 = 1) and enable the software reset (RSZ[0xC0], bit 7 = 1).
3. Enable the JPEG module circuit (JCTL[0x00], Bit 0 = 1) and write 000 in the JPEG Operation Mode Setting Register (JCTL[0x00], bits 3-1).
4. Reset the JPEG Codec circuit through software (JCODEC[0x04], bit 7 = 1).
5. Read (dummy read) the RST Marker Operation Status Register (JCODEC[0x3C]) in the JPEG Codec circuit.
6. Read (dummy read) the JPEG Operation Status Register (JCODEC[0x08]) in the JPEG Codec circuit.
7. Write 0x00 in the Operation Mode Setting Register (JCODEC[0x00]) in the JPEG Codec circuit.
8. Reset the JPEG module circuit through software (JCTL[0x00], bit 7 = 1).
9. Write 0x0000 in the JPEG Line Buffer Interrupt Control Register (JLB[0x8C]) and disable all line buffer interrupts.
10. Write 0xFFFF in the JPEG Line Buffer Status Flag Register (JLB[0x80]).
11. Write 0x0000 in the JPEG Interrupt Control Register (JCTL[0x0C]) and disable all JPEG interrupts.
12. Write 0xFFFF in the JPEG Status Flag Register (JCTL[0x04]).
13. Reset the JPEG DMA circuit through software (JDMA[0x20], bit 15 = 1).
14. Disable the DMA Enable and JPEG Interrupt Enable in the JPEG DMA circuit (JDMA[0x0C], bit 0 = 1, JDMA[0x0C], bit 21 = 0).
15. Clear the Interrupt Flag in the JPEG DMA circuit (JDMA[0x0C], bit 1 = 0).
16. Disable the JPEG Controller Interrupt and JPEG DAM Interrupt in the interrupt controller.
17. Disable the JPEG module (JCTL[0x00], bit 0 = 0).
18. Disable the resizer (RSZ[0xC0], bit 0 = 0).

Note: The JPEG module must be disabled before disabling the resizer. If the JPEG module is disabled last, clock supply to the JPEG Coded circuit may not stop.

9. JPEG_DMAC (JDMA)

9.1 Description

JPEG_DMAC is a DMA controller dedicated to the processing of image data sent from the camera interface.

JPEG_DMAC uses a function of DMAC2 to transfer the necessary register data to DMAC2 which performs the DMA transfer. Request/acknowledge to DMAC2 is performed by FIFO built in the JPEG controller.

9.2 Block Diagram

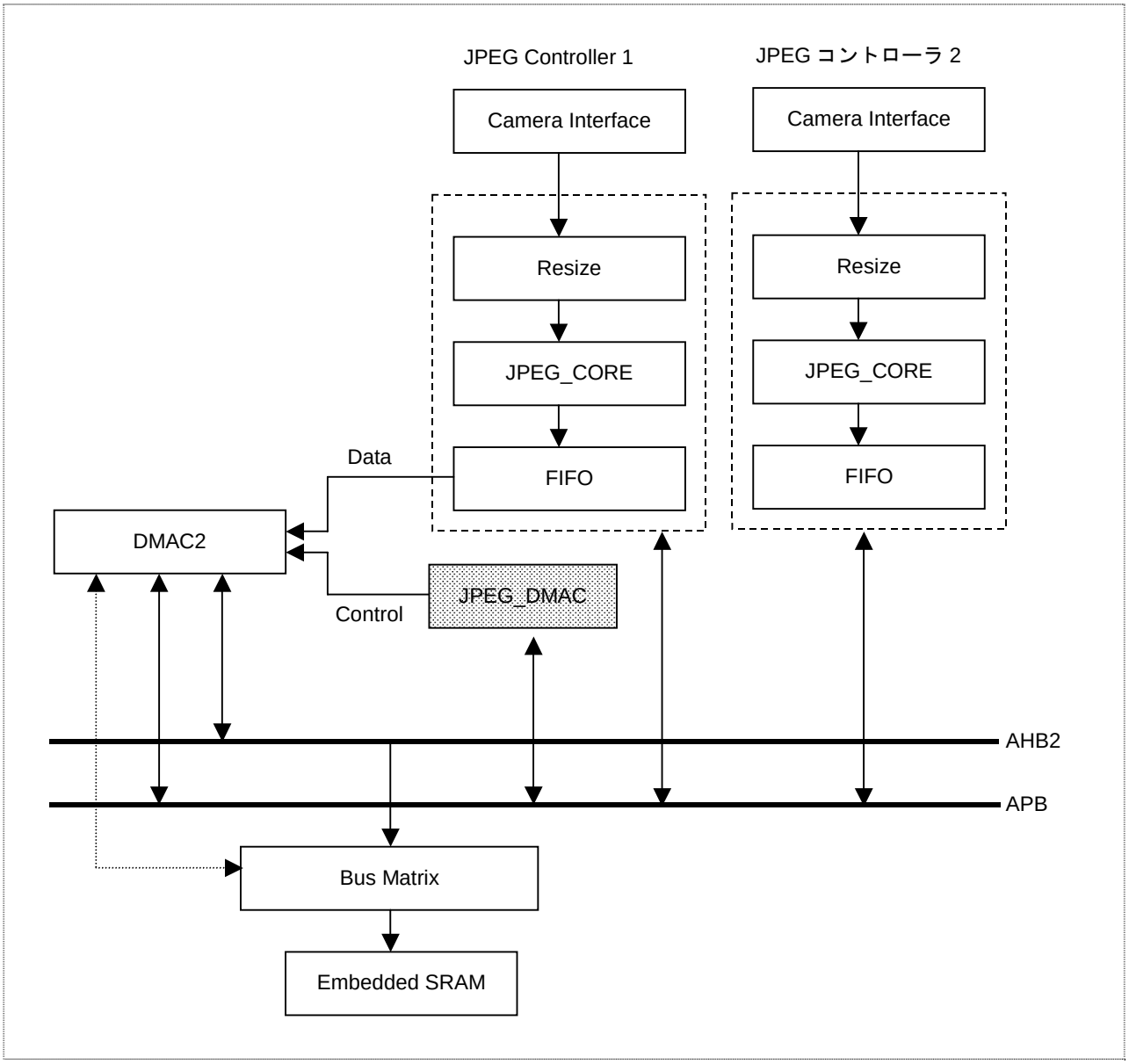


Fig.9.1 Block diagram of the relation between JPEG_DMAC and JPEG controllers, DMAC2, etc.

9.3 External Pins

The IC has no external pins for JPEG_DMAC.

9.4 Registers

9.4.1 List of Registers

The base address of the registers of JPEG_DMAC is 0xFFFE_C000.

Table 9.1 List of registers (Base address: 0xFFFE_C000)

Address offset	Register name	Register name's abbreviation	Default Value *	R/W	Data access size
0x00	DMA channel 0 JPEG source address register	JSAR0	0xXXXX_XXXX	R/W	32
0x04	DMA channel 0 JPEG destination address register	JDAR0	0xXXXX_XXXX	R/W	32
0x08	DMA channel 0 JPEG transfer count register	JTCR0	0x0000_0000	R/W	32
0x0C	DMA channel 0 JPEG control register	JCTL0	0x0000_0000	R/W	32
0x10	DMA channel 0 JPEG block count register	JBCR0	0x00XX_XXXX	R/W	32
0x14	DMA channel 0 JPEG destination offset address register	JOFR0	0x0000_0000	R/W	32
0x18	DMA channel 0 JPEG block-end count register	JBER0	0x00XX_XXXX	R/W	32
0x1C	—	—	—	—	—
0x20	DMA channel 1 JPEG source address register	JSAR1	0xXXXX_XXXX	R/W	32
0x24	DMA channel 1 JPEG destination address register	JDAR1	0xXXXX_XXXX	R/W	32
0x28	DMA channel 1 JPEG transfer count register	JTCR1	0x0000_0000	R/W	32
0x2C	DMA channel 1 JPEG control register	JCTL1	0x0000_0000	R/W	32
0x30	DMA channel 1 JPEG block count register	JBCR1	0x00XX_XXXX	R/W	32
0x34	DMA channel 1 JPEG destination offset address register	JOFR1	0x0000_0000	R/W	32
0x38	DMA channel 1 JPEG block-end count register	JBER1	0x00XX_XXXX	R/W	32
0x3C	—	—	—	—	—
0x40	DMA channel JPEG FIFO data selection mode register	JFSM	0x0000_0000	R/W	32
0x48	DMA channel JPEG extended register	JHID	0x0000_0000	R/W	32

* Note: X: Undecided (h)
Supplemental remark: It is recommended to use DMA channel 0 with JPEG controller 1 and DMA channel 1 with JPEG controller 2.

9.4.2 Detailed Description of Registers

DMA channel 0/1 JPEG source address register (JSAR0/1)

JDMA[0x00], [0x20] Default value = 0xXXXX_XXXX Read/Write

DMA channel 0/1 JPEG source address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA channel 0/1 JPEG source address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **DMA channel 0/1 JPEG source address [31:0]**

Set the source addresses used for the JPEG_DMA transfer with the software. The settings are not updated by hardware.
Set 0xE000_004C to use JPEG1.
Set 0xD000_004C to use JPEG2.

DMA channel 0/1 JPEG destination address register (JDAR0/1)

JDMA[0x04], [0x24] Default value = 0xXXXX_XXXX Read/Write

DMA channel 0/1 JPEG destination address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA channel 0/1 JPEG destination address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **DMA channel 0/1 JPEG destination address [31:0]**

Set the destination addresses used for the JPEG_DMA transfer with the software. Each time the DMA transfer of one block is completed, the offset value set for the JPEG destination offset address register of the DMA channel is added automatically to the current value set.

DMA channel 0/1 JPEG transfer count register (JTCR0/1)

JDMA[0x08], [0x28] Default value = 0x0000_0000 Read/Write

n/a								DMA channel 0/1 JPEG transfer count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA channel 0/1 JPEG transfer count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [23:0]: **DMA channel 0/1 JPEG transfer count [23:0]**

Set the count of the JPEG_DMA transfer with the software. The settings are not updated by hardware
Bits [31:24] get "0" if this register is read.

DMA channel 0/1 JPEG control register (JCTL0/1)

JDMA[0x0C], [0x2C] Default value = 0x0000_0000 Read/Write

n/a								RSV	JS	JIE	JCS1	JCS0	RSV	AM	AL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV		TM	TS		IE	JTE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 23: **RSV Reserved (0)**

Bit 22: **JS JPEG_DMA transfer start**

0: This bit is cleared automatically to "0" upon completion of the transfer of one block or upon assertion of FIFO_END.
1: "1" is set upon start of the JPEG_DMA transfer.

Bit 21: **JIE JPEG interrupt enable**

If this bit is "1", interrupt is asserted upon setting of JTE (Bit 1).

0: Interrupt disabled
1: Interrupt enabled

Bit [20:19]: **JCS[1:0] DMA channel selection**

Select a DMA channel of the DMAC2 used.

00: DMA channel 0
01: DMA channel 1
10: DMA channel 2
11: DMA channel 3

Bit 18: **RSV Reserved (0)**

Set "0" to this bit.

Bit 17: **AM Acknowledge mode**

Select the output timing for the DACK signal.

0: Active while in the DMA read cycle
1: Active while in the DMA write cycle

Bit 16: **AL Acknowledge level**

Select the output polarity for the DACK signal.

0: LOW active
1: HIGH active

Bits [15:14]: **DAM Destination Address Mode [1:0]**

Select an update mode entered upon completion of one count of transfer for the destination address register.

00: Destination address fixed (not updated)
01: Destination address incremented corresponding to the size of data transferred.
(+1 for 8 bits; +2 for 16 bits, +4 for 32 bits)
10: Destination address decremented corresponding to the size of data transferred.
(-1 for 8 bits; -2 for 16 bits, -4 for 32 bits)
11: Reserved

Bits [13:12]: **SAM Source Address Mode [1:0]**

Select an update mode entered upon completion of one count of transfer for the source address register.

00: Source address fixed (not updated)
01: Source address incremented corresponding to the size of data transferred.
(+1 for 8 bits; +2 for 16 bits, +4 for 32 bits)
10: Source address decremented corresponding to the size of data transferred.
(-1 for 8 bits; -2 for 16 bits, -4 for 32 bits)
11: Reserved

Bits [11:8]: **RS Resource selection [3:0]**

Select the cause which starts up the DMA transfer.

0010: JPEG1
0011: JPEG2
Other: Reserved

Bits [7:6]: **RSV Reserved (0)**

Bit 5: **TM Transmission Mode**

Select a transmission mode for the DMA transfer.

0: Single transfer
1: Demand transfer

Bits [4:3]: **TS Size of data transferred**

Select the size of data sent by one count of transfer.

00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2: **IE Interrupt enable**

0: No interrupt is caused upon completion of one count of block transfer.
1: Interrupt is caused upon completion of one count of block transfer.

Bit 1: **JTE JPEG_DMA transfer completed**

0 (when reading): Transferring or standing by
1 (when writing): JPEG_DMA transfer completed
0 (when writing): Clears this bit.
1 (when writing): Invalid

This bit is set if the DMA channel's JPEG transfer block count register gets a value "0". Once set, it holds "1" until it is cleared by "0" written in. The DMA transfer is disabled for this channel until this bit is cleared. This bit functions also as an interrupt flag.

Bit 0: **DE DMA enable**

This bit can enable the JPEG_DMA transfer.

0: JPEG_DMA transfer disabled
1: JPEG_DMA transfer enabled

DMA channel 0/1 JPEG block count register (JBCR0/1)

JDMA[0x10], [0x30] Default value = 0x00XX_XXXX Read/Write

n/a								DMA channel 0/1 JPEG block count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA channel 0/1 JPEG block count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [23:0]: **DMA channel 0/1 JPEG block count [23:0]**

Set the count of block transfer for the JPEG_DMA transfer with the software. Decremented by "1" automatically upon completion of transfer of one block in the JPEG_DMA transfer.
Bits [31:24] get "0" if this register is read.

DMA channel 0/1 JPEG destination offset address register (JOFR0/1)

JDMA[0x14], [0x34] Default value = 0x0000_0000 Read/Write

n/a								DMA channel 0/1 JPEG destination offset address [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA channel 0/1 JPEG destination offset address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [23:0]: **DMA channel 0/1 JPEG destination offset address [23:0]**

Set the offset value for the JPEG_DMA transfer with the software. The settings are not updated by hardware
Bits [31:24] get "0" if this register is read.

DMA channel 0/1 JPEG block-end count register (JBER0/1)

JDMA[0x18], [0x38] Default value = 0x00XX_XXXX Read/Write

n/a								DMA channel JPEG block-end count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA channel JPEG block-end count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [23:0]: **DMA channel 0/1 JPEG transfer block-end count [23:0]**

Set the count of block transfer for the JPEG_DMA transfer with the software. The register value is incremented by "1" automatically upon completion of transfer of one block in the JPEG_DMA transfer. Normally, it shows the current count of block transfer.

Bits [31:24] get "0" if this register is read.

DMA channel JPEG FIFO data selection mode register (JFSM)

JDMA[0x40] Default value = 0x0000_0000 Read/Write

n/a		RSV						n/a		RSV		JF1B	RSV		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a		RSV						n/a		RSV		JF0B	RSV		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [29:24]: **RSV Reserved (0)**
Bits [21:20]: **RSV Reserved (0)**
Bits [18:16]: **RSV Reserved (0)**
Bits [13:8]: **RSV Reserved (0)**
Bits [5:4]: **RSV Reserved (0)**

Bit 19, 3: **JF[1:0]B FIFO mode**

Makes the settings for the data output bus of JPEG FIFO[1:0].

0: Responds to access from the APB bus side.
1: Responds to access from the AHB bus side.

JF1B responds to JPEG2, and JF0B to JPEG1.

Bits [2:0]: **RSV Reserved (0)**

DMA channel JPEG extended register (JHID)

JDMA[0x48] Default value = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SW	n/a														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 15: **SW Software reset**

Causes the software reset of all the registers in JDMA. The software reset is performed by "1" written in this bit. Upon completion of the software reset, this bit returns to "0" automatically.

10. DMA Controller 2 (DMAC2)

10.1 Description

DMAC2 is a DMA controller which performs the DMA transfer based on the control information sent from JPEG_DMAC or set by software.

10.2 Block Diagram

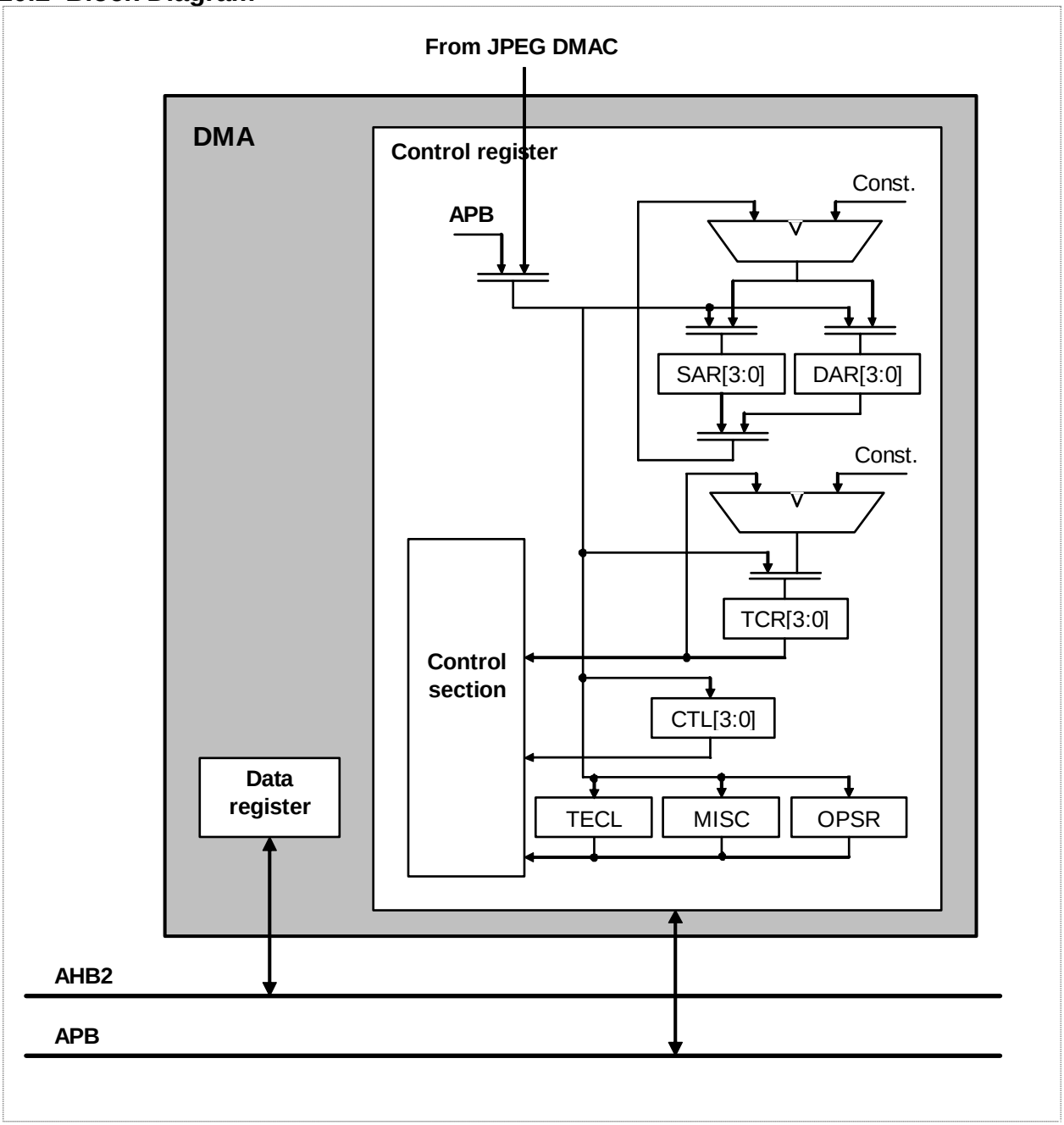


Fig.10.1 Block diagram of DMA controller 2

10.3 External Pins

The IC has no external pins for DMA controller 2.

10.4 Registers

10.4.1 List of Registers

The base address of the registers for DMAC2 is 0xFFFF_9000.

Table 10.1 List of registers (Base address: 0xFFFF_9000)

Address offset	Register name	Register name's abbreviation	Default Value	R/W	Data access size
0x00	DMA channel 0 Source address register	SAR0	0xXXXX_XXXX	R/W	32
0x04	DMA channel 0 Destination address register	DAR0	0xXXXX_XXXX	R/W	32
0x08	DMA channel 0 Transfer count register	TCR0	0x00XX_XXXX	R/W	32
0x0C	DMA channel 0 Control register	CTL0	0x0000_0000	R/W	32
0x10	DMA channel 1 Source address register	SAR1	0xXXXX_XXXX	R/W	32
0x14	DMA channel 1 Destination address register	DAR1	0xXXXX_XXXX	R/W	32
0x18	DMA channel 1 Transfer count register	TCR1	0x00XX_XXXX	R/W	32
0x1C	DMA channel 1 Control register	CTL1	0x0000_0000	R/W	32
0x20	DMA channel 2 Source address register	SAR2	0xXXXX_XXXX	R/W	32
0x24	DMA channel 2 Destination address register	DAR2	0xXXXX_XXXX	R/W	32
0x28	DMA channel 2 Transfer count register	TCR2	0x00XX_XXXX	R/W	32
0x2C	DMA channel 2 Control register	CTL2	0x0000_0000	R/W	32
0x30	DMA channel 3 Source address register	SAR3	0xXXXX_XXXX	R/W	32
0x34	DMA channel 3 Destination address register	DAR3	0xXXXX_XXXX	R/W	32
0x38	DMA channel 3 Transfer count register	TCR3	0x00XX_XXXX	R/W	32
0x3C	DMA channel 3 Control register	CTL3	0x0000_0000	R/W	32
0x40 -0x5C	Reserved	—	—	—	—
0x60	DMA channel Operating selection register	OPSR	0x0000_0000	R/W	32
0x64	DMA channel MISC register	MISC	0x0000_0000	R/W	32
0x70	DMA channel Transfer completion control register	TECL	0x0000_0000	R/W	32

10.4.2 Detailed Description of Registers

DMA channel [3:0] source address register (SAR[3:0])

DMAC2[0x00], [0x10], [0x20], [0x30] Default value = 0xXXXX_XXXX Read/Write

DMA channel [3:0] source address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA channel [3:0] source address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **DMA channel [3:0] source address [31:0]**

Set the source address for the DMA transfer through channel [3:0] with the software. The source address set must be a boundary address compatible with the size of data transferred. In the case of 32-bit transfer, for example, "00b" must be set to bits [1:0]. Once the DMA transfer is started, the source address is updated automatically to another one for the next transfer each time one count of transfer completes according to the size of data transferred (TS: channel [3:0] control register bits [4:3]) and the source address mode (SAM: channel [3:0] control register bits [13:12]).

DMA channel [3:0] destination address register (DAR[3:0])

DMAC2[0x04], [0x14], [0x24], [0x34] Default value = 0xXXXX_XXXX Read/Write

DMA channel [3:0] destination address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA channel [3:0] destination address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **DMA channel [3:0] destination address [31:0]**

Set the destination address for the DMA transfer through channel [3:0] with the software. The destination address set must be a boundary address compatible with the size of data transferred. In the case of 32-bit transfer, for example, "00b" must be set to bits [1:0]. Once the DMA transfer is started, the destination address is updated automatically to another one for the next transfer each time one count of transfer completes according to the size of data transferred (TS: channel [3:0] control register bits [4:3]) and the destination address mode (DAM: channel [3:0] control register bits [15:14]).

DMA channel [3:0] transfer count register (TCR[3:0])

DMAC2[0x08], [0x18], [0x28], [0x38] Default value = 0x00XX_XXXX Read/Write

n/a								DMA channel [3:0] transfer count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA channel [3:0] transfer count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [23:0]: **DMA channel [3:0] transfer count [23:0]**

Set the DMA transfer count with the software. Once the DMA transfer starts, the setting is decremented each time one count of transfer completes. If "0" is set, the transfer count is 2^{24}=16777216. The counter value counted-down to "0" causes the DMA interrupt.

Bits [31:24] get "0" if this register is read.

DMA channel [3:0] control register (CTL[3:0])

DMAC2[0x0C], [0x1C], [0x2C], [0x3C] Default value = 0x0000_0000 Read/Write

n/a								RSV					IB4	AM	AL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV	RIM	TM	TS		IE	TE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [23:19]: **RSV Reserved (0)**

Bit 18: **IB4 Increment burst 4**

Bit value "1" causes the increment burst 4 transfer to be performed. The size of the data transferred by the increment burst 4 transfer, though, must be a multiple of four data-sizes. Otherwise, any fraction of the size of the data transferred must be transferred by separate DMA transfer by setting this bit to "0".
This function is used only for block transfer from a memory to another.

Bit 17: **AM Acknowledge mode**

Select the output timing for the DACK signal.

0: Active while in the DMA read cycle
1: Active while in the DMA write cycle

Bit 16: **AL Acknowledge level**

Select the output polarity for the DACK signal.

0: LOW active
1: HIGH active

Bits [15:14]: **DAM Destination Address Mode [1:0]**

Select an update mode entered upon completion of one count of transfer for the destination address register.

00: Destination address fixed (not updated)
01: Destination address incremented corresponding to the size of data transferred.
(+1 for 8 bits; +2 for 16 bits, +4 for 32 bits)
10: Destination address decremented corresponding to the size of data transferred.
(-1 for 8 bits; -2 for 16 bits, -4 for 32 bits)
11: Reserved

Bits [13:12]: **SAM Source Address Mode [1:0]**

Select an update mode entered upon completion of one count of transfer for the source address register.

00: Source address fixed (not updated)
01: Source address incremented corresponding to the size of data transferred.
(+1 for 8 bits; +2 for 16 bits, +4 for 32 bits)
10: Source address decremented corresponding to the size of data transferred.
(-1 for 8 bits; -2 for 16 bits, -4 for 32 bits)
11: Reserved

Bits [11:8]: **RS Resource selection [3:0]**

Select the cause which starts up the DMA transfer.

1111: **SW-Request** Software request
Setting bits [11:8] to "1111" starts up the DMA transfer of the software.
Only the addresses mapped in 4.2.2 Memory Map (AHB2) can be set.
Other: Reserved

Bit 7: **RSV Reserved (0)**

Bit 6: **RIM Request input mode**

Select an input mode for the DMA request signal.

0: LOW active (level trigger)
1: Fall edge (edge trigger)

Bit 5: **TM Transfer mode**

Select a transfer mode for the DMA transfer.

0: Single transfer
1: Demand transfer

Bits [4:3]: **TS Transfer size [1:0]**

Select the size of data transferred by one count of transfer.

00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2: **IE Interrupt enable**

Enables/disables the interrupt into DMA channel [3:0] upon completion of transfer.

0: Interrupt disabled
1: Interrupt enabled

Bit 1: **TE DMA transfer completed**

0 (when reading): Transferring or standing by
1 (when reading): JPEG_DMA transfer completed
0 (when writing): Clears this bit.
1 (when writing): Disabled

This bit is set if the transfer block count register of DMA channel [3:0] gets a value “0”. Once set, it holds “1” until it is cleared by “0” written in. The DMA transfer is disabled for this channel until this bit is cleared. This bit functions also as an interrupt flag.

Bit 0: **DE DMA enable**

This bit can enable the DMA transfer through channel [3:0].

0: DMA transfer disabled
1: DMA transfer enabled

DMA channel operating selection register (OPSR)

DMAC2[0x60] Default value = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a						DPE	DPM	n/a							DGE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 9: **DPE DMA priority change enable**

0: Priority not changed

1: Priority changed

Bit value "1" changes DPM of bit 8 under the following conditions:

Single transfer: Changes DPM at each count of transfer.

Demand transfer: Changes DPM if transfer is suspended because of negated request or if transfer counter gets "0".

Bit 8: **DPM DMA priority mode**

0: Priority is CH0>CH1>CH2>CH3.

1: Priority is CH1>CH0>CH2>CH3.

Bit 0: **DGE DMA global enable**

Enables/disables all the DMA channels.

0: Disables

1: Enables

DMA channel MISC register (MISC)

DMAC2[0x64] Default value = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SR	n/a											DPL			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 15: **SR Software reset**

"1" written in this bit initializes the registers in DMAC2.

All the registers in DMAC2 are initialized, so make settings again for registers as required.

Bits [3:0]: **DPL DMA polarity selection [1:0]**

DPL3 Select the polarity of DMA channel 3.

0: Positive

1: Negative

DPL2 Select the polarity of DMA channel 2.

0: Positive

1: Negative

DPL1 Select the polarity of DMA channel 1.

0: Positive

1: Negative

DPL0 Select the polarity of DMA channel 0.

0: Positive

1: Negative

DMA channel transfer completion control register (TECL)

DMAC2[0x70] Default value = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a		STTE	ENTE	n/a											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 13: **STTE TE set upon reception of completion of transfer**

"1" written in this bit sets TE upon receiving the transfer completion signal from a request-issuing cause. This bit becomes valid when bit 12 is set to "1".

Bits 12: **ENTE Transfer completion reception enable**

"1" written in this bit enables reception of the transfer completion signal from a request-issuing cause, without setting TE; Write "1" to bit 13 to make TE set.

11. USB HS-Device

11.1 Description

This USB HS-Device Macro is made by extracting the USB HS-Device Controller section mounted on the USB2.0-compatible SEIKO EPSON High Speed USB-Host/Device Combination Controller (S1R72V05) and rebuilding it as a macro for the use on a SoC (System On Chip).

See "Appendix2 USB Device Controller" for more details of the HS-Device Macro.

11.2 Block Diagram

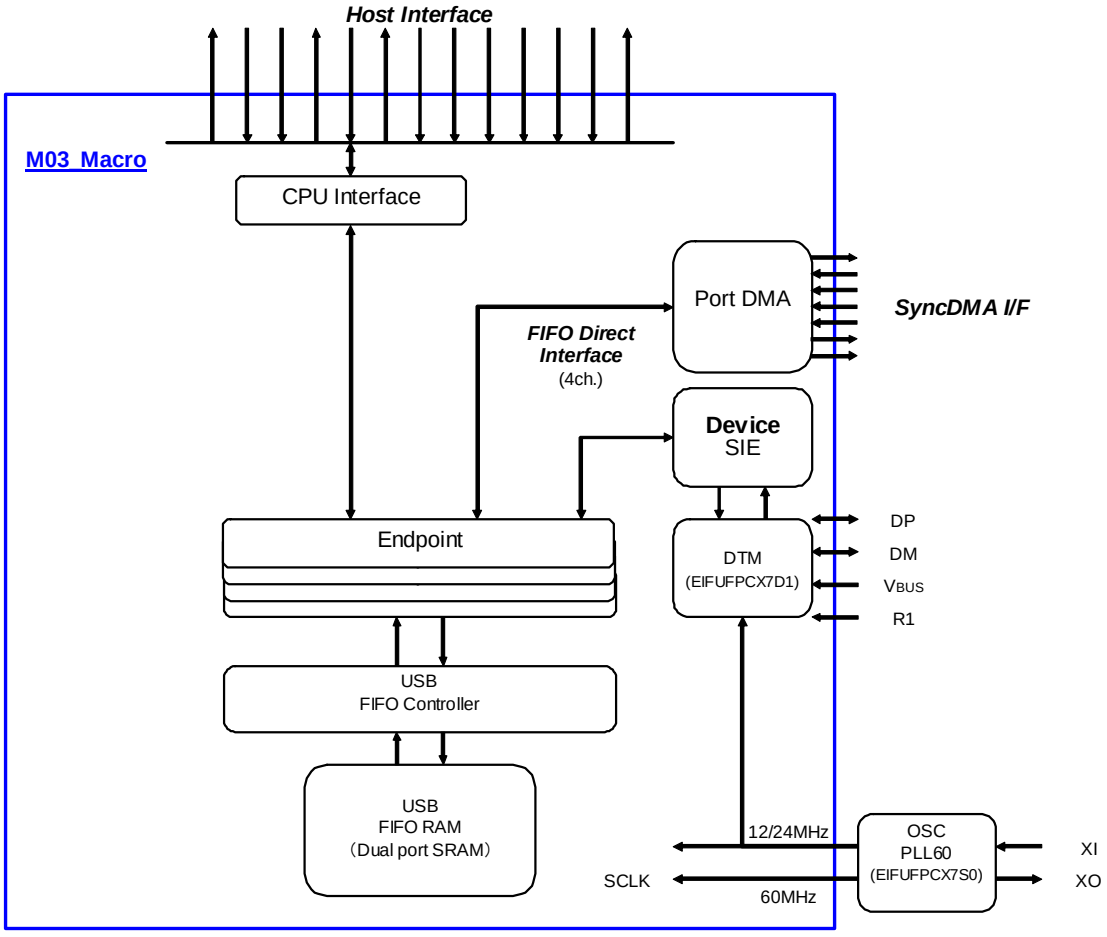


Fig.11.1 Block diagram of USB HS-Device

11.3 Features

The USB HS-Device Macro features the following basic functions.

<USB2.0 Device Function>
- Supports transfer of HS (480Mbps) and FS (12Mbps).
- Built in FS/HS Termination function (no external circuit needed)
- VBUS 5V Interface (an external protection circuit needed))
- Supports Control Bulk and Interrupt Transfer.
- Supports the End Points (8 in total) shared by Control (End Point 0) and Bulk/Interrupt.

<CPU Interface>
- General-purpose CPU Interface with the 16-bit or 8-bit width
- Compatible with Littler Endian.
- Register Table compiled by addition/deletion made to that of the HS-Device of S1R72V05.

< Oscillation Circuit >
- USB clock input compatible with the 12MHz or 24MHz crystal transducer.
  (1MΩ built in for the oscillation circuit and as the feedback resistor)
- The following frequencies are available for internal clock based on the USB input clock.
  USB input clock: 12MHz or 24MHz
  Internal USB clock: 60MHz (via PLL for the built-in USB)

12. APB BRIDGE (APB)

12.1 Description

This module, mounted on the APB bus connecting the internal high-speed bus AHB1 and the low-speed APB-functional devices (APB Devices), is a slave device on the AHB bus performing the bridge function to control the AHB bus on behalf of each APB Device. With this APB bridge, each APB Device is discharged from the responsibility to control the AHB bus, and it has only to control the simple APB bus. This APB bridge is irrelevant to a software programmers usually; they are required occasionally, though, to make a setting of the wait function (1-3 wait) to the APB Devices. Use it normally as it is reset.

12.2 Block Diagram

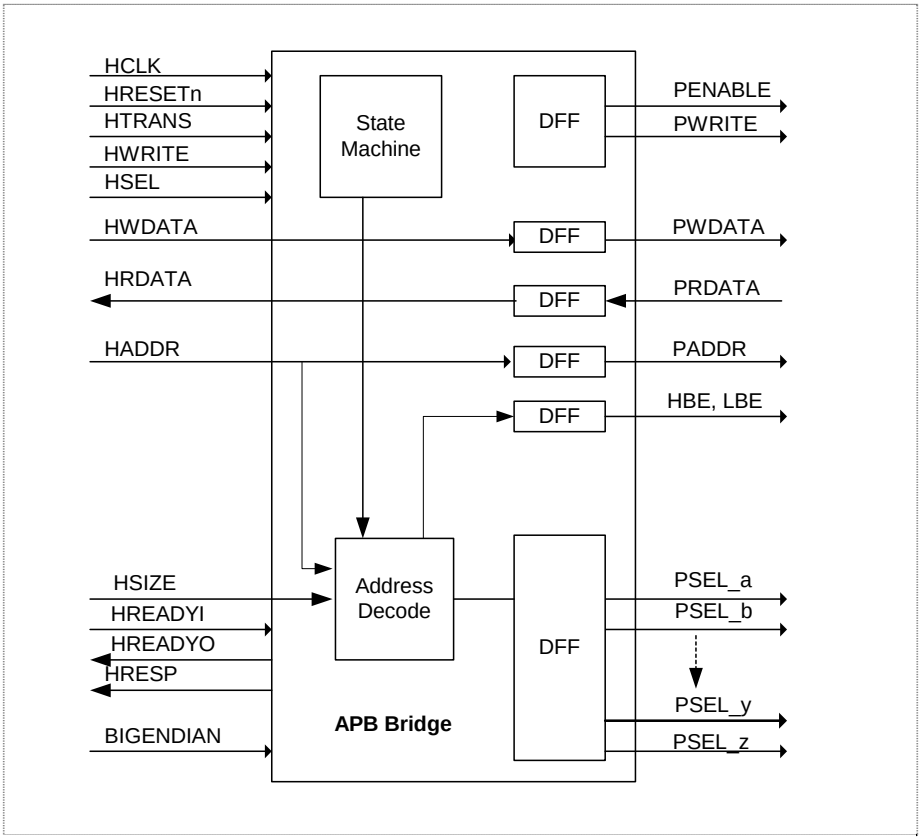


Fig.12.1 Block diagram of the APB bridge

12.3 External Pins

The IC has no external pins for the APB bridge.

12.4 Registers

12.4.1 List of Registers

The base address of the registers for APB is 0xFFFE_0000.

Table 12.1 List of registers (Base address: 0xFFFE_0000)

Address offset	Register name	Register name's abbreviation	Default	R/W	Data access size
0x00	APB WAIT0 Register	APBWAIT0	0x0050_0500	R/W	32bit
0x04	APB WAIT1 Register	APBWAIT1	0x0000_0000	R/W	32bit
0x08	APB WAIT2 Register	APBWAIT2	0x0050_0000	R/W	32bit

12.4.2 Detailed Description of Registers

APB WAIT0 register (APBWAIT0)

APB[0x00] Default value = 0x0050_0500 Read/Write

PW0FCNF [1:0]		PW0ECNF [1:0]		PW0DCNF [1:0]		PW0CCNF [1:0]		PW0BCNF [1:0]		PW0ACNF [1:0]		PW09CNF [1:0]		PW08CNF [1:0]	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PW07CNF [1:0]		PW06CNF [1:0]		PW05CNF [1:0]		PW04CNF [1:0]		PW03CNF [1:0]		PW02CNF [1:0]		PW01CNF [1:0]		PW00CNF [1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **PWxCNF[1:0] (x=00 to 0F)**

00: 0 wait Basic 2-APB cycle (default)
01: 1 wait 2-APB cycle + 1-wait cycle = 3-APB cycle
10: 2 wait 2-APB cycle + 2-wait cycle = 4-APB cycle
11: 3 wait 2-APB cycle + 3-wait cycle = 5-APB cycle

APB WAIT1 register (APBWAIT1)

APB[0x04] Default value = 0x0000_0000 Read/Write

PW1FCNF [1:0]		PW1ECNF [1:0]		PW1DCNF [1:0]		PW1CCNF [1:0]		PW1BCNF [1:0]		PW1ACNF [1:0]		PW19CNF [1:0]		PW18CNF [1:0]	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PW17CNF [1:0]		PW16CNF [1:0]		PW15CNF [1:0]		PW14CNF [1:0]		PW13CNF [1:0]		PW12CNF [1:0]		PW11CNF [1:0]		PW10CNF [1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **PWxCNF[1:0] (x=10 to 1F)**

00: 0 wait Basic 2-APB cycle (default)
01: 1 wait 2-APB cycle + 1-wait cycle = 3-APB cycle
10: 2 wait 2-APB cycle + 2-wait cycle = 4-APB cycle
11: 3 wait 2-APB cycle + 3-wait cycle = 5-APB cycle

APB WAIT2 register (APBWAIT2)

APB[0x08] Default value = 0x0050_0000 Read/Write

PW2FCNF [1:0]		PW2ECNF [1:0]		PW2DCNF [1:0]		PW2CCNF [1:0]		PW2BCNF [1:0]		PW2ACNF [1:0]		PW29CNF [1:0]		PW28CNF [1:0]	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PW27CNF [1:0]		PW26CNF [1:0]		PW25CNF [1:0]		PW24CNF [1:0]		PW23CNF [1:0]		PW22CNF [1:0]		PW21CNF [1:0]		PW20CNF [1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **PWxCNF[1:0] (x=20 to 2F)**

00: 0 wait Basic 2-APB cycle (default)
01: 1 wait 2-APB cycle + 1-wait cycle = 3-APB cycle
10: 2 wait 2-APB cycle + 2-wait cycle = 4-APB cycle
11: 3 wait 2-APB cycle + 3-wait cycle = 5-APB cycle

Table 12.2 List of APBWAIT[2:0] registers and corresponding APB Devices

APBWAIT0		APBWAIT1		APBWAIT2	
PW0xCNF	APB Device	PW1xCNF	APB Device	PW2xCNF	APB Device
PW00CNF	APB bridge	PW10CNF	Reserved	PW20CNF	Reserved
PW01CNF	Reserved	PW11CNF	GPIO	PW21CNF	Reserved
PW02CNF	Reserved	PW12CNF	SPI	PW22CNF	Reserved
PW03CNF	DMA1C	PW13CNF	Reserved	PW23CNF	Timer B
PW04CNF	CF Attribute CF Common	PW14CNF	Reserved	PW24CNF	UART2
PW05CNF	CF I/O	PW15CNF	UART1	PW25CNF	UART3
PW06CNF	CF CTL	PW16CNF	Reserved	PW26CNF	Reserved
PW07CNF	ARS	PW17CNF	Reserved	PW27CNF	Reserved
PW08CNF	Camera Interface-1	PW18CNF	RTC	PW28CNF	Camera Interface-2
PW09CNF	JPEG Resize-1	PW19CNF	DMA2C	PW29CNF	JPEG Resize-2
PW0ACNF	JPEG module/FIFO-1	PW1ACNF	Memory CNTL	PW2ACNF	JPEG module/FIFO-2
PW0BCNF	JPEG Codec-1	PW1BCNF	Timer A	PW2BCNF	JPEG Codec-2
PW0CCNF	JPEG-DMAC	PW1CCNF	WDT	PW2CCNF	ADC
PW0DCNF	I^2C	PW1DCNF	SYS-CNTL	PW2DCNF	SD-Memory
PW0ECNF	I^2S	PW1ECNF	Reserved	PW2ECNF	Reserved
PW0FCNF	(INT-CNTL)	PW1FCNF	INT-CNTL	PW2FCNF	USB_2.0 Device

Table 12.3 Built-in I/O map

Base Address	Space Size	S2S65A00
0xFFFD_0000	4KB	Reserved
0xFFFD_1000	4KB	Reserved
0xFFFD_2000	4KB	Reserved
0xFFFD_3000	4KB	**Timer B**
0xFFFD_4000	4KB	**UART2**
0xFFFD_5000	4KB	**UART3**
0xFFFD_6000	4KB	Reserved
0xFFFD_7000	4KB	Reserved
0xFFFD_8000	4KB	**Camera Interface-2**
0xFFFD_9000	4KB	**JPEG Resize-2**
0xFFFD_A000	4KB	**JPEG module/FIFO-2**
0xFFFD_B000	4KB	**JPEG Codec-2**
0xFFFD_C000	4KB	**ADC**
0xFFFD_D000	4KB	**SD-Memory**
0xFFFD_E000	4KB	Reserved
0xFFFD_F000	4KB	**USB_2.0 Device**
0xFFFE_0000	4KB	**APB bridge**
0xFFFE_1000	4KB	Reserved
0xFFFE_2000	4KB	Reserved
0xFFFE_3000	4KB	**DMA1C**
0xFFFE_4000	2KB	CF attribute
0xFFFE_4800	2KB	CF common
0xFFFE_5000	2KB	CF I/O
0xFFFE_5800	1KB	CF ture IDE CS1#
0xFFFE_5C00	1KB	CF ture IDE CS2#
0xFFFE_6000	4KB	CF control
0xFFFE_7000	4KB	**ARS**
0xFFFE_8000	4KB	**Camera Interface-1**
0xFFFE_9000	4KB	**JPEG Resize-1**
0xFFFE_A000	4KB	**JPEG module/FIFO-1**
0xFFFE_B000	4KB	**JPEG Codec-1**
0xFFFE_C000	4KB	**JPEG-DMAC**
0xFFFE_D000	4KB	**I^2C**
0xFFFE_E000	4KB	**I^2S**
0xFFFE_F000	4KB	(INT-CNTL)
0xFFFF_0000	4KB	Reserved
0xFFFF_1000	4KB	**GPIO**
0xFFFF_2000	4KB	**SPI**
0xFFFF_3000	4KB	Reserved
0xFFFF_4000	4KB	Reserved
0xFFFF_5000	4KB	**UART1**
0xFFFF_6000	1KB	Reserved
0xFFFF_7000	4KB	Reserved
0xFFFF_8000	4KB	**RTC**
0xFFFF_9000	4KB	**DMA2C**
0xFFFF_A000	4KB	**Memory CNTL**
0xFFFF_B000	4KB	**Timer A**
0xFFFF_C000	4KB	**WDT**
0xFFFF_D000	4KB	**SYS-CNTL**
0xFFFF_E000	4KB	Reserved
0xFFFF_F000	4KB	**INT-CNTL**

13. System Controller (SYS)

13.1 Description

This block controls clocks, means to get the system lower powered, matters involving the chip and the entire system, and memory map mainly.

This block features:

- Supports transition to the low-power mode (IDLE) by means of HALT.
- Allows selection of ON/OFF of the CPU and bus clocks within HALT.
- Allows dynamic switching of frequencies of the CPU/AHB/APB clocks.
- Allows turning ON/OFF the clock for each I/O built in.
- Supports the operation mode in 32KHz.
- Supports the means to switch the PLL clock.
- Supports software reset.
- Programmable Clock generator for UART

13.2 Operation mode

S2S65A00 has four operational states: Low-speed, low-speed HALT, high-speed, and high-speed HALT modes.

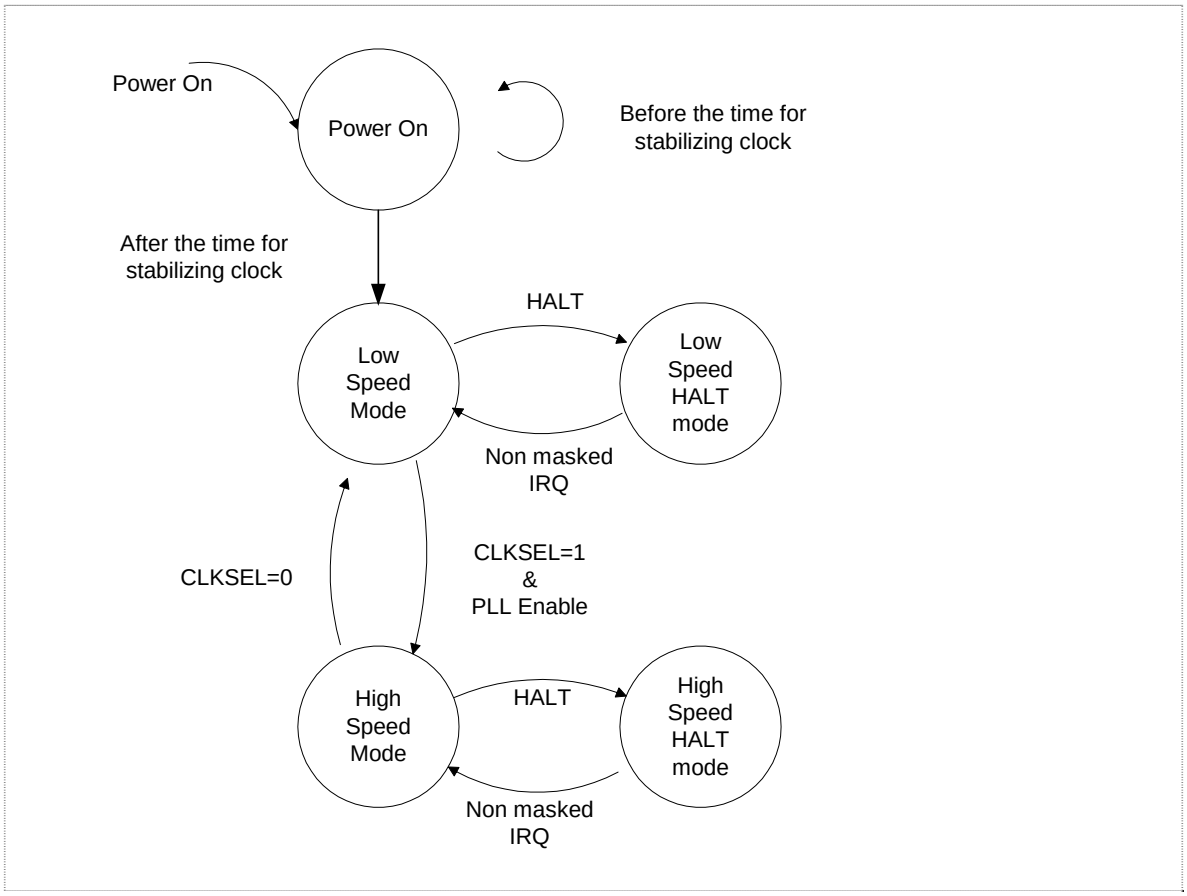


Fig.13.1 Operational states inside the system

Note: Transition from the Low Speed Mode to High Speed Mode can be made by the software after waiting the PLL stabilizing time (Max. 100ms).

Each operational state (mode) is described in the following pages.

13.2.1 Power-On State

This is not a special operational mode. The reset signal generated by turning power on makes the system enter this mode. The clock input of 32KHz is used. Assuming external 32KHz, it is necessary to secure 3 seconds at the maximum for the rise time, and this state is held for such duration. The state transits to the next operational mode, or the low-speed mode, about 3 seconds after the resetting is released.

13.2.2 Low-Speed Mode

PLL is stopped in this mode immediately after resetting. This mode is operational only in 32KHz. The CPU starts in this state immediately after power-on resetting.

Setting PLL in this mode makes it possible to set any multiplying factor to the clock (though there are some restrictions on UART and the basic clock generated by a timer). PLL is operated at the frequency set (i.e. in the high-speed mode) switched to by the software after waiting some time under the software control because PLL takes some time (Max:100ms) before it is stabilized. Any change in the PLL settings required in the high-speed mode must be made in the low-speed mode.

13.2.3 Low-Speed HALT Mode

A command issued in the low-speed mode instructing the HALT operation makes the system transit into this mode, which consumes power the least among all the modes. This mode can stop both built-in I/O bus and the supply of the clock to it, which are dependent on their register settings. Exit from this mode (returning to the low-speed mode always) is caused by unmasked interrupt. For example, change in the interrupt-enabled GPIO input, external interrupt pin, or interrupt from the timer operating at 32KHz can cause exit from this mode.

13.2.4 High-Speed Mode

The system transits into this mode under the software control after PLL is started in the low-speed mode, operating with the basic clock set to the PLL at a multiplying factor which is supplied to the CPU and internal buses.

13.2.5 High-Speed HALT Mode

A command issued in the high-speed mode instructing the HALT operation makes the system transit into this mode. In this mode, both clock-stop of each I/O bus and CPU set before entering this mode are valid at the same time. Accordingly, the low-power operation is attained by using this mode frequently, i.e. by issuing HALT whenever CPU is not required to operate. Unmasked interrupt returns this mode to the high-speed mode. For example, enabled timer-interrupt or UART reception interrupt can cause interrupt, allowing subsequent transition to the high-speed mode promptly.

13.3 External Pins

The IC has the external pins for the system controller as follows.

Pin name	I/O	Pin functions	Multiplex pin/Remarks
RESET#	I	Hardware reset input	None
SYSCLKI	I/O	32KHz crystal oscillator clock input or Internal clock output. (When SYSCKSEL is High.)	Depends on the SYSCKSEL pin,
SYSCKSEL	I	32KHz system clock select signal Low : crystal transducer select High : crystal oscillator select	None

13.4 Registers

13.4.1 List of Registers

Shown below are the system control registers. The base address of these registers is 0xFFFF_D000.

Table 13.1 List of registers (Base address: 0xFFFF_D000)

Address Offset	Register Name	Abbreviation Name	Initial Value	R/W	Data Access Size
0x00	Chip ID Register	CHIPID	0x065A_000X	RO	32bit
0x04	Chip Configuration Register	CHIPCFG	0x0000_XXXX	RO	16/32bit
0x08	PLL Setting Register 1	PLLSET1	0x0421_84AE	R/W	32bit
0x0C	PLL Setting Register 2	PLLSET2	0x0000_0000	(R/W)	16/32bit
0x10	HALT Mode Clock Control Register	HALTMODE	0x0000_0000	R/W	16/32bit
0x14	IO Clock Control Register	IOCLKCTL	0x0000_0000	R/W	16/32bit
0x18	Clock Select Register	CLK_SEL	0x0000_0000	R/W	16/32bit
0x1C	HALT Control Register	HALTCTL	—	WO	16/32bit
0x20	Memory Remap Register	REMAP	0x0000_0000	R/W	16/32bit
0x24	Software Reset Register	SOFTRST	—	WO	32bit
0x28	UART1 Clock Divider Register	UART1DIV	0x0000_0000	R/W	16/32bit
0x2C	UART2 Clock Divider Register	UART2DIV	0x0000_0000	R/W	16/32bit
0x30	UART3 Clock Divider Register	UART3DIV	0x0000_0000	R/W	16/32bit
0x34	Timer-B Clock Select Register	TIMBCKSEL	0x0000_0000	R/W	16/32bit
0x40	MD Bus Pull-down Control Register	MDPLDCTL	0x0000_0000	R/W	16/32bit
0x44	SDD Bus Pull-down Control Register	SDDPLDCTL	0x0000_0000	R/W	16/32bit
0x48	GPIOE Resistor Control Register	PORTERCTL	0x0000_0000	R/W	16/32bit
0x4C	GPIOF Resistor Control Register	PORTFRCTL	0x0000_0000	R/W	16/32bit
0x50	GPIOG Resistor Control Register	PORTGRCTL	0x0000_0000	R/W	16/32bit
0x54	GPIOH Resistor Control Register	PORTHRCTL	0x0000_0000	R/W	16/32bit
0x58	GPIOI Resistor Control Register	PORTIRCTL	0x0000_0000	R/W	16/32bit
0x5C	GPIOJ Resistor Control Register	PORTJRCTL	0x0000_0000	R/W	16/32bit
0x60	GPIOK Resistor Control Register	PORTKRCTL	0x0000_0000	R/W	16/32bit
0x64	Internal TEST Mode Register	ITESTM	0x0000_0000	R/W	32bit
0x68	Embedded Memory Control Register	EMBMEMCTL	0x0000_0000	R/W	32bit
0x6C	Misc Register	MISC	0x0000_0000	R/W	32bit

13.4.2 Detailed Description of Registers

Chip ID Register (CHIPID)

SYS[0x00] Default value = 0x065A_000X Read Only

PRODUCT ID [23:16]							
31	30	29	28	27	26	25	24
PRODUCT ID [15:8]							
23	22	21	20	19	18	17	16
PRODUCT ID [7:0]							
15	14	13	12	11	10	9	8
Reserved					REVISION CODE		
7	6	5	4	3	2	1	0

Bits[31:8]: **Product ID Code [23:0]**
This chip has “065A00h” in the hexadecimal format embedded here.

Bits[7:3]: **Reserved**

Bits[2:0]: **Revision Code [2:0]**
Shows the revision number of this IC. The first chip is revision 1, expressed as “01h”. The revision number is incremented by 1 each time the chip is revised.

Chip Configuration Register (CHIPCFG)

SYS[0x04] Default value = 0x0000_XXXX Read Only

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
CONF [15:8]							
15	14	13	12	11	10	9	8
CONF [7:0]							
7	6	5	4	3	2	1	0

Bits[15:0]: **CONF [15:0]**
CNF[15:0] are used to decide the internal operation by sampling the state of the pull-up/pull-down resistor connected to the MD[15:0] data bus at the rise of RESET#.
For more detailed functions, see “4.1 System Configuration.”

PLL Setting Register 1 (PLLSET1)

SYS[0x08] Default value = 0x0421_D46A Read/Write

n/a	CS [1:0]		CP [4:0]				
31	30	29	28	27	26	25	24
RS [3:0]				VC[3:0]			
23	22	21	20	19	18	17	16
N-Counter [3:0]				W-Divider [1:0]]		L-Counter [9:8]	
15	14	13	12	11	10	9	8
L-Counter [7:0]							
7	6	5	4	3	2	1	0

This register decides the operational frequency of the built-in PLL. To change a setting, be sure to put PLL in the power-down state first by setting "0" to bit 1 (PLLN) of the PLL Setting Register 2 (SYS[0x0C]). For more concrete values recommended, see Appendices A and B.

Bits [30:29]: **CS [1:0]**
Adjust the capacity of LPF of the built-in PLL. Set the recommended values.

Bits [28:24]: **CP [4:0]**
Adjust the current of CP of the built-in PLL. Set the recommended values.

Bits [23:20]: **RS [3:0]**
Adjust the resistance of LPF of the built-in PLL. Set the recommended values.

Bits [19:16]: **VC[3:0]**
Parameters to operate VCO in the built-in PLL. Set the recommended values.

Bits [15:12]: **NN value N-Counter [3:0]**
N-Counter is used to decide the multiplying factor for PLL together with the L-Counter mentioned below. For more details, see the descriptions of LL Value and the examples of settings for PLL.

Bits [11:10]: **V-Divider [1:0]**
Specify a value to divide the PLL-Out frequency with in the built-in PLL.
00: Disabled
01: 1/2 division
10: 1/4 division
11: 1/8 division
Set the recommended values.

Bits [9:0]: **LL Value L-Counter [9:0]**
Values which decide the multiplying factor for the built-in PLL. Set the recommended values.

PLL Output = (N-Counter+1) × (L-Counter+1) × CLKI_L
= NN × LL × CLKI_L

NN = N-Counter, LL = L-Counter, CLKI_L = External clock input (32.768KHz)

PLL Setting Register 2 (PLLSET2)

SYS[0x0C] Default value = 0x0000_0000 (Read/Write)

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
n/a							PLLEN
7	6	5	4	3	2	1	0

Bits [7:4]: **Reserved**

Bit 0: **PLLEN**

Specify whether Enable or Disable the built-in PLL. The value of PLL Setting Register1 can be changed only if PLLEN = 0.

0: PLL is Disabled. (PLL stays in the power-down state.)
1: PLL is Enabled.

HALT Mode Clock Control Register (HALTMODE)

SYS[0x10] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
Reserved (0)							
23	22	21	20	19	18	17	16
Reserved (0)							
15	14	13	12	11	10	9	8
Reserved (0)							
7	6	5	4	3	2	1	0
CPUCKSEL [1:0]		n/a		HALT_MDCLK [4:0]			

This register decides the clock frequencies of CPU and internal buses (AHB1, AHB2, APB Bus) and which clock to stop during the HALT mode.

Bits [31:8]: **Reserved (0)**

Bits [7:6]: **CPUCKSEL [1:0]**

Specify the clock division for CPU/AHB1/AHB2/APB.
CPU, AHB1, AHB2, and APB have all the same clock value. If the CPU clock frequency is divided, the AHB1, AHB2, and APB clock frequencies all get the same value. Any change in the clock frequency is reflected immediately after it is written in without any glitch.

00: 1/1 of the PLL output is supplied to CPU.
01: 1/2 of the PLL output is supplied to CPU.
10: 1/4 of the PLL output is supplied to CPU.
11: 1/8 of the PLL output is supplied to CPU.

Bits [3:0]: **HALT_MDCLK [3:0]**

Bits [4:0], if "0" is written in a bit or bits, stop supplying clock to the corresponding internal bus or buses (CPU, AHB1-Bus, AHB2-Bus, or APB-bus, or all of them) during the HALT mode. Unmasked interrupt causes exit from the HALT mode. Accordingly, it is necessary to keep supplying clock to an internal device (UART, Timer, Ethernet, or GPIO) in which to cause interrupt.

0: Clock keeps to be supplied. (Clock On)
1: Clock stops to be supplied. (Clock Off)
Bit3: Specify clock On/Off for ARM720T.
Bit2: Specify clock On/Off for AHB1.*
Bit1: Specify clock On/Off for AHB2.
Bit0: Specify clock On/Off for APB.

"1" written in bits [3:0] to specify clock Off does not stop the clock immediately; this specification becomes valid only when the HALT mode is entered in by means of HALT Control Register. This function serves for saving power while CPU is idle and standing by, for example when the CPU has no job to execute or it is waiting for some interrupt event.

* **Restriction on use: If AHB1 bus clock is stopped by HALT, CPUCLK must be specified to be stopped by HALT, too. CPUCLK may be specified to be stopped by HALT without restriction independently from the AHB1 bus clock.**

IO Clock Control Register (IOCLKCTL)

SYS[0x14] Default value = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
Reserved (0)	IOCLKCTL [14:8]						
15	14	13	12	11	10	9	8
IOCLKCTL [7:0]							
7	6	5	4	3	2	1	0

Each bit of this register specifies whether to supply or stop supplying clock to the corresponding I/O (Timer, UART, SPI, etc.). "1" written in causes supply of clock, and "0" stops the supply. Writing "0" for any unused I/O function serves for saving power consumption.

Bit 15: **Reserved (0)**

Bits [6:0]: **IOCLKCTL [14:0]**

Bit14 (ADC_CLKEN)	: Specify clock On/Off for ADC.
Bit13 (UART1_CLKEN)	: Specify clock On/Off for UART1.
Bit12 (UART2_CLKEN)	: Specify clock On/Off for UART2.
Bit11 (UART3_CLKEN)	: Specify clock On/Off for UART3.
Bit10 (DMA1C_CLKEN)	: Specify clock On/Off for DMA1C.
Bit9 (DMA2C_CLKEN)	: Specify clock On/Off for DMA2C.
Bit8 (SPI_CLKEN)	: Specify clock On/Off for SPI.
Bit7 (I^2C_CLKEN)	: Specify clock On/Off for I^2C.
Bit6 (I^2S_CLKEN)	: Specify clock On/Off for I^2S.
Bit5 (TIMERA_CLKEN)	: Specify clock On/Off for Timer Ch0/1/2.
Bit4 (TIMERB_CLKEN)	: Specify clock On/Off for Timer Ch0/1/2.
Bit3 (CF_CLKEN)	: Specify clock On/Off for Timer Ch0/1/2.
Bit2 (SD-MEM)	: Specify clock On/Off for CF Card Interface.
Bit1 (USB_CLKEN)	: Specify clock On/Off for SD-Memory Interface.
Bit0 (Reserved)	: Reserved

Clock Select Register (CLK_SEL)

SYS[0x18] Default value = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
n/a		Reserved		N/a			CLKSEL
7	6	5	4	3	2	1	0

Bit [5:4]: **Reserved[1:0]**

Bit 0: **CLKSEL**

Decides which of 32KHz or PLL output to use as the system clock. To specify "use PLL output", first make settings for the PLL parameters, enable PLL, secure proper PLL stabilization time (100ms), and set "1" to this bit for changing to PLL output.

0: 32KHz

1: PLL output

HALT Control Register (HALTCTL)

SYS[0x1C] Default value = — Write Only

Halt Control [31:24]							
31	30	29	28	27	26	25	24
Halt Control [23:16]							
23	22	21	20	19	18	17	16
Halt Control [15:8]							
15	14	13	12	11	10	9	8
Halt Control [7:0]							
7	6	5	4	3	2	1	0

Bits [31:0]: **Halt Control [31:0]**

Any value written in this register causes the chip to enter the HALT mode.

Memory Remap Register (REMAP)

SYS[0x20] Default value = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
n/a						REMAP2	REMAP1
7	6	5	4	3	2	1	0

This register allows making changes in memory map after reset. The Remap function assigns the SDRAM space to the space visible at address “0x0” after resetting. For more details, see Appendix to SYSTEM CONTROLLER. The register needs not be used usually. If an OS running on a RAM basis is used, though, the remapping may make the use of the memory map easier.

Bit1: **REMAP2**

Allows making a change in memory map on the AHB2 Bus side.

0: Applies the memory map after reset.

1: Makes a change in the memory map after reset.

Bit0: **REMAP1**

Allows making a change in memory map on the AHB1 Bus side.

0: Applies the memory map after reset.

1: Makes a change in the memory map after reset.

Note: Take care in the operation to make a change in the memory map by executing the change code in a field unaffected by the memory map, for example.
Also it is recommended to set REMAP1 and REMAP2 simultaneously to avoid any contradiction in memory maps between AHB1 and AHB2.

Software Reset Register (SOFTRST)

SYS[0x24] Default value = — Write Only

Software Reset [31:24]							
31	30	29	28	27	26	25	24
Software Reset [23:16]							
23	22	21	20	19	18	17	16
Software Reset [15:8]							
15	14	13	12	11	10	9	8
Software Reset [7:0]							
7	6	5	4	3	2	1	0

Bits [31:0]: **Software Reset [31:0]**

Writing "AA5555AAh" in this register initializes all the registers in this IC (S2S65A00) and resets the CPU.

UART1/2 Clock Divider Register (UART1/2DIV)

SYS[0x28/2C] Default value = 0x0000_0000 Read/Write

N/a							
31	30	29	28	27	26	25	24
N/a							
23	22	21	20	19	18	17	16
N/a							
15	14	13	12	11	10	9	8
UART 1/2 CLKDIV [7:0]							
7	6	5	4	3	2	1	0

This register is used for a baud-rate TIME-based divider for UART1/2. In this IC, the PCLK clock is divided basically into its division rate × 1/2 frequency, and supplied to the UART's SCLK (= baud-rate generating clock).

Bits [7:0]: **UART1/2CLKDIV [7:0]**

Generate a division rate 1/N. (N = Bits[7:0]+1)

0: 1/1

1: 1/2 division

•

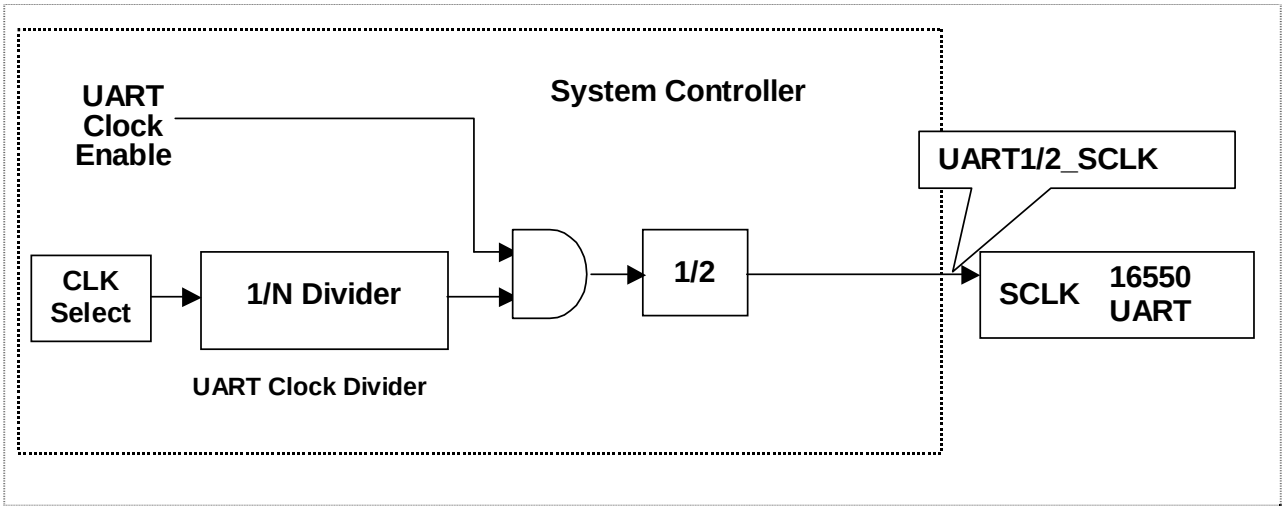
•

255: 1/256 division

UART1/2 is supplied with UART1/2_SCLK which is a clock divided into 1/2 of the divided UART1/2CLKDIV.

Thus, UART1/2_SCLK = (PCLK's frequency) * 1/N * 1/2. (See the following chart.)

Note: This UART1/2_SCLK is very different from SCLK, the clock for SPI Interface.



UART3 Clock Divider Register (UART3DIV)

SYS[0x30] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
N/a							
23	22	21	20	19	18	17	16
N/a							
15	14	13	12	11	10	9	8
N/a							UART3_CKSEL
7	6	5	4	3	2	1	0
UART 3 CLKDIV [7:0]							

This register is used for a baud-rate TIME-based divider for UART1/2/3. In this IC, the PCLK clock is divided basically into its division rate × 1/2 frequency, and supplied to the UART's SCLK (= baud-rate generating clock).

Bits8: **UART3CKSEL**

Allows selection of the external input clock for UART3 only.

0: Internal clock (Refer to Bit[7:0] for division rate.)

1: External clock

Bits [7:0]: **UART3CLKDIV [7:0]**

Generate a division rate 1/N. (N = Bits[7:0]+1)

0: 1/1

1: 1/2 division

•

•

255: 1/256 division

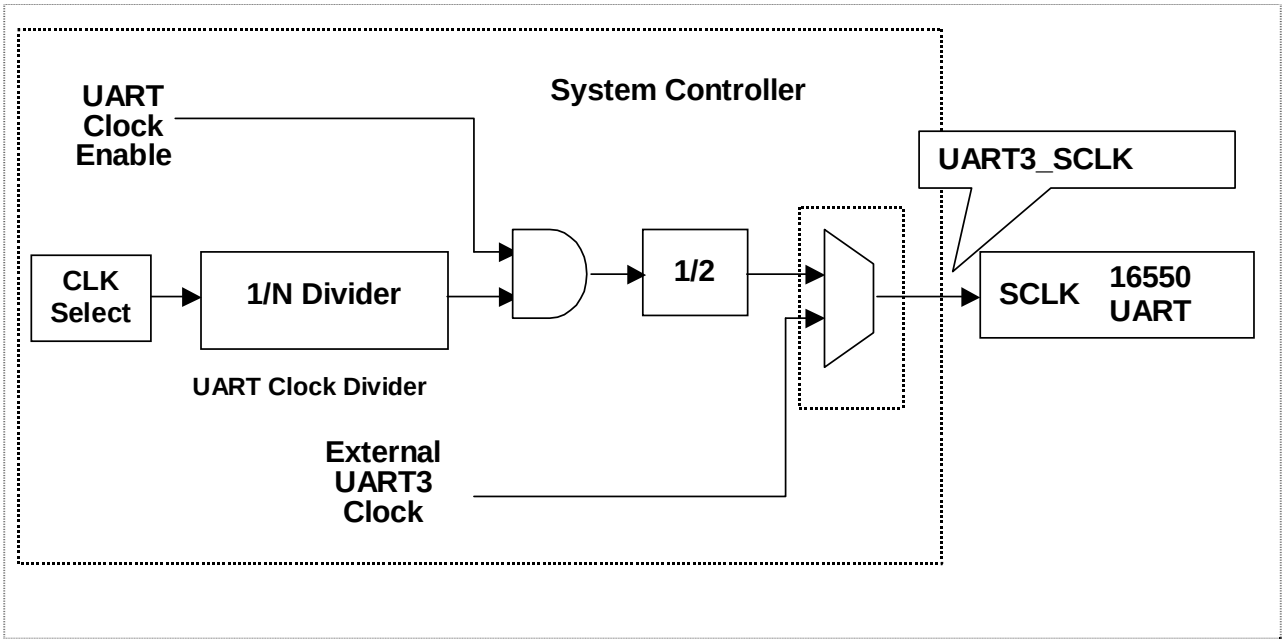
When you select internal clock, UART3 is supplied with UART3_SCLK which is a clock divided into 1/2 of the divided UART3CLKDIV.

Thus, UART3_SCLK = (PCLK's frequency) * 1/N * 1/2.

When you select external clock, input external clock is supplied to UART3 as it is.

(See the following chart.)

Note: This UART3_SCLK is very different from SCLK, the clock for SPI Interface.



Timer B Clock Select Register (TIMBCKSEL)							
SYS[0x34] Default value = 0x0000_0000							Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
N/a							CLKSEL
7	6	5	4	3	2	1	0

Bit 0: **CLKSEL**

This register selects the clock supplied to Timer-B.

0: 1/8 of PLL output

1: 32KHz

MD Bus Pulldown Control Register (MDPLDCTL)							
SYS[0x40] Default value = 0x0000_0000							Read/Write
Reserved							
31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
MDPLDNDIS [15:8]							
15	14	13	12	11	10	9	8
MDPLDNDIS [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **MDPLDNDIS [15:0]**

Control the connection/separation of the pull-down resistors built in MD[15:0] Bus. The separation is possible as required (in particular, if an external pull-up resistor is connected for a low-power application) after resetting.

Each bit corresponds to each pin of MD [15:0].

0: Pull-down resistor enabled (default after resetting)

1: Pull-down resistor disabled

SDD Bus Pulldown Control Register (SDDPLDCTL)

SYS[0x44] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SDDPLDNDIS [15:8]							
7	6	5	4	3	2	1	0
SDDPLDNDIS [7:0]							

Bits [15:0]: **SDDPLDNDIS [15:0]**

Control the connection/separation of the pull-down resistors built in SDD[15:0] Bus. The separation is possible as required (in particular, if an external pull-up resistor is connected for a low-power application) after resetting.

Each bit corresponds to each pin of SDD [15:0].

0: Pull-down resistor enabled (default after resetting)

1: Pull-down resistor disabled

GPIOE Resistor Control Register (PORTERCTL)

SYS[0x48] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PORTEPDDIS [7:0]							

Bits [7:0]: **PORTEPDDIS [7:0]**

This register controls the connection/separation of the pull-down resistors built in GPIOE [7:0] pins.

Each bit corresponds to each pin of GPIOE [7:0].

0: Pull-down resistor enabled (default after resetting)

1: Pull-down resistor disabled

GPIOF Resistor Control Register (PORTFRCTL)

SYS[0x4C] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PORTFPDDIS [7:0]							

Bits [7:0]: **PORTFPDDIS [7:0]**

Control the connection/separation of the pull-down resistors built in GPIOF [7:0] pins.

Each bit corresponds to each pin of GPIOF [7:0].

0: Pull-down resistor enabled (default after resetting))

1: Pull-down resistor disabled

GPIOG Resistor Control Register (PORTGRCTL)

SYS[0x50] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PORTGPDDIS [7:0]							

Bits [7:0]: **PORTGPDDIS [7:0]**

This register controls the connection/separation of the pull-down resistors built in GPIOG [7:0] pins.
Each bit corresponds to each pin of GPIOG [7:0].

0: Pull-down resistor enabled (default after resetting)
1: Pull-down resistor disabled

GPIOH Resistor Control Register (PORTHRCTL)

SYS[0x54] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PORTHPUDIS [7:0]							

Bits [7:0]: **PORTHPUDIS [7:0]**

This register controls the connection/separation of the pull-up resistors built in GPIOH [7:0] pins.
Each bit corresponds to each pin of GPIOH [7:0].

0: Pull-up resistor enabled (default after resetting)
1: Pull-up resistor disabled

GPIOI Resistor Control Register (PORTIRCTL)

SYS[0x58] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						PORTIPUDIS [1:0]	

Bits [1:0]: **PORTIPUDIS [1:0]**

This register controls the connection/separation of the pull-up resistors built in GPIOI [1:0] pins.
Each bit corresponds to each pin of GPIOI [1:0].

0: Pull-up resistor enabled (default after resetting)
1: Pull-up resistor disabled

GPIOJ Resistor Control Register (PORTJRCTL)

SYS[0x5C] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PORTJPDDIS [7:0]							

Bits [7:0]: **PORTJPDDIS [7:0]**

This register controls the connection/separation of the pull-down resistors built in GPIOJ [7:0] pins.
Each bit corresponds to each pin of GPIOJ [7:0].

0: Pull-down resistor enabled (default after resetting)
1: Pull-down resistor disabled

GPIOK Resistor Control Register (PORTKRCTL)

SYS[0x60] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PORTKPDDIS [7:0]							

Bits [7:0]: **PORTKPDDIS [7:0]**

This register controls the connection/separation of the pull-down resistors built in GPIOK [7:0] pins.
Each bit corresponds to each pin of GPIOK [7:0].

0: Pull-down resistor enabled (default after resetting)
1: Pull-down resistor disabled

Internal TEST Mode Register (ITESTM)

SYS[0x64] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0

This register is used for the internal testing of the LSI. No user may operate it.
Leave it as it is reset.

Embedded Memory Control Register (EMBMEMCTL)

SYS[0x68] Default value = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
Reserved		EMBRAMSEL[1:0]		Reserved		EMBWAITEN[1:0]	
7	6	5	4	3	2	1	0

Bits [31:6]: **Reserved (0)**

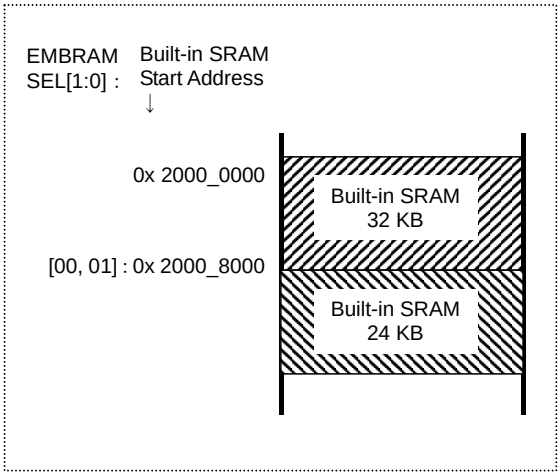
Bits [5:4]: **EMBRAMSEL[1:0] Embedded SRAM Select**

00: Built-in SRAM 24KB is allocated to the JPEG1 line buffer.
(Use JPEG Line Buffer [0xA4] of JPEG controller 1 by setting it to default "0x0020") 32KB is made available as the built-in SRAM with the start address "0x 2000_0000".

01: 4KB is allocated to the JPEG1 line buffer, and 32KB to the JPEG2 line buffer.
(Use JPEG Line Buffer [0xA4] of JPEG controller 1 by setting it to "0x0000", and that of JPEG controller 2 by setting it to the default "0x0020").

10: 56KB is allocated to the JPEG1 line buffer. (Use JPEG Line Buffer [0xA4] of JPEG controller 1 by setting it to "0x0000".)

11: The full size 56KB is made available for the built-in SRAM with the start address "0x 2000_0000".



Bits [3:2]: **Reserved (0)**

Bits [1:0]: **EMBWAITEN[1:0] Embedded SRAM Wait Control**

00: No Wait
01: Read Access Wait ON (Read: 1 wait, Write: no wait)
10: Read Access Wait ON, Read Data Wait ON,
(Read: 2 wait, Write: no wait)
11: Read Access Wait ON, Read Data Wait ON, Write Access Wait On
(Read: 2 wait, Write: 1 wait)

MISC Register (MISC)

SYS[0x6C] Default value = 0x0000_0000 (Read/Write)

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
n/a					Reserved	USB_CLK_CTL	Reserved
7	6	5	4	3	2	1	0

Bits [31:3]: **Not used**

Bit 2: **Reserved**

Bit 1: **USB_CLK_CTL**

Specify the supply of clock to USB HS-Device. Set this bit to "1" to use USB HS-Device.

0: No clock is supplied to USB HS-Device.

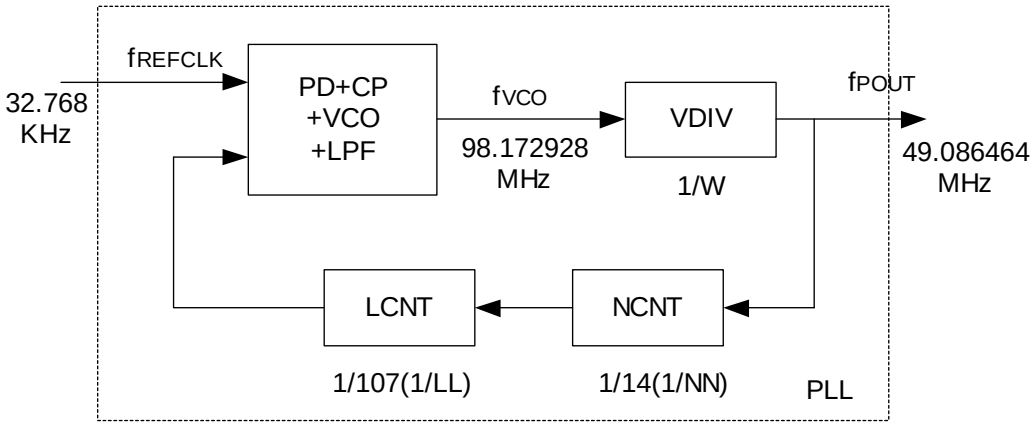
1: A 60MHz clock is supplied to USB HS-Device.

Bit 0: **Reserved**

Notes) Set "0x0000_0002" to use USB HS-Device.

13.5 Appendix A: PLL Setting Example

Shown below are examples of the PLL settings.
If CPU Clock = 49.086464 MHz:



If f_{POUT}= 49.086464 MHz is selected:
f_{REFCLK} = 32.768KHz
f_{POUT} / f_{REFCLK} = 49.086464 MHz / 32.768KHz = 1498 (division rate)
Division rate: 1498 = 2 × 7 × 207 = 107 × 14
Thus, NN = 14 and LL = 107 can be selected as the division values.
NN = 14, or N-Counter [3:0] = 13 = 1101 (Binary)
LL = 107, or L-Counter [9:0] = 106 = 00_0110_1010 (Binary)

V_{CO} frequency: f_{VCO} = f_{POUT} × W = 49.086464 MHz × 2 = 98.172928 MHz
Here, W = 2, or about 100MHz;
thus W = 2 or W-Divider [1:0] = 01 (Binary) suffices.

13.6 Appendix B: PLL Parameter table

To make settings for PLL Setting Register1, use the values shown below corresponding to the frequency range used. PLL requires 100ms at the maximum for its output stabilizing time. Be sure to wait for this time to change 32KHz after turning on PLL, then switch it to desired clock by setting CLKSEL=1. The PLL's frequency can be changed only if the system is running at 32KHz and with PLLEN=0 (= PLL Disable). If it is running at an frequency other than 32KHz, the frequency may not changed to another directly. Switch it to 32KHz first, then to the frequency desired. Shown below is an example of the PLL Setting Register1 settings.

Target frequency (example)	Multiplying factor for 32.768KHz	PLL_Setting Register1 (Hex)	NN N-Counter	LL L-Counter
49.086464MHz	1498 = 14 x 107	0x0421_D46A	1101b	00_0110_1010b

Use the values shown below for W-Divider, VC, RS, CP and CS if VCO frequency is 90 - 100MHz.

VV_Divider	VC[3:0]	RS[3:0]	CP[4:0]	CS[1:0]
01b	0001b	0010b	0_0100b	00b

13.7 Appendix C: Memory Map after Remapping

13.7.1 Memory Map after Remapping (AHB1)

The remap function allows changing the memory map after resetting.

The memory map after remapped may be easier to use if an OS running on a RAM basis is used, though this does not apply to the OS running on a ROM basis.

Table 13.2 Memory Map of AHB1 after Remapping

Start Address	End Address	Size	Device	External Chip Select	Device Bus size
0x0000_0000	0x0FFF_FFFF	256MByte	External SDRAM	SDCS0#	32
0x1000_0000	0x1FFF_FFFF	256MByte	Reserved		
0x2000_0000	0x2FFF_FFFF	128MByte	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256MByte	Reserved		
0x4000_0000	0x4FFF_FFFF	256MByte	External SDRAM	SDCS0#	32
0x5000_0000	0x5FFF_FFFF	256MByte	External SDRAM	SDCS1#	32
0x5000_0000	0x5FFF_FFFF	256Mbyte	Reserved		
0x6000_0000	0x6FFF_FFFF	256Mbyte	Reserved		
0x7000_0000	0x7FFF_FFFF	256Mbyte	Reserved		
0x8000_0000	0x8FFF_FFFF	256Mbyte	Reserved		
0x9000_0000	0x9FFF_FFFF	256Mbyte	Reserved		
0xA000_0000	0xAFFF_FFFF	256Mbyte	Reserved		
0xB000_0000	0xBFFF_FFFF	256Mbyte	Reserved		
0xC000_0000	0xC1FF_FFFF	32MByte	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32MByte	Reserved		
0xC400_0000	0xC5FF_FFFF	32MByte	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32MByte	Reserved		
0xC800_0000	0xC9FF_FFFF	32MByte	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32MByte	Reserved		
0xCC00_0000	0xCDFF_FFFF	32MByte	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32MByte	Reserved		
0xD000_0000	0xDFFF_FFFF	256Mbyte	Reserved		
0xE000_0000	0xEFFF_FFFF	256Mbyte	USB DMA Port		
0xF000_0000	0xFFFF_FFFF	256Mbyte	Built-in I/O Area		32/16/8

Table 13.3 Memory Map of AHB1 before Remapping

Start Address	End Address	Size	Device	External Chip Select	Device Bus size
0x0000_0000	0x01FF_FFFF	32MByte	External ROM	MCS0#	16
0x0200_0000	0x03FF_FFFF	32MByte	Reserved		
0x0400_0000	0x05FF_FFFF	32MByte	External SRAM	MCS1#	16
0x0600_0000	0x07FF_FFFF	32MByte	Reserved		
0x0800_0000	0x09FF_FFFF	32MByte	External SRAM	MCS2#	16
0x0A00_0000	0x0BFF_FFFF	32MByte	Reserved		
0x0C00_0000	0x0DFF_FFFF	32MByte	External SRAM	MCS3#	16
0x0E00_0000	0x0FFF_FFFF	32MByte	Reserved		
0x1000_0000	0x1FFF_FFFF	256MByte	Reserved		
0x2000_0000	0x2FFF_FFFF	256Mbyte	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256MByte	Reserved		
0x4000_0000	0x4FFF_FFFF	256MByte	External SDRAM	SDCS0#	16/32
0x5000_0000	0x5FFF_FFFF	256MByte	External SDRAM	SDCS1#	16/32
0x6000_0000	0x6FFF_FFFF	256Mbyte	Reserved		
0x7000_0000	0x7FFF_FFFF	256Mbyte	Reserved		
0x8000_0000	0x8FFF_FFFF	256Mbyte	Reserved		
0x9000_0000	0x9FFF_FFFF	256Mbyte	Reserved		
0xA000_0000	0xAFFF_FFFF	256Mbyte	Reserved		
0xB000_0000	0xBFFF_FFFF	256Mbyte	Reserved		
0xC000_0000	0xC1FF_FFFF	32MByte	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32MByte	Reserved		
0xC400_0000	0xC5FF_FFFF	32MByte	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32MByte	Reserved		
0xC800_0000	0xC9FF_FFFF	32MByte	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32MByte	Reserved		
0xCC00_0000	0xCDFF_FFFF	32MByte	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32MByte	Reserved		
0xD000_0000	0xDFFF_FFFF	256Mbyte	Reserved		
0xE000_0000	0xEFFF_FFFF	256Mbyte	USB DMA Port		
0xF000_0000	0xFFFF_FFFF	256Mbyte	Built-in I/O Area		32/16/8

13.7.2 Memory Map after Remapping (AHB2)

Table 13.4 Memory Map after Remapping (AHB2)

Start Address	End Address	Size	Device	External Chip Select	Device Bus size
0x0000_0000	0x0FFF_FFFF	256MByte	External SDRAM	SDCS0#	32
0x1000_0000	0x1FFF_FFFF	256MByte	Reserved		
0x2000_0000	0x2FFF_FFFF	128MByte	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256MByte	Reserved		
0x4000_0000	0x4FFF_FFFF	256MByte	External SDRAM	SDCS0#	32
0x5000_0000	0x5FFF_FFFF	256MByte	External SDRAM	SDCS1#	32
0x5000_0000	0x5FFF_FFFF	256Mbyte	Reserved		
0x6000_0000	0x6FFF_FFFF	256Mbyte	Reserved		
0x7000_0000	0x7FFF_FFFF	256Mbyte	Reserved		
0x8000_0000	0x8FFF_FFFF	256Mbyte	Reserved		
0x9000_0000	0x9FFF_FFFF	256Mbyte	Reserved		
0xA000_0000	0xAFFF_FFFF	256Mbyte	Reserved		
0xB000_0000	0xBFFF_FFFF	256Mbyte	Reserved		
0xC000_0000	0xC1FF_FFFF	32MByte	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32MByte	Reserved		
0xC400_0000	0xC5FF_FFFF	32MByte	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32MByte	Reserved		
0xC800_0000	0xC9FF_FFFF	32MByte	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32MByte	Reserved		
0xCC00_0000	0xCDFF_FFFF	32MByte	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32MByte	Reserved		
0xD000_0000	0xDFFF_FFFF	256Mbyte	JPEG2 DMA Port		
0xE000_0000	0xEFFF_FFFF	256Mbyte	JPEG1 DMA Port		32
0xF000_0000	0xFFFF_FFFF	256Mbyte	Reserved		

Table 13.5 Memory Map of AHB2 before Remapping

Start Address	End Address	Size	Device	External Chip Select	Device Bus size
0x0000_0000	0x01FF_FFFF	32MByte	External ROM	MCS0#	16
0x0200_0000	0x03FF_FFFF	32MByte	Reserved		
0x0400_0000	0x05FF_FFFF	32MByte	External SRAM	MCS1#	16
0x0600_0000	0x07FF_FFFF	32MByte	Reserved		
0x0800_0000	0x09FF_FFFF	32MByte	External SRAM	MCS2#	16
0x0A00_0000	0x0BFF_FFFF	32MByte	Reserved		
0x0C00_0000	0x0DFF_FFFF	32MByte	External SRAM	MCS3#	16
0x0E00_0000	0x0FFF_FFFF	32MByte	Reserved		
0x1000_0000	0x1FFF_FFFF	256MByte	Reserved		
0x2000_0000	0x2FFF_FFFF	256Mbyte	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256MByte	Reserved		
0x4000_0000	0x4FFF_FFFF	256MByte	External SDRAM	SDCS0#	16/32
0x5000_0000	0x5FFF_FFFF	256MByte	External SDRAM	SDCS1#	16/32
0x6000_0000	0x6FFF_FFFF	256Mbyte	Reserved		
0x7000_0000	0x7FFF_FFFF	256Mbyte	Reserved		
0x8000_0000	0x8FFF_FFFF	256Mbyte	Reserved		
0x9000_0000	0x9FFF_FFFF	256Mbyte	Reserved		
0xA000_0000	0xAFFF_FFFF	256Mbyte	Reserved		
0xB000_0000	0xBFFF_FFFF	256Mbyte	Reserved		
0xC000_0000	0xC1FF_FFFF	32MByte	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32MByte	Reserved		
0xC400_0000	0xC5FF_FFFF	32MByte	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32MByte	Reserved		
0xC800_0000	0xC9FF_FFFF	32MByte	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32MByte	Reserved		
0xCC00_0000	0xCDFF_FFFF	32MByte	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32MByte	Reserved		
0xD000_0000	0xDFFF_FFFF	256Mbyte	JPEG2 DMA Port		
0xE000_0000	0xEFFF_FFFF	256Mbyte	JPEG1 DMA Port		32
0xF000_0000	0xFFFF_FFFF	256Mbyte	Reserved		

13.8 Appendix D: Clock Control Block Diagram

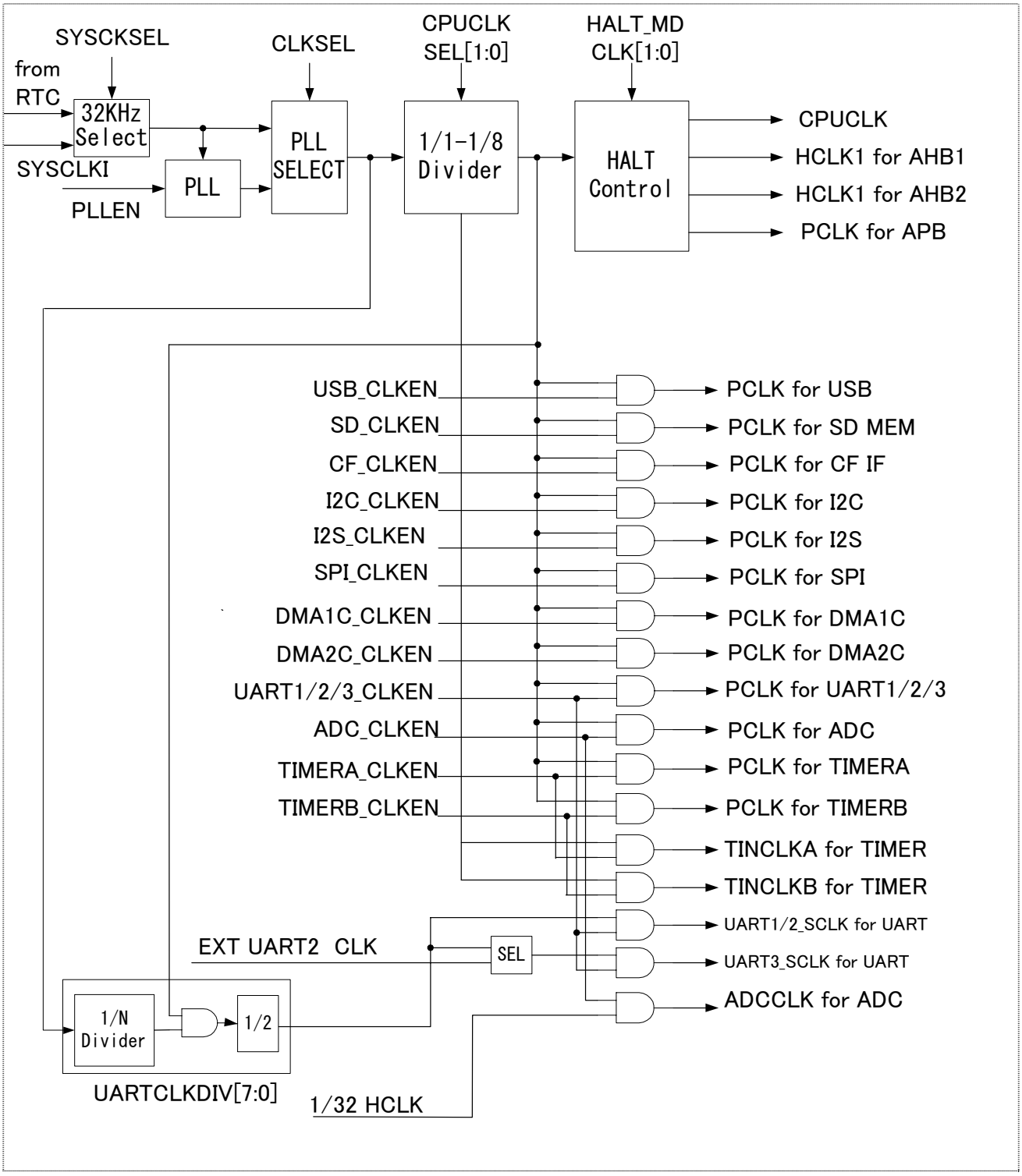


Fig.13.2 Clock Control Block Diagram

A clock signal name “PCLK_for_block-name” in the diagram indicates the clock used for controlling the buses and registers in each block. Also, UART1/2/3_SCLK is used for the clock input for generating the baud-rate at UART1/2/3, respectively; TINCLKA/B is used for the basic clock input for counting in the TimerA/B block, respectively.

13.9 Appendix D: Example of Settings for UART Clock

If CPU Clock = 49.086464 MHz:

Baudrate	Ideal x16 Clock (Hz)	SYS[0x28]	16550 Divisor Value (DEC)	16550 Divisor Value (HEX)	Percent Error %	Actual x16 Clock (Hz)	UART_SCLK
110	1760	0	13945	3679	0.00	1760.0	24543232
300	4800	0	5113	13F9	0.00	4800.2	24543232
600	9600	0	2557	09FD	0.02	9598.4	24543232
1200	19200	0	1278	04FE	0.02	19204.4	24543232
2400	38400	0	639	027F	0.02	38408.8	24543232
4800	76800	0	320	0140	0.13	76697.6	24543232
9600	153600	0	160	00A0	0.13	153395.2	24543232
14400	230400	0	107	006B	0.44	229376.0	24543232
19200	307200	0	80	0050	0.13	306790.4	24543232
38400	614400	0	40	0028	0.13	613580.8	24543232
57600	921600	0	27	001B	1.37	909008.6	24543232
115200	1843200	0	13	000D	2.43	1887940.9	24543232

14. MEMORY CONTROLLER (MEMC)

14.1 Description

The memory controller, built in the AHB bus interface, controls the asynchronous SRAM- and SDRAM-type devices. It supports up to four asynchronous SRAM-type and up to two SDRAM-type devices.

Shown below are the features of the memory controller.

- Compatible with the SRAM timing devices.
- Compatible with the SDRAM devices.
- Allows adjustment of the SDRAM auto-refresh intervals in accordance with devices. If the memory controller is unavailable because it is engaged in memory access or otherwise, it can accumulate plural waiting refreshing operations, performing the burst refreshing as soon as it becomes available. It performs refreshing on the background.
- Supports the SDRAM's self refreshing. Also it can read/write the SDRAM in the self-refreshing state. In this case, it allows selection whether to keep the SDRAM in idle state or to get the SDRAM enter in the self-refresh mode again.

14.1.1 SRAM Controller

- Supports the following devices.
  Asynchronous SRAM, ROM, FLASH, EEPROM
- External 16-bit bus
- Supports burst transfer.
- Wait state is programmable.
- Timing of inserting WE# and OE# is programmable.
- Connectable to a x16-type static memory with common WE# and independent byte-enable without any external logic.

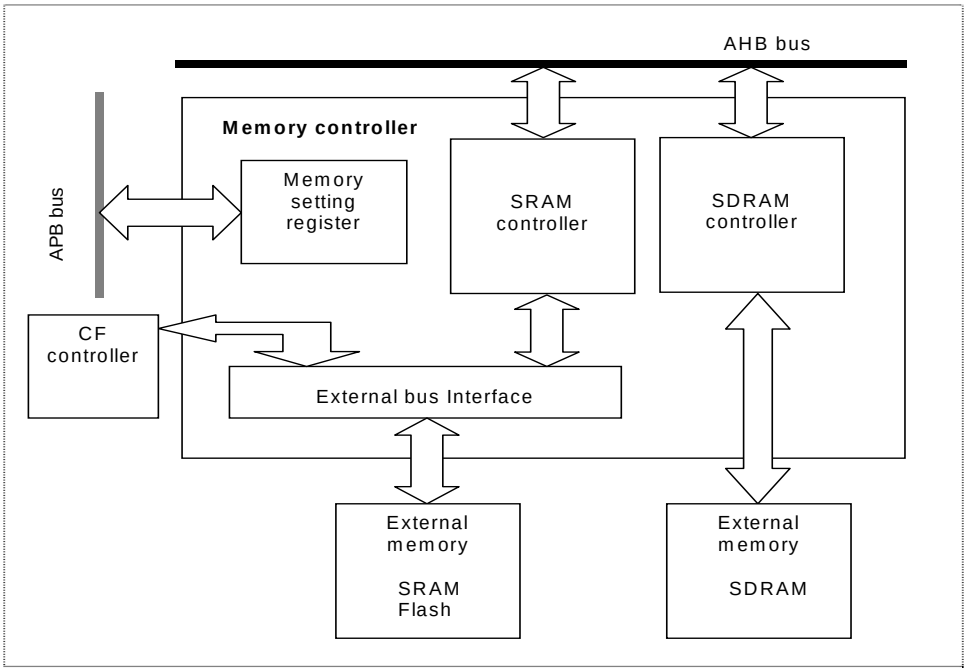
14.1.2 SDRAM Controller

- Supports the following devices.
  SDRAM
- 16/32-bit bus
- Supports burst transfer.
- Auto-precharge ON/OFF is programmable.
- Supports the mode of auto-setting of IN/OUT from/to a device in the self-refresh mode.
- Supports the manual SDRAM initialization mode.

14.1.3 External Bus Interface Module

- This module is required to use the SRAM controller and CF controller modules with common address/data buses. It arbitrates the requirements for the external bus from the SRAM controller and CF controller modules to access external memories. The requirements for the bus are met by handshaking.

14.2 Block Diagram



14.3 External Pins

The IC has the external pins for the memory controller as follows.

Table 14.1 List of external pins for memory controller

Pin name	I/O	Pin functions	Multiplex pin/Remarks
MA[23:20]	O	SRAM address output signal [23:20]	GPIOD[3:0]*
MA [19:12]	o	SRAM address output signal [19:12]	
MA11	O	SRAM address output signal 11	CFREG#**
MA [10:0]	O	SRAM address output signal [10:0]	CFADDR [10:0]**
MD [15:0]	I/O	SRAM data I/O signal [15:0]	MODESEL[15:0]***
MCS [3:2]#	O	Chip select signal for SRAM [3:2]	GPIOD[5:4]*
MCS [1:0]#	O	Chip select signal for SRAM [1:0]	
MOE#	O	Strobe signal for SRAM output	CFOE#**
MWE#	O	Write-enable signal for SRAM	CFWE#**
MBEH#	O	Upper-bites-enable signal for SRAM	
MBEL#	O	Lower-bites-enable signal for SRAM	
MWAIT	I	Wait signal for memory controller	The pin is shared with CFWAIT#.

Note (*):These external pins for the memory controller is multiplexed with GPIO, etc. They become available by setting “Non-GPIO function 1” in the register of the GPIO pin functions.

Note (**):The pin for the memory controller operates as the external pin for the compact flash interface (CF) when CR is in operation.

Note (***):Operates as the MODESEL pin to decide the internal operation mode at the time of power-on reset.

Table 14.2 List of external pins for memory controller

Pin name	I/O	Pin functions	Multiplex pin/Remarks
SDA [14:13]	O	SDRAM bank address output signal [1:0]	
SDA [12:0]	O	SDRAM address output signal [12:0]	
SDD [31:16]	I/O	SDRAM data I/O signal [31:16]	GPIOJ[7:0]*,GPIOK[7:0]*
SDD [15:0]	I/O	SDRAM data I/O signal [15:0]	
SDCS[1:0]#	O	Chip select signal for SDRAM [1:0]	
SDWE#	O	Write-enable signal for SDRAM	
SDCLK	O	Clock output signal for SDRAM	
SDCLKEN	O	Clock-enable output signal for SDRAM	
SDRAS#	O	RAS signal for SDRAM	
SDCAS#	O	CAS signal for SDRAM	
SDDQM3#	I/O	DQM signal for highest-order byte (SDRAM)	GPIOD7*
SDDQM2#	I/O	DQM signal for upper-middle-order byte (SDRAM)	GPIOD6*
SDDQM1#	O	DQM signal for lower-middle-order byte (SDRAM)	
SDDQM0#	O	DQM signal for lowest-order byte (SDRAM)	

Note (*):These external pins for the memory controller is multiplexed with GPIO, etc. They become available by setting “Non-GPIO function 1” in the register of the GPIO pin functions.

14.4 Memory controller

14.4.1 Number of Devices

Up to four SRAM timing devices (device 0 – 3) can be connected to the controller.
Up to eight (512MB at the maximum) SDRAM devices with 8-bit data width can be connected by using two chip-select signal lines.

14.4.2 External Memory Width

The bus width of external memory is switchable between 16 bits for the SRAM timing devices and 16 or 32 bits for the SDRAM devices by making a corresponding setting.

14.4.3 Setting of Device Segments

The SRAM interface has up to 32MB of space allocated for each chip-select signal. 128MB (32MBx4) is available for the SRAM timing space. Whereas the SDRAM interface has up to 256MB of space allocated for each chip-select signal, having up to 512MB assigned to the signals (256MB to each).

14.5 SRAM control

14.5.1 Setting of Timing

The timing is set for each device controller by multiplying a factor to the clock cycle supplied by the memory controller. Set the most appropriate timing values matching the clock cycle.
Adjustable timings include those of reading, writing, and inserting the OE or WE signal.

14.5.2 Write Protect

Setting “1” to the WPROTECT bit of a control register disables writing in the relevant device. Attempting to write in the prohibited area aborts bus access and causes an error notification to CPU; treat such conditions by software properly.

14.6 SDRAM control

14.6.1 Setting of Mode Register

Set first the SDRAM mode register, then issue the LMR instruction to devices with the initialization control register. Use INIT_SD(MEMC [0x80] bit15) normally. Select at the same time the target devices the instruction is issued for. Set the timing for each device controller with number of clocks. Set the most appropriate timing values matching the clock cycle.

14.6.2 Compatibility with Burst

The SDRAM is compatible with burst lengths 1, 2, 4, and 8. Also, the addresses are limited to the increment burst. The SDRAM incurs no access penalty even if a burst length is surpassed in the same ROW. If an interrupt is caused in R/W of a device by an requirement for access to a different device, the SDRAM controller issues the pre-charge or burst-terminate command automatically to pass over the bus promptly.

14.6.3 Setting of Auto-Precharge

Set whether to issue pre-charge after completion of the R/W operation. Selecting “No precharge” keeps the ROW system to be selected, accelerating the speed of R/W of the data in the same page. An active ROW address is precharged at the time of auto-refreshing, then goes idle. Also manual precharge operation is available. See SDRAM Setting Register (MEMC [0x70]).

14.6.4 Reduction of Power Consumption

If a device is in the state left idle after data access (including self-refresh and ROW active states), it can have MCLKEN de-asserted. This is made effective by “1” set to the CKECTRL bit of the SDRAM setting register (MEMC [0x70]).

If the CKECTRL bit has “1” set, it is possible to stop the clock itself supplied to the memory for the purpose of further reduction of power consumption. This is made effective by “1” set to the CLKCTRL bit.

If the accumulated number of auto-refreshing requirements surpasses the value set, or R/W is requested of a device, CKE is asserted as “HIGH” and data transfer is resumed even if the CKECTRL bit has “1”.

14.6.5 Stopping Memory Clock

Stopping the supply of clocks to SDRAM can lead to further reduction of power consumption other than keeping MCLKEN at “LOW” with CKECTRL. This mode is made effective by setting the CLKCTRL bit to “1”. Be sure to make the CKECTRL bit effective for making the CLKCTRL bit effective. The memory controller stops the memory clock MCLK if all the SDRAMs connected are in one of the following states:

memory not initialized yet, or chip idle in the self-refresh mode or with ROW inactive.

Any request for R/W or auto-refreshing of memory causes supply of clocks to be resumed even if in the state where the memory clock is stopped.

As an exceptional case, setting “1” to the CLKFORCE bit in the SDRAM detailed settings register (MEMC [0x74]) causes the output from MCLK to be continued regardless of the state of the SDRAM memory.

14.6.6 Support of Energy-Saving Mode

When the system enters the energy-saving mode, the supply of clocks to the memory controller itself may be stopped. An SDRAM requires to be auto-refreshed or self-refreshed to hold the data in it. Stop of supply of clocks to the memory controller necessitates that all the SDRAMs be entered into the self-refresh mode.

First enter all the SDRAMs into the self-refresh mode with the software, then set the system controller to the energy-saving mode.

14.6.7 Control of Auto-Refresh

Auto-refreshing is performed when at least one SDRAM has been initialized and is in the IDLE state (but not self-refreshing). Its cycle is equal to the number of HCLK set in the SDRAM refresh timer register. The memory controller also refreshes the SDRAM connected at the same time; so set the refreshing cycle in accordance with the device which requires the most frequent refreshing.

The refreshing is performed in a distributed way basically; but plural requirements for refreshing can be accumulated to prevent data transfer from being severed by auto-refreshing. Specify the number of requirements accumulable by setting AREFWAIT bits [3:0] in the SDRAM detailed setting register (MEMC [0x74]). If the refreshing requirements accumulated surpass the specified count, The SDRAM controller recognizes the requirements, starting auto-refreshing next time the memory controller goes idle.

If the device is accessing a bus at the timing its refreshing is requested, the access to the bus takes priority, and the auto-refreshing is performed upon clearance of the bus request.

14.6.8 Control of Self-Refresh

The memory controller supports the SDRAM's self-refreshing. To enter a device into the self-refresh mode, set "1" to the SELF bit for the relevant device in the SDRAM detailed setting register (MEMC [0x74]). To exit from the self-refresh mode, set "0" instead.

If a SDRAM in the self-refresh mode is accessed (for Read/Write), it exit from the mode automatically to execute the command required. After that, the device goes idle basically; it can be entered again into the self-refresh mode, though, by making an appropriate setting. To do so, set "1" to the RESELF bit in the SDRAM detailed setting register (MEMC [0x74]). The SDRAM enters the self-refresh mode automatically if the count of accesses to it as set in the SREFCNT bits is exhausted.

14.6.9 Status Register

Shows the statuses of the SDRAM controller and the devices connected.

14.7 Registers

14.7.1 List of Registers

Shown below is the list of the memory controller registers. The base address of these registers is 0xFFFF_A000.

Table 14.2 List of registers (Base address: 0xFFFF_A000)

Address offset	Register name	Register name's abbreviation	Default Value	R/W	Data access size
SRAM controller registers					
0x20	SRAM device 0 timing register	RAMTMG0	0x0000_1C7F	R/W	32
0x24	SRAM device 0 control register	RAMCNTL0	0x0000_0001	R/W	32
0x30	SRAM device 1 timing register	RAMTMG1	0x0000_1C7F	R/W	32
0x34	SRAM device 1 control register	RAMCNTL1	0x0000_0001	R/W	32
0x40	SRAM device 2 timing register	RAMTMG2	0x0000_1C7F	R/W	32
0x44	SRAM device 2 control register	RAMCNTL2	0x0000_0001	R/W	32
0x50	SRAM device 3 timing register	RAMTMG3	0x0000_1C7F	R/W	32
0x54	SRAM device 3 control register	RAMCNTL3	0x0000_0001	R/W	32
SDRAM controller registers					
0x60	SDRAM mode register	SDMR	0x0000_0032	R/W	16/32
0x64	Reserved	—	—	—/—	—
0x68	Reserved	—	—	—/—	—
0x70	SDRAM setting register	SDCNFG	0x0600_C700	R/W	32
0x74	SDRAM detailed setting register	SDADVCNFG	0x000F_0300	R/W	32
0x80	Initialization control register	SDINIT	0x0000_0000	R/W	16/32
0x90	SDRAM refresh timer register	SDREF	0x0000_00A0	R/W	16/32
0xA0	SDRAM status register	SDSTAT	0x0000_0202	RO	32

14.7.2 Detailed Description of Registers

SRAM device [3:0] Timing register (RAMTMG[3:0])

MEMC[0x20, 0x30, 0x40, 0x50] Default value = 0x0000_1C7F Read/Write

RSV (0)	WAITWE [4:0]					RSV (0)	WAITOE [4:0]					RSV (0)			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)	WAITWR [4:0]					RSV (0)	WAITRD [4:0]					RSV (0)			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bits [30:26]: **WAITWE [4:0] Write-enable signal insertion delay control**
Set the timing to insert the WE# signal. (Default = 0x0)

Bit 25: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bits [24:20]: **WAITOE [4:0] Output-enable signal insertion delay control**
Set the timing to insert the OE# signal. (Default = 0x0)

Bits [19:15]: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bits [14:10]: **WAITWR [4:0] Write cycle wait control**
Set the number of writing wait cycle for an SRAM/ROM device. (Default = 0x07)

Bit 9: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bits [8:4]: **WAITRD [4:0] Read cycle wait control**
Set the number of reading wait cycle for to an SRAM/ROM device. (Default = 0x07)

Bits [3:0]: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

SRAM device [3:0] Control register (RAMCNTL[3:0])

MEMC[0x24, 0x34, 0x44, 0x54] Default value = 0x0000_0001 Read/Write

RSV (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)												WPROTECT	MWAITPOL[1:0]		RBLE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:4]: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bit 3: **WPROTECT Write protect**
No write can be performed in a device having the write-protect bit set.
0: Not write-protected (default)
1: Write-protected

Bits [2:1]: **MWAITPOL [1:0]**

Set the polarity for the MWAIT signal.

00: Disabled (default)
01: Enabled if LOW-active
10: Enabled if HIGH-active
11: Reserved

Bit 0: **RBLE Byte lane control setting**

To implement the byte control on writing with a device lacking in byte-lane control, set this bit to "0" and connect DQM of this device to WE# of the target device.

0: Reading makes DQM [1:0] HIGH, preventing writing.
1: Reading makes DQM [1:0] LOW, allowing reading of all the byte lanes. (Default)

SDRAM mode register (SDMR)

MEMC[0x60] Default value = 0x0000_0032 Read/Write

RSV (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)						WBM	OP Mode [1:0]		CL [2:0]			BT	BL [2:0]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:10]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bit 9: **WBM Write burst mode**

0: Written with the burst length specified (default)
1: Single location access

Bits [8:7]: **OP Mode [1:0] Operation mode**

00: Normal operation
xx: All the other combinations of bits are reserved.

Bits [6:4]: **CL [2:0] CAS latency**

000: Reserved
001: CL=1
010: CL=2
011: CL=3 (default)
1xx: Reserved

Bit 3: **BT Burst type**

0: Sequential (default)
1: Reserved

Bits [2:0]: **BL [2:0] Burst length**

000: BL=1
001: BL=2
010: BL=4 (default)
011: BL=8
100: Reserved
101: Reserved
110: Reserved
111: Reserved

Reserved register

MEMC[0x64, 0x68] Default value = 0x xxxx_xxxx - / -

RSV															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved register; don't access it.

SDRAM setting register (SDCNFG)

MEMC[0x70] Default value = 0x0600_C700 Read/Write

RSV (0)					CLK-CTRL	CKE-CTRL	XBW (W)	COLW [7:4]				RSV (0)			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)					TRCD [1:0]		APCG	RSV(0)		RSV (0)		BNUM [1:0]		RSV (0)	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:27]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bit 26: **CLKCTRL MCLK control**

This bit can be set to "1" only if the dynamic MCLKEN control (CKECTRL) is enabled.

0: MCLK is output continuously. Initialization of SDRAM is required, though.

1: MCLK is stopped if SDRAM is idle (including the self-refreshing state). (Default)

Note: To change the value of this bit (CLKCTRL) after initializing SDRAM (MEMC [0x80] bit15:INIT_SD = 1), first enter the SDRAM into the self-refresh mode. That is, perform: self-refresh mode IN, change of CLKCTRL, then self-refresh mode OUT.

Bit 25: **CKECTRL Dynamic MCLKEN control**

0: MCLKEN=H is output continuously (except in the self-refreshing state).

1: MCLK is put into LOW if SDRAM is idle (including the self-refreshing state and the bank active state getting no access). (Default)

Note: To change the value of this bit (CKECTRL) after initializing SDRAM (MEMC [0x80] bit15:INIT_SD = 1), first enter the SDRAM into the self-refresh mode. That is, perform: self-refresh mode IN, change of CKECTRL, then self-refresh mode OUT.

Bit 24: **XBW External bus width (for write only)**

Select the external bus width.

0: External bus width =16 bits (default)

1: External bus width =32 bits

Bits [23:20]: **COLW [7:4] Column address width**

Set the column address width for SDRAM.

COLW [5:4]: For setting for SDRAM device 0.

COLW [7:6]: For setting for SDRAM device 1.

00: Column address A0 - A7 (default)

01: Column address A0 - A8

10: Column address A0 - A9

11: Column address A0 - A9, A11

Bits [19:16]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [15:11]: **RSV Reserved**
This bit is reserved. Be sure to set "0", though the bit defaults to "0b11000".

Bits [10:9]: **TRCD [1:0] RAS-CAS delay**
Set the delay time from MRAS# to MCAS# (in number of cycles).
00: Reserved
01: 1 cycle
10: 2 cycles
11: 3 cycles (default)

Bit 8: **APCG Auto-precharge control**
Make setting for auto-precharge.
0: No auto-precharge (All the banks are precharged at the timing auto-refresh is performed.)
1: Auto-precharged (default)

Bits [7:6]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [5:4]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [3:2]: **BNUM [1:0] Number of banks**
Set the number of banks in the SDRAM connected.
BNUM0: For setting for SDRAM device 0
BNUM1: For setting for SDRAM device 1
0: 4-bank device (default)
1: Reserved (for 2-bank device)

Bits [1:0]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

SDRAM detailed setting register (SDADVCNFG)
MEMC[0x74] Default value = 0x000F_0300 Read/Write

RSV (0)												SREFCNT [3:0]			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				AREFWAIT [3:0]				CLK-FORCE	RESELF	RSV (0)		SELF [1:0]		RSV (0)	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:20]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [19:16]: **SREFCNT [3:0] Number of cycles before re-entering self-refresh mode**
Set the number of cycles before the self-refresh mode is re-entered after SDRAM lost accesses. The setting is effective only when "1" is set to the RESELF bit (MEMC [0x74] bit 6). (Default = 0x07)

Bits [15:12]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [11:8]: **AREFWAIT [3:0] Count of auto-refresh held**

If the memory controller interface is occupied at the timing of distributed refreshing performed, the requirements for refreshing are held, and are met after the occupancy is released. (Default = 0x03)

Bit 7: **CLKFORCE**

0: MCLK clock is output subject to the state of the memory controller. (Default)
1: MCLK clock is output regardless of the state of the memory controller.

Bit 6: **RESELF Self-refresh re-entry mode**

Controls whether to re-enter the SDRAM into the self-refresh mode after the SDRAM in the self-refresh mode is activated for R/W.

0: The SDRAM is not re-entered in the self-refresh mode. (Default)
1: The SDRAM is re-entered in the self-refresh mode if it gets no access for SREFCNT [3:0] cycles.

Bits [5:4]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bits [3:2]: **SELF [1:0] Self-refresh mode ON/OFF**

Cleared to zero automatically if the SDRAM in the self-refresh mode is activated for R/W (except the RESELF mode is set).
SELF0: For setting for SDRAM device 0
SELF1: For setting for SDRAM device 1

0: Exits from the self-refresh mode. (Default)
1: The SDRAM is set to the self-refresh mode.

Bits [1:0]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Initialization control register (SDINIT)
MEMC[0x80] Default value = 0x0000_0000 Read/Write

RSV (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INIT_SD	RSV (0)							LMR	AREF	PCG-ALL	RSV (0)	DEVSEL [1:0]		RSV (0)	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Note: Don't set plural instructions in this register at the same time.
Use normally INIT_SD to initialize the SDRAM.
Use other instructions if manual initialization is required.
Select the target devices with DEVSEL[1:0] (MEMC[0x80] bits[3:2]) for each instruction.

Bits [31:16]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bit 15: **INIT_SD Initialize SDRAM**

Performs the sequence for initializing the SDRAM automatically. Select the target devices with DEVSEL (bits [3:0] in this register).

0: No action (default)
1: SDRAM initialized. (Returns to "0" automatically upon completion of initialization.)

Bits [14:8]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bit 7: **LMR Mode register set**

Issues the mode register set instruction to SDRAM.

0: No action (default)

1: Issues the mode register set instruction. (Returns to "0" automatically upon issuance.)

Bit 6: **AREF Auto-refresh**

Issues the auto-refresh instruction to SDRAM.

0: No action (default)

1: Issues the auto-refresh instruction. (Returns to "0" automatically upon issuance.)

Bit 5: **PCGALL Precharge all**

Issues the all-bank precharge instruction to SDRAM.

0: No action (default)

1: Issues the all-bank precharge instruction. (Returns to "0" automatically upon issuance.)

Bit 4: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bits [3:2]: **DEVSEL [1:0] Select device**

Selects the device an instruction is issued to.

DEVSEL0: For setting for SDRAM device 0

DEVSEL1: For setting for SDRAM device 1

0: No selection (default)

1: Selected

Bits [1:0]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

SDRAM refresh timer register (SDREF)

MEMC[0x90] Default value = 0x0000_00A0 Read/Write

RSV (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)				REFTIME [11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:12]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bits [11:0]: **REFTIME [11:0] Refreshing interval**

Set the interval for issuing distributed refresh.

Set the number of the HCLK cycles compatible with the interval of distributed auto-refreshing.

For example, if the refreshing interval is 16μsec and the frequency of the system clock HCLK is 10MHz, then set:

16μs × 10MHz = 160 cycles (=1010_0000b).

SDRAM status register (SDSTAT)

MEMC[0xA0] Default value = 0x0000_0202 Read Only

RSV															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV				DEVST1 [3:0]				RSV				DEVST0 [3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:12]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bits [11:8]: **DEVST1 [3:0] Status of SDRAM device 1**

Shows the current state of SDRAM device 1.

DEVST1 [0] 1: Device available (enabled and initial settings completed)
0: Device unavailable

DEVST1 [1] 1: Device idle or suspended
0: Device in other states

DEVST1 [2] 1: ROW of either bank of the device is activated.
0: ROW inactive

DEVST1 [3] 1: Device in the self-refresh mode
0: Device not in the self-refresh mode

Bits [7:4]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bits [3:0]: **DEVST0 [3:0] Status of SDRAM device 0**

Shows the current state of SDRAM device 0.

DEVST0 [0] 1: Device available (enabled and initial settings completed)
0: Device unavailable

DEVST0 [1] 1: Device idle or suspended
0: Device in other states

DEVST0 [2] 1: ROW of either bank of the device is activated.
0: ROW inactive

DEVST0 [3] 1: Device in the self-refresh mode
0: Device not in the self-refresh mode

15. INTERRUPT CONTROLLER (INT)

15.1 Description

This interrupt controller (INTC) can handle two fast interrupt requests (FIQs) and 32 normal interrupt requests (IRQs). Table 15.1 shows the correspondence of peripheral circuits/functions to their respective interrupt request inputs. After reset, all interrupt requests are active HIGH level trigger inputs. Although INT [7:0] is set to the low level trigger setting right after reset, this interrupt controller has made it possible to select the polarity and level/edge trigger for the input signal if necessary.

The interrupt controller processes FIQ and IRQ inputs and then outputs two interrupt request signals to the ARM720T core: nFIQ and nIRQ.

For hardware reason, no interrupt priority is configured.

