Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM33LG0xx

Low-power MCU Chip

User Manual

2022. 01

This information is provided as a reference material for users to choose the appropriate products of Shanghai Fudan Microelectronics Group Co., Ltd. (hereinafter referred to as Fudan Microelectronics) according to their use. It's not allowed to transfer intellectual property and other rights owned by Fudan Microelectronics or a third party. Before using the information recorded in this document to finally make a judgment on whether the information and products are applicable, please be sure to evaluate all the information as a whole system. The purchaser is solely responsible for the selection and use of the products and services of Fudan Microelectronics described in this article. Fudan Microelectronics is not responsible for the purchaser's selection and use of the products and services described in this article. Fudan Micros also does not guarantee that the manual is error-free. Unless expressly approved in writing, Fudan Microelectronics products are not recommended, authorized, or guaranteed for use in military, aviation, aerospace, life-saving and life-support systems. In the product or system that may cause personal injury or death, serious property or environmental damage due to failure or malfunction. Without the permission of Fudan Microelectronics, it is not allowed to reprint or copy all or part of the content of this material. In the future, daily product updates will be released in due course without notice. When purchasing the products described in this document, please contact the local sales office of Fudan Microelectronics in advance. To confirm the latest information, and please follow the information published by Fudan Microelectronics in various ways, including Fudan Microelectronics website (http://www.fmsh.com/).

If you need to know the details of the information or products recorded in this document, please contact the local sales office of Shanghai Fudan Microelectronics Group Co., Ltd.

Contents

List of tables

List of figures

1 Features

1.1 Introduction

The main features of FM33LG0 are as follows:

- Wide voltage range: 1.65~5.5V
- -40℃~+85℃ temperature range
- Core
  - ARM Cortex-M0
  - Support MPU
  - Unprivileged/Priviledged support
  - 64Mhz Maximum Frequency
  - SWD Debug Interface
  - 24bit SystickTimer
- Lower-power platform
  - 130uA/MHz@48MHz
  - VBAT Backup Power Switch
  - 5uA Sleepmode
  - 1.5uA DeepSleep mode (RTC on + all RAM retention + CPU retention)
  - 0.8uA VBAT mode (RTC on+ Backup Register)
- Memories
  - 64/128/256KB Flash memory
  - Flash cycling endurance: 100,000
  - Flash data retention: 10years@85℃
  - User code protection
  - 16/32KB Maximum SRAM
- Analog peripherals
  - BOR with 4 programmable thresholds
  - Ultra-low-power PDR with 4 programmable thresholds

- Programmable Supply Voltage Detector
- 3x low-power analog comparator
- 12bit 2Msps SAR-ADC
- 12bit 1Msps DAC
- Internal reference voltage generating circuit
- High precision temperature sensor, +/-2°C over full temperature range

- Communication interfaces
  - UART*5
  - LPUART*3
  - 7816 master*1
  - SPI*3, master and slave mode
  - I2C*2, master and slave mode
  - CAN2.0B*1
  - 7-channel DMA
  - Programable CRC
- Timers
  - 16bits basic timer*1, 120MHz Maximum PWM resolution
  - 16bits general timer*3
  - 32bits basic timer*1, 16bits basic timer*1
  - 24bits Systick*1
  - 32bits low-power timer*1, 16bits low-power timer*1
  - Watchdog timer *1
  - Low-power real-time clock calendar (RTCC), with digital calibration up to +/-0.476ppm Segment LCD Controller
- Segment LCD controller
  - Up to 4COM×44SEG / 6COM×42SEG / 8COM×40SEG
  - 1/3 bias, 1/4bias
  - Internal resistor voltage divider
  - Display under DeepSleep
- Security

- AES 128/192/256bits hardware accelerator
- Support ECB/CBC/CTR/GCM/GMAC
- True Random Number generator

- Clocks
  - Programmable high speed RC oscillator, 8/16/24/32MHz, factory-trimmed to +/-0.5%, variation less than +/-2% for 8MHz over -40~+85℃
  - Low power 32K crystal oscillator with fail detector
  - Low-power and low-speed RC oscillator, 32KHz, full temperature range +/-3%
  - 4~24MHz high-frequency crystal oscillator
  - PLL_L up to 128MHz
  - PLL_H up to 64Mhz
  - Package: LQFP80/64/48, QFN32, TSSOP20

1.2 Block Diagram

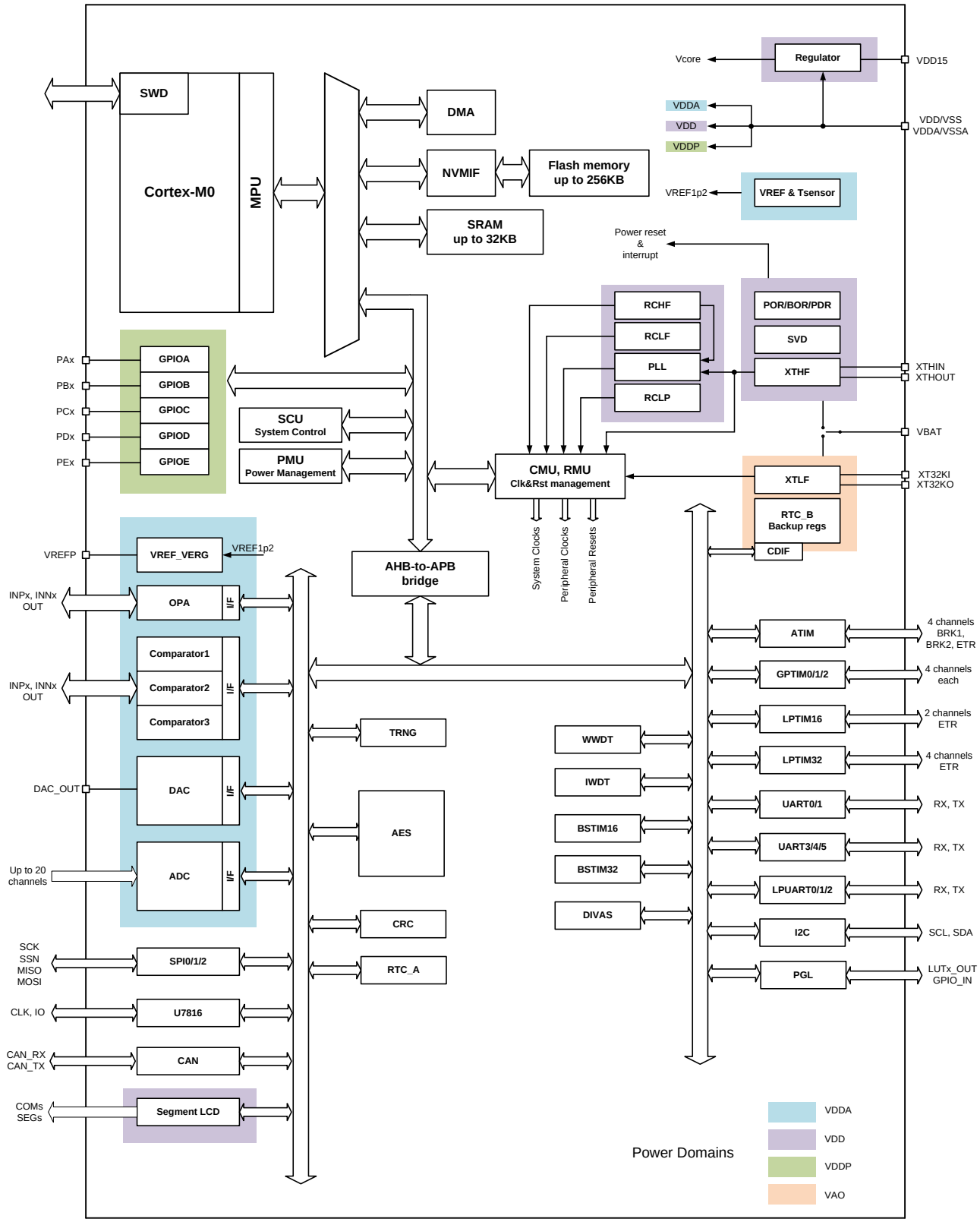


Figure 1–1 FM33LG0xx Block Diagram

1.3 Device Lineup

Part code	Flash (KBytes)	RAM (KBytes)	Package
FM33LG048	256	32	LQFP80
FM33LG046	256	32	LQFP64
FM33LG045	256	32	LQFP48
FM33LG043	256	32	QFN32
FM33LG026	128	32	LQFP64
FM33LG025	128	32	LQFP48
FM33LG023	128	32	QFN32
FM33LG016	64	16	LQFP64
FM33LG013	64	16	QFN32
FM33LG012	64	16	TSSOP20

Table 1-1 FM33LG0xx Device Lineup

1.4 FM33LG0xx Selection Guide

Model		FM33LG048	FM33LG046	FM33LG045	FM33LG043	FM33LG028	FM33LG026	FM33LG025	FM33LG023	FM33LG016	FM33LG015	FM33LG013
CPU		Cortex-M0										
MPU		Y										
Max Freq.		64MHz										
Flash		256KB				128KB				64KB		
RAM		32KB				32KB				16KB		
AES		1										
RNG		1										
Timer	ATIM	1										
	GTIM	3										
	BSTIM32	1										
	BSTIM16	1										
	LPTIM32	1										
	LPTIM16	1										
	Systick	1										
RTC/WWDT/IWDT		2/1/1										
SPI		3	3	2	2	3	3	2	2	3	2	2
I2C		1	1	1	1	1	1	1	1	1	1	1
UART		5	5	5	4	5	5	5	4	5	5	4
LPUART		3	3	3	2	3	3	3	2	3	3	2

Model	FM33LG048	FM33LG046	FM33LG045	FM33LG043	FM33LG028	FM33LG026	FM33LG025	FM33LG023	FM33LG016	FM33LG015	FM33LG013
ISO7816	1	1	-	-	1	1	-	-	1	-	-
GPIO	71	56	40	26	71	56	40	26	56	40	26
LCD	4*44 6*42 8*40	4*36 6*34 8*32	4*25 6*23 8*21	-	4*44 6*42 8*40	4*36 6*34 8*32	4*25 6*23 8*21	-	4*36 6*34 8*32	4*25 6*23 8*21	-
OPA	1	1	1	1	1	1	1	1	1	1	1
12bit SAR-ADC	20ch	18ch	9ch	8ch	20ch	18ch	9ch	8ch	18ch	9ch	8ch
TempSensor	1										

Table 1-2 FM33LG0xx Selection Guide

2 Pinout

2.1 Package and Pin

2.1.1 FM33LG0x8 (LQFP80)

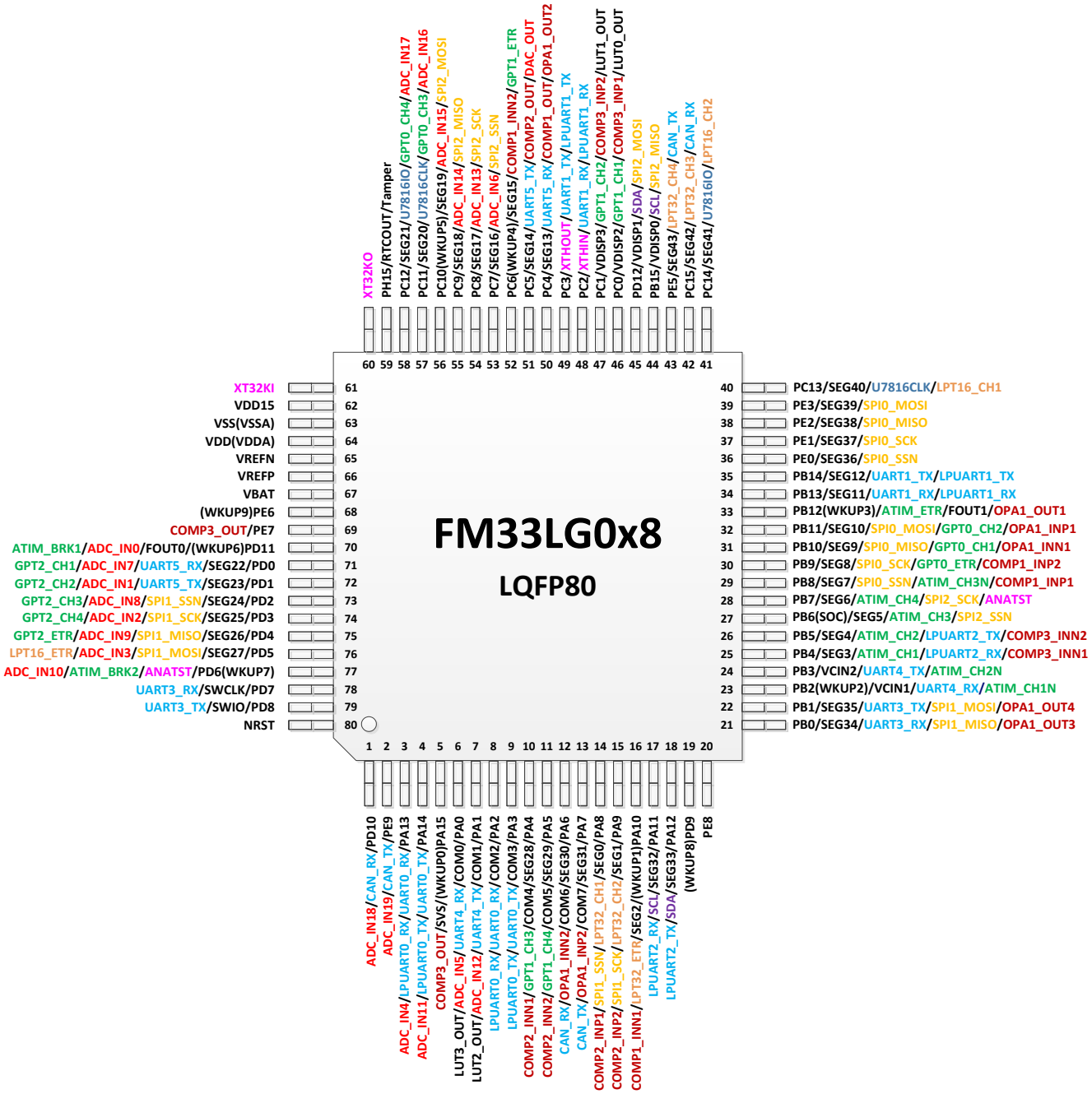


Figure 2–1 FM33LG0x8 LQFP80 Package

Typical resources:

LCD 4*44

ADC (21 external channels)

Support VBAT, with Tamper pin (PH15)

2.1.2 FM33LG0x6 (LQFP64)

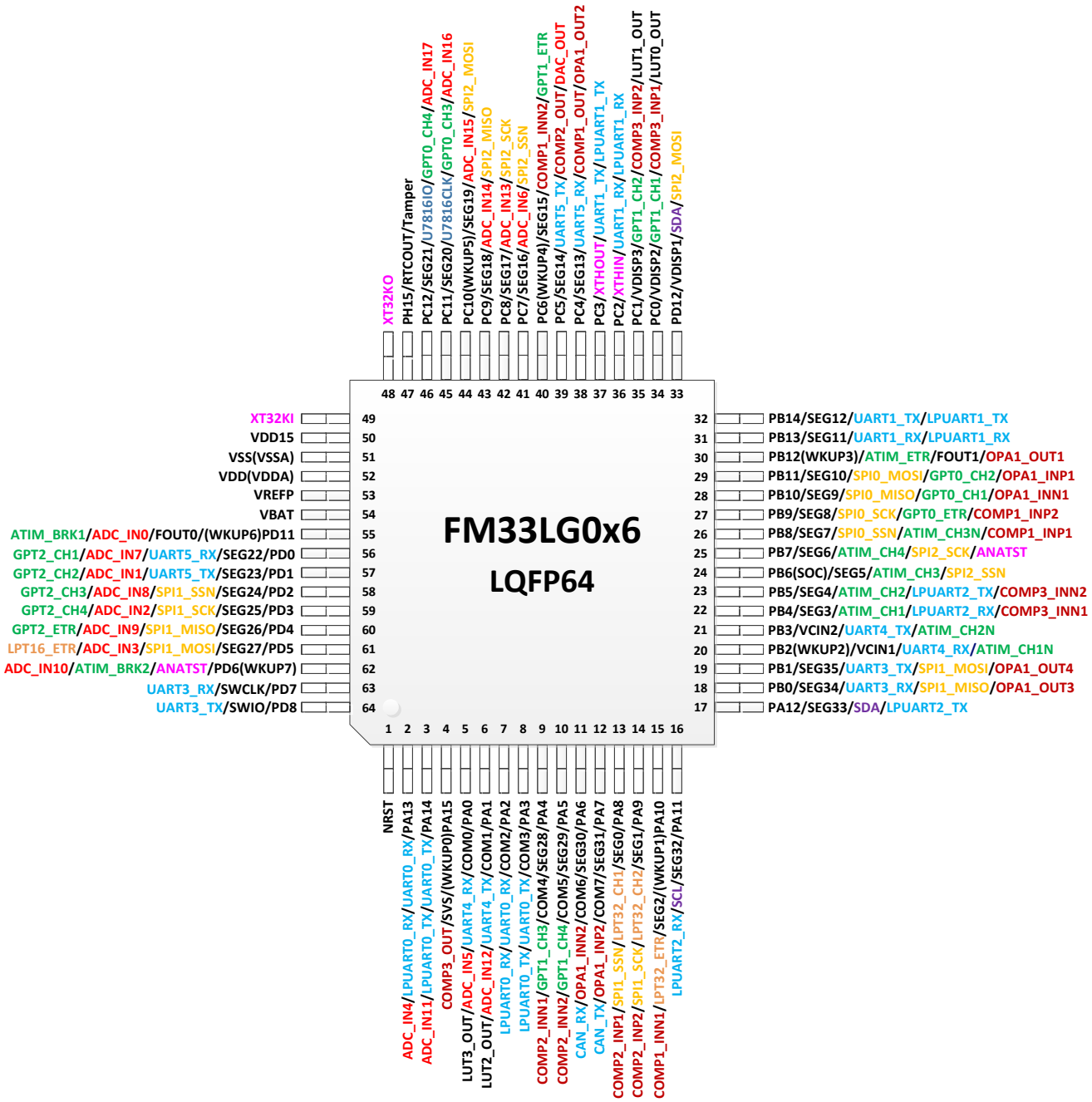


Figure 2–2 FM33LG0x6 LQFP64 Package

Typical resources:

LCD 4*36

ADC (18 external channels)

Support VBAT, with Tamper pin (PH15)

2.1.3 FM33LG0x5 (LQFP48)

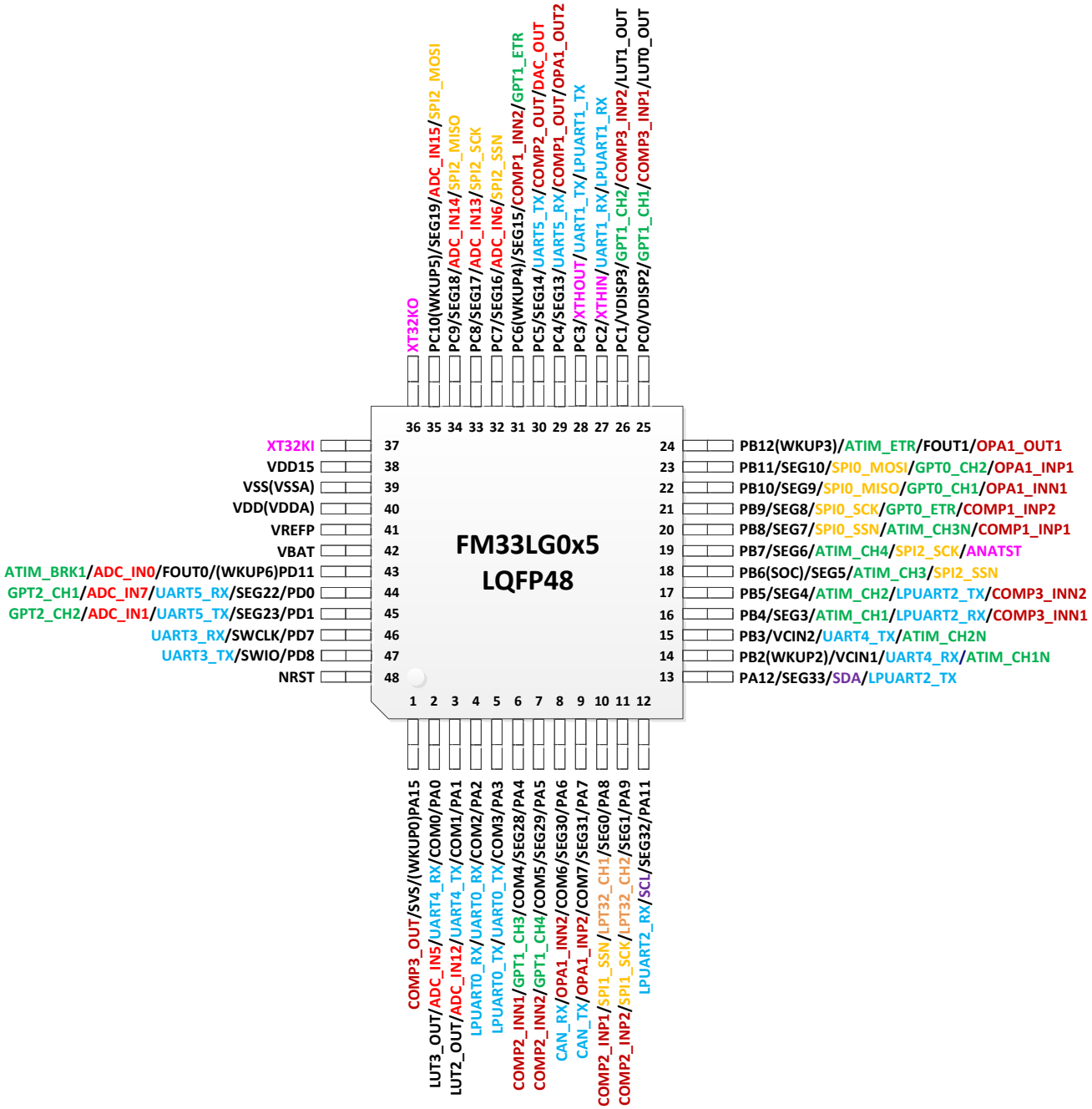


Figure 2–3 FM33LG0x5 LQFP48 Package

Typical resources:

LCD 4*25,6*23,8*21

ADC (9 external channels)

Support VBAT

2.1.4 FM33LG0x3 (QFN32)

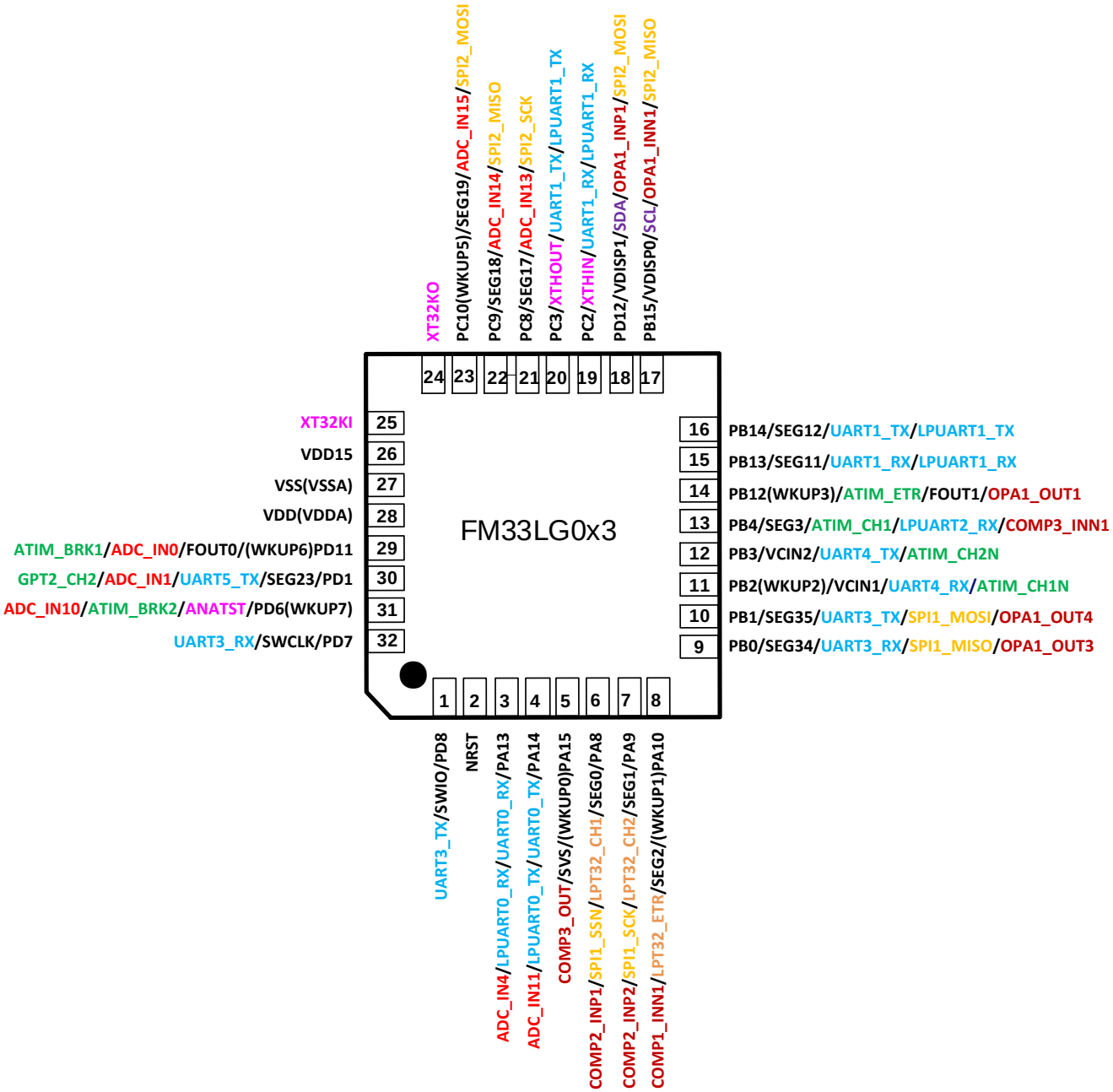


Figure 2–4 FM33LG0x3 QFN32 Package

2.1.5 Pin Descriptions (FM33LG0xx)

Pin Number			Pin Function	Description
LQFP80	LQFP64	LQFP48		
1	-	-	PD10	GPIO
			CAN_RX	CAN Receive
			ADC_IN18	ADC input channel
2	-	-	PE9	GPIO
			CAN_TX	CAN Transmit
			ADC_IN19	ADC input channel
3	2	-	PA13	GPIO
			UART0_RX	UART Receive
			LPUART0_RX	Low-power UART Receive
			ADC_IN4	ADC input channel
4	3	-	PA14	GPIO
			UART0_TX	UART Transmit
			LPUART0_TX	Low-power UART Transmit
			ADC_IN11	ADC input channel
5	4	1	PA15	GPIO
			WKUP0	External wakeup
			SVS	External power detection
			COMP3_OUT	Comparator OUTPUT
6	5	2	PA0	GPIO
			COM0	LCD COM
			UART4_RX	UART Receive
			ADC_IN5	ADC input channel
			LUT3_OUT	PGL Output
7	6	3	PA1	GPIO
			COM1	LCD COM
			UART4_TX	UART Transmit
			ADC_IN12	ADC input channel
			LUT2_OUT	PGL Output
8	7	4	PA2	GPIO
			COM2	LCD COM
			UART0_RX	UART Receive
			LPUART0_RX	Low-power UART receive
9	8	5	PA3	GPIO
			COM3	LCD COM
			UART0_TX	UART Transmit
			LPUART0_TX	Low-power UART transmit
10	9	6	PA4	GPIO
			COM4/SEG28	LCD COM/SEG
			GPT1_CH3	General timer external channel
			COMP2_INN1	Comparator input

Pin Number			Pin Function	Description
LQFP80	LQFP64	LQFP48		
11	10	7	PA5	GPIO
			COM5/SEG29	LCD COM/SEG
			GPT1_CH4	General timer external channel
			COMP2_INN2	Comparator input
12	11	8	PA6	GPIO
			COM6/SEG30	LCD COM/SEG
			OPA1_INN2	OPA input channel
			CAN_RX	CAN receive
13	12	9	PA7	GPIO
			COM7/SEG31	LCD COM/SEG
			OPA1_INP2	OPA input channel
14	13	10	PA8	GPIO
			SEG0	LCD SEG
			LPT32_CH1	Low-power timer external channel
			COMP2_INP1	Comparator input
15	14	11	PA9	GPIO
			SEG1	LCD SEG
			LPT32_CH2	Low-power timer external channel
			COMP2_INP2	Comparator input
16	15	-	PA10	GPIO
			WKUP1	External WKUP
			SEG2	LCD SEG
			LPT32_ETR	Low-power timer external channel
			COMP1_INN1	Comparator input
17	16	12	PA11	GPIO
			SEG32	LCD SEG
			SCL	I2C CLOCK
			LPUART2_RX	LPUART Receive
18	17	13	PA12	GPIO
			SEG33	LCD SEG
			SDA	I2C Data
			LPUART2_TX	LPUART transmit
19	-	-	PD9	GPIO
			WKUP8	External WKUP
20	-	-	PE8	GPIO
21	18	-	PB0	GPIO
			SEG34	LCD SEG
			UART3_RX	UART Receive
			SPI1_MISO	SPI Data
			OPA1_OUT3	OPA Output

Pin Number			Pin Function	Description
LQFP80	LQFP64	LQFP48		
22	19	-	PB1	External WKUP
			SEG35	LCD SEG
			UART3_TX	UART transmit
			SPI1_MOSI	SPI Data
			OPA1_OUT4	OPA Output
23	20	14	PB2	GPIO
			WKUP2	External WKUP
			VCIN1	LCD Capacitor Pin (Capacitor Drive Mode)
			UART4_RX	UART receive
			ATIM_CH1N	Advanced timer external channel
24	21	15	PB3	GPIO
			VCIN2	LCD Capacitor Pin (Capacitor Drive Mode)
			UART4_TX	UART receive
			ATIM_CH2N	Advanced timer external channel
25	22	16	PB4	GPIO
			SEG3	LCD SEG
			ATIM_CH1	Advanced timer external channel
			LPUART2_RX	LPUART receive
			COMP3_INN1	COMP input
26	23	17	PB5	GPIO
			SEG4	LCD SEG
			ATIM_CH2	Advanced timer external channel
			LPUART2_TX	LPUART receive
			COMP3_INN2	COMP input
27	24	18	PB6	GPIO
			SEG5	LCD SEG
			ATIM_CH3	Advanced timer external channel
			SPI2_SSN	SPI SSN
28	25	19	PB7	GPIO
			SEG6	LCD SEG
			ATIM_CH4	Advanced timer external channel
			SPI2_SCK	SPI SCK
			ANATST	Original factory test channel
29	26	20	PB8	GPIO
			SEG7	LCD SEG
			SPI0_SSN	SPI SSN
			ATIM_CH3N	Advanced timer external channel
			COMP1_INP1	COMP input
30	27	21	PB9	GPIO
			SEG8	LCD SEG
			COMP1_INP2	COMP input

Pin Number			Pin Function	Description
LQFP80	LQFP64	LQFP48		
			SPI0_SCK	SPI SCK
			GPT0_ETR	General timer external channel
31	28	22	PB10	GPIO
			SEG9	LCD SEG
			OPA1_INN1	OPA input
			SPI0_MISO	SPI data
			GPT0_CH1	General timer external channel
32	29	23	PB11	GPIO
			SEG10	LCD SEG
			OPA1_INP1	OPA input
			SPI1_MOSI	SPI data
			GPT0_CH2	General timer external channel
33	30	24	PB12	GPIO
			WKUP3	External WKUP
			FOUT1	Clock frequency output
			ATIM_ETR	Advanced timer external channel
			OPA1_OUT1	OPA output
34	31	-	PB13	GPIO
			SEG11	LCD SEG
			UART1_RX	UART receive
			LPUART1_RX	LPUART receive
35	32	-	PB14	GPIO
			SEG12	LCD SEG
			UART1_TX	UART transmit
			LPUART1_TX	LPUART transmit
36	-	-	PE0	GPIO
			SEG36	LCD SEG
			SPI0_SSN	SPISSN
37	-	-	PE1	GPIO
			SEG37	LCD SEG
			SPI0_SCK	SPISCK
38	-	-	PE2	GPIO
			SEG38	LCD SEG
			SPI0_MISO	SPI data
39	-	-	PE3	GPIO
			SEG39	LCD SEG
			SPI0_MOSI	SPI data
40	-	-	PC13	GPIO
			SEG40	LCD SEG
			U7816CLK	7816 Interface clock
			LPT16_CH1	LPTIM16 channel

Pin Number			Pin Function	Description
LQFP80	LQFP64	LQFP48		
41	-	-	PC14	GPIO
			SEG41	LCD SEG
			U7816IO	7816 interface data (100K pull-up)
			LPT16_CH2	LPTIM16 channel
42	-	-	PC15	GPIO
			SEG42	LCD SEG
			LPT32_CH3	LPTIM32 channel
			CAN_RX	CAN receive
43	-	-	PE5	GPIO
			SEG43	LCD SEG
			LPT32_CH4	LPTIM32 channel
			CAN_TX	CAN transmit
44	-	-	PB15	GPIO
			VDISP0	LCD off-chip capacitance
			SCL	I2C clock
			SPI2_MISO	SPI data
45	33	-	PD12	GPIO
			VDISP1	LCD off-chip capacitance
			SDA	I2C data
			SPI2_MOSI	SPI data
46	34	25	PC0	GPIO
			VDISP2	LCD off-chip capacitance
			COMP3_INP1	Analog comparator input
			LUT0_OUT	PGL output
			GPT1_CH1	General timer external channel
47	35	26	PC1	GPIO
			VDISP3	LCD off-chip capacitance
			COMP3_INP2	Analog comparator input
			LUT1_OUT	PGL output
			GPT1_CH2	General timer external channel
48	36	27	PC2	GPIO
			XTHIN	High frequency crystal input
			UART1_RX	UART receive
			LPUART1_RX	LPUART receive
49	37	28	PC3	GPIO
			XTHOUT	High frequency crystal output
			UART1_TX	UART transmit
			LPUART1_TX	LPUART transmit
50	38	29	PC4	GPIO
			SEG13	LCD SEG
			OPA1_OUT2	OPA output

Pin Number			Pin Function	Description
LQFP80	LQFP64	LQFP48		
			COMP1_OUT	COMP output
			UART5_RX	UART recive
51	39	30	PC5	GPIO
			SEG14	LCD SEG
			DAC_OUT	DAC output
			COMP2_OUT	COMP2 out
			UART5_TX	UART transmit
52	40	31	PC6	GPIO
			WKUP4	External WKUP4
			SEG15	LCD SEG
			GPT1_ETR	General timer external trigger input
			COMP1_INN2	COMP input
53	41	32	PC7	GPIO
			SEG16	LCD SEG
			SPI2_SSN	SPI SSN
			ADC_IN6	ADC Input channel
54	42	33	PC8	GPIO
			SEG17	LCD SEG
			SPI2_SCK	SPI clock
			ADC_IN13	ADC input channel
55	43	34	PC9	GPIO
			SEG18	LCD SEG
			SPI2_MISO	SPI data
			ADC_IN14	ADC input channel
56	44	35	PC10	GPIO
			WKUP5	External WKUP
			SEG19	LCD SEG
			SPI2_MOSI	SPI data
			ADC_IN15	ADC input channel
57	45	-	PC11	GPIO
			SEG20	LCD SEG
			U7816CLK	7816 Interface clock
			GPT0_CH3	General timer external channel
			ADC_IN16	ADC input channel
58	46	-	PC12	GPIO
			SEG21	LCD SEG
			U7816IO	7816 interface data (10K pull-up)
			GPT0_CH4	General timer external channel
			ADC_IN17	ADC input channel
59	47	-	PH15	GPIO
			RTCOUT	RTCB output signal

Pin Number			Pin Function	Description
LQFP80	LQFP64	LQFP48		
			Tamper	Tamper detection signal input
60	48	36	XT32KO	32768Hz Crystal output pin
61	49	37	XT32KI	32768Hz Crystal input pin
62	50	38	VDD15	LDO output, external 100nF capacitor to ground
63	51	39	VSS	GND
64	52	40	VDD	Source
65	-	-	VREFN	Base Ground
66	53	41	VREFP	Base Source
67	54	42	VBAT	Backup power
68	-	-	PE6	GPIO
			WKUP9	Wake-up pin
69	-	-	PE7	GPIO
			COMP3_OUT	Comp output
70	55	43	PD11	GPIO
			WKUP6	External wake-up pin
			FOUT0	Clock frequency output
			ATIM_BRK1	Advanced timer brake input
			ADC_IN0	ADC input channel
71	56	44	PD0	GPIO
			SEG22	LCD SEG
			UART5_RX	UART receive
			GPT2_CH1	General timer channel
			ADC_IN7	ADC input channel
72	57	45	PD1	GPIO
			SEG23	LCD SEG
			UART5_TX	UART transmit
			GPT2_CH2	General timer channel
			ADC_IN1	ADC input channel
73	58	-	PD2	GPIO
			SEG24	LCD SEG
			SPI1_SSN	SPI SSN
			GPT2_CH3	General timer channel
			ADC_IN8	ADC input channel
74	59	-	PD3	GPIO
			SEG25	LCD SEG
			SPI1_SCK	SPI clock
			GPT2_CH4	General timer channel
			ADC_IN2	ADC input channel
75	60	-	PD4	GPIO
			SEG26	LCD SEG

Pin Number			Pin Function	Description
LQFP80	LQFP64	LQFP48		
			SPI1_MISO	SPI data
			GPT2_ETR	General-purpose timer external trigger
			ADC_IN9	ADC input channel
76	61	-	PD5	GPIO
			SEG27	LCD SEG
			SPI1_MOSI	SPI data
			LPT16_ETR	LPTIM16 external trigger
			ADC_IN3	ADC input channel
77	62	-	PD6	GPIO
			WKUP7	External wake-up pin
			ANATST	Analog test channel
			ATIM_BRK2	Advanced timer brake input
			ADC_IN10	ADC input channel
78	63	46	PD7	GPIO
			UART3_RX	UART receive
			SWCLK	SWD interface clock
79	64	47	PD8	GPIO
			UART3_TX	UART transmit
			SWIO	SWD Interface data
80	1	48	NRST	Reset pin

Table 2-1 FM33LG0xx Pin Descriptions

2.1.6 Package information

2.1.6.1 LQFP80

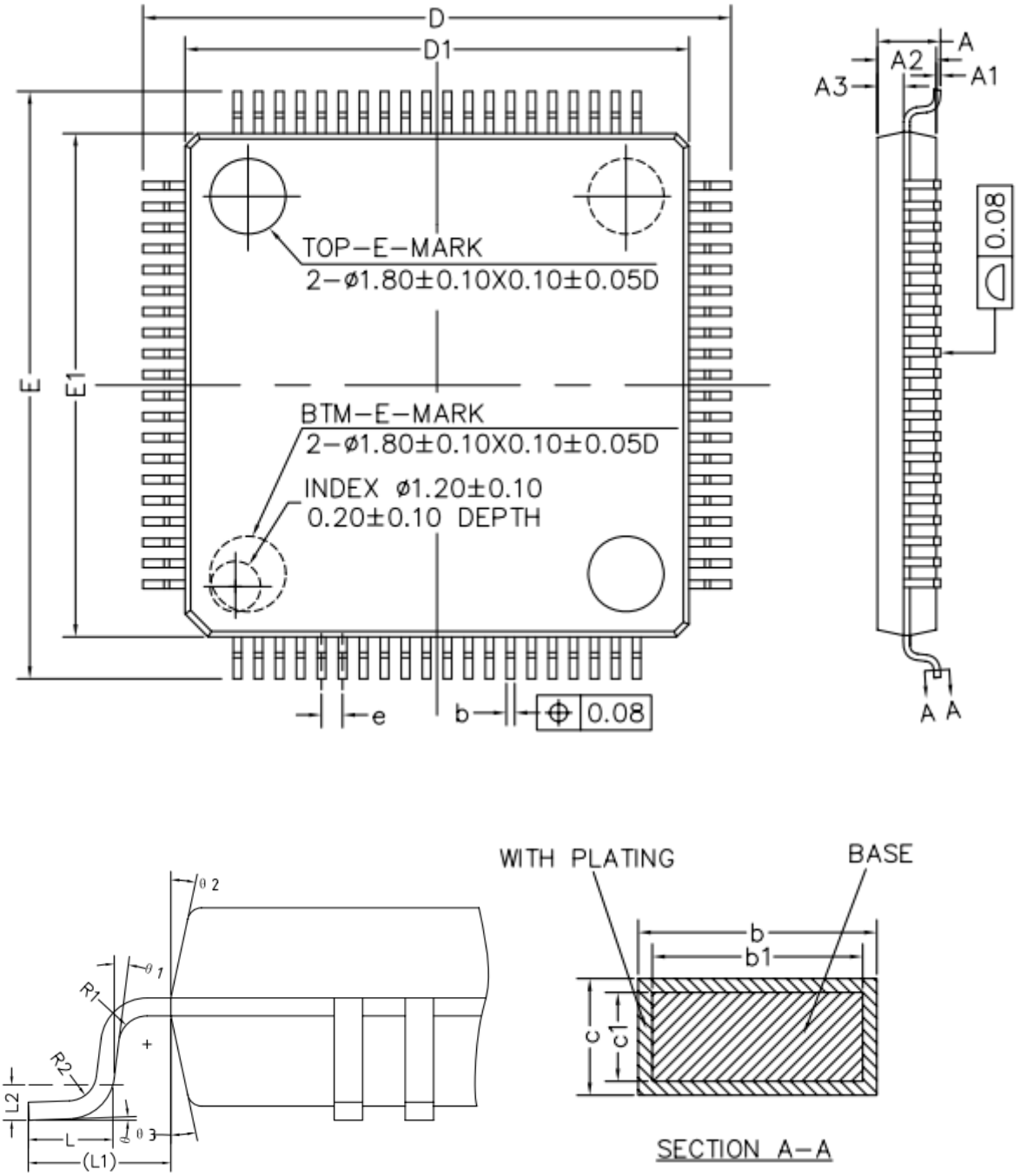


Figure 2–5 LQFP80 Package Information

Symbol	MIN	NOM	MA
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69

Symbol	MIN	NOM	MA
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	13.80	14.00	14.20
D1	11.90	12.00	12.10
E	13.80	14.00	14.20
E1	11.90	12.00	12.10
e	0.40	0.50	0.60
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE: ALL DIMENSIONS REFER TO JEDEC STANDARD MO-220 WMMD-4.

2.1.6.2 LQFP64

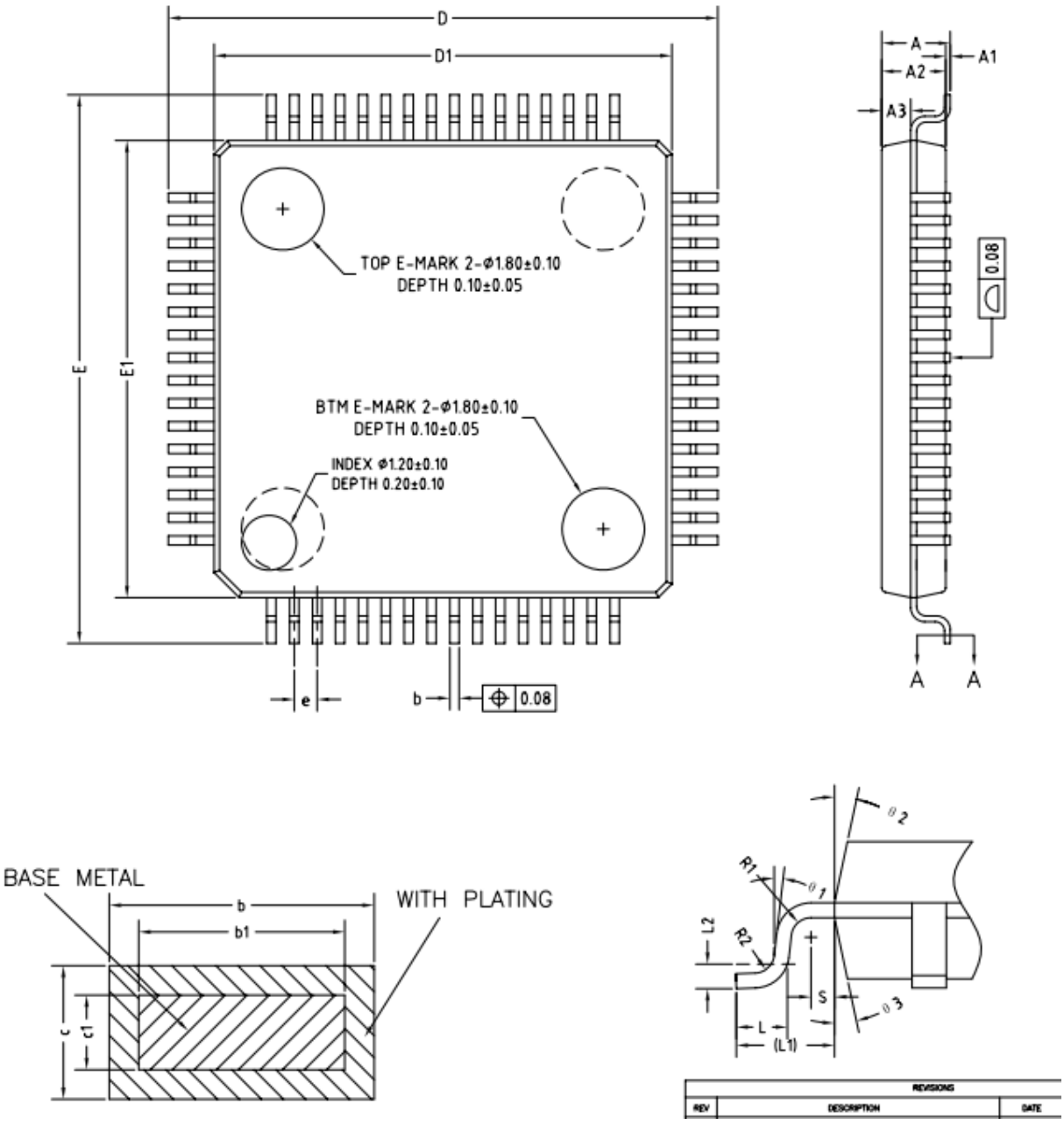


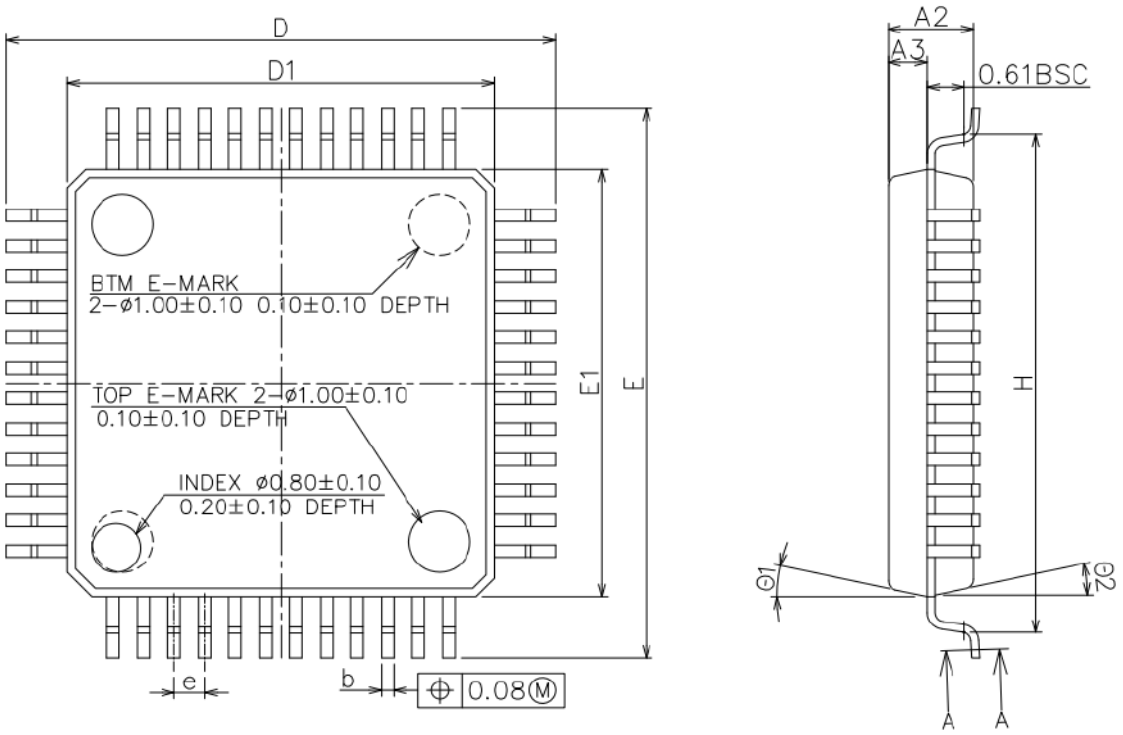
Figure 2–6 LQFP64 Package Information

Symbol	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18

Symbol	MIN	NOM	MAX
c1	0.12	0.127	0.134
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e		0.50BSC	
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE: ALL DIMENSIONS REFER TO JEDEC STANDARD MO-220 WMMD-4.

2.1.6.3 LQFP48



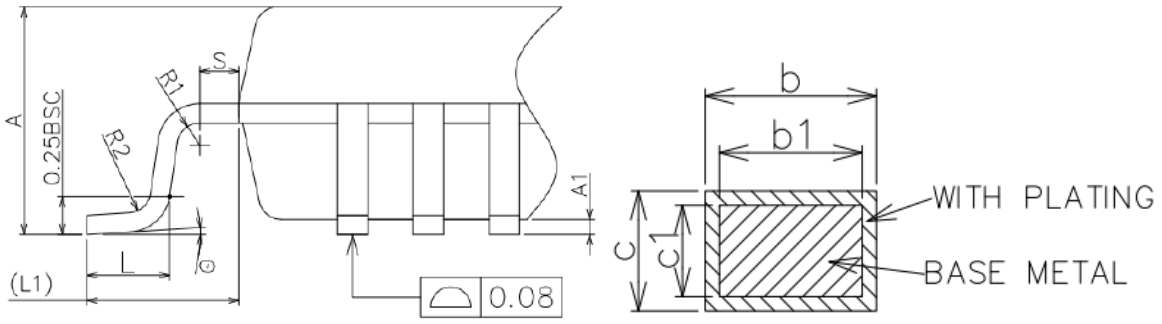


Figure 2–7 LQFP48 Package Information

Symbol	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
e	0.50BSC		
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE: ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BDD.

2.1.6.4 QFN32

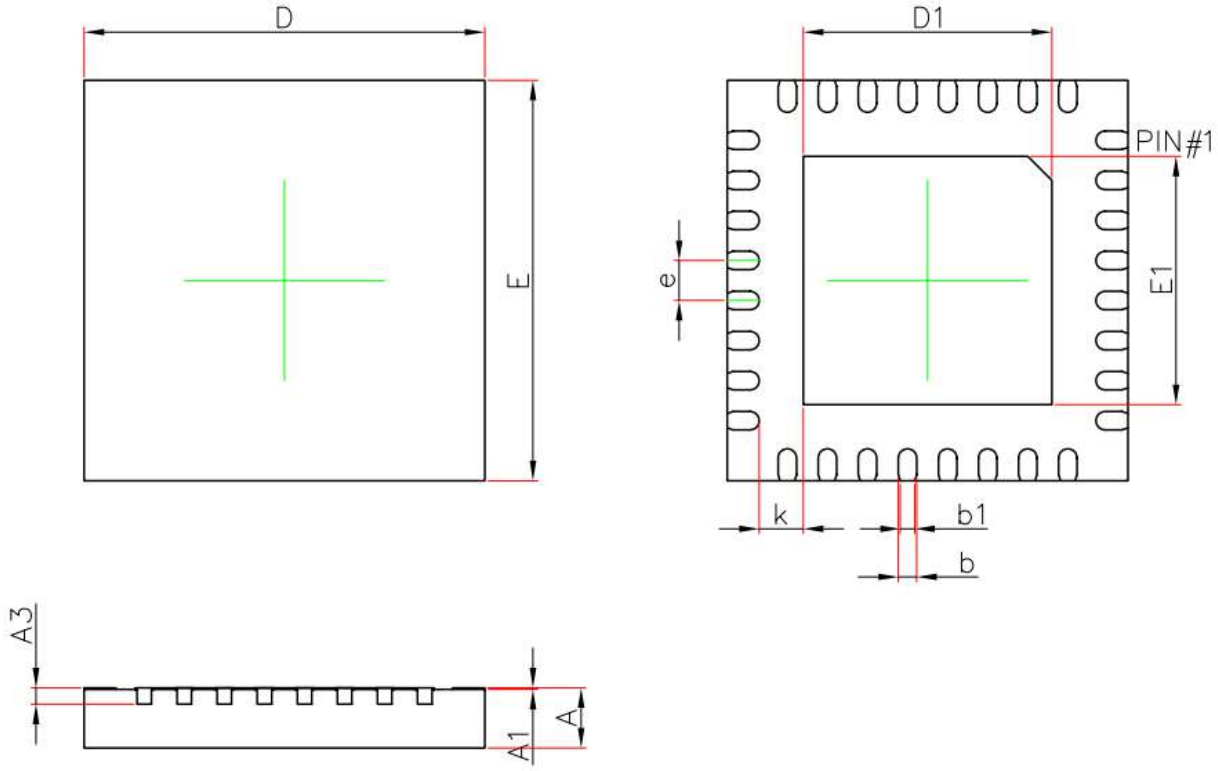


Figure 2–8 QFN32 Package Information

Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	0.700	0.800	0.028	0.031
A1	0.000	0.050	0.000	0.002
A3	0.203 REF.		0.008 REF.	
b	0.180	0.300	0.007	0.012
b1	0.130	0.230	0.005	0.009
D	4.900	5.100	0.193	0.201
D1	3.000	3.200	0.118	0.126
E	4.900	5.100	0.193	0.201
E1	3.000	3.200	0.118	0.126
e	0.500 BSC.		0.020 BSC.	
k	0.550 REF.		0.022 REF.	
L	0.324	0.476	0.013	0.019

NOTE: ALL DIMENSIONS REFER TO JEDEC STANDARD MO-220WMMD-4.

2.2 Welding Installation

Fudan microelectronics chips are packaged in lead-free process. The reflow welding process parameters are recommended to be set in accordance with JEDEC standards.

According to JEDEC standard J-STD-020, the recommended peak temperature setting for lead-free process reflow welding is shown in the following table. The user can select the appropriate peak reflow welding temperature in the table below according to the specifications of the different thickness and volume of the chip.

Package thickness	Package volume mm^3 <350	Package volume mm^3 350 - 2000	Package volume mm^3 >2000
<1.6mm	260℃	260℃	260℃
1.6~2.5 mm	260℃	250℃	245℃
>2.5mm	250℃	245℃	245℃

The following table shows the peak reflow temperatures for various packages:

Package type	Package thickness mm	Package volume mm^3	Reflow welding peak temperature
LQFP80	1.4	201.6	260℃

Please refer to JEDEC standard J-STD-020 for setting the temperature curve of lead-free reflow weld ring.

Profile Feature	Pb-Free Assembly
Preheat/Soak Temperature Min (T_{smin}) Temperature Max (T_{smax}) Time (t_s) from (T_{smin} to T_{smax})	 150 °C 200 °C 60-120 seconds
Ramp-up rate (T_L to T_p)	3 °C/second max.
Liquidous temperature (T_L) Time (t_L) maintained above T_L	217 °C 60-150 seconds
Peak package body temperature (T_p)	For users T_p must not exceed the Classification temp in Table 4-2. For suppliers T_p must equal or exceed the Classification temp in Table 4-2.
Time (t_p)* within 5 °C of the specified classification temperature (T_c), see Figure 5-1.	30* seconds
Ramp-down rate (T_p to T_L)	6 °C/second max.
Time 25 °C to peak temperature	8 minutes max.

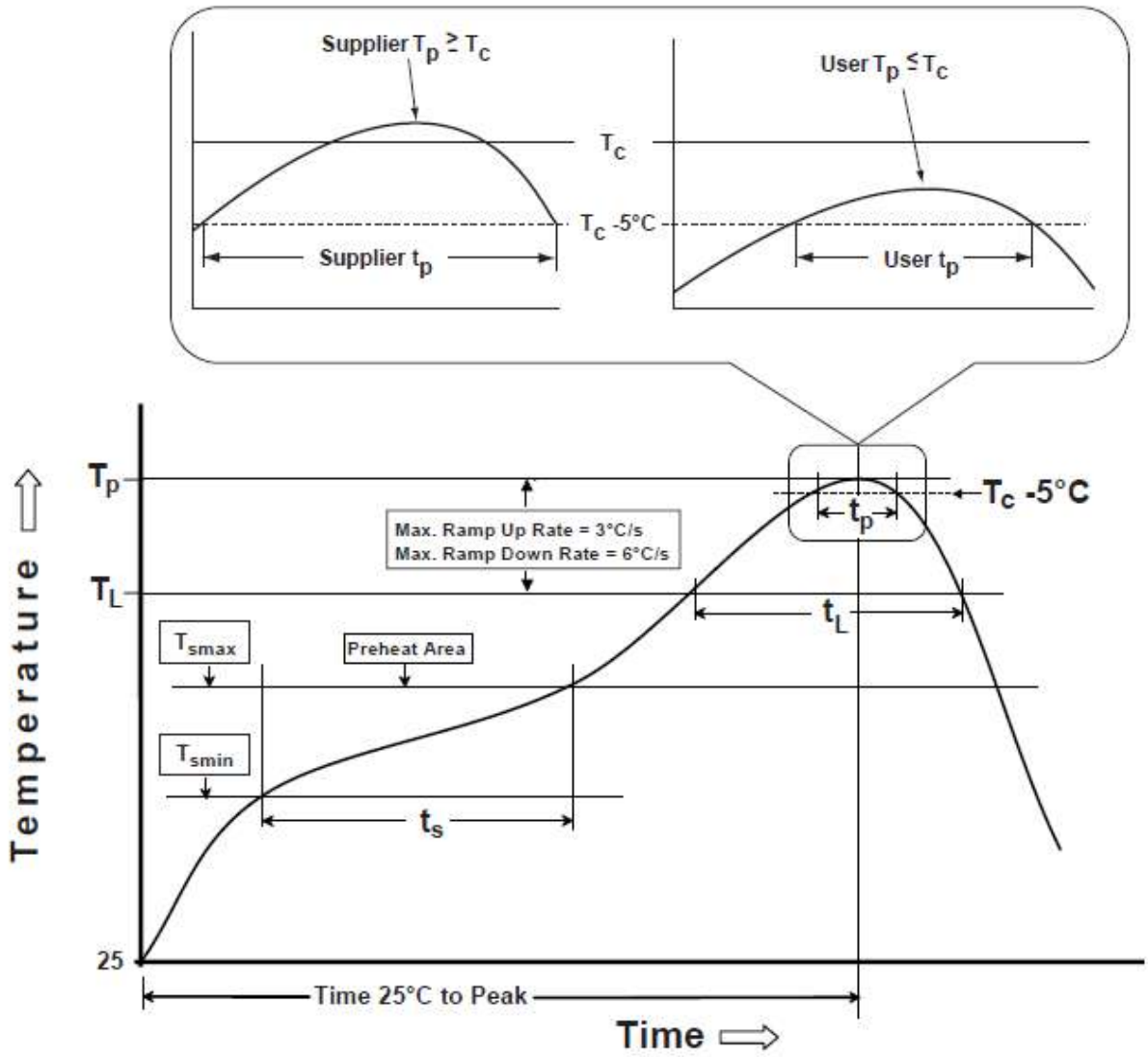


Figure 2–9 JEDEC Standard Heat Resistance Reflow Temperature Curve

Note:

- Before the chip is welded on the upper board, please observe whether the humidity card changes color to confirm whether the humidity sensitive packaging is intact. Unless otherwise specified, the chip package is MSL3 level, please complete the welding operation within one week after the package is opened and restructured in a non-dry environment
- Unless otherwise specified, do not reflow more than 3 times

2.3 MSL Level

FM33LG0 chip moisture sensitivity grade is MSL3, according to JEDEC standard: J-STD-020. Please open the package and place it in a non-dry environment within one week for soldering operations.

2.4 Heat Resistance Characteristics

The chip junction temperature (TJ) can be calculated by the following formula.

$$T_J = T_A + P_D \times \Theta_{JA}$$

Among them:

- T_A is the working environment temperature, the unit is °C
- Θ_{JA} is the package thermal resistor coefficient, the unit is °C/W
- P_D is the chip power, including core power and IO power, the unit is W; the IO power can be calculated by the following formula:

$$P_{IO} = \sum(V_{OL} \times I_{OL}) + \sum((V_{DDIO} - V_{OH}) \times I_{OH})$$

The thermal resistor coefficient of different package types can be referred to the following table:

Package type	Package dimension	Reference thermal resistorΘ_{JA} (℃/W)
LQFP80	12x12x1.4mm	50
LQFP100	14x14x1.4mm	45
LQFP64	10x10x1.4mm	48
LQFP48	7x7x1.4mm	55
QFN32	5x5x0.75mm	35

Example:

- Assumed average working environment temperature T_A=55°C
- Chip core current IDD=5mA, VDD=3.6V
- 10 IO low voltage output, IO sink 5mA，VOL=0.3V
- 10 IO high voltage output, IO source 5mA，VOH=2.9V

Chip power can be calculated as:

$$P_D = P_{INT} + P_{IO} = 5mA \times 3.6V + 10 \times 5mA \times 0.3V + 10 \times (3.6V - 2.9V) \times 5mA = 68mW$$

ForLQFP64, Θ_{JA} = 48°C /W, the junction temperature can be calculated as:

$$T_J = T_A + P_D \times \Theta_{JA} = 55^\circ C + 0.068W \times 48^\circ C / W = 58.264^\circ C$$

3 Electrical Parameters

3.1 Introduction

The typical values listed in the section of electrical parameters are the central values of the distribution of a large number of sample data, and the min/max value at room temperature is guaranteed by the chip mass production test. Electrical parameters at high and low temperatures are based on characterization. The min/max data represent the mean value plus or minus three times the standard deviation (mean +/- 3σ).

3.2 Parameter Test Conditions

3.2.1 Power Supply Scheme

The power supply scheme as shown in the figure below is adopted for chip test.

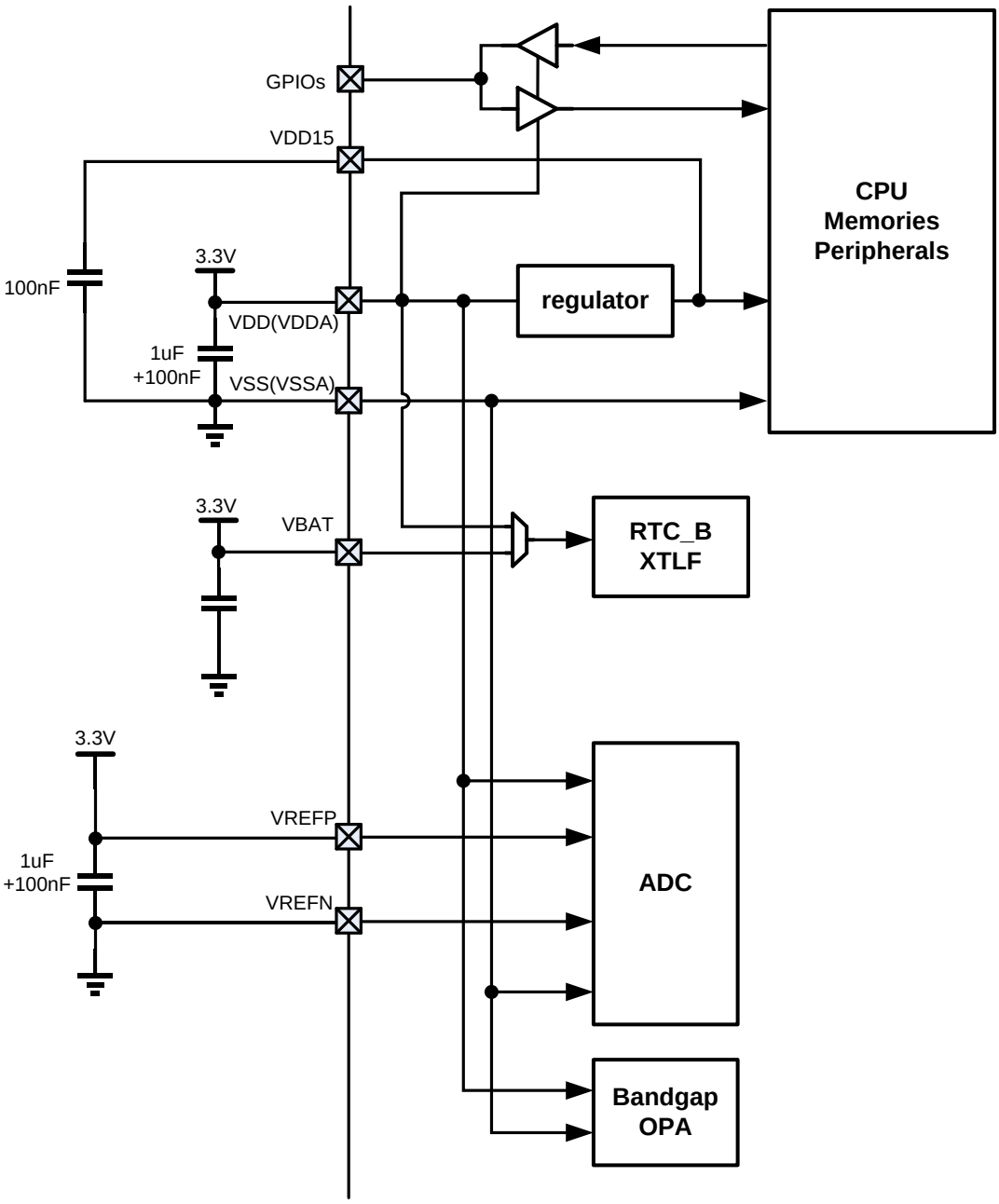


Figure 3–1 FM33LG0xx Power Supply Scheme

3.3 Absolute Maximum Ratings

When the voltage and current applied to the chip exceed the absolute maximum rating defined in the limit parameter table, the chip may be permanently damaged. Exceeding the absolute maximum ratings for a short time may affect the reliability and operating life of the device.

Symbol	Parameter		min	max	unit
V_{DD}-V_{SS}	Supply voltage （VDD、VDDA）		-0.3	6.5	V
V_{PIN}	Pin voltage		V_{SS}-0.3	6.5	V
VREFP-VREFN	ADC reference voltage		-0.3	6.5	
$\|\Delta V_{SS}\|$	Voltage difference between all ground pins		-	50	mV
T_A	Working temperature		-40	85	℃
T_{STG}	Storage temperature		-55	150	℃
HBM	ESD HBM mode TA=25℃ The test standard conforms to JEDEC JS-001	PC5(DAC_OUT)		+/-2000	V
		All others		+/-4000	V
CDM	ESD CDM mode TA=25℃ The test standard conforms to JEDEC JS-002			+/-1000	V
LU	IO Latch up -(0.5VDD) < VI < (1.5VDD) TA=25℃ The test standard conforms to JEDEC JS-002			+/-210	mA
$\sum I_{VDD}$	Total current flows into VDD(Source)			120	mA
$\sum I_{VSS}$	Total current flows out of VSS(Sink)			100	mA
$\sum I_{IO}$	Total current of all IO sinks			100	mA
	Total current of all IO sources			120	mA

Table 3-1 FM33LG0xx Absolute Maximum Ratings

3.4 Operating Conditions

3.4.1 General Operating Conditions

Symbol	Parameter	Conditions	min	max	unit
f_{HCLK}	AHB clock frequency	-	0	64	MHz
f_{PCLK}	APB clock frequency	-	0	64	
VDD	Typical operating voltage range		1.65	5.5	V
VBAT	Backup power supply operating voltage range	Do not use the power switching function, should be shorted to VDD	1.5	5.5	V
		Use the power switch function, VBAT power supply alone	1.5	4.2	V
T_J	Junction temperature		-40	125	°C

Table 3-2 FM33LG0xx General Operating Conditions

3.4.2 Current consumption characteristics

The current consumptions are factory-tested at ambient temperature, and the high and low temperature current parameters are based on characterization.

When power consumption parameters are measured, the MCU is configured as follows:

- All functional pins are configured in GPIO mode, and the input and output enable is turned off to avoid floating leakage
- All peripherals are turned off and the operating clocks are stopped except as otherwise stated
- The maximum current consumption data at room temperature represents the test upper limit standard
- Typical current consumption data at room temperature represent the mean values of a large number of sample distributions
- Unless otherwise specified, all current consumption are tested under VDD=VDDA=3.3V

3.4.2.1 Active Mode

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
IDD_{RUN}	Current consumption in active mode CPU fetches from Flash Core Mark	f_{AHB}=16MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	2.32	-	mA
			TA=85℃	-	2.31	-	
		f_{AHB}=24MHz (RCHF) PLL off Flash 0 wait	TA=25℃	-	3.43	-	mA
			TA=85℃		3.40		
		f_{AHB}=48MHz PLL on Flash 1 wait	TA=25℃	-	6.16	-	mA
			TA=85℃		6.15		
		f_{AHB}=614KHz (RCLF) PLL off Flash 0 wait	TA=25℃		150		uA
			TA=85℃		150		

Table 3-3 ACTIVE Current Parameters

Note:

The parameters in the above table are based on characterization and are not included in the mass production test.

3.4.2.2 LP Active Mode

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
IDD_{RUN}	Current consumption in LP Active，CPU fetches from Flash, Coremark	f_{AHB}=614KHz (RCLF) PLL, RCHF off Flash 0 wait	TA=25℃	-	125	-	uA
			TA=85℃	-	150	-	
IDD_{RUN}	Current consumption in LP Active，CPU fetches from Flash，while(1)	f_{AHB}=614KHz (RCLF) PLL, RCHF off Flash 0 wait	TA=25℃	-	120	-	uA
			TA=85℃	-	140	-	

Table 3-4 LP ACTIVE Current Parameters

3.4.2.3 LP RUN Mode

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
IDD_{LPRUN}	Current consumption in LP RUN，CPU fetches from Flash, Coremark	f_{AHB}=32768Hz (XTLF) PLL, RCHF, RCLF off Flash 0 wait	TA=25℃	-	29	-	uA
			TA=85℃	-	32	-	
IDD_{LPRUN}	Current consumption in LP RUN, CPU fetches from Flash，while(1)	f_{AHB}=32768Hz (XTLF) PLL, RCHF, RCLF off Flash 0 wait	TA=25℃	-	28	-	uA
			TA=85℃	-	30	-	

Table 3-5 LP RUN Current Parameters

3.4.2.4 SLEEP Mode

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
I_{sleep1}	Sleep mode current	BOR、SVD disable RTC running with XTLF CPU, RAM, peripheral retained LCD display enabled, no load	TA=25℃	-	3	5	uA
			TA=85℃	-	7.5	-	

Table 3-6 SLEEP Current Parameters

3.4.2.5 DEEPSLEEP Mode

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
I_{sleep1}	DeepSleep mode current	BOR, SVD closed RTC uses XTLF to travel time CPU, RAM, peripheral data retention LCD display off	TA=25℃	-	1.2	2	uA
			TA=85℃	-	5.5	-	

Table 3-7 DEEPSLEEP Current Parameters

3.4.2.6 VBAT Power Consumption

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
I_{VBAT}	VBAT current	VDD power down VBAT=3.3V XTLF drive current 250nA RTCB travel time	TA=25℃	-	0.8	-	uA
			TA=85℃	-	1	-	

Table 3-8 VBTA Current Parameters

3.4.3 Reset and Supply Detection

The reset and power monitoring parameters of the chip are as follows.

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
t_{VDD}	VDD rising slope		2		∞	us/V
	VDD falling slope	PDR	100		∞	us/V
		BOR	30		∞	us/V
T_{reset_delay}	Power on reset time delay			0.5		ms
T_{pdr_filter}	Power down reset filtering time			4		us
V_{POR}	Power on reset voltage [1]	-40℃≤T_A≤85℃	1.45	1.55	1.65	V
V_{BOR}	Power down reset voltage -40℃≤T_A≤85℃[1]	BORCFG==2'b00	1.65	1.75	1.85	V
		BORCFG==2'b01	1.85	1.95	2.05	
		BORCFG==2'b10	2.05	2.15	2.25	
		BORCFG==2'b11	2.25	2.35	2.45	

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
V_{PDR}	Low-power power-off reset voltage -40℃≤T_A≤85℃[1]	PDRCFG==2'b00		1.23	1.33	1.4	V
		PDRCFG==2'b01		1.28	1.38	1.45	
		PDRCFG==2'b10		1.32	1.42	1.48	
		PDRCFG==2'b11		1.4	1.48	1.55	
I_{BOR}	BOR power consumption VDD=3.3V	BORCFG==2'b00			1		uA
I_{PDR}	PDR power consumption VDD=3.3V				45		nA
V_{SVD}	Voltage monitoring threshold level	SVD[3:0]=0000	Fall		1.800		V
			Rise		1.900		
		SVD[3:0]=0001	Fall		2.014		V
			Rise		2.114		
		SVD[3:0]=0010	Fall		2.229		V
			Rise		2.329		
		SVD[3:0]=0011	Fall		2.443		V
			Rise		2.543		
		SVD[3:0]=0100	Fall		2.657		V
			Rise		2.757		
		SVD[3:0]=0101	Fall		2.871		V
			Rise		2.971		
		SVD[3:0]=0110	Fall		3.086		V
			Rise		3.186		
		SVD[3:0]=0111	Fall		3.300		V
			Rise		3.400		
		SVD[3:0]=1000	Fall		3.514		V
			Rise		3.614		
		SVD[3:0]=1001	Fall		3.729		V
			Rise		3.829		
		SVD[3:0]=1010	Fall		3.943		V
			Rise		4.043		
		SVD[3:0]=1011	Fall		4.157		V
			Rise		4.257		
		SVD[3:0]=1100	Fall		4.371		V
			Rise		4.471		
		SVD[3:0]=1101	Fall		4.586		V
			Rise		4.686		
		SVD[3:0]=1110	Fall		4.800		V
			Rise		4.900		
		SVD[3:0]=1111	Fall		-		V

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
			Rise		-		

Table 3-9 Reset and Supply Detection

Note:

[1] Based on feature parameter extraction

3.4.4 High Precision Reference

A high-precision reference voltage source is built in the chip to provide a high-precision and high-stability reference voltage for ADC and OPA.

When the chip leaves the factory, Fudan Microelectronics will use the on-chip ADC to sample the reference source output under a specific power supply voltage and temperature, and save the conversion result in the chip's Flash. This conversion value can be used as a reference in user applications. For detailed usage, please refer to Fudan Microelectronics Driver Library Function.

Symbol	Parameter
REF_CAL	ADC to VREF output conversion Test conditions: T_A=30±1℃ VDDA=3V±10mV
REF_RAW	VREF output voltage Test conditions: T_A=30±1℃ VDDA=3V±10mV

The main parameters of high precision reference are as follows:

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
V_{REF}	Reference output voltage[1]	-40℃≤T_A≤85℃	1.192	1.208	1.223	V
T_{setup}	Internal reference start-up time	Typical time: VDD=3V Maximum time: VDD=1.6V	-	0.5	2	ms
V_{VREF_MEAS}	Factory measurement conversion VDDA voltage of VREF	-	2.99	3	3.01	V
T_{coeff}	Internal reference temperature	-40℃≤T_A≤85℃		25	85	ppm/℃

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
	coefficient						
T_{S_VREF}	Sampling time of VREF measured by ADC	Pre enable VREF Buffer		10			us
T_{ADC_BUF}	Output fully build delay after VREF buffer enable[1]	VDD=3V ADC sample value stabilized to1LSB				100	us
I_{REF}	Reference working current[2]	T_A = 25℃ VDD=3V	PTAT_EN=0		1.8		uA
			PTAT_EN=1		2.6		

Table 3-10 High Precision Reference Parameters

Note:

[1] The parameters in the above table are based on characterization and are not included in the mass production test

[2] Based on circuit simulation

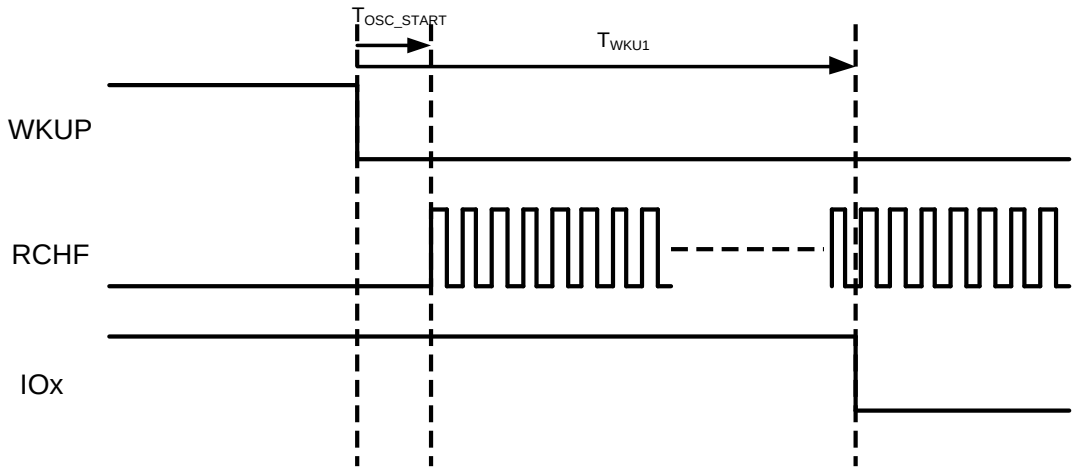
3.4.5 Wake Up Time in Low-power Mode

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
T_{WKU1}	Sleep/DeepSleep wake up time[1]	Use the WKUP pin to wake up, PRIMASK=1 to disable interrupts; after the CPU wakes up, execute the program to flip a certain IO output, and measure the time between the edge of the WKUP signal and the IO output flip FSYSCLK=8Mhz	-	6.8	-	us
T_{WKU2}	LP RUN mode wake up time		-	0	-	us

Table 3-11 Wake Up Time Parameters

[1] Based on characterization

Typical wake up event waveform, for design reference only



In the figure above, TOSC_START represents the RCHF start-up time after the wake event, and the typical value is less than 3.5us

TWKU1 is the time between the arrival of the wake-up event and the toggle of IO after the program runs, with a typical value of 6.8us.

If interrupts are not masked through PRIMASK, the wake event will cause the CPU enters the interrupt service routine. The process of the CPU entering the interrupt service routine introduces an additional delay time.

Note: *RCHF 8Mhz is used as the system clock after wake-up for time assessment. If 16Mhz or24Mhz frequency is selected after wake-up, the wake-up time will be shortened accordingly.*

3.4.6 External Clock Source Characteristics

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
f_{XTLF}	XTLF oscillation frequency	External 32768Hz crystal		32768		Hz
T_{start}	XTLF start-up time	External 32768Hzcrystal C_{load}=12pF XTLFIPW==3'b000		1	3	s

Table 3-12 Low-frequency Crystal Parameters

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
F_{XTHF}	XTHF oscillation frequency	VDD=3.3V		4	-	24	MHz
R_{fb}	Feedback resistor	-		-	200	-	KΩ
gm	Transconductance gain	VDD=5V	HF_CFG =00000	-	1.75	-	mA/V
			HF_CFG =00001	-	3.5	-	
		VDD=3V	HF_CFG =00000	-	1.2	-	
			HF_CFG =00001	-	2.4	-	
			HF_CFG =00010	-	3.48	-	
		VDD=1.6V	HF_CFG =00000	-	0.32	-	
			HF_CFG =00100	-	1.59	-	
			HF_CFG =00111	-	2.55	-	
VDD_{rise}	XTHF minimum supply voltage HF_CFG=00000	4MHz		1.08	-	-	V
		8MHz,		1.6	-	-	
		24MHz		1.8	-	-	
IDD	XTHF supply current HF_CFG=00000	4MHz		-	170	-	uA
		8MHz		-	200	-	
	XTHF supply current HF_CFG=11111	4MHz		-	1050	-	uA
		8MHz		-	1100	-	
T_{start1}	XTHF 8Mstart-up time	VDD=3.3V HF_CFG=00000		-	1.3	-	ms
		VDD=3.3V, HF_CFG=11111		-	0.3	-	ms
T_{start2}	XTHF 16Mstart-up time	VDD=3.3V, HF_CFG=00000		-	0.9	-	ms
		VDD=3.3V, HF_CFG=11111		-	0.1	-	ms
C_L	Load capacitance	-		5	-	25	pF

Table 3-13 High-frequency Crystal Parameters

Note:

[1] The above indicators are based on feature parameter extraction

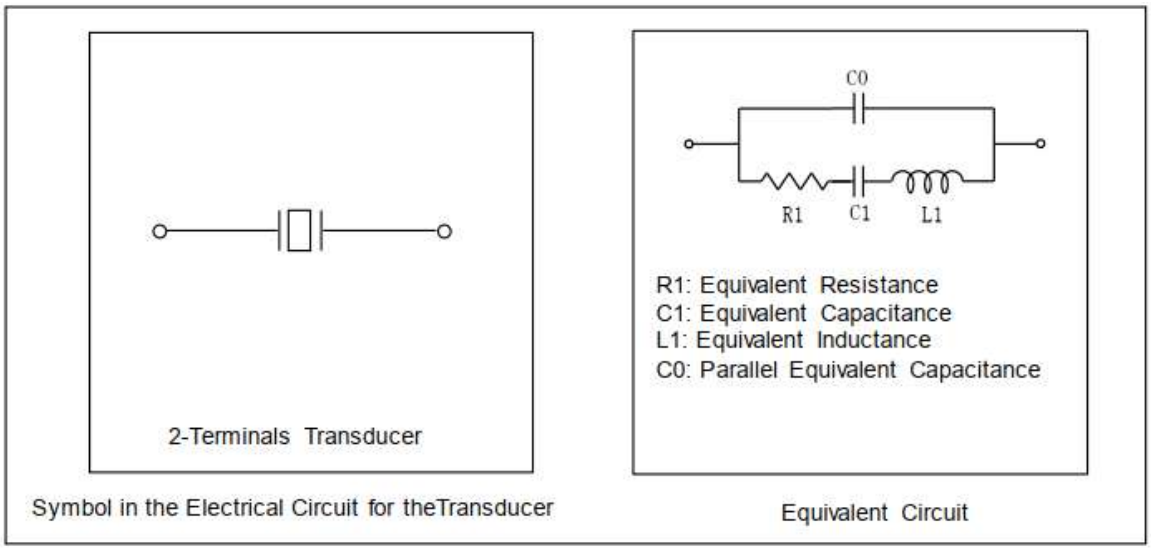
[2] It is recommended to use 8~16MHz crystal oscillator or ceramic oscillator with the recommended

oscillation intensity configuration to reduce the power consumption and noise radiation of the clock oscillator

XTHF typical gm parameters (for design reference only):

HF_CFG	Gm(mA/V)		
	VDD=5V	VDD=3V	VDD=1.6V
00000	1.75	1.2	0.32
00001	3.5	2.4	0.64
00010	4.91	3.48	0.95
00011	6.67	4.68	1.28
00100	8.07	5.77	1.59
00101	9.83	6.96	1.91
00110	11.2	8.05	2.23
00111	13	9.25	2.55
01000	14.4	10.3	2.86
01001	16.2	11.5	3.18
01010	17.6	12.6	3.5

The equivalent circuit model of crystal oscillator or ceramic oscillator is shown in the figure below:



The minimum loop gain for oscillator start-up can be calculated by the following formula:

$$g_{mcrit} = 4 \times ESR \times (2\pi F)^2 \times (C_0 + C_L)^2$$

C_L is the load capacitance required by the crystal oscillator or the built-in load capacitance of the ceramic oscillator. ESR can be calculated by the following formula:

$$ESR = R_1 \times \left(1 + \frac{C_0}{C_L}\right)^2$$

In order to ensure a sufficient and safe oscillation margin, g_m is usually required to be greater than 5 times g_{mcrit}. Based on this, a suitable XTHF oscillation intensity configuration can be selected according to the crystal oscillator or ceramic oscillator manual.

The following table takes the crystal oscillator and ceramic oscillator of muRata company as examples to give calculation examples:

Model	Type	Frequency (MHz)	ESR(typ) (Ohm)	C0 (pF)	CL (pF)	g_{mcrit} (mA/V)	Minimumgm (mA/V)
XRCGB24M000F2P00R0	Crystal	24	49.06	0.526	6pF	0.19	0.95
CSTNE8M00G550000R0	Ceramic	8	19.95	13.15	33	0.4294	2.147
CSTNE12M0G550000R0	Ceramic	12	12.81	14.44	33	0.6557	3.279

3.4.7 Internal Clock Characteristics

Internal high-frequency RC oscillator

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
f_{RCHF}[1]	RCHF frequency	VDD=1.8~5.5V	FSEL==2'b00	7.96	8	8.04	MHz
			FSEL==2'b01	15.92	16	16.08	
			FSEL==2'b10	23.88	24	24.12	
			FSEL==2'b11	31.84	32	32.16	
ACC_{RCHF}[2]	Frequency accuracy over full temperature range	VDD=1.8~5.5V	FSEL==2'b00 T= -40 ~+85℃	-1	-	1.5	%
			FSEL==2'b01 T= -40 ~+85℃	-2.5	-	3	%
			FSEL==2'b10 T= -40 ~+85℃	-3	-	4	%
ACC_{RCHF}[2]	Frequency accuracy over partial temperature range	VDD=1.8~5.5V	FSEL==2'b00 T= -10 ~+70℃	TBD	-	TBD	%
			FSEL==2'b01 T= -10 ~+70℃	TBD	-	TBD	%
			FSEL==2'b10 T= -10 ~+70℃	TBD	-	TBD	%

Table 3-14 Internal High-frequency RC Oscillator Parameters

Note [1]: Guaranteed by the mass production test.

Note [2]: Based on characterization.

Internal middle-frequency RC Oscillator

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
f_{RCLF}	RCLF oscillation frequency	VDD=1.65~5.5V T=25℃	612	615	618	KHz
I_{DD_RCLF}	RCLF current consumption	VDD=1.65~5.5V T=25℃		1		uA
t_{START}	RCLF start-up time	VDD=1.65~5.5V T=25℃		120		us

Table 3-15 Internal Middle-frequency RC Oscillator Parameters

Internal low-frequency RC Oscillator

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
f_{RCLP}	RCLP low power oscillating frequency	VDD=1.65~5.5V T=25℃	32.4	32.8	33.2	KHz
I_{DD_RCLP}	RCLP power consumption	VDD=1.65~5.5V T=25℃		350		nA
t_{START}	RCLP start time	VDD=1.65~5.5V T=25℃		380		us

Table 3-16 Internal Low-frequency RC Oscillator Parameters

3.4.8 PLL Characteristics

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
F_{PLL}	PLL output frequency	VDD=1.65~5.5V T=-40~85℃	32	-	64	MHz
I_{DD_PLL}	PLL current consumption	Input frequency 1Mhz，output frequency 32Mhz	-	450	-	uA
		Input frequency1Mhz，output frequency64Mhz	-	450	-	
t_{LOCK}	PLL lock		-	65	-	us

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
	time					

Table 3-17 PLLParameters

3.4.9 ADC Characteristics

3.4.9.1 Parameter Description

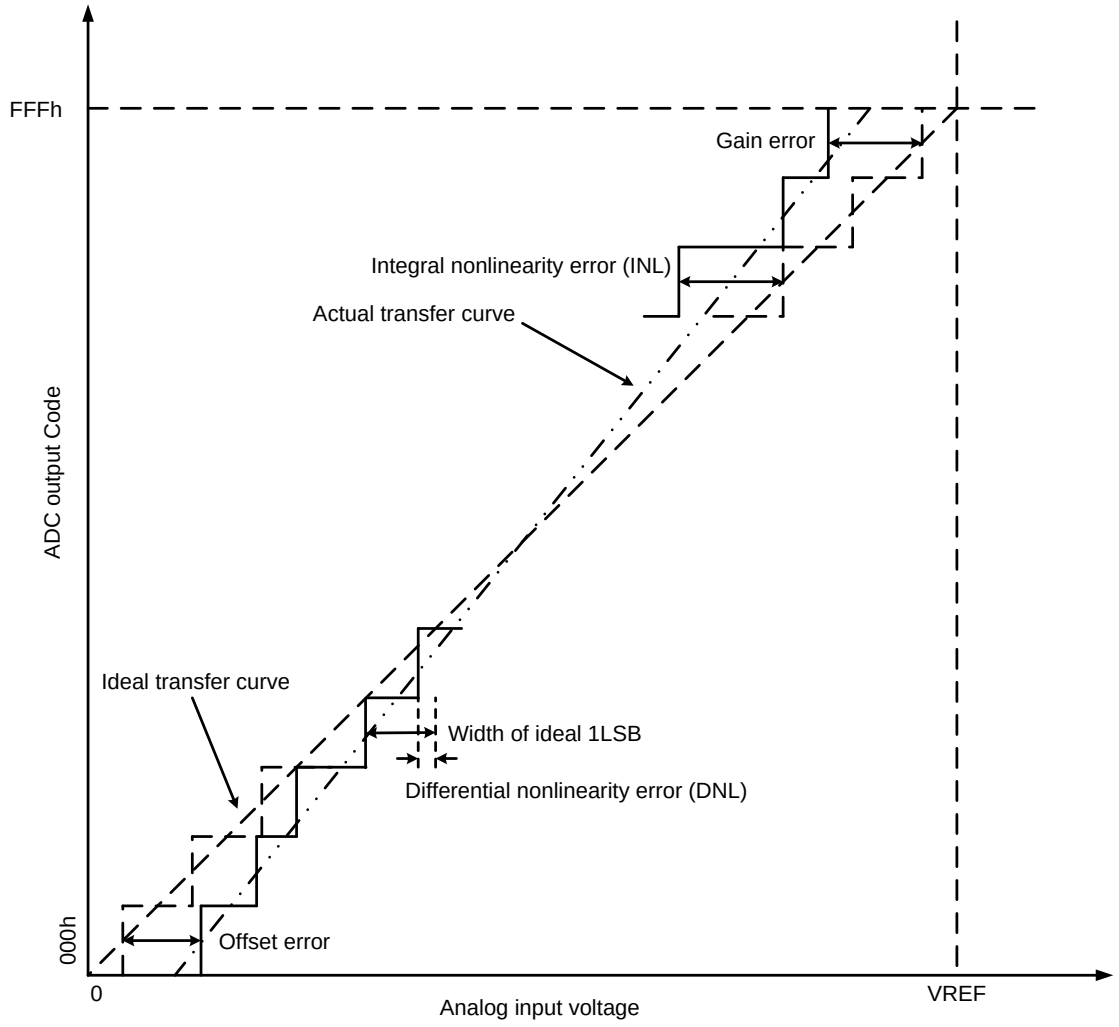


Figure 3–2 ADC Parameter Description

Differential Nonlinearity (DNL)

DNL represents the difference between the ideal ADC conversion curve 1LSB width and the actual ADC conversion curve 1LSB width.

Integral Nonlinearity (INL)

INL represents the maximum deviation between the actual ADC conversion curve and the ideal ADC

conversion curve.

Offset error

The offset error represents the difference between the position of the first codeword transition of the actual ADC and the position of the first codeword change of the ideal ADC.

Gain error

Gain error represents the difference between the position of the last codeword change of the actual ADC and the position of the last codeword change of the ideal ADC during full-scale input.

3.4.9.2 Performance Characteristic

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
VDDA	Operating voltage	-		1.65		5.5	V
VREF+	Positive reference voltage	-		1.5		VDDA	V
VREF-	Negative reference voltage	-		0		0.5	V
T_J	Junction temperature	-		-40		125	°C
V_{AIN}	Input voltage range	-		VREF-		VREF+	V
Vcm	Differential signal common mode input range	-		0.2		VREF-0.2	V
C_s	Sample and hold capacitor	-			3		pF
F_{CLK}	ADC working clock frequency	-				32	MHz
F_S	ADC sampling frequency	VDDA=2.0~5.5V				2	Msps
		VDDA=1.6~2.0V				1.5	
T_{SAMP}	Sample hold time	-		2		512	F_{CLK}
T_{CONV}	Conversion time	-			14		F_{CLK}
T_{CAL}	Self-calibration time	-			128	4096	F_{CLK}
IDD	ADC working current	VDDA=3.3V	Fs=1Msps	-	250	-	uA
			Fs=2Msps	-		-	
			Fs=250Ksps	-		-	
		VDDA=5V	Fs=1Msps	-	380	-	uA
			Fs=2Msps	-		-	

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
			Fs=250Ksps	-		-	
ADC Dynamic performance [1]							
ENOB	The relationship between the effective number of bits and the frequency of the input signal VDDA=3.3V VREF+=VDDA F_S=2Msps T_A=25℃ ADCCLK=XTHF	Single-ended mode F_{AIN}=29KHz		-	10.8	-	bits
		Differential mode F_{AIN}=29KHz		-	11.3	-	bits
		Single-ended mode F_{AIN}=499KHz		-	10.6	-	bits
		Differential mode F_{AIN}=499KHz		-	11.3	-	bits
	The relationship between the effective number of bits and the working voltage VREF+=VDDA F_{AIN}=29KHz T_A=25℃ ADCCLK=XTHF	VDDA=5.0V F_S=1Msps	Differential mode	-	11.4	-	bits
			Single-ended mode	-	10.9	-	
		VDDA=3.3V F_S=1Msps	Differential mode	-	11.4	-	bits
			Single-ended mode	-	10.8	-	
		VDDA=1.8V F_S=1Msps	Differential mode	-	11.1	-	bits
			Single-ended mode	-	10.3	-	
SNDR	Signal-to-noise distortion ratio VDDA=3.3V VREF+=VDDA F_S=1Msps -40℃<T_A<85℃	Single-ended mode F_{AIN}=29KHz		-	65	-	dB
		Single-ended mode F_{AIN}=499KHz		-	63	-	dB
SFDR	Spurious free dynamic range VDDA=3.3V VREF+=VDDA F_S=1Msps F_{AIN}=29KHz -40℃<T_A<85℃	Single-ended mode		-	78	-	dB
ADCStatic performance							
DNL	Differential nonlinearity VDDA=3.3V	Single-ended mode		-1	-	1	LSB
		Differential mode		-1	-	1	

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
	Fs=1Msps					
INL	Integral nonlinearity VDDA=3.3V Fs=1Msps	Single-ended mode	-1	-	2	LSB
		Differential mode	-1	-	1	
Offset Error	Offset error After calibration	Single-ended mode		-0.5		LSB
Gain Error	Gain error After calibration	Single-ended mode		-0.1		%

Table 3-18 ADC Parameters

Note:

[1] Based on feature parameter extraction

ADC typical static performance parameter chart

- **Differential input mode**

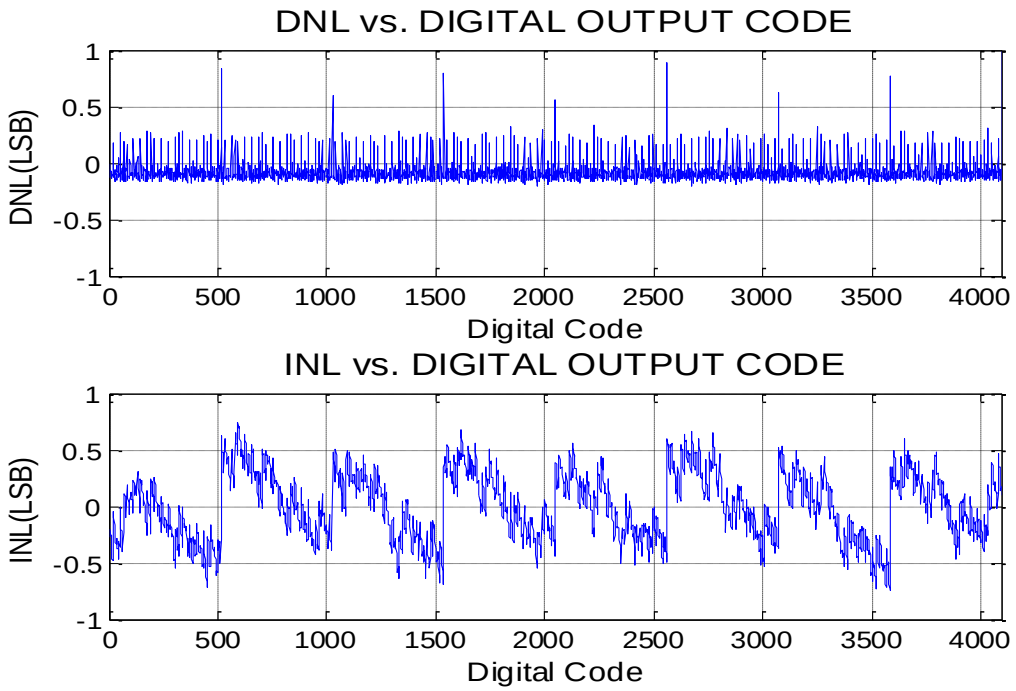


Figure 3–3 Typical DNL and INL of ADC Differential Input

- **Single-ended input mode**

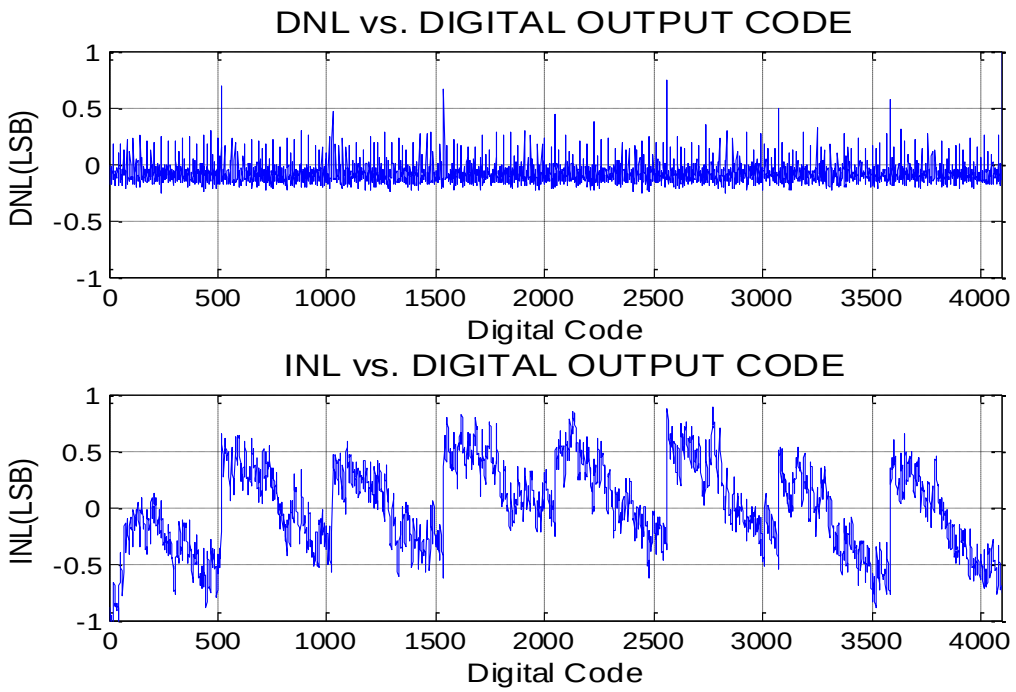


Figure 3–4 Typical DNL and INL of ADC Single-Ended Input

ADC typical signal-to-noise ratio spectrogram

- **VDD=VREFP=3.3V, F_S=2Msps, ADCCLK=XTHF32M, FAIN=29KHz**

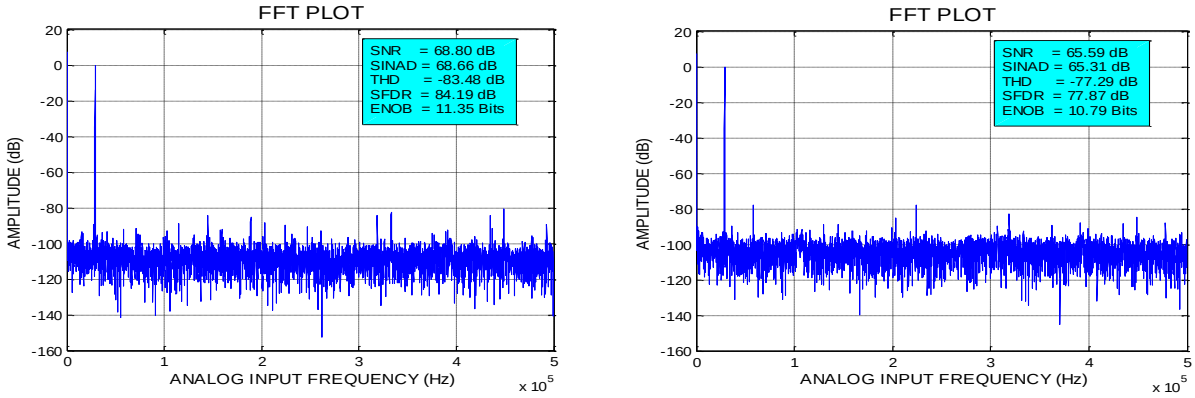


Figure 3–5 3.3V ADC Differential (Left) and Single-Ended Input (Right) Typical Signal-to-Noise Ratio

- **VDD=VREFP=1.8V, FS=2Msps, ADCCLK=XTHF32M, FAIN=29KHz**

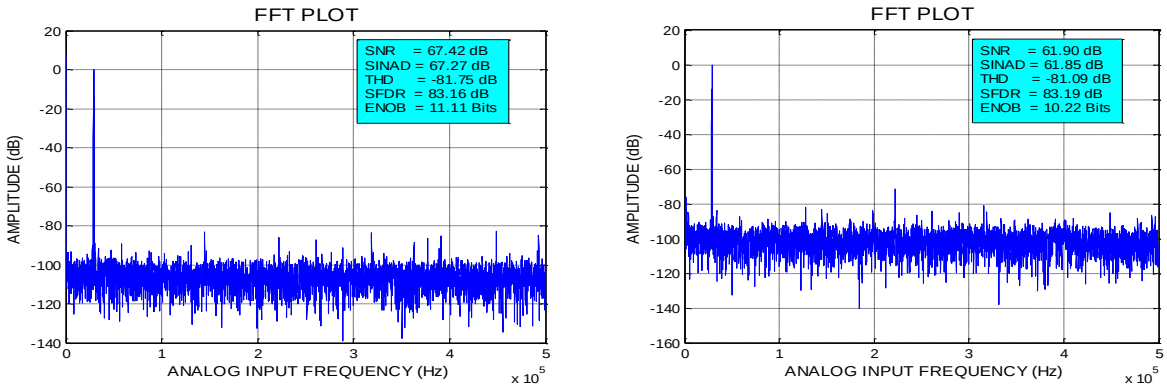


Figure 3–6 1.8V ADC Differential (Left) and Single-Ended Input (Right) Typical Signal-to-Noise Ratio

3.4.9.3 Input Channel Impedance

The following figure shows the ADC input channel impedance distribution.

- ADC_Inx means fast external channel
- ADC_Iny means slow external channel
- RIO represents the pin input switch impedance, RADC1 and RADC2 represent the ADC input fast channel impedance and slow channel impedance
- CS represents the internal sampling capacitance of ADC, with a typical value of 12.8pF
- Refer to the following table for impedance parameters

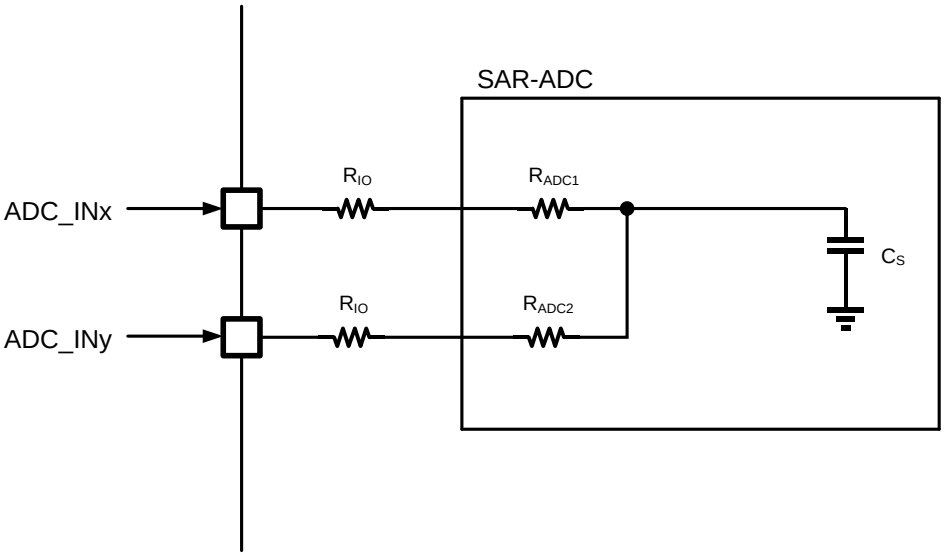


Figure 3–7 ADC Channel Input Impedance

3.4.9.4 Sampling Time

The minimum sampling time of ADC input signal is determined by the internal resistor of analog signal source, input channel impedance, pin parasitic capacitance and sampling capacitance.

The minimum sampling time of ADC sampling external input signal can be calculated according to the following formula:

$$T_{samp} = \ln\left(\frac{2^n}{SA}\right) \times \left(R_{AIN} + R_{ADC} + R_{IO}\right) \times C_{ADC}$$

Among them, n=12,SA=0.25LSB (The voltage on the sampling capacitor is established to within 0.25LSB error of the sampled signal level), R_{AIN} represents the internal resistor of the sampled signal source, R_{IO} represents input IO impedance, R_{ADC} represents ADC input channel impedance, C_{ADC} represents ADC sampling capacitance. And R_{IO} is 100Ω.

The R_{DAC} of the slow channel is affected by the power supply voltage, temperature and the amplitude of the input signal. The switch impedance is the largest when the input signal is VDDA/2. The fast channel has nothing to do with the input signal amplitude. The following table provides the R_{DAC} parameters of different channels under different power supply and temperature conditions. Users can calculate the minimum sampling time required based on these parameters and the characteristics of the signal source.

Symbol	VDDA	Temperature	Value			Unit
			Min	Typ	Max	
Fast channel, differential input						
R_{ADC}	5V	25C	-	182	-	Ω
		85C	-	240	303	
		-40C	-	126	159	
	3.3V	25C	-	280	-	
		85C	-	360	467	
		-40C	-	200	259	
	1.6V	25C	-	972	-	
		85C	-	1100	1612	
		-40C	-	819	1285	
Fast channel, single-ended input						
R_{ADC}	5V	25C	-	2222	-	Ω
		85C	-	2391	2724	
		-40C	-	2172	2471	
	3.3V	25C	-	2320	-	
		85C	-	2513	2887	
		-40C	-	2244	2569	
	1.6V	25C	-	2978	-	
		85C	-	3230	3973	
		-40C	-	2823	3511	
Slow channel, single-ended input, input signal level VDDA/2						

Symbol	VDDA	Temperature	Value			Unit
			Min	Typ	Max	
R_{ADC}	5V	25C	-	1285	-	Ω
		85C	-	-	1704	
		-40C	-	-	1437	
	3.3V	25C	-	1568	-	
		85C	-	-	2169	
		-40C	-	-	1795	
	1.6V	25C	-	5123	-	
		85C	-	-	8088	
		-40C	-	-	16970	

Table 3-19 ADC Input Impedance

- Suppose a fast channel is used to sample a signal source, and the internal resistor of the signal source is 1KΩ,When VDDA=3.3V and temperature is 25C, the minimum sampling time is 107ns according to Tsamp formula. If the working clock of ADC is 16Mhz, the sampling time configuration should be greater than 2 ADC clocks.
- Suppose a slow channel is used to sample a signal source, and the internal resistor of the signal source is 100KΩ, When the working power supply VDDA=1.6V and the working temperature is -40C~85C, the recommended minimum sampling time is 3.4us according to Tsamp formula. If the working clock of ADC is 8 Mhz, the sampling time configuration should be greater than 28 ADC clocks.