Table 15.1 List of Interrupt Sources

Interrupt Type	Interrupt Level	Interrupt Source	Description
Fast Interrupt Request FIQ	FIQ0	Watchdog Timer	
	FIQ1	GPIOB0 Pin (***)	Ex. Battery Low
Normal Interrupt Request IRQ	IRQ0	Watchdog Timer	
	IRQ1	Interrupt Controller	Software request by the register
	IRQ2	ARM720T COMMRx	Debug Communication Port
	IRQ3	ARM720T COMMTx	Debug Communication Port
	IRQ4	Timer A	16-bit Timer Channel 0
	IRQ5	Timer A	16-bit Timer Channel 1
	IRQ6	Timer A	16-bit Timer Channel 2
	IRQ7	Reserved	
	IRQ8	JPEG1 Control	
	IRQ9	DMAC1	DMAC on AHB1
	IRQ10	JPEG1 DMAC	
	IRQ11	Camera 1 Interface	
	IRQ12	ARS	
	IRQ13	DMAC2	DMA INT (DMAC for JPEG on AHB2)
	IRQ14 (*)	GPIOA or GPIOB	Selects an interrupt input pin between GPIOA and GPIOB
	IRQ15	SPI	SPI TXRDY/RXRDY
	IRQ16	I^2C	Transfer Complete
	IRQ17	UART1	UART TXRDY/RXRDY
	IRQ18	RTC	Alarm or Timer tick
	IRQ19	CF card Interface	
	IRQ20 (**)	INTGB	GPIOB7 to GPIOB0 direct input representative bit
	IRQ21	Timer B	
	IRQ22	Reserved	
	IRQ23	JPEG2 Control	
	IRQ24	JPEG2 DMAC	
	IRQ25	Camera 2 Interface	
	IRQ26	UART2	
	IRQ27	UART3	
	IRQ28	SD Memory	
	IRQ29	USB DEV	
	IRQ30	ADC	
	IRQ31	I^2S	

Note (*): Sets an interrupt request based on GPIOA [7:0] or GPIOB [7:0]. For details, see the detailed description of registers from GPIO [0x60] to GPIO [0x6C] in “26. GPIO Detailled Description of Registers”.

Note (**): Sets an interrupt request based on GPIOB [7:0]. For details, see the detailed description of IRQxx registers.

Note (***): Direct input from the GPIOB0 pin. (The default is active-low level interrupt request. However, the interrupt level on the inside becomes active-high level because it reverses internally.)

Table 15.2 List of Interrupt Sources

Interrupt Type	Interrupt Level	Interrupt Source	Description
GPIOB Interrupt Request IRQxx	IRQGB0(*)	INT0	GBIOB0 direct input
	IRQGB1(*)	INT1	GBIOB1 direct input
	IRQGB2(*)	INT2	GBIOB2 direct input
	IRQGB3(*)	INT3	GBIOB3 direct input
	IRQGB4(*)	INT4	GBIOB4 direct input
	IRQGB5(*)	INT5	GBIOB5 direct input
	IRQGB6(*)	INT6	GBIOB6 direct input
	IRQGB7(*)	INT7	GBIOB7 direct input

Note (*): Direct input from the GPIOB [7:0] pin. (The default is active-low level interrupt request. However, the interrupt level on the inside becomes active-high level because it reverses internally.) These interrupt settings (enable, polarity, level, etc.) can be changed only by the control register of the interrupt controller. Unlike IRQ14 (*), they cannot be changed by the GPIO[0x60] ~ GPIO[0x6C].

15.2 Block Diagram

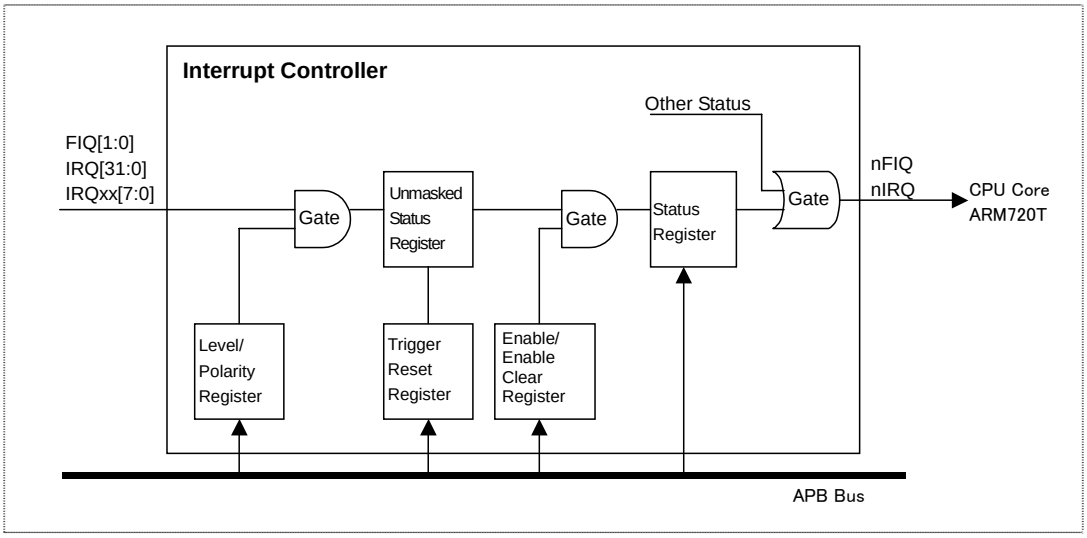


Fig.15.1 Interrupt Controller Block Diagram

15.3 FIQ

As an interrupt output, a watchdog timer and GPIO port B [0] are assigned to FIQ0 and FIQ1, respectively. These two FIQ signals are used to generate nFIQ to be input to ARM720T.

15.4 IRQ

Interrupt requests from the internal devices are assigned to the IRQs. All IRQ interrupt requests are ORed to generate nIRQ signal to be input to ARM720T. The interrupt controller has a total of 32 interrupt request inputs, IRQ31-IRQ0, each of which corresponds to a bit of the interrupt control register with a ratio of 1:1. For example, IRQ0 is assigned to Bit 0, while IRQ1 is assigned to Bit 1. For the correspondence, see Table 15.1.

For the external interrupt, INT0-INT7, interrupt requests are received at IRQGB0-IRQGB7 and then assigned as an external representative interrupt to IRQ20.

15.5 External Pins

The following lists the external pins related to the interrupt controller (INT).

Pin Name	Input/Output	Pin Functions	Multiplex Pin*/Remarks
FIQ1	I	Fast Interrupt Pin 1	GPIOB0/I^{2}S_WS
INT0	I	External Interrupt Pin 0	GPIOB0/I^{2}S_WS
INT1	I	External Interrupt Pin 1	GPIOB1/I^{2}S_SCK
INT2	I	External Interrupt Pin 2	GPIOB2/I^{2}S_SDO
INT3	I	External Interrupt Pin 3	GPIOB3/I^{2}S_SDI
INT4	I	External Interrupt Pin 4	GPIOB4/TimerA0out
INT5	I	External Interrupt Pin 5	GPIOB5/TimerA1out
INT6	I	External Interrupt Pin 6	GPIOB6/TimerA2out
INT7	I	External Interrupt Pin 7	GPIOB7/TimerBIn

Note (*): The INT-related external interrupt pins use the GPIO pin inputs. If you use a GPIO pin as an interrupt input, set the internal register for interrupt control. To use for other pin functions, set the value to “Function 1 or 2 of other than GPIO” by using the GPIO pin function register.

15.6 Registers

15.6.1 List of Registers

The base address where the control register of the interrupt controller is located is 0xFFFF_F000.

Table 15.2 List of Registers (Base Address: 0xFFFF_F000)

Address Offset	Register Name	Default Value	R/W	Data Access Size
0x000	IRQ Status Register	0x0000_0000	RO	32
0x004	IRQ Unmasked Status Register	0x0000_0000 (*)	RO	32
0x008	IRQ Enable Register	0x0000_0000	R/W	32
0x00C	IRQ Enable Clear Register	0x0000_0000	WO	32
0x010	Software IRQ Register	0x0000_0000	WO	32
0x020	IRQxx Status Register	0x0000_0000	RO	32
0x024	IRQxx Unmasked Status Register	0x0000_0000 (*)	RO	32
0x028	IRQxx Enable Register	0x0000_0000	R/W	32
0x02C	IRQxx Enable Clear Register	0x0000_0000	WO	32
0x080	IRQ Level Register	0x0000_0000	R/W	32
0x084	IRQ Polarity Register	0xFFFF_FFFF	R/W	32
0x088	IRQ Trigger Reset Register	0x0000_0000	WO	32
0x0A0	IRQxx Level Register	0x0000_0000	R/W	32
0x0A4	IRQxx Polarity Register	0x0000_0FFF	R/W	32
0x0A8	IRQxx Trigger Reset Register	0x0000_0000	WO	32
0x100	FIQ Status Register	0x0000_0000	RO	32
0x104	FIQ Unmasked Status Register	0x0000_0000 (*)	RO	32
0x108	FIQ Enable Register	0x0000_0000	R/W	32
0x10C	FIQ Enable Clear Register	0x0000_0000	WO	32
0x180	FIQ Level Register	0x0000_0000	R/W	32
0x184	FIQ Polarity Register	0x0000_0003	R/W	32
0x188	FIQ Trigger Reset Register	0x0000_0000	WO	32

Note (*): Default value of IRQ/IRQxx/FIQ Unmask Status Registers are different according to the system configuration.

15.6.2 Detailed Description of Registers

If not otherwise specified, the default value of any register bit that is not reserved is “0”.

IRQ Status Register

INT[0x000] Default value = 0x0000_0000 Read Only

IRQ[31:16] Status															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ[15:0] Status															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQ[31:0] Status**

Indicates the (masked) generation status of an interrupt request from each unit.

0 (r): No interrupt request exists

1 (r) : An interrupt request exists

These status bits indicate the status of the IRQ for which interrupt is enabled by the IRQ enable register (INT[0x008]). If an interrupt request from an interrupt disabled unit is generated, the bit will not be “1”. The interrupt request that is set to “1” is sent to the CPU. These bits return to “0” by clearing the corresponding bits of the IRQ unmasked status register.

Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

IRQ Unmasked Status Register

INT[0x004] Default value = 0x0000_0000 Read Only

IRQ[31:16] Unmasked Status															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ[15:0] Unmasked Status															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQ[31:0] Unmasked Status**

Indicates the (unmasked) generation status of an interrupt cause of each unit.

0 (r) : No interrupt cause exists

1 (r) : An interrupt cause exists

These status bits indicate the status of the IRQ where the interrupt has not been masked by the IRQ enable register (INT[0x008]). If an interrupt cause from an interrupt disabled unit is generated, the bit will also be “1”. For a level trigger interrupt, these bits return to “0” by clearing the interrupt flag of each unit. For an edge trigger interrupt, they return to “0” by writing “1” to the corresponding bits of the IRQ trigger reset register (INT[0x088])

Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

IRQ Enable Register

INT[0x008] Default value = 0x0000_0000 Read/Write

IRQ[31:16] Enable															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ[15:0] Enable															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQ[31:0] Enable**

Enables interrupt request (IRQ) input from each unit.

0 (r): Interrupt request is disabled
1 (r): Interrupt request is enabled
0 (w): Invalid
1 (w): Enables interrupt request input

By reading this register, you can determine for each interrupt request (IRQ) input whether it is currently enabled (bit is set to "1") or disabled (bit is set to "0").

When performing a write operation to the register, writing "1" enables interrupt request (IRQ) input. The interrupt controller receives the IRQ input corresponding to the bit and then outputs an interrupt request to the CPU. When "0" is written to the register, it is ignored and accessing the register does not disable interrupt request (IRQ) input. To clear the bits, the IRQ enable clear register (INT[0x00C]) is used.

When reset, all interrupts are set to disabled.

Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

IRQ Enable Clear Register

INT[0x00C] Default value = 0x0000_0000 Write Only

IRQ[31:16] Enable Clear															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ[15:0] Enable Clear															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQ[31:0] Enable Clear**

Disables (masks) the interrupt request (IRQ) input from each unit.

0 (w): Invalid
1 (w): Disables interrupt request input

Writing "1" to these bits clears the corresponding IRQ enable bits of the IRQ enable register (INT[0x008]) and the interrupt request (IRQ) input is disabled. Writing "0" is ignored.

Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

Software IRQ Register

INT[0x010] Default value = 0x0000_0000 Write Only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														Software IRQ	RSV

Bit 1: **Software IRQ**

Controls an IRQ1 software interrupt.

0 (w): Clears a software interrupt request

1 (w): Generates a software interrupt request

The status of the software interrupt cause that is currently set can be read from bit 1 of the IRQ unmasked status register (INT[0x004]).

Bit 0: **RSV Reserved**

IRQxx Status Register

INT[0x020] Default value = 0x0000_0000 Read Only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQxx[31:16] Status															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQxx[15:0] Status															

Bits [31:0]: **IRQxx[31:0] Status**

IRQxx[31:12] : n/a

IRQxx[11:8] : Reserved

IRQxx[7:0] : IRQxx (IRQGB[7:0]) status

Indicates the (masked) generation status of an interrupt request from each unit.

0 (r) : No interrupt request exists

1 (r) : An interrupt request exists

These status bits indicate the status of the IRQxx for which interrupt is enabled by the IRQxx enable register (INT[0x028]). If an interrupt request from an interrupt disabled unit is generated, the bit will not be "1". The interrupt request that is set to "1" is sent to the CPU. These bits return to "0" by clearing the corresponding bits of the IRQ unmasked status register.

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQxx Unmasked Status Register

INT[0x024] Default value = 0x0000_0000 Read Only

IRQxx[31:16] Unmasked Status															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQxx[15:0] Unmasked Status															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQxx[31:0] Unmasked Status**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Unmasked Status

Indicates the (unmasked) generation status of an interrupt cause of each unit.

0 (r): No interrupt cause exists
1 (r): An interrupt cause exists

These status bits indicate the status of the IRQxx where the interrupt has not been masked by the IRQxx enable register (INT[0x028]). If an interrupt cause from an interrupt disabled unit is generated, the bit will also be "1". For a level trigger interrupt, these bits return to "0" by clearing the interrupt flag of each unit. For an edge trigger interrupt, they return to "0" by writing "1" to the corresponding bits of the IRQxx trigger reset register (INT[0x0A8])

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQxx Enable Register

INT[0x028] Default value = 0x0000_0000 Read/Write

IRQxx[31:16] Enable															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQxx[15:0] Enable															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQxx[31:0] Enable**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Enable Register

Enables the interrupt request (IRQ) input from each unit.

0 (r): Interrupt request is disabled
1 (r): Interrupt request is enabled
0 (w): Ignored
1 (w): Enables interrupt request input

By reading this register, you can determine for each interrupt request (IRQxx) input whether it is currently enabled (bit is set to "1") or disabled (bit is set to "0").

When performing a write operation to the register, writing "1" enables interrupt request (IRQxx) input. The interrupt controller receives the IRQxx input corresponding to the bit and then outputs an interrupt request to the CPU. When "0" is written to the register, it is ignored and accessing the register does not disable the interrupt request (IRQxx) input. To clear the bits, the IRQxx enable clear register (INT[0x02C]) is used.

When reset, all interrupts are set to disabled.

Bits [7:0] correspond to IRQ [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQxx Enable Clear Register

INT[0x02C] Default value = 0x0000_0000 Write Only

IRQxx[31:16] Enable Clear															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQxx[15:0] Enable Clear															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQxx[31:0] Enable Clear**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Enable Clear

Disables (masks) the interrupt request (IRQxx) input from each unit.

0 (w): Ignored
1 (w): Disables interrupt request input

Writing "1" to these bits clears the corresponding IRQxx enable bits of the IRQxx enable register (INT[0x028]) and the interrupt request (IRQxx) input is disabled. Writing "0" is ignored.

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQ Level Register

INT[0x080] Default value = 0x0000_0000 Read/Write

IRQ[31:16] Level															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ[15:0] Level															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQ[31:0] Level**

Sets a trigger mode for each IRQ input.

0 (r/w): Level trigger mode
1 (r/w): Edge trigger mode

This setting determines whether interrupt request (IRQ) signal is sampled according to the level or falling/rising edge of the signal. The polarity of a signal (LOW level/rising edge or HIGH level/falling edge) is set by the IRQ polarity register (INT[0x084]). Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

Note: Normally, use the value of this register that is set at reset without changing it.

IRQ Polarity Register

INT[0x084] Default value = 0xFFFF_FFFF Read/Write

IRQ[31:16] Polarity															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ[15:0] Polarity															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQ[31:0] Polarity**

Sets the polarity for each IRQ signal.

0 (r/w): LOW level/falling edge

1 (r/w): HIGH level/rising edge

An interrupt request (IRQ) signal is sampled according to this setting. Level/edge selection is done by the IRQ level register (INT[0x080]). Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

Note: Normally, use the value of this register that is set at reset without changing it.
Please set bit28 to 0 when you use the interrupt of the SD card detection for the factor.

IRQ Trigger Reset Register

INT[0x088] Default value = 0x0000_0000 Write Only

IRQ[31:16] Trigger Reset															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ[15:0] Trigger Reset															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQ[31:0] Trigger Reset**

Clears the unmasked status of the interrupt that is set to edge trigger mode.

0 (w): Invalid

1 (w): Clears the interrupt status

Writing “1” clears the corresponding status bits in the IRQ unmasked status register (INT[0x004]) that are set to edge trigger mode. Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

IRQxx Level Register

INT[0x0A0] Default value = 0x0000_0000 Read/Write

IRQxx[31:16] Level															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQxx[15:0] Level															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQxx[31:0] Level**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Level

Sets a trigger mode for each IRQxx input.
0 (r/w): Level trigger mode
1 (r/w): Edge trigger mode

This setting determines whether interrupt request (IRQxx) signal is sampled according to the level or falling/rising edge of the signal. The polarity of a signal (LOW level/rising edge or HIGH level/falling edge) is set by the IRQxx polarity register (INT[0x0A4]).

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQxx Polarity Register

INT[0x0A4] Default value = 0x0000_0FFF Read/Write

IRQxx[31:16] Polarity															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQxx[15:0] Polarity															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQxx[31:0] Polarity**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Polarity

Sets the polarity for each IRQxx signal.
0 (r/w): LOW level/falling edge
1 (r/w): HIGH level/rising edge

An interrupt request (IRQxx) signal is sampled according to this setting. Level/edge selection is done by the IRQxx level register (INT[0x0A0]).

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQxx Trigger Reset Register

INT[0x0A8] Default value = 0x0000_0000 Write Only

IRQxx[31:16] Trigger Reset															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQxx[15:0] Trigger Reset															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **IRQxx[31:0] Trigger Reset**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Trigger Reset

Clears the unmasked status of the interrupt that is set to edge trigger mode.

0 (w): Ignored
1 (w): Clears the interrupt status

Writing "1" clears the corresponding status bits in the IRQxx unmasked status register (INT[0x024]) that are set to edge trigger mode.

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

FIQ Status Register

INT[0x100] Default value = 0x0000_0000 Read Only

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a														FIQ [1:0] Status	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [1:0]: **FIQ [1:0] Status**

Indicates the (masked) generation status of an FIQ interrupt request.

0 (r) : No interrupt request exists
1 (r) : An interrupt request exists

These status bits indicate the status of the FIQ for which interrupt is enabled by the FIQ enable register (INT[0x108]). If an interrupt request from an interrupt disabled unit is generated, the bit will not be "1". The interrupt request that is set to "1" is sent to the CPU. These bits return to "0" by clearing the corresponding bits of the FIQ unmasked status register.

Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Unmasked Status Register

INT[0x104] Default value = 0x0000_0000 Read Only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														FIQ[1:0] Unmasked Status	

Bits [1:0]: **FIQ[1:0] Unmasked Status**

Indicates the (unmasked) generation status of an FIQ interrupt cause.

0 (r) : No interrupt cause exists

1 (r) : An interrupt cause exists

These status bits indicate the status of the FIQ where the interrupt has not been masked by the FIQ enable register (INT[0x108]). If an interrupt cause from an interrupt disabled unit is generated, the bit will also be "1". For a level trigger interrupt, these bits return to "0" by clearing the interrupt flag of each unit. For an edge trigger interrupt, they return to "0" by writing "1" to the corresponding bits of the FIQ trigger reset register (INT[0x188]).

Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Enable Register

INT[0x108] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														FIQ [1:0] Enable	

Bits [1:0]: **FIQ [1:0] Enable**

Enables the fast interrupt request (FIQ) input.

0 (r) : Interrupt request is disabled

1 (r) : Interrupt request is enabled

0 (w) : Invalid

1 (w): Enables interrupt request input

By reading this register, you can determine for each interrupt request (FIQ) input whether it is currently enabled (bit is set to "1") or disabled (bit is set to "0").

When performing a write operation to the register, writing "1" enables interrupt request (FIQ) input. The interrupt controller receives the FIQ input corresponding to the bit and then outputs an interrupt request to the CPU. When "0" is written to the register, it is ignored and accessing the register does not disable the interrupt request (FIQ) input. To clear the bits, the FIQ enable clear register (INT[0x10C]) is used.

When reset, all interrupts are set to disabled.

Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Enable Clear Register

INT[0x10C] Default value = 0x0000_0000 Write Only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														FIQ [1:0] Enable Clear	

Bits [1:0]: **FIQ [1:0] Enable Clear**

Disables (masks) the fast interrupt request (FIQ) input.

0 (w) : Invalid

1 (w) : Disables interrupt request input

Writing "1" to these bits clears the corresponding FIQ enable bits of the FIQ enable register (INT[0x108]) and the interrupt request (FIQ) input is disabled. Writing "0" is ignored. Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Level Register

INT[0x180] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														FIQ [1:0] Level	

Bits [1:0]: **FIQ [1:0] Level**

Sets a trigger mode for each FIQ input.

0 (r/w): Level trigger mode

1 (r/w): Edge trigger mode

This setting determines whether interrupt request (FIQ) signal is sampled according to the level or falling/rising edge of the signal. The polarity of a signal (LOW level/rising edge or HIGH level/falling edge) is set by the FIQ polarity register (INT[0x184]). Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Polarity Register

INT[0x184] Default value = 0x0000_0003 Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														FIQ [1:0] Polarity	

Bits [1:0]: **FIQ [1:0] Polarity**

Sets the polarity for each FIQ signal.

0 (r/w): LOW level/falling edge

1 (r/w): HIGH level/rising edge

An interrupt request (FIQ) signal is sampled according to this setting. Level/edge selection is done by the FIQ level register (INT[0x180]). Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Trigger Reset Register

INT[0x188] Default value = 0x0000_0000 Write Only

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a														FIQ [1:0] Trigger Reset	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [1:0]: **FIQ [1:0] Trigger Reset**

Clears the unmasked status of the interrupt that is set to edge trigger mode.

0 (w): Invalid

1 (w): Clears the interrupt status

Writing “1” to these bits clears the corresponding status bits in the FIQ unmasked status register (INT[0x104]) that are set to edge trigger mode. Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

16. UART1/2/3

16.1 Description

UART1/2/3 are asynchronous data transfer interfaces that are compatible with the industry standard 16550. They perform serial conversion of CPU parallel data and transfer it to the peripheral devices, as well as receive serial data from the peripheral devices and convert it to parallel data.

S2S65A00 has three UART modules (UART1/2/3) whose functionalities are identical.

16.2 Block Diagram

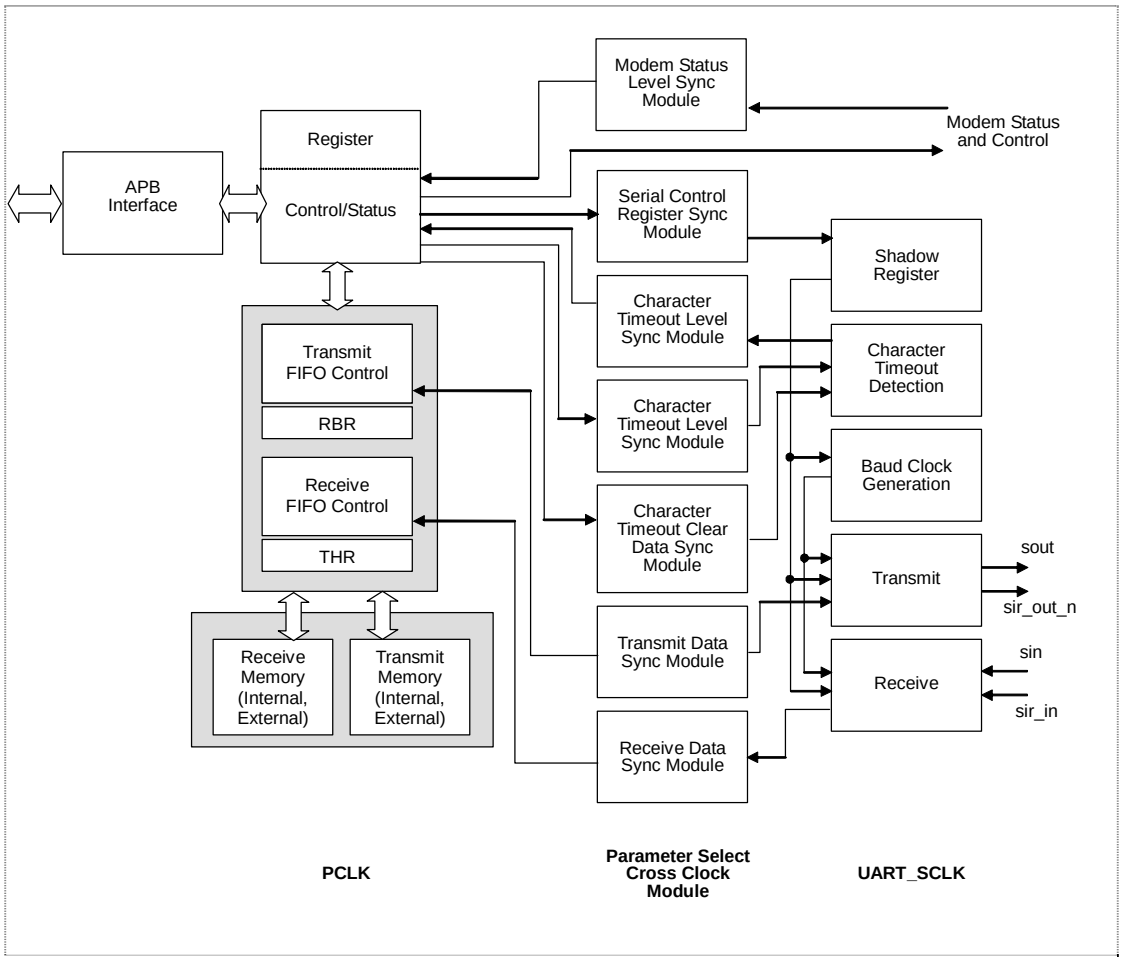


Fig.16.1 UART1/2/3 Block Diagram

16.3 External Pins

The external pins related to UART1/2/3 are shown below.

Pin Name	Input/Output	Pin Functions	Multiplex Pin
TXD1	Output	UART1 transmit data output	GPIOA0(*)
RXD1	Input	UART1 transmit data input	GPIOA1(*)
RTS1	Output	UART1 request to send output	GPIOA2(*)
CTS1	Input	UART1 clear to send input	GPIOA3(*)
TXD2	Output	UART2 transmit data output	GPIOA4(*)
RXD2	Input	UART2 transmit data input	GPIOA5(*)
RTS2	Output	UART2 request to send output	GPIOA6(*)
CTS2	Input	UART2 clear to send input	GPIOA7(*)
UART3_CLK	Input	UART3 clock input	GPIOC3(**)
TXD3	Output	UART3 transmit data output	GPIOC4(**)
RXD3	Input	UART3 transmit data input	GPIOC5(**)
RTS3	Output	UART3 request to send output	GPIOC6(**)
CTS3	Input	UART3 clear to send input	GPIOC7(**)

Note (*): The external pins for UART1/2 are multiplexed with GPIO pins or other pins. You can use the functions for UART1/2 by setting “Function 1 of other than GPIO” through the GPIO pin function register.

Note (**):The external pins for UART3 are multiplexed with GPIO pins or other pins. You can use the functions for UART1/2 by setting “Function 2 of other than GPIO” through the GPIO pin function register.

16.4 Description of Registers

The default base addresses where the UART1/2/3 control registers are located are as follows.
UART1: 0x FFFF_5000
UART2: 0x FFFD_4000
UART3: 0x FFFD_5000

If not otherwise specified, the default value of any register bit that is not reserved is “0”.

Note: Accesses to UART1/2/3 can read and write regardless whether the access size is 8 bits, 16 bits, or 32 bits, as long as UART1/2/3 are accessed at 32-bit boundary offset.

16.4.1 List of Registers

Table 16.1 List of Registers
Base Address: 0xFFFF_5000 (UART1)
0xFFFD_4000 (UART2)
0xFFFD_5000 (UART3)

Address Offset	DLAB	Register Name	Register Abbreviation	Default Value	R/W	Data Access Size
0x00	0	Receive Buffer Register	RBR	0x00	RO	8/16/32bit
0x00	0	Transmit Holding Register	THR	—	WO	8/16/32bit
0x00	1	Divisor Latch LSB Register	DLL	0x00	R/W	8/16/32bit
0x04	0	Interrupt Enable Register	IER	0x00	R/W	8/16/32bit
0x04	1	Divisor Latch MSB Register	DLM	0x00	R/W	8/16/32bit
0x08	—	Interrupt Identify Register	IIR	0x01	RO	8/16/32bit
0x08	—	FIFO Control Resister	FCR	—	WO	8/16/32bit
0x0C	—	Line Control Register	LCR	0x00	R/W	8/16/32bit
0x10	—	Modem Control Register	MCR	0x00	R/W	8/16/32bit
0x14	—	Line Status Register	LSR	0x60	RO	8/16/32bit
0x18	—	Modem Status Register	MSR	0xEX	RO	8/16/32bit
0x1C	—	Scratch Register	SCR	0x00	R/W	8/16/32bit
0x20	—	Test 0 Register	T0	0x00	R/W	8/16/32bit
0x24	—	Test 1 Register	T1	0x00	R/W	8/16/32bit
0x28	—	Test Status 0 Register	TS0	-	RO	8/16/32bit
0x2C	—	Test Status 1 Register	TS1	0x01	RO	8/16/32bit
0x30	—	Test Status 2 Register	TS2	0x0F	RO	8/16/32bit
0x3C	—	Test Status 3 Register	TS3	0x02	RO	8/16/32bit

16.4.2 Precautions on Register Access

The behavior when an address between the UART1/2/3 control registers are accessed is not guaranteed. For example, the behavior when address offset 01h is accessed in byte units is not guaranteed. Be sure to use the specified address offsets to access the registers.

Registers 20h through 3Ch are for debugging UART itself. They cannot be used for other than debugging. The specifications of these registers may be changed in the future.

16.4.3 Detailed Description of Registers

Receive Buffer Register (RBR)

UART1/2/3[0x00] DLAB [0] Default value = 0x00 Read Only

Serial Receive Data (RBR[7:0])							
7	6	5	4	3	2	1	0

Bits [7:0]: **RBR[7:0] Serial Receive Data Bits [7:0]**

This register works as a receive buffer during a read operation when the DLAB bit (UART1/2/3[0x0C] Bit 7) of the line control register is "0". From Bits [7:0], the byte data received on a serial port can be read. The read data is effective only if the data ready bit (UART1/2/3[0x14] Bit 0) of the line status register is "1". Accessing this register reads the first data in the receive FIFO. When the receive FIFO is full, data that has already received in the FIFO is retained but data that is sent after that is lost.

Transmit Holding Register (THR)

UART1/2/3[0x00] DLAB [0] Default value = — Write Only

Serial Transmit Data (THR [7:0])							
7	6	5	4	3	2	1	0

Bits [7:0]: **THR[7:0] Serial Transmit Data Bits [7:0]**

This register works as a transmit buffer during a write operation when the DLAB bit (UART1/2/3[0x0C] Bit 7) of the line control register is "0". Up to 16 bytes of data can be written until the transmit FIFO becomes full. When the transmit FIFO becomes full, data written to this register after that is lost.

Divisor Latch LSB Register (DLL)

UART1/2/3[0x00] DLAB [1] Default value = 0x00 Read/Write

Divisor Latch LSB (DL[7:0])							
7	6	5	4	3	2	1	0

Bits [7:0]: **DL[7:0] Divisor Latch LSB Bits**

This register is used to set the dividing ratio of the source clock that determines the baud rate, when the DLAB bit (UART1/2/3[0x0C] Bit 7) of the line control register is "1". A setting value is 16 bits and this register is for the lower 8 bits. The higher 8 bits are set in the divisor latch MSB register (UART1/2/3[0x04]). A baud rate is determined by the following expression.

Baud rate = Input clock/(16×DL[15:0])

Table 16.4 shows the correspondence of baud rates and division values.

Interrupt Enable Register (IER)

UART1/2/3[0x04] DLAB [0] Default value = 0x00 Read/Write

7	6	5	4	3	2	1	0
Programmable Transmit Holding Empty Interrupt Enable (EPTBEI)	Reserved (0)			Modem Status Interrupt Enable (EDSSI)	Receive Line Status Interrupt Enable (ELSI)	Transmit Holding Empty Interrupt Enable (ETBEI)	Receive Data Ready Interrupt Enable (ERBFI)

This register works as an interrupt enable register for UART1/2/3 when the DLAB bit (UART1/2/3[0x0C] Bit 7) of the line control register is "0", and enables/disables five UART1/2/3 interrupts. The bit 7 function does not exist for 16550. Do not use Bit 7 to create a firmware compatible with 16550.

Bit 7: **EPTBEI Programmable Transmit Holding Register Empty Interrupt Enable**

Becomes effective only if the operation is in FIFO mode (UART1/2/3[0x08] Bit 0 = 1) and the transmit holding register empty interrupt is enabled (UART1/2/3[0x04] Bit 1 = 1).

0 (r/w): Interrupt is disabled

1 (r/w): Interrupt is enabled

Note: this bit does not exist in 16550. (not compatible with 16550.)

Bit 3: **EDSSI Modem Status Interrupt Enable**

0 (r/w): Interrupt is disabled

1 (r/w): Interrupt is enabled

Note that in auto CTS control mode (when the modem control register Bit 5 is "1" and the FIFO control register Bit 0 is set to "1"), interrupt does not occur because the CTS# input changes.

Bit 2: **ELSI Receive Line Status Interrupt Enable**

0 (r/w): Interrupt is disabled

1 (r/w): Interrupt is enabled

Bit 1: **ETBEI Transmit Holding Register Empty Interrupt Enable**

0 (r/w): Interrupt is disabled

1 (r/w): Interrupt is enabled

Bit 0: **ERBFI Receive Data Ready Interrupt Enable**

0 (r/w): Interrupt is disabled

1 (r/w): Interrupt is enabled

For description of each interrupt cause, see Table 16.2.

Divisor Latch MSB Register (DLM)

UART1/2/3[0x04] DLAB [1] Default value = 0x00 Read/Write

7	6	5	4	3	2	1	0
Divisor Latch MSB (DL[15:8])							

Bits [7:0]: **DL[15:8] Divisor Latch MSB Bits**

This register is used to set the dividing ratio of the source clock that determines the baud rate, when the DLAB bit (UART1/2/3[0x0C] Bit 7) of the line control register is "1". A setting value is 16 bits and this register is for the higher 8 bits. The lower 8 bits are set in the divisor latch LSB register (UART1/2/3[0x00]). A baud rate is determined by the following expression.

Baud rate = Input clock/(16×DL[15:0])

For the correspondence of baud rates and division values, see Table 16.4.

Interrupt Identify Register (IIR)

UART1/2/3[0x08] Default value = 0x01 Read Only

FIFO Enable (FFEN [1:0])		Reserved		Interrupt ID (IID [3:0])			
7	6	5	4	3	2	1	0

Bits [7:6]: **FFEN [1:0] FIFO Enable Status Bits**

Indicate if the transmit and receive FIFOs are enabled or disabled.

00: FIFO is disabled

11: FIFO is enabled

Bits [3:0]: **IID [3:0] Interrupt ID Bits [3:0]**

Identify the source of the interrupt that currently occurs on UART1/2/3.

Table 16.2 UART1/2/3 Interrupt

IID [3:0]	Interrupt Type	Interrupt Source	How to Reset Interrupt ID	Priority
0001	No	No	n/a	n/a
0110	Receive line status interrupt	・Overrun error ・Parity error ・Framing error ・Break received	・Read out the line status register.	1 (Highest)
0100	Receive data ready interrupt	・Receive data ready	・Read out the line status register.	2
	Receive data trigger level reached interrupt	・In the receive FIFO, data has been received until reaching the trigger level.	・Reduce the amount of the data In the receive FIFO below the trigger level.	
1100	Character timeout interrupt	・When the receive FIFO has one or more characters of data, data is not read from the FIFO or is not input to the FIFO for the period equivalent to four characters.	・Read out the receive buffer register.	2
0010	Transmit holding register empty interrupt	・The transmit holding register is empty	・Read out the interrupt identify register. ・Write data to the transmit holding register.	3
	Transmit data trigger level reached interrupt	・Data in the transmit FIFO has been transmitted until reaching the trigger level.	・Read out the interrupt identify register. ・Write data to the transmit FIFO so that there is more data than the trigger level.	
0000	Modem status interrupt	・CTS# input. DSR input, RI input, or DCD input is changed.	・Read out the modem status register.	4 (Lowest)

FIFO Control Resister (FCR)

UART1/2/3[0x08] Default value = — Write Only

Receive Data Trigger Level (RCVRT[1:0])		Transmit Data Trigger Level (XMITT[1:0])		DMA Mode Select (DMAMS)	Transmit FIFO Reset (XMITFR)	Receive FIFO Reset (RCVRFR)	FIFO Enable (EFIFO)
7	6	5	4	3	2	1	0

Bits [7:6]: **RCVRT[1:0] Receive Data Trigger Level Setting**

Specifies the maximum number of bytes received in the receive FIFO above which a receive data trigger level reached interrupt should occur if the FIFO is enabled (FIFO enable (Bit 0)= 1).

00 (w): 1 byte
01 (w): 4 bytes
10 (w): 8 bytes
11 (w): 14 bytes

When the number of data bytes in the receive FIFO exceeds the value, a receive data trigger level reached interrupt occurs. The receive data interrupt cause is cleared when receive data is read out from the receive buffer register and the number of data bytes in the FIFO drops below the value.

Bits [5:4]: **XMITT[1:0] Transmit Data Trigger Level Setting**

Specifies the minimum number of data bytes left in the transmit FIFO below which a transmit data trigger level reached interrupt should occur if the programmable transmit holding register empty interrupt is enabled (UART1/2/3[0x04] Bit 7=1) and the FIFO is enabled (FIFO enable (Bit 0)= 1). This bit function does not exist for 16550. Do not use this bit to create a firmware compatible with 16550.

00 (w): 0 byte
01 (w): 2 bytes
10 (w): 4 bytes
11 (w): 8 bytes

When the number of data bytes in the transmit FIFO drops to the value, a transmit data trigger level reached interrupt occurs. The transmit data interrupt cause is cleared when transmit data is written to the transmit buffer register and the number of data bytes in the FIFO exceeds the value.

Note: This bit does not exist in 16550. (not compatible with 16550.)

Bit 3: **DMAMS DMA Mode Selection Bit**

Sets an operation mode of the status signal (internal signal) for the DMA controller.

0 (w): Single word mode
1 (w): Multi-word mode

Bit 2: **XMITFR Transmit FIFO Reset**

Clears data in the transmit FIFO if the FIFO is enabled (FIFO enable (Bit 0)= 1). The shift register is not reset. A "1" written to this bit is automatically reset.

0 (w): Disabled
1 (w): Clear

Note: If data is left in the FIFO, up to two characters are sent before the data is cleared.

Bit 1: **RCVRFR Receive FIFO Reset**

Clears data in the receive FIFO if the FIFO is enabled (FIFO enable (Bit 0)= 1). The shift register is not reset. A "1" written to this bit is automatically reset.

0 (w): Disabled
1 (w): Clear

Note: Depends on the receive FIFO reset, the data ready bit, which indicates that there is receive data, is not cleared. After you clear the receive FIFO, be sure to clear the data ready bit by reading out the receive buffer register (UART1/2/3[0x00]).

Bit 0: **EFIFO FIFO Enable**

Sets the transmit and receive FIFOs to enabled or disabled.

0 (w): Disabled. The FIFOs will not be used for transmission and reception.

1 (w): Enabled. Transmission and reception is done via the FIFOs.

Line Control Register (LCR)

UART1/2/3[0x0C] Default value = 0x00 Read/Write

DLAB	Break Control (SBRK)	Reserved (0)	Even Parity (EPS)	Parity Enable (PEN)	No. of Stop Bits (STB)	Word Length (WLS[1:0])	
7	6	5	4	3	2	1	0

Bit 7: **DLAB Divisor Latch Access Bit**

Specifies whether the divisor latch LSB/MSB register should be accessed or the receive buffer register, transmit holding register, and interrupt enable register should be accessed from UART1/2/3[0x00] and UART1/2/3[0x04].

0 (r/w): Accesses to the receive buffer register, transmit holding register, and interrupt enable register.

1 (r/w): Accesses to the divisor latch LSB/MSB register.

Bit 6: **SBRK Break Control**

Controls the output of the break signal.

0 (r/w): Normal output.

1 (r/w): Break signal output. When this bit is set to "1" (before "0" is written to it), the serial output is LOW.

Bit 4: **EPS Even Parity**

Selects even/odd parity.

0 (r/w): Odd parity

1 (r/w): Even parity

This setting is effective only if the parity enable (Bit 3) is "1".

Bit 3: **PEN Parity Enable**

Sets the parity check and parity bit attachment to enabled/disabled.

0 (r/w): Parity disabled. Receive data is treated as data without parity. No parity bit is attached to transmit data.

1 (r/w): Parity enabled. Data is received as data with a parity bit attached and parity check is performed. A parity bit is attached to transmit data.

Bit 2: **STB No. of Stop Bits**

Selects the number of stop bits.

0 (r/w): 1 bit

1 (r/w): 2 bits (if the data length is 6, 7, or 8 bits)
1.5 bits (if the data length is 5 bits)

The specified stop bits are attached to the transmit data. When receiving data, only the first 1 bit is checked regardless the specified number of stop bits.

Bits [1:0]: **WLS[1:0] Word Length Bits [1:0]**

Specifies the number of bits for transmit/receive characters (excluding parity and stop bits).

00 (r/w): 5 bits

01 (r/w): 6 bits

10 (r/w): 7 bits

11 (r/w): 8 bits

Modem Control Register (MCR)

UART1/2/3[0x10] Default value = 0x00 Read/Write

n/a		Auto Flow Control Enable (AFCE)	Loop Back (LOOP)	Output 2 Control (OUT2)	Output 1 Control (OUT1)	RTS Control (RTS)	DTR Control (DTR)
7	6	5	4	3	2	1	0

Bit 5: **AFCE Auto Flow Control Enable**

Enables auto flow control that uses CTS# and RTS# signals from the modem.

0 (r/w): Manual flow control mode

1 (r/w): Auto flow control mode

Setting this bit to "1" and Bit 0 of the FIFO control register to "1" enables auto CTS control mode, and automatically aborts data transmission when the CTS# input becomes HIGH while there is data in the transmit FIFO. When the CTS# input becomes LOW, data transmission is automatically resumed. In addition, setting Bit 1 of the modem control register to "1" enables auto RTS control mode, and changes the RTS# output to HIGH when the receive FIFO reaches the receive data trigger level. When the receive FIFO becomes empty, the RTS# output is automatically changed to LOW.

Note: Tis bit does not exist in 16550. (not compatible with 16550.)

Bit 4: **LOOP Loop Back**

Runs a local loop test on the modem.

0 (r/w): Normal operation

1 (r/w): Local loop test

When this bit is set to "1", the serial output line is linked to the serial input line and the test can be run by inputting transmit data.

The table below shows differences between loop back mode and normal mode.

As shown the table, you can run simple stand-alone self-diagnoses in loop back mode, since the output system for serial data and modem control feeds back to the input system. Even in this mode, the interrupt and break functions continue their operation.

Table 16.3 Differences between Loop Back Mode and Normal Mode

Difference Location		Loop Back Mode	Normal Mode
DTR# pin		Always outputs high level (inactive)	Outputs an inverted value of the DTR bit
RTS# pin		Always outputs high level (inactive)	Outputs an inverted value of the RTS bit
TXD pin		Always outputs high level (mark state)	Sequentially outputs serial data from the transmit shift register (TSR)
Modem Status Register	CTS bit	The setting of the RTS bit is read.	The inverted value of the CTS# pin is read.
	DSR bit	The setting of the DTR bit is read.	The inverted value of the DSR# pin is read.
	RI bit	The setting of the OUT1 bit is read.	The inverted value of the RI# pin is read.
	DCD bit	The setting of the OUT2 bit is read.	The inverted value of the DCD# pin is read.
	DCTS bit	Captures changes of the RTS bit.	Captures changes of the CTS# pin.
	DDSR bit	Captures changes of the DTR bit.	Captures changes of the DSR# pin.
	TERI bit	Captures falling edge changes of the OUT1 bit.	Captures changes of the RI# pin.
	DDCD bit	Captures changes of the OUT1 bit.	Captures changes of the DCD# pin.
Receive shift register		Captures serial data from the transmit shift register (TSR).	Captures serial data from the RXD pin.

Bit 3: **OUT2 Output 2# Control**

Directly controls OUT2# output. However, in loop back test mode, it is always high level output and connects to internal signal equivalent to DCD#.

0 (r/w): OUT2# = HIGH
1 (r/w): OUT2# = LOW

Note: This bit is available only in loop back mode.

Bit 2: **OUT1 Output 1# Control**

Directly controls OUT1# output. However, in loop back test mode, it is always high level output and connects to internal signal equivalent to RI#.

0 (r/w): OUT1# = HIGH
1 (r/w): OUT1# = LOW

Note: This bit is available only in loop back mode.

Bit 1: **RTS RTS Control**

Directly controls RTS# output if manual flow mode is enabled (modem control register Bit 5 is set to 0).

0 (r/w): RTS# = HIGH
1 (r/w): RTS# = LOW

Becomes auto RTS control mode if auto flow mode is enabled (modem control register Bit 5 is set to 1).

0 (r/w): No control for RTS
1 (r/w): Auto RTS control mode

However, in loop back test mode, it is always high level output and connects to internal signal equivalent to CTS#.

Bit 0: **DTR DTR# Control**

Directly controls DTR# output. However, in loop back test mode, it is always high level output and connects to internal signal equivalent to DSR#.

0 (r/w): DTR# = HIGH
1 (r/w): DTR# = LOW

Note: This bit is available only in loop back mode.

Line Status Register (LSR)

UART1/2/3[0x14] Default value = 0x60 Read Only

Receive FIFO Error	Transmit Empty (TEMT)	Transmit Holding Register Empty (THRE)	Break Interrupt (BI)	Framing Error (FE)	Parity Error (PE)	Overrun Error (OE)	Data Ready (DR)
7	6	5	4	3	2	1	0

Bit 7: **RCVRE Receive FIFO Error**

Indicates if there is a receive FIFO error (parity error, framing error, or break indicator). This bit is effective only if the FIFO is enabled (the FIFO control register UART1/2/3[08h] Bit 0 (FIFO enable bit)= 1); if the FIFO is disabled (the FIFO enable bit = 0), this bit is always set to 0.

0: No error
1: Error occurred

If there is no data in the FIFO other than the error data to be read next from the receive buffer, reading this register resets this flag to “0”.

Bit 6: **TEMT Transmit Empty**

Indicates that the transmit shift register and the FIFO/transmit holding register are empty.

0: Transmit data exists
1: No transmit data exists

Indicates that both the transmit FIFO register and the transmit shift register are empty, when the FIFO is enabled (FIFO enable bit = 1). Indicates that both the transmit holding register and the transmit shift register are empty, when the FIFO is disabled (FIFO enable bit = 0). This bit returns to 0 when transmit data is written.

Bit 5: **THRE Transmit Holding Register Empty**

The behavior of this bit varies depending on the setting of the interrupt enable register UART1/2/3[0x04] Bit 7 (EPTBEI bit).

When programmable transmit holding register empty interrupt is disabled (EPTBEI bit = 0), this bit indicates that the transmit FIFO/holding registers are empty.

0: Transmit data exists
1: No transmit data exists

Indicates that the transmit FIFO register is empty, when the FIFO is enabled (FIFO enable bit = 1). Indicates that the transmit holding register is empty, when the FIFO is disabled (FIFO enable bit = 0). When this flag is set, an interrupt request is generated if the transmit holding register empty interrupt is enabled. Note that this bit is set when the transmit FIFO/holding registers become empty as a result of transmission. It is not set when the registers become empty as a result of another operation such as reset and FIFO clear. This bit returns to 0 when transmit data is written.

When programmable transmit holding register empty interrupt is enabled (EPTBEI bit = 1), this bit indicates that the transmit FIFO is full.

0 (r) : Data can be written to the transmit FIFO
1 (r) : The transmit FIFO is full

By reading this bit before writing data to the transmit FIFO when a programmable transmit holding register empty interrupt occurs, transmit data can be written to the transmit FIFO until it becomes full. Using this bit and programmable transmit holding register empty interrupt ensures that the transmit FIFO always has data, and efficient data transfer can be carried out even in a system that cannot respond to an interrupt immediately.

Bit 4: **BI Break Interrupt Flag**

Indicates a break interrupt.

0: No break interrupt exists
1: A break interrupt exists

This flag is set if the input line is 0 during a period equivalent to one character.
It indicates that the error is occurring at the first character in the FIFO, when the FIFO is enabled (FIFO enable bit = 1). If receive line status interrupt is enabled, an interrupt request is generated when this flag is set. This flag resets to “0” when the register is read out.

Bit 3: **FE Framing Error**

Indicates that a framing error occurs.

0: No error exists
1: An error exists

This flag is set if the receive data contains no valid stop bit. It indicates that the error is occurring at the first character in the FIFO, when the FIFO is enabled (FIFO enable bit = 1). If receive line status interrupt is enabled, an interrupt request is generated when this flag is set. This flag resets to “0” when the register is read out.

Bit 2: **PE Parity Error**

Indicates that a parity error occurs.

0: No error exists
1: An error exists

This flag is set when a parity error is detected in the parity check that is performed when the parity enable bit is set . It indicates that the error is occurring at the first character in the FIFO, when the FIFO is enabled (FIFO enable bit = 1). If receive line status interrupt is enabled, an interrupt request is generated when this flag is set. This flag resets to “0” when the register is read out.

Bit 1: **OE Overrun Error**

Indicates that an overrun error occurs.

0: No error exists
1: An error exists

An overrun error occurs when the next character is received before the current character is read from the receive buffer register, if the FIFO is disabled (FIFO enable bit = 0). An overrun error also occurs when a new character is received while the FIFO is full, if the FIFO is enabled (FIFO enable bit = 1). If receive line status interrupt is enabled, an interrupt request is generated when this flag is set. This flag resets to "0" when the register is read out.

Bit 0: **DR Data Ready**

Indicates that there is receive data.

0: No receive data exists
1: Receive data exists

This bit is set to "1" if there is data of one or more valid characters in the receive buffer register or the FIFO. If receive data ready interrupt is enabled, an interrupt request is generated when this flag is set. This flag resets to "0" when all receive data is read from the receive FIFO.

Modem Status Register (MSR)

UART1/2/3[0x18] Default value = 0xEX Read Only

DCD Status (DCD) 7	RI Status (RI) 6	DSR Status (DSR) 5	CTS Status (CTS) 4	DCD Change (DDCD) 3	RI Falling Edge Change (TERI) 2	DSR Change (DDSR) 1	CTS Change (DCTS) 0

Bit 7: **DCD DCD Status**

Indicates the input state of the DCD pin.

0: DCD input = HIGH
1: DCD input = LOW

Note: This bit is available only in loop back mode.

Bit 6: **RI RI RI Status**

Indicates the input state of the RI pin.

0: RI input = HIGH
1: RI input = LOW

Note: This bit is available only in loop back mode.

Bit 5: **DSR DSR Status**

Indicates the input state of the DSR pin.

0: DSR input = HIGH
1: DSR input = LOW

Note: This bit is available only in loop back mode.

Bit 4: **CTS CTS Status**

Indicates the input state of the CTS# pin.

0: CTS1# input = HIGH
1: CTS#1 input = LOW

Bit 3: **DDCD DCD Change**

Indicates if there has been any change in the input of the DCD pin since the previous read operation. Note that, however, the change in the output 2 control bit (UART1/2/3[0x10] Bit 3) is reflected regardless the setting of the DCD pin if loop back mode is enabled (UART1/2/3[0x10] Bit 4 = 1).

0: No change exists
1: Change exists

This bit is cleared when the register is read out.

Note: This bit is available only in loop back mode.

Bit 2: **TERI RI Falling Edge Change**

Indicates if there has been any falling edge change in the input of the RI pin since the previous read operation. Note that, however, the falling edge change in the output 1 control bit (UART1/1/3[0x10] Bit 2) is reflected regardless the setting of the RI pin if loop back mode is enabled (UART1/2/2[0x10] Bit 4 = 1).

0: No falling edge change exists
1: Falling edge change exists

This bit is cleared when the register is read out.

Note: This bit is available only in loop back mode.

Bit 1: **DDSR DSR Change**

Indicates if there has been any change in the input of the DSR pin since the previous read operation. Note that, however, the change in the DTR bit (UART1/2/3[0x10] Bit 0) is reflected regardless the setting of the DSR pin if loop back mode is enabled (UART1/2/0[0x10] Bit 4 = 1).

0: No change exists
1: Change exists

This bit is cleared when the register is read out.

Note: This bit is available only in loop back mode.

Bit 0: **DCTS CTS Change**

Indicates if there has been any change in the input of the CTS# pin since the previous read operation. Note that, however, the change in the RTS control bit (UART1/2/3[0x10] Bit 1) is reflected regardless the setting of the CTS# pin if loop back mode is enabled (UART1/2/1[0x10] Bit 4 = 1).

0: No change exists
1: Change exists

This bit is cleared when the register is read out.

Scratch Register (SCR)

UART1/2/3[0x1C] Default value = 0x00 Read/Write

Scratch Bits (SCR [7:0])							
7	6	5	4	3	2	1	0

Bits [7:0]: **SCR [7:0] Scratch Bits**

Can be used as a general purpose register that does not affect the hardware operations.

Test 0 Register (T0)

UART1/2/3[0x20] Default value = 0x00 Read/Write

n/a							Test Mode
7	6	5	4	3	2	1	0

Bit 0: **Test Mode**

Select the test mode. In test mode, a test equivalent to a loop back test can be carried out.

0 (r/w): Normal mode
1 (r/w): Test mode

Test 1 Register (T1)

UART1/2/3[0x24] Default value = 0x00 Read/Write

n/a				DCD Test	RI Test	DSR Test	CTS Test
7	6	5	4	3	2	1	0

Bit 3: **DCD Test**

Controls the DCD input in test mode (UART1/2/3[0x20] Bit 0=1).

0 (r/w): Inputs low level (active)

1 (r/w): Inputs high level (inactive).

Bit 2: **RI Test**

Controls the RI input in test mode (UART1/2/3[0x20] Bit 0=1).

0 (r/w): Inputs low level (active)

1 (r/w): Inputs high level (inactive).

Bit 1: **DSR Test**

Controls the DSR input in test mode (UART1/2/3[0x20] Bit 0=1).

0 (r/w): Inputs low level (active)

1 (r/w): Inputs high level (inactive).

Bit 0: **CTS Test**

Controls the CTS# input in test mode (UART1/2/3[0x20] Bit 0=1).

0 (r/w): Inputs low level (active)

1 (r/w): Inputs high level (inactive).

Test Status 0 Register (TS0)

UART1/2/3[0x28] Default value = — Read Only

n/a				DCD raw Status	RI raw Status	DSR raw Status	CTS raw Status
7	6	5	4	3	2	1	0

Bit 3: **DCD Raw Status**

Always indicates the input state of the DCD pin.

0: Low level (active) is input

1: High level (inactive) is input

Bit 2: **RI Raw Status**

Always indicates the input state of the RI pin.

0: Low level (active) is input

1: High level (inactive) is input

Bit 1: **DSR Raw Status**

Always indicates the input state of the DSR pin.

0: Low level (active) is input

1: High level (inactive) is input

Bit 0: **CTS Raw Status**

Always indicates the input state of the CTS pin.

0: Low level (active) is input

1: High level (inactive) is input

Test Status 1 Register (TS1)

UART1/2/3[0x2C] Default value = 0x01 Read Only

n/a				DCD Status	RI Status	DSR Status	CTS Status
7	6	5	4	3	2	1	0

Bit 3: **DCD Status**

Indicates the input state of the DCD signal that the UART1/2/3 circuits recognize.

0: Low level (active) is input
1: High level (inactive) is input

Bit 2: **RI Status**

Indicates the input state of the RI signal that the UART1/2/3 circuits recognize.

0: Low level (active) is input
1: High level (inactive) is input

Bit 1: **DSR Status**

Indicates the input state of the DSR signal that the UART1/2/3 circuits recognize.

0: Low level (active) is input
1: High level (inactive) is input

Bit 0: **CTS Status**

Indicates the input state of the CTS# signal that the UART1/2/3 circuits recognize.

0: Low level (active) is input
1: High level (inactive) is input

Test Status 2 Register (TS2)

UART1/2/3[0x30] Default value = 0x0F Read Only

n/a			BAUDOUT Status	OUT2 Status	OUT1 Status	RTS Status	DTR Status
7	6	5	4	3	2	1	0

Bit 4: **BAUDOUT Status**

Indicates the state of the BAUDOUT signal that the UART1/2/3 circuits output.

0: Outputs low level
1: Outputs high level

Bit 3: **OUT2 Status**

Indicates the state of the OUT2 signal that the UART1/2/3 circuits output.

0: Outputs low level
1: Outputs high level

Bit 2: **OUT1 Status**

Indicates the state of the OUT1 signal that the UART1/2/3 circuits output.

0: Outputs low level
1: Outputs high level

Bit 1: **RTS Status**

Indicates the state of the RTS# signal that the UART1/2/3 circuits output.

0: Outputs low level (active)
1: Outputs high level (inactive)

Bit 0: **DTR Status**

Indicates the state of the DTR signal that the UART1/2/3 circuits output.

0: Outputs low level (active)
1: Outputs high level (inactive)

Test Status 3 Register (TS3)

UART1/2/3[0x3C] Default value = 0x02 Read Only

n/a					TXRDY Status	RXRDY Status	INTR Status
7	6	5	4	3	2	1	0

Bit 2: **TXRDY Status**

Indicates the state of the TXRDY signal that the UART1/2/3 circuits output.

0: Outputs low level (active)
1: Outputs high level (inactive)

Bit 1: **RXRDY Status**

Indicates the state of the RXRDY signal that the UART1/2/3 circuits output.

0: Outputs low level (active)
1: Outputs high level (inactive)

Bit 0: **INTR Status**

Indicates the state of the INTR signal that the UART1/2/3 circuits output.

0: Outputs low level (inactive)
1: Outputs high level (active)

16.4.4 Baud Rate Setup Example

The clock division value to set up the baud rate in UART1/2/3 is obtained from the following expression.

Division value = UART1/2/3_SCLK input clock frequency (Hz) ÷ baud rate (bps) ÷ 16

For example, the clock division values when input clock to UART1/2/3 (=UART1/2/3_SCLK) is at 24.543232 MHz are as follows.

Table 16.4 Baud Rate and Division Value

Baud Rate	×16 Ideal Clock Value	24.543232MHz UART1/2/3 Source Clock		
		×16 Clock Division Value	Error Percent %	Actual ×16 Clock
300	4800	5113	0.00	4800.2
600	9600	2557	0.02	9598.4
1200	19200	1278	0.02	19204.4
2400	38400	639	0.02	38408.8
4800	76800	320	0.13	76697.6
9600	153600	160	0.13	153395.2
14400	230400	107	0.44	229376.0
19200	307200	80	0.13	306790.4
28800	460800	53	0.49	463079.8
38400	614400	40	0.13	613580.8
57600	921600	27	1.37	909008.6
115200	1843200	13	2.43	1887940.9
125000	2000000	12	2.26	2045269.3
250000	4000000	6	2.26	4090538.7
500000	8000000	3	2.26	8181077.3
750000	12000000	2	2.26	12271616.0
1500000	24000000	1	2.26	24002560.0

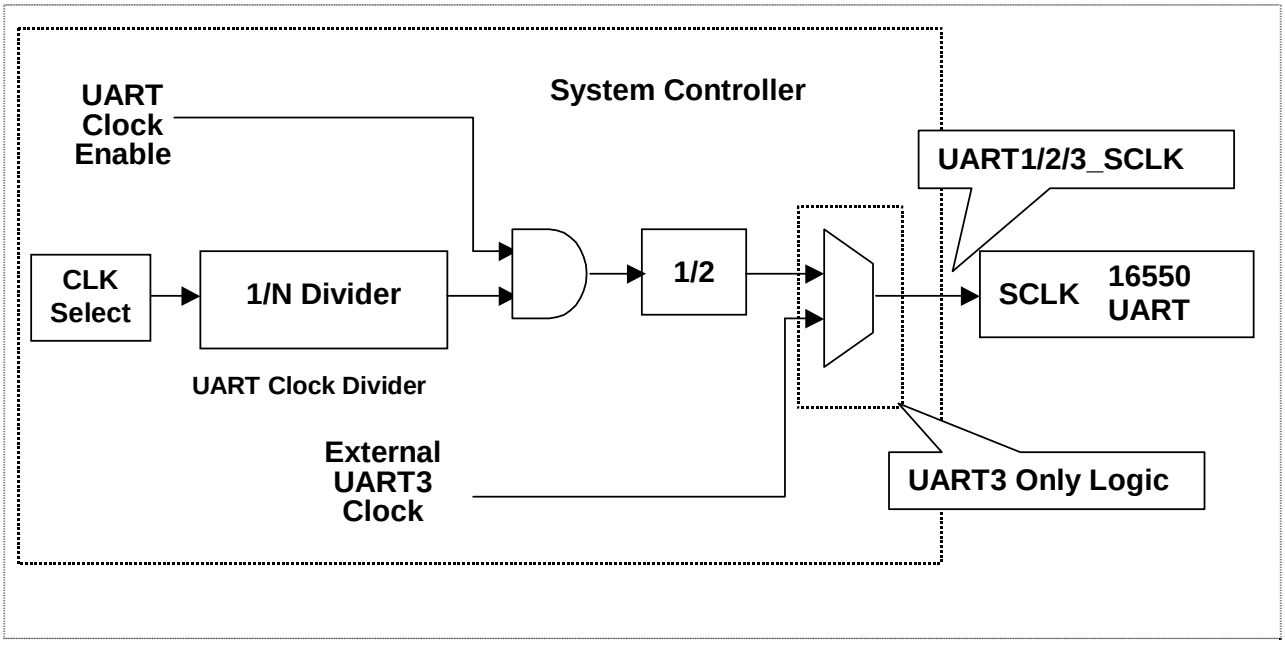


Fig.16.2 Conceptual Diagram of UART1/2/3 Clock

16.5 Usage Restrictions for This UART1/2/3

Although the chip circuits themselves have most functions for 16550, restrictions are applied to some registers because this chip does not provide all signals as I/O pins that are compatible with 16550. This chip applies restrictions to the following registers.

Offset Address	Register Bit Name	Restrictions
UART1/2/3[0x10]Bit 0	DTR: DTR# Control	Only loop back mode available
UART1/2/3[0x10]Bit 2	OUT1: Output 1# Control	Only loop back mode available
UART1/2/3[0x10]Bit 3	OUT2: Output 2# Control	Only loop back mode available
UART1/2/3[0x18]Bit 1	DDSR: DSR Change	Only loop back mode available
UART1/2/3[0x18]Bit 2	TERI: RI Falling Edge Change	Only loop back mode available
UART1/2/3[0x18]Bit 3	DDCD: DCD Change	Only loop back mode available
UART1/2/3[0x18]Bit 5	DSR: DSR Status	Only loop back mode available
UART1/2/3[0x18]Bit 6	RI: RI Status	Only loop back mode available
UART1/2/3[0x18]Bit 7	DCD: DCD Status	Only loop back mode available

The following register setting is available for 16550 but not for this UART1/2/3.

Offset Address	Register Bit Name	Restrictions
UART1/2/3[0x0C]Bit 5	Stick parity	Always unavailable

The following register settings are available for this UART1/2/3 but not for 16550: These registers should not be used when you create a firmware compatible with 16550.

Offset Address	Register Bit Name	Restrictions
UART1/2/3[0x04]Bit 7	EPTBEI: Programmable Transmit Holding Register Empty Interrupt Enable	Not compatible with 16550
UART1/2/3[0x08]Bit [5:4]	XMITT[1:0]: Transmit Data Trigger Level Setting	Not compatible with 16550
UART1/2/3[0x10]Bit 5	AFCE: Auto Flow Control Enable	Not compatible with 16550

17. I²C Single Master Core Module (I²C)

17.1 Description

17.1.1 Master Mode

- Supports I²C bus I²C single master mode.
- Does not support I²C bus I²C multi-master mode.
- Multiple slave devices can be connected to the I²C bus.
- In addition to the shift register involved in I²C data transfer, this module has transmit and receive buffers (TBUF and RBUF) to ease the timing of software read/write.
- Detects bus error statuses and can report them through the status register.
- Supports the I²C clock wait function.

17.1.2 Slave Mode

- Does not support slave mode.

17.2 Block Diagram

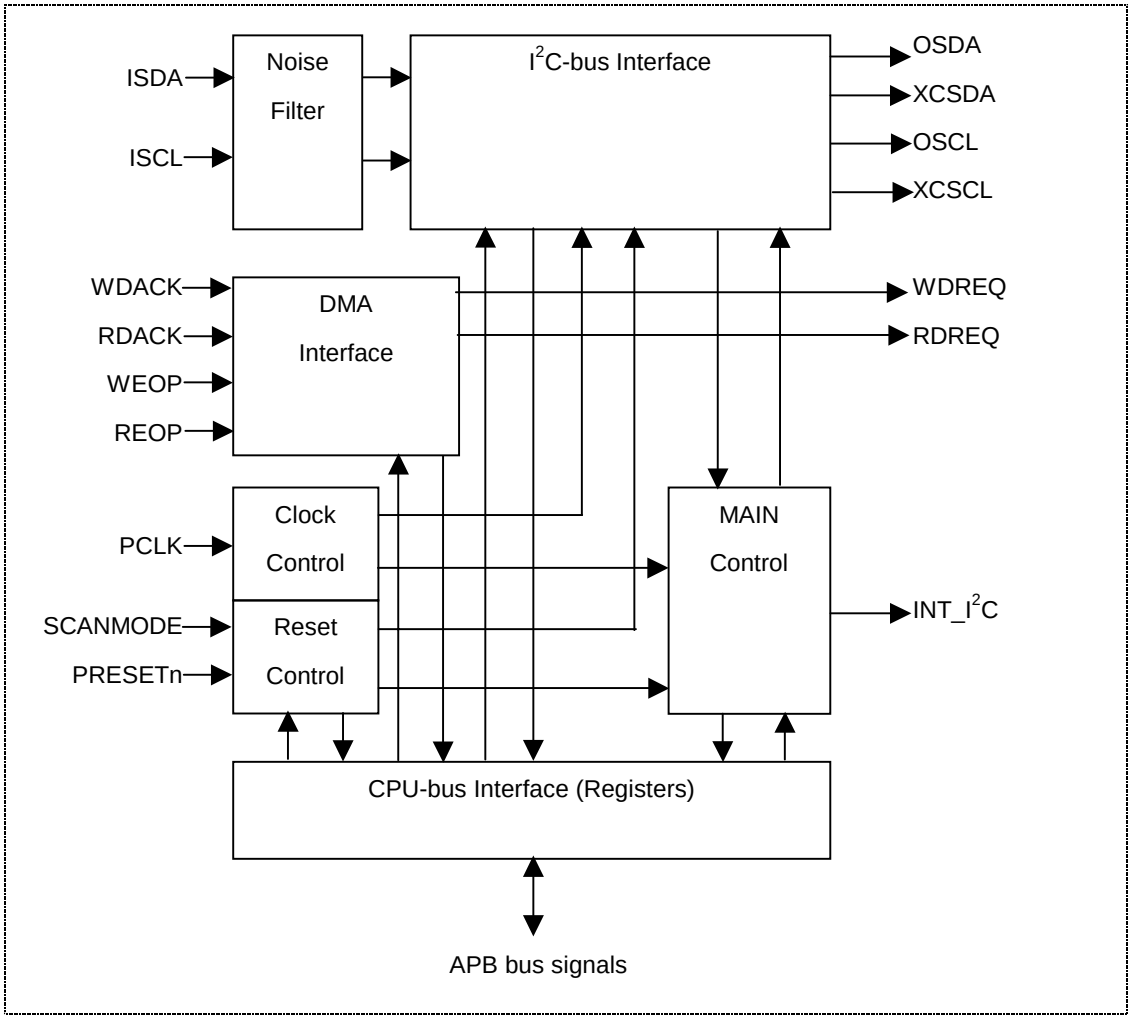


Fig.17.1 I^2C Internal Block Diagram

17.3 External Pins

Pin Name	Input/Output	Pin Functions	Multiplex Pin*/Remarks
SCL	Input and output	I^2C clock input and output	GPIOA6
SDA	Input and output	I^2C data input and output	GPIOA7

Note (*): The external pins for I^2C are multiplexed with GPIO pins. You can use the functions for I^2C by setting "Function 2 of other than GPIO" through the GPIO pin function register.

17.4 Registers

17.4.1 List of Registers

The address offset values in the table below are offsets from the base address that is assigned to the I2C logic module. The base address of these registers is 0xFFFE_D000.

Table 17.1 List of Registers (Base Address: 0xFFFE_D000)

Address Offset	Register Name	Default Value	R/W	Access Size *1
0x00	I^2C transmit data register	0000 0000b	R/W	8 (16/32)
0x04	I^2C receiving data register	0000 0000b	RO	8 (16/32)
0x08	I^2C control register	0000 0000b	R/W	8 (16/32)
0x0C	I^2C bus status register	00xx 0000b *2	RO	8 (16/32)
0x10	I^2C error status register	0000 0000b	RO	8 (16/32)
0x14	I^2C interrupt control/status register	0000 0000b	R/W	8 (16/32)
0x18	I^2C–BUS sample clock dividing setting register	0000 0000b	R/W	8 (16/32)
0x1C	I^2C SCL clock dividing setting register	0000 0000b	R/W	8 (16/32)
0x20	I^2C I/O control register	0001 0001b	R/W	8 (16/32)
0x24	I^2C DMA mode register	0000 0000b	R/W	8 (16/32)
0x28	I^2C DMA counter value (LSB) register	0000 0000b	R/W	8 (16/32)
0x2C	I^2C DMA counter value (MSB) register	0000 0000b	R/W	8 (16/32)
0x30	I^2C DMA status register	0000 1000b	RO	8 (16/32)
0x34 – 0x38	Reserved	—	—	—

*1: The registers in the List of Registers are defined with 8 bits and firmware usually uses these registers in 8-bit access. Although 16-bit or 32-bit access is possible if the firmware uses 16-bit or 32-bit access instruction, only the lower 8 bits can be used as valid values.

*2: Bit 5 (SDA) and Bit 4 (SCL) of this register monitor the external pins: SDA pin and SCL pin. The status of these pins determine the default value. Because these pins are generally pulled up externally, the settings are SDA=1 and SCL=1, which correspond to the settings of the external pins.

17.4.2 Detailed Description of Registers

The following explains the details of each register:

I²C transmit data register

I²C[0x00] Default value = 0000 0000b Read/write

I²C Transmit Data TD [7:0]

7	6	5	4	3	2	1	0

Bits [7:0]: **TD[7:0] I²C Transmit Data**

This register is an 8-bit buffer that stores transmit data of I²C-BUS transfer.

The MSB bit through the LSB bit are transmitted in sequence when data is transmitted from the master. (See the diagram below.)

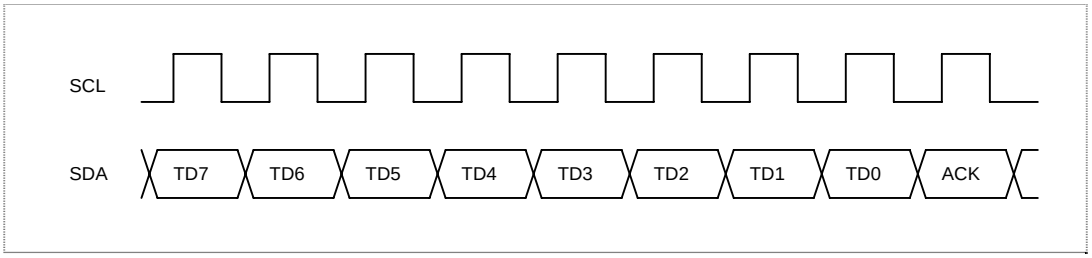


Fig.17.2 Bit Sequence When transmitting

I²C receiving data register

I²C[0x04] Default value = 0000 0000b Read Only

I²C Receive Data RD [7:0]*

7	6	5	4	3	2	1	0

Bits [7:0]: **RD[7:0] I²C Receive Data**

This register is an 8-bit buffer that stores receive data of I²C-BUS transfer.

Transmitted data from a slave device, the MSB bit through the LSB bit, is received in sequence. (See the diagram below.)

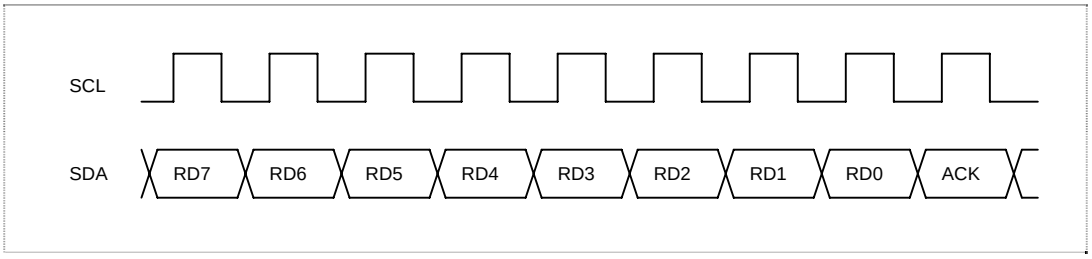


Fig.17.3 Bit Sequence When Receiving

I^2C control register

I^2C[0x08] Default value = 0000 0000b Read/write

Reserved (0)		SR	CLKW*	TACK*	TRNS [2:0] *		
7	6	5	4	3	2	1	0

Note: (*) can also be initialized by software reset.

This register controls occurrence of start and stop conditions, data transmission/reception start, and others.
When TRNS[2:0] are written, the specified operation is performed.
If TRNS[2:0] are written when the I^2C bus status register I^2C[0x0C] bit 7 is set to RUN =1, the command is not executed.

Bits [7:6]: **Reserved (0)**

Bit 5: **SR Software Reset**

Forces reset by the software to perform an initialization.

0: Disables software reset
1: Enables software reset

Bit 4: **CLKW Clock wait Mode Enable**

Selects whether to use the clock wait function.

0: Clock wait mode disable
1: Clock wait mode enable

Bit3: **TACK Data Reception Acknowledgement Setting (only for receive [Read] mode)**

Sets an acknowledgement to send to the slave device at data reception.

0: Does not send an acknowledgement to the slave device.
1: Sends an acknowledgement to the slave device.

Note: In DMA transfer mode, the value of this bit (TACK) is used only when the last byte is transferred; “0” is sent in all other byte transfer.

Bits [2:0]: **TRNS[2:0] Transmit Control Command**

Specifies the operation and instructs about the start.

001: I^2C start condition
010: I^2C stop condition
011: I^2C data receive
100: I^2C data transfer
101, 110: Unavailable
000, 111: Clears the error flag of the error status register.

I²C bus status register

I²C[0x0C] Default value = 00xx 0000b Read Only

RUN*	Reserved	SDA	SCL	Using*	Busy*	Error*	Finish*
7	6	5	4	3	2	1	0

Note: (*) can also be initialized by software reset.
x: unknown (Irregularity)

This register represents the status of the I²C-BUS.

Bit 7: **RUN Command Execution Status**

Indicates if there is a command operation.

0: I²C-BUS (command) wait state
1: I²C-BUS (command) execution is in progress

Bit 6: **Reserved**

Bit [5:4]: **SDA, SCL I²C-BUS Monitor**

Indicates the current status of the SDA and SCL. Because SDA and SCL are usually pulled up externally, these bits are set to HIGH and indicate values accordingly.

Bit 3: **Using I²C-BUS Use**

Indicates if the I²C single master uses the I²C-BUS.

0: The I²C single master is not using the I²C-BUS.
1: The I²C single master is using the I²C-BUS.

Bit 2: **Busy I²C-BUS Busy**

Indicates the operating state of the I²C-BUS.

0: I²C-BUS BusFree state
1: I²C-BUS is in use

Bit 1: **Error Error occurrence**

Indicates if an error has occurred.

0: No error exists
1: Error state has occurred
This bit can be cleared by writing operation to the I²C control register.

Bit 0: **Finish Command Execution Finished**

Indicates that the command finished.

0: Command is not executed or currently in progress
1: Command execution finished
This bit is cleared when reset is enabled (RESET# = LOW), when the I²C control register bit 5 (software reset) is set to 1, or by writing a valid command to the I²C control register bits[2:0] (TRNS).

I^2C error status register

I2C[0x10] Default value = 0000 0000b Read Only

Reserved			Reception Acknowledgement *	SCL Mismatch *	SDA Mismatch *	Stop Condition *	Start Condition *
7	6	5	4	3	2	1	0

Note: (*): Can also be initialized by software reset.

This register represents error states.

Bits [7:5]: **Reserved**

Bit 4: **Reception Acknowledgement Error**

Indicates if there is an error in a reception acknowledgement.

0: No error exists

1: An error exists

Bit 3: **SCL Mismatch Error**

Indicates if there is an error due to a mismatch between ISCL and OSCL.

0: Normal operation

1: Clock mismatch is detected.

Bit 2: **SDA Mismatch Error**

Indicates if there is an error due to a mismatch between ISDA and OSDA.

0: Normal operation

1: Data mismatch is detected.

Bit 1: **Stop Condition Detection**

Indicates if there is an error due to an occurrence of a stop condition.

0: Normal operation

1: A stop condition other than one caused by a command is detected.

Bit 0: **Start Condition Detection**

Indicates if there is an error due to an occurrence of a start condition.

0: Normal operation

1: A start condition other than one caused by a command is detected.

Note: These error statuses are cleared when reset is enabled (RESET# = LOW), when the I^2C control register bit 5 (software reset) is set to 1, or by writing a valid command to the I^2C control register bits[2:0] (TRNS).

I²C interrupt control/status register

I²C[0x14] Default value = 0000 0000b — Read/Write

7	6	5	4	3	2	1	0
Reserved (0)				Error INT Status Flag *	Completion INT Status Flag *	Error INT Enable *	Completion INT Enable *

Note: (*): Can also be initialized by software reset.

Bits [7:4]: **Reserved (0)**

Bit 3: **Status Flag for Interrupt Caused by Error Occurrence**

This bit is effective only when Bit 1 is set to "1".

0: No error has occurred

1: An interrupt request caused by error occurrence has been generated
This flag is cleared by writing "1" to this bit.

Bit 2: **Command Completion Interrupt Status Flag**

This bit is effective only when Bit 0 is set to "1".

0: The command completion interrupt enable bit (Bit 0) is set to 0, or the command has not been completed.

1: An interrupt request caused by command completion has been generated
This flag is cleared by writing "1" to this bit.

Bit 1: **Error Occurrence Interrupt Enable**

Controls interrupts caused by error occurrence.

0: Disables Interrupt generation caused by error occurrence

1: Enables interrupt generation caused by error occurrence

Bit 0: **Command Completion Interrupt Enable**

Controls interrupt generation caused by command completion.

0: Disables interrupt generation caused by command completion

1: Enables interrupt generation caused by command completion

I²C-BUS sample clock dividing setting register

I²C[0x18] Default value = 0000 0000b — Read/Write

7	6	5	4	3	2	1	0
Reserved (0)				Dividing Ratio (I²C Sample) [3:0]			

Generates an I²C-BUS sample clock by dividing the master clock.

Bits [7:4]: **Reserved bit (0)**

Bits [3:0]: **Dividing Ratio (I²C Sample) [3:0]**

Sets the dividing ratio to generate an I²C-BUS sample clock from the master clock.

The frequency of the I²C-BUS sample clock ($f_{I2C\ sample}$) is obtained from the following expression.

Bits [3:0] > 0:

$f_{I2C\ sample} = f_{PCLK}/(4*m)$[Hz]

Bits [3:0] = 0:

$f_{I2C\ sample} = f_{PCLK}/2$[Hz]

Note: In the expression above, m = dividing ratio (I²C sample) [3:0].

I²C SCL clock dividing setting register

I²C[0x1C] Default value = 0000 0000b Read/Write

Reserved (0)					Dividing Ratio (SCL) [2:0]		
7	6	5	4	3	2	1	0

This register is for setting the dividing ratio to generate SCL.

Bits [7:3]: **Reserved (0)**

Bit [2:0]: **Dividing Ratio (SCL) [2:0]**

Sets the dividing ratio to generate SCL from an I²C-BUS sample clock.

The frequency of the SCL for I²C-BUS transfer is obtained from the following expression.

Dividing Ratio (I²C Sample)[3:0] > 0:

$f_{SCL} = f_{I2C\ sample}/(2n*4)$

$= f_{PCLK}/\{(4*m) * (2n*4)\}$

$= f_{PCLK}/(16*m*2n)$[Hz]

Dividing ratio (I²C sample)[3:0] = 0:

$f_{SCL} = f_{PCLK}/\{2*(2n*4)\}$

$= f_{PCLK}/(8*2n)$ [Hz]

Note: In the expression above, “m” and “n” are specified as follows:

m = dividing ratio (I²C sample)[3:0]

(See I²C-bus sample clock dividing setting register Bits[3:0].)

n = dividing ratio (SCL) [2:0]

I²C I/O control register

I²C[0x20] Default value = 0001 0001b Read/Write

Reserved (0)		High Drive SDA	Sample SDA	Reserved (0)		High Drive SCL	Sample SCL
7	6	5	4	3	2	1	0

This register is for selecting output mode and determining whether the noise filter should be enabled or disabled.

Bits [7:6]: **Reserved (0)**

Bit 5: **SDA HIGH Drive Enable**

0: “SDA=1” is controlled by the pull-up resistor external to the IC.

1: “SDA=1” is controlled by this IC itself by driving "High".

Bit 4: **SDA Sampling Enable**

0: As SDA input, one data sample is sampled with the I²C-BUS sample clock.

1: As SDA input, two data samples are sampled with the I²C-BUS sample clock.

Note: Use this bit with the value of “0”.

Bits [3:2]: **Reserved (0)**

Bit 1: **SCL HIGH Drive Enable**

0: “SCL=1” is controlled by the pull-up resistor external to the IC.

1: “SCL=1” is controlled by this IC itself by driving “High”.

Bit 0: **SCL Sampling Enable**

0: As SCL input, one data sample is sampled with the I²C-BUS sample clock.

1: As SCL input, two data samples are sampled with the I²C-BUS sample clock.

Note: Use this bit with the value of "0".

I²C DMA mode register

I²C[0x24] Default value = 0000 0000b Read/Write

Reserved (0)						DMA_MODE [1:0]	
7	6	5	4	3	2	1	0

Bits[7:2]: **Reserved (0)**

Bits [1:0]: **DMA_MODE[1:0] DMA Mode Setting**

00: Do not use DMA transfer.

01: Single address mode

10: Dual address mode (without EOP). Use the DMA counter.

11: Dual address mode (with EOP).

Although DMA modes can be turned ON/OFF during a transfer (from Start(restart) to Stop), do not switch the DMA mode (Single, Dual (with EOP), Dual (without EOP)) that you intend to use.

Burst transfer is not supported.

I²C DMA counter value (LSB) register

I²C[0x28] Default value = 0000 0000b Read/Write

DMA Counter Value (LSB)							
7	6	5	4	3	2	1	0

Write: This register is for setting the lower byte [7:0] of the DMA counter value.
Read: This register returns the lower byte [7:0] of the DMA counter value.

I²C DMA counter value (MSB) register

I²C[0x2C] Default value = 0000 0000b Read/Write

DMA Counter Value (MSB)							
7	6	5	4	3	2	1	0

Write: This register is for setting the higher byte [15:8] of the DMA counter value.
Read: This register returns the higher byte [15:8] of the DMA counter value.

I²C DMA status register

I²C[0x30] Default value = 0000 1000b Read Only

Reserved				TBUF Empty*	RBUF Update*	RDREQ Monitor*	WDREQ Monitor*
7	6	5	4	3	2	1	0

Note: (*):Can also be initialized by software reset.

Bits[7:4]: **Reserved**

Bit 3: **TBUF_Empty Transmit Buffer Empty**

Indicates whether the write operation to the transmit data buffer (TBUF) of the I²C single master core is possible.

0: The transmit buffer has data to transmit.

1: The transmit buffer is empty. Write operation is possible.

Bit 2: **RBUF_Update Receive Buffer Update**

Indicates whether the data in the I²C receive data buffer (RBUF) is updated.

0: The receive buffer has not been updated.

1: The data in the receive buffer has been updated.

Bit 1: **Reserved（RDREQ Monitor RDREQ Signal Monitor）**

Indicates the state of the RDREQ signal.

0: Indicates that the signal on the RDREQ pin is Low.

1: Indicates that the signal on the RDREQ pin is High.

Note: The state may change but you cannot use this bit.

Bit 0: **Reserved（WDREQ Monitor WDREQ Signal Monitor）**

Indicates the state of the WDREQ signal.

0: Indicates that the signal on the WDREQ pin is Low.

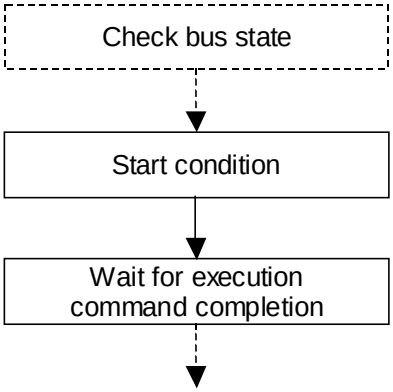
1: Indicates that the signal on the WDREQ pin is Low.

Note: The state may change but you cannot use this bit.

17.5 Explanation of Operation (Example of Use: Bus Control Command)

The following is examples for controlling I^2C bus with this module. In practice, it may be required to check the states or respond to an error. In addition, a specific control method depends on the specifications of the slave device.

17.5.1 Start (S) Flow Example

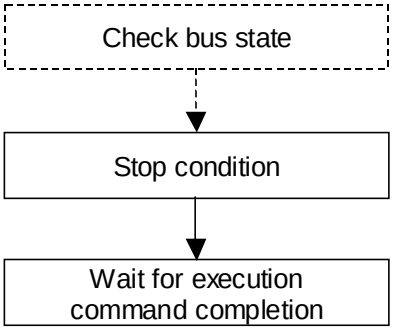


Check that I2C[OxOC] bit 2 is set to “0”

Start condition setting:
Set the I2C[OxO8] bit [2:0] to “001”

Wait for the I2C[OxOC] bit 2 to be set to “1”

17.5.2 Stop (P) Flow Example

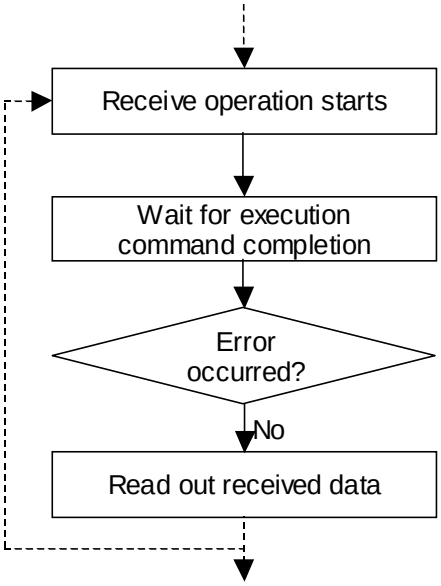


Check that I2C[OxOC] bit 3 is set to “1”

Stop condition setting:
Set the I2C[OxO8] bit [2:0] to “010”

Wait for the I2C[OxOC] bit 0 to be set to “1”

17.5.3 Receive (R) Flow Example



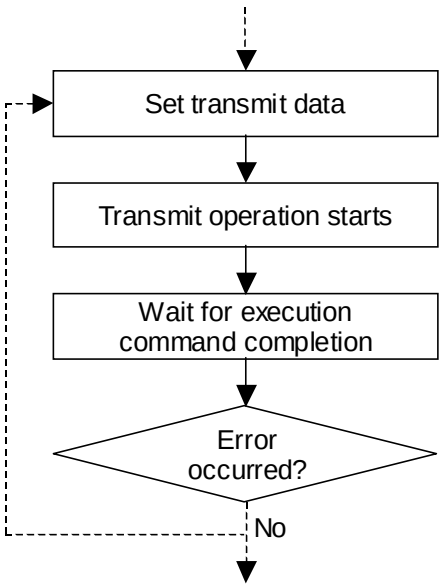
Receive setting:
Set the I2C[OxO8] bit [2:0] to “011”

Wait for the I2C[OxOC] bit 0 to be set to “1”

Check the I2C[OxOC] bit 1
“0” indicates that no error exists; “1”, an error has occurred.

Read out received data of I2C[OxO4]

17.5.4 Transfer (T) Flow Example



Set transmit data in I2C[OxO0]

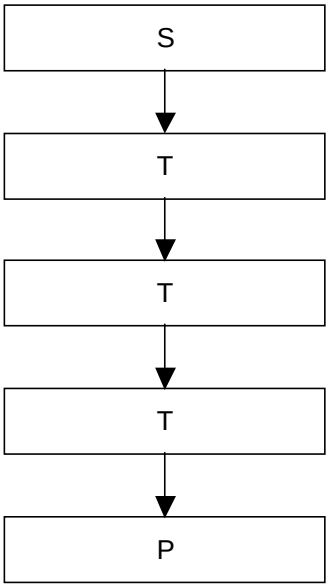
Transfer setting:
Wait for the I2C[OxOC] bit 0 to be set to "100"

Wait for the I2C[OxOC] bit 0 to be set to "1"

Check the I2C[OxOC] bit 1
"0" indicates that no error exists; "1", an error has occurred.

17.5.5 Example of Sequence When Writing to Slave Device

The S/P/R/T in the figure below correspond to above-mentioned Start/Stop/Receive/Transfer flows.



Start condition occurs

Transmits the slave address and Write mode

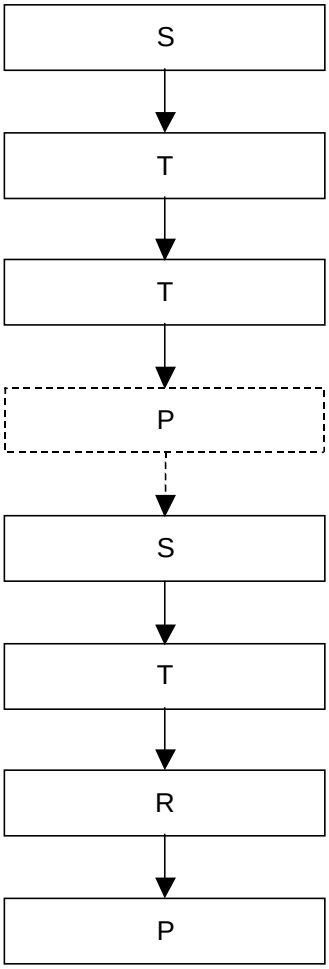
Transmits the internal address of the slave device

Transmits the write data

Stop condition occurs

17.5.6 Example of Sequence When Reading from Slave

The S/P/R/T in the figure below correspond to above-mentioned Start/Stop/Receive/Transfer flows.



Start condition occurs

Transmits the slave address and Write mode

Transmits the inner address of the slave device

Stop condition occurs

Start condition occurs

Transmits the address and Read mode of the slave device

Transmits the read data

Stop condition occurs

17.6 Usage Restrictions for This I²C Single Master Core Module (I²C)

This chip applies restrictions to the following registers.

Offset Address	Register Bit Name	Restrictions
I²C[0x30] Bit 0	WDREQ Signal Monitor	Unavailable
I²C[0x30] Bit 1	RDREQ Signal Monitor	Unavailable

18. I²S (I2S)

18.1 Description

The I²S module complies with the I²S standard defined by Philips. This module is mainly used for voice/audio data communications. I²S supports 2-Channel communications. You can choose transmit or receive for each channel. For example, you can not only receive voice/audio data from an audio device while transmitting voice/audio data to an audio device, but also receive data from two different audio device at the same time.

18.1.1 Function

Functions that I²S provides.

- Allows you to select between master mode (outputs SCK and WS) and slave mode (inputs SCK and WS)
- Allows you to select between transmit mode (outputs SD) and receive mode (inputs SD)
- Supports 16-, 14-, an 8-bit data widths
- Allows you to select between stereo and monaural
- Supports frame cycles, 32fs, 64fs, 128fs, and 256fs
- Allows you to select a clock dividing setting from 1/2 to 1/512 (256 steps) of the source clock (only for master mode)
- Supports DMA
- Clock sharing function (Makes both channels use the same clock)
- Detects FIFO overflow and underflow
- Interrupt generation based on the six types of the FIFO states
- 0 output function (in transmit mode and when the FIFO underflows)
- Conversion from monaural to stereo (only for transmit mode)

18.2 Block Diagram

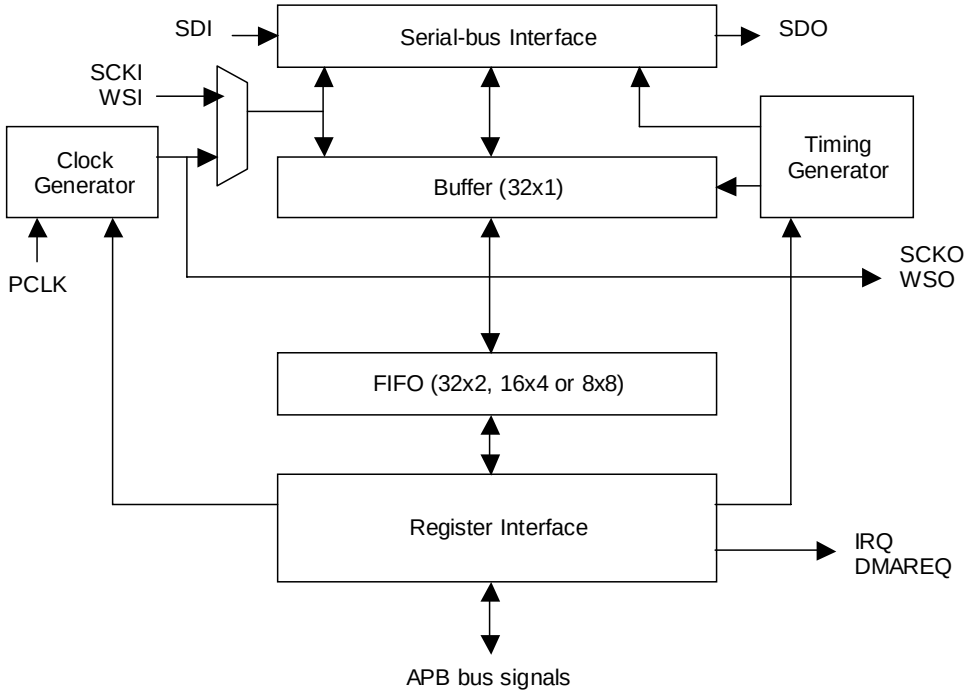


Fig.18.1 I²S Block Diagram

18.3 External Pins

The external pins related to I^2S are as follows.

Pin Name	Input/Output	Pin Functions	Multiplex Pin/Remarks
I^2S_SCK (I2S0_SCK)	Input and output	I^2S0 serial clock	GPIOB1 (*)
I^2S_WS (I^2S0_WS	Input and output	I^2S0 word select	GPIOB0 (*)
I^2S_SDO (I^2S0_SD)	Input and output	I^2S0 serial data	GPIOB2 (*)
(I^2S1_SCK)	Input and output	I^2S1 serial clock	GPIOA3 (**)
(I^2S1_WS)	Input and output	I^2S1 word select	GPIOA2 (**)
I^2S_SDI (I^2S1_SD)	Input and output	I^2S1 serial data	GPIOB3 (*)

Note (*): The external pins for I^2S are multiplexed with GPIO pins or other pins. You can use the functions for I^2S by setting "Function 2 of other than GPIO" through the GPIO pin function register.

Note (**): For the external WS/SCK pins for I^2S1, set transmit mode on I^2S0 and receive mode on I^2S1, and you do not need external connections. When you connect the external pins, note that they are multiplexed with GPIO pins or other pins. You can use the functions for them by setting "Function 2 of other than GPIO" through the GPIO pin function register.

18.4 Description of Registers

18.4.1 List of Registers

The default base address where the I^2S control registers are located is 0xFFFE_E000.
If not otherwise specified, the default value of any register bit that is not reserved is "0".

Table 18.1 I^2S[1:0] List of Registers

Address Offset	Register Name	Default Value	R/W	Data Access Size
I^2S0 Control Register Base Address: 0xFFFE_E000				
0x00	I^2S0 Control Register	0x0000	R/W	16/32
0x04	I^2S0 Clock Dividing Register	0x0000	R/W	16/32
0x08	I^2S0 Transmit/Receive Port Register	—	R/W	8/16/32
0x10	I^2S0 Interrupt Status Register	0x0000	R/W	16/32
0x14	I^2S0 Interrupt Raw Status Register	0x0009	RO	16/32
0x18	I^2S0 Interrupt Enable Register	0x0000	R/W	16/32
0x1C	I^2S0 Current Status Register	0x0009	RO	16/32
I^2S1 Control Register Base Address: 0xFFFE_E000				
0x40	I^2S1 Control Register	0x0000	R/W	16/32
0x44	I^2S1 Clock Dividing Register	0x0000	R/W	16/32
0x48	I^2S1 Transmit/Receive Port Register	—	R/W	8/16/32
0x50	I^2S1 Interrupt Status Register	0x0000	R/W	16/32
0x54	I^2S1 Interrupt Raw Status Register	0x0009	RO	16/32
0x58	I^2S1 Interrupt Enable Register	0x0000	R/W	16/32
0x5C	I^2S1 Current Status Register	0x0009	RO	16/32

18.4.2 Detailed Description of Registers

I^2S[1:0] Control Register

I^2S0[0x00], I^2S1[0x40] Default value = 0x0000 Read/Write

n/a			CNVM2S	FRAMECYC [1:0]		CLKOUTEN	CLKSEL
15	14	13	12	11	10	9	8
SFTRST (WO)	DATAWIDTH [1:0]		MONO/ STEREO	DMAEN	TX/RX	MST/SLV	I2SEN
7	6	5	4	3	2	1	0

Bit 12: **CNVM2S Conversion from Monaural to Stereo (Only for transmit mode)**

Converts monaural data to stereo data before transmitting. Concretely speaking, monaural data to be transmitted on the L channel is also transmitted on the R channel.

0: Disables monaural to stereo conversion
1: Enables monaural to stereo conversion

Bits [11:10]: **FRAMECYC [1:0] Frame Cycle Select (Only for master mode)**

Controls the number of frame cycles. This register bit is effective only in master mode.
In slave mode, an even number of frame cycles that is more than or equal to twice as much as the data width and less than or equal to 256fs can be used.

00: 32fs
01: 64fs
10: 128fs
11: 256fs

Bit 9: **CLKOUTEN Clock Output Enable (Only for master mode)**

Controls clock output.

0: Clock output disable
1: Clock output enable

Bit 8: **CLKSEL Clock Select**

Selects the clock source.

0: Uses SDI/WSI from outside (slave mode) or self-generated clock/word select (master mode).
1: Shares the clock and word select that the other channel uses.

This register bit is used to share I^2S clock and I^2S word select signals with the I^2S of the other channel.

Note: If you set this register bit to "1", set Bit 1 of the same register to "0" to enable slave mode. You cannot use the clock sharing function in master mode.

Bit 7: **SFTRST Software Reset (Write Only)**

Resets the FIFO data, shift register, and internal control circuit. This does not reset the clock generation circuit used in master mode. In receive mode, this does not reset the word select generation circuit, which is also used in master mode. In transmit mode, however, this resets the circuit so that the output becomes "1".

0: n/a
1: Software reset

Bits [6:5]: **DATAWIDTH [1:0] Data Bit Width**

Selects the bit width of the data.

00: 16 bits
01: 14 bits
10: 8 bits
11: Reserved

Bit 4: **MONO/ STEREO Monaural/Stereo Selection**

Selects a data type between stereo and monaural.

0: Stereo type
1: Monaural type

Bit 3: **DMAEN DMA Enable**

Selects whether to use DMA.

0: DMA Disable (Does not issue any DMA request)
1: DMA enable (Issues a DMA request to DMA controller 1)

Bit 2: **TX/RX Transmit/Receive Select**

Selects whether to use the I^2S for transmit mode or for receive mode.

0: Receive mode (Inputs data)
1: Transmit mode (Outputs data)

Bit 1: **MST/SLV Master/Slave Select**

Selects whether to use the I^2S for master mode or for slave mode.

0: Slave mode (Inputs clock and word select)
1: Master mode (Outputs clock and word select)

Bit 0: **I2SEN I^2S Enable**

Controls enable/disable of the I^2S module.

0: I^2S disable
1: I^2S enable

I^2S[1:0] Clock Division Register

I^2S0[0x04], I^2S1[0x44] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
CLKDIV [7:0]							
7	6	5	4	3	2	1	0

Bit[7:0]: **CLKDIV [7:0] Clock Division (Only for master mode)**

In master mode, sets the number of divisions for the output clock based on the source clock.

Number of divisions = (CLKDIV + 1) × 2

With this setting, the sampling frequency is obtained from the following expression.

Sampling frequency =
source clock frequency / (number of clock divisions × frame cycle)

In S2S65A00, the source clock frequency is the same as the system clock frequency.

I²S[1:0] Transmit/Receive Port Register

I²S0[0x08], I²S1[0x48] Default value = — Read/Write

TXD/RXD [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXD/RXD [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit[31:0]: **TXD/RXD [31:0] Transmit/Receive Port [31:0]**

In receive mode

Read: Reads I²S data accumulated in the FIFO.

Write: n/a

In transmit mode

Read: n/a

Write: Writes I²S data to the FIFO.

The amount of data that can be read or write at one time differs depending on the data width setting and the stereo/monaural setting.

16-bit stereo:

TXD/RXD[31:16] Right data
TXD/RXD[15:0] Left data

14-bit stereo:

TXD/RXD[31:30] 0 data
TXD/RXD[29:16] Right data
TXD/RXD[15:14] 0 data
TXD/RXD[13:0] Left data

8-bit stereo:

TXD/RXD[31:16] Invalid data
TXD/RXD[15:8] Right data
TXD/RXD[7:0] Left data

16-bit monaural:

TXD/RXD[31:16] Invalid data
TXD/RXD[15:0] Monaural data

14-bit monaural:

TXD/RXD[31:16] Invalid data
TXD/RXD[15:14] 0 data
TXD/RXD[13:0] Monaural data

8-bit monaural:

TXD/RXD[31:8] Invalid data
TXD/RXD[7:0] Monaural data

Note: A whole piece of sampling data must be read or written by one register access because the transmit/receive port register handles data on a sampling basis. For example, if you try to read 16-bit stereo data by 16-bit access, only the left data may be read and the right data may be overwritten by the next data. So, in this case, you should perform 32-bit access to read both the left data and the right data at one time.

I²S[1:0] Interrupt Status Register

I²S0[0x10], I²S1[0x50] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a		OVERFLOWFLG	UNDERFLOWFLG	NOTFULLFLG	NOTEMPTYFLG	FULLFLG	EMPTYFLG
7	6	5	4	3	2	1	0

This register represents a result that is obtained by bit ANDing of the raw interrupt status register (I2S0[0x14], I2S1[0x54]) and the interrupt enable register (I2S0[0x18], I2S1[0x58]).

Bit 5: **OVERFLOWFLG FIFO Overflow Interrupt Flag**

Indicates whether the FIFO has ever overflowed.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never overflowed.
Read 1: The FIFO has overflowed at least once.

Whether writing "1" to this bit clears the flag depends on modes.

Transmit mode: The flag is always cleared. Overflowing data is lost at the time when the overflow occurs. Data that is already in the FIFO remains there.
Receive mode: The flag is cleared if the write operation is performed after the FIFO has not been full for a period of one I2S clock. The flag is also cleared if the operation performed after executing a software reset.

Bit 4: **UNDERFLOWFLG FIFO Underflow Interrupt Flag**

Indicates whether the FIFO has ever underflowed.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never underflowed.
Read 1: The FIFO has underflowed at least once.

Whether writing "1" to this bit clears the flag depends on modes.

Transmit mode: The flag is cleared if the write operation is performed in one I²S clock later after writing data to the FIFO. The flag is also cleared if the operation performed after executing a software reset.
Receive mode: The flag is always cleared. The value of the data that is obtained in an underflow condition is not guaranteed.

Bit 3: **NOTFULLFLG FIFO Not Full Interrupt Flag**

Indicates whether the FIFO has ever been in the not full state.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never been in the not-full state.
Read 1: The FIFO has been in the not full state at least once.

Whether writing "1" to this bit clears the flag depends on the state of the FIFO.

When the FIFO is full: The flag is cleared.
When the FIFO is not full: The flag is not cleared.

Bit 2: **NOTEMPTYFLG FIFO Not Empty Interrupt Flag**

Indicates whether the FIFO has ever been in the not empty state.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never been in the not empty state.
Read 1: The FIFO has been in the not empty state at least once.

Whether writing "1" to this bit clears the flag depends on the state of the FIFO.

When the FIFO is empty: The flag is cleared.
When the FIFO is not empty: The flag is not cleared.

Bit 1: **FULLFLG FIFO Full Interrupt Flag**

Indicates whether the FIFO has ever been in the full state.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never been in the full state.
Read 1: The FIFO has been in the full state at least once.

Whether writing "1" to this bit clears the flag depends on the state of the FIFO.

When the FIFO is full: The flag is not cleared.
When the FIFO is not full: The flag is cleared.

Bit 0: **EMPTYFLG FIFO Empty Interrupt Flag**

Indicates whether the FIFO has ever been in the empty state.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never been in the empty state.
Read 1: The FIFO has been in the empty state at least once.

Whether writing "1" to this bit clears the flag depends on the state of the FIFO.

When the FIFO is empty: The flag is not cleared.
When the FIFO is not empty: The flag is cleared.

I^2S[1:0] Raw Interrupt Status Register

I^2S0[0x14], I^2S1[0x54] Default value = 0x0009 Read Only

n/a							
15	14	13	12	11	10	9	8
n/a		RAW OVERFLOWFLG	RAW UNDERFLOWFLG	RAW NOTFULLFLG	RAW NOTEMPTYFLG	RAWFULLFLG	RAW EMPTYFLG
7	6	5	4	3	2	1	0

Bit 5: **RAWOVERFLOWFLG Raw FIFO Overflow Interrupt Flag**

Indicates whether the FIFO has ever overflowed.

0: The FIFO has never overflowed.
1: The FIFO has overflowed at least once.

Bit 4: **RAWUNDERFLOWFLG Raw FIFO Underflow Interrupt Flag**

Indicates whether the FIFO has ever underflowed.

0: The FIFO has never underflowed.
1: The FIFO has underflowed at least once.

Bit 3: **RAWNOTFULLFLG Raw FIFO Not Full Interrupt Flag**

Indicates whether the FIFO has ever been in the not full state.

0: The FIFO has never been in the not full state.
1: The FIFO has been in the not full state at least once.

Bit 2: **RAWNOTEMPTYFLG Raw FIFO Not Empty Interrupt Flag**

Indicates whether the FIFO has ever been in the not empty state.

0: The FIFO has never been in the not empty state.
1: The FIFO has been in the not empty state at least once.

Bit 1: **RAWFULLFLG Raw FIFO Full Interrupt Flag**

Indicates whether the FIFO has ever been in the full state.

0: The FIFO has never been in the full state.
1: The FIFO has been in the full state at least once.

Bit 0: **RAWEMPTYFLG Raw FIFO Empty Interrupt Flag**

Indicates whether the FIFO has ever been in the empty state.

0: The FIFO has never been in the empty state.
1: The FIFO has been in the empty state at least once.

I^2S[1:0] Interrupt Enable Register

I^2S0[0x18], I^2S1[0x58] Default value = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a		OVERFLOW IRQEN	UNDERFLOW IRQEN	NOTFULL IRQEN	NOTEMPTY IRQEN	FULL IRQEN	EMPTY IRQEN
7	6	5	4	3	2	1	0

Bit 5: **OVERFLOWIRQEN FIFO Overflow Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has overflowed at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

Bit 4: **UNDERFLOWIRQEN FIFO Underflow Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has undeflowed at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

Bit 3: **NOTFULLIRQEN FIFO Not Full Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has been in the not full state at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

Bit 2: **NOTEMPTYIRQEN FIFO Not Empty Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has been in the not empty state at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

Bit 1: **FULLIRQEN FIFO Full Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has been in the full state at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

Bit 0: **EMPTYIRQEN FIFO Empty Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has been in the empty state at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

I²S[1:0] Current Status Register

I²S0[0x1C], I²S1[0x5C] Default value = 0x0009 Read Only

FIFOWPNTR [3:0]				FIFORPNTR [3:0]			
15	14	13	12	11	10	9	8
DMASTS	n/a			NOTFULLSTS	NOTEMPTYSTS	FULLSTS	EMPTYSTS
7	6	5	4	3	2	1	0

Bit [15:12]: **FIFOWPNTR [3:0] FIFO Write Pointer**

Indicates the current write pointer of the FIFO. (0x0 to 0xF)

Bit [11:8]: **FIFORPNTR [3:0] FIFO Read Pointer**

Indicates the current read pointer of the FIFO. (0x0 to 0xF)

The real address of the FIFO varies depending on the data size and the type (stereo or monaural).
The lower one bit of the pointer: 16-bit stereo, 14-bit stereo
The lower two bits of the pointer: 16-bit monaural, 14-bit monaural, 8-bit monaural
The lower three bits of the pointer: 8-bit monaural

Bit 7: **DMASTS DMA Status**

Indicates whether a DMA request is currently issued.

0: No DMA request is currently issued.
1: A DMA request is currently issued.

Bit 3: **NOTFULLSTS FIFO Not Full Current Status**

Indicates whether the FIFO is currently in the not full state.

0: The FIFO is currently full.
1: The FIFO is not currently full.

Bit 2: **NOTEMPTYSTS FIFO Not Empty Current Status**

Indicates whether the FIFO is currently in the not empty state.

0: The FIFO is currently empty.
1: The FIFO is not currently empty.

Bit 1: **FULLSTS FIFO Full Current Status**

Indicates whether the FIFO is currently in the full state.

0: The FIFO is not currently full.
1: The FIFO is currently full.

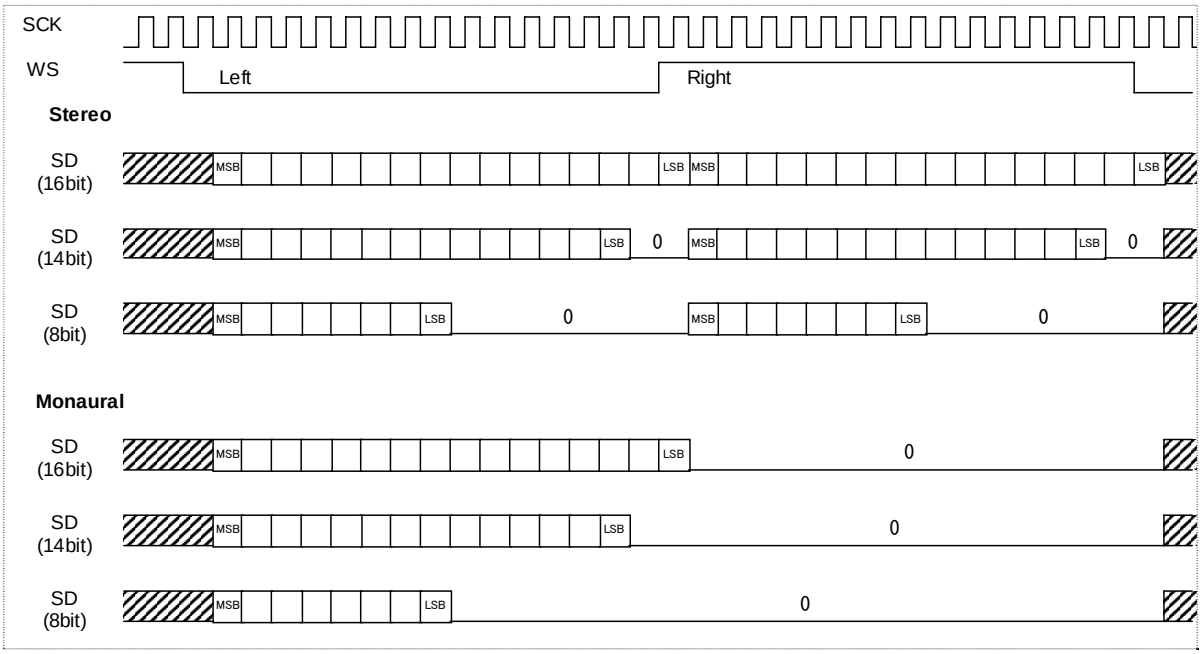
Bit 0: **EMPTYSTS FIFO Empty Current Status**

Indicates whether the FIFO is currently in the empty state.

0: The FIFO is not empty.
1: The FIFO is empty.

18.5 Functional description

18.5.1 I²S Timing Chart (32fs)



18.5.2 Data Width and Number of FIFO Stages

The amount of data that can be read or write at one time differs depending on the data width setting and the stereo/monaural setting.

16-bit stereo:
- TXD/RXD[31:16] Right data
- TXD/RXD[15:0] Left data

14-bit stereo:
- TXD/RXD[31:30] 0 data
- TXD/RXD[29:16] Right data
- TXD/RXD[15:14] 0 data
- TXD/RXD[13:0] Left data

8-bit stereo:
- TXD/RXD[31:16] Invalid data
- TXD/RXD[15:8] Right data
- TXD/RXD[7:0] Left data

16-bit monaural:
- TXD/RXD[31:16] Invalid data
- TXD/RXD[15:0] Monaural data

14-bit monaural:
- TXD/RXD[31:16] Invalid data
- TXD/RXD[15:14] 0 data
- TXD/RXD[13:0] Monaural data

8-bit monaural:
- TXD/RXD[31:8] Invalid data
- TXD/RXD[7:0] Monaural data

Note: A whole piece of sampling data must be read or written by one register access because the transmit/receive port register handles data on a sampling basis. For example, if you try to read 16-bit stereo data by 16-bit access, only the left data may be read and the right data may be overwritten by the next data. So, in this case, you should perform 32-bit access to read both the left data and the right data at one time.

The number of FIFO stages varies depending on the data width and the stereo/monaural setting.

Two-stage FIFO: 16-bit stereo, 14-bit stereo
Four-stage FIFO: 16-bit monaural, 14-bit monaural, 8-bit stereo
Eight-stage FIFO: 8-bit monaural

18.5.3 DMA Transfer

I^2S supports DMA transfer using the DMA controller 1. However, available channels will be 2/3. In transmit mode, a DMA request is asserted when the FIFO is not full; in receive mode, when the FIFO is not empty. A DMA request is negated at the point when the FIFO overflows. This is done for the I^2S in order not to perform unnecessary operations in an abnormal condition of FIFO overflow.

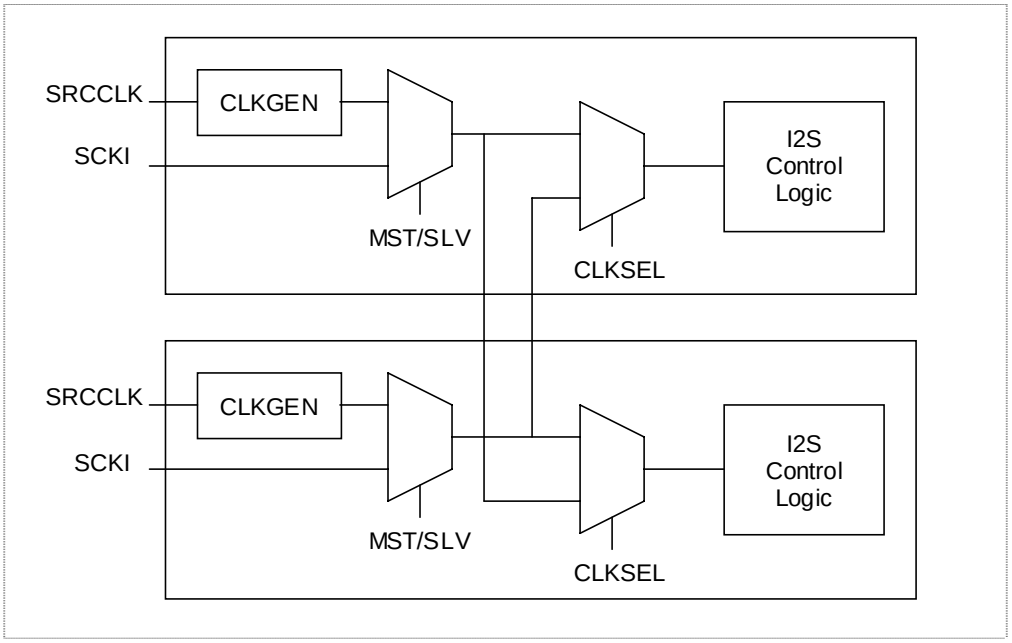
DMA transfer is enabled by setting Bit 3 of I2S0[0x00]/I2S1[0x40] to "1".

18.5.4 Interrupt to INTC

The I^2S, which has a two-channel configuration, transmits an interrupt to INTC (IRQ31) by ORing the two channels. To determine the interrupt cause, you should check both of the I^2S[1:0] interrupt status registers (I2S0[0x10], I2S1[0x50]).

18.5.5 Clock Select (Clock Sharing)

The clock to use can be shared by using the clock select function provided by Bit 8 of I2S0[0x00]/I2S1[0x40]. The following shows the configuration of sharing circuits:



The channel that starts to share the clock as a result of clock input from the other channel is no longer able to output clock in master mode. Therefore, the channel should be used in slave mode by setting Bit 1 of I2S0[0x00]/I2S1[0x40] to "1". The channel that provides clock can be used either in master mode or in slave mode.

Use this clock select function if you want to use the same clock to transmit/receive data by the I^2S simultaneously by using four signal pins.

18.5.6 Monaural-to-Stereo Conversion Function

Monaural data can be converted to stereo data before outputting it to a stereo type audio device. This is done by copying the L data of the monaural data, which only has L data, to R data.

Data is output as stereo data by setting Bit 12 of I2S0[0x00]/I2S1[0x40] to "1" in transmit mode.

Received stereo data cannot be converted to monaural data.

18.6 Setup example

The following is an I2S register setup example where an audio chip with transfer/receive functions is connected by using SCK, WS, SDI, and SDO to receive clock and word select (LR clock) from the audio chip. If the I/O of the I^2S and other I/Os are in sharing state, you must first configure the GPIO registers so that the I/O of the I^2S is enabled, before configuring this setup.

Setup Conditions for Channel 0

- Slave mode
- Transmit mode
- DMA enable
- 16-bit stereo
- 32fs

Setup Conditions for Channel 1

- Slave mode
- Receive mode
- DMA enable
- 16-bit stereo
- 32fs
- Clock sharing

Setup Procedure

```
I2S0[0x00] = 0x00000005    Slave mode, transmit mode, 16-bit stereo, 32fs
I2S0[0x18] = 0x00000030    Overflow and underflow interrupts enable
I2S0[0x00] = 0x00000085    Software reset
I2S0[0x00] = 0x0000000D    DMA enable
I2S1[0x40] = 0x00000101    Slave mode, receive mode, 16-bit stereo, 32fs, clock sharing
I2S1[0x58] = 0x00000030    Overflow and underflow interrupts enable
I2S1[0x40] = 0x00000181    Software reset
I2S1[0x40] = 0x00000109    DMA enable
```

19. SERIAL PERIPHERAL INTERFACE (SPI)

19.1 Description

The Serial Peripheral Interface (SPI) has a one-channel configuration.

The SPI supports operations in either master or slave mode and transfers data in bits 1 to 32. It can insert a delay ranging from 0 to 65535 clocks between each data transfer and can also generate an internal interrupt. It has data buffers for transmission and reception. The SPI provides four pins. The pin for SRDY# signal is internally fixed to LOW and cannot be used as an external pin.

19.1.1 Master Mode

An SPI set to master mode controls data transfer with a slave device connected to the SPI bus. An SPI supplies serial clock through the SCLK pin. Serial data is output from the MOSI pin and input to the MISO pin. An SPI has an SS (slave select) pin. Although this pin is not required for data transfer, it can be used for mode violation error detection. A mode violation error occurs when more than one device are set as the master at the same time in a multi-master SPI system. When an SPI in master mode detects that the SS pin becomes active level, a mode violation error interrupt is generated and the SPI is automatically reset to slave mode in order to avoid signal contention. If you do not have to detect mode violation errors, you can use the SS pin as a general I/O port.

To start a data transfer, enable the SPI (operable) and then write data to the transmit data register (TXD).

Fig.19.1 shows the control/operation flow in master mode.

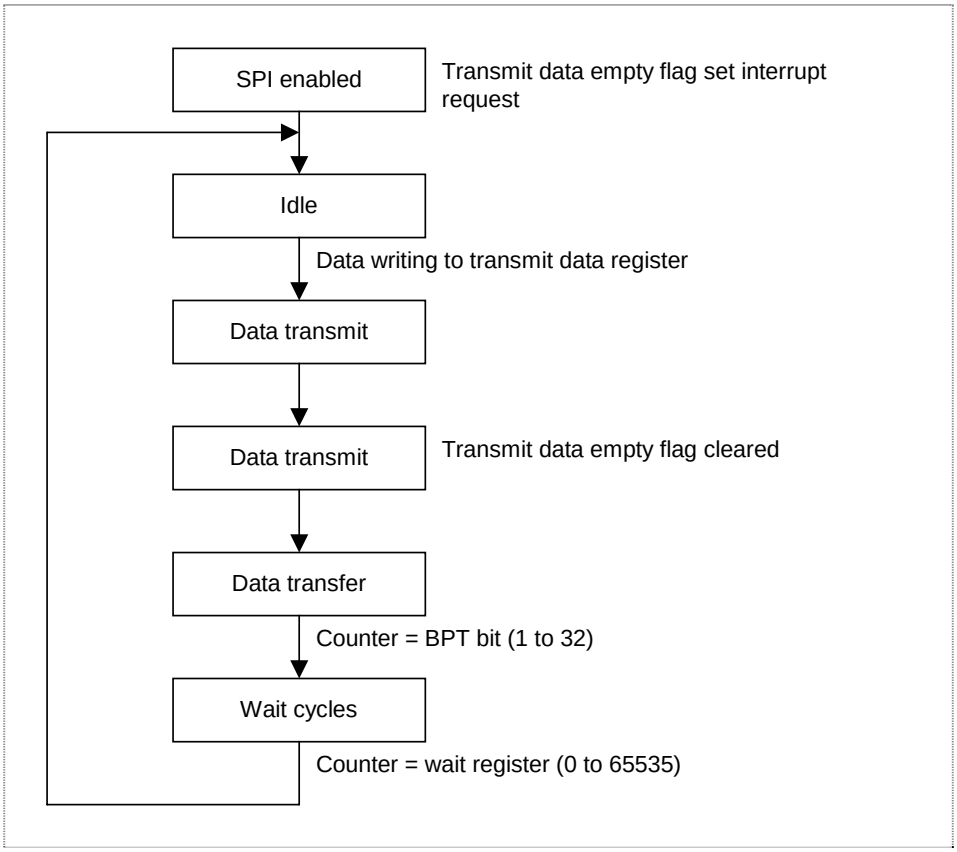


Fig.19.1 Transmission Flow in SPI Master Mode

19.1.2 Slave Mode

If an SPI is set to slave mode, an external SPI master controls data transfer with the SPI. A slave SPI receives operation clock from the external master at the SCLK pin and uses it. Serial data is input to the MOSI pin and output from the MISO pin. SS (slave select) pin is used for input.