3.4.10 DAC Characteristics

Symbol	VDDA	Temperature	Value			Unit
			Min	Typ	Max	
VDDA	Operating voltage range	-	1.6	-	5.5	V
VREF	Reference voltage range	-	1.6	-	VDDA	V
IDD_{VREF}	VREF power consumption [1] VDDA=VREF=3.3V Buffer ON，No load	Middle code (0x800)		120		uA
		Worst code (0x000)		200		
	VREF power consumption [1] VDDA=VREF=3.3V Buffer OFF，No load	Middle code (0x800)		100		uA
		Worst code (0x000)		50		
IDD_{VDDA}	VDDA power consumption [1]	Middle code (0x800)		300		uA
		Worst code (0xF5C)		360		

Symbol	VDDA	Temperature	Value			Unit
			Min	Typ	Max	
	VDDA=VREF=3.3V Buffer ON，No load					
	VDDA power consumption [1] VDDA=VREF=3.3V Buffer OFF，No load	Middle code (0x800)		0		uA
		Worst code (0xF5C)		0		
V_{DAC}	Output voltage range[1]	Buffer ON	0.15	-	VREF-0.15	V
		Buffer OFF	0	-	VREF	
R_O	DAC output internal resistance	Buffer OFF[1]		16.8		KΩ
		Buffer ON[1]		5		Ω
R_L	Output resistive load [1]	Buffer ON R_L connected to VSSA	5			KΩ
C_L	Output capacitive load [1]	Buffer ON			50	pF
DNL	Differential nonlinearity [1]	Buffer ON	-1		+1	LSB
		Buffer OFF	-1		+1	LSB
INL	Integral nonlinearity [1]	Buffer ON VDDA=VREF=3.3V	-2		+2	LSB
		Buffer OFF VDDA=VREF=3.3V	-1		+2	LSB
Offset Error	Offset error[1] VDDA=VREF=3.3V	Buffer OFF Code=0x000		0.1		mV
Gain Error	Gain error[1] VDDA=VREF=3.3V	Buffer OFF Code=0xFFF		6		mV
t_{setup}	Full-scale output settling time	Code=0xFFF Buffer OFF		3		us
		Code=0xFFF Buffer ON		1		

Table 3-20 DAC Parameters

[1] Based on feature parameter extraction

DAC typical output curve (VDD=3.3V, BUFFER ON, RL=5KΩ)

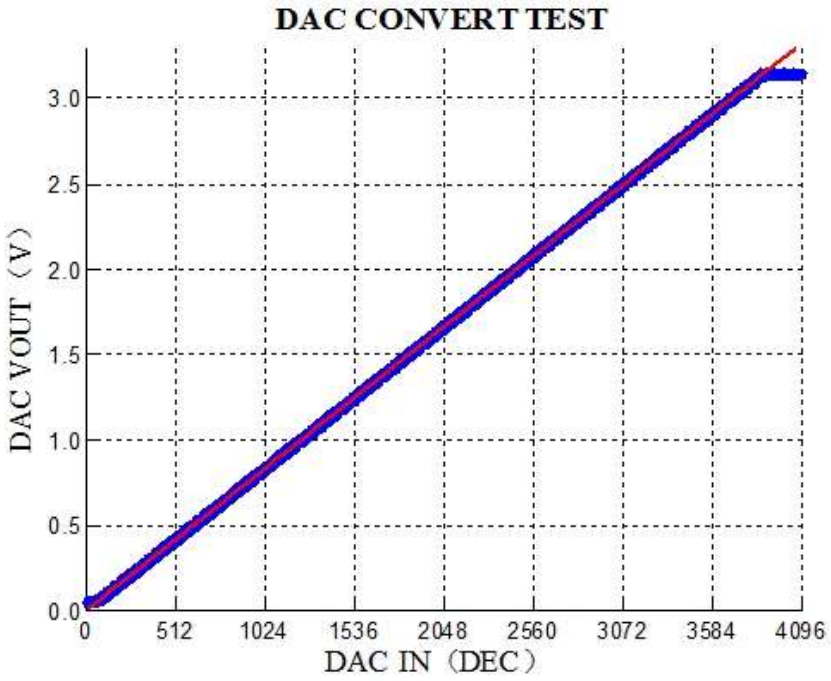


Figure 3–8 ADC Typical Output Curve

DAC Typical DNL and INL Curves

- **VDD=5V, BUFFER OFF, no load**

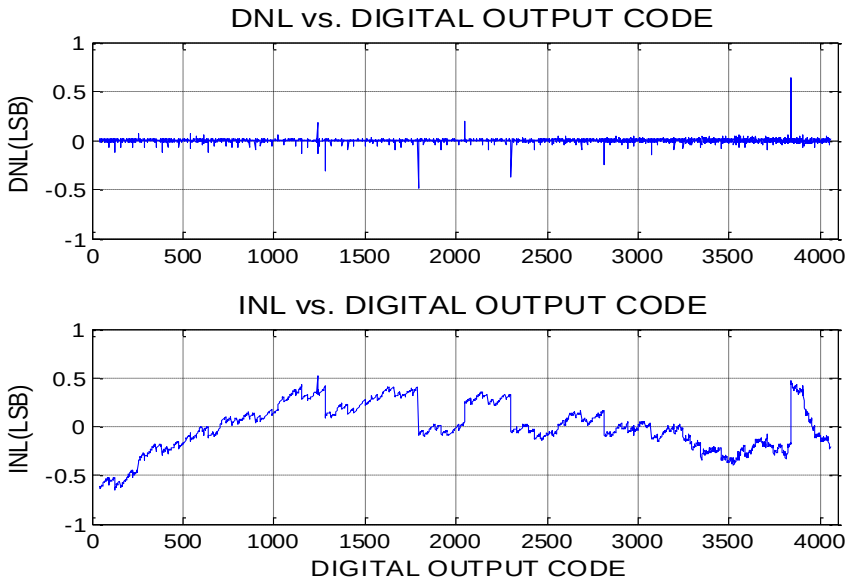


Figure 3–9 DAC Typical Static Characteristics (5V)

- **VDD=3.3V, BUFFER OFF, no load**

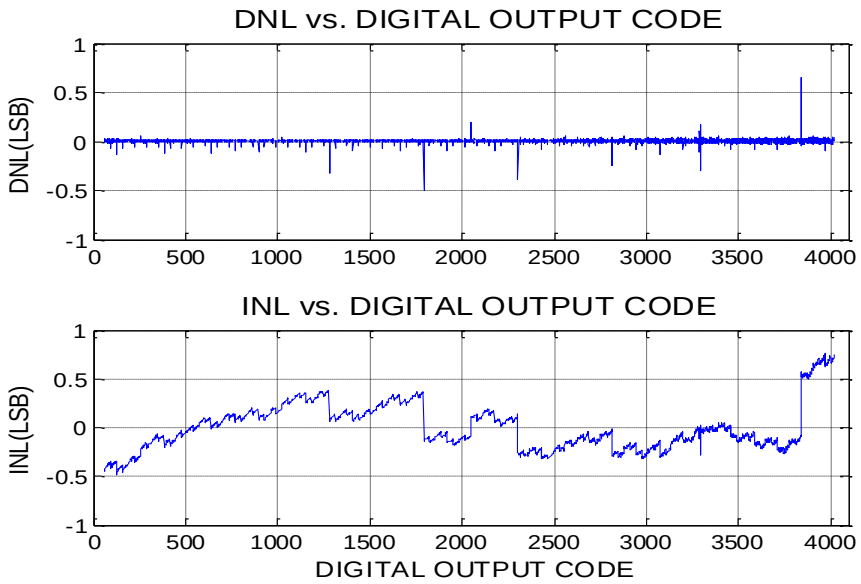


Figure 3–10 DAC Typical Static Characteristics (3.3V)

- **VDD=1.8V, BUFFER OFF, no load**

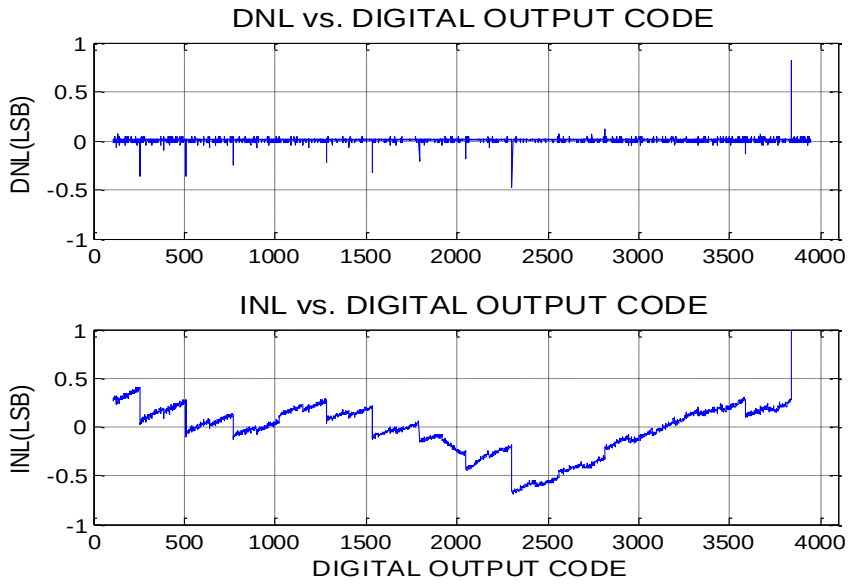


Figure 3–11 DAC Typical Static Characteristics (1.8V)

3.4.11 Temperature Sensor

Temperature sensor is factory-trimmed under such condition: VDD=5.0V and TA=30+/-1°C. Under this condition, ADC is used to sample and convert the temperature sensor output voltage. The conversion result is stored in Flash for user application. For detailed usage, please refer to the library functions

provided by Fudan Microelectronics.

Symbol	Parameter	Test conditions
TS_CAL1	Temperature sensor calibration value 1	VDD=3.0V, T_A=30+/-1°C

Note: According to the value of TS_CAL1, the temperature sensor output voltage absolute value at 30°C during temperature calibration can be calculated.

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
Reso	Resolution [1]	VDDA=VREF+=5V		2.04		LSB/°C
		VDDA=VREF+=3V		3.14		
Slope	Output slope [1]	T_A=-40~+85°C VDDA=1.8~5.5V		2.53		mV/°C
Linerity	Linearity in the whole temperature range[1]		-	+/-1	+/-2	°C
I_{DDA}	Temperature sensor consumption (not include ADC)[2]	VDDA=3.3V		0.8		uA
t_{START}	Temperature sensor start-time, include output buffer start-up time[2]	VREF1p2 enabled, set PTAT_EN register, VPTATBUFFER_OUTE , VPTATBUFFER_EN			50	us
		VREF1p2 disenabled			1.4	ms
t_{SAMPLE}	Sampling time required for ADC sampling temperature sensor output[2]		10	-	-	us

Table 3-21 Temperature Sensor Parameters

[1]: Based on characterization

[2]: Based on simulation

The temperature sensor output curve schematic diagram is as follows.

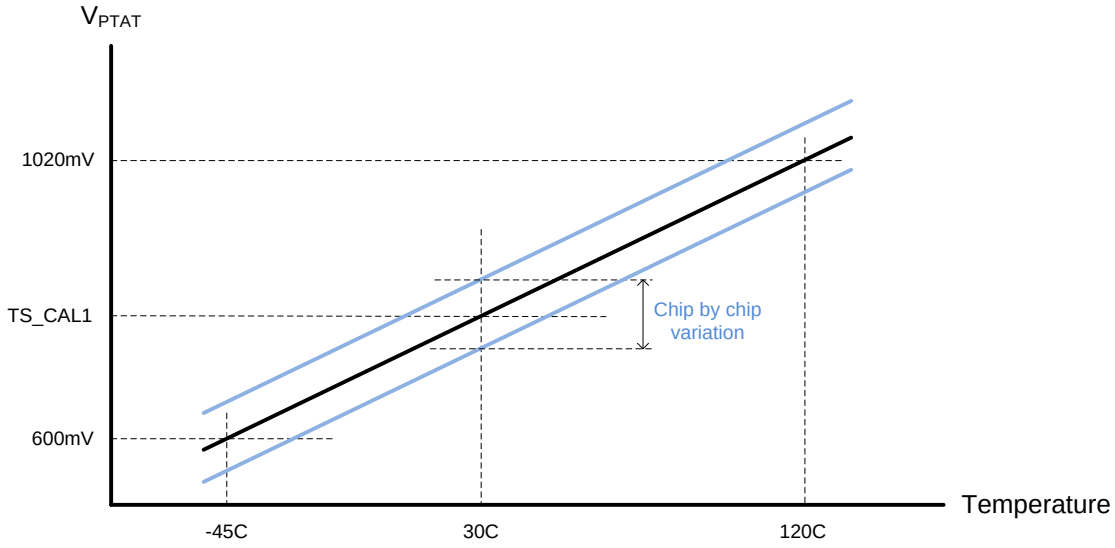


Figure 3–12 Temperature Sensor Output Curve

The temperature sensor output voltage is only related to the substrate temperature of the chip, and has nothing to do with the current working power voltage of the chip.

3.4.12 OPA Characteristics

Unless otherwise stated, the following parameters are based on feature parameter extraction.

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
VDDA	Operating voltage range		1.8		5.5	V
Vos	Input offset voltage	Before calibration VDDA=1.8~5.5V VCM=VDDA/2 T_A=25C		±1.5		mV
Ibias	Input bias current	VDDA=3.3V VCM=VDDA/2 T_A=25C			±100	pA
Ios	Input offset current				±100	pA
TRIMSTEP_P	Low common mode input voltage offset trim step size（0.1xVDDA）	VDDA=3.3V VCM=VDDA/2 T_A=25C		0.04		mV
TRIMSTEP_N	High common mode input voltage offset trim step size（0.9xVDDA）			0.05		

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
VOHsat	The difference between output high saturation voltage and VDDA	VCM=VDDA/2 R_L=4KOhm	VDDA=2.0V	-	200	-	mV
			VDDA=3.3V	-	300	-	
			VDDA=5.5V	-	400	-	
		VCM=VDDA/2 R_L=50KOhm	VDDA=2.0V	-	40	-	mV
			VDDA=3.3V	-	60	-	
			VDDA=5.5V	-	100	-	
VOLsat	The difference between output low saturation voltage and VSSA	VCM=VDDA/2 R_L=4KΩ	VDDA=2.0V	-	160	-	mV
			VDDA=3.3V	-	190	-	
			VDDA=5.5V	-	300	-	
		VCM=VDDA/2 R_L=50KΩ	VDDA=2.0V	-	20	-	mV
			VDDA=3.3V	-	20	-	
			VDDA=5.5V	-	30	-	
I_{LOAD}	Output saturation drive current	R_L=4KΩ	VDDA=2.0V	-	400	-	uA
			VDDA=3.3V	-	680	-	
			VDDA=5.5V	-	1150	-	
		R_L=50KΩ	VDDA=2.0V	-	40	-	uA
			VDDA=3.3V	-	65	-	
			VDDA=5.5V	-	110	-	
C_{LOAD}	Capacitive load					50	pF
CMIR	Common mode input range	VDDA=3.3V Voltage follower or PGA configuration		0.01		VDDA-0.01	V
CMRR	Common mode rejection ratio	VDDA=3.3V VCM=0.3~3V			66		dB
		VDDA=5V VCM=0.5~4.8V			70		dB
PSRR	Power supply rejection ratio	VDDA=1.8~5.5V VCM=VDDA/2		80			dB
GBW	-3dB bandwidth	VDDA=3.3V	VCM=1.0~2.5V		3500		KHz
			VCM=0.2V		2200		
			VCM=0.015V		1200		
			VCM=3V		1800		

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
SR	Rising slew rate (The output voltage varies from 10% to 90%)	VDDA=2.0V			1.1		V/us
		VDDA=3.3V			1.25		
		VDDA=5V			1.45		
	Decrease slew rate (The output voltage varies from 90% to 10%)	VDDA=2.0V			1.3		V/us
		VDDA=3.3V			1.4		
		VDDA=5V			1.6		
AO	Open loop gain	VDDA=3.3V			100		dB
Phi	Phase margin[1]	C_{LOAD}=50pF VDDA=5V			80		°
GM	Gain margin [1]	C_{LOAD}=50pF VDDA=5V			11		dB
t_{START}	Start Time	Buffer mode VDDA=3.3V			2.2		us
PGA gain	PGA gain	VDDA=3.3V In-phase amplification	Gain=2		2		-
			Gain=4		3.99		
			Gain=8		7.96		
			Gain=16		15.85		
		VDDA=3.3V Inverted amplification	Gain=1		0.99		-
			Gain=3		2.95		
			Gain=7		6.79		
			Gain=15		14.1		
PGA BW	PGA band width	Gain=2			GBW/2		
		Gain=4			GBW/4		
		Gain=8			GBW/8		
		Gain=16			GBW/16		
I_{DDA}	consumption	Standalone，Normal mode，No load VDDA=3.3V VCM=VDDA/2			150		uA
		Buffer，Low power mode，No load VDDA=3.3V VCM=VDDA/2			1.5		

Table 3-22 OPAParameters

[1] Based on circuit design simulation

3.4.13 Analog Comparator Characteristics

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
VDDA	Comparator operating voltage range	-		1.8	-	5.5	V
VI_{comp1}	Comparator1 input voltage range	-		0	-	VDDA	V
VI_{comp2}	Comparator2 input voltage range	-		0	-	VDDA	V
VI_{comp3}	Comparator3 input voltage range	-		0	-	VDDA-0.7	V
I_{comp12}	Comparator1/2 operating current	VDDA=3.3V 50Khz Square wave, ±100mV overdrive	Low power mode	-	9	-	uA
			Medium speed mode	-	9.2	-	
			High speed mode	-	32	-	
		VDDA=3.3V DC Input	Low power mode	-	1	-	
			Medium speed mode	-	1.5	-	
			High speed mode	-	25	-	
$T_{propagation12}$	Comparator 1/2 propagation delay	VDDA=3.3V 200mV step 100mV overdrive	Low power mode	-	0.8	-	us
			Medium speed mode	-	0.6	-	
			High speed mode	-	150	-	ns
$T_{setup12}$	Comparator 1/2 setup time	VDDA=3.3V	Low power mode	-	20	-	us
			Medium speed mode	-	15	-	
			High speed mode	-	5	-	
I_{comp3}	Comparator 3 operating current	VDDA=3.3V VDDA=3.3V 50Khz Square wave, ±100mV overdrive		-	8	-	uA

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
		VDDA=3.3V DC Input	-	150	-	nA
$T_{propagation3}$	Comparator 3 propagation delay	VDDA=3.3V	-	1.5	-	us
T_{setup3}	Comparator 3 setup time	VDDA=3.3V	-	20	-	us

Table 3-23 Analog Comparator Parameters

3.4.14 Flash Characteristics

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
	Flash size		64K	-	256K	bytes
T_{PROG}	Byte Program Time		6	-	7.5	μs
T_{ERASE}	Sector/Block Erase		4	-	5	ms
	Chip Erase		30	-	40	ms
N_{ED}	Sector Endurance		100,000			Erase/Write cycles
T_{DR}	Data Retention	T=85℃ After 20K cycling	10			yrs

Table 3-24 FlashParameters

3.4.15 GPIO Characteristics

General-purpose IO

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
V_{IL}	Input low voltage			0		$0.3V_{DD}$	V
V_{IH}	Input high voltage			$0.7V_{DD}$		V_{DD}	V
I_{IL}	Input low leakage	V_{IL}=0V		-1		1	μA
I_{IH}	Input high leakage	V_{IH}=5V		-1		1	μA
V_{OL}	Output low voltage	V_{DD}=5V I_{SINK}=20mA	PA11,PA12		1.1		V
		V_{DD}=5V I_{SINK}=5mA	PC12		1		
			PH15		0.3		
		V_{DD}=5V I_{SINK}=10mA	other		1.05		

Symbol	Parameter	Test conditions		Value			Unit
				Min	Typ	Max	
V_{OH}	Output high voltage	V_{DD}=5V I_{SOURCE}=20mA	PA11,PA12		3.75		V
		V_{DD}=5V I_{SOURCE} =5mA	PC12		3.95		
			PH15		3.9		
		V_{DD}=5V I_{SOURCE} =10mA	Other		3.8		
V_{OL}	Output low voltage	V_{DD}=3.3V I_{SINK}=10mA	PA11,PA12		0.6		V
		V_{DD}=3.3V I_{SINK}=5mA	PC12		1.1		
			PH15		0.4		
		V_{DD}=3.3V I_{SINK}=5mA	Other		0.6		
V_{OH}	Output high voltage	V_{DD}=3.3V I_{SOURCE}=10mA	PA11,PA12		2.5		V
		V_{DD}=3.3V I_{SOURCE} =1.5mA	PC12		2.9		
			PH15		2.8		
		V_{DD}=3.3V I_{SOURCE} =5mA	Other		2.6		
R_{PU}	Pull-up resistor				100		KΩ

Table 3-25 GPIO Parameters

NRST Pin

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
V_{IL}	Input low voltage		0		$0.3V_{DD}$	V
V_{IH}	Input high voltage		$0.7V_{DD}$		V_{DD}	V
I_{IL}	Input low leakage	V_{IL}=0V	-1		1	μA
I_{IH}	Input high leakage	V_{IH}=5V	-1		1	μA
R_{PU}	Pull-up resistor			5		KΩ
$T_{AFILTER}$	Analog filter length[1]	VDD=5V		100		ns
		VDD=3V				ns
$T_{DFILTER}$	Digital filter length[1]	VDD=1.65~5.5V $-40℃<T_A<85℃$	50		100	us

Table 3-26 NRST Pin Parameters

Note [1]: Based on characterization

GPIO AC characteristics

IO	Symbol	Parameter[1]	Test conditions	min	max	Unit
Non FM+	Fmax	Maximum frequency	C=30pF,2.7V<Vdd<3.6V	-	45	MHz
			C=30pF,1.6V<Vdd<2.7V	-	22	
			C=10pF,2.7V<Vdd<3.6V	-	80	
			C=10pF,1.6V<Vdd<2.7V	-	40	
	Tr/Tf	Output rise and fall time	C=30pF,2.7V<Vdd<3.6V	-	8.7	ns
			C=30pF,1.6V<Vdd<2.7V	-	16.9	
			C=10pF,2.7V<Vdd<3.6V	-	3.4	
			C=10pF,1.6V<Vdd<2.7V	-	6.7	
FM+	Fmax	Maximum frequency	C=50pF,1.6V<Vdd<3.6V	-	10	MHz
	Tf	Output fall time		-	27	ns

Table 3-27 Pin AC Parameters

Note [1]: Based on simulation and are not included in the mass production test

3.4.16 LCD Characteristics

On-chip resistor voltage divider

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
I_{LCD}	Operating current of LCD(no load) w/ internal resistor voltage divider[1]	VDD=5V		2		uA
		VDD=3V				
V_{LCD}	LCD bias voltage	-	0.547× VDD		VDD	V

Table 3-28 LCD On-Chip Resistor Voltage Divider

[1] Based on feature parameter extraction

Off-chip capacitor mode

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
I_{LCD}	LCD operating current in on-chip resistor divider mode (no load) [1]	VDD=5V				uA
		VDD=3V				
V_{LCD}	LCD bias voltage	-	0.547× VDD		VDD	V

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
C1	Decoupling capacitor between VCIN1 and VCIN2	-		0.1		uF
C20	VDISP0 decoupling capacitor to ground	-		0.1		uF
C21	VDISP1 decoupling capacitor to ground	-		0.1		uF
C22	VDISP2 decoupling capacitor to ground	-		0.1		uF
C23	VDISP3 decoupling capacitor to ground	-		0.1		uF

Table 3-29 LCD Off-Chip Capacitor Drive

[1] Based on feature parameter extraction

The off-chip connection of the LCD capacitor drive mode is shown in the figure below:

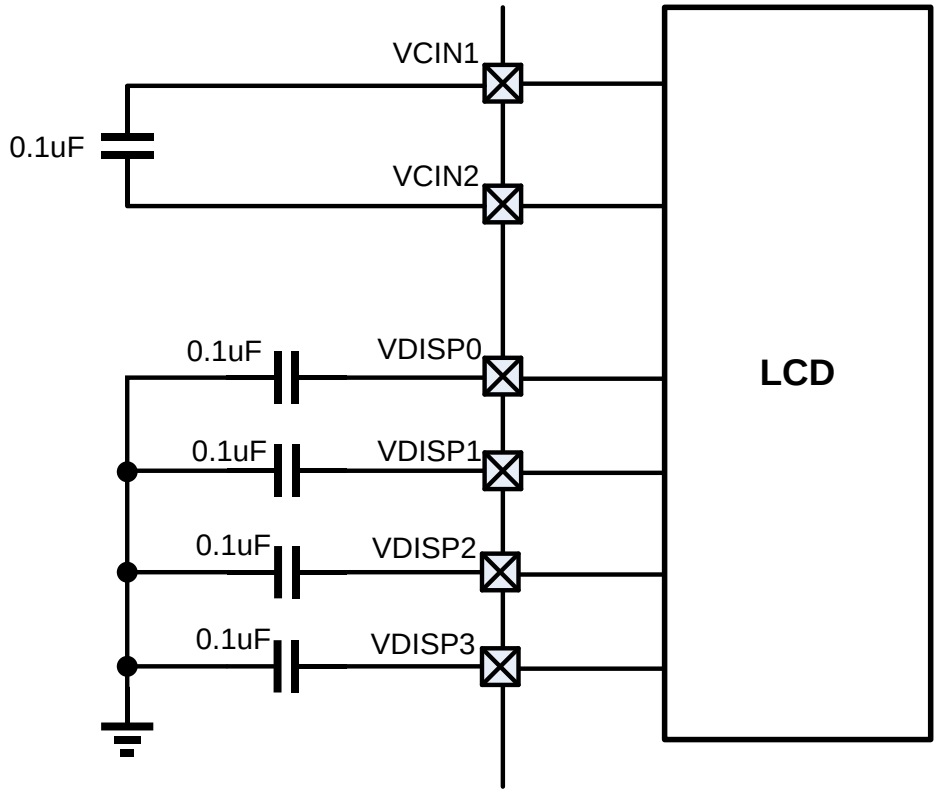


Figure 3–13 LCD Off-chip Capacitor Drive Mode Capacitor Connection

3.4.17 VBAT Characteristics

Symbol	Parameter	Test conditions	Value			Unit
			Min	Typ	Max	
R	VBAT measuring circuit voltage divider resistance			100		KΩ
Q	VBAT divider ratio	VBAT_MEASURE=VBAT /Q		3		
Er	Partial pressure ratio error [1]		-1	-	1	%
t_{S_VBAT}	ADC sampling time when measuring VBAT[1]	ADC internal buffer enable				µs
		ADC internal buffer disable				

Table 3-30 VBAT Measurement Characteristics

Note [1]: Based on simulation and are not included in the mass production test

4 Bus and Memories

4.1 System Bus

The FM33LG0 bus architecture consists of the following main components:

- 2 Masters
  - Cortex-M0
  - DMA controller
- 4 Slaves
  - Internal Flash
  - Internal SRAM
  - GPIO controller module
  - AHB-APB bus transfer bridge

The system bus diagram of FM33LG0XX is as follows, including an AHB-Lite bus and an APB bus

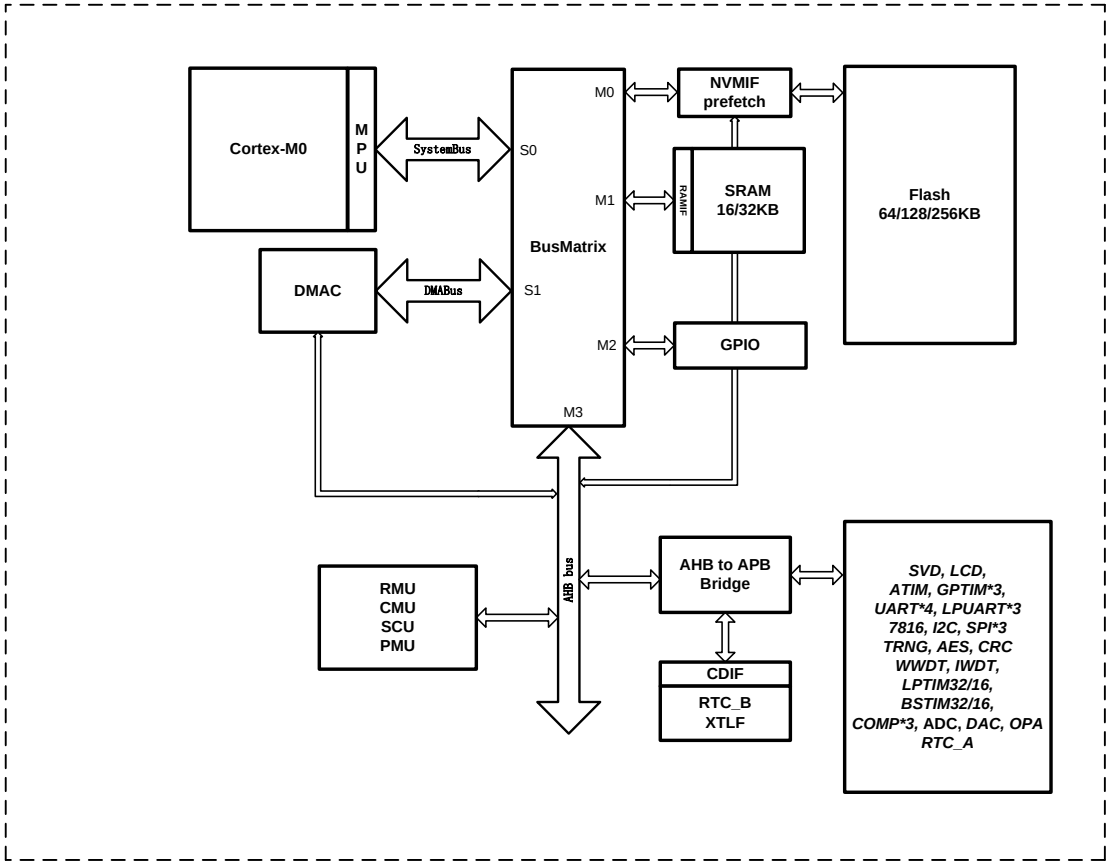


Figure 4–1 System Bus Diagram

4.2 Memory Space Allocation

4.2.1 Introduction

The Flash Sector size is 512 bytes, 4 pages can constitute a sector of 2K bytes.

Flash contains 4 INFO pages, 2 LDT pages, 1 redundant page, and 1 DCT page. Among them, DCT and LDT are reserved and are not available to users. INFO pages are used to hold user configuration Information. All Option pages are logically isolated from the Flash main area.

The address space of FM33LC0XX is allocated as follows (256KB Flash, 32KB RAM):

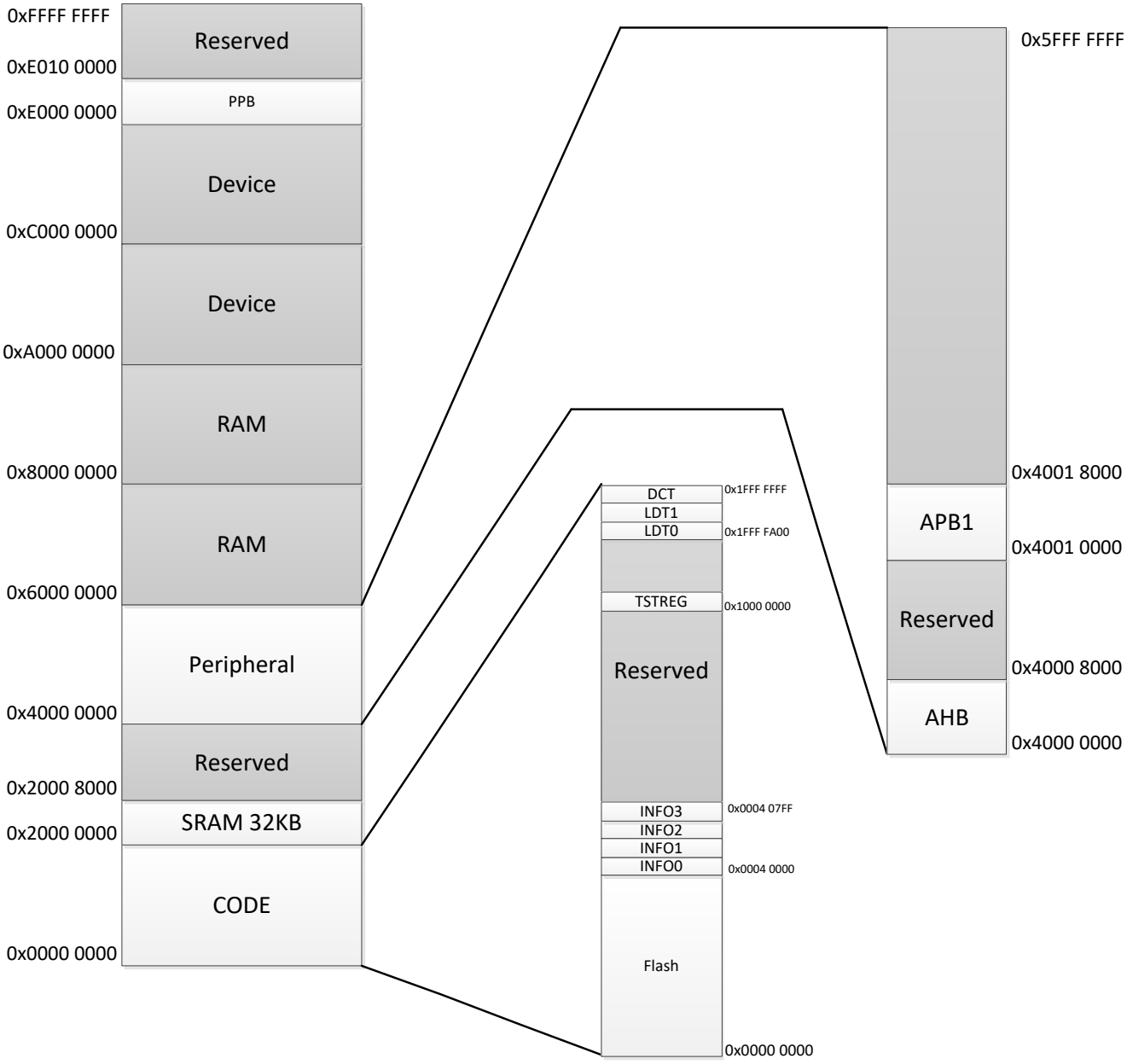


Figure 4–2 FM33LG04x Bus Address

128KB flash+32KB RAM：

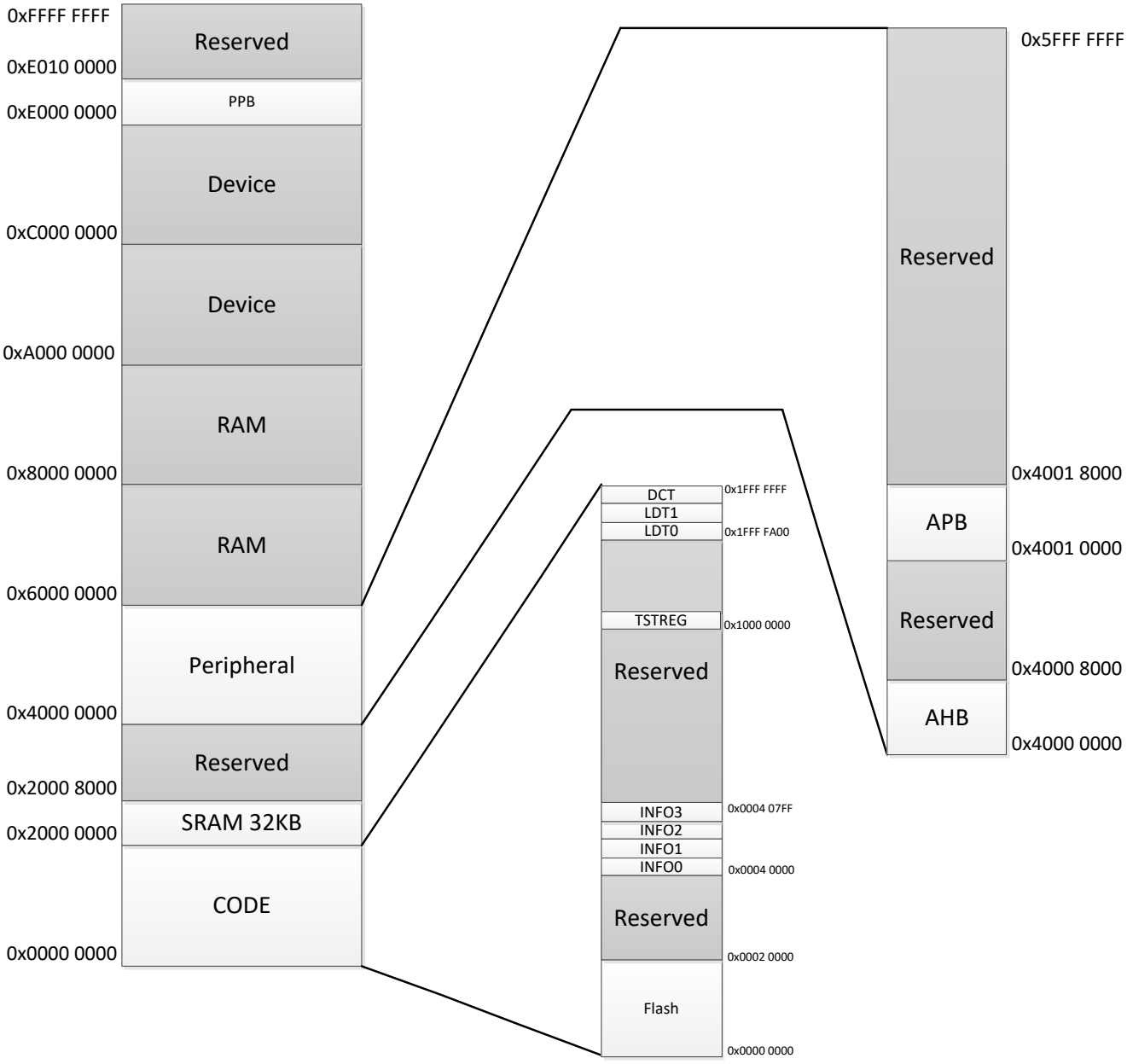


Figure 4–3 FM33LG02x Bus Address

64KB flash+16KB RAM:

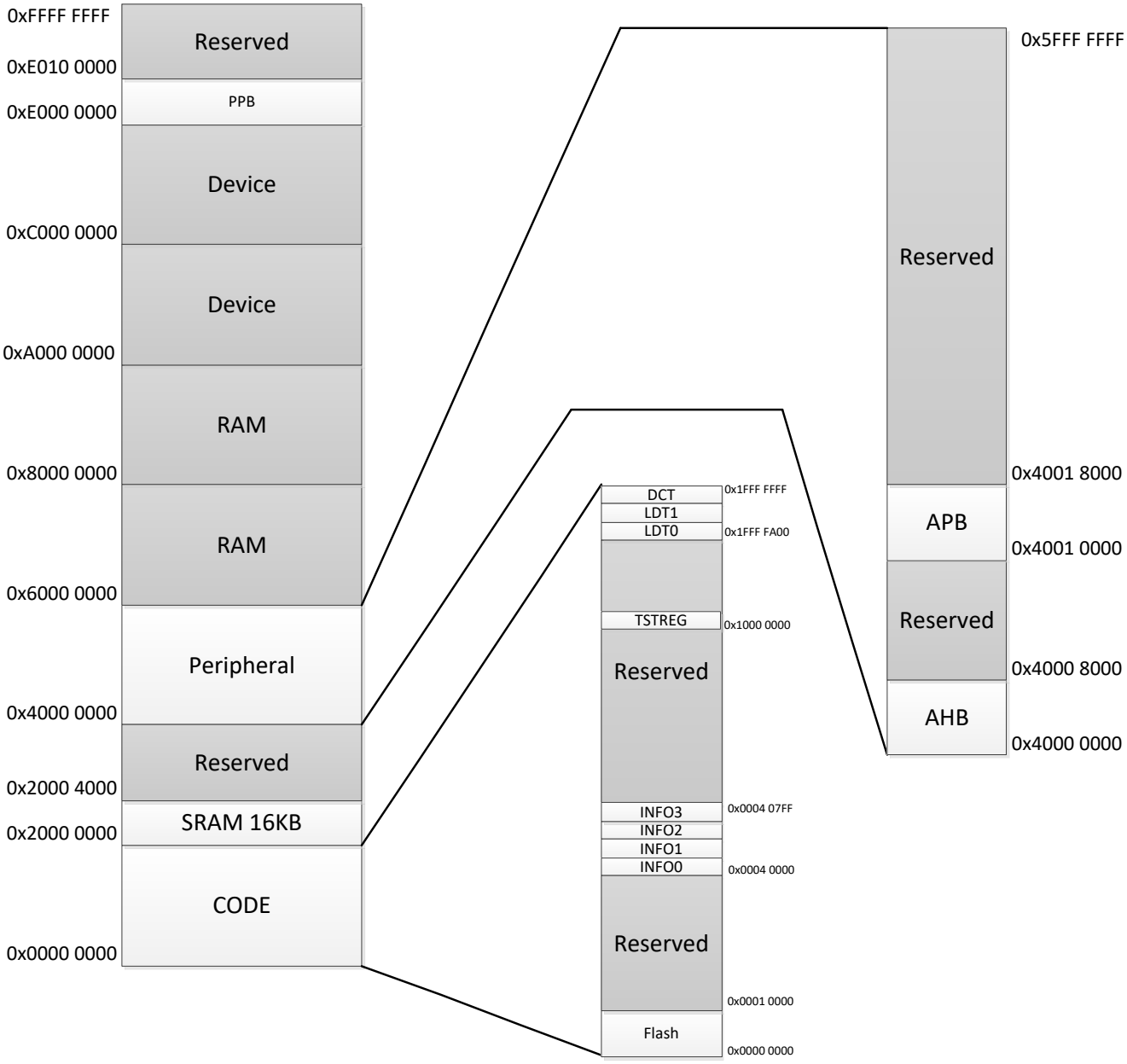


Figure 4–4 FM33LG01x Bus Address

4.2.2 Peripherals Address Assignment

The following table lists the address space allocation for all peripherals, each occupying 1KB of address space.

Bus	Range of address	Memory	Peripheral
AHB	0x0000_0000~0x0007_FFFF	512KB	Flash main array
	0x1FFF_F000~0x1FFF_FFFF	4KB	Flash Option cell array
	0x2000_0000~0x2000_7FFF	32KB	SRAM

	0x2000_8000~0x2001_3FFF	-	-
	0x4000_0000~0x4000_03FF	1KB	SCU
	0x4000_0400~0x4000_07FF	1KB	DMA
	0x4000_0800~0x4000_0CFF	-	-
	0x4000_0C00~0x4000_0FFF	1KB	GPIO
	0x4000_1000~0x4000_13FF	1KB	NVMIF Registers
	0x4000_1400~0x4000_17FF	-	-
	0x4000_1800~0x4000_1BFF	-	-
	0x4000_1C00~0x4000_1FFF	-	-
	0x4000_2000~0x4000_23FF	1KB	PMU
	0x4000_2400~0x4000_27FF	1KB	CMU
	0x4000_2800~0x4000_2BFF	1KB	RMU
	0x4001_0000~0x4001_FFFF	64KB	APB
APB	0x4001_0000~0x4001_03FF	1KB	CRC
	0x4001_0400~0x4001_07FF	1KB	SPI0
	0x4001_0800~0x4001_0BFF	1KB	SPI1
	0x4001_0C00~0x4001_0FFF	1KB	LCD
	0x4001_1000~0x4001_13FF	1KB	RTC_A
	0x4001_1400~0x4001_17FF	1KB	IWDT
	0x4001_1800~0x4001_1BFF	1KB	WWDT
	0x4001_1C00~0x4001_1FFF	1KB	U7816
	0x4001_2000~0x4001_23FF	1KB	UART0
	0x4001_2400~0x4001_27FF	1KB	I2C
	0x4001_2800~0x4001_2BFF	1KB	SVD
	0x4001_2C00~0x4001_2FFF	1KB	RAMBIST
	0x4001_3000~0x4001_33FF	1KB	ATIM
	0x4001_3400~0x4001_37FF	1KB	LPTIM32
	0x4001_3800~0x4001_3BFF	1KB	AES
	0x4001_3C00~0x4001_3FFF	1KB	TRNG
	0x4001_4000~0x4001_43FF	1KB	LPUART0
	0x4001_4400~0x4001_47FF	1KB	LPUART1
	0x4001_4800~0x4001_4BFF	1KB	SPI2
	0x4001_4C00~0x4001_4FFF	1KB	GPTIM0
	0x4001_5000~0x4001_53FF	1KB	LPUART2
	0x4001_5400~0x4001_57FF	1KB	COMPx
	0x4001_5800~0x4001_5BFF	1KB	AutoTrim
	0x4001_5C00~0x4001_5FFF	1KB	ADC
	0x4001_6000~0x4001_63FF	1KB	BSTIM32
	0x4001_6400~0x4001_67FF	1KB	GPTIM1
	0x4001_6800~0x4001_6BFF	1KB	UART1
	0x4001_6C00~0x4001_6FFF	1KB	PGL
	0x4001_7000~0x4001_73FF	1KB	UART3

	0x4001_7400~0x4001_77FF	1KB	UART4
	0x4001_7800~0x4001_7BFF	1KB	UART5
	0x4001_7C00~0x4001_7FFF	1KB	UARTIR
	0x4001_8000~0x4001_83FF	1KB	GPTIM2
	0x4001_8400~0x4001_87FF	1KB	-
	0x4001_8800~0x4001_8BFF	1KB	LPTIM16
	0x4001_9000~0x4001_93FF	1KB	ANTEST(BUF4TST)
	0x4001_9800~0x4001_9BFF	1KB	DAC
	0x4001_9C00~0x4001_9FFF	1KB	DIVAS
	0x4001_A000~0x4001_A3FF	1KB	OPA
	0x4001_A400~0x4001_A7FF	1KB	VREF1p2
	0x4001_E000~0x4001_E3FF	1KB	CDIF controller
	0x4001_F000~0x4001_FFFF	4KB	CDIF(RTC_B, XTLF, VAO)

Table 4-1 Peripherals Address Assignment

4.3 RAM

4.3.1 Introduction

The FM33LG0XX contains a 32KB RAM (8K*32), address space range is 0x2000_0000 ~ 0x2000_7FFF, the software can carry out byte, half word, word access to SRAM, CPU and DMA can achieve single-cycle read and write without waiting for SRAM at the maximum system frequency. CPU can also execute program in SRAM. Program code can be imported into SRAM to achieve the highest frequency of wait-free execution.

4.4 Flash

4.4.1 Introduction

FM33LG0XX implements up to 256KB of Flash memory; the array organization includes Page (512B), Sector (2KB).

Main Array contains a total of 512 pages and supports page erase, sector erase and matrix erase.

4.4.2 Special Information Sector Description

There are several special sectors for users to use, the description is as follows:

Area	Description	Use
LDT0	FMSH data area	Save FMSH adjustment information, mode words, test data, etc.; the software is read-only
LDT1	User option data area	User option byte（OPTBYTES）
RED	Redundant information	Save information for replacement of failed sectors
IF	Information area	4 pages total 2KB, for users to use; software can read and write

Table 4-2 Special Information Sector

4.4.2.1 LDT0 Page

The chip parameters written by the original factory are stored in LDT0, and the software can only be read and cannot be rewritten.

The bus address of LDT0 is 0x1FFF_FA00~0x1FFF_FBFF; the following parameter software can be read from LDT0 and written to the corresponding control register in the application to achieve analog parameter calibration.

AHB address	Bit[31:16]	Bit[15:0]	Description
0x1FFF_FA84	~VREFREG45_TRIM	VREFREG45_TRIM	VREFP_REGU 4.5V trim value
0x1FFF_FA88	~VREFREG30_TRIM	VREFREG30_TRIM	VREFP_REGU 3.0V trim value
0x1FFF_FA8C	~VREFREG25_TRIM	VREFREG25_TRIM	VREFP_REGU 2.5V trim value
0x1FFF_FA90	~VREFREG20_TRIM	VREFREG20_TRIM	VREFP_REGU 2.0V trim value
0x1FFF_FA94	~VREFREG15_TRIM	VREFREG15_TRIM	VREFP_REGU 1.5V trim value
0x1FFF_FA98	~ULPBG_TRIM	ULPBG_TRIM	ULPBG trim value
0x1FFF_FB08	~VREFCAL	VREFCAL	ADC conversion value to VREF1p2 under 3V, 30C
0x1FFF_FB0C	VREFRAW		VREF1p2 actual voltage value
0x1FFF_FB10	~TS_CAL	TS_CAL	ADC to PTAT conversion value at 3V, 30C
0x1FFF_FB20	~RCLP_TRIM	RCLP_TRIM	RCLP adjustment value
0x1FFF_FB38	~RCHF24TRIM	RCHF24TRIM	RCHF 24MHz tuning value
0x1FFF_FB3C	~RCHF16TRIM	RCHF16TRIM	RCHF 16MHz tuning value
0x1FFF_FB40	~RCHF8TRIM	RCHF8TRIM	RCHF 8MHz tuning value (auto-load)
0x1FFF_FB44	~RCLFTRIM	RCLFTRIM	RCLF tuning value

Table 4-3 Flash LDT0 Sector

In order to ensure the reliability of the data, the parameters in LDT0 are stored in a way that the high and low halfwords are positive and negative codes for each other. When the software uses these parameters, the positive and negative code should be checked first, and it can be used if the result is correct, otherwise the default parameters should be kept.

The detailed parameter format is defined as follows:

Symbol	Bit[31:16]	Bit[15:0]	Description

Symbol	Bit[31:16]	Bit[15:0]	Description
VREFCAL	{4'h0, ~VREFCAL}	{4'hF, VREFCAL}	VDDA=VREFP=3V+/-10 mV, ADC conversion value to VREF1p2 under 30C+/-1C
VREFRAW			The actual measured value of VREF1p2 output voltage, data format TBD
RCLP_TRIM	{8'h00, ~trim}	{8'hFF, trim}	trim[7:0] means 8bit trim value
RCHF24TRIM RCHF16TRIM RCHF8TRIM	{8'b0000_0000, ~RCHFtrim[7:0]}	{8'b1111_1111,RCHFtrim[7:0]}	RCHFtrim[7:0] means 8bit adjustment value
VREFREG45_TRIM	{8'b0000_0000, ~VREFREGU_trim}	{8'b1111_1111, VREFREGU_trim}	VREFP_VREG tuning value 4.5V
VREFREG30_TRIM	{8'b0000_0000, ~VREFREGU_trim}	{8'b1111_1111, VREFREGU_trim}	VREFP_VREG tuning value 3.0V
VREFREG25_TRIM	{8'b0000_0000, ~VREFREGU_trim}	{8'b1111_1111, VREFREGU_trim}	VREFP_VREG tuning value 2.5V
VREFREG20_TRIM	{8'b0000_0000, ~VREFREGU_trim}	{8'b1111_1111, VREFREGU_trim}	VREFP_VREG tuning value 2.0V
VREFREG15_TRIM	{8'b0000_0000, ~VREFREGU_trim}	{8'b1111_1111, VREFREGU_trim}	VREFP_VREG tuning value 1.5V
ULPBG_TRIM	{11'b0000_0000_000, ~ULPBG_trim}	{11'b1111_1111_111, ULPBG_trim}	ULPBG_VDD tuning value 1.2V

Table 4-4 LDT0 Data Format

4.4.2.2 LDT1 Page

LDT1 is the user configuration information area, which is mainly used to save user option bytes and Flash lock information. LDT1 can only be rewritten using SWD, that is, rewritten by the user through the programmer.

The bus address of LDT1 is 0x1FFF_FC00~0x1FFF_FDFF, LDT1 can be erased only after performing flash full erase.

AHB address	Bit[31:16]	Bit[15:0]	Description

0x1FFF_FC00	~OPTBYTES[15:0]	OPTBYTES[15:0]	Low halfword of user option byte
0x1FFF_FC04	~OPTBYTES[31:16]	OPTBYTES[31:16]	High half word of user option byte
0x1FFF_FC08	LOCK1		ACLOCK configuration word, control low 16 blocks
0x1FFF_FC0C	LOCK2		ACLOCK configuration word, control high 16 blocks

Table 4-5 Flash LDT1 Format

The OPTBYTES option byte is defined as follows:

Bit field	Name	Functional description	Default value
31:24	BTSWPEN	Boot address swapping enable 0x55: Boot swap function allowed Others: Boot swap function forbidden	0xFF
23:20	IWDTSLP	Configure whether IWDT is allowed to stop counting in low-power mode 0xA: Allow to use stopping IWDT counting in Sleep/DeepSleep/RTCBKP mode Others: Forbidden to use stopping IWDT in any mode	0xA
19:16	DFLSEN	Data flash enable 0x5: Enable data flash, the highest 16KB address of main array is defined as data flash Others: Disable data flash	0xF
15:8	ACLKEN	Application code protection enable 0x33: Disable ACLOCK Others: Enable ACLOCK	0x33
7:0	DBRDPEN	Debug access protection enable 0xAA: No debug protection Others: Enable debug protection	0xAA

Table 4-6 Flash Option Byte Definition

Note: OPTBYTES can be written by user software or SWD interface for a fresh device. But once ACLKEN or DBRDP is enabled, the user must erase all flash (matrix erase) with SWD before rewriting OPTBYTES.

LOCK information is used to prepare Flash content protection, and access rights are protected in units of 8KB blocks. See the "Flash Content Protection" chapter for details.

LOCK configuration byte definition:

Bit field	Name	Functional description	Default value
31:0	LOCK1	Block Lockword 1, every 2itcorresponds to one 8KB Block 11: Unprotected 01,10: Software read-write prohibited, only fetching allowed 00: Software read-write prohibited, only fetching allowed; SWD read-write prohibited LOCK1[1:0]corresponds to Block0(Flash minimum address 8KB space), LOCK1[31:30]corresponds to Block15 (Flash address space 120~128KB), and so on	0xFFFFFFFF
31:0	LOCK2	Block Lockword 2, every 2it corresponds to one 8KB Block 11: Unprotected 01,10: Software read-write prohibited, only fetching allowed 00: Software read-write prohibited, only fetching allowed; SWD read-write prohibited LOCK2[1:0]corresponds to Block16 (Flash address space 128~136KB), LOCK2[31:30] corresponds to Block31 (Flash address space 248~256KB), and so on	0xFFFFFFFF

Table 4-7 Flash LOCK Config

The address mapping for LOCK bits is shown in the following table:

Address	LOCK bits
0x0000_0000 ~ 0x0000_1FFF	LOCK1[1:0]
0x0000_2000 ~ 0x0000_3FFF	LOCK1[3:2]
0x0000_4000 ~ 0x0000_5FFF	LOCK1[5:4]
0x0000_6000 ~ 0x0000_7FFF	LOCK1[7:6]
0x0000_8000 ~ 0x0000_9FFF	LOCK1[9:8]
0x0000_A000 ~ 0x0000_BFFF	LOCK1[11:10]
0x0000_C000 ~ 0x0000_DFFF	LOCK1[13:12]
0x0000_E000 ~ 0x0000_FFFF	LOCK1[15:14]
0x0001_0000 ~ 0x0001_1FFF	LOCK1[17:16]
0x0001_2000 ~ 0x0001_3FFF	LOCK1[19:18]
0x0001_4000 ~ 0x0001_5FFF	LOCK1[21:20]
0x0001_6000 ~ 0x0001_7FFF	LOCK1[23:22]
0x0001_8000 ~ 0x0001_9FFF	LOCK1[25:24]

Address	LOCK bits
0x0001_A000 ~ 0x0001_BFFF	LOCK1[27:26]
0x0001_C000 ~ 0x0001_DFFF	LOCK1[29:28]
0x0001_E000 ~ 0x0001_FFFF	LOCK1[31:30]
0x0002_0000 ~ 0x0002_1FFF	LOCK2[1:0]
0x0002_2000 ~ 0x0002_3FFF	LOCK2[3:2]
0x0002_4000 ~ 0x0002_5FFF	LOCK2[5:4]
0x0002_6000 ~ 0x0002_7FFF	LOCK2[7:6]
0x0002_8000 ~ 0x0002_9FFF	LOCK2[9:8]
0x0002_A000 ~ 0x0002_BFFF	LOCK2[11:10]
0x0002_C000 ~ 0x0002_DFFF	LOCK2[13:12]
0x0002_E000 ~ 0x0002_FFFF	LOCK2[15:14]
0x0003_0000 ~ 0x0003_1FFF	LOCK2[17:16]
0x0003_2000 ~ 0x0003_3FFF	LOCK2[19:18]
0x0003_4000 ~ 0x0003_5FFF	LOCK2[21:20]
0x0003_6000 ~ 0x0003_7FFF	LOCK2[23:22]
0x0003_8000 ~ 0x0003_9FFF	LOCK2[25:24]
0x0003_A000 ~ 0x0003_BFFF	LOCK2[27:26]
0x0003_C000 ~ 0x0003_DFFF	LOCK2[29:28]
0x0003_E000 ~ 0x0003_FFFF	LOCK2[31:30]

Table 4-8 Lock Bit and Flash Address Corresponding Table

4.4.2.3 Information3Page

Flash also contains four information pages, in which Information3 is used to control the BootSwap function. The Information3 page address is 0x0004_0600~0x0004_07FF.