Serial clock input and transfer operation are enabled as a result of the SS pin becoming active level.

After SPI is enabled (brought to operable), the external SPI master starts a transfer. The built-in counter that operates with SCLK clock controls transmission and reception for the specified number of transfer bits.

If more SCLK clock than the specified number of transfer bits is input, the transfer data is guaranteed only for the specified number of transfer bits.

Fig.19.2 shows the control/operation flow in slave mode.

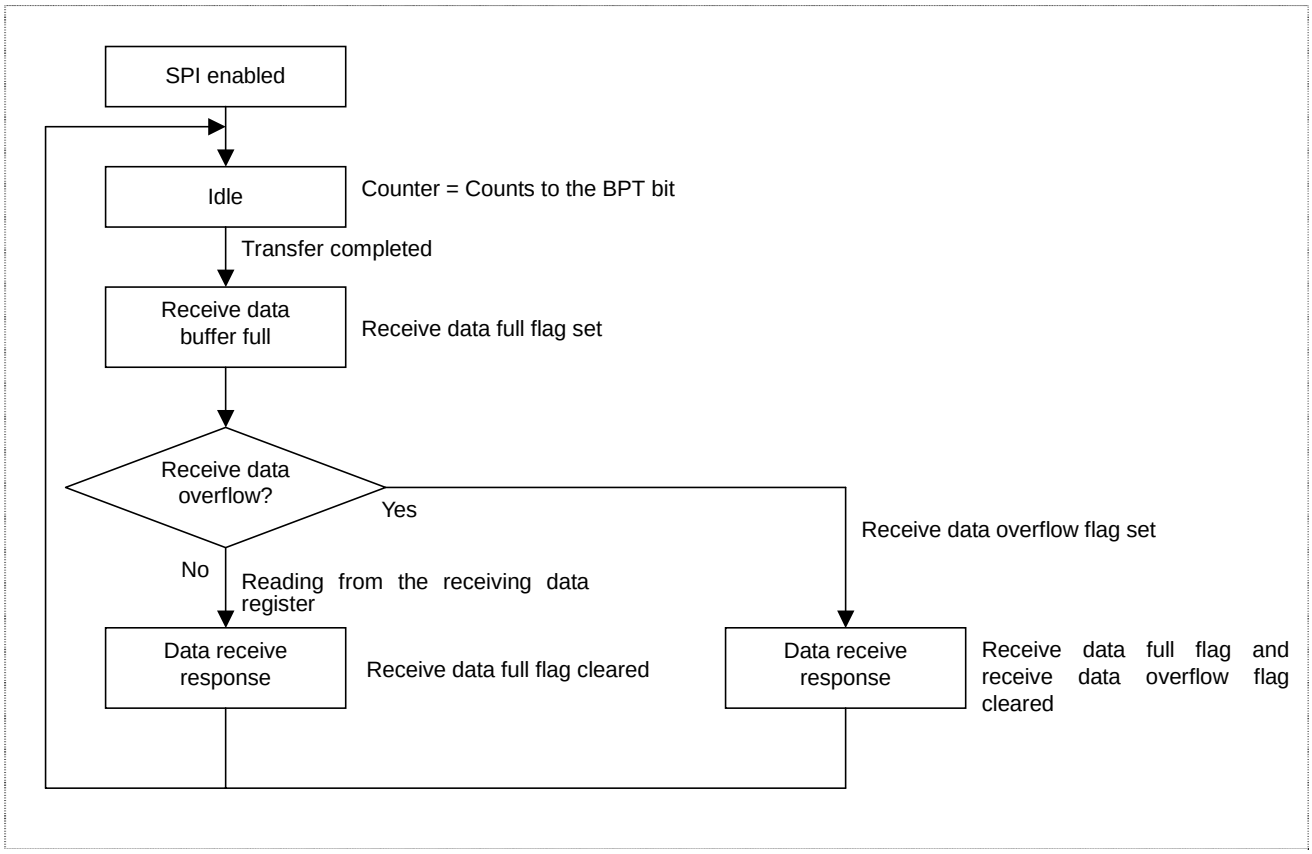


Fig.19.2 Reception Flow in SPI Slave Mode

19.2 Block Diagram

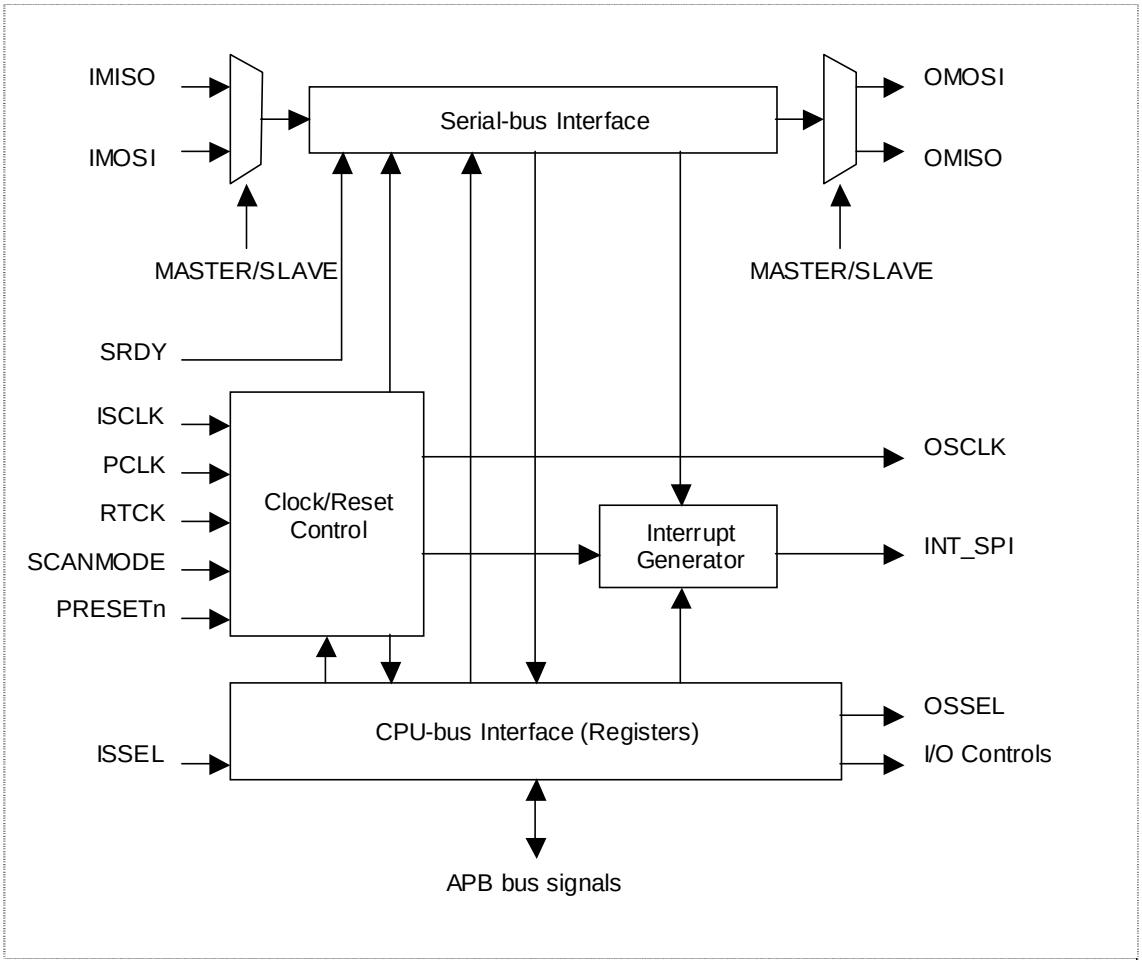


Fig.19.3 SPI Block Diagram

19.3 External Pins

The external pins related to the serial peripheral interface are as follows.

Pin Name	Input/Output	Pin Functions	Multiplex Pin/Remarks
SPI_SCLK	Input and output	SPI serial clock	GPIOC5*
SPI_SS	Input and output	SPI chip select	GPIOC4*
SPI_MISO	Input and output	SPI serial data master input/slave output	GPIOC6*
SPI_MOSI	Input and output	SPI serial data master output/slave input	GPIOC7*

Note (*): The external pins for SPI are multiplexed with GPIO pins or other pins. You can use the functions for SPI by setting “Function 2 of other than GPIO” through the GPIO pin function register.

19.4 Clock and Data Transfer Timing

If you use an SPI in master mode, use internal SCLK clock to make the shift register operate that is responsible for input and output of transfer data. There are four types of SCLKs to choose from that have different combination of clock phases and polarities.

You select a clock phase by using the CPHA bit (Bit 9 of the SPI control register 1). If the CPHA bit is set to "0", the output data changes at the falling edge of the clock (output from the shift register) and the input data is captured into the shift register at the rising edge of the clock (the bits in the sift register sequentially shift). Writing data to the transmit data register outputs MSB. If the CPHA bit is set to "1", the output changes at the rising edge and the input is captured at the falling edge. The MSB of the data is output at the first rising edge of the SCLK.

You select a clock polarity by using the CPOL bit (Bit 8 of the SPI control register 1). If the CPOL bit is set to "0", it means active high; "1", active low. In the above mentioned description of CPHA, the input and output timings assume that the clock is active high. If the CPOL bit is set to "1", the rising and falling edges are reversed. However, the timings of the edge trigger events inside the SIP are not reversed.

Fig.19.4 illustrates SCLK clock waveform for each option. This flexibility allows the interface to support most of the commercially available serial peripheral devices.

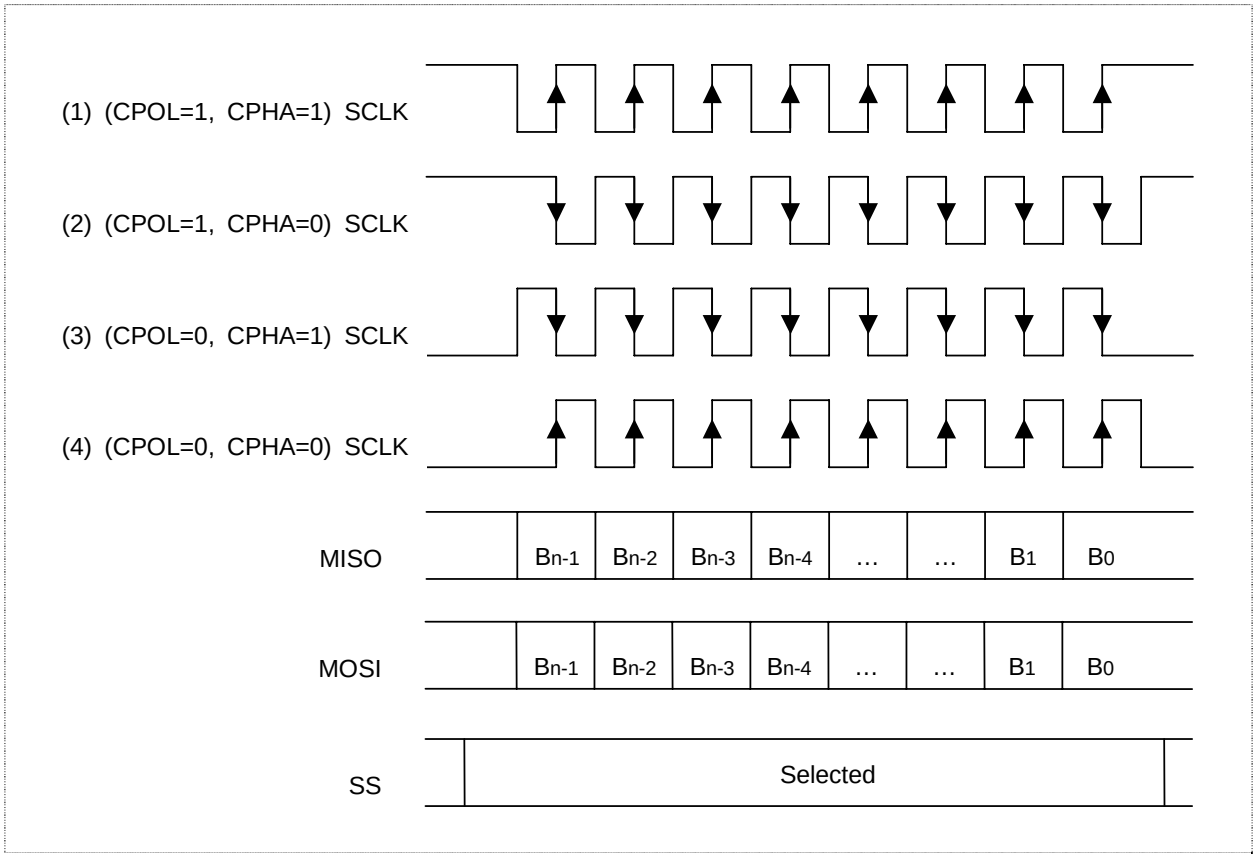


Fig.19.4 SPI Master Mode Clock Settings (where the number of bits of the transfer data is n)

Fig.19.5 illustrates SCLK clock waveform of SPI Slave mode.

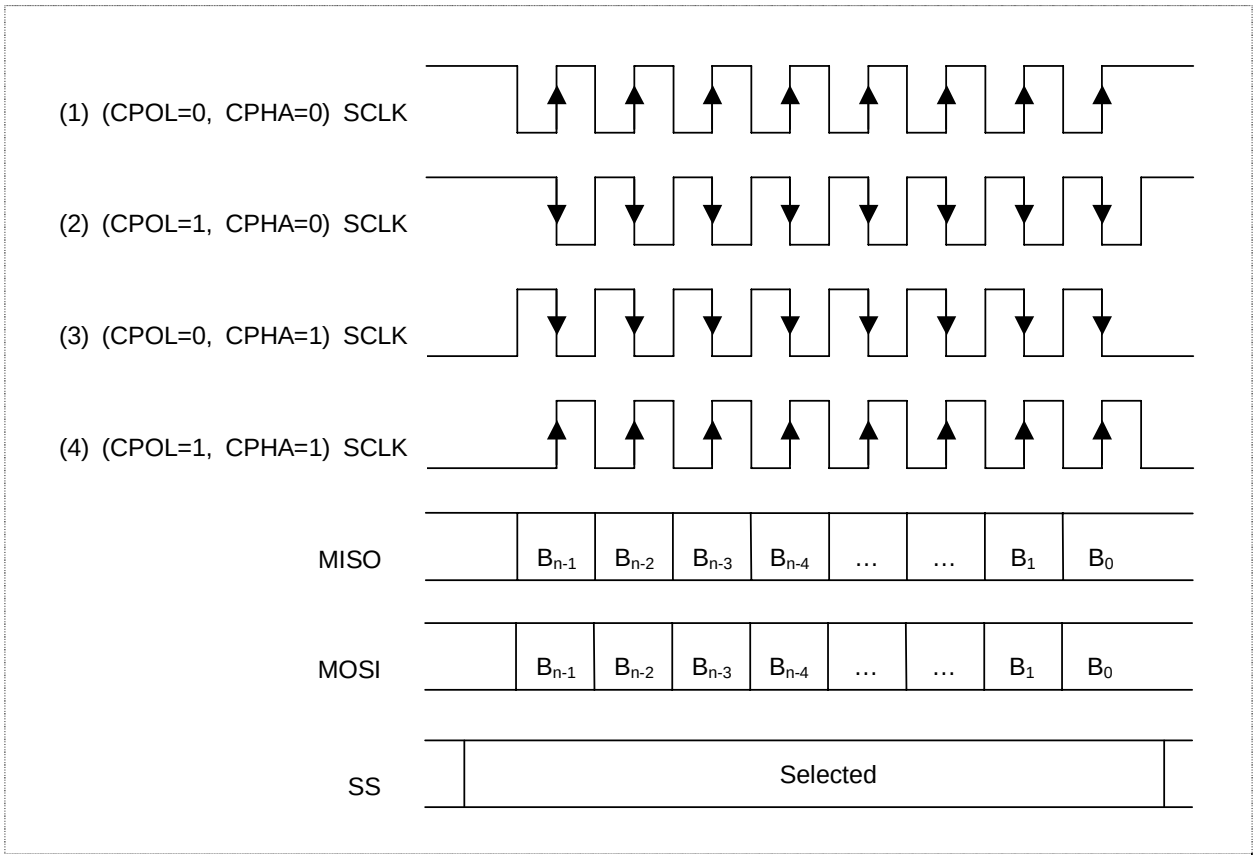


Fig.19.5 SPI Slave Mode Clock Settings (where the number of bits of the transfer data is n)

19.5 Description of Registers

19.5.1 List of Registers

The default base address where the SPI control registers are located is 0xFFFF_2000.
If not otherwise specified, the default value of any register bit that is not reserved is "0".

Table 19.1 List of SPI Registers (Base Address: 0xFFFF_2000)

Address Offset	Register Name	Default Value	R/W	Data Access Size
SPI Control Register				
0x00	SPI Receiving Data Register	0x0000_0000	RO	32
0x04	SPI Sending Data Register	0x0000_0000	R/W	32
0x08	SPI Control Register 1	0x0000_0000	R/W	32
0x0C	SPI Control Register 2	0x0000_0000	R/W	32
0x10	SPI Wait Register	0x0000_0000	R/W	32
0x14	SPI Status Register	0x0000_0010	RO	32
0x18	SPI Interrupt Control Register	0x0000_0000	R/W	32

19.5.2 Detailed Description of Registers

SPI Receiving Data Register

SPI[0x00] Default value = 0x0000_0000 Read Only

Receive Data [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Receive Data [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **Receive Data Bits [31:0]**

Data received from an external serial peripheral device can be read.

SPI Sending Data Register

SPI[0x04] Default value = 0x0000_0000 Read/Write

Transmit Data [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Transmit Data [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **Transmit Data Bits [31:0]**

This is a buffer to which transmit data is written. Data can be written when the TDEF bit (Bit 4 of the SPI status register) is set to "1", which indicate that this register is empty.

SPI Control Register 1

SPI[0x08] Default value = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
N/a	BPT [4:0]					CPHA	CPOL	n/a	MCBR [2:0]			CLKS	RX RAW	Mode	ENA
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [14:10]: **BPT[4:0] Number of transfer bits**

Specifies the bit length of the data transmitted/received in one transfer.

00000: 1 bit
00001: 2 bits
:
11110: 31 bits
11111: 32 bits

Bit 9: **CPHA Serial Clock Phase Select**

Selects the phase of the SCLK clock.

0: Generates a clock pulse in the last half of the data cycle (See Fig.20.4. (2)(4))
1: Generates a clock pulse in the first half of the data cycle (See Fig.20.4. (1)(3))

Bit 8: **CPOL Serial Clock Polarity Select**

Selects the polarity of the SCLK clock.

0: Active HIGH (Generates a HIGH pulse as clock) (See Fig.20.4. (3)(4))
1: Active LOW (Generates a LOW pulse as clock) (See Fig.20.4. (1)(2))

Bits [6:4]: **MCBR [2:0] Master Clock Bit Rate Select**

Sets the SCLK clock rate in master mode. Using the setting of this bit, the dividing ratio of the source clock (bus clock) is obtained from the following expression.

Dividing ratio = $4 * 2^{MCBR[2:0]}$

Therefore the SPI master clock is:

Master clock frequency (f_{SCLK}) = bus clock frequency/$(4 * 2^{MCBR[2:0]})$

Note: In slave mode, and in master mode as well, if you select a real time clock (32.768KHz) as the source clock (Bit 3 of this register is set to "1"), this bit is ignored.

Bit 3: **CLKS Source Clock Select**

Selects the source clock to generate SCLK clock in master mode.

0: Bus clock
1: Real time clock (32.768KHz)

Bit 2: **RXDATA RAW**

0: RXDATA is masked by the BPT width.
1: RXDATA is unmasked data of the shift register.

Bit 1: **Mode SPI Mode Select**

Selects whether to use this interface in master mode or in slave mode.

0: Slave Mode
1: Master Mode

Bit 0: **ENA SPI Enable**

Enables the transmit/receive circuits of the SPI.

0: Disable
1: Enable

SPI Control Register 2

SPI[0x0C] Default value = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a				SSA	SS	SSP	SSC	n/a					予約 (0)		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 11: **SSA Slave Select Pin (SS) Auto Control**

This bit sets the following when operating in master mode.

0: The SS pin is controlled by SS (Bit10) if the pin is set to output.

1: The SS pin is controlled by internal transfer timing if the pin is set to output.

This bit has no effect when operating in slave mode.

Bit 10: **SS Slave Select Pin (SS) Control**

Controls the output of the SS pin if the pin is set to output when operating in master mode.

0: The SS pin outputs inactive level.

1: The SS pin outputs active level.

If the SS pin is set as invalid input (SSC = 0) when operating in slave mode:

0: This SPI is not selected.

1: This SPI is selected.

In other cases, this bit has no effect.

Bit 9: **SSP Slave Select Pin (SS) Polarity Select**

0: Active LOW

1: Active HIGH

Bit 8: **SSC Slave Select Pin (SS) Setting**

Switches the input/output direction of the SS pin when operating in master mode.

0: Input (Mode violation detection)

1: Output (Slave select output)
HIGH/LOW output can be set by using the SS bit (Bit 10 of this register).
Mode violation cannot be detected.

In slave mode:

0: Sets the SS pin as invalid input. Selection in the SS bit (Bit 10) becomes effective.

1: Sets the SS pin as valid input.

The following table summaries the settings in Bits [10:8].

Table 19.2 Settings in the SS Pin

Mode selected *	Bit 8: SSC SS Setting	Bit 11: SSA SS Auto Select	Bit 9: SSP SS Polarity Select	Bit 10: SS SS Control	State of SS Pin (Active Level)
Master Mode	0: SS pin is for input (Mode violation detection)	No effect	0: Active LOW	No effect	SS input (LOW)
			1: Active HIGH		SS input (HIGH)
	1: SS pin is for output	0: Controlled by SS the bit	0: Active LOW	0: Inactive	SS output = HIGH (LOW)
				1: Active	SS output = LOW (LOW)
			1: Active HIGH	0: Inactive	SS output = LOW (HIGH)
				1: Active	SS output = HIGH (HIGH)
		1: Auto control	0: Active LOW	No effect	SS output = auto control (LOW)
			1: Active HIGH		SS output = auto control (HIGH)
Slave Mode	0: SS pin as invalid input	No effect	No effect	0: Not Selected	SS input (HIGH) <Not selected>
				1: Selected	SS input (HIGH) <Selected>
	1: SS pin as valid input		0: Active LOW	No effect	SS input (LOW)
			1: Active HIGH		SS input (HIGH)

*: You select a mode by using the Mode bit (Bit 1 of the SPI control register 1).

Bits [2:0]: **Reserved (0)**

SPI Wait Register

SPI[0x10] Default value = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WAIT Cycles [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [15:0]: **WAIT Cycles [15:0] WAIT Cycles**

Sets the wait time to insert between each data transmit/receive operation in the unit of SCLK clock. (Wait time = WAIT Cycles [15:0] × SCLK cycles)

0000h (w): 0 clock
0001h (w): 1 clock
0002h (w): 2 clocks
:
FFFFh (w): 65535 clocks

The setting of this bit is effective only in master mode.

SPI Status Register

SPI[0x14] Default value = 0x0000_0010 Read Only

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a									BSYF	MFEF	TDEF	RDOF	RDFF	n/a	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 6: **BSYF Transfer Busy Flag**

Indicates the SPI is performing a transmit/receive operation.

0: Standby

1: Transmission or reception is in progress.

This flag is automatically set when a transmission or reception starts. The set flag is automatically cleared when the transfer completes and the interface becomes standby.

This flag is effective only in master mode. In slave mode, it always remains "0".

Bit 5: **MFEF Mode Violation Flag**

Indicates that a mode violation error occurs.

0: No error exists

1: Error occurred

This flag is set if the SS pin is brought to active level by an external serial device when the SPI is in master mode. This flag is automatically cleared by canceling the error. To cancel the error, the SPI operates in slave mode while the flag and the MFIE bit (Bit 5 of the SPI interrupt control register) are set and disables all output without starting data transfer.

Bit 4: **TDEF Transmit Data Empty Flag**

Indicates that the sending data register is empty.

0: Transmit data exists

1: No transmit data exists (Default)

This flag is set when the data written to the sending data register is sent to a serial interface (or at reset). Owing to this function, the next transmit data can be written to the sending data register and the set flag is cleared by writing data to the sending data resister.

Bit 3: **RDOF Receive Data Overflow Flag**

Indicates that a receive data overflow occurs.

0: No overflow occurred

1: Overflow occurred

This flag is set when the next receive data is sent to the receive data register from a serial interface while the receive data full flag is set (the receive data is not read). The set flag is cleared by reading the receive data register.

Bit 2: **RDFF Receive Data Full Flag**

Indicates that there is receive data in the receive data register.

0: No receive data exists

1: Receive data exists

This flag is set when receive data is sent to the receive data register through the serial interface. The set flag is cleared by reading the receive data register.

Note: This register is also cleared entirely when the SPI Enable bit (Bit 0 of the SPI control register 1) is set to "0" to disable the SPI.

SPI Interrupt Control Register

SPI[0x18] Default value = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a										MFIE	TEIE	ROIE	RFIE	MIRQ	IRQE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 5: **MFIE Mode Violation Interrupt Enable**

Enables (permits)/disables (prohibits) a mode violation interrupt.

0: Interrupt disable
1: Interrupt enable

This setting is effective when the SPI operates in master mode and the SS pin is configured for mode violation detection.

Bit 4: **TEIE Transmit Data Register Empty Interrupt Enable**

Enables/disables a transmit data register empty interrupt.

0: Interrupt disable
1: Interrupt enable

Bit 3: **ROIE Receive Data Overflow Error Interrupt Enable**

Enables/disables a receive data overflow error interrupt.

0: Interrupt disable
1: Interrupt enable

Bit 2: **RFIE Receive Data Register Full Interrupt Enable**

Enables/disables a receive data full interrupt.

0: Interrupt disable
1: Interrupt enable

Bit 1: **MIRQ Manual Interrupt Request Set/Clear**

Sets/Clears a manual interrupt request of the SPI.

0: Clears the interrupt request.
1: Sets the interrupt request.

This bit allows the software to generate an SPI interrupt. If IRQE (Bit 0) is set to "0 (interrupt is prohibited)", controls by this bit has no effect.

Bit 0: **IRQE Interrupt Request Enable**

Enables/disables an SPI interrupt request.

0: Interrupt request disable
1: Interrupt request enable

20. Compact Flash (CF) Card Interface

20.1 Overview

The CF card interface has the following features.

- CF card attribute memory space (2KB space)
- CF card common memory space (2KB space)
- CF card IO space (2KB space)
- Support of interrupt output (STSCHG#, IRQ)
- Command strobe timing output in various ranges of internal PCLK clocks (50 MHz to 6 MHz)
- Support of programmable idle cycle insertion and programmable command cycle insertion for CFIORD#,CFIOWR#
- Support of True IDE mode on CF interface (if CFOE# pull-down circuit and CSSEL signal are used, and Low Active Reset is externally supported)

Note: The S2S65A00 does not have the following signal lines due to the limited number of pins. They are internally fixed to logical LOW.

- **CD [2:1]#**
- **VS [2:1]#**
- **BVD2#**
- **WP/IOIS16#**

20.2 Block Diagram

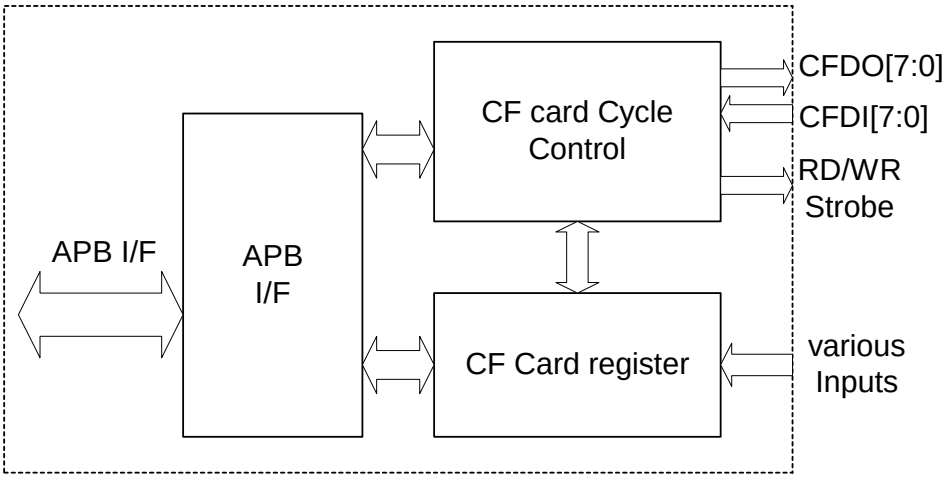


Fig.20.1 CF Card I/F Block Diagram

20.3 CF Card I/F Space Assignment

Table 20.1 CF Card I/F Space Assignment

Description	Address Range	Size
CF card attribute space	0xFFFE4000 - 0xFFFE47FF	2KB
CF card common space	0xFFFE4800 - 0xFFFE4FFF	2KB
CF card I/O space	0xFFFE5000 - 0xFFFE57FF	2KB
CF card true IDE CS1# space	0xFFFE5800 - 0xFFFE5BFF	1KB
CF card true IDE CS2# space	0xFFFE5C00 - 0xFFFE5FFF	1KB
CF card interface setting	0xFFFE6000 - 0xFFFE6FFF	4KB

Caution: Either 8-bit or 16-bit data of each CF card space must be accessed. An access to 32-bit data of the card causes an abnormal operation.
The 16- or 32-bit data access must be used for the register access during CF card interface setting.

To allow true IDE access:

The True IDE access via the CF I/F of the S2S65A00 is not realized by access to the specified space only. The signals must be operated at the system level (on the board).

Actually, the following two signals must be used correctly.

- OE# (Alias: ATASEL)
- CSSEL

Specify the OE# signal of CF Card I/F to be sampled as logical Low at Power-On Reset time (the power must be switched from OFF to ON state).

Furthermore, be sure to set the CSSEL signal to Pull Up or Pull Down status or to the Open status when determining if the IDE device operates as the master or a slave unit.

During True IDE access, the CS1# and CS2# spaces have the following addresses.

- The accessible registers of CS1# are the address space having low-order 3 bits of 0x0 to 0x7 shown on the above table.
- The accessible registers of CS2# are the address space having low-order 3 bits of 0x6 to 0x7 shown on the above table.
  For example, the Alternate Status register has been assigned the address space having low-order 3 bits of 0x7 within the CS2# space.

20.4 External Pins

The following defines the external pins of the CF card interface.

Table 20.2 External Pins (CF)

Pin Name	Input/Output	Pin Functions	Multiplex Pin/Remarks
CFCE2#	Output	CF card enable 2 (CE2#) output	GPIOH0*
CFCE1#	Output	CF card enable 1 (CE1#) output	GPIOH1*
CFIORD#	Output	CF I/O read strobe output	GPIOH2*
CFIOWR#	Output	CF I/O write strobe output	GPIOH3*
CFWAIT#	Input	Wait request input from CF card	GPIOH4*
CFRST	Output	Reset output to CF card	GPIOH5*
CFIRQ	Input	Interrupt request input from CF card	GPIOH6*
CFSTSCHG#	Input	Status change input from CF card	GPIOH7*
CFDEN#	Output	CF card external buffering data enable output	GPIOI0*
CFDDIR	Output	CF data bus direction indication output	GPIOI1*
CFREG#	Output	CF attribute space and I/O space selection REG signal	MA11**
CFADDR [10:0]	Output	CF address signal	MA [10:0]**
CFDATA [15:0]	Input/Output	16-bit CF data signal	MD [15:0]**
CFOE#	Output	CF interface memory and attribute space output enable signal	MOE#**
CFWE#	Output	CF interface memory and attribute space write enable signal	MWE0#**

* As the CF external pin is multiplexed with the GPIO pin, this function can be used if “Function 1 other than GPIO” is set using the GPIO Pin Function register.

** When the CF is operating, the memory controller pin functions as the CF external pin.

20.5 Registers

20.5.1 List of Registers

The CF card interface registers have base address 0xFFFE_6000.

Table 20.3 Register List **(Base address 0xFFFE_6000)**

Address Offset	Register Name	Abbreviation Name	Default Value	R/W	Data Access Size
0x00	CF Card Interface Control Register	CFCTL	0x1000	(R/W)	16 (/32)
0x04	CF Card Pin Status Register	CFPINSTS	0x0XXX	RO	16 (/32)
0x08	CF Card IRQ Source & Clear Register	CFINTRSTS	0x0XXX	R/W	16 (/32)
0x0C	CF Card IRQ Enable Register	CFINTMSTS	0x0000	R/W	16 (/32)
0x10	CF Card IRQ Status Register	CFINTSTS	0x0000	RO	16 (/32)
0x14	CF Card MISC Register	CFMISC	0x0000	R/W	16 (/32)

20.5.2 Detailed Description of Registers

CF Card Interface Control Register (CFCTL)

CF[0x00] Default value = 0x1000 (Read/Write)

PROG CYCEN	PROG IDLE [2:0]			PROG CYC [3:0]			
			R/W				
15	14	13	12	11	10	9	8
Reserved (0)	IOIS8_IO	IOIS8_MEM	PROG IDLE EN	CFRST	CFCARDEN	PCKMD[1:0]	
RO				R/W			
7	6	5	4	3	2	1	0

Bit 15: **PROG_CYCEN**

Enables the programmable command cycle function for CFIORD#/CFIOWR# signals.
Usually, it must be disabled.

0: Disable (After RESET/ Default)
1: Enable

Bits [14:12]: **PROG_IDLE[2:0]**

Set the idle cycle count of CF card interface. (Default is logical 1.)

Bits [11:8]: **PROG_CYC[3:0]**

Set the command active cycle count of CFIORD#/CFIOWR# signals.
Valid if PROG_CYCEN=1 only.

Bit 7: **Reserved**

Bit 6: **IOIS8_IO**

Sets the device size of CF card I/O space.

0: Allows operating as a 16-bit CF card device.
1: Allows operating as an 8-bit CF card device.

Bit 5: **IOIS8_MEM**

Sets the device size of CF card common memory space.

0: Allows operating as a 16-bit CF card device.
1: Allows operating as an 8-bit CF card device.

Bit 4: **PROG_IDLE_EN**

Enables the programmable command cycle function for CFIORD#/CFIOWR# signals until commands from CFCE1# ・CFCE2# are made active. Usually, it must be disabled.

0: Disable
1: Enable

Bit 3: **CFRST**

Allows the direct control of CFRST pin if the CF I/F function is selected.

0: Sets the CFRST pin to Low.
1: Sets the CFRST pin to High.

Bit 2: **CFCARDEN**

0: Disables the CF card interface.
1: Enables the CF card interface.

Bits [1:0]: **PCKMD[1:0]**

Allows the value change so that the CF card I/F operates according to the clock frequency.

00: Use this value if the PCLK is approximately 50 MHz (-25 MHz).
01: Use this value if the PCLK is approximately 24 MHz.
10: Use this value if the PCLK is approximately 12 MHz.
11: Use this value if the PCLK is approximately 6 MHz.

CF Card Pin Status Register (CFPINSTS)

CF[0x04] Default value = 0x0XXX — Read Only

15	14	13	12	11	10	9	8
0							IREQ#2
WP	IREQ#1	BVD2#	BVD1#/ STSCHG	VS2#	VS1#	CD2#	CD1#
7	6	5	4	3	2	1	0

The pin status value (except for bits 1 and 0) is reflected on each bit.
Bits [1:0] (CD2#, CD1#) The pin status after signal noise filtering is indicated.

Bit 8: **IREQ#2 input**

This bit reference is useful for interrupt input if the IREQ# pin is active high.
This bit indicates the IREQ# pin status of CF card interface.

Bit 7: **WP pin input**

Indicates the WP pin status of CF card interface.

Bit 6: **RDY/BSY, IREQ pin input (The name changes depending on the mode selected for the CF card.)**

This bit reference is useful for interrupt input if the IREQ# pin is active low.
The inverted input status of RDY/BSY or IREQ pin of CF card interface can be checked.

Bit 5: **BVD2# pin input**

Indicates the BVD2# pin status of CF card interface.

Bit 4: **BVD1#/STSCHG# pin input**

Indicates the BVD1#/STSCHG# pin status of CF card interface.

Bit 3: **VS2# pin input**

Indicates the VS2# pin status of CF card interface.

Bit 2: **VS1# pin input**
Indicates the VS2# pin status of CF card interface.

Bit 1: **CD2# pin input**
Indicates the CD2# pin status of CF card interface after noise filtering.

Bit 0: **CD1# pin input**
Indicates the CD1# pin status of CF card interface after noise filtering.

Note: Bits 7, 5, and 3 to 0 are not supported by the S2S65A00.

CF Card IRQ Source & Clear Register (CFINTRSTS)
CF[0x08] Default value = 0x0XXX Read/Write

Reserved							IRQ#2
15	14	13	12	11	10	9	8
Reserved	IRQ#1	Reserved	BVD1/ STSCHG	Reserved		CD2	CD1
7	6	5	4	3	2	1	0

This register is used to indicate the interrupt request source (not masked), and if value 1 is written in it, the respective bit is cleared.

Bits [15:7]: **Reserved bits**
Zeros (0s) are read. Zeros (0s) must be entered during writing.

Bit 8: **IRQ#2 input**
This bit reference is useful for interrupt input if the CFIRQ pin is active high.
After the CF card has been programmed to the I/O mode, if this bit is logical 1, it indicates an occurrence of CFIRQ interrupt request. If this bit is logical 0, no interrupt request has occurred at the CFIRQ pin.

Bit 6: **IRQ#1 (Inverting input)**
This bit reference is useful for interrupt input if the CFIRQ pin is active low.
After the CF card has been programmed to the I/O mode, if this bit is logical 1, it indicates an occurrence of CFIRQ interrupt request. If this bit is logical 0, no interrupt request has occurred at the CFIRQ pin.

Bit 4: **BVD1/STSCHG#**
After the CF card has been programmed to the I/O mode, if this bit is logical 0, it indicates that STSCHG# is logical 0, that is, the RDY/BSY# or WP signal status has been changed. If this bit is logical 1, the RDY/BSY# or WP signal status has not changed.

Bit 1: **CD2 pin status change**
If this signal (after noise filtering) status changes from Low to High or from High to Low, it is set to logical 1. This is one of interrupt signal sources, and the signal is cleared if this bit is set to 1.

Bit 0: **CD1 pin status change**
If this signal (after noise filtering) status changes from Low to High or from High to Low, it is set to logical 1. This is one of interrupt signal sources, and the signal is cleared if this bit is set to 1.

Note: Bits 1 and 0 are not supported by the S2S65A00.

CF Card IRQ Enable Register (CFINTMSTS)

CF[0x0C] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
Reserved							IRQEN#EN2
7	6	5	4	3	2	1	0
Reserved	IREQ#EN1	Reserved	BVD1EN/ STSCHGEN	Reserved		CD2EN	CD1EN

Each bit corresponds to the respective bit of CF Card IRQ Source & Clear register.
Each bit is set as follows.

0: Masks an interrupt.
1: Unmasks (or enables) an interrupt.

Note: Bits 1 and 0 are not supported by the S2S65A00.

CF Card IRQ Status Register (CFINTSTS)

CF[0x10] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
Reserved							IREQ#2
7	6	5	4	3	2	1	0
Reserved	IREQ#1	Reserved	BVD1/ STSCHG	Reserved		CD2	CD1

The respective bits of the IRQ Source register and IRQ Enable register are ANDed and set in this register.

0: No interrupt has occurred, or the interrupt is masked.
1: The interrupt is enabled and an interrupt cause already exists.

Note: Bits 1 and 0 are not supported by the S2S65A00.

CF Card MISC Register (CFMISC)

CF[0x14] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CSRDEN

This register is reserved for the certain hardware. Do not write any value in usual application.

Bits [15:2]: **Reserved bits**

Bit 0: **CSRDEN**

Activates both CFCE1# and CFCE2# or not during reading.

0: Normal operation
1: Activates both CFCE1# and CFCE2# during reading.

20.6 Application Restrictions of CF Card Interface

This device does not have the following pins due to the limited number of mount pins, and those pin signals are internally fixed to Low. Therefore, the application of some registers is restricted.

- CD [2:1]#
- VS [2:1]#
- BVD2#
- WP/IOIS16#

The following lists the registers that are restricted to use for this device.

Offset address	Register bit name	Restrictions
CF[0x04] Bit 7	WP pin input	Unavailable
CF[0x04] Bit 5	BVD2# pin input	Unavailable
CF[0x04] Bit 3	VS2# pin input	Unavailable
CF[0x04] Bit 2	VS1# pin input	Unavailable
CF[0x04] Bit 1	CD2# pin input	Unavailable
CF[0x04] Bit 0	CD1# pin input	Unavailable
CF[0x08] Bit 1	CD2 pin status change	Unavailable
CF[0x08] Bit 0	CD1 pin status change	Unavailable
CF[0x0C] Bit 1	CD2EN	Unavailable
CF[0x0C] Bit 0	CD1EN	Unavailable
CF[0x10] Bit 1	CD2	Unavailable
CF[0x10] Bit 0	CD1	Unavailable

21. SD Memory Card Interface

21.1 Overview

The SD memory card interface conforms the SD Memory Card Physical Layer Specification Version 1.01.

The customer is requested to “acknowledge the membership in the SDA or MMCA.” The “SD memory card interface” specifications are provided after this confirmation. If the customer is not a member of SDA or MMCA, no specification publication is provided.

(SDA = SD Card Association; MMCA = MultiMedia Card Association)

22. Timer A (TIMA)

22.1 Overview

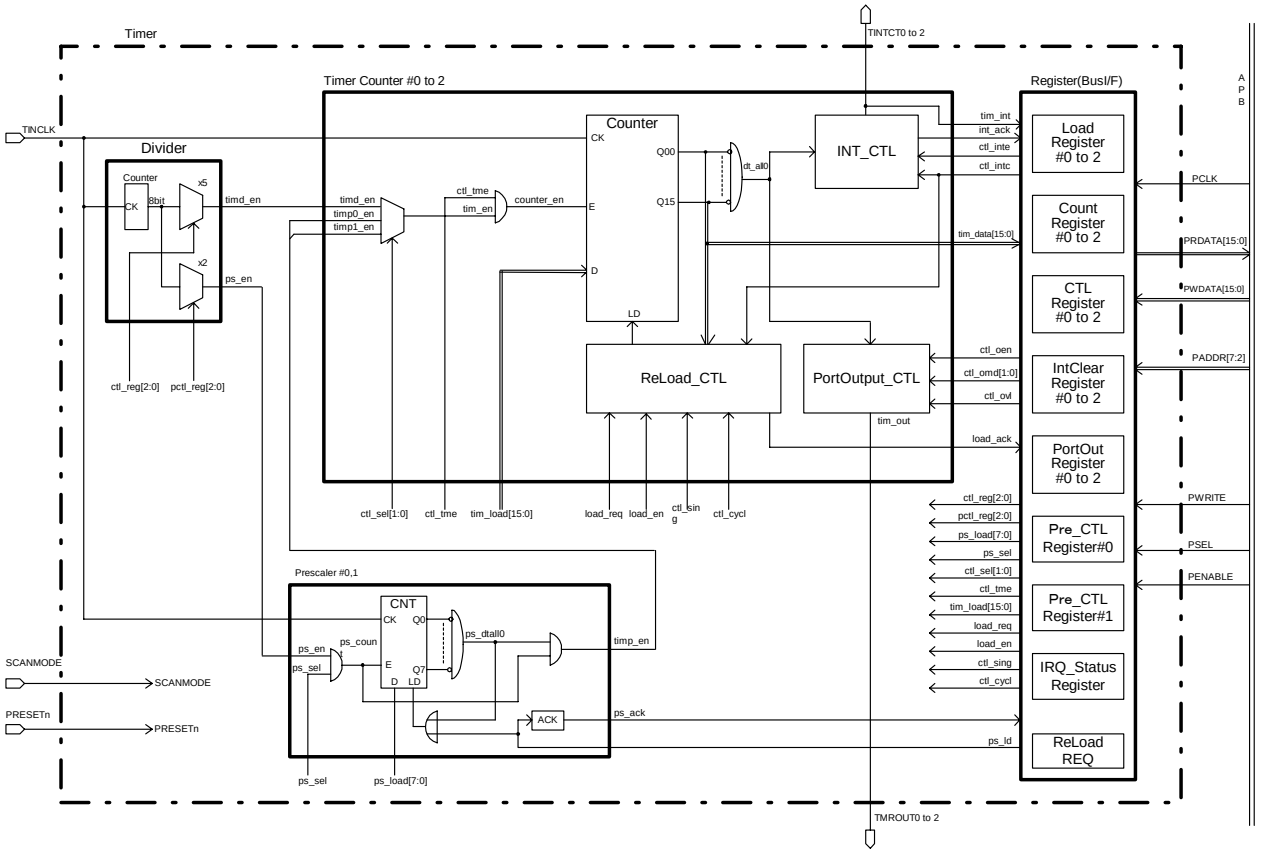
Timer A has the following features.

- Built-in 3-channel, 16-bit down count timer (The 3-channel timers have the same structure.)
- Supports two types of timer mode (Cyclic mode and Single mode).
- Generates an interrupt with unmasked IRQ each time the timer count reaches zero (0).
- Built-in 8-bit counter for signal dividing (from 1/1 to 1/256)
- Allows setting any counting of clocks (divided by the divider) using the 8-bit counter for 2-channel prescaler.
- Timer output mode functions during signal underflow (3 types):
  Underflow signal, any value-0 or 1 output, and underflow frequency toggle output

22.2 Block Diagram

The following shows a block diagram of Timer A. The timer block consists of the Register block (Register (Bus I/F)), Divider block (Divider), 2-channel Prescaler block (Prescaler #0, 1), and 3-channel Timer Counter block (Timer Counter #0 to 2).

As every prescaler and timer counter channels are the same, the typical structure of channel 0 is only shown in the following block diagram. All channels are controlled by the divider and registers. The clock for timer counter (TINCLK) is supplied by the system controller, and the PCLK clock divided by 8 (the 1/8 PCLK) is used. For details, see the System Controller section.



＜16bit TIMER BLOCK＞

Fig.22.1 Block Diagram

22.3 External Pins

The following defines the external pins for Timer A.

Pin Name	Input/Output	Pin Functions	Multiplex Pin*/Remarks
TimerA0out	Output	Timer A0 output	GPIOB4/INT4 (*)
TimerA1out	Output	Timer A1 output	GPIOB4/INT5 (*)
TimerA2out	Output	Timer A2 output	GPIOB5/INT6 (*)

* As timer-A external pins are multiplexed with GPIO and other pins, this function can be used if "Function 1 other than GPIO" is set using the GPIO Pin Function register.

22.4 Registers

22.4.1 List of Registers

These registers have base address 0xFFFF_B000.

Table 22.1 List of Registers (Base Address: 0xFFFF_B000)

Address Offset	Register Name	Register Abbreviation	Default Value	R/W	Data Access Size
0x00	Timer A0 Load register	TMA0LD	0x0000	R/W	16 (/32) *1
0x04	Timer A0 Count register	TMA0CNT	0x0000	RO	16 (/32) *1
0x08	Timer A0 Control register	TMA0CTRL	0x0000	(R/W)	16 (/32) *1
0x0C	Timer A0 IRQ Flag Clear register	TMA0IRQ	—	WO	8 (/16/32) *2
0x10	Timer A0 Port Output Control register	TMA0POUT	0x0000	(R/W)	8 (/16/32) *2
0x20	Timer A1 Load register	TMA1LD	0x0000	R/W	16 (/32) *1
0x24	Timer A1 Count register	TMA1CNT	0x0000	RO	16 (/32) *1
0x28	Timer A1 Control register	TMA1CTRL	0x0000	(R/W)	16 (/32) *1
0x2C	Timer A1 IRQ Flag Clear register	TMA1IRQ	—	WO	8 (/16/32) *2
0x30	Timer A1 Port Output Control register	TMA1POUT	0x0000	(R/W)	8 (/16/32) *2
0x40	Timer A2 Load register	TMA2LD	0x0000	R/W	16 (/32) *1
0x44	Timer A2 Count register	TMA2CNT	0x0000	RO	16 (/32) *1
0x48	Timer A2 Control register	TMA2CTRL	0x0000	(R/W)	16 (/32) *1
0x4C	Timer A2 IRQ Flag Clear register	TMA2IRQ	—	WO	8 (/16/32) *2
0x50	Timer A2 Port Output Control register	TMA2POUT	0x0000	(R/W)	8 (/16/32) *2
0x60-0x9C	Reserved	—	—	—	—
0xA0	Prescaler 0 Control register	PS0CTRL	0x0000	(R/W)	16 (/32) *1
0xA4	Prescaler 1 Control register	PS1CTRL	0x0000	(R/W)	16 (/32) *1
0xB0	Timer A IRQ Status register	TMAIRQSTS	0x0000	RO	8 (/16/32) *2

*1: Access to 16-bit or 32-bit data is enabled.
*2: Access to various size of data (8, 16 and 32-bit data) is enabled.

22.4.2 Detailed Description of Registers

Timer A0 Load Register (TMA0LD)

TIM[0x00] Default value = 0x0000 Read/Write

Timer A0 load value [15:8]							
15	14	13	12	11	10	9	8
Timer A0 load value [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **TMA0LD [15:0] Timer A0 load value bits [15:0]**

This register sets the (16-bit) initial count to be loaded in timer A0.

Timer A0 Count Register (TMA0CNT)

TIM[0x04] Default value = 0x0000 Read only

Timer A0 current count [15:8]							
15	14	13	12	11	10	9	8
Timer A0 current count [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **TMA0CNT [15:0] Timer A0 current count bits [15:0]**

The current count of timer A0 can be read.

Timer A0 Control Register (TMA0CTRL)

TIM[0x08] Default value = 0x0000 (Read/Write)

n/a RO						Divider/Prescaler [1:0] R/W	
15	14	13	12	11	10	9	8
Timer A0 enable R/W	n/a RO	Mode selection R/W	Divider dividing ratio [2:0] R/W			Immediate load request R/W	IRQ request R/W
7	6	5	4	3	2	1	0

Bits [9:8]: **Divider/prescaler selection [1:0]**

Sets the dividing source of the clock to be used for timer A0.

- 0x: Divider
- 10: Prescaler #0
- 11: Prescaler #1

Bit 7: **Timer A0 enable**

When this bit is enabled, the countdown starts in the mode being set by the mode selection bit of this register.

- 0: Disable timer A0.
- 1: Enables timer A0 (in Cyclic mode or Single mode).

Bit 5: **Selects a mode.**

- 0: Cyclic mode
- 1: Single mode

Bits [4:2]: **Divider dividing ratio [2:0]**

000: Do not divide. →1/1
001: Divide by 4. →1/4
010: Divide by 8. →1/8
011: Divide by 16. →1/16
100: Divide by 32. →1/32
101: Divide by 64. →1/64
110: Divide by 128. →1/128
111: Divide by 256. →1/256

Bit 1: **Immediate load request (in Cyclic mode)**

Requests for a reload timing of the load value of timer A0 in the Cyclic mode.

0: Reloads the written load value when the counter reaches "0x0000."
1: Reloads the load value immediately when it is written in the register.

Note: In the Single mode, however, the load value is reflected immediately when it is written regardless of this bit setting.

Bit 0: **Timer A0 interrupt request enable**

0: Disables an interrupt request (IRQ mask).
1: Enables an interrupt request.

Timer A0 IRQ Flag Clear Register (TMA0IRQ)

TIM[0x0C] Default value = — Write Only

15	14	13	12	11	10	9	8
Any data							
7	6	5	4	3	2	1	0
Any data							

This register is a write-only port to clear the interrupt request (IRQ) of timer A0. The IRQ flag is cleared if any data is written in this register.

Timer A0 Port Output Control Register (TMA0POUT)

TIM[0x10] Default value = 0x0000 (Read/Write)

15	14	13	12	11	10	9	8
				n/a RO			
7	6	5	4	3	2	1	0
n/a RO				Output mode select R/W		Output enable R/W	Output value R/W

Bits [3:2]: **Output mode select**

00: Outputs a value (bit 0) during underflow.
01: Underflow output
10: Toggle output in each occurrence of underflow
11: Reserved

Bit 1: **Output enable**

0: Disables the output mode.
1: Enables the output mode.
If the corresponding port has been set as GPIO port and if this bit is set to 1, the port operates in the output mode selected by bits [3:2].

Bit 0: **Output value**

If the output mode (Bits [3:2]) is "00", this bit value is output.

Timer A1 Load Register (TMA1LD)

TIM[0x20] Default value = 0x0000 Read/Write

Timer A1 load value [15:8]							
15	14	13	12	11	10	9	8
Timer A1 load value [7:0]]							
7	6	5	4	3	2	1	0

Bits [15:0]: **TMA1LD [15:0] Timer A1 load value bits [15:0]**

This register sets the (16-bit) initial count to be loaded in timer A1.

Timer A1 Count Register (TMA1CNT)

TIM[0x24] Default value = 0x0000 Read Only

Timer A1 current count [15:8]							
15	14	13	12	11	10	9	8
Timer A1 current count [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **TMA1CNT [15:0] Timer A1 current count bits [15:0]**

The current count of timer A1 can be read.

Timer A1 Control Register (TMA1CTRL)

TIM[0x28] Default value = 0x0000 (Read/Write)

n/a RO						Divider/Prescaler [1:0 R/W	
15	14	13	12	11	10	9	8
Timer A1 enable R/W	n/a RO	Mode select R/W	Divider dividing ratio [2:0] R/W			Immediate load request R/W	IRQ request R/W
7	6	5	4	3	2	1	0

Bits [9:8]: **Divider/prescaler selection [1:0]**

Sets the dividing source of the clock to be used for timer A1.

- 0x: Divider
- 10: Prescaler #0
- 11: Prescaler #1

Bit 7: **Timer A1 enable**

When this bit is enabled, the countdown starts in the mode being set by the mode selection bit of this register.

- 0: Disables timer A1.
- 1: Enables timer A1 (in Cyclic mode or Single mode).

Bit 5: **Mode select**

- 0: Cyclic mode
- 1: Single mode

Bits [4:2]: **Divider dividing ratio [2:0]**

000: Do not divide. →1/1
001: Divide by 4. →1/4
010: Divide by 8. →1/8
011: Divide by 16. →1/16
100: Divide by 32. →1/32
101: Divide by 64. →1/64
110: Divide by 128. →1/128
111: Divide by 256. →1/256

Bit 1: **Immediate load request (in Cyclic mode)**

Sets a reload timing of the load value of timer A1 in the Cyclic mode.

0: Reloads the written load value when the counter reaches "0x0000".
1: Reloads the load value immediately when it is written in the register.

Caution: In the Single mode, however, the load value is reloaded immediately when it is written regardless of this bit value.

Bit 0: **Timer A1 interrupt request enable**

0: Disables an interrupt request (IRQ mask).
1: Enables an interrupt request.

Timer A1 IRQ Flag Clear Register (TMA1IRQ)

TIM[0x2C] Default value = —— Write Only

15	14	13	12	11	10	9	8
Any data							
7	6	5	4	3	2	1	0
Any data							

This register is a write-only port to clear the interrupt request (IRQ) of timer A1. The IRQ flag is cleared if any data is written in this register.

Timer A1 Port Output Control Register (TMA1POUT)

TIM[0x30] Default value = 0x0000 (Read/Write)

15	14	13	12	11	10	9	8
				n/a RO			
7	6	5	4	3	2	1	0
n/a RO				Output mode select R/W		Output enable R/W	Output value R/W

Bits [3:2]: **Output mode select**

00: Outputs a value (bit 0) during underflow.
01: Underflow output
10: Toggle output in each occurrence of underflow
11: Reserved

Bit 1: **Output enable**

0: Disables the output mode.
1: Enables the output mode.
If the corresponding port has been set as GPIO port and if this bit is set to 1, the port operates in the output mode selected by bits [3:2].

Bit 0: **Output value**

If the output mode (Bits [3:2]) is "00", this bit value is output.

Timer A2 Load Register (TMA2LD)

TIM[0x40] Default value = 0x0000 Read/Write

Timer A2 load value [15:8]							
15	14	13	12	11	10	9	8
Timer A2 load value [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **TMA2LD [15:0] Timer A2 load value bits [15:0]**

This register sets the (16-bit) initial count to be loaded in timer A2.

Timer A2 Count Register (TMA2CNT)

TIM[0x44] Default value = 0x0000 Read only

Timer A2 current count [15:8]							
15	14	13	12	11	10	9	8
Timer A2 current count [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **TMA2CNT [15:0] Timer A2 current count bits [15:0]**

The current count of timer A2 can be read.

Timer A2 Control Register (TMA2CTRL)

TIM[0x48] Default value = 0x0000 (Read/Write)

n/a RO						Divider/Prescaler [1:0] R/W	
15	14	13	12	11	10	9	8
Timer A2 enable R/W	n/a RO	Mode select R/W	Divider dividing ratio [2:0] R/W			Immediate load request R/W	IRQ request R/W
7	6	5	4	3	2	1	0

Bits [9:8]: **Divider/prescaler selection [1:0]**

Sets the dividing source of the clock to be used for timer A2.

- 0x: Divider
- 10: Prescaler #0
- 11: Prescaler #1

Bit 7: **Timer A2 enable**

When this bit is enabled, the countdown starts in the mode being set by the mode selection bit of this register.

- 0: Disables timer A2.
- 1: Enables timer A2 (in Cyclic mode or Single mode).

Bit 5: Mode select

- 0: Cyclic mode
- 1: Single mode

Bits [4:2]: **Divider dividing ratio [2:0]**

000 : Do not divide. →1/1
001 : Divide by 4. →1/4
010 : Divide by 8. →1/8
011 : Divide by 16. →1/16
100 : Divide by 32. →1/32
101 : Divide by 64. →1/64
110 : Divide by 128. →1/128
111 : Divide by 256. →1/256

Bit 1: **Immediate load request (in Cyclic mode)**

Sets a reload timing of the load value of timer A2 in the Cyclic mode.

0: Reloads the written load value when the counter reaches "0x0000."
1: Reloads the load value immediately when it is written in the register.

Caution: In the Single mode, however, the load value is reflected immediately when it is written regardless of this bit setting.

Bit 0: **Timer A2 interrupt request enable**

0: Disables an interrupt request (IRQ mask).
1: Enables an interrupt request.

Timer A2 IRQ Flag Clear Register (TMA2IRQ)

TIM[0x4C] Default value = — Write Only

15	14	13	12	11	10	9	8
Any data							
7	6	5	4	3	2	1	0
Any data							

This register is a write-only port to clear the interrupt request (IRQ) of timer A2. The IRQ flag is cleared if any data is written in this register.

Timer A2 Port Output Control Register (TMA2POUT)

TIM[0x50] Default value = 0x0000 (Read/Write)

15	14	13	12	11	10	9	8
				n/a RO			
7	6	5	4	3	2	1	0
n/a RO				Output mode select R/W		Output enable R/W	Output value R/W

Bits [3:2]: **Output mode select**

00: Enables an output (bit 0) during underflow.
01: Underflow output
10: Toggle output in each occurrence of underflow
11: Reserved

Bit 1: **Output enable**

0: Disables the output mode.
1: Enables the output mode.
If the corresponding port has been set as GPIO port and if this bit is set to 1, the port operates in the output mode selected by bits [3:2].

Bit 0: **Output value**

This bit value is output if the output mode (bits 3 and 2) are set to "00."

Prescaler 0 Control Register (PS0CTRL)

TIM[0xA0] Default value = 0x0000 (Read/Write)

Divider dividing ratio [2:0]			n/a				
R/W			RO				
15	14	13	12	11	10	9	8
Prescaler 0 load value [7:0]							
R/W							
7	6	5	4	3	2	1	0

Bits [15:13]: **Divider dividing ratio [2:0]**

000: Do not divide. →1/1
001: Divide by 4. →1/4
010: Divide by 8. →1/8
011: Divide by 16. →1/16
100: Divide by 32. →1/32
101: Divide by 64. →1/64
110: Divide by 128. →1/128
111: Divide by 256. →1/256

Bits [7:0]: **Prescaler 0 load value bits [7:0]**

This register sets the count of the clock divided by the divider in prescaler 0.

Prescaler 1 Control Register (PS1CTRL)

TIM[0xA4] Default value = 0x0000 (Read/Write)

Divider dividing ratio			n/a				
R/W			RO				
15	14	13	12	11	10	9	8
Prescaler 1 load value [7:0]							
R/W							
7	6	5	4	3	2	1	0

Bits [15:13]: **Divider dividing ratio [2:0]**

000: Do not divide. →1/1
001: Divide by 4. →1/4
010: Divide by 8. →1/8
011: Divide by 16. →1/16
100: Divide by 32. →1/32
101: Divide by 64. →1/64
110: Divide by 128. →1/128
111: Divide by 256. →1/256

Bits [7:0]: **Prescaler 1 load value bits [7:0]**

This register sets a value to count the clock divided by the divider in prescaler 1.

Timer-A IRQ Status Register (TMAIRQSTS)

TIM[0xB0] Default value = 0x0000 Read Only

n/a							
15	14	5	4	3	2	1	0
n/a			Reserved		Timer2 IRQ	Timer1 IRQ	Timer0 IRQ
7	6	5	4	3	2	1	0

Bits [4:3]: **Reserved**

Bit 2: **Timer A2 IRQ status**

Indicates the interrupt status of timer A2.

0: No interrupt request was issued (or IRQ masked).

1: An interrupt request was issued.

This interrupt status is cleared by the timer-A2 IRQ flag clear register.

Bit 1: **Timer A1 IRQ status**

Indicates the interrupt status of timer A1.

0: No interrupt request was issued (or IRQ masked).

1: An interrupt request was issued.

This interrupt status is cleared by the timer-A1 IRQ Flag Clear register.

Bit 0: **Timer A0 IRQ status**

Indicates the interrupt status of timer A0.

0: No interrupt request was issued (or IRQ masked).

1: An interrupt request was issued.

This interrupt status is cleared by the timer-A0 IRQ Flag Clear register.

22.5 Setting the Load Value in Each Mode

22.5.1 Timer Counter Modes

Each of 3-channel timers can be operated in any of the following two modes.
This section explains how to set the load value in each mode.

(1) Cyclic mode

The load value being set by the Load register is counted down to value 0.
The value is loaded in the following two types of timing.

a) If an immediate load request is issued

If bit 1 (Immediate Load Request bit) of Timer-A Control register is set to logical 1, the load value is reflected on the counter immediately.

To do so, set the Immediate Load Request bit to 1 when you wish to reflect, and set the load value that you wish to reflect using the Load register. The load value is forcibly written in the counter regardless of the bit 7 (Timer-A Enable bit) status of Timer-A Control register.

b) If an immediate load request is NOT issued

If bit 1 (Immediate Load Request bit) of Timer-A Control register is set to logical 0, an interrupt is generated when all bits of the down counter are set to 0. At this time, the value being set in the Load register is reloaded and the countdown starts.

To use it in the free-running mode, set value FFFFh as the load value.

(2) Single mode

The load value being set by the Load register is counted down. When all counter outputs reach value 0, an interrupt is generated and the countdown is stopped. The counter continues to stop even if the interrupt is cleared by the IRQ Flag Clear register.

When a load value is set by the Load register, it is reloaded and reflected immediately on the counter regardless of the bit 1 (Immediate Load Request bit) value of Timer-A Control register. The load value can be written forcibly in the counter regardless of the bit 7 (Timer-A Enable bit) status of Timer-A Control register.

22.6 Timer Internal Clock Setting Examples (1 KHz, 1 MHz)

22.6.1 Divider and Prescaler Settings

You can set the dividing ratio and the count value for three channel timers using their dividers and 2-channel prescalers. The timers can be counted down in any of two cycles.

If you operate the timer with the dividing ratio of the diver without using the prescaler, you can set the dividing ratio for each of three channel timers.

The APB bus clock (PCLK) divided by 8 is entered as the input clock (TINCLK) of the timer.

Table 22.2 gives the 1-msec (1KHz) and 1-µsec (1MHz) cycle setting examples when the 6MHz TINCLK (if PCLK=48MHz) is entered. Set the corresponding Prescaler Control registers as follows. If you set the count to 1/1, set the prescaler load value to "0x00." To set the count to 1/2, set the load value to "0x01" and to set the count to 1/3, set the load value to "0x02" and so on.

Table 22.2 Millisecond and Microsecond Cycle Setting Examples *

Cycle	Divider dividing ratio	Prescaler load value	Frequency
1 ms	Bits [15:13] = 100b (1/ 32 ratio selection)	Bits [7:0] = 1011 1010b (0xBA) (1/ 187 ratio selection)	1.002673KHz
	Bits [15:13] = 101b (1/ 64 ratio selection)	Bits [7:0] = 0101 1101b (0x5D) (1/ 94 ratio selection)	0.997340KHz
1 µs	Bits [15:13] = 000b (1/ 1 ratio selection)	Bits [7:0] = 0000 0101b (0x05) (1/ 6 ratio selection)	1.000000MHz

* An example during 6MHz f_{TINCLK} input (48MHz f_{PCLK} input).

The following gives an expression to calculate the frequency you create using the PCLK frequency.

Set the divider's dividing ratio and the prescaler load value in the Prescaler [1:0] Control register so that its frequency closes to the frequency you wish to create.

$$\text{Frequency (f) to be created} = \underbrace{\frac{f_{PCLK} \div 8}{f_{TINCLK}}}_{} \div \underbrace{\text{Value set by bits [15:13]}}_{\text{Divider dividing ratio}} \div \underbrace{\text{Value set by bits [7:0]}}_{\text{Prescaler load value}}$$

* : Prescaler [1:0] Control registers (TIM[0xA0], TIM[0xA4])

Calculation examples if settings of Table 22.2 are used

Example 1: 1kHz ≒ 48MHz ÷ 8 ÷ 32 ÷ 187 = 1.002673KHz

Example 2: 1KHz ≒ 48MHz ÷ 8 ÷ 64 ÷ 94 = 0.997340KHz

Example 3: 1MHz ≒ 48MHz ÷ 8 ÷ 1 ÷ 6 = 1.000000MHz

22.7 Timing Diagrams

22.7.1 Immediate Load Request in Cyclic Mode

The following shows a timing chart if an immediate load request is issued in the Cyclic mode and if the load value is reflected immediately. The Enable cycle of the counter is set to 1/3 by the prescaler.

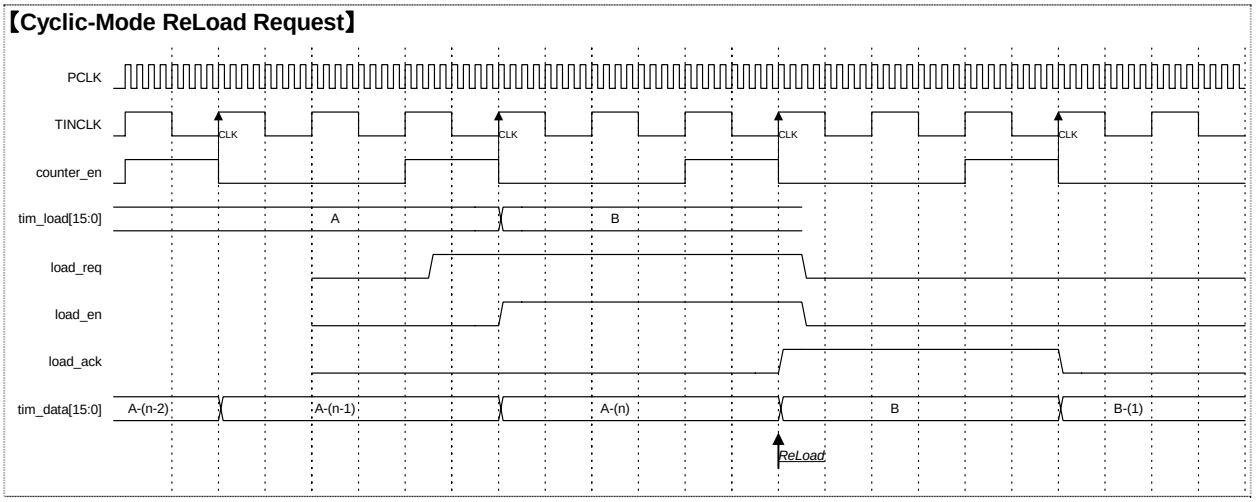


Fig.22.2 Immediate Load Request in Cyclic Mode

22.7.2 Normal Reloading in Cyclic Mode

The following shows a timing chart of normal cycle countdown if an immediate load request is NOT issued in the Cyclic mode and if the load value is reloaded when all counter outputs reach zero (0). This chart shows the interrupt occurrence and the reloading without clock dividing.

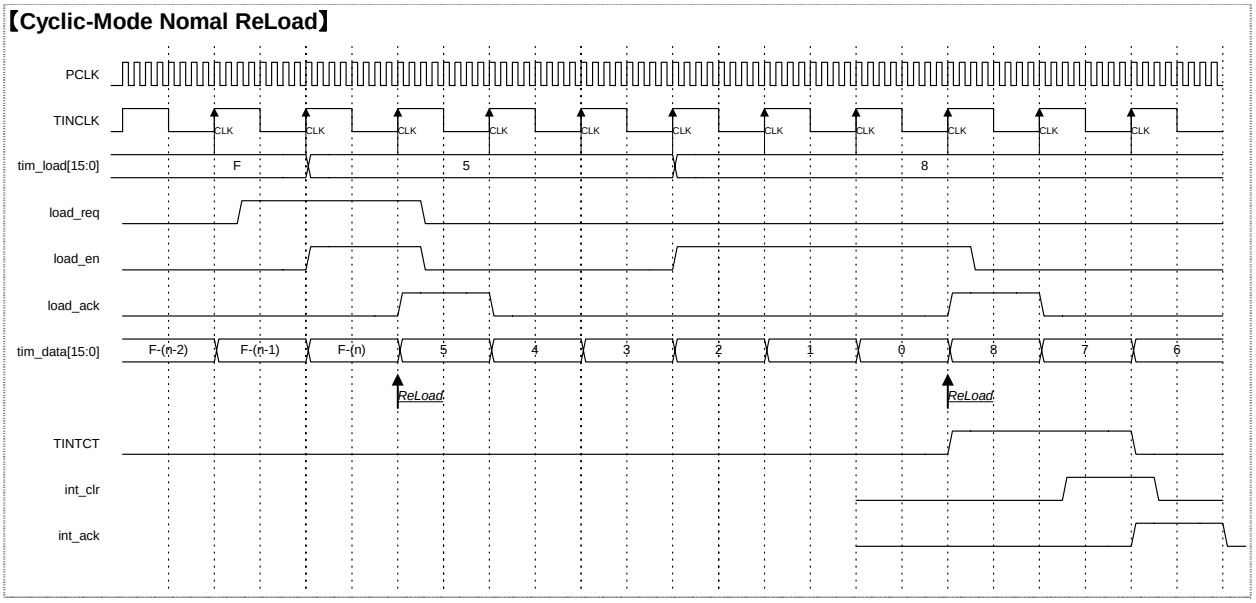


Fig.22.3 Normal Reloading in Cyclic Mode

22.7.3 Normal Reloading in Single Mode

The following shows a timing chart where an interrupt occurs when all counter outputs reach zero (0) in the Single mode and the load value is written. The clock is not divided in this example.

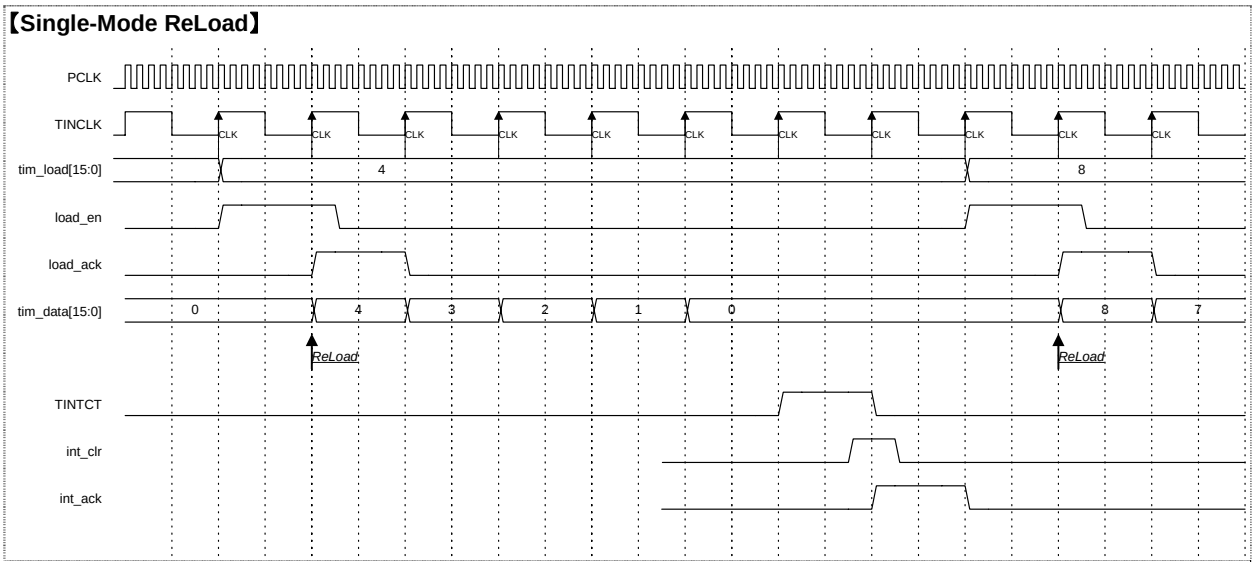


Fig.22.4 Normal Reloading in Single Mode

22.7.4 Port Outputs

The following shows the timing charts for port outputs if an underflow occurs in each mode.
Note that each chart shows the waveforms when the default divider and prescaler values (1/1) are set.

(1) Outputs a value (bit 0) during underflow (Output mode="00")

If an underflow occurs, bit 0 (output value bit) value of the Port Output Control register is held and it is output.

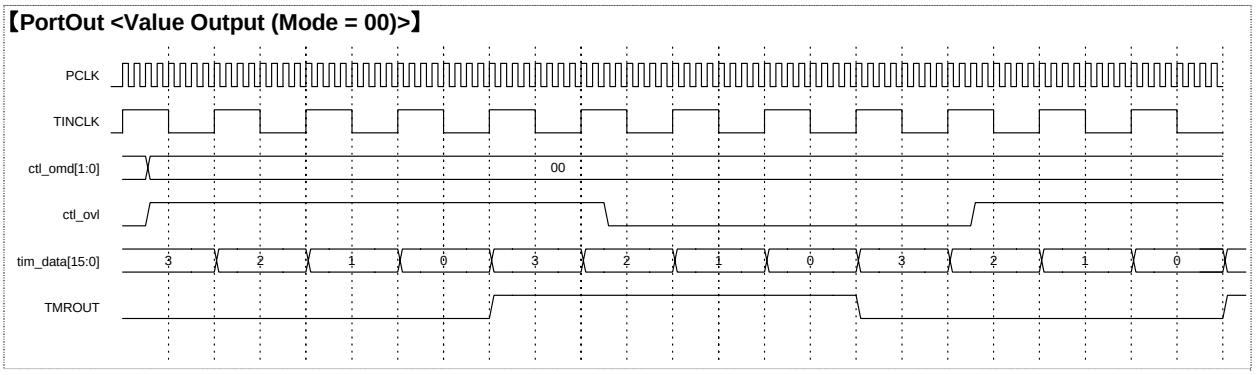


Fig.22.5 Outputs (Output Values) during Underflow

(2) Output of underflow (Output mode = “01”)

If an underflow occurs, the underflow data itself is output.

The TMROUT underflow pulse width is the same as the downcount data width. When the count changes from zero (0) to the load value (“3” in this figure), the underflow signal is output.

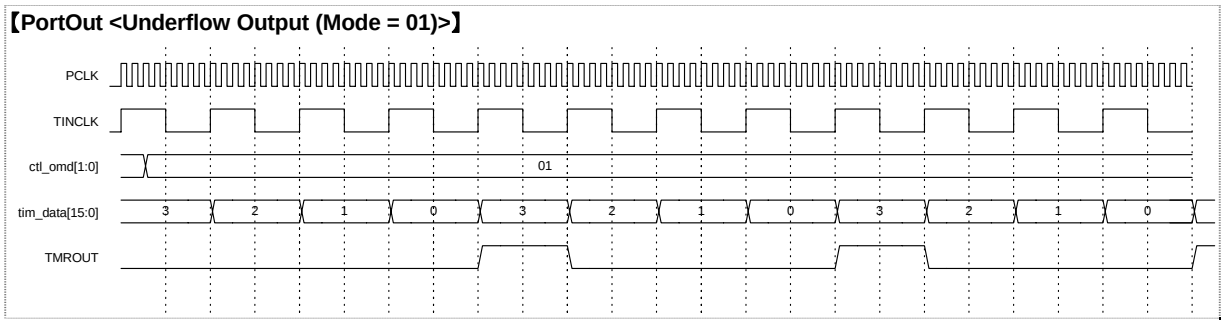


Fig.22.6 Outputs during Underflow (Underflow Output)

(3) Toggle output during underflow (Output mode=“10”)

If an underflow occurs, the signal is inverted and output.

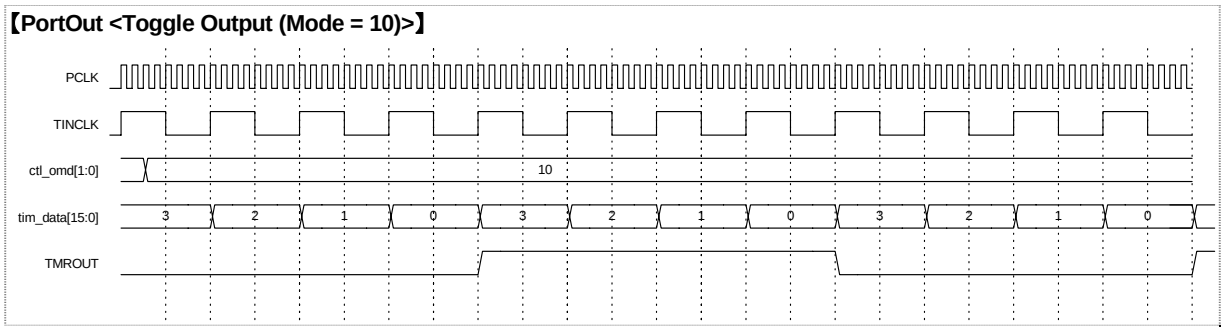


Fig.22.7 Outputs during Underflow (Toggle Output)

23. Timer B (TIMB)

23.1 Description

Timer B is a 16-bit timer having the Output Compare function and the Input Capture function. This timer can also be operated as an event counter using externally input pulses. This is the multi-functional timer that can output pulses with any duty ratio based on the compare signals between the timer counter and four Common registers.

23.2 Features

- Counter input clock:
  1/16 to 1/2048 of external clock or internal system clock
- Can process up to 4 lines of pulse output.
- Four Common registers
  Each of them can be set as an Output register or an Input Capture register.
- Timer I/O function:
  Output compare: Allows 0 output, 1 output, or toggle output.
  Input capture: Can detect a rising edge, a falling edge, or both edges of signal waveforms.
  Allows PWM output.
- When operated as Input Capture registers, the current count and the previous count are held.
- Can count events using the externally input pulses.
- Five types of interrupt causes:
- Four (4) interrupt causes for compare matching or input capturing, and one (1) overflow interrupt

23.3 Block Diagram

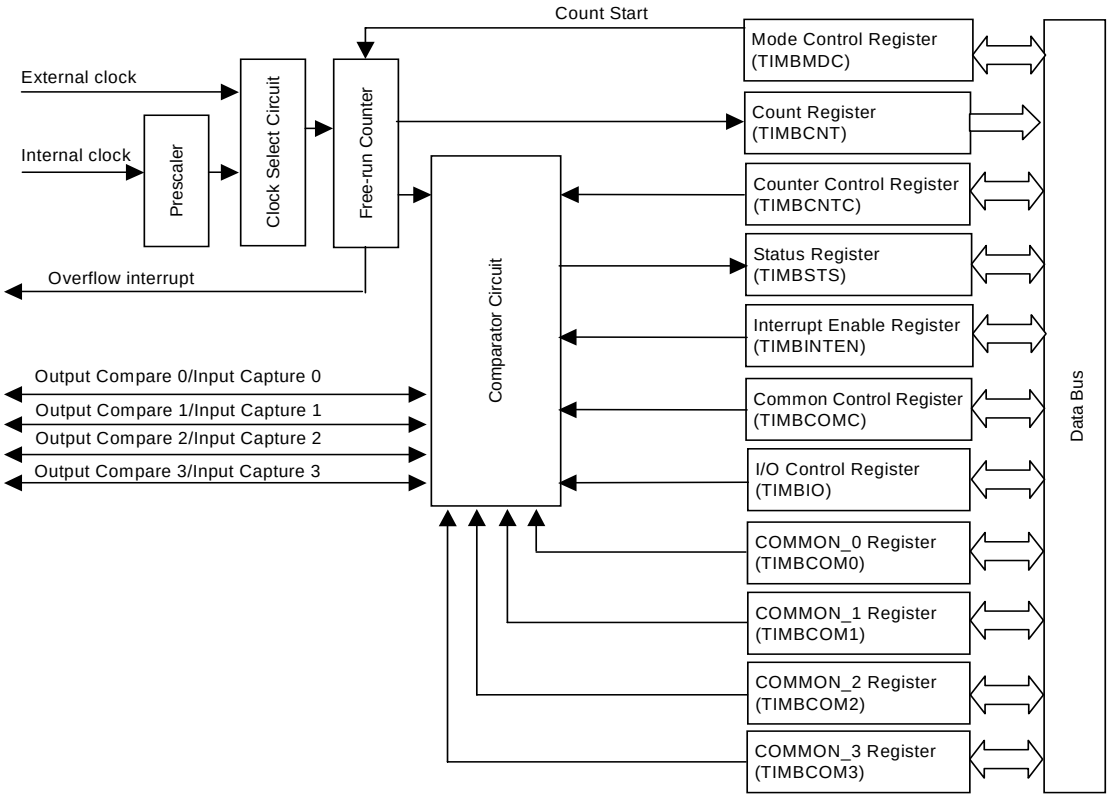


Fig. 23.1 Timer-B Block Diagram

23.4 External Pins

The following defines the external pins for Timer B.

Pin Name	Input/Output	Pin Functions	Multiplex Pin*/Remarks
TimerBIn	Input	External clock input pin	GPIOB7
TimerB0IO	Input/Output	COMMON_0 register: Output compare output/Input capture input pin	GPIOC0
TimerB1IO	Input/Output	COMMON_1 register: Output compare output/Input capture input pin	GPIOC1
TimerB2IO	Input/Output	COMMON_2 register: Output compare output/Input capture input pin	GPIOC2
TimerB3io	Input/Output	COMMON_3 register: Output compare output/Input capture input pin	GPIOC3/ UART3_CLK

* As timer-B external pins are multiplexed with GPIO and other pins, this function can be used if "Function 1 other than GPIO" is set using the GPIO Pin Function register.

23.5 Registers

23.5.1 List of Registers

These registers have base address 0xFFFD_3000.

Table 23.1 List of Registers (Base Address: 0xFFFD_3000)

Address Offset	Register Name	Register Abbreviation	Default value	R/W	Data Access Size
0x00	Timer B Mode Control register	TIMBMDC	0x0000	R/W	16 (/32)
0x04	Timer B Counter Control register	TIMBCNTC	0x0000	R/W	16 (/32)
0x08	Timer B Status register	TIMBSTS	0x0000	R/W	16 (/32)
0x0C	Timer B Interrupt Enable register	TIMBINTEN	0x0000	R/W	16 (/32)
0x10	Timer B Count register	TIMBCNT	0x0000	R/W	16 (/32)
0x14	Timer B Common Control register	TIMBCOMC	0x0000	R/W	16 (/32)
0x18	Timer B I/O Control register	TIMBIO	0x0000	R/W	16 (/32)
0x1C	Timer B Cycle register	TIMBDUTY	0x0000	R/W	16 (/32)
0x20	Timer B COMMON_0 register	TIMBCOM0	0x0000	R/W	16 (/32)
0x24	Timer B COMMON_1 register	TIMBCOM1	0x0000	R/W	16 (/32)
0x28	Timer B COMMON_2 register	TIMBCOM2	0x0000	R/W	16 (/32)
0x2C	Timer B COMMON_3 register	TIMBCOM3	0x0000	R/W	16 (/32)
0x30	Timer B COMINP_0 register	TIMBINP0	0x0000	RO	16 (/32)
0x34	Timer B COMINP_1 register	TIMBINP1	0x0000	RO	16 (/32)
0x38	Timer B COMINP_2 register	TIMBINP2	0x0000	RO	16 (/32)
0x3C	Timer B COMINP_3 register	TIMBINP3	0x0000	RO	16 (/32)

23.5.2 Detailed Description of Registers

Timer B Mode Control Register (TIMBMDC)							
TIMB[0x00] Default value = 0x0000							Read/Write
clr_B3o	clr_B2o	clr_B1o	clr_B0o	set_B3o	set_B2o	set_B1o	set_B0o
15	14	13	12	11	10	9	8
Reserved				B3pwm	B2pwm	B1pwm	B0pwm
7	6	5	4	3	2	1	0

Bit 15: **clr_B3o Timer output 3 forcible clear**
- (W) 1: Forcibly sets the TimerB3io output to Low. (Invalid during input capturing)
- 0: Disabled
- (R) : Shows the current TimerB3io output status. (Invalid during input capturing)

Bit 14: **clr_B2o Timer output 2 forcible clear**
- (W) 1: Forcibly sets the TimerB2io output to Low. (Invalid during input capturing)
- 0: Disabled
- (R) : Shows the current TimerB2io output status. (Invalid during input capturing)

Bit 13: **clr_B1o Timer output 1 forcible clear**
- (W) 1: Forcibly sets the TimerB1io output to Low. (Invalid during input capturing)
- 0: Disabled
- (R) : Shows the current TimerB1io output status. (Invalid during input capturing)

Bit 12: **clr_B0o Timer output 0 forcible clear**
- (W) 1: Forcibly sets the TimerB0io output to Low. (Invalid during input capturing)
- 0: Disabled
- (R) : Shows the current TimerB0io output status. (Invalid during input capturing)

Bit 11: **set_B3o Timer output 3 forcible set**
- (W) 1: Forcibly sets the TimerB3io output to High. (Invalid during input capturing)
- 0: Disabled
- (R) : Shows the current TimerB3io output status. (Invalid during input capturing)

Bit 10: **set_B2o Timer output 2 forcible set**
- (W) 1: Forcibly sets the TimerB2io output to High. (Invalid during input capturing)
- 0: Disabled
- (R) : Shows the current TimerB2io output status. (Invalid during input capturing)

Bit 9: **set_B1o Timer output 1 forcible set**
- (W) 1: Forcibly sets the TimerB1io output to High. (Invalid during input capturing)
- 0: Disabled
- (R) : Shows the current TimerB1io output status. (Invalid during input capturing)

Bit 8: **set_B0o Timer output 0 forcible set**
- (W) 1: Forcibly sets the TimerB0io output to High. (Invalid during input capturing)
- 0: Disabled
- (R) : Shows the current TimerB0io output status. (Invalid during input capturing)

Bits [7:4]: **Reserved**

Bits 3: **B3pwm PWM mode 3 (TimerB3io output)**
- 0: Compare output
- 1: PWM output

Bits 2: **B2pwm PWM mode 2 (TimerB2io output)**
- 0: Compare output
- 1: PWM output

Bits 1: **B1pwm PWM mode 1 (TimerB1io output)**
- 0: Compare output
- 1: PWM output

Bits 0: **B0pwm PWM mode 0 (TimerB0io output)**
- 0: Compare output
- 1: PWM output

Timer B Counter Control Register (TIMBCNTC)

TIMB[0x04] Default value = 0x0000 Read/Write

BcntTST	Reserved			Bclksel			
15	14	13	12	11	10	9	8
BcntCLR	Reserved						BcntST
7	6	5	4	3	2	1	0

Bits 15: **BcntTST counter test mode (Reserved)**
Do not write value 1 here.

Bits [14:12]: **Reserved**

Bits [11:8]: Bclksel clock select *
- 0000: The internal system clock divided by 16
- 0001: The internal system clock divided by 32
- 0010: The internal system clock divided by 64
- 0011: The internal system clock divided by 128
- 0100: The internal system clock divided by 256
- 0101: The internal system clock divided by 512
- 0110: The internal system clock divided by 1024
- 0111: The internal system clock divided by 2048
- 1xxx: External clock

* The clock division by 8 by the system controller is included in these dividing ratios of internal system clock.

Bits 7: **BcntCLR count mode set**
- 0: Operates as a free-run counter. (When the counter reaches 0xFFFF, it returns to 0x0000.)
- 1: Operates as a cycle counter. (When the counter reaches the TIMBDUTY Cycle register value, it returns to 0x0000.)

Bits [6:1] : **Reserved**

Bits 0: **BcntST counter start**
- 0: Stops timer counting.
- 1: Starts timer counting.

Timer B Status Register (TIMBSTS)

TIMB[0x08] Default value = 0x0000 Read/Write

Reserved						Bmatch	Bovf
15	14	13	12	11	10	9	8
Reserved				B3sts	B2sts	B1sts	B0sts
7	6	5	4	3	2	1	0

Bits [15:10]: **Reserved**

Bits 9: **Bmatch Cycle register compare match status**

(R) 1: When the Cycle register value matches the timer counter value
(R) 0: No counter matching

(W) 1: Clears the Status register.
(W) 0: Invalid

Bits 8: **Bovf overflow**

(R) 1: When the timer counter overflows from FFFFh to 0000h
(R) 0: No overflow

(W) 1: Clears the Status register.
(W) 0: Invalid

Bits [7:4]: **Reserved**

Bits 3: **B3sts TimerB3io input capture/compare match status**

(R) 1: When the counter value matches during output compare setting, or
When the counter value is transferred to TIMBCOM3 by input capture setting
(R) 0: No status flag interrupt occurs.

(W) 1: Clears the Status register.
(W) 0: Invalid

Bits 2: **B2sts TimerB2io input capture/compare match status**

(R) 1: When the counter value matches during output compare setting, or
When the counter value is transferred to TIMBCOM2 by input capture setting
(R) 0: No status flag interrupt occurs.

(W) 1: Clears the Status register.
(W) 0: Invalid

Bits 1: **B1sts TimerB1io input capture/compare match status**

(R) 1: When the counter value matches during output compare setting, or
When the counter value is transferred to TIMBCOM1 by input capture setting
(R) 0: No status flag interrupt occurs.

(W) 1: Clears the Status register.
(W) 0: Invalid

Bits 0: **B0sts TimerB0io input capture/compare match status**

(R) 1: When the counter value matches during output compare setting, or
When the counter value is transferred to TIMBCOM0 by input capture setting
(R) 0: No status flag interrupt occurs.

(W) 1: Clears the Status register.
(W) 0: Invalid

Timer B Interrupt Enable Register (TMBINTEN)

TIMB[0x0C] Default value = 0x0000 Read/Write

Reserved						BmatchEN	BovfEN
15	14	13	12	11	10	9	8
reserved				B3intEN	B2intEN	B1intEN	B0intEN
7	6	5	4	3	2	1	0

Bits [15:10]: **Reserved**

Bits 9: **BmatchEN Cycle register compare match interrupt control flag**

1: Interrupt enabled
0: Interrupt disabled

Bits 8: **BovfEN TimerBovf overflow interrupt control flag**

1: Interrupt enabled
0: Interrupt disabled

Bits [7:4]: **Reserved**

Bits 3: **B3intEN TimerB3io input capture/compare match interrupt control flag**

1: Interrupt enabled
0: Interrupt disabled

Bits 2: **B2intEN TimerB2io input capture/compare match interrupt control flag**

1: Interrupt enabled
0: Interrupt disabled

Bits 1: **B1intEN TimerB1io input capture/compare match interrupt control flag**

1: Interrupt enabled
0: Interrupt disabled

Bits 0: **B0intEN TimerB0io input capture/compare match interrupt control flag**

1: Interrupt enabled
0: Interrupt disabled

Timer B Count Register (TIMBCNT)

TIMB[0x10] Default value = 0x0000 Read/Write

Bcount [15:8]							
15	14	13	12	11	10	9	8
Bcount [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **Bcount current count bits [15:0]**

The current count of timer B can be read.
If the timer counting is stopped (BcntST is set to 0), any count can be set.

Timer B Common Control Register (TIMBCOMC)

TIMB[0x14] Default value = 0x0000 Read/Write

Reserved							
15	14	13	12	11	10	9	8
Reserved				B3comFUNC	B2comFUNC	B1comFUNC	B0comFUNC
7	6	5	4	3	2	1	0

Bits [15:4]: **Reserved**

Bits 3: **B3comFUNC TIMBCOM3 function select**

0: Functions as the Output Compare register.

1: Functions as the Input Capture register.

Bits 2: **B2comFUNC TIMBCOM2 function select**

0: Functions as the Output Compare register.

1: Functions as the Input Capture register.

Bits 1: **B1comFUNC TIMBCOM1 function select**

0: Functions as the Output Compare register.

1: Functions as the Input Capture register.

Bits 0: **B0comFUNC TIMBCOM0 function sele**

0: Functions as the Output Compare register.

1: Functions as the Input Capture register.

Timer B I/O Control Register (TIMBIO)

TIMB[0x18] Default value = 0x0000 Read/Write

Reserved[15:8]							
15	14	13	12	11	10	9	8
B3comIO		B2comIO		B1comIO		B0comIO	
7	6	5	4	3	2	1	0

Bits [15:8]: **Reserved**

Bits [7:6]: **B3comIO**

If the TIMBCOM3 functions as the Output Compare register:

00: Inhibits pin output during compare matching.

01: Outputs logical 0 to TimerB3io during compare matching.

10: Outputs logical 1 to TimerB3io during compare matching.

11: Outputs toggle signal to TimerB3io during compare matching.

If the TIMBCOM3 functions as the Input Capture register:

00: Captures the input in TIMBCOM3 at the TimerB3io rising edge.

01: Captures the input in TIMBCOM3 at the TimerB3io falling edge.

10: Captures the input in TIMBCOM3 at TimerB3io rising and falling edges.

11: Reserved

Bits [5:4]: **B2comIO**

If the TIMBCOM2 functions as the Output Compare register:
- 00: Inhibits pin output during compare matching.
- 01: Outputs logical 0 to TimerB2io during compare matching.
- 10: Outputs logical 1 to TimerB2io during compare matching.
- 11: Outputs toggle signal to TimerB2io during compare matching.

If the TIMBCOM2 functions as the Input Capture register:
- 00: Captures the input in TIMBCOM2 at the TimerB2io rising edge.
- 01: Captures the input in TIMBCOM2 at the TimerB2io falling edge.
- 10: Captures the input in TIMBCOM2 at TimerB2io rising and falling edges.
- 11: Reserved

Bits [3:2]: **B1comIO**

If the TIMBCOM1 functions as the Output Compare register:
- 00: Inhibits pin output during compare matching.
- 01: Outputs logical 0 to TimerB1io during compare matching.
- 10: Outputs logical 1 to TimerB1io during compare matching.
- 11: Outputs toggle signal to TimerB1io during compare matching.

If the TIMBCOM1 functions as the Input Capture register:
- 00: Captures the input in TIMBCOM1 at the TimerB1io rising edge.
- 01: Captures the input in TIMBCOM1 at the TimerB1io falling edge.
- 10: Captures the input in TIMBCOM1 at TimerB1io rising and falling edges.
- 11: Reserved

Bits [1:0]: **B0comIO**

If the TIMBCOM0 functions as the Output Compare register:
- 00: Inhibits pin output during compare matching.
- 01: Outputs logical 0 to TimerB0io during compare matching.
- 10: Outputs logical 1 to TimerB0io during compare matching.
- 11: Outputs toggle signal to TimerB0io during compare matching.

If the TIMBCOM0 functions as the Input Capture register:
- 00: Captures the input in TIMBCOM0 at the TimerB0io rising edge.
- 01: Captures the input in TIMBCOM0 at the TimerB0io falling edge.
- 10: Captures the input in TIMBCOM0 at TimerB0io rising and falling edges.
- 11: Reserved

Timer B Cycle Register (TIMBDUTY)

TIMB[0x1C] Default value = 0x0000 Read/Write

Bduty[15:8]							
15	14	13	12	11	10	9	8
Bduty [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **Bduty Duty register**

Set a cycle count value in the Cycle Count mode.

Timer B COMMON_0 Register (TIMBCOM0)

TIMB[0x20] Default value = 0x0000 Read/Write

B0com[15:8]							
15	14	13	12	11	10	9	8
B0com [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **B0com COMMON_0 register**

Set a counter value in the Output Compare mode.

The immediately preceding B0inp content is held in the Input Capture mode.

Timer B COMMON_1 Register (TIMBCOM1)

TIMB[0x24] Default value = 0x0000 Read/Write

B1com[15:8]							
15	14	13	12	11	10	9	8
B1com [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **B1com COMMON_1 register**

Set a counter value in the Output Compare mode.

The immediately preceding B1inp content is held in the Input Capture mode.

Timer B COMMON_2 Register (TIMBCOM2)

TIMB[0x28] Default value = 0x0000 Read/Write

B2com[15:8]							
15	14	13	12	11	10	9	8
B2com[7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **B2com COMMON_2 register**

Set a counter value in the Output Compare mode.

The immediately preceding B2inp content is held in the Input Capture mode.

Timer B COMMON_3 Register (TIMBCOM3)

TIMB[0x2C] Default value = 0x0000 Read/Write

B3com[15:8]							
15	14	13	12	11	10	9	8
B3com[7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **B3com COMMON_3 register**

Set a counter value in the Output Compare mode.

The immediately preceding B3inp content is held in the Input Capture mode.

Timer B COMINP_0 Register (TIMBINP0)

TIMB[0x30] Default value = 0x0000 Read Only

B0inp[15:8]							
15	14	13	12	11	10	9	8
B0inp [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **B0inp COMINP_0 register**

The counter value is entered in the Input Capture mode.

Timer B COMINP_1 Register (TIMBINP1)

TIMB[0x34] Default value = 0x0000 Read Only

B1inp[15:8]							
15	14	13	12	11	10	9	8
B1inp [7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **B1inp COMINP_1 register**

The counter value is entered in the Input Capture mode.

Timer B COMINP_2 Register (TIMBINP2)

TIMB[0x38] Default value = 0x0000 Read Only

B2inp[15:8]							
15	14	13	12	11	10	9	8
B2inp[7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **B2inp COMINP_2 register**

The counter value is entered in the Input Capture mode.

Timer B COMINP_3 register (TIMBINP3)

TIMB[0x3C] Default value = 0x0000 Read Only

B3inp[15:8]							
15	14	13	12	11	10	9	8
B3inp[7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **B3inp COMINP_3 register**

The counter value is entered in the Input Capture mode.

23.6 Explanation of Operations

23.6.1 Count Operations

There are two types of timer counter operations: the free-running count operation using the 16-bit up counter, and the cycle count operation.

(1) Free-running count operation

If the count mode setting bit (Bit[7]) of Counter Control register (TIMBCNTC) is set to 0 and if the counter start bit (Bit[0]) is set to 1, the free-running count operation starts. The count clock is divided by the built-in prescaler based on the clock select bits (Bits[3:1]) of Counter Control register (TIMBCNTC). When the counter overflows from FFFFh to 0000h, the overflow bit (Bovf bit[8]) of Status register (TIMBSTS) is set to 1. If the overflow control flag (Bit[8]) of Interrupt Enable register (TIMBINTEN) has been set to interrupt enabled, an interrupt request (TimerBovf) occurs.

The interrupt request (Bovf) is cleared by the software. Fig.23.2 illustrates the free-running count operation.

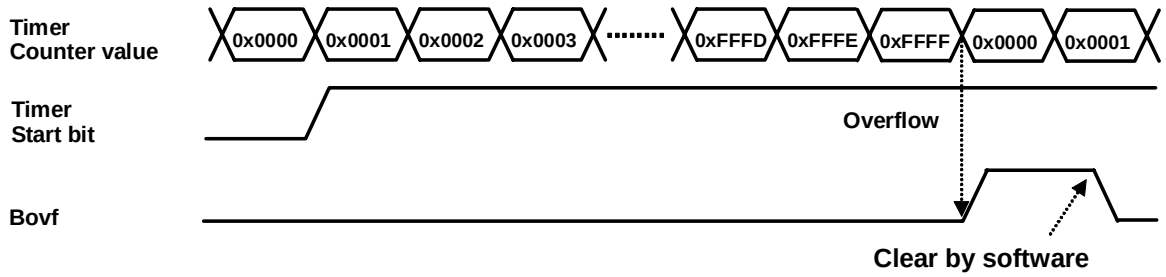


Fig.23.2 Free-running Count Operation

(2) Cycle count operation

If the count mode setting bit (Bit[7]) of Counter Control register (TIMBCNTC) is set to 1, the cycle count operation starts with the Cycle register value. When the counter value matches the Cycle register value, the timer counter is cleared to 0000h and the cycle register compare match (Bit[9]) of Status register (TIMBSTS) is set to 1. If the cycle register compare match interrupt control flag (Bit[9]) of Interrupt Enable register (TIMBINTEN) has been set to interrupt enabled, an interrupt request (TimerBovf) occurs. The interrupt request (Bmatch) is cleared by the software. Figure 23-3 illustrates the cycle count operation if the Cycle register value is 12FFh.

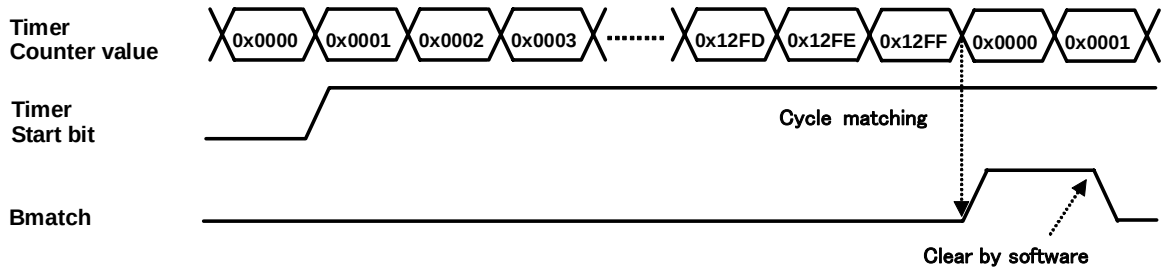


Fig.23.3 Cycle Count Operation (if Cycle Register Value is 12FFh)

23.6.2 Timer I/O Functions

Timer B has the following three input and output function modes. Each mode can be set for each pin (TimerB0/1/2/3io) separately.

- Output compare
- PWM output
- Input capture

(1) Output compare operations

If the TimerB0io, TimerB1io, TimerB2io, and TimerB3io functions of Common Control registers (TIMBCOM) are set in the Output Compare registers and if the function of I/O Control register (TIMBIO) is selected, the logical 0, logical 1 or toggle signals can be output at TimerB0io, TimerB1io, and TimerB3io pins.

Fig.23.4 shows the operations if the timer counter operates in the free-running mode, if value AAAAh is set in the COMMON_0 register, value BBBBh is set in the COMMON_1 register, and value AAAAh is set in the COMMON_2 register, and if TimerB0io pin is set for logical 1 output, TimerB1io pin is set for logical 0 output, and TimerB2io pin is set for toggle output.

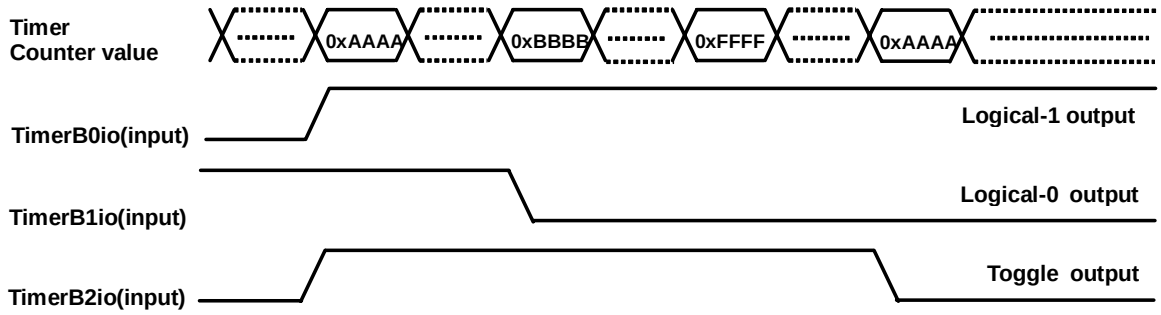


Fig.23.4 Output Settings

When the Common register value matches the counter value, the “B*sts” interrupt cause occurs regardless of the change of “imerB*io” pin output.

(2) PWM operations

In the PWM mode, the PWM waveforms are output at each of TimerB0io, TimerB1io, TimerB2io, and TimerB3io output pins based on the values of Common registers (TIMBCOM0, TIMBCOM1, TIMBCOM2, and TIMBCOM3) against the Cycle register (TIMBDUTY). Up to 4 phases of PWM waveforms can be output. If the PWM mode output function (Bits[3:0]) is selected using the Mode Control register (TIMBMDC), the selected Common register functions as the Compare register for PWM. If the Cycle register (TIMBDUTY) and Common register values are identical, the output value does not change even when a compare match occurs.

Fig.23.5 shows the operation if the Cycle register is set to FF00h, the TIMBCOM0 is set to AA00h, and TIMBCOM1 is set to BB00h and if the output level is high at TimerB0io and TimerB1io pins.

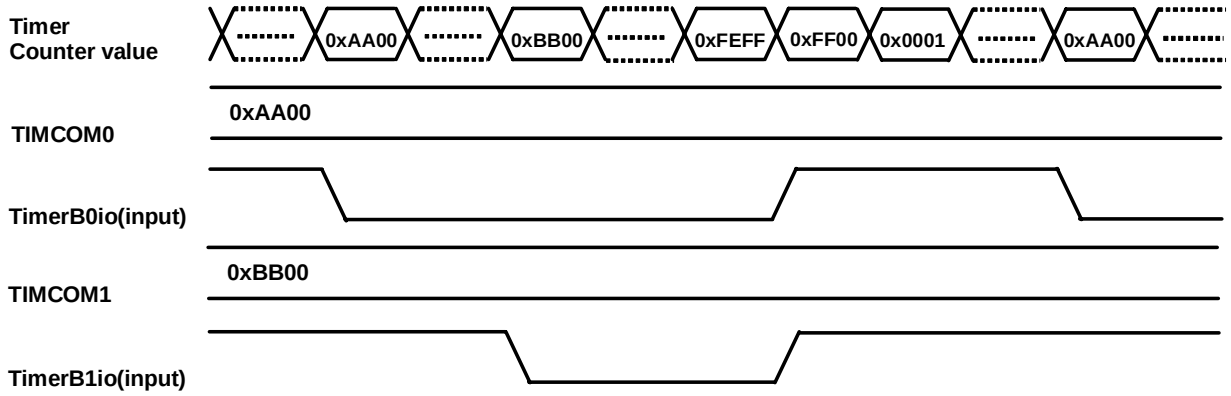


Fig.23.5 PWM Operations

(3) Input capture operations

If the TimerB0io, TimerB1io, TimerB2io, and TimerB3io functions of common control registers (TIMBCOMC) are set as Input Capture registers, an input signal edge is detected at TimerB0io, TimerB1io, TimerB2io, and TimerB3io pins and the timer counter value can be sent to the COMINP_0 register, COMINP_1 register, COMINP_2 register, and COMINP_3 register. Also, the previous COMINP_* register content is saved in the COMMON_* register. Therefore, the interval of input signal edges can be calculated accurately based on the contents of both registers and the occurrence count of overflow interrupts. A rising edge, a falling edge, or both edges of signals can be detected if its function is set by the I/O Control register (TIMBIO). You can measure the signal pulse width and cycle using these functions.

Fig.23.6 shows the operations if the timer counter is set to the free-running operation, if both rising and falling edges of a signal are detected at TimerB0io pin with the input capture function, and if a raising edge of signal is detected at TimerB1io pin with the input capture function.

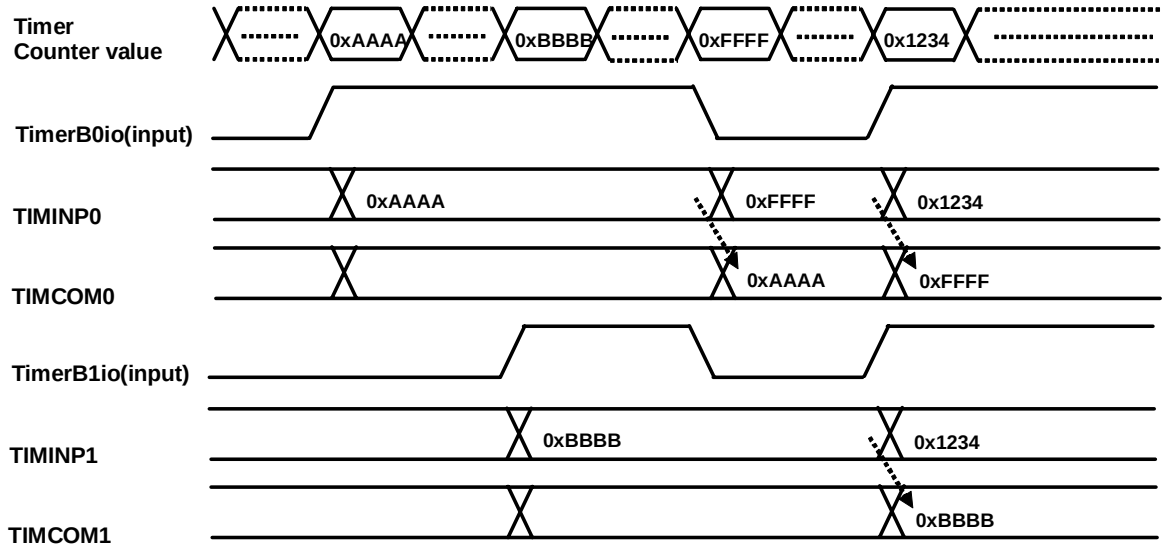


Fig.23.6 Operation Settings

24. Real-Time Clock (RTC)

24.1 Description

The real-time clock (called the RTC) receives the 32,768Hz input clock, measures the time using the 256Hz clock output prescaler, 8-bit dividing timer, and second, minute, hour, day, month and year counters. Also, this RTC has various time measuring functions including the watch and stopwatch. Each data can be read by the software. Also, the RTC can generate an interrupt when 32Hz, 8Hz, 2Hz, 1 second, 1 minute, 1 hour or 1 day is counted up. It can be used for cyclic interrupt and as the wakeup source. Also, an alarm can be generated by specifying the minutes, hours, day, month and year. The RTC can be used as the wakeup source and for alarming. If the 32,768Hz clock is running, the RTC can operate even when the CPU and other built-in peripheral circuits are standing by. Because the system reset does not affect on the RTC's time measurement, it can continue to measure the time even if an external reset signal is entered.

The RTC has the 8-byte, built-in RAM as the backup memory.

The RTC can operate in the backup mode that can greatly reduce its power consumption. This can be realized if all device circuits are powered, if the BUP# pin is set to logical 0, and if the power supply to all pins is stopped except for the RTCVDD pin. During this time, the RTC cannot be accessed.

24.2 Block Diagram

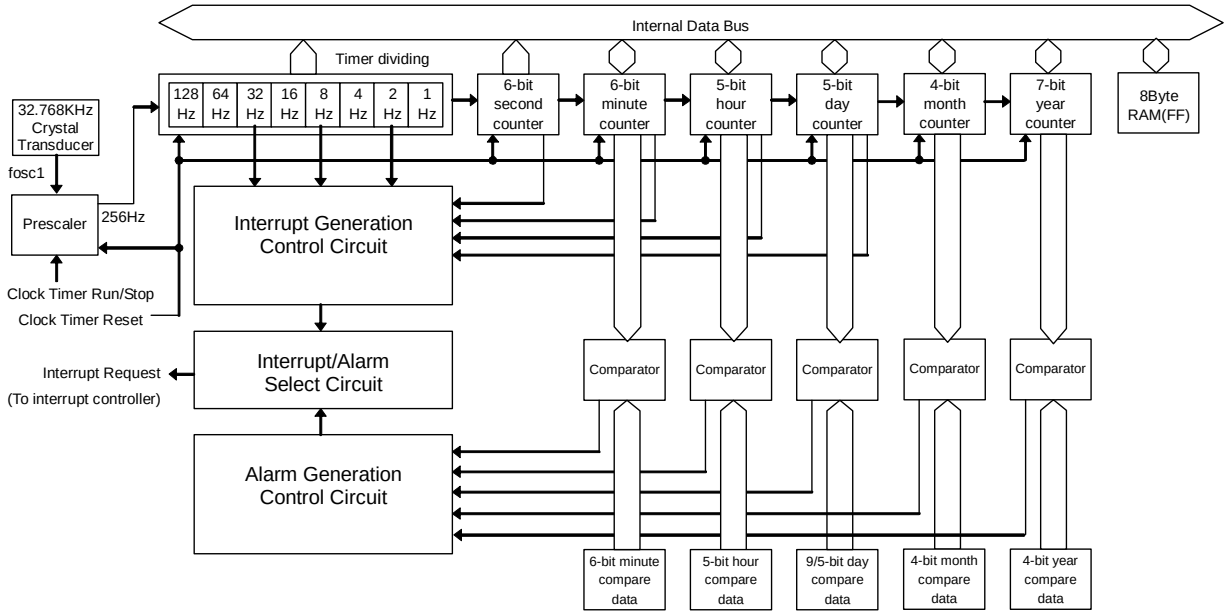


Fig.24.1 Block Diagram

24.3 External Pins

The following defines the RTC external pins.

Pin Name	Input/Output	Bus Width	Pin Functions	Multiplex Pin/Remarks
BUP#	Input	1	Switches the Backup mode.	
SYS_OSCI	Input	1	Crystal transducer connect pin	
SYS_OSCO	Output	1	Crystal transducer connect pin	

24.4 Registers

24.4.1 List of Registers

These registers have base address 0xFFFF_8000.

Table 24.1 List of Registers (Base Address: 0xFFFF_8000)

Address Offset	Register Name	Register Abbreviation	Default value	Data Access Size
0x00	RTC Run/Stop Control Register	xxx- --xx b	R/W	8 (/16/32)
0x04	RTC Interrupt Register	0x XXXX	R/W	16 (/32)
0x08	RTC Timer Dividing Register	xxxx xxxx b	R/(W)	8 (/16/32)
0x0C	RTC Second Counter Register	--xx xxxx b	R/W	8 (/16/32)
0x10	RTC Minute Counter Register	--xx xxxx b	R/W	8 (/16/32)
0x14	RTC Hour Counter Register	---x xxxx b	R/W	8 (/16/32)
0x18	RTC Day Counter Register	---x xxxx b	R/W	8 (/16/32)
0x1C	RTC Month Counter Register	---- xxxx b	R/W	8 (/16/32)
0x20	RTC Year Counter Register	-xxx xxxx b	R/W	8 (/16/32)
0x24	RTC Alarm Minute Compare Register	--xx xxxx b	R/W	8 (/16/32)
0x28	RTC Alarm Month Compare Register	---x xxxx b	R/W	8 (/16/32)
0x2C	RTC Alarm Day Compare Register	---x xxxx b	R/W	8 (/16/32)
0x30	RTC Alarm Month Compare Register	---- xxxx b	R/W	8 (/16/32)
0x34	RTC Alarm Year Compare Register	-xxx xxxx b	R/W	8 (/16/32)
0x38 - 0x3C	n/a			
0x40	RTC Test Register	---x xxxx b	R/W	8 (/16/32)
0x44	RTC Prescaler Register	-xxx xxxx b	R/(W)	8 (/16/32)
0x48	RTC Test Clock Register	xxxx xxxx b	RO	8 (/16/32)
0x4C – 0x5C	n/a			
0x60	RTC RAM0	xxxx xxxx b	R/W	8 (/16/32)
0x64	RTC RAM1	xxxx xxxx b	R/W	8 (/16/32)
0x68	RTC RAM2	xxxx xxxx b	R/W	8 (/16/32)
0x6C	RTC RAM3	xxxx xxxx b	R/W	8 (/16/32)
0x70	RTC RAM4	xxxx xxxx b	R/W	8 (/16/32)
0x74	RTC RAM5	xxxx xxxx b	R/W	8 (/16/32)
0x78	RTC RAM6	xxxx xxxx b	R/W	8 (/16/32)
0x7C	RTC RAM7	xxxx xxxx b	R/W	8 (/16/32)

24.4.2 Detailed Description of Registers

All reserved bits must be set to logical 0 unless otherwise specified. If these reserved bits are read by the software, those values should be ignored.

RTC Run/Stop Control Register (8 bit)

RTC[0x00] Default value = xxx- --xx b — Read/Write

BUSY	BUSYWIDTH[1:0]		Reserved			TCADJ	TCRUN
x	xx		—			—	0
RO	R/W					WO	R/W
7	6	5	4	3	2	1	Bit0

x: Undefined bit

Bit 7: **BUSY Busy (Read only)**

When this bit is logical 0, the software can write or read data into/from RTC register. When this bit is logical 1, the update cycle is running in the RTC. Therefore, this data is unreliable even if the registers are accessed by the software. This update cycle occurs once per second. This bit is set to logical 1 in the period being set by bits [6:5] (BUSYWIDTH) of RTC Run/Stop Control register. Therefore, if this bit is 1, you must until the preset period expires and access to registers again.

Bits [6:5]: **BUSYWIDTH busy interval**

Sets a busy interval.

00: Approx. 244 µsec
01: Approx. 122 µsec
10: Approx. 61 µsec
11: Reserved

This bit is not initialized by system reset.
To assure the normal operation, you must reset the system when the RTC is stopped.

Bits [4:2]: **Reserved Reserved**

Bit 1: **TCADJ RTC adjust (Write Only)**

0: Normal mode
When this bit is read, the signal is always set to logical 0.
1: Adjust RTC counter
If this bit is set to logical 1 and if bit 0 of RTC run/stop control register is set to logical 0 simultaneously, only the prescaler counter and the 128-1Hz counter are reset.

If the RTC is operating, this adjustment is made invalid.

Bit 0: **TCRUN**

Read
0: When RTC is stopped
1: When RTC is operating

Write
0: Stops the RTC.
1: Starts the RTC.

The operation or shutdown occurs after approximately 30 to 61 µsec for synchronization with the 32KHz clock.

RTC Interrupt Register (16 bit)

RTC[0x04] Default value = 0xXXXX Read/Write

Reserved					TCISE[2:0]		
—	—	—	—	—	x	x	x
					R/W	R/W	R/W
15	14	13	12	11	10	9	8

Reserved	TCASE[4:0]					TCIF	TCAF
—	x	x	x	x	x	x	x
	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [15:11]: **Reserved Reserved**

Bits [10:8]: **TCISE[2:0] interrupt source enable select bits**

000: Carry from “32Hz” (Once every 1/32 sec)
001: Carry from “8Hz” (Once every 1/8 sec)
010: Carry from “2Hz” (Once every 1/2 sec)
011: Carry from “1Hz” (Once every second)
100: Carry from “1 minute” (Once every minute)
101: Carry from “1 hour” (Once every hour)
110: Carry from “Day” (Once every day)
111: No interrupt source (Default)

Because the status change of carry is used as the interrupt source, the interrupt occurs only once in the specified internal.

Bits [6:2]: **TCASE[4:0] RTC alarm source select bit**

00000: No alarm (Default)
xxxx1: Minute alarm enable
xxx1x: Hour alarm enable
xx1xx: Day Hour alarm enable
x1xxx: Month alarm enable
1xxxx: Year alarm enable

If the alarm source counter being set to logical 1 matches the Alarm Compare register value, an interrupt occurs. If multiple sources have been enabled and if all of them match the Alarm Compare register value, an interrupt occurs.

This interrupt request continues to occur during the time when the counter matches the Alarm Compare register value. It continues even if you clear the interrupt request by setting the TCAF bit to logical 1. Therefore, if you wish to wait for the next interrupt using the same alarm setting, you need to disable all sources until the counter having the minimum time unit changes among the alarm sources you have specified. For example, if you have enabled the “Hour” and “Day” alarms, you must not select the same alarm source settings within one hour after an interrupt occurrence.

Bit 1: **An interrupt request flag from TCIF RTC timer**

Read
0: No pending interrupt exists.
1: A pending interrupt request from RTC timer exists.
Write
0: N/A
1: The interrupt request from RTC timer is cleared.

Bit 0: **An interrupt request flag from TCAF alarm**

Read
0: No pending interrupt exists.
1: A pending interrupt request from the alarm exists.
Write
0: N/A

1: The interrupt request from the alarm is cleared.