With BOOTSWAPEN=0x55 in LDT1, BootSwap can be controlled by the INFO3 lowest address content. When the data is 0x5454_ABAB, the chip swaps the logical address of the lowest two 8KB spaces in Flash (note that ACLOCK is only applied to logical address, not the physical address), thus enabling risk-free upgrading of the boot code.

The schematic diagram of BootSwap is as follows:

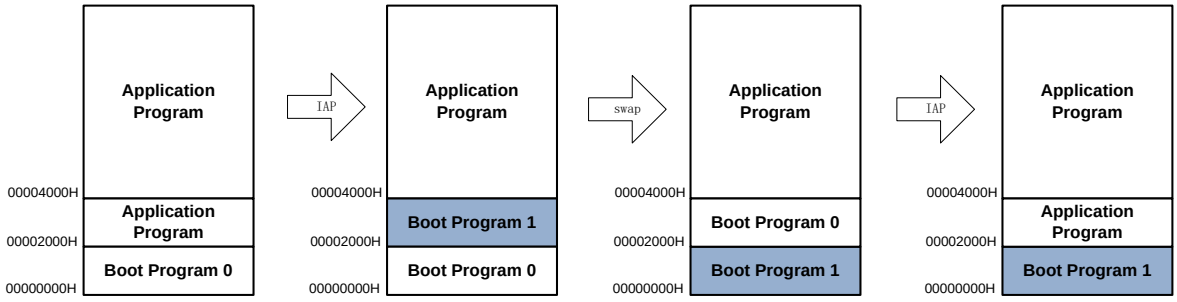


Figure 4–5 BootSwap Schematic Diagram

Its main purpose is to prevent the occurrence of unexpected interruption (power failure, abnormal reset, etc.) when the system is updating the boot code. If the original boot code has been erased at this time, it will lead to the failure of normal operation after the chip is reset.

When the BootSwap function is enabled, it is assumed that the boot code occupies a total space of 8KBfrom 0000 to 1FFF. When the system is being upgraded, the new boot code should be written into the address of 0x2000~3FFF first, and then swap boot code area.

There are several possible conditions:

- Power off occurs when erasing 2000~3FFF address. Because the original boot code is still present, system will boot using original code after reset
- The chip successfully writes boot Program1, then enables BootSwap and perform a soft reset. System will boot using new boot code after reset
- The chip loses power when erasing boot Program0. Since boot Program1 has already been written, system will boot using new boot code after reset

The recommended upgrading procedure is as follows:

- Update the application program
- To upgrade boot code, write the new boot program into the second 8KB space
- Configure INFO3 to enable BootSwap
- Perform a soft reset and boot from the new Boot program
- Rewrite the second 8KB space to the new application

The remapping of the logical address to the Flash physical address is automatically done by chip, and both software and Debugger can only access to the logical address.

Note: *The BootSwap function of IF3 actually only uses the lowest word on this page, and the rest of the address space is open for reading and writing (no specific functions), and software or Debugger can write data at will.*

4.4.2.4 Information1~2 Page (Debugger Only, Lockable)

These two information pages are open to users. It can only be modified by SWD, and read-only to software.

The bus address is 0x0004_0200~0x0004_05FF, low address is IF1, high address is IF2, with 1KB in total.

If SWD rewrites the highest address byte to 0x55, the current sector will be prohibited from

programming after chip reset. SWD can erase the page and reprogram it.

SWD and software can always read these pages.

4.4.2.5 Information0 Page (OTP)

IF0 is an OTP page that can only be programmed once and cannot be erased or modified after programming. The bus address of IF0 is 0x0004_0000~0x0004_01FF, 512 bytes in total.

There is on restriction on reading IF0 pages.

4.4.3 Command Prefetch

The highest frequency of FM33LG0 is 64MHz. When the system frequency is higher than 24Mhz, you need to insert wait when accessing flash. Wait cycle has a certain impact on the efficiency of program execution. In order to improve the operating efficiency of the CPU at high frequencies, FM33LG0 implements instruction prefetch instructions to reduce the impact of flash waiting at high frequencies.

The instruction prefetch function is enabled by setting the PFTBUF_EN and PFTPHS_EN registers.

4.4.4 Flash Program

4.4.4.1 Introduction

FM33LG0XXsupports the following Flash programming methods:

- In system programming (ISP): Chip programming via FMSH dedicated programmer or online simulation, using SWD interface
- In application programming (IAP): Bootloader code can perform self-programming, the user can define any communication interface to realize online upgrade

The Flash must be erased before programming, and repeated programming of flash addresses that have not been erased is prohibited. Flash supports three erasing operations: full erasing, sector erasing, and page erasing.

4.4.4.2 Flash Erase Clock

RCHF clock is used to perform Flash erase/program, while the system clock can be any clock. The supported RCHF frequencies of programming are 8M, 16M and 24M.

4.4.4.3 Flash Erase Method

FM33LG0XX supports Flash erase operations, as well as single and continuous programming.

Flash must perform Key verification before erase/program. The key register must be written with

correct values, as well as in correct order, right before flash erase/perform operation. Otherwise, an error interrupt will be issued and flash erase/program will not be performed by hardware. When there is a Flash Key authentication error, Flash erase/program is prohibited until next system reset. Writing any value to the KEY register after a normal erase/program operation causes the state machine to return to its original write-protected state. The state transition is as follows:

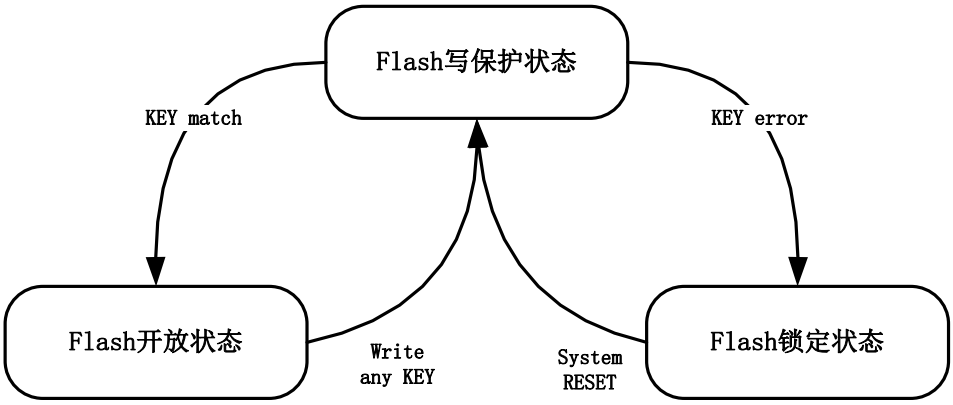


Figure 4–6 Flash Erasing Key Authentication

The software can confirm the current Key input status by querying FLS_ISR.KEYSTA. For details, refer to the register description.

4.4.4.4 Matrix Erase

Matrix erase operation can only be initiated by SWD interface. The matrix erase operation only erases the main array, and does not erase the special information sector. SWD can initiate matrix erase in manufacturer or user mode, the operation process is as follows:

- Write 10 to ERTYPE register by SWD
- Clear PREQ register by SWD, and then set the EREQ register
- Write the Flash matrix erase Key: 0x9696_9696 and 0x7D7D_7D7D by SWD
- SWD writes erase request 0x1234_ABCD to any Flash address
- Chip starts the matrix erase to Flash and suspends any Master access to Flash
- Interrupt flag and matrix erase flag are set after the matrix erase is finished (The matrix erase flag means that the main array is all erased, and any programming to the main array will clear this flag)
- After confirming the end of the erasing, the software writes any value to FLS_KEY register to restore write-protection

Note: *The matrix erase operation can only erase the main array of Flash, and will not affect the special information sector.*

4.4.4.5 Sector Erase

Both SWD and application code can perform sector erase. The procedure is as follows:

- Write 01 to ERTYPE register
- Clear the PREQ register and set the EREQ register
- Write the Flash block erase Key: 0x9696_9696 and 0x3C3C_3C3C
- Write erase request 0x1234_ABCD to any address in the sector that needs to be erased
- Chip checks whether the target sector is locked by ACLOCK, starts erasing the target sector if there is no lock, and triggers an error flag if there is a lock
- After sector erasing is complete, set the interrupt flag
- The software writes any value to FlashKEY register to restore write-protection

4.4.4.6 Page Erase

Both SWD and application code can perform page erase. The procedure is as follows:

- Write 00 or 11 to ERTYPE register
- Clear the PREQ register and set the EREQ register
- Write the Flash block erase Key: 0x9696_9696 and 0xEAEA_EAEA
- Write erase request 0x1234_ABCD to any address in the page that needs to be erased
- Chip checks whether the target sector is locked by ACLOCK, starts erasing the target sector if there is no lock, and triggers an error flag if there is a lock
- After sector erasing is complete, set the interrupt flag
- The software writes any value to FLS_KEY register to restore write-protection

4.4.4.7 Single Programming

The single programming is initiated by software and write operation is performed directly to Flash through the bus. The smallest unit of programming is 32bit. The procedure is as follows:

- Clear the EREQ register and set the PREQ register
- Clear the multiple word programming enable register
- Write the Flash programming Key: 0xA5A5_A5A5 and 0xF1F1_F1F1
- Write data to the Flash target address, an error flag will be set by hardware if the target address is locked by ACLOCK, and programming will be performed if there is no lock

- The interrupt flag is set after the programming is completed
- The software writes any value to FLS_KEY register to restore write protection

4.4.4.8 Multiple Programming

Multiple words programming can program half-sector (256 bytes) to Flash at one time over DMA Memory channels. During multi-word programming the DMA reads data from SRAM, and Flash target address must be aligned to half-sector, which means lowest 6bits of Flash address must be 0. In this way, fixed length of data could be programmed into Flash continuously and efficiently.

During the continuous programming, Flash interface is fully occupied by DMA, so any access from CPU will be halted. The procedure of continuous programming is as follows:

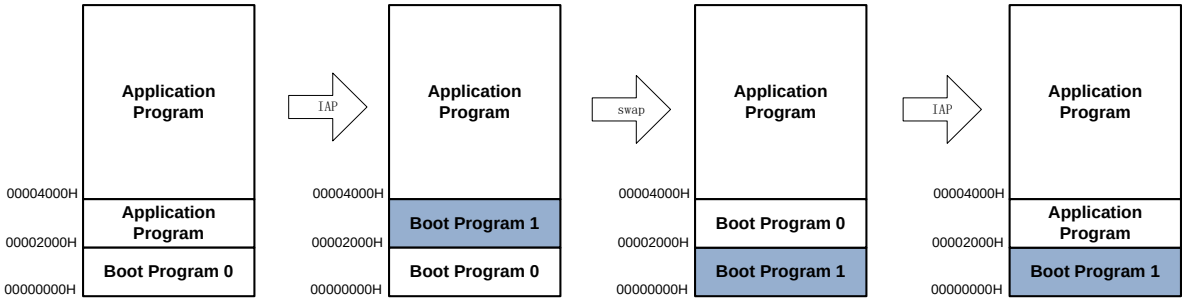
- Clear the EREQ register and set the PREQ register
- Set the multiple word programming enable register (DMA mode enable)
- Write 256 bytes data to RAM
- Configure DMA memory channels, set the transfer direction, read address, and write address
- Enable DMA memory channels
- Write the Flash programming Key: 0xA5A5_A5A5 and 0xF1F1_F1F1
- Software triggers DMA memory channels, which will read RAM 64 times in a row and program Flash
- Chip will check whether the programmed sector is locked by ACLOCK, and if locked an error interrupt will be triggered and DMA will stop programming
- An interrupt is triggered when 256-byte programming is completed, then Flash interface is released
- The software writes any value to FLS_KEY register to restore write protection

Note: If Flash erase/program is initiated while CPU is executing from Flash, CPU fetching will be halted until erase/program cycle is finished. If the CPU is executing from RAM, Flash erase/program will not halt CPU execution. During Flash erase/program, if user still wants to respond to interrupts in time, it is recommended to remap vector table into RAM.

4.4.4.9 BootSwap

Main purpose of BootSwap is to prevent the occurrence of unexpected interruption (power failure, abnormal reset, etc.) when the system is updating the boot code. If the original boot code has been erased at this time, it will lead to the failure of normal operation after the chip is reset.

BootSwap diagram is shown as follows:

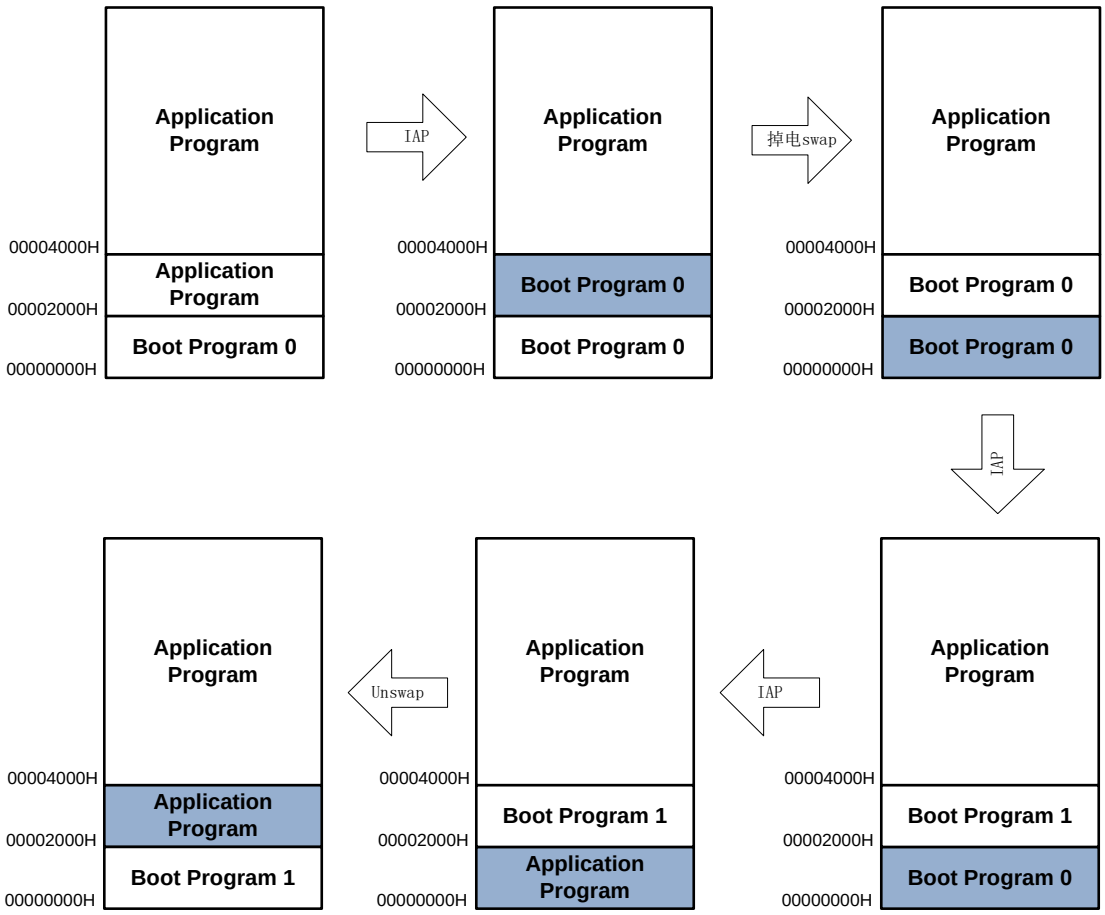


When the BootSwap function is enabled, it is assumed that the boot code occupies a total space of 8KB from 0000 to 1FFF. When the system is being upgraded, the new boot code should be written into the address of 0x2000~3FFF first, and then swap boot code area.

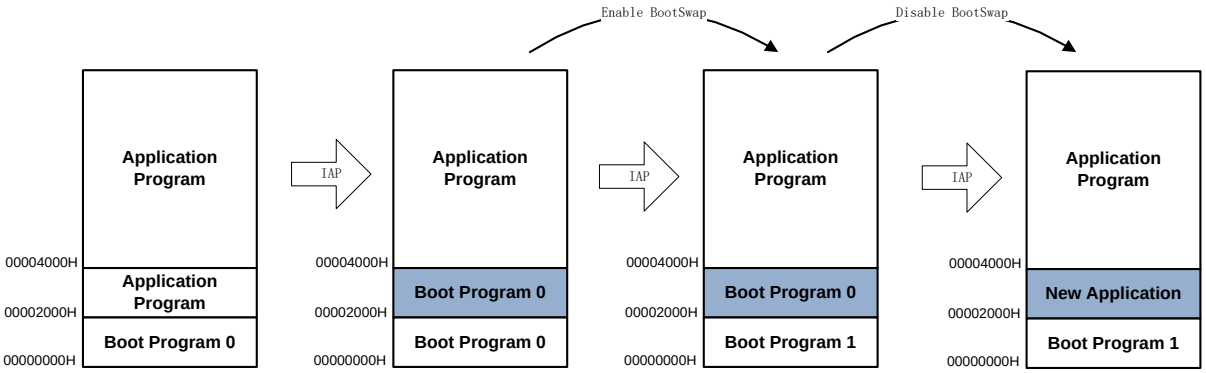
There are several possible conditions:

- Power off occurs when writing 2000~3FFF address. Because the original boot code is still present, system will boot using original code after reset
- The chip successfully writes boot Program1, then enables BootSwap and perform a soft reset. System will boot using new boot code after reset
- The chip loses power when erasing boot Program0. Since boot Program1 has already been written, system will boot using new boot code after reset

Another BootSwap application is shown in the figure below. In order to ensure reliable update of boot code, the 2nd 8KB physical space is used as a backup of the original Boot program. If an abnormal power loss occurs during programming, BootSwap will be triggered:



If there is no abnormal power loss during the Boot program update, soft reset can be skipped, and no swap is needed. Instead, BootSwap can be enabled before the original Boot program is modified, and BootSwap can be disabled after successful update:



The recommended boot code updating procedure is as follows:

- Update application program
- Write the new boot program to a second 8KB space
- Configure the information block, enable BootSwap
- Perform a soft reset, execute the new boot program after reboot
- Rewrite the second 8KB space as a new application

Register flag (FLSIF.BTSF) is used to indicate whether the current Boot area is physically located in 1st 8KB or 2nd 8KB. Application code can use this flag to determine the current boot status.

4.4.5 Data Flash

After the user configures OPTBYTES to enable DFLSEN, FM33LG0 will open 16KB data flash to users for data storage. After the data flash is enabled, the flash capacity of different models of products are divided as shown in the following table:

Model	Data flash size	Data flash address	Program flash size
FM33LG04x	16KB	0x0003_C000~0x0003_FFFF	240KB
FM33LG02x	16KB	0x0001_C000~0x0001_FFFF	112KB
FM33LG01x	N/A	N/A	64KB

Table 4-9 Dataflash Config

Note: *64KB flash version does not support data flash*

When the data flash is enabled, the corresponding program flash space will be reduced by 16KB. Data flash is always located at the highest 16KB of the flash logical address space.

Data flash and program flash have no difference in access rights, and are also controlled by DBRDP and ACLOCK. But when the chip performs a matrix erase operation, the data flash will not be erased. When the data flash is not enabled, all data in the main array will be erased when the chip executes a matrix erase.

Note: *With data flash enabled, the chip's matrix erasing time is significantly increased. For the 256KB capacity model, the matrix erasing time is increased from 8ms to 240ms, and for the 128KB capacity model, the time is increased from 8msto 112ms.*

4.4.6 Flash Memory Protection

Flash memory protection can be used to protect user code, user data and user configuration information in Flash from being read and tampered with by an unauthorized third party.

Flash Protection includes two types: DBRDP-DeBug ReaD Protection and ACLOCK-Application Code Block Locking. Flash protection is controlled via OPTBYTES in LDT1.

4.4.6.1 Debug Interface Protection (DBRDP)

The primary purpose of DBRDP is to prevent unauthorized access to the Flash content through the Debug interface.

DBRDP is enabled or disabled by the DBRDPEN configuration word in LDT1 sector (0xAA means DBRDP is disabled, which is default state of a fresh device).When DBRDP is enabled, the Flash main array cannot be read or erased through the SWD interface, and the RAM cannot be accessed through the SWD interface.

Ways to exit DBRDP: After the matrix erase of the flash is completed through SWD, SWD can rewrite OPTBYTES to disable DBRDP at will, and then reset the chip. After the reset is completed, the chip will be in a non-debug protection state.

4.4.6.2 Application Code Lock (ACLOCK)

The main purpose of ACLOCK is to prevent any unauthorized reading or tempering to the application code in Flash from hacking code. With the ACLOCK function, you can set some part of Flash as fetch-only, any read-as-data or modifying are prohibited.

ACLOCK works in the granularity of 8KB. The whole Flash contains 32 Blocks with 2bit LOCK information for each Block. The default LOCK word is 0xFFFF_FFFF for a fresh device, which is unprotected. When the corresponding LOCK bits are set to 01 or 10, this Block can only be fetched by CPU. When the corresponding LOCK bits are 00, both CPU and SWD are prohibited to read or modify the Block. ACLOCK function is disabled by default. The user needs to enable ACLOCK through the programmer, and the user code should conform to the ACLOCK configuration when compiling (for example, literal pool cannot be compiled to the locked Block).

Functions of ACLOCK:

- No protection: All blocks allow CPU to fetch, read, and modify. No restriction on SWD access.
- Read-write protection: Specified blocks allow CPU to fetch only, read & modify by CPU and DMA is prohibited. No restriction on SWD access.
- Software and SWD protection: Specified blocks allow CPU to fetch only, read & modify by CPU, DMA and SWD is prohibited.

The relationship between LOCK bit and Block access permissions is shown in the following table:

LOCK bit	CPU read	CPU fetch	SWDread and eraseprogram
11	√	√	√
01/10	×	√	√
00	×	√	×

Table 4-10 LOCK Bit Permission Definition

ACLOCK information is loaded into registers after system reset. These registers can also be set by software, but cannot be cleared by software (it is only possible for software to escalate the protection level).

LOCK register contents are invalid when ACLOCK is not enabled.

Note: *ACLOCK is independent of DBRDP for each Block in Flash. For SWD interfaces, DBRDP has higher priority than ACLOCK, that is, after DBRDP is enabled, SWD cannot access Flash regardless of whether ACLOCK is enabled or not.*

It is forbidden to use ACLOCK to disable the read permission of 1st block. Since the MSP pointer must be read from address 0 after the CPU is reset, ACLOCK will cause the CPU to fail to start normally.

Exit ACLOCK: Full-space matrix erase must be performed by SWD. After matrix erasing, SWD can modify OPTBYTES to disable ACLOCK, and then reset the chip. After reset, ACLOCK is unactivated.

4.4.6.3 Flash Access Authorization Description

Flash space access authorization allocation:

Flash area	DBRDP	LOCK bits (per Block)[3]	Last byte in page	SWD	Application
Main array	ON	00	x	-	Block fetch only
		01/10	x	-	Block fetch only
		11	x	-	R/E/W/F
	OFF	00	x	Block cannot be accessed	Block fetch only
		01/10	x	R/E/W	Block fetch only
		11	x	R/E/W	R/E/W/F
LDT1	ON	x	x	R[2]	R
	OFF	x	x	R/E/W	R
IF3	x	x	x	R/E/W	R/E/W
IF2,1,0	x	x	55	R/E	R
			others	R/E/W	R

Table 4-11 Flash Access Authorization Table

Note:

[1] R: Read, E: Erase, W: Write, F: Fetch

[2] LDT1 erase can be performed after matrix erase

[3] ACLOCKEN is assumed to be valid in the above description. If ACLOCKEN is disabled, LOCK bits have no effect.

4.5 Register

Offset	Name	Symbol
FLS (Base address: 0x40001000)		
0x00000000	Flash Read Control Register	FLS_RDCR
0x00000004	Flash Prefetch Control Register	FLS_PFCR
0x00000008	Flash Option Bytes Register	FLS_OPTBR
0x0000000C	Flash Application Code Lock Register1	FLS_ACLOCK1
0x00000010	Flash Application Code Lock Register2	FLS_ACLOCK2
0x00000014	Flash Erase/Program Control Register	FLS_EPCR
0x00000018	Flash Key Register	FLS_KEY
0x0000001C	Flash Interrupt Enable Register	FLS_IER
0x00000020	Flash Interrupt Status Register	FLS_ISR

4.5.1 Flash Read Control Register (FLS_RDCR)

NAME	**FLS_RDCR**							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						WAIT	
access	U-0						R/W-00	

bit	name	functional description
31:2	-	RFU: **Reserved, read as 0**
1:0	WAIT	Flash Wait Cycles Config 00/11:0 wait cycle 01:1 wait cycle 10:2 wait cycles When the system frequency is less than or equal to 24MHz, there is no need to insert wait cycle. If the system frequency is greater than 24MHz and less than 48MHz, insert 1 wait; if the system frequency is greater than 48MHz, insert 2 waits.

4.5.2 Flash Prefetch Control Register (FLS_PFCR)

NAME	FLS_PFCR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							PRFTEN
access	U-0							R/W-0

bit	name	functional description
31:2	-	RFU: **Reserved, read as 0**
1	PFTBUF_EN	Prefetch Buffer Enable, in the case of WAIT==00, writing 1 is invalid 1: Enable prefetch buffer 0: Disable prefetch buffer
0	PFTPHS_EN	Prefetch Phase Enable, in the case of WAIT==00, writing 1 is invalid 1: Enable phase buffer 0: Disable phase buffer

4.5.3 Flash Option Bytes Register (FLS_OPTBR)

NAME	FLS_OPTBR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	IWDTSLP	-						
access	R-0	U-0						
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-					IF2LOCK	IF1LOCK	-
access	U-0					R-0	R-0	U-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-					DFLSEN	BTSEN	
access	U-0					R-0	R-01	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

name	-	ACLOCKEN	DBRDPEN
access	U-0	R-01	R-01

bit	name	functional description
31	IWDTSLP	IWDT Sleep Enable 1: Allows the application to suspend IWDT counting in Sleep mode 0: Prohibit the application from suspending IWDT counting in Sleep mode
30:19	-	RFU: **Reserved, read as 0**
18	IF2LOCK	Information2 Lock Flag (IF2 Lock Enable) 0: Unlock 1: Locked, software cannot modify this page
17	IF1LOCK	Information1Lock Flag (IF1 Lock Enable) 0: Unlock 1: Locked, software cannot modify this page
16:11	-	RFU: **Reserved, read as 0**
10	DFLSEN	Data Flash Enable 0: No data flash 1: Have data flash
9:8	BTSEN	BootSwap Enable 00/01/11: Disable BootSwap 10: Enable BootSwap
7:4	-	RFU: **Reserved, read as 0**
3:2	ACLOCKEN	App Code Lock Enable 00/01/11: ACLOCK disable 10: ACLOCK enable
1:0	DBRDPEN	Debug Port Read Protection Enable 00/01/11: DBRDP disable 10: DBRDP enable

4.5.4 Flash Application Code Lock Register1 (FLS_ACLOCK1)

NAME	FLS_ACLOCK1							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	LOCK1[31:24]							
access	R/W-1111 1111							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	LOCK1[23:16]							
access	R/W-1111 1111							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	LOCK1[15:8]							

access	R/W-1111 1111							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	LOCK1[7:0]							
access	R/W-1111 1111							

bit	name	functional description
31:0	LOCK1	ACLOCK lower 32 LOCK bits, which is used to control the Block15~Block0 respectively. Each Block is 8KB size, and each Block uses 2bits for access control.(Lock bits) 11: The current Block allows SWD and software to read and write 01/10: The current Block allows SWD to read and write, prohibits the software from reading and writing, and the software can fetch 00: The current Block prohibits SWD & software from reading and writing, and the software can only fetch Software can only write 0 to these bits, write 1 is ignored.

4.5.5 Flash Application Code Lock Register2 (FLS_ACLOCK2)

NAME	FLS_ACLOCK2							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	LOCK2[31:24]							
access	R/W-1111 1111							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	LOCK2[23:16]							
access	R/W-1111 1111							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	LOCK2[15:8]							
access	R/W-1111 1111							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	LOCK2[7:0]							
access	R/W-1111 1111							

bit	name	functional description
31:0	LOCK2	ACLOCK higher 32 LOCK bits, which is used to control the Block31~Block16 respectively. Each Block is 8KB size, and each Block uses 2bits for access control.(Lock Bits) 11: The current Block allows SWD and software to read and write 01/10: The current Block allows SWD to read and write,

bit	name	functional description
		prohibits the software from reading and writing, and the software can fetch 00: The current Block prohibits SWD & software from reading and writing, and the software can only fetch Software can only write 0 to these bits, write 1 is ignored.

4.5.6 Flash Erase/Program Control Register (FLS_EPCR)

NAME	FLS_EPCR							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						ERTYPE	
access	U-0						R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						PREQ	EREQ
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:10	-	RFU: **Reserved, read as 0**
9:8	ERTYPE	Flash Erase Type 00/11: Page Erase 01: Sector Erase 10: Chip Erase (SWD only)
7:2	-	RFU: **Reserved, read as 0**
1	PREQ	Program Request Set by software, cleared by hardware after programming is finished
0	EREQ	Erase Request Set by software, cleared by hardware after erasing is finished

Note: *When PREQ and EREQ are in the enabled state at the same time, the execution of FLASH erasing or writing will cause the CPU to hang all the time. This situation needs to be avoided.*

4.5.7 Flash Key Register (FLS_KEY)

NAME	FLS_KEY							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	KEY[31:24]							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	KEY[23:16]							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	KEY[15:8]							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	KEY[7:0]							
access	W-0000 0000							

bit	name	functional description
31:0	KEY	Flash Key Input Register Software or SWD must correctly write a valid Key sequence into this address to initiate erase/program

4.5.8 Flash Interrupt Enable Register (FLS_IER)

NAME	FLS_IER							
Offset	0x1C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				OTPIE	AUTHIE	KEYIE	CKIE
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						PRDIE	ERDIE
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:12	-	RFU: **Reserved, read as 0**
11	OTPIE	OTP Program Error Interrupt Enable, 1enable
10	AUTHIE	Flash Authentication Error Interrupt Enable, 1 enable

bit	name	functional description
9	KEYIE	Flash Key Error Interrupt Enable, 1 enable
8	CKIE	Erase/Program Clock Error Interrupt Enable, 1 enable
7:2	-	RFU: **Reserved, read as 0**
1	PRDIE	Program Done Interrupt Enable, 1 enable
0	ERDIE	Erase Done Interrupt Enable, 1 enable

4.5.9 Flash Interrupt Status Register (FLS_ISR)

NAME	FLS_ISR							
Offset	0x20							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				KEYSTA			BTSF
access	U-0				R-000			R-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				OTPERR	AUTHERR	KEYERR	CKERR
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						PRD	ERD
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:20	-	RFU: **Reserved, read as 0**
19:17	KEYSTA	Flash Key Input Status 000: Flash write-protect status, no KEY inputted 001: Matrix erase unlocked state 010: Page erase unlocked state 011: Programming unlocked state 100: Locked state caused by KEY error. Reset is required to unlock. 101: Sector erase unlocked state 110/111: RFU
16	BTSF	BootSwap Flag Register 0: The boot area is Flash physical address 0000H~1FFFH 1: The boot area is Flash physical address 2000H~3FFFH
15:12	-	RFU: **Reserved, read as 0**
11	OTPERR	OTP Program Error Flag. Write 1 to clear. 1: Try to program the OTP bytes that have been programmed 0: No OTP programming error

bit	name	functional description
10	AUTHERR	Flash Authentication Error Flag. Set when reading or erasing a LOCK block, write 1 to clear. 1: Flash access error 0: No Flash access error
9	KEYERR	Flash Key Error Flag, write 1 to clear
8	CKERR	Erase/Program Clock Error Flag, CKERR interrupts are triggered if RCHF is not enabled while writing Flash with NVMIF, write 1 to clear
7:2	-	RFU: **Reserved, read as 0**
1	PRD	Program Done Flag, hardware set, write 1 to clear
0	ERD	Erase Done Flag, hardware set, write 1 to clear

5 Power Management Unit (PMU)

5.1 Power Supply

5.1.1 Power Domains

- VDD(VDDA)

The typical operating voltage range of the chip's main power supply (VDD) is 1.65~5.5V.

When the chip is powered on, the reset release threshold is mainly determined by the BOR circuit, and its typical reset release voltage is 1.65V.

When the chip is powered off, if BOR is enabled, the power-off reset threshold is determined by the BOR circuit. The software can configure BORCR.BOR_PDRCFG to obtain 4 threshold levels, the default value is 1.8V. If BOR is not enabled and PDR is enabled, the power-off reset threshold is determined by the PDR circuit. The software can configure 4 gears through PDRCR.CFG, the default value is 1.4V.

In summary, the actual operating voltage range of the chip's VDD will be determined by the BOR and PDR circuit configuration.

Note: *BOR and PDR must not be turned off at the same time under any circumstance, as a normal reset may not occur when the chip powers down and the chip may not work properly when it is powered up again.*

- VREFP

Only a few packages have independent VREFP pins. VREFP is the reference voltage input for ADC and DAC. When ADC is working, it will draw tens to hundreds of uA current from VREFP pin. In most packages, VREFP is packaged with VDDA.

- VBAT

Some models support the VBAT power supply pin, which can automatically switch the RTC circuit to the VBAT power supply when the VDD main power supply loses power. The VBAT pin should be connected to a suitable backup power source, usually a backup battery or super capacitor. If the backup power supply is not used, VBAT and VDD should be short-circuited on the system.

Note that when the power switching function is used, the working range of VBAT is 1.5~4.2V. When the power switching function is not used, VBAT should be short-circuited with VDD. At this time, VBAT can withstand 5.5V power supply.

- VDD15

VDD15 is the core power supply of the chip, and a 1.5V power output is generated by a linear power regulator. All digital circuits, Flash, SRAM and some analog circuits work under this power supply. The VDD15 pin requires an external 0.1~1uF regulator capacitor. When the main power supply VDD drops below 1.5V, the voltage regulator output will follow VDD changes.

5.1.2 Power Supply Structure

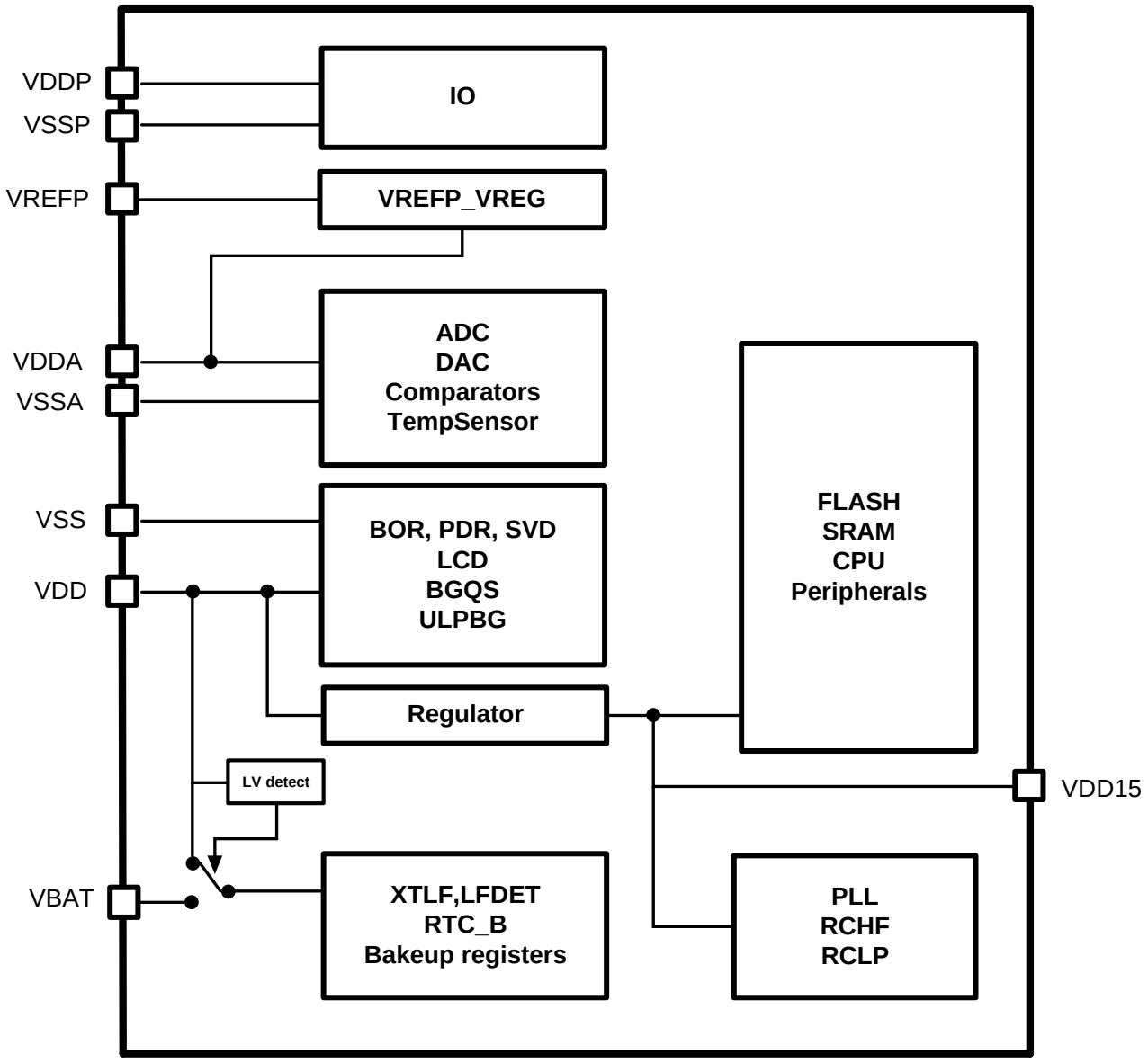


Figure 5–1 Power Structure

5.1.3 ADC and Independent Power Supply of Internal Reference

The power ground used by ADC is VDDA and VSSP, and its reference voltage is VREFP. In some packaging forms, VREFP will be separately led to the chip pins, where VREFP can be provided by the system, can also be generated on-chip, or directly short-circuited with VDDA on the PCB.

When VREFP and VDDA are independent, its input reference can be different from VDDA. The allowable input range is:

1.6V ≤ VREFP ≤ VDDA

5.1.4 On-chip Fast Reference Source (AVREF)

The AVREF voltage is 1.0V, and the startup speed is very fast. After enabling, it only takes about 3us to establish a 1V output voltage, and the typical power consumption is less than 3uA. When the chip sleeps, AVREF is automatically turned off to save power.

5.1.5 On-chip High-precision Reference Source (VREF1p2)

FM33LGx0 integrates a high-precision reference source, the typical output voltage is about 1.2V, and it can work stably in the range of 1.6V≤VDDA≤5.5V. After this reference voltage is output from the Buffer, it can be sampled by the ADC and also used as the reference voltage input of the comparator. After this reference voltage is boosted and driven, internal references of 2.0V, 2.5V, 3.0V, and 4.5V can be obtained (not higher than the power supply voltage), which can be used as a regulated reference source for ADCs, DACs and comparators.

In the entire operating temperature range, the typical temperature coefficient of this reference source is less than 25ppm/°C, and a built-in temperature sensor output is used for ADC sampling and measurement of the current chip substrate temperature.

The software can turn on or turn off this reference source. After turning on the reference source, the VREF1p2 output setup time is less than 1ms, and the typical power consumption is about 1.5uA. When the temperature sensor is turned on, the power consumption of VREF1p2 is less than 2uA.

After the software enables VREF1p2, after waiting for enough time to ensure that the VREF output is fully established inside the chip, the VREF_DRY status flag register is set, and the VREF_IF interrupt flag is set. The software can confirm the effective establishment of VREF1p2 by timing or according to the VREF_RDY register.

When the software closes VREF1p2, the VREF_RDY register is automatically cleared, and VREF_IF is cleared by software writing 1.

The maximum temperature measurement range supported by the temperature sensor is -55~125°C. The output voltage of the temperature sensor changes with temperature as a straight line with a positive temperature coefficient, with a typical slope of 5.1mV/°C. Before the chip leaves the factory, the temperature sensor will be calibrated under the condition of 30°C +/- 1°C. Under this condition, the temperature measurement error in the range of -40~+85°C is within +/-2°C .

5.1.6 VREFP Generation (VREFP_VREG)

See "VREFP Reference Voltage" chapter.

5.2 Consumption Mode

5.2.1 Introduction

After power-on reset, the chip runs in ACTIVE mode by default. The CPU fetches instructions from Flash, and all peripherals can work normally. The chip supports a variety of low power modes, and the software can choose the appropriate low power mode in the appropriate scenario to balance the power consumption, performance, wake-up time and wake-up conditions.

Supported power modes:

- ACTIVE mode
- LP Active: LDO enters low-power mode, CPU frequency does not exceed 4MHz, all peripherals can work
- LP Run mode: LDO operate in ultra-low power mode, and CPU and peripherals can only operate at low-frequency
- SLEEP mode: CPU halts, Flash halts, LDO works in low-power mode, only some of the peripherals can work
- DEEPSLEEP mode: CPU halts, Flash halts, internal reference disabled, LDO works in low-power mode, only some of the peripherals can work
- VBAT mode: VDD is powered down, VBAT is powered by the backup power supply

In addition, the consumptions can be reduced by the several methods below in ACTIVE mode.

- Lowering system clock frequency
- Turn off the bus clock and operating clock for unused peripherals

Mode	Typical consumptions	Wakeup conditions	Chip status	Typical wakeup time[1]
ACTIVE	150uA/MHz		Working normally	-
LP Active	500uA@4Mhz	Software quit	LDO enters low-power mode Disable XTHF, PLL CPU operating frequency is not higher than 600Khz	-
LP Run	30uA@32KHz	Software quit	Low-speed work	-
SLEEP	6uA	SVD interrupt	CPU sleep Disable RCHF, PLL, XTHF,	3us

		COMP interrupt RTC interrupt IO interrupt WKUPx interrupt 32K crystal oscillator fail interrupt Watch dog reset NRST pin reset	etc. Keep BGQS enable, disable LDO15, RTC timekeeping VREF1p22is determined by the software configuration whether to turn on, if it is turned on, the power consumption will increase by 1.5uA	
DEEPSLEEP	1.5uA		CPU sleep[2] Disable RCHF, PLL, XTHF, etc. Disable BG_QS and LDO15, RTC timekeeping VREF1p22is determined by the software configuration whether to turn on, if it is turned on, the power consumption will increase by 1.5uA	5us
VBAT	0.8uA	VDD power up again	VDD power down, only VBAT power supply	-

Table 5-1 Consumption Mode Table

Note:

[1] Typical wake-up time means the time between the arrival of the wake-up event and the CPU's execution of the wake-up interrupt service routine.

[2] Refer to the ARMv6-M Architecture Reference manual for CPU sleep-entry.

[3] When the CPU tries to enter the low-power mode, if Flash is under erasing/programming, the chip will automatically wait for Flash to finish erasing/programming before entering the low-power mode.

5.2.2 Power Mode and System Frequency

In different power mode, the range of system frequency is shown as below:

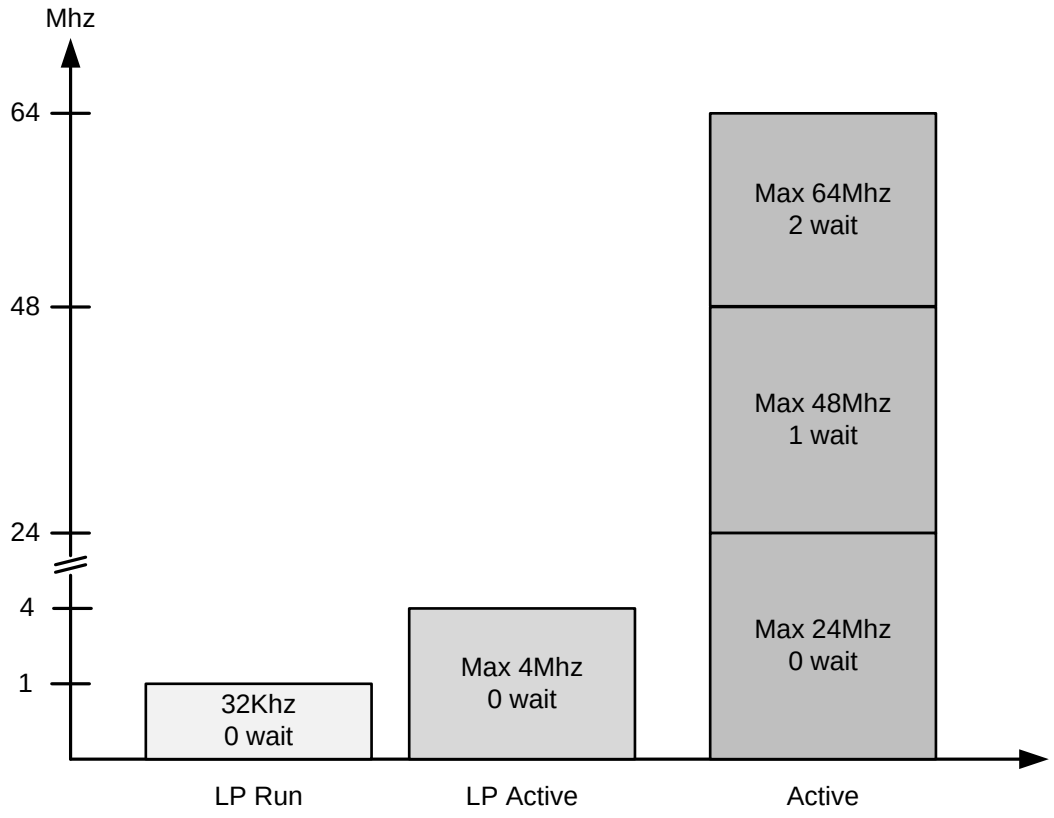


Figure 5–2 Consumption Mode and System Clock

The acceptable system frequency and available clock sources in different power modes are shown in the table below. Application software should strictly follow the rules of this table. Using high system frequency in low-power mode may cause the system to fail to function properly.

Mode	CPU frequency	Available clock source	Flash wait	Peripheral clock
ACTIVE	<=24Mhz	All	0	All
	>24Mhz, <=48Mhz		1	
	>48Mhz, <=64Mhz		2	
LP Active	4Mhz	RCHF, RCLF, XTLF, RCLP	0	RCHF, RCLF, XTLF, RCLP
LP Run	32Khz	RCLF, XTLF, RCLP	0	RCLF, XTLF, RCLP

Table 5-2 Consumption Mode and Available System Clock

5.2.3 Active Mode

The chip is in normal working mode. The chip will enter Active mode after power reset. The default CPU frequency is 8MHz and it can run up to 64MHz. All digital and analog peripheral can run at full speed in Active mode.

When the system frequency is higher than 24MHz, wait cycle must be inserted when accessing Flash, and Flash Prefetch is recommended to be enabled by the software to improve the efficiency of instruction execution.

5.2.4 LP Active Mode

Software can enter LP Active mode by configuring the LPMCFG.LDO_LPM register. At this time, LDO is placed in a low-power mode, while its own power consumption is reduced, the driving capability is also reduced. Therefore, in LP Active mode, it is recommended that the CPU frequency should not exceed 4MHz. At the same time, the peripheral modules can still work with RCHF, RCMF, XTLF, RCLP, but the PLL and XTHF are forcibly disabled by the hardware and cannot be used.

The typical application scenario of LP Active mode is to keep 1~2 peripherals (such as UART, Timer) running normally for a long time when the CPU is in standby or running at low speed in a scenario that does not require high CPU processing capabilities, so as to provide the best energy efficiency ratio for some special low-power scenario.

Entering LP Active mode

- Set the system clock to 4MHz or lower
- Ensure that no peripherals are using XTHF or PLL clock
- Configure the LDO_LPM register

Hardware behavior in LP Active mode

After entering LP Active, the hardware automatically disables XTHF and PLL, and then LDO enters the low-power mode. All digital and analog peripheral can work.

Exiting LP Active mode

- Software clears the LDO_LPM register
- Waiting for a few NO Pin instructions
- Set the system clock as needed and resume normal Active mode operation

In LP Active mode, the chip can directly enter the LPRUN / SLEEP/DEEPSLEEP mode by rewriting the PMOD register by CPU.

5.2.5 LP Run Mode

When the chip needs low-power and low-speed operation, it can enter the LP RUN mode. At this time, the LDO enters the low-power mode, and the core uses XTLF, RCLP or RCLF to run, with a typical

frequency of 32KHz. When high-speed operation is required, the software can actively exit LP RUN and enter ACTIVE mode, and then switch the system clock to a higher frequency.

Entering LPRUN mode

The steps of entering LPRUN mode:

- The software sets the system clock (SYSCLK) to the XTLF or RCLP
- Configure the PMOD register as 01
- If the system clock configuration does not meet the register conditions above, error interrupt is flagged and access to the LPRUN is forbidden

Hardware behavior in LP RUN mode

After entering LP Run, the hardware automatically turns off RCHF, XTHF, PLL, TRNG, and then turns off LDO15. SVD, COMP, ADC, DAC, OPA can still work in LPRUN mode (VDDA power supply). Since the high-speed clocks are all turned off, the highest ADC working clock is only RCLF, which is equivalent to the fastest sampling rate of 38Ksps.

If the software executes WFI/WFE instructions in LPRUN mode, the CPU and Flash will stop activity, but the peripherals can still continue to work.

Exiting LPRUN mode

Follow these steps to exit the LPRUN mode:

- The software sets the PMOD register to 00
- Software enables RCHF, XTHF or PLL as needed
- Configure the system clock to be RCHF, XTHF or PLL after waiting for the clock start-up time

The CPU modifies the PMOD register in LPRUN mode to return to ACTIVE, or to enter SLEEP / DEEPSLEEP mode. If ACTIVE is returned, the hardware automatically puts the LDO in normal mode and unlocks the high-speed clock module.

5.2.6 SLEEP Mode

By entering SLEEP mode, you can significantly reduce the power consumption of the chip and keep in a state waiting for an event to wakeup.

Entering SLEEP mode

The software enters SLEEP mode according to the following steps:

- Configure the PMOD register to 10
- Execute WFI or WFE instructions

Hardware behavior in SLEEP mode

After entering SLEEP mode, the chip turns off the CPU clock, Flash enters STOP mode, the hardware automatically turns off RCHF, PLL, XTHF, TRNG. SVD, OPA, ADC, DAC, and COMP can still work in SLEEP mode. Among them, because the high-speed clocks are all turned off, the highest ADC working clock is only RCLF, which is equivalent to the fastest sampling rate of 38Ksps.

Digital peripheral modules can continue to work with low-speed clocks such as RCLF, XTLF, and RCLP.

Exiting SLEEP mode

According to the steps below to exit the SLEEP mode:

- A specific interrupt event occurs
- The system clock is automatically configured as RCHF
- When the CPU is awakened, it can enter or not enter the interrupt service routine, depending on the software configuration

5.2.7 DEEPSLEEP Mode

DEEPSLEEP is the lowest power consumption mode when the chip VDD is powered. In this mode, the internal reference source (AVREF) is turned off, so the sleep power consumption is further reduced than SLEEP.

Entering DEEPSLEEP mode

The software enters DEEPSLEEP mode according to the following steps:

- Set the VREFOFF register
- Configure the PMOD register to 10
- Execute WFI or WFE instruction

Hardware behavior in DEEPSLEEP mode

In the DEEPSLEEP mode, the chip automatically turns off the CPU clock, turns off the internal reference source, Flash enters STOP mode, and the hardware automatically turns off RCHF, PLL, and TRNG. SVD, OPA, ADC, DAC, and COMP can still work in the DEEPSLEEP mode. Among them, because the high-speed clocks are all turned off, the highest ADC working clock is only RCLF, which is equivalent to the fastest sampling rate of 38Ksps. In DEEPSLEEP mode, VREF1p2 can be selectively enabled or disabled, and can be flexibly set by the software according to the needs of peripheral functions.

Digital peripheral modules can continue to work with low-speed clocks such as RCLF, XTLF, and RCLP.

Exiting DEEPSLEEP mode

Exiting the DEEPSLEEP mode as follows:

- A specific interrupt event occurs
- The system clock is automatically configured as RCHF
- When the CPU is awakened, it can enter or not enter the interrupt service routine, depending on the software configuration

5.2.8 VBAT Mode

For chips that include the VBAT pin in the package pin, when the VDD is powered off, the on-chip power switching circuit will automatically switch the power supply of the backup power domain to the VBAT pin. At this time, the chip can only maintain the travel time of RTC_B and the contents of the backup register. For details, see Chapter 8 Backup Power Domain.

5.3 Wake-up Events

Wake up events	Application	Effective mode	
		Sleep	DeepSleep
Oscillation stop detection	Maskable, Wake up the chip when 32786Hz crystal fails	√	√
VREF	Maskable, Wake up the chip when an interrupt occurs after VREF1p22 is established	√	√
SVD	Maskable, Wake up the chip when the supply voltage falls below or rises above the threshold	√	√
COMP	Maskable, Used for external event wake-up	√	√
ADC	Maskable, Various interrupts of ADC can be used to wake up	√	√
RTCA RTCB	Maskable, Set the wake-up cycle as needed	√	√
IO pin interrupt	Maskable, Used for external event wake-up	√	√
Debug	Nonmaskable, For debug wake-up	√	√
LPUART	Maskable, Wake-up on data receiving	√	√
UART0/1_RXD	Maskable, wake up on falling edge	√	√
WKUPx pin	Maskable, Used for external input to wake up	√	√
NRST	Nonmaskable, For global reset	√	√
LPTIM32	Maskable, Used for periodic wake-up	√	√
BSTIM32	Maskable, Used for periodic wake-up	√	√
LPTIM16	Maskable, Used for periodic wake-up	√	√
BSTIM16	Maskable, Used for periodic wake-up	√	√
I2C slave	Maskable,Used for slave receiving wake-up	√	√

Table 5-3 Wake-up Sources in Sleep Mode Table

By enabling PRIMASK feature in Cortex-M0, you can wake up the chip with the above interrupt events, but the CPU does not execute the interrupt handler. At this point, the CPU will continue to run from the

instruction right after WFI/WFE.

Note: *After the chip is awakened from sleep mode, the software can quickly identify the current wake-up source by polling the PMU.WKPFLAG register. The clear of wake-up sources needs to be performed individually for each peripheral module.*

5.3.1 VREF1p2 Delayed Wake-up Function

In the application scenario of periodic wake-up, you may want to wake up the MCU after VREF1p2 is established. The delayed wake-up function can achieve this goal.

When the delayed wake-up function is enabled, when a periodic timing wake-up event occurs, the MCU will not be awakened immediately, but will automatically enable VREF1p2 and wait for the configurable delay time before waking up the MCU. If VREF1p2 is originally enabled, this function is invalid. And delayed wake-up is only valid for timing wake-up events (RTC and timer, that is, there must be a working clock), and it is invalid for wake-up events such as WKUPx, pins, and analog wake-up sources; regardless of whether the delayed wake-up is enabled or not, when an untimed event arrives, the MCU wakes up immediately.

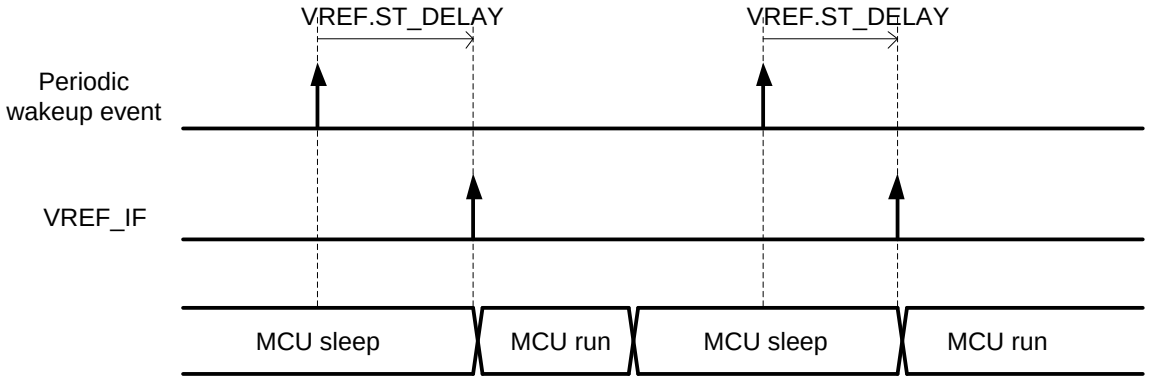


Figure 5–3 VREF1p2 Delayed Wake-up Timing Diagram

Wake-up source	VREF Delayed wake up
Stop vibration detection	X
VREF	X
SVD	X
COMP	X
ADC	X
RTC	O
IO Pin Interrupt	X
Debug	X
LPUART	X
WKUPx Pin	X

NRST	X
LPTIM16/32	O
BSTIM16/32	O
I2CSlave	X

Table 5-4 VREF1p2 Delayed Wake-up Applicable Wake-up Sources

5.4 The System Clock after Wake Up

When the chip wakes up from Sleep/DeepSleep mode, the chip uses RCHF as the clock source. The register will retain the frequency configuration and trim value of RCHF before sleep, so the CPU operating frequency after wake-up will be determined by the software configuration register before sleep (PMU.WKFSEL). In the fastest case, the chip will start with a 24MHz clock after it wakes up.

During sleep, the AHBPRES register will be reset to 000, the APBPRES register will remain unchanged, and the SYSCLKSEL register will be reset to 000 (RCHF is selected). Therefore, if the system clock is not RCHF before sleep, RCHF will be used by default after wake-up, AHBPRES does not divide the frequency, and the APBPRES register keeps the configuration before sleep.

5.5 Register

Offset	Name	Symbol
PMU(Base address: 0x40002000)		
0x00	Power Management Control Register	PMU_CR
0x04	Wakeup Time Register	PMU_WKTR
0x08	Wakeup Source Flags Register	PMU_WKFR
0x0C	PMU Interrupt Enable Register	PMU_IER
0x10	PMU Interrupt and Status Register	PMU_ISR
0x38	ULPBG Trim Register	PMU_ULPB_TR
0x3C	VREFP Control Register	PMU_VREFP_CR
0x40	VREFP Config Register	PMU_VREFP_CFGR
0x44	VREFP Interrupt Status Register	PMU_VREFP_ISR
0x48	VREFP Trim Register	PMU_VREFP_TR

5.5.1 Power Management Control Register (PMU_CR)

NAME	PMU_CR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	RFUI	-						
access	R/W-0	U-0						
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				LDO_LPM		LDO15EN	LDO15EN_B
access	U-0				R/W-01		R/Dy-1	R/Dy-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				WKFSEL		SLPDP	CVS
access	U-0				R/W-00		R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				RFUI		PMOD	
access	U-0				R/W-00		R/W-00	

bit	name	functional description
31	**RFUI**	Dummy register
30:20	**--**	RFU: **Reserved, read as 0**
19:18	**LDO_LPM**	LDO low power mode configuration 00/01/11: Normal mode 10: LDO enters low power consumption mode
17	**LDO15EN**	LDO15 enable flag 1: LDO15 is in working condition 0: LDO15 is disabled
16	**LDO15EN_B**	The inversed check bit for LDO15EN (LDO Enable Inversed)

bit	name	functional description
15:12	--	RFU: **Reserved, read as 0**
11:10	**WKFSEL**	Sleep/DeepSleep system frequency after wake up 00: RCHF-8MHz 01: RCHF-16MHz 10: RCHF-24MHz 11: RFU
9	**SLPDP**	DeepSleep control register (Sleep Deep) 1: Enable DeepSleep mode 0: Sleep mode In the Sleep mode, if the SLPDP is set, it will be switch into DeepSleep mode. This bit is only valid in Sleep mode.
8	**CVS**	Core-Voltage-Scaling configuration 0: Disable CVS 1: Enable CVS in Sleep/DeepSleep This bit is only valid in Sleep/DeepSleep mode
7:4	--	RFU: **Reserved, read as 0**
3:2	**RFUI**	Dummy register
1:0	**PMOD**	Low-power mode control register 00: Active mode / LP Active mode 01: LPRUN mode 10: Sleep mode / DeepSleep mode 11: RFU

5.5.2 Wakeup Time Register (PMU_WKTR)

NAME	PMU_WKTR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				VREFDLY	STPCLR	T1a	
access	U-0				R/W-0	R/W-0	R/W-01	

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3	**VREFDLY**	VREF wake-up delay function 0: wake up the MCU immediately after the wake-up event arrives 1: Start VREF1p2 after the wake-up event arrives, and wait for VREF1p2 to be established before waking up the MCU ***Note:*** *1) This function is only valid for timing wake-up* *2) If VREF1p2 is not closed during sleep, this function is invalid*
2	**STPCLR**	Flash Stop wake up control (Stop clear) 0: The Stop signal waits for the clock to be synchronously cleared 1: The Stop signal is asynchronously cleared
1:0	**T1a**	Programmable extra wake-up delay In the Sleep/DeepSleep mode, when the RCHF clock arrives, wait for additional delay time according to this register configuration. 00: 0us 01: 2us 10: 4us 11: 8us

5.5.3 Wakeup Source Flags Register (PMU_WKFR)

NAME	PMU_WKFR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	ADCWKF	UART1WKF	UART0WKF	RTCWKF	SVDWKF	LFDETWKF	VREFWKF	IOWKF
access	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	I2CWKF	LPU2WKF	LPU1WKF	LPU0WKF	--	COMP3WKF	COMP2WKF	COMP1WKF
access	R-0	R-0	R-0	R-0	U-0	R-0	R-0	R-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	--	LPT32WKF	LPT16WKF	BST32WKF	BST16WKF	DBGWKF	WKPxF[9:8]	
access	U-0	R-0	R-0	R-0	R-0	R/W-0	R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	WKPxF[7:0]							
access	R/W-00000000							

bit	name	functional description
31	**ADCWKF**	ADC interrupt wake up flag, automatically reset when the interrupt is cleared (ADC wakeup Flag, auto to clear)
30	**UART1WKF**	UART1 receives the falling edge asynchronous wake-up flag, and the hardware is automatically cleared when the interrupt is cancelled
29	**UART0WKF**	UART0 receives the falling edge asynchronous wake-up flag, and the hardware is automatically cleared when the interrupt is cancelled
28	**RTCWKF**	RTC interrupt wake up flag, automatically reset when the interrupt is cleared (RTC wakeup flag, auto to clear)
27	**SVDWKF**	SVD interrupt wake up flag, automatically reset when the interrupt is cleared (SVD wakeup flag, auto to clear)
26	**LFDETWKF**	32768Hz crystal fail interrupt wake up flag, automatically reset when the interrupt is cleared (XTLF fail detect wakeup flag, auto to clear)
25	**VREFWKF**	VREF1P22 reference source establishment interrupt wake up flag, automatically reset when the interrupt is cleared (Vref wakeup flag, auto to clear)
24	**IOWKF**	IO interrupt wake up flag, automatically reset when the interrupt is cleared (GPIO wakeup flag, auto to clear)
23	**I2CWKF**	I2C interrupt wake up flag, automatically reset when the interrupt is cleared (I2C wakeup flag, auto to clear)
22	**LPU2WKF**	LPUART2 interrupt wake-up flag, automatically cleared by hardware when the interrupt is cancelled
21	**LPU1WKF**	LPUART1 interrupt wake-up flag, automatically cleared by hardware when the interrupt is cancelled
20	**LPU0WKF**	LPUART0 interrupt wake-up flag, automatically cleared by hardware when the interrupt is cancelled
19	**--**	RFU: **Reserved, read as 0**
18	**COMP3WKF**	Comparator 3 interrupt wake-up flag, automatically cleared by hardware when the interrupt is cancelled

bit	name	functional description
17	**COMP2WKF**	Comparator 2 interrupt wake-up flag, automatically cleared by hardware when the interrupt is cancelled
16	**COMP1WKF**	Comparator 1 interrupt wake-up flag, automatically cleared by hardware when the interrupt is cancelled
15	**--**	RFU: **Reserved, read as 0**
14	**LPT32WKF**	LPTIM32 interrupt wake-up flag, automatically cleared by hardware when the interrupt is cancelled
13	**LPT16WKF**	LPTIM16 interrupt wake-up flag, automatically cleared by hardware when the interrupt is cancelled
12	**BST32WKF**	BSTIM32 interrupt wake-up flag, automatically cleared by hardware when the interrupt is cancelled
11	**BST16WKF**	BSTIM16 interrupt wake-up flag, automatically cleared by hardware when the interrupt is cancelled
10	**DBGWKF**	CPU Debugger wake-up flag, software write 1 to clear
9:0	**WKPxF**	NWKUPx Pin wake-up flag, software write 1 to clear

5.5.4 PMU Interrupt Enable Register (PMU_IER)

NAME	**PMU_IER**							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					LPACTIE	SLPEIE	LPREIE
access	U-0					R/W-0	R/W-0	R/W-0

bit	name	functional description
31:3	--	RFU: **Reserved, read as 0**
2	**LPACTIE**	LPACTIVE mode error interrupt enable 1: Enable LPACTIVE error interrupt 0: Disable LPACTIVE error interrupt
1	**SLPEIE**	SLEEP error interrupt enable 1: Enable SLEEP error interrupt 0: Disable SLEEP error interrupt
0	**LPREIE**	LPRUN mode error interrupt enable 1: Enable LPRUN error interrupt 0: Disable LPRUN error interrupt

5.5.5 PMU Interrupt and Status Register (PMU_ISR)

NAME	PMU_ISR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					LPACTIF	SLPEIF	LPREIF
access	U-0					R/W-0	R/W-0	R/W-0

bit	name	functional description
31:3	--	RFU: **Reserved, read as 0**
2	**LPACTIF**	LPACT error interrupt flag, set by hardware, cleared by software writing 1 1: When the LDO15LPM register is set, the system working clock is RCHF and greater than 4Mhz, or the system clock is PLL, XTHF 0: enter LPACTIVE normally
1	**SLPEIF**	SLEEP error interrupt flag, set by hardware, cleared by software writing 1 1: After PMOD=2'h2, it is set when the SLEEPDEEP register is set before the CPU executes the WFI/WFE instruction 0: After PMOD=2'h2, CPU enters SLEEP correctly
0	**LPREIF**	LPRUN error interrupt flag, set by hardware, cleared by software by writing 1; if LPREIF is triggered when the software enters

bit	name	functional description
		LPRUN mode, the chip will still stay in ACTIVE mode 1: LPRUN Condition Error, that is, the following conditions are met when entering LPRUN: 1) The system clock is not LSCLK or RCLF, or 2) RCHF enable is not closed 0: LPRUN enters normally

5.5.6 ULPBG Trim Register (PMU_ULPB_TR)

NAME	PMU_ULPB_TR							
Offset	0x38							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			ULPBG_TRIM				
access	U-0			R/W-10000				

bit	name	functional description
31:5	--	RFU: **Reserved, read as 0**
4:0	ULPBG_TRIM	1.16V reference voltage adjustment register output by ULPBG

5.5.7 VREFP Control Register (PMU_VREFP_CR)

NAME	PMU_VREFP_CR							
Offset	0x3C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

name	-	DEND_IE	POV_IE	EN
access	U-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:3	--	RFU: **Reserved, read as 0**
2	DEND_IE	Driving end interrupt enable 1: Allow DEND interrupt 0: Disable DEND interrupt
1	POV_IE	Periodic overflow interrupt enable 1: Allow POV interrupt 0: Disable POV interrupt
0	EN	VREFP_VREG enable register (enable) 0: Close and bypass VREFP_VREG 1: Enable VREFP_VREG

5.5.8 VREFP Config Register (PMU_VREFP_CFGR)

NAME	PMU_VREFP_CFGR							
Offset	0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-					VRS		
access	U-0					R/W-000		
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TPERIOD			TDRV			LPM	-
access	R/W-000			R/W-000			R/W-0	U-0

bit	name	functional description
31:10	--	RFU: **Reserved, read as 0**
10:8	VRS	Output voltage selection (Voltage regulation select) 000: VREFP output 2.0V 001: VREFP output 2.5V 010: VREFP output 3.0V 011: VREFP output 4.5V 1xx: VREFP output 1.5V
7:5	TPERIOD	Enable period in intermittent enable mode (Time of Period) 000: 1ms

bit	name	functional description
		001: 4ms 010: 16ms 011: 32ms 100: 64ms 101: 256ms 110: 1s 111: 4s
4:2	TDRV	Time of Driving in Intermittent Enable Mode (Time of Driving) 000: 4*TLSCLK 001: 8*TLSCLK 010: 16*TLSCLK 011: 32*TLSCLK 100: 64*TLSCLK 101: 128*TLSCLK 110: 256*TLSCLK 111: 512*TLSCLK
1	LPM	Intermittent enable register (Low power mode) 0: Always enable mode 1: Intermittent enable mode
0	--	RFU: **Reserved, read as 0**

5.5.9 VREFP Interrupt Status Register (PMU_VREFP_ISR)

NAME	PMU_VREFP_ISR							
Offset	0x44							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					BUSY	DEND	POV
access	U-0					R-0	R/W-0	R/W-0

bit	name	functional description
31:3	--	RFU: **Reserved, read as 0**
2	BUSY	Drive flag, read only (Busy) 0: VREG is not currently driving

bit	name	functional description
		1: VREG is driving
1	DEND	End of driving, only valid in intermittent enable mode (Driving End) Set by hardware after each driving cycle, and clear by software by writing 1 When DEND_IE is set, DEND is set to trigger an interrupt.
0	POV	Intermittent enable period overflow, only valid in intermittent enable mode (Periodic Overflow) Set by hardware after each intermittent enable period, and clear by software by writing 1 When POV_IE is set, POV is set to trigger an interrupt.

5.5.10 VREFP Trim Register (PMU_VREFP_TR)

NAME	PMU_VREFP_TR							
Offset	0x48							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TRIM							
access	R/W-1000 0000							

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:0	TRIM	VREFP output voltage adjustment (trimming) Each LSB corresponds to a step length of 0.1%, up to 127steps or minus 127steps can be added, corresponding to the adjustment range of -12.7% ~ +12.7%

6 High-Precision Reference Source (VREF1p2)

6.1 Introduction

FM33LG0 integrates a high-precision reference source with a typical output voltage of about 1.2V, which can work stably within the entire operating power supply range of the chip. After this reference voltage is output from the Buffer, it can be sampled by the ADC and also used as the reference voltage input of the comparator.

In the entire operating temperature range, the typical temperature coefficient of this reference source is less than 25ppm/°C, and a built-in temperature sensor output is used for ADC sampling and measurement of the current chip substrate temperature.

The software can turn on or turn off this reference source. After turning on the reference source, the VREF1p2 output setup time is less than 1ms, and the typical power consumption is about 1.5uA. When the temperature sensor is turned on, the power consumption of VREF1p2 is less than 2uA.

After the software enables VREF1p2, there is a hardware delay circuit inside the chip. After waiting for enough time to ensure that the VREF output is fully established, the VREF_DRY status flag register is set, and the VREF_IF interrupt flag is set. The software can confirm the effective establishment of VREF1p2 by timing or according to the VREF_RDY register.

When the software closes VREF1p2, the VREF_RDY register is automatically cleared, and VREF_IF is cleared by software writing 1.

The maximum temperature measurement range supported by the temperature sensor is -55~125°C. The output voltage of the temperature sensor changes with temperature as a straight line with a positive temperature coefficient, with a typical slope of 2.6mV/°C. Before the chip leaves the factory, the temperature sensor will be calibrated under the condition of 30°C +/- 1°C. Under this condition, the temperature measurement error in the range of -40~+85°C is within +/-2°C.

6.2 Application of Reference Voltage

The VREF1p2 output can be used for the following purposes:

- Reference input of VREFP_VREG
- ADC measurement input (VREF1p2 and PTAT)
- COMP input reference voltage

6.3 Temperature Sensor

The built-in PTAT circuit of VREF1p2 can output a voltage signal with a positive temperature coefficient. Since the slope of the output voltage and temperature change is determined, the PTAT voltage can be

measured by ADC to measure the current chip substrate temperature. In order to ensure the accuracy of absolute temperature measurement, the chip will perform temperature sensor output calibration under the conditions of VDDA=3V+/-10mV and TA=30C+/-1C before leaving the factory, and write the calibration data (TS_CAL1) into Flash. In application, according to the temperature calibration data and the slope of the voltage-temperature fitting line, the absolute temperature of the current chip substrate can be calculated. In the full temperature range of -40~+85C, the absolute temperature measurement error is less than +/-2C.

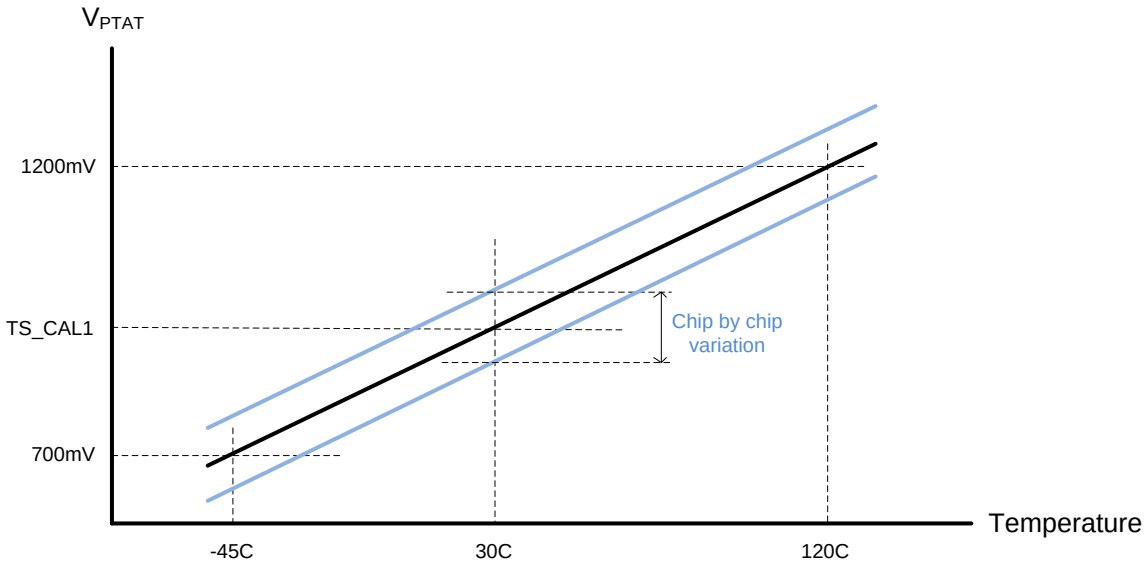


Figure 6–1 Temperature Sensor Output and Calibration

In the range of -40~+85C, VDD=1.8~5.5V, the typical slope of PTAT output relative to temperature change is 2.6mV/C, and the PTAT output voltage range is roughly 700mv~1100mV.

6.4 Output Buffer

Both VREF1p2 output and PTAT output have built-in output buffers to enhance signal driving. When using ADC to sample VREF1p2 and PTAT output, it is recommended to enable the built-in buffer to shorten the sample hold time. When using the built-in buffer of VREF1p2, it is recommended that the sampling time is not less than 10us.

6.5 Chip Sleep

When the chip sleeps, whether VREF1p2 keeps working is determined by the software.

If the software turns off VREF1p2, VREFP_VREG will not be able to keep the output. The software can use the intermittent enable mode of VREF_VREG. In this mode, VREF_VREG is turned on regularly. If VREF1p2 is turned off, the hardware will automatically turn on VREF1p2. See 7VREFP reference voltage (VREFP_VREG) for details

6.6 Register

Base address: 0x4001A400

Offset	Name	Symbol
VREF1p2(Base address: 0x4001A400)		
0x00	VREF Control Register	VREF_CR
0x04	VREF Config Register	VREF_CFGR
0x08	VREF Status Register	VREF_ISR
0x0C	VREF Interrupt Enable Register	VREF_IER
0x10	Buffer Control Register	VREF_BUFCR

6.6.1 VREF Control Register (VREF_CR)

NAME	VREF_CR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						PTAT_EN	VREF_EN
access	U-0						R/W-0	R/W-1

bit	name	functional description
31:2	-	RFU: **Reserved, read as 0**
1	PTAT_EN	Band gap temperature sensor enable (Temperature sensor enable) 0: Turn off the temperature sensor output 1: Enable temperature sensor output
0	VREF_EN	VREF1p2 enable register (Voltage reference enable) 0: Turn off VREF1p2 1: Enable VREF1p2

6.6.2 VREF Config Register (VREF_CFGR)

NAME	VREF_CFGR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						ST_DELAY	
access	U-0						R/W-00	

bit	name	functional description
31:2	-	RFU: **Reserved, read as 0**
1:0	ST_DELAY	Start Delay, after setting the VREF_EN register to enable VREF1p2 output, wait for a configurable time, and the hardware automatically sets the VREF_RDY register to indicate that the reference voltage has been established. 00: 1.5ms 01: 1ms 10: 750us 11: 500us

6.6.3 VREF Status Register (VREF_ISR)

NAME	VREF_ISR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							FLAG_B
access	U-0							R-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						RDY	IF

access	U-0	R-0	R/W-0

bit	name	functional description
31:9	-	RFU: **Reserved, read as 0**
8	FLAG_B	VREF Settable Flag from Analog, auto to clear
7:2	-	RFU: **Reserved, read as 0**
1	RDY	VREF1p2 reference voltage establishment flag (VREF Ready Flag, auto to clear) After VREF1p2 is enabled, it is delayed set by the digital circuit, and the software is read-only. After closing the VREF1p2 module, this register is automatically cleared. Digital circuit delay time 1.5ms
0	IF	VREF1p2 reference voltage establishment interrupt (VREF Ready Interrupt Flag, write 1 to clear) 0: VREF1p2 is not established 1: VREF1p2 is established After VREF1p2 is enabled, this flag is delayed by the digital circuit to be set, set by hardware, and cleared by software writing 1

6.6.4 VREF Interrupt Enable Register (VREF_IER)

NAME	VREF_IER							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							IE
access	U-0							R/W-0

bit	name	functional description
31:1	-	RFU: **Reserved, read as 0**
0	IE	VREF1p2 Reference Voltage Establishment Interrupt Enable (VREF Ready Interrupt Enable) 0: Disable the generation of VREF establishment completion interrupt

bit	name	functional description
		1: Allow to generate VREF establishment completion interrupt When this register is 1, when VREF1p2 is established, the output will be interrupted to the CPU

6.6.5 Buffer Control Register (VREF_BUFCR)

NAME	VREF_BUFCR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		AVREFBUF_OUTEN	AVREFBUF_EN	VPTATBUFFER_OUTEN	VPTATBUFFER_EN	VREFBUFFER_OUTEN	VREFBUFFER_EN
access	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:6	-	RFU: **Reserved, read as 0**
5	AVREFBUF_OUTEN	AVREF output buffer output enable, it is recommended to enable when ADC samples AVREF
4	AVREFBUF_EN	AVREF output buffer is enabled, it is recommended to enable when ADC samples AVREF
3	VPTATBUFFER_OUTEN	Vptat Buffer module switch channel output enable signal, high level enable is effective. (PTAT Buffer Output Enable)
2	VPTATBUFFER_EN	Vptat Buffer module enable signal, high level enable is valid. (PTAT Buffer Enable)
1	VREFBUFFER_OUTEN	Vref Buffer module switch channel output enable signal, high level enable is effective. (VREF Buffer Output Enable)
0	VREFBUFFER_EN	Vref Buffer module enable signal, high level enable is valid. (VREF Buffer Enable)

7 VREFP Reference Voltage (VREFP_VREG)

7.1 Introduction

FM33LG0 can obtain an external reference voltage from the VREFP pin, or generate a multi-level configurable reference voltage from the on-chip circuit.

The internal reference voltage is generated by the VREFP_VREG circuit. VREFP_VREG uses VREF1p2 as the reference voltage to obtain 4.5V, 3.0V, 2.5V, 2.0V, 1.5V output, and a 1uF voltage regulator capacitor needs to be connected to the VREFP pin.

When using an external reference, the external reference voltage is directly applied to the VREFP pin, and at the same time, the VREFP_VREG circuit must be closed to avoid level conflicts.

The reference voltage generated by VREFP_VREG can not only be used for internal circuits (such as ADC, DAC, comparator, etc.), but also for off-chip use. The drive capability of VREFP_VREG is 10mA.

In order to reduce the overall power consumption of the chip, VREFP_VREG also supports intermittent startup mode, which maintains the voltage on the off-chip capacitor through intermittent enable, which can greatly reduce sleep power consumption and still maintain the VREFP reference voltage within an acceptable range.

7.2 Function Description

7.2.1 Introduction

The VREFP_VREG circuit can be used to generate a constant internal reference voltage. At this time, the reference power supply of the ADC and DAC can be made independent of the external power supply, so it is not affected by the fluctuation of the external power supply.

The reference source of ADC can be switched between VREFP and VDDA, and the reference source of DAC is fixedly connected to VREFP.

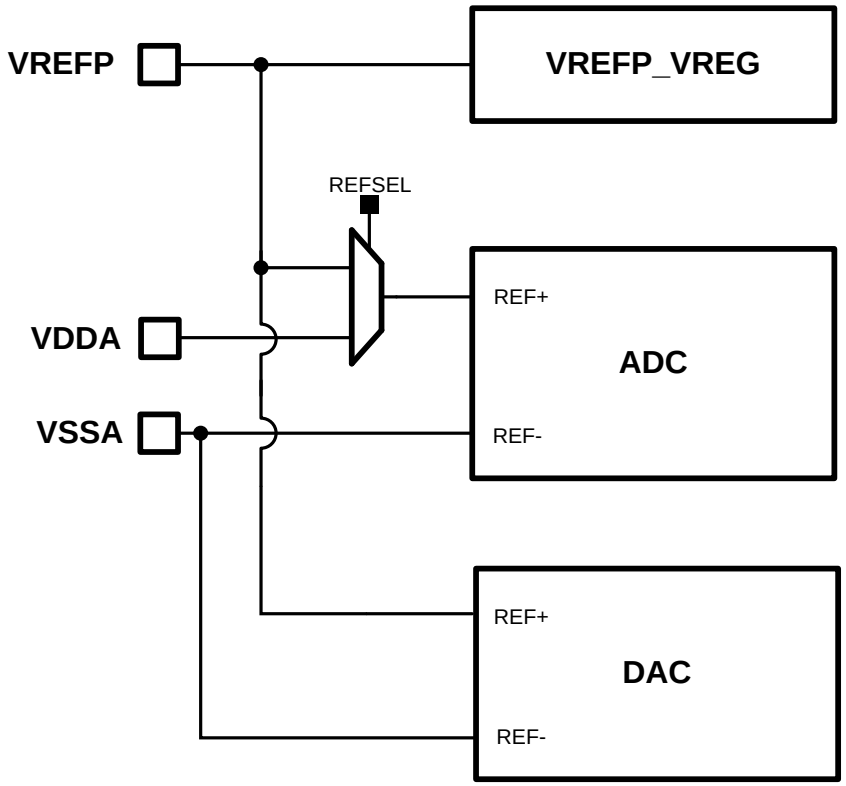


Figure 7–1 ADC and DAC Reference Source

The system scheme can support both external benchmarks and internal benchmarks.

7.2.1.1 External Power Supply as Reference

When using an external power supply as a reference, the application should keep VREFP_VREG off, and short-circuit the VREFP and VDDA pins on the system. At this time, the reference sources of ADC and DAC are equivalent on the system.

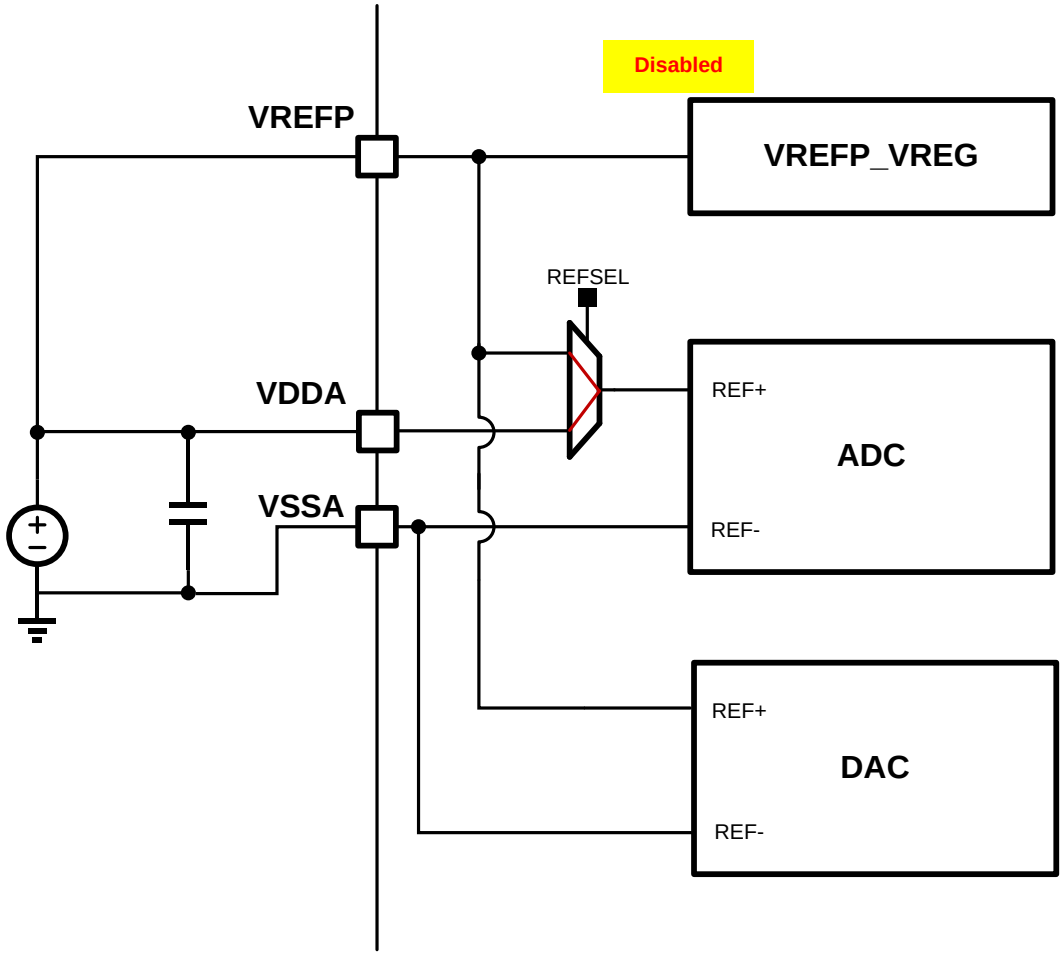


Figure 7–2 External Power Supply Reference Scheme 1

7.2.1.2 External Input Independent Reference

It can also support the system to provide a reference voltage independent of the power supply. At this time, VREFP_VREG remains closed, and the reference source of ADC and DAC is selected from VREFP. In this solution, VREFP and VDDA cannot be short-circuited on the PCB, and the corresponding package must be an independent VREFP pin.

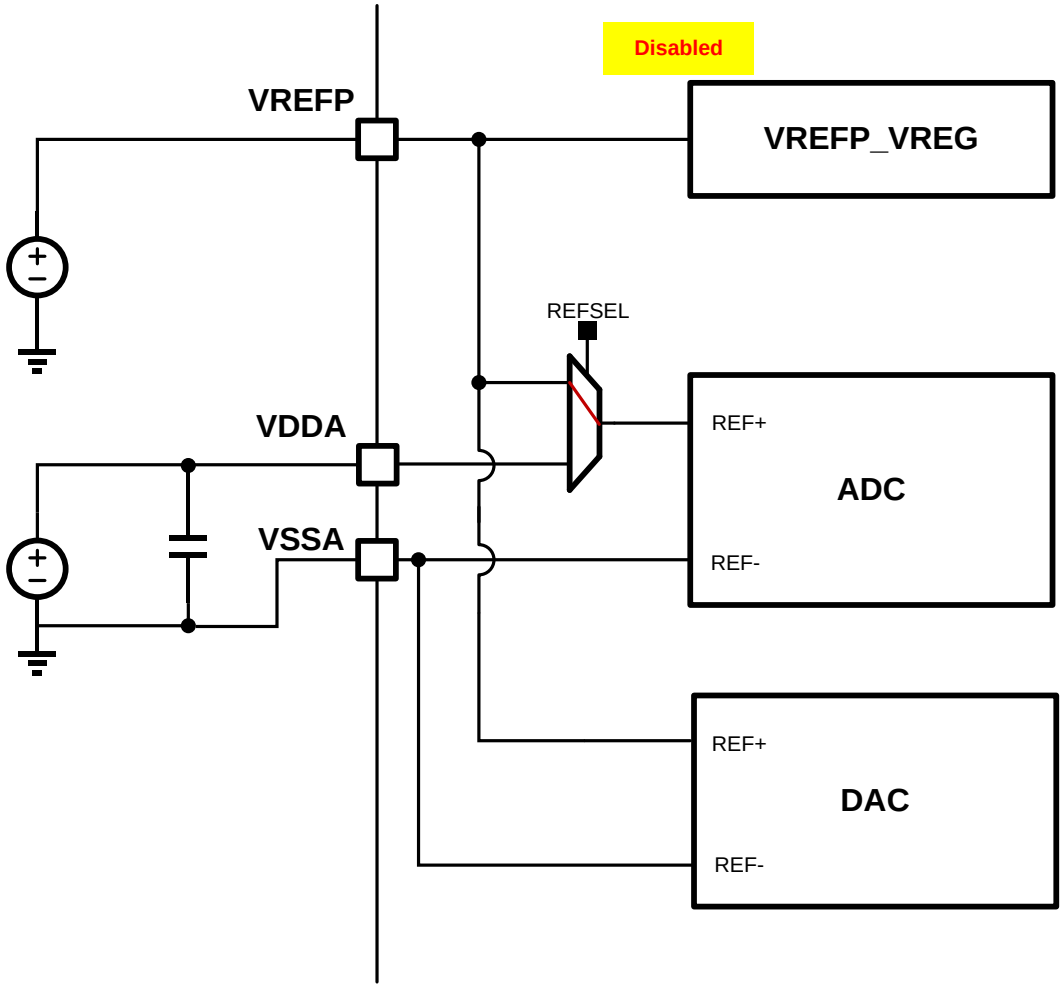


Figure 7–3 External Independent Reference Scheme

7.2.1.3 Internal Reference

When the internal reference is enabled, VREFP_REG behaves as a voltage regulator, and a 1uF capacitor must be connected to the VREFP pin; the following figure shows the scheme of using internal reference sources for both ADC and DAC.

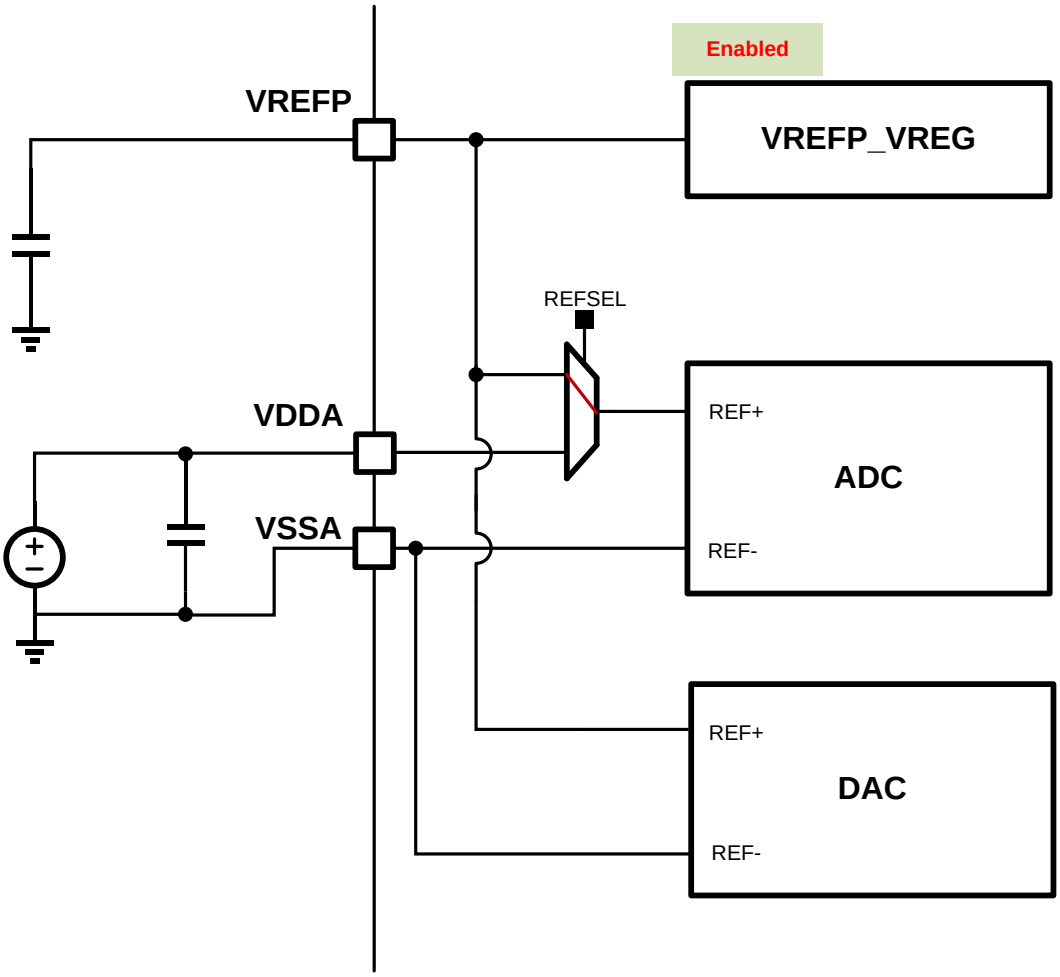


Figure 7–4 ADC and DAC are System Connections Using Internal Reference Schemes

In the figure below, ADC uses VDDA as a reference, while DAC uses an internal reference.

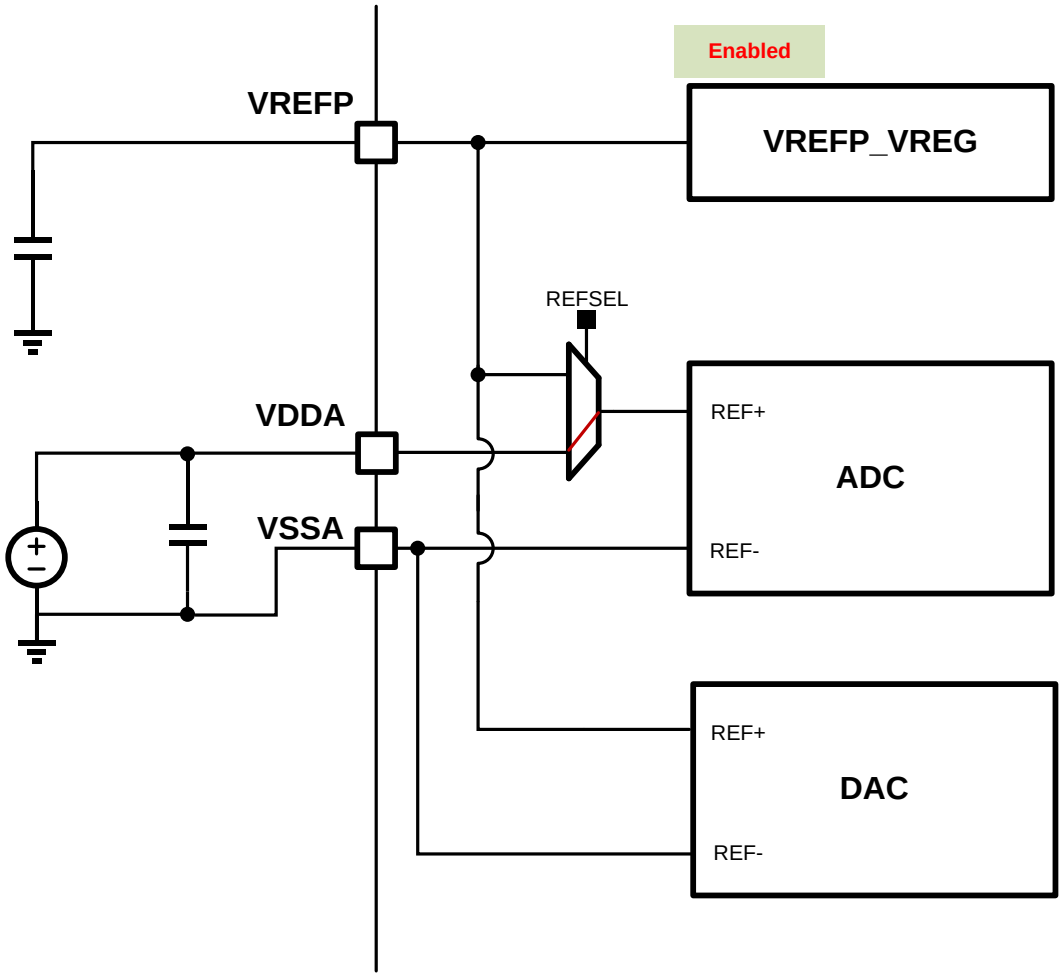


Figure 7–5 ADC Uses an External Reference, DAC Uses an Internal Reference

Only chips with independent VREFP pin packages can support the internal reference scheme.

Note: *While this chip cannot support ADC using internal reference, DAC uses external reference.*

7.2.2 Clock and Reset

The VREFP_VREG controller uses two working clocks: APBCLK and LSCLK.

The bus access of the register only uses APBCLK, and the intermittent start control uses LSCLK.

The VREFP_VREG module does not have a separate peripheral reset function.

7.2.3 Reference Voltage

The reference voltage of VREFP_VREG comes from VREF1p2, so when using the internally generated VREFP, VREF1p2 must be enabled.

7.2.4 Output Voltage

VREFP_VREG supports 5 output gears, and supports at least +/-10% of the output adjustment range, the adjustment step is 0.1%, that is, the output accuracy of +/-0.05% can be achieved after trimming.

Because ADC uses VREFP as the reference voltage, the application can confirm whether VREFP has been established within the normal range by measuring the converted value output by the reference source.

At the same time, VREFP is also used as the reference power supply of the DAC and can be used as the reference voltage input of the comparator.

7.2.5 Working Mode

VREFP_VREG supports two working modes: always enable and intermittent enable, controlled by LPM register.

When LPM=0, VREFP_VREG remains always enabled, unless the software turns off the module by clearing the EN register. Note that the software cannot turn off VREF1p2 in the always-on mode.

When LPM=1, VREFP_VREG works in intermittent enable mode, that is, it is turned on intermittently at a fixed time, and then turned off automatically after driving for a programmable period of time. The period of timing on can be programmed by software to balance power consumption and VREFP ripple amplitude.

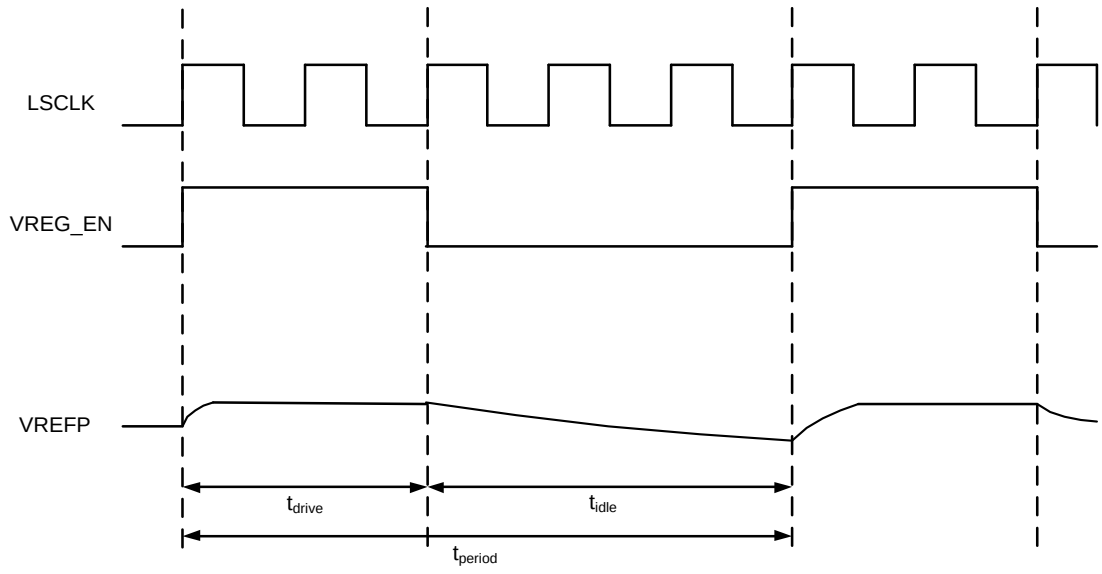


Figure 7–6 VREFP Intermittent Enable Waveform Diagram

The intermittent enable mode can maintain the VREFP voltage with very low power consumption when the chip is sleeping. If the software turns off VREF1p2, the control circuit will automatically enable VREF1p2 at the beginning of each intermittent enable, wait for its establishment completion

flag, and then enable VREFP_VREG to start output driving. In this case, the drive time configured in the TDRV register does not include the setup time of VREF1p2. If VREF1p2 is not turned off, turn on VREFP_VREG directly when intermittently is enabled.

Note that when VREF1p2 is turned off, the intermittent startup period configuration must be greater than the sum of VREF1p2 setup time (1.5ms) and drive time.

$$t_{period} > t_{VREF\_setup} + t_{drive}$$

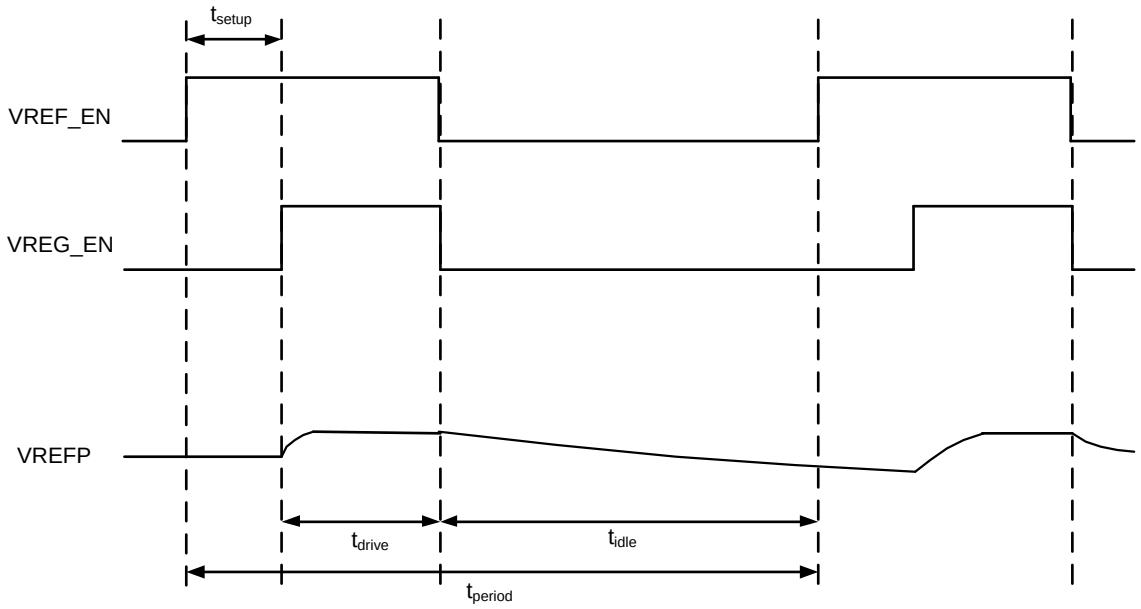


Figure 7–7 Schematic Diagram of Intermittent Enable Waveform When VREF1p2 is Turned Off

7.2.6 Interrupts and Flags

VREFP_VREG outputs the BUSY flag when it is enabled. In the intermittently enabled state, the software can check the BUSY register to confirm whether the current VREFP_VREG is enabled.

In intermittent enable mode, VREFP_VREG will also generate two interrupt flags; when a start is completed, a DEND flag is generated, and when an intermittent period is over, a POV flag is generated.

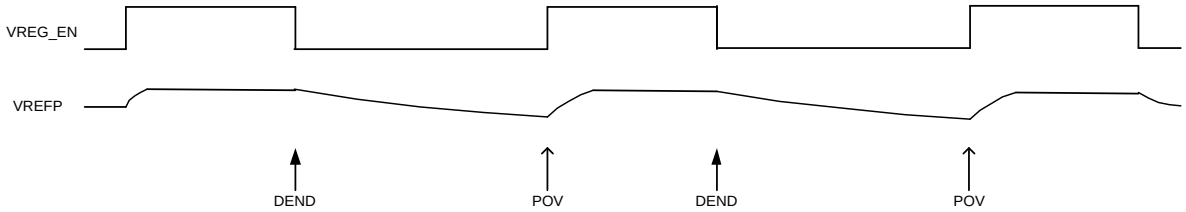


Figure 7–8 Schematic Diagram of Interrupt Flag

7.3 Register

See Chapter 5.5 register of PMU.

8 VAO Domain

8.1 Introduction

The VAO power domain adopts the main backup power supply to supply power after switching. When the VDD power supply drops and the PDR power-off reset occurs, the chip automatically switches the VAO power supply to VBAT; when VDD is re-powered and returns to the PDR power-on reset threshold, the chip automatically switches the VAO power supply to VDD.

The VAO power domain contains the following circuits:

- Power switch
- XTLF and LFDET
- RTCB
- Backup register
- 32768 crystal oscillator pins, PH15 is GPIO+Tamper
- Power-on reset circuit
- Simulation test Buffer

8.2 Block Diagram

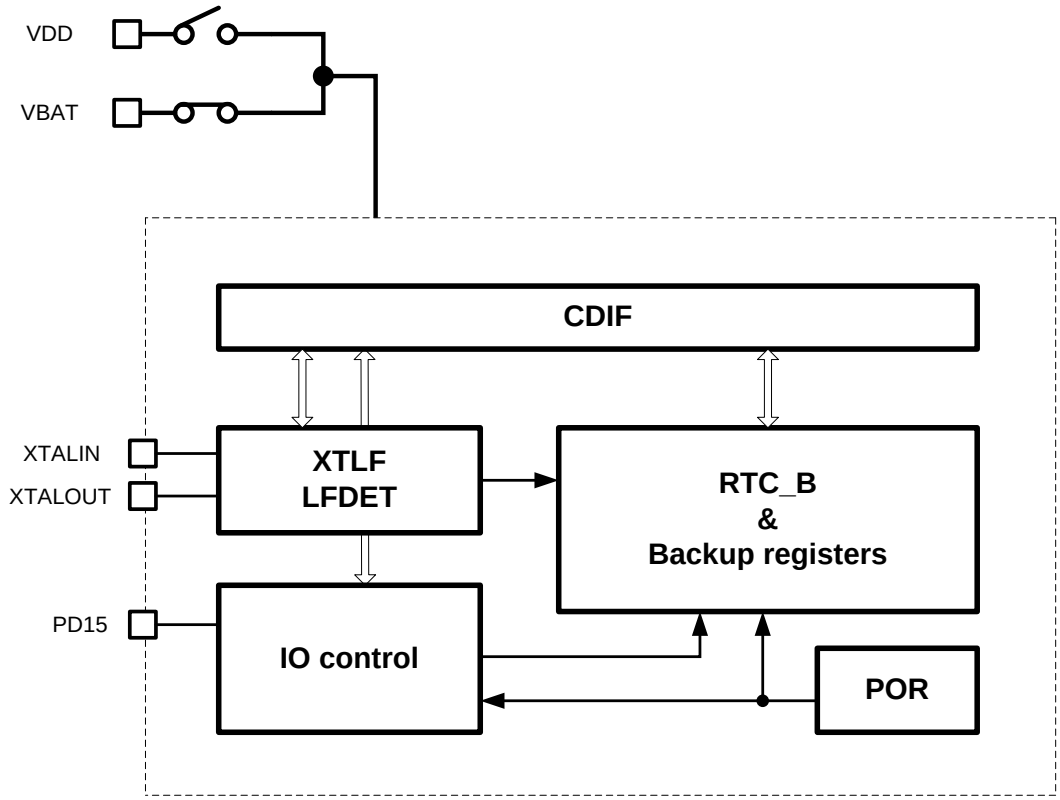


Figure 8–1 Backup Power Domain Structure Block Diagram

8.3 Power Switch

The Power Switch circuit uses the power-off reset signal generated by the PDR to switch the backup power supply.