RTC Timer Dividing Register (8 bit)

RTC[0x008] Default value = xxxx xxxx b Read/(Write)

TCD[7:0]							
x	x	x	x	x	x	x	x
R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)
7	6	5	4	3	2	1	0

x: Undefined bit

Bit 7:	**TCD7: 1Hz indicator**	1: High; 0: Low
Bit 6:	**TCD6: 2Hz indicator**	1: High; 0: Low
Bit 5:	**TCD5: 4Hz indicator**	1: High; 0: Low
Bit 4:	**TCD4: 8Hz indicator**	1: High; 0: Low
Bit 3:	**TCD3: 16Hz indicator**	1: High; 0: Low
Bit 2:	**TCD2: 32Hz indicator**	1: High; 0: Low
Bit 1:	**TCD1: 64Hz indicator**	1: High; 0: Low
Bit 0:	**TCD0: 128Hz indicator**	1: High; 0: Low

This register is reset to 0 when bit 1 (TCADJ) of RTC Run/Stop Control register (RTC[0x00]) is set to logical 1. The software can consider this register as the up counter. Because this register does not have the circuit to synchronize the 32KHz signal with the system clock, the clock value must be read multiple times and the same value must be used as the correct one.

This register is write enabled only when the dividing counter is set to the write enabled mode by the RTC Test register (RTC[0x40]). Because this is the hardware test function, the data written in this register during dividing counter in the write enabled mode is unreliable.

RTC Second Counter Register (8 bit)

RTC[0x0C] Default value = --xx xxxx b Read/Write

Reserved		TCMD[5:0]					
—	—	x	x	x	x	x	x
		R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [7:6]: **Reserved Reserved**

Bits [5:0]: **TCMD[5:0]**

These 6 bits show the binary data indicating from 0 to 59 seconds.
TCMD5=MSB, TCMD0=LSB.
If a value greater than 59 is written, it is reflected as it is.

RTC Minute Counter Register (8 bit)

RTC[0x10] Default value = --xx xxxx Read/Write

Reserved		TCHD[5:0]					
—	—	x	x	x	x	x	x
		R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [7:6]: **Reserved Reserved**

Bits [5:0]: **TCHD[5:0]**
These 6 bits show the binary data indicating from 0 to 59 minutes.
TCHD5=MSB, TCHD0=LSB.
If a value greater than 59 is written, it is reflected as it is.

RTC Hour Counter Register (8 bit)
RTC[0x14] Default value = ---x xxxx b Read/Write

Reserved			TCDD[4:0]				
—	—	—	x	x	x	x	x
			R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [7:5]: **Reserved Reserved**

Bits [4:0]: **TCDD[4:0]**
These 5 bits show the binary data indicating from 0 to 23 hours.
TCDD4=MSB, TCDD0=LSB.
If a value greater than 23 is written, it is reflected as it is.

RTC Day Counter Register (8 bit)
RTC[0x18] Default value = ---x xxxx b Read/Write

Reserved			TCND [4:0]				
—	—	—	x	x	x	x	x
			R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [7:5]: **Reserved Reserved**

Bits [4:0]: **TCND[4:0]**
These 5 bits show the binary data indicating from the first to 31st day.
TCND4 = MSB, TCND0 = LSB.
If value 0 or a value greater than 31 is written, it is reflected as it is.

RTC Month Counter Register (8 bit)
RTC[0x1C] Default value = ---- xxxx b Read/Write

Reserved				TCDD[3:0]			
—	—	—	—	x	x	x	x
				R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [7:4]: **Reserved Reserved**

Bits [3:0]: **TCTD[3:0]**
These 4 bits show the binary data indicating from January to December.
TCTD3=MSB, TCTD0=LSB.
If value 0 or a value greater than 12 is written, it is reflected as it is.

RTC Year Counter Register (8 bit)

RTC[0x20] Default value = -xxx xxxx b Read/Write

Reserved	TCYD[6:0]						
—	x	x	x	x	x	x	x
	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bit 7: **Reserved Reserved**

Bits [6:0]: **TCYD[6:0]**
These 7 bits show the binary data indicating from year 01 to year 99.
TCYD6=MSB, TCTD0=LSB.
If value 0 or a value greater than 99 is written, it is reflected as it is.

RTC Alarm Minute Compare Register (8 bit)

RTC[0x24] Default value = --xx xxxx b Read/Write

Reserved		TCCH[5:0]					
—	—	x	x	x	x	x	x
		R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [7:6]: **Reserved Reserved**

Bits [5:0]: **TCCH[5:0]**
These 6 bits show the binary data indicating from 0 to 59 minutes.
TCCH5=MSB, TCCH0=LSB.
If a value greater than 59 is written, it is reflected as it is.

RTC Alarm Hour Compare Register (8 bit)

RTC[0x28] Default value = ---x xxxx b Read/Write

Reserved			TCCD[4:0]				
—	—	—	x	x	x	x	x
			R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [7:5]: **Reserved Reserved**

Bits [4:0]: **TCCD[4:0]**
These 5 bits show the binary data indicating from 0 to 23 hours.
TCCD4=MSB, TCCD0=LSB.
If a value greater than 23 is written, it is reflected as it is.

RTC Alarm Day Compare Register (8 bit)

RTC[0x2C] Default value = ---- ---x xxxx xxxx b Read/Write

Reserved			TCCN[4:0]				
—	—	—	X	x	x	x	x
			R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [4:0]: **TCCN[4:0]**

These 4 bits show the binary data indicating from 0 to 31st day.
TCCN4=MSB, TCCN0=LSB.
If value 0 or a value greater than 31 is written, it is reflected as it is.

RTC Alarm Month Compare Register (8 bit)

RTC[0x30] Default value = ---- xxxx b Read/Write

Reserved				TCCT[3:0]			
—	—	—	—	x	x	x	x
				R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit (b)

Bits [7:4]: **Reserved Reserved**

Bits [3:0]: **TCCT[3:0]**

These 4 bits show the binary data indicating from 0 to 15 months.
TCCT3=MSB, TCCT0=LSB.
If value 0 or a value greater than 12 is written, it is reflected as it is.

RTC Alarm Year Compare Register (8 bit)

RTC[0x34] Default value = -xxx xxxx b Read/Write

Reserved	TCCY[6:0]						
—	x	x	x	x	x	x	x
	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bit 7: **Reserved Reserved**

Bits [3:0]: **TCCY[6:0]**

These 7 bits show the binary data indicating from 0 to 127 years.
TCCY6=MSB, TCCY0=LSB.
If value 0 or a value greater than 99 is written, it is reflected as it is.

RTC Test Register (8 bit)

RTC[0x40] Default value = ---x xxxx b Read/Write

Reserved			RTST4	RTST3	RTST2	RTST1	RTST0
—			x	x	x	x	x
—			R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [7:5]: **Reserved Reserved**

Bits [4:1]: **RTST[4:1] Test mode set**

0000: Dividing counter write-enabled mode
The dividing counter is write enabled in this mode.

xx10: Test Clock mode
In the Test Clock mode, clock signals are written in the RTC Test Clock register instead of the 32KHz clock, and the generated pulses are used.

x1xx: Second, minute, hour, day, year and month counter carry bypass mode
Carries of second, minute, hour, day, year and month counters are used as clocks.

1xxx: Dividing counter carry bypass mode
The dividing counter carry is used as the clock.

Bit 0 **RTST0 Test mode enable**

If logical 1 is written in the previous time and if logical 0 is written next, the Test mode is selected.

RTC Prescaler Register (8 bit)

RTC[0x44] Default value = -xxx xxxx b Read/(Write)

Reserved	TCP [6:0]						
—	x	x	x	x	x	x	x
	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)
7	6	5	4	3	2	1	0

x: Undefined bit

Bits 7: **Reserved Reserved**

Bits [6:0]: **TCP[6:0]**

These 7 bits show the prescaler value.
TCP8=MSB, TCP0=LSB.
This register is not initialized by system reset.
This register is write enabled only when the dividing counter is set to the write enabled mode by the RTC Test register. Because this is the hardware test function, the data written in this register during dividing counter in the write enabled mode is unreliable.

RTC Test Clock Register (8 bit)

RTC[0x48] Default value = -xxx xxxx b Write Only

TSTCLK[7:0]							
x	x	x	x	x	x	x	x
WO	WO	WO	WO	WO	WO	WO	WO
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [7:0]: **TSTCLK[7:0] Test clock**

If the Test Clock mode is set by the RTC Test register and if any value is written in this register, a single pulse of test clock is generated. Because this function is for the hardware test only, the operation is unreliable if this register is used.

RTC RAM0-7 Register (8 bit)

RTC[0x60 - 0x7C] Default value = xxxx xxxx b Read/Write

TCRAM0[7:0]							
x	x	x	x	x	x	x	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined bit

Bits [7:0]: **TCRAM0 to 7[7:0]**
These 8 bits can be used as the RAM.
TCRAM0-7[7]=MSB, TCRAM0-7 [0]=LSB.
This register is not initialized by the system reset or ADJ bit.

24.5 Setting RTC Registers

If the Test register or Test Clock register is used or if the device is operated in the Test mode, the resulting operation is unreliable and it is not guaranteed. Also, all registers are NOT initialized by the system reset.

24.5.1 Initialization after Power-On

All registers in the RTC are unstable immediately after the system power-On. The following initialization procedure must be used.

A) Set all bits of RTC Test register (RTC[0x40]) to “0x00” twice.
B) Because you must stop the RTC during various settings, set bit 0 of RTC Run/Stop Control register (RTC[0x00]) to “0b0” to stop writing.
C) Make sure that bit 0 of RTC Run/Stop Control register (RTC[0x00]) is “0b0.”
(When this bit is “0b0,” the RTC has been stopped. When it is “0b1,” the RTC is operating. You must securely check that the RTC has been stopped. This is because the RTC stops with a delay of 30 to 61 µsec so that various register settings are synchronized with the 32KHz system circuit operations. Bit 0 of RTC Run/Stop Control register (RTC[0x00]) is set based on the Run/Stop Control signal of the 32KHz system circuit that has been synchronized again. Therefore, a certain delay occurs due to the synchronization.)
D) Set bits 1 and 0 of RTC Run/Stop Control register (RTC[0x00]) to “0b10” so that the prescaler and the dividing timer are reset.
(Because all registers are unstable during power-On, both the prescaler and dividing timer can be initialized to zero (0) by this operation. Note that bit 1 must be set to “0b1” and bit 0 must be set to “0b0.”)
E) Before setting the date and time, be sure to disable an interrupt and clear all flags. Write value “0x0703” in the RTC Interrupt register (RTC[0x04]) so that the interrupt controller is not affected.
F) After you have set the RTC Interrupt register (RTC[0x04]), set the date and time. Set the Second, Minute, Hour, Day, Month and Year counters correctly.
(You must enter these values correctly because the counter may malfunction if you write nonexistent time and if a lower digit is carried up.)
G) To set an alarm, set the Minute, Hour, Day, Month, and Year Alarm Compare registers. Also, if you set an interrupt, enable both the cycle interrupt and the alarm interrupt of RTC Interrupt register (RTC[0x04]).
H) We recommend you to clear an interrupt flag and initialize the 8-byte RAM because its initial value is still unreliable.
I) Finally, set the BUSY signal width of the RTC Run/Stop Control register (RTC[0x00]). We recommend you to set the BUSY signal width that is greater than the time the software process ends. For example, if the setting process (including a process that is terminated by another process) can ends within 100 µsec,

you should set the BUSYWIDTH value to "01" (122 μsec). Because the RTC has been stopped, the Second counter setting through the BUSY signal width setting may be made in a different order. You can set them in any order without any problem.

J) After these settings, set bit 0 of RTC Run/Stop Control register (RTC[0x00]) to "0b1" so that the RTC starts. The RTC starts operating after 30 to 61 μsec.

24.5.2 Stop and Restart of Operations

Set bit 0 of RTC Run/Stop Control register (RTC[0x00]) to "0b0" and wait until bit 0 of RTC Run/Stop Control register (RTC[0x00]) is set to "0b0" (approximately 30 to 61 μsec). When bit 0 is set to "0b0," the operation is stopped. Each counter value is held even when the operation is stopped. To restart the operation, set bit 0 of RTC Run/Stop Control register (RTC[0x00]) to "0b1." Counting starts from the held value. If you restart the operation after interrupt processing, clear the interrupt flag first. Then, set bits 1 and 0 of RTC Run/Stop Control register to "10" and reset both prescaler and dividing timer. Then, set bit 0 of RTC Run/Stop Control register (RTC[0x00]) to "0b1" so that the write operation restarts.

24.5.3 Repeated Setting without Operation Stop

If you access to any bit during operation, except for bit 0 of RTC Run/Stop Control register (RTC[0x000]), the operation becomes unreliable. Especially, if you reset the prescaler and the dividing timer during operation, a malfunction may result.

You can change the settings of RTC Interrupt register and the Minute, Hour, Day, Month and Year Alarm Compare register when the BUSY bit is logical 0. However, disable an RTC interrupt using the interrupt controller before changing the register settings. After the change, you must clear the interrupt controller and RTC interrupt flag. Then, you must enable an RTC interrupt using the interrupt controller. This is required to inhibit unnecessary interrupts that may occur during change of settings.

You can change the Second, Minute, Hour, Day, Month and Year Counter register contents when the BUSY bit is logical 0. Because the repeated setting of this counter register completes instantaneously, you can complete this setting by checking the BUSY bit only once if you have disabled any interrupt of peripheral circuits during this time.

24.5.4 Repeated Setting during Operation after System Reset

If a system reset occurs when the RTC is operating, the operation continues without affecting on the RTC. (All registers in the RTC are not affected by the system reset.)

The RTC continues to operate without missing the RTCVDD signal (1.8V typical).

24.5.5 Cautions during Programming

- All registers in the RTC are unstable immediately after the power-On. Initialize the registers by following the procedure given in Section 24.5.1 "Initialization after Power-On."
- If you reset the prescaler and the dividing timer, first stop the RTC from operating. Then, set bits 1 and 0 of RTC Run/Stop Control register (RTC[0x00]) to "0b10." You must set bit 1 to logical 1 and set bit 1 to logical 0. If the system is reset when the RTC is operating, this setting is made invalid.
- If you change the interrupt cause and alarm cause settings to prevent undesired interrupts, first disable the RTC interrupt using the interrupt controller. Also, before enabling the interrupts, be sure to clear the interrupt cause flag and the alarm cause flag.

25. Watchdog Timer (WDT)

25.1 Description

The watchdog timer (WDT) is a system overrun monitoring unit, and it consists of 16-bit down counter whose value can be preset by the software. Counting down starts from the initially set value, and when the counter reaches zero (0), an interrupt request or a reset request is generated by following the settings of watchdog timer operation select bit (WDT[0x08], bit 4). The software periodically loads the preset data on the counter so that the counter does not reach zero (0). Therefore, if an interrupt request or a reset request occurs, it means that the program has failed to execute normally.

After the counter has reached 0, it returns to "0xFFFF." The counter value can be read by the software any time when desired. Note that the counter is set to "0xFFFF" when the system is reset or when the watchdog timer enable bit (WDT[0x08], bit 5) is set to logical 0. This counting is stopped if the watchdog timer enable bit is set to logical 0. The watchdog timer operates even in the HALT mode.

The watchdog timer uses the APB clock as the source clock. Because the watchdog timer has the built-in prescaler, this timer can control the dividing ratio of source clock and set the frequency of the count clock.

25.2 Block Diagram

The following shows a watchdog timer block diagram.

The watchdog timer consists of the register block (Read, Load and Control registers, the 16-bit Down counter, and the prescaler block having 11-bit Down counter.

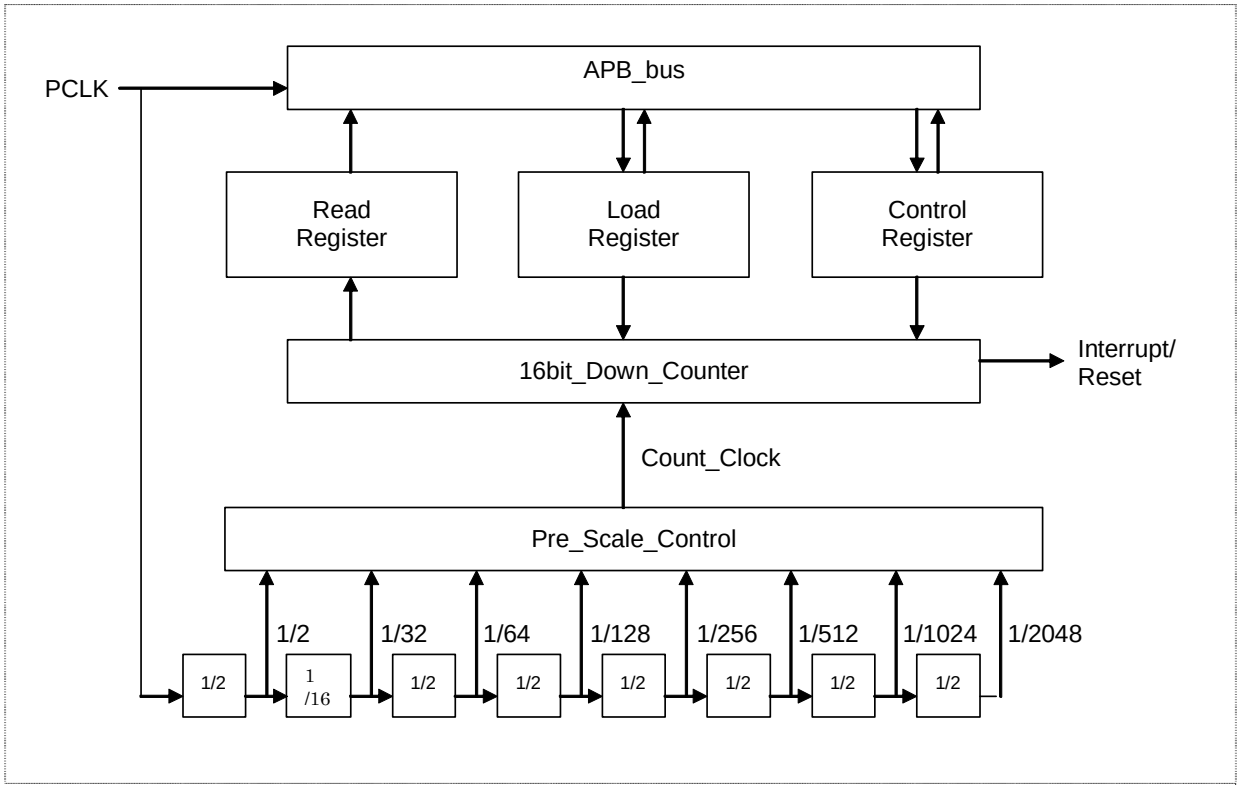


Fig.25.1 WDT Block Diagram

25.3 External Pins

There is no external pin that relates to the watchdog timer.

25.4 Registers

25.4.1 List of Registers

These registers have base address 0xFFFF_C000.

Table 25.1 List of Registers (Base Address: 0xFFFF_C000)

Address Offset	Register Name	Default Value	R/W	Data Access Size
0x00	Watchdog Timer Load register	0x0000_FFFF	R/W	16 (/32)
0x04	Watchdog Timer Count register	0x0000_FFFF	RO	16 (/32)
0x08	Watchdog Timer Control register	0x0000_0000	R/W	16 (/32)

25.4.2 Detailed Description of Registers

The Watchdog Timer Control registers have base address 0xFFFF_C000. If not reserved, all register bits are zeros (0s) unless otherwise noted.

Watchdog Timer Load Register

WDT[0x00] Default value = 0x0000_FFFF Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timer load value															

Bits [15:0]: **Timer load value bits [15:0]**
If data is written here, it is loaded on the counter.

Watchdog Timer Count Register

WDT[0x04] Default value = 0x0000_FFFF Read Only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Current timer count value															

Bits [15:0]: **Current timer count value bits [15:0]**
The current counter value can be read.

Watchdog Timer Control Register

WDT[0x08] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								n/a							
Reserved (Fixed to A5h)								n/a	WDT status (RO)	WDT enable	WDT operation select	n/a	Prescaler dividing ratio		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [15:8]: **Reserved**

Must always be 0xA5 during writing.

Bit 6: **Watchdog timer status (Read only)**

Indicates whether the counter has reached zero (0) or not.

0: The counter is not zero.

1: The counter is zero.

This bit is set to logical 1 when the counter reaches zero (0), and its status is kept. To clear this status, you must once set the watchdog timer enable bit (bit 5 of this register) to logical 0 so that the watchdog timer is reset.

Bit 5: **Watchdog timer enable**

Enables or disables the watchdog timer counting.

0 (r/w): Disables the timer.

If this bit is set to logical 0, the current counting is stopped. At the same time, the counter is reset to "0xFFFF" and the watchdog timer status bit (bit 6 of this register) is also reset to logical 0.

1 (r/w): Enables the timer.

When this bit is set to logical 1, the watchdog timer starts countdown.

Bit 4: **Watchdog timer operation**

Specifies an operation that starts when the counter reaches zero (0).

0 (r/w): Outputs a reset request signal.

When the counter reaches zero (0), the watchdog timer outputs a reset request signal (logical High WRST signal) to the system controller. The reset request signal returns to logical Low when the counter becomes a non-zero value.

1 (r/w): Outputs an interrupt request signal.

When the counter reaches zero (0), the watchdog timer outputs an interrupt request signal (logical High WINT signal). The interrupt request signal returns to logical Low when the watchdog timer enable bit (bit 5 of this register) is disabled.

Bits [2:0]: **Prescaler dividing ratio bits [2:0]**

Select a dividing ratio of the prescaler for count clock generation. The prescaler uses the APB clock as the source clock.

Bits [2:0]	Prescaler dividing ratio n*
000	2
001	32
010	64
011	128
100	256
101	512
110	1024
111	2048

*: Prescaler dividing ratio $n = 2$ ・・・ If Bits[2:0]=0

$= 2^{(4 + Bits[2:0])}$ ・・・ If Bits[2:0] are not equal to 0

The count clock frequency ("f" $_{count\ clock}$) can be expressed using the APB system clock (PCLK) frequency (f_{PCLK}) and the prescaler dividing ratio (n) as follows:

$$f_{count\ clock} = f_{PCLK} / n$$

26. GPIO

26.1 Description

This functional block consists of the GPIO function and the registers which are used to select pin functions multiplexed with GPIO pins.

This GPIO has the following features.

- Supports ten (10) 8-bit GPIO ports such as:
  GPIOA, GPIOB, GPIOC, GPIOD, GPIOE, GPIOF, GPIOG, GPIOH, GPIOJ, GPIOK
- Supports a single 2-bit GPIO port such as:
  GPIOI
- Allows changing the input or output direction of all GPIO pins separately.
- Some pins are multiplexed with other functions.
- Allows selecting the multi-function I/O pins using the Pin Function register.
- Supports the interrupt input function at GPIOA and GPIOB pins.
- Allows selecting an interrupt signal edge, its level, and logical High or Low polarity from GPIOA and GPIOB pins.

26.2 External Pins

The following defines the external pins relating to the GPIO.

Port	Pin Name	Input/Output	Pin Functions	Multiplex Pin/Remarks		
				Shared	Function 1	Function 2
A	GPIOA0	Input/Output	General-purpose I/O port A0 input/output		TXD1	
	GPIOA1	Input/Output	General-purpose I/O port A1 input/output		RXD1	
	GPIOA2	Input/Output	General-purpose I/O port A2 input/output		RTS1	I2S1_WS
	GPIOA3	Input/Output	General-purpose I/O port A3 input/output		CTS1	I2S1_SCK
	GPIOA4	Input/Output	General-purpose I/O port A4 input/output		TXD2	
	GPIOA5	Input/Output	General-purpose I/O port A5 input/output		RXD2	
	GPIOA6	Input/Output	General-purpose I/O port A6 input/output		RTS2	SCL
	GPIOA7	Input/Output	General-purpose I/O port A7 input/output		CTS2	SDA
B	GPIOB0	Input/Output	General-purpose I/O port B0 input/output	INT0	I2S0_WS	
	GPIOB1	Input/Output	General-purpose I/O port B1 input/output	INT1	I2S0_SCK	
	GPIOB2	Input/Output	General-purpose I/O port B2 input/output	INT2	I2S0_SD	
	GPIOB3	Input/Output	General-purpose I/O port B3 input/output	INT3	I2S1_SD	
	GPIOB4	Input/Output	General-purpose I/O port B4 input/output	INT4	TimerA0out	
	GPIOB5	Input/Output	General-purpose I/O port B5 input/output	INT5	TimerA1out	
	GPIOB6	Input/Output	General-purpose I/O port B6 input/output	INT6	TimerA2out	
	GPIOB7	Input/Output	General-purpose I/O port B7 input/output	INT7	TimerBIn	
C	GPIOC0	Input/Output	General-purpose I/O port C0 input/output		TimerB0IO	
	GPIOC1	Input/Output	General-purpose I/O port C1 input/output		TimerB1IO	
	GPIOC2	Input/Output	General-purpose I/O port C2 input/output		TimerB2IO	
	GPIOC3	Input/Output	General-purpose I/O port C3 input/output		TimerB3IO	UART3_CLK
	GPIOC4	Input/Output	General-purpose I/O port C4 input/output		SPI_SS	TXD3
	GPIOC5	Input/Output	General-purpose I/O port C5 input/output		SPI_SCLK	RXD3
	GPIOC6	Input/Output	General-purpose I/O port C6 input/output		SPI_MISO	RTS3
	GPIOC7	Input/Output	General-purpose I/O port C7 input/output		SPI_MOSI	CTS3
D	GPIOD0	Input/Output	General-purpose I/O port D0 input/output		MA20	
	GPIOD1	Input/Output	General-purpose I/O port D1 input/output		MA21	
	GPIOD2	Input/Output	General-purpose I/O port D2 input/output		MA22	
	GPIOD3	Input/Output	General-purpose I/O port D3 input/output		MA23	
	GPIOD4	Input/Output	General-purpose I/O port D4 input/output		MCS2#	
	GPIOD5	Input/Output	General-purpose I/O port D5 input/output		MCS3#	
	GPIOD6	Input/Output	General-purpose I/O port D6 input/output		SDDQM2	
	GPIOD7	Input/Output	General-purpose I/O port D7 input/output		SDDQM3	
E	GPIOE0	Input/Output	General-purpose I/O port E0 input/output		CM1DATA0	
	GPIOE1	Input/Output	General-purpose I/O port E1 input/output		CM1DATA1	
	GPIOE2	Input/Output	General-purpose I/O port E2 input/output		CM1DATA2	
	GPIOE3	Input/Output	General-purpose I/O port E3 input/output		CM1DATA3	
	GPIOE4	Input/Output	General-purpose I/O port E4 input/output		CM1DATA4	
	GPIOE5	Input/Output	General-purpose I/O port E5 input/output		CM1DATA5	
	GPIOE6	Input/Output	General-purpose I/O port E6 input/output		CM1DATA6	
	GPIOE7	Input/Output	General-purpose I/O port E7 input/output		CM1DATA7	
F	GPIOF0	Input/Output	General-purpose I/O port F0 input/output		CM1VREF	
	GPIOF1	Input/Output	General-purpose I/O port F1 input/output		CM1HREF	
	GPIOF2	Input/Output	General-purpose I/O port F2 input/output		CM1CLKOUT	
	GPIOF3	Input/Output	General-purpose I/O port F3 input/output		CM1CLKIN	
	GPIOF4	Input/Output	General-purpose I/O port F4 input/output		CM2VREF	
	GPIOF5	Input/Output	General-purpose I/O port F5 input/output		CM2HREF	
	GPIOF6	Input/Output	General-purpose I/O port F6 input/output		CM2CLKOUT	
	GPIOF7	Input/Output	General-purpose I/O port F7 input/output		CM2CLKIN	

Port	Pin Name	Input/Output	Pin Functions	Multiplex Pin/Remarks		
				Shared	Function 1	Function 2
G	GPIOG0	Input/Output	General-purpose I/O port G0 input/output		CM2DATA0	
	GPIOG1	Input/Output	General-purpose I/O port G1 input/output		CM2DATA1	
	GPIOG2	Input/Output	General-purpose I/O port G2 input/output		CM2DATA2	
	GPIOG3	Input/Output	General-purpose I/O port G3 input/output		CM2DATA3	
	GPIOG4	Input/Output	General-purpose I/O port G4 input/output		CM2DATA4	
	GPIOG5	Input/Output	General-purpose I/O port G5 input/output		CM2DATA5	
	GPIOG6	Input/Output	General-purpose I/O port G6 input/output		CM2DATA6	
	GPIOG7	Input/Output	General-purpose I/O port G7 input/output		CM2DATA7	
H	GPIOH0	Input/Output	General-purpose I/O port H0 input/output		CFCE2#	SDMDATA0
	GPIOH1	Input/Output	General-purpose I/O port H1 input/output		CFCE1#	SDMDATA1
	GPIOH2	Input/Output	General-purpose I/O port H2 input/output		CFIORD#	SDMDATA2
	GPIOH3	Input/Output	General-purpose I/O port H3 input/output		CFIOWR#	SDMDATA3
	GPIOH4	Input/Output	General-purpose I/O port H4 input/output		CFWAIT#	SDMCMD
	GPIOH5	Input/Output	General-purpose I/O port H5 input/output		CFRST	SDMCLK
	GPIOH6	Input/Output	General-purpose I/O port H6 input/output		CFIRQ	SDMCD#
	GPIOH7	Input/Output	General-purpose I/O port H7 input/output		CFSTSCHG#	SDMWP
I	GPIOI0	Input/Output	General-purpose I/O port I0 input/output		CFDEN#	SDMGPO
	GPIOI1	Input/Output	General-purpose I/O port I1 input/output		CFDDIR	
	~~GPIOI2~~	~~Input/Output~~	~~General-purpose I/O port I2 input/output~~			
	~~GPIOI3~~	~~Input/Output~~	~~General-purpose I/O port I3 input/output~~			
	~~GPIOI4~~	~~Input/Output~~	~~General-purpose I/O port I4 input/output~~			
	~~GPIOI5~~	~~Input/Output~~	~~General-purpose I/O port I5 input/output~~			
	~~GPIOI6~~	~~Input/Output~~	~~General-purpose I/O port I6 input/output~~			
	~~GPIOI7~~	~~Input/Output~~	~~General-purpose I/O port I7 input/output~~			
J	GPIOJ0	Input/Output	General-purpose I/O port J0 input/output		SDD16	
	GPIOJ1	Input/Output	General-purpose I/O port J1 input/output		SDD17	
	GPIOJ2	Input/Output	General-purpose I/O port J2 input/output		SDD18	
	GPIOJ3	Input/Output	General-purpose I/O port J3 input/output		SDD19	
	GPIOJ4	Input/Output	General-purpose I/O port J4 input/output		SDD20	
	GPIOJ5	Input/Output	General-purpose I/O port J5 input/output		SDD21	
	GPIOJ6	Input/Output	General-purpose I/O port J6 input/output		SDD22	
	GPIOJ7	Input/Output	General-purpose I/O port J7 input/output		SDD23	
K	GPIOK0	Input/Output	General-purpose I/O port K0 input/output		SDD24	
	GPIOK1	Input/Output	General-purpose I/O port K1 input/output		SDD25	
	GPIOK2	Input/Output	General-purpose I/O port K2 input/output		SDD26	
	GPIOK3	Input/Output	General-purpose I/O port K3 input/output		SDD27	
	GPIOK4	Input/Output	General-purpose I/O port K4 input/output		SDD28	
	GPIOK5	Input/Output	General-purpose I/O port K5 input/output		SDD29	
	GPIOK6	Input/Output	General-purpose I/O port K6 input/output		SDD30	
	GPIOK7	Input/Output	General-purpose I/O port K7 input/output		SDD31	

26.3 Registers

26.3.1 List of Registers

These registers have base address 0xFFFF_1000.

Table 26.1 List of Registers (Base Address: 0xFFFF_1000)

Address Offset	Register Name	Abbreviation Name	Default Value	R/W	Data Access Size
0x00	GPIOA Data register	GPIOA_DATA	0x0000	R/W	8 (/16/32) *1
0x04	GPIOA Pin Function register	GPIOA_FNC	0x0000	R/W	16 (/32) *2
0x08	GPIOB Data register	GPIOB_DATA	0x0000	R/W	8 (/16/32) *1
0x0C	GPIOB Pin Function register	GPIOB_FNC	0x0000	R/W	16 (/32) *2
0x10	GPIOC Data register	GPIOC_DATA	0x0000	R/W	8 (/16/32) *1
0x14	GPIOC Pin Function register	GPIOC_FNC	0x0000	R/W	16 (/32) *2
0x18	GPIOD Data register	GPIOD_DATA	0x0000	R/W	8 (/16/32) *1
0x1C	GPIOD Pin Function register	GPIOD_FNC	0x0000	R/W	16 (/32) *2
0x20	GPIOE Data register	GPIOE_DATA	0x0000	R/W	8 (/16/32) *1
0x24	GPIOE Pin Function register	GPIOE_FNC	0x0000	R/W	16 (/32) *2
0x28	GPIOF Data register	GPIOF_DATA	0x0000	R/W	8 (/16/32) *1
0x2C	GPIOF Pin Function register	GPIOF_FNC	0x0000	R/W	16 (/32) *2
0x30	GPIOG Data register	GPIOG_DATA	0x0000	R/W	8 (/16/32) *1
0x34	GPIOG Pin Function register	GPIOG_FNC	0x0000	R/W	16 (/32) *2
0x38	GPIOH Data register	GPIOH_DATA	0x0000	R/W	8 (/16/32) *1
0x3C	GPIOH Pin Function register	GPIOH_FNC	0x0000	R/W	16 (/32) *2
0x40	GPIOI Data register	GPIOI_DATA	0x0000	R/W	8 (/16/32) *1
0x44	GPIOI Pin Function register	GPIOI_FNC	0x0000	R/W	16 (/32) *2
0x48	GPIOJ Data register	GPIOJ_DATA	0x0000	R/W	8 (/16/32) *1
0x4C	GPIOJ Pin Function register	GPIOJ_FNC	0x0000	R/W	16 (/32) *2
0x50	GPIOK Data register	GPIOK_DATA	0x0000	R/W	8 (/16/32) *1
0x54	GPIOK Pin Function register	GPIOK_FNC	0x0000	R/W	16 (/32) *2
0x58-5C	N/a				
0x60	GPIOA&B IRQ Type register	GPIOAB_ITYP	0x0000	R/W	16 (/32) *2
0x64	GPIOA&B IRQ Polarity register	GPIOAB_IPOL	0x0000	R/W	16 (/32) *2
0x68	GPIOA&B IRQ Enable register	GPIOAB_IEN	0x0000	R/W	16 (/32) *2
0x6C	GPIOA&B IRQ Status & Clear register	GPIOAB_ISTS	0x0000	R/W	16 (/32) *2

*1: Access to 8-, 16- or 32-bit data is enabled
*2: Access to 16- or 32-bit data is enabled

26.3.2 Detailed Description of Registers

26.3.2.1 GPIOA Registers

GPIOA Data Register (GPIOA_DATA)

GPIO[0x00] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
GPIOADATA [7:0]							

This is the GPIOA data register. The register is write and read enabled. If the GPIOA is set to output, the contents of this register are read. If GPIOA is set to input, the pin status can be read.

GPIOA Pin Function Register (GPIOA_FNC)

GPIO[0x04] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
GPA7MD [1:0]		GPA6MD [1:0]		GPA5MD [1:0]		GPA4MD [1:0]	
7	6	5	4	3	2	1	0
GPA3MD [1:0]		GPA2MD [1:0]		GPA1MD [1:0]		GPA0MD [1:0]	

Selects a GPIOA pin function. Two bits are used to select a pin function for each GPIOA port.

Table 26.2 Port-A Pin Select Function

GPAxMD1	GPAxMD0	Pin Function (“x” identifies the bit position on the port.)
0	0	GPIOAx port input (Default)
0	1	Function 1 other than GPIO
1	0	GPIOAx port output
1	1	Function 2 other than GPIO (Only GPIOA[7:6], [3:2] can be set.)

26.3.2.2 GPIOB Register

GPIOB Data Register (GPIOB_DATA)

GPIO[0x08] Default value = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
GPIOBDATA [7:0]							
7	6	5	4	3	2	1	0

This is the GPIOB data register. Data can be written into and read from the register. If the GPIOB is set to output, data is read from the register. If set to input, the pin status can be read.

GPIOB Pin Function Register (GPIOB_FNC)

GPIO[0x0C] Default value = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPB7MD [1:0]		GPB6MD [1:0]		GPB5MD [1:0]		GPB4MD [1:0]	
15	14	13	12	11	10	9	8
GPB3MD [1:0]		GPB2MD [1:0]		GPB1MD [1:0]		GPB0MD [1:0]	
7	6	5	4	3	2	1	0

Selects a GPIOB pin function. Two bits are used to select a pin function for each GPIOB port.

Table 26.3 Port-B Pin Select Function

GPBxMD1	GPBxMD0	Pin Function (“x” identifies the bit position on the port.)
0	0	GPIOBx port input (Default)
0	1	Function 1 other than GPIO
1	0	GPIOBx port output
1	1	Reserved

26.3.2.3 GPIOC Register

GPIOC Data Register (GPIOC_DATA)

GPIO[0x10] Default value = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
GPIOCDATA [7:0]							
7	6	5	4	3	2	1	0

This is the GPIOC data register. Data can be written into and read from the register. If the GPIOC is set to output, data is read from the register. If set to input, the pin status can be read.

GPIOC Pin Function Register (GPIOC_FNC)

GPIO[0x14] Default value = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPC7MD [1:0]		GPC6MD [1:0]		GPC5MD [1:0]		GPC4MD [1:0]	
15	14	13	12	11	10	9	8
GPC3MD [1:0]		GPC2MD [1:0]		GPC1MD [1:0]		GPC0MD [1:0]	
7	6	5	4	3	2	1	0

Selects a GPIOC pin function. Two bits are used to select a pin function for each GPIOC port.

Table 26.4 Port-C Pin Select Function

GPCxMD1	GPCxMD0	Pin Function (“x” identifies the bit position on the port.)
0	0	GPIOCx port input (Default)
0	1	Function 1 other than GPIO
1	0	GPIOCx port output
1	1	Function 2 other than GPIO (Only GPIOC[7:3] can be set.)

26.3.2.4 GPIOD Register

GPIOD Register (GPIOD_DATA)							
GPIO[0x18] Default value = 0x0000_0000							Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
GPIODDATA [7:0]							
7	6	5	4	3	2	1	0

This is the GPIOD data register. Data can be written into and read from the register. If the GPIOD is set to output, data is read from the register. If set to input, the pin status can be read.

GPIOD Pin Function Register (GPIOD_FNC)							
GPIO[0x1C] Default value = 0x0000_0000							Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPD7MD [1:0]		GPD6MD [1:0]		GPD5MD [1:0]		GPD4MD [1:0]	
15	14	13	12	11	10	9	8
GPD3MD [1:0]		GPD2MD [1:0]		GPD1MD [1:0]		GPD0MD[1:0]	
7	6	5	4	3	2	1	0

Selects a GPIOD pin function. Two bits are used to select a pin function for each GPIOD port.

Table 26.5 Port-D Pin Select Function

GPDxMD1	GPDxMD0	Pin Function (“x” identifies the bit position on the port.)
0	0	GPIODx port input (Default)
0	1	Function 1 other than GPIO
1	0	GPIODx port output
1	1	Reserved

26.3.2.5 GPIOE Register

GPIOE Data Register (GPIOE_DATA)

GPIO[0x20] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
GPIOEDATA [7:0]							
7	6	5	4	3	2	1	0

This is the GPIOE data register. Data can be written into and read from the register. If the GPIOE is set to output, data is read from the register. If set to input, the pin status can be read.

GPIOE Pin Function Register (GPIOE_FNC)

GPIO[0x24] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPE7MD [1:0]		GPE6MD [1:0]		GPE5MD [1:0]		GPE4MD [1:0]	
15	14	13	12	11	10	9	8
GPE3MD [1:0]		GPE2MD [1:0]		GPE1MD [1:0]		GPE0MD [1:0]	
7	6	5	4	3	2	1	0

Selects a GPIOE pin function. Two bits are used to select a pin function for each GPIOE port.

Table 26.6 Port-E Pin Select Function

GPExMD1	GPExMD0	Pin Function (“x” identifies the bit position on the port.)
0	0	GPIOEx port input (Default)
0	1	Function 1 other than GPIO
1	0	GPIOEx port output
1	1	Reserved

26.3.2.6 GPIOF Register

GPIOF Data Register (GPIOF_DATA)

GPIO[0x28] Default value = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
GPIOFDATA [7:0]							
7	6	5	4	3	2	1	0

This is the GPIOF data register. Data can be written into and read from the register. If the GPIOF is set to output, data is read from the register. If set to input, the pin status can be read.

GPIOF Pin Function Register (GPIOF_FNC)

GPIO[0x2C] Default value = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPF7MD [1:0]		GPF6MD [1:0]		GPF5MD [1:0]		GPF4MD [1:0]	
15	14	13	12	11	10	9	8
GPF3MD [1:0]		GPF2MD [1:0]		GPF1MD [1:0]		GPF0MD [1:0]	
7	6	5	4	3	2	1	0

Selects a GPIOF pin function. Two bits are used to select a pin function for each GPIOF port.

Table 26.7 Port-F Pin Select Function

GPFxMD1	GPFxMD0	Pin Function (“x” identifies the bit position on the port.)
0	0	GPIOFx port input (Default)
0	1	Function 1 other than GPIO
1	0	GPIOFx port output
1	1	Reserved

26.3.2.7 GPIOG Register

GPIOG Data Register (GPIOG_DATA)

GPIO[0x30] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
GPIOGDATA [7:0]							
7	6	5	4	3	2	1	0

This is the GPIOG data register. Data can be written into and read from the register. If the GPIOG is set to output, data is read from the register. If set to input, the pin status can be read.

GPIOG Pin Function Register (GPIOG_FNC)

GPIO[0x34] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPG7MD [1:0]		GPG6MD [1:0]		GPG5MD [1:0]		GPG4MD [1:0]	
15	14	13	12	11	10	9	8
GPG3MD [1:0]		GPG2MD [1:0]		GPG1MD [1:0]		GPG0MD [1:0]	
7	6	5	4	3	2	1	0

Selects a GPIOG pin function. Two bits are used to select a pin function for each GPIOG port.

Table 26.8 Port-G Pin Select Function

GPGxMD1	GPGxMD0	Pin Function (“x” identifies the bit position on the port.)
0	0	GPIOGx port input (Default)
0	1	Function 1 other than GPIO
1	0	GPIOGx port output
1	1	Reserved

26.3.2.8 GPIOH Register

GPIOH Data Register (GPIOH_DATA)

GPIO[0x38] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
GPIOHDATA[7:0]							
7	6	5	4	3	2	1	0

This is the GPIOH data register. Data can be written into and read from the register. If the GPIOH is set to output, data is read from the register. If set to input, the pin status can be read.

GPIOH Pin Function Register (GPIOH_FNC)

GPIO[0x3C] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPH7MD [1:0]		GPH6MD [1:0]		GPH5MD [1:0]		GPH4MD [1:0]	
15	14	13	12	11	10	9	8
GPH3MD [1:0]		GPH2MD [1:0]		GPH1MD [1:0]		GPH0MD [1:0]	
7	6	5	4	3	2	1	0

Selects a GPIOH pin function. Two bits are used to select a pin function for each GPIOH port.

Table 26.9 Port-H Pin Select Function

GPHxMD1	GPHxMD0	Pin Functions
0	0	GPIOHx port input (Default)
0	1	Function 1 other than GPIO
1	0	GPIOHx port output
1	1	Function 2 other than GPIO

26.3.2.9 GPIOI Registers

GPIOI Data Register (GPIOI_DATA)

GPIO[0x40] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
n/a						GPIOIDATA[1:0]	

This is the GPIOI data register. Data can be written into and read from the register. If the GPIOI is set to output, data is read from the register. If set to input, the pin status can be read.

GPIOI Pin Function Register (GPIOI_FNC)

GPIO[0x44] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
n/a				GPI1MD [1:0]		GPI0MD [1:0]	

Selects a GPIOI pin function. Two bits are used to select a pin function for each GPIOI port.

Table 26.10 Port-I Pin Select Function

GPIxMD1	GPIxMD0	Pin Functions
0	0	GPIOIx port input (Default)
0	1	Function 1 other than GPIO (Only GPIOI[1:0] can be set.)
1	0	GPIOIx port output
1	1	Function 2 other than GPIO (Only GPIO[0] can be set.)

26.3.2.10 GPIOJ Register

GPIOJ Data Register (GPIOJ_DATA)

GPIO[0x48] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
GPIOJDATA[7:0]							

This is the GPIOJ data register. Data can be written into and read from the register. If the GPIOJ is set to output, data is read from the register. If set to input, the pin status can be read.

GPIOJ Pin Function Register (GPIOJ_FNC)

GPIO[0x4C] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
GPJ7MD [1:0]		GPJ6MD [1:0]		GPJ5MD [1:0]		GPJ4MD [1:0]	
7	6	5	4	3	2	1	0
GPJ3MD [1:0]		GPJ2MD [1:0]		GPJ1MD [1:0]		GPJ0MD [1:0]	

Selects a GPIOJ pin function. Two bits are used to select a pin function for each GPIOJ port.

Table 26.11 Port-J Pin Select Function

GPJxMD1	GPJxMD0	Pin Functions
0	0	GPIOJ0 port input (Default)
0	1	Function 1 other than GPIO
1	0	GPIOJ0 port output
1	1	Reserved

26.3.2.11 GPIOK Register

GPIOK Data Register (GPIOK_DATA)

GPIO[0x50] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
GPIOKDATA[7:0]							

This is the GPIOK data register. Data can be written into and read from the register. If the GPIOK is set to output, data is read from the register. If set to input, the pin status can be read.

GPIOK Pin Function Register (GPIOK_FNC)

GPIO[0x54] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
GPK7MD [1:0]		GPK6MD [1:0]		GPK5MD [1:0]		GPK4MD [1:0]	
15	14	13	12	11	10	9	8
GPK3MD [1:0]		GPK2MD [1:0]		GPK1MD [1:0]		GPK0MD [1:0]	
7	6	5	4	3	2	1	0

Selects a GPIOK pin function. Two bits are used to select a pin function for each GPIOK port.

Table 26.12 Port-K Pin Select Function

GPKxMD1	GPKxMD0	Pin Functions
0	0	GPIOK0 port input (Default)
0	1	Function 1 other than GPIO
1	0	GPIOK0 port output
1	1	Reserved

GPIOA&B IRQ Registers

IRQ14 interrupt requests are set by GPIO[0x60] to GPIO[0x6C] registers as follows.

GPIOA&B IRQ TYPE

GPIOA&B IRQ Type Register (GPIOAB_ITYP)

GPIO[0x60] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
PORTB_IRQ_TYPE [7:0]							
7	6	5	4	3	2	1	0
PORTA_IRQ_TYPE [7:0]							

Bits [15:8]: **PORTB_IRQ_TYPE**

Bits [7:0]: **PORTA_IRQ_TYPE**

Each bit can be set for level triggering or signal edge interruption as follows.

0: Uses an interrupt request as a level trigger signal.

1: Uses an interrupt as a signal edge for detection.

GPIOA&B IRQ Polarity

GPIOA&B IRQ Polarity Register (GPIOAB_IPOL)

GPIO[0x64] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
PORTB_IRQ_POL [7:0]							
7	6	5	4	3	2	1	0
PORTA_IRQ_POL [7:0]							

Bits [15:8]: **PORTB_IRQ_POL**

Bits [7:0]: **PORTA_IRQ_POL**

0: Considers a logical High level as an interrupt request if the level signal of interrupt request is used. If the signal edge of interrupt request is used, a rising edge of signal is considered as an interrupt request.

1: Considers a logical Low level as an interrupt request if the level signal of interrupt request is used. If the signal edge of interrupt request is used, a falling edge of signal is considered as an interrupt request.

GPIOA&B IRQ ENABLE

GPIOA&B IRQ Enable Register (GPIOAB_IEN)

GPIO[0x68] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
PORTB_IEN [7:0]							
7	6	5	4	3	2	1	0
PORTA_IEN [7:0]							

Bits [15:8]: **PORTB_IEN[7:0]**

These bits correspond to PORTB[7:0] bits for interrupt enable/disable setting.

0: Disables an interrupt from GPIOB.

1: Enables an interrupt from GPIOB.

Bits [7:0]: **PORTA_IEN[7:0]**

These bits correspond to PORTA[7:0] bits for interrupt enable/disable setting.

0: Disables an interrupt from GPIOA.

1: Enables an interrupt from GPIOA.

GPIOA&B IRQ STATUS & Clear

GPIOA&B IRQ Status & Clear Register (GPIOAB_ISTS)

GPIO[0x6C] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
PORTB_IRQ [7:0]							
7	6	5	4	3	2	1	0
PORTA_IRQ [7:0]							

Bits [15:8]: **PORTB_IRQ[7:0]**

These bits correspond to PORTB[7:0] bits for interrupt status display and clearing.

[Read]

0: No interrupt request has been issued.

1: An interrupt request has been issued.

[Write]

0: No status changes.

1: The interrupt request cause is cleared if logical 1 is written.

Bits [7:0]: **PORTA_IRQ[7:0]**

These bits correspond to PORTA[7:0] bits for interrupt status display and clearing.

[Read]

0: No interrupt request has been issued.

1: An interrupt request has been issued.

[Write]

0: No status changes.

1: The interrupt request cause is cleared if logical 1 is written.

Note: The interrupt requests of these bits from GPIOA and GPIOB pins are "logically ORed" and posted to the controller. Therefore, the port where the interrupt request has been generated must be determined within the GPIO IRQ handler using the software.

26.4 GPIOA and GPIOB Interrupt Logic

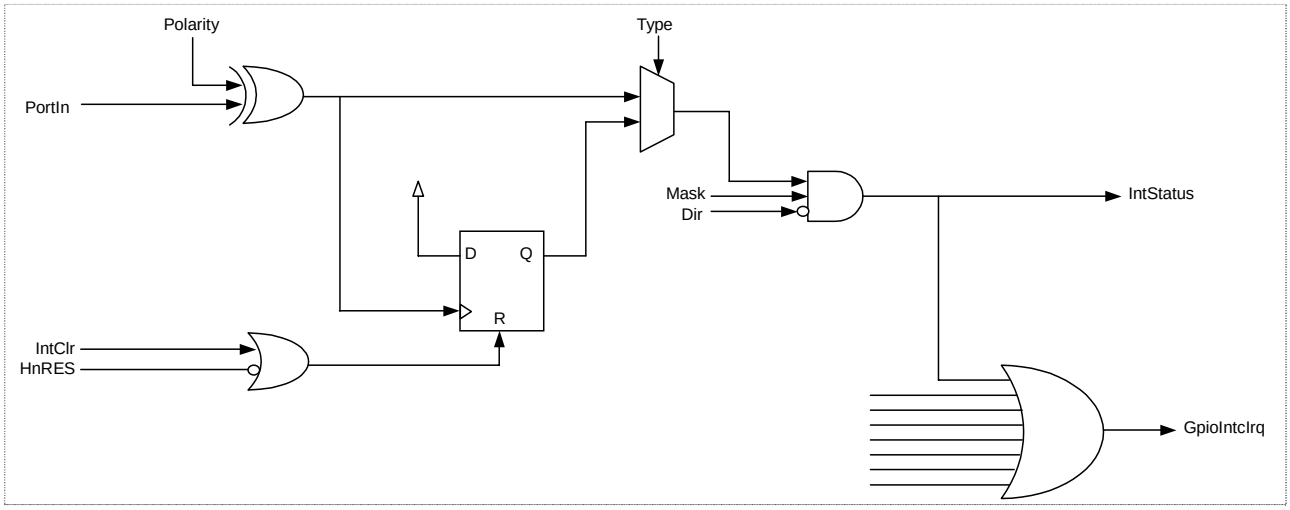


Fig.26.1 GPIOA and GPIOB Interrupt Logic

Note: If the interrupt type or its signal polarity is changed, a false interrupt may be inserted. To avoid this problem, be sure to clear the interrupt cause before use if you have changed the interrupt type or its signal polarity.

27. A/D Controller (ADC)

27.1 Description

When an 8-pin analog signal (ADIN[7:0]) is entered in the circuit, the built-in 10-bit ADC converts the analog signal into digital form, and the digital signal is output. This device also has the A/D converter controller (called the ADC).

Analog input voltage can be from 0 to 3.3 volts, and the signal at 8 channels can be sampled continuously.

The time of single A/D conversion is 20 µsec or less (when system clock sets about 49MHz), and an interrupt can be generated at the completion of signal conversion.

27.2 Block Diagram

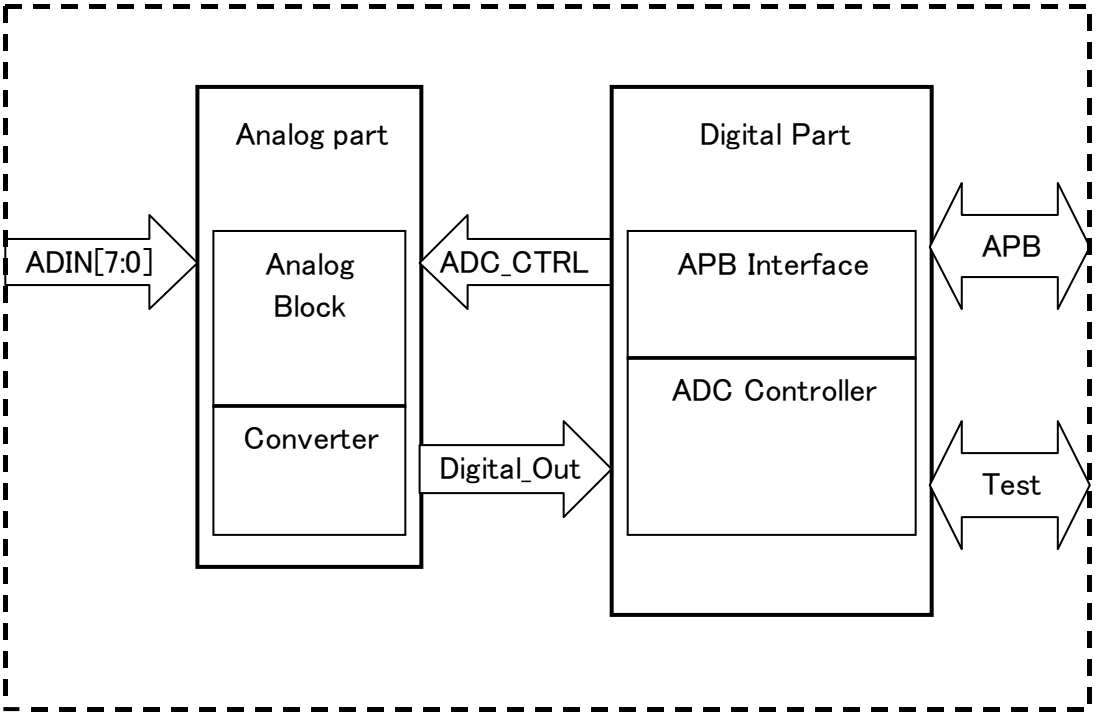


Fig.27.1 ADC Block Diagram

27.3 External Pins

The following defines the external pins that relate to the ADC.

Table 27.2 External Pins (CF)

Pin Name	Input/Output	Pin Functions	Multiplex Pin/Remarks
ADIN[7:0]	Input	Analog signal input and output	

27.4 Operation State

The ADC operates in the following state.

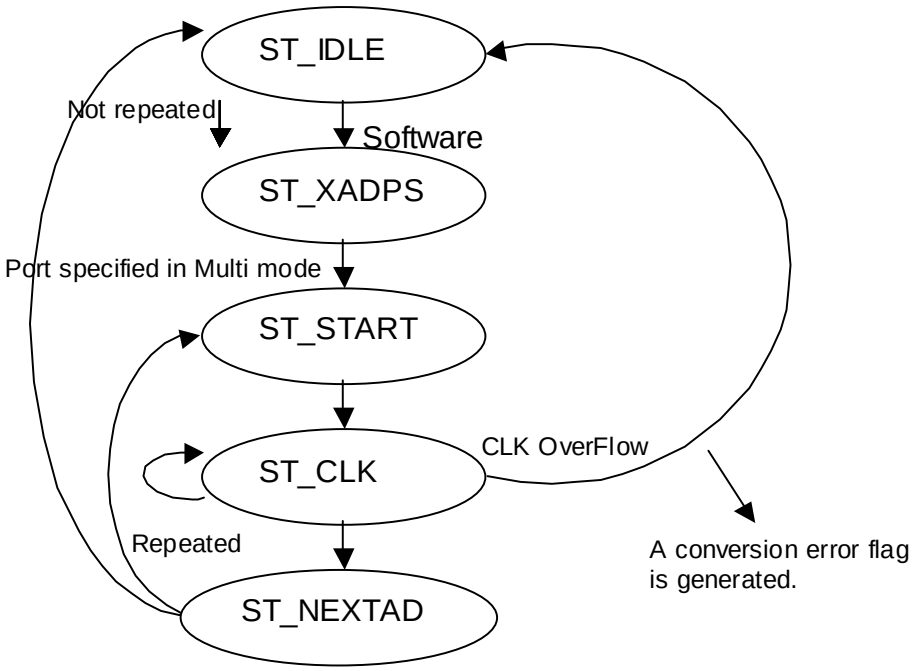


Fig.27.2 ADC Operation State

ST_IDLE	:	This is the initial status. This status is kept until an input by the software occurs.
ST_XADPS	:	Activates the XADPS signal to start the A/D conversion. The A/D power is turned ON in this state.
ST_START	:	Activates the START signal.
ST_CLK	:	The A/D conversion is executed in this state. The A/D conversion takes a certain time. However, if it exceeds this time, this state may return to the ST_IDLS state due to the error. It it has occurred, bit 3 of ADC Flag register is set to logical 1. (During this time, the A/D conversion result is unreliable.) If the Single port is used, the operation state returns to ST_IDLE after the A/D conversion.
ST_NEXTAD	:	If the A/D conversion is specified for multiple ports and if all of A/D conversions are not complete, the operation state returns to ST_START and the A/D conversion is restarted. When the A/D conversion is complete at all of multiple ports, the operation state returns to the ST_IDLE.

27.5 Registers

27.5.1 List of Registers

The ADC registers have base address 0xFFFD_C000.

Table 27.3 List of Registers (Base Address: 0xFFFD_C000)

Address Offset	Register Name	Abbreviation Name	Default Value	R/W	Data Access Size
0x00	ADC Data Register 0	ADCDT0	0x0000	RO	16 (/32)
0x04	ADC Data Register 1	ADCDT1	0x0000	RO	16 (/32)
0x08	ADC Data Register 2	ADCDT2	0x0000	RO	16 (/32)
0x0C	ADC Data Register 3	ADCDT3	0x0000	RO	16 (/32)
0x10	ADC Data Register 4	ADCDT4	0x0000	RO	16 (/32)
0x14	ADC Data Register 5	ADCDT5	0x0000	RO	16 (/32)
0x18	ADC Data Register 6	ADCDT6	0x0000	RO	16 (/32)
0x1C	ADC Data Register 7	ADCDT7	0x0000	RO	16 (/32)
0x20	ADC Control Register	ADCCTL	0x00	R/W	8 (/16/32)
0x24	ADC Flag Register	ADCFLG	0x0000	R/W	16 (/32)

27.5.2 Detailed Description of Registers

ADC Data Register 0 (ADCDT0)

ADC[0x00] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
Data 0 [9:2] RO							

7	6	5	4	3	2	1	0
Data 0 [1:0] RO		Reserved RO					

Bit [15:6]: **Data 0 [9:0]**
Channel 0 10-bit A/D conversion data [9:0]
The conversion result of channel 0 (analog input in ADIN0 pin) is read.

Bits [5:0]: **Reserved**

ADC Data Register 1 (ADCDT1)

ADC[0x04] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
Data 1 [9:2] RO							

7	6	5	4	3	2	1	0
Data 1 [1:0] RO		Reserved RO					

Bit [15:6]: **Data 1 [9:0]**
Channel 1 10-bit A/D conversion data [9:0]
The conversion result of channel 1 (analog input in ADIN1 pin) is read.

Bits [5:0]: **Reserved**

ADC Data Register 2 (ADCDT2)

ADC[0x08] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
Data 2 [9:2] RO							
Data 2 [1:0] RO		Reserved RO					
7	6	5	4	3	2	1	0

Bit [15:6]: **Data 2 [9:0]**

Channel 2 10-bit A/D conversion data [9:0]

The conversion result of channel 2 (analog input in ADIN2 pin) is read.

Bits [5:0]: **Reserved**

ADC Data Register 3 (ADCDT3)

ADC[0x0C] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
Data 3 [9:2] RO							
Data 3 [1:0] RO		Reserved RO					
7	6	5	4	3	2	1	0

Bit [15:6]: **Data 3 [9:0]**

Channel 3 10-bit A/D conversion data [9:0]

The conversion result of channel 3 (analog input in ADIN3 pin) is read.

Bits [5:0]: **Reserved**

ADC Data Register 4 (ADCDT4)

ADC[0x10] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
Data 4 [9:2] RO							
Data 4 [1:0] RO		Reserved RO					
7	6	5	4	3	2	1	0

Bit [15:6]: **Data 4 [9:0]**

Channel 4 10-bit A/D conversion data [9:0]

The conversion result of channel 4 (analog input in ADIN4 pin) is read.

Bits [5:0]: **Reserved**

ADC Data Register 5 (ADCDT5)

ADC[0x14] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
Data 5 [9:2] RO							
Data 5 [1:0] RO		Reserved RO					
7	6	5	4	3	2	1	0

Bit [15:6]: **Data 5 [9:0]**
Channel 5 10-bit A/D conversion data [9:0]
The conversion result of channel 5 (analog input in ADIN5 pin) is read.

Bits [5:0]: **Reserved**

ADC Data Register 6 (ADCDT6)

ADC[0x18] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
Data 6 [9:2] RO							
Data 6 [1:0] RO		Reserved RO					
7	6	5	4	3	2	1	0

Bit [15:6]: **Data 6 [9:0]**
Channel 6 10-bit A/D conversion data [9:0]
The conversion result of channel 6 (analog input in ADIN6 pin) is read.

Bits [5:0]: **Reserved**

ADC Data Register 7 (ADCDT7)

ADC[0x1C] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
Data 7 [9:2] RO							
Data 7 [1:0] RO		Reserved RO					
7	6	5	4	3	2	1	0

Bit [15:6]: **Data 7 [9:0]**
Channel 7 10-bit A/D conversion data [9:0]
The conversion result of channel 7 (analog input in ADIN7 pin) is read.

Bits [5:0]: **Reserved**

ADC Control Register (ADCCTL)							
ADC[0x20] Default value = 0x00							Read/Write
ADCFLG	ADCIEN	ADCSTT	ADCMLT	ADCEXT	ADCCH2	ADCCH1	ADCCH0
RW	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

Bit 7: **A/D End Flag**

Indicates that the A/D conversion has completed.

0 (r) During A/D conversion or stopped conversion
1 (r) A/D conversion has completed.
0 (w) This flag is cleared.
1 (w) Invalid

When the A/D conversion has completed on all of the specified channels, this flag is set to logical 1. When the ADC registers are updated (including reading), this flag is automatically reset to logical 0. Also, when this bit is changed from logical 1 to logical 0, the flag is reset to logical 0.

If you do not use the interrupt control (that is, if you have set bit 6 to logical 0), use the ADCNV signal of [0x24] bit 2 without monitoring this register.

This bit is the register that monitors the end of A/D conversion, and this value does not guarantee the data set. To guarantee the data set, use the ADCNV signal of [0x24] bit 2.

Bit 6: **A/D interrupt enable**

Enables or disables an A/D conversion end interrupt.

0 (r/w): Disables the interrupt.
1 (r/w): Enables the interrupt.

This interrupt enable bit must always be set to logical 1. If you wish to suppress an interrupt, disable an interrupt of the interrupt control module.

Bit 5: **Starts A/D conversion.**

0 (r): Ends the A/D conversion.
1 (r/w): Starts the A/D conversion.

When the A/D conversion has completed on all of the specified channels, this bit automatically returns to logical 0.

The A/D conversion does not end even if this bit is set to logical 0 during A/D conversion.

If the A/D conversion does not end at the specified clock, the 0x24[3] bit may have been set to logical 1. You can check the error status by referring to this bit.

Bit 4: **Multi-mode select**

Selects an A/D conversion mode (Single mode or Multiple mode).

0 (r/w): Single mode
1 (r/w): Multiple mode

In the Single mode, the A/D conversion of only single-channel input occurs. In the Multiple mode, the A/D conversion is continued for the input of all channels. In both modes, you can switch the channel of input using the channel select bits (bits [2:0]).

Bit 3: **External trigger enable**

0 Disables an external trigger.
1 Enables an external trigger.

If this register is enabled, a falling edge of signal sent from Timer B is detected and the A/D conversion is triggered.

When a falling edge of signal from Timer B is detected during A/D conversion, the operation is stopped when the current conversion ends. You must restart the A/D conversion.

The signal sent from Timer B must be 1 µsec or longer.

Bit [2:0]: **Channel select bits [2:0]**

Selects a channel for A/D signal conversion. The options vary depending on the Single or Multiple mode selection.

Bits[2:0]	Functional Description	
CH[2:0]	Single Mode	Multi Mode
000	AN0	AN0
001	AN1	AN0-AN1
010	AN2	AN0AN2
011	AN3	AN0AN3
100	AN4	AN0-AN4
101	AN5	AN0-AN5
110	AN6	AN0-AN6
111	AN7	AN0-AN7

The A/D conversion can be triggered with a signal sent from Timer B if the external trigger enable bit is set to logical 1. Therefore, if you transfer the converted data to another place, we recommend you to detect an interrupt after A/D conversion and to read the data after the interrupt occurrence.

When the converted data is read, the End flag is set to logical 0. If the End flag is logical 1 and if another input is received from Timer B, the End flag is set to logical 1 and the next A/D conversion is started.

If data is transferred during its A/D conversion, the value is unreliable.

ADC Flag Register (ADCFLG)

ADC[0x24] Default value = 0x00 Read/Write

Reserved				ADCERR	ADCCNV	Reserved	
—	—	—	—			—	—
7	6	5	4	3	2	1	0

Bit [7:4]: **Reserved**

Bit 3: **ADCERR**

If an abnormality occurs during A/D conversion, this bit is set to logical 1.

The logical-1 bit means that the A/D conversion is not completed although the A/D conversion clock has been generated 15 times or more. Because the converted data is not written in the data register, the register value before the A/D conversion is read. However, if the A/D conversion is executed normally even if this flag has occurred, the data register value is updated correctly.

The logical-1 bit is set to 0 when the system is reset or when this bit is set to 1.

Bit 2: **ADCCNV**

Indicates that the A/D conversion has completed normally and the converted data has been stored in the data register.

In the Multiple mode, this bit shows that A/D conversion has completed on all of the specified channels and all data has been stored in the data register. To make this bit effective, you must enable the A/D interrupt of ADC[0x20], Bit 6.

The ADCCNV signal that is output from the Analog part is traced. When the A/D conversion has completed and when the Conversion End flag is output from the Analog part and the converted data is stored in the register, this bit is set to logical 1. When the A/D conversion is started again, this bit is returned to logical 0.

Bit [1:0]: **Reserved**

27.6 Application Examples

- Single mode

A single-channel analog input can be converted into digital form in the Single mode.

When bit 5 of ADC Control register (called "ADC[0x20]" hereafter) is set to logical 1 by the software or by the external trigger input, A/D conversion of the specified channel data starts.

The following explains the A/D conversion if only ADIN1 pin is used in the Single mode.

1. Select the Single mode and enable an ADIN1 input and an interrupt (that is, set bits [7:0] of ADC[0x20] to 61h and start the A/D conversion).
2. After the A/D conversion has completed, bit 7 of ADC[0x20] is set to logical 1 but bit 5 of ADC[0x20] is reset to logical 0. The next A/D conversion is waited. Then, the A/D conversion result is set in Data Register 1.
3. If an ADC conversion interrupt is enabled by the interrupt controller, an interrupt is accepted after the A/D conversion. If you read data from Data Register 1 and set it in the interrupt process routine, you can read the A/D conversion result of ADIN1.
4. After execution of interrupt process routine, you can start the next A/D conversion by setting bit 5 of ADC[0x20] to logical 1.

- Multiple mode

The multiple-channel analog inputs (including the input of channel 1) can be converted into digital form sequentially in the Multiple mode.

If you select the Multiple mode and set bits [2:0] of ADC[0x20] to "001", the A/D conversion of ADIN1 starts immediately after A/D conversion of ADIN0. The A/D conversion ends when analog data of all channels has been converted into digital data. The End flag (ADCFLG) is set to logical 1, and it shows that the data conversion has completed.

The converted digital data is held in ADC Data registers of the respective channels.

The following explains the A/D conversion if ADIN0, ADIN1 and ADIN2 pins are used in the Multiple mode.

1. Select the Multiple mode and enable ADIN0, ADIN1 and ADIN2 inputs and an interrupt (that is, set bits [7:0] of ADC[0x20] to 72h and start A/D conversion).
2. The ADIN0 analog data is first converted into digital form. After its conversion, the result is set in Data Register 0. Similarly, ADIN1 and ADIN2 analog data are converted sequentially, and the conversion results are set in the respective data registers. The last ADIN2 data is stored in registers only after the End flag has been set.
3. After the ADIN2 data conversion has completed, bit 7 of ADC[0x20] is set to logical 1 but bit 5 of ADC[0x20] is reset to logical 0. The next A/D conversion is waited. Then, the A/D conversion result is set in Data Register 2.
4. If an ADC conversion end interrupt is enabled by the interrupt controller, an interrupt is accepted after the A/D conversion. If you read data from Data Register 1, 2 or 3 and set it in the interrupt process routine, you can read the A/D conversion result of ADIN1, ADIN2 or ADIN3.
5. After execution of interrupt process routine, you can start the next A/D conversion by setting bit 5 of ADC[0x20] to logical 1.

27.7 Input Voltages and Converted Data Values

The values listed below can be entered in ADC Data Registers 0 to 7 [15:0] using the input voltages.

This ADC can convert 10-bit analog data into digital form but it has an error within ±2 bits. Therefore, bits 15 to 8 of ADC Data Register [7:0] must be checked.

Input Voltage [V]	Conversion Reference Value	ADC Data Register0-7[15:0]		Remarks
		Min.	Max.	
0.0	0	0x00_00	0x00_C0	
0.3	0x5D	0x16_80	0x18_00	
0.6	0xB5	0x2D_C0	0x2F_40	
0.9	0x117	0x45_00	0x46_80	
1.2	0x1174	0x5C_40	0x5D_C0	
1.5	0x1D1	0x73_80	0x75_00	
1.8	0x22E	0x8A_C0	0x8C_40	
2.1	0x28B	0xA2_00	0xA3_80	
2.4	0x2E8	0xB9_40	0xBA_C0	
2.7	0x345	0xD0_80	0xD2_00	
3.0	0x3A2	0xE7_C0	0xE9_40	
3.3	0x3FF	0xFF_00	0xFF_C0	

28. Area Sensor (ARS)

28.1 Description

A screen image of YUV input signals sent from the camera I/F can be divided into 16 square partitions, and any of YUV components can be selected and each partition of signal components can be calculated and set in registers. If this calculation value is compared with the previous calculation value, a partial image movement can be detected in real-time mode without increasing the CPU load.

[An example of 2MB pixel camera (1632*1220 pixels)]
Although the calculated data width is 1632x1224x63/16 = 7865424 (decimal) = 780450 (hex) = 23 bits, the actual width is 21 bits because the calculation is made only once for 4 pixels of data.

Area 0	Area 1	Area 2	Area 3
Area 4	Area 5	Area 6	Area 7
Area 8	Area 9	Area 10	Area 11
Area 12	Area 13	Area 14	Area 15

Fig.28.1 Distribution of Image Partition Areas (Areas 0 to 15)

28.1.1 Functions

The Area Sensor (ARS) provides the following basic functions.

- Divides a picture image into 16 partitions, selects any of Y, U and V components in each partition, and calculates the components.
- Calculates data of 16 partitioned areas automatically according to the data size (H/V Sync).
- Can respond to the quick movement and slow movement because any calculation and comparison cycle is selectable.
- Allows to set a change ratio of calculation value and to generate an interrupt by detecting a change that exceeds the preset change ratio.
- Allows to determine a contrast change in a square partitioned area by reading data of status register.
- Can enable and clear an interrupt in each partitioned area separately.
- Can read the calculated value (21-bit data) and the magnitude relation data when compared with the previous frame in each area.
- Can switch between Camera 1 and Camera 2 using the registers.

28.2 Block Diagram

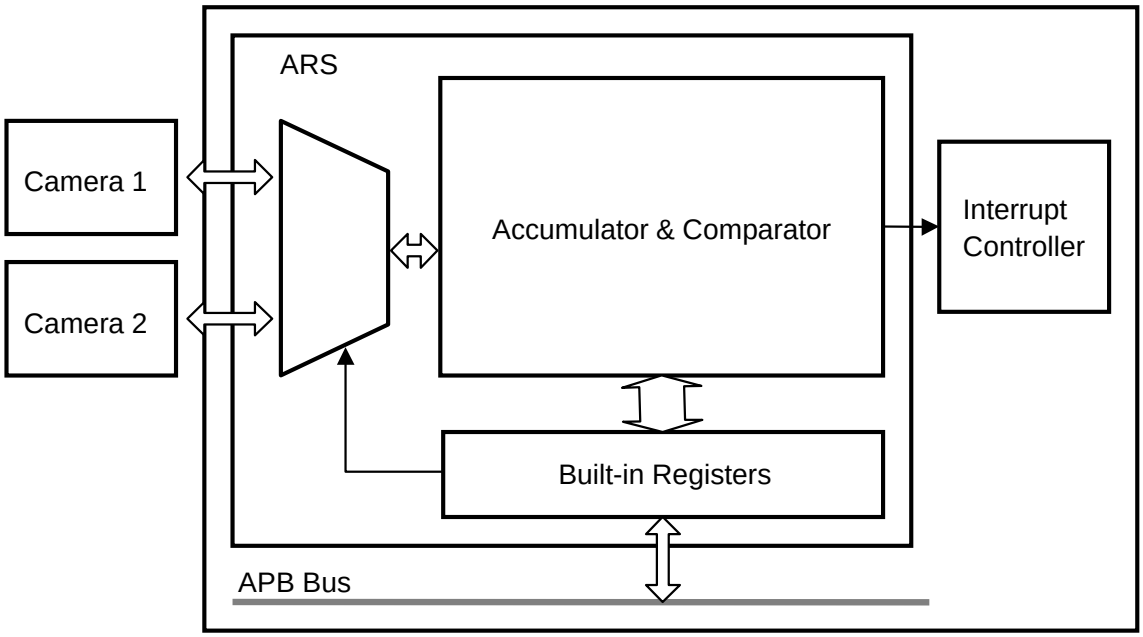


Fig.28.4 ARS Block Diagram

28.3 External Pins

There is no external pin that relates to the ARS. The internal camera input module supplies each camera input and clock.

28.4 Description of Registers

28.4.1 List of Registers

The ARS Control register locates at the default base address of 0xFFFE_7000.
Unless otherwise specified, all register bits are set to logical 0 by default if not reserved.

Table 28.1 ARS Register List

Address Offset	Register Name	Default Value	R/W	Data Access Size
ARS Control register Base address: 0xFFFE_7000				
0x00	ARS Control register (ARSCTRL)	0x0000_0080	R/W	32
0x04	ARS Area Select register (ARSASEL)	0x0000_0000	R/W	32
0x0C	ARS Status register (ARSSTAT)	0x0000_0000	R/W	32
0x10	ARS Interrupt Cause register (ARSINT)	0x0000_0000	R/W	32
0x40	ARS Accumulation register 0 (ARSADD0)	0x0000_0000	RO	32
0x44	ARS Accumulation register 1 (ARSADD1)	0x0000_0000	RO	32
0x48	ARS Accumulation register 2 (ARSADD2)	0x0000_0000	RO	32
0x4C	ARS Accumulation register 3 (ARSADD3)	0x0000_0000	RO	32
0x50	ARS Accumulation register 4 (ARSADD4)	0x0000_0000	RO	32
0x54	ARS Accumulation register 5 (ARSADD5)	0x0000_0000	RO	32
0x58	ARS Accumulation register 6 (ARSADD6)	0x0000_0000	RO	32
0x5C	ARS Accumulation register 7 (ARSADD7)	0x0000_0000	RO	32
0x60	ARS Accumulation register 8 (ARSADD8)	0x0000_0000	RO	32
0x64	ARS Accumulation register 9 (ARSADD9)	0x0000_0000	RO	32
0x68	ARS Accumulation register 10 (ARSADD10)	0x0000_0000	RO	32
0x6C	ARS Accumulation register 11 (ARSADD11)	0x0000_0000	RO	32
0x70	ARS Accumulation register 12 (ARSADD12)	0x0000_0000	RO	32
0x74	ARS Accumulation register 13 (ARSADD13)	0x0000_0000	RO	32
0x78	ARS Accumulation register 14 (ARSADD14)	0x0000_0000	RO	32
0x7C	ARS Accumulation register 15 (ARSADD15)	0x0000_0000	RO	32

28.4.2 Detailed Description of Registers

ARS Control Register

ARS[0x00] Default value = 0x0000_0080 Read/Write

SWRST (WO)	Reserved		ADDCYC (R/W)				
15	14	13	12	11	10	9	8
ADDOFF (RO)	CHGRATE (R/W)			CAMSEL (R/W)	YUVSEL (R/W)		ARSEN (R/W)
7	6	5	4	3	2	1	0

Bit 15: **SWRST Software Reset**

Allow to reset the area sensor module.

0: Nothing occurs.

1: Executes the reset by software.

Bits [14:13]: **(Reserved)**

Bits [12:8]: **ADDCYC Accumulation cycle set**

Sets a cycle for accumulation and comparison. The value set in this register is used to accumulate the YUV components of picture images, which are sent through camera interface, and to compare them with the previous value.

00000 : Does not accumulate. (Default)

00001: Accumulates for each frame.

00010: Accumulates for every two frames.

:

11111: Accumulates for every 31 frames.

Bit 7: **ADDOFF Accumulation On/Off status display (Read Only)**

Shows the accumulation On/Off status. Refer to this register and make sure that the accumulation status is Off before reading the accumulated values and status register values.

0: Accumulation On (Default)
1: Accumulation Off

Bits [6:4]: **CHGRATE Change rate set**

If the accumulated value changes greater than the threshold, each status bit of the Status register (ARS[0x08]) is set to logical 1 and an interrupt is generated.

000: Approx. 1.5% (Default)
001: Approx. 3%
010: Approx. 6%
011: Approx. 13%
100: Approx. 25%
101: Approx. 50%
11x: (Reserved)

Bit 3 **CAMSEL Camera select**

Selects a camera input for accumulation.

0: Camera 0 (Default)
1: Camera 1

Bits [2:1] **YUVSEL Accumulation element select**

Select the Y/U/V value to be accumulated.

00: Does not accumulate. (Default)
01: Accumulates Y values.
10: Accumulates U values.
11: Accumulates V values.

Bit 0: **ARSEN ARS enable**

Turns the ARS accumulation function On or Off. If this bit is set to logical 1, the accumulation starts at the next frame. Also, if this bit is set to logical 0, the ARS accumulation function is stopped after the current frame has ended.

0: Disables the ARS.
1: Enables the ARS.

ARS Area Select Register

ARS[0x04] Default value = 0x0000_0000 Read/Write

ASEL[15:8]							
15	14	13	12	11	10	9	8
ASEL[7:0]							
7	6	5	4	3	2	1	0

Bit[15:0]: **ASEL Area Select register**

Selects a change detection area or areas. (Multiple areas can be selected.) For the relationship between divided areas, see Fig.28.1 “Distribution of Split Areas.”

ARS Status Register

ARS[0x0C] Default value = 0x0000_0000 Read/Write

ARSSTAT [15:8]							
15	14	13	12	11	10	9	8
ARSSTAT[7:0]							
7	6	5	4	3	2	1	0

Bit[15:0]: **ARSSTAT Area status register**

Indicates a change exceeding the change ratio being set for each of divided areas.

0: No change occurs in the area.

1: A change occurred in the area.

This indication is cleared if each bit is set to logical 1.

ARS Interrupt Status Register

ARS[0x10] Default value = 0x0000_0000 Read/Write

ARSINT [15:8]							
15	14	13	12	11	10	9	8
ARSINT[7:0]							
7	6	5	4	3	2	1	0

Bits [15:0]: **ARSINT[15:0] ARS Interrupt Cause register**

Indicates that an interrupt cause exists or not in each of divided areas. The interrupt cause is cleared if each bit is set to logical 1.

ARS Accumulation Registers 0 to 15

ARS[0x40] to ARS[0x7C] Default value = 0000_0000 Read Only

GTO	(Reserved)										ADDV[20:16]				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDV[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit [31]: **GTO Accumulation Value Increment/Decrement register**

Indicates an increase or decrease of the current accumulation value when compared with the previous one.

0: Decreased

1: Increased

Bits [20:0]: **ADDV [20:0] Area accumulation value register**

Indicates the accumulated value of each area.

29. Absolute Maximum Ratings

29.1 Absolute Maximum Ratings

(VSS = 0 [V])

Item	Symbol	Rated Value	Unit
Power voltage	HVDD, UVDD3, C1VDD, C2VDD, SDVDD, AVDD	-0.3 to 4.0	V
	LVDD, UPVDD, UXVDD, RTCVDD, PLLVDD	-0.3 to 2.5	V
Input voltage	HVI	-0.3 to HVDD+0.5	V
	LVI	-0.3 to LVDD+0.5	V
	USBVDD	-0.3 to 6.0	V
Output voltage	HVO	-0.3 to HVDD+0.5	V
	LVO	-0.3 to LVDD+0.5	V
Output current/pin	I_{OUT}	± 10	mA
Storage temperature	T_{stg}	-65 to 150	°C

29.2 Recommended Operating Conditions (Dual Power Supplies, 3.3V I/O Buffers)

(UVSS = PVSS = XVSS = 0 [V]
VSS = PLLVSS = AVSS = 0 [V])

Item		Symbol	Min.	Typ.	Max.	Unit
Power voltage (High-voltage)	I/O cell power supply	HVDD	3.00	3.30	3.60	V
	USB power supply	UVDD3	3.00	3.30	3.60	V
	Camera-1 I/F power supply	C1VDD	2.40	3.00	3.60	V
	Camera-2 I/F power supply	C2VDD	2.40	3.00	3.60	V
	SDRAM I/F power supply	SDVDD	2.70	3.00	3.60	V
	A/D converter power supply	AVDD	3.00	3.30	3.60	V
Power voltage (Low-voltage)	(Internal) core power supply	LVDD	1.65	1.80	1.95	V
	USB power supply	UPVDD	1.65	1.80	1.95	V
	USB power supply	UXVDD	1.65	1.80	1.95	V
	Analog (PLL) power supply	PLLVDD	1.65	1.80	1.95	V
	RTC power supply	RTCVDD	1.65	1.80	1.95	V
Input voltage	I/O cell power supply	HV_I	VSS	—	HVDD	V
	USB power supply	$UV3_I$	UVSS	—	UVDD3	V
	Camera-1 I/F power supply	$C1V_I$	VSS	—	C1VDD	V
	Camera-2 I/F power supply	$C2V_I$	VSS	—	C2VDD	V
	SDRAM I/F power supply	SDV_I	VSS	—	SDVDD	V
	A/D converter power supply	AV_I	AVSS	—	AVDD	V
	(Internal) core power supply	LV_I	VSS	—	LVDD	V
	USB power supply	UPV_I	PVSS	—	UPVDD	V
	USB power supply	UXV_I	XVSS	—	UXVDD	V
	Analog (PLL) power supply	$PLLV_I$	PLLVSS	—	PLLVDD	V
	RTC power supply	$RTCV_I$	VSS		RTCVDD	V
Ambient temperature		Ta	-40	25	85*	°C
Input rise time (normal input)		t_{ri}	—	—	50	ns
Input fall time (normal input)		t_{fa}	—	—	50	ns
Input rise time (Schmitt input)		t_{ri}	—	—	5	ms
Input fall time (Schmitt input)		t_{fa}	—	—	5	ms

*: This temperature range is the recommended ambient temperature range if Tj=-40 to 125°C.

29.3 Power-ON Timing

The 3.3V power supply (HVDD) and 1.8V power supply (LVDD) must be turned On in the following sequence.

(1) First, turn the 3.3Vdc power supply On. Within 1 msec, turn the 1.8Vdc power supply On. We recommend you to reduce this delay as much as possible.
(2) After the HVDD and LVDD signals have become stable, keep the RESET# signal in logical Low more than the 32KHz oscillation start time.

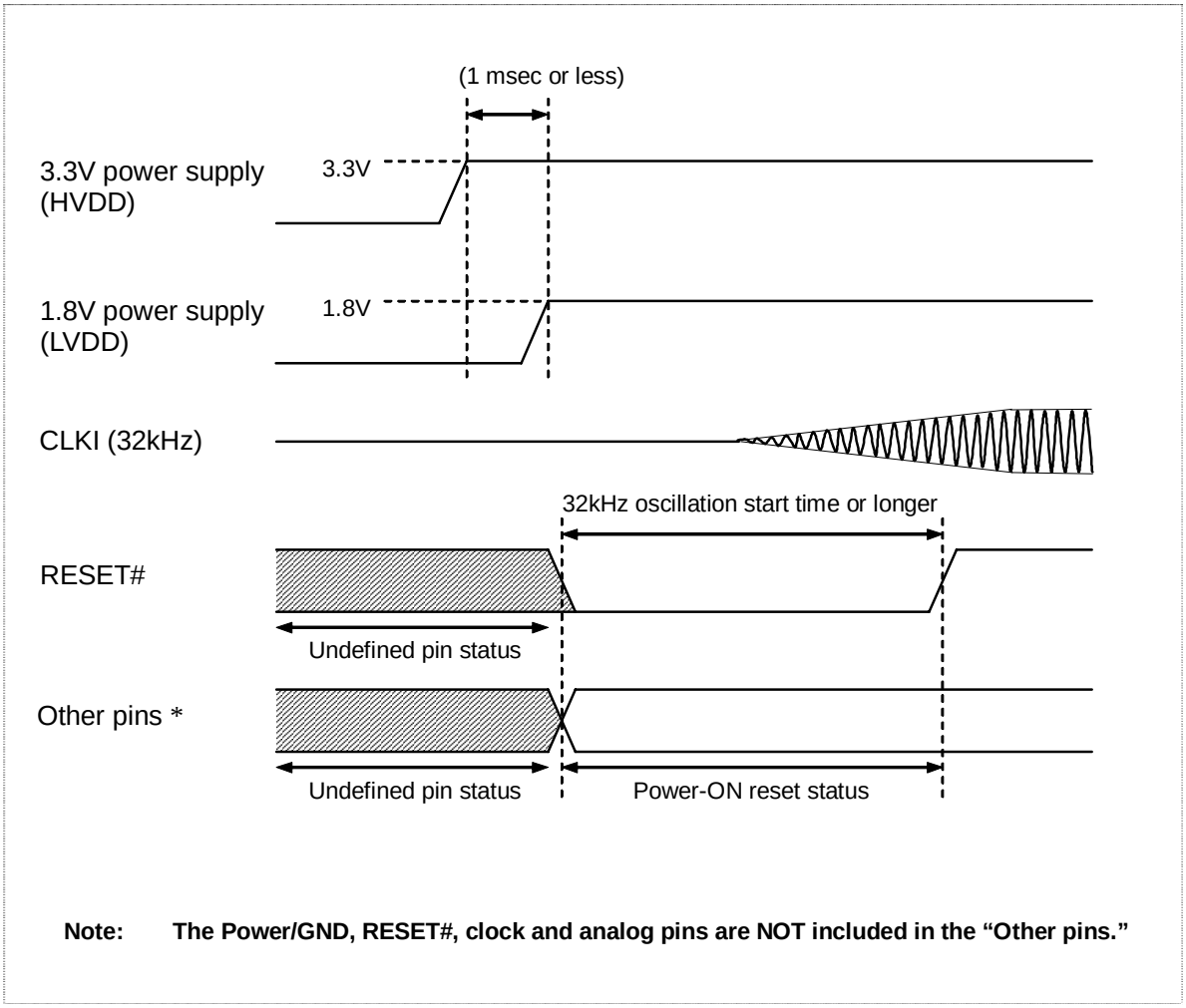


Fig.29.1 Power-ON Timing

29.4 Power-OFF Timing

The 3.3V power supply (HVDD) and 1.8V power supply (LVDD) must be turned Off in the following sequence.

(1) First, turn the 1.8Vdc power supply Off. Within 1 msec, turn the 3.3Vdc power supply Off. We recommend you to reduce this delay as much as possible.
(2) If the 1.8V power supply is only turned Off, the pin status is unstable. You must design the system to avoid the system malfunction due to this unstable pin status.

30. Electrical Characteristics

30.1 DC Characteristics

Table 30.1 DC Characteristics (3.3V)

(HVDD = 3.3V ± 0.3V, VSS = 0V, Ta= -40～85℃)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input leak current	I_{LI}	—		-5	—	5	μA
Off-state leak current	I_{OZ}	—		-5	—	5	μA
High-level output voltage (*1)	V_{OH}	I_{OH} = -4mA HVDD=Min.		HVDD -0.4	—	—	V
Low-level output voltage (*1)	V_{OL}	I_{OL} = 4mA HVDD=Min.		—	—	0.4	V
High-level input voltage	V_{IH1}	LVCMOS level, HVDD=Max.		2.2	—	—	V
Low-level input voltage	V_{IL1}	LVCMOS level, HVDD=Min.		—	—	0.8	V
High-level input voltage	V_{T1+}	LVCMOS Schmitt input		1.4	—	2.7	V
Low-level input voltage	V_{T1-}	LVCMOS Schmitt input		0.6	—	1.8	V
Hysteresis voltage	V_{H1}	LVCMOS Schmitt input		0.3	—	—	V
High-level input voltage	V_{IH2}	LVTTL level, HVDD=Max.		2.0	—	—	V
Low-level input voltage	V_{IL2}	LVTTL level, HVDD=Min.		—	—	0.8	V
Pull-up resistance	P_{PU}	VI=0V		25	50	120	kΩ
Pull-down resistance	P_{PD}	VI=HVDD	Other pins *2	25	50	120	kΩ
			MD[15:0] pin	50	100	240	
Input pin capacity	C_I	f=1MHz, HVDD = 0V		—	—	8	pF
Output pin capacity	C_O	f=1MHz, HVDD = 0V		—	—	8	pF
Input and output pin capacity	C_{IO}	f=1MHz, HVDD = 0V		—	—	8	pF
Current consumption (LVDD)	I_{LOW}	Low-speed mode (32KHz)		—	220	—	μA
	I_{FO}	High-speed mode *3		—	150	—	mA
Current consumption (RTCVDD)	I_{RLOW}	Low-speed mode (32KHz)	BUP# = HIGH	—	170	—	μA
	I_{RLOWS}		BUP# = LOW	—	2	—	mA
	I_{RQ}	High-speed mode *3		—	735	—	μA
Current consumption (PLLVDD)	I_{DDPLL}	If PLL frequency is 50 MHz		—	1	—	mA

*1: Applied to all output and I/O pins.
*2: The pin has pull-down resistance except for MD[15:0] pin.
*3: During VGA image transfer at 30 fps. (each of 2 cameras.)

Table 30.2 DC Characteristics (1.8V)

(RTCVDD = 1.8V ± 0.15V, VSS = 0V, Ta= -40～85℃)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input leak current	I_{LI}	—	-5	—	5	μA
Off-state leak current	I_{OZ}	—	-5	—	5	μA
High-level input voltage	V_{T1+}	LVCMOS Schmitt input	0.6	—	1.4	V
Low-level input voltage	V_{T1-}	LVCMOS Schmitt input	0.3	—	1.1	V
Hysteresis voltage	V_{H1}	LVCMOS Schmitt input	0.02	—	—	V
Input pin capacity	C_I	f=1MHz, HVDD = 0V	—	—	8	pF

Table 30.3 USBVBUS Judgment voltage

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
High-level trigger voltage	VB_{TH}	UVDD3 = 3.6V	1.86	—	2.85	V
Low-level trigger voltage	VB_{TL}	UVDD3 = 3.0V	1.48	—	2.23	V
Hysteresis voltage	VB_H	UVDD3 = 3.0V	0.31	—	0.64	V

30.2 AC Characteristics

30.2.1 AC Characteristics Measuring Conditions

HVDD, AVDD, UVDD3, = 3.3V ± 0.3V
C1VDD, C2VDD = 3.0V ± 0.6V
SDVDD = 2.7V to 3.6V
LVDD, UPVDD, UXVDD, PLLVDD, RTCVDD = 1.8V ± 0.15V
T_A = -40°C to 85°C
CL=50 pF (unless otherwise noted)

30.2.2 AC Characteristics Signal Timing List

30.2.2.1 Clock Timing

Table 30.2 Clock (CLKI) Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
SYSCLKI frequency	f_{clkI}	—	32.768	—	KHz	—
SYSCLKI input cycle time	t_{clkI}	—	1/ f_{OSC}	—	s	—
SYSCLKI high-level pulse width	t_{CLKIH}	5	—	—	µs	—
SYSCLKI low-level pulse width	t_{CLKIL}	5	—	—	µs	—
SYSCLKI rise time (from 10% to 90%)	t_{CLKIR}	—	—	12	µs	—
SYSCLKI fall time (from 90% to 10%)	t_{CLKIF}	—	—	12	µs	—
System clock frequency	fsys	—	—	50	MHz	—
System clock frequency cycle	Ts	1/fsys —	—	—	ns	—

30.2.2.2 CPU Control Signal Timing

Table 30.3 CPU Control Signal Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
RESET# pulse width	t_{RESW}	10	—	—	T_{CLKI}	*1
IRQ/FIQ pulse width	t_{IRQW}	10	—	—	T_S	*2
Clock restart time	t_{WAK}	—	—	4	T_S	*2
PLL stability time	t_{PLLST}	—	—	100	ms	—

*1: T_{CLKI}=32 KHz unit. A signal amplitude to turn off the threshold voltage is required.
*2: T_S=System clock cycle time

30.2.2.3 Battery Backup Mode Timing

Table 30.4 Battery Backup Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
BUP# pin delay time at power supply start time	t_{BUPPO}	0	—	—	ns	
Power shutdown time at backup start time	t_{BUPS}	10	—	—	ns	
Power supply stability time at backup recovery time	t_{BUPH}	10	—	—	T_{CLKI}	*1
RESET# effective time after backup release	t_{BUPRST}	10	—	—	T_{CLKI}	*1

*1: T_{CLKI}=32 KHz unit. A signal amplitude to turn off the threshold voltage is required.

30.2.2.4 Camera Interface (CAM) Timing

Table 30.5 Camera Interface (CAM) Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
From CMVREF rise edge to CMHREF rise edge	t_{CAM1}	0	—	—	T_C	*3
Horizontal blank period	t_{CAM2}	4	—	—	T_C	*3
From CMHREF fall edge to CMVREF fall edge	t_{CAM3}	0	—	—	T_C	*3
Vertical blank period	t_{CAM4}	1	—	—	Line	—
Camera input clock period	t_{CAM5}	1.6 (3.2)	—	—	T_S	*2
Camera input clock low-level pulse width	t_{CAM6}	0.8 (1.6)	—	—	T_S	*2
Camera input clock high-level pulse width	t_{CAM7}	0.8 (1.6)	—	—	T_S	*2
Data setup time	t_{CAM8}	10	—	—	ns	—
Data hold time	t_{CAM9}	10	—	—	ns	—
CMVREF and CMHREF setup time	t_{CAM10}	10	—	—	ns	—
CMVREF and CMHREF hold time	t_{CAM11}	10	—	—	ns	—

*2: T_S = System clock cycle time
The minimum value is obtained during high-speed sampling, and the minimum value during normal sampling is shown in parentheses.
*3: T_C = Camera interface input clock cycle time

30.2.2.5 Memory Controller (MEMC) Timing

■ Static Memory Controller Timing

Table 30.6 Static Memory Timing 1 (MCS0#/MCS1#)

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
Address signal setup time	t_{MAS}	– 3.5	—	– 1	ns	—
Address signal hold time (to MOE#)	t_{MAHOE}	– 0.5	—	0	ns	—
Address signal hold time (to MWE#)	t_{MAHWE}	1Ts – 1	—	—	ns	*2
Address signal hold time (to MCSx#)	t_{MAHCE}	1Ts – 1	—	—	ns	*2
Data output enable signal delay time	t_{MOED}	n_1Ts – 0.5	—	—	ns	*2 *4
Data output enable signal effective period	t_{MOEV}	n_2Ts – 0.5	—	—	ns	*2 *5
Data output enable signal interval period	t_{MOEI}	—	n_1	—	Ts	*2 *4
Chip select hold time (to MOE#)	t_{MCEHOE}	– 0.5	—	0.5	ns	—
Read data setup time	t_{MDRS}	—	—	17	ns	—
Read data hold time	t_{MDRH}	0	—	—	ns	—
Write effective signal delay time	t_{MWED}	n_3Ts – 0.5	—	—	ns	*2 *6
Write effective signal effective period	t_{MWEV}	n_4Ts +1 – n_3Ts – 0.5	—	—	ns	*2 *6 *7
Write effective signal interval period	t_{MWEI}	—	n_3	—	Ts	*2 *6
Chip select signal hold time (to MWE#)	t_{MCEHWE}	1Ts – 1	—	1Ts	ns	*2
Byte enable signal delay time	t_{MBED}	—	—	1	ns	—
Byte enable signal hold time	t_{MBEH}	1Ts – 2	—	—	ns	*2
Write data setup time	t_{MDWS}	—	—	n_4Ts – 6	ns	*2 *7
Write data hold time	t_{MDWH}	1Ts – 0.5	—	—	ns	*2

Table 30.7 Static Memory Timing 2 (MCS2#/MCS3#)

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
Address signal setup time	t_{MAS}	– 1	—	1	ns	—
Address signal hold time (to MOE#)	t_{MAHOE}	– 0.5	—	0	ns	—
Address signal hold time (to MWE#)	t_{MAHWE}	1Ts – 1	—	—	ns	*2
Address signal hold time (to MCSx#)	t_{MAHCE}	1Ts – 1	—	—	ns	*2
Data output enable signal delay time	t_{MOED}	n_1Ts – 3.5	—	—	ns	*2 *4
Data output enable signal effective period	t_{MOEV}	n_2Ts – 0.5	—	—	ns	*2 *5
Data output enable signal interval period	t_{MOEI}	—	n_1	—	Ts	*2 *4
Chip select hold time (to MOE#)	t_{MCEHOE}	0	—	3	ns	—
Read data setup time	t_{MDRS}	—	—	18	ns	—
Read data hold time	t_{MDRH}	0	—	—	ns	—
Write effective signal delay time	t_{MWED}	n_3Ts – 3.5	—	—	ns	*2 *6
Write effective signal effective period	t_{MWEV}	n_4Ts + 1 – n_3Ts –0.5	—	—	ns	*2 *6 *7
Write effective signal interval period	t_{MWEI}	—	n_3	—	Ts	*2 *6
Chip select signal hold time (to MWE#)	t_{MCEHWE}	1Ts	—	1Ts + 2	ns	*2
Byte enable signal delay time	t_{MBED}	—	—	1	ns	—
Byte enable signal hold time	t_{MBEH}	1Ts – 2	—	—	ns	*2
Write data setup time	t_{MDWS}	—	—	n_4Ts – 6	ns	*2 *7
Write data hold time	t_{MDWH}	1Ts – 0.5	—	—	ns	*2

*2 T_S = System clock cycle time
*4 n_1 = The value set by WAITOE register (MEMC:0x20, 0x30, 0x40, 0x50)
*5 n_2 = The value set by WAITRD register (MEMC:0x20, 0x30, 0x40, 0x50)
*6 n_3 = The value set by WAITWE register (MEMC:0x20, 0x30, 0x40, 0x50)
*7 n_4 = The value set by WAITWR register (MEMC:0x20, 0x30, 0x40, 0x50)

■ SDRAM Controller Timing

Table 30.8 SDRAM Controller Timing (SDVDD=3.3V±0.3V)

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
SDCKE delay time	t_{CKED}	2	—	8	ns	—
SDCS[1:0]# delay time	t_{CSD}	2	—	10	ns	—
SDRAS# delay time	t_{RASD}	2	—	8	ns	—
SDCAS# delay time	t_{CASD}	2	—	10	ns	—
SDWE# delay time	t_{WED}	2	—	8	ns	—
SDDQM[3:0] delay time	t_{DQMD}	1	—	10	ns	—
Address delay time	t_{ADD}	1	—	10	ns	—
Write data delay time	t_{WDD}	1	—	8	ns	—
Read data setup time	t_{RDS}	12	—	—	ns	—
Read data hold time	t_{RDH}	0	—	—	ns	—

Table 30.9 SDRAM Controller Timing (SDVDD=3.0V±0.3V)

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
SDCKE delay time	t_{CKED}	(TBD)	—	(TBD)	ns	—
SDCS[1:0]# delay time	t_{CSD}	(TBD)	—	(TBD)	ns	—
SDRAS# delay time	t_{RASD}	(TBD)	—	(TBD)	ns	—
SDCAS# delay time	t_{CASD}	(TBD)	—	(TBD)	ns	—
SDWE# delay time	t_{WED}	(TBD)	—	(TBD)	ns	—
SDDQM[3:0] delay time	t_{DQMD}	(TBD)	—	(TBD)	ns	—
Address delay time	t_{ADD}	(TBD)	—	(TBD)	ns	—
Write data delay time	t_{WDD}	(TBD)	—	(TBD)	ns	—
Read data setup time	t_{RDS}	(TBD)	—	—	ns	—
Read data hold time	t_{RDH}	(TBD)	—	—	ns	—

30.2.2.6 I²C Single Master Core Module (I2C) Timing

Table 30.10 I²C Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
SCL cycle time	$t_{C(SCL)}$	8	—	30720	T_S	*2
SCL pulse width (High)	$t_{WH(SCL)}$	—	1/2	—	$T_{C(SCL)}$	*8
SCL pulse width (Low)	$t_{WL(SCL)}$	—	1/2	—	$T_{C(SCL)}$	*8
SDA output delay time	$t_{D(OSDA)}$	—	1/4	—	$T_{C(SCL)}$	*8
SDA input setup time	$t_{SU(ISDA)}$	0	—	—	ns	*9
SDA input hold time	$t_{HD(ISDA)}$	0	—	—	ns	*9
SDA sample time	$t_{SMP(SDA)}$	—	1/4	—	$T_{C(SCL)}$	*8
Start condition startup time	$t_{S(ST)}$	1/4	—	—	$T_{C(SCL)}$	*8
Start condition completion time	$t_{E(ST)}$	1/2	—	—	$T_{C(SCL)}$	*8
Stop condition startup time	$t_{S(SP)}$	1/4	—	—	$T_{C(SCL)}$	*8
Stop condition completion time	$t_{E(SP)}$	1/2	—	—	$T_{C(SCL)}$	*8

*2: T_S = System clock cycle time
*8: $T_{C(SCL)}$ = SCL (I²C clock) cycle time
*9: See SDA sample time ($T_{smp(SDA)}$).

30.2.2.7 I²STiming

Table 30.11 I²S Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
SCK cycle time	t_{SCKCT}	2	—	512	T_S	*2
SCK pulse width (High)	t_{SCKWH}	1	—	—	Ts	*2
SCK pulse width (Low)	t_{SCKWL}	1	—	—	Ts	*2
SCK duty ratio	t_{SCKDT}	—	50	—	%	*10
WS cycle time	t_{WSCT}	32	—	256	t_{SCKCT}	*11
WS output delay time	t_{WSOD}	—1	—	1	Ts	*2
WS input setup time	t_{WSISU}	1	—	—	Ts	*2
WS input hold time	t_{WSIHD}	1	—	—	Ts	*2
SD output delay time	t_{SDOD}	—1	—	1	Ts	*2
SD input setup time	t_{SDISU}	1	—	—	Ts	*2
SD input hold time	t_{SDIHD}	1	—	—	Ts	*2

*2: TS = System clock cycle time
*10: $t_{SCKDT} = t_{SCKWH} / (t_{SCKWH} + t_{SCKWL})$
*11: t_{SCKCT} = SCK cycle time

30.2.2.8 Serial Peripheral Device Interface (SPI) Timing

Table 30.12 SPI Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
SCLK cycle time	$t_{C(SCLK)}$	4	—	512	T_S	*2
SCLK pulse width (first half)	$t_{WH1(SCLK)}$	—	1/2	—	$T_{C(SCLK)}$	*12
SCLK pulse width (second half)	$t_{WH2(SCLK)}$	—	1/2	—	$T_{C(SCLK)}$	*12
SS output start time (during auto control)	$t_{S(OSS)}$	3	—	—	T_S	*2
SS output completion time (during auto control)	$t_{E(OSS)}$	1	—	—	T_S	*2
SS input setup time	$t_{SU(ISS)}$	3	—	—	T_S	*2
SS input hold time	$t_{HD(ISS)}$	1	—	—	T_S	*2
MISO input setup time	$t_{SU(MI)}$	30	—	—	ns	
MISO input hold time	$t_{HD(MI)}$	0	—	—	ns	
MISO output delay time	$t_{D(SO)}$	—	—	30	ns	
MOSI input setup time	$t_{SU(SI)}$	10	—	—	ns	
MOSI input hold time	$t_{HD(SI)}$	10	—	—	ns	
SCLK cycle time	$t_{D(MO)}$	—	—	0	ns	

*2: T_S = System clock cycle time
*12: $T_{C(SCLK)}$ = SCLK (SPI clock) cycle time = (4×2^{MCBR}) T_S

30.2.2.9 Compact Flash Memory Interface (CF) Timing

■ CF Attribute Memory Timing

Table 30.13 CF Attribute Memory Read Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
Read cycle time	t_{ATRC}	—	20	—	T_S	*2
Address setup time	t_{ADSAR}	—	4	—	T_S	*2
Address hold time (from MOE# inactive state)	t_{ADHMOE}	—	2	—	T_S	*2
CE effective time before read	t_{CEVBR}	—	3	—	T_S	*2
CE effective time after read	t_{CEVAR}	—	2	—	T_S	*2
MOE# active time	t_{MOEW}	—	14	—	T_S	*2
Read data setup time	t_{RDS}	$1T_S$+16	—	—	ns	
Read data hold time	t_{RDH}	0	—	—	ns	

*2: T_S = System clock cycle time

Table 30.14 CF Attribute Memory Write Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
Write cycle time	t_{ATWC}	—	16	—	T_S	*2
Address setup time	t_{ADSAW}	—	3	—	T_S	*2
MWE0# active time	t_{MWE0W}	—	9	—	T_S	*2
Write recovery time	t_{WREC}	—	2	—	T_S	*2
Write data effective time 1	t_{WDV1}	—	11	—	T_S	*2
Write data effective time 2	t_{WDV2}	—	2	—	T_S	*2

■ CF Common Memory Timing

Table 30.15 CF Common Memory Read Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
Read cycle time	t_{CMRC}	—	17	—	T_S	*2
Address setup time	t_{CRADS}	—	4	—	T_S	*2
Address hold time (from MOE# inactive state)	t_{ADHMOE}	—	2	—	T_S	*2
CE effective time before read	t_{CEVBR}	—	3	—	T_S	*2
CE effective time after read	t_{CEVAR}	—	2	—	T_S	*2
Wait active allowable time after read	t_{WTATAR}	—	—	6	T_S	*2
Data setup time after wait release	t_{DSAWT}	—	—	0	T_S	*2
Wait active time	t_{WTW}			3000	ns	
Read data setup time	t_{RDS}	$1T_S$+16	—	—	ns	
Read data hold time	t_{RDH}	0	—	—	ns	

Table 30.16 CF Common Memory Write Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
Write cycle time	t_{CMWC}	—	17	—	T_S	*2
Address setup time	t_{ADS}	—	4	—	T_S	*2
Address hold time	t_{ADH}	—	4	—	T_S	*2
CE effective time before write	t_{CEVBW}	—	3	—	T_S	*2
CE effective time after write	t_{CEVAW}	—	2	—	T_S	*2
MWE0# active time	t_{MWE0W}	—	9	—	T_S	*2
Data effective time before write	t_{DVBW}	—	11	—	T_S	*2
Data effective time after write	t_{DVAW}	—	2	—	T_S	*2
Write recovery time	t_{WREC}	—	2	—	T_S	*2
Wait active allowable time after write	t_{WTATAW}	—	—	6	T_S	*2
Write active time after wait release	t_{WWAWT}	—	—	3	T_S	*2
Wait active time	t_{WTW}	—	—	3000	ns	

■ CF I/O Space/IDE Timing

Table 30.17 CF I/O Space/IDE Read Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
Read cycle time	t_{IORC}	—	20	—	T_S	*2
IORD# active time	t_{IORW}	—	10	—	T_S	*2
Address setup time	t_{ADSIO}	—	6	—	T_S	*2
Address hold time	t_{ADHIO}	—	4	—	T_S	*2
CE effective time before IO read	$t_{CEVBIOR}$	—	5	—	T_S	*2
CE effective time after IO read	$t_{CEVAIOR}$	—	3	—	T_S	*2
REG effective time before IO read	$t_{REGVBIOR}$	—	6	—	T_S	*2
REG effective time after IO read	$t_{REGVAIOR}$	—	4	—	T_S	*2
Wait allowable time after IO read active	$t_{WTATIOR}$	—	—	6	T_S	*2
Data delay allowable time after wait release	t_{DATAWT}	—	—	0	T_S	*2
Wait active time	t_{WTW}	—	—	3000	ns	
Read data setup time	t_{RDS}	$1T_S$+16	—		ns	
Read data hold time	t_{RDH}	0			ns	

Table 30.18 CF I/O Space/IDE Write Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
Write cycle time	t_{IOWC}	—	20	—	T_S	*2
IOWR# active time	t_{IOWW}	—	10	—	T_S	*2
Address setup time	t_{ADSIO}	—	6	—	T_S	*2
Address hold time	t_{ADHIO}	—	4	—	T_S	*2
CE effective time before IO write	$t_{CEVBIOW}$	—	5	—	T_S	*2
CE effective time after IO write	$t_{CEVAIOW}$	—	3	—	T_S	*2
REG effective time before IO write	$t_{REGVBIOW}$	—	6	—	T_S	*2
REG effective time after IO write	$t_{REGVAIOW}$	—	4	—	T_S	*2
Data effective time before IO write	t_{DVBIOW}	—	14	—	T_S	*2
Data effective time after IO write	t_{DVAIOW}	—	3	—	T_S	*2
Wait allowable time after IO write	$t_{WTATIOW}$	—	—	6	T_S	*2
IO write inactive time after wait release	t_{WITAWT}	—	—	2	T_S	*2
Wait active time	t_{WTW}	—	—	3000	ns	

30.2.3 Timing Charts

30.2.3.1 Clock Timing

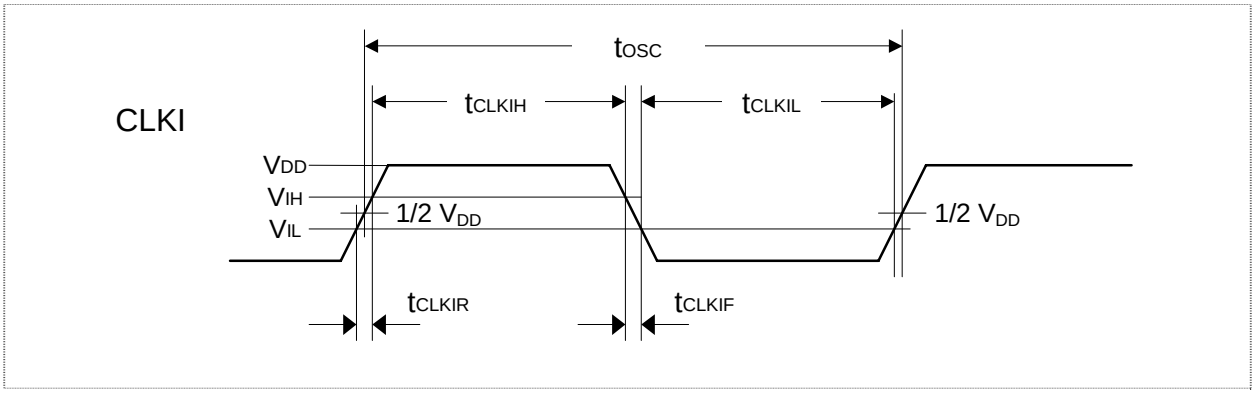


Fig.30.1 Clock Timing

30.2.3.2 CPU Control Signal Timing

■ RESET# Timing

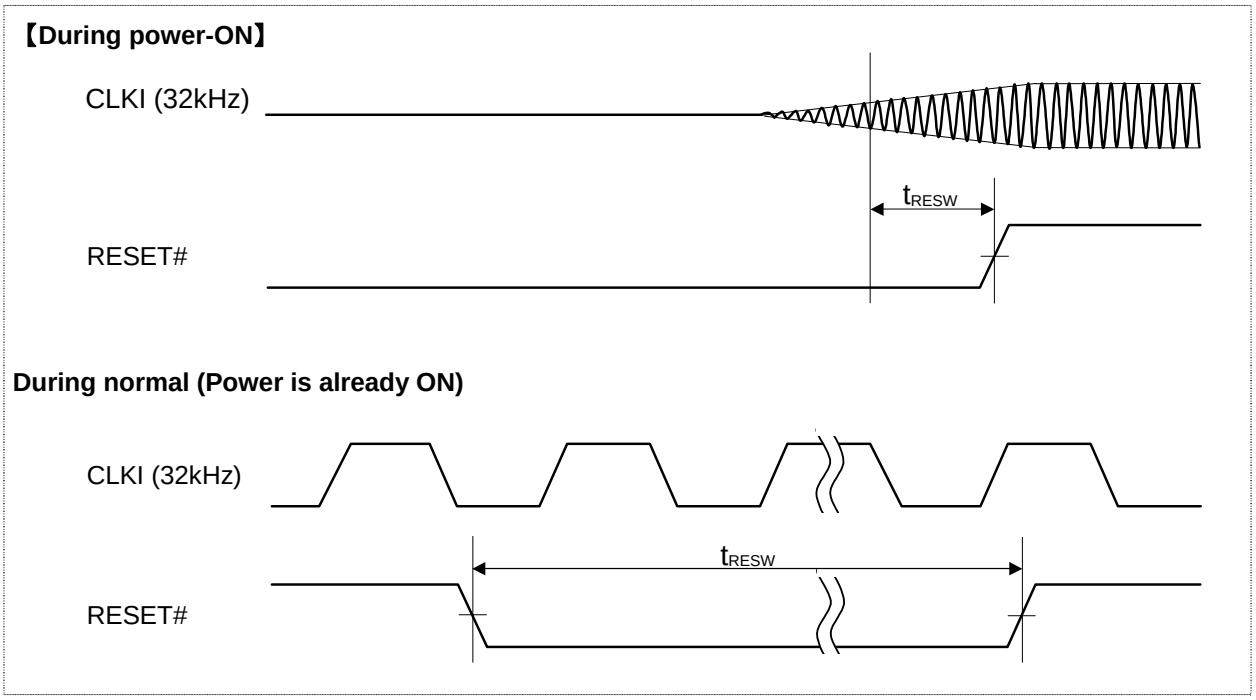


Fig.30.2 RESET# Timing

■ Interrupt Signal Timing

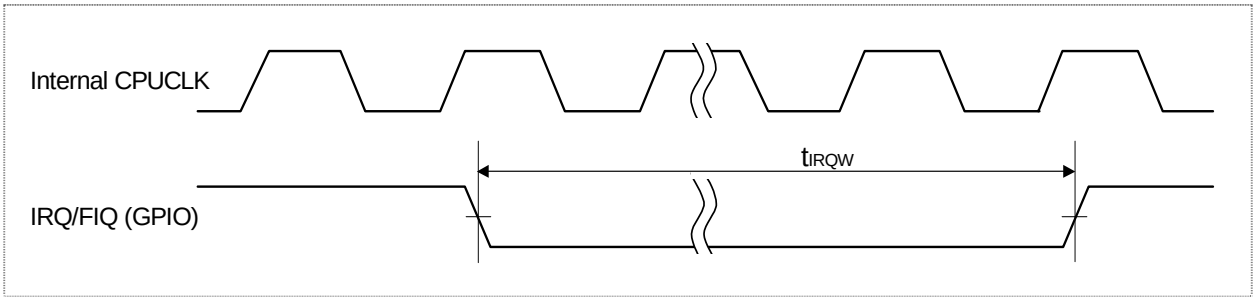
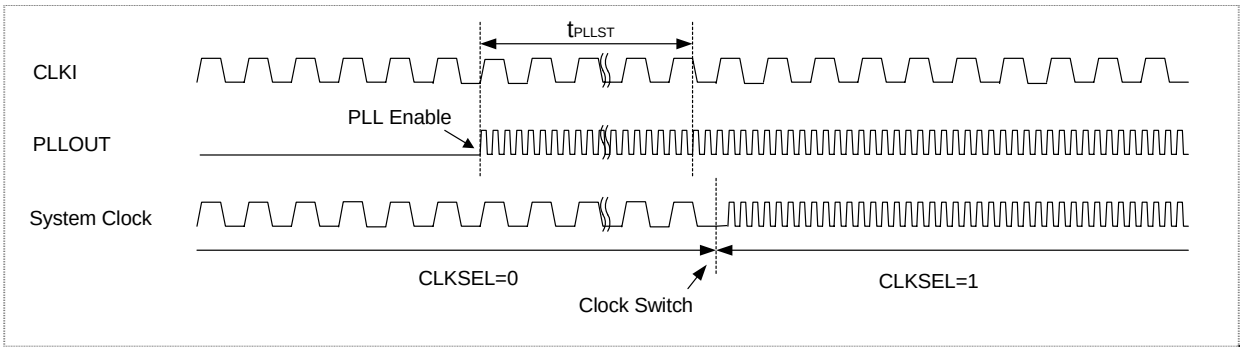


Fig.30.3 Interrupt Signal Timing

■ PLL Related Timing

(1) Clock Switch 1 (PLL Enable)



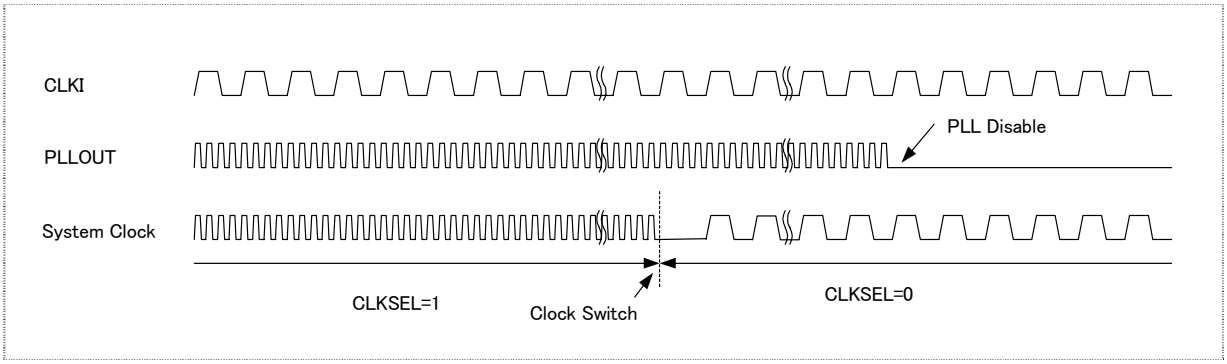
System Clock = CPUCLK / HCLK1 / HCLK2 / PCLK

Fig.30.4 Clock Switch 1 (PLL Enable)

This is the internal clock timing that is used to switch the 32KHz system clock to the PLL output.

Set bit 0 (PLLEN) of PLL Setting Register 2 (SYS[0x0C]) of the system controller to logical 1 and enable the PLL output. After the PLL stabilization time (t_{PLLST}) has passed, set bit 0 (CLKSEL) of Clock Select Register (SYS[0x18]) to logical 1 and select the PLL output as the system clock.

(2) Clock Switch 2 (PLL Disable)



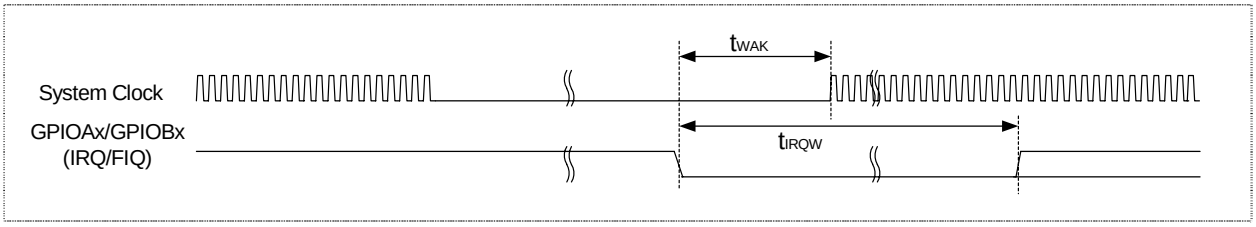
System Clock = CPUCLK / HCLK1 / HCLK2 / PCLK

Fig.30.5 Clock Switch 2 (PLL Disable)

This is the internal clock timing that is used to switch the PLL output to the 32KHz system clock.

Set bit 0 (CLKSEL) of Clock Select Register (SYS[0x18]) to logical 0, and select the CLKI (32KHz) as the system clock. Then, set bit 0 (PLLEN) of PLL Setting Register 2 (SYS[0x0C]) of the system controller to logical 1, and disable the PLL signal.