If you apply VBAT power switching in the system solution, you must ensure the following precautions:

- The software should close the BOR circuit and detect the power supply voltage through SVD
- The PDR reset threshold should use the highest gear, that is, RMU_PDRCR.CFG=11

It is recommended to use a 3.3V battery to supply power to the VBAT pin, and the maximum value does not exceed 4.2V

8.4 Reset

The power-on reset circuit under VAO power supply is used to reset the control register under VAO when VDD or VBAT is powered on, to ensure that the PH15 pin has a certain initial state after

power-on. XTLF and some RTCB control registers are certain Initial value.

After the VAO power supply is established, the power-on reset circuit is automatically closed to save power consumption. At this time, the software can still manually reset the VBAT_RST register through CDIF.

The VAO power domain has the following two reset methods:

- Realize software reset by setting VBAT_RST register
- When both VDD and VBAT are powered down, power-on of VDD or VBAT will trigger VAO power-on reset

The schematic diagram of VAO power-on reset is as follows.

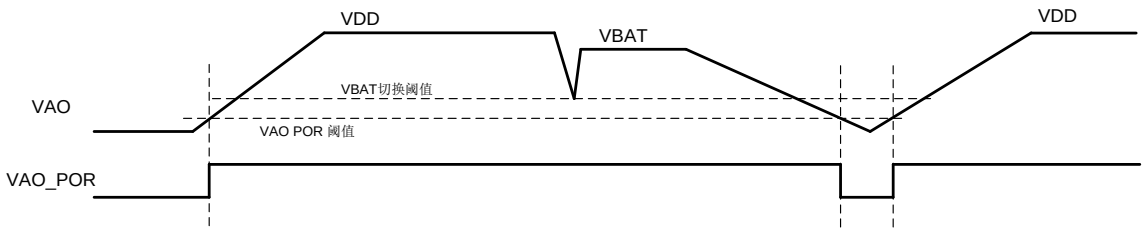


Figure 8–2 Schematic Diagram of VAO Power-on Reset

After the VAO power-on reset, XTLF is turned on by default, the PH15 control register defaults to high-impedance state, turning off the input and output.

8.5 Low Frequency Crystal Oscillator Circuit (XTLF)

8.5.1 Introduction

The low-frequency crystal oscillation circuit provides a stable oscillation source through an external 32768Hz crystal, with extremely low power consumption, and is mainly used to provide an input clock for the real-time clock (RTC) module. The oscillation intensity of XTLF is adjustable, and users can select the oscillation intensity according to their needs to achieve a balance between oscillation capacity and power consumption. The feedback resistor of XTLF is integrated inside the chip, and the user needs to add a load capacitor to the oscillation pin.

A stop-vibration detection circuit is integrated inside the chip to detect whether the XTLF stops vibration. Once the XTLF vibration stop is detected, an XTLF vibration stop interrupt will be generated, and the CPU will be notified to deal with it in time.

Software can enable or disable XTLF. In order to improve the anti-interference ability, the 4-bit XTLFEN control bit is used, and the 4-bit reset value is 0101, which must be rewritten to 1010 to turn off XTLF, and any other data will keep XTLF enabled.

8.5.2 Working Mode

XTLF is enabled by default after power-on, and the software can be turned off. By default, the medium intensity is used to shorten the start-up time, and the corresponding oscillation power consumption is also larger. The typical start-up time is less than 1s. When the oscillator is fully started, the software can reduce the oscillation power consumption through the configuration register.

8.5.3 Oscillation Failure Detection

FM33LG0 has an on-chip oscillation failure detection circuit. After it is enabled, it can continuously detect the XTLF output. When XTLF is found to stop vibration, an alarm interrupt will be generated. The software can determine whether to automatically switch LSCLK to RCLP through the LSCATS register.

When LSCATS=1, when FDET detects that XTLF has stopped oscillation, the hardware will automatically enable RCLP and switch LSCLK to RCLP output; when LSCATS=0, oscillation failure detection will only generate an alarm interrupt, and will not automatically switch the clock.

When XTLF is stopped, the software can also switch XTLF by setting LSCTS.

The vibration stop detection circuit is always opened or closed at the same time as XTLF. It cannot be closed separately. Once XTLF is enabled, the oscillation failure detection circuit will automatically open; when XTLF is closed, the oscillation failure detection will also be automatically closed to avoid false triggering of the oscillation failure alarm.

8.6 Real-time Clock (RTCB)

See Chapter 36 Real-time Clock (RTCB) for details

8.7 IO

The GPIO under the VAO power supply is only PH15.

There are also two XTLF oscillation pins under VAO, which are connected to a 32768hz crystal; they cannot be configured as GPIO.

PH15 is mainly RTCOUT and Tamper function pins.

The above pins are powered after the power switch, so the drive capability is limited, and the maximum source current cannot exceed 1mA.

The control registers are all located under the VAO power supply, and the CPU needs to be accessed through the CDIF interface.

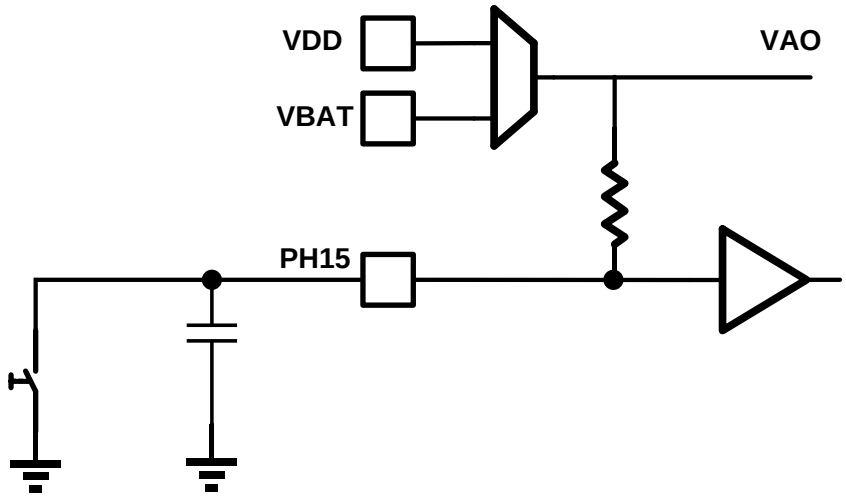
PH15 is a 5V tolerant IO with a built-in pull-up resistor, which is powered by the switched VAO power

supply. When the chip is powered by VDD, PH15 is also powered by VDD, so the high level of its push-pull output is VDD; when VDD is powered off, the chip backup power domain is powered by VBAT, at this time PH15 is also powered by VBAT, and the push-pull output is high The level is VBAT.

8.7.1 Use PH15 Input to Achieve Tamper Detection

In the scenario of VDD power supply and VDD power failure, the tamper detection function needs to be implemented.

The figure below is an example of using PH15 for tamper detection. It is pulled up to VAO through the internal pull-up resistor of the PAD. When a tamper event occurs, a logic 0 is input.



It is not recommended to connect PH15 to VBAT through an external pull-up, otherwise, VBAT and VDD are not equal, resulting in leakage. Therefore, in this solution, only the PH15 internal pull-up resistor can be used to achieve the pull-up.

PH15 is a 5V tolerant pin, which can accept an external input level higher than VAO.

8.8 Register

Offset	Name	Symbol
VAO(Base address: 0x4001F000)		
0x800	VBAT Reset Control Register	VAO_RSTCR
0x804	XTLF Control Register	VAO_XTLFCR
0x808	XTLF Power Register	VAO_XTLFPR
0x80C	XTLF Oscillation Fail Detection Interrupt Enable Register	VAO_FDIER
0x810	XTLF Oscillation Fail Detection Interrupt Status Register	VAO_FDISR
0xC00	VAO IO Input Enable Register	VAO_INEN
0xC04	VAO IO Pull-up Enable Register	VAO_PUEN
0xC08	VAO IO Open Drain Enable Register	VAO_ODEN
0xC0C	VAO IO Function Control Register	VAO_FCR
0xC10	VAO IO Data Output Register	VAO_DOR
0xC14	VAO IO Data Input Register	VAO_DIR
0xC18	VAO IO Voltage Input Low Register	VAO_VILR

8.8.1 VAO Reset Control Register (VAO_RSTCR)

NAME	VAO_RSTCR							
Offset	0x800							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							VBAT_RST
access	U-0							R/W-0

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
0	VBAT_RST	VBAT power domain register reset control 1: Reset the registers under the VBAT power domain (including some internal registers of RTCB)

bit	name	functional description
		0: Undo reset

8.8.2 XTLF Control Register (VAO_XTLFCR)

NAME	VAO_XTLFCR							
Offset	0x804							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				XTLFEN			
access	U-0				R/W-xxxx			

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3:0	XTLFEN	XTLF enable register, XTLF is enabled by default after power-on 1010: Close XTLF and FDET 0101: Enable XTLF and FDET When XTLF is working, the software must write 1010 to close it, when XTLF is not working, the software must write 0101 to start it Physically only 1 bit

8.8.3 XTLF Power Register (VAO_XTLFPR)

NAME	VAO_XTLFPR							
Offset	0x808							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							

bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	DRVCFG			XTLFIPW			
access	U-0	R/W-001			R/W-1000			

bit	name	functional description
31:7	--	RFU: **Reserved, read as 0**
6:4	DRVCFG	Output stage drive capability configuration 000: No drive 001: the weakest drive ... 111: The strongest drive
3:0	XTLFIPW	XTLF working current selection. The larger the current, the higher the oscillation intensity. After power-on reset, use the 0000 gear to start the vibration. It is recommended to use the 1000 gear during normal operation. Actually, the appropriate current should be selected according to the measured negative resistance characteristics of the adapted crystal 0000: 850nA 0001: 800nA 0010: 750nA 0011: 700nA 0100: 650nA 0101: 600nA 0110: 550nA 0111: 500nA 1000: 450nA (default) 1001: 400nA 1010: 350nA 1011: 300nA 1100: 250nA 1101: 200nA 1110: 150nA 1111: 100nA

8.8.4 XTLF Oscillation Fail Detection Interrupt Enable Register (VAO_FDIER)

NAME	VAO_FDIER							
Offset	0x80C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							

access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							LFDET_IE
access	U-0							R/W-0

bit	name	functional description
31:1	-	RFU: **Reserved, read as 0**
0	LFDET_IE	XTLF fail detect interrupt enable, 1 is valid

8.8.5 XTLF Oscillation Fail Detection Interrupt Status Register (VAO_FDISR)

NAME	VAO_FDISR							
Offset	0x810							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						LFDETO	LFDETIF
access	U-0						R-0	R/W-0

bit	name	functional description
31:2	-	RFU: **Reserved, read as 0**
1	LFDETO	XTLF fail detect output 1: XTLF oscillation does not fail 0: XTLF oscillation fails
0	LFDETIF	Low-frequency oscillation fail detection interrupt flag register. When XTLF oscillation fails, the hardware is asynchronously set, and software writes 1 to clear it; this register can only be cleared when LFDETO is not 0 (XTLF fail detect interrupt flag, write 1 to clear)

8.8.6 VAO IO Input Enable Register (VAO_INEN)

Name	VAO_INEN							
Offset	0xC00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PHINEN15	-						
access	R/W-0	U-0						
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							
access	U-0							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15	PHINEN	PH15 pin input enable register 0: Turn off input 1: Enable input
14:13	--	RFU: **Reserved, read as 0**
12:0	--	RFU: **Reserved, read as 0**

8.8.7 VAO IO Pull-up Enable Register (VAO_PUEN)

NAME	VAO_PUEN							
Offset	0xC04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PHPUEN15	-						
access	R/W-0	U-0						
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							
access	U-0							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15	PHPUEN	GPIO PH15 pull-up control 0: Turn off the pull-up

bit	name	functional description
		1: Enable pull-up
14:13	--	RFU: **Reserved, read as 0**
12:0	--	RFU: **Reserved, read as 0**

8.8.8 VAO IO Open Drain Enable Register (VAO_ODEN)

NAME	VAO_ODEN							
Offset	0xC08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PHODEN15	-						
access	R/W-0	U-0						
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							
access	U-0							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15	PHODEN	GPIO PH15 open drain output enable 0: Turn off the open-drain output 1: Enable open-drain output
14:13	--	RFU: **Reserved, read as 0**
12:0	--	RFU: **Reserved, read as 0**

8.8.9 VAO IO Function Control Register (VAO_FCR)

NAME	VAO_FCR							
Offset	0xC0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	Px15FCR		-					
access	R/W-0		U-0					
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							

access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							
access	U-0							

bit	name	functional description
31:30	PH15FCR	PH[15]Pin Function Select 00: GPIO Input 01: GPIO Output 10: RTCOUT 11: RFU
29:0	--	RFU: **Reserved, read as 0**

8.8.10 VAO IO Data Output Register (VAO_DOR)

NAME	VAO_DOR							
Offset	0xC10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PHDO15	-						
access	R/W-0	U-0						
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							
access	U-0							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15	PHDO	GPIO output data register PH15
14:13	--	RFU: **Reserved, read as 0**
12:0	--	RFU: **Reserved, read as 0**

8.8.11 VAO IO Data Input Register (VAO_DIR)

NAME	VAO_DIR							
Offset	0xC14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PxDIN15	-						
access	R	U-0						
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							
access	U-0							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15	PHDIN	Portx input data register PH15 This register only occupies the address space and has no physical implementation. Software reads this register and returns the pin input signal directly, the chip does not latch the pin input
14:13	--	RFU: **Reserved, read as 0**
12:0	--	RFU: **Reserved, read as 0**

8.8.12 VAO IO Voltage Input Low Register (VAO_VILR)

NAME	**VAO_VILR**							
Offset	0xC18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PHVIL15	-						
access	R/W-0	U-0						
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							
access	U-0							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15	PHVIL15	PH15 input low threshold configuration

bit	name	functional description
		0: Input low threshold is normal 1: Input low threshold is reduced
14:0	--	RFU: **Reserved, read as 0**

9 Cross Power Domain Interface (CDIF)

9.1 Introduction

The CPU accesses the modules under the VAO power domain through the CDIF interface

CDIF only supports word access, not DMA access. Before accessing the registers under the VAO, the INTF_OEN and INTF_IEN registers must be set and the interface frequency divider register must be configured. The highest frequency for CDIF interface access is recommended not to exceed 8Mhz. For example, when APBCLK is 32Mhz, it is recommended to configure the PRSC register to achieve frequency division by 4.

After the register access is over, it is recommended to clear INTF_OEN, otherwise it will bring more than ten uA of extra current.

9.2 Bus Address

4KB address space is allocated on APB to accommodate all module registers under VAO, including RTCB, XTLF, etc., see the Chapter 8 VAO Domain.

This address segment does not support DMA operations.

9.3 Clock and Reset

The CDIF module does not have an independent RMU reset register.

9.4 Register (CPU Domain)

Offset	Name	Symbol
CDIF(Base address: 0x4001E000)		
0x00	Interface Control Register	CDIF_CR
0x04	Interface Prescaler Register	CDIF_PRSC

9.4.1 Interface Control Register (CDIF_CR)

NAME	CDIF_CR							
offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-						INTF_IEN	INTF_OEN
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
1	**INTF_IEN**	Cross power domain interface VAO->CPU direction enable 1: Enable interface (CPU input enable) 0: Close the interface (CPU input is closed)
0	**INTF_OEN**	Cross power domain interface CPU->VAO direction enable 1: Enable interface 0: close the interface

9.4.2 Interface Prescaler Register (CDIF_PRSC)

NAME	INTF_PRSC							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					PRSC		
access	U-0					R/W-000		

bit	name	functional description
31:4	**--**	RFU: **Reserved, read as 0**
3:0	**PRSC**	Cross power domain timing control, configure the timing ratio relative to APBCLK 000: 1 frequency division 001: divide by 2 010: divide by 4 011: divide by 8 100: divide by 16 101: divide by 32 110: divide by 64 111: divide by 128

10 CPU

10.1 Introduction

The CPU core used by FM33LG0 is Cortex-M0, which complies with the ARMv6-M architecture and programming model; for more information, please refer to the ARM official website www.arm.com

Its basic characteristics are as follows:

- User/Privilege Mode
- VTOR (Interrupt vector table redirection)
- NVIC supports 32 external interrupts
- Data monitoring point: 1
- Hardware breakpoint: 4
- Single-cycle 32-bit hardware multiplier
- SWD debugging interface

10.1.1 CPU Config

Feature	Options	FM33LG0Config
Interrupts	1~32	32
Data endianness	little/big	little
SysTick Timer	Present or absent	Present
watchpoints	0,1,2	1
breakpoints	0,1,2,3,4	4
halting debug support	Present or absent	Present
multiplier	Fast or Small	Fast
Single-Cycle IO	Present or absent	Absent
wake-up interrupt controller(WIC)	Present or absent	Present
Vector Table Offset Register	Present or absent	Present
Unprivileged/Priviledged support	Present or absent	Present
JTAGnSW	JTAG or SWD for DAP	SWD
Memory Protection Unit	Present or absent	Present

Table 10-1 FM33LGx0xx CPU Configuration Summary Table

10.2 Core Register

List of main core registers:

Name	Description
R0-R12	General Register
MSP (R13)	Stack pointer; use MSP (Main Stack Pointer) in Handler mode, select MSP or PSP (Process Stack Pointer) through CONTROL register in Thread mode
PSP (R13)	
LR (R14)	Link register, save the return information of sub-function/function call/exception handling
PC (R15)	Program pointer
PSR	Including application program status (APSR), interrupt program status (IPSR) and program execution status (EPSR)
PRIMASK	PRIMASK is used to shield all interrupt responses of the specified priority and below
CONTROL	Set the stack pointer used in Thread mode

Table 10-2 Cortex-M0 Core Register Summary Table

Refer to the ARMv6-M Architecture Reference Manual for detailed register definitions.

10.3 Exceptions and Interrupts

The exception and interrupt management of the kernel is completed through the NVIC. The programmable management register of the NVIC is located in the SCS space of the PPB bus. The NVIC has the following characteristics:

- Support 32 external interrupts, 5 internal exceptions
- 1 NMI interrupt
- Support interrupt nesting
- Vectorized exception entry
- Interrupt mask

After the processor core receives an exception request, it first pushes the core registers R0~R3, R12, R14, PC, and xPSR onto the stack. The link register LR (R14) is updated to the special value (EXC_RETURN) used when returning from the exception, and then the exception handler is located according to the exception vector table and executed. Note that registers that are not automatically pushed onto the stack during exception handling must be saved and restored by software.

10.3.1 Interrupt Vector Table

Position	Priority	Priority type	Acronym	Description	Address
0	-	-	MSP Initial value	Main stack pointer initialization address	0x0000_0000
1	-3	fixed	Reset	Reset vector	0x0000_0004
2	-2	fixed	NMI	WKUPx interrupt Low power mode error interrupt	0x0000_0008
3	-1	fixed	HardFault	HardFault interrupt vector	0x0000_000C
4-10	-	-	-	Reserved	0x0000_0010~0x0000_002B
11	3	settable	SVC	SVCall system service request	0x0000_002C
12-13	-	-	-	Reserved	0x0000_0030~0x0000_0037
14	5	settable	PendSV	Suspend system service request	0x0000_0038
15	6	settable	Systick	Internal timer interrupt vector	0x0000_003C

Position	Priority	Priority type	Acronym	Description	Address
16	7	settable	WDT	Window watchdog or independent watchdog interrupt	0x0000_0040
17	8	settable	SVD	Power monitoring alarm interruption	0x0000_0044
18	9	settable	RTC_A/RTC_B	Real-time clock interrupt	0x0000_0048
19	10	settable	FLASH	NVMIF interrupt	0x0000_004C
20	11	settable	FDET	XTLF or XTHF vibration stop detection interrupted System clock selection error interrupt	0x0000_0050
21	12	settable	ADC	ADC conversion complete interrupt	0x0000_0054
22	13	settable	DAC	DAC interrupt	0x0000_0058
23	14	settable	SPI0	SPI interrupt	0x0000_005C
24	15	settable	SPI1		0x0000_0060
25	16	settable	SPI2		0x0000_0064
26	17	settable	UART0	UART interrupt	0x0000_0068
27	18	settable	UART1		0x0000_006C
28	19	settable	UART3		0x0000_0070
29	20	settable	UART4		0x0000_0074
30	21	settable	UART5		0x0000_0078
31	22	settable	U7816	U7816 interrupt	0x0000_007C
32	23	settable	LPUARTx	LPUART0/1/2 interrupt	0x0000_0080
33	24	settable	I2C	I2C interrupt	0x0000_0084
34	25	settable	CCL	Clock calibration interrupt	0x0000_0088
35	26	settable	AES	AES interrupt	0x0000_008C
36	27	settable	LPTIM	LPTIM16 or LPTIM32 interrupt	0x0000_0090
37	28	settable	DMA	DMA interrupt	0x0000_0094
38	29	settable	WKUPx	WKUP pin interrupt	0x0000_0098
39	30	settable	LUT	LUT interrupt	0x0000_009C
40	31	settable	BSTIM	BSTIM16 or BSTIM32 interrupt	0x0000_00A0
41	32	settable	COMPx	COMPx interrupt	0x0000_00A4
42	33	settable	GPT0,1	General-purpose timer 0,1 interrupt	0x0000_00A8
43	34	settable	GPT2	General-purpose timer 2 interrupt	0x0000_00AC
44	35	settable	ATIM	Advanced timer interrupt	0x0000_00B0

Position	Priority	Priority type	Acronym	Description	Address
45	36	settable	VREF1p2 /VREF_V REG	1.2V internal reference voltage establishment interrupted VREF_VREG interrupt	0x0000_00B4
46	37	settable	EXTI	External pin interrupt	0x0000_00B8
47	38	settable	CAN	CAN2.0 interrupt	0x0000_00BC

Table 10-3 FM33LG0xx Interrupt Vector Table

Among them, WKUPx interrupt can be connected to NMI or 38# entry. The interrupt entry address is selected through the PINWKEN.WKISEL register of the GPIO module. When configured as 38# entry, the WKUPx interrupt can be shielded through PRIMASK. After waking up, the CPU does not enter the interrupt service routine, but continues to execute downward from the sleep instruction.

10.3.2 Interrupt Priority

The processor supports 3 fixed highest priority levels and 4 programmable priority levels. When two exceptions of the same priority occur at the same time, the exception with the smaller exception number will be executed first.

10.3.3 Error Handling

The processor only supports one hardware error handling method: HardFault exception. HardFault priority is -1, and only NMI can preempt it.

The triggering reasons of HardFault include the following situations:

Error type	Error condition
Memory related	Bus error. Bus error caused by using illegal address in bus transmission.
	Attempt to execute program in XN area
Program error	Execute undefined command
	Attempt to switch to ARM state
	Attempt to perform unaligned memory access Execute SVC instructions in higher priority exception handling The value of EXC_RETURN is illegal when executing an exception return Attempt to execute BKPT instruction when debugging is not enabled

The HardFault trigger cause of FM33LG0 can be queried through registers to help software developers locate the cause of the error.

10.3.4 Lockup

When another HardFault occurs during HardFault processing, or a HardFault occurs during NMI processing, the processor will enter the locked state (stop execution) and output the LOCKUP signal. At this time, the chip will automatically reset the processor core Instead of waiting for the watchdog to overflow.

10.4 MPU

MPU complies with ARMv6-M Protected Memory System Architecture (PMSAv6). MPU supports the following features:

- Support 8 programmable storage areas (Region)
- Support background area features
- Overlapping areas, support 0~7 area priority (0 is the lowest priority, 7 is the highest priority)
- Access control
- Output memory attributes
- Incorrect permission access will be blocked and a HardFault will be triggered

For simple systems without embedded OS, MPU can be programmed as static configuration, examples of common functions:

- Set part of the RAM area as read-only to avoid accidental destruction of important data
- Set the bottom space of the stack as inaccessible to detect stack overflow
- Set the SRAM area to XN (not executable) to avoid code injection attacks

For systems with embedded OS, the OS can dynamically configure the MPU during each context switch (Context Switch), so that each application task has a different MPU configuration, and realizes more complex authority management:

- Define SRAM access permissions to ensure that application tasks can only access their own stack space, avoid stack leakage and damage other stacks
- Define memory access permissions so that application tasks can only access limited peripherals
- Restrict application tasks to only access their own data pool (literal pool) or program code

10.4.1 MPU Register

The MPU related registers are located in the system control space (SCS) and include the following

registers. Note that the MPU registers only support word access:

Address	Register	Function
0xE000ED90	MPU Type Register (TYPE)	Read only, provide MPU related query information
0xE000ED94	MPU Control Register (CTRL)	MPU enable/disable and background region control
0xE000ED98	MPU Region Number Register (RNR)	Select the MPU region to be configured
0xE000ED9C	MPU Base Address Register (RBAR)	Define the base address of the MPU region
0xE000EDA0	MPU Region Attribute and Size Register (RASR)	Define the attributes and size of the MPU region

10.4.1.1 MPU Type Register (TYPE)

Name: MPU_TYPE			
Address: 0xE000ED90			
Field	Description	Reset	Access
31:24	-		
23:16	IREGION	0x00	R
15:8	DREGION	0x08	R
7:1	-		
0	I	0	R

10.4.1.2 MPU Control Register (CTRL)

Name: MPU_CTRL			
Address: 0xE000ED94			
Field	Description	Reset	Access
31:3	-		
2	PRIVDEFENA The default memory map of the privilege level is enabled. When it is 1 and the MPU is enabled, the privileged access will use the default memory map as the background area; if this bit is 0, the background area is forbidden. Access to the energy zone will trigger HardFault	0	R/W

1	HFNMIENA 1-MPU is also enabled during HardFault and NMI processing 0-MPU is not enabled during HardFault and NMI processing	0	R/W
0	ENABLE 1-Enable MPU 0-disable MPU	0	R/W

10.4.1.3 MPU Region Number Register (RNR)

Name: MPU_RNR			
Address: 0xE000ED98			
Field	Description	Reset	Access
31:8	-		
7:0	REGION Before setting each area, write to this register to select the area to be programmed; since the processor only supports 8 Regions, write values other than 0-7 should be avoided	-	R/W

10.4.1.4 MPU Base Address Register (RBAR)

Name: MPU_RBAR			
Address: 0xE000ED9C			
Field	Description	Reset	Access
31:8	ADDR Base address of the area	-	R/W
7:5	-		
4	VALID 1 – The REGION number written in Bit[3:0] will take effect during the base address programming and will cover the lowest 4bit in MPU_RNR at the same time 0-The value in the MPU_RNR register will take effect when the base address is programmed	-	R/W
3:0	REGION	-	R/W

	If VALID=1 when writing, MPU_RNR[3:0] will be overwritten; when reading, MPU_RNR[3:0] will be returned		

10.4.1.5 MPU Region Attribute and Size Register (RASR)

Name: MPU_RASR			
Address: 0xE000EDA0			
Field	Description	Reset	Access
31:29	-	-	
28	XN Forbidden to fetch 1-The CPU is prohibited from fetching instructions from this area, which will trigger HardFault 0-Allow to fetch instructions from this area	0	R/W
27	-		
26:24	AP Access control	000	R/W
23:22	-		
21:19	TEX Type expansion field, only supports 000, other values are reserved	000	R/W
18	S Shareable	-	R/W
17	C Cacheable	-	R/W
16	B Bufferable	-	R/W
15:8	SRD Sub-regions are prohibited; each region is divided into 8 sub-regions equally by MPU, and 8bit SRD is used to individually enable or disable each sub-region. 1-Prohibit the corresponding sub-area 0-enable the corresponding sub-area Bit8 controls the sub-area of the address in the	0x00	R/W

	lowest area, Bit15 controls the sub-area of the highest address in the area		
7:6	-		
5:1	SIZE Area size setting, the allowed value range is 7-31, and the corresponding area size is 2 (SIZE+1) bytes, and settings smaller than 7 are not supported, because the minimum area size is 256bytes (32bytes*8)	-	R/W
0	ENABLE Zone enable 0-prohibit this area 1-Enable this area (on the premise of enabling MPU)	0	R/W

TEX (Type Expansion), S (Shareable), C (Cachable), B (Bufferable) represent the attributes of the memory area. These attributes will be output to the bus every time data and instructions are accessed for write-buffer or cache and other bus devices use.

The following table defines the coding rules for zone attributes and access control:

TEX[a]	C	B	Memory type	Description, or Normal region cacheability	Shareable?
000	0	0	Strongly-ordered	Strongly ordered	Shareable
000	0	1	Device	Shared device	Shareable
000	1	0	Normal	Outer and inner write-through, no write allocate	S bit[b]
000	1	1	Normal	Outer and inner write-back, no write allocate	S bit[b]

a. All other combinations of TEX, C, and B are reserved.
b. Shareable if the S bit is set to 1, Non-shareable if the S bit is set to 0.

AP[2:0]	Privileged access	Unprivileged access	Notes
000	No access	No access	Any access generates a permission fault
001	Read and write	No access	Privileged access only
010	Read and write	Read only	Any unprivileged write generates a permission fault
011	Read and write	Read and write	Full access
100	UNPREDICTABLE	UNPREDICTABLE	Reserved
101	Read-only	No access	Privileged read-only
110	Read-only	Read-only	Privileged or unprivileged read-only
111	Read-only	Read-only	Privileged or unprivileged read-only

10.5 Debugging Features

CPU supports the following debugging features

- Pause, resume and single-step execution of the program
- Access to core registers and special registers
- 4 hardware breakpoints
- Software breakpoints (unlimited number of BKPT instructions)
- A data monitoring point
- Dynamic non-intrusive memory access (no need to stop the processor)
- SWD interface

The debugging feature of Cortex-M0 is based on the ARM CoreSight debugging architecture. For details, please refer to "CoreSight Technology System Design Guide" and "ARM Debug Interface Architecture Specification ADIv5.0 to ADIv5.2".

10.5.1 Debug Function Pins

FM33LG0 uses the SWD debug interface, in user mode, only 4 wires (NRST, GND, SWIO, SWCLK) are needed to realize the debugging function. The 2-wire debug pin can be multiplexed as GPIO, and its function is selected and configured by software.

The NRST pin is used to reset the chip. Through the cooperation of NRST and SWD, the Halt can be at the first instruction after the chip is reset.

Refer to the I/O control chapter for the multiplexing description of the debug function pins.

10.5.2 Watchdog Control in Debug State

The watchdog can remain enabled or disabled in debug mode. Software or Debugger can configure the watchdog to open or close through the DBG_CR register.

10.5.3 DEBUG Reset

The DEBUG part of the kernel is only affected by power-on and power-on reset. Other system reset sources such as watchdog, pin reset, software reset, etc. will not reset the DAP circuit. In this way, the CPU core can be in the reset state through the pin reset after the chip is powered on, but the debugger can still normally establish communication with the DAP and set breakpoints. After the reset is released, the CPU can immediately enter the debug mode.

It is recommended that the debugger connect to the core when the system is reset (set a breakpoint at the reset vector).

11 Reset Management Unit (RMU)

11.1 Introduction

Main features:

- Support multiple reset sources, such as POR, PDR, WDG reset, software reset, pin reset, etc
- BOR monitoring main power supply
- BOR power-on reset typical release voltage is 1.65V
- BOR power-down reset with programmable reset voltage: 1.8/2.0/2.2/2.4V, can be disabled
- by software
- Ultra-low-power PDR with programmable reset voltage: 1.4/1.45/1.5/1.55V
- Filtering and delay function to ensure robust system reset

During system reset phase, all registers are restored to their default values (except RTC internal registers);

After exiting system reset, the MCU uses internal high speed RC oscillator (RCHF, default frequency of 8MHz) as the system clock by default.

11.2 Block Diagram

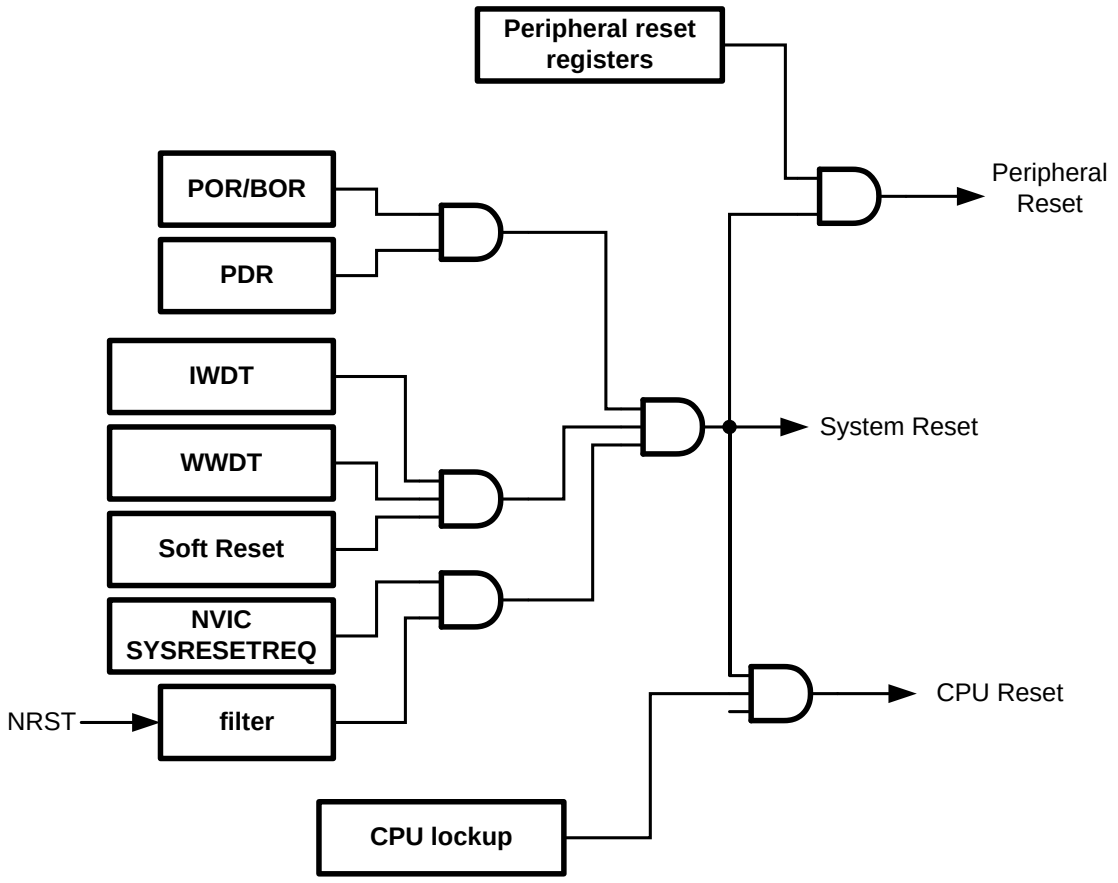


Figure 11-1 Reset Block Diagram

11.3 Power-on Reset and Power-down Reset

The power-on reset circuit monitors the VDD power supply. In order to prevent power jitter and ensure the anti-interference ability of the power-on reset circuit, the power-on reset signal is filtered and delayed.

VPOR threshold is fixed at 1.6V (typical value)

VPDR power-off reset threshold is software configurable, 4 levels in total: 1.4V, 1.45V, 1.5V, 1.55V

VBOR power-off reset threshold is software configurable, 4 levels in total: 1.8V, 2.0V, 2.2V, 2.4V

After the chip power-on reset is released, the BOR power-off reset is disabled by default and can be turned on by software.

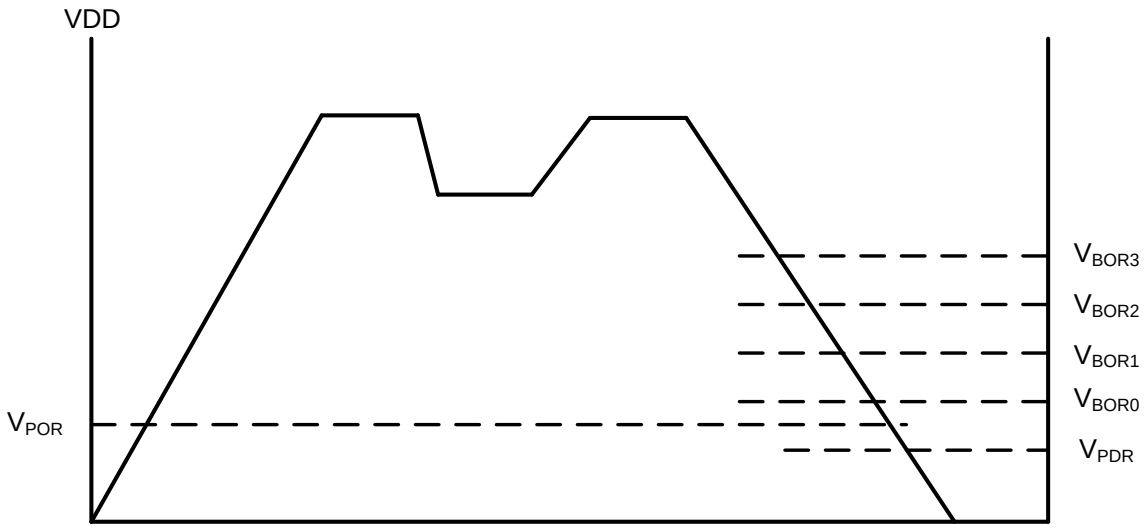


Figure 11-2 Power-on/Power-down Reset Diagram

In order to ensure a safe power-on reset, it is recommended that the power-on speed cannot be faster than 2us/V, that is, the power supply rises from 0 to 5V, and the time cannot be shorter than 10us.

Note that if the software turns off BOR, the chip is still running when the power supply voltage is between VBOR0 and VPDR. Since the minimum voltage for the chip to run at full speed is 1.65V, in order to ensure the reliable operation of the CPU, it is recommended to use SVD to monitor the power supply. When the power supply voltage is low, the program should actively go to sleep.

11.4 Software Reset

Soft reset is initiated by the CPU writing 0x5C5C_AABB to the SOFTRST register.

11.5 NRST Pin Reset

Pulling down the NRST pin can generate a global reset. In order to enhance the anti-interference ability, the NRST pin has a digital filtering function. In order to ensure a reliable reset, it is recommended to pull down the NRST and keep the low level for more than 2ms.

11.6 Register

Offset	Name	Symbol
RMU(Base address: 0x40002800)		
0x00	PDR Control Register	RMU_PDRCR
0x04	BOR Control Register	RMU_BORCR
0x08	Lockup Reset Control Register	RMU_LKPCR
0x0C	Software Reset Register	RMU_SOFTRST
0x10	Reset Flag Register	RMU_RSTFR
0x14	Peripheral Reset Enable Register	RMU_PRSTEN
0x18	AHB Peripherals Reset Register	RMU_AHBRSTCR
0x1C	APB Peripherals Reset Register1	RMU_APBRSTCR1
0x20	APB Peripherals Reset Register2	RMU_APBRSTCR2

11.6.1 PDR Control Register (RMU_PDRCR)

NAME	RMU_PDRCR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					CFG		EN
access	U-0					R/W-00		R/W-1

bit	name	functional description
31:3	--	RFU: **Reserved, read as 0**
2:1	**CFG**	Ultra-low-power PDR threshold configuration 00—1.4V 01—1.45V 10—1.5V 11—1.55V
0	**EN**	Ultra-low-power PDR enable 0：disable 1：enable

11.6.2 BOR Control Register (RMU_BORCR)

NAME	RMU_BORCR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				BOR_PDRCFG		-	BOR_ENB
access	U-0				RW-00		U-0	R/W-1

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3:2	**BOR_PDRCFG**	BOR power-down reset voltage configuration 00—1.8V 01—2.0V 10—2.2V 11—2.4V
1	--	RFU: **Reserved, read as 0**
0	**BOR_ENB**	BOR power-off reset enable, it is turned off by default after power-on 0: Enable BOR power-off reset 1: Turn off BOR power-off reset

11.6.3 Lockup Reset Control Register (RMU_LKPCR)

NAME	RMU_LKPCR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							

access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						EN	-
access	U-0						R/W-0	U-0

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
1	**EN**	LOCKUP reset enable 1: Enable CPU LOCKUP reset 0: Mask CPU LOCKUP reset
0	--	RFU: **Reserved, read as 0**

11.6.4 Software Reset Register (RMU_SOFTRST)

NAME	RMU_SOFTRST							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	SOFTRST							
access	W							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	SOFTRST							
access	W							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	SOFTRST							
access	W							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SOFTRST							
access	W							

bit	name	functional description
31:0	**SOFTRST**	The software writes 0x5C5C_AABB to trigger a global reset

11.6.5 Reset Flag Register (RMU_RSTFR)

NAME	RMU_RSTFR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

name	-			MDF_FLAG	NRSTN_FLAG	PRCN_FLAG	PORN_FLAG	PDRN_FLAG
access	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		SOFTN_FLAG	IWDTN_FLAG	-	WWDTN_FLAG	LKUPN_FLAG	NVICN_FLAG
access	U-0		R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:13	--	RFU: **Reserved, read as 0**
12	MDF_FLAG	Mode diagnosis timeout reset flag, high effective, software write 1 to clear
11	NRSTN_FLAG	NRST pin reset flag, high level effective, write 1 to clear
10	PRCN_FLAG	TESTN pin reset flag, high level effective, write 1 to clear
9	PORN_FLAG	Power-on reset flag, high level effective, write 1 to clear
8	PDRN_FLAG	Power-down reset flag, high level effective, write 1 to clear
7:6	--	RFU: **Reserved, read as 0**
5	SOFTN_FLAG	Software reset flag, high level effective, write 1 to clear
4	IWDTN_FLAG	IWDT reset flag, high level effective, write 1 to clear
3	--	RFU: **Reserved, read as 0**
2	WWDTN_FLAG	WWDT reset flag, high level effective, write 1 to clear
1	LKUPN_FLAG	LOOKUP reset flag, high level effective, write 1 to clear
0	NVICN_FLAG	NVIC reset flag, high level effective, write 1 to clear

11.6.6 Peripheral Reset Enable Register (RMU_PRSTEN)

NAME	**RMU_PRSTEN**							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	PERHRSTEN[31:24]							
access	W							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	PERHRSTEN[23:16]							
access	W							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PERHRSTEN[15:8]							
access	W							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PERHRSTEN[7:0]							
access	W							

bit	name	functional description
31:0	PERHRSTEN	Peripheral module reset enable, 32bit virtual register, write only The software writes 0x1357_9BDF to this address to enable the peripheral reset function, and then each module can be reset through the peripheral module reset register The software writes any other data to this address, the peripheral reset function will be disabled

11.6.7 AHB Peripherals Reset Register (RMU_AHBRSTCR)

NAME	RMU_AHBRSTCR							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							DMARST
access	U-0							R/W-0

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
0	DMARST	DMA reset, software write 1 reset, write 0 to release reset 0: No reset 1: Reset

11.6.8 APB Peripherals Reset Register1 (RMU_APBRSTCR1)

NAME	RMU_APBRSTCR1							
Offset	0x1C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	UART5RST	UART4RST	UART3RST	-	UART1RST	UART0RST	UARTIRRST	U7816RST
access	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

name	GPT2RST	GPT1RST	GPT0RST	ATIMRST	BT32RST	BT16RST	-	-
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	SPI2RST	SPI1RST	SPI0RST	-	I2CRST	LPUART2RST	LPUART1RST	LPUART0RST
access	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	VREFRST	PGLRST	LCDRST	DACRST	OPARST	LPT16RST	LPT32RST
access	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31	UART5RST	UART5 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
30	UART4RST	UART4 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
29	UART3RST	UART3 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
28	--	RFU: **Reserved, read as 0**
27	UART1RST	UART1 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
26	UART0RST	UART0 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
25	UARTIRRST	UART infrared modulation module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
24	U7816RST	U7816 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
23	GPT2RST	GPTIM2 module reset, software write 1 to reset, write 0 to cancel reset

bit	name	functional description
		0: Do not reset 1: Reset
22	GPT1RST	GPTIM1 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
21	GPT0RST	GPTIM0 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
20	ATIMRST	ATIM module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
19	BT32RST	BSTIM32 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
18	BT16RST	BSTIM16 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
17:16	--	RFU: **Reserved, read as 0**
15	SPI2RST	SPI2 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
14	SPI1RST	SPI1 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
13	SPI0RST	SPI0 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
12	--	RFU: **Reserved, read as 0**
11	I2CRST	I2C module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
10	LPUART2RST	LPUART2 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset

bit	name	functional description
		1: Reset
9	LPUART1RST	LPUART1 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
8	LPUART0RST	LPUART0 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
7	--	RFU: **Reserved, read as 0**
6	VREFRST	VREF1p2 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
5	PGLRST	PGL module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
4	LCDRST	LCD module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
3	DACRST	DAC module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
2	OPARST	OPA module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
1	LPT16RST	LPTIM16 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
0	LPT32RST	LPTIM32 module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset

11.6.9 APB Peripherals Reset Register2 (RMU_APBRSTCR2)

NAME	RMU_APBRSTCR2							
Offset	0x20							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							ADCCRST
access	U-0							R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	ADCRST	-				AESRST	CRCRST	RNGRST
access	R/W-0	U-0				R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				DIVASRST	CANRST	SVDRST	COMPRST
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:25	--	RFU: **Reserved, read as 0**
24	ADCCRST	ADC controller reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
23	ADCRST	ADC module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
22:19	--	RFU: **Reserved, read as 0**
18	AESRST	AES module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
17	CRCRST	CEC module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
16	RNGRST	RNG module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
15:4	--	RFU: **Reserved, read as 0**

bit	name	functional description
3	DIVASRST	DIVAS module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
2	CANRST	CAN module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
1	SVDRST	SVD module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset
0	COMPRST	COMP module reset, software write 1 to reset, write 0 to cancel reset 0: Do not reset 1: Reset

12 Independent Watchdog (IWDT)

12.1 Introduction

The independent watchdog is used to monitor the operation of the system. If the CPU is running abnormally and the dog cannot be cleared regularly, the watchdog will generate a global reset signal after overflow and restart the system to avoid system lockup. The independent watchdog is started by software after the chip is powered on, and cannot be turned off after it is started until the chip is reset.

In order to facilitate debugging, IWDT will stop running under the following conditions:

- When the chip is in debugging mode, the software can suspend IWDT during debugging by configuring the DBG_CR register
- When IWDTSLP in OPTBYTES is valid, the software can suspend IWDT counting in sleep mode

The core of IWDT is a 12-bit up counter, which starts to increment from 0 after reset, and triggers IWDT reset after counting to 0xFFF. IWDT reset is a global reset.

IWDT uses LSCLK to work, combined with a stop-vibration detection circuit, to ensure that the XTLF low-frequency crystal oscillator will not stop running when it stops. IWDT has a prescaler divided by 128, and the counter length is 12bit.

IWDT supports programmable window function, the software can only clear the dog in the allowed window, and clearing the dog outside the window will trigger the IWDT reset.

12.2 Block Diagram

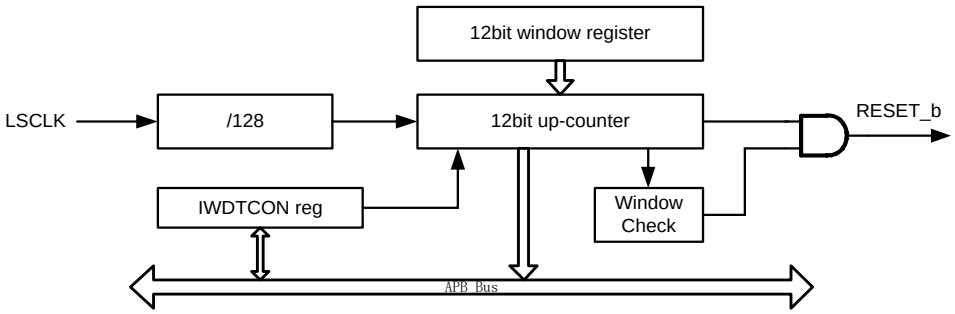


Figure 12-1 IWDT Block Diagram

12.3 IWDT Functional Description

The watchdog should use a shorter overflow period when the CPU is running normally, while in low power modes such as SLEEP/DEEPSLEEP, the watchdog should use a longer overflow period in order to keep the chip in low power mode for as long as possible.

To fullfile different application requirements, software can modify the IWDT's overflow period configuration in real time. To avoid unpredictable consequences of improper operation, software should follow the following procedure when updating the overflow period configuration.

- Ensure that the watchdog is running
- Feed watchdog
- Rewrite the IWDT_CFGR register to select the appropriate overflow period
- Read IWDT_CFGR and make sure it is written correctly
- The overflow period is updated and the CPU is running normally

IWDT uses LSCLK to work with an internal prescaler of 128. The overflow length of the counter after the frequency division can be configured from 1 to 4096 (a total of 8 available gears). The overflow time length calculation formula is as follows:

$tWWDT = T_{LSCLK} * 128 * OVP$

LSCLK frequency	Overflow length config	Overflow time (ms)
32768Hz	32	125
	64	250
	128	500
	256	1000
	512	2000
	1024	4000
	2048	8000
	4096	16000

Table 12-1 IWDT Overflow Periodic Table

12.4 IWDT Window Function

IWDT supports programmable clear dog window function. The IWDT_WIN register is used to define the allowed dog-clearing window. Only when the counter count value is greater than or equal to the value of IWDT_WIN, the dog-clearing operation is legal. Clearing the dog outside the window will directly initiate an IWDT reset.

After the chip is reset, IWDT_WIN is all 0, which means that the software is allowed to clear the dog at

any position by default.

The software can modify the IWDT_WIN register in real time while the IWDT is running. When the software clears the dog, it must read and confirm whether the current count value is within the allowed range of dog clearing.

When the IWDT count value enters the clear dog window, IWDT will trigger an interrupt flag register to notify the software that the current count value has entered the clear dog window.

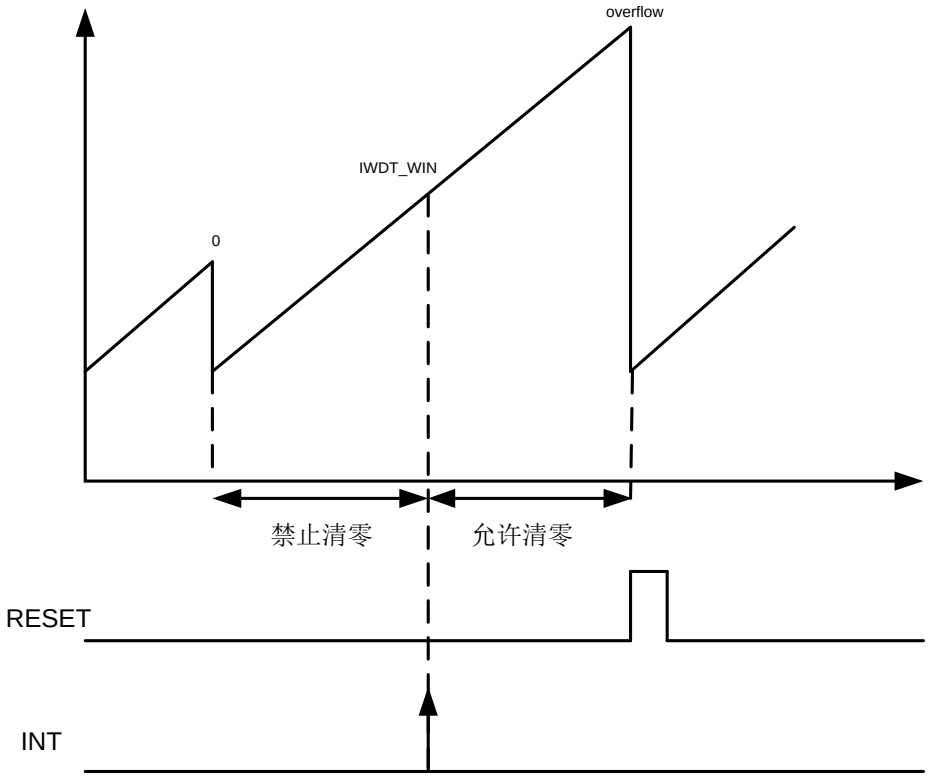


Figure 12-2 IWDT Window Diagram

12.5 IWDT Freeze

The user can configure through OPTBYTES whether to allow IWDT to freeze counting in sleep mode.

When the IWDTSLP in OPTBYTES is valid and the software sets the IWDT_FREEZE register, when the chip enters Sleep/DeepSleep mode, the IWDT count value is automatically frozen (note that IWDT is not turned off, but the count value keeps the current value and no longer increments).

12.6 Register

Offset	Name	Symbol
IWDT(Base address: 0x40011400**)**		
0x00	IWDT Service Register	IWDT_SERV
0x04	IWDT Config Register	IWDT_CR
0x08	IWDT Counter Register	IWDT_CNT
0x0C	IWDT Window Register	IWDT_WIN
0x10	IWDT Interrupt Enable Register	IWDT_IER
0x14	IWDT Interrupt Status Register	IWDT_ISR

12.6.1 IWDT Service Register (IWDT_SERV)

NAME	IWDT_SERV							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	SERV[31:24]							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	SERV[23:16]							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	SERV[15:8]							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SERV[7:0]							
access	W-0000 0000							

bit	name	functional description
31:0	**SERV**	IWDT is turned off by default after power-on reset. Software writes 0x1234_5A5A to this register and then starts IWDT. After that, IWDT cannot be turned off until the next chip reset. After IWDT is started, the software will clear the dog when writing 0x1234_5A5A to this address (IWDT Service Register, write only)

12.6.2 IWDT Config Register (IWDT_CR)

NAME	IWDT_CR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				FREEZE	-		
access	U-0				R/W-0	U-0		
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					OVP		
access	U-0					R/W-001		

bit	name	functional description
31:12	-	RFU: **Reserved, read as 0**
11	FREEZE	IWDT sleep freezes, only works when the IWDTSLP configuration in OPTBYTES is valid (Freeze in Sleep Enable) 1: Sleep/DeepSleep mode, freeze IWDT count 0: Sleep/DeepSleep mode, keep IWDT running
10:3	-	RFU: **Reserved, read as 0**
2:0	OVP	Configure IWDT overflow period 000: 125ms 001: 250ms 010: 500ms 011: 1s 100: 2s 101: 4s 110: 8s 111: 16s

12.6.3 IWDT Counter Register (IWDT_CNT)

Name	IWDT_CNT							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				CNT[11:8]			
access	U-0				R-0000			
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CNT[7:0]							

access	R-0000 0000

bit	name	functional description
31:12	-	RFU: **Reserved, read as 0**
11:0	CNT	IWDT Counter Value, read only Due to the asynchronous relationship between the counter working clock and the APB bus, the software should read the count value more than twice in a row, and it is considered a stable result when the value is the same

12.6.4 IWDT Window Register (IWDT_WIN)

NAME	IWDT_WIN							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				WIN[11:8]			
access	U-0				R/W-0000			
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	WIN[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:12	-	RFU: **Reserved, read as 0**
11:0	WIN	IWDT window register

12.6.5 IWDT Interrupt Enable Register (IWDT_IER)

NAME	IWDT_IER							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							

access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							IE
access	U-0							R/W-0

bit	name	functional description
31:1	-	RFU: **Reserved, read as 0**
0	IE	IWDT interrupt enable 0: Disabled interrupt 1: Enable interrupt

12.6.6 IWDT Interrupt Status Register (IWDT_ISR)

NAME	IWDT_ISR							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							WINF
access	U-0							R/W-0

bit	name	functional description
31:1	-	RFU: **Reserved, read as 0**
0	WINF	IWDT enters window flag, write 1 to clear 0: No interruption 1: The counting value enters the IWDT cleaning window IWDT enters the window interrupt flag, write 1 to clear

13 Window Watchdog (WWDT)

13.1 Introduction

The window watchdog is synchronous to CPU, which will reset the whole chip when program fails to feed WWDT in time.

WWDT is disabled by default after the chip is powered on. After the software starts WWDT, it cannot be disabled again until the next reset. WWDT is stopped under sleep mode.

WWDT uses APBCLK with an internal prescaler circuit to guarantee synchronous operation with CPU.

WWDT generates a system reset in the following conditions:

- Counter overflow
- Write a value other than 0xAC to the WWDT reset register (can be used to trigger a soft reset)
- Write 0xAC to the WWDT reset register during the window closed period

A warning interrupt is triggered when the counter reaches 75% of the overflow period.

13.2 Block Diagram

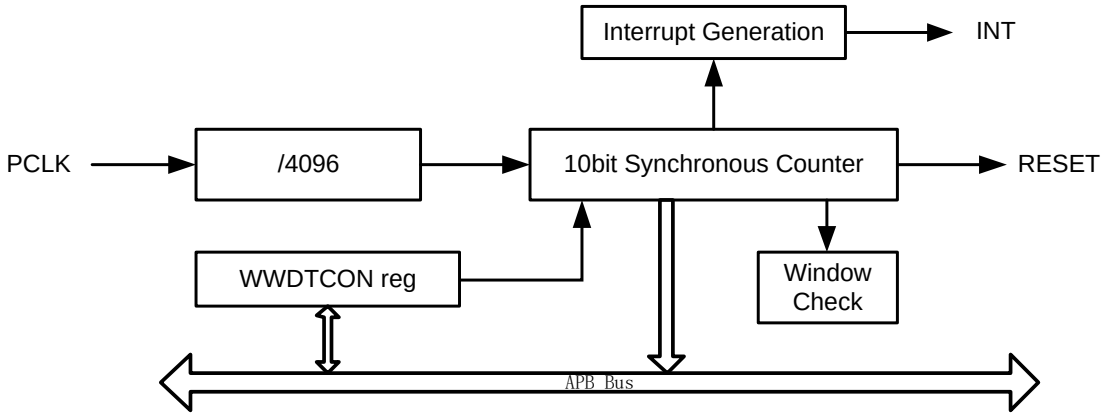


Figure 13-1 WWDT Block Diagram

13.3 WWDT Functional Description

After a chip reset, WWDT is disabled by default. WWDT is started by writing 0x5A to the WWDTCON register. If 0xAC is written to WWDTCON after WWDT is started, it will clear the counter. WWDT cannot be disabled once enabled until the next reset.

WWDT operates on APBCLK with an internal prescaler of 4096, the overflow period of the counter can be configured from 1 to 1024 (8 available steps in total). The overflow time period is calculated as follows:

tWWDT = TAPBCLK * 4096 * NCFG

The following table shows calculation examples.

APBCLK freq	Overflow period configuration	Overflow time (ms)
48MHz	1	0.085
	4	0.341
	16	1.365
	64	5.461
	128	10.922
	256	21.845
	512	43.69
	1024	87.38
32MHz	1	0.128
	4	0.512
	16	2.048
	64	8.192
	128	16.384
	256	32.768
	512	64.536
	1024	131.072
16MHz	1	0.256
	4	1.024
	16	4.096
	64	16.384
	128	32.768
	256	65.536
	512	129.072
	1024	262.144
8MHz	1	0.512
	4	2.048
	16	8.192
	64	32.768

APBCLK freq	Overflow period configuration	Overflow time (ms)
	128	65.536
	256	131.072
	512	262.144
	1024	524.288

Table 13-1 WWDT Overflow Periodic Table

The counter must be cleared in a limited window (2nd half of the period). Otherwise, a reset is generated. Software should read counter value before feeding watchdog.

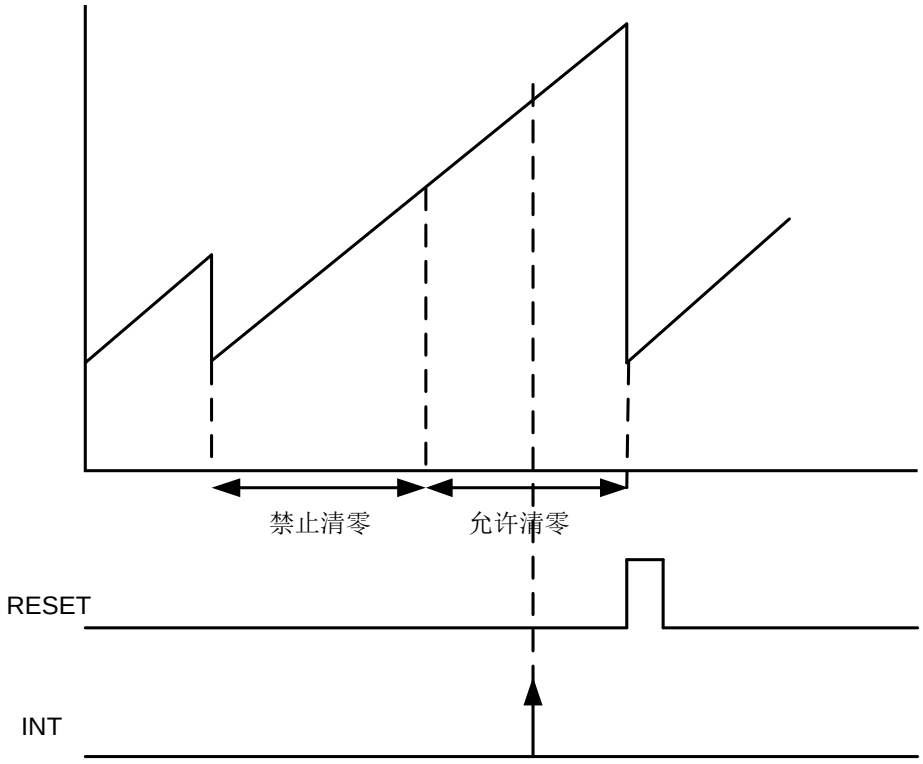


Figure 13-2 WWDT Window Watchdog Timing Diagram

13.4 Register

Offset	Name	Symbol
WWDT(Base address: 0x40011800)		
0x00	WWDT Control Register	WWDT_CR
0x04	WWDT Config Register	WWDT_CFGR
0x08	WWDT Counter Register	WWDT_CNT
0x0C	WWDT Interrupt Enable Register	WWDT_IER
0x10	WWDT Interrupt Status Register	WWDT_ISR
0x14	WWDT Prescaler Register	WWDT_PSC

13.4.1 WWDT Control Register (WWDT_CR)

NAME	WWDT_CR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CON							
access	W-0000 0000							

bit	name	functional description
31:8	-	RFU: **Reserved, read as 0**
7:0	CON	Write 0x5A to enable WWDT Write 0xAC to clear WWDT

13.4.2 WWDT Config Register (WWDT_CFGR)

Name	WWDT_CFGR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							

access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					CFG		
access	U-0					R/W-011		

bit	name	functional description
31:3	-	RFU: **Reserved, read as 0**
2:0	CFG	WWDT overflow period. The default overflow period is approximately 32ms as the system clock defaults to 8Mhz after power up 000: T_{PCLK} * 4096 * 1 001: T_{PCLK} * 4096 * 4 010: T_{PCLK} * 4096 * 16 011: T_{PCLK} * 4096 * 64 100: T_{PCLK} * 4096 * 128 101: T_{PCLK} * 4096 * 256 110: T_{PCLK} * 4096 * 512 111: T_{PCLK} * 4096 * 1024

13.4.3 WWDT Counter Register (WWDT_CNT)

NAME	WWDT_CNT							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						CNT[9:8]	
access	U-0						R-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CNT[7:0]							
access	R-0000 0000							

bit	name	functional description
31:10	-	RFU: **Reserved, read as 0**
9:0	CNT	WWDT Counter value, read only

13.4.4 WWDT Interrupt Enable Register (WWDT_IER)

Name	WWDT_IER							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							IE
access	U-0							R/W-0

bit	name	functional description
31:1	-	RFU: **Reserved, read as 0**
0	IE	WWDT Interrupt Enable 0: Disable Interrupt 1: Enable Interrupt

13.4.5 WWDT Interrupt Status Register (WWDT_ISR)

NAME	WWDT_ISR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							NOVF
access	U-0							R/W-0

bit	name	functional description
31:1	-	RFU: **Reserved, read as 0**
0	NOVF	WWDT count to 75%interrupt flag, cleared by software writing 1

bit	name	functional description
		0: No interruption 1: Interrupt Flag

13.4.6 WWDT Prescaler Register (WWDT_PSC)

NAME	WWDT_PSC							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				DIV_CNT[11:8]			
access	U-0				R-0000			
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DIV_CNT[7:0]							
access	R-0000 0000							

bit	name	functional description
31:12	-	RFU: **Reserved, read as 0**
11:0	DIV_CNT	WWDT prescaler Divider Counter, read only

14 Clock Manage Unit (CMU)

14.1 Introduction

The chip contains 32.768KHz low-frequency crystal oscillator circuit (XTLF), 4~24MHz high-frequency crystal oscillator, up to 32MHz high-frequency RC oscillator (RCHF), 32KHz low-power internal ring oscillator (RCLP), 614KHz low-power ring oscillator (RCLF), high frequency crystal oscillator (XTHF) and a phase locked loop (PLL). The clock generation module inside the chip integrates these clock sources to generate the clock required by each module.

Features:

- Multiple clock sources can be selected for the system clock
- Clocks can be switched on-the-fly during system operation
- Fail detection for XTLF
- Independent operating clocks for certain peripherals (decoupled from CPU and bus clocks)
- System frequency up to 64MHz

14.2 Clock Diagram

14.2.1 Clock Tree

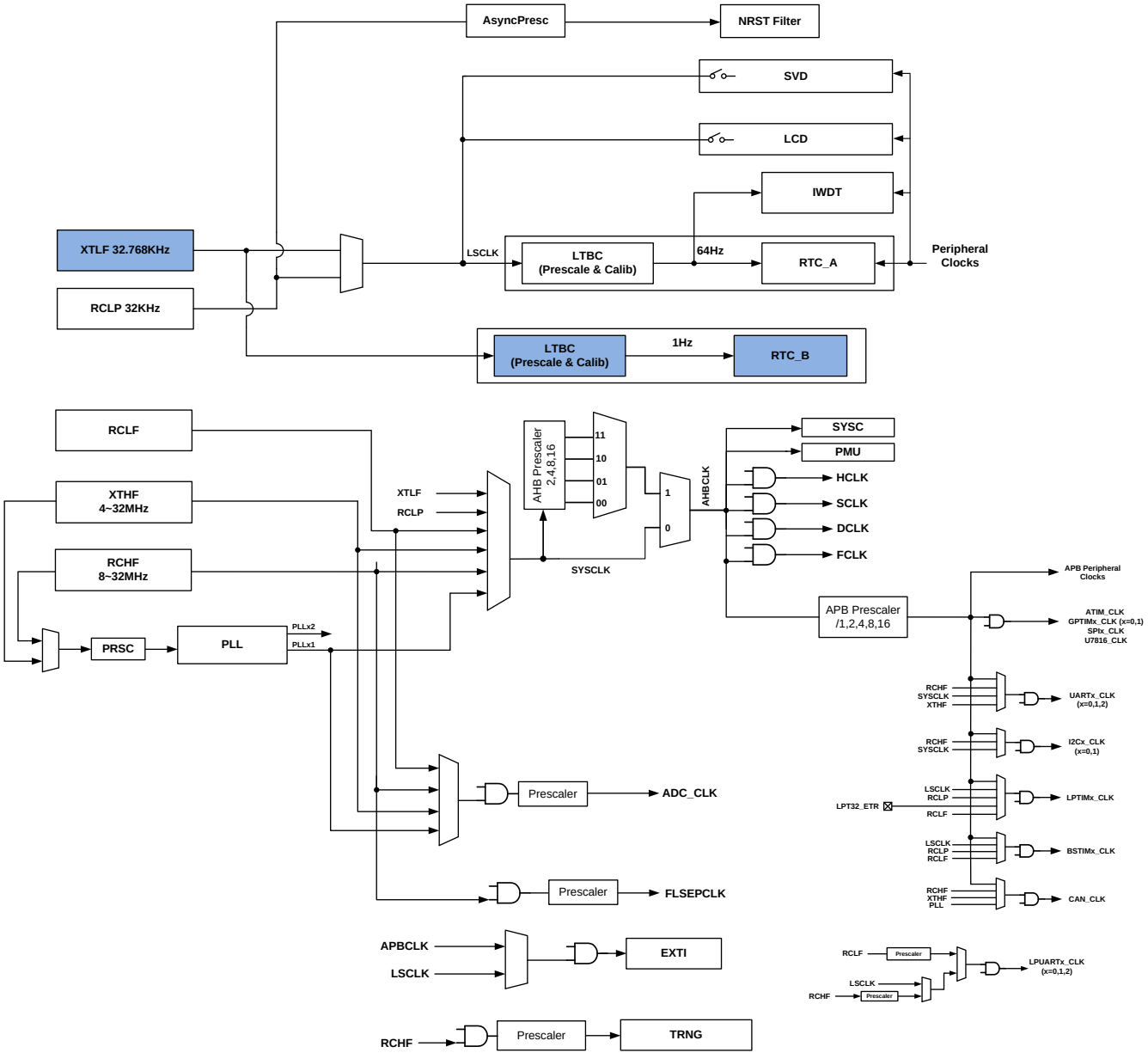


Figure 14-1 Clock Tree Diagram

The peripheral clocks (SYSCLK) can be generated from XTLF, RCHF, LPOSC, PLL, XTHF, RCMF and their divided clocks. By default, 8MHz RCHF is used as the system clock after power-up. The clock of each peripheral can be controlled separately. APBCLK is divided from AHBCLK and is used to drive low speed peripherals.

14.2.2 Introduction for SYSCLK Switching

1SYSCLK is the main clock of the system. From SYSCLK, bus clocks such as AHBCLK and APBCLK and the clock required for CPU operation can be obtained.

When SYSCLK selects any clock source, the hardware must check whether the corresponding clock source is turned on. If the clock source is not enabled (or the oscillation is stopped), the software switching operation is invalid, the SYSCLKSEL register will not be overwritten, and the clock switching will not be possible. Occurs, and the SYSCKE_IF interrupt flag register is set at the same time, and an interrupt event can be generated.

Target clock	Switching condition
RCHF	RCHF enabled
RCLF	RCMF enabled
XTHF	XTHF enabled and does not fail
PLL	PLL enabled, and: 1， If the PLL reference clock is XTHF, XTHF must be enabled and does not fail 2， If the PLL reference clock is RCHF,RCHF must be enabled
XTLF	XTLF enabled and does not fail
RCLP	RCLP enable

Table 14-1 System Clock Switching Conditions

14.2.3 Clock Security

1When using XTLF or XTHF as the system clock, or when using the PLL as the system clock and the PLL reference clock is XTHF, you must consider clock security.

Under the above conditions, if the clock stops oscillating, the stop-oscillation detection output will forcibly start RCHF and reset the SYSCLK switching logic to the RCHF channel.

- SYSCLK is XTLF, when XTLF stops oscillating, it will automatically switch SYSCLK to RCHF
- SYSCLK is XTHF, when XTHF stops oscillating, it will automatically switch SYSCLK to RCHF
- SYSCLK is PLL and PLL reference clock is selected as XTHF, when XTHF stops oscillating, it will automatically switch SYSCLK to RCHF

At the same time that the above switching occurs, the vibration stop detection circuit will generate a stop vibration interrupt, notifying the software to handle the abnormality.

14.2.4 Introduction for Main Clocks

Name	Source	Description
LSCLK	XTLF, RCLP	32KHz low frequency system clock, automatically switched to LPOSC when

Name	Source	Description
		XTLF fails Mainly used for RTC,IWDT, pin filtering, SVD,LCD
SYSCLK	RCHF, PLL, RCLP, XTLF, XTHF, RCLF	32K~64MHz,AHBCLK is the synchronous or divided version of SYSCLK
HCLK(AHBCLK)	SYSCLK	AHB bus clock driving CPU, RAM, Flash and high-speed peripherals
SCLK	SYSCLK	Core clock
DCLK	SYSCLK	Core Debug clock (This clock must be on when the debug probe is connected)
FCLK	SYSCLK	Core Free-Running clock
APBCLK	AHBCLK	The peripheral bus clock

Table 14-2 Main System Clocks Description

14.2.5 Bus Clocks and Operating Clocks for Peripherals

The bus clocks and operating clocks of some peripherals are independent from each other.

The bus clock is used for AHB or APB bus access. When the software accesses the function registers of the peripheral, the corresponding bus clock must first be enabled through the peripheral bus clock control register.

The operating clock of the peripheral is the clock actually used for peripheral operation, which may be different from APBCLK or AHBCLK. Before the peripheral module works, it needs to select the required clock source through the peripheral working clock register and open the clock gating.

For peripheral modules where the operating clock and the bus clock are unified, only the bus clock needs to be enabled for normal operation.

Module	Bus clocks	Operating clocks
Independent operating clock peripheral		
UARTx (x=0,1)	APBCLK	APBCLK
		RCHF
		SYSCLK
		XTHF
LPUARTx (x=0,1,2)	APBCLK	LSCLK
		RCHF
		RCLF
I2Cx (x=0,1)	APBCLK	APBCLK
		RCHF
		SYSCLK
		RCLF
ATIM	APBCLK	APBCLK
		PLLx2_CLK

Module	Bus clocks	Operating clocks
LPTIM32 LPTIM16	APBCLK	APBCLK
		LSCLK
		RCLP
		LPT32_ETR, LPT16_ETR
		RCLF
BSTIM32 BSTIM16	APBCLK	APBCLK
		LSCLK
		RCLP
		RCLF
ADC	APBCLK	RCLF
		XTHF
		RCHF
		PLL
NVMIF (Flash erase/program)	AHBCLK	RCHF
EXTI (PADCFG)	AHBCLK	AHBCLK
		LSCLK
TRNG	APBCLK	RCHF
IWDT	APBCLK	LSCLK
LCD	APBCLK	LSCLK
RTC	APBCLK	LSCLK
Non-independent operating clock peripherals		
PMU	AHBCLK	
DMA	AHBCLK	
GPTIMx (x=0,1,2)	APBCLK	
UARTy (y=3,4,5)	APBCLK	
SPIx (x=0,1,2)	APBCLK	
7816	APBCLK	
AES	APBCLK	
CRC	APBCLK	
WWDT	APBCLK	
OPAx	APBCLK	
COMPx (x=0,1,2)	APBCLK	

Table 14-3Peripheral Module Operating Clocks and Bus Clocks

Note that before the peripherals work, the bus clock and working clock need to be enabled. When

enabling the working clock of the peripheral module, the clock source and prescaler register should be configured first, and finally the working clock output should be enabled through the bus clock register.

14.2.6 Peripheral Clock in Sleep Mode

In Sleep/DeepSleep mode, SYSCLK is turned off, so AHBCLK and APBCLK do not work in sleep mode, and all peripherals based on AHBCLK or APBCLK stop working. However, peripherals that work independently with the bus clock can still continue to work, such as UARTx, LPUARTx, I2Cx, LPTIM32, LPTIM16, BSTIM32, and BSTIM16.

In order to allow the above peripherals to continue working in sleep mode, the software needs to ensure that the above peripherals use clocks other than SYSCLK and bus clock to work before sleep.

14.2.7 LSCLK Switching Logic

LSCLK is a low-speed clock for RTC, IWDT, SVD and LCD drivers, with a typical frequency of about 32K. The source of LSCLK is XTLF or RCLP, and the chip supports automatic switching between the two or manual switching by software.

The automatic switching function of LSCLK is configured by the LSACTS register and is only valid when XTLF is enabled. At this time, it is assumed that XTLF is the main clock and RCLP is the backup clock, which is only used to prevent XTLF from abnormally stopping oscillation. So LSCATS only works when XTLF is enabled. When XTLF stops oscillating unexpectedly, the FDET output stop signal will automatically switch LSCLK to RCLP.

When the LSCLK automatic switch is not enabled, the software can realize the manual switch of LSCLK through the LSCLKSEL register.

When LSCLK is used as the system clock (SYSCLK), it is recommended that the software enable the automatic switching function of stop vibration.

14.3 High Frequency RC Oscillators (RCHF)

14.3.1 Introduction

The typical oscillation frequency of the high-frequency RC oscillator is 8/16/24/32MHz, which can be used as the main clock of the system. When the system needs a higher main frequency, the PLL can be used to multiply the RCHF frequency to a maximum of 64Mhz. The RCHF output frequency can be adjusted. Before leaving the factory, it is adjusted to within +/-0.5% of the target frequency, and the frequency accuracy of the 8/16MHz output in the full temperature range (-40~+85°C) is less than +/-2%.

14.3.2 Software Control Description

The chip operates with the RCHF 8MHz clock by default after power-up. The hardware will automatically load the 8MHz calibration value from Flash to ensure that the 8MHz room temperature frequency error is less than +/-0.5%.

If the software needs to use other frequencies, follow the steps below:

- Rewrite CMU_RCHFCR.FSEL
- Read the adjustment value of the corresponding frequency from LDT0 (normal temperature calibration value)
- Write the frequency adjustment value into the CMU_RCHTR register to get a clock with a target frequency error of less than +/-0.5% at room temperature

If you want to obtain an accurate ring oscillator clock in high and low temperature environments, you can use the timer and XTLF clock to calibrate the RCHF in real time. For details, please contact Shanghai Fudan Microelectronics Company for technical support.

14.4 Low Frequency RC Oscillation(RCLF)

14.4.1 Introduction

RCLF is a low-power intermediate frequency ring oscillator with a typical frequency of 614.4KHz. It is used for low-power and low-speed operation of the CPU and provides a working clock for the LPUART to achieve accurate 9600 baud rate transmission and reception.

14.5 Low Power RC Oscillation(RCLP)

14.5.1 Introduction

RCLP is a low-frequency ring oscillator with polar power consumption, with a typical frequency of 32Khz and a typical power consumption of only 300nA. This ring oscillator can be used for the backup clock of XTLF, and can also be used for CPU low-speed operation and some peripherals.

14.5.2 Software User Guide

RCLP has extremely low power consumption, so it is turned on by default in ACTIVE and LPRUN modes.

If the function of XTLF stopping oscillation to automatically switch the backup clock is enabled, RCLP is forcibly turned on when XTLF stops oscillation.

When the software closes XTLF, LFDET will not output a stop signal. If the software also closes RCLP at this time, both IWDT and RTC will stop running; if the software wants to keep RCLP running in sleep mode, you need to ensure that it is enabled before sleep RCLP.

Power mode	RCLP control status description
Active/LP Run	Enable by default, software can control enable or disable
Sleep/DeepSleep	The state during sleep is determined by the enable register If RCLP is turned off, when XTLF stops vibration, it is determined whether to automatically start RCLP and output 32KHz according to the register configuration

Table14-4 RCLP State Description

14.6 High-Frequency Crystal Oscillator Circuit (XTHF)

14.6.1 Introduction

By connecting external high frequency crystal, the XTHF is able to provide a high precision high frequency clock source for the MCU. Load capacitors should be placed as close as possible to the XTHF pins, and capacitance should be chosen to suit the type of crystal.

The XTHF can accommodate crystals from 4 to 32MHz. The software can enable or disable the XTHF clock via the XTHFEN register.

14.6.2 Working Method

XTHF is turned off by default after power-on. After the power-on reset is complete, the software can open XTHF as needed. Because the crystal oscillator pins are multiplexed with GPIO, the PC2 and PC3 pins need to be configured as analog functions before XTHF is enabled by the software.

14.6.3 Fail Detection (HFNET)

FM33LG0 has an on-chip vibration stop detection circuit, which can be enabled or closed together with the XTHF circuit. After the vibration stop detection is enabled, the XTHF output can be continuously detected. When XTHF is found to stop vibration, an alarm interrupt will be generated, and an advanced timer brake signal will be generated at the same time; if XTHF is being used directly or indirectly as the system working clock (directly refers to the SYSCLK selection XTHF, indirectly means that SYSCLK is selected as PLL and PLL uses XTHF as input reference clock), the stop signal will automatically enable RCHF and switch SYSCLK to RCHF to avoid accidental stop of high-frequency crystals and system crashes.

The vibration stop detection circuit is always opened or closed at the same time as XTHF. It cannot be closed separately. Once XTHF is enabled, the vibration stop detection circuit will automatically open; when XTHF is closed, the vibration stop detection will also be automatically closed to avoid false triggering of the vibration stop alarm.

14.7 Phase Locked Loop (PLL)

14.7.1 Introduction

The input reference clock of the phase-locked loop can be divided by RCHF or XTHF. The prescaler register needs to be configured according to the frequency of RCHF and XTHF to obtain an input reference clock of 1MHz. The maximum output frequency can reach 64MHz (duty cycle 50%) and its 2 times the frequency. Before the software uses the PLL as the system clock, it is necessary to configure the input reference clock and frequency multiplication factor.

14.7.2 Software Control Description

In order to improve reliability, the following points need to be noted in the configuration.

- The software must ensure that RCHF or XTHF is enabled when the PLL input is selected
- PLL cannot be turned off when the PLL output is selected as SYSCLK
- The software should wait for PLL to lock before configuring SYSCLK as PLL output

Configure the PLL to output 64MHz and make the system operate at 64MHz frequency:

- Configure the PLLCON register, select the input clock source and output clock frequency
- Configure Flash wait to 2 cycles
- (Optional) Enable Flash prefetch instruction
- Select AHB clock as PLL output

14.8 Clock Calibration

The clock calibration circuit is a synchronous counter used to calibrate each other between different clocks without occupying timer resources.

The core of the clock calibration circuit is a 16-bit synchronous counter, prescaler and input signal source selection. As shown below.

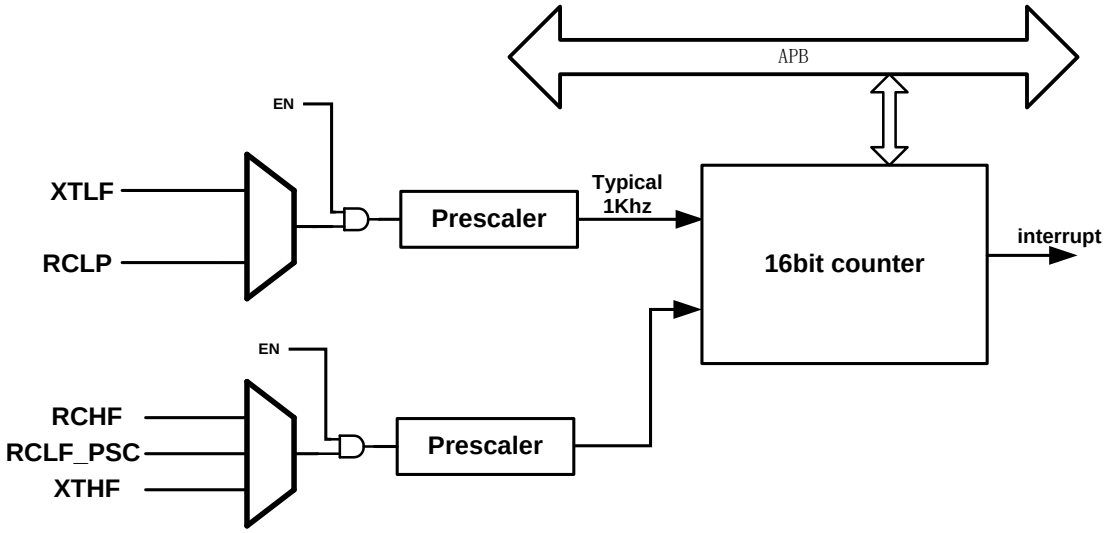


Figure 14-2 Clock Calibration Circuit Block Diagram

The main reference for clock calibration is XTLF or RCLP. After pre-frequency division, a periodic reference signal is obtained, with a typical frequency of about 1Khz. The rising edge of the reference signal triggers the counter to count and stops the counter at the next rising edge. The count value in the counter reflects the frequency relationship between the reference signal and the clock under test.

For example, the reference signal is 32768Hz divided by 32, which is 1024Hz; the measured clock is RCHF 8Mhz. If the frequency is accurate 8Mhz, the timer count value in one period of the reference signal should be around 7812. If more, you can calculate the frequency error based on the count value and calibrate the RCHF output.

If the frequency of the measured signal is high, such as RCHF 32Mhz, the measured clock should be prescaled before calibration to ensure that the count value does not overflow.

Normally, XTLF is used to calibrate RCHF and RCLF. After RCHF is calibrated, RCLP can be calibrated in reverse. XTHF can also be used to calibrate RCLP.

14.9 Clock Source in Low Power Mode

In low power consumption mode, some clock sources are forcibly turned off by the hardware, while other clock sources can still keep working. See the table below for details:

Clock source	LPRUN/Sleep/DeepSleep	Description	VBAT	Description
RCHF	X	Hardware forced shutdown	X	Power down
PLL	X		X	
XTHF	X		X	
RCLF	O	Software configuration enable or disable	X	
RCLP	O		X	
XTLF	O		O	Software configuration enable or disable

Table 14-5 Description of Clock Source Status in Low Power Mode

14.10 Clock Selection after Wake Up from Sleep

When the chip wakes up from Sleep/DeepSleep mode, the hardware automatically turns on RCHF and restores to the frequency output set before sleep; at the same time, the SYSCLKSEL register is reset to 000, the system clock is selected as RCHF, the AHBPRES register is reset to 000, and the APBPRES register remains Configure before sleep; therefore, the chip will work with the RCHF clock by default after waking up.

14.11 Register

Offset	Name	Symbol
CMU(Base address:0x40000200)		
0x00	System Clock Control Register	CMU_SYSCLKCR
0x04	RCHF Control Register	CMU_RCHCR
0x08	RCHF Trim Register	CMU_RCHFTR
0x0C	PLL Control Register	CMU_PLLCR
0x10	RCLP Control Register	CMU_RCLPCR
0x14	RCLP Trim Register	CMU_RCLPTR
0x1C	LSCLK Select Register	CMU_LSCLKSEL
0x20	XTHF Control Register	CMU_XTHFCR
0x24	RCLF Control Register	CMU_RCLFCR
0x28	RCLF Trim Register	CMU_RCLFTR
0x2C	Interrupt Enable Register	CMU_IER
0x30	Interrupt Status Register	CMU_ISR
0x34	Peripheral Bus Clock Control Register1	CMU_PCLKCR1
0x38	Peripheral Bus Clock Control Register2	CMU_PCLKCR2
0x3C	Peripheral Bus Clock Control Register3	CMU_PCLKCR3
0x40	Peripheral Bus Clock Control Register4	CMU_PCLKCR4
0x44	Peripheral Clock Config Register1	CMU_OPCCR1
0x48	Peripheral Clock Config Register2	CMU_OPCCR2
0x4C	Peripheral Clock Config Register3	CMU_OPCCR3
0x50	AHB Master Control Register	CMU_AHBMCR
0x54	Clock Calibration Control Register	CMU_CCCR
0x58	Clock Calibration Config Register	CMU_CCFR
0x5C	Clock Calibration Counter Register	CMU_CCNR
0x60	Clock Calibration Interrupt Status Register	CMU_CCISR

14.11.1 System Clock Control Register (CMU_SYSCLKCR)

NAME	CMU_SYSCLKCR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-				LSCATS	-	SLP_EN EXTI	-
access	U-0				R/W-1	U-0	R/W-1	U-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-					APBPRES		
access	U-0					R/W-000		
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	RFUI		-			AHBPRES		

access	R/W-00		U-0			R/W-011		
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	STCLKSEL		-			SYSCLKSEL		
access	R/W-00		U-0			R/W-000		

bit	name	functional description
31:28	--	RFU: **Reserved, read as 0**
27	LSCATS	LSCLK automatic switch enable 0: When detecting abnormal XTLF vibration stop, LSCLK will not be automatically switched to RCLP, software can manually switch to RCLP by writing the LSCLKSEL register 1: When detecting abnormal XTLF vibration stop, automatically enable RCLP and switch LSCLK to RCLP
26	--	RFU: **Reserved, read as 0**
25	SLP_ENEXTI	EXTI sampling settings in Sleep/DeepSleep mode 1: Enable external pin interrupt sampling in Sleep/DeepSleep mode (the sampling clock is LSCLK) 0: Disable external pin interrupt sampling in Sleep/DeepSleep mode (the EXTI interrupt will not be generated)
24:19	--	RFU: **Reserved, read as 0**
18:16	APBPRES	APB clock division selection 0xx: No frequency division 100: Divide by 2 101: Divide by 4 110: 8 frequency division 111: Divide by 16
15:11	--	RFU: **Reserved, read as 0**
10:8	AHBPRES	AHB clock division selection 0xx: No frequency division 100: Divide by 2 101: Divide by 4 110: 8 frequency division 111: Divide by 16
7:6	STCLKSEL	CPU core systick working clock selection 00: SCLK 01: LSCLK 10: RCLF 11: RFU
5:3	--	RFU: **Reserved, read as 0**
2:0	SYSCLKSEL	System clock source selection 000: RCHF 001: XTHF 010: PLL 011: RCHF

bit	name	functional description
		100: RCLF 101: XTLF 110: RCLP 111: RCHF

14.11.2 RCHF Control Register (CMU_RCHCR)

NAME	CMU_RCHCR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				FSEL			
access	U-0				R/W-0000			
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							RCHFEN
access	U-0							R/W-1

bit	name	functional description
31:20	--	RFU: **Reserved, read as 0**
19:16	FSEL	RCHF frequency selection register 0000: 8MHz 0001: 16MHz 0010: 24MHz 0011: 32MHz Others: RFU
15:1	--	RFU: **Reserved, read as 0**
0	RCHFEN	RCHF enable register 1: Enable RCHF 0: Disable RCHF

14.11.3 RCHF Trim Register (CMU_RCHFTR)

NAME	CMU_RCHFTR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							

bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RCHFTRIM							
access	R/W-1000 0000							

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:0	RCHFTRIM	RCHF frequency trim register, 8'h00 means the lowest frequency, 8'hFF means the highest frequency, the trimming range is +/-30% of the centre frequency, the trimming step is 0.25% After power on, the chip automatically reads the 8MHz trimming value from LDT0 and writes it into this register. When the software uses frequencies other than 8Mhz, it can read the trimming information from the address specified in LDT0 and write it to this register, thus ensuring the output frequency is accurate at room temperature.

14.11.4 PLL Control Register (CMU_PLLCR)

NAME	CMU_PLLCR							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-	PLLDB						
access	U-0	R/W-0011111						
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	LOCKED	REFPRSC			PLLOSEL	-	PLLINSEL	PLLEN
access	R/Dy-0	R/W-000			R/W-0	U-0	R/W-0	R/W-0

bit	name	functional description
31:23	--	RFU: **Reserved, read as 0**
22:16	PLLDB	PLL frequency multiplication ratio, output clock frequency is

bit	name	functional description
		1M*(PLLDB+1) 0011111: Output 32 times the frequency ..... 0111111: Output 64 times the frequency ***Note:*** *The setting range is 31~63, please do not use a setting higher than 64 times.*
15:8	--	RFU: **Reserved, read as 0**
7	LOCKED	PLL_L Locked Flag 1: PLL_L is locked 0: PLL_L is not locked
6:4	REFPRSC	PLL reference clock prescaler (the goal is to generate a 1MHz reference clock for the PLL) 000: No frequency division 001: Divide by 2 010: Divide by 4 011: Divide by 8 100: Divide by 12 101: Divide by 16 110: 24 frequency division 111: Divide by 32
3	PLLOSEL	PLL output selection register 0: Select PLL double output as the PLL clock in the digital circuit 1: Select PLL twice the output as the PLL clock in the digital circuit
2	--	RFU: **Reserved, read as 0**
1	PLLINSEL	PLL input selection register 0: RCHF 1: XTHF
0	PLLEN	PLL enable register 1: Enable PLL 0: Disable PLL

14.11.5 RCLP Control Register (CMU_RCLPCR)

NAME	**CMU_RCLPCR**							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							ENB
access	U-0							R/W-0

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
0	ENB	RCLP Enable Register 0: Enable RCLP 1: Close RCLP

14.11.6 RCLP Trim Register (CMU_RCLPTR)

NAME	CMU_RCLPTR							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RCLP_TRIM							
access	R/W-1000 0000							

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:0	RCLP_TRIM	RCLP adjustment value register 0000 0000: The lowest frequency 1111 1111: The highest frequency

14.11.7 LSCLK Select Register (CMU_LSCLKSEL)

NAME	CMU_LSCLKSEL							
Offset	0x1C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	LSCLKSEL							
access	R/W-01010101							

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:0	LSCLKSEL	LSCLK clock manual switch register, physical realization is only 1bit; the reset value selects RCLP; When LSCLK is XTLF, the software writes 0x55 to this address, which will switch the source of LSCLK to RCLP When LSCLK is RCLP, the software writes 0xAA to this address, which will switch the source of LSCLK to XTLF Write any other value without changing the current LSCLK; this register is only valid when LSCATS is 0

14.11.8 XTHF Control Register (CMU_XTHFCR)

NAME	CMU_XTHFCR							
Offset	0x20							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			HF_CFG				
access	U-0			R/W-00000				
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							XTHFEN
access	U-0							R/W-0

bit	name	functional description
31:13	--	RFU: **Reserved, read as 0**
12:8	HF_CFG	XTHF oscillation intensity configuration 00000: the weakest

bit	name	functional description
		.....
		11111: the strongest
		Note: *Setting range 0~0x1F*
0	XTHFEN	XTHF enable register
		0: Turn off XTHF
		1: Enable XTHF

14.11.9 RCLF Control Register (CMU_RCLFCR)

NAME	CMU_RCLFCR							
Offset	0x24							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-						RCLF_PSC	
access	U-0						R/W-00	
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							RCLF_EN
access	U-0							R/W-0

bit	name	functional description
31:18	--	RFU: **Reserved, read as 0**
17:16	RCLF_PSC	RCLF output prescaler 00: No frequency division 01: 4 frequency division 10: 8 frequency division 11: 16 frequency division
15:1	--	RFU: **Reserved, read as 0**
0	RCLF_EN	RCLF enable register 0: Turn off RCLF 1: Open RCLF

14.11.10 RCLF Trim Register (CMU_RCLFTR)

NAME	CMU_RCLFTR							
Offset	0x28							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RCLF_TRIM							
access	R/W -1000_0000							

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:0	RCLF_TRIM	RCLF frequency adjustment register, 8'h00 means the lowest frequency, 8'hFF means the highest frequency, the adjustment range is +/-30% of the center frequency, and the step size is 1% of the center frequency

14.11.11 Interrupt Enable Register (CMU_IER)

NAME	CMU_IER							
Offset	0x2C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					SYSCSE_IE	HFDET_IE	-
access	U-0					R/W-0	R/W-0	U-0

bit	name	functional description
31:3	--	RFU: **Reserved, read as 0**
2	SYSCKE_IE	SYSCLK clock selection error interrupt enable register, 1 is valid
1	HFDET_IE	XTHF high frequency detection alarm interrupt enable, 1 is valid
0	--	RFU: **Reserved, read as 0**

14.11.12 Interrupt Status Register (CMU_ISR)

NAME	CMU_ISR							
Offset	0x30							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						HFDETO	-
access	U-0						R-0	U-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					SYSCSE_IF	HFDETIF	-
access	U-0					R/W-0	R/W -0	U-0

bit	name	functional description
31:10	--	RFU: **Reserved, read as 0**
9	HFDETO	High-frequency crystal stop vibration detection module output 1: XTHF does not stop vibrating 0: XTHF stops vibration
8:3	--	RFU: **Reserved, read as 0**
2	SYSCSE_IF	SYSCLK clock selection error interrupt flag. When the selected target clock is not enabled or has stopped oscillating, clock switching is prohibited, and this flag register is set at the same time. Software writes 1 to clear.
1	HFDETIF	High-frequency vibration stop detection interrupt flag register, when XTHF stops vibration, the hardware is asynchronously set, and software writes 1 to clear it; this register can be cleared only when FFDETO is not 0
0	--	RFU: **Reserved, read as 0**

14.11.13 Peripheral Bus Clock Control Register1 (CMU_PCLKCR1)

NAME	CMU_PCLKCR1							
Offset	0x34							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			VREF1p2_PCE	OPA_PCE	ATT_PCE	COMP_PCE	SVD_PCE
access	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PAD_PCE	ANAC_PCE	IWDT_PCE	SCU_PCE	PMU_PCE	RTCA_PCE	LPT16_PCE	LPT32_PCE
access	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:13	--	RFU: **Reserved, read as 0**
12	VREF1p2_PCE	VREF1p2 module bus clock enable, high effective
11	OPA_PCE	OPA bus clock enable, high effective
10	ATT_PCE	AUTOTRIM bus clock enable, high effective
9	COMP_PCE	Comparator bus clock enable, high effective
8	SVD_PCE	SVD bus clock enable, high effective
7	PAD_PCE	GPIOs et bus clock enable, high effective
6	ANAC_PCE	Analog test buffer bus clock enable, high effective This register is used to control the bus clock of BUF4TST
5	IWDT_PCE	IWDT bus clock enable, high effective
4	SCU_PCE	SCU bus clock enable, high effective
3	PMU_PCE	PMU bus clock enable, high effective
2	RTCA_PCE	RTCA bus clock enable, high effective
1	LPT16_PCE	LPTIM16 bus clock enable, high effective
0	LPT32_PCE	LPTIM32 bus clock enable, high effective

14.11.14 Peripheral Bus Clock Control Register2 (CMU_PCLKCR2)

NAME	CMU_PCLKCR2							
Offset	0x38							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				PGL_PCE	DAC_PCE	DIVAS_PCE	ADC_PCE
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

name	WWDT_PCE	RAMBIST_PCE	FLASH_PCE	DMA_PCE	LCD_PCE	AES_PCE	TRNG_PCE	CRC_PCE
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:12	--	RFU: **Reserved, read as 0**
11	PGL_PCE	PGL bus clock enable, high effective
10	DAC_PCE	DAC bus clock enable, high effective
9	DIVAS_PCE	Hardware divider bus clock enable, high effective
8	ADC_PCE	ADC bus clock enable, high effective
7	WWDT_PCE	WWDT bus clock enable, high effective
6	RAMBIST_PCE	RAMBIST bus clock enable, high effective
5	FLASH_PCE	NVMIF (Flash erase and write controller) bus clock enable, high effective
4	DMA_PCE	DMA bus clock enable, high effective
3	LCD_PCE	LCD bus clock enable, high effective
2	AES_PCE	AES bus clock enable, high effective
1	TRNG_PCE	RNG bus clock enable, high effective
0	CRC_PCE	CRC bus clock enable, high effective

14.11.15 Peripheral Bus Clock Control Register3 (CMU_PCLKCR3)

NAME	CMU_PCLKCR3							
Offset	0x3C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							I2C_PCE
access	U-0							R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				CAN_PCE	LPUART2_PCE	LPUART1_PCE	LPUART0_PCE
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	U7816_PCE	UARTIR_PCE	UART5_PCE	UART4_PCE	UART3_PCE	-	UART1_PCE	UART0_PCE
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					SPI2_PCE	SPI1_PCE	SPI0_PCE
access	U-0					R/W-0	R/W-0	R/W-0

bit	name	functional description
31:25	-	RFU: **Reserved, read as 0**

bit	name	functional description
24	I2C_PCE	I2C bus clock enable, high effective
23:20	-	RFU: **Reserved, read as 0**
19	CAN_PCE	CAN bus clock enable, high effective
18	LPUART2_PCE	LPUART2 bus clock enable, high effective
17	LPUART1_PCE	LPUART1 bus clock enable, high effective
16	LPUART0_PCE	LPUART0 bus clock enable, high effective
15	U7816_PCE	7816 bus clock enable, high effective
14	UARTIR_PCE	UART infrared modulation working clock enable, high effective
13	UART5_PCE	UART5 bus clock enable, high effective
12	UART4_PCE	UART4 bus clock enable, high effective
11	UART3_PCE	UART3 bus clock enable, high effective
10	-	RFU: **Reserved, read as 0**
9	UART1_PCE	UART1 bus clock enable, high effective
8	UART0_PCE	UART0 bus clock enable, high effective
7:3	-	RFU: **Reserved, read as 0**
2	SPI2_PCE	SPI2 bus clock enable, high effective
1	SPI1_PCE	SPI1 bus clock enable, high effective
0	SPI0_PCE	SPI0 bus clock enable, high effective

14.11.16 Peripheral Bus Clock Control Register4 (CMU_PCLKCR4)

Name	**CMU_PCLKCR4**							
Offset	0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							BT16_PCE
access	U-0							R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			AT_PCE	GT2_PCE	GT1_PCE	GT0_PCE	BT32_PCE
access	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:9	-	RFU: **Reserved, read as 0**
8	BT16_PCE	BSTIM16 bus clock enable, high effective
7:5	-	RFU: **Reserved, read as 0**

bit	name	functional description
4	AT_PCE	ATIM bus clock enable, high effective
3	GT2_PCE	GPTIM2 bus clock enable, high effective
2	GT1_PCE	GPTIM1 bus clock enable, high effective
1	GT0_PCE	GPTIM0 bus clock enable, high effective
0	BT32_PCE	BSTIM32 bus clock enable, high effective

14.11.17 Peripheral Clock Config Register1 (CMU_OPCCR1)

NAME	CMU_OPCCR1							
Offset	0x44							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-	EXTICKS	-		LPUART1CKS		LPUART0CKS	
access	U-0	R/W-0	U-0		R/W-00		R/W-00	
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	LPUART2CKS		-				I2CCKS	
access	W-00		U-0				R/W-00	
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	BT16CKS		BT32CKS		LPT16CKS		LPT32CKS	
access	R/W-00		R/W-00		R/W-00		R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ATCKS	-	CANCKS		UART1CKS		UART0CKS	
access	R/W-0	U-0	R/W-00		R/W-00		R/W-00	

bit	name	functional description
31	-	RFU: **Reserved, read as 0**
30	EXTICKS	EXTI interrupt sampling clock selection 1: External pin interrupts use LSCLK sampling 0: Use AHBCLK to sample the external pin interrupt *It is recommended to set up when all EXTI interrupts are turned off, and then enable EXTI interrupts after the setting is completed
29	-	RFU: **Reserved, read as 0**
28	-	RFU: **Reserved, read as 0**
27:26	LPUART1CKS	LPUART1 working clock selection 00: LSCLK 01: RCHF frequency division (automatic frequency division to around 32768Hz according to RCHF gear) 10: RCLF frequency division (divide by 16 to get 38.4Khz) 11: RFU
25:24	LPUART0CKS	LPUART0 working clock selection 00: LSCLK

bit	name	functional description
		01: RCHF frequency division (automatic frequency division to around 32768Hz according to RCHF gear) 10: RCLF frequency division (divide by 16 to get 38.4Khz) 11: RFU
23:22	LPUART2CKS	LPUART2 working clock selection 00: LSCLK 01: RCHF frequency division (automatic frequency division to around 32768Hz according to RCHF gear) 10: RCLF frequency division (divide by 16 to get 38.4Khz) 11: RFU
21:18	-	RFU: **Reserved, read as 0**
17:16	I2CCKS	I2C master working clock selection 00: APBCLK 01: RCHF 10: SYSCLK 11: RCLF_PSC (set according to RCLFCR.RCLF_PSC)
15:14	BT16CKS	BSTIM16 working clock selection 00: APBCLK 01: LSCLK 10: RCLP 11: RCLF_PSC (set according to RCLFCR.RCLF_PSC)
13:12	BT32CKS	BSTIM32 working clock selection 00: APBCLK 01: LSCLK 10: RCLP 11: RCLF_PSC (set according to RCLFCR.RCLF_PSC)
11:10	LPT16CKS	LPTIM16 working clock selection 00: APBCLK 01: LSCLK 10: RCLP 11: RCLF_PSC (set according to RCLFCR.RCLF_PSC)
9:8	LPT32CKS	LPTIM32 working clock selection 00: APBCLK 01: LSCLK 10: RCLP 11: RCLF_PSC (set according to RCLFCR.RCLF_PSC)
7	ATCKS	ATIM working clock source selection register 0: APBCLK 1: PLL double frequency
6	-	RFU: **Reserved, read as 0**
5:4	CANCKS	CAN working clock selection (CAN_CLK working frequency range is 8~24Mhz) 00: RCHF

bit	name	functional description
		01: XTHF 10: PLL 11: APBCLK
3:2	UART1CKS	UART1 working clock selection 00: APBCLK 01: RCHF 10: SYSCLK 11: XTHF
1:0	UART0CKS	UART0 working clock selection 00: APBCLK 01: RCHF 10: SYSCLK 11: XTHF

14.11.18 Peripheral Clock Config Register2 (CMU_OPCCR2)

NAME	CMU_OPCCR2							
Offset	0x48							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			RNGPSC			-	
access	U-0			R/W-000			U-0	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			ADCPSC			ADCCKS	
access	U-0			R/W-000			R/W-00	

bit	name	functional description
31:13	--	RFU: **Reserved, read as 0**
12:10	RNGPSC	Random number generator clock division 000: No frequency division 001: Divide by 2 010: Divide by 4 011: Divide by 8 100: Divide by 16 101: Divide by 32 110,111: RFU
9:5	--	RFU: **Reserved, read as 0**

bit	name	functional description
4:2	ADCPSC	ADC working clock frequency division 000: No frequency division 001: Divide by 2 010: Divide by 4 011: Divide by 8 100: Divide by 16 101: Divide by 32 110,111: RFU
1:0	ADCCKS	ADC working clock selection 00: RCLF_PSC (set according to RCLFCR.RCLF_PSC) 01: RCHF 10: XTHF 11: PLL

14.11.19 Peripheral Clock Config Register3 (CMU_OPCCR3)

NAME	CMU_OPCCR3							
Offset	0x4C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	EXTICKE	FLASHCKE	LPU1CKE	LPU0CKE	-			RNGCKE
access	R/W-0	R/W-0	R/W-0	R/W-0	U-0			R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-		LPU2CKE	I2CCKE	-			ADCCKE
access	U-0		R/W-0	R/W-0	U-0			R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ATCKE	CANCKE	-				UART1CKE	UART0CKE
access	R/W-0	R/W-0	U-0				R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				BT16CKE	BT32CKE	LPT16CKE	LPT32CKE
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31	EXTICKE	GPIO external interrupt working clock enable, high effective
30	FLASHCKE	Flash erasing clock enable, high effective
29	LPU1CKE	LPUART1 working clock enable, high effective
28	LPU0CKE	LPUART0 working clock enable, high effective
27:25	--	RFU: **Reserved, read as 0**
24	RNGCKE	Random number generator working clock enable, high effective

bit	name	functional description
23:22	--	RFU: **Reserved, read as 0**
21	LPU2CKE	LPUART2 working clock enable, high effective
20	I2CCKE	I2C working clock enable, high effective
19:17	--	RFU: **Reserved, read as 0**
16	ADCCKE	ADC working clock enable, high effective
15	ATCKE	ATIM working clock enable, high effective
14	CANCKE	CAN bus module working clock enable, high effective
13:10	--	RFU: **Reserved, read as 0**
9	UART1CKE	UART1 working clock enable, high effective
8	UART0CKE	UART0 working clock enable, high effective
7:4	--	RFU: **Reserved, read as 0**
3	BT16CKE	BSTIM16 working clock enable, high effective
2	BT32CKE	BSTIM32 working clock enable, high effective
1	LPT16CKE	LPTIM16 working clock enable, high effective
0	LPT32CKE	LPTIM32 working clock enable, high effective

14.11.20 Clock Calibration Control Register (CMU_CCCR)

NAME	CMU_CCCR							
Offset	0x54							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						CCLIE	EN
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
1	CCLIE	Clock calibration interrupt enable 0: Disable interrupt 1: Allow interrupt
0	EN	Clock calibration enable 1: Start calibration 0: Close calibration and reset the calibration circuit

14.11.21 Clock Calibration Config Register (CMU_CCFR)

NAME	CMU_CCFR							
Offset	0x58							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				CALPSC		REFPSC	
access	U-0				R/W-00		R/W-10	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					CALSEL		REFSEL
access	U-0					R/W-00		R/W-0

bit	name	functional description
31:12	--	RFU: **Reserved, read as 0**
11:10	CALPSC	Calibration clock prescaler 00: No frequency division 01: divide by 2 10: 4 frequency division 11: 8 frequency division
9:8	REFPSC	Reference clock prescaler 00: Divide by 8 01: 16 frequency division 10: 32 frequency division 11: 64 frequency division
7:3	--	RFU: **Reserved, read as 0**
2:1	CALSEL	Calibration clock select 00: RFU 01: RCHF 10: RCLF_PSC 11: XTHF
0	REFSEL	Reference clock select 0: XTLF 1: RCLP

14.11.22 Clock Calibration Counter Register (CMU_CCNR)

Name	CMU_CCNR							
Offset	0x5C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCL_CNT[15:8]							
access	R-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCL_CNT[7:0]							
access	R-0000 0000							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	CCNT	Clock calibration counter After the calibration period is over, the software calculates the target clock frequency by reading the count value

14.11.23 Clock Calibration Interrupt Status Register (CMU_CCISR)

NAME	CMU_CCISR							
Offset	0x60							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							CCLIF
access	U-0							R/W-0

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
0	CCLIF	Clock calibration interrupt flag After the calibration cycle is completed, the hardware is set, and the software writes 1 to clear it.