(3) Clock Restart by Interrupt in High-Speed HALT Mode

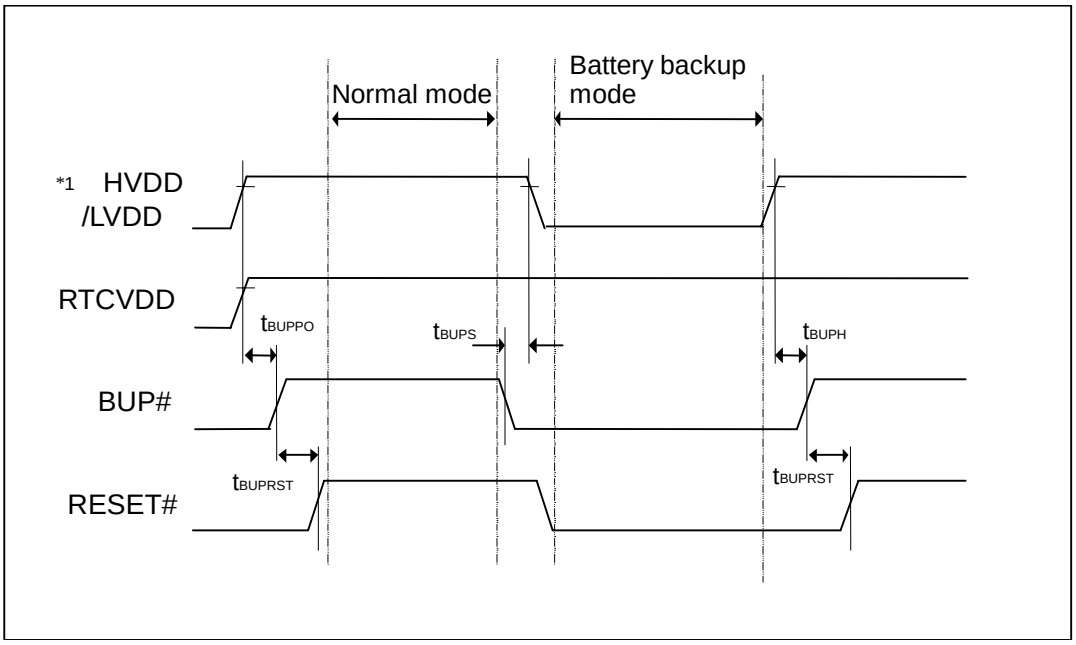


System Clock = CPUCLK / HCLK1 / HCLK2 / PCLK

Fig. 30.6 Clock Restart Timing

This is the signal timing that is used to transition to the High-Speed mode when an interrupt occurs in the High-Speed HALT mode. The system clock restarts after the clock restart time (t_{WAK}) has passed. The interrupt pulse width (t_{IRQW}) must be longer enough than this timing.

30.2.3.3 Battery Backup Mode Timing



*1 : Contains C1VDD,C2VDD,SDVDD,AVDD,ULDD3,UPVDD,UXVDD,PLLVDD

30.2.3.4 Camera Interface Timing

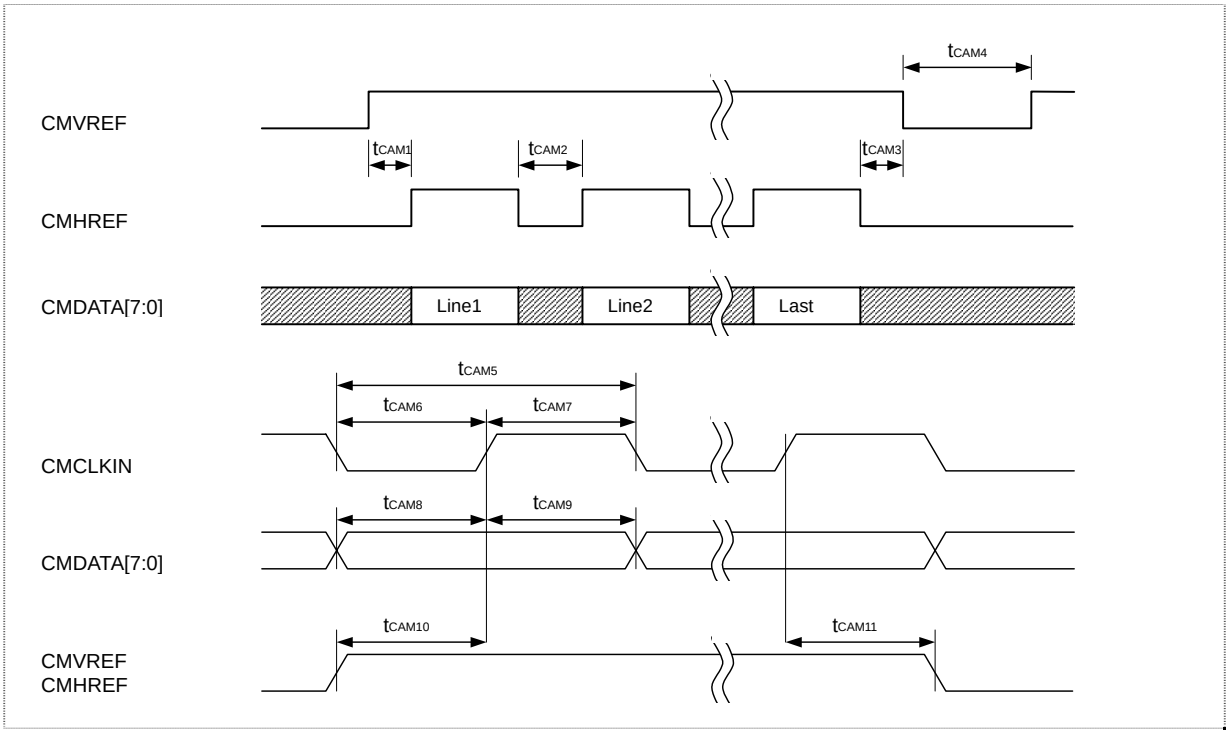


Fig.30.7 Camera Interface Timing

The effective CMCLKIN signal edge can be changed by the software. This figure shows a timing where data is read when the CMCLKIN signal changes from logical Low to logical High.

30.2.3.5 Memory Interface Controller

Static Memory Controller Timing (Flash EEPROM, SRAM, etc.)

■ Static Memory Read Timing

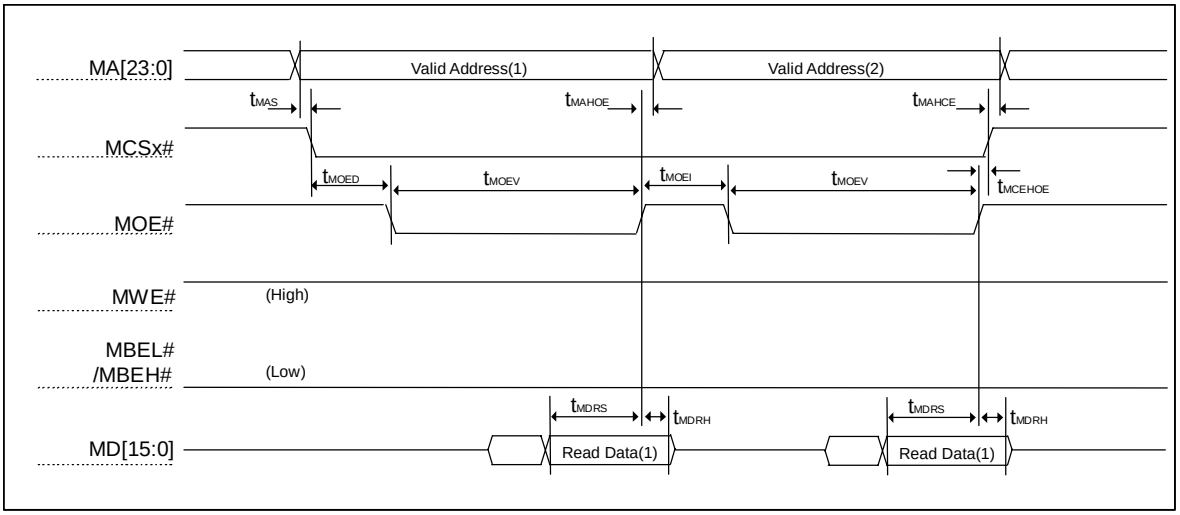


Fig.30.8 Static Memory Read Timing

■ Static Memory Write Timing

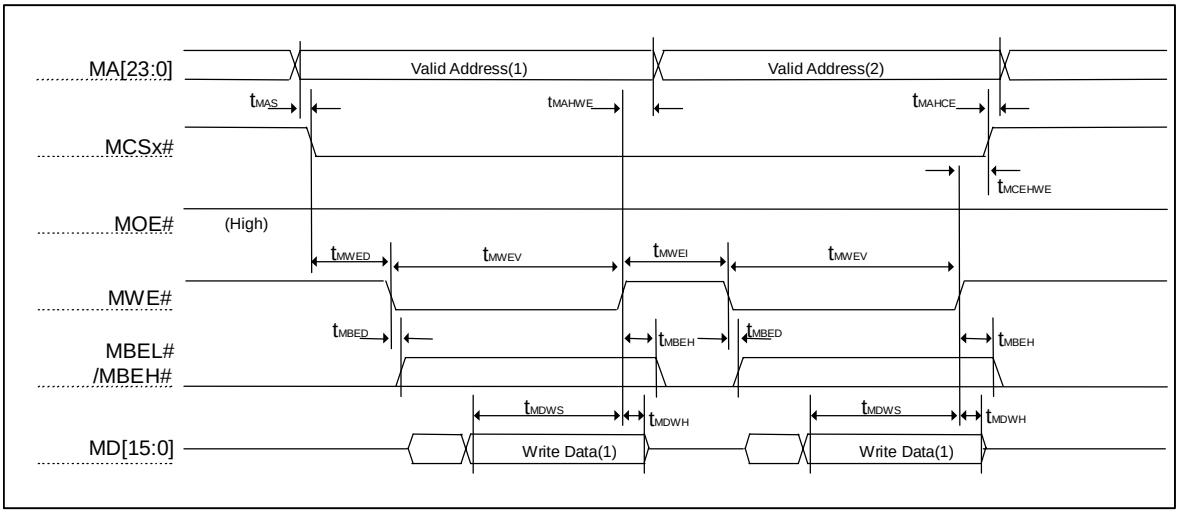


Fig.30.9 Static Memory Write Timing

SDRAM Controller AC Timing

The following shows an AC timing of the SDRAM controller.
The commands used in the figure are defined on this table.

Command	Function	MCS2#	MRAS#	MCAS#	MWE1#	Address and others
ACT	Bank active	L	L	H	H	Bank/Row
RD	Read	L	H	L	H	Bank/Col
WR	Write	L	H	L	L	Bank/Col
BT	Burst terminate	L	H	H	L	—
PCGA	Precharge all bank	L	L	H	L	SDA10 = HIGH
PCG	Precharge	L	L	H	L	SDA10 = LOW
AREF	Auto refresh	L	L	L	H	SDCLKEN = HIGH
SELF_IN	Self-refresh start	L	L	L	H	SDCLKEN = LOW
SELF_OUT	Self-refresh terminate	H	x	x	x	SDCLKEN = HIGH
LMR	Mode register set	L	L	L	L	—

■ SDRAM Read Cycles

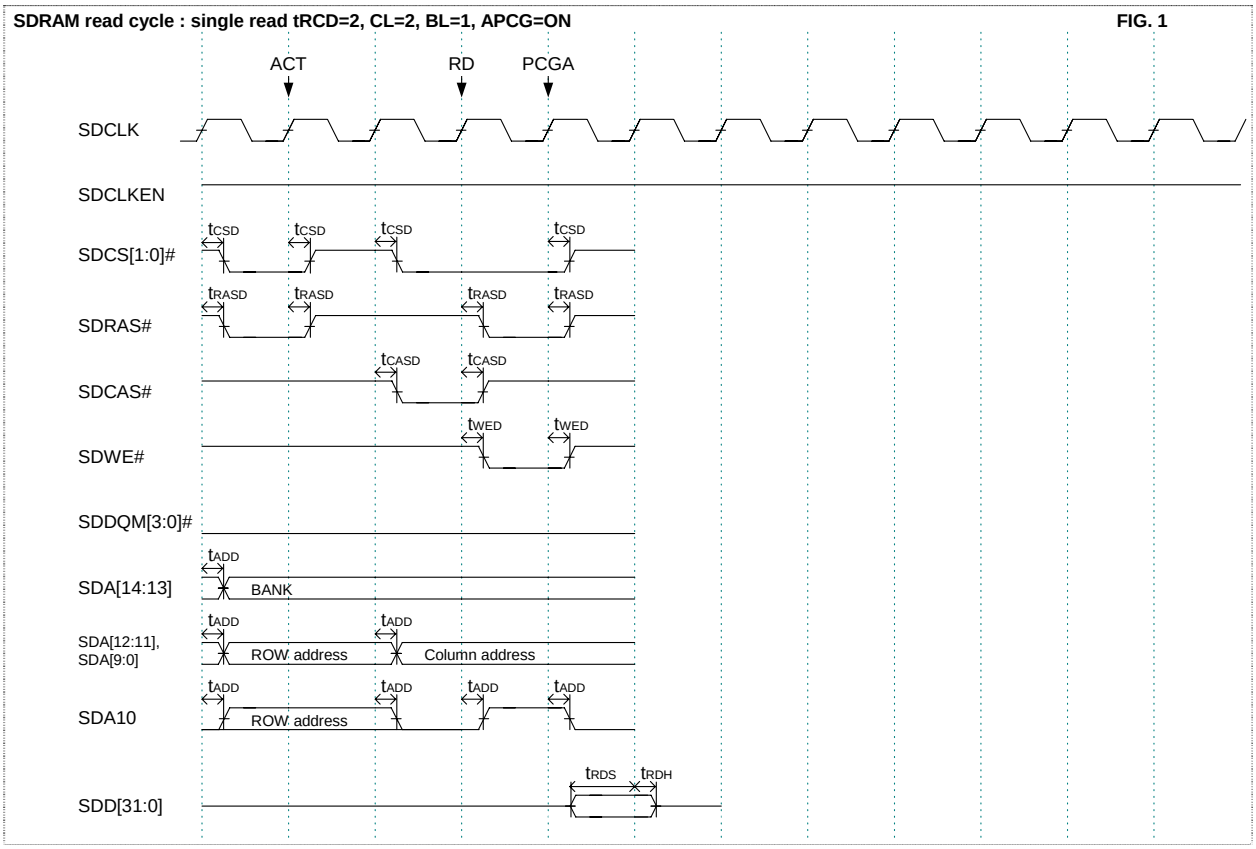


Fig.30.10 SDRAM Read Cycle 1: Single read cycle; tRCD=2, CL=2, BL=1, APCG=ON

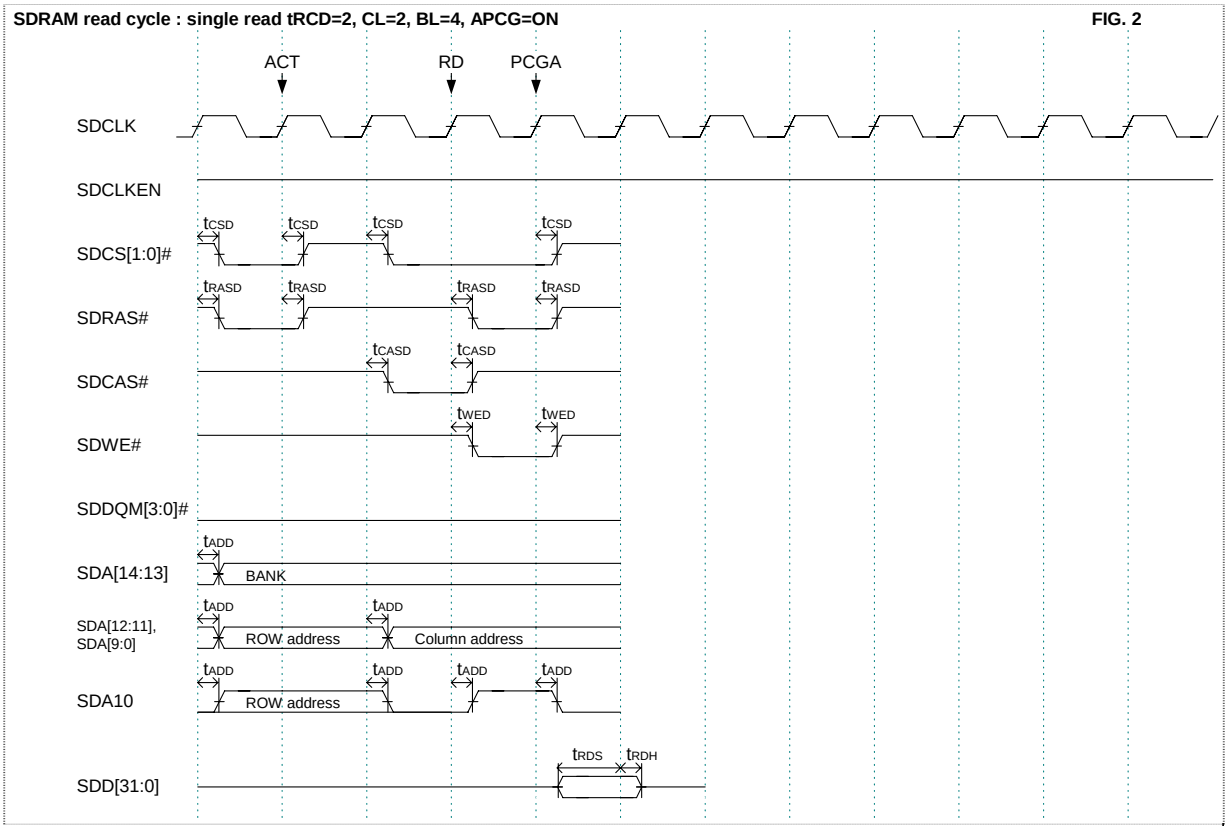


Fig.30.11 SDRAM Read Cycle 2: Single read cycle; tRCD=2, CL=2, BL=4, APCG=ON

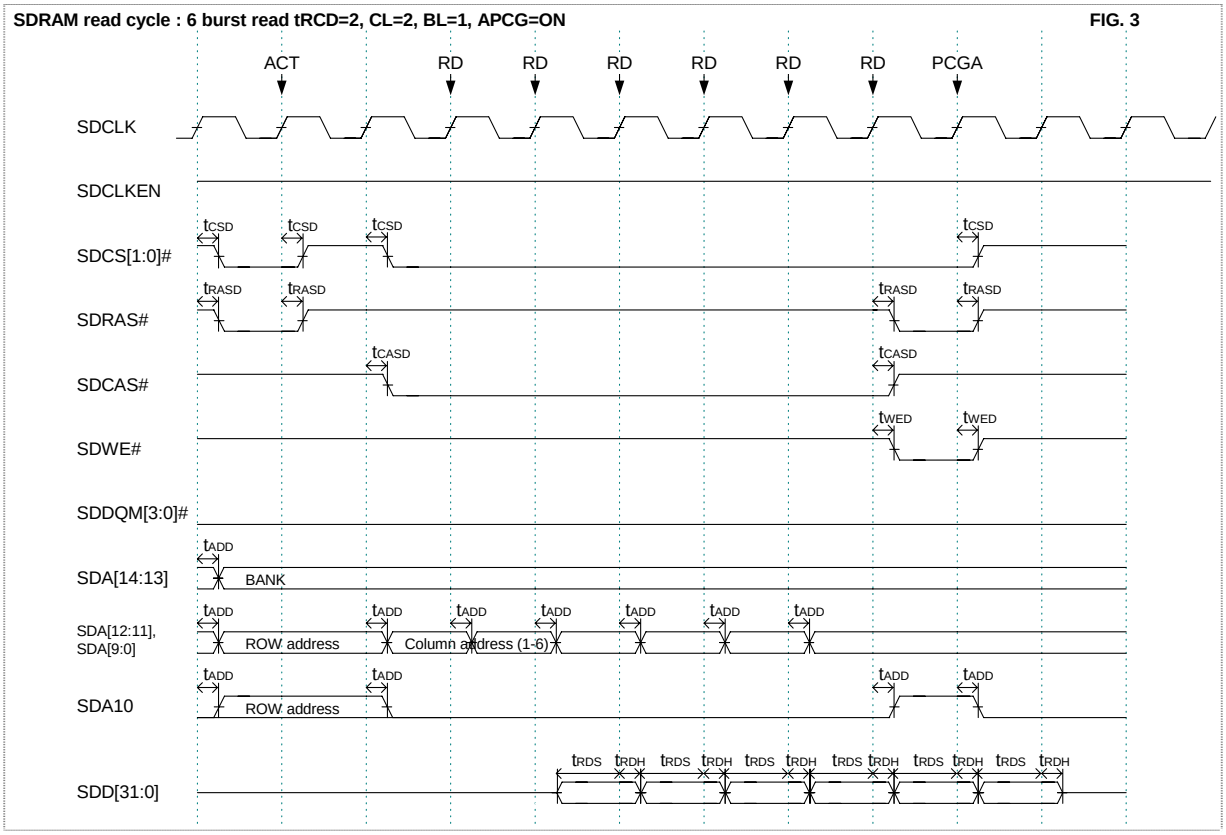


Fig.30.12 SDRAM Read Cycle 3: 6 burst read cycles; tRCD=2, CL=2, BL=1, APCG=ON

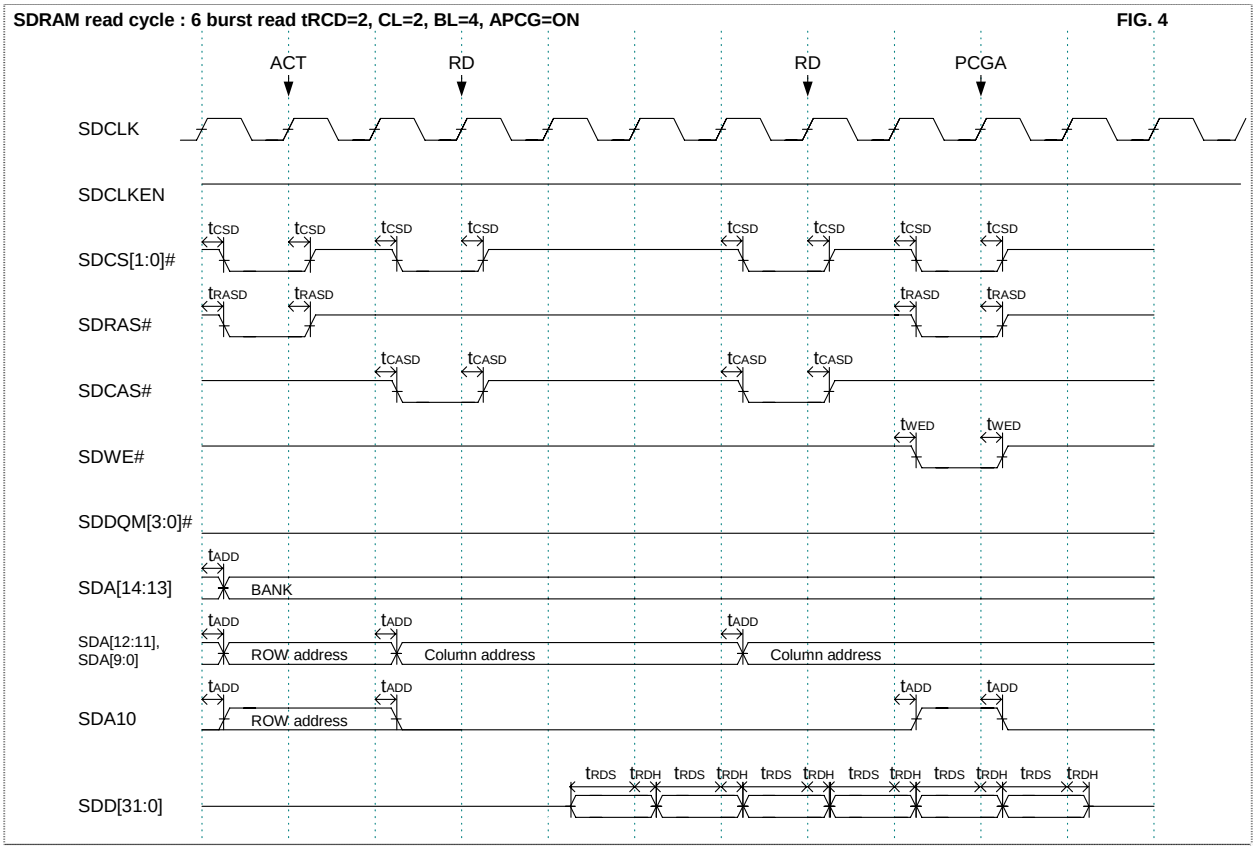


Fig.30.13 SDRAM Read Cycle 4: 6 burst read cycles; tRCD=2, CL=2, BL=4, APCG=ON

■ SDRAM Write Cycles

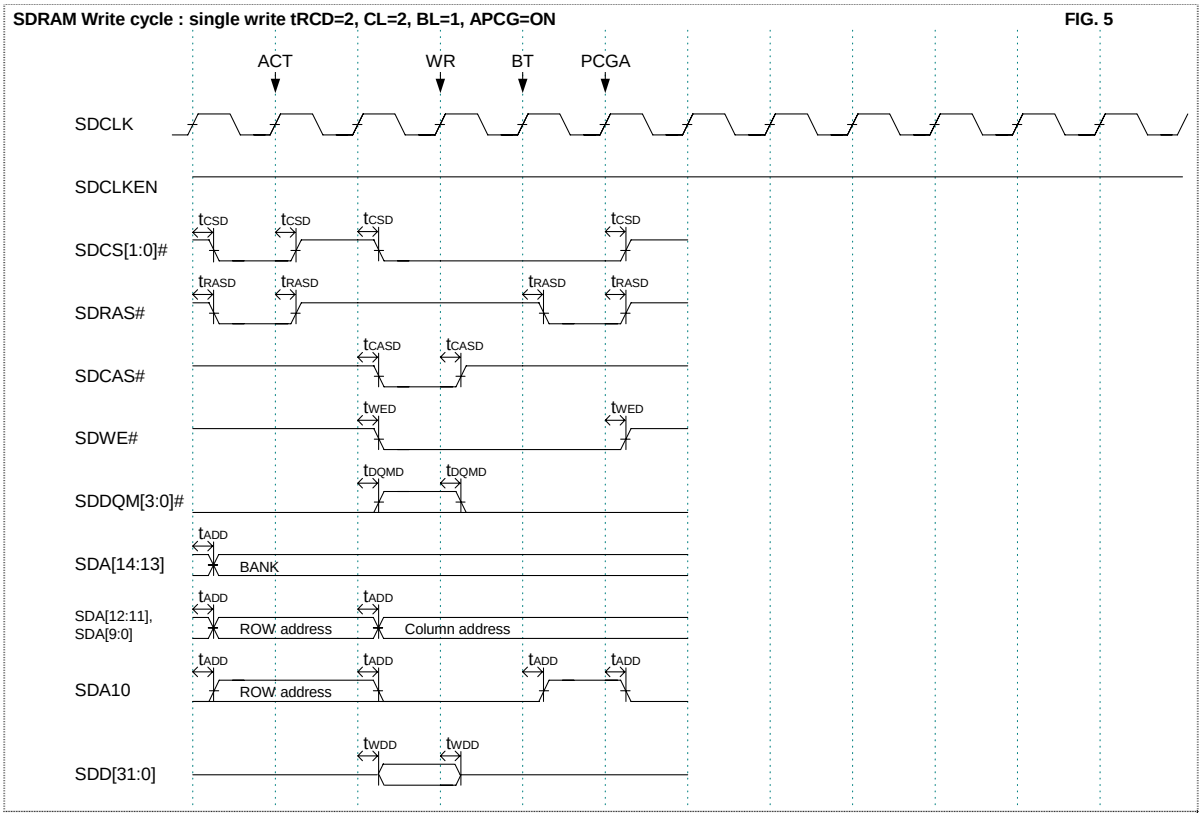


Fig.30.14 SDRAM Write Cycle 1: Single write cycle; tRCD=2, CL=2, BL=1, APCG=ON

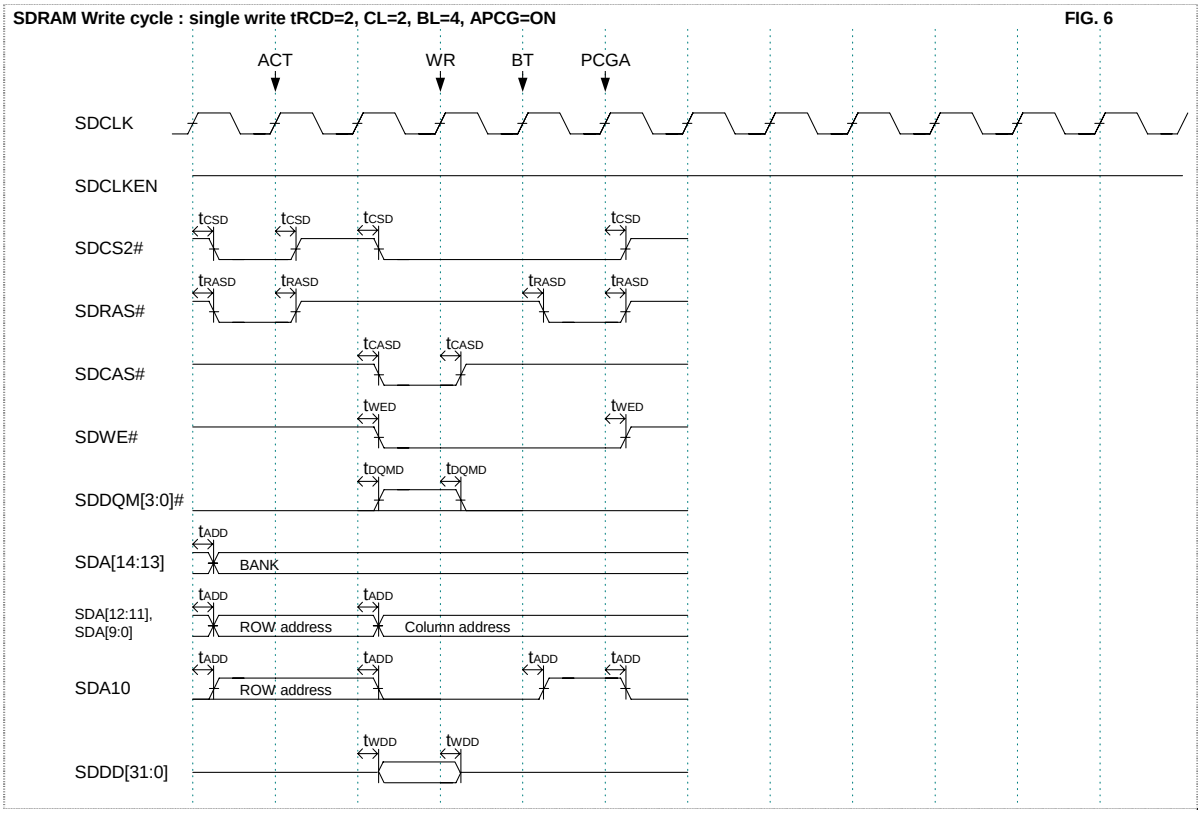


Fig.30.15 SDRAM Write Cycle 2: Single write cycle; tRCD=2, CL=2, BL=4, APCG=ON

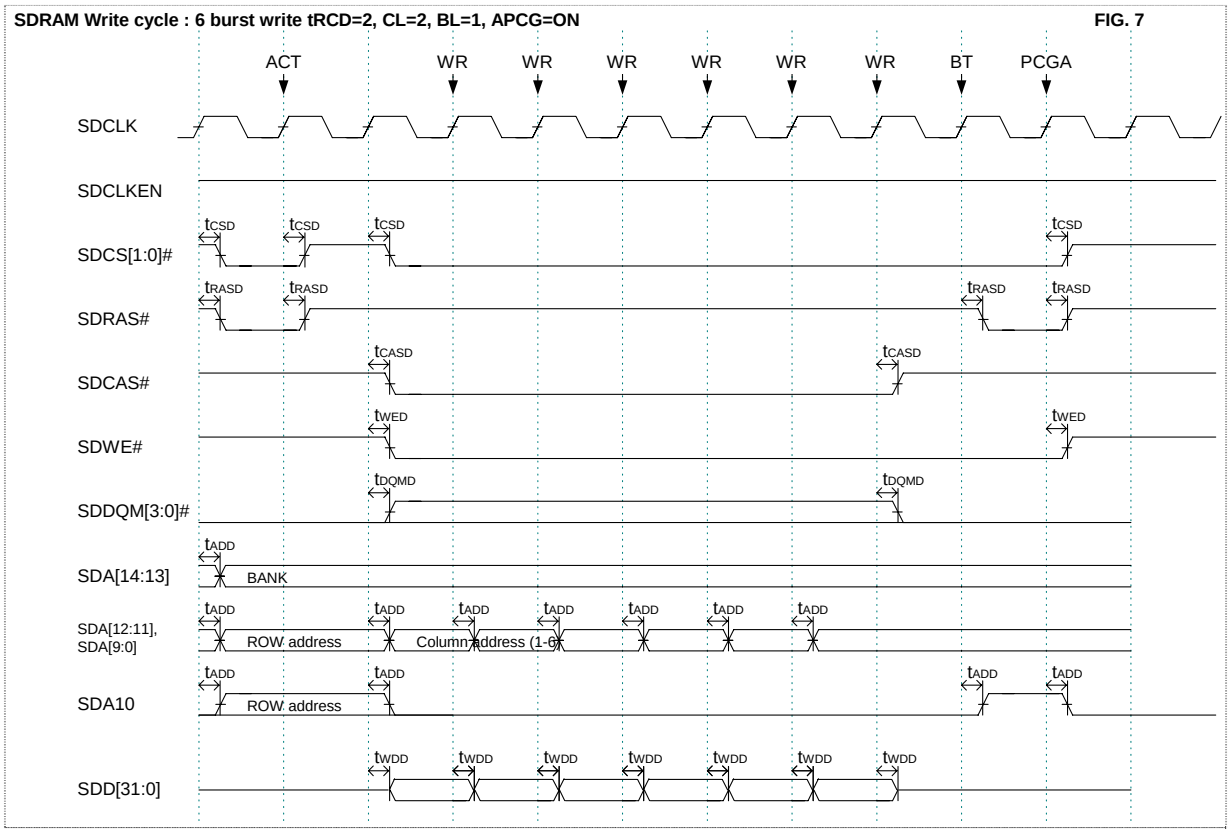


Fig.30.16 SDRAM Write Cycle 3: 6 burst write cycles; tRCD=2, CL=2, BL=1, APCG=ON

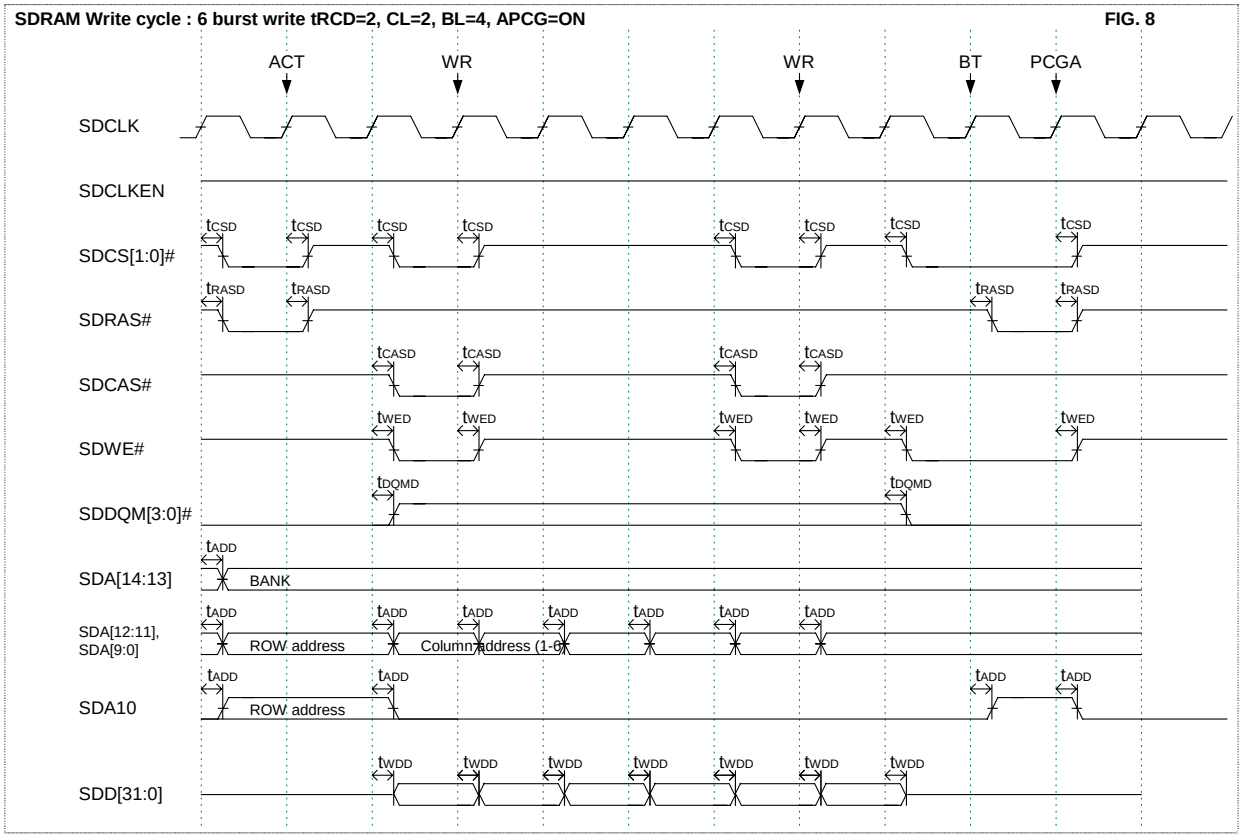


Fig.30.17 SDRAM Write Cycle 4: 6 burst write cycles; tRCD=2, CL=2, BL=4, APCG=ON

■ SDRAM Read Cycles (Row Active Mode)

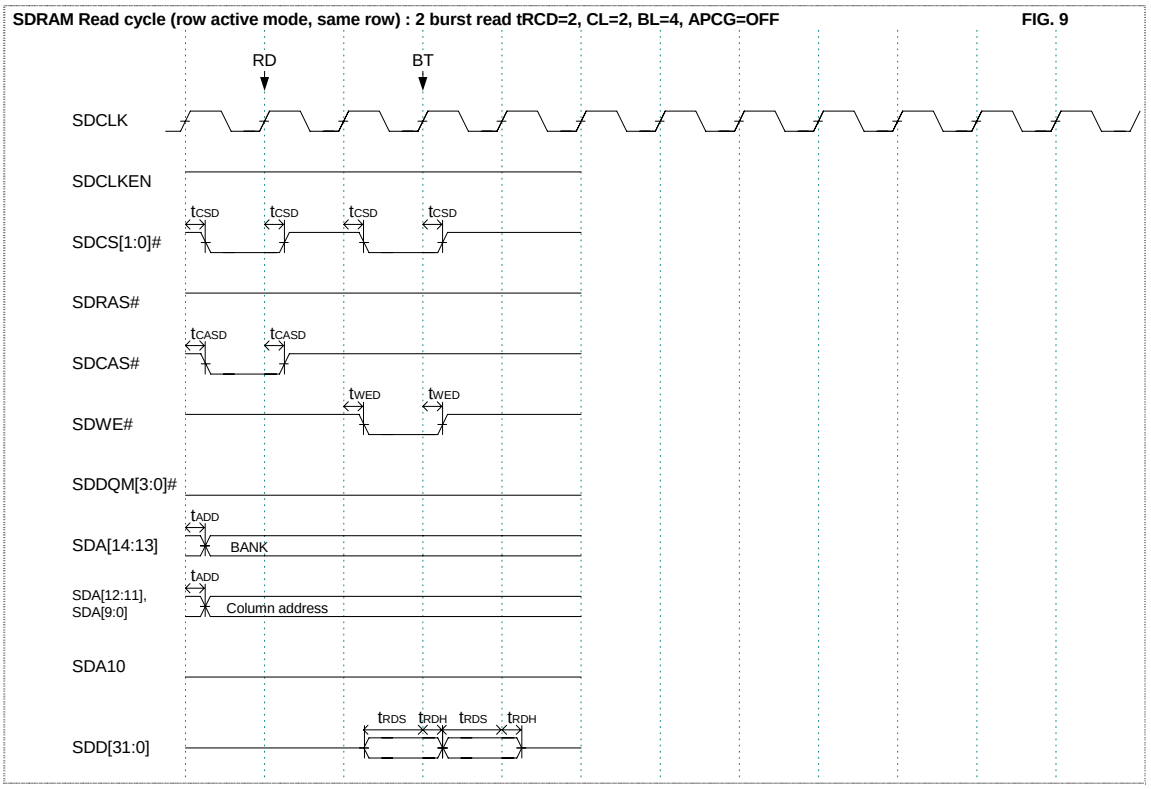


Fig.30.18 SDRAM Read Cycle, Row Active Mode 1 (Same row):
2 burst read cycles;tRCD=2, CL=2, BL=4, APCG=OFF

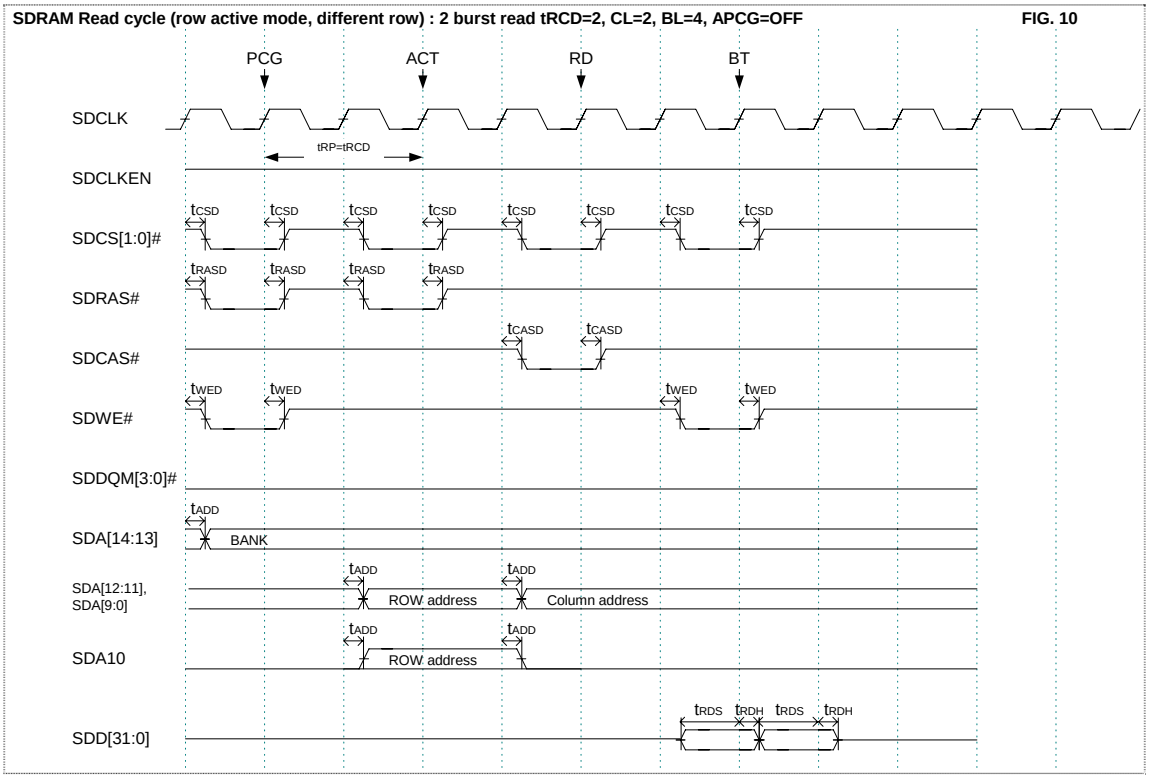


Fig.30.19 SDRAM Read Cycle, Row Active Mode 2 (Different row):
2 burst read cycles;tRCD=2, CL=2, BL=4, APCG=OFF

■ SDRAM Write Cycle (Row Active Mode)

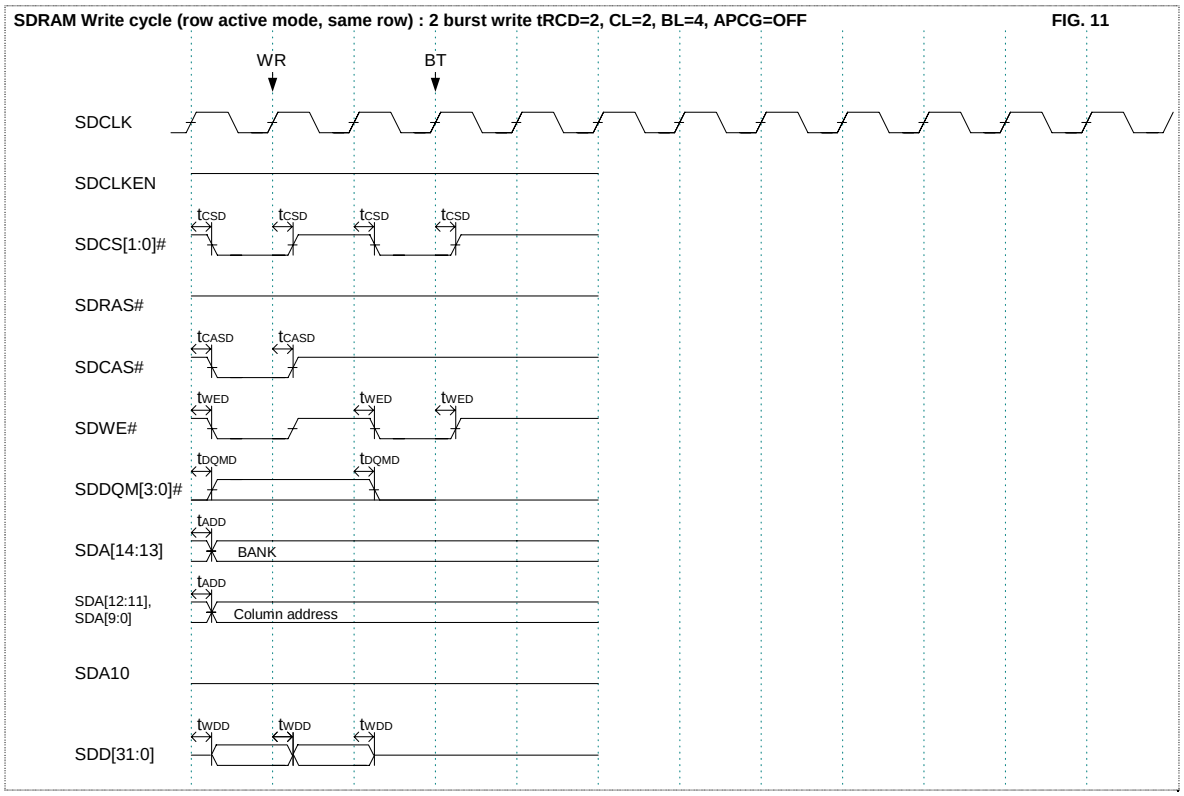


Fig.30.20 SDRAM Write Cycle, Row Active Mode 1 (Same row): 2 burst write cycles;tRCD=2, CL=2, BL=4, APCG=OFF

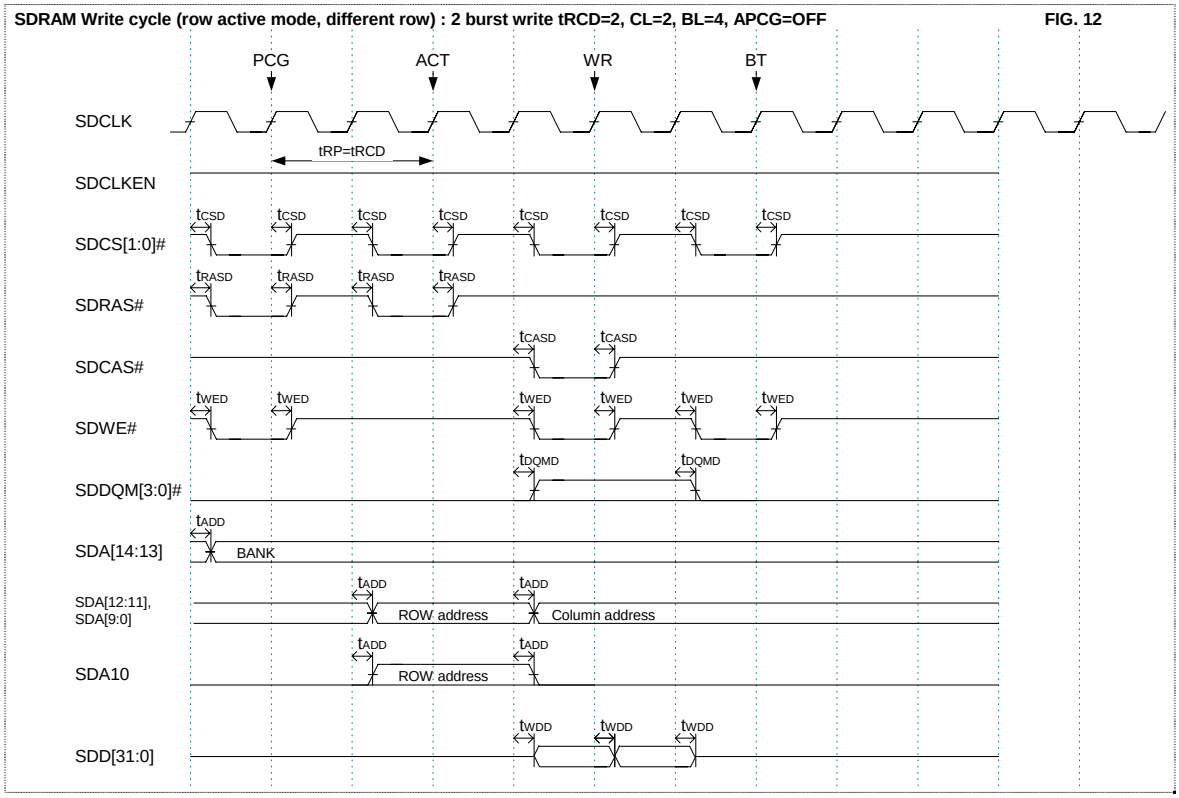


Fig.30.21 SDRAM Write Cycle, Row Active Mode 2 (Different row): 2 burst write cycles;tRCD=2, CL=2, BL=4, APCG=OFF

■ SDRAM Auto Refresh Cycle

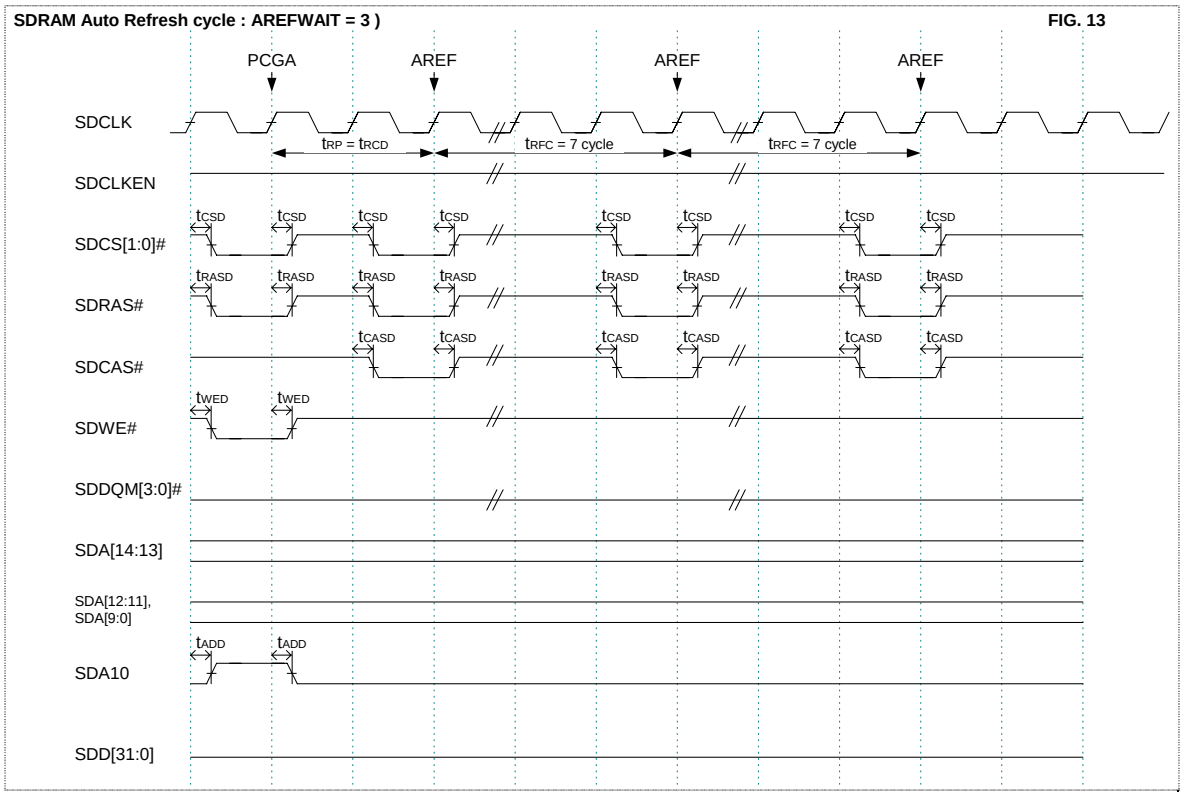


Fig.30.22 SDRAM Auto Refresh Cycle: AREFWAIT=3

■ SDRAM Self-Refresh Cycle

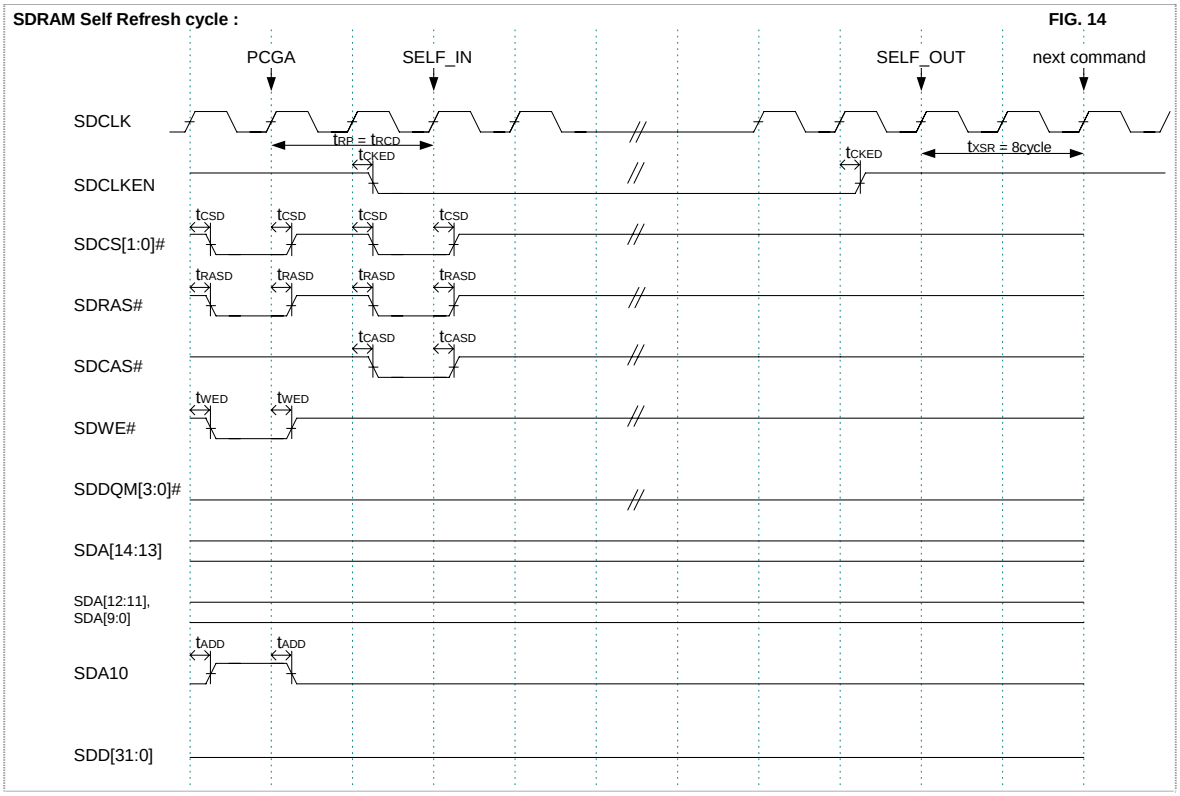


Fig.30.23 SDRAM Self-Refresh Cycle

■ SDRAM Initialization Cycle

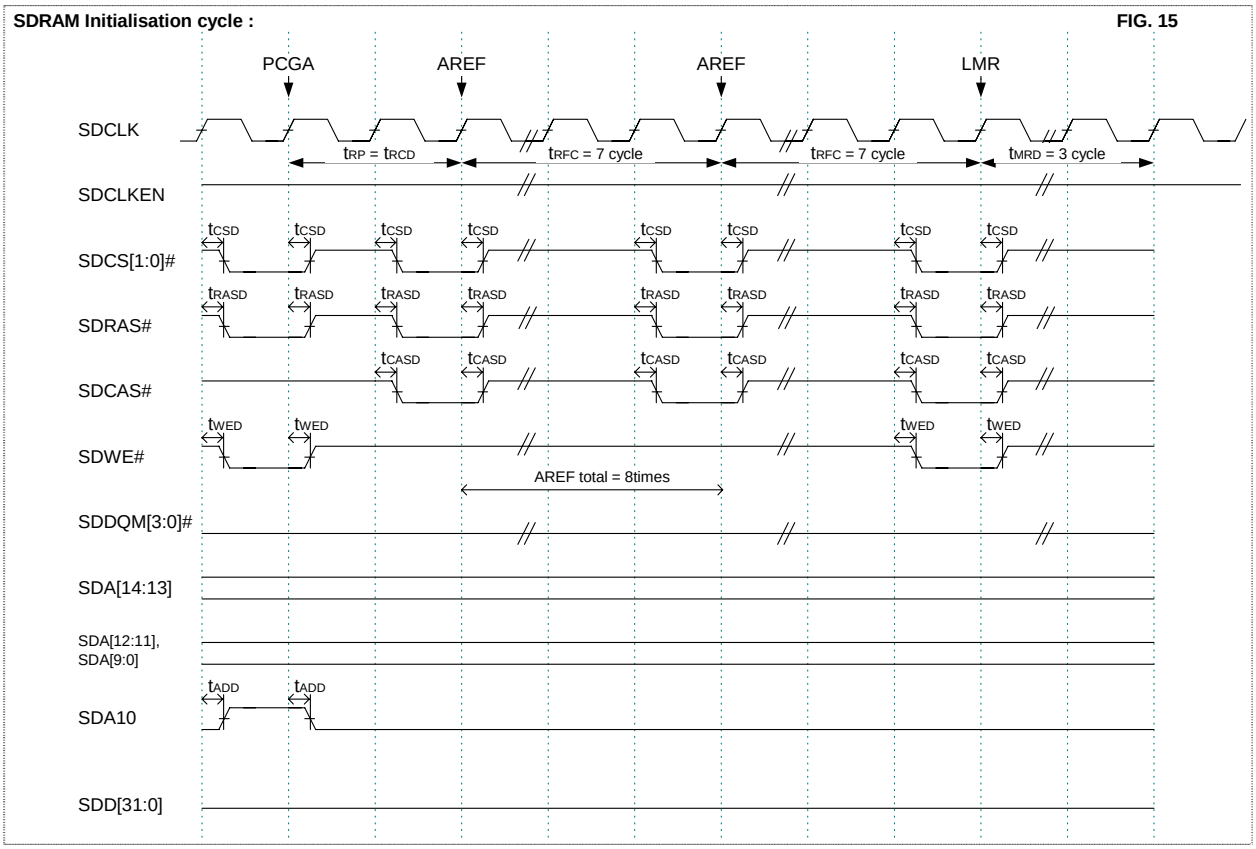


Fig.30.24 SDRAM Initialization Cycle

■ SDCLK/SDCLKEN Control

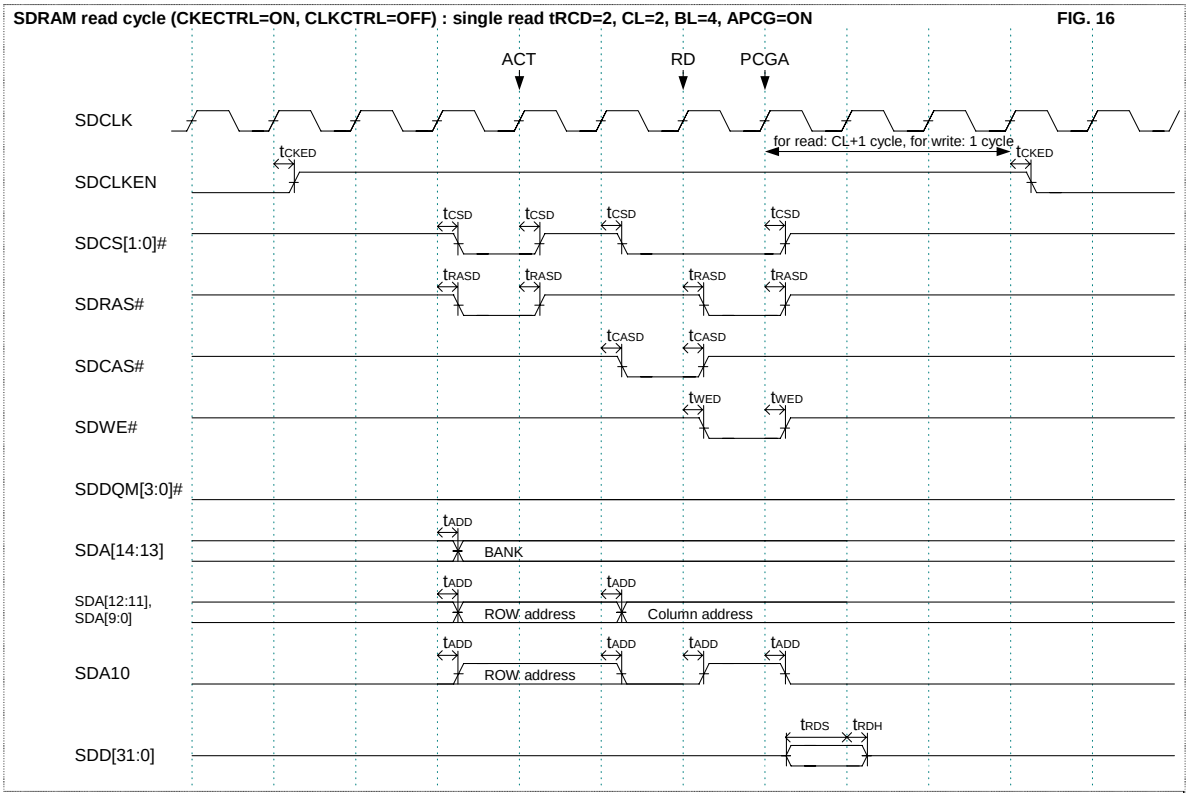


Fig.30.25 SDCLK/SDCLKEN Control 1 (CKECTRL=ON, CLKCTRL=OFF): Single read cycle; tRCD=2, CL=2, BL=4, APCG=ON

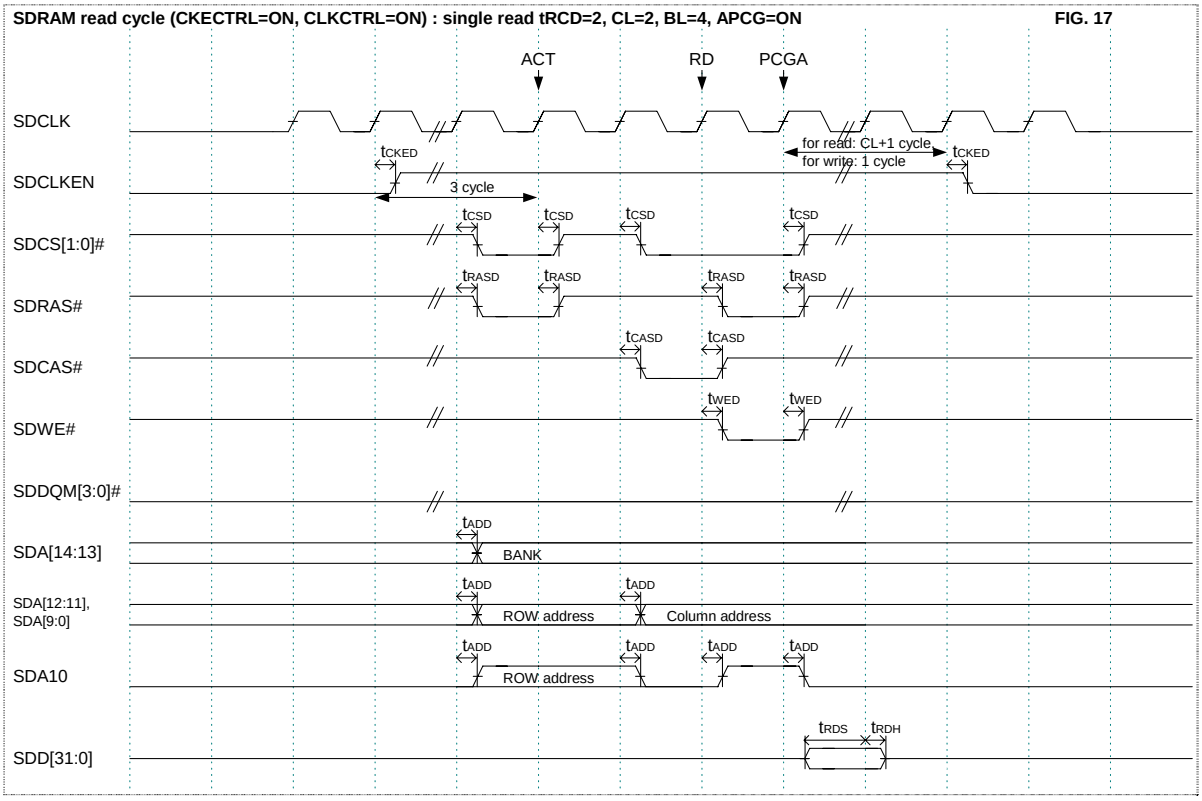


Fig.30.26 SDCLK/SDCLKEN Control 2 (CKECTRL=ON, CLKCTRL=ON): Single read cycle; tRCD=2, CL=2, BL=4, APCG=ON

30.2.3.6 I2C Single Master Core Module Timing

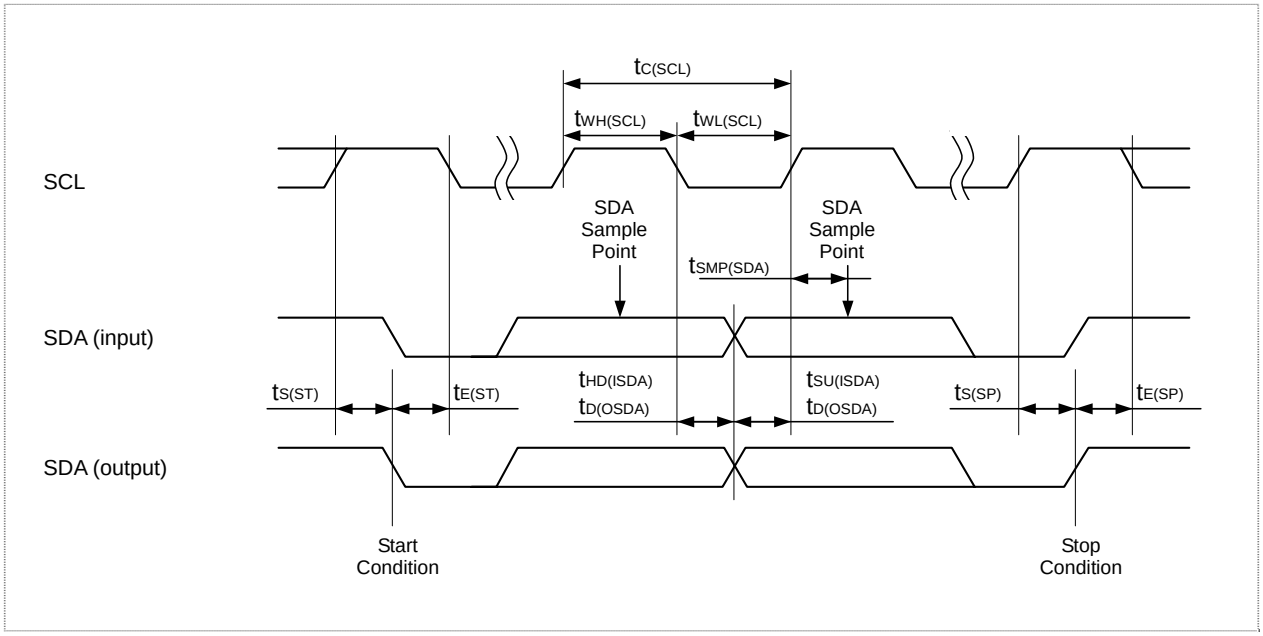


Fig.30.27 I2C Single Master Core Module Timing

30.2.3.7 I2STiming

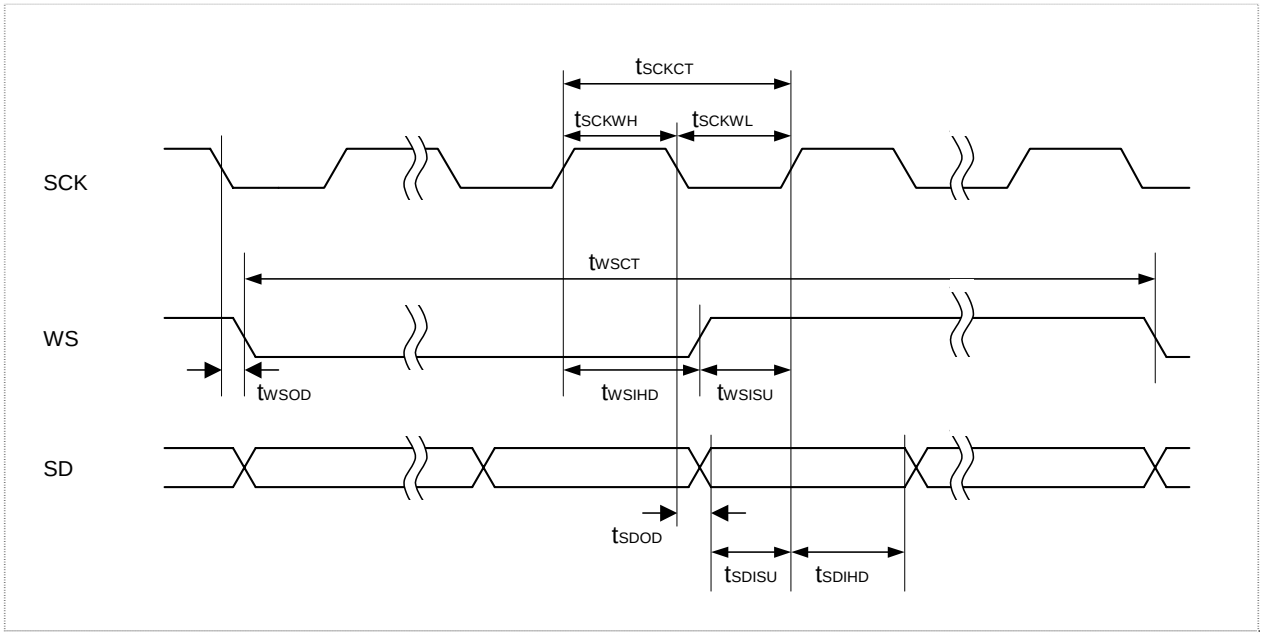


Fig.30.28 I2S Timing

30.2.3.8 Serial Peripheral Device Interface Timing

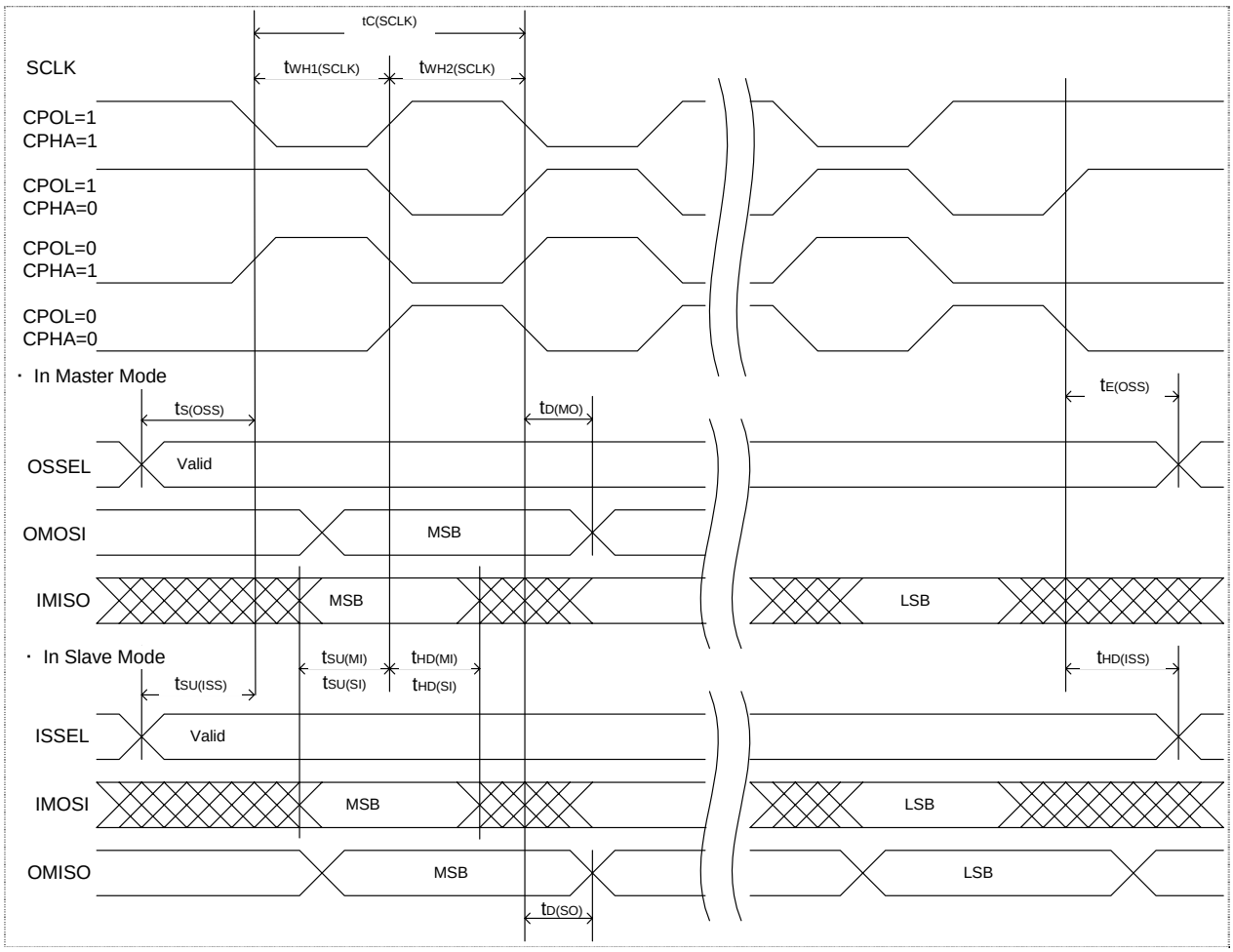


Fig.30.29 Serial Peripheral Device Interface Timing

30.2.3.9 Compact Flash Memory Interface (CF) Timing

■ CF Attribute Memory Read Cycle

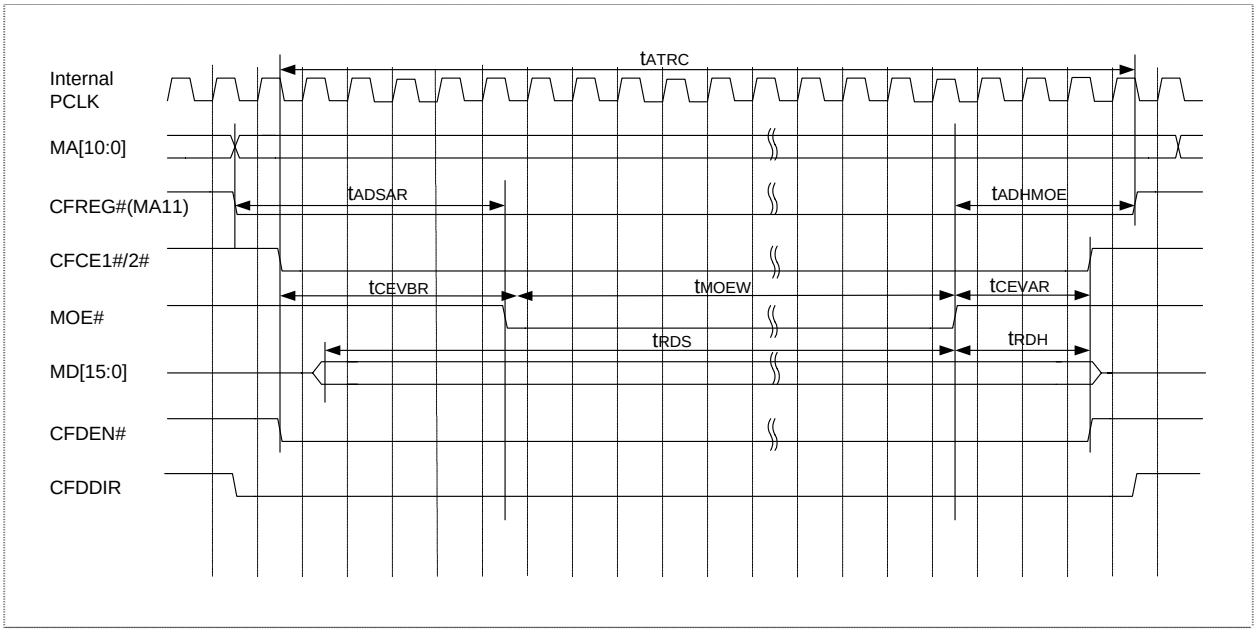


Fig.30.30 CF Attribute Memory Read Cycle

■ CF Attribute Memory Write Cycle

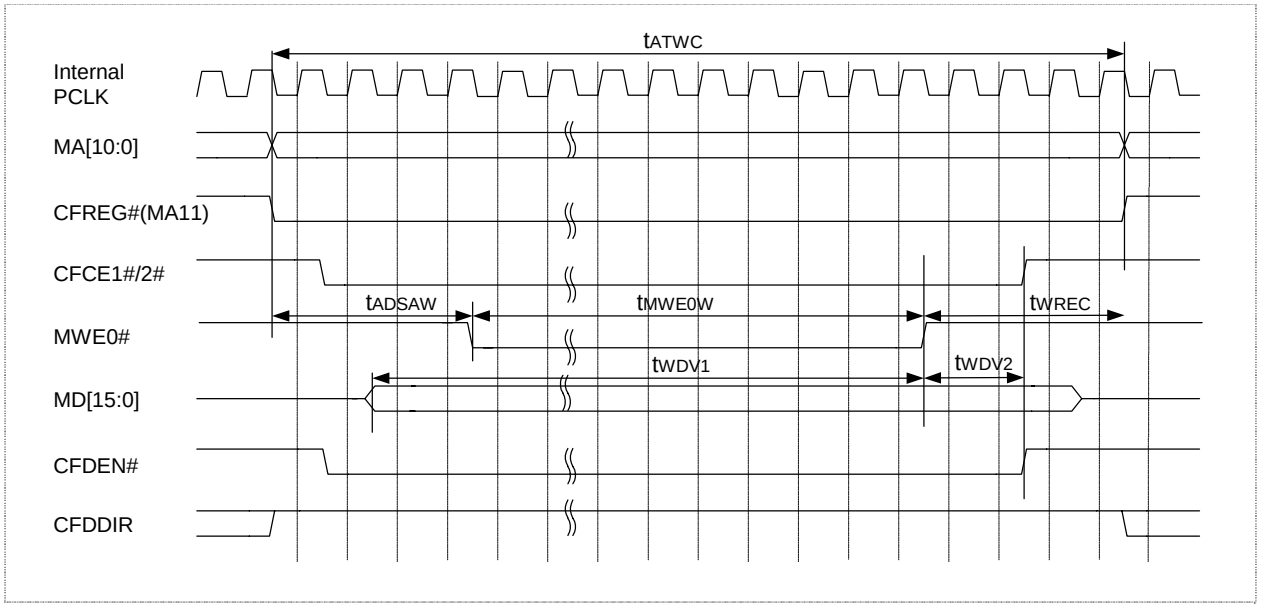


Fig.30.31 CF Attribute Memory Write Cycle

■ CF Common Memory Read Cycle

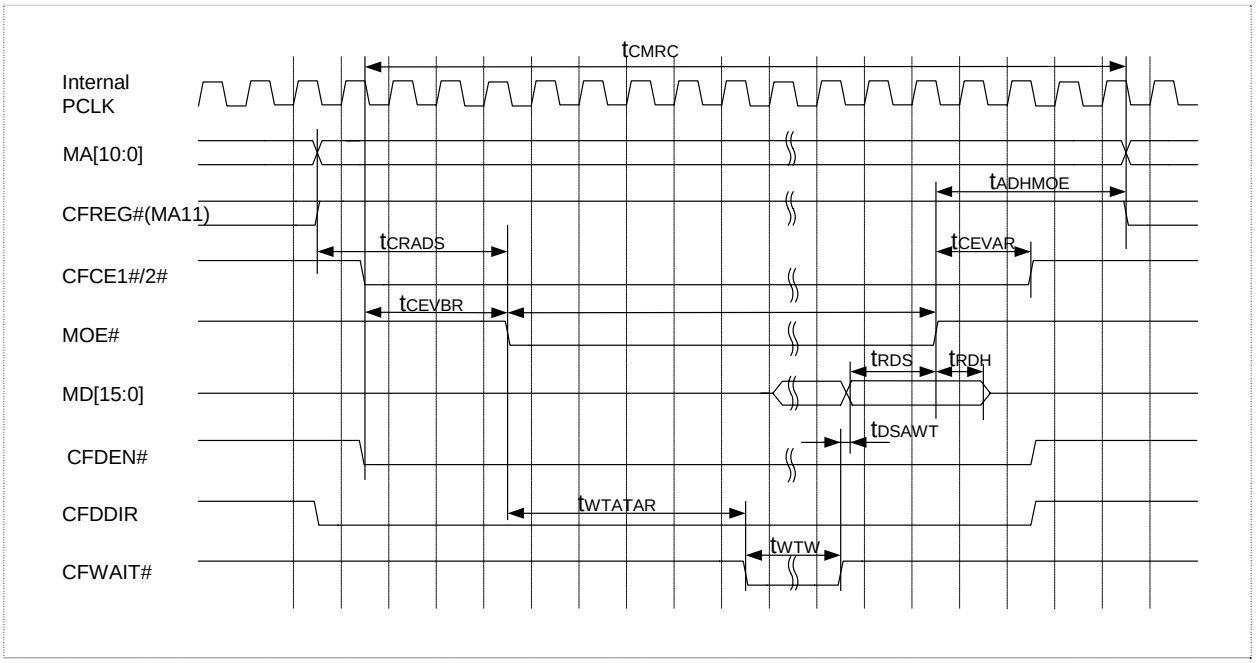


Fig.30.32 CF Common Memory Read Cycle

■ CF Common Memory Write Cycle

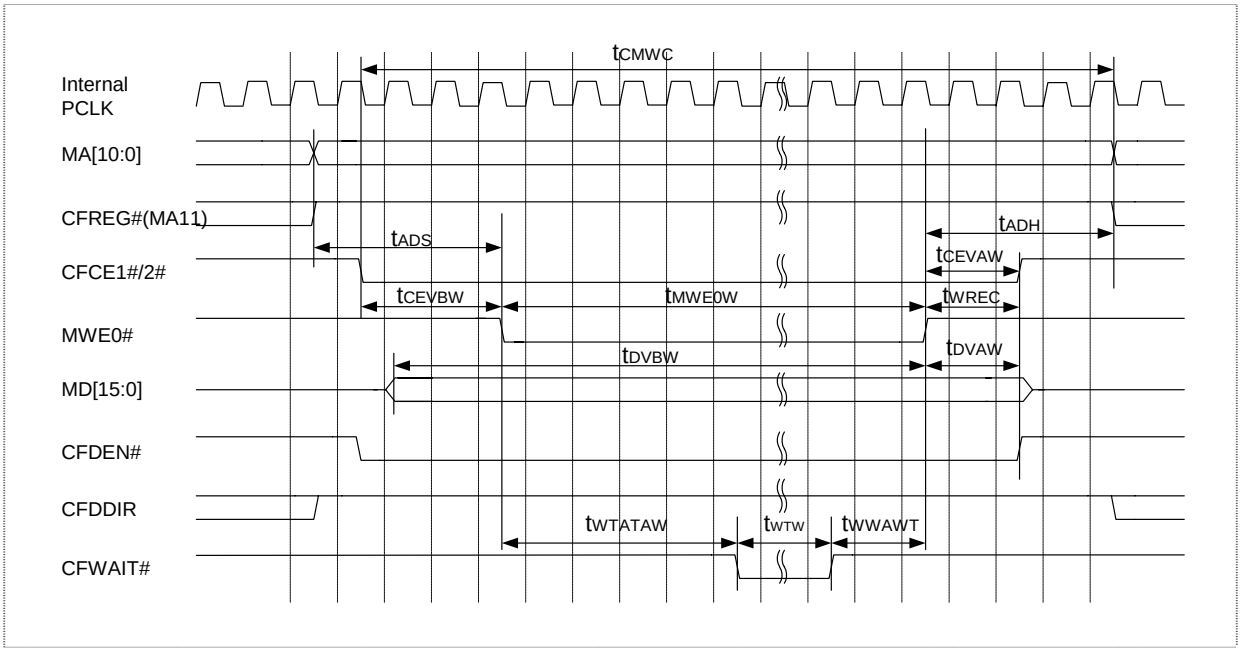


Fig.30.33 CF Common Memory Write Cycle

■ CF I/O Space/IDE Read Cycle

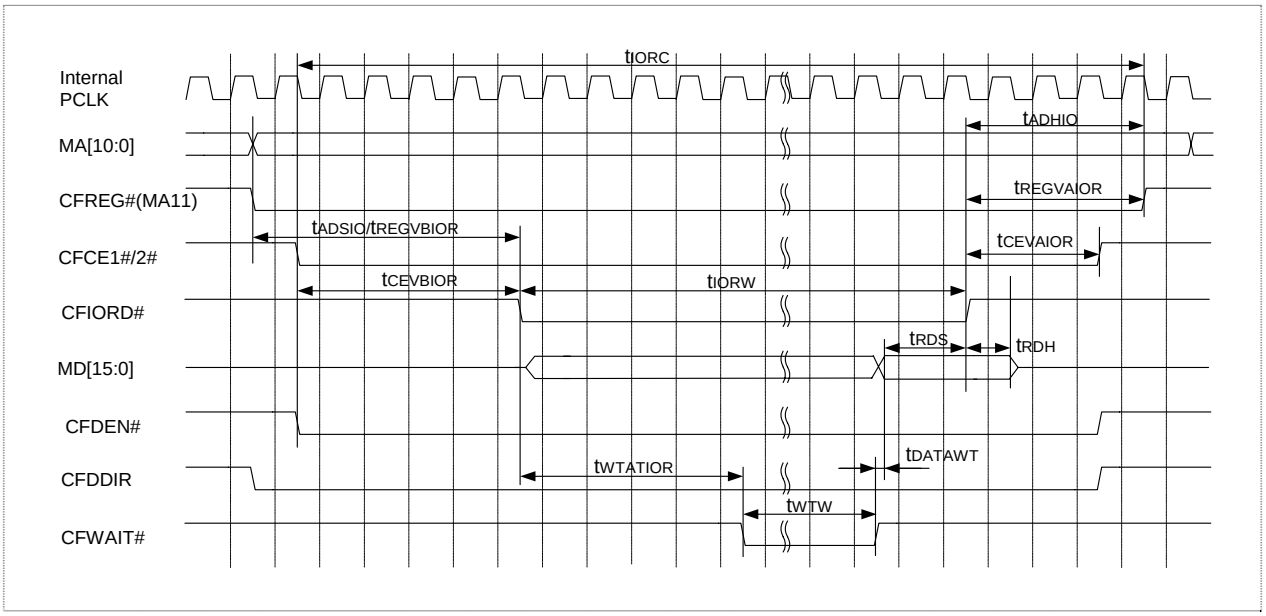


Fig.30.34 CF I/O Space/IDE Read Cycle

■ CF I/O Space/IDE Write Cycle

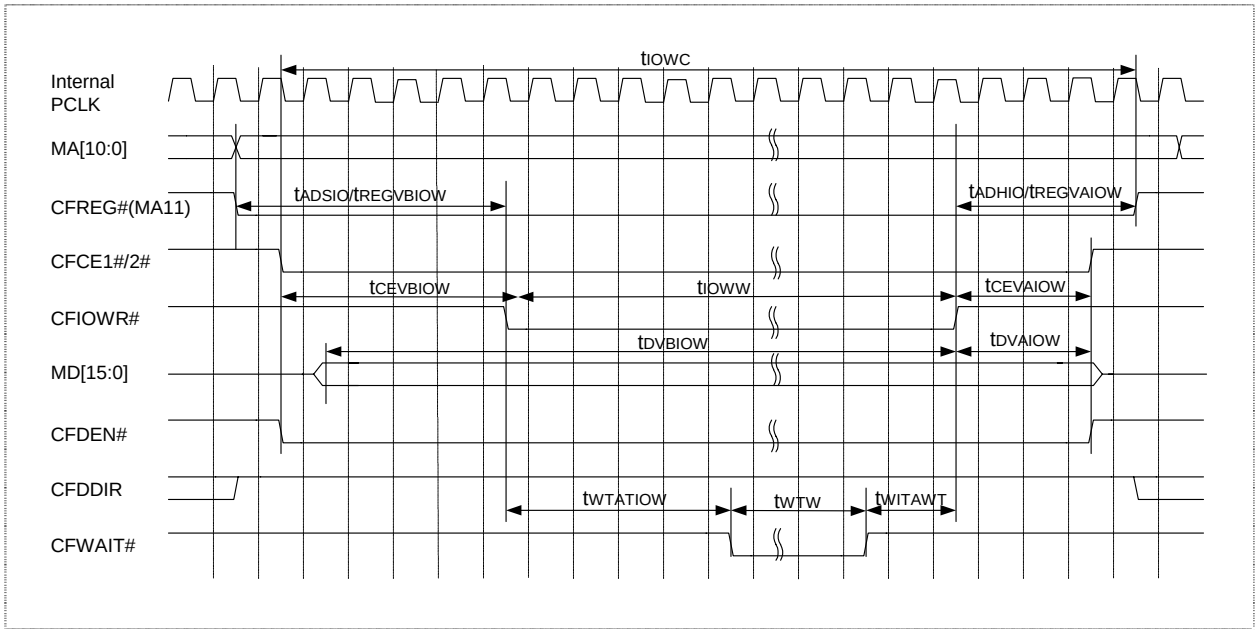


Fig.30.35 CF I/O Space/IDE Write Cycle

31. External Connection Examples (Reference)

31.1 Memory Connection Examples

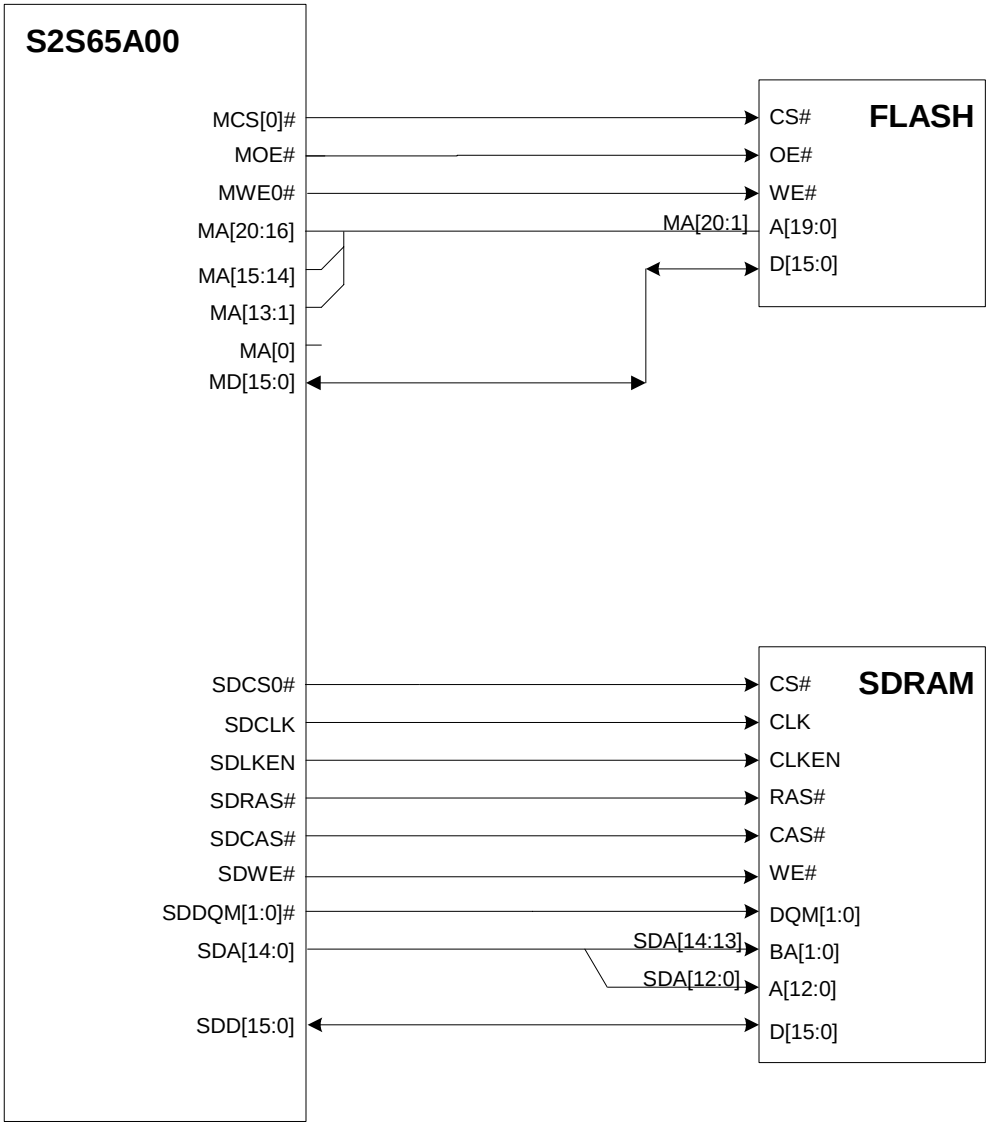


Fig.31.1 Memory Connection Example (1)

Note: Connect SDMA[14:13] to the SDRAM bank address (BA[1:0]).

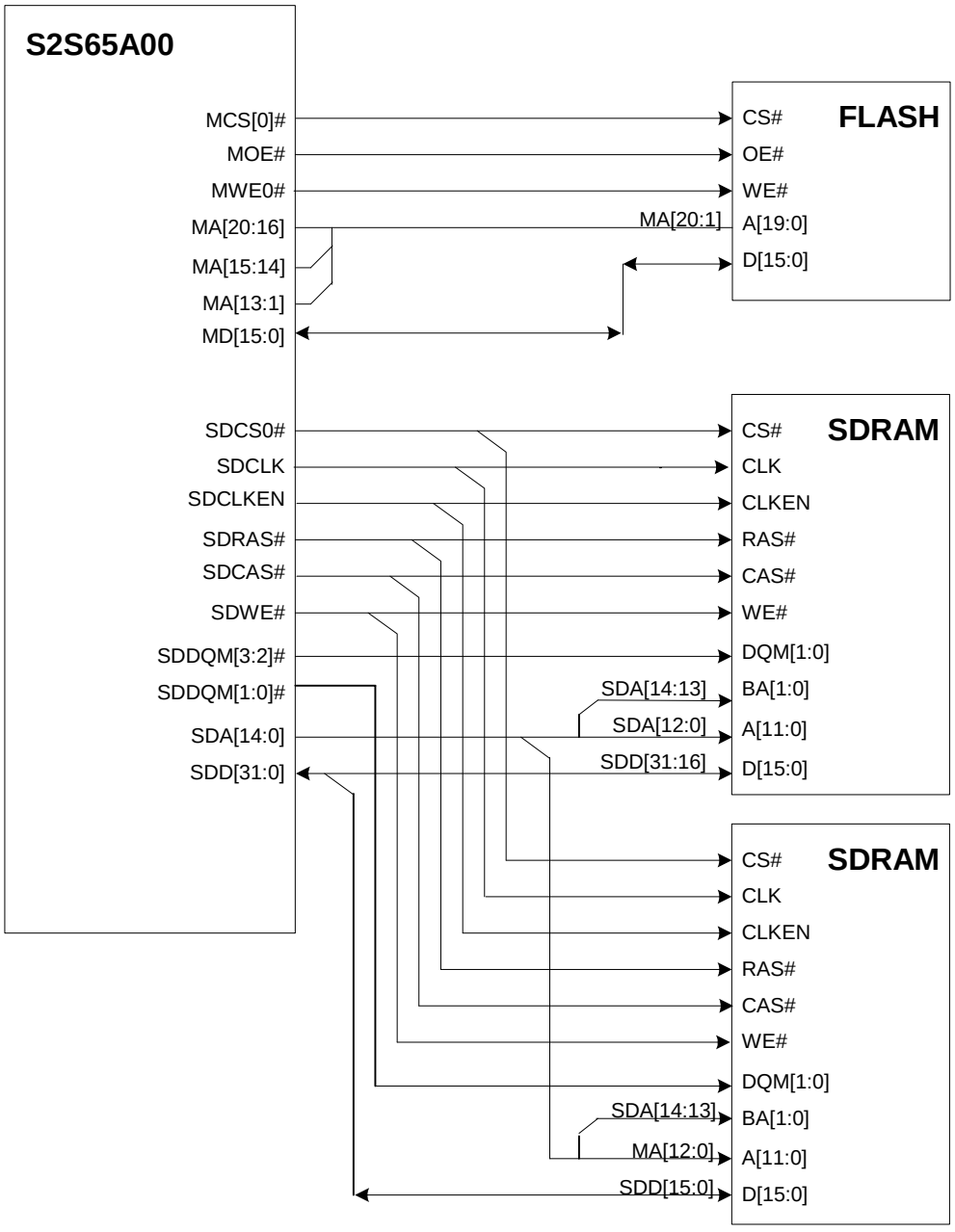


Fig.31.2 Memory Connection Example (2)

Note: Connect SDA[14:13] to the SDRAM bank address (BA[1:0]).

31.2 Compact Flash Memory Connection Example (for 16-Bit Bus Support Model)

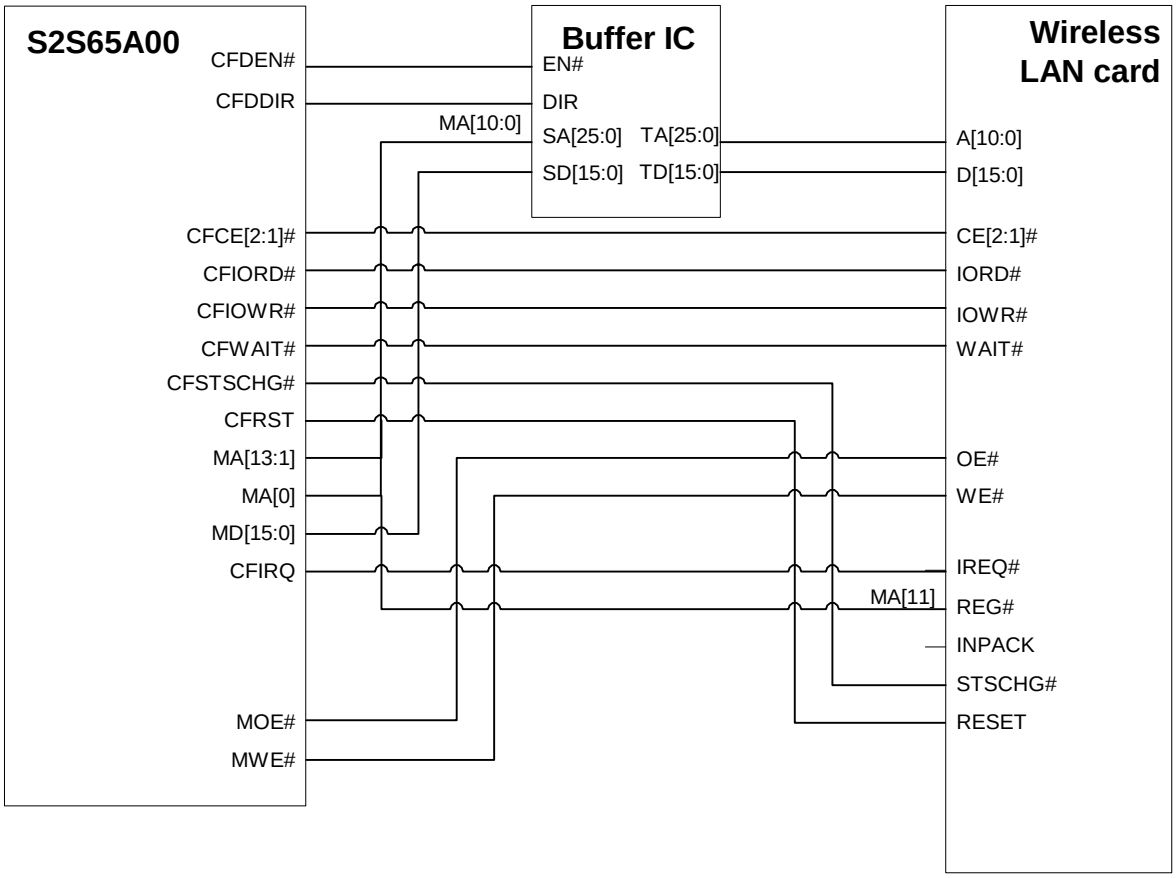
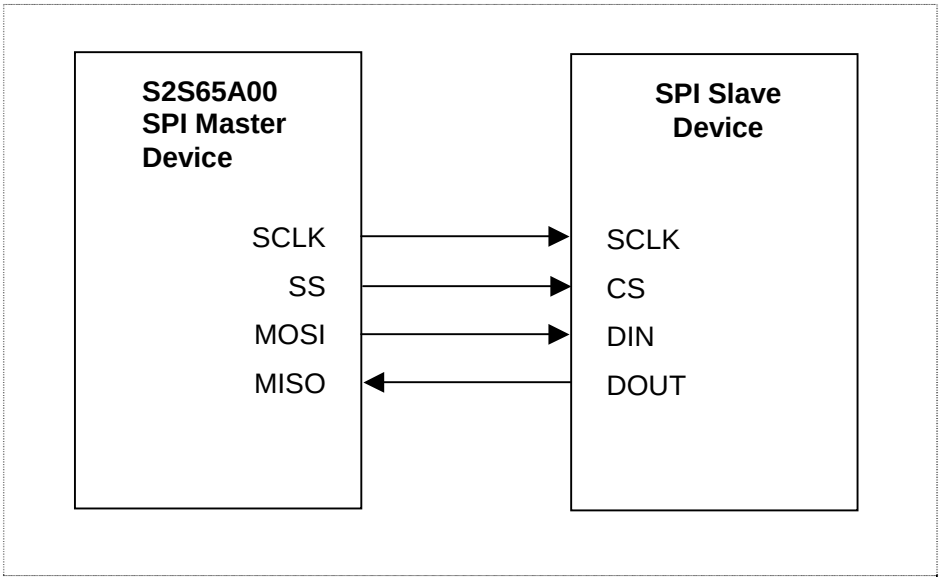


Fig.31.3 Compact Flash Memory I/F Connection Example

31.3 Serial Peripheral Device Interface (SPI) Connection Examples

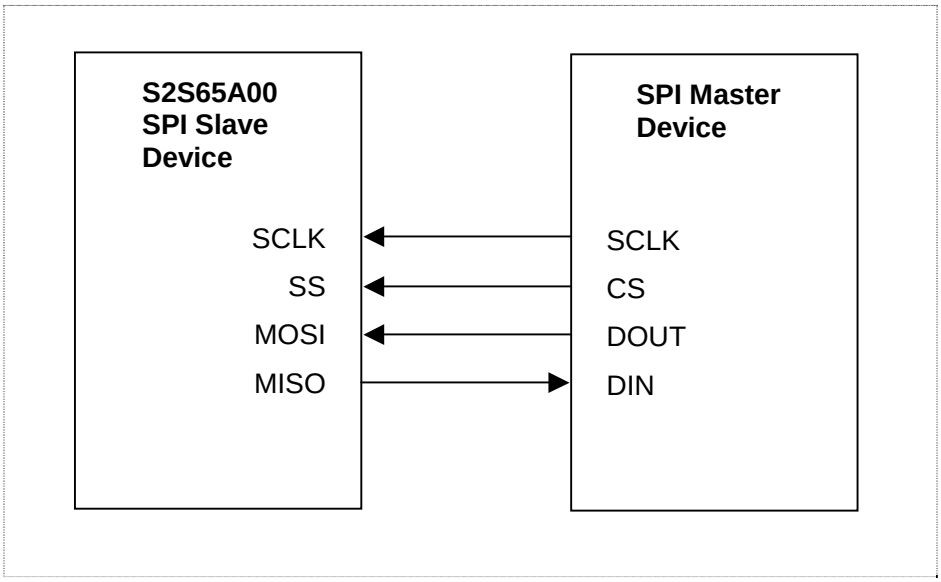
31.3.1 Master Mode

A connection example if S2S65A00 is used as master device



31.3.2 Slave Mode

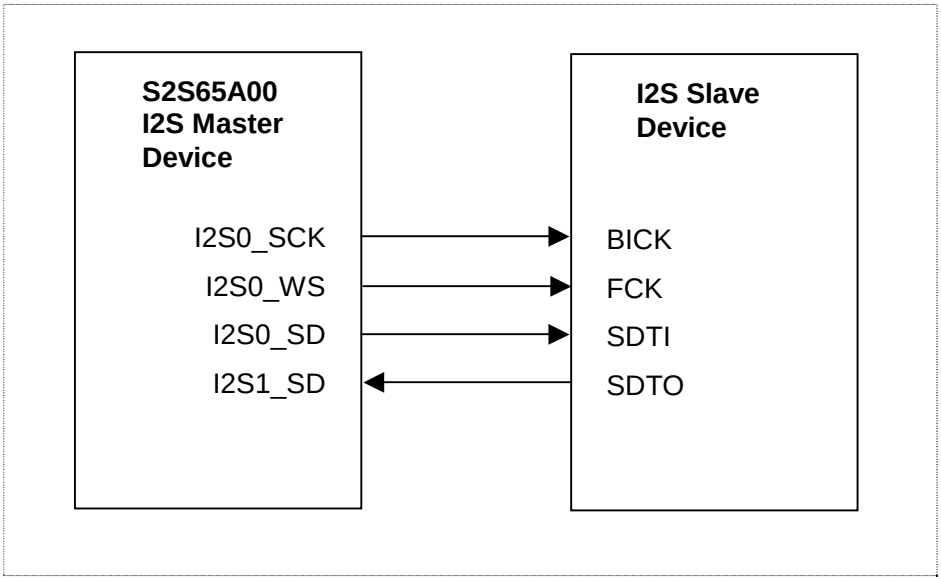
A connection example if S2S65A00 is used as slave device



31.4 I²S Connection Examples

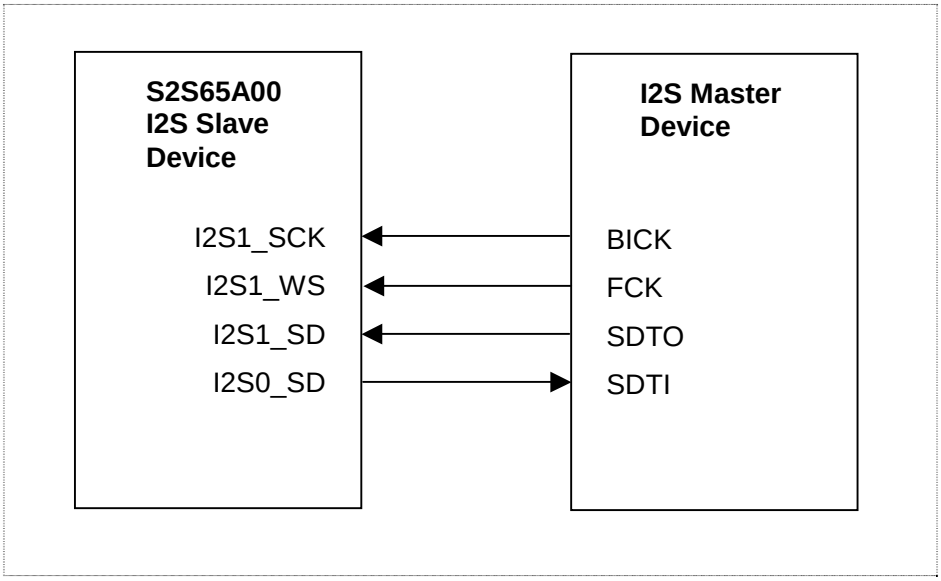
31.4.1 Master Mode

A connection example if S2S65A00 is used as master device



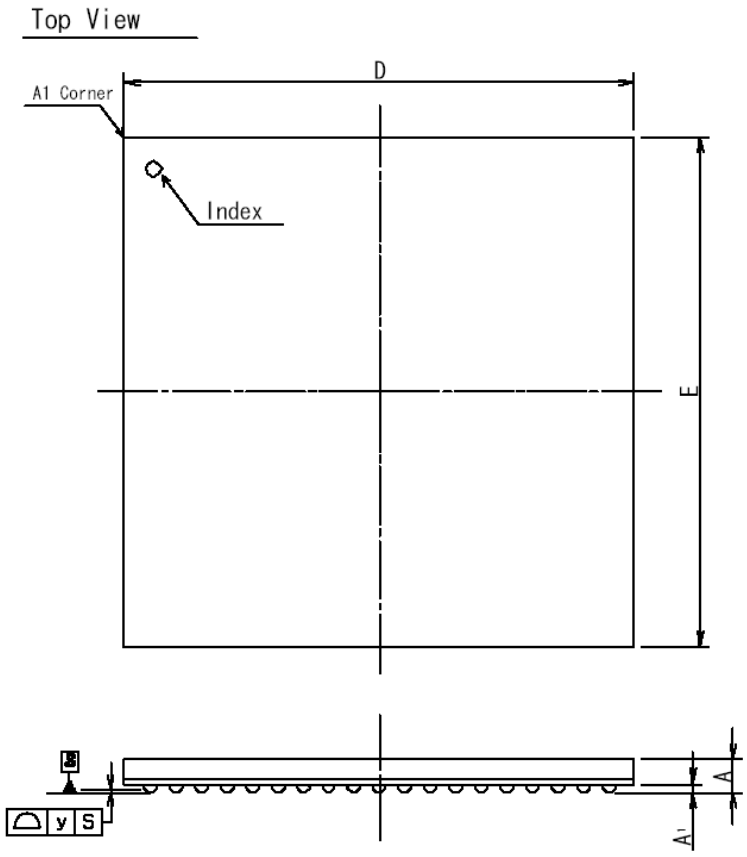
31.4.2 Slave Mode

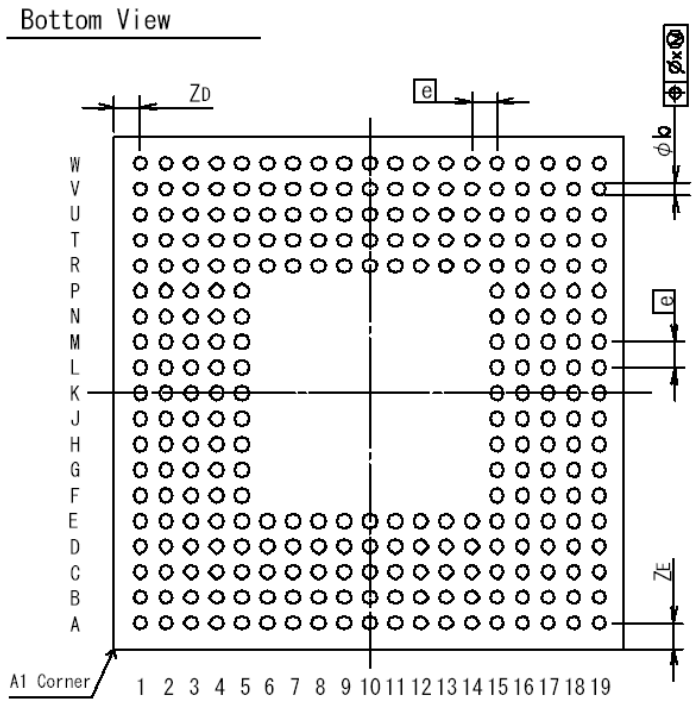
A connection example if S2S65A00 is used as slave device



32. External Dimensions

32.1 Plastic TFBGA 280pin Body size 16x16x1.2mm (PFBGA16U-280)





Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	16	—
E	—	16	—
A	—	—	1.2
A1	—	0.3	—
e	—	0.8	—
b	0.38	—	0.48
x	—	—	0.08
y	—	—	0.1
ZD	—	0.8	—
ZE	—	0.8	—

1 = 1mm

Fig.32.1 Package Dimensions (PFBGA16U-280PIN)

33. Revision History

Revision	Date of Issue	Description		
		Revision data	Before revision	After revision
1.0	2007/06/22	First edition		
1.1	2007/09/13	P16 TESTCK Description	Test clock?	Test clock This pin has a built-in pull-down resistor. Connect this pin to VSS or make it open at the time of normal operation.
		P20 3.4 Pin Status during/after Reset SDCLK Value during RESET	Low	SDCLK(32KHz)
		P175 16.4 Description of Registers	UART2:0xFFFD_5000	UART3:0xFFFD_5000
		P221 19.4 Clock and Data Transfer Timinng		Add Fig.19.5 SPI Slave mode Clock Settings
		P267 24.4.1 List of Registers		Add Table 24.1 List of Registers
		P285 Table 26.2 Port-A Pin Select Function GPAxMD[1:0] = 0b11	Reserved	Function 2 other than GPIO (Only GPIOA[7:6],[3:2] can be set.)
		P286 Table 26.3 Port-B Pin Select Function GPBxMD[1:0] = 0b10	Function 1 other than GPIO (Only GPIOB[7:1] can be set.)	Function 1 other than GPIO
		P287 Table 26.4 Port-C Pin Select Function GPCxMD[1:0] 0b11	Function 2 other than GPIO (Only GPIOC[6],[3:0] can be set.)	Function 2 other than GPIO (Only GPIOC[7:3] can be set.)
		P288 Table 26.5 Port-D Pin Select Function GPDxMD[1:0] = 0b11	Function 2 other than GPIO (Only GPIOD[7:4] can be set.)	Reserved
		P290 Table 26.7 Port-F Pin Select Function GPFxMD[1:0] = 0b11	Function 2 other than GPIO (Only GPIOE[7:4] can be set.)	Reserved
		P291 Table 26.8 Port-G Pin Select Function GPGxMD[1:0] = 0b11	Function 2 other than GPIO	Reserved
		P292 Table 26.9 Port-H Pin Select Function GPHxMD[1:0] = 0b11	Reserved	Function 2 other than GPIO
		P293 Table 26.10 Port-I Pin Select Function GPIxMD[1:0] = 0b11	Reserved	Function 2 other than GPIO (Only GPIOI[0] can be set.)
		P294 Table 26.11 Port-J Pin Select Function GPJxMD[1:0] = 0b11	Reserved	Function 1 other than GPIO

33. Revision History

Revision	Date of Issue	Description		
		Revision data	Before revision	After revision
1.1	2007/09/13	P295 Table 26.12 Port-K Pin Select Function GPKxMD[1:0] = 0b11	Reserved	Function 1 other than GPIO
		P315 30.1 DC Characteristics Current consumption	TBD	Current consumption(LVDD) Low-speed mode : 220µA High-speed mode : 150mA Current comsunption(RTCVDD) Low-speed mode BUP# = HIGH : 170µA BUP# = LOW : 2µA High-speed mode : 735µA Current consumption(PLLVDD) If PLL frequency is 50MHz : 1mA
1.2	2008/06/13	P1 1.1 Feature	Wireless LAN interface (802.11b)	Wireless LAN interface (802.11b/g)
		P2, 3 1.2 Built-in Functions Memory Controller	Supports up to two SDRAMs	Supports up to eight SDRAMs Add AD Converter: • 8-channel analog signal inputs. • 10-bits resolution AD converter. • AD conversion time is 20µsec or less.
		P4 2.BLOCK DIAGRAM	CPU Work:32KB Max.	CPU Work:56KB Max. *1 Note (*1) : Internal SRAM is shared with Line Buffer of JPG[2:1]. When JPG[2:1] is used, CPU-Work cannot be used.
		P6 3.1 Pin Assignment B19 C19 R6	 USBOSCO USBOSCI SD2	 USBCK_OSCO USBCK_OSCI SDD2
		P7 Table 3.1 Cell Type Description Cell Type	 BLNC4	 Add power supply ILS (Add) (Delete) USBDM (Add) USBDP (Add) USBVBUS (Add)

Revision	Date of Issue	Description		
		Revision data	Before revision	After revision
1.2	2008/06/13	P12,13,15,16 Table 3.2 Pin Description Pin Name:CFIRQ Pin Name:R1 Cell Type Pin Name:ADIN[7:0] Cell Type Pin Name:GPIOA2 Pin Name:GPIOA3 Pin Name:GPIOB0 Pin Name:GPIOB1 Pin Name:GPIOB2 Pin Name:GPIOB3 Pin Name:GPIOJ[7:0] Pin Name:GPIOK[7:0] Pin Name:SYSCLKI Pin Name:SYS_OSCI Pin Name:SYS_OSCO Pin Name:SYSVCP Pin Name:USBCK_OSCI Pin Name:USBCK_OSCO Pin Name:BUP#	 To use as the CFIREQ pin LLIN HLIN I2S_WS:Word select for I2S (Function 1 of other than GPIO) I2S_SCK:Serial clock for I2S (Function 1 of other than GPIO) I2S_SDO:Serial data output for I2S (Function 1 of other than GPIO) I2S_SDI:Serial data input for I2S (Function 1 of other than GPIO) BLNS4 GPIOJ[7]0] I/O BLNS4 GPIOK[7]0] I/O I 1/8 of 32KHz or system clock becomes ouput when SYSCKSEL=”LOW”. LLIN LLOT LLOT LLIN LLOT ICS	 To use as the CFIRQ pin ILTR IHTR I2S1_WS:Word select for I2S1 (Function 2 of other than GPIO) I2S1_SCLK:Serial clock for I2S1 (Function 2 of other than GPIO) I2S0_WS:Word select for I2S0 (Function 1 of other than GPIO) I2S0_SCK:Serial clock for I2S0 (Function 1 of other than GPIO) I2S0_SD:Serial data for I2S0 (Function 1 of other than GPIO) I2S1_SD:Serial data for I2S1 (Function 1 of other than GPIO) BLNC4D2 GPIOJ[7:0] I/O BLNC4D2 GPIOK[7:0] I/O I/O 1/4 of system clock or 32KHz becomes output by setting it when SYSCKSEL is “LOW”. ILTR OLTR OLTR ILTR OLTR ILS
		P18,19 3.3 Multiple Pin Function of GPIO Pins, GPIOA2 GPIOA3 GPIOB0 GPIOB1 GPIOB2 GPIOB3 CFIREQ -> CFIRQ	 I2S_WS:Function 1 I2S_SCK:Functon 1 I2S_SDO:Function 1 I2S_SDI:Function 1 CFIREQ:Function 1	 I2S1_WS:Function 2 I2S1_SCK:Function 2 I2S0_WS:Function 1 I2S0_SCK:Function 1 I2S0_SD:Function 1 I2S1_SD:Function 1 CFIRQ:Function 1
		P20 3.4 Pin Status during Reset Pin Name:MA[19:0] Pin Name:MCS[2:0] Pin Name:MBEL# Pin Name:MBEH# Pin Name:SYSCLKI Input of value during RESER	 Low High High High High Low	 Low(However, only bit11 is High.) MCS[1:0]:High MCS[3:2]:High-Z(GPIOD[5:4]) Low Low Depends on the SYSCKSEL Pin High-Z(Pull-up) High-Z(Pull-down)

33. Revision History

Revision	Date of Issue	Description		
		Revision data	Before revision	After revision
1.2	2008/06/13	P31 DMA Channel[1:0] Control Register Bits[11:0]	0100:UART#1 output(UART#0-TX) 0101:UART#1 input(UART#0-RX) 0110:UART#2 output(UART#1-TX) 0111:UART#2 input(UART#1-RX) 1000:UART#3 output(UART#2-TX) 1001:UART#3 input(UART#2-RX)	0100:UART#1 output(TX) 0101:UART#1 input(RX) 0110:UART#2 output(TX) 0111:UART#2 input(RX) 1000:UART#3 output(TX) 1001:UART#3 input(RX)
		P37,44 Camera 1/2 status register Default Value	0x0034	0x0004
		P48 8.2 Block Diagram	30KB SRAM	SRAM *1 *1 : Refer to bit[5:4](EMBRAMSEL[1:0]) of SYS[0x68](EMBMEMCTL) for the allocation of SRAM.
		P49,50 8.4 Registers Reserved Default Value R/W	0x000 R/W	- -
		P115 11.1 Description	See “S2S65A00 USB HS-Device Macro Specification” for more detail of the HS-Device Macro.	See “Appendix2 USB Device Controller” for more detail of the HS-Device Macro.
		P123 13.3 External Pins Pin Name:SYSCLKI Table 13.1 List of register	I TIMBCLSEL	I/O Add Pin Name:SYSCKSEL : I 32KHz system clock select signal Low : crystal transducer select High : crystal oscillator select TIMBCKSEL
		P129 UART1/2 Clock Divider Register	This register is used for a baud-rate TIME-based divider for UART1/2/3.	This register is used for a baud-rate TIME-based divider for UART1/2.
		P130 UART3 Clock Divider Register Bits8	SYS[30] UART1/2/3 0 : 1/1 1 : 1/2 divisionith UART3_SCLK which is a clock divided into 1/2 of the divided UART3CLKDIV. Thus, UART3_SCLK=(PCLK’s frequency)*1/N*1/2. (See the following chart.)	SYS[0x30] UART3 0 : Internal clock (Refer to Bit[7:0] for division rate.) 1 : External Clock. When you select internal clock, UART3 is supplied with UART3_SCLK which is a clock divided into 1/2 of the divided UART3CLKDIV. Thus, UART3_SCLK = (PCLK’s frequency) * 1/N * 1/2. When you select external clock, input external clock is supplied to UART3 as it is. (See the following chart.)

Revision	Date of Issue	Description		
		Revision data	Before revision	After revision
1.2	2008/06/13	P132 SDD Bus Pulldown Control Register GPIOF Register Control Register	 Each bit corresponds to each pin of MD[15:0] Bits[7:4] PRTDPDDIS[7:0]	 Each bit corresponds to each pin of SDD[15:0] Bits[7:0] PORTFPDDIS[7:0]
		P133 GPIOH Register Control Register GPIOI Register Control Register	 Pull-down Bits[7:0] Pull-down	 Pull-up Bits[1:0] Pull-up
		P134 PORTK Register Control Register	 Each bit corresponds to each pin of GPIOG[7:0]	 Each bit corresponds to each pin of GPIOK[7:0]
		P135 Embedded Memory Control Register Bits[1:0]	 11 : Read Access Wait ON, Write Access Wait ON, Write Access Wait ON	 11 : Read Access Wait ON, Read Data Wait ON, Write Access Wait ON
		P136 MISC Register Bit 1	 USB_CTL_CTL	 USB_CLK_CTL
		P137 13.5 Appendix A : PLL Setting Example	 89.1726MHz	 89.172928MHz
		P147 Table 14.2 List of external pins of memory controller	 SDD[31:0] SDDQMHH# SDDQMHL# SDDQMLH# SDDQMLL#	 SDD[31:16] : GPIOJ/K SDD[15:0] SDDQM3# : GPIOD7 SDDQM2# : GPIOD6 SDDQM1# SDDQM0#
		P150,151 14.7 Registers	 Device [3:0] timing register Device [3:0] control register	 SRAM Device [3:0] timing register SRAM Device [3:0] control register
		P153,154 SDRAM setting register Bits[23:20] COLW[5:4] COLW[7:6] Bits[7:6] Bits[3:2]	 For setting for device 2 Reserved(for setting for device 3, Be sure to set “0”.) REF[3:2] Refresh cycle BNUM[3:2] BNUM2:For setting for SDRAM devide 0 BNUM3:For setting for SDRAM device 1	 For setting for SDRAM device 0 For setting for SDRAM device 1 Reserved This bit is reserved. Be sure to set “0”. BNUM[1:0] BNUM0:For setting for SDRAM device 0 BNUM1:For setting for SDRAM device 1
		P154,155 SDRAM detailed setting register Bits[3:2]	 SELF[3:2] SELF2:For setting for SDRAM device 0 SELF3:For setting for SDRAM device 1	 SELF[1:0] SELF0:For setting for SDRAM device 0 SELF1:For setting for SDRAM device 1
		P155,156 Initialization control register Bits[3:2]	 Select the target devices with DEVSEL(MEMC[0x60] bits[3:2]) for each instruction. DEVSEL[3:2] DEVSEL2:For setting for SDRAM device 0 DEVSEL3:For setting for SDRAM device 1	 Select the target devices with DEVSEL[1:0](MEMC[0x80] bits[3:2]) for each instruction. DEVSEL[1:0] DEVSEL0:For setting for SDRAM device 0 DEVSEL1:For setting for SDRAM device 1

33. Revision History

Revision	Date of Issue	Description		
		Revision data	Before revision	After revision
1.2	2008/06/13	P157 SDRAM status register Bits[11:8]	DEVST3[3:0] Status of device 3 Shows the current state of device 3 DEVST2[3:0] Status of device 2 Shows the current state of device 2	DEVST1[3:0] Status of SDRAM device 1 Shows the current state of SDRAM device 1 DEVST0[3:0] Status of SDRAM device 0 Shows the current state of SDRAM device 0
		P158 Table 15.1 List of Interrupt Sources FIQ1	GPIOB0 Pin Ex. Battery Low(*)	GPIOB0 Pin(***) Ex Battery Low Note (***): Direct input from the GPIOB0 pin. (The default is active-low level interrupt request. However, the interrupt level on the inside becomes active-high level because it reverses internally.)
		P159 Table 15.2 List of Interrupt Sources	Note (*): Direct input from the GPIOB [7:0] pin. (The default is active-low level interrupt request.) These interrupt settings (enable, polarity, level, etc.) can be changed only by the control register of the interrupt controller. Unlike IRQ14 (**), they cannot be changed by the GPIO control register.	Note (*): Direct input from the GPIOB [7:0] pin. (The default is active-low level interrupt request. However, the interrupt level on the inside becomes active-high level because it reverses internally.) These interrupt settings (enable, polarity, level, etc.) can be changed only by the control register of the interrupt controller. Unlike IRQ14 (*), they cannot be changed by the GPIO[0x60] ~ GPIO[0x6C].
		P161 Table 15.2 List of Register 0x004 Default Value 0x024 Default Value 0x104 Default Value	 0x0000_0000 0x0000_0000 0x0000_0000	 0x0000_0000(*) 0x0000_0000(*) 0x0000_0000(*) Note (*): Initial value of IRQ/IRQxx/FIQ Unmask Status Registers are different according to the system configuration.
		P164 IRQxx Status Register	See Table 15.1	See Table 15.2
		P165 IRQxx Unmasked Status Register IRQxx[7:0] IRQxx Enable Register IRQxx[7:0]	GPIOB Unmasked Status See Table 15.1 GPIOB Enable Register See Table 15.1	IRQxx(IRQGB[7:0]) Unmasked Status See Table 15.2 IRQxx(IRQGB[7:0]) Enable Register See Table 15.2
		P166 IRQxx Enable Clear Regisgter IRQxx[7:0]	GPIOB Enable Clear See Table 15.1	IRQxx(IRQGB[7:0]) Enable Clear See Table 15.2

Revision	Date of Issue	Description		
		Revision data	Before revision	After revision
1.2	2008/06/13	P168 IRQxx Level Register IRQxx[7:0] IRQxx Polarity Register IRQxx[7:0]	 GPIOB Enable Clear See Table 15.1 Note:Normally, use the value of this regisgter that is set at reset without changing it. GPIOB Enable Clear See Table 15.1 Note:Normally, use the value of this register that is set at reset without changing it.	 IRQxx(IRQGB[7:0]) Level See Table 15.2 (delete) IRQxx(IRQGB[7:0]) Polarity See Table 15.2 (delete)
		P169 IRQxx Trigger Reset Register IRQxx[7:0]	 GPIOB Enable Clear See Table 15.1	 IRQxx(IRQGB[7:0]) Trigger Reset See Table 15.2
		P175,178 Interrupt Identify Register Default Value	 0x00	 0x01
		P207 I2S[1:0] Control Register	 Read Only	 Read/Write
		P228 20.1 Onerview	 Support of interrupt output (STSCHG#, IREQ)	 Support of interrupt output (STSCHG#, IRQ)
		P230 20.4 External Pins Pin Name	 CFIREQ	 CFIRQ
		P231,232,233 CF Card Pin Status Register Default Value CF Card IRQ Source & Clear Register Default Value	 0x00XX 0x0000	 0x0XXX 0x0XXX
		P233 CF Card IRQ Source & Clear Register	 IREQ#2 IREQ#1 IREQ	 IRQ#2 IRQ#1 CFIRQ
		P253 Table 23.1 List of Register Timer B COMINP_0/1/2/3 register R/W	 Data Access Size Address Offset R	 R/W Data Access Size RO
		P266 24.3 External Pins SYS_OSCO/SYS_OSCI Pin Functions	 Crystal oscillator connect pin	 Crystal ransducer connect pin
		P278,279 25.4 Register Watchdog Timer Control Register Default Value	 0x0000_A500	 0x0000_0000
		P282,283 26.2 External Pins GPIOA0 GPIOA1 GPIOA2 GPIOA3 GPIOA4 GPIOA5 GPIOA6 GPIOA7 GPIOH6	 TXD0 RXD0 RTS0 CTS0 TXD1 RXD1 RTS1 CTS1 CFIREQ	 TXD1 RXD1 RTS1 CTS1 TXD2 RXD2 RTS2 CTS2 CFIRQ

33. Revision History

Revision	Date of Issue	Description		
		Revision data	**Before revision**	**After revision**
1.2	2008/06/13	P293 GPIOI Data Data Register GPIOI Pin Function Register	 GPIOIDATA[7:0] GPI[7:0]MD[1:0]	 GPIOIDATA[7:2] : n/a GPIOIDATA[1:0] GPI[7:2]MD[1:0] : n/a GPI[1:0]MD[1:0]
		P298 26.4 GPIOA and GPIOB Interrupt Logic	Caution: Both GPIOI and GPIOK have the same circuit configuration. The same signal as for GPIOx port input must be connected to the “Reserved” function select pin or the port.	Note: (delete)
		P299 27.1 Description	The time of single A/D conversion is 10 µsec or less,	The time of single A/D conversion is 20 µsec or less(when system clock sets about 49MHz),
		P307 27.2 Input Voltages and Converted Data Value 1.8V Max	 0x_8C40	 0x8C_40
		P310 Table 28.1 ARS Register List ARS Status Register : Default Value	 -0x0000_0000	 0x0000_0000
		P313 29.1 Absokute Maximum Ratings 29.2 Recommended Operating Conditions	HVDD+ LVDD+ (VSS = PLLVSS = 0[V])	HVDD, UVDD3, C1VDD, C2VDD, SDVDD, AVDD LVDD, UPVDD, UXVDD, RTCVDD, PLLVDD (add) USBVBUS : -0.3 ~ 6.0 V (UVSS = PVSS = XVSS = 0 [V] VSS = PLLVSS = AVSS = 0 [V])
		P314 29.3 Power-ON Timing 29.4 Power-OFF Timing	HVDD1	HVDD
		P315		(add) Table 30.2 DC Characteristics (1.8V) Table 30.3 USBVBUS Judgment voltage
		P317,318 Table 30.6 Static Memory Timing 1 Table 30.7 Static Memory Timing 2 Data output enable signal interval period Write effective signal delay time Wirte effective signal effective period	 n2 *6 n4Ts - 0.5 *2, *7	 n1 *2, *6 n4Ts + 1 - n3Ts - 0.5 *2, *6, *7
		P327 Static Memory Read Timing Static Memory Write Timing	 tMAD tBEH	 tMAS tMBEH
		P345 Fig 31.3 Compact Flash Memory I/F Connection Example	CFIREQ	CFIRQ
		P376 A2.2 Overview A2.3 Block Diagram	 FIFO for 4.5KB endpoint FIFO RAM (4.5KB) USBOSCI USBOSCO	 FIFO for 4.0KB endpoint FIFO RAM (4.0KB) USBCK_OSCI USBCK_OSCO

Revision	Date of Issue	Description		
		Revision data	Before revision	After revision
1.2	2008/06/13	P377 Fig A2..4.2 USB External Terminal	 USBOSCI USBOSCO	 USBCK_OSCI USBCK_OSCO
1.3	2011/10/05	P29, P31, P33 DMA control register channel 0/1/2/3 initial value	0x0000_0000	0x00XX_XXXX
		P49, P69 JPEG line buffer current status register initial value	0x0009	0xX009
		P175, P182 Line statues register initial value	0x00	0x60
		P175, P184 Modem status register initial value	0x00	0xEX
		P175, P187 Test statue register 1 initial value	0x00	0x01
		P175, P187 Test statue register 2 initial value	0x00	0x0F
		P175, P187 Test statue register 3 initial value	0x00	0x02

Appendix 1 S2S65A00 Internal Register List

Address (Hex)	Register Name		Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFD_3000	**TIMB Related Register**		**TIMB**			
0x FFFD_3000	Timer-B Mode Control Register		TIMBMDC	0x0000	R/W	16(/32)
0x FFFD_3004	Timer-B Counter Control Register		TIMBCNTC	0x0000	R/W	16(/32)
0x FFFD_3008	Timer-B Status Register		TIMBSTS	0x0000	R/W	16(/32)
0x FFFD_300C	Timer-B Interrupt Enable Register		TIMBINTEN	0x0000	R/W	16(/32)
0x FFFD_3010	Timer-B Count Register		TIMBCNT	0x0000	R/W	16(/32)
0x FFFD_3014	Timer-B Common Control Register		TIMBCOMC	0x0000	R/W	16(/32)
0x FFFD_3018	Timer-B I/O Control Register		TIMBIO	0x0000	R/W	16(/32)
0x FFFD_301C	Timer-B Cycle Register		TIMBDUTY	0x0000	R/W	16(/32)
0x FFFD_3020	Timer-B COMMON_0 Register		TIMBCOM0	0x0000	R/W	16(/32)
0x FFFD_3024	Timer-B COMMON_1 Register		TIMBCOM1	0x0000	R/W	16(/32)
0x FFFD_3028	Timer-B COMMON_2 Register		TIMBCOM2	0x0000	R/W	16(/32)
0x FFFD_302C	Timer B COMMON_3 Register		TIMBCOM3	0x0000	R/W	16(/32)
0x FFFD_3030	Timer B COMINP_0 Register		TIMBINP0	0x0000	RO	16(/32)
0x FFFD_3034	Timer B COMINP_1 Register		TIMBINP1	0x0000	RO	16(/32)
0x FFFD_3038	Timer B COMINP_2 Register		TIMBINP2	0x0000	RO	16(/32)
0x FFFD_303C	Timer B COMINP_3 Register		TIMBINP3	0x0000	RO	16(/32)
0x FFFD_4000	**DLAB**	**UART2 Related Register**	**UART2**			
0x FFFD_4000	0	Receive Buffer Register	RBR	0x 00	RO	8 (/16/32)
0x FFFD_4000	0	Send Holding Register	THR	—	WO	8 (/16/32)
0x FFFD_4000	1	Divider Latch LSB Register	DLL	0x 00	R/W	8 (/16/32)
0x FFFD_4004	0	Interrupt Enable Register	IER	0x 00	R/W	8 (/16/32)
0x FFFD_4004	1	Divider Latch MSB Register	DLM	0x 00	R/W	8 (/16/32)
0x FFFD_4008	Interrupt Identification Register		IIR	0x 01	RO	8 (/16/32)
0x FFFD_4008	FIFO Control Register		FCR	—	WO	8 (/16/32)
0x FFFD_400C	Line Control Register		LCR	0x 00	R/W	8 (/16/32)
0x FFFD_4010	Modem Control Register		MCR	0x 00	R/W	8 (/16/32)
0x FFFD_4014	Line Status Register		LSR	0x 60	RO	8 (/16/32)
0x FFFD_4018	Modem Status Register		MSR	0x EX	RO	8 (/16/32)
0x FFFD_401C	Scratch Register		SCR	0x 00	R/W	8 (/16/32)
0x FFFD_4020	Test-0 Register		T0	0x 00	R/W	8 (/16/32)
0x FFFD_4024	Test-1 Register		T1	0x 00	R/W	8 (/16/32)
0x FFFD_4028	Test Status 0 Register		TS0	—	RO	8 (/16/32)
0x FFFD_402C	Test Status 1 Register		TS1	0x 01	RO	8 (/16/32)
0x FFFD_4030	Test Status 2 Register		TS2	0x 0F	RO	8 (/16/32)
0x FFFD_403C	Test Status 3 Register		TS3	0x 02	RO	8 (/16/32)
0x FFFD_5000	**DLAB**	**UART3 Related Register**	**UART3**			
0x FFFD_5000	0	Receive Buffer Register	RBR	0x 00	RO	8 (/16/32)
0x FFFD_5000	0	Send Holding Register	THR	—	WO	8 (/16/32)
0x FFFD_5000	1	Divider Latch LSB Register	DLL	0x 00	R/W	8 (/16/32)
0x FFFD_5004	0	Interrupt Enable Register	IER	0x 00	R/W	8 (/16/32)
0x FFFD_5004	1	Divider Latch MSB Register	DLM	0x 00	R/W	8 (/16/32)
0x FFFD_5008	Interrupt Identification Register		IIR	0x 01	RO	8 (/16/32)
0x FFFD_5008	FIFO Control Register		FCR	—	WO	8 (/16/32)
0x FFFD_500C	Line Control Register		LCR	0x 00	R/W	8 (/16/32)
0x FFFD_5010	Modem Control Register		MCR	0x 00	R/W	8 (/16/32)
0x FFFD_5014	Line Status Register		LSR	0x 00	RO	8 (/16/32)
0x FFFD_5018	Modem Status Register		MSR	0x 00	RO	8 (/16/32)
0x FFFD_501C	Scratch Register		SCR	0x 00	R/W	8 (/16/32)

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFD_5020	Test-0 Register	T0	0x 00	R/W	8 (/16/32)
0x FFFD_5024	Test-1 Register	T1	0x 00	R/W	8 (/16/32)
0x FFFD_5028	Test Status 0 Register	TS0	—	RO	8 (/16/32)
0x FFFD_502C	Test Status 1 Register	TS1	0x 00	RO	8 (/16/32)
0x FFFD_5030	Test Status 2 Register	TS2	0x 00	RO	8 (/16/32)
0x FFFD_503C	Test Status 3 Register	TS3	0x 00	RO	8 (/16/32)
0x FFFD_8000	**Camera-2 Interface Related Register**	**CAM2**			
0x FFFD_8000	Camera-2 Clock Cycle Setting Register		0x 0000	R/W	16
0x FFFD_8004	Camera-2 Signal Setting Register		0x 0000	R/W	16
0x FFFD_8008 to 0x FFFD_801C	Reserved		—	—	—
0x FFFD_8020	Camera-2 Mode Setting Register		0x 0000	R/W	16
0x FFFD_8024	Camera-2 Frame Control Register		0x 0000	R/W	16
0x FFFD_8028	Camera-2 Control Register		0x 0000	WO	16
0x FFFD_802C	Camera-2 Status Register		0x 0004	RO	16
0x FFFD_8030 to 0x FFFD_805C	Reserved		—	—	—
0x FFFD_9000	**JPEG2 Resize Related Register**	**RSZ2**			
0x FFFD_9060	Global Resizer Control Register		0x 0000	WO	16
0x FFFD_9064	Capture Control State Register		0x 0000	RO	16
0x FFFD_9068	Capture Data Setting Register		0x 0000	R/W	16
0x FFFD_9070 to 0x FFFD_907C	Reserved Register		—	—	—
0x FFFD_90C0	Capture Resize Control Register		0x 0000	R/W	16
0x FFFD_90C8	Capture Resize Start X-Coordinate Register		0x 0000	R/W	16
0x FFFD_90CC	Capture Resize Start Y-Coordinate Register		0x 0000	R/W	16
0x FFFD_90D0	Capture Resize End X-Coordinate Register		0x 027F	R/W	16
0x FFFD_90D4	Capture Resize End Y-Coordinate Register		0x 01DF	R/W	16
0x FFFD_90D8	Capture Resize Scaling Rate Register		0x 8080	R/W	16
0x FFFD_90DC	Capture Resize Scaling Mode Register		0x 0000	R/W	16
0x FFFD_A000	**JPEG2 Module Register**	**JCTL2**			
0x FFFD_A000	JPEG Control Register		0x 0000	R/W	16
0x FFFD_A004	JPEG Status Flag Register		0x 8080	R/W	16
0x FFFD_A008	JPEG Raw Status Flag Register		0x 8080	RO	16
0x FFFD_A00C	JPEG Interrupt Control Register		0x 0000	R/W	16
0x FFFD_A010	Reserved Register		—	—	—
0x FFFD_A014	JPEG Codec Start/Stop Control Register		0x 0000	WO	16
0x FFFD_A018 to 0x FFFD_A01C	Reserved Register		—	—	—
0x FFFD_A020	Huffman Table Auto Setting Register		0x 0000	R/W	16
0x FFFD_A040	**JPEG2 FIFO Setting Register**	**JFIFO2**			
0x FFFD_A040	JPEG FIFO Control Register		0x 0000	R/W	16
0x FFFD_A044	JPEG FIFO Status Register		0x 8001	RO	16
0x FFFD_A048	JPEG FIFO Size Register		0x 003F	R/W	16
0x FFFD_A04C	JPEG FIFO Read/Write Port Register		0x 0000_0000	R/W	32
0x FFFD_A050 to 0x FFFD_A058	Reserved Register		—	—	—
0x FFFD_A060	Encode Size Limit Register 0		0x 0000	R/W	16
0x FFFD_A064	Encode Size Limit Register 1		0x 0000	R/W	16
0x FFFD_A068	Encode Size Result Register 0		0x 0000	RO	16
0x FFFD_A06C	Encode Size Result Register 1		0x 0000	RO	16

Appendix 1 S2S65A00 Internal Register List

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFD_A070 to 0x FFFD_A078	Reserved Register		—	—	—
0x FFFD_A080	**JPEG2 Line Buffer Setting Register**	**JLB2**			
0x FFFD_A080	JPEG Line Buffer Status Flag Register		0x 0000	R/W	16
0x FFFD_A084	JPEG Line Buffer Raw Status Flag Register		0x 0000	RO	16
0x FFFD_A088	JPEG Line Buffer Current Status Flag Register		0x 0009	RO	16
0x FFFD_A08C	JPEG Line Buffer Interrupt Control Register		0x 0000	R/W	16
0x FFFD_A090 to 0x FFFD_A09C	Reserved Register		—	—	—
0x FFFD_A0A0	JPEG Line Buffer Horizontal Pixel Allowable Size Register		0x 2800	R/W	16
0x FFFD_A0A4	JPEG Line Buffer Memory Address Offset Register		0x 0020	R/W	16
0x FFFD_A0A8 to 0x FFFD_A0BC	Reserved Register		—	—	—
0x FFFD_A0C0	JPEG Line Buffer Read/Write Port Register		0x 0000	R/W	16
0x FFFD_B000	**JPEG2 Codec Register**	**JCOCEC2**			
0x FFFD_B000	Operation Mode Setting Register		0x 0000	R/W	16
0x FFFD_B004	Command Setting Register		Not applicable	WO	16
0x FFFD_B008	JPEG Operation Status Register		0x 0000	RO	16
0x FFFD_B00C	Quantization Table Number Register		0x 0000	R/W	16
0x FFFD_B010	Huffman Table Number Register		0x 0000	R/W	16
0x FFFD_B014	DRI Setting Register 0		0x 0000	R/W	16
0x FFFD_B018	DRI Setting Register 1		0x 0000	R/W	16
0x FFFD_B01C	Vertical Pixel Size Register 0		0x 0000	R/W	16
0x FFFD_B020	Vertical Pixel Size Register 1		0x 0000	R/W	16
0x FFFD_B024	Horizontal Pixel Size Register 0		0x 0000	R/W	16
0x FFFD_B028	Horizontal Pixel Size Register 1		0x 0000	R/W	16
0x FFFD_B02C to 0x FFFD_B034	Reserved Register		—	—	—
0x FFFD_B038	RST Marker Operation Setting Register		0x 0000	R/W	16
0x FFFD_B03C	RST Marker Operation Status Register		0x 0000	RO	16
0x FFFD_B040 to 0x FFFD_B0CC	Insertion Marker Data Register		0x 00FF	R/W	16
0x FFFD_B400 to 0x FFFD_B4FC	Quantization Table No. 0 Register		Not applicable	R/W	16
0x FFFD_B500 to 0x FFFD_B5FC	Quantization Table No.1 Register		Not applicable	R/W	16
0x FFFD_B800 to 0x FFFD_B83C	DC Huffman Table No. 0 Register 0		Not applicable	WO	16
0x FFFD_B840 to 0x FFFD_B86C	DC Huffman Table No. 0 Register 1		Not applicable	WO	16
0x FFFD_B880 to 0x FFFD_B8BC	AC Huffman Table No.0 Register 0		Not applicable	WO	16
0x FFFD_B8C0 to 0x FFFD_BB44	AC Huffman Table No.0 Register 1		Not applicable	WO	16
0x FFFD_BC00 to 0x FFFD_BC3C	DC Huffman Table No.1 Register 0		Not applicable	WO	16
0x FFFD_BC40 to 0x FFFD_BC6C	DC Huffman Table No.1 Register 1		Not applicable	WO	16
0x FFFD_BC80 to 0x FFFD_BCBC	AC Huffman Table No.1 Register 0		Not applicable	WO	16

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFD_BCC0 to 0x FFFD_BF44	AC Huffman Table No.1 Register 1		Not applicable	WO	16
0x FFFD_C000	**ADC Related Register**	**ADC**			
0x FFFD_C000	ADC Data Register 0	ADCDT0	0x 0000	RO	16(/32)
0x FFFD_C004	ADC Data Register 1	ADCDT1	0x 0000	RO	16(/32)
0x FFFD_C008	ADC Data Register 2	ADCDT2	0x 0000	RO	16(/32)
0x FFFD_C00C	ADC Data Register 3	ADCDT3	0x 0000	RO	16(/32)
0x FFFD_C010	ADC Data Register 4	ADCDT4	0x 0000	RO	16(/32)
0x FFFD_C014	ADC Data Register 5	ADCDT5	0x 0000	RO	16(/32)
0x FFFD_C018	ADC Data Register 6	ADCDT6	0x 0000	RO	16(/32)
0x FFFD_C01C	ADC Data Register 7	ADCDT7	0x 0000	RO	16(/32)
0x FFFD_C020	ADC Control Register	ADCCTL	0x 00	RW	8(/16/32)
0x FFFD_C024	ADC Flag Register	ADCFLG	0x 0000	RW	16(/32)
0x FFFD_D000	**SD Memory Card Control Related Register**	**SDC**			
Separate list					
0x FFFD_F000	**USB2.0 HS Device Control Related Register**	**USB**			
0x FFFD_F000	Main Interrupt Status	MainIntStat	0x 00	R/(W)	8(/16)
0x FFFD_F001	Reserved		—	—	
0x FFFD_F002	Reserved		—	—	8(/16)
0x FFFD_F003	CPU Interrupt Status	CPU_IntStat	0x 00	R/(W)	
0x FFFD_F004 to 0x FFFD_F005	Reserved		—	—	
0x FFFD_F006	Reserved		—	—	8(/16)
0x FFFD_F007	DMA Interrupt Status	DMA_IntStat	0x 00	R/(W)	
0x FFFD_F008 to 0x FFFD_F00F	Reserved		—	—	—
0x FFFD_F010	Main Interrupt Enable	MainIntEnb	0x 00	R/W	8(/16)
0x FFFD_F011	Reserved		—	—	
0x FFFD_F012	Reserved		—	—	8(/16)
0x FFFD_F013	CPU Interrupt Enable	CPU_IntEnb	0x 00	R/W	
0x FFFD_F014 to 0x FFFD_F015	Reserved		—	—	—
0x FFFD_F016	Reserved		—	—	8(/16)
0x FFFD_F017	DMA Interrupt Enable	DMA_IntEnb	0x 00	R/W	
0x FFFD_F018 to 0x FFFD_F01F	Reserved		—	—	—
0x FFFD_F020	Power Management Control 0	PM_Control0	0x 00	R/W	8(/16)
0x FFFD_F021	Power Management Control 1	PM_Control1	0x 00	R	
0x FFFD_F022	Wakeup Time Low(BE)	WakeupTim_L	0x 00	R/W	8(/16)
0x FFFD_F023	Wakeup Time High(BE)	WakeupTim_H	0x 00	R/W	
0x FFFD_F024 to 0x FFFD_F02F	Reserved		—	—	—
0x FFFD_F030	Reserved		—	—	8(/16)
0x FFFD_F031	Macro Reset	MacroReset	0x XX	W	
0x FFFD_F032	Reserved		—	—	8(/16)
0x FFFD_F033	Mode Protection	ModeProtect	0x 56	R/W	
0x FFFD_F034	Reserved		—	—	8(/16)
0x FFFD_F035	Macro Configuration 0	MacroConfig0	0x 41	R/W	
0x FFFD_F036	Reserved		—	—	8(/16)
0x FFFD_F037	Macro Configuration 1	MacroConfig1	0x 06	R/W	
0x FFFD_F038	Reserved		—	—	—
0x FFFD_F039	Reserved		—	—	—

Appendix 1 S2S65A00 Internal Register List

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFD_F03A	Macro Type 0	MacroType0	0x 44	R	8(/16)
0x FFFD_F03B	Macro Type 1	MacroType1	0x 08	R	
0x FFFD_F03C	Macro Type 2	MacroType2	0x 02	R	8(/16)
0x FFFD_F03D	Macro Type 3	MacroType3	0x 30	R	
0x FFFD_F03E	FIFO Capacity Low(BE)	FIFO_CapacityL	0x 00	R	8(/16)
0x FFFD_F03F	FIFO Capacity High(BE)	FIFO_CapacityH	0x 12	R	
0x FFFD_F040	Reserved		—	—	8(/16)
0x FFFD_F041	DMA0 Configuration	DMA0_Config	0x 00	R/W	
0x FFFD_F042	DMA0 Control	DMA0_Control	0x 00	R/W	8(/16)-
0x FFFD_F043	Reserved		—	—	
0x FFFD_F044	DMA0 FIFO Remain Low(BE)	DMA0_RemainL	0x 00	R	8(/16)
0x FFFD_F045	DMA0 FIFO Remain High(BE)	DMA0_RemainH	0x 00	R	
0x FFFD_F046 to 0x FFFD_F047	Reserved		—	—	—
0x FFFD_F048	DMA0 Transfer Byte Counter High/Low(BE)	DMA0_Count_HL	0x 00	R/W	8(/16)
0x FFFD_F049	DMA0 Transfer Byte Counter High/High(BE)	DMA0_Count_HH	0x 00	R/W	
0x FFFD_F04A	DMA0 Transfer Byte Counter Low/Low(BE)	DMA0_Count_LL	0x 00	R/W	8(/16)
0x FFFD_F04B	DMA0 Transfer Byte Counter Low/High(BE)	DMA0_Count_LH	0x 00	R/W	
0x FFFD_F04C to 0x FFFD_F04F	Reserved		—	—	—
0x FFFD_F050	Reserved		—	—	8(/16)
0x FFFD_F051	DMA1 Configuration	DMA1_Config	0x 00	R/W	
0x FFFD_F052	DMA1 Control	DMA1_Control	0x 00	R/W	8(/16)
0x FFFD_F053	Reserved		—	—	
0x FFFD_F054	DMA1 FIFO Remain Low(BE)	DMA1_RemainL	0x 00	R	8(/16)
0x FFFD_F055	DMA1 FIFO Remain High(BE)	DMA1_RemainH	0x 00	R	
0x FFFD_F056 to 0x FFFD_F057	Reserved		—	—	—
0x FFFD_F058	DMA1 Transfer Byte Counter High/Low(BE)	DMA1_Count_HL	0x 00	R/W	8(/16)
0x FFFD_F059	DMA1 Transfer Byte Counter High/High(BE)	DMA1_Count_HH	0x 00	R/W	
0x FFFD_F05A	DMA1 Transfer Byte Counter Low/Low(BE)	DMA1_Count_LL	0x 00	R/W	8(/16)
0x FFFD_F05B	DMA1 Transfer Byte Counter Low/High(BE)	DMA1_Count_LH	0x 00	R/W	
0x FFFD_F05C to 0x FFFD_F05F	Reserved		—	—	—
0x FFFD_F060 to 0x FFFD_F07F	Reserved		—	—	—
0x FFFD_F080	Device Interrupt Status	DeviceIntStat	0x 00	R/W	8(/16)
0x FFFD_F081	EPr Interrupt Status	EPrIntStat	0x 00	R	
0x FFFD_F082	SIE Interrupt Status	SIE_IntStat	0x 00	R/(W)	8(/16)
0x FFFD_F083	Reserved		—	—	
0x FFFD_F084	FIFO Interrupt Status	FIFO_IntStat	0x 00	R/(W)	8(/16)
0x FFFD_F085	Bulk Interrupt Status	BulkIntStat	0x 00	R/(W)	
0x FFFD_F086	Reserved		—	—	8(/16)
0x FFFD_F087	EP0 Interrupt Status	EP0IntStat	0x 00	R/(W)	
0x FFFD_F088	EPa Interrupt Status	EPaIntStat	0x 00	R/(W)	8(/16)
0x FFFD_F089	EPb Interrupt Status	EPbIntStat	0x 00	R/(W)	

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFD_F08A	EPc Interrupt Status	EPcIntStat	0x 00	R/(W)	8(/16)
0x FFFD_F08B	EPd Interrupt Status	EPdIntStat	0x 00	R/(W)	
0x FFFD_F08C	EPe Interrupt Status	EPeIntStat	0x 00	R/(W)	8(/16)
0x FFFD_F08D	EPf Interrupt Status	EPfIntStat	0x 00	R/(W)	
0x FFFD_F08E	EPg Interrupt Status	EPgIntStat	0x 00	R/(W)	8(1/6)
0x FFFD_F08F	EPh Interrupt Status	EPhIntStat	0x 00	R/(W)	
0x FFFD_F090	Device Interrupt Enable	DeviceIntEnb	0x 00	R/W	8(/16)
0x FFFD_F091	EPr Interrupt Enable	EprIntEnb	0x 00	R/W	
0x FFFD_F092	SIE Interrupt Enable	SIE_IntEnb	0x 00	R/W	8(/16)
0x FFFD_F093	Reserved		—	—	
0x FFFD_F094	FIFO Interrupt Enable	FIFO_IntEnb	0x 00	R/W	8(/16)
0x FFFD_F095	Bulk Interrupt Enable	BulkIntEnb	0x 00	R/W	
0x FFFD_F096	Reserved		—	—	8(/16)
0x FFFD_F097	EP0 Interrupt Enable	EP0IntEnb	0x 00	R/W	
0x FFFD_F098	EPa Interrupt Enable	EPaIntEnb	0x 00	R/W	8(/16)
0x FFFD_F099	EPb Interrupt Enable	EPbIntEnb	0x 00	R/W	
0x FFFD_F09A	EPc Interrupt Enable	EPcIntEnb	0x 00	R/W	8(/16)
0x FFFD_F09B	EPd Interrupt Enable	EPdIntEnb	0x 00	R/W	
0x FFFD_F09C	EPe Interrupt Enable	EPeIntEnb	0x 00	R/W	8(/16)
0x FFFD_F09D	EPf Interrupt Enable	EPfIntEnb	0x 00	R/W	
0x FFFD_F09E	EPg Interrupt Enable	EPgIntEnb	0x 00	R/W	8(/16)
0x FFFD_F09F	EPh Interrupt Enable	EPhIntEnb	0x 00	R/W	
0x FFFD_F0A0	Reset DTM	ResetDTM	0x 01	R/W	8(/16)
0x FFFD_F0A1	Reserved		—	—	
0x FFFD_F0A2	Nego Control	NegoControl	0x 00	R/W	8(/16)
0x FFFD_F0A3	Reserved		—	—	
0x FFFD_F0A4	USB Status	USB_Status	0x XX	R/W	8(/16)
0x FFFD_F0A5	Xcvr Control	XcvrControl	0x 41	R/W	
0x FFFD_F0A6	USB Test	USB_Test	0x 00	R/W	8(/16)
0x FFFD_F0A7	Reserved		—	—	
0x FFFD_F0A8	Endpoint Control	EPnControl	0x XX	W	8(/16)
0x FFFD_F0A9	Endpoint Clear	EPrFIFO_Clr	0x XX	W	
0x FFFD_F0AA	Clear All EPn Join	ClrAllEPnJoin	0x XX	W	8(/16)
0x FFFD_F0AB	Reserved		—	—	
0x FFFD_F0AC	BulkOnly Control	BulkOnlyControl	0x 00	R/W	8(/16)
0x FFFD_F0AD	BulkOnly Configuration	BulkOnlyConfig	0x 00	R/W	
0x FFFD_F0AE	Reserved		—	—	—
0x FFFD_F0AF	Reserved		—	—	—
0x FFFD_F0B0	EP0 SETUP 0	EP0SETUP_0	0x 00	R	8(/16)
0x FFFD_F0B1	EP0 SETUP 1	EP0SETUP_1	0x 00	R	
0x FFFD_F0B2	EP0 SETUP 2	EP0SETUP_2	0x 00	R	8(/16)
0x FFFD_F0B3	EP0 SETUP 3	EP0SETUP_3	0x 00	R	
0x FFFD_F0B4	EP0 SETUP 4	EP0SETUP_4	0x 00	R	8(/16)
0x FFFD_F0B5	EP0 SETUP 5	EP0SETUP_5	0x 00	R	
0x FFFD_F0B6	EP0 SETUP 6	EP0SETUP_6	0x 00	R	8(/16)
0x FFFD_F0B7	EP0 SETUP 7	EP0SETUP_7	0x 00	R	
0x FFFD_F0B8	USB Address	USB_Address	0x 00	R/(W)	8(/16)
0x FFFD_F0B9	Reserved		—	—	
0x FFFD_F0BA	SETUP Control	SETUP_Control	0x 00	R/W	8(/16)
0x FFFD_F0BB	Reserved		—	—	
0x FFFD_F0BC to 0x FFFD_F0BD	Reserved		—	—	—

Appendix 1 S2S65A00 Internal Register List

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFD_F0BE	FrameNumber Low	FrameNumber_L	0x 00	R	8(/16)
0x FFFD_F0BF	FrameNumber High	FrameNumber_H	0x 80	R	
0x FFFD_F0C0	EP0 Max Packet Size	EP0MaxSize	0x 40	R/W	8(/16)
0x FFFD_F0C1	EP0 Control	EP0Control	0x 00	R/W	
0x FFFD_F0C2	EP0 Control IN	EP0ControlIN	0x 00	R/W	8(/16)
0x FFFD_F0C3	EP0 Control OUT	EP0ControlOUT	0x 00	R/W	
0x FFFD_F0C4	Reserved		—	—	8(/16)
0x FFFD_F0C5	EP0 Join	EP0Join	0x 00	—	
0x FFFD_F0C6 to 0x FFFD_F0CF	Reserved		—	—	—
0x FFFD_F0D0	EPa Max Packet Size Low	EPaMaxSize_L	0x 00	R/W	8(/16)
0x FFFD_F0D1	EPa Max Packet Size High	EPaMaxSize_H	0x 00	R/W	
0x FFFD_F0D2	EPa Configuration	EPaConfig	0x 00	R/W	8(/16)
0x FFFD_F0D3	Reserved		—	—	
0x FFFD_F0D4	EPa Control	EPaControl	0x 00	R/W	8(/16)
0x FFFD_F0D5	EPa Join	EPaJoin	0x 00	R/W	
0x FFFD_F0D6	Reserved		—	—	—
0x FFFD_F0D7	Reserved		—	—	—
0x FFFD_F0D8	EPb Max Packet Size Low	EPbMaxSize_L	0x 00	R/W	8(/16)
0x FFFD_F0D9	EPb Max Packet Size High	EPbMaxSize_H	0x 00	R/W	
0x FFFD_F0DA	EPb Configuration	EPbConfig	0x 00	R/W	8(/16)
0x FFFD_F0DB	Reserved		—	—	
0x FFFD_F0DC	EPb Control	EPbControl	0x 00	R/W	8(/16)
0x FFFD_F0DD	EPb Join	EPbJoin	0x 00	R/W	
0x FFFD_F0DE	Reserved		—	—	—
0x FFFD_F0DF	Reserved		—	—	—
0x FFFD_F0E0	EPc Max Packet Size Low	EPcMaxSize_L	0x 00	R/W	8(/16)
0x FFFD_F0E1	EPc Max Packet Size High	EPcMaxSize_H	0x 00	R/W	
0x FFFD_F0E2	EPc Configuration	EPcConfig	0x 00	R/W	8(/16)
0x FFFD_F0E3	Reserved		—	—	
0x FFFD_F0E4	EPc Control	EPcControl	0x 00	R/W	8(/16)
0x FFFD_F0E5	EPc Join	EPcJoin	0x 00	R/W	
0x FFFD_F0E6	Reserved		—	—	—
0x FFFD_F0E7	Reserved		—	—	—
0x FFFD_F0E8	EPd Max Packet Size Low	EPdMaxSize_L	0x 00	R/W	8(/16)
0x FFFD_F0E9	EPd Max Packet Size High	EPdMaxSize_H	0x 00	R/W	
0x FFFD_F0EA	EPd Configuration	EPdConfig	0x 00	R/W	8(/16)
0x FFFD_F0EB	Reserved		—	—	
0x FFFD_F0EC	EPd Control	EPdControl	0x 00	R/W	8(/16)
0x FFFD_F0ED	EPd Join	EPdJoin	0x 00	R/W	
0x FFFD_F0EE	Reserved		—	—	—
0x FFFD_F0EF	Reserved		—	—	—
0x FFFD_F0F0	EPe Max Packet Size Low	EPeMaxSize_L	0x 00	R/W	8(/16)
0x FFFD_F0F1	EPe Max Packet Size High	EPeMaxSize_H	0x 00	R/W	
0x FFFD_F0F2	EPe Configuration	EPeConfig	0x 00	R/W	8(/16)
0x FFFD_F0F3	Reserved		—	—	
0x FFFD_F0F4	EPe Control	EPeControl	0x 00	R/W	8(/16)
0x FFFD_F0F5	EPe Join	EPeJoin	0x 00	R/W	
0x FFFD_F0F6	Reserved		—	—	—
0x FFFD_F0F7	Reserved		—	—	—
0x FFFD_F0F8	EPf Max Packet Size Low	EPfMaxSize_L	0x 00	R/W	8(/16)
0x FFFD_F0F9	EPf Max Packet Size High	EPfMaxSize_H	0x 00	R/W	

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFD_F0FA	EPf Configuration	EPfConfig	0x 00	R/W	8(/16)
0x FFFD_F0FB	Reserved		—	—	
0x FFFD_F0FC	EPf Control	EPfControl	0x 00	R/W	8(/16)
0x FFFD_F0FD	EPf Join	EPfJoin	0x 00	R/W	
0x FFFD_F0FE	Reserved		—	—	—
0x FFFD_F0FF	Reserved		—	—	—
0x FFFD_F100	EPg Max Packet Size Low	EPgMaxSize_L	0x 00	R/W	8(/16)
0x FFFD_F101	EPg Max Packet Size High	EPgMaxSize_H	0x 00	R/W	
0x FFFD_F102	EPg Configuration	EPgConfig	0x 00	R/W	8(/16)
0x FFFD_F103	Reserved		—	—	
0x FFFD_F104	EPg Control	EPgControl	0x 00	R/W	8(/16)
0x FFFD_F105	EPg Join	EPgJoin	0x 00	R/W	
0x FFFD_F106	Reserved		—	—	—
0x FFFD_F107	Reserved		—	—	—
0x FFFD_F108	EPh Max Packet Size Low	EPhMaxSize_L	0x 00	R/W	8(/16)
0x FFFD_F109	EPh Max Packet Size High	EPhMaxSize_H	0x 00	R/W	
0x FFFD_F10A	EPh Configuration	EPhConfig	0x 00	R/W	8(/16)
0x FFFD_F10B	Reserved		—	—	
0x FFFD_F10C	EPh Control	EPhControl	0x 00	R/W	8(/16)
0x FFFD_F10D	EPh Join	EPhJoin	0x 00	R/W	
0x FFFD_F10E	Reserved		—	—	—
0x FFFD_F10F	Reserved		—	—	—
0x FFFD_F110	Endpoint a Start Address Low	EPaStartAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F111	Endpoint a Start Address High	EPaStartAdrs_H	0x 00	R/W	
0x FFFD_F112	Endpoint b Start Address Low	EPbStartAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F113	Endpoint b Start Address High	EPbStartAdrs_H	0x 00	R/W	
0x FFFD_F114	Endpoint c Start Address Low	EPcStartAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F115	Endpoint c Start Address High	EPcStartAdrs_H	0x 00	R/W	
0x FFFD_F116	Endpoint d Start Address Low	EPdStartAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F117	Endpoint d Start Address High	EPdStartAdrs_H	0x 00	R/W	
0x FFFD_F118	Endpoint e Start Address Low	EPeStartAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F119	Endpoint e Start Address High	EPeStartAdrs_H	0x 00	R/W	
0x FFFD_F11A	Endpoint f Start Address Low	EPfStartAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F11B	Endpoint f Start Address High	EPfStartAdrs_H	0x 00	R/W	
0x FFFD_F11C	Endpoint g Start Address Low	EPgStartAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F11D	Endpoint g Start Address High	EPgStartAdrs_H	0x 00	R/W	
0x FFFD_F11E	Endpoint h Start Address Low	EPhStartAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F11F	Endpoint h Start Address High	EPhStartAdrs_H	0x 00	R/W	
0x FFFD_F120	Endpoint n End Address Low	EP_EndAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F121	Endpoint n End Address High	EP_EndAdrs_H	0x 12	R/W	
0x FFFD_F122	Reserved		—	—	—
0x FFFD_F123	Reserved		—	—	—
0x FFFD_F124	Descriptor Address Low	DescAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F125	Descriptor Address High	DescAdrs_H	0x 00	R/W	
0x FFFD_F126	Descriptor Size High	DescSize_L	0x 00	R/W	8(/16)
0x FFFD_F127	Descriptor Size Low	DescSize_H	0x 00	R/W	
0x FFFD_F128	DMA0 FIFO Control	DMA0_FIFO_ Control	0x 00	R/W	8(/16)
0x FFFD_F129	Reserved		—	—	
0x FFFD_F12A	DMA1 FIFO Control	DMA1_FIFO_ Control	0x 00	R/W	8(/16)
0x FFFD_F12B	Reserved		—	—	

Appendix 1 S2S65A00 Internal Register List

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFD_F12C to 0x FFFD_F12F	Reserved		—	—	—
0x FFFD_F130	FIFO Read 0	FIFO_Rd_0	0x XX	R	8(/16)
0x FFFD_F131	FIFO Read 1	FIFO_Rd_1	0x XX	R	
0x FFFD_F132	FIFO Write 0	FIFO_Wr_0	0x XX	W	8(/16)
0x FFFD_F133	FIFO Write 1	FIFO_Wr_1	0x XX	W	
0x FFFD_F134	FIFO Read Remain Low	FIFO_RdRemain_L	0x 00	R	8(/16)
0x FFFD_F135	FIFO Read Remain High	FIFO_RdRemain_H	0x 00	R	
0x FFFD_F136	FIFO Write Remain Low	FIFO_WrRemain_L	0x 00	R	8(/16)
0x FFFD_F137	FIFO Write Remain High	FIFO_WrRemain_H	0x 00	R	
0x FFFD_F138	FIFO Byte Read	FIFO_ByteRd	0x XX	R	8
0x FFFD_F139 to 0x FFFD_F13F	Reserved		—	—	—
0x FFFD_F140	RAM Read Address Low	RAM_RdAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F141	RAM Read Address High	RAM_RdAdrs_H	0x 00	R/W	
0x FFFD_F142	RAM Read Control	RAM_RdControl	0x 00	R/W	8(/16)
0x FFFD_F143	RAM Read Counter	RAM_RdCount	0x 00	R/W	
0x FFFD_F144	RAM Write Address Low	RAM_WrAdrs_L	0x 00	R/W	8(/16)
0x FFFD_F145	RAM Write Address High	RAM_WrAdrs_H	0x 00	R/W	
0x FFFD_F146	RAM Write Door 0	RAM_WrDoor_0	0x XX	W	8(/16)
0x FFFD_F147	RAM Write Door 1	RAM_WrDoor_1	0x XX	W	
0x FFFD_F148 to 0x FFFD_F14F	Reserved		—	—	—
0x FFFD_F150	RAM Read 00	RAM_Rd_00	0x XX	R	8(/16)
0x FFFD_F151	RAM Read 01	RAM_Rd_01	0x XX	R	
0x FFFD_F152	RAM Read 02	RAM_Rd_02	0x XX	R	8(/16)
0x FFFD_F153	RAM Read 03	RAM_Rd_03	0x XX	R	
0x FFFD_F154	RAM Read 04	RAM_Rd_04	0x XX	R	8(/16)
0x FFFD_F155	RAM Read 05	RAM_Rd_05	0x XX	R	
0x FFFD_F156	RAM Read 06	RAM_Rd_06	0x XX	R	8(/16)
0x FFFD_F157	RAM Read 07	RAM_Rd_07	0x XX	R	
0x FFFD_F158	RAM Read 08	RAM_Rd_08	0x XX	R	8(/16)
0x FFFD_F159	RAM Read 09	RAM_Rd_09	0x XX	R	
0x FFFD_F15A	RAM Read 0A	RAM_Rd_0A	0x XX	R	8(/16)
0x FFFD_F15B	RAM Read 0B	RAM_Rd_0B	0x XX	R	
0x FFFD_F15C	RAM Read 0C	RAM_Rd_0C	0x XX	R	8(/16)
0x FFFD_F15D	RAM Read 0D	RAM_Rd_0D	0x XX	R	
0x FFFD_F15E	RAM Read 0E	RAM_Rd_0E	0x XX	R	8(/16)
0x FFFD_F15F	RAM Read 0F	RAM_Rd_0F	0x XX	R	
0x FFFD_F160	RAM Read 10	RAM_Rd_10	0x XX	R	8(/16)
0x FFFD_F161	RAM Read 11	RAM_Rd_11	0x XX	R	
0x FFFD_F162	RAM Read 12	RAM_Rd_12	0x XX	R	8(/16)
0x FFFD_F163	RAM Read 13	RAM_Rd_13	0x XX	R	
0x FFFD_F164	RAM Read 14	RAM_Rd_14	0x XX	R	8(/16)
0x FFFD_F165	RAM Read 15	RAM_Rd_15	0x XX	R	
0x FFFD_F166	RAM Read 16	RAM_Rd_16	0x XX	R	8(/16)
0x FFFD_F167	RAM Read 17	RAM_Rd_17	0x XX	R	
0x FFFD_F168	RAM Read 18	RAM_Rd_18	0x XX	R	8(/16)

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFD_F169	RAM Read 19	RAM_Rd_19	0x XX	R	
0x FFFD_F16A	RAM Read 1A	RAM_Rd_1A	0x XX	R	8(/16)
0x FFFD_F16B	RAM Read 1B	RAM_Rd_1B	0x XX	R	
0x FFFD_F16C	RAM Read 1C	RAM_Rd_1C	0x XX	R	8(/16)
0x FFFD_F16D	RAM Read 1D	RAM_Rd_1D	0x XX	R	
0x FFFD_F16E	RAM Read 1E	RAM_Rd_1E	0x XX	R	8(/16)
0x FFFD_F16F	RAM Read 1F	RAM_Rd_1F	0x XX	R	
0x FFFD_F170 to 0x FFFD_F2FF	Reserved		—	—	—
0x FFFE_0000	**APB Bridge Related Register**	**APB**			
0x FFFE_0000	APB WAIT0 Register	APBWAIT0	0x 0050_0500	R/W	32
0x FFFE_0004	APB WAIT1 Register	APBWAIT1	0x 0000_0000	R/W	32
0x FFFE_0008	APB WAIT2 Register	APBWAIT2	0x 0050_0000	R/W	32
0x FFFE_3000	**DMA Controller 1 Related Register**	**DMAC1**			
0x FFFE_3000	DMA Channel-0 Source Address Register	SAR0	0x XXXX_XXXX	R/W	32
0x FFFE_3004	DMA Channel-0 Destination Address Register	DAR0	0x XXXX_XXXX	R/W	32
0x FFFE_3008	DMA Channel-0 Transfer Count Register	TCR0	0x 00XX_XXXX	R/W	32
0x FFFE_300C	DMA Channel-0 Control Register	CTL0	0x 0000_0000	R/W	32
0x FFFE_3010	DMA Channel-1 Source Address Register	SAR1	0x XXXX_XXXX	R/W	32
0x FFFE_3014	DMA Channel-1 Destination Address Register	DAR1	0x XXXX_XXXX	R/W	32
0x FFFE_3018	DMA Channel-1 Transfer Count Register	TCR1	0x 00XX_XXXX	R/W	32
0x FFFE_301C	DMA Channel-1 Control Register	CTL1	0x 0000_0000	R/W	32
0x FFFE_3020	DMA Channel-2 Source Address Register	SAR2	0x XXXX_XXXX	R/W	32
0x FFFE_3024	DMA Channel-2 Destination Address Register	DAR2	0x XXXX_XXXX	R/W	32
0x FFFE_3028	DMA Channel-2 Transfer Count Register	TCR2	0x 00XX_XXXX	R/W	32
0x FFFE_302C	DMA Channel-2 Control Register	CTL2	0x 0000_0000	R/W	32
0x FFFE_3030	DMA Channel-3 Source Address Register	SAR3	0x XXXX_XXXX	R/W	32
0x FFFE_3034	DMA Channel-3 Destination Address Register	DAR3	0x XXXX_XXXX	R/W	32
0x FFFE_3038	DMA Channel-3 Transfer Count Register	TCR3	0x 00XX_XXXX	R/W	32
0x FFFE_303C	DMA Channel-3 Control Register	CTL3	0x 0000_0000	R/W	32
0x FFFE_3060	DMA Channel Operating Select Register	OPSR	0x 0000_0000	R/W	32
0x FFFE_6000	**CF Interface Control Related Register**	**CF**			
0x FFFE_6000	CF Card Interface Control Register	CFCTL	0x 1000	(R/W)	16(/32)
0x FFFE_6004	CF Card Pin Status Register	CFPINSTS	0x 0XXX	RO	16(/32)
0x FFFE_6008	CF Card IRQ Source & Clear Register	CFINTRSTS	0x 0XXX	R/W	16(/32)
0x FFFE_600C	CF Card IRQ Enable Register	CFINTMSTS	0x 0000	R/W	16(/32)
0x FFFE_6010	CF Card IRQ Status Register	CFINTSTS	0x 0000	RO	16(/32)
0x FFFE_6014	CF Card MISC Register	CFMISC	0x 0000	R/W	16(/32)
0x FFFE_7000	**ARS Control Register**	**ARS**			
0x FFFE_7000	ARS Control Register	ARSCTRL	0x0000_0080	R/W	32
0x FFFE_7004	ARS Area Select Register	ARSASEL	0x0000_0000	R/W	32
0x FFFE_700C	ARS Status Register	ARSSTAT	0x0000_0000	R/W	32
0x FFFE_7010	ARS Interrupt Cause Register	ARSINT	0x0000_0000	R/W	32
0x FFFE_7040	ARS Accumulation Register 0	ARSADD0	0x0000_0000	RO	32

Appendix 1 S2S65A00 Internal Register List

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFE_7044	ARS Accumulation Register 1	ARSADD1	0x0000_0000	RO	32
0x FFFE_7048	ARS Accumulation Register 2	ARSADD2	0x0000_0000	RO	32
0x FFFE_704C	ARS Accumulation Register 3	ARSADD3	0x0000_0000	RO	32
0x FFFE_7050	ARS Accumulation Register 4	ARSADD4	0x0000_0000	RO	32
0x FFFE_7054	ARS Accumulation Register 5	ARSADD5	0x0000_0000	RO	32
0x FFFE_7058	ARS Accumulation Register 6	ARSADD6	0x0000_0000	RO	32
0x FFFE_705C	ARS Accumulation Register 7	ARSADD7	0x0000_0000	RO	32
0x FFFE_7060	ARS Accumulation Register 8	ARSADD8	0x0000_0000	RO	32
0x FFFE_7064	ARS Accumulation Register 9	ARSADD9	0x0000_0000	RO	32
0x FFFE_7068	ARS Accumulation Register 10	ARSADD10	0x0000_0000	RO	32
0x FFFE_706C	ARS Accumulation Register 11	ARSADD11	0x0000_0000	RO	32
0x FFFE_7070	ARS Accumulation Register 12	ARSADD12	0x0000_0000	RO	32
0x FFFE_7074	ARS Accumulation Register 13	ARSADD13	0x0000_0000	RO	32
0x FFFE_7078	ARS Accumulation Register 14	ARSADD14	0x0000_0000	RO	32
0x FFFE_707C	ARS Accumulation Register 15	ARSADD15	0x0000_0000	RO	32
0x FFFE_8000	**Camera-1 Interface Related Register**	**CAM1**			
0x FFFE_8000	Camera-1 Clock Cycle Setting Register		0x 0000	R/W	16
0x FFFE_8004	Camera-1 Signal Setting Register		0x 0000	R/W	16
0x FFFE_8008 to 0x FFFE_801C	Reserved		—	—	—
0x FFFE_8020	Camera-1 Mode Setting Register		0x 0000	R/W	16
0x FFFE_8024	Camera-1 Frame Control Register		0x 0000	R/W	16
0x FFFE_8028	Camera-1 Control Register		0x 0000	WO	16
0x FFFE_802C	Camera-1 Status Register		0x 0004	RO	16
0x FFFE_8030 to 0x FFFE_805C	Reserved		—	—	—
0x FFFE_9000	**JPEG1 Resize Related Register**	**RSZ1**			
0x FFFE_9060	Global Resizer Control Register		0x 0000	WO	16
0x FFFE_9064	Capture Control State Register		0x 0000	RO	16
0x FFFE_9068	Capture Data Setting Register		0x 0000	R/W	16
0x FFFE_9070 to 0x FFFE_907C	Reserved Register		—	—	—
0x FFFE_90C0	Capture Resize Control Register		0x 0000	R/W	16
0x FFFE_90C8	Capture Resize Start X-Coordinate Register		0x 0000	R/W	16
0x FFFE_90CC	Capture Resize Start Y-Coordinate Register		0x 0000	R/W	16
0x FFFE_90D0	Capture Resize End X-Coordinate Register		0x 027F	R/W	16
0x FFFE_90D4	Capture Resize End Y-Coordinate Register		0x 01DF	R/W	16
0x FFFE_90D8	Capture Resize Scaling Rate Register		0x 8080	R/W	16
0x FFFE_90DC	Capture Resize Scaling Mode Register		0x 0000	R/W	16
0x FFFE_A000	**JPEG1 Module Register**	**JCTL1**			
0x FFFE_A000	JPEG Control Register		0x 0000	R/W	16
0x FFFE_A004	JPEG Status Flag Register		0x 8080	R/W	16
0x FFFE_A008	JPEG Raw Status Flag Register		0x 8080	RO	16
0x FFFE_A00C	JPEG Interrupt Control Register		0x 0000	R/W	16
0x FFFE_A010	Reserved Register		—	—	—
0x FFFE_A014	JPEG Codec Start/Stop Control Register		0x 0000	WO	16
0x FFFE_A018 to 0x FFFE_A01C	Reserved Register		—	—	—
0x FFFE_A020	Huffman Table Auto Setting Register		0x 0000	R/W	16
0x FFFE_A040	**JPEG1 FIFO Setting Register**	**JFIFO1**			
0x FFFE_A040	JPEG FIFO Control Register		0x 0000	R/W	16
0x FFFE_A044	JPEG FIFO Status Register		0x 8001	RO	16

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFE_A048	JPEG FIFO Size Register		0x 003F	R/W	16
0x FFFE_A04C	JPEG FIFO Read/Write Port Register		0x 0000_0000	R/W	32
0x FFFE_A050 to 0x FFFE_A058	Reserved Register		—	—	—
0x FFFE_A060	Encode Size Limit Register 0		0x 0000	R/W	16
0x FFFE_A064	Encode Size Limit Register 1		0x 0000	R/W	16
0x FFFE_A068	Encode Size Result Register 0		0x 0000	RO	16
0x FFFE_A06C	Encode Size Result Register 1		0x 0000	RO	16
0x FFFE_A070 to 0x FFFE_A078	Reserved Register		—	—	—
0x FFFE_A080	**JPEG1 Line Buffer Setting Register**	**JLB1**			
0x FFFE_A080	JPEG Line Buffer Status Flag Register		0x 0000	R/W	16
0x FFFE_A084	JPEG Line Buffer Raw Status Flag Register		0x 0000	RO	16
0x FFFE_A088	JPEG Line Buffer Current Status Flag Register		0x 0009	RO	16
0x FFFE_A08C	JPEG Line Buffer Interrupt Control Register		0x 0000	R/W	16
0x FFFE_A090 to 0x FFFE_A09C	Reserved Register		—	—	—
0x FFFE_A0A0	JPEG Line Buffer Horizontal Pixel Allowable Size Register		0x 2800	R/W	16
0x FFFE_A0A4	JPEG Line Buffer Memory Address Offset Register		0x 0020	R/W	16
0x FFFE_A0A8 to 0x FFFE_A0BC	Reserved Register		—	—	—
0x FFFE_A0C0	JPEG Line Buffer Read/Write Port Register		0x 0000	R/W	16
0x FFFE_B000	**JPEG1 Codec Register**	**JCOCEC1**			
0x FFFE_B000	Operation Mode Setting Register		0x 0000	R/W	16
0x FFFE_B004	Command Setting Register		Not applicable	WO	16
0x FFFE_B008	JPEG Operation Status Register		0x 0000	RO	16
0x FFFE_B00C	Quantization Table Number Register		0x 0000	R/W	16
0x FFFE_B010	Huffman Table Number Register		0x 0000	R/W	16
0x FFFE_B014	DRI Setting Register 0		0x 0000	R/W	16
0x FFFE_B018	DRI Setting Register 1		0x 0000	R/W	16
0x FFFE_B01C	Vertical Pixel Size Register 0		0x 0000	R/W	16
0x FFFE_B020	Vertical Pixel Size Register 1		0x 0000	R/W	16
0x FFFE_B024	Horizontal Pixel Size Register 0		0x 0000	R/W	16
0x FFFE_B028	Horizontal Pixel Size Register 1		0x 0000	R/W	16
0x FFFE_B02C to 0x FFFE_B034	Reserved Register		—	—	—
0x FFFE_B038	RST Marker Operation Setting Register		0x 0000	R/W	16
0x FFFE_B03C	RST Marker Operation Status Register		0x 0000	RO	16
0x FFFE_B040 to 0x FFFE_B0CC	Insertion Marker Data Register		0x 00FF	R/W	16
0x FFFE_B400 to 0x FFFE_B4FC	Quantization Table No. 0 Register		Not applicable	R/W	16
0x FFFE_B500 to 0x FFFE_B5FC	Quantization Table No.1 Register		Not applicable	R/W	16
0x FFFE_B800 to 0x FFFE_B83C	DC Huffman Table No. 0 Register 0		Not applicable	WO	16
0x FFFE_B840 to 0x FFFE_B86C	DC Huffman Table No. 0 Register 1		Not applicable	WO	16
0x FFFE_B880 to 0x FFFE_B8BC	AC Huffman Table No.0 Register 0		Not applicable	WO	16

Appendix 1 S2S65A00 Internal Register List

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFE_B8C0 to 0x FFFE_BB44	AC Huffman Table No.0 Register 1		Not applicable	WO	16
0x FFFE_BC00 to 0x FFFE_BC3C	DC Huffman Table No.1 Register 0		Not applicable	WO	16
0x FFFE_BC40 to 0x FFFE_BC6C	DC Huffman Table No.1 Register 1		Not applicable	WO	16
0x FFFE_BC80 to 0x FFFE_BCBC	AC Huffman Table No.1 Register 0		Not applicable	WO	16
0x FFFE_BCC0 to 0x FFFE_BF44	AC Huffman Table No.1 Register 1		Not applicable	WO	16
0x FFFE_C000	**JPEG_DMAC Related Register**	**JDMA**			
0x FFFE_C000	DMA Channel 0 JPEG Source Address Register	JSAR0	0x XXXX_XXXX	R/W	32
0x FFFE_C004	DMA Channel 0 JPEG Destination Address Register	JDAR0	0x XXXX_XXXX	R/W	32
0x FFFE_C008	DMA Channel 0 JPEG Transfer Count Register	JTCR0	0x 0000_0000	R/W	32
0x FFFE_C00C	DMA Channel 0 JPEG Control Register	JCTL0	0x 0000_0000	R/W	32
0x FFFE_C010	DMA Channel 0 JPEG Block Count Register	JBCR0	0x 00XX_XXXX	R/W	32
0x FFFE_C014	DMA Channel 0 JPEG Destination Offset Address Register	JOFR0	0x 0000_0000	R/W	32
0x FFFE_C018	DMA Channel 0 JPEG Block End Count Register	JBER0	0x 00XX_XXXX	R/W	32
0x FFFE_C020	JPEG Source Address Register DMA Channel 1	JSAR1	0x XXXX_XXXX	R/W	32
0x FFFE_C024	DMA Channel 1 JPEG Destination Address Register	JDAR1	0x XXXX_XXXX	R/W	32
0x FFFE_C028	DMA Channel 1 JPEG Transfer Count Register	JTCR1	0x 0000_0000	R/W	32
0x FFFE_C02C	DMA Channel 1 JPEG Control Register	JCTL1	0x 0000_0000	R/W	32
0x FFFE_C030	DMA Channel 1 JPEG Block Count Register	JBCR1	0x 00XX_XXXX	R/W	32
0x FFFE_C034	DMA Channel 1 JPEG Destination Offset Address Register	JOFR1	0x 0000_0000	R/W	32
0x FFFE_C038	DMA Channel 1 JPEG Block End Count Register	JBER1	0x 00XX_XXXX	R/W	32
0x FFFE_C040	DMA Channel JPEG FIFO Data Select Mode Register	JFSM	0x 0000_0000	R/W	32
0x FFFE_C048	DMA Channel JPEG Expansion Register	JHID	0x 0000_0000	R/W	32
0x FFFE_D000	**I2C Related Register**	**I2C**			
0x FFFE_D000	I2C Send Data Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D004	I2C Receive Data Register		0000 0000 b	RO	8 (16/32)
0x FFFE_D008	I2C Control Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D00C	I2C Bus Status Register		00xx 0000 b	RO	8 (16/32)
0x FFFE_D010	I2C Error Status Register		0000 0000 b	RO	8 (16/32)
0x FFFE_D014	I2C Interrupt Control/Status Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D018	I2C-Bus Sample Clock Dividing Setting Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D01C	I2C SCL Clock Dividing Setting Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D020	I2C I/O Control Register		0001 0001 b	R/W	8 (16/32)

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFE_D024	I2C DMA Mode Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D028	I2C DMA Counter Value (LSB) Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D02C	I2C DMA Counter Value (MSB) Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D030	I2C DMA Status Register		0000 1000 b	RO	8 (16/32)
0x FFFE_D034 to 0x FFFE_D038	Reserved		—	—	—
0x FFFE_E000	**I2S Related Register**	**I2S**			
0x FFFE_E000	I2S0 Control Register		0x 0000	R/W	16(/32)
0x FFFE_E004	I2S0 Clock Dividing Register		0x 0000	R/W	16(/32)
0x FFFE_E008	I2S0 Transmission Port Register		—	R/W	8(/16/32)
0x FFFE_E010	I2S0 Interrupt Status Register		0x 0000	R/W	16(/32)
0x FFFE_E014	I2S0 Interrupt Raw Status Register		0x 0009	RO	16(/32)
0x FFFE_E018	I2S0 Interrupt Enable Register		0x 0000	R/W	16(/32)
0x FFFE_E01C	I2S0 Current Status Register		0x 0009	RO	16(/32)
0x FFFE_E040	I2S1 Control Register		0x 0000	R/W	16(/32)
0x FFFE_E044	I2S1 Clock Dividing Register		0x 0000	R/W	16(/32)
0x FFFE_E048	I2S1 Transmission Port Register		—	R/W	8(/16/32)
0x FFFE_E050	I2S1 Interrupt Status Register		0x 0000	R/W	16(/32)
0x FFFE_E054	I2S1 Interrupt Raw Status Register		0x 0009	RO	16(/32)
0x FFFE_E058	I2S1 Interrupt Enable Register		0x 0000	R/W	16(/32)
0x FFFE_E05C	I2S1 Current Status Register		0x 0009	RO	16(/32)
0x FFFF_1000	**GPIO Related Register**	**GPIO**			
0x FFFF_1000	GPIOA Data Register	GPIOA_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_1004	GPIOA Pin Function Register	GPIOA_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1008	GPIOB Data Register	GPIOB_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_100C	GPIOB Pin Function Register	GPIOB_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1010	GPIOC Data Register	GPIOC_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_1014	GPIOC Pin Function Register	GPIOC_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1018	GPIOD Data Register	GPIOD_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_101C	GPIOD Pin Function Register	GPIOD_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1020	GPIOE Data Register	GPIOE_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_1024	GPIOE Pin Function Register	GPIOE_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1028	GPIOF Data Register	GPIOF_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_102C	GPIOF Pin Function Register	GPIOF_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1030	GPIOG Data Register	GPIOG_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_1034	GPIOG Pin Function Register	GPIOG_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1038	GPIOH Data Register	GPIOH_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_103C	GPIOH Pin Function Register	GPIOH_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1040	GPIOI Data Register	GPIOI_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_1044	GPIOI Pin Function Register	GPIOI_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1048	GPIOJ Data Register	GPIOJ_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_104C	GPIOJ Pin Function Register	GPIOJ_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1050	GPIOK Data Register	GPIOK_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_1054	GPIOK Pin Function Register	GPIOK_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1060	GPIOA&B IRQ Type Register	GPIOAB_ITYP	0x 0000_0000	R/W	16 (/32)
0x FFFF_1064	GPIOA&B IRQ Polarity Register	GPIOAB_IPOL	0x 0000_0000	R/W	16 (/32)
0x FFFF_1068	GPIOA&B IRQ Enable Register	GPIOAB_IEN	0x 0000_0000	R/W	16 (/32)
0x FFFF_106C	GPIOA&B IRQ Status & Clear Register	GPIOAB_ISTS	0x 0000_0000	R/W	16 (/32)
0x FFFF_2000	**SPI Related Register**	**SPI**			
0x FFFF_2000	SPI Receive Data Register		0x 0000_0000	RO	32
0x FFFF_2004	SPI Send Data Register		0x 0000_0000	R/W	32
0x FFFF_2008	SPI Control Register 1		0x 0000_0000	R/W	32

Appendix 1 S2S65A00 Internal Register List

Address (Hex)	Register Name		Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFF_200C	SPI Control Register 2			0x 0000_0000	R/W	32
0x FFFF_2010	SPI Wait Register			0x 0000_0000	R/W	32
0x FFFF_2014	SPI Status Register			0x 0000_0010	RO	32
0x FFFF_2018	SPI Interrupt Control Register			0x 0000_0000	R/W	32
0x FFFF_5000	**DLAB**	**UART Related Register**	**UART**			
0x FFFF_5000	0	Receive Buffer Register	RBR	0x 00	RO	8 (/16/32)
0x FFFF_5000	0	Send Holding Register	THR	—	WO	8 (/16/32)
0x FFFF_5000	1	Divider Latch LSB Register	DLL	0x 00	R/W	8 (/16/32)
0x FFFF_5004	0	Interrupt Enable Register	IER	0x 00	R/W	8 (/16/32)
0x FFFF_5004	1	Divider Latch MSB Register	DLM	0x 00	R/W	8 (/16/32)
0x FFFF_5008	Interrupt Identification Register		IIR	0x 01	RO	8 (/16/32)
0x FFFF_5008	FIFO Control Register		FCR	—	WO	8 (/16/32)
0x FFFF_500C	Line Control Register		LCR	0x 00	R/W	8 (/16/32)
0x FFFF_5010	Modem Control Register		MCR	0x 00	R/W	8 (/16/32)
0x FFFF_5014	Line Status Register		LSR	0x 00	RO	8 (/16/32)
0x FFFF_5018	Modem Status Register		MSR	0x 00	RO	8 (/16/32)
0x FFFF_501C	Scratch Register		SCR	0x 00	R/W	8 (/16/32)
0x FFFF_5020	Test-0 Register		T0	0x 00	R/W	8 (/16/32)
0x FFFF_5024	Test-1 Register		T1	0x 00	R/W	8 (/16/32)
0x FFFF_5028	Test Status 0 Register		TS0	—	RO	8 (/16/32)
0x FFFF_502C	Test Status 1 Register		TS1	0x 00	RO	8 (/16/32)
0x FFFF_5030	Test Status 2 Register		TS2	0x 00	RO	8 (/16/32)
0x FFFF_503C	Test Status 3 Register		TS3	0x 00	RO	8 (/16/32)
0x FFFF_8000	**RTC Related Register**		**RTC**			
0x FFFF_8000	RTC Run/Stop Control Register			xxx- --xx b	R/W	8(/16/32)
0x FFFF_8004	RTC Interrupt Register			0x xxxx	R/W	16(/32)
0x FFFF_8008	RTC Timer Dividing Register			xxxx xxxx b	R/(W)	8(/16/32)
0x FFFF_800C	RTC Second Counter Register			--xx xxxx b	R/W	8(/16/32)
0x FFFF_8010	RTC Minute Counter Register			--xx xxxx b	R/W	8(/16/32)
0x FFFF_8014	RTC Hour Counter Register			---x xxxx b	R/W	8(/16/32)
0x FFFF_8018	RTC Day Counter Register			---x xxxx b	R/W	8(/16/32)
0x FFFF_801C	RTC Month Counter Register			---- xxxx b	R/W	8(/16/32)
0x FFFF_8020	RTC Year Counter Register			-xxx xxxx b	R/W	8(/16/32)
0x FFFF_8024	RTC Alarm Minute Compare Register			--xx xxxx b	R/W	8(/16/32)
0x FFFF_8028	RTC Alarm Month Compare Register			---x xxxx b	R/W	8(/16/32)
0x FFFF_802C	RTC Alarm Day Compare Register			---x xxxx b	R/W	8(/16/32)
0x FFFF_8030	RTC Alarm Month Compare Register			---- xxxx b	R/W	8(/16/32)
0x FFFF_8034	RTC Alarm Year Compare Register			-xxx xxxx b	R/W	8(/16/32)
0x FFFF_8040	RTC Test Register			---x xxxx b	R/W	8(/16/32)
0x FFFF_8044	RTC Prescaler Register			-xxx xxxx b	R/(W)	8(/16/32)
0x FFFF_8048	RTC Test Clock Register			xxxx xxxx b	WO	8(/16/32)
0x FFFF_8060	RTC RAM0			xxxx xxxx b	R/W	8(/16/32)
0x FFFF_8064	RTC RAM1			xxxx xxxx b	R/W	8(/16/32)
0x FFFF_8068	RTC RAM2			xxxx xxxx b	R/W	8(/16/32)
0x FFFF_806C	RTC RAM3			xxxx xxxx b	R/W	8(/16/32)
0x FFFF_8070	RTC RAM4			xxxx xxxx b	R/W	8(/16/32)
0x FFFF_8074	RTC RAM5			xxxx xxxx b	R/W	8(/16/32)
0x FFFF_8078	RTC RAM6			xxxx xxxx b	R/W	8(/16/32)
0x FFFF_807C	RTC RAM7			xxxx xxxx b	R/W	8(/16/32)
0x FFFF_9000	**DMA Controller 2 Related Register**		**DMAC2**			
0x FFFF_9000	DMA Channel-0 Source Address Register		SAR0	0x XXXX_XXXX	R/W	32

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFF_9004	DMA Channel-0 Destination Address Register	DAR0	0x XXXX_XXXX	R/W	32
0x FFFF_9008	DMA Channel-0 Transfer Count Register	TCR0	0x 00XX_XXXX	R/W	32
0x FFFF_900C	DMA Channel-0 Control Register	CTL0	0x 0000_0000	R/W	32
0x FFFF_9010	DMA Channel-1 Source Address Register	SAR1	0x XXXX_XXXX	R/W	32
0x FFFF_9014	DMA Channel-1 Destination Address Register	DAR1	0x XXXX_XXXX	R/W	32
0x FFFF_9018	DMA Channel-1 Transfer Count Register	TCR1	0x 00XX_XXXX	R/W	32
0x FFFF_901C	DMA Channel-1 Control Register	CTL1	0x 0000_0000	R/W	32
0x FFFF_9020	DMA Channel-2 Source Address Register	SAR2	0x XXXX_XXXX	R/W	32
0x FFFF_9024	DMA Channel-2 Destination Address Register	DAR2	0x XXXX_XXXX	R/W	32
0x FFFF_9028	DMA Channel-2 Transfer Count Register	TCR2	0x 00XX_XXXX	R/W	32
0x FFFF_902C	DMA Channel-2 Control Register	CTL2	0x 0000_0000	R/W	32
0x FFFF_9030	DMA Channel-3 Source Address Register	SAR3	0x XXXX_XXXX	R/W	32
0x FFFF_9034	DMA Channel-3 Destination Address Register	DAR3	0x XXXX_XXXX	R/W	32
0x FFFF_9038	DMA Channel-3 Transfer Count Register	TCR3	0x 00XX_XXXX	R/W	32
0x FFFF_903C	DMA Channel-3 Control Register	CTL3	0x 0000_0000	R/W	32
0x FFFF_9060	DMA Channel Operating Select Register	OPSR	0x 0000_0000	R/W	32
0x FFFF_9064	DMA Channel MISC Register	MISC	0x 0000_0000	R/W	32
0x FFFF_9070	DMA Channel Transfer End Control Register	TECL	0x 0000_0000	R/W	32
0x FFFF_A000	**Memory Controller Related Register**	**MEMC**			
0x FFFF_A020	SRAM Device-0 Timing Register	RAMTMG0	0x 0000_1C7F	R/W	32
0x FFFF_A024	SRAM Device-0 Control Register	RAMCNTL0	0x 0000_0001	R/W	32
0x FFFF_A030	SRAM Device-1 Timing Register	RAMTMG1	0x 0000_1C7F	R/W	32
0x FFFF_A034	SRAM Device-1 Control Register	RAMCNTL1	0x 0000_0001	R/W	32
0x FFFF_A040	SRAM Device-2 Timing Register	RAMTMG2	0x 0000_1C7F	R/W	32
0x FFFF_A044	SRAM Device-2 Control Register	RAMCNTL2	0x 0000_0001	R/W	32
0x FFFF_A050	SRAM Device-3 Timing Register	RAMTMG3	0x 0000_1C7F	R/W	32
0x FFFF_A054	SRAM Device-3 Control Register	RAMCNTL3	0x 0000_0001	R/W	32
0x FFFF_A060	SDRAM Mode Register	SDMR	0x 0000_0032	R/W	16/32
0x FFFF_A064	Reserved	—	—	-/-	—
0x FFFF_A068	Reserved	—	—	-/-	—
0x FFFF_A070	SDRAM Setting Register	SDCNFG	0x 0600_C700	R/W	32
0x FFFF_A074	SDRAM Detailed Setting Register	SDADVCNFG	0x 000F_0300	R/W	32
0x FFFF_A080	Initialization Control Register	SDINIT	0x 0000_0000	R/W	16/32
0x FFFF_A090	SDRAM Refresh Timer Register	SDREF	0x 0000_00A0	R/W	16/32
0x FFFF_A0A0	SDRAM Status Register	SDSTAT	0x 0000_0202	RO	32
0x FFFF_B000	**TimerA Related Register**	**TIM**			
0x FFFF_B000	TimerA-0 Load Register	TM0LD	0x 0000	R/W	16 (/32)
0x FFFF_B004	TimerA-0 Count Register	TM0CNT	0x 0000	RO	16 (/32)
0x FFFF_B008	TimerA-0 Control Register	TM0CTRL	0x 0000	(R/W)	16 (/32)
0x FFFF_B00C	TimerA-0 IRQ Flag Clear Register	TM0IRQ	—	WO	8 (/16/32)
0x FFFF_B010	TimerA-0 Port Output Control Register	TM0POUT	0x 0000	(R/W)	8 (/16/32)
0x FFFF_B020	TimerA-1 Load Register	TM1LD	0x 0000	R/W	16 (/32)
0x FFFF_B024	TimerA-1 Count Register	TM1CNT	0x 0000	RO	16 (/32)
0x FFFF_B028	TimerA-1 Control Register	TM1CTRL	0x 0000	(R/W)	16 (/32)
0x FFFF_B02C	TimerA-1 IRQ Flag Clear Register	TM1IRQ	—	WO	8 (/16/32)

Appendix 1 S2S65A00 Internal Register List

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFF_B030	TimerA-1 Port Output Control Register	TM1POUT	0x 0000	(R/W)	8 (/16/32)
0x FFFF_B040	TimerA-2 Load Register	TM2LD	0x 0000	R/W	16 (/32)
0x FFFF_B044	TimerA-2 Count Register	TM2CNT	0x 0000	RO	16 (/32)
0x FFFF_B048	TimerA-2 Control Register	TM2CTRL	0x 0000	(R/W)	16 (/32)
0x FFFF_B04C	TimerA-2 IRQ Flag Clear Register	TM2IRQ	—	WO	8 (/16/32)
0x FFFF_B050	TimerA-2 Port Output Control Register	TM2POUT	0x 0000	(R/W)	8 (/16/32)
0x FFFF_B060 to 0x FFFF_B09C	Reserved	—	—	—	—
0x FFFF_B0A0	Prescaler 0 Control Register	PS0CTRL	0x 0000	(R/W)	16 (/32)
0x FFFF_B0A4	Prescaler 1 Control Register	PS1CTRL	0x 0000	(R/W)	16 (/32)
0x FFFF_B0B0	TimerA IRQ Status Register	TMAIRQSTS	0x 0000	RO	8 (/16/32)
0x FFFF_C000	**WDT Related Register**	**WDT**			
0x FFFF_C000	Watchdog Timer Load Register		0x 0000_FFFF	R/W	16 (/32)
0x FFFF_C004	Watchdog Timer Count Register		0x 0000_FFFF	RO	16 (/32)
0x FFFF_C008	Watchdog Timer Control Register		0x 0000_A500	R/W	16 (/32)
0x FFFF_D000	**System Controller Related Register**	**SYS**			
0x FFFF_D000	Chip ID Register	CHIPID	0x 065A_000X	RO	32
0x FFFF_D004	Chip Configuration Register	CHIPCFG	0x 0000_XXXX	RO	16 (/32)
0x FFFF_D008	PLL Setting Register 1	PLLSET1	0x 0421_84AE	R/W	32
0x FFFF_D00C	PLL Setting Register 2	PLLSET2	0x 0000_0000	(R/W)	16 (/32)
0x FFFF_D010	HALT Mode Clock Control Register	HALTMODE	0x 0000_0000	R/W	16 (/32)
0x FFFF_D014	IO Clock Control Register	IOCLKCTL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D018	Clock Select Register	CLK_SEL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D01C	HALT Control Register	HALTCTL	—	WO	16 (/32)
0x FFFF_D020	Memory Remap Register	REMAP	0x 0000_0000	R/W	16 (/32)
0x FFFF_D024	Software Reset Register	SOFTRST	—	WO	32
0x FFFF_D028	UART1 Clock Divider Register	UART1DIV	0x 0000_0000	R/W	16 (/32)
0x FFFF_D02C	UART2 Clock Divider Register	UART2DIV	0x 0000_0000	R/W	16 (/32)
0x FFFF_D030	UART3 Clock Divider Register	UART3DIV	0x 0000_0000	R/W	16 (/32)
0x FFFF_D034	Timer-B Clock Select Register	TIMBCKSEL	0x 0000_0000	R/W	16(/32)
0x FFFF_D040	MD Bus Pull-down Control Register	MDPLDCTL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D044	SDD Bus Pull-down Control Register	SDDPLDCTL	0x 0000_0000	R/W	16(/32)
0x FFFF_D048	GPIOE Resistor Control Register	PORTERCTL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D04C	GPIOF Resistor Control Register	PORTFRCTL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D050	GPIOG Resistor Control Register	PORTGRCTL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D054	GPIOH Resistor Control Register	PORTHRCTL	0x 0000_0000	R/W	16(/32)
0x FFFF_D058	GPIOI Resistor Control Register	PORTIRCTL	0x 0000_0000	R/W	16(/32)
0x FFFF_D05C	GPIOJ Resistor Control Register	PORTJRCTL	0x 0000_0000	R/W	16(/32)
0x FFFF_D060	GPIOK Resistor Control Register	PORTKRCTL	0x 0000_0000	R/W	16(/32)
0x FFFF_D064	Internal TEST Mode Register	ITESTM	0x 0000_0000	R/W	32
0x FFFF_D068	Embedded Memory Control Register	EMBMEMCTL	0x 0000_0000	R/W	32
0x FFFF_D06C	MISC Register	MISC	0x 0000_0000	R/W	32
0x FFFF_F000	**Interrupt Controller Related Register**	**INT**			
0x FFFF_F000	IRQ Status Register		0x 0000_0000	RO	32
0x FFFF_F004	Status Register Before IRQ Mask		0x 0000_0000	RO	32
0x FFFF_F008	IRQ Enable Register		0x 0000_0000	R/W	32
0x FFFF_F00C	IRQ Enable Clear Register		0x 0000_0000	WO	32
0x FFFF_F010	Software IRQ Register		0x 0000_0000	WO	32
0x FFFF_F020	IRQxx Status Register		0x 0000_0000	RO	32
0x FFFF_F024	Status Register Before IRQxx Mask		0x 0000_0000	RO	32
0x FFFF_F028	IRQxx Enable Register		0x 0000_0000	R/W	32
0x FFFF_F02C	IRQxx Enable Clear Register		0x 0000_0000	WO	32

Address (Hex)	Register Name	Register Abbreviation	Default (Hex) *1	R/W	Data Access Size (bits) *2
0x FFFF_F080	IRQ Level Register		0x 0000_0000	R/W	32
0x FFFF_F084	IRQ Polarity Register		0x FFFF_FFFF	R/W	32
0x FFFF_F088	IRQ Trigger Reset Register		0x 0000_0000	WO	32
0x FFFF_F0A0	IRQxx Level Register		0x 0000_0000	R/W	32
0x FFFF_F0A4	IRQxx Polarity Register		0x 0000_0FFF	R/W	32
0x FFFF_F0A8	IRQxx Trigger Reset Register		0x 0000_0000	WO	32
0x FFFF_F100	FIQ Status Register		0x 0000_0000	RO	32
0x FFFF_F104	Status Register Before FIQ Mask		0x 0000_0000	RO	32
0x FFFF_F108	FIQ Enable Register		0x 0000_0000	R/W	32
0x FFFF_F10C	FIQ Enable Clear Register		0x 0000_0000	WO	32
0x FFFF_F180	FIQ Level Register		0x 0000_0000	R/W	32
0x FFFF_F184	FIQ Polarity Register		0x 0000_0003	R/W	32
0x FFFF_F188	FIQ Trigger Reset Register		0x 0000_0000	WO	32

*1 : The default value (or the initial value) is shown in hexadecimal notation and character “h” is added after the value. If character “b” is added after the value, it is a binary value.
Also, “X” represents the undefined hexadecimal value, and “x” represents the undefined binary value.

*2 : The data access size is shown in bits which is equal to the register access size. Value 8 (or 16 or 32) is usually used to access to 8-bit data. If the 16- or 32-bit data access instruction is used, the 16- or 32-bit data can also be accessed. Similarly, value 16 (or 32) is usually used to access to 16-bit data but the 32-bit data can also be accessed. In these cases, only the low-order bits must be significant.

Appendix 2 USB Device Controller

A2.1 Scope

This specification describes the functional specification of USB HS Device Controller built in S2S65A00* made by Seiko Epson.

A2.2 Overview

S2S65A00*is equipped with a USB controller compatible with the device controller of the USN controller "S1R72V05*" made by Seiko Epson.

- HS (High Speed=480Mbps) AND FS (Full Speed=12Mbps) Transfer support
- Built-in FS/HS Termination (external circuit not required).
- VBUS 5V I/F (external protection circuit required)
- Control/bulk/interrupt transfer support
- Support of 7 endpoints for control transfer (EPO), bulk transfer and bulk/interrupt transfer
- FIFO for 4.0KB endpoint
- 2ch. DMA I/F (can be combined with DMAC1 for use)
- Clock entry by 12MHz / 24MHz crystal transducer (built-in feedback resister 1MΩ)

A2.3 Block Diagram

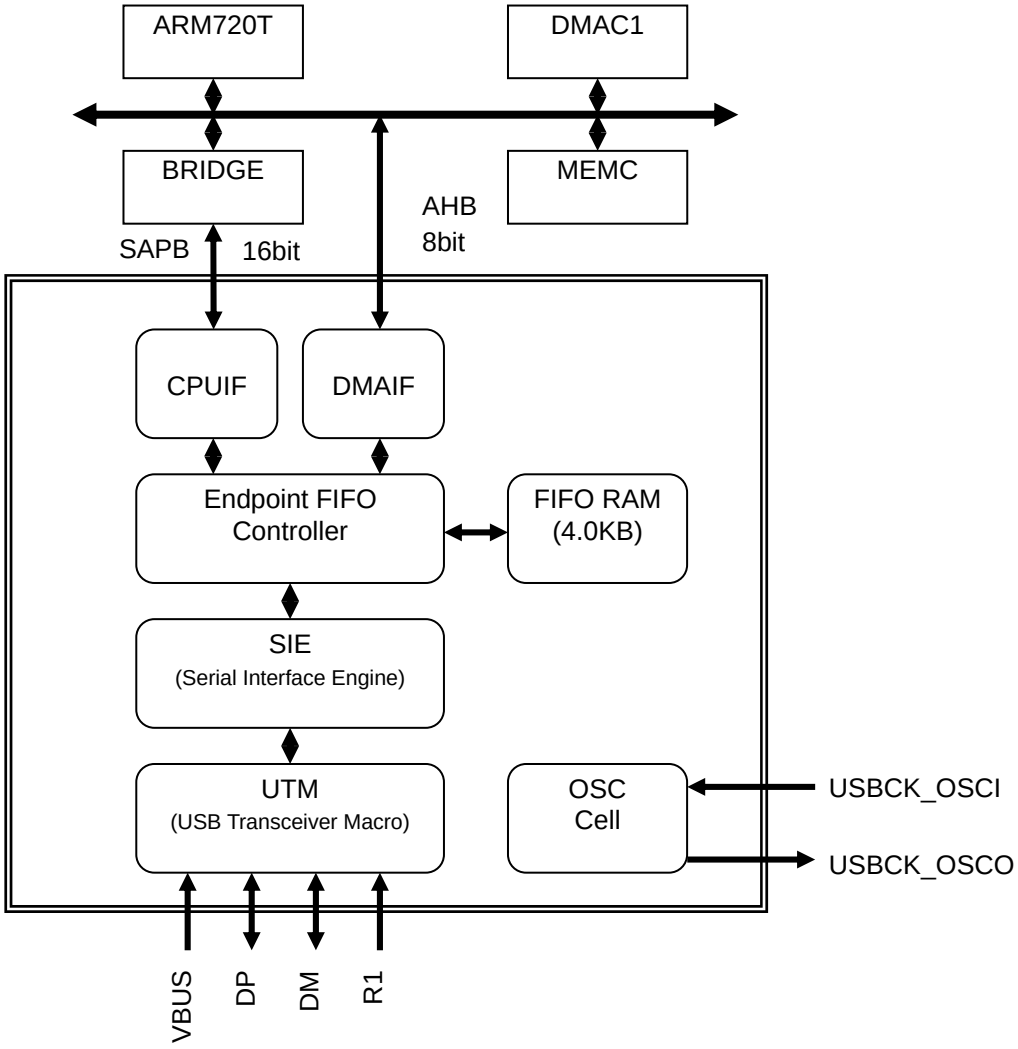


Fig.A2.3.1 S2S65A00 USB Controller Block Diagram

A2.4 Pin List

The USB interface-related external terminals are listed below.

Fig.A2.4.2 USB External Terminal

Pin Name	Input/Output	Pin Functions
VBUS	I	USB bus detection signal
DP	I/O	USB data line Data+
DM	I/O	USB data line Data−
R1	I	Internal Operation Settings 6.2KΩ±1% resistance connected between VSS's
USBCK_OSCI	I	USB internal crystal transducer input (12/24MHz)
USBCK_OSCO	O	USB internal crystal tarnsducer output

A2.5 Register

A2.5.1 Register List

The register map of control registers in nthis block is shown in TablJPEG Destination Offset Address Register. A2.5.1 - Table A2.5.4. Addresses shown in tables are those in APB bus. These registers have base address **0xFFFD_F000**.

Table A2.5.1 USB Register List (1/4)

Address Offset	Register Name	Description	R/W	Default value
Interrupt control register				
0x000	MainIntStat	Interrupt Status register	R/(W)	0x00
0x003	CPU_IntStat	CPUIF-related interrupt Status register	R/(W)	0x00
0x007	DMA_IntStat	DMA-related interrupt Status register	R/(W)	0x00
0x010	MainIntEnb	Interrupt enable register	R/W	0x00
0x013	CPU_IntEnb	CPUIF-related interrupt enable register	R/W	0x00
0x017	DMA_IntEnb	DMA-related interrupt enable register	R/W	0x00
Power Management Register				
0x020	PM_Control0	Power Management Control Register	R/W	0x00
0x021	PM_Control1	Power Management Status Register	R	0x00
0x022	WakeupTim_L	Wakeup Time setting register	R/W	0x00
0x023	WakeupTim_H		R/W	0x00
MISC Register				
0x031	MacroReset	Software reset register	R/W	0x00
0x033	ModeProtect	Setting Protect Register	R/W	0x56
0x035	MacroConfig0	Clock Setting Register	R/W	0x41
0x037	MacroConfig1	Macro Operation Mode Setting Register	R/W	0x06
0x039	(fix_CPU_Swap)		R	
0x03A	MacroType0	Macro Type Register 0	R	0x44
0x03B	MacroType1	Macro Type Register 1	R	0x08
0x03C	MacroType2	Macro Type Register 2	R	0x02
0x03D	MacroType3	Macro Type Register 3	R	0x30
0x03E	FIFO_CapacityL	FIFO RAM Capacity Register	R	0x00
0x03F	FIFO_CapacityH		R	0x12
DMA Control Register				
0x041	DMA0_Config	DMA0 Setting Register	R/W	0x00
0x042	DMA0_Control	DMA0 Control Register	R/W	0x00
0x044	DMA0_Remain_L	DMA0 FIFO Remaining Quantity Register	R	0x00
0x045	DMA0_Remain_H		R	0x00
0x048	DMA0_Count_HL	DMA0 Transfer Counter Register	R/W	0x00
0x049	DMA0_Count_HH		R/W	0x00
0x04A	DMA0_Count_LL		R/W	0x00
0x04B	DMA0_Count_LH		R/W	0x00
0x051	DMA1_Config	DMA1 Setting Register	R/W	0x00
0x052	DMA1_Control	DMA1 Control Register	R/W	0x00
0x054	DMA1_Remain_L	DMA1 FIFO Remaining Quantity Register	R	0x00
0x055	DMA1_Remain_H		R	0x00
0x058	DMA1_Count_HL	DMA1 Transfer Counter Register	R/W	0x00
0x059	DMA1_Count_HH		R/W	0x00
0x05A	DMA1_Count_LL		R/W	0x00
0x05B	DMA1_Count_LH		R/W	0x00

Table A2.5.2 USB Register List (2/4)

Address Offset	Register Name	Description	R/W	Default value
USB Control Register				
0x080	DeviceIntStat	USB Interrupt Status register	R/(W)	0x00
0x081	EPrIntStat	Endpoint Interrupt Status register	R	0x00
0x082	SIE_IntStat	SIE Interrupt Status register	R/(W)	0x00
0x084	FIFO_IntStat	FIFO Interrupt Status Register	R/(W)	0x00
0x085	BulkIntStat	Bulk Transfer Support Interrupt Status Register	R/(W)	0x00
0x087	EP0IntStat	EP0 Interrupt Status Register	R/(W)	0x00
0x088	EPaIntStat	EPa Interrupt Status Register	R/(W)	0x00
0x089	EPbIntStat	EPb Interrupt Status Register	R/(W)	0x00
0x08A	EPcIntStat	EPc Interrupt Status Register	R/(W)	0x00
0x08B	EPdIntStat	EPd Interrupt Status Register	R/(W)	0x00
0x08C	EPeIntStat	EPe Interrupt Status Register	R/(W)	0x00
0x08D	EPfIntStat	EPf Interrupt Status Register	R/(W)	0x00
0x08E	EPgIntStat	EPg Interrupt Status Register	R/(W)	0x00
0x08F	EPhIntStat	EPh Interrupt Status Register	R/(W)	0x00
0x090	DeviceIntEnb	USB Interrupt Control Register	R/W	0x00
0x091	EPrIntEnb	Endpoint Interrupt Control Register	R/W	0x00
0x092	SIE_IntEnb	SIE Interrupt Control Register	R/W	0x00
0x094	FIFO_IntEnb	FIFO Interrupt Control Register	R/W	0x00
0x095	BulkIntEnb	Bulk Transfer Support Interrupt Control Register	R/W	0x00
0x097	EP0IntEnb	EP0 Interrupt Control Register	R/W	0x00
0x098	EPaIntEnb	EPa Interrupt Control Register	R/W	0x00
0x099	EPbIntEnb	EPb Interrupt Control Register	R/W	0x00
0x09A	EPcIntEnb	EPc Interrupt Control Register	R/W	0x00
0x09B	EPdIntEnb	EPd Interrupt Control Register	R/W	0x00
0x09C	EPeIntEnb	EPe Interrupt Control Register	R/W	0x00
0x09D	EPfIntEnb	EPf Interrupt Control Register	R/W	0x00
0x09E	EPgIntEnb	EPg Interrupt Control Register	R/W	0x00
0x09F	EPhIntEnb	EPh Interrupt Control Register	R/W	0x00
0x0A0	ResetDTM	Transceiver macro Reset Register	R/W	0x01
0x0A2	NegoControl	Negotiation Control Register	R/W	0x00
0x0A4	USB_Status	USB Bus Status Register	R/W	0xXX
0x0A5	XcvrControl	Transceiver Macro Control Register	R/W	0x41
0x0A6	USB_Test	USB Test Register	R/W	0x00
0x0A8	EPnControl	EP Common control Register	W	0xXX
0x0A9	EPrFIFO_Clr	FIFO Clear Register	W	0xXX
0x0AA	ClrAllEPnJoin	EP Join Clear register	W	0xXX
0x0AC	BulkOnlyControl	Bulk Transfer Support Control Register	R/W	0x00
0x0AD	BulkOnlyConfig	Bulk Transfer Support Setting Register	R/W	0x00
0x0B0	EP0SETUP_0	EP0 Setup Data Register	R	0x00
0x0B1	EP0SETUP_1		R	0x00
0x0B2	EP0SETUP_2		R	0x00
0x0B3	EP0SETUP_3		R	0x00
0x0B4	EP0SETUP_4		R	0x00
0x0B5	EP0SETUP_5		R	0x00
0x0B6	EP0SETUP_6		R	0x00
0x0B7	EP0SETUP_7		R	0x00
0x0B8	USB_Address	USB Address Register	R/(W)	0x00
0x0BA	SETUP_Control	Setup Stage control Register	R/W	0x00
0x0BE	FrameNumber_L	Frame Number Register	R	0x00
0x0BF	FrameNumber_H		R	0x80

Table A2.5.3 USB Register List (3/4)

Address Offset	Register Name	Description	R/W	Default value
USB Control Register				
0x0C0	EP0MaxSize	EP0 Max Packet Size Setting register	R/W	0x40
0x0C1	EP0Control	EP0 Control Register	R/W	0x00
0x0C2	EP0ControlIN	EP0 IN Transfer Control Register	R/W	0x00
0x0C3	EP0ControlOUT	EP0 OUT Transfer Control Register	R/W	0x00
0x0C5	EP0JOin	EP0 Join Setting Register	R/W	0x00
0x0D0	EPaMaxSize_L	EPa Max Packet Size Setting Register	R/W	0x00
0x0D1	EPaMaxSize_H		R/W	0x00
0x0D2	EPaConfig	EPa Setting Register	R/W	0x00
0x0D4	EPaControl	EPa Control Register	R/W	0x00
0x0D5	EPaJoin	EPa Join Setting Register	R/W	0x00
0x0D8	EPbMaxSize_L	EPa Max Packet Size Setting Register	R/W	0x00
0x0D9	EPbMaxSize_H		R/W	0x00
0x0DA	EPbConfig	EPb Setting Register	R/W	0x00
0x0DC	EPbControl	EPb Control Register	R/W	0x00
0x0DD	EPbJoin	EPb Join Setting Register	R/W	0x00
0x0E0	EPcMaxSize_L	EPc Max Packet Size Setting Register	R/W	0x00
0x0E1	EPcMaxSize_H		R/W	0x00
0x0E2	EPcConfig	EPc Setting Register	R/W	0x00
0x0E4	EPcControl	EPc Control Register	R/W	0x00
0x0E5	EPcJoin	EPc Join Setting Register	R/W	0x00
0x0E8	EPdMaxSize_L	EPd Max Packet Size Setting Register	R/W	0x00
0x0E9	EPdMaxSize_H		R/W	0x00
0x0EA	EPdConfig	EPd Setting Register	R/W	0x00
0x0EC	EPdControl	EPd Control Register	R/W	0x00
0x0ED	EPdJoin	EPd Join Setting Register	R/W	0x00
0x0F0	EPeMaxSize_L	EPe Max Packet Size Setting Register	R/W	0x00
0x0F1	EPeMaxSize_H		R/W	0x00
0x0F2	EPeConfig	EPe Setting Register	R/W	0x00
0x0F4	EPeControl	EPe Control Register	R/W	0x00
0x0F5	EPeJoin	EPe Join Setting Register	R/W	0x00
0x0F8	EPfMaxSize_L	EPf Max Packet Size Setting Register	R/W	0x00
0x0F9	EPfMaxSize_H		R/W	0x00
0x0FA	EPfConfig	EPf Setting Register	R/W	0x00
0x0FC	EPfControl	EPf Control Register	R/W	0x00
0x0FD	EPfJoin	EPf Join Setting Register	R/W	0x00
0x100	EPgMaxSize_L	EPg Max Packet Size Setting Register	R/W	0x00
0x101	EPgMaxSize_H		R/W	0x00
0x102	EPgConfig	EPg Setting Register	R/W	0x00
0x104	EPgControl	EPg Control Register	R/W	0x00
0x105	EPgJoin	EPg Join Setting Register	R/W	0x00
0x108	EPhMaxSize_L	EPh Max Packet Size Setting Register	R/W	0x00
0x109	EPhMaxSize_H		R/W	0x00
0x10A	EPhConfig	EPh Setting Register	R/W	0x00
0x10C	EPhControl	EPh Control Register	R/W	0x00
0x10D	EPhJoin	EPh Join Setting Register	R/W	0x00

Table A2.5.4 USB Register List (4/4)

Address Offset	Register Name	Description	R/W	Default value
USB FIFO Setting Register				
0x110	EPaStartAdrs_L	EPa FIFO Start Address Setting Register	R/W	0x00
0x111	EPaStartAdrs_H		R/W	0x00
0x112	EPbStartAdrs_L	EPb FIFO Start Address Setting Register	R/W	0x00
0x113	EPbStartAdrs_H		R/W	0x00
0x114	EPcStartAdrs_L	EPc FIFO Start Address Setting Register	R/W	0x00
0x115	EPcStartAdrs_H		R/W	0x00
0x116	EPdStartAdrs_L	EPd FIFO Start Address Setting Register	R/W	0x00
0x117	EPdStartAdrs_H		R/W	0x00
0x118	EPeStartAdrs_L	EPe FIFO Start Address Setting Register	R/W	0x00
0x119	EPeStartAdrs_H		R/W	0x00
0x11A	EPfStartAdrs_L	EPf FIFO Start Address Setting Register	R/W	0x00
0x11B	EPfStartAdrs_H		R/W	0x00
0x11C	EPgStartAdrs_L	EPg FIFO Start Address Setting Register	R/W	0x00
0x11D	EPgStartAdrs_H		R/W	0x00
0x11E	EPhStartAdrs_L	EPh FIFO Start Address Setting Register	R/W	0x00
0x11F	EPhStartAdrs_H		R/W	0x00
0x120	EP_EndAdrs_L	EP FIFO End Address Setting Register	R/W	0x00
0x121	EP_EndAdrs_H		R/W	0x12
0x124	DescAdrs_L	Descriptor Initial Address Setting Register	R/W	0x00
0x125	DescAdrs_H		R/W	0x00
0x126	DescSize_L	Descriptor Size Setting Register	R/W	0x00
0x127	DescSize_H		R/W	0x00
0x128	DMA0_FIFO_Control	DMA0 FIFO Control Register	R/W	0x00
0x12A	DMA1_FIFO_Control	DMA1 FIFO Control Register	R/W	0x00
0x130	FIFO_Rd_0	FIFO Read Register	R	0xXX
0x131	FIFO_Rd_1		R	0xXX
0x132	FIFO_Wr_0	FIFO Write Register	W	0xXX
0x133	FIFO_Wr_1		W	0xXX
0x134	FIFO_RdRemain_L	FIFO Read Remaining Quantity Register	R	0x00
0x135	FIFO_RdRemain_H		R	0x00
0x136	FIFO_WrRemain_L	FIFO Write Remaining Quantity Register	R	0x00
0x137	FIFO_WrRemain_H		R	0x00
0x138	FIFO_ByteRd	FIFO Byte Read Register	R	0xXX
0x140	RAM_RdAdrs_L	RAM Read Address Setting Register	R/W	0x00
0x141	RAM_RdAdrs_H		R/W	0x00
0x142	RAM_RdControl	RAM Read Control Register	R/W	0x00
0x143	RAM_RdCount	RAM Read Count Setting Register	R/W	0x00
0x144	RAM_WrAdrs_L	RAM Write Address Setting Register	R/W	0x00
0x145	RAM_WrAdrs_H		R/W	0x00
0x146	RAM_WrDoor_0	RAM Write Register	W	0xXX
0x147	RAM_WrDoor_1		W	0xXX
0x150 to 0x16F	RAM_Rd_00 to RAM_Rd_1F	RAM Read Data Register	R	0xXX
Test Register (Reserved)				
0x170 to 0x17F	Reserved	Test Register	—	

A2.5.2 Register Detail Description

A2.5.2.1 Notes on Register Access

(1) Do not write anything for the address to which no register is allocated.
(2) Write only “0x0” for the bits to which no register is allocated.
(3) Read/write enabled registers even in SLEEP and SNOOZE are shown in bold italic.
(4) Other registers can be read and written in ACTIVE60 / ACT_DEVICE.

A2.5.2.2 Interrupt control register

Main Interrupt Status
USB[0x000] Default value = 0x00

Device IntStat(R)	—	CPU_ IntStat(R)	—	—	DMA_ IntStat(R)	—	***Finished PM***
7	6	5	4	3	2	1	0

Interrupt factors of USB macros are shown.

This register has a bit to indirectly instruct an interrupt factor and a bit to directly instruct it. The bit that indirectly instructs an interrupt factor allows you to the bit that directly instructs the interrupt factor by reading the corresponding interrupt status register. The bit that indirectly instructs an interrupt factor is read-only and is automatically cleared by clearing the bit that directly instructs the original interrupt factor. The bit that directly instructs an interrupt factor is writable and you can clear the interrupt factor by writing “1” in the corresponding bit. When the interrupt of the corresponding bit is enabled by the MainInterruptEnb register and the interrupt factor is set to “1”, an interrupt occurs against INTC.

Bit 7: **DeviceIntStat**
Indirectly instruct an interrupt factor.
This is set to “1” when the Device_IntStat register has an interrupt factor and the bit of the SIE_DeviceIntEnb register corresponding to the interrupt factor is enabled. This bit is valid for reading even during SLEEP / SNOOZE.

Bit 6: **Reserved**

Bit 5: **CPU_IntStat**
Indirectly instruct an interrupt factor.
This is set to “1” when the CPU_IntStat register has an interrupt factor and the bit of the CPU_IntEnb register corresponding to the interrupt factor is enabled.

Bit [4:3]: **Reserved**

Bit 2: **DMAIntStat**
Indirectly instruct an interrupt factor.
This is set to “1” when the DMAIntStat register has an interrupt factor and the bit of the DMAIntEnb register corresponding to the interrupt factor is enabled.

Bit 1: **Reserved**

Bit 0: **FinishedPM**
Directly instruct an interrupt factor.
This bit is set to “1” when GoSLEEP, GoSNOOZE, GoActive60 and GoActDevice are set by the PM_Control register and have reached each status instructed. This bit is valid even during SLEEP / SNOOZE.

CPU Interrupt Status
USB[0x003] Default value = 0x00

7	6	5	4	3	2	1	0
RAM_RdCmp	—	—	—	—	—	—	—

Display interrupts related to a CPU interface.
You can clear interrupt factors by writing "1" for all bits.

Bit 7: **RAM_RdCmp**
Directly instruct an interrupt factor.
This is set to "1" when the RAM_Rd function reads data from RAM and enabled the RAM_Rd_XX data.

Bit [6:0]: **Reserved**

DMA Interrupt Status
USB[0x007] Default value = 0x00

7	6	5	4	3	2	1	0
—	—	—	—	DMA1_CountUp	DMA1_Cmp	DMA0_CountUp	DMA0_Cmp

Display interrupts related to DMA.
You can clear interrupt factors by writing "1" for all bits.

Bit [7:4]: **Reserved**

Bit 3: **DMA1_CountUp**
Directly instruct an interrupt factor.
This is set to "1" when the transfer mode is in free-run mode and DMA1_Count_HH, HL, LH and LL values overflowed. DMA1_Count_HH, HL, LH and LL values are cleared and the DMA operation continues.

Bit 2: **DMA1_Cmp**
Directly instruct an interrupt factor.
This is set to "1" when DMA transfer is stopped or when data has been transferred for the specified times and the termination process is completed.

Bit 1: **DMA0_CountUp**
Directly instruct an interrupt factor.
This is set to "1" when the transfer mode is in free-run mode and DMA0_Count_HH, HL, LH and LL values overflowed. DMA0_Count_HH, HL, LH and LL values are cleared and the DMA operation continues.

Bit 0: **DMA0_Cmp**
Directly instruct an interrupt factor.
This is set to "1" when DMA transfer is stopped or when data has been transferred for the specified times and the termination process is completed.

Main Interrupt Enable
USB[0x010] Default value = 0x00

EnDevice IntStat	—	EnCPU IntStat	—	—	EnDMA_ IntStat	—	***EnFinished PM***
7	6	5	4	3	2	1	0

This is the register to permit/prohibit asserting interrupt signals for the INTC module due to interrupt factors of the MainIntStat register. Setting the corresponding bit to "1" permits the interrupt.
EnDeviceIntStat and EnFinishedPM bits are valid even during SLEEP / SNOOZE.

CPU Interrupt Enable
USB[0x013] Default value = 0x00

EnRAM_ RdCmp	—	—	—	—	—	—	—
7	6	5	4	3	2	1	0

This permits/prohibits asserting the CPU_IntStat bit of the MainIntStat register due to interrupt factors of the CPU_IntStat register.

DMA Interrupt Enable
USB[0x017] Default value = 0x00

—	—	—	—	EnDMA1_ CountUp	EnDMA1_ Cmp	EnDMA0_ CountUp	EnDMA0_ Cmp
7	6	5	4	3	2	1	0

This permits/prohibits asserting the DMAIntStat bit of the MainIntStat register due to interrupt factors of the DMAIntStat register.

A2.5.2.3 Power Management Control Register

Power Management Control 0							
USB[0x020] Default value = 0x00							
Go SLEEP	***Go SNOOZE***	***Go Active60***	***Go ActDevice***	—	—	—	—
7	6	5	4	3	2	1	0

Set operations related to power management of this LSI.
This register is valid even during SLEEP / SNOOZE.

Bit 7: **GoSLEEP**

Start transition to the SLEEP state from other states.
Setting this bit to "1" in the SNOOZE state stops oscillation of the oscillator and makes the transition to the SLEEP state.
Setting this bit to "1" in the ACTIVE60 state stops oscillation of PLL60 and then the oscillator and makes the transition to the SLEEP state.
Setting this bit to "1" in the ACT_DEVICE state stops oscillation of DevicePLL480 first and then PLl60 and the oscillator and makes the transition to the SLEEP state.
This bit is automatically cleared as soon as the transition has been made from any state and the MainIntStat.FinishedPM bit is set at the same time.

Bit 6: **GoSNOOZE**

Start transition to the SNOOZE state from other states.
Setting this bit to "1" in the SLEEP state starts oscillation of the oscillator and makes the transition to the SNOOZE state after the oscillation stable time (the time set for WakeupTim_H and L) has elapsed.
Setting this bit to "1" in the ACTIVE60 state stops oscillation of PLL60 and makes the transition to the SNOOZE state.
Setting this bit to "1" in the ACT_DEVICE state stops oscillation of DevicePLL480 and then PLL60 and makes the transition to the SNOOZE state.
This bit is automatically cleared as soon as the transition has been made from any state and the MainIntStat.FinishedPM bit is set at the same time.

Bit 5: **GoActive60**

Start transition to the ACTIVE60 state from other states.
Setting this bit to "1" in the SLEEP state starts oscillation of the oscillator and PLL60 after the oscillation stable time (the time set for WakeupTim_H and L) has elapsed and makes the transition to the ACTIVE60 state after the PLL60 oscillation stable time (about 250us) has elapsed.
Setting this bit to "1" in the SNOOZE state starts oscillation of PLL60 and makes the transition to the ACTIVE60 state after the PLL oscillation stable time (about 250us) has elapsed.
Setting this bit to "1" in the ACT_DEVICE state stops oscillation of DevicePLL480 and makes the transition to the ACTIVE60 state.
This bit is automatically cleared as soon as the transition has been made from any state and the MainIntStat.FinishedPM bit is set at the same time.

Bit 4: **GoActDevice**

Start transition to the ACT_DEVICE state from other states.
Setting this bit to "1" in the SLEEP state starts oscillation of the oscillator and PLL60 after the oscillation stable time (the time set for WakeupTim_H and L) has elapsed, starts oscillation of DevicePLL480 after the PLL60 oscillation stable time (about 250us) has elapsed and makes the transition to the ACT_DEVICE state after the PLL480 oscillation stable time (about 250us) has elapsed.
Setting this bit to "1" in the SNOOZE state starts oscillation of PLL60, starts oscillation of DevicePLL480 after the PLL oscillation stable time (about 250us) has elapsed and makes the transition to the ACT_DEVICE state after the PLL480 oscillation stable time (about 250us) has elapsed.
Setting this bit to "1" in the ACTIVE60 state starts oscillation of DevicePLL480 and makes the transition to the ACT_DEVICE state after the PLL480 oscillation stable time (about 250us) has elapsed.
This bit is automatically cleared as soon as the transition has been made from any state and the MainIntStat.FinishedPM bit is set at the same time.

* Though this LSI is masked to prevent the XINT signal from being asserted during SNOOZE by the interrupt status (hereafter, synchronous status) inaccessible during SLEEP / SNOOZE, apply the following processing through F/W to prevent the XINT terminal from being asserted as soon as SNOOZE is released.
  <Before starting SLEEP / SNOOZE>
  Process the synchronous status and clear it. (to IntStat)
  Disable the synchronous status. (to IntEnb)
  <After releasing SLEEP / SNOOZE>
  Clear the synchronous status. (to IntStat)
  Enable the synchronous status. (to IntEnb)

Bit [3:0]: **Reserved**

Power Management Control 1
USB[0x021] Default value = 0x00

—	—	—	—	***PM_State[3:0]***(R)			
7	6	5	4	3	2	1	0

You can monitor the operating state related to power management of this LSI.
This register is valid even during SLEEP / SNOOZE.

Bit [7:4]: **Reserved**

Bit [3:0]: **PM_State[3:0]**

Display the state in the power mode.

0000 : SLEEP state	(OSC off, PLL60 off, DevicePLL480 off)
0001 : SNOOZE state	(OSC on, PLL60 off, DevicePLL480 off)
0011 : ACTIVE60 state	(OSC on, PLL60 on, DevicePLL480 off)
0111 : ACT_DEVICE state	(OSC on, PLL60 on, DevicePLL480 on)

Others: Disable

Do not refer to this state as it successively changes toward the corresponding state until PM_Contorl0.GoXXXX is set, the MainIntStat.FinishedPM interrupt status is set and the PM_Control0.GoXXXX bit is cleared.

Wakeup Time[7:0]							
USB[0x022] Default value = 0x00							
WakeupTim[7:0]							
7	6	5	4	3	2	1	0

Wakeup Time[15:8]							
USB[0x023] Default value = 0x00							
WakeupTim[15:8]							
7	6	5	4	3	2	1	0

Set the oscillation stable time to return to the SNOOZE state from the SLEEP state. This register is accessible even during SLEEP.

When "1" is written in PM_Control0.GoActDevice, PM_Control0.GoActive60 and PM_Control0. GoSNOOZE bits during the SLEEP state, the oscillation cell is enabled and oscillation of the oscillator is started. Then, load the set value of Wakeup_Tim and start countdown at the startup of OSC. After finishing countdown, open the gate of the internal OSCCLK and start sending CLK to circuits of PLL and so on.

This oscillation stable time changes depending on the oscillator, oscillation cell, board and load capability. To go down to the SLEEP state in SUSPEND of USB during device operation, the internal SCLK must be stable in 60MHz±10% within 5.1ms from detecting RESET of USB.

Therefore, the total of the oscillation stable time + PLL60 stable time (less than 250us) + PLL480 stable time (less than 250us) must be 5.1ms or less.

A2.5.2.4 MISC register

Macro Reset							
USB[0x031] Default value = 0xXX							
—	—	—	—	—	—	—	***AllReset*** (W)
7	6	5	4	3	2	1	0

Reset this register.

Access is valid even during SLEEP / SNOOZE.

Bit [7:1]: **Reserved**

Bit 0: **AllReset**

Reset all circuits of this register. Synonymous with the external reset terminal (XRST)

Do not write anything in this register for other purposes than resetting it.

Note that writing anything in this register for other purposes than resetting it against the AC specification can cause malfunction.

Mode Protect							
USB[0x033] Default value = 0x56							
ModeProtect[7:0]							
7	6	5	4	3	2	1	0

Bit [7:0]: **ModeProtect[7:0]**

Protect values of the MacroConfig0 register and ClkSelect.ClkSelect bit. Writing "56h" in this register enables write access to the MacroConfig0 register and the ClkSelect.ClkSelect bit.

In normal usage, set the MacroConfig0 register and the ClkSelect.ClkSelect bit to any value and set other values (for example, "00h") than "56h" to this register to protect the setting of the MacroConfig0 register and the ClkSelect.ClkSelect bit.

Access to this bit is valid even during SLEEP / SNOOZE.

Macro Configuration 0
USB[0x035] Default value = 0x41

—	—	—	—	—	—	—	*ClkSelect*
7	6	5	4	3	2	1	0

Bit [7:1]: **Reserved**

Bit 0: **ClkSelect**
Select a clock to be used by this LSI. Access to this bit is valid even during SLEEP / SNOOZE.
0: 12MHz
1: 24MHz

Macro Configuration 1
USB[0x037] Default value = 0x06

IntLevel	*IntMode*	*DREQ_ Level*	*DACK_ Level*	*CS_Mode*	*CPU_Swap*	*BusMode*	*Bus8x16*
7	6	5	4	3	2	1	0

Specify the operation mode of this LSI. As the default value of this register is optimized for S2S65A00, do not change the setting unless otherwise required.

Bit 7: **IntLevel**
Set the logical level of XINT. Access to this bit is valid even during SLEEP / SNOOZE.
0: Negative logic
1: Positive logic

Bit 6: **IntMode**
Set the output level of XINT. Access to this bit is valid even during SLEEP / SNOOZE.
0: 1/0 mode
1: Hi-z/0 mode

Bit 5: **DREQ_Level**
Set the logical level of XDREQ0 and 1. Access to this bit is valid even during SLEEP / SNOOZE.
0: Negative logic
1: Positive logic

Bit 4: **DACK_Level**
Set the logical level of XDACK0 and 1. Access to this bit is valid even during SLEEP / SNOOZE.
0: Negative logic
1: Positive logic

Bit 3: **CS_Mode**
Set the operation mode of DMA0 and 1. Access to this bit is valid even during SLEEP / SNOOZE.
0: Operates as a valid DMA access when XDACK0 and 1 are asserted.
1: Operates as a valid DMA access when XCS and XDACK0 and 1 are asserted.

Bit 2: **CPU_Swap**
Set the CPU bus in the 16bit mode. Access to this bit is valid even during SLEEP / SNOOZE. Do not set this bit in the 8bit mode.
0: The even number address shall be the upper side and the odd number address shall be the lower side.
1: The even number address shall be the upper side and the odd number address shall be the lower side.
This bit setting will be enabled by reading the address “039h” after writing registers. When you reset the circuit with the ChipReset.ResetAll bit, the register is initialized. The setting is enabled after reading the address “039h” as described above.

Bit [1:0]: **BusMode, Bus8x16**

Set the operation mode of the CPU. Access to this bit is valid even during SLEEP / SNOOZE.

Operation Mode	bit1.BusMode	bit0.Bus8x16
16bit Strobe mode	0	0
16bit BE mode	1	*
8bit mode	0	1

Macro Type 0
USB[0x03A] Default value = 0x44

Device_Host[7:0](R)							
7	6	5	4	3	2	1	0

It indicates the macro type of this macro.

Bit [7:0]: **Device/Host[7:0]**

It indicates Device or Host.
Device: 0x44
Host: 0x48

Macro Type 1
USB[0x03B] Default value = 0x08

—	—	—	***Exist_ IDE***(R)	***EP_Num[3:0]*** (R)			
7	6	5	4	3	2	1	0

It indicates the macro type of this macro.

Bit [7:5]: **Reserved**

Bit 4: **Exist_IDE**

It indicates whether IDE is implemented or not.
Not mount : 0x0
Mount : 0x1

Bit [3:0]: **EP_Num[3:0]**

It indicates the End Point number.
It is expressed in hexadecimal.

Macro Type 2
USB[0x03C] Default value = 0x02

—	—	—	—	***DMAch_Count[3:0]*** (R)			
7	6	5	4	3	2	1	0

It indicates the macro type of this macro.

Bit [7:4]: **Reserved**

Bit [3:0]: **DMAch_Count[3:0]**

It indicates the total number of DMA channels.
It indicates the number of channels in hexadecimal.

Macro Type 3
USB[0x03D] Default value = 0x30

PortDMA_Mount[3:0] (R)				***CPUDMA_Mount[3:0]*** (R)			
7	6	5	4	3	2	1	0

It indicates the macro type of this macro.

Bit [7:4]: **PortDMA_Mount[3:0]**
It indicates the implemented Sync DMA channel.
It indicates channel implemented in binary number.

Bit [3:0]: **CPUDMA_Mount[3:0]**
It indicates the implemented CPU DMA channel.
It indicates channel implemented in binary number.

FIFO Capacity[7:0]
USB[0x03E] Default value = 0x00

FIFO_Capacity[7:0]							
7	6	5	4	3	2	1	0

FIFO Capacity[15:8]
USB[0x03F] Default value = 0x12

FIFO_Capacity[15:8]							
7	6	5	4	3	2	1	0

It indicates the capacity of the implemented FIFO RAM.

A2.5.2.5 DMA Control Register

DMA0 Configuration
USB[0x041] Default value = 0x00

FreeRun	—	—	—	ActiveDMA	—	—	—
7	6	5	4	3	2	1	0

DMA1 Configuration
USB[0x051] Default value = 0x00

FreeRun	—	—	—	ActiveDMA	—	—	—
7	6	5	4	3	2	1	0

Set the operation mode of DMA0/1.

Bit 7: **FreeRun**
Set the operation mode of DMA0/1.
0: Count Mode
1: Free-Running Mode

Bit [6:4]: **Reserved**

Bit 3: **ActiveDMA**

Enable PortDMA0/1. Clearing this bit during transfer immediately stops transfer. Initialize USB FIFO and PortDMA for retransmission. Use DMA_Stop for normal shutdown.
Note that this bit is automatically cleared when USBFIFO is cleared or when the join is changed.

1: PortDMA0/1 Enable
0: PortDMA0/1 Disable

Bit [2:0]: **Reserved**

DMA0 Control
USB[0x042] Default value = 0x00

Running (R)	—	—	Counter Clr（W）	Dir	—	Stop (W)	Go (W)
7	6	5	4	3	2	1	0

DMA1 Control
USB[0x052] Default value = 0x00

Running (R)	—	—	Counter Clr（W）	Dir	—	Stop (W)	Go (W)
7	6	5	4	3	2	1	0

Display the control and the status of DMA0/1.

Bit 7: **Running**

This bit is et to “1” during transfer of DMA0/1. You can’t update EPx{x=0,a~h}Join.JoinDMA0/1 while this bit is “1”.

Bit [6:5]: **Reserved**

Bit 4: **CounterClr**

If you set this bit to “1”, Count_HH, HL, LH and LL registers will be cleared to “0x00”. When the Running bit is “1”, rewriting this bit is ignored.

Bit 3: **Dir**

Set the transfer direction of DMA0/1.

0: Port → FIFO RAM (DMA Write)
1: Port ← FIFO RAM (DMA Read)

Bit 2: **Reserved**

Bit 1: **Stop**

Setting this bit to “1” terminates transfer of DMA0/1. Stopping transfer of DMA0/0 clears the DMA_Running bit to “0”. And it sets “1” to the DMA0/1_Cmp bit of the CPU_IntStat register. To restart transfer of DMA, check the Running and DMA0/1_Cmp bits and wait until DMa finishes.

Bit 0: **Go**

Setting this bit to “1” starts transfer of DMA0/1.

DMA0 Remain[7:0]
USB[0x044] Default value = 0x00

Remain[7:0](R)							
7	6	5	4	3	2	1	0

DMA1 Remain[7:0]
USB[0x54] Default value = 0x00

Remain[7:0](R)							
7	6	5	4	3	2	1	0

DMA0 Remain[12:8]
USB[0x045] Default value = 0x00

—	—	—	Remain[12:8](R)				
7	6	5	4	3	2	1	0

DMA1 Remain[12:8]
USB[0x055] Default value = 0x00

—	—	—	Remain[12:8](R)				
7	6	5	4	3	2	1	0

Bit [7:0]: **Remain[7:0]**

Bit [7:5]: **Reserved**

Bit [4:0]: **Remain[12:8]**

For reading, it indicates the remaining data in FIFO at the endpoint connected to DMA by the EPx{x=0,a to h}Join.JoinDMA0/1 bit.
For writing, it indicates the free space in FIFO at the endpoint connected to DMA by the EPx{x=0,a to h}Join.JoinDMA0/1 bit. You can't refer to the correct free space in FIFO with this register immediately after rewriting DMA. Check the free space in FIFO at an interval of at least one CPU cycle.
Read the upper byte and lower byte of this register in order.

DMA0 Count[23:16]
USB[0x048] Default value = 0x00

Count[23:16]							
7	6	5	4	3	2	1	0

DMA1 Count[23:16]
USB[0x058] Default value = 0x00

Count[23:16]							
7	6	5	4	3	2	1	0

DMA0 Count[31:24]
USB[0x049] Default value = 0x00

Count[31:24]							
7	6	5	4	3	2	1	0

DMA1 Count[31:24]
USB[0x059] Default value = 0x00

Count[31:24]							
7	6	5	4	3	2	1	0

DMA0 Count[7:0]
USB[0x04A] Default value = 0x00

Count[7:0]							
7	6	5	4	3	2	1	0

DMA1 Count[7:0]
USB[0x05A] Default value = 0x00

Count[7:0]							
7	6	5	4	3	2	1	0

DMA0 Count[15:8]
USB[0x04B] Default value = 0x00

Count[15:8]							
7	6	5	4	3	2	1	0

DMA1 Count[15:8]
USB[0x05B] Default value = 0x00

Count[15:8]							
7	6	5	4	3	2	1	0

Bit [7:0]: **Count[23:16]**

Bit [7:0]: **Count[31:24]**

Bit [7:0]: **Count[7:0]**

Bit [7:0]: **Count[15:8]**

Set the transfer data length of DMA0/1 in bytes in the count mode. You can set up to 0xFFFF_FFFF bytes. The set value is counted down. Set the transfer count to this register and set “1” to the DMA0/1_Control.Go bit to start DMA transfer. When the number of bytes set in this register have been transferred, DMA transfer terminates.

Count up the set value for the free-running mode. When the value of the DMA0/1_Count register overflows, “1” is set to the DMA0/1_CountUp bit of the DMAIntStat register. Counting continues even after the value overflowed. You can refer to the DMA transfer count in this mode.

You can’t check thecorrect count with this register immediately after rewriting DMA. Check the count at an interval of at least one CPU cycle. Read this register from the upper byte in sequence.

A2.5.2.6 USB Control Register

Device Interrupt Status							
USB[0x080]	Default value = 0x00						
VBUS_ Changed	Descriptor Cmp	***SIE_ IntStat***(R)	Bulk_ IntStat(R)	RcvEP0 SETUP	FIFO_ IntStat(R)	EP0_ IntStat(R)	EPr_ IntStat(R)
7	6	5	4	3	2	1	0

Display interrupts related to USB devices.

This register has a bit to indirectly instruct an interrupt factor and a bit to directly instruct it. The bit that indirectly instructs an interrupt factor allows you to the bit that directly instructs the interrupt factor by reading the corresponding interrupt status register. The bit that indirectly instructs an interrupt factor is read-only and is automatically cleared by clearing the bit that directly instructs the original interrupt factor. The bit that directly instructs an interrupt factor is writable and you can clear the interrupt factor by writing "1" in the corresponding bit.

Bit 7: **VBUS_Changed**

Directly instruct an interrupt factor.
This is set to "1" when the status of the VBUS terminal has changed.
Check the status of VBUS with the VBUS bit of the USB_Status register. If VBUS is "0", it indicates that the cable is pulled out. This bit is valid even during SLEEP / SNOOZE.

Bit 6: **DescriptorCmp**

Directly instruct an interrupt factor.
When the amount of data specified in the DescriptorSize register has been sent by the Descriptor return functionh, it is set to "1".
When it made the transition (received an OUT token) to the status stage before sending the amount of data specified in the DescriptorSize register, it is set to "1" together with the OUT_TranNAK bit of the EP0IntStat register.

Bit 5: **SIE_IntStat**

Indirectly instruct an interrupt factor.
This is set to "1" when the SIE_IntStat register has an interrupt factor and the bit of the SIE_IntEnb register corresponding to the interrupt factor is enabled. This bit is valid for reading even during SLEEP / SNOOZE.

Bit 4: **BulkIntStat**

Indirectly instruct an interrupt factor.
This is set to "1" when the BulkIntStat register has an interrupt factor and the bit of the BulkIntEnb register corresponding to the interrupt factor is enabled.

Bit 3: **RcvEP0SETUP**

Directly instruct an interrupt factor.
It is set to "1" when the setup stage for control transfer finishes and the received data is stored in EP0SETUP_0 through EP0SETUP_7 registers. At the same time, the ForceSTALL bit of the EP0ControlIN and EP0ControlOUT registers is set to "0" and the ForceNAK and ToggleStat bits of the EP0ControlIN and EP0ControlOUT registers and the ProtectEP0 bit of the SETUP_Control register are automatically set to "1". The AutoSetAddress function automatically replies to a SetAddress() request and does not set this status.

Bit 2: **FIFO_IntStat**

Indirectly instruct an interrupt factor.
This is set to "1" when the FIFO_IntStat register has an interrupt factor and the bit of the FIFO_IntEnb register corresponding to the interrupt factor is enabled.

Bit 1: **EP0IntStat**
Indirectly instruct an interrupt factor.
This is set to "1" when the EP0IntStat register has an interrupt factor and the bit of the EP0IntEnb register corresponding to the interrupt factor is enabled.

Bit 0: **EPrIntStat**
Indirectly instruct an interrupt factor.
This is set to "1" when the EPrIntStat register has an interrupt factor and the bit of the EPaIntEnb register corresponding to the interrupt factor is enabled.

EPr Interrupt Status
USB[0x081] Default value = 0x00

EPh IntStat(R) 7	EPg IntStat(R) 6	EPf IntStat(R) 5	EPe IntStat(R) 4	EPd IntStat(R) 3	EPc IntStat(R) 2	EPb IntStat(R) 1	EPa IntStat(R) 0

Display interrupts of the endpoint EPr.

Bit 7: **EPhIntStat**
Indirectly instruct an interrupt factor.
This is set to "1" when the EPhIntStat register has an interrupt factor and the bit of the EPhIntEnb register corresponding to the interrupt factor is enabled.

Bit 6: **EPgIntStat**
Indirectly instruct an interrupt factor.
This is set to "1" when the EPgIntStat register has an interrupt factor and the bit of the EPgIntEnb register corresponding to the interrupt factor is enabled.

Bit 5: **EPfIntStat**
Indirectly instruct an interrupt factor.
This is set to "1" when the EPfIntStat register has an interrupt factor and the bit of the EPfIntEnb register corresponding to the interrupt factor is enabled.

Bit 4: **EPeIntStat**
Indirectly instruct an interrupt factor.
This is set to "1" when the EPeIntStat register has an interrupt factor and the bit of the EPeIntEnb register corresponding to the interrupt factor is enabled.

Bit 3: **EPdIntStat**
Indirectly instruct an interrupt factor.
This is set to "1" when the EPdIntStat register has an interrupt factor and the bit of the EPdIntEnb register corresponding to the interrupt factor is enabled.

Bit 2: **EPcIntStat**
Indirectly instruct an interrupt factor.
This is set to "1" when the EPcIntStat register has an interrupt factor and the bit of the EPcIntEnb register corresponding to the interrupt factor is enabled.

Bit 1: **EPbIntStat**
Indirectly instruct an interrupt factor.
This is set to "1" when the EPbIntStat register has an interrupt factor and the bit of the EPbIntEnb register corresponding to the interrupt factor is enabled.

Bit 0: **EPaIntStat**
Indirectly instruct an interrupt factor.
This is set to "1" when the EPaIntStat register has an interrupt factor and the bit of the EPaIntEnb register corresponding to the interrupt factor is enabled.

SIE Interrupt Status
USB[0x082] Default value = 0x00

—	***NonJ***	RcvSOF	Detect Reset	Detect Suspend	Chirp Cmp	Restore Cmp	SetAddress Cmp
7	6	5	4	3	2	1	0

Display interrupts related to the device SIE.
You can clear interrupt factors by writing "1" for all bits.

Bit 7: **Reserved**

Bit 6: **NonJ**
Directly instruct an interrupt factor.
This is set to "1" when other statuses than J state are detected on the USB bus. This bit is valid when this LSI is in the SNOOZE state (the InSnooze bit of the PM_Control register is "1") and the InSUSPEND of the USB_Control register is set to "1" to use the AutoNegotiation function.

Bit 5: **RcvSOF**
Directly instruct an interrupt factor.
This is set to "1" when the SOF token is received.

Bit 4: **DetectReset**
Directly instruct an interrupt factor.
This is set to "1" when the reset state of USB is detected. When this bit is set, the suspend state of USB cannot be detected (DetectSUSPEND is not set).
This reset detection is enabled when the ActiveUSB bit of the NegoControl register is set to "1".
For "HS" operation mode, when a bus activity stops for a certain period, FS termination is automatically set to detect reset/suspend of USB. When SE0 is detected, this bit is set to "1" by assessing it as reset.
To disable the AutoNegotiation function, set the DisBusDetect bit of the NegoControl register to "1" to disable detecting the reset/suspend state of USB so that the continuing reset may not be detected by mistake when this bit is set to "1". After finishing the reset processing, clear the DisBusDetect bit and enable detecting the reset/suspend state of USB.
To detect a reset, you can start "HS Detection Handshake" with the GoChirp of the NegoControl register.
See the section of the EnAutoNego bit of the NegoControl register for the AutoNegotiation function.

Bit 3: **DetectSuspend**
Directly instruct an interrupt factor.
This is set to "1" when the suspend state of USB is detected. When this bit is set, the reset state of USB cannot be detected (DetectRESET is not set).
For "HS" operation mode, when a bus activity stops for a certain period, FS toperation mode is automatically set to detect reset/suspend of USB. After detecting the suspend state of USB, you can change this LSI to the snooze mode (stop oscillation of the built-in PLL) by setting the GoSnooze bit of the PM_Control0 register.

Bit 2: **ChirpCmp**
Directly instruct an interrupt factor.
This is set to "1" when "HS Detection Handshake" started by the GoChirp bit of the NegoControl register is completed,
You can assess the current operation mode (FS or HS) by reading the FSxHS bit of the USB_Status register after the interrupt is generated.

Bit 1: **RestoreCmp**
Directly instruct an interrupt factor.
This is set to "1" when Restore process started by the RestoreUSB bit of the NegoControl register finishes. When this bit is set to "1", the operation mode (FS or HS) returns to the status before Suspend.

Bit 0: **SetAddressCmp**
Directly instruct an interrupt factor.
When a SetAddress() request is received, the AutoSetAddress function (see USB_Address register) automatically runs the control transfer process. This status is set to "1" when the control transfer related to the SetAddress() request is completed through the status stage. An addressis set to the USB_Address register at the same time.

Though you can read synchronous bits (Bit5 to 0) in the ACTIVE60 state, you can't rewrite it (clear the interrupt factor).

To leave the ACT_DEVICE state, carry out the following steps with F/W so that the interrupt signal XINT may nor be asserted by these interrupt statuses.

<To leave the ACT_DEVICE state>
1) Handle the interrupt status and clear it (SIE_IntStat.Bit5 to 0).
2) Disable the interrupt status (SIE_IntEnb.Bit5 to 0).

<To enter the ACT_DEVICE state>
1) Clear the interrupt status (SIE_IntStat.Bit5 to 0).
2) Enable the interrupt status (SIE_IntEnb.Bit5 to 0).

FIFO Interrupt Status
USB[0x084] Default value = 0x00

—	—	FIFO_ DMA1_Cmp	FIFO_ DMA0_Cmp	—	FIFO_ NotEmpty	FIFO_ Full	FIFO_ Empty
7	6	5	4	3	2	1	0

Display interrupt statuses related to the device FIFO.
You can clear interrupt factors by writing "1" for all bits.

Bit [7:6]: **Reserved**

Bit 5: **FIFO_DMA1_Cmp**
Directly instruct an interrupt factor.
This is set to "1" when the endpoint joined to DMA1 is in IN direction and FIFO gets empty after DMA1 transfer has finished. This is set to "1" when the endpoint joined to DMA1 is in OUT direction and DMA1 transfer has finished.

Bit 4: **FIFO_DMA0_Cmp**
Directly instruct an interrupt factor.
This is set to "1" when the endpoint joined to DMA0 is in IN direction and FIFO gets empty after DMA0 transfer has finished. This is set to "1" when the endpoint joined to DMA0 is in OUT direction and DMA0 transfer has finished.

Bit 3: **Reserved**

Bit 2: **FIFO_NotEmpty**
Directly instruct an interrupt factor.
This is set to "1" when the EPx{x=0,a to h}Join.JoinFIFO_Stat bit is set to "1" and the FIFO area of the corresponding endpoint is not empty.

Bit 1: **FIFO_Full**
Directly instruct an interrupt factor.
This is set to "1" when the EPx{x=0,a to h}Join.JoinFIFO_Stat bit is set to "1" and the FIFO area of the corresponding endpoint is full.

Bit 0: **FIFO_Empty**
Directly instruct an interrupt factor.
This is set to “1” when the EPx{x=0,a to h}Join.JoinFIFO_Stat bit is set to “1” and the FIFO area of the corresponding endpoint is empty.

Bulk Interrupt Status
USB[0x085] Default value = 0x00

CBW_ Cmp 7	CBW_ LengthErr 6	CBW_ Err 5	— 4	CSW_ Cmp 3	CSW_ Err 2	— 1	— 0

Display interrupt statuses related to the bulk transfer function. You can clear interrupt factors by writing “1” for all bits.

Bit 7: **CBW_Comp**
Directly instruct an interrupt factor.
This is set to “1” when 31 bytes of CBW have been correctly received.

Bit 6: **CBW_LengthErr**
Directly instruct an interrupt factor.
This is set to “1” when the packet lengtgh of the received CBW is not 31 bytes.

Bit 5: **CBW_Err**
Directly instruct an interrupt factor.
This is set to “1” when transaction errors such as the CRC error of the received CBW are detected.

Bit 4: **Reserved**

Bit 3: **CSW_Cmp**
Directly instruct an interrupt factor.
This is set to “1” when 13 bytes of CSW have been normally sent.

Bit 2: **CSW_Err**
Directly instruct an interrupt factor.
This is set to “1” when an error is found in sending CSW (when ACK is not returned).

Bit [1:0]: **Reserved**

EP0 Interrupt Status

USB[0x087] Default value = 0x00

—	OUT_ ShortACK	IN_ TranACK	OUT_ TranACK	IN_ TranNAK	OUT_ TranNAK	IN_ TranErr	OUT_ TranErr
7	6	5	4	3	2	1	0

Display interrupt statuses of the endpoint EP0. You can clear interrupt factors by writing "1" for all bits.

Bit 7: **Reserved**

Bit 6: **OUT_ShortACK**
Directly instruct an interrupt factor.
This is set to "1" at the same time as OUT_TranACK when a short packet is received for OUT transaction and ACK is returned.

Bit 5: **IN_TranACK**
Directly instruct an interrupt factor.
This is set to "1" when ACK is received for IN transaction.

Bit 4: **OUT_TranACK**
Directly instruct an interrupt factor.
This is set to "1" when ACK is returned for OUT transaction.

Bit 3: **IN_TranNAK**
Directly instruct an interrupt factor.
This is set to "1" when NAK is returned for IN transaction.

Bit 2: **OUT_TranNAK**
Directly instruct an interrupt factor.
This is set to "1" when NAK is returned for OUT and PING transactions.

Bit 1: **IN_TranErr**
Directly instruct an interrupt factor.
This is set to "1" when STALL is returned for an IN transaction, an error is found in the packet and handshake has caused time-out.

Bit 0: **OUT_TranErr**
Directly instruct an interrupt factor.
This is set to "1" when STALL is returned for an OUT transaction and an error is found in the packet.

EPa Interrupt Status
USB[0x088] Default value = 0x00

—	OUT_ ShortACK	IN_ TranACK	OUT_ TranACK	IN_ TranNAK	OUT_ TranNAK	IN_ TranErr	OUT_ TranErr
7	6	5	4	3	2	1	0

EPb Interrupt Status
USB[0x089] Default value = 0x00

—	OUT_ ShortACK	IN_ TranACK	OUT_ TranACK	IN_ TranNAK	OUT_ TranNAK	IN_ TranErr	OUT_ TranErr
7	6	5	4	3	2	1	0

EPc Interrupt Status
USB[0x08A] Default value = 0x00

—	OUT_ ShortACK	IN_ TranACK	OUT_ TranACK	IN_ TranNAK	OUT_ TranNAK	IN_ TranErr	OUT_ TranErr
7	6	5	4	3	2	1	0

EPd Interrupt Status
USB[0x08B] Default value = 0x00

—	OUT_ ShortACK	IN_ TranACK	OUT_ TranACK	IN_ TranNAK	OUT_ TranNAK	IN_ TranErr	OUT_ TranErr
7	6	5	4	3	2	1	0

EPe Interrupt Stauts
USB[0x08C] Default value = 0x00

—	OUT_ ShortACK	IN_ TranACK	OUT_ TranACK	IN_ TranNAK	OUT_ TranNAK	IN_ TranErr	OUT_ TranErr
7	6	5	4	3	2	1	0

EPf Interrupt Status
USB[0x08D] Default value = 0x00

—	OUT_ ShortACK	IN_ TranACK	OUT_ TranACK	IN_ TranNAK	OUT_ TranNAK	IN_ TranErr	OUT_ TranErr
7	6	5	4	3	2	1	0

EPg Interrupt Status
USB[0x08E] Default value = 0x00

—	OUT_ ShortACK	IN_ TranACK	OUT_ TranACK	IN_ TranNAK	OUT_ TranNAK	IN_ TranErr	OUT_ TranErr
7	6	5	4	3	2	1	0

EPh Interrupt Status
USB[0x08F] Default value = 0x00

—	OUT_ ShortACK	IN_ TranACK	OUT_ TranACK	IN_ TranNAK	OUT_ TranNAK	IN_ TranErr	OUT_ TranErr
7	6	5	4	3	2	1	0

Display interrupt statuses of the endpoint EPx (a-h). You can clear interrupt factors by writing “1” for all bits.

Bit 7: **Reserved**

Bit 6: **OUT_ShortACK**
Directly instruct an interrupt factor.
This is set to "1" at the same time as OUT_TranACK when a short packet is received for OUT transaction and ACK is returned.

Bit 5: **IN_TranACK**
Directly instruct an interrupt factor.
This is set to "1" when ACK is received for IN transaction.

Bit 4: **OUT_TranACK**
Directly instruct an interrupt factor.
This is set to "1" when ACK is returned for OUT transaction.

Bit 3: **IN_TranNAK**
Directly instruct an interrupt factor.
This is set to "1" when NAK is returned for IN transaction.

Bit 2: **OUT_TranNAK**
Directly instruct an interrupt factor.
This is set to "1" when NAK is returned for OUT and PING transactions.

Bit 1: **IN_TranErr**
Directly instruct an interrupt factor.
This is set to "1" when STALL is returned for an IN transaction, an error is found in the packet and handshake has caused time-out.

Bit 0: **OUT_TranErr**
Directly instruct an interrupt factor.
This is set to "1" when STALL is returned for an OUT transaction and an error is found in the packet.

Device Interrupt Enable
USB[0x090] Default value = 0x00

EnVBUS_ Changed 7	EnDescriptor Cmp 6	***EnSIE_ IntStat*** 5	EnBulk IntStat 4	EnRcvEP0 SETUP 3	EnFIFO_ IntStat 2	EnEP0 IntStat 1	EnEPr IntStat 0

This permits/prohibits asserting the DeviceIntStat bit of the MainIntStat register due to interrupt factors of the DeviceIntStat register.

EnVBUS_Changed and EnSIE_IntStat bits are valid even during SLEEP / SNOOZE.

EPr Interrupt Enable
USB[0x091] Default value = 0x00

EnEPh IntStat 7	EnEPg IntStat 6	EnEPf IntStat 5	EnEPe IntStat 4	EnEPd IntStat 3	EnEPc IntStat 2	EnEPb IntStat 1	EnEPa IntStat 0

This permits/prohibits asserting the EPrIntStat bit of the DeviceIntStat register due to interrupt factors of the EPrIntStat register.

SIE Interrupt Enable
USB[0x092] Default value = 0x00

—	***EnNonJ***	EnRcvSOF	EnDetect RESET	EnDetect SUSPEND	EnChirp Cmp	EnRestore Cmp	EnSet AddressCmp
7	6	5	4	3	2	1	0

This permits/prohibits asserting the SIE_IntStat bit of the DeviceIntStat register due to interrupt factors of the SIE_IntStat register.

The EnNonJ bit is valid even during SLEEP / SNOOZE.

Though you can read synchronous bits (Bit5 to 0) in the ACTIVE60 state, you can't rewrite it. See the description of the SIE_IntStat register for the process to leave the ACT_DEVICE state of these synchronous bits.

FIFO Interrupt Enable
USB[0x094] Default value = 0x00

—	—	EnFIFO_ DMA1_Cmp	EnFIFO_ DMA0_Cmp	—	EnFIFO_ NotEmpty	EnFIFO_ Full	EnFIFO_ Empty
7	6	5	4	3	2	1	0

This permits/prohibits asserting the FIFO_IntStat bit of the DeviceIntStat register due to interrupt factors of the FIFO_IntStat register.

Bulk Interrupt Enable
USB[0x095] Default value = 0x00

EnCBW_ Cmp	EnCBW_ LengthErr	EnCBW_ Err	—	EnCSW_ Cmp	EnCSW_Err	—	—
7	6	5	4	3	2	1	0

This permits/prohibits asserting the BulkIntStat bit of the DeviceIntStat register due to interrupt factors of the BulkIntStat register.

EP0 Interrupt Enable
USB[0x097] Default value = 0x00

—	EnOUT_ ShortACK	EnIN_ TranACK	EnOUT_ TranACK	EnIN_ TranNAK	EnOUT_ TranNAK	EnIN_ TranErr	EnOUT_ TranErr
7	6	5	4	3	2	1	0

This permits/prohibits asserting the EP0IntStat bit of the DeviceIntStat register due to interrupt factors of the EP0IntStat register.

EPa Interrupt Enable
USB[0x098] Default value = 0x00

—	EnOUT_ ShortACK	EnIN_ TranACK	EnOUT_ TranACK	EnIN_ TranNAK	EnOUT_ TranNAK	EnIN_ TranErr	EnOUT_ TranErr
7	6	5	4	3	2	1	0

EPb Interrupt Enable
USB[0x099] Default value = 0x00

—	EnOUT_ ShortACK	EnIN_ TranACK	EnOUT_ TranACK	EnIN_ TranNAK	EnOUT_ TranNAK	EnIN_ TranErr	EnOUT_ TranErr
7	6	5	4	3	2	1	0

EPc Interrupt Enable
USB[0x09A] Default value = 0x00

—	EnOUT_ ShortACK	EnIN_ TranACK	EnOUT_ TranACK	EnIN_ TranNAK	EnOUT_ TranNAK	EnIN_ TranErr	EnOUT_ TranErr
7	6	5	4	3	2	1	0

EPd Interrupt Enable
USB[0x09B] Default value = 0x00

—	EnOUT_ ShortACK	EnIN_ TranACK	EnOUT_ TranACK	EnIN_ TranNAK	EnOUT_ TranNAK	EnIN_ TranErr	EnOUT_ TranErr
7	6	5	4	3	2	1	0

EPe Interrupt Enable
USB[0x09C] Default value = 0x00

—	EnOUT_ ShortACK	EnIN_ TranACK	EnOUT_ TranACK	EnIN_ TranNAK	EnOUT_ TranNAK	EnIN_ TranErr	EnOUT_ TranErr
7	6	5	4	3	2	1	0

EPf Interrupt Enalbe
USB[0x09D] Default value = 0x00

—	EnOUT_ ShortACK	EnIN_ TranACK	EnOUT_ TranACK	EnIN_ TranNAK	EnOUT_ TranNAK	EnIN_ TranErr	EnOUT_ TranErr
7	6	5	4	3	2	1	0

EPg Interrupt Enable
USB[0x09E] Default value = 0x00

—	EnOUT_ ShortACK	EnIN_ TranACK	EnOUT_ TranACK	EnIN_ TranNAK	EnOUT_ TranNAK	EnIN_ TranErr	EnOUT_ TranErr
7	6	5	4	3	2	1	0

EPh Interrupt Enable
USB[0x09F] Default value = 0x00

—	EnOUT_ ShortACK	EnIN_ TranACK	EnOUT_ TranACK	EnIN_ TranNAK	EnOUT_ TranNAK	EnIN_ TranErr	EnOUT_ TranErr
7	6	5	4	3	2	1	0

This permits/prohibits asserting the EPx{a-h}IntStat bit of the EPrIntStat register due to interrupt factors of the EPx{a-h}IntStat register.

Reset DTM							
USB[0x0A0] Default value = 0x01							
—	—	—	—	—	—	—	***ResetDTM***
7	6	5	4	3	2	1	0

Reset the transceiver macro of the device.
Access is valid even during SLEEP / SNOOZE.

Bit [7:1]: **Reserved**

Bit 0: **ResetDTM**
Setting this bit to "1" initializes the device transceiver macro of this LSI.
To cancel reset, clear this bit.

Negotiation Control							
USB[0x0A2] Default value = 0x00							
DisBus Detect	EnAuto Nego	In SUSPEND	Disable HS	Send Wakeup	Restore USB	GoChirp	Active USB
7	6	5	4	3	2	1	0

Set the operation on the device negotiation.

Bit 7: **DisBusDetect**
Setting this bit to "1" disables automatic detection of reset/suspend state of USB. When this bit is cleared, the bus activity on the USB bus is monitored to detect reset/suspend state of USB.
If the bus activity is not detected for 3ms in "HS" mode, the mode is automatically changed to "FS" mode to determine reset or suspend state of USB and then set the corresponding interrupt factor (DetectReset, DetectSuspend). If the bus activity is not detected for 3ms in "FS" mode, the device determines it to be the suspend state of USB. If the device detects 2.5µses or more "SE0", it assesses reset and sets the corresponding interrupt factor.
When "1" is set to DetectReset and DetectSuspend bits, set "1" to the DisBusDetect bit and disable detection while the reset/suspend state of USB is ongoing. When you use the AutoNegotiation function, do not set "1" to this bit.

Bit 6: **EnAutoNego**
Enable the AutoNegotiation function. The AutoNegotiation function automates the sequence from the termination of speed negoatiation to the decision of the speed mode whenit detects a reset. See the chapter on the operation description for details of the AutoNegotiation function.

Bit 5: **InSUSPEND**
This is automatically set to "1" and enables the function to detect the NonJ state when the AutoNegotiation function is in use and the suspend state of USB is detected. Clear this bit to return from the suspend state of USB.
When you use the AutoNegotiation function, see "Functional description: Auto negoatiation function".

Bit 4: **DisableHS**
When GoChirp is set to "1" and this bit is set to "1", the device is compeled to be in the FS mode without sending DeviceChirp and generates a ChirpCmp interruption.

Bit 3: **SendWakeup**
Setting this bit to "1" outputs the RemoteWakeup signal (K) to the USB port.
Clear this bit after 1ms or more and 15ms or less passed since starting to send the RemoteWakeup signal and stop sending it.

Bit 2: **RestoreUSB**

Setting this bit to "1" to resume from the suspend state of USB automatically changes the operation mode (FS or HS) saved before suspending USB and sets the corresponding interrupt factor (RestoreCmp).
This bit is automatically cleared after the operation finishes.
When you use the AutoNegotiation function, do not set or clear this bit as the function of this bit is automatically controlled.

Bit 1: **GoChirp**

When the USB bus is in a reset condition, setting this bit to "1" causes "HS Detection Handshake" between the host and the hub and automatically sets the TermSelect and XcvrSelect bits of the XcvrControl register and the FSxHS bit of the USB_Status register. As soon as the operation finishes, the interrupt factor (ChirpCmp) is set.
This bit is automatically cleared after the operation finishes. You can check the result of "HS Detection Handshake" by referring to the FSxHS bit of the USBStauts register after the operation has finished.
When you use the AutoNegotiation function, do not set or clear this bit as the function of this bit is automatically controlled.

Bit 0: **ActiveUSB**

As this bit is cleared after resetting the hardware of this LSI, all functions of the USB device stop. The operation of the USB device is enabled by setting this bit to "1" after having set this LSI.

USB Status
USB[0x0A4] Default value = 0xXX

VBUS(R)	FSxHS	—	—	—	—	***LineState[1:0]*** (R)	
7	6	5	4	3	2	1	0

Display statuses related to the device.

Bit 7: **VBUS**

The status of the VBUS terminal is displayed. This bit is valid even during SLEEP / SNOOZE.

Bit 6: **FSxHS**

This indicates the current operation mode. This is automatically set by executing "HS Detection Handshake" (see the functional description) with the NegoControl.GoChirp bit. Though you can mandatorily change the operation mode by rewriting this bit, handle this bit only when you want to change the operation mode by simulation without executing "HS Detection Handshake".
Set it to "FS(1)" to attach cable.
This bit allows reading in case of ACTIVE60 / ACT_DEVICE and allows writing in case of ACT_DEVICE.

Bit [5:2]: **Reserved**

Bit [1:0]: **LineState [1:0]**

This indicates the status of the signal on the USB cable. This bit is valid even during SLEEP / SNOOZE.
When the TermSelect bit of the XcvrControl register is "1" (when FS termination is selected), it indicates the received value of the FS receiver of DP/DM if the XcvrSelect bit is "1" (FS transceiver is selected) and indicates the received value of the HS receiver if the XcvrSelect bit is "0" (HS transceiver is selected).
It indicates the bus activity of USB when TermSelect is "0".

LineState		
TermSelect	DP / DM	LineState [1:0]
0	Don't Care	Bus activity
1	SE0	0b00
1	J	0b01
1	K	0b10
1	SE1	0b11

Xcvr Control
USB[0x0A5] Default value = 0x41

Term Select	XcvrSelect	—	—	—	—	OpMode[1:0]	
7	6	5	4	3	2	1	0

Make the setting related to transceiver macros of the device.

Bit 7: **TermSelect**
Select either FS or HS termination to enable it. This bit is automatidcally set when "HS Detection Handshake" is executed with the GoChirp bit of the USB_Control register or when the EnAutoNego bit of the D_NegoControl register is set to execute the AutoNegotiation function.

Bit 6: **XcvrSelect**
Select either FS or HS transceiver to enable it. This bit is automatidcally set when "HS Detection Handshake" is executed with the GoChirp bit of the D_NegoControl register or when the EnAutoNego bit of the D_NegoControl register is set to execute the AutoNegotiation function.

Bit [5:2]: **Reserved**

Bit [1:0]: **OpMode**
Set the operation mode of UTM.
You may not usually set it when the USB cable is pulled out (*), when USB is suspended or when it is not in the test mode.

OpMode		
00	"Normal Operation"	"Normal use state"
01	"Non-Driving"	Enable this state when the USB cable is pulled out.
10	"Disable Bitstuffing and NRZI encoding"	Enable this state in the USB test mode.
11	"Power-Down"	Enable this state when the USB is suspended.

* It is recommended that this register is set to "41h" when the USB cable is pulled out.

USB Test
USB[0x0A6] Default value = 0x00

EnHS_ Test	—	—	—	Test_ SE0_NAK	Test_J	Test_K	Test_ Packet
7	6	5	4	3	2	1	0

Set the operation on the test mode of USB 2.0 of the device. Enable the operation in the test mode defined by the USB 2.0 standard by setting the bit corresponding to the test mode specified by the SetFeature request and setting "1" to the EnHS_Test bit after finishing the status stage.

Bit 7: **EnHS_Test**

Setting "1" to this bit enters the test mode corresponding to the bit when any one of the lower four bits of the USB_Test register is set to "1". You need to set "1" to the DisBusDetect bit of the NegoControl register to disable detecting USB suspend and reset to conduct a test mode. Clear the EnAutoNego bit of the NegoControl register to disable the AutoNegotiation function.
Note that you transfer to the test mode after finishing the status stage by the SetFeature request.

Bit [6:4]: **Reserved**

Bit 3: **Test_SE0_NAK**

You can enter the Test_SE0_NAK test mode by setting "1" to this bit and the EnHS_Test bit.

Bit 2: **TEST_J**

You can enter the Test_J test mode by setting "1" to this bit and the EnHS_Test bit. Set TermSelect and XcvrSelect of the XcvrControl register according to the speed and set "10" (Disable Bitstuffing and NRZI encoding) to OpMode before setting "1" to the EnHS_Test in this test mode.

Bit 1: **TEST_K**

You can enter the Test_K test mode by setting "1" to this bit and the EnHS_Test bit. Set TermSelect and XcvrSelect of the XcvrControl register according to the speed and set "10" (Disable Bitstuffing and NRZI encoding) to OpMode before setting "1" to the EnHS_Test bit in this test mode.

Bit 0: **Test_Packet**

You can enter the Test_packet test mode by setting "1" to this bit and the EnHS_Test bit.
Set up as follows as this test mode is available at any endpoint except EP0.

1) Set the MaxPacketSize of the endpoint EPx{x=a to h} to 64 or more, the transmission direction to IN and the EndpointNumber to "0xF" to enable them. Allocate 64 bytes or more to FIFO of the endpoint EPx{x=a to h}.
2) Set the endpoint so that it may not overlap the setting of EPx{x=a to h} described above. Otherwise, clear the EPx{x=a to h}Config.EnEndpoint bit.
3) Clear FIFo of EPx{x=a to h} and write the data for test packet described below in this FIFO. Clear the IN_TranErr bit of the EPx{x=a to h}IntStat register.
4) The IN_TranErr status is set to "1" every time a test packet has been transferred. The followingb 53 bytes are written in FIFO in the packet transfer test mode.
   00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h,
   00h, Aah, Aah, Aah, Aah, Aah, Aah, Aah,
   Aah, Eeh, Eeh, Eeh, Eeh, Eeh, Eeh, Eeh,
   Eeh, Feh, FFh, FFh, FFh, FFh, FFh, FFh,
   FFh, FFh, FFh, FFh, FFh, 7Fh, BFh, DFh,
   Efh, F7h, FBh, FDh, FCh, 7Eh, BFh, DFh,
   Efh, F7h, FBh, FDh, 7Eh

As SIE adds PID and CRC in sending a test packet, the data next to DATA0 PID and that except CRC16 among test packet data described in the USB stabndard 2.0 are written in FIFO.

EPn Control
USB[0x0A8] Default value = 0xXX

AllForce NAK(W) 7	EPrForce STALL(W) 6	AllFIFO_ Clr(W) 5	— 4	— 3	— 2	— 1	EP0FIFO_ Clr(W) 0

Set the operation of the endpoint. This is the write-only register.

Bit 7: **AllForceNAK**
Set "1" to the ForceNAK bit of all endpoints.

Bit 6: **EPrForceSTALL**
Set "1" to the ForceSTALL bit of the endpoint EPx{x=a to h}.

Bit 5: **AllFIFO_Clr**
FIFO of all endpoints is cleared. When you have set the area of each endpoint, be sure to set "1" to this bit after finishing the setting and clear FIFO of all endpoints. This bit is automatically cleared after FIFO has been cleared.
Do not set "1" to the bit of the corresponding endpoint when DMAx{x=0 to 3} is joined to the endpoint and the corresponding DMA is running (the running bit is "1").

Bit [4:1]: **Reserved**

Bit 0: **EP0FIFO_Clr**
Clear FIFO of the endpoint EP0.
When this bit is set to "1", only FIFO is cleared and the set value is not kept.
Do not set "1" to this bit when DMAx{x=0 to 13} is joined to the endpoint EP0 and the corresponding DMA is running (the running bit is "1").

EPr FIFO Clr
USB[0x0A9] Default value = 0xXX

EPh FIFO_Clr(W) 7	EPg FIFO_Clr(W) 6	EPf FIFO_Clr(W) 5	EPe FIFO_Clr(W) 4	EPd FIFO_Clr(W) 3	EPc FIFO_Clr(W) 2	EPb FIFO_Clr(W) 1	EPa FIFO_Clr(W) 0

Clear FIFO of the corresponding endpoint. This is the write-only register.
When each bit of this register is set to "1", only FIFO is cleared and the set value is not kept.
Do not set "1" to the bit of the corresponding endpoint when DMAx{x=0 to 1} is joined to the endpoint and the corresponding DMA is running (the running bit is "1").

Clr All EPn Join
USB[0x0AA] Default value = 0xXX

— 7	ClrJoin FIFO_Stat(W) 6	— 5	— 4	ClrJoin DMA1(W) 3	ClrJoin DMA0(W) 2	ClrJoin CPU_Rd(W) 1	ClrJoin CPU_Wr(W) 0

Clear the connection of the corresponding port and each endpoint. This is the write-only register.
Bits of this register are automatically cleared after the connection is cleared.
Do not set a bit of this register to "1" when the endpoint is connected to the port (the corresponding bit of the EPx{x=0,a to h}Join register is set to "1") and each port is active. It can cause a malfunction.

Bulk Only Control
USB[0x0AC] Default value = 0x00

AutoForce NAK_CBW	—	—	—	—	GoCBW_ Mode	GoCSW_ Mode	—
7	6	5	4	3	2	1	0

It controls the bulk only support function.

Bit 7: **AutoForceNAK_CBW**
When this bit is set to "1", the ForceNAK bit of the corresponding endpoint is set to "1" as soon as the OUT transaction received by CBW is completed by the CBW support.

Bit [6:3]: **Reserved**

Bit 2: **GoCBW_Mode**
When this bit is set to "1", the CBW support is executed at the corresponding endpoint. See the section on the BulkOnlyConfig register for the endpoint to execute the CBW support.

Bit 1: **GoCSW_Mode**
When this bit is set to "1", the CSW support is executed at the corresponding endpoint. See the section on the BulkOnlyConfig register for the endpoint to execute the CSW support.

Bit 0: **Reserved**

Bulk Only Configuration
USB[0x0AD] Default value = 0x00

EPh BulkOnly	EPg BulkOnly	EPf BulkOnly	EPe BulkOnly	EPd BulkOnly	EPc BulkOnly	EPb BulkOnly	EPa BulkOnly
7	6	5	4	3	2	1	0

It enables the bulk only support function.

Bit [7:0]: **EPx{a-h}BulkOnly**
If this bit is set to "1", the bulk only support function is enabled at the endpoint EPx{a-h}. When the bulk only support is enabled, the CBW support is executed by setting the BulkOnlyControl.GoCBW_Mode bit when the endpoint EPx{a-h} is OUT endpoint. The CBW support is executed by setting the BulkOnlyControl.GoCSW_Mode bit when the endpoint EPx{a-h} is IN endpoint.
Do not enable the bulk only support function at two or more OUT endpoints at the same time.

EP0 SETUP 0
USB[0x0B0] Default value = 0x00

EP0 SETUP_0 to 7[7:0](R)							
7	6	5	4	3	2	1	0

EP0 SETUP 1
USB[0x0B1] Default value = 0x00

EP0 SETUP_0 to 7[7:0](R)							
7	6	5	4	3	2	1	0

EP0 SETUP 2
USB[0x0B2] Default value = 0x00

EP0 SETUP_0 to 7[7:0](R)							
7	6	5	4	3	2	1	0

EP0 SETUP 3
USB[0x0B3] Default value = 0x00

EP0 SETUP_0 to 7[7:0](R)							
7	6	5	4	3	2	1	0

EP0 SETUP 4
USB[0x0B4] Default value = 0x00

EP0 SETUP_0 to 7[7:0](R)							
7	6	5	4	3	2	1	0

EP0 SETUP 5
USB[0x0B5] Default value = 0x00

EP0 SETUP_0 to 7[7:0](R)							
7	6	5	4	3	2	1	0

EP0 SETUP 6
USB[0x0B6] Default value = 0x00

EP0 SETUP_0 to 7[7:0](R)							
7	6	5	4	3	2	1	0

EP0 SETUP 7
USB[0x0B7] Default value = 0x00

EP0 SETUP_0 to 7[7:0](R)							
7	6	5	4	3	2	1	0

8-byte data received in the stup stage at the endpoint EP0 is sequentially stored from EP0SETUP_0.

EP0SETUP_0
BmRequestType is set.

EP0SETUP_1
BRequest is set.

EP0SETUP_2
The lower eight bits of Wvalue are set.

EP0SETUP_3
The upper eight bits of Wvalue are set.

EP0SETUP_4
The lower eight bits of WIndex are set.

EP0SETUP_5
The upper eight bits of WIndex are set.

EP0SETUP_6

The lower eight bits of WLength are set.

EP0SETUP_7

The upper eight bits of WLength are set.

USB Address
USB[0x0B8] Default value = 0x00

—	USB Address[6:0]						
7	6	5	4	3	2	1	0

The USB address is set by the AutoSetAddress function.

When a SetAddress() request is received, the AutoSetAddress function automatically runs the control transfer process. The AutoSetAddress function issues a SetAddressCmp status after the status stage of the control transfer related to the SetAddress() request is completed and the USB_Address is set.

Bit 7: **Reserved**

Bit [6:0]: **USB_Address**

The USB address is set.
It is automatically written by the AutoSetAddress function.
Though it can be written, it is automatically rewritten when it receives a SetAddress() request.

SETUP Control
USB[0x0BA] Default value = 0x00

—	—	—	—	—	—	—	Protect EP0
7	6	5	4	3	2	1	0

Make a setting related to control transfer.

Bit [7:1]: **Reserved**

Bit 0: **ProtectEP0**

It is set to "1" when the setup stage for control transfer finishes and the received data is stored in EP0SETUP_0 through EP0SETUP_7 registers.
At the same time, the ForceSTALL bit of the EP0ControlIN and EP0ControlOUT registers is set to "0" and the ForceNAK and ToggleStat bits are automatically set to "1".
The ProtectEP0 bit is set when SETUP transaction is processed. Therefore, it is set for a SetAddress() request.
When this bit is set to "1", you cannot change the setting of ForceNAK and ForceSTALL bits of EP0.

Frame Number L
USB[0x0BE] Default value = 0x00

FrameNumber[7:0](R)							
7	6	5	4	3	2	1	0

Frame Number H
USB[0x0BF] Default value = 0x00

FN_ Invalid(R)	—	—	—	—	FrameNumber[10:8] (R)		
7	6	5	4	3	2	1	0

The frame number of USB updated every time the SOF taken is received is displayed. You need to access FrameNumber_H and FrameNumber_L registers in pairs to get a frame number. At that time, access the FrameNumber_H register first.

Bit 7: **Fn_Invalid**
This is set to "1" when an error is found in the received SOF packet.

Bit [6:3]: **Reserved**

Bit [2:0]: **FrameNumber[10:8]**

Bit [7:0]: **FrameNumber[7:0]**
FrameNumber of the received SOF packet is displayed.

EP0 Max Size
USB[0x0C0] Default value = 0x40

—	EP0 Max Size[6:3]				—	—	—
7	6	5	4	3	2	1	0

Make the setting of the endpoint EP0.

Bit 7: **Reserved**

Bit [6:3]: **EP0MaxSize[6:3]**
Set MaxPacketSize of the endpoint EP0.
This endpoint is available by selecting any size from the following ones.
For FS: 8, 16, 32, 64 bytes
For HS: 64 bytes

Bit [2:0]: **Reserved**

EP0 Control
USB[0x0C1] Default value = 0x00

INxOUT	—	—	—	—	—	—	Reply Descriptor
7	6	5	4	3	2	1	0

Make the setting of the endpoint EP0.

Bit 7: **INxOUT**
Set a transfer direction of the endpoint EP0.
Determine the request received in the setuop stage and set a value to this bit.
If a data stage is available, set a transfer direction in the data stage to this bit. Completing the setup stage sets the ForceNAK bit of the EP0ControlIN and EP0ControlOUT registers. Clear it when you run the data stage or the status stage.
When you finish the data stage, update this bit according to the direction of the status stage. When the transfer direction of the data stage is IN, the status stage is in OUT direction. Set "0" to this bit. When the transfer direction of the data stage is OUT or when no data stage is available, the status stage is in IN direction. Clear FIFO of the endpoint EP0 and Set "1" to this bit.
Send NAK reply to IN or OUT transaction which is different from the set valu of this bit. However, send STALL reply if the ForceSTALL bit of the EP0ControlIN or EP0ControlOUT register corresponding to the transaction is set.

Bit [6:1]: **Reserved**

Bit 0: **ReplyDescriptor**

Execute the Descriptor return function.
When this bit is set to "1", send a reply to the IN transaction of the endpoint EP0 and return Descriptor data from FIFO by MaxPacketSize. The Descriptor points the data of the size set in the DescSize_H and L registers starting from the address set in the DescAdrs_H and L registers. As these set values are updated while the Descriptor return function is running, set them every time you set the ReplyDescriptor bit.
The DescAdrs_H and L registers are incremented by the number of times the data is sent while the DescSize_H and L registers are decremented by the number of times the data is sent per transaction.
When the amount of data set by DescSize_H and L has been sent for other transactions than IN transaction, the Descriptor return function terminates. Then, the ReplyDescriptor is cleared. The DescriptorCmp bit of the FIFO_IntStat register and the IN_TranACK bit of the EP0IntStat register are set to "1".
See the chapter on the operation description for more details.

EP0 Control IN
USB[0x0C2] Default value = 0x00

—	En ShortPkt	—	Toggle Stat(R)	Toggle Set(W)	Toggle Clr(W)	Force NAK	Force STALL
7	6	5	4	3	2	1	0

Set the operation and display the status related to the IN transaction of the endpoint EP0.

Bit 7: **Reserved**

Bit 6: **EnShortPkt**

You can send data less than MaxPacketSize in FIFO as short packets to the IN transaction of the endpoint EP0 by setting this bit to "1". When the IN transaction to which short packets have been sent is completed, this bit is automatically cleared. When a MaxPacketSize packet is sent, this bit is not cleared.
If you set this bit to "1" when there is no data in FIFO, you can send a zero-length packet to the IN token from the host. If you write data in the corresponding FIFO while you set this bit to send a packet, the data may be included and sent depending on the timing. Do not write data in FIFO until packet transmission finishes and this bit is cleared.

Bit 5: **Reserved**

Bit 4: **ToggleStat**

This indicates the status of the toggle sequence bit of the IN transaction of the endpoint EP0.

Bit 3: **ToggleSet**

Set the toggle sequence bit of the IN transaction of the endpoint EP0 to "1". When you set it at the same time as the ToggleClr bit, the function of the ToggleClr bit takes precedence.

Bit 2: **ToggleClr**

Clear the toggle sequence bit of the IN transaction of the endpoint EP0. When you set it at the same time as the ToggleSet bit, the function of this bit takes precedence.

Bit 1: **ForceNAK**

A NAK reply is sent to the IN transaction of the endpoint EP0 regardless of the amount of data in FIFO when you set this bit to "1".
When the RcvEP0SETUP bit of the MainIntStat register is set to "1" by the completion of the setup stage, this bit is set to "1" and cannot be cleared while the RcvEP0SETUP bit is "1". When the IN transaction to which short packets have been sent is completed, this bit is set to "1".
If a transaction is being executed, this bit is not set until the transaction finishes. As soon as it finishes, this bit is set to "1". If the transaction is not being executed, the bit is immediately set to "1".

Bit 0: **ForceSTALL**
A STALL reply is sent to the IN transaction of the endpoint EP0 when you set this bit to “1”. This bit takes precedence of the setting of the ForceNAK bit.
When the RcvEP0SETUP bit of the DeviceIntStat register is set to “1” by the completion of the setup stage, this bit is cleared and cannot be set to “1” while the RcvEP0SETUP bit is “1”.
If a transaction is being executed, this bit set after a certain period of time since the start of the transaction will be valid from the next transaction.

EP0 Control OUT
USB[0x0C3] Default value = 0x00

Auto ForceNAK	—	—	Toggle Stat(R)	Toggle Set(W)	Toggle Clr(W)	Force NAK	Force STALL
7	6	5	4	3	2	1	0

Set the operation and display the status related to the OUT transaction of the endpoint EP0.

Bit 7: **AutoForceNAK**
When the OUT transaction of the endpoint EP0 completes successfully, the ForceNAK bit of this register is set to “1”.

Bit [6:5]: **Reserved**

Bit 4: **ToggleStat**
This indicates the status of the toggle sequence bit of the OUTtransaction of the endpoint EP0.

Bit 3: **ToggleSet**
Set the toggle sequence bit of the OUT transaction of the endpoint EP0 to “1”. When you set it at the same time as the ToggleClr bit, the function of the ToggleClr bit takes precedence.

Bit 2: **ToggleClr**
Clear the toggle sequence bit of the OUT transaction of the endpoint EP0. When you set it at the same time as the ToggleSet bit, the function of this bit takes precedence.

Bit 1: **ForceNAK**
A NAK reply is sent to the OUT transaction of the endpoint EP0 regardless of the free space in FIFO when you set this bit to “1”.
When the RcvEP0SETUP bit of the DeviceIntStat register is set to “1” by the completion of the setup stage, this bit is set to “1” and cannot be cleared while the RcvEP0SETUP bit is “1”.
If a transaction is being executed, this bit is not set until the transaction finishes. As soon as it finishes, this bit is set to “1”. If the transaction is not being executed, the bit is immediately set to “1”.

Bit 0: **ForceSTALL**
A STALL reply is sent to the OUT transaction of the endpoint EP0 when you set this bit to “1”. This bit takes precedence of the setting of the ForceNAK bit.
When the RcvEP0SETUP bit of the DeviceIntStat register is set to “1” by the completion of the setup stage, this bit is cleared and cannot be set to “1” while the RcvEP0SETUP bit is “1”.
If a transaction is being executed, this bit set after a certain period of time since the start of the transaction will be valid from the next transaction.

EP0 Join

USB[0x0C5] Default value = 0x00

—	Join FIFO_Stat	—	—	Join DMA1	Join DMA0	Join CPU_Rd	Join CPU_Wr
7	6	5	4	3	2	1	0

Specify a port to transfer data to the endpoint 0.

Bit 7: **Reserved**

Bit 6: **JoinFIFO_Stat**
Enable the Full and Empty status of FIFO of the endpoint EP0 to be monitored in FIFO_IntStat.FIFO_Full, FIFO_IntStat.FIFO_Empty and FIFO_IntStat.FIFO_NotEmpty.

Bit [5:4]: **Reserved**

Bit 3: **JoinDMA1**
Transfer DMA1 in FIFO of the endpoint EP0. The transfer direction depends on the setting of the DMA1_Control.Dir bit.

Bit 2: **JoinDMA0**
Transfer DMA0 in FIFO of the endpoint EP0. The transfer direction depends on the setting of the DMA0_Control.Dir bit.

Bit 1: **JoinCPU_Rd**
Read and transfer the CPU register access in FIFO of the endpoint EP0. That is, data is read from FIFO of this endpoint when the FIFO_Rd_H,L register or the FIFO_ByteRd register is read.

Bit 0: **JoinCPU_Wr**
Write and transfer the CPU register access in FIFO of the endpoint EP0. That is, data is written in FIFO of this endpoint when the FIFO_Wr_H,L register is rewritten.

When you set the JoinDMAx{x=0 to 1} bit, you can refer to the remaining amount of data when the DMAx_Control.Dir is “1” while you can refer to the free space when it is “0” with the DMAx{x=0 to 1}_Remain_H,L register.

When you set the JoinCPU_Rd and JoinCPU_Wr bits, you can refer to FIFO_RdRemain_H,L and FIFO_WrRemain_H,L and read or write data from/to FIFO_Rd_H,L, FIFO_ByteRd and FIFO_Wr_H,L registers.

You can set any one of JoinDMA1, JoinDMA0, JoinCPU_Rd and JoinCPU_Wr bits to “1” at the same time. If you set “1” to multiple bits at the same time, the high-order bit is assumed to be valid.

EPa Max Size L
USB[0x0D0] Default value = 0x00

7	6	5	4	3	2	1	0
MaxSize[7:3]					—	—	—

EPb Max Size L
USB[0x0D8] Default value = 0x00

7	6	5	4	3	2	1	0
MaxSize[7:3]					—	—	—

EPc Max Size L
USB[0x0E0] Default value = 0x00

7	6	5	4	3	2	1	0
MaxSize[7:3]					—	—	—

EPd Max Size L
USB[0x0E8] Default value = 0x00

7	6	5	4	3	2	1	0
MaxSize[7:3]					—	—	—

EPe Max Size L
USB[0x0F0] Default value = 0x00

7	6	5	4	3	2	1	0
MaxSize[7:3]					—	—	—

EPf Max Size L
USB[0x0F8] Default value = 0x00

7	6	5	4	3	2	1	0
MaxSize[7:3]					—	—	—

EPg Max Size L
USB[0x100] Default value = 0x00

7	6	5	4	3	2	1	0
MaxSize[7:3]					—	—	—

EPh Max Size L
USB[0x108] Default value = 0x00

7	6	5	4	3	2	1	0
MaxSize[7:3]					—	—	—

EPa Max Size H
USB[0x0D1] Default value = 0x00

7	6	5	4	3	2	1	0
—	—	—	—	—	—	MaxSize[9:8]	

EPb Max Size H
USB[0x0D9] Default value = 0x00

7	6	5	4	3	2	1	0
—	—	—	—	—	—	MaxSize[9:8]	

EPc Max Size H
USB[0x0E1] Default value = 0x00

7	6	5	4	3	2	1	0
—	—	—	—	—	—	MaxSize[9:8]	

EPd Max Size H
USB[0x0E9] Default value = 0x00

—	—	—	—	—	—	MaxSize[9:8]	
7	6	5	4	3	2	1	0

EPe Max Size H
USB[0x0F1] Default value = 0x00

—	—	—	—	—	—	MaxSize[9:8]	
7	6	5	4	3	2	1	0

EPf Max Size H
USB[0x0F9] Default value = 0x00

—	—	—	—	—	—	MaxSize[9:8]	
7	6	5	4	3	2	1	0

EPg Max Size H
USB[0x101] Default value = 0x00

—	—	—	—	—	—	MaxSize[9:8]	
7	6	5	4	3	2	1	0

EPh Max Size H
USB[0x109] Default value = 0x00

—	—	—	—	—	—	MaxSize[9:8]	
7	6	5	4	3	2	1	0

Set MaxPacketSize.

Bit [7:3]: **MaxSize[7:3]**

Bit [2:0]: **Reserved**

Bit [7:2]: **Reserved**

Bit [1:0]: **MaxSize[9:8]**
Set MaxPacketSize of the endpoint EPx{a-h}.
To use this endpoint as bulk transfer,
For FS: 8, 16, 32, 64 bytes
For HS: 512 bytes
Set the value to either of the above.
To use this endpoint as interrupt transfer,
For HS: Up to 64 bytes
For HS: Up to 512 bytes
You can set any value within the range shown above.

EPa Configuration
USB[0x0D2] Default value = 0x00

INxOUT	IntEP_Mode	En Endpoint	—	EndpointNumber[3:0]			
7	6	5	4	3	2	1	0

EPb Configuration
USB[0x0DA] Default value = 0x00

INxOUT	IntEP_Mode	En Endpoint	—	EndpointNumber[3:0]			
7	6	5	4	3	2	1	0

EPc Configuration
USB[0x0E2] Default value = 0x00

INxOUT	IntEP_Mode	En Endpoint	—	EndpointNumber[3:0]			
7	6	5	4	3	2	1	0

EPd Configuration
USB[0x0EA] Default value = 0x00

INxOUT	IntEP_Mode	En Endpoint	—	EndpointNumber[3:0]			
7	6	5	4	3	2	1	0

EPe Configuration
USB[0x0F2] Default value = 0x00

INxOUT	IntEP_Mode	En Endpoint	—	EndpointNumber[3:0]			
7	6	5	4	3	2	1	0

EPf Configuration
USB[0x0FA] Default value = 0x00

INxOUT	IntEP_Mode	En Endpoint	—	EndpointNumber[3:0]			
7	6	5	4	3	2	1	0

EPg Configuration
USB[0x102] Default value = 0x00

INxOUT	IntEP_Mode	En Endpoint	—	EndpointNumber[3:0]			
7	6	5	4	3	2	1	0

EPh Configuration
USB[0x10A] Default value = 0x00

INxOUT	IntEP_Mode	En Endpoint	—	EndpointNumber[3:0]			
7	6	5	4	3	2	1	0

Set the endpoint EPx{a-h}.
Make settings so that a combination of EndpointNumber and INxOUT may not overlap other endpoints.

Bit 7: **INxOUT**

Set a transfer direction of the endpoint.

Bit 6: **IntEP_Mode**

Make the setting related to Interrupt transfer.
Do not set "1" to this bit at the bulk endpoint.
The setting of this bit varies depending on the direction (IN/OUT) of the endpoint (the direction of the endpoint is set by Bit7 "INxOUT").
Set the operation mode of the toggle sequence bit for IN direction (INxOUT = 1). The operation mode of the toggle sequence depends on the application. Select either one of operation modes for the Interrupt IN endpoint.

0: Normal toggle - Normal toggle sequence
1: Always toggle - Always toggle per transaction See section 5.7.5 in USB2.0 written standards for this mode.

Set whether or not to enable PING flow control at this endpoint for OUT direction (INxOUT = 0). Set this bit to "1" for Interrupt OUT endpoint.

0: Bulk OUT - Choose this setting for Bulk OUT endpoint.
1: Interrupt OUT - Choose this setting for Interrupt OUT endpoint.

Bit 5: **EnEndpoint**

Set this bit to "1" to enable this endpoint.
When this bit is "0", an access to the endpoint is ignored.
Make settings according to the SetConfiguration request from the host.

Bit 4: **Reserved**

Bit [3:0]: **EndpointNumber[3:0]**

Set any endpoint number from 0x1 to 0xF.

EPa Control
USB[0x0D4] Default value = 0x00

Auto ForceNAK 7	EnShort Pkt 6	DisAF_ NAK_Short 5	Toggle Stat(R) 4	Toggle Set(W) 3	Toggle Clr(W) 2	Force NAK 1	Force STALL 0

EPb Control
USB[0x0DC] Default value = 0x00

Auto ForceNAK 7	EnShort Pkt 6	DisAF_ NAK_Short 5	Toggle Stat(R) 4	Toggle Set(W) 3	Toggle Clr(W) 2	Force NAK 1	Force STALL 0

EPc Control
USB[0x0E4] Default value = 0x00

Auto ForceNAK 7	EnShort Pkt 6	DisAF_ NAK_Short 5	Toggle Stat(R) 4	Toggle Set(W) 3	Toggle Clr(W) 2	Force NAK 1	Force STALL 0

EPd Control
USB[0x0EC] Default value = 0x00

Auto ForceNAK 7	EnShort Pkt 6	DisAF_ NAK_Short 5	Toggle Stat(R) 4	Toggle Set(W) 3	Toggle Clr(W) 2	Force NAK 1	Force STALL 0

EPe Control
USB[0x0F4] Default value = 0x00

Auto ForceNAK	EnShort Pkt	DisAF_ NAK_Short	Toggle Stat(R)	Toggle Set(W)	Toggle Clr(W)	Force NAK	Force STALL
7	6	5	4	3	2	1	0

EPf Control
USB[0x0FC] Default value = 0x00

Auto ForceNAK	EnShort Pkt	DisAF_ NAK_Short	Toggle Stat(R)	Toggle Set(W)	Toggle Clr(W)	Force NAK	Force STALL
7	6	5	4	3	2	1	0

EPg Control
USB[0x104] Default value = 0x00

Auto ForceNAK	EnShort Pkt	DisAF_ NAK_Short	Toggle Stat(R)	Toggle Set(W)	Toggle Clr(W)	Force NAK	Force STALL
7	6	5	4	3	2	1	0

EPh Control
USB[0x10C] Default value = 0x00

Auto ForceNAK	EnShort Pkt	DisAF_ NAK_Short	Toggle Stat(R)	Toggle Set(W)	Toggle Clr(W)	Force NAK	Force STALL
7	6	5	4	3	2	1	0

Set the operation of the endpoint EPx{a-h}.

Bit 7: **AutoForceNAK**
When the transaction of the endpoint EP x{a-h} completes successfully, the ForceNAK bit of this register is set to "1".

Bit 6: **EnShortPkt**
You can send data less than MaxPacketSize in FIFO as short packets to the IN transaction of the endpoint EP x{a-h} by setting this bit to "1". When the IN transaction to which short packets have been sent is completed, this bit is automatically cleared. When a MaxPacketSize packet is sent, this bit is not cleared.
If you set this bit to "1" when there is no data in FIFO, you can send a zero-length packet to the IN token from the host. If you write data in the corresponding FIFO while you set this bit to send a packet, the data may be included and sent depending on the timing. Do not write data in FIFO until packet transmission finishes and this bit is cleared.

Bit 5: **DisAF_NAK_Short**
Enable/disable the Auto Force NAK Short (hereafter, AF_NAK_Short*) function.
When a normal OUT transaction completes and the received packet is a short packet, the ForceNAK bit is automatically set to "1".
The AF_NAK_Short function is valid for the default setting.
If this bit is set to "1", the AF_NAK_Short function is disabled.
When the AutoForceNAK bit is set to "1", the AutoForceNAK bit takes precedence.

Bit 4: **ToggleStat**
This indicates the status of the toggle sequence bit of the endpoint EP x{a-h}.

Bit 3: **ToggleSet**
Set the toggle sequence bit of the endpoint EP x{a-h}. When you set it at the same time as the ToggleClr bit, the function of the ToggleClr bit takes precedence.

Bit 2: **ToggleClr**
Clear the toggle sequence bit of the endpoint EP x{a-h}. When you set it at the same time as the ToggleSet bit, the function of this bit takes precedence.

Bit 1: **ForceNAK**
A NAK reply is sent to the transaction of the endpoint EP x{a-h} regardless of the amount of data or the free space in FIFO when you set this bit to "1".
If a transaction is being executed, this bit is not set until the transaction finishes. As soon as it finishes, this bit is set to "1". If the transaction is not being executed, the bit is immediately set to "1".

Bit 0: **ForceSTALL**
A STALL reply is sent to the transaction of the endpoint EP x{a-h} when you set this bit to "1". This bit takes precedence of the setting of the ForceNAK bit.
If a transaction is being executed, this bit set after a certain period of time since the start of the transaction will be valid from the next transaction.

EPa Join
USB[0x0D5] Default value = 0x00

—	Join FIFO_Stat	—	—	Join DMA1	Join DMA0	Join CPU_Rd	Join CPU_Wr
7	6	5	4	3	2	1	0

EPb Join
USB[0x0DD] Default value = 0x00

—	Join FIFO_Stat	—	—	Join DMA1	Join DMA0	Join CPU_Rd	Join CPU_Wr
7	6	5	4	3	2	1	0

EPc Join
USB[0x0E5] Default value = 0x00

—	Join FIFO_Stat	—	—	Join DMA1	Join DMA0	Join CPU_Rd	Join CPU_Wr
7	6	5	4	3	2	1	0

EPd Join
USB[0x0ED] Default value = 0x00

—	Join FIFO_Stat	—	—	Join DMA1	Join DMA0	Join CPU_Rd	Join CPU_Wr
7	6	5	4	3	2	1	0

EPe Join
USB[0x0F5] Default value = 0x00

—	Join FIFO_Stat	—	—	Join DMA1	Join DMA0	Join CPU_Rd	Join CPU_Wr
7	6	5	4	3	2	1	0

EPf Join
USB[0x0FD] Default value = 0x00

—	Join FIFO_Stat	—	—	Join DMA1	Join DMA0	Join CPU_Rd	Join CPU_Wr
7	6	5	4	3	2	1	0

EPg Join
USB[0x105] Default value = 0x00

—	Join FIFO_Stat	—	—	Join DMA1	Join DMA0	Join CPU_Rd	Join CPU_Wr
7	6	5	4	3	2	1	0

EPh Join
USB[0x10D] Default value = 0x00

—	Join FIFO_Stat	—	—	Join DMA1	Join DMA0	Join CPU_Rd	Join CPU_Wr
7	6	5	4	3	2	1	0

Specify a port to transfer data to the endpoint EPx{a-h}.

Bit 7: **Reserved**

Bit 6: **JoinFIFO_Stat**
Enable the Full and Empty status of FIFO of the endpoint EP x{a-h} to be monitored in FIFO_IntStat.FIFO_Full, FIFO_IntStat.FIFO_Empty and FIFO_IntStat.FIFO_NotEmpty.

Bit [5:4]: **Reserved**

Bit 3: **JoinDMA1**
Transfer DMA1 in FIFO of the endpoint EP x{a-h}. The transfer direction depends on the setting of the DMA1_Control.Dir bit.

Bit 2: **JoinDMA0**
Transfer DMA0 in FIFO of the endpoint EP x{a-h}. The transfer direction depends on the setting of the DMA0_Control.Dir bit.

Bit 1: **JoinCPU_Rd**
Read and transfer the CPU register access in FIFO of the endpoint EP x{a-h}. That is, data is read from FIFO of this endpoint when the FIFO_Rd_H,L register or the FIFO_ByteRd register is read.

Bit 0: **JoinCPU_Wr**
Write and transfer the CPU register access in FIFO of the endpoint EP x{a-h}. That is, data is written in FIFO of this endpoint when the FIFO_Wr_H,L register is rewritten.

When yoiu set the JoinDMAx{x=0 to 1} bit, you can refer to the remaining amount of data when the DMAx_Control.Dir is “1” while you can refer to the free space when it is “0” with the DMAx{x=0 to 1}_Remain_H,L register.

When you set the JoinCPU_Rd and JoinCPU_Wr bits, you can refer to FIFO_RdRemain_H,L and FIFO_WrRemain_H,L and read or write data from/to FIFO_Rd_H,L, EPnCHnFIFO_ByteRd and FIFO_Wr_H,L registers.

You can set any one of JoinDMA1, JoinDMA0, JoinCPU_Rd and JoinCPU_Wr bits to “1” at the same time. If you set “1” to multiple bits at the same time, the high-order bit is assumed to be valid.

A2.5.2.7 USB FIFO Setting Register

EPa Start Address L
USB[0x110] Default value = 0x00

Start Adrs[7:2]						—	—
7	6	5	4	3	2	1	0

EPb Start Address L
USB[0x112] Default value = 0x00

Start Adrs[7:2]						—	—
7	6	5	4	3	2	1	0

EPc Start Address L
USB[0x114] Default value = 0x00

Start Adrs[7:2]						—	—
7	6	5	4	3	2	1	0

EPd Start Address L
USB[0x116] Default value = 0x00

Start Adrs[7:2]						—	—
7	6	5	4	3	2	1	0

EPe Start Address L
USB[0x118] Default value = 0x00

Start Adrs[7:2]						—	—
7	6	5	4	3	2	1	0

EPf Start Address L
USB[0x11A] Default value = 0x00

Start Adrs[7:2]						—	—
7	6	5	4	3	2	1	0

EPg Start Address L
USB[0x11C] Default value = 0x00

Start Adrs[7:2]						—	—
7	6	5	4	3	2	1	0

EPh Start Address L
USB[0x11E] Default value = 0x00

Start Adrs[7:2]						—	—
7	6	5	4	3	2	1	0

EPa Start Address H
USB[0x111] Default value = 0x00

—	—	—	Start Adrs[12:8]				
7	6	5	4	3	2	1	0

EPb Start Address H
USB[0x113] Default value = 0x00

—	—	—	Start Adrs[12:8]				
7	6	5	4	3	2	1	0

EPc Start Address H

USB[0x115] Default value = 0x00

7	6	5	4	3	2	1	0
–	–	–	Start Adrs[12:8]				

EPd Start Address H

USB[0x117] Default value = 0x00

7	6	5	4	3	2	1	0
–	–	–	Start Adrs[12:8]				

EPe Start Address H

USB[0x119] Default value = 0x00

7	6	5	4	3	2	1	0
–	–	–	Start Adrs[12:8]				

EPf Start Address H

USB[0x11B] Default value = 0x00

7	6	5	4	3	2	1	0
–	–	–	Start Adrs[12:8]				

EPg Start Address H

USB[0x11D] Default value = 0x00

7	6	5	4	3	2	1	0
–	–	–	Start Adrs[12:8]				

EPh Start Address H

USB[0x11F] Default value = 0x00

7	6	5	4	3	2	1	0
–	–	–	Start Adrs[12:8]				

Set the area in FIFO used for EP x{a-h}.

Bit [7:2] : **StartAdrs[7:2]**

Bit [1:0] : **Reserved**

Bit [7:5] : **Reserved**

Bit [4:0] : **StartAdrs[12:8]**

Set the initial address of FIFO allocated to the endpoint EP x{a-h}.
The address value is specified in 4 bytes as it is set in the higher-order 12-2 bits.
The area up to one byte before the address set in the initial address of the next EP can be allocated to the channel EP x{a-g}. The area up to one byte before the address set in EP_EndAdrs can be allocated to EPh.
After setting EP x{a-h} StartAdrs, be sure to set “1” to the EP x{a-h}FIFO_Clr bit of the EPrFIFO_Clr register and clear FIFO of the endpoint EP x{a-h}.
If MaxSize of EP x{a-h} is greater than the area set here, the device does not operate properly.
Set the value so that the total of FIFO area, EP0 area, descriptor area, CSW area and CSW area obtained at all endpoints may not exceed the total of the built-in RAM.

EP End Address L
USB[0x120] Default value = 0x00

7	6	5	4	3	2	1	0
End Adrs[7:2]						–	–

EP End Address H
USB[0x121] Default value = 0x12

7	6	5	4	3	2	1	0
–	–	–	End Adrs[12:8]				

Set the area of the endpoint FIFO.

Bit [7:2]: **EndAdrs[7:2]**

Bit [1:0]: **Reserved**

Bit [7:5]: **Reserved**

Bit [4:0]: **EndAdrs[12:8]**
Set the end address of FIFO allocated to the endpoint EPh.
The address value is specified in 4 bytes as it is set in the higher-order 12-2 bits.
The last address of the loaded FIFO RAM is set to the default when it is reset.
If MaxSize of the endpoint is greater than the area set here, the device does not operate properly.
Set the value so that the total of FIFO area, EP0 area, descriptor area, CSW area and CSW area obtained at all endpoints may not exceed the total of the built-in RAM.
The allocated address of this register is not changed by the change of the number of endpoints.
Though the reset default is the address value indicating the size of the FIFO area, it can be changed.

Descriptor Address L
USB[0x124] Default value = 0x00

7	6	5	4	3	2	1	0
Desc Adrs[7:0]							

Descriptor Address H
USB[0x125] Default value = 0x00

7	6	5	4	3	2	1	0
–	–	–	–	Desc Adrs[11:8]			

Specify the descriptor address.

Bit [7:0]: **DescAdrs[7:0]**

Bit [7:4]: **Reserved**

Bit [3:0]: **DescAdrs[11:8]**
Specify the initial address of FIFO to start the Descriptor return operation in the Descriptor return finction.
Descriptor Address does not allocate FIFo area to the Descriptor return function. Descriptor Address can specify the whole area of FIFO RAM.
DescAdrs is updated for the amount of sending data every time an IN taransaction at the endpoint EP0 is completed for Descriptor return. See the section of ReplyDescriptor of the EP0Control register for the Descriptor return function.
As you can't explicitly allocate FIFO area for the Descriptor return function, avoid overlapping with FIFO of other endpoints by specifying the DescAdrs_H,L and DescSize_H,L registers. The area from the end address (0x40) of the reserved area of the endpoint EP0 to the initial address (0x190) of the CBW area is appropriate.
To refer to Desriptor Address, read DescAdrs_H and DescAdrs_L in sequence.

Descriptor Size L
USB[0x126] Default value = 0x00

Desc Size[7:0]							
7	6	5	4	3	2	1	0

Descriptor Size H
USB[0x127] Default value = 0x00

–	–	–	Desc Size[12:8]				
7	6	5	4	3	2	1	0

Specify the descriptor size.

Bit [7:0]: **DescSize[7:0]**

Bit [7:5]: **Reserved**

Bit [1:0]: **DescSize[12:8]**
Specify the total amount of data to be returned by the Descriptor return function for Descriptor Size. See the section of ReplyDescriptor bit of the EP0Control register for the Descriptor return function.
You can specify a value from “0x000” to “0x1FFF” for Descriptor Size regardless of the size of FIFO and the area setting. DescSize is updated for the amount of sending data every time an IN taransaction at the endpoint EP0 is completed for Descriptor return.
As you can’t explicitly allocate FIFO area for the Descriptor return function, avoid overlapping with FIFO of other endpoints byb specifying the DescAdrs_H,L and DescSize_H,L registers. Use the area from the end address (0x40) of the reserved area of the endpoint EP0 to the initial address (0x190) of the CBW.
To refer to Desriptor Size, read DescSize_H and DescSize_L in sequence.

DMA0 FIFO Control
USB[0x128] Default value = 0x00

FIFO_ Running(R)	AutoEn Short	–	–	–	–	–	–
7	6	5	4	3	2	1	0

DMA1 FIFO Control
USB[0x12A] Default value = 0x00

FIFO_ Running(R)	AutoEn Short	–	–	–	–	–	–
7	6	5	4	3	2	1	0

Display and set the status of FIFO to transfer DMA0/1.

Bit 7 : **FIFO_Running**
This indicates that FIFO of the endpoint connected to DMA0/1 is operating. This is set to “1” when DMA0/1 is started and it is cleared when FIFO gets empty after DMA0/1 finishes.

Bit 6 : **AutoEnShort**
If the amount of data less than the max packet size remains in FIFO when DMA0/1 finishes, the EnShortPkt bit of the endpoint is set to “1”.
This is valid when the endpoint connected to DMA0/1 is in IN direction.

Bit [5:0] : **Reserved**

FIFO Rd 0
USB[0x130] Default value = 0xXX

FIFO Rd0[7:0](R)							
7	6	5	4	3	2	1	0

FIFO Rd 1
USB[0x131] Default value = 0xXX

FIFO Rd1[7:0] (R)							
7	6	5	4	3	2	1	0

Bit [7:0]: **FIFO_Rd0[7:0]**

Bit [7:0]: **FIFO_Rd1[7:0]**

You can read FIFO data of the endpoint whose EPx{x=0,a to h}Join.JoinCPU_Rd bit is set.
You can read FIFO data by accessing either FIFO_Rd_H or L register in 8 bit mode.
If you read this register when FIFO has byte boundary in 16 bit mode, valid data is outputted to only one side. For details, see the functional description “Rounding of FIFO Access”.
To read FIFO data using this register, be sure to check the amount of read-enabled data with the FIFO_RdRemain_H and L registers before reading.

FIFO Wr 0
USB[0x132] Default value = 0xXX

FIFO Wr0[7:0](W)							
7	6	5	4	3	2	1	0

FIFO Wr 1
USB[0x133] Default value = 0xXX

FIFO Wr1[7:0] (W)							
7	6	5	4	3	2	1	0

Bit [7:0]: **FIFO_Wr0[7:0]**

Bit [7:0]: **FIFO_Wr1[7:0]**

You can write data to FIFO of the endpoint whose EPx{x=0,a to h}Join.JoinCPU_Wr bit is set.
You can write data to FIFO by accessing either FIFO_Wr_H or L register in 8 bit mode.
If you write data to this register when FIFO has byte boundary in 16 bit mode, data is written to only one side. For details, see the functional description “Rounding of FIFO Access”.
To wwrite data to FIFO using this register, be sure to check the amount of write-enabled data with the FIFO_WrRemain_H and L registers before writing.

FIFO Rd Remain L
USB[0x134] Default value = 0x00

RdRemain[7:0](R)							
7	6	5	4	3	2	1	0

FIFO Rd Remain H
USB[0x135] Default value = 0x00

RdRemain Valid(R)	–	–	RdRemain[12:8] (R)				
7	6	5	4	3	2	1	0

Bit [7:0]: **RdRemain[7:0]**

Bit 7: **RdRemainValid**
This is set to "1" when the endpoint is joined to CPU I/F by the EPx{x=0,a to h}Join.JoinCPU_Rd bit and the value of FIFO_RdRemain is valid. The value of RdRemain when this bit is cleared is invalid.

Bit [6:5]: **Reserved**

Bit [4:0]: **RdRemain[12:8]**
It indicates the amount of read-enabled data in FIFO at the endpoint connected to CPU I/F by the EPx{x=0,a to h}Join.JoinCPU_Rd bit. You need to access data with FIFO_RdRemain_H and FIFO_RdRemain_L registers in pairs to get read-enabled data in FIFO. At that time, access the FIFO_RdRemain_H register first.

FIFO Wr Remain L
USB[0x136] Default value = 0x00

WrRemain[7:0] (R)							
7	6	5	4	3	2	1	0

FIFO Wr Remain H
USB[0x137] Default value = 0x00

–	–	–	WrRemain[12:8] (R)				
7	6	5	4	3	2	1	0

Bit [7:0]: **WrRemain[7:0]**

Bit [7:5]: **Reserved**

Bit [4:0]: **WrRemain[12:8]**
It indicates the free space of the channel FIFO connected to CPU I/F by the endpoint FIFO connected to COPU I/F by the EPx{x=0,a to h}Join.JoinCPU_Wr bit. Note that you can't check the exact free space of FIFO immediately after writing to FIFO. Check the free space in FIFO at an interval of at least one CPU cycle. You need to access data with FIFO_WrRemain_H and FIFO_WrRemain_L registers in pairs to get the free space in FIFO. At that time, access the FIFO_WrRemain_H register first.

FIFO Byte Rd
USB[0x138] Default value = 0xXX

FIFO ByteRd[7:0] (R)							
7	6	5	4	3	2	1	0

Bit [7:0]: **FIFO_ByteRd[7:0]**
You can read FIFO data (in byte) of the endpoint whose EPx{x=0,a to h}Join.JoinCPU_Rd bit is set. To read FIFO data using this register, be sure to check the amount of read-enabled data with the FIFO_RdRemain_H and L registers before reading.

RAM Rd Adrs L

USB[0x140] Default value = 0x00

RAM Read Address[7:2]						–	–
7	6	5	4	3	2	1	0

RAM Rd Adrs H

USB[0x141] Default value = 0x00

–	–	–	RAM Read Address[12:8]				
7	6	5	4	3	2	1	0

Bit [7:2]: **RAM_RdAdrs[8:2]**

Bit [1:0]: **Reserved**

Bit [7:5]: **Reserved**

Bit [4:0]: **RAM_RdAdrs[12:8]**

Set the initial address for RAM_Rd. After setting this register, set the RAM_RdCount register and set the bits of the RAM_RdControl register. The RAM_Rd function starts. The value of this register changes according to the internal operation while the RAM_Rd function operates. Therefore, after you have set bits of the RAM_RdControl and started the RAM_Rd function, do not read the value of this register until the CPU_IntStat.RAM_RdCmp bit is set. If this register is read when the RAM_Rd function is operating, the value is not guranteed. Note that rewriting the register during operation of the RAM_Rd function can cause malfunction.

RAM Rd Control

USB[0x142] Default value = 0x00

RAM_GoRd CBW_CSW	RAM_GoRd	–	–	–	–	–	–
7	6	5	4	3	2	1	0

Bit 7: **RAM_GoRdCBW_CSW**

This is the bit to start the RAM_Rd function to read data that has been received in the CBW area.
Writing "1" to this bit starts the RAM_Rd function and reads data from the CBW area. When the value of the RAM_Rd_00 to RAM_Rd_1E register becomes valid, the CPU_IntStat.RAM_RdCmp bit is set to "1" and this bit is automatically cleared.
You don't need to set the RAM_RdAdrs_H,L and RAM_RdCount registers.
When you set it at the same time as theRAM_GoRd bit, the function of this bit takes precedence.

Bit 6: **RAM_GoRd**

This is the bit to start the RAM_Rd function.
When you set the initial address to activate RAM_Rd to the RAM_RdAdrs_H,L registers and then the RAM_RdCount register and write "1" to this bit, the RAM_Rd function starts. When data is read from the specified initial address for the specified count and the value of the RAM_Rd_xx{xx=00 to 1F} register becomes valid, the CPU_IntStat.RAM_RdCmp bit is set to "1" and this bit is automatically cleared.
When you set it at the same time as the RAM_GoRdCBW_CSW bit, the function of the RAM_GoRdCBW_CSW bit takes precedence.

Bit [5:0]: **Reserved**

RAM Rd Count
USB[0x143] Default value = 0x00

7	6	5	4	3	2	1	0
–	–	RAM Rd Count[5:2]				–	–

Bit [7:6]: **Reserved**

Bit [5:2]: **RAM_RdCount [5:2]**
Set the amount of data to be read in the RAM_Rd_xx{xx=00 to 1F} register using the RAM_Rd function. After setting RAM_RdAdrs_H,L registers, set this register and set the bit of the RAM_RdControl register to start the RAM_Rd function. The value of this register changes according to the internal operation while the RAM_Rd function operates. Therefore, after you have set bits of the RAM_RdControl and started the RAM_Rd function, do not read the value of this register until the CPU_IntStat.RAM_RdCmp bit is set. If this register is read when the RAM_Rd function is operating, the value is not guranteed. Note that rewriting the register during operation of the RAM_Rd function can cause malfunction.
You can set up to 32 bytes to this register. Note that setting data exceeding 32 bytes can cause malfunction.

Bit [1:0]: **Reserved**

RAM Wr Address L
USB[0x144] Default value = 0x00

7	6	5	4	3	2	1	0
RAM WrAdrs[7:0]							

RAM Wr Address H
USB[0x144] Default value = 0x00

7	6	5	4	3	2	1	0
–	–	–	RAM WrAdrs[12:8]				

Specify an address to write to RAM by the RAM_WrDoor_H,L registers.

Bit [7:0]: **RAM_WrAdrs[7:0]**

Bit [7:5]: **Reserved**

Bit [4:0]: **RAM_WrAdrs[12:8]**
Specify an address to write to RAM. The address is incremented according to the number of bytes written to RAM_WrDoor_H and L registers. As it is hard to check correct AM_WrAdrs immediately after writing data to RAM_WrDoor_H and L registers, check RAM_WrAdrs at an interval of at least one CPU cycle. See the section of RAM_WrDoor_H and L registers for writing data.
To refer to RAM_WrAdrs, read RAM_WrAdrs_H and RAM_WrAdrs_L in sequence.

RAM Wr Door 0

USB[0x146] Default value = 0xXX

RAM WrDoor0[7:0](W)							
7	6	5	4	3	2	1	0

RAM Wr Door 1

USB[0x147] Default value = 0xXX

RAM WrDoor1[7:0] (W)							
7	6	5	4	3	2	1	0

Bit [7:0]: **RAM_WrDoor0[7:0]**

Bit [7:0]: **RAM_WrDoor1[7:0]**

This is an access register to write data to RAM. This is the write-only register.

Set the initial address to write RAM data to the RAM_WrAdrs_H and L registers before starting writing. Then, when you write data to this register, RAM_WrAdrs_H and L registers will be automatically incremented according to the number of bytes to be written in sequence.

You can write data to the descriptor area and the CSW area with RAM_WrDoor_H and L registers in the device mode. You can use data written in the descriptoer area with RAM_WrDoor_H and L registers as many times as you want by the ReplyDescriptor function. That is, the data is never erased or overwritten by the Descriptor return function. Note that the data may be overwritten if the area in which Descriptor data is written overlaps the area acquired by other endpoints.

RAM Rd 00 to RAM Rd 1F

USB[0x150] to USB[0x16F] Default value = 0x00

RAM Rd 00[7:0](R) to RAM Rd 1F[7:0] (R)							
7	6	5	4	3	2	1	0

Bit [7:0]: **RAM_Rd_xx[7:0]**

This is the register to store the data read from RAM using the RAM_Rd function. Set RAM_RdAdrs_H and L registers and RAM_RdCount register and start the RAM_Rd function using the bit of the RAM_RdControl register. When the value of this register is enabled, the FIFO_IntaStat.RAM_RdCmp bit is set to “1”. When the value set in the RAM_RdCount register is less than 32 bytes, the data read from RAM is stored from RAM_Rd_00 in sequence. Registers exceeding the count set in the RAM_RdCount register (for example, if the count is set to “16”, RAM_Rd_10 to RAM_Rd_1F) will be disabled.

A2.6 Functional description

A2.6.1 Default setting

Make the following default setting to use the USB controller.

A2.6.1.1 Access setting to the USB controller

The following default setting is required to access this USN controller.

① Selection of APB bus Wait
Set the access weight (+2wait or more) from CPU to the USB controller to bit[31:30] of the PB WAIT2n register (APBWAIT2) of the APB bridge.
Address 0xFFFE_0008/bit[31:30] = 10

② I/O clock On (CPU⇔USB controller)
Write "1" to bit1 of the IO Clock Control Register (IOCLKCTL) of the system controller.
ADDRESS: 0xFFFF_D014/bit[1] = 1

③ Clock inside the USB controller On
Enable the clock used in the aUSB controller.
The register in the USB controller operates in 60MHz which is one eighths frequency of PLL480 buil in the USB controller by writing "1" to bit1 of the IO Clock Control Register (MISC) of the system controller.
Address: 0xFFFF_D06C/bit[1] = 1

bit[1]	:	1	60MHz ON
		0	60MHz OFF

A2.6.1.2 Interrupt setting

Interrupts from this USB controller are allocated to the normal interrupt IRQ29 of the interrupt controller (INTC). Interrupt signals from the USB controller are outputted in low active level signals. As they are logically inverted in the IC when INTC is inputted, they are handles as high active level signals for the setting of INTC.

① Level register setting
Set "0" to bit[29] of the IRQxx level register of the interrupt controller.
Address: 0xFFFF_F0A0/bit[29] = 0

② Polarity register setting
Set "1" to bit[29] of the IRQxx polarity register of the interrupt controller.
Address: 0xFFFF_F0A4/bit[29] = 1

A2.6.1.3 Notes on Macro Config1[0x37]

This register is optimized for this LSI. Do not change the setting unless required.

bit[7]	0 : Negative logic
bit[6]	0 : 1/0 mode
bit[5]	0 : Negative logic
bit[4]	0 : Negative logic
bit[3]	0 : Operates as a valid DMA access when XDACK0 and 1 are asserted.
bit[2]	1 : The even number address shall be the upper side and the odd number address shall be the lower side.
bit[1:0]	1x : 16bit BE mode (fixed)

This is in Little Endian format in the initial condition by the setting of this register (bit[2]).

A2.6.2 USB Device Control

This section describes the USB device function.

A2.6.2.1 Endpoint

This LSI has an endpoint for control transfer (EP0) and eight general-purpose endpoints (EPa-h). Endpoints EPa-h can be used as endpoints for bulk or interrupt transfer separately.

The hardware of LSI provides endpoints and controls transactions. On the other hand, it does not provide the control function of the interface defined in USB (hereafter, USB definition interface). Implement the USB definition interface as firmware. Set endpoints according to the descriptor definition specific to the device , combine them and configure the USB definition interface.

Each endpoint has fixed basic setting items determined by the USB definition interface and variable control items and statuses controlled for every transfer. Set basic setting items when you initialize the chip or when you change the USB definition interface.

Table A2.6.1 shows basic setting items of the end point EP0 (default control pipe)

The endpoint EP0 shares the register set and FIFO area in IN and OUT directions. Set the direction of the data transaction as required by the firmware in the data stage and status stage at the endpoint EP0 before execution.

Table A2.6.1 Basic setting items of the end point EP0

Item	Register/bit	Description
Max packet size	EP0MaxSize	Set the max packet size to any value of 8, 16, 32 and 64 in FS operation. Set it to 64 in HS operation. Sixty four byte area is allocated to the endpoint EP0 from the address 0 of FIFO.

Table A2.6.2 shows basic setting items of the end point (EPa-h). As you can arbitrarily set a transaction number and endpoint number to endpoints EPa-h, you can use up to eight independent endpoints. Configure the USB definition interface by setting it according to the definition and enabling the setting.

Set the FIFO area of endpoints EPa-h by defining the start and end addresses.

Table A2.6.2 Basic setting items of the universal point

Item	Register/bit	Description
Transaction direction	EPx(x=a-h)Config.INxOUT	Set a transfer direction of each endpoint.
Max packet size	EPx(x=a-h)MaxSize_H EPx(x=a-h)MaxSize_L	Set the max packet size of each endpoint to any value of 8, 16, 32, 64 and 512 bytes. However, set it to any value of 8, 16, 32 and 64 bytes in the FS mode and set it to 512 bytes in the HS mode for endpoints requiring bulk transfer.
Endpoint number	EPx(x=a-h)Config.EndpointNumber	Set the endpoint number of each endpoint to any value between 0x1 and 0xF2.
Toggle mode	EPx(x=a0h)Config.IntEP_Mode	Set the operation mode of interrupt transfer. Set it to “0” for the endpoint requiring bulk transfer regardless of the direction. Set the mode of the toggle sequence for the endpoint in IN direction. Set it to “0” for the endpoint for OUT transfer to carry out interrupt transfer.
Endpoint enabled	EPx(x=a-h)Config.EnEndpoint	Enable each endpoint. Set this when the USB definition interface using the endpoint is enabled.
FIFO area	EPx(x=a-h)StartAdrs_H EPx(x=a-h)StartAdrs_L EP_EndAdrs_H EP_EndAdrs_L	Set the area allocated to each endpoint in FIFO address. Allocate area equal to or more than the max packet size of each channel to the FIFO area. The size of the FIFO area has an impact on the throughput of data transfer. See the FIFO section of the functional description for the detail of FIFO area allocation.

A2.6.2.2 Transaction

LSI provides the function to execute transactions and the interface to execute transactions for the firmware. The interface for the firmware is implemented as the control register, status register and interrupt signals asserted by the status register. See the chapter on the register description for the setting to assert interruptions by the status.

LSI issues a status for the firmware for each transaction. However, the firmware does not need to control each transaction. LSI refers to FIFO before responding to a transaction, determines whether or not to transfer data based on the amount of data or the free space and automatically process data.

For example, the firware can read data from FIFO through CPU interface (DMA read or register read) to create free space in FIFO and automatically process OUT transactions in series for OUT endpoints. On the other hand, the firware can write data to FIFO through CPU interface (DMA write or register write) to create valid data in FIFO and automatically process IN transactions in series for IN endpoints.

Table A2.6.3 shows control items ans statuses related to transaction control of the endpoint EP0.

Table A2.6.3 Control items and statuses of the end point EP0

Item	Register/bit	Description
Transaction direction	EP0Control.INxOUT	Set a transfer direction in the data stage and the status stage.
Descriptor Return Enabled	EP0Control.ReplyDescriptor	Start the automatic response of the descriptor.
Descriptor Return Address	DescAdrs_H DescAdrs_L	Specify the initial address on FIFO of the data to be returned by automatic response of the descriptor.
Descriptor size	DescSize_H DescSize_L	Specify the amount of data to be returned by automatic response of the descriptor.
Control prohibited	SETUP_Control.ProtectEP0	When this bit is set, access to ForceNAK and ForceSTALL bits of the EP0ControlIN and EP0ControlOUT registers is not allowed. This bit is set by H/W of LSI and can be cleared by accessing the register when the RcvEP0SETUP status is enabled.
Short packet transmission enable	EP0ControlIN.EnShortPkt	This enables short packet transmission less than the max packet size. This bit is cleared when the IN transaction that has sent the short packet is completed.
Toggle sequence bit	EP0ControlIN.ToggleStat EP0ControlOUT.ToggleStat	This indicates the status of the toggle sequence bit. It is automatically initialized by the SETUP stage.
Toggle set	EP0ControlIN.ToggleSet EP0ControlOUT.ToggleSet	Set a toggle sequence bit.
Toggle clear	EP0ControlIN.ToggleClr EP0ControlOUT.ToggleClr	Clear a toggle sequence bit.
Forced NAK response	EP0ControlIN.ForceNAK EP0ControlOUT.ForceNAK	A NAK response is sent to IN or OUT transactions (including PING) regardless of the amount of data or free space of FIFO.
STALL response	EP0ControlIN.ForceSTALL EP0ControlOUT.ForceSTALL	A STALL response is sent to IN or OUT transactions (including PING).
Automatic ForceNAK set	EP0ControlOUT.AutoForceNAK	The EP0ControlOUT.ForceNAK bit is set for every completion of OUT transactions.
Receiving SETUP status	DeviceIntStat.RcvEP0SETUP	This indicates that SETUP transactions have been executed.
Transaction status	EP0IntStat.OUT_ShortACK EP0IntStat.IN_TranACK EP0IntStat.OUT_TranACK EP0IntStat.IN_TranNAK EP0IntStat.OUT_TranNAK EP0IntStat.IN_TranErr EP0IntStat.OUT_TranErr	This shows the result of the transaction.
Descriptor return data stage end status	FIFO_IntStat.DescriptorCmp	This indicates that the data stage of the automatic descriptor response has finished.

Table A2.6.4 shows control items ans statuses related to transaction process of endpoints EPa-h.

Table A2.6.4 Control items and statuses of the universal point

Item	Register/bit	Description
Automatic ForceNAK set	EPx{x=a-h}Control.AutoForceNAK	The EPx{x=a-h}Control.ForceNAK bit of the endpoint is set for every completion of OUT transactions.
Short packet transmission enable	EP x{x=a-h}Control.EnShortPkt	This enables short packet transmission less than the max packet size for IN transactions. This bit is cleared when the IN transaction that has sent the short packet is completed.
Prohibition of automatic ForceNAK set by receiving a short packet	EPx{x=a-h}Control.DisAF_NAK_Short	The function(*) to automatically set the EPx{x=a-h}Control.ForceNAK bit of the endpoint is prohibited when a short packet is received for OUT transactions. *: This is enabled when it is not prohibited by this bit.
Toggle sequence bit	EP x{x=a-h}Control.ToggleStat	This indicates the status of the toggle sequence bit.
Toggle set	EP x{x=a-h}Control.ToggleSet	Set a toggle sequence bit.
Toggle clear	EP x{x=a-h}Control.ToggleClr	Clear a toggle sequence bit.
Forced NAK response	EP x{x=a-h}Control.ForceNAK	A NAK response is sent to transactions regardless of the amount of data or free space of FIFO.
STALL response	EP x{x=a-h}Control.ForceSTALL	A STALL response is sent to transactions.
Transaction status	EP x{x=a-h}IntStat.OUT_ShortACK EP x{x=a-h}IntStat.IN_TranACK EP x{x=a-h}IntStat.OUT_TranACK EP x{x=a-h}IntStat.IN_TranNAK EP x{x=a-h}IntStat.OUT_TranNAK EP x{x=a-h}IntStat.IN_TranErr EP x{x=a-h}IntStat.OUT_TranErr	This shows the result of the transaction.

A2.6.2.2.1 SETUP Transaction

Enforce SETUP transactions addressed to the endpoint EP0 of its node unconditionally. (The USB function needs to be enabled by the NegoControl.ActiveUSB bit.)

When a SETUP transaction is issued, all contents in the data packet (8Byte) are stored in EP0SETUP_0 - EP0SETUP_7 registers and a ACK response is returned. The RcvEP0SETUP status is issuet to the firmware except SetAddress() requests.

If an error occurs during a SETUP transaction, no response is sent and no status is issued.

When the SETUP transaction completes, the ForceNAK of the EP0ControlIN and EP0ControlOUT registers is set and the ForceSTALL bit is cleared. In addition, set the ToggleStat bit. Also, set the SETUP_Control.ProtectEP0 bit. When the firmware finished setting the endpoint EP0 and is ready to transfer to the stage, clear the SETUP_Control.ProtectEP0 bit and the ForceNAK in the corresponding direction of the EP0ControlIN and EP0ControlOUT registers.

Fig.A2.6.1 shows how the SETUP transaction looks. (a) The host issues a SETUP token to the endpoint 0 of this node. (b) The host goes on sending 8-byte length data packets. LSI writes the data to EP0SETUP_0 - EP0SETUP_7 registers. (c) LSI automatically sends an ACK response. Set a register for automatic setting and issue a status to the firmware.

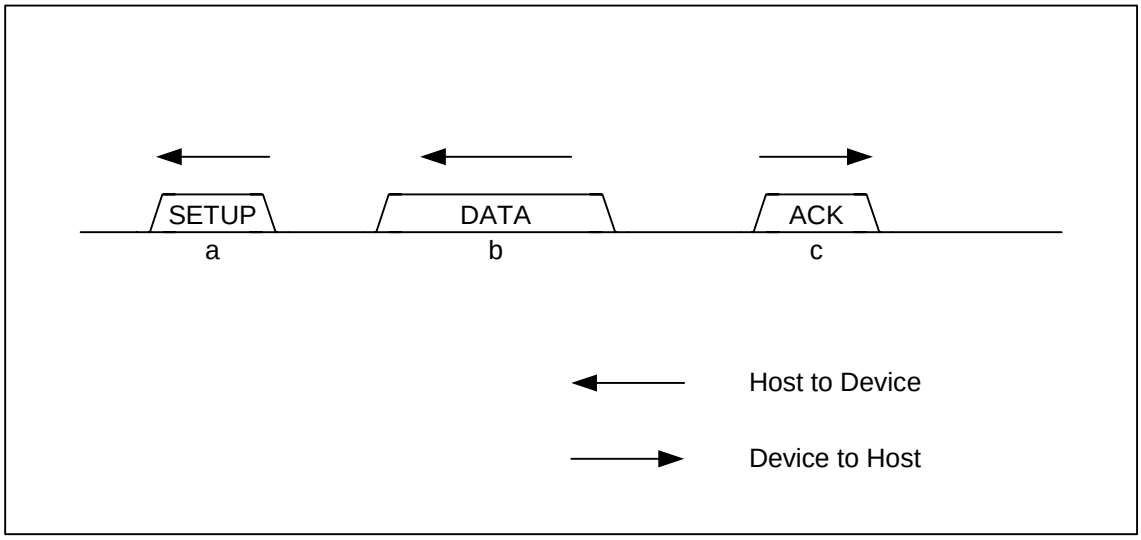


Fig.A2.6.1 SETUP Transaction

A2.6.2.2.2 Bulk/interrupt OUT transaction

Bulk and interrupt transactions start receiving data when the free space of FIFO is equal to or larger than the max packet size.

If bulk and interrupt transactions have successfully received all data, they complete and send an ACK response. The OUT_TranACK status (EPx{x=0,a-h}IntStat.OUT_TranACK bit) of the corresponding endpoint is issued to the firmware. It also updates FIFO and secures area regarding the data as received.

When all data in short packets are received for bulk and interrupt transactions, the OUT_ShortACK status (EPx{x=0,a-h}IntStat.OUT_ShortACK bit) is issued adding to the transaction completion process described above. When the EPx{x=a-h}Control.DisAF_NAK_Short bit is cleared, the EPx{x=a-h}ForceNAK bit of the endpoint is set.

When toggle mismatch occurs in bulk and interrupt OUT transactions, an ACK response is sent to the transaction but a status is not issued. FIFO is not updated.

When an error occurs in bulk and interrupt OUT transactions, no response is sent to the transaction. The OUT_TranErr status (EPx{x=0,a-h}IntStat.OUT_TranErr bit) is issued. FIFO is not updated.

If bulk and interrupt transactions have not received all data, they an ACK response to transactions. The OUT_TranNAK status (EPx{x=0,a-h}IntStat.OUT_TranNAK bit) is issued. FIFO is not updated.

Fig.A2.6.2 shows how the bulk or interrupt OUT transaction looks when it is completed. (a) The host issues an OUT token to the endpoint in OUT direction in this node. (b) The host goes on sending data packets less than the max packet size. LSI writes data to FIFO of the corresponding endpoint. (c) LSI automatically sends an ACK response when it receives data. Set a register for automatic setting and issue a status to the firmware.

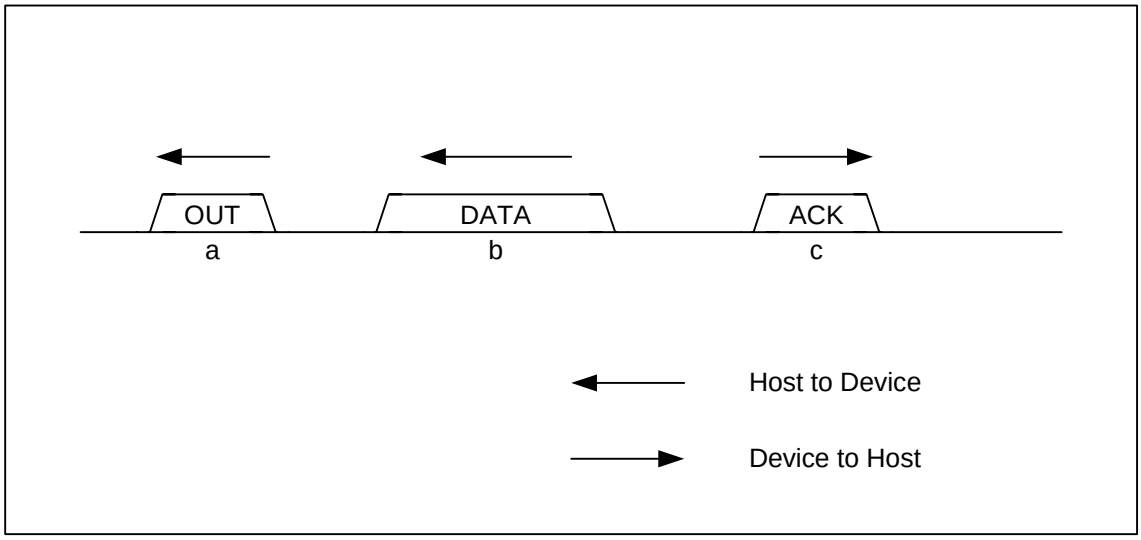


Fig.A2.6.2 OUT Transaction

A2.6.2.2.3 Bulk/Interrupt IN transaction

If FIFO has data as much as the max packet size at the bulk or interrupt endpoint in IN direction or short packet transmission is allowed by the firmware, a response is sent to the IN transaction and data packets are returned.

Transmission of short packets (including packets whose data length is zero) is allowed by setting the EP0ControlIN.EnShortPkt or EPx{x=a-h}Control.EnShortPkt bit. Do not write new data to FIFO of the endpoint until the transaction is completed after transmission is allowed to send a short packet.

When the IN transaction to send a short packet is completed at the endpoint EP0, the EP0ControlIN. ForceNAK bit is set.

When ACK is received by the IN transaction that received data, the transaction is completed and the IN_TranACK status (EPx{x=0,a-h}IntStat.IN_TranACK bit) is issued to the firmware. It also updates FIFO and releases the area regarding the data as sent.

When ACK is not received by the IN transaction that returned data, the transaction is assumed to be a failure and the IN_TranErr status (EPx{x=0,a-h}IntStat.IN_TranErr bit) is issued to the firmware. FIFO is not updated and the area is not released.

If FIFO doesn't have data as much as the max packet size at the bulk or interrupt endpoint in IN direction and short packet transmission is not allowed, a NAK response is sent to the IN transaction and the IN_TranNAK status (EPx{x=0,a-h}IntStat.IN_TranNAK bit) is issued to the firmware. FIFO is not updated and the area is not released.

Fig.A2.6.3 shows how the bulk or interrupt IN transaction looks when it is completed. (a) The host issues an IN token to the endpoint in IN direction in this node. (b) LSI sends a data packet less than the max packet size if it can respond to this IN transaction. (c) The host sends an ACK response. When LSI receives an ACK response, it sets a register for automatic setting and issues a status to the firmware.

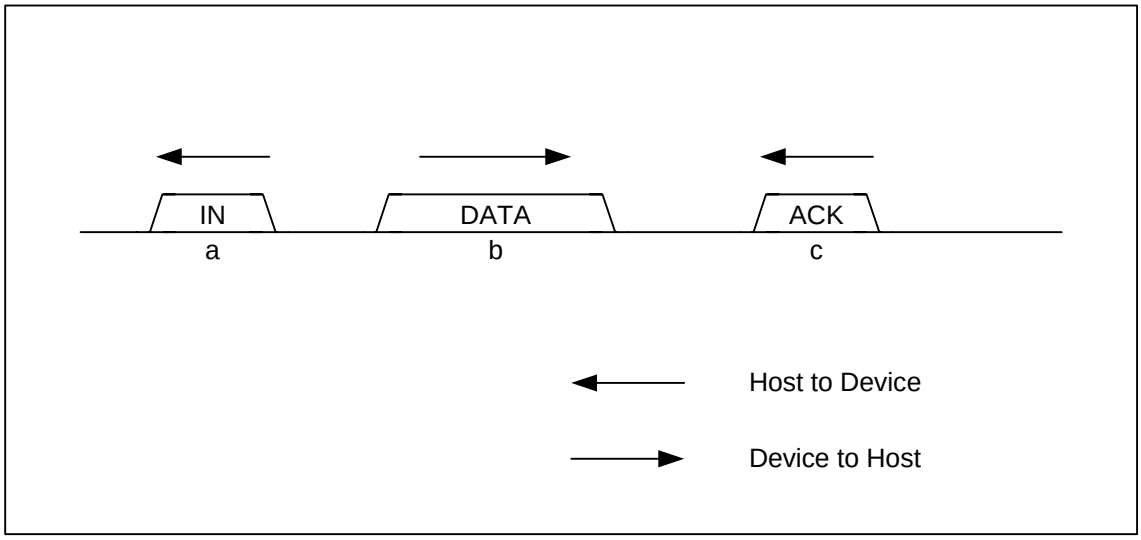


Fig.A2.6.3 IN Transaction

A2.6.2.2.4 PING Transaction

PING transactions are executed at the endpoint in OUT direction for bulk transfer during HS operation.

When the free space of FIFO at the corresponding endpoint is equal to or larger than the max packet size, an ACK response is sent to the PING transaction. No status is issued to the firmware.

When the free space of FIFO at the corresponding endpoint is less than the max packet size, an NAK response is sent to the PING transaction. The OUT_TranNAK status (EPx{x=0,a-h}IntStat.OUT_TranNAK bit) is issued to the firmware.

FIFO ios never updated for the PING transaction.

Fig.A2.6.4 shows how an ACK response is sent to the PING transaction. (a) The host issues an PING token to the endpoint in OUT direction in this node. When FIFO has free space as much as the max packet size, LSI sends an ACK response to this PING transaction. It also issues a status to the firmware.

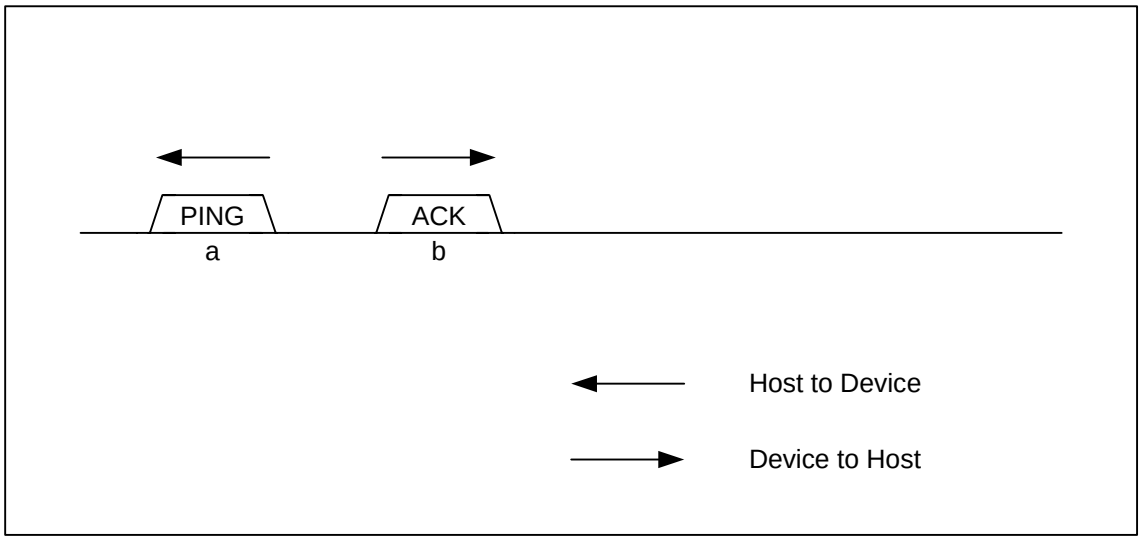


Fig.A2.6.4 PING Transaction

A2.6.2.3 Control Transfer

During control transfer at endpoint EP0, individual transactions are combined and controlled with together except for SetAddress() requests. The SetAddress() requests are processed automatically by the Automatic Address Setting function described later.

Fig.A2.6.5 shows control transfer when the data stage is in the OUT direction. (a) The host starts control transfer using the SETUP transaction. The device firmware analyzes the contents of request and prepares to respond to the data stage. (b) The host issues an OUT transaction and executes the data stage. The device receives data. (c) The host issues an IN transaction and executes the status stage. The device returns a packet having the zero data length.

Control transfer without data stage is executed without using the data stage as shown in this example.

Control transfers to the status stage when the host issues a transaction having the direction reverse to the data stage. The firmware must monitor and detect a change of IN_TranNAK status (EP0IntStat.IN_TranNAK bit) to transfer control from the data stage to the status stage.

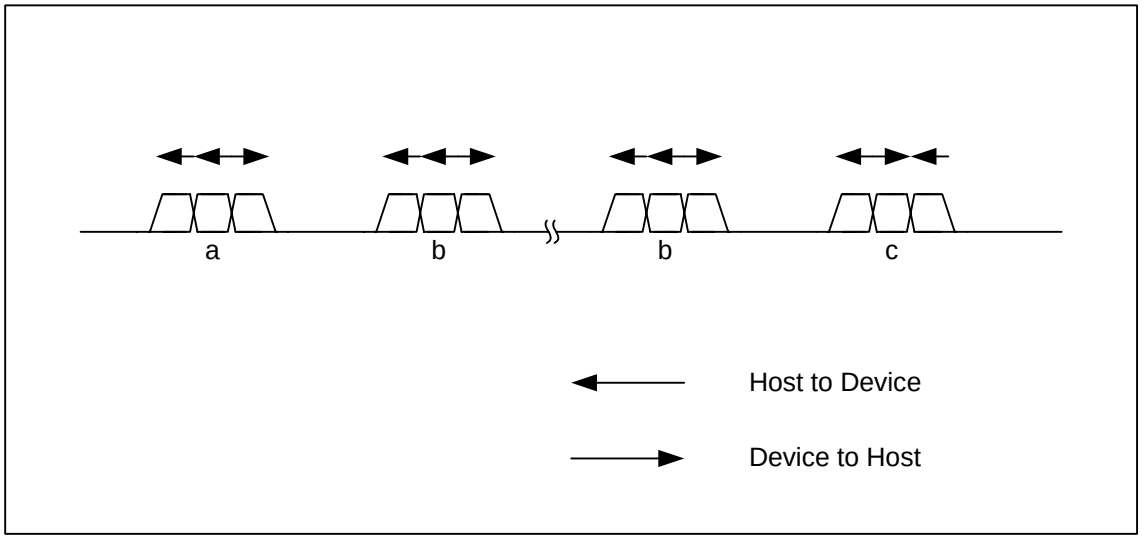


Fig.A2.6.5 Control transfer if the data stage is in the OUT direction

Fig.A2.6.6 shows control transfer when the data stage is in the IN direction. (a) The host starts control transfer using the SETUP transaction. The device firmware analyzes the contents of request and prepares to respond to the data stage. (b) The host issues an IN transaction and executes the data stage. The device sends data. (c) The host issues an OUT transaction and executes the status stage. The device responds with an ACK.

Control transfers to the status stage when the host issues a transaction having the direction reverse to the data stage. The firmware must monitor and detect a change of OUT_TranNAK status (EP0IntStat.OUT_TranNAK bit) to transfer control from the data stage to the status stage.

As the normal OUT and IN transactions are executed in the data stage and status stage during control transfer, the NAK-based flow control is effective. The device is allowed to prepare its response within the specified time period.

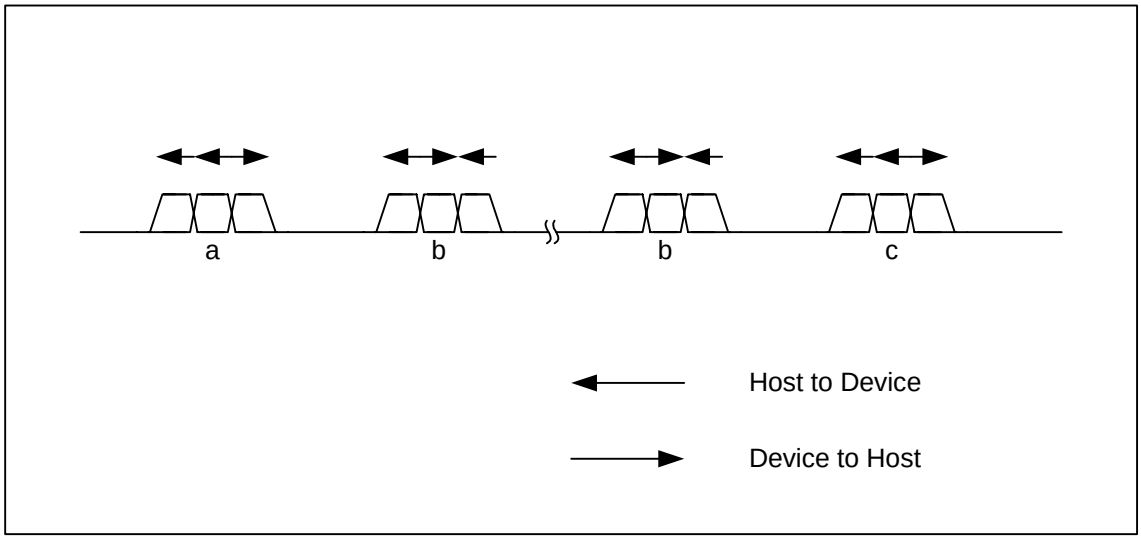


Fig.A2.6.6 Control transfer if the data stage is in IN direction

A2.6.2.3.1 Setup Stage

When receiving a SETUP token directed to the local node, the device automatically executes the setup transaction.

The firmware must monitor the RcvEP0SETUP status, analyze the request using EP0SETUP_0 to EP0SETUP_7 registers, and execute control transfer.

If the received request has the data stage in the OUT direction, clear the INxOUT bit of EP0Control register to transfer to the data stage, and set endpoint EP0 in the OUT direction.

If the received request has the data stage in the IN direction, set the INxOUT bit of EP0Control register to transfer to the data stage, and set endpoint EP0 in the IN direction.

If the received request does not have the data stage, set the INxOUT bit of EP0Control register to transfer to the status stage, and set endpoint EP0 in the IN direction.

A2.6.2.3.2 Data Stage and Status Stage

Read the EP0SETUP_0 to EP0SETUP_7 register data, and transfer to the next stage by following the analyzed request contents.

If this stage is in the OUT direction, clear the INxOUT bit of EP0Control register to set the OUT direction, and control the stage by setting the EP0ControlOUT register appropriately. When the process of Setup stage ends, the ForceNAK bit is set. Also, the SETUP_Control.ProtectEP0 bit is set.

If this stage is in the IN direction, set the INxOUT bit of EP0Control register to set the IN direction, and control the stage by setting the EP0ControlIN register appropriately. When the process of Setup stage ends, the ForceNAK bit is set. Also, the SETUP_Control.ProtectEP0 bit is set.

A2.6.2.3.3 Automatic Address Setting Function

This LSI has the function that can automatically process the SetAddress() request with control transfer at endpoint EP0.

The LSI hardware (H/W) checks the request contents by reading EP0SETUP_0 to EP0SETUP_7 registers. If it is a valid SetAddress() request, control is transferred to the status stage of this request without notification to the firmware. When process in Status stage has completed, the address is set in the USB_Address register and the SetAddressCmp status (SIE_IntStat.SetAddressCmp bit) is issued to the firmware.

The firmware monitors the SetAddressCmp status, and when it is issued, the firmware can check the address by reading the USB_Address register.

A2.6.2.3.4 Descriptor Reply Function

This LSI has the Descriptor Reply function that is effective to the GetDescriptor() and other data requests to be issued multiple times.

The firmware can use this function for a request issued in the Data stage IN direction.

Clear the EP0ControlIN.ForceNAK bit, and before starting a response to the Data stage, set the first address of reply data of the FIFO descriptor area in DescAdrs_H, L registers. Also, set the total byte count of the reply data in DescSize_H, L registers and set the EP0Control.ReplyDescriptor bit.

The Descriptor Reply function replies data packets in response to the Data stage IN transaction and executes IN transactions until the specified number of data sets are sent. If an IN transaction is issued after the specified number of data sets have been sent, this function responds with a NAK. If a fragment data exists for the maximum packet size, the Descriptor Reply function sets the EP0ControlIN.EnShortPkt bit so that it can respond to an IN transaction until all data is returned.

When this function receives an OUT token and when detects a transition to the Status Stage, it clears the EP0Control.ReplyDescriptor bit and issues a DescriptorCmp status (FIFO_IntStat.DescriptorCmp bit) to the firmware. If the DescriptorCmp status is detected, the firmware must execute the Status Stage.

For the descriptor areas, see the FIFO Memory section in the Functional Description.

A2.6.2.4 Bulk Transfer and Interrupt Transfer

Both Bulk Transfer and Interrupt Transfer at EPa-h generic endpoints can be controlled as a data flow (see Paragraph A2.6.2.5) or continuous individual transactions (see Paragraph A2.6.2.2).

A2.6.2.5 Data Flow

The following explains the general data flow control for OUT Transfer and IN Transfer.

A2.6.2.5.1 OUT Transfer

Data received by OUT transfer is written in FIFO memory at each endpoint. The data can be read from FIFO memory by the register read through CPU interface or through DMA transfer.

If you use the register read through CPU interface for data reading, select only one endpoint by setting the EPx{x=0,a-h}Join.JoinCPU_Rd bit. At the selected endpoint, data can be read from its FIFO memory in the data receive sequence, using the FIFO_Rd registers or FIFO_ByteRd registers. Also, you can check the data count that can be read form FIFO memory by referring to the FIFO_RdRemain_H,L registers. Because you cannot read data from an empty FIFO memory, be sure to check the readable data count by referring to the FIFO_RdRemain_H,L registers and read the data that do not exceed the limit.

If you use the DMA read for data reading from FIFO memory, select only one endpoint for each of DMA channel by setting the EPx{x=0,a-h}Join.JoinDMAx{x=0,1} bits. Also, set the DMAx{x=0,1}_Control.Dir bit to 1. Data can be read from FIFO memory of the selected endpoint in the data receiving sequence when the DMA transfer is executed with an external DMA controller or others. Also, you can check the data count remaining in the FIFO memory by referring to the DMAx{x=0,1}_Remain_H,L registers. When the FIFO memory is emptied, the DMA transfer is stopped and the flow is controlled automatically.

When the FIFO memory has a space to receive data packets, an OUT transaction is responded automatically and the data is received. Therefore, the OUT transfer can be made without individual transaction control by the firmware. However, if the EPx{x=a-h}Control.DisAF_NAK_Short bit has been cleared (default value) and if a short packet is received (including a packet having zero data length), the EPx{x=a-h}Control.ForceNAK bit of this endpoint is set.. Therefore, the EPx{x=a-h}Control.ForceNAK bit must be cleared when the next data transfer is ready to start.

A2.6.2.5.2 IN Transfer

Data sent by IN transfer must be written in FIFO memory at each endpoint. The data can be written in FIFO memory by the register write through CPU interface or through DMA transfer.

If you use the register write through CPU interface for data writing, select only one endpoint by setting the EPx{x=0,a-h}Join.JoinCPU_Wr bit. At the selected endpoint, data can be written in its FIFO memory using the FIFO_Wr register. The data packets are sent in the same data writing sequence. You can check the space of FIFO memory by referring to FIFO_WrRemain_H,L registers. You cannot write data in FIFO memory if it is filled with data. Be sure to check the memory space by referring to the FIFO_WrRemain_H,L registers, and write data that does not exceed the limit.

If you use the DMA write for data writing in FIFO memory, select only one endpoint for each of DMA channel by setting the EPx{x=0,a-h}Join.JoinDMAx{x=0,1} bits. Also, set the DMAx{x=0,1}_Control.Dir bit to 0. At the selected endpoint, data is written in its FIFO memory through DMA transfer by an external DMA controller or others. The data packets are sent in the same data writing sequence. When the FIFO memory is filled with data, the DMA transfer is stopped and the flow is controlled automatically.

If the FIFO memory contains data that exceeds the maximum packet size, the IN transaction is responded automatically and the data can be sent. Therefore, the IN transfer can be made without individual transaction control by the firmware. However, if you need to send a short packet at the end of data transfer, be sure to set the EnShortPkt bit. This bit is cleared when the IN transaction that sends the short packet is completed. It can be set at the time when data writing in FIFO memory has completed. Also, if the DMAx{x=0,1}_FIFO_Control.AutoEnShort bit is set and if a fragment data that is smaller than the maximum packet size exists in FIFO memory at the end of DMA writing, the EnShortPkt bit of this endpoint is set automatically.

A2.6.2.6 Bulk Only Support

This LSI provides the Bulk Only Support function to support the Command Block Wrapper (CBW) reception and the Command Status Wrapper (CSW) sending, which are unique to the USB Mass Storage Class (Bulk Only Transport Protocol) during Bulk Transfer at generic endpoints EPa to EPh.

If the BulkOnlyConfig.EPx{x=a-h}BulkOnly bit is set, the Bulk Only Support function is enabled at the selected endpoint.

When the CBW Support or CSW Support of the Bulk Only Support function is being executed, the area assigned as CBW area or CSW area is used for packet reception (for CBW) or transmission (for CSW), instead of the FIFO memory area that is usually assigned at the endpoint.

If the EPx{x=a-h}Control.ForceSTALL bit is set at the target endpoint and if the OUT transaction is responded with STALL, the CBW Error Status (BulkIntStat.CBW_Err bit) is issued to the firmware. The BulkOnlyControl.GoCBW_Mode bit is cleared, and the CBW Support is terminated. If the BulkOnlyControl.GoCSW_Mode bit is set at this time, it is also cleared simultaneously.

If a CRC error or another transaction error has occurred in the OUT transaction, the data is not received and the CBW Transaction Error Status (BulkIntStat.CBW_TranErr bit) is issued to the firmware. In this case, the BulkOnlyControl.GoCBW_Mode bit is not cleared and the CBW Support is continued to execute. If the BulkOnlyControl.GoCSW_Mode has been set at this time, it is not cleared.

Data received in the CBW area can be read by the RAM_Rd function.

A2.6.2.6.1 CBW Support

The firmware can use the CBW Support during command transport of the Bulk Only Transport protocol. If the BunlOnlyConfig.EPx{x=a-h}BulkOnly bit is set, the CBW Support is enabled at the corresponding OUT-direction endpoint. The CBW Support must be enabled only at one endpoint. If the BulkOnlyContol.GoCBW_Mode bit is set when the CBW Support is enabled, the CBW Support is executed. The data received by the OUT transaction at the corresponding endpoint is handled as the CBW.

If the data length of data packet is 31 bytes that is equal to the expected CBW length, the data is stored in the CBW area and the CBW Completion Status (BulkIntStat.CBW_Cmp bit) is issued to the firmware. Also, the BulkOnlyControl.GoCBW_Mode bit is cleared automatically, and the CBW Support is terminated. If the BulkOnlyControl.GoCSW_Mode bit is set at this time, it is also cleared simultaneously.

If the data length of data packet is less than or greater than 31 bytes, the data is not received. The CBW Data Length Error Status (BulkIntStat.CBW_LengthErr bit) is issued to the firmware. Also, the BulkOnlyControl.GoCBW_Mode bit is cleared automatically, and the CBW Support is terminated. If the BulkOnlyControl.GoCSW_Mode bit is set at this time, it is also cleared simultaneously. If the CBW_Err status is issued, a phase mismatching has occurred in the Bulk Only Transport protocol. The firmware must “stall” the

endpoint or others to recover the communication.

If the EPx{x=a-h}Control.ForceSTALL bit is set at the target endpoint and if the OUT transaction is responded with STALL, the CBW Error Status (BulkIntStat.CBW_Err bit) is issued to the firmware. The BulkOnlyControl.GoCBW_Mode bit is cleared, and the CBW Support is terminated. If the BulkOnlyControl.GoCSW_Mode bit is set at this time, it is also cleared simultaneously.

If a CRC error or another transaction error has occurred in the OUT transaction, the data is not received and the CBW Transaction Error Status (BulkIntStat.CBW_TranErr bit) is issued to the firmware. In this case, the BulkOnlyControl.GoCBW_Mode bit is not cleared and the CBW Support is continued to execute. If the BulkOnlyControl.GoCSW_Mode has been set at this time, it is not cleared.

Data received in the CBW area can be read by the RAM_Rd function.

A2.6.2.6.2 CSW Support

The firmware can use the CSW Support during status transport of the Bulk Only Transport protocol. If the BulkOnlyConfig.EPx{x=a-h}BulkOnly bit is set, the CBW Support is enabled at the corresponding IN-direction endpoint. The CSW Support must be enabled only at one endpoint. If the D_BulkOnlyControl.GoCSW_Mod bit is set when the CSW Support is enabled, the CSW Support is executed. The data that is sent by IN transaction at the corresponding endpoint is handled as the CSW.

If the 13-byte CSW data is returned to the host by IN transaction and if the ACK signal is received from the host and this transaction is completed, the CSW Completion status (BulkIntStat.CSW_Cmp bit) is issued to the firmware. Also, the BulkOnlyControl.GoCSW_Mode bit is cleared automatically, and the CSW Support is terminated. Also, the BulkOnlyControl.GoCBW_Mode bit is set simultaneously, and the CBW Support is started.

If the 13-byte data is returned to the host by IN transaction and if the ACK signal is NOT received from the host, the CSW Error Status (BulkIntStat.CSW_Err bit) is issued to the firmware. In this case, the BulkOnlyControl.GoCSW_Mode bit is not cleared and the CSW Support is continued to execute. Also, the hardware sets the BulkOnlyControl.GoCBW_Mode bit and starts the CBW Support. In such case, both CSW Support and CBW Support are executed simultaneously. If the host cannot receive the CSW and an error occurs on it, the CSW is retried. As the CSW Support is being executed, a response can be made. If the device cannot receive the ACK and an error occurs on it, the next CBW is executed. Because the CBW Support is being executed, a response can be made. The CBW Support is executed, and the CSW Support is terminated.

Data can be written in the CSW area by the RAM_WrDoor function.

A2.6.2.7 Auto Negotiation Function

The Auto-Negotiation function sequentially checks the USB bus status and executes suspend detection, reset detection, HS Detection handshaking, resume detection, and restore execution automatically. The actual operations can be checked by referring to each interrupt (DetectRESET, DetectSUSPEND, ChirpCmp, and RestoreCmp).

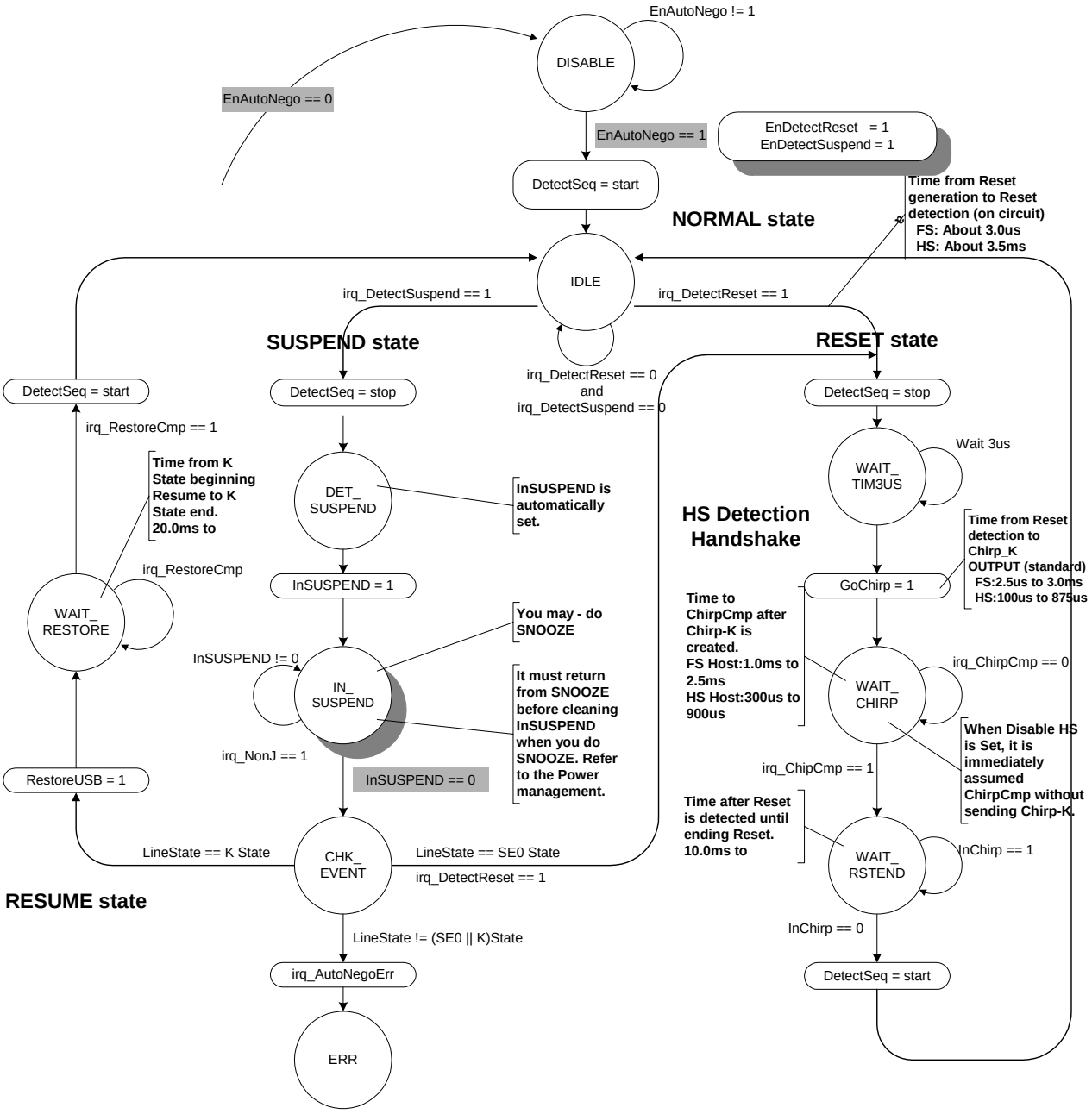


Fig.A2.6.7 Auto-Negotiator

A2.6.2.7.1 DISABLE State

The device enters the DISABLE state when the NegoControl.EnAutoNego bit is cleared.

When you enable the Auto-Negotiation function, first set the Reset Detection Interrupt Enable bit (SIE_IntEnb.EnDetectRESET) and the Suspend Detection Interrupt Enable bit (SIE_IntEnb.EnDetectSUSPEND) to enable those event detection interrupts before you set the NegoControl.EnAutoNego bit.

When the Auto-Negotiation function is enabled, the internal event detection functions are activated. While the Auto-Negotiation function is enable, never set the NegoControl.DisBusDetect bit.

A2.6.2.7.2 IDLE

A reset detection or a suspend detection is waited in the IDLE state.

If the current USB speed is HS and if no bus activity is detected on USB bus for 3 msec or longer, the FS termination is made valid. If an FS-J is detected, the suspend is determined. If the SE0 state is detected, the reset is determined. If the current USB speed is FS and if the 2.5-μsec or longer SE0 state is detected, the reset is determined. If no bus activity is detected for 3 msec or longer, the suspend is determined. At the same time, a reset detection interrupt or a suspend detection interrupt occurs and the SIE_IntStat.DetectRESET bit or SIE_IntStat.DetectSUSPEND bit is set.

If the suspend is determined, the event detection function is once stopped and the device enters the DET_SUSPEND state.

If the reset is determined, the event detection function is once stopped and the device enters the WAIT_TIM3US state.

A2.6.2.7.3 WAIT_TIM3US

After the reset detection, the time period until HS Detection Handshaking is adjusted in the WAIT_TIM3US state. After a certain time (approx. 3 μsec), the device enters the WAIT_CHIRP state.

A2.6.2.7.4 WAIT_CHIRP

The NegoControl.GoChirp bit is set automatically, and HS Detection Handshaking takes place in the WAIT_CHIRP state. After the HS Detection Handshaking, the “Chirp” end interrupt status (SIE_IntStat.ChirpCmp) bit is set and the device enters the WAIT_RSTEND state. For the detailed HS Detection Handshaking, see Paragraph A2.6.2.7.11.4.

Also, if the NegoControl.DisableHS bit has been set, the HS Detection Handshaking is not executed. The “Chirp” end interrupt status (SIE_IntStat.ChirpCmp) bit is set and the device enters the WAIT_RSTEND state.

After this state has ended, the device operates at the transmission speed being set by the USB_Status.FSxHS bit. If you need to detect this transmission speed change, set the SIE_IntEnb.EnChirpCmp bit to enable the “Chirp” end interrupt described above.

A2.6.2.7.5 WAIT_RSTEND

The device waits in this state until the reset period ends. If the USB speed is HS, the end of reset period is determined when the “Chirp” transmission (or its reception by this IC) ends. If FS, the end of reset period is determined when the SH0 signal transitions to J signal.

When the end of reset period is determined, the event detection function is enabled and the device enters the IDLE state again.

A2.6.2.7.6 DET_SUSPEND

If the suspend is determined, the NegoControl.InSUSPEND bit is set automatically and the device enters in the IN_SUSPEND state. This NegoControl.InSUSPEND bit enables the FS-J Bus Transition Detect Function so that a resume request and a reset request from the host is detected.

The reduction of actual current consumption in Suspend state depends on the application. This LSI supports the two-stage (Snooze and Sleep) current consumption reduction measures. For the detailed information and control, see the Power Management function section (A2.6.3).

To allow the resume detection (FS-K) that indicates the end of suspend state, set the SIE_IntEnb.EnNonJ bit and enable the NonJ interrupt using the firmware.

A2.6.2.7.7 IN_SUSPEND

If the NonJ interrupt status (SIE_IntStat.NonJ) is set, it is determined to be a recovery request from the suspend state. When the NegoControl.InSUSPEND bit is cleared by the firmware, the device enters the CHK_EVENT state.

If the Remote Wakeup function is enabled by the application and if the device is recovered autonomously from the suspend state, set the NegoControl.SendWakeup bit in this state and output the FS-K signal more than 1 msec but less than 15 msec.

A2.6.2.7.8 CHK_EVENT

The USB cable is checked and if an FS-K signal is detected, the resume is determined. If the SE0 state is detected, the reset is determined. If the resume is determined, the NegoControl.RestoreUSB bit is set and the device returns to the transmission speed (that follows the USB_Status.FSxHS value) before being suspended. If the reset is determined, the event detection function is once stopped and the device enters the WAIT_TIM3US state in the similar way as for the transition from IDLE state.

If a state other than FS-K or SE0 state is detected, the Auto-Negotiation Error Interrupt Status (SIE_IntStat.AutoNegoErr) bit is set and the device enters the ERR state.

A2.6.2.7.9 WAIT_RESTORE

If the SIE_IntStat.RestoreCmp bit is set, the Event Detection function is enabled and the device enters the IDLE state.

A2.6.2.7.10 ERR

Once the device has entered in the ERR state, it cannot exit this state except when the Auto-Negotiation function is stopped. This state is not defined by USB standards.

The USB cable disconnection is not determined in all states. If the USB cable is unplugged from its port, you must stop the Auto-Negotiation function immediately.

A2.6.2.7.11 Individual Negotiation Functions

A2.6.2.7.11.1 Suspend Detection (in HS Mode)

If the LSI is operating in the HS mode and if no signal transmission or reception is detected more than 3 msec (T1), the device automatically transitions to the FS mode (the HS termination is disabled and the FS termination (Rpu) is enabled). At this time, signal DP goes High, and the “J” state can be check by referring to USB_Status.LineState[1:0] bits. (Note that the Reset (described later) is determined if the SE0 state is detected.) If the “J” state is detected again at T2, the SIE_IntStat.DetectSUSPEND bit is set.

At this time, if both SIE_IntEnb.EnDetectSUSPEND bit and DeviceIntEnb.EnSIE_IntStat bit are set and if MainIntEnb.EnDeviceIntStat bit is set, an interrupt signal is asserted at the same time and the USB device is determined in the Suspend state. The following shows the signal timing during snoozing.

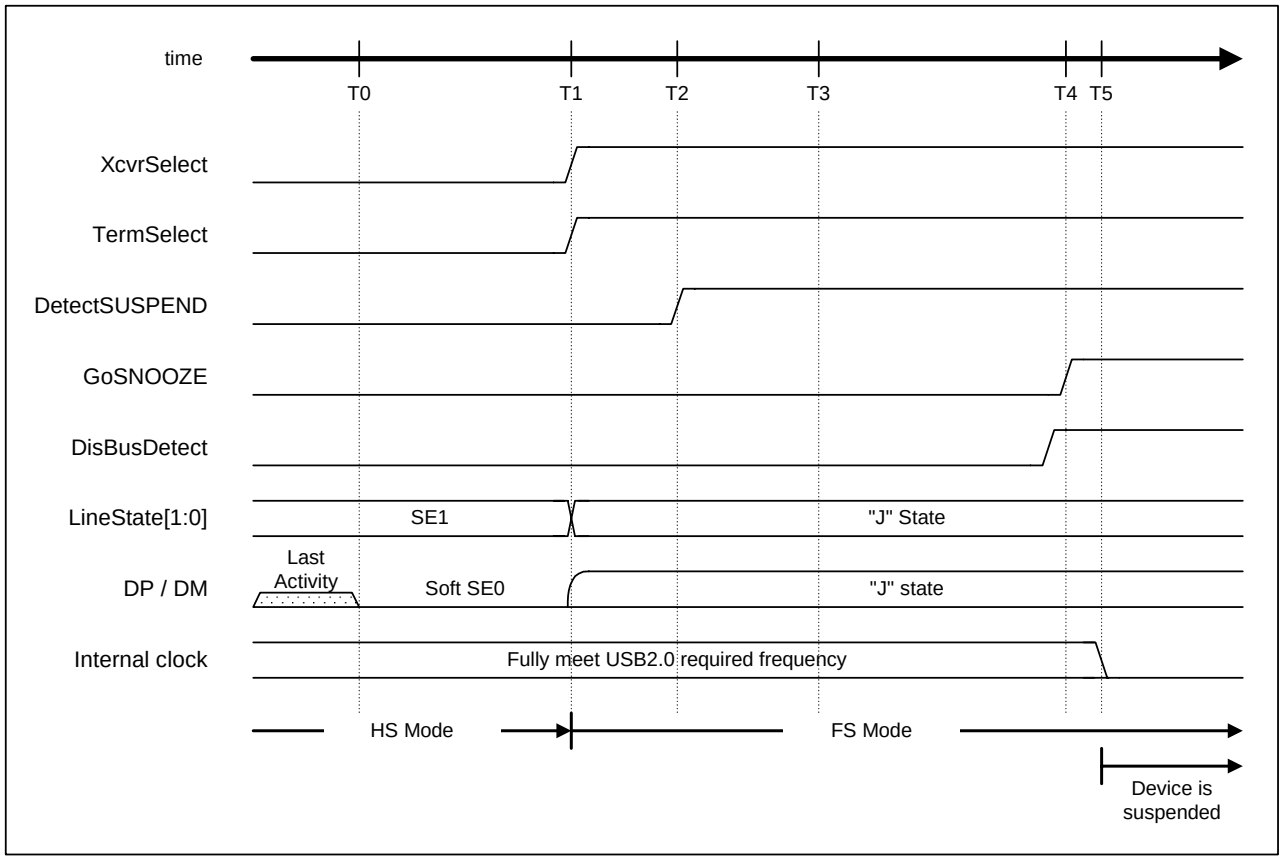


Fig.A2.6.8 Suspend Timing (HS mode)

Table A2.6.5 Suspend Timing Values (HS mode)

Timing Parameter	Description	Values
T0	Last bus activity	0(Reference)
T1	If a bus activity is not detected yet at T1, the XcvrSelect and TermSelect signals are set to 1 and the HS mode is switched to the FS mode.	HS Reset T0 + 3.0ms < T1 {T_{WTREV}} < HS Reset T0 + 3.125ms
T2	The LineState[1:0] signal is sampled. If it is in the “J” state, the DetectSUSPEND signal is set to 1 and the USB is determined to be in the Suspend state.	T1 + 100us < T2 {$T_{WTWRSTHS}$} < T1 + 875us
T3	No RESUME request must be issued before T3.	HS Reset T0 + 5ms {T_{WTRSM}}
T4	The device transitions to the SNOOZE state completely. After T4, the current that is greater than the specified suspend current of USB cannot be sent from the VBUS. (The DisBusDetect signal must be set to 1 before transition to the SNOOZE state.)	HS Reset T0 + 10ms {T_{2SUSP}}
T5	The internal clock is stopped completely.	T5 < T4 + 10us

Note: The signal name in braces { } has been defined by the USB2.0 Standard manual.

A2.6.2.7.11.2 Suspend Detection (in FS Mode)

If this LSI is operating in the FS mode, and if no signal transmission or reception is detected for 3 msec or longer or if the "J" state is detected by USB_Status.LineState[1:0] bits at T1 and this "J" state is still detected at T2, the USB is determined in the Suspend state and the SIE_IntStat.DetectSUSPEND bit is set.

At this time, if both SIE_IntEnb.EnDetectSUSPEND bit and DeviceIntEnb.EnSIE_IntStat bit are set and if MainIntEnb.EnDeviceIntStat bit is set, an interrupt signal is asserted at the same time. The following shows the signal timing during snoozing.

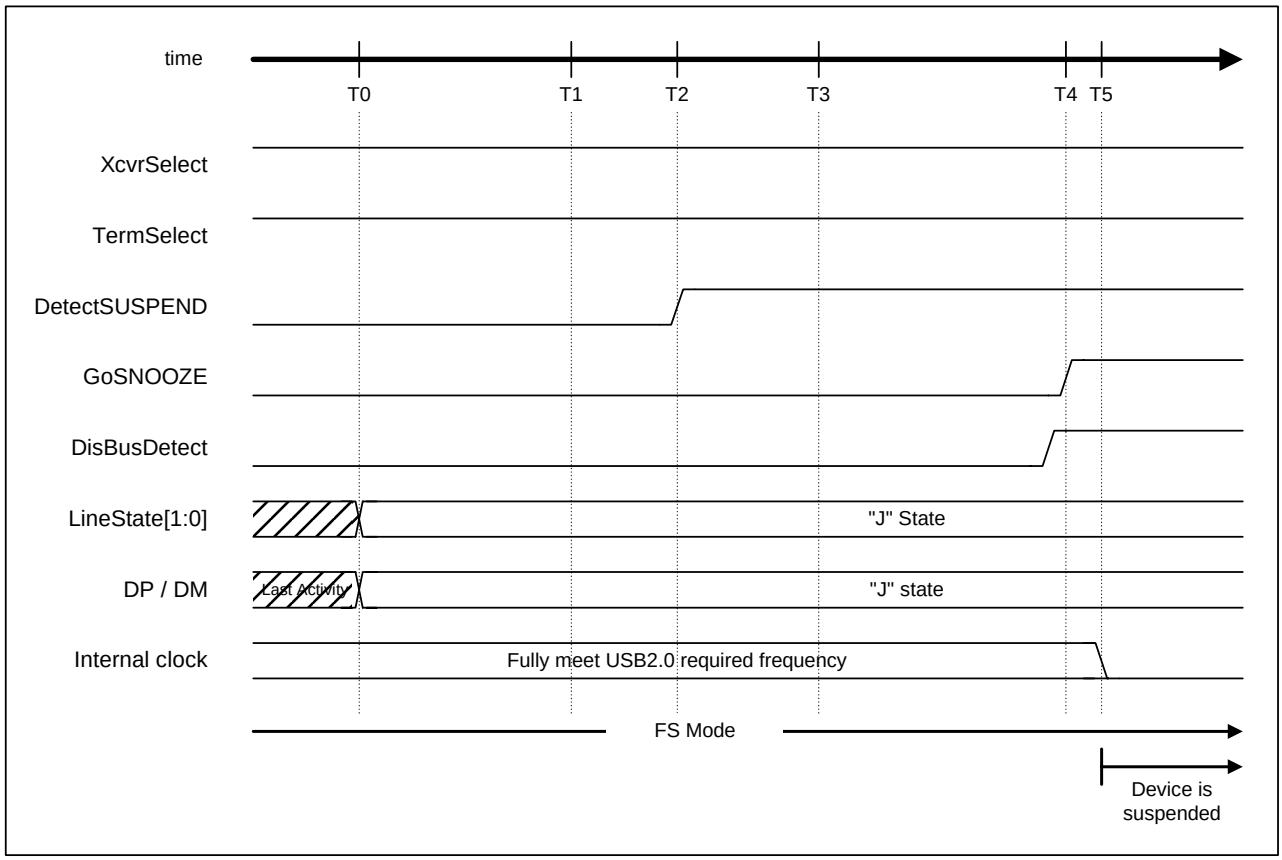


Fig.A2.6.9 Suspend Timing (FS mode)

Table A2.6.6 Suspend Timing Values (FS mode)

Timing Parameter	Description	Values
T0	Last bus activity	0(Reference)
T1	No bus activity is detected yet at T1.	T0 + 3.0ms < T1 {T_{WTREV}} < T0 + 3.125ms
T2	The LineState[1:0] signal is sampled. If it is in the "J" state, the DetectSUSPEND signal is set to 1 and the USB is determined to be in the Suspend state.	T1 + 100us < T2 {$T_{WTWRSTHS}$} < T1 + 875us
T3	No RESUME request must be issued before T3.	T0 + 5ms {T_{WTRSM}}
T4	The device transitions to the SNOOZE state completely. After T4, the current that is greater than the specified suspend current of USB cannot be sent from the VBUS. (The DisBusDetect signal must be set to 1 before transition to the SNOOZE state.)	T0 + 10ms {T_{2SUSP}}
T5	The internal clock is stopped completely.	T5 < T4 + 10us

Note: The signal name in braces { } has been defined by the USB2.0 Standard manual.

A2.6.2.7.11.3 Reset Detection (in HS Mode)

If the LSI is operating in the HS mode and if no signal transmission or reception is detected more than 3 msec, the device automatically transitions to the FS mode (the HS termination is disabled and the FS termination (Rpu) is enabled). After this operation, the DP line continues in Low state and, therefore, the SE0 state can also be checked by USB_Status.LineState[1:0] bits. If the SE0 state is still detected at T2, the SIE_IntStat.DetectRESET bit is set.

At this time, if both SIE_IntEnb.EnDetectRESET bit and DeviceIntEnb.EnSIE_IntStat bit are set and if MainIntEnb.EnDeviceIntStat bit is set, an interrupt signal is asserted at the same time and the signal is determined to be the Reset command. After this time, set the NegoControl.DisBusDetect bit first, then start HS Detection Handshaking (described later).

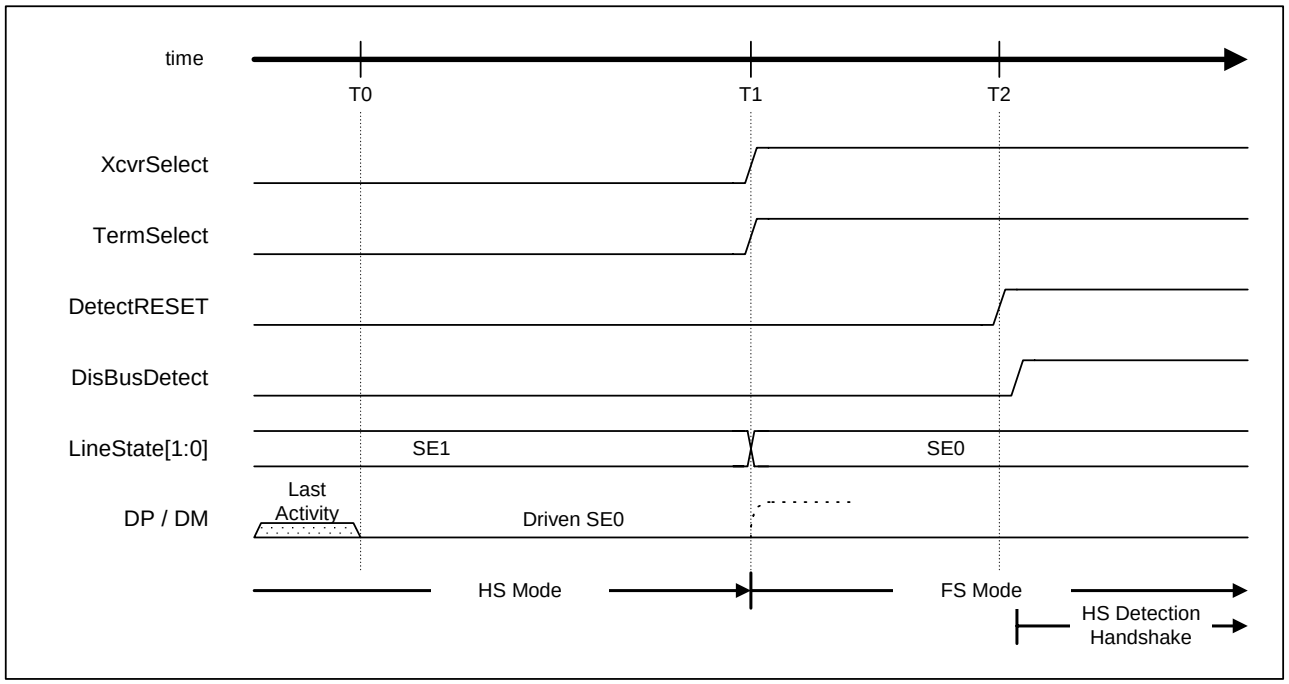


Fig.A2.6.10 Reset Timing (HS mode)

Table A2.6.7 Reset Timing Values (HS mode)

Timing Parameter	Description	Values
T0	Last bus activity	0(Reference)
T1	If a bus activity is not detected yet at T1, the XcvrSelect and TermSelect signals are set to 1 and the HS mode is switched to the FS mode.	HS Reset T0 + 3.0ms < T1 {T_{WTREV}} < HS Reset T0 + 3.125ms
T2	The LineState[1:0] signal is sampled. If it is in the "SE0" state, the DetectRESET signal is set to 1 and the transition to the Reset state is determined. After the reset detection, the DisBusDetect signal is set to 1 and HS Detection Handshaking takes place.	T1 + 100us < T2 {$T_{WTWRSTHS}$} < T1 + 875us

Note: The signal name in braces { } has been defined by the USB2.0 Standard manual.

Reset Detection (in FS mode)

If this LSI is operating in the FS mode, and if the SE0 detection by USB_Status.LineState[1:0] bits is continued for 2.5 us or longer (at T1), the SIE_IntStat.DetectRESET bit is set.

At this time, if both SIE_IntEnb.EnDetectRESET bit and DeviceIntEnb.EnSIE_IntStat bit are set and if MainIntEnb.EnDeviceIntStat bit is set, an interrupt signal is asserted at the same time and the signal is determined to be the Reset command. After this time, set the NegoControl.DisBusDetect bit first, then start HS Detection Handshaking (described later).

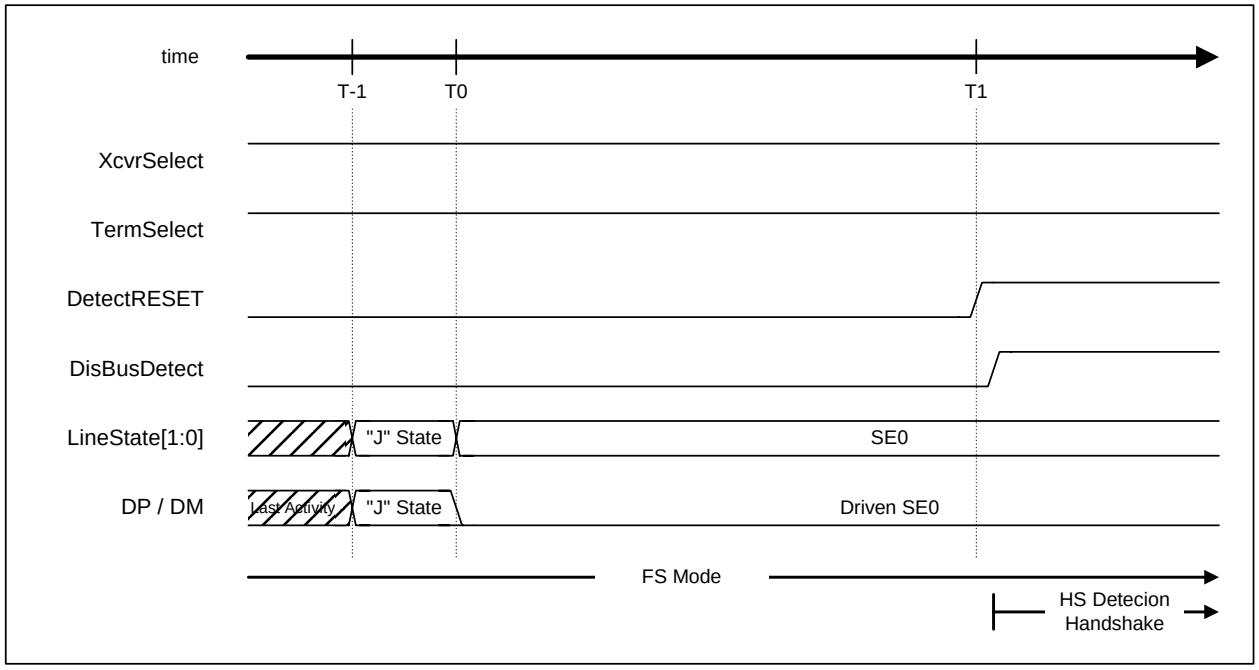


Fig.A2.6.11 Reset Timing (FS mode)

Table A2.6.8 Reset Timing Values (FS mode)

Timing Parameter	Description	Values
T-1	Last bus activity	
T0	Start of Reset command from a downstream port	0(Reference)
T1	If the “SE0” state continues, the DetectRESET signal is set to 1 and the transition to the Reset state is determined. After the reset detection, the DisBusDetect signal is set to 1 and HS Detection Handshaking takes place.	HS Reset T0 + 2.5us < T1 {T_{WTREV}}

Note: The signal name in braces { } has been defined by the USB2.0 Standard manual.

A2.6.2.7.11.4 HS Detection Handshake

HS Detection Handshaking starts when the SE0 signal is asserted from a downstream port during Suspend state, during FS operation, or during HS operation (that is, when the Reset operation is started in any of the above states). For details, see the related USB2.0 Standard manual.

The following explains how to transition to the HS Detection Handshaking state from any of the above three states.

If this LSI is in the Suspend state, it must transition to the HS Detection Handshaking state just when SE0 state is detected on the bus.

If this LSI is operating in the FS mode, it must transition to the HS Detection Handshaking state when 2.5 us or more time has passed after SE0 detection.

If this LSI is operating in the HS mode and if the SE0 state continues 3.0 msec or longer, the LSI is temporarily switched to the FS mode so that this signal is determined to be the Suspend state or Reset of the USB. During this time, both XcvrControl.XcvrSelect and XcvrControl.TermSelect bits are switched to the FS mode, the HS termination is disabled, and the FS termination is enabled. These mode switching must complete within 3.125 msec. The USB_Status.LineState[1:0] bits are checked when 100 us or more have passed (but within 875 us) after the above mode switching has completed. If the SE0 state is detected, the Reset is determined. If the Reset is determined, the device must transition to the HS Detection Handshaking state.

In any case, the Reset continues 10 msec minimum, but its timing slightly varies depending on the operation state (HS or FS mode) before the state transition. In this section, the reset state start time is defined as “HS Reset T0”. The following explains the device operations after this “HS Reset T0” time.

When the device is operating, its internal clock is stable and there is no problem. However, if the device is set to the SLEEP or SNOOZE state when suspended, the internal clock is not output during Reset detection. Therefore, the PM_Control_0.GoActDevice bit must be set to 1 and the internal clock must be operated to allow HS Detection Handshaking. For this operation details, see the Power Management function section (A2.6.3).

A2.6.2.7.11.5 If connected to a Downstream Port supporting FS Mode

The following explains how this LSI operates if it is connected to a downstream port that does not support the HS mode operation. When HS Detection Handshaking starts (at T0), both XcvrControl.XcvrSelect and XcvrControl.TermSelect bits must be the FS mode (the FS termination, that is, the DP’s pull-up resistance (Rpu) is enabled, but the HS termination is disabled).

First, set the NegoControl.GoChirp bit. Then, XcvrControl.OpMode[1:0] bits become the "Disable Bit Stuffing and NRZI encoding" state, and the data padded with zeros is prepared (at T1). This is used to sent the “HS K” (chirp) signal onto the bus. At the same time, the XcvrControl.XcvrSelect bit is set to the HS mode and the transmission is enabled. the “HS K” (chirp) signal is sent to the downstream port. After the signal transmission, the device waits for a 2chirp” from the downstream port (at T2). The downstream port that supports the HS mode operation usually sends “HS K” and “HS J” signals sequentially after T3 (explained later). However, if the downstream port does not support the HS mode operation (in this case), it does not sent the “chirp” even at T4. Therefore, the XcvrControl.XcvrSelect bit is automatically switched to the FS mode, and the NegoControl.GoChirp bit is cleared. Also, the USB_Status.FSxHS bit is set, and the SIE_IntStat.ChirpCmp bit is set.

At this time, if both SIE_IntEnb.EnChirpCmp bit and DeviceIntEnb.EnSIE_IntStat bit are set and if MainIntEnb.EnDeviceIntStat bit is set, an interrupt signal is asserted at the same time and the end of HS Detection Handshaking must be determined.

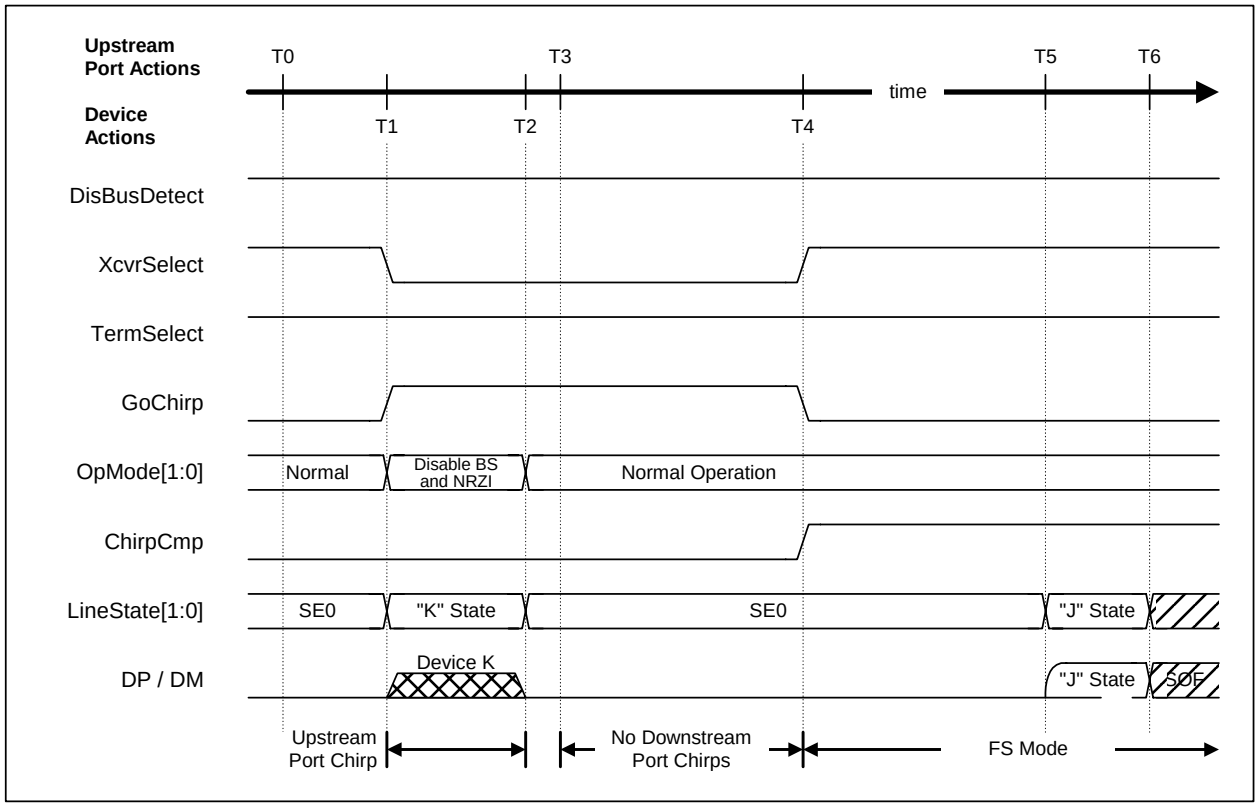


Fig.A2.6.12 HS Detection Handshake (FS mode)

Table A2.6.9 HS Detection Handshake Timing Values (FS mode)

Timing Parameter	Description	Values
T0	Start of HS Detection Handshaking	0(Reference)
T1	The HS transceiver is enabled, and GoChirp signal is set to 1 and “Chirp K” transmission starts.	T0 < T1 < HS Reset T0 + 6.0ms
T2	“Chirp K” transmission has completed. It must be sent at least 1 msec.	T1 + 1.0ms {T_{UCH}} < T2 HS Reset T0+7.0ms {T_{UCHEND}}
T3	If the downstream port supports HS mode operation, the “Chirp K” transmission starts at this time.	T2 < T3 < T2 + 100us {T_{WTDCH}}
T4	If no Chirp is detected, the operation is returned to the FS mode. The ChirpCmp signal is set to 1, and the end of Reset sequence is waited.	T2 + 1.0ms < T4 {T_{WTFS}} < T2 + 2.5ms
T5	End of reset sequence	HS Rest T0 + 10ms {T_{DRST}(MIN)}
T6	The signal name in braces { } has been defined by the USB2.0 Standard manual.	T6

Note: The signal name in braces { } has been defined by the USB2.0 Standard manual.

Note: 66000 cycles (for 60MHz internal clock) are used for determination to generate 1-msec minimum “Chirp K” signal.

A2.6.2.7.11.6 If connected to a Downstream Port supporting HS Mode

The following explains how this LSI operates if it is connected to a downstream port that supports the HS mode operation. When HS Detection Handshaking starts (at T0), both XcvrControl.XcvrSelect and XcvrControl.TermSelect bits must be the FS mode (the FS termination, that is, the DP's pull-up resistance (Rpu) is enabled, but the HS termination is disabled).

First, set the NegoControl.GoChirp bit. Then, XcvrControl.OpMode[1:0] bits become the "Disable Bit Stuffing and NRZI encoding" state, and the data padded with zeros is prepared (at T1). This is used to sent the "HS K" (chirp) signal onto the bus. At the same time, the XcvrControl.XcvrSelect bit is set to the HS mode and the transmission is enabled. the "HS K" (chirp) signal is sent to the downstream port. After the signal transmission, the device waits for a 2chirp" from the downstream port (at T2). As the downstream port supports HS mode operation, the "HS K" (Chirp K) and "HS J" (Chirp J) signals are alternatively output (at T3). When this state is detected at least 6 times (as "Chirp K-J-K-J-K-J") by USB_Status.LineState[1:0] bits (at T6), the XcvrControl.TermSelect bit is automatically switched to the HS mode (at T7) and the device transitions to the HS mode completely. During this time, the NegoControl.GoChirp and NegoStatus.FSxHS bits are cleared, but the SIE_IntStat.ChirpCmp bit is set.

At this time, if both SIE_IntEnb.EnChirpCmp bit and DeviceIntEnb.EnSIE_IntStat bit are set and if MainIntEnb.EnDeviceIntStat bit is set, an interrupt signal is asserted at the same time and the end of HS Detection Handshaking must be determined.

The Chirp K and Chirp J signals sent form the downstream port must be considered as the bus activities, but they must not be determined as the Suspend state of USB. In the HS mode, these Chirp K and Chirp J signals are sequentially detected, and fed into the internal Suspend Timer.

The USB_Status.LineState[1:0] bits are used for the repeated "Chirp K-J-K-J-K-J" signal detection. Because the Chirp K and Chirp J transmission speed is very slow when compared with the normal HS packets, the USB_Status.LineState[1:0] bits can be used for their signal detection. However, if the bus signals are sent together with USB_Status.LineState[1:0] bits during usual packet reception, they are very noisy. If the presence of bus activity is determined when the XcvrControl.TermSelect bit is HS mode, the "J" state is output by USB_Status.LineState[1:0] bits. If the absence of bus activity is determined, the "SE0" state is output.

In the following figure, the "Chirp" signal height changes after T6. It means that the HS termination of the device is enabled by the XcvrControl.TermSelect bit. If the XcvrControl.TermSelect bit is FS mode, the "Chirp" signal is usually approx. 800 mV. If the XcvrControl.TermSelect bit is HS mode, the "Chirp" signal is approx. 400 mV (this is similar to the normal send and receive packets in HS mode).

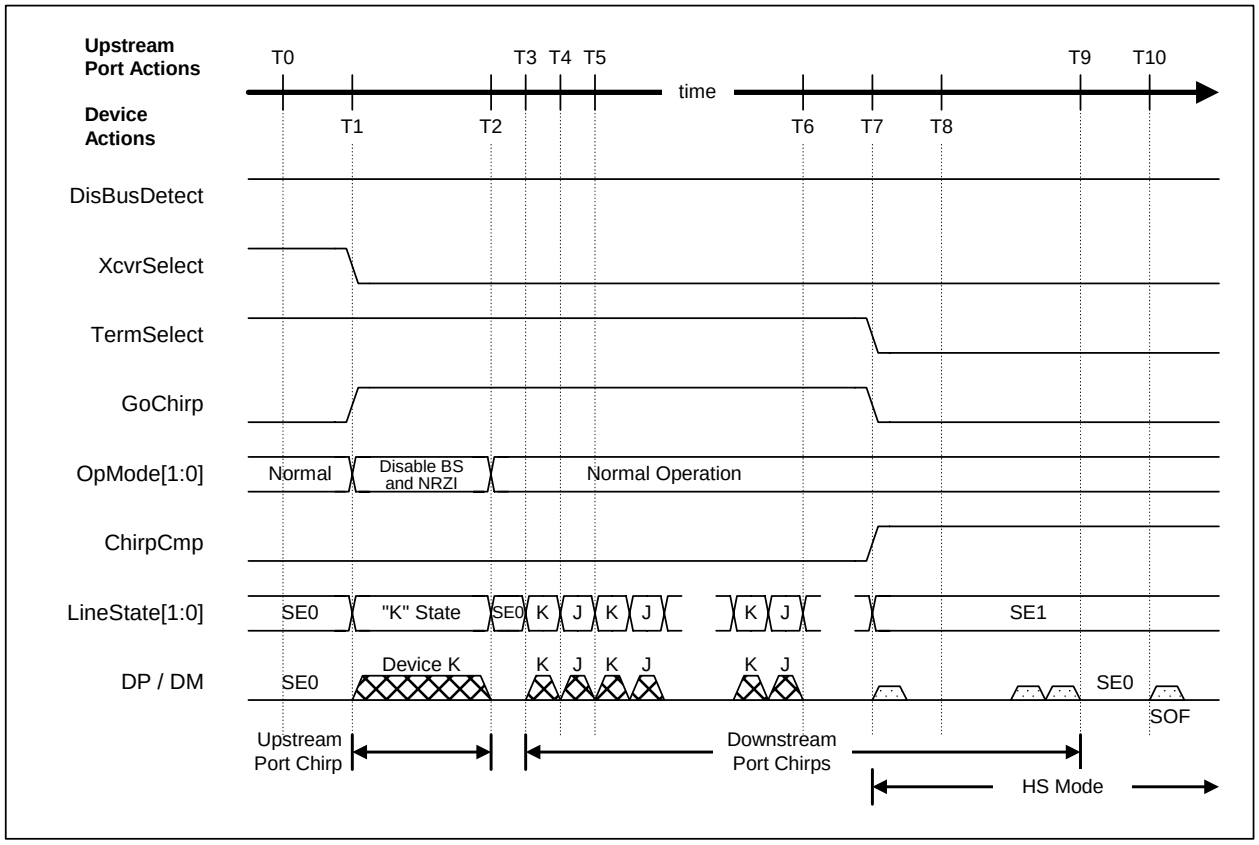


Fig.A2.6.13 HS Detection Handshake Timing (HS mode)

Table A2.6.10 HS Detection Handshake Timing Values (HS mode)

Timing Parameter	Description	Values
T0	Start of HS Detection Handshaking	0(Reference)
T1	The HS transceiver is enabled, and GoChirp signal is set to 1 and “Chirp K” transmission starts.	T0 < T1 < HS Reset T0 + 6.0ms
T2	“Chirp K” transmission has completed. It must be sent at least 1 msec.	T1 + 1.0ms {T_{UCH}} < T2 HS Reset T0 + 7.0ms {T_{UCHEND}}
T3	The downstream port sends the first “Chirp K” signal onto the bus.	T2 < T3 < T2 + 100us {T_{WTDCH}}
T4	The downstream port switches the signal from “Chirp K” to “Chirp J” and sends it onto the bus.	T3 + 40us {T_{DCHBIT}(MIN)} < T4 < T3 + 60us {T_{DCHBIT}(MAX)}
T5	The downstream port switches the signal from “Chirp J” to “Chirp K” and sends it onto the bus.	T4 + 40us {T_{DCHBIT}(MIN)} < T5 < T4 + 60us {T_{DCHBIT}(MAX)}
T6	The “Chirp K-J-K-J-K-J” signals are detected in this order.	T6
T7	When “Chirp K-J-K-J-K-J” signals are received in this order, the FS termination is disabled but the HS termination is enabled. The ChirpCmp signal is set to 1 and the end of reset is waited.	T6 < T7 < T6 + 500us
T8	The bus activity is recognized by the Chirp K and Chirp J signals. It is not recognized erroneously as the packet receive state because no SYNC signal is detected.	T8
T9	The downstream port terminates the Chirp K and Chirp J signal transmission.	T10 - 500us {T_{DCHSE0}(MAX)} < T9 < T10 - 100us {T_{DCHSE0}(MIN)}
T10	End of reset sequence	HS Rest T0 + 10ms {T_{DRST}(MIN)}

Note: The signal name in braces { } has been defined by the USB2.0 Standard manual.

Note: 66000 cycles (for 60MHz internal clock) are used for determination to generate 1-msec minimum “Chirp K” signal.

A2.6.2.7.11.7 If Device is Reset during SNOOZE State

This LSI does not output its internal clock when it is in the SNOOZE state. The following explains the device operations by assuming that the oscillator circuit is operating (in the SNOOZE state, but not in the SLEEP state).

If a Reset signal is detected (at T0) when the device is in the SNOOZE state, the SIE_IntStat.NonJ bit is set. Also, the SIE_IntEnb.EnNonJ and DeviceIntEnb.EnSIE_IntStat bits are set. If the MainIntEnb.EnDeviceIntStat bit is set, an interrupt signal is asserted simultaneously. During this time, to immediately recover from the SNOOZE state and to transition to the Reset sequence, set the PM_Control_0.GoActDevice bit to 1 (at T1). After the PLL power-up time has passed (at T2), the PM_Control_1.PM_State[2:0] bits are set to "ACT_DEVICE" and the output of internal clock starts. After this time, start HS Detection Handshaking (as described before).

If the oscillator circuit is not stopped (not restored from the SLEEP state), the internal clock having the frequency defined by the USB2.0 Standard is output.

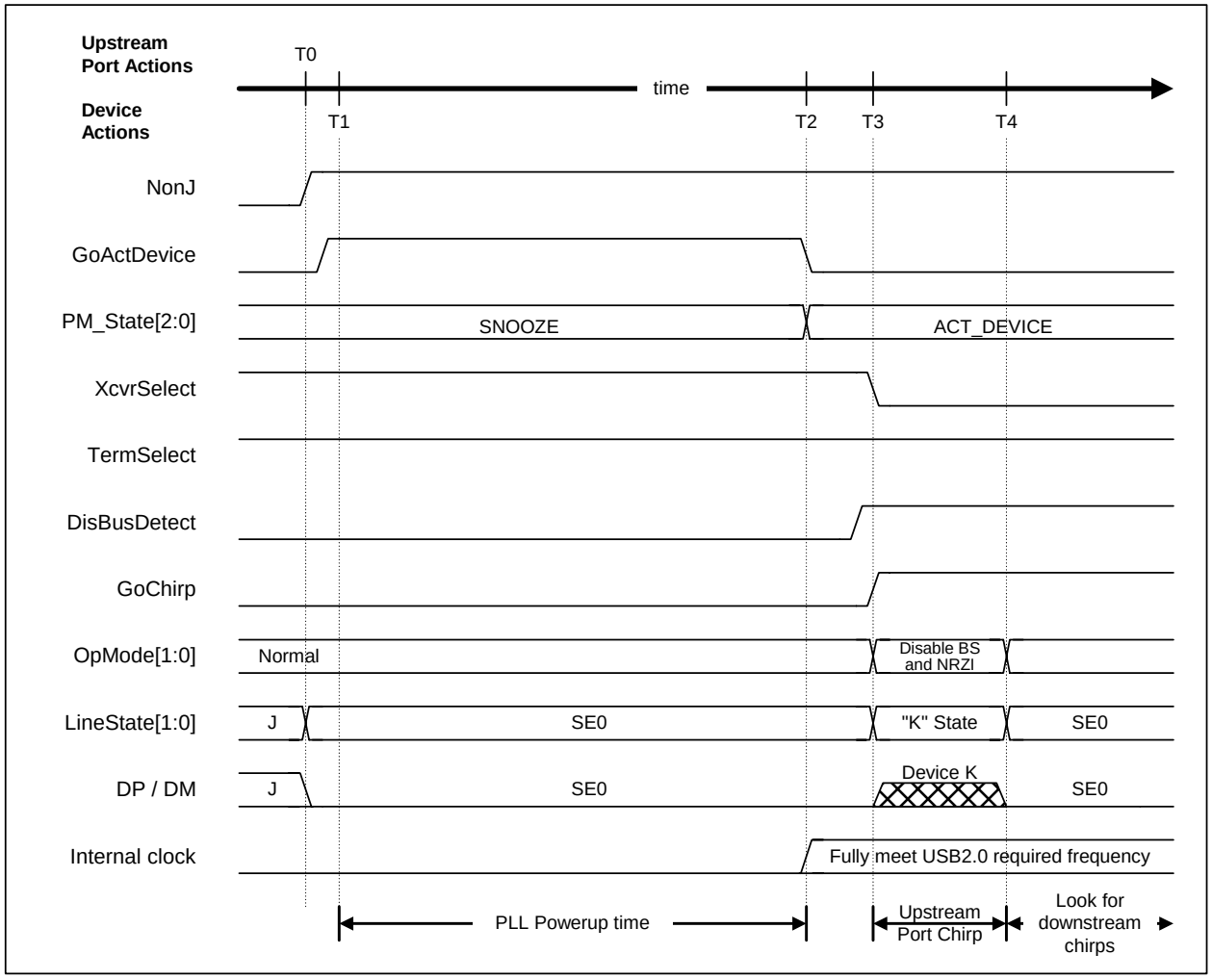


Fig.A2.6.14 HS Detection Handshake Timing from Suspend

Table A2.6.11 HS Detection Handshake Timing Values from Suspend

Timing Parameter	Description	Values
T0	If NonJ signal is set to 1 and if “SE0” state is determined by LineState[1:0] bits, the Reset signal is detected in the SNOOZE state.	0（HS Reset T0）
T1	After the Reset signal has been detected, the GoActDevice signal is set to 1.	T1
T2	The PM_State becomes “ACT_DEVICE”. The stable internal clock is output.	T1 + 250us < T2
T3	The GoChirp signal is set to 1, and the Chirp K signal is sent onto the bus. (Set the DisBusDetect signal to 1 before Chirp K signal transmission.)	T2 < T3 < HS Reset T0 + 5.8ms
T4	“Chirp K” transmission terminates.	T3 + 1.0ms {T_{UCH}} < T4 < HS Reset T0 + 7.0ms {T_{UCHEND}}

Note: The signal name in braces { } has been defined by the USB2.0 Standard manual.

Note: 66000 cycles (for 60MHz internal clock) are used for determination to generate 1-msec minimum “Chirp K” signal.

Note: If the oscillator circuit is also stopped (in the SLEEP state), the device reset operation is described later. (The OSC power-up time is required in addition to the PLL power-up time.)

A2.6.2.7.11.8 Issue of Resume Signal

The following explains how the device can resume with certain factors if the Remote Wakeup function is supported and if this Remote Wakeup is enabled by the host. However, the Remote Wakeup must start at least 5 msec have passed after the bus has become the Idle state. Also, if 10 msec have not passed yet after the output of Resume signal, the current used for the USB device to enter the Suspend state cannot be supplied from the VBUS pin.

The device must be restored from the SLEEP or SNOOZE state before the Remote Wakeup is used. When the SIE_IntEnb. EnNonJ bit is cleared, the PM_Control_0.GoActDevice bit is set (at T0), and the PLL power-up time has passed (at T1), the PM_Control_1.PM_State[2:0] bits become "ACT_DEVICE" and the internal clock starts to be output. If the oscillator circuit is not stopped at this time, the internal clock having the frequency defined by the USB2.0 Standard is output.

Then, the NegoControl.SendWakeup bit is set and the Resume signal is sent (at T2). At this time, the XcvrControl.OpMode[1:0] bits are internally set to "Disable Bit Stuffing and NRZI encoding" state. The “0” send data is prepared, the packet transmission mode is set, and the “K” (Resume) signal is sent. The downstream port detects this Resume signal, and returns the “K” (Resume) signal onto the bus (at T3).

If the NegoControl.SendWakeup bit is cleared approx. 1 msec after the start of Resume signal transmission, the Resume signal transmission is stopped (at T4). At this time, the downstream port still holds the bus in the Resume signal state.

Therefore, the NegoControl.RestoreUSB bit must be set. After the specific time has passed, the downstream port stops sending the Resume signal (at T5), sends the 2-bit LS-EOP (2*SE0) signal, and returns to the signal transmission speed mode before the USB was suspended. When this status change is detected (when the non-K signal is detected), both XcvrControl.XcvrSelect and XcvrControl.TermSelect bits are switched to the desired mode (HS mode in this case). The NegoControl.RestoreUSB bit is cleared, and the SIE_IntStat.RestoreCmp bit is set. At this time, the SIE_IntEnb.EnRestoreCmp and DeviceIntEnb.EnSIE_IntStat bits are set . If the MainIntEnb.EnDeviceIntStat bit has been set, an interrupt signal is asserted simultaneously.

If the USB’s Suspend mode is started, its speed mode (HS or FS mode) is kept by USB_Status.FSxHS bits and the USB device is returned to the mode specified by the USB_Status.FSxHS bit if reset by Resume operation. During this time, no HS Detection Handshaking is required for each Resume operation. Note that this section explains the device operation if the USB device was in the HS mode before it was suspended. If the device was in the FS mode, the normal FS mode continues after T5. There is no large difference in the signal sequence.

When this LSI is in the SNOOZE state (the PM_Control_1.PM_State[2:0] bits are “SNOOZE”), the internal clock is not output. However, this section explains the device operations by assuming that the oscillator circuit is operating (in the SNOOZE state, but not in the SLEEP state).

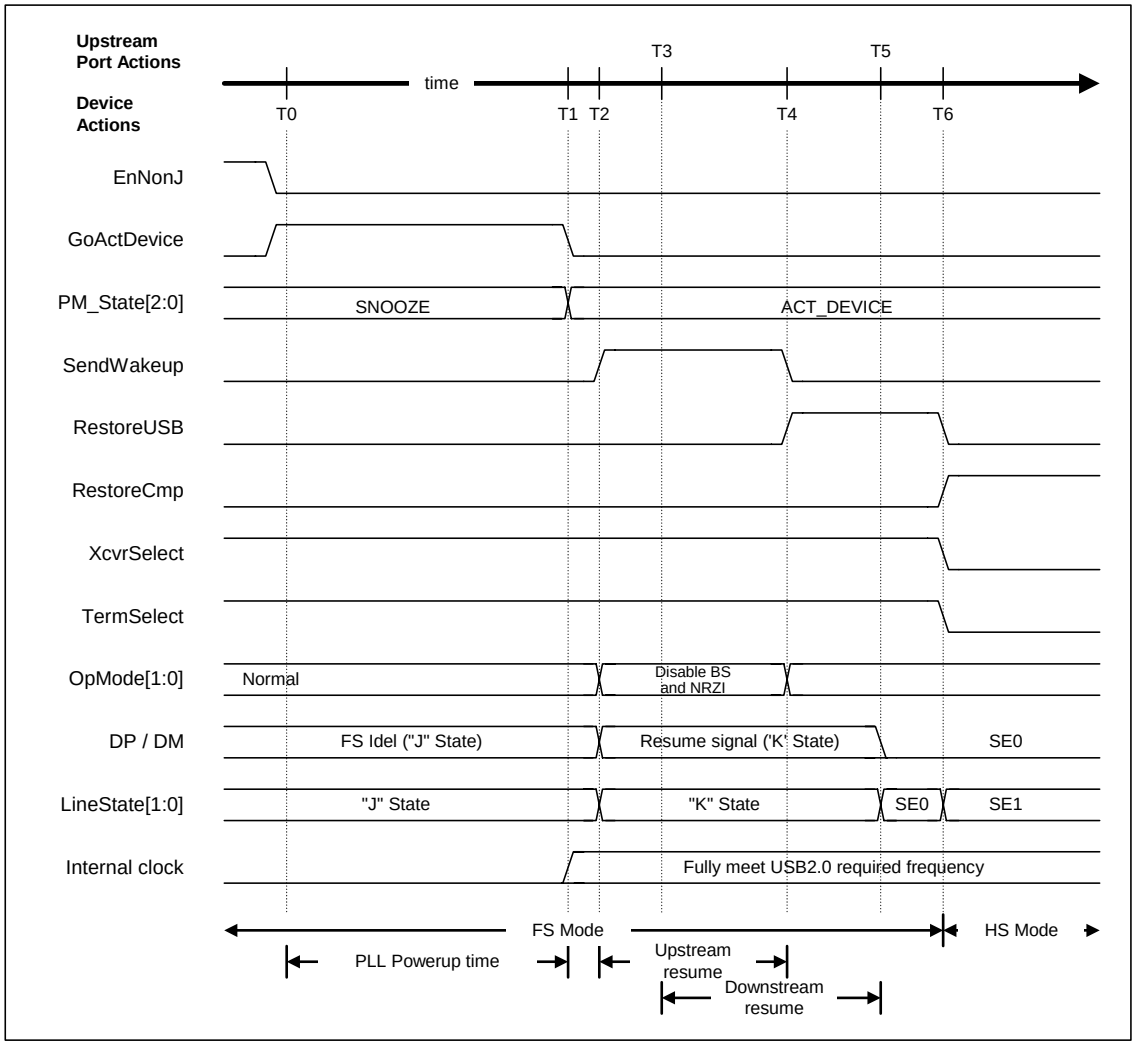


Fig.A2.6.15 Assert Resume Timing (HS mode)

Table A2.6.12 Assert Resume Timing Values (HS mode)

Timing Parameter	Description	Values
T0	Start of resume operation. The GoActDevice signal is set to 1. (The EnNonJ signal must be cleared (to 0) before the start of resume operation.)	0（Reference）
T1	The PM_State becomes “ACT_DEVICE”. The stable internal clock is output.	T0 + 250us < T1
T2	The SendWakeup signal is set to 1, and the FS “K” signal transmission starts. The current before the USB was suspended cannot be used from this pin within 10 msec.	T0 < T2 < T0 + 10ms
T3	The downstream port returns the FS “K” signal.	T2 < T3 < T2 + 1.0ms
T4	The SendWakeup signal is cleared (to 0), and the FS “K” transmission terminates. The “K” state is checked by the LineState[1:0] signal, and the RestoreUSB signal is set to 1.	T2 + 1.0ms {T_{DRSMUP}(MIN)} < T4 < T2 + 15ms {T_{DRSMUP}(MAX)}
T5	The downstream port terminates the FS “K” transmission.	T2 + 20ms {T_{DRSMDN}}
T6	The RestoreCmp signal is set to 1. If the USB device was in HS mode before suspended, it transitions to the HS mode automatically.	T5 + 1.33us {2 Low-speed bit time}

Note: The signal name in braces { } has been defined by the USB2.0 Standard manual.

A2.6.2.7.11.9 Detection of Resume Signal

When this LSI is in the SNOOZE state, the “J” signal is observed (USB_Status.LineState[1:0] is “J” state) on the bus. If the “K” state signal is observed on the bus, it means that the Wakeup request (the Resume command) has been received from a downstream port. (at T0). At this time, if the oscillator circuit is not stopped (that is, not in the SLEEP state), the SIE_IntStat.NonJ bit is set. If the SIE_IntEnb.EnNonJ and DeviceIntEnb.EnSIE_IntStat bits are set and if the MainIntEnb.EnDeviceIntStat bit is set, an interrupt signal is asserted simultaneously.

First, the PM_Control_0.GoActDevice bit is set to 1 (at T1). When the PLL power-up time has passed (at T2), the PM_Control_1.PM_State[2:0] signal becomes the “ACT_DEVICE” state and the output of internal clock starts simultaneously. If the oscillator circuit is not stopped at this time, the internal clock having the frequency defined by the USB2.0 Standard is output.

Therefore, the NegoControl.RestoreUSB bit must be set. After the specific time has passed, the downstream port stops sending the Resume signal (at T3), and returns to the signal transmission speed mode before the USB was suspended. When this status change is detected (when the non-K signal is detected), both XcvrControl.XcvrSelect and XcvrControl.TermSelect bits are switched to the desired mode (HS mode in this case). The NegoControl.RestoreUSB bit is cleared, and the SIE_IntStat.RestoreCmp bit is set. At this time, the SIE_IntEnb.EnRestoreCmp and DeviceIntEnb.EnSIE_IntStat bits are set . If the MainIntEnb.EnDeviceIntStat bit has been set, an interrupt signal is asserted simultaneously.

However, this section explains the device operations by assuming that the oscillator circuit is operating (in the SNOOZE state, but not in the SLEEP state).

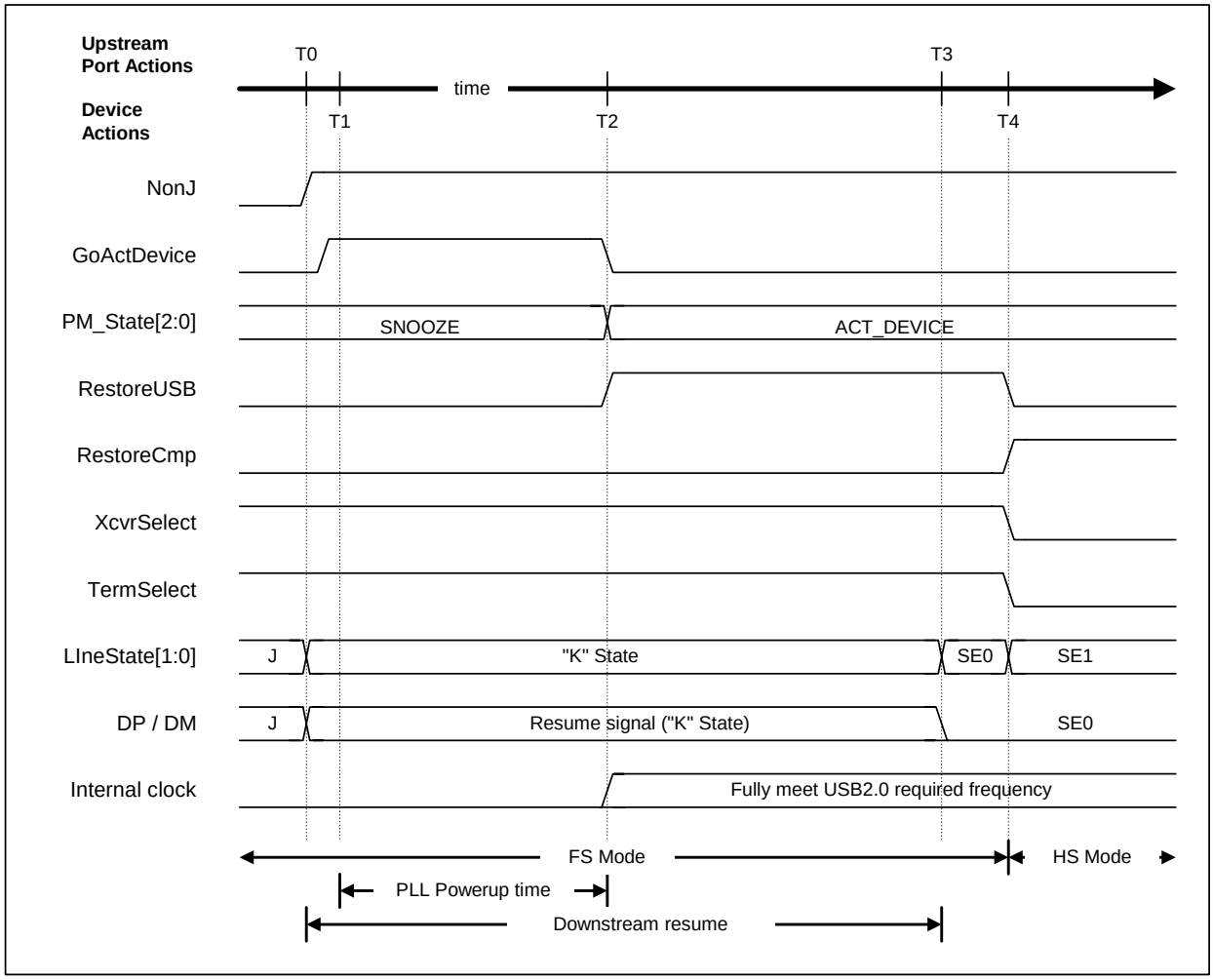


Fig.A2.6.16 Detect Resume Timing (HS mode)

Table A2.6.13 Detect Resume Timing Values (HS mode)

Timing Parameter	Description	Values
T0	The downstream port sends the FS “K” state signal. The NonJ signal is set to 1.	0（Reference）
T1	The GoActDevice signal is set to 1.	T1
T2	The PM_State becomes “ACT_DEVICE”. The stable internal clock is output. The “K” state is checked by the LineState[1:0] signal, and the RestoreUSB signal is set to 1.	T1 + 250us < T2
T3	The downstream port terminates the FS “K” transmission. At the same time, the downstream port transitions to the HS mode before being suspended.	T2 + 20ms {T_{DRSMDN}}
T4	If the USB device was in HS mode before suspended, it transitions to the HS mode automatically.	T3 + 1.33us {2 Low-speed bit time}

Note: The signal name in braces { } has been defined by the USB2.0 Standard manual.

A2.6.2.7.11.10 Cable Insertion

The following explains the device operation when a device is connected to the hub or to the host, that is, when a cable is inserted.

If you have unplugged a cable or if you have intentionally disconnected a cable, set the XcvrControl.XcvrSelect bit to FS mode or set the XcvrControl.TermSelect bit to HS mode as default value.

If a cable is disconnected (at T0) and if the cable is connected, the VBUS signal goes High and the USB_Status.VBUS bit is set simultaneously (at T1). If the SNOOZE state has been set, the PM_Control_0. GoActDevice bit is set to 1 (at T2). After the PLL power-up time has passed (at T3), when the PM_Control_1.PM_State[2:0] signal becomes “ACT_DEVICE” state, the output of internal lock starts. To make a connection of FS device, you must set the XcvrControl.TermSelect bit to the FS mode temporarily (at T4).

Then, the downstream port sends a Reset signal (at T5) and the HS Detection Handshaking is started.

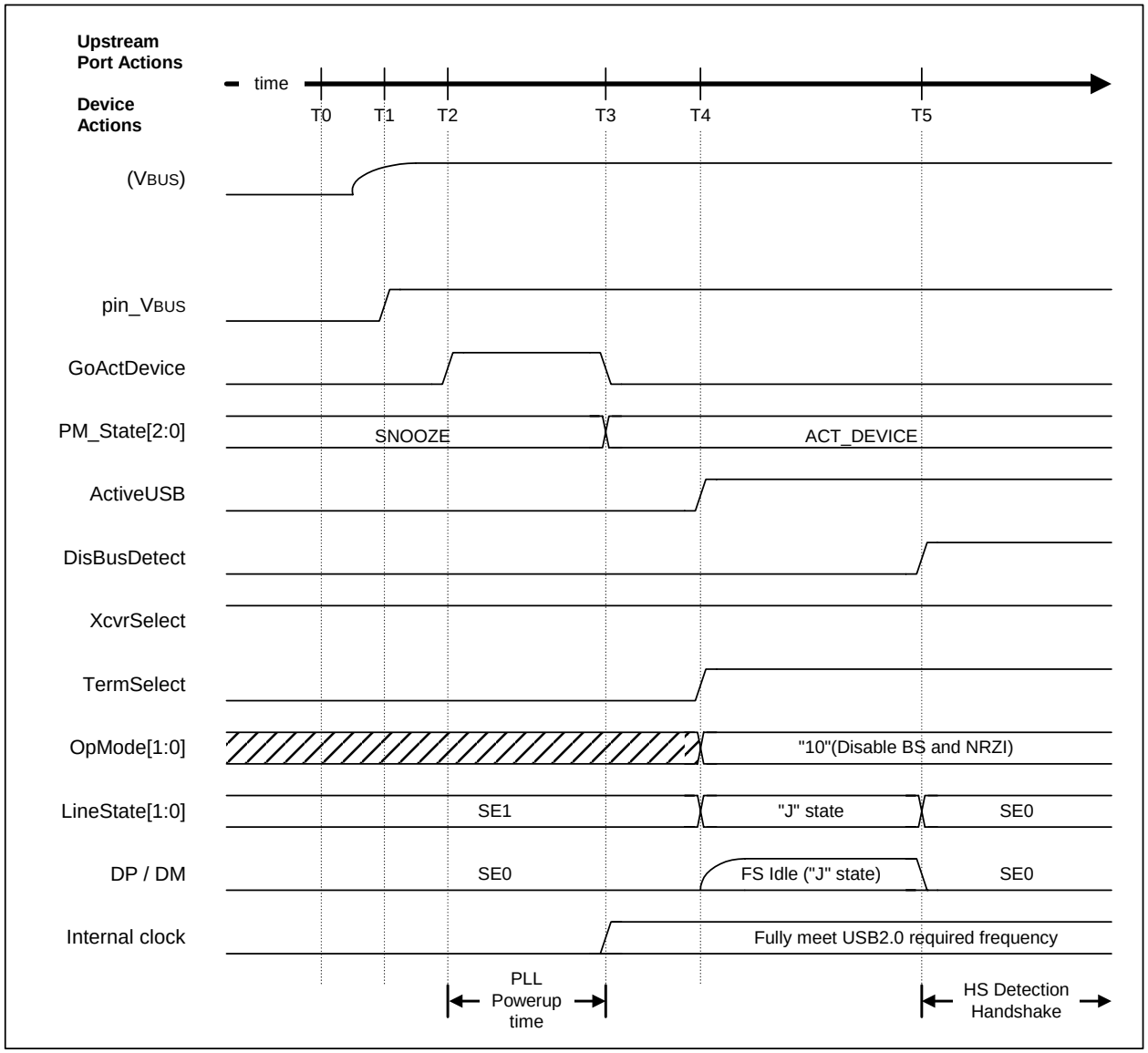


Fig.A2.6.17 Device Attach Timing

Table A2.6.14 Device Attach Timing Values

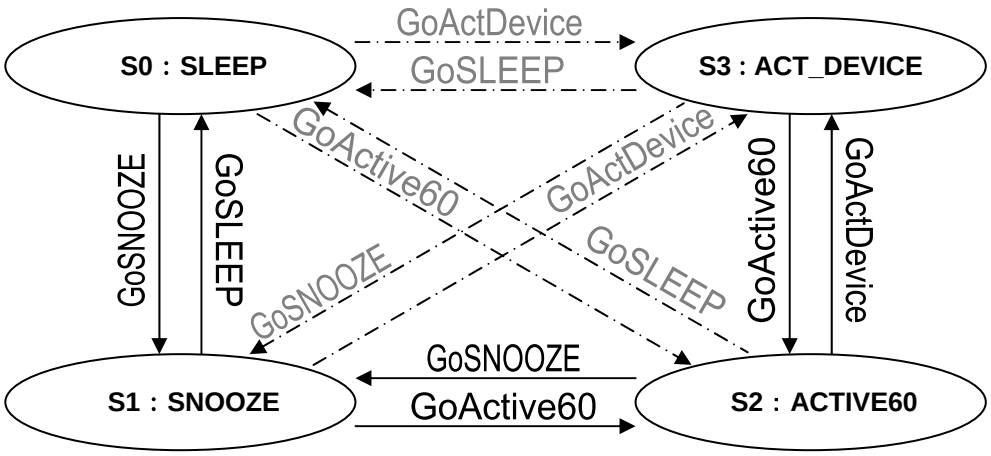
Timing Parameter	Description	Values
T0	The cable is not inserted.	0（Reference）
T1	The cable is inserted, and the VBUS input pin goes to the High.	T1
T2	The GoActDevice signal is set to 1.	T2
T3	The PM_State becomes “ACT_DEVICE”. The stable internal clock is output.	T2 + 250us < T3
T4	The ActiveUSB signal is set to 1. The TermSelect signal is set to 1. OpMode[1:0] is set to “00”. The device transitions to the FS mode. The FS termination is enabled.	T1 + 100ms {T_{SIGATT}} < T4
T5	The downstream port sends a Reset signal. The DisBusDetect signal is set to 1.	T4 + 100ms {T_{ATTDB}} < T5

Note: The signal name in braces { } has been defined by the USB2.0 Standard manual.

A2.6.3 Power Management Function

The oscillator and PLL (DevicePLL480) operations are controlled by the Power Management function, and the status transition occurs between four states: the SLEEP, SNOOZE, ACTIVE60, and ACT_DEVICE states. The device transitions to another state when the PM_Control_0.GoSLEEP, PM_Control_0.GoSNOOZE, PM_Control_0.GoActive60, or PM_Control_0.GoActDevice bit is set. After its processing has completed, its state terminates. The current device state can be checked by referring to the PM_Control_1.PM_State[3:0] bits. When the status transition has completed, a MainIntStat.FinishedPM event occurs. During this time, if the MainIntEnb.EnFinishedPM bit is set and if the MainIntEnb.EnSIE_IntStat bit is set, an interrupt occurs.

The device can transition from any state to any other state. If the PM_Control.GoSLEEP bit is set when the device is in the ACT_DEVICE state, it transitions to the SLEEP state through the ACTIVE60 state and through the SNOOZE state. When its status transition has completed, a SIE_IntStat.FinishedPM event occurs. Also, if the PM_Control.GoActDevice bit is set when the device is in the SLEEP state, it transitions to the ACT_DEVICE state through the SNOOZE state and through the ACTIVE60 state. When its status transition has completed, a MainIntStat.FinishedPM event occurs. Similarly, if the PM_Control.GoSLEEP bit is set when the device is in the ACTIVE60 state, it transitions to the SLEEP state through the SNOOZE state. When its status transition has completed, a MainIntStat.FinishedPM event occurs. Also, if the PM_Control_0.GoActive60 bit is set when the device is in the SLEEP state, it transitions to the ACTIVE60 state through the SNOOZE state. When its status transition has completed, a MainIntStat.FinishedPM event occurs.



Note: The status transition shown by alternate long and short dash lines actually occurs through the path shown by the full line.

Fig.A2.6.18 Power Management

Cautions: The S2S65A00 does not have the dedicate 60MHz PLL circuit. Although the “ACTIVE60” state is supported by the Power Management function, this ACTIVE60 state is logically the same as the SNOOZE state inside of the LSI.

A2.6.3.1 SLEEP State

The oscillator is not operating in the SLEEP state. Therefore, the PLL circuit is not oscillating in this state.

If the device is in the SNOOZE, ACTIVE60, or ACT_DEVICE state and if you need to transition the device to the SLEEP state by setting the PM_Control_0.GoSLEEP bit, you must first stop both the PLL and OSC circuits from operating. Actually, after you have stopped the OSC CLK output, you must stop the oscillation.

While if the device is transitioned form the SLEEP state to the SNOOZE state by setting the PM_Control_0.GoSNOOZE, PM_Control_0.GoActve60, an d PM_Control. GoActDevice bit, it is gated for the time until the oscillation becomes stable. The OSC CLK signal is not entered in the internal circuit. Because this oscillation stabilization time varies depending on the used oscillator cell, oscillator device, peripheral circuits, and the board, you must set an appropriate time using the WakeUpTim_H,L registers.

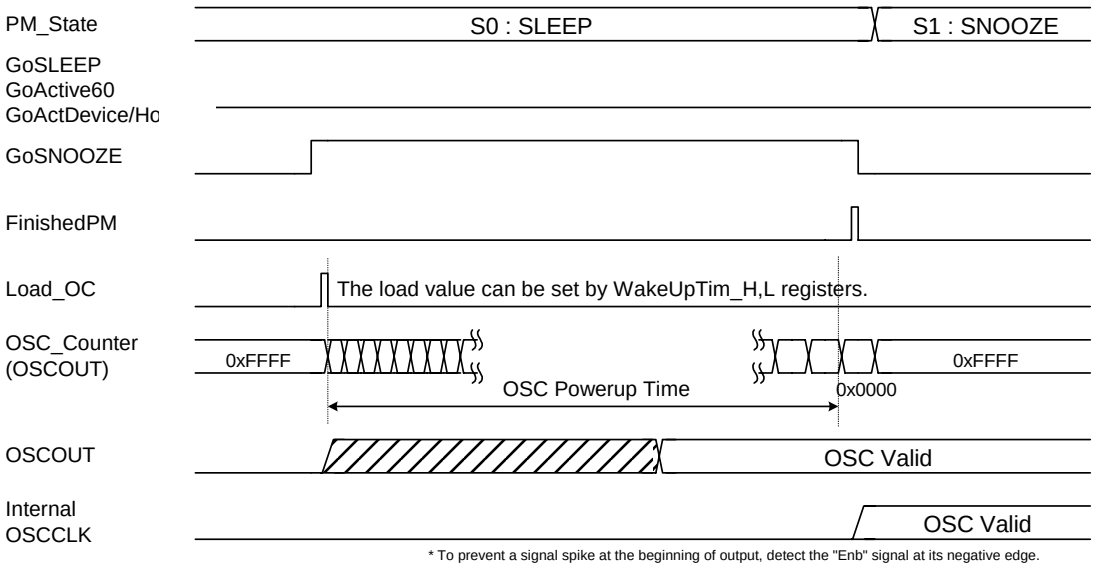


Fig.A2.6.19 Transition from SLEEP State (by GoSNOOZE)

A2.6.3.2 SNOOZE State

The oscillator is operating but the PLL circuit is not oscillating in the SNOOZE state.

If you transition the device from the ACTIVE60 state or ACT_DEVICE state to the SNOOZE state by setting the PM_Control_0.GoSNOOZE bit, stop the current clock output and stop the DevicePLL480 from oscillating.

However, if the device is transitioned from the SNOOZE state to an active state by setting the PM_Control_0.GoActDevice and PM_Control_0.GoActive60 bits, they are gated for the time the PLL oscillation becomes stable (approx. 250 μsec). The SCLK signal is not entered in the internal circuit until the PLL oscillation becomes stable.

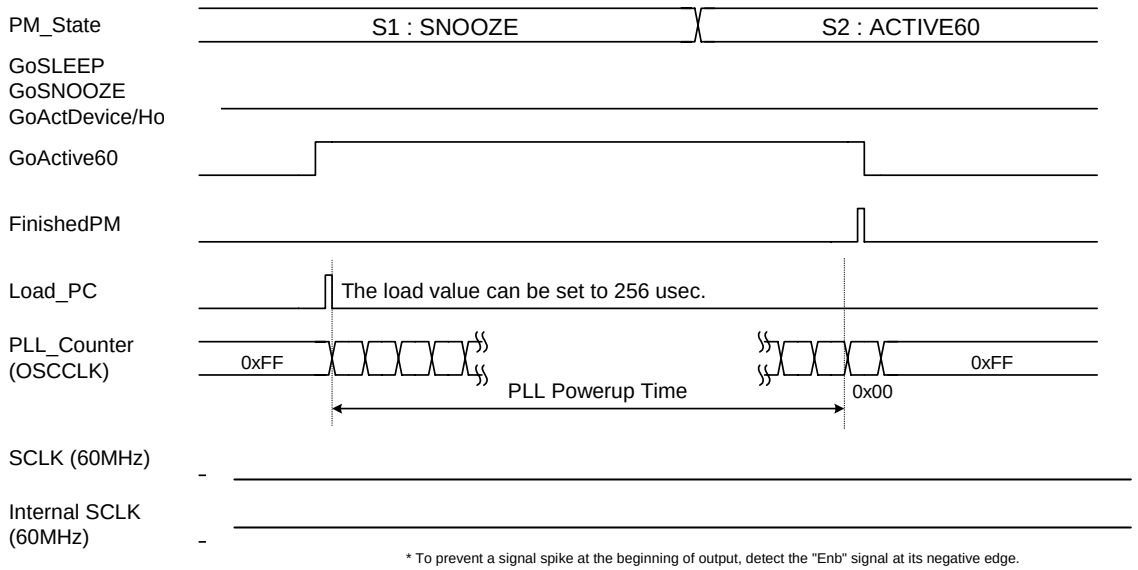


Fig.A2.6.20 Transition from SNOOZE State (by GoActive60)

A2.6.3.3 ACTIVE60 State

As the S2S65A00 does not have the dedicate 60MHz PLL circuit, the ACTIVE60 state is the same as the SNOOZE state. The transition from the SNOOZE state to the ACTIVE60 state takes approx. 256 μsec as illustrated in Figure A2.6.20.

A2.6.3.4 ACT_DEVICE State

The oscillator and DevicePLL480 operates in the ACT_DEVICE state. The registers and bits shown in italic characters on the register map can be read and written even in the SNOOZE or SLEEP state. However, the registers shown by non-italic characters can be read and written in the ACT_DEVICE state only.

Also, the USB device circuits can operate in the ACT_DEVICE state only.

A2.6.4 FIFO Memory Management

A2.6.4.1 FIFO Memory Map

The following shows the standard FIFO memory map.

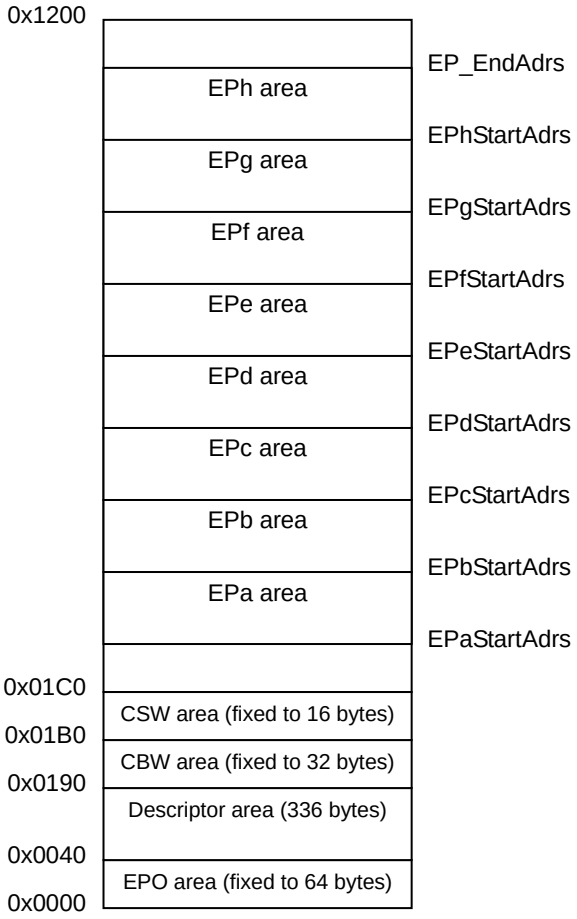


Fig.A2.6.21 FIFO Memory Map (Example)

The FIFO memory can be divided into 12 areas maximum and used. They are: EP0 area, descriptor area, CBW area, CSW area, and EPa to EPh areas. Each of EP0 area, descriptor area, CBW area, and CSW area is assigned a fixed size space as illustrated in Figure A2.6.21. While the other EPx{x=a-h} area size can be set flexibly using FIFO area setting registers (EPx{x=a-h}StartAdrs_H,L, and EP_EndAdrs_H,L).

The EP0 area is used for endpoint 0 that is always required for the USB, and it may be used in both IN and OUT directions. Although this area is assigned 64 bytes, only the area that begins at address 0x000 and has the maximum packet size of endpoint 0 can be used. Therefore, endpoint 0 is always the single buffer.

The descriptor area is used by the Descriptor Reply function. This area is assigned 336 bytes, and a space area beginning at any position can be used. The actual applications are described later in Paragraph A2.6.4.2. Although the entire FIFO memory area can be used for the Descriptor Reply function, we recommend you to use only the area space explained here as the descriptor area in order to prevent a possible area competition.

The CBW area is used for CBW support of the Bulk Only Support function. Although 32 bytes are assigned, the 31-byte area beginning at address 0x190 is used. The actual applications are described later in Paragraph A2.6.4.3.

The CSW area is used for CSW support of the Bulk Only Support function. Although 16 bytes are assigned, the 13-byte area beginning at address 0x1B0 is used. The actual applications are described later in Paragraph A2.6.4.4.

The EPa to EPh areas are generic endpoint areas, and their endpoint numbers and IN or OUT direction can be set as desired.

The EP0 and EPa to EPh areas are controlled as FIFO memory areas, and their data storage count is held. To clear this hold status, set the EPnControl.AllFIFO_Clr or EPnControl.EP0FIFO_Clr bit or set each of EPrFIFO_Clr.EPx{x=a-h}FIFO_Clr bits that correspond to each area.

Note that this status clear means the initialization of data hold information only. No data is written or cleared. Therefore, data in the RAM is not cleared by this bit. The information recorded in the descriptor area is not erased, and you need not write the data again after clearing of the status data.

A2.6.4.2 Using the Descriptor Area

The descriptor area is used for the Descriptor Reply function. The Descriptor Reply function can be used if data stage IN transfer occurs at endpoint 0.

To start the IN direction data stage, set both the first address of data written in this area and the size of reply data. When you execute the Descriptor Reply function, the data stage is processed automatically.

If the device descriptor and other data uniquely determined by the system device are already written in the descriptor area, the data of this area can be returned automatically when its request is issued. Because those data need not be written in the EP0 area for each request, the system can quickly respond to the request.

A2.6.4.2.1 Writing Data in Descriptor Area

To write data in the descriptor area, use the RAM_WrDoor function. Set the start address for data writing in RAM_WrAdrs_H, L registers, and write data in RAM_WrDoor_0, 1 registers. The RAM_WrAdrs_H, L register values are updated when the data write count is incremented for each data writing. For data writing in continuous addresses, you can write data in RAM_WrDoor_0, 1 registers continuously.

Note that the RAM_WrDoor_0, 1 registers are used for data writing only.

A2.6.4.2.2 Executing the Data Stage (IN-Direction) in Descriptor Area

When using the written data with the Descriptor Reply function, set the first address of data sent to the data stage in DescAdrs_H, L registers, set the reply data size in DescSize_H, L registers, and set the EP0Control.ReplyDescriptor bit to 1. Also, set the EP0Control.INxOUT bit to 1 to enable IN transactions. Also remember that you must clear the SETUP_Control.Protect EP0 bit first, then clear the EP0Control_IN.ForceNAK bit so that data packets can be sent to the IN transaction of the data stage.

After these settings, the device responds to the IN transaction from the host, automatically divides data in the maximum packet size (set by EP0MaxSize), and replies data packets of data count being set in DescSize_H, L registers back to the host. If the DescSize_H, L register values are less than the maximum packet size or if the data count after the data division is less than the maximum packet size, a short packet is sent automatically.

When the OUT transaction is issued by the host, the EP0Control.ReplyDescriptor bit is cleared but the FIFO_IntStat. DescriptorCmp bit is set. The firmware needs to transition to the Status stage.

A2.6.4.3 Using the CBW Area

The CBW area is used for CBW support of the Bulk Only Support function. If the Command transport of the Bulk Only Transport protocol is used at the Bulk OUT endpoint, data can be received in this area. This can simplify DMA and other transfer control because only data can be received at the endpoint FIFO memory.

A2.6.4.3.1 Receiving Data in CBW Area

During CBW support execution, the OUT transaction is executed at the target endpoint and, if the data is 31 bytes long, it is received in the CBW area. If the data length is not less than 31 bytes, an error status is issued and the data is discarded.

A2.6.4.3.2 Reading Data from CBW Area

To read data that has been received in the CBW area, use the RAM_Rd function. If RAM_RdControl and RAM_GoRdCBW_CSW bits are set, data is read from the CBW area and copied in RAM_Rd_00 to RAM_Rd_1F registers. Then, the completion status (CPU_IntStat.RAM_RdCmp bit) is issued.

A2.6.4.4 Using the CSW Area

The CSW area is used for CSW support of the Bulk Only Support function. If the Status transport of the Bulk Only Transport protocol is used at the Bulk IN endpoint, data can be sent from this area. This can simplify DMA and other transfer control because only data can be sent from the endpoint FIFO memory.

A2.6.4.4.1 Sending Data from CSW Area

If the IN transaction is executed at the target endpoint during CSW support execution, the 13-byte data is sent as the data packet from the CSW area.

A2.6.4.4.2 Writing Data in CSW Area

To write data in the CSW area, use the RAM_WrDoor function. Write the first address (0x1B0) of CSW area in RAM_WrAdrs_H, L registers, and write the 13-byte significant data through RAM_WrDoor_0, 1 registers. When 14-byte data is written with word access, it does not violate another area because the CSW area is assigned 16 bytes.

A2.6.4.5 Accessing to FIFO Memory

Access cause to FIFO memory can be the CPU (registers), DMA transfer, or USB interface.

A2.6.4.5.1 Accessing to FIFO Memory (RAM_Rd)

To allow read access to FIFO memory using RAM_Rd registers of CPUIF, set the first address of FIFO area and the read data size in the respective RAM_RdAdrs_H, L and RAM_RdCount registers, and set the RAM_RdControl.RAM_GoRd bit. When the specified FIFO area data becomes ready to read with RAM_Rd registers, the CPU_IntStat.RAM_RdCmp bit is set to 1. Check the RAM_RdCmp bit status, and read the data from RAM_Rd registers. The RAM_Rd register data is stored in the RAM_Rd_00 and subsequent places sequentially. If a data size less than 32 bytes is set in the RAM_RdCount register, the RAM_Rd register value exceeding the set size is made invalid.

The FIFO memory data can be read using RAM_Rd registers any time regardless of the current FIFO area setting of the channel.

When the RAM_Rd function is active, the RAM_RdAdrs_H, L and RAM_RdCount register values are updated sequentially. If the RAM_Rd function is just started, do not access to these registers until the CPU_IntStat.RAM_RdCmp bit is set. If these register values are read when the RAM_Rd function is operating, the read values are unreliable. Also, if data is written in these registers, a malfunction may result.

A2.6.4.5.2 Accessing to FIFO Memory (RAM_WrDoor)

During write access to FIFO memory using RAM_WrDoor_0, 1 registers of CPUIF, set the start address for data writing in RAM_WrAdrs_H, L registers and write data using RAM_WrDoor_0, 1 registers. The RAM_WrAdrs_H, L register values are automatically incremented each time a write access is made. For data writing in continuous addresses, you can write data in RAM_WrDoor_0, 1 registers continuously.

Data can be written in FIFO memory using RAM_WrDoor_0, 1 registers any time regardless of the current FIFO area setting of the channel.

A2.6.4.5.3 Register Access to FIFO Memory

To allow read access to FIFO memory with the CPU register access, set D_EPx{x=0,a-h}Join.JoinCPU_Rd bits to 1 for any single channel, and read memory data using FIFO_Rd_0, 1 registers or FIFO_ByteRd register.

While to allow write access to FIFO memory with the CPU register access, set D_EPx{x=0,a-h}Join.JoinCPU_Wr bits to 1 for any single channel, and write data in FIFO_Wr_0, 1 registers.

The FIFO_RdRemain_H, L registers indicate the remaining data count that can be read from FIFO memory through the single channel that has been set by EPx{x=0,a-h}Join.JoinCPU_Rd. The FIFO_WrRemain_H, L registers indicate the remaining FIFO memory area available for data writing through the single channel that has been set by D_EPx{x=0,a-h}Join.JoinCPU_Wr.

Note that if you dump register data to debug firmware using ICE or others, data is read from FIFO memory during register dump if any of D_EPx{x=0,a-h}Join.JoinCPU_Rd registers is set.

A2.6.4.5.3.1 Write Access to FIFO Memory

A write access to FIFO memory means data writing in FIFO_Wr_0, 1 registers.

The following restrictions apply to the write access to FIFO memory.

- Set the EPx{x=0,a-h}Join.JoinCPU_Wr bits, check the writable data count of FIFO_WrRemain_H, L registers, and access to FIFO memory. Basically, access to a single word (two-byte data) at a time. If you write odd-numbered bytes of data, you must control the access size by considering the FIFO byte boundaries. For details, see the “Rounding of FIFO Access” section.
- If you refer to FIFO_WrRemain_H, L registers immediately after data writing in FIFO_Wr_0, 1 registers, you cannot check the accurate space of FIFO memory. You must always wait for a single CPU cycle or more, and check those registers.

A2.6.4.5.3.2 Read Access to FIFO Memory

A read access to FIFO memory means data reading from FIFO_Rd_0, 1 registers and from FIFO_ByteRd register.

The following restrictions apply to the read access to FIFO memory.

- Set the EPx{x=0,a-h}Join.JoinCPU_Wr bits, check the readable data count and RdRemainValid bit of FIFO_RdRemain_H, L registers, and access to FIFO memory.
- Use FIFO_Rd_0, 1 registers for a word read. Use FIFO_ByteRd register for a byte read. If the byte boundaries exist, use the byte read. In this case, if you use the word read with FIFO_Rd_0, 1 registers, significant data is output in a single side only. For details, see the “Rounding of FIFO Access” section.

A2.6.4.5.3.3 Rounding of FIFO Access

The following explains the storage of odd-numbered data in FIFO memory and the access to it. Although the FIFO memory has the 4-byte width, the 2-byte memory width is used for your easy understanding in the following sections. There is no operational difference between 4-byte and 2-byte memory width.

[Data writing]

Basically, we recommend you to write data in FIFO memory having no byte boundary.

You should set the EPnControl.EP0FIFO_Clr and EPrFIFO_Clr.EPx{x=a-h}FIFO_Clr bits to clear the byte boundary, and writes word data in the memory. If an odd-numbered data exists, write only the last byte of continuous data (data Z) in the High side. This status is illustrated in Fig.A2.6.22 (1). Data sets A, B, C, D, ..., X, Y, and Z are output in this order from USB and other memories.

If the FIFO memory already has a byte boundary, first write data (data K) in the Low side to clear the byte boundary. Then, write word data (data L and M) in this memory. This status is illustrated in Fig.A2.6.22 (2).

This is the normal data write.

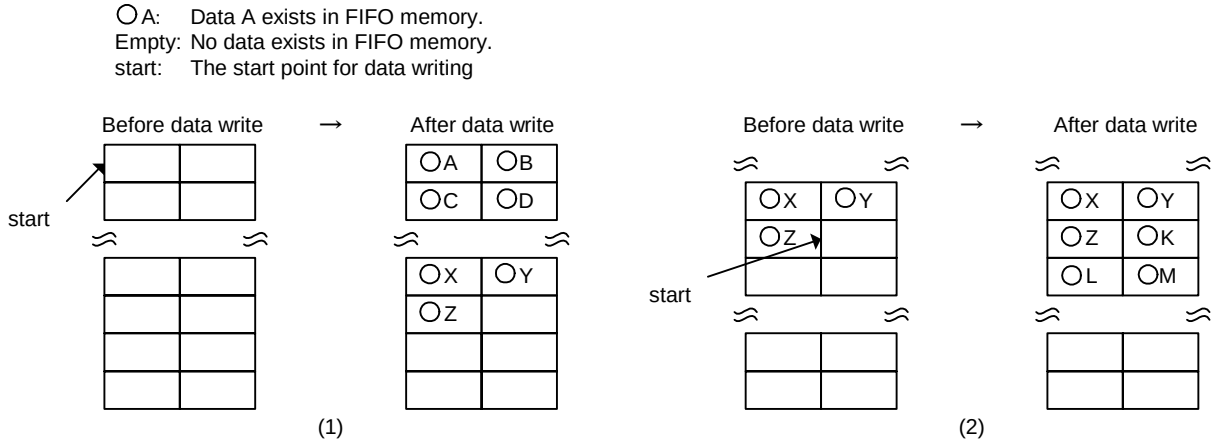


Fig. A2.6.22 FIFO Write Process (Normal operation)

The following explains the data write where certain cares must be taken.

If the FIFO memory already has a byte boundary and if you use the word write, the High-side data writing is ignored and data is written in the Low side only (see case (3) of Fig.A2.6.23). This operation is the same as a byte write in the Low side. Similarly, if the FIFO memory has a byte boundary and if you write data in the High side only, this writing is ignored (see case (4) of Fig.A2.6.23).

If the FIFO memory has no byte boundary and if you write data in the Low side only, this writing is ignored (see case (5) of Fig.A2.6.23). Also, if the FIFO memory has no byte boundary and if you use the word write when the writable data count is equal to 1, the data writing in the Low side is ignored and data is written in the High side only (see case (6) of Fig.A2.6.23). This operation is the same as a byte write in the High side.

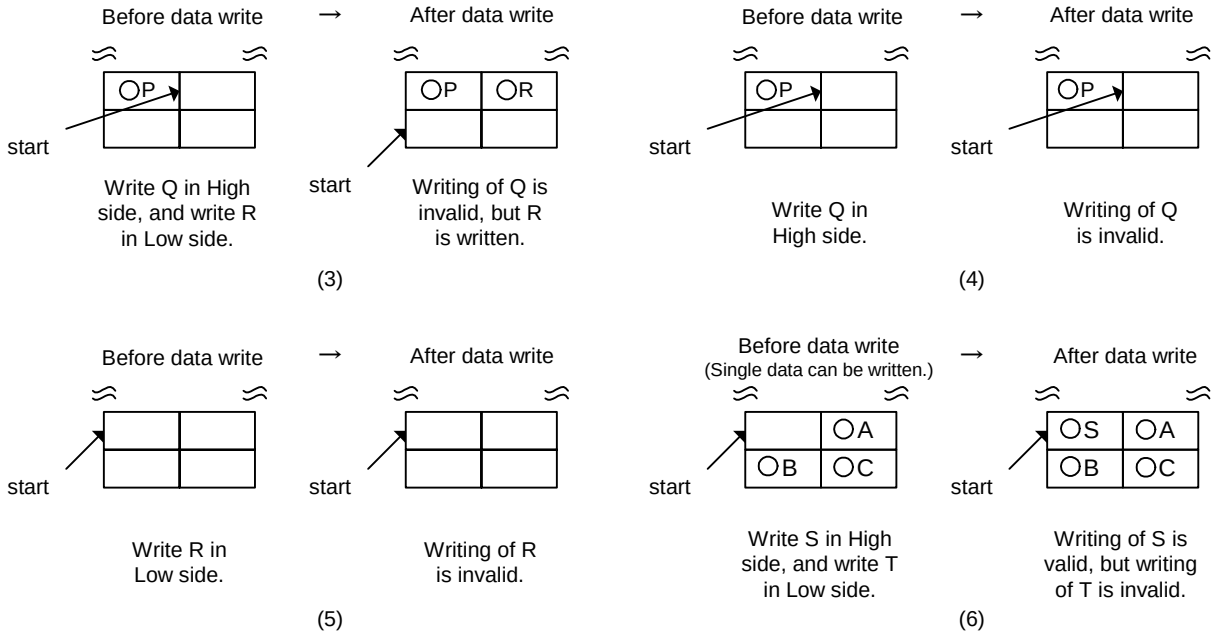


Fig. A2.6.23 FIFO Write Process (Operation requiring caution)

[Data reading]

If FIFO memory has no byte boundary, the word read with FIFO_Rd_0,1 registers or byte read with FIFO_ByteRd register can be used. However, if memory has a byte boundary, use the byte read with FIFO_ByteRd register. Once the byte boundary is cleared, both word read and byte read can be used without problem.

Item (1) of Fig.A2.6.24 shows the word read from memory having no byte boundary. Data A and B, and data C and D are read in this order for each memory access. Case (2) of Fig.A2.6.24 (2) shows the byte read. Data A, data B, data C, and data D are read in this order for each memory access. This is the normal data read.

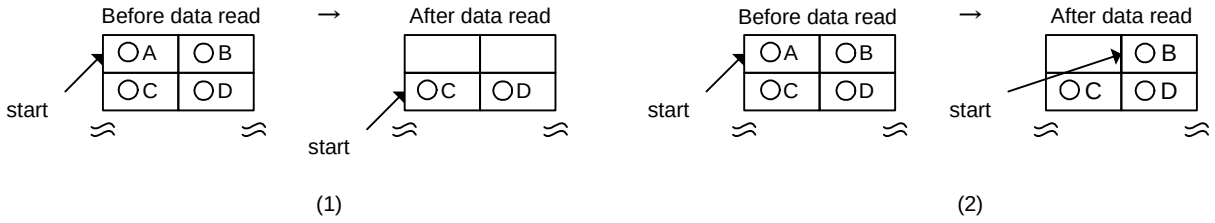


Fig. A2.6.24 FIFO Read Process (Normal operation)

The following explains the data read where certain cares must be taken.

Case (3) of Fig.A2.6.25 shows the word read from memory having a byte boundary, using FIFO_Rd_0,1 registers. Undefined data is output to the High side, and data J is output to the Low side. The read pointer is incremented for a single byte only. Case (4) of Fig.A2.6.25 shows the word read from memory using FIFO_Rd_0,1 registers if a single byte of data remains but no byte boundary exists in the memory. Data X is output to the High side, but undefined data is output to the Low side. The read pointer is incremented for a single byte only.

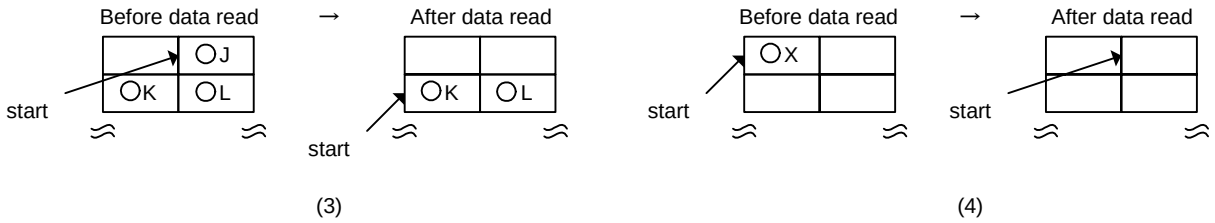


Fig. A2.6.25 FIFO Read Process (Operation requiring caution)

Using the above examples, the following explains the data read from FIFO memory with fraction rounding.

1) To read 31-byte data first, then read 33-byte data if 64-byte data is received from USB memory
   - ① The hardware latches a 64-byte data read ready signal, and starts its read sequence.
   - ② Read 30-byte data by the word read with FIFO_Rd_0,1 registers, or read it by the byte read with the FIFO_ByteRd register.
   - ③ Read the 31st byte data by the byte read with the FIFO_ByteRd register. → A byte boundary occurs.
   - ④ Read the 32nd byte data by the byte read. We recommend you to use the byte read with the FIFO_ByteRd register. If you use the word read with FIFO_Rd_0,1 registers, the data is output to the Low side. → The byte boundary is cleared.
   - ⑤ Read the remaining 32-byte data by the word read with FIFO_Rd_0,1 registers, or read it by the byte read with the FIFO_ByteRd register.

2) To read the entire 64-byte data by the word read using FIFO_Rd_0,1 registers if the JoinCPU_Rd bit is set and if 31-byte data and 33-byte data are received sequentially from USB memory
   - ① When receiving the 31-byte data from USB memory, the hardware latches the 31-byte data ready signal and start its read sequence.
   - ② Read the 30-byte data by the word read.
   - ③ To clear the byte boundary of the 31st byte data being cached, disconnect its join.
   - ④ After the 33-byte data has been sent from USB memory, connect the join again. (1 + 33 bytes)
   - ⑤ The CPU IF latches the 34-byte data read ready signal, and starts its read sequence.
   - ⑥ Read the 34-byte data by the word read.

A2.6.4.5.4 DMA Access to FIFO Memory

When you use the DMA access to FIFO memory for data reading, select only a single endpoint of each DMA channel by setting EPx{x=0,a-h}Join. JoinDMAx{x=0,1} bits. Set the DMAx{x=0,1}_Control.Dir bit to logical 1 and read data by executing the DMA procedure.

Also, when you use the DMA access to FIFO memory for data writing, select only a single endpoint of each DMA channel by setting EPx{x=0,a-h} Join.JoinDMAx{x=0,1} bits. Set the DMAx{x=0,1}_Control.Dir bit to logical 0 and write data by executing the DMA procedure.

The DMAx{x=0,1}_Remain_H,L registers show the remaining data count that can be read from FIFO memory at the endpoint of each DMA channel that has been selected by EPx{x=0,a-h}Join. JoinDMAx{x=0,1} bits. Also, the registers show the remaining FIFO area space where data can be written from the endpoint of each DMA channel that has been selected by EPx{x=0,a-h}Join.JoinDMAx{x=0,1} bits.

A2.6.4.5.5 Restricted Access to FIFO Memory

The FIFO memory of this LSI is used for data transmission to/from USB memory, the register write and read from CPU buses, and DMA write and read simultaneously. The look-ahead process is used to read data from the CPU bus.

For this reason, the following rules apply exclusively to select an access method (Join) to FIFO memory of each channel.

- Only one of JoinCPU_Wr, JoinCPU_Rd, and JoinDMAx{x=0,1} can be set at a single endpoint.
- Each of JoinCPU_Wr, JoinCPU_Rd, and JoinDMAx{x=0,1} can be set only at a single endpoint at a

time.

Also, the following restrictions apply to the access from USB.

- When data is being written in a FIFO area from the USB, no data can be written by any other cause.
- When data is being read from a FIFO area to the USB, no data can be read from it by any other cause.

For example, data can be written in the FIFO area of the OUT transaction endpoint by setting the JoinCPU_Wr bit. However, the data must be written from the CPU when the OUT transaction does not take place. Also, data can be read from FIFO area of IN transaction endpoint by setting the JoinCPU_Rd bit. However, the data must be read from the CPU when the IN transaction does not take place. The transaction does not take place if the ActiveUSB bit is cleared, if the EnEndpoint bit of each endpoint is cleared, or if the ForceNAK bit is set.

A2.6.5 DMA Transfer

A2.6.5.1 Description

This USB Controller has two channels of 8-bit DMA interface which are directly connected to the AHB bus (1). When linked to DMA controller 1, DMA transfer can start between the USB controller and memory.

The DMA I/F ports of S2S65A00 are mapped to the following addresses on the AHB bus (1).

DMA0 0xE000_0000（Fixed）
DMA1 0xE000_0004（Fixed）

Note: Always use the fixed address mode for DMA transfer to the DMA port of USB controller.

A2.6.5.2 Basic Functions

The DMA interface (I/F) of USB controller supports two operation modes (Count mode and Free-running mode).

This Specification describes the DMA transfer directions as follows.

DMA Write transfer (Write): S2S65A00 internal/external memory ⇒ USB controller
DMA Read transfer (Read): USB controller ⇒ S2S65A00 internal/external memory

A2.6.5.2.1 Count Mode

The data count (in bytes) stored in DMA0/1_Count[31:0] is sent through DMA transfer. When the specified count of data has been sent by DMA transfer, a DMA_IntStat.DMA0/1_Cmp interrupt cause occurs and the DMA circuit stops from operating.

After the data count has been set in the DMA0/1_Count register, if the DMA0/1_Control.Go bit is set to logical 1, the status of USB controller FIFO memory is checked and a data transfer request is issued to the DMA controller. The DMA0/1_Count register value is counted down (or decremented) each time data is transferred, and when the count reaches zero (0), the DMA transfer is stopped and a DMA_IntStat.DMA0/1_Cmp interrupt cause occurs.

The DMA transfer is stopped (or interrupted) when the DMA0/1_Control.Stop bit is set to 1. However, the DMA0/1_Control.Running bit is kept to 1 until DMA transfer is completely stopped even if the DMA0/1_Control.Stop bit has been set to 1. When DMA transfer is stopped (or interrupted) completely, this Running bit is set to logical 0 and a DMA_IntStat.DMA0/1_Cmp interrupt cause occurs. You should confirm this bit status before starting any subsequent process.

A2.6.5.2.2 Free-Running Mode

If you start DMA transfer in Free-Running mode, only the status of USB controller FIFO memory is checked and DMA transfer is continued permanently. If an external DMA controller is used to control the data transfer count, you can skip the DMA transfer settings in the free-running mode.

If DMA0/1_Config.FreeRun bit is set to 1 and if DMA0/1_Control.Go bit is set to 1, the status of USB controller FIFO memory is checked and a data transfer request is issued to the DMA controller. The DMA0/1_Count register value is counted up (or incremented) each time data is transferred, and when the count changes from "0xFFFF_FFFF" to "0x0000_0000", a DMA_IntStat.DMA0/1_CountUp interrupt cause occurs. However, the DMA transfer continues even if a DMA0/1_CountUp interrupt cause has occurred.

The DMA transfer is stopped (or interrupted) when the DMA0/1_Control.Stop bit is set to 1. However, the DMA0/1_Control.Running bit is kept to 1 until DMA transfer is completely stopped even if the DMA0/1_Control.Stop bit has been set to 1. When DMA transfer is stopped (or interrupted) completely, this Running bit is set to logical 0 and a DMA_IntStat.DMA0/1_Cmp interrupt cause occurs. You should confirm this bit status before starting any subsequent process.

A2.6.5.3 Operation Procedure

The following gives an example of DMA transfer procedure if DMA controller 1 is used.

Conditions: 64-byte data (DMA write) transfer from SDRAM (0x4000_0000) to USB memory
DMAC1 CH.0 is used (Count mode).
USB DMA port CH.0 is used.
EPa is used (FIFO area, USB protocol and others have been set.)
End of DMA transfer is determined through polling of DMA_Cmp bit.

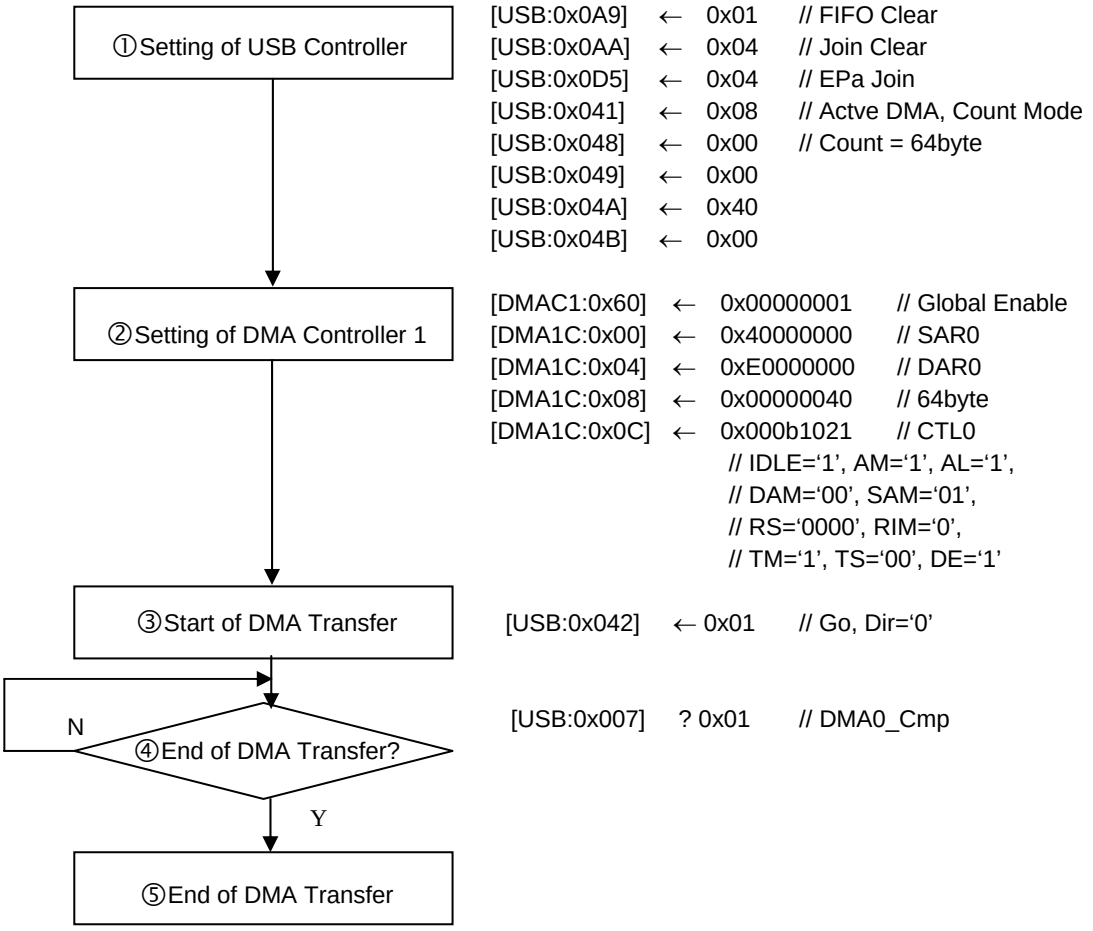


Fig. A2.6.26 Example of DMA Transfer Procedure

A2.6.5.4 Forced Termination

You can forcibly terminate DMA transfer by setting the DMA0/1_Config.ActiveDMA bit to 0. In this case, however, the continuity of current DMA transfer data is not guaranteed. Therefore, usually use the DMA0/1_Control.Stop bit to stop (or interrupt) the DMA transfer. If you repeat DMA transfer after its forced termination, initialize the USB FIFO memory and set the Join and DMA parameters again.

If the USB FIFO memory is initialized or if the Join settings are changed during DMA transfer, the DMA transfer is forcibly terminated and the DMA0/1_Config.ActiveDMA bit is automatically set to 0.

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
HEADQUARTERS
214 Devcon Drive,
San Jose , CA 95112,USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

SALES OFFICES
Northeast
301 Edgewater Place, Suite 210
Wakefield, MA 01880, U.S.A.
Phone: +1-800-922-7667 FAX: +1-781-246-5443

EUROPE

EPSON EUROPE ELECTRONICS GmbH
HEADQUARTERS
Riesstrasse 15
80992 Munich, GERMANY
Phone: +49-89-14005-0 FAX: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.
23F, Beijing Silver Tower 2# North RD DongSanHuan
ChaoYang District, Beijing, CHINA
Phone: +86-10-6410-6655 FAX: +86-10-6410-7320

SHANGHAI BRANCH
7F, High-Tech Bldg., 900, Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5522 FAX: +86-21-5423-5512

EPSON HONG KONG LTD.
Unit 715-723, 7/F Trade Square, 681 Cheung Sha Wan Road,
Kowloon, Hong Kong.
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON Electronic Technology Development (Shenzhen) LTD.
12/F, Dawning Mansion, Keji South 12th Road,
Hi- Tech Park, Shenzhen
Phone: +86-755-2699-3828 FAX: +86-755-2699-3838

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
14F, No. 7, Song Ren Road,
Taipei 110
Phone: +886-2-8786-6688 FAX: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.
1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 FAX: +65-6271-3182

SEIKO EPSON CORPORATION
KOREA OFFICE
5F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

GUMI OFFICE
2F, Grand B/D, 457-4 Songjeong-dong,
Gumi-City, KOREA
Phone: +82-54-454-6027 FAX: +82-54-454-6093

SEIKO EPSON CORPORATION
SEMICONDUCTOR OPERATIONS DIVISION

IC Sales Dept.
IC International Sales Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 FAX: +81-42-587-5117

Document Code: 411076203
First Issue June 2007
Printed in JAPAN (H)

Printed Oct, 2011 in Japan

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Display Drivers & Controllers ***category:***

Click to view products by Epson ***manufacturer:***

Other Similar products are found below :

ICB2FL01G HV5812PJ-G-M904 TW8813-LB2-GR TW8811-PC2-GR MAX1839EEP+ TW9907-TA1-GR LX27901IDW SSD2828QN4 ICB2FL01GXUMA2 DLP2000FQC PAD1000YFFR S1D13746F01A600 FIN324CMLX AD8387JSVZ DLPC6421ZPC HV852K7-G HV859K7-G HV857K7-G DIO2133CT14 S1D13506F00A200 S1D13L03F00A100-40 TW2836-BA1-GR SSD2829QL9 MAX749CSA+T MAX4820EUP+T ICL7135CAI+ ICL7135CMH+D ICL7137CMH+D MAX25221BATJ/V+ S1D13748B00B100 S1D13A05B00B200 MAX3738ETG+T MAX8722CEEG+ MAX749CPA+ MAX8785AETI+ ICL7135CQI+ HV518PJ-G-M903 HV5812P-G HV5812PJ-G HV5812WG-G HV7224PG-G HV853K7-G HV860K7-G HV6810WG-G HV823LG-G HV857MG-G HV833MG-G HV857LMG-G HV850MG-G HV859MG-G