15 Supply Voltage Detection (SVD)

15.1 Introduction

The supply voltage detection circuit is mainly used to detect the supply of the external mains power supply, to detect the under-voltage or recovery of the external mains power supply in time and to give an interrupt signal. The supply voltage detection circuit can be switched off or periodically enabled to save power.

Features:

- Detect mains power. Interrupt generated when voltage is below or above set threshold
- Under voltage detection range 1.8V~4.8V, 15 level programmable threshold steps, step interval 0.214V
- Voltage detection hysteresis window of 0.1V
- Can be switched off or operated intermittently
- Supports 1 external channel direct input for comparison with internal reference voltage source
- External channel supports 100mV window by setting reference voltage

15.2 Block Diagram

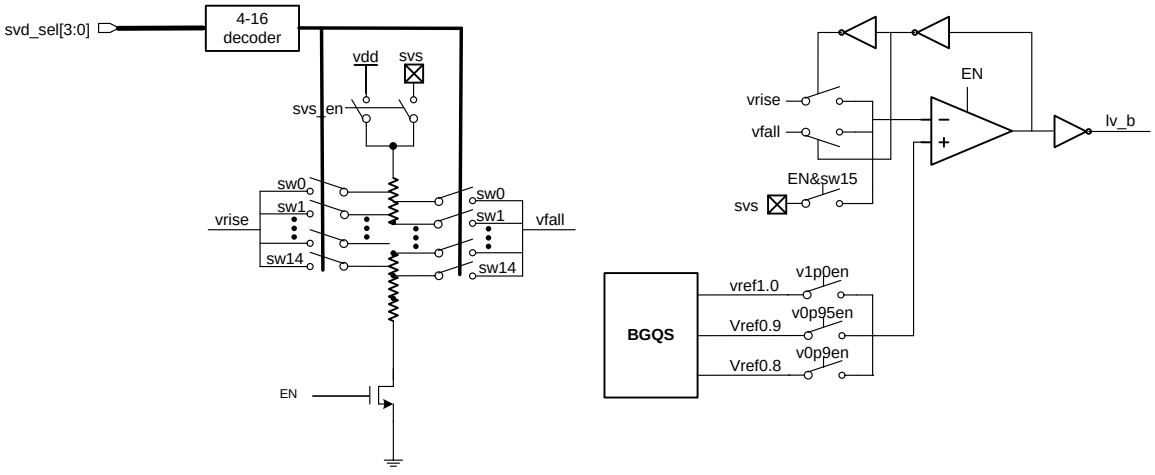


Figure 15-1 Under Voltage Detection Circuit Diagram

The SVD has 15 internal channels and one external channel. The internal channel is used for chip power detection and the external channel is used to compare the external input signal with the internal reference voltage.

SVD working sequence diagram:

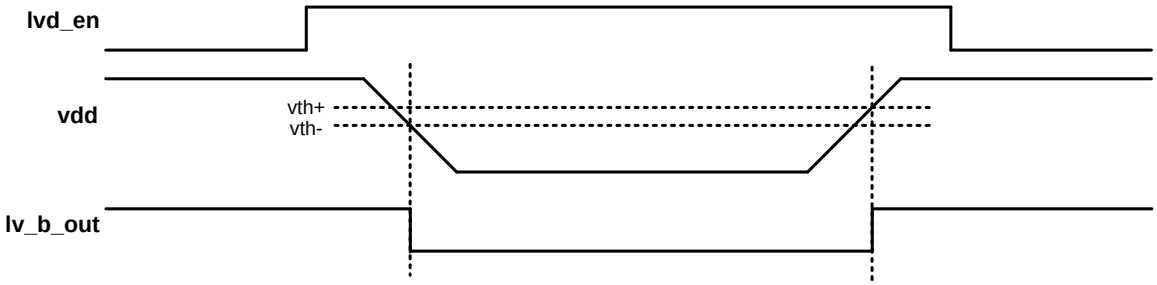


Figure 15-2 Timing of Under Voltage Detection Circuit Operating

15.3 Pin Definition

The SVD module can directly detect the chip power supply (VDD), or it can detect an external voltage signal through an SVS pin.

When detecting SVS input, you need to configure the FCR register of GPIO to 11 (analog function).

15.4 Functional Description

The power detection circuit can be used to detect the main power supply voltage and external voltage. The power supply voltage generates 15 levels of detection levels through the voltage divider resistance, the detection range is 1.8V~4.8V, and the difference of each level is 0.214V; in addition, it also supports external input detection channels. The VDD divided voltage is sent to the comparator through the multiplexer and compared with the internal reference voltage. According to the low voltage alarm threshold setting, if the detected level is lower than the reference voltage, the output voltage will jump and an undervoltage interrupt will be generated and the MCU will be notified Deal with this event in time; and when VDD recovers above the threshold (with a hysteresis window of approximately 0.1V), an undervoltage recovery interrupt will be generated.

The power detection circuit can be enabled or disabled by software configuration. In order to save power consumption, it can be divided into two modes: always-enable and intermittent work when it is enabled. When working intermittently, the turn-on time interval can be set by setting the CFGR register.

SVD has a hysteresis window of 0.1V from under-voltage to over-voltage under always-enabled conditions, but there is no hysteresis window under intermittent enable conditions; for internal channels, software can be used to cooperate, that is, a higher value can be artificially set after under-voltage interruption. To solve the window problem. As for the SVS channel, the digital circuit latches the decision result of the last intermittent window as the basis for selecting the threshold in this intermittent window, so as to realize the selection of the falling threshold and the rising threshold of the SVS. Correspondingly, the comparison reference voltage has three gears of 1.0V, 0.9V, and 0.8V for

software selection.

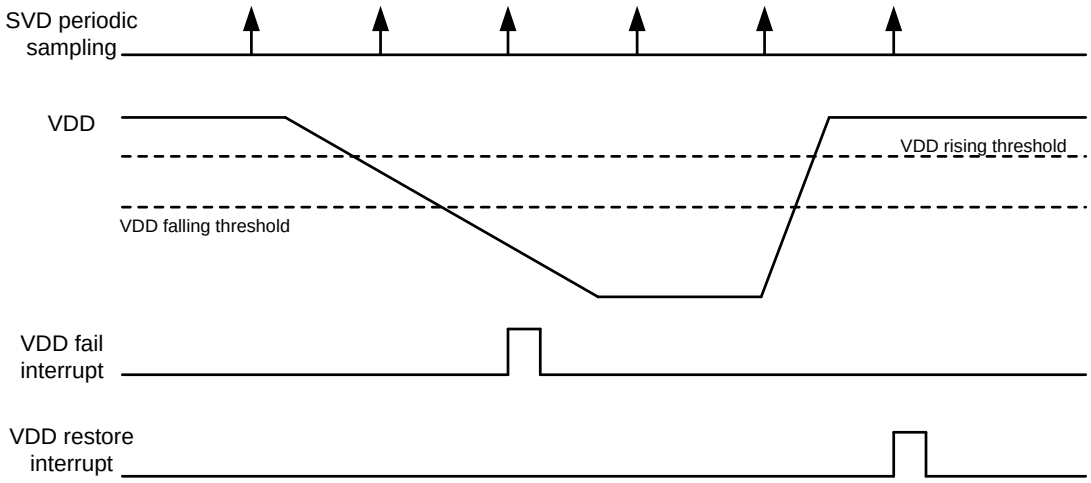


Figure 15-3 Power Supply Detection Circuit Intermittent Operation Mode

When working intermittently, when the software enables the gap of SVD, SVD does not necessarily work immediately, but waits for the next open window to arrive. When it is always enabled, the SVD will start to work after one or two LSCLK clock synchronization cycles after the software turns on the SVD. After the SVD is turned on, it takes about 100us to stabilize the output. Pay attention when the software reads the SVD output.

If the chip turns off all clocks after entering the sleep mode, and want to use SVD, you need to set SVD to always enable before sleep, and turn off the digital filtering function.

Working mode description:

- In the always-enabled/internal channel mode, the detection threshold has a window, the falling threshold and the rising threshold window are 0.1V, and the falling threshold is detected when it is not enabled until it is enabled.
- In the intermittent enable/internal channel mode, the falling threshold is detected every time the intermittent enable is started (that is, when the intermittent is not enabled to enable), so there is no threshold window, and software cooperation is required, that is, the detection is enabled in the previous intermittent When it is undervoltage, the software will increase the threshold gear by one gear; when the previous intermittent enable detects non-undervoltage, the software will restore the threshold gear.
- In the constant enable/external channel mode, the input reference voltage is three-level input, 1.0V, 0.9V, 0.8V, and the detection threshold has no window. Software cooperation is required, that is, when an undervoltage is detected, the software Increase the threshold gear by one gear; when non-undervoltage is detected, the software will restore the gear.
- In the intermittent enable/external channel mode, the input reference voltage is three-level input,

1.0V, 0.9V, 0.8V, and the detection threshold has no window. Software cooperation is required, that is, the detection is detected in the previous intermittent enable When undervoltage, the software increases the threshold gear by one gear; when the previous intermittent enable detects non-undervoltage, the software restores the gear.

15.5 Intermittent Enable Mode

In sleep mode, the average power consumption of SVD can be reduced by enabling intermittently.

In the SVD open window, the total power consumption of the comparator plus the reference voltage is less than 5uA. Each turn-on time is about 100us. If the turn-on interval is set to 1s, the average current is less than 5nA.

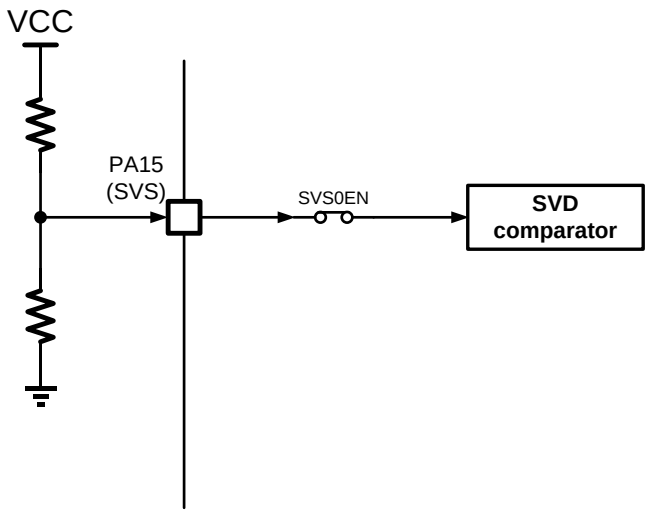
Through the intermittent enable mode, the chip can maintain the monitoring of the VDD power supply, while almost not bringing additional sleep power consumption.

15.6 External Voltage Detection

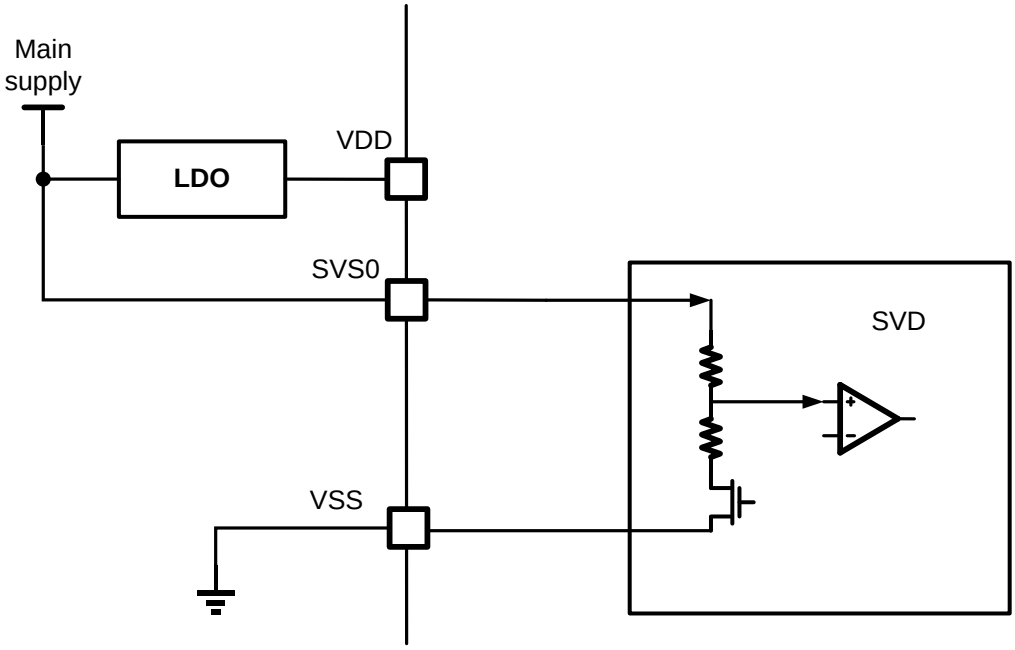
In addition to detecting chip power, SVD can also perform power-down or power-on detection on external voltage signals.

The external power supply detection is realized through the SVS pin. The input of SVS can be divided by external resistance or internal resistance, and then input to the comparator for detection. Note that the detected voltage input to the SVS pin must not be higher than the chip power supply voltage.

The following figure shows the external power supply detection with external resistor divider.



The following figure shows the external power supply detection with internal resistor divider:



The register configuration method is as follows:

SVSEN	SVDLVL	Description
0	X	The external power supply detection channel is closed, and only the internal power supply voltage is detected
1	1111	The external voltage input is not used as an internal voltage divider, and is directly input to the comparator to compare with the internal reference voltage ***Note:*** *At this time, the external input voltage cannot be higher than the power supply voltage*
	0000~1110	The external voltage input is first divided by the internal resistance, and then input to the comparator to compare with the internal reference voltage. Refer to the description in the subsequent chapters for the gear position after the partial pressure

15.7 Detection Thresholds

The voltage detection object and detection threshold can be selected through the SVSEN and SVDLVL registers.

Internal power supply detection: SVSxEN = 0, {VREF1P0EN, VREF0P9EN, VREF0P8EN} = 100, comparison reference 1.0V

SVDLVL	Rising Threshold (V)	Falling Threshold (V)
0000	1.900	1.800
0001	2.114	2.014
0010	2.329	2.229
0011	2.543	2.443
0100	2.757	2.657
0101	2.971	2.871
0110	3.186	3.086
0111	3.400	3.300
1000	3.614	3.514
1001	3.829	3.729
1010	4.043	3.943
1011	4.257	4.157
1100	4.471	4.371
1101	4.686	4.586
1110	4.900	4.800
1111	N/A	N/A

Internal power detection: SVSxEN = 0, { VREF1P0EN, VREF0P9EN, VREF0P8EN } = 010, comparison reference 0.9V

SVDLVL	Rising Threshold (V)	Falling Threshold (V)
0000	1.742	1.650
0001	1.938	1.846
0010	2.135	2.043
0011	2.331	2.239
0100	2.527	2.436
0101	2.723	2.632
0110	2.921	2.829
0111	3.117	3.025
1000	3.313	3.221
1001	3.510	3.418
1010	3.706	3.614
1011	3.902	3.811
1100	4.098	4.007
1101	4.296	4.204
1110	4.492	4.400
1111	N/A	N/A

Internal power detection：SVSxEN = 0, { VREF1P0EN, VREF0P9EN, VREF0P8EN } = 001, comparison reference 0.8V

SVDLVL	Rising Threshold (V)	Falling Threshold (V)
0000	1.583	1.500
0001	1.762	1.678
0010	1.941	1.858
0011	2.119	2.036
0100	2.298	2.214
0101	2.476	2.393
0110	2.655	2.572
0111	2.833	2.750
1000	3.012	2.928
1001	3.191	3.108
1010	3.369	3.286
1011	3.548	3.464
1100	3.726	3.643
1101	3.905	3.822
1110	4.083	4.000
1111	N/A	N/A

External power detection: SVSxEN =1，{ VREF1P0EN, VREF0P9EN, VREF0P8EN } = 100, comparison reference 1.0V

SVDLVL	Rising Threshold (V)	Falling Threshold (V)
0000	1.900	1.800
0001	2.114	2.014
0010	2.329	2.229
0011	2.543	2.443
0100	2.757	2.657
0101	2.971	2.871
0110	3.186	3.086
0111	3.400	3.300
1000	3.614	3.514
1001	3.829	3.729
1010	4.043	3.943
1011	4.257	4.157
1100	4.471	4.371
1101	4.686	4.586
1110	4.900	4.800
1111	1.2	1.2

External power detection: SVSxEN =1, { VREF1P0EN, VREF0P9EN, VREF0P8EN } = 010, comparison reference 0.9V

SVDLVL	Rising Threshold (V)	Falling Threshold (V)
0000	1.742	1.650
0001	1.938	1.846
0010	2.135	2.043
0011	2.331	2.239
0100	2.527	2.436
0101	2.723	2.632
0110	2.921	2.829
0111	3.117	3.025
1000	3.313	3.221
1001	3.510	3.418
1010	3.706	3.614
1011	3.902	3.811
1100	4.098	4.007
1101	4.296	4.204
1110	4.492	4.400
1111	1.1	1.1

External power detection: SVSxEN =1, {VREF1P2EN, VREF1P1EN, VREF1P0EN} = 001, comparison reference 0.8V

SVDLVL	Rising Threshold (V)	Falling Threshold (V)
0000	1.583	1.500
0001	1.762	1.678
0010	1.941	1.858
0011	2.119	2.036
0100	2.298	2.214
0101	2.476	2.393
0110	2.655	2.572
0111	2.833	2.750
1000	3.012	2.928
1001	3.191	3.108
1010	3.369	3.286
1011	3.548	3.464
1100	3.726	3.643
1101	3.905	3.822
1110	4.083	4.000
1111	1.0	1.0

15.8 Register

Offset	Name	Symbol
SVD(Base address: 0x40012800)		
0x00	SVD Config Register	SVD_CFGR
0x04	SVD Control Register	SVD_CR
0x08	SVD Interrupt Enable Register	SVD_IER
0x0C	SVD Interrupt Status Register	SVD_ISR
0x10	SVD Reference Voltage Select Register	SVD_VSR

15.8.1 SVD Config Register (SVD_CFGR)

NAME	SVD_CFGR							
Offset	错误!未找到引用源。x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	LVL				DFEN	MOD	ITVL	
access	R/W-0000				R/W-1	R/W-0	R/W-00	

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:4	LVL	For SVD detect threshold levels, see 11.3.3 Detection thresholds (SVD threshold level)
3	DFEN	Digital Filter Enable (must be set to 1 when MOD=1) (Digital Filter Enable) 1: Enables digital filtering of the SVD output 0: Disables digital filtering of the SVD output
2	MOD	SVD operating mode selection (SVD Mode) 1: Intermittent mode 0: Normal mode Note: Digital filtering must be enabled in intermittent mode
1:0	ITVL	SVD intermittent period 00: 62.5ms 01: 256ms

bit	name	functional description
		10: 1s 11: 4s

15.8.2 SVD Control Register (SVD_CR)

NAME	SVD_CR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							TE
access	U-0							R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						SVS0EN	EN
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:9	--	RFU: **Reserved, read as 0**
8	TE	SVD test enable, reserved by FMSH
7:2	--	RFU: **Reserved, read as 0**
1	SVS0EN	SVS external channel control 0: SVS channel disabled 1: SVS channel enabled When EN = 1, the SVS input can be set to be divided by internal resistors according to the SVDLVL register; if LVL = 1111, then the SVS input is not divided, if LVL ! = 1111, then the SVS input is divided by the internal resistor.
0	EN	SVD enable 1: Enable SVD 0: disable SVD

15.8.3 SVD Interrupt Enable Register (SVD_IER)

NAME	SVD_IER							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							

bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						PFIE	PRIE
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
1	PFIE	Power Fall Interrupt Enable 1: Allow power drop interruption 0: Disable interrupt
0	PRIE	Power Rise Interrupt Enable 1: Allow power recovery to be interrupted 0: Disable interrupt

15.8.4 SVD Interrupt Status Register (SVD_ISR)

NAME	SVD_ISR							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							SVDO
access	U-0							R
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SVDR	-					PFF	PRF
access	R	U-0					R/W-0	R/W-0

bit	name	functional description
31:9	--	RFU: **Reserved, read as 0**
8	SVDO	SVD power detection output 1: The power supply voltage is higher than the current threshold of SVD 0: The power supply voltage is lower than the current threshold of SVD

bit	name	functional description
7	SVDR	SVD output latch signal, SVD state latched by digital circuit
6:2	--	RFU: **Reserved, read as 0**
1	PFF	Power drop interrupt flag register, set when the power supply voltage drops below the SVD threshold, and cleared by software writing 1
0	PRF	Power recovery interrupt flag register, set when the power supply voltage rises above the SVD threshold, and cleared by software by writing 1

15.8.5 SVD Reference Voltage Select Register (SVD_VSR)

NAME	SVD_VSR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit0	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					V1P0EN	V0P95EN	V0P9EN
access	U-0					R/W-1	R/W-0	R/W-0

bit	name	functional description
31:3	--	RFU: **Reserved, read as 0**
2	V1P0EN	1.0V reference input enable signal (1.0V reference enable) 1: Enable 1.0V reference input 0: Disable 1.0V reference input
1	V0P95EN	0.95V reference input enable signal (0.95V reference enable) 1: Enable 0.95V reference input 0: Disable 0.95V reference input
0	V0P9EN	0.9V reference input enable signal (0.9V reference enable) 1: Enable 0.9V reference input 0: Disable 0.9V reference input

16 AES Algorithm Module (AES)

16.1 Functional Description

The main functions of the AES unit are as follows:

- Support decryption key extension
- Support 128bit/192bit/256bit key length
- Support ECB, CBC, CTR, GCM
- Support DMA for automatic data transmission
- Support multiplication in GF (2^128) domain, support GMAC

16.2 Working Mode

ES module has four operating modes, which are set by MODE[1:0] registers:

- **MODE=1**: Encryption using the key stored in the AES_KEYRx register.
- **MODE=2**: Key extension, which overwrites the encryption key initially stored in the AES_KEYRx register with the key calculation result stored in the internal register after the key extension is completed.
- **MODE=3**: Decryption using the decryption key stored in the AES_KEYRx register.
- **MODE=4**: Key extension and decryption with the encryption key stored in the AES_KEYRx register (not used in CTR mode).

First, determine the working mode by configuring the MODE[1:0] register. The MODE register must be configured before AES is enabled (EN=0). The KEY register should also be configured before AES is enabled. Then configure the data stream processing mode register CHMOD[1:0], and also need to configure the IV register in the CBC/CTR/GCM mode.

Then you can enable EN. In mode 1/mode 3/mode 4, the AES module waits for the software to write input data to the AES_DINR register, and AES starts to calculate after 4 writes of 128bit. In mode 2, the key expansion operation is performed immediately after EN is enabled.

After the calculation is completed, the flag CCF will be set. If CCFIE=1, an interrupt signal will be generated. The software reads the result of 128 bits in total from the AES_DOUTR register 4 times.

AES also supports DMA mode. By configuring DMAOUTEN=1 and DMAINEN=1, AES can continuously process data with DMA without CPU intervention.

The error flags RDERR and WRERR will be set during an error read and write operation. If ERRIE is enabled, a corresponding error interrupt will also be generated. AES will continue to work normally after an error occurs.

The AES module can be reset at any time by resetting the EN register.

16.3 AES Data Stream Processing Modes

AES module has 4 data stream processing modes: ECB, CBC, CTR, GCM.

16.3.1 ECB Mode

In the default working mode, there is no need to use the IV register in this mode, and each block performs encryption and decryption calculations separately. The encryption and decryption process is shown in Figure 16-1 and Figure 16-2.

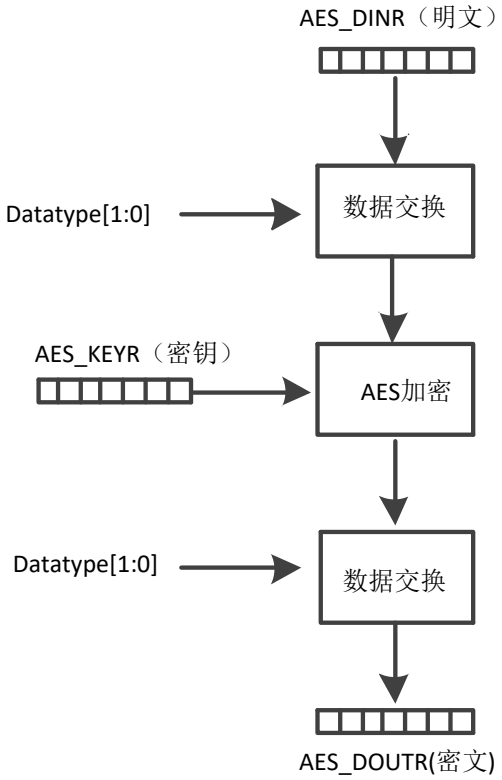


Figure 16-1 ECB Mode Encryption

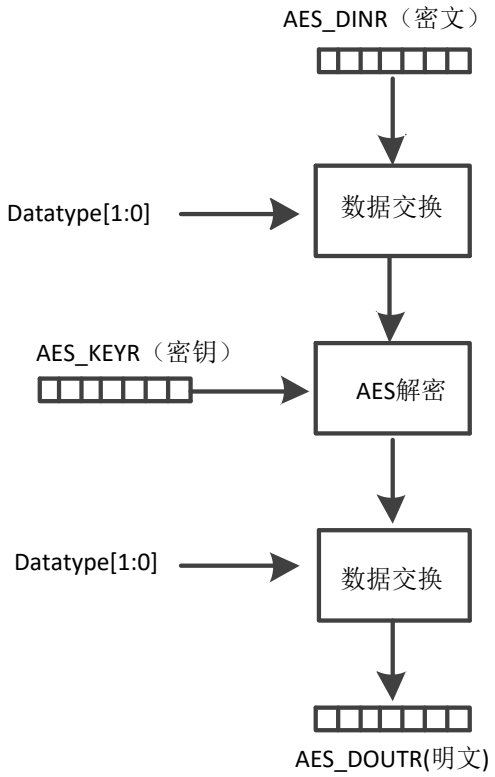


Figure 16-2 ECB Mode Decryption

16.3.2 CBC Mode

The plaintext data of each block is XORed with the encryption result of the previous block and then entered as encrypted data. The first block requires an initial IVRx register value. The XOR operation during encryption is before encryption and the XOR operation during decryption is after encryption. The workflow is shown in Figure 16-3 and Figure 16-4.

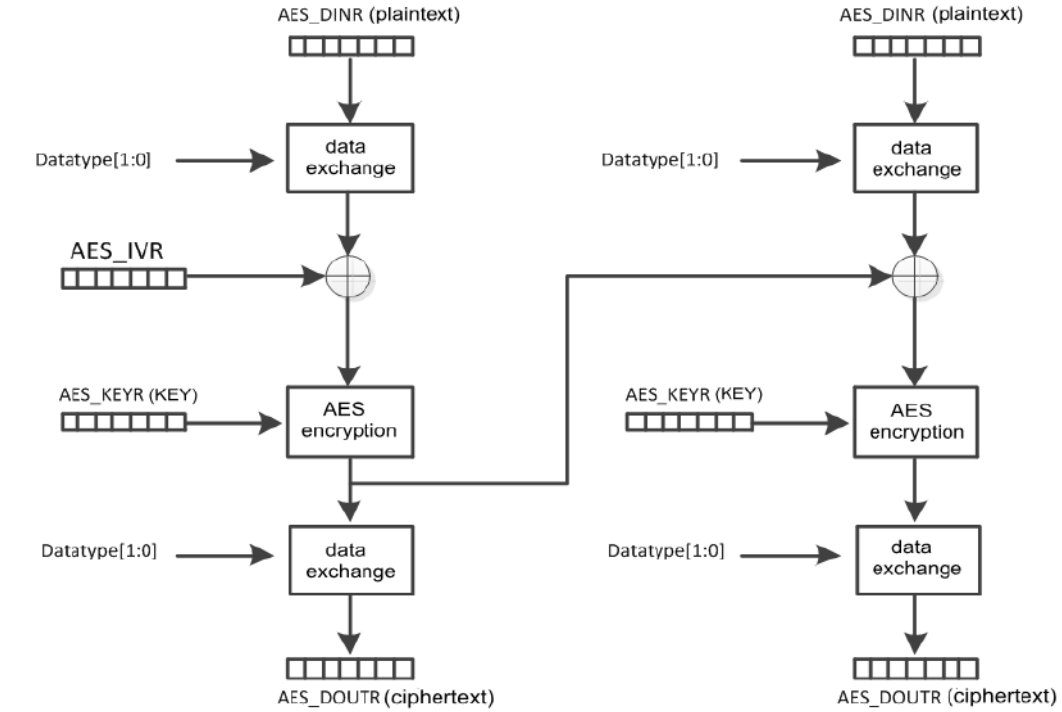


Figure 16-3 CBC Mode Encryption

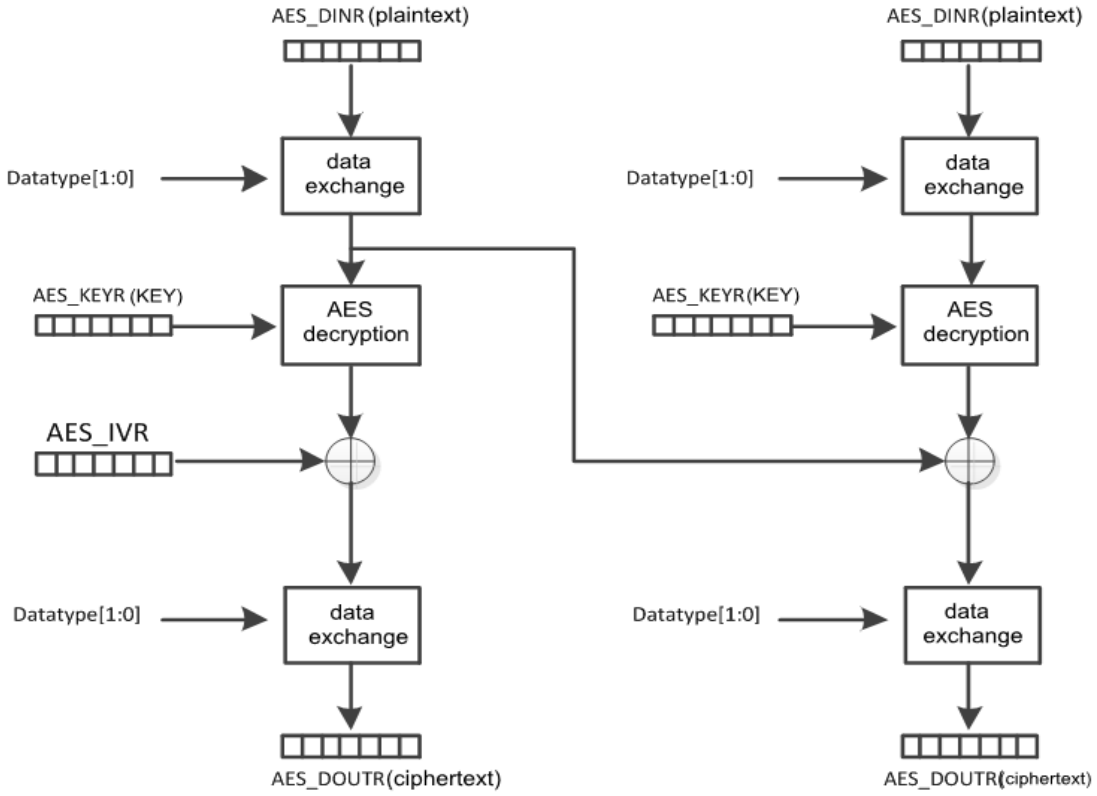


Figure 16-4 CBC Mode Decryption

Note: *While AES is working, the AES_IVR register reads as 0x00000000.*

16.3.3 Suspend Mode

If a higher priority data needs to be processed, the current data operation can be suspended. The suspended data processing can be resumed in the encryption and decryption operation mode. It is only available in the mode where the CPU participates, and not available in the DMA mode.

The correct workflow is: data is paused after the result of a block is read.

Pause AES by writing 0 to the EN bit. The software reads the value in the AES_IVRx register and stores it. The value needs to be written into the AES_IVRx register during the restoration operation.

The process is shown in Figure 16-5:

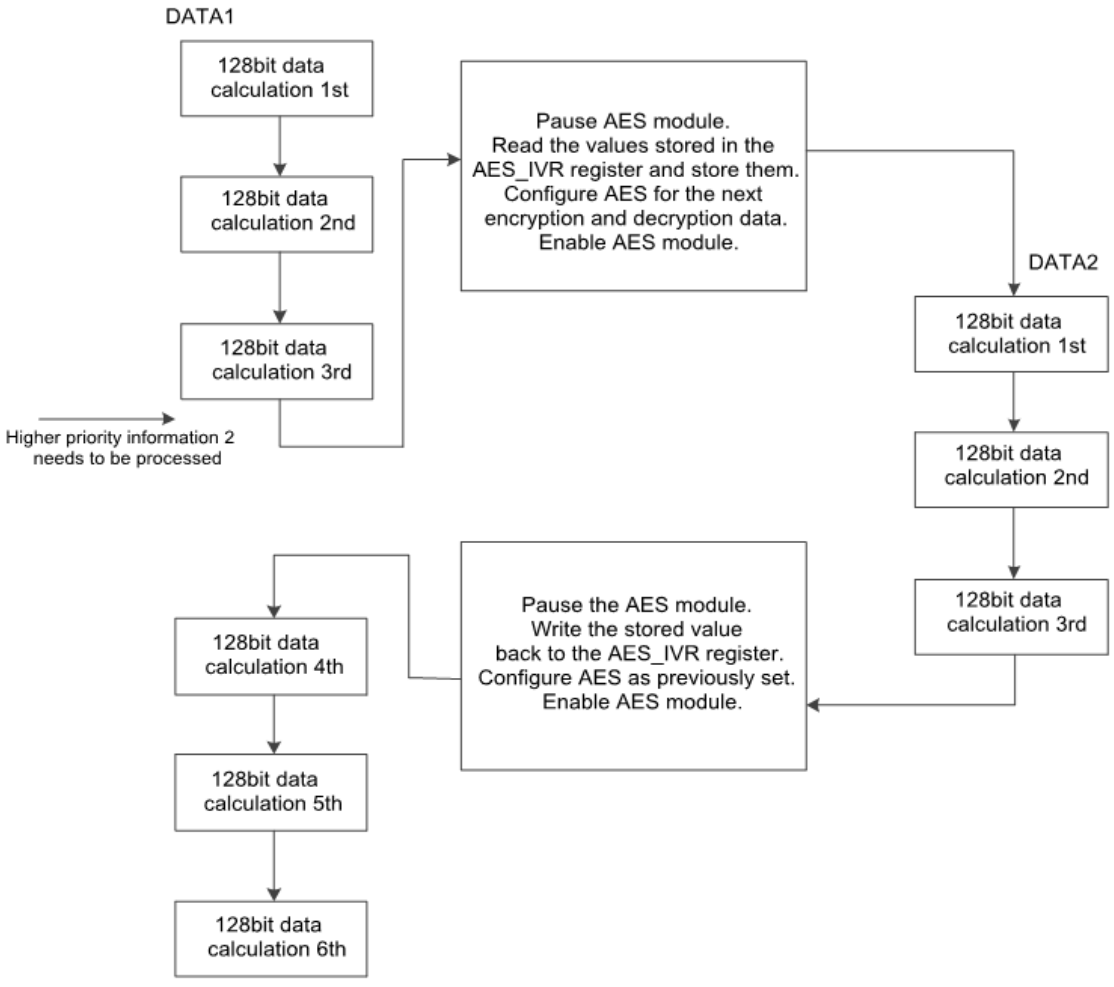


Figure 16-5 Suspend Mode Sequence

16.3.4 CTR Mode

In this mode, a 32-bit counter and a random number are used as the input of the encryption and decryption module. The result is XORed with the plaintext data. The process is shown in Figure 16-6 and Figure 16-7.

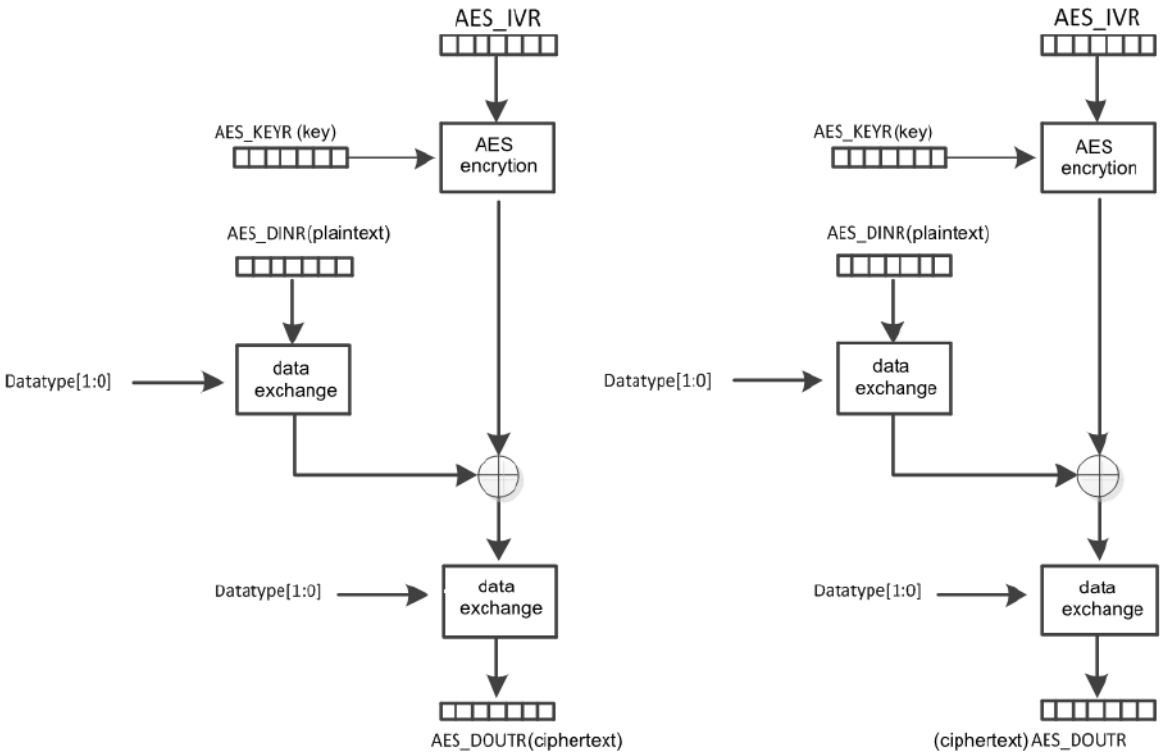


Figure 16-6 CTR Mode Encryption

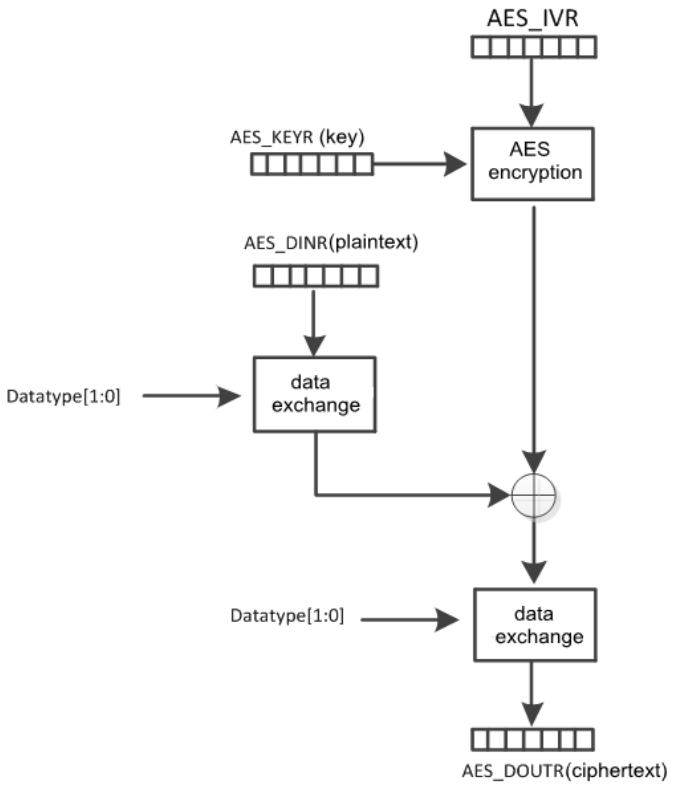


Figure 16-7 CTR Mode Decryption

The random number (nonce) and 32-bit counter are stored in the IV register, as shown in Figure 16-8

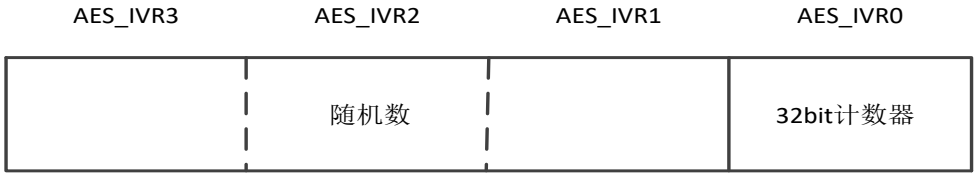


Figure 16-8 Storage Format of 32-bit Counter Value and Random Number

The key extension and decryption mode under CTR mode is regardless.

16.3.5 Suspend Mode under CTR Mode

Similar to the suspend mode under CBC mode. Refer to suspend mode under CBC mode.

16.3.6 GCM Mode

Refer to the documentation *The Galois/Counter Mode of Operation (GCM)* for details.

The GCM encryption is defined according to the following formula:

$$
\begin{aligned}
H &= E(K, 0^{128}) \\
Y_0 &= \begin{cases} IV \| 0^{31}1 & \text{if } \mathrm{len}(IV) = 96 \\ \mathrm{GHASH}(H, \{\}, IV) & \text{otherwise.} \end{cases} \\
Y_i &= \mathrm{incr}(Y_{i-1}) \text{ for } i = 1, \ldots, n \\
C_i &= P_i \oplus E(K, Y_i) \text{ for } i = 1, \ldots, n-1 \\
C_n^* &= P_n^* \oplus \mathrm{MSB}_u(E(K, Y_n)) \\
T &= \mathrm{MSB}_t(\mathrm{GHASH}(H, A, C) \oplus E(K, Y_0))
\end{aligned}
$$

where the GHASH function is defined as $\mathrm{GHASH}(H, A, C) = X_{m+n+1}$, and X is defined as

$$
X_i = \begin{cases}
0 & \text{for } i = 0 \\
(X_{i-1} \oplus A_i) \cdot H & \text{for } i = 1, \ldots, m-1 \\
(X_{m-1} \oplus (A_m^* \| 0^{128-v})) \cdot H & \text{for } i = m \\
(X_{i-1} \oplus C_i) \cdot H & \text{for } i = m+1, \ldots, m+n-1 \\
(X_{m+n-1} \oplus (C_m^* \| 0^{128-u})) \cdot H & \text{for } i = m+n \\
(X_{m+n} \oplus (\mathrm{len}(A) \| \mathrm{len}(C))) \cdot H & \text{for } i = m+n+1.
\end{cases}
$$

The encryption and decryption process of GCM mode is shown in Figure 16-9 and Figure 16-10

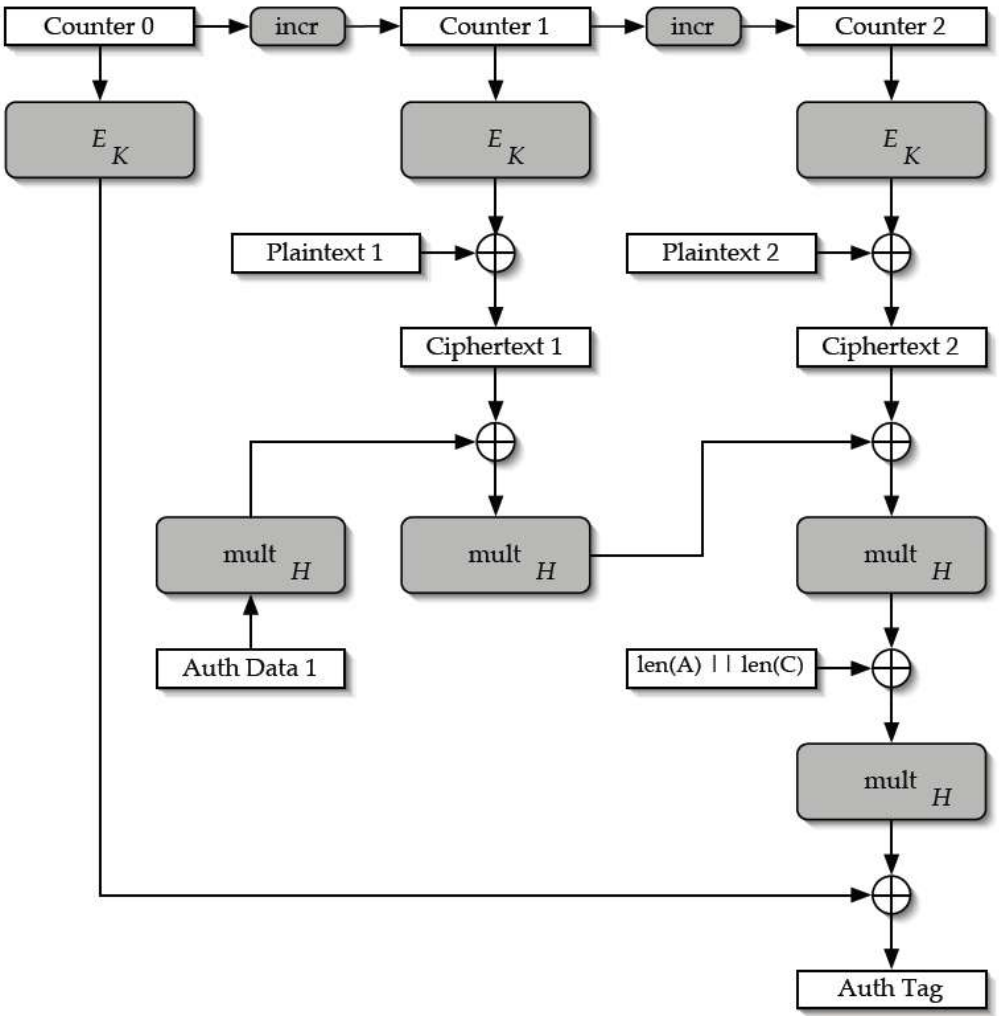


Figure 16-9 GCM Mode Encryption

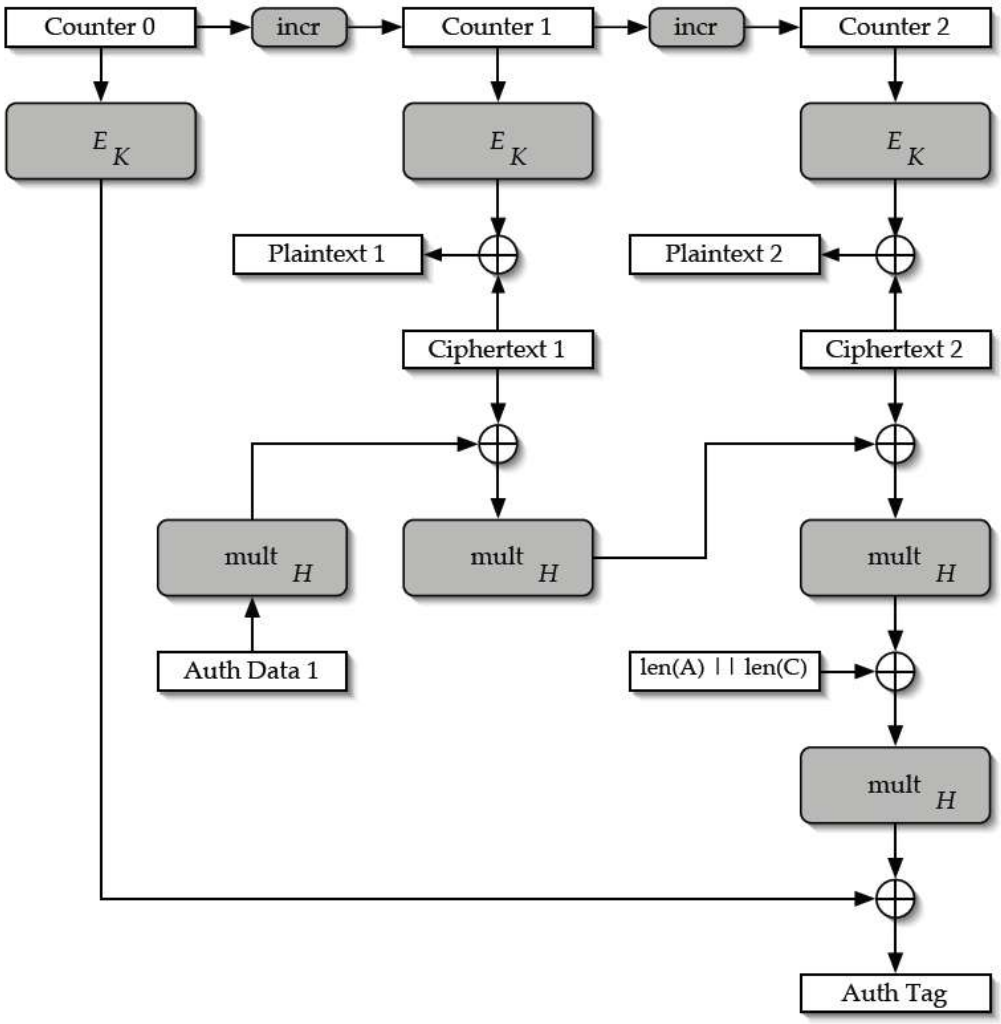


Figure 16-10 GCM Mode Decryption

The E_k in the diagram indicates the AES encryption module. "$mult_H$" module performs a multiplication under GF(2^128) domain. "incr" indicates the counter plus one.

In GCM mode the hardware AES module and $mult_H$ module are dispatched by software. The process of GCM mode encryption and decryption is same as CTR mode. The authentication process is implemented by the software using $mult_H$ module.

16.3.7 MultH Module

Multiplication under GF(2^128) domains implemented using the following algorithm.

Algorithm 1 Multiplication in $GF(2^{128})$. Computes the value of $Z = X \cdot Y$, where X, Y and $Z \in GF(2^{128})$.

$Z \leftarrow 0, V \leftarrow X$
for $i = 0$ to 127 **do**
  if $Y_i = 1$ **then**
    $Z \leftarrow Z \oplus V$
  end if
  if $V_{127} = 0$ **then**
    $V \leftarrow \text{rightshift}(V)$
  else
    $V \leftarrow \text{rightshift}(V) \oplus R$
  end if
end for
return Z

The input and output registers of the MultH module are multiplexed with the AES registers. The block diagram of the module is shown in Figure 16-11.

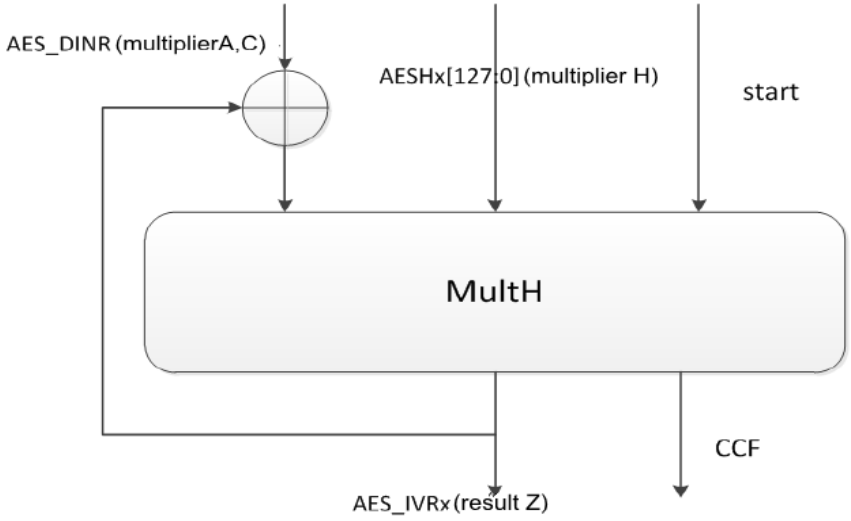


Figure 16-11 MultH Module Block Diagram

The input registers of the multH module multiplex the lower 128 bits of the AES input registers AES_DINR and AES_KEYx. The output register is multiplexed with the AES_IVR register. When using, configure the CHMOD[1:0] register to MultH mode, then configure the AES_KEYx and AES_IVR registers to input and output 128 bits each, enable EN, input data to AES_DINR, and wait for the CCF to be set to complete the calculation.

16.3.8 Recommended GCM Process

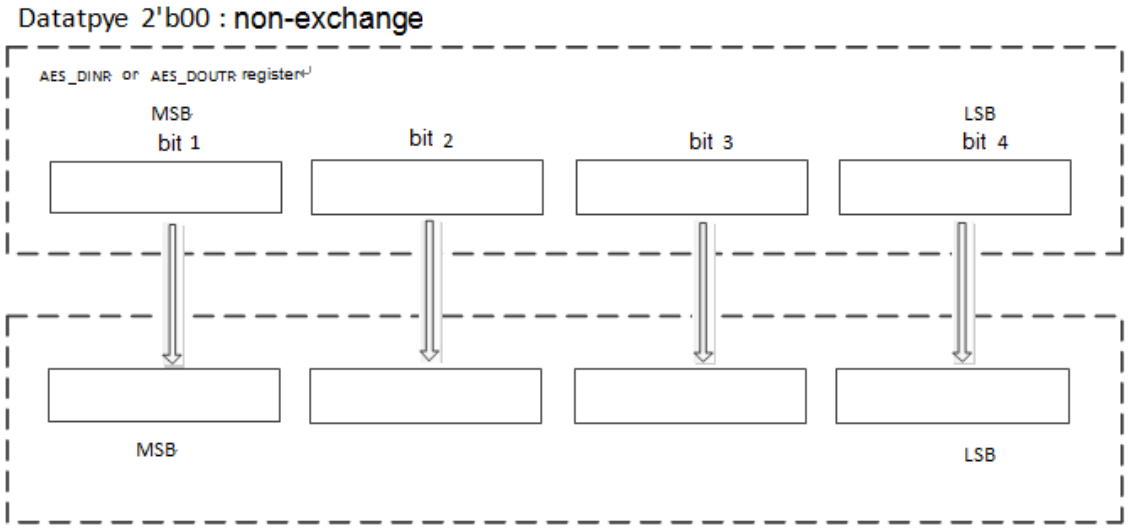
The implementation of GCM mode requires hardware and software cooperation, and this document provides a recommended way to implement it.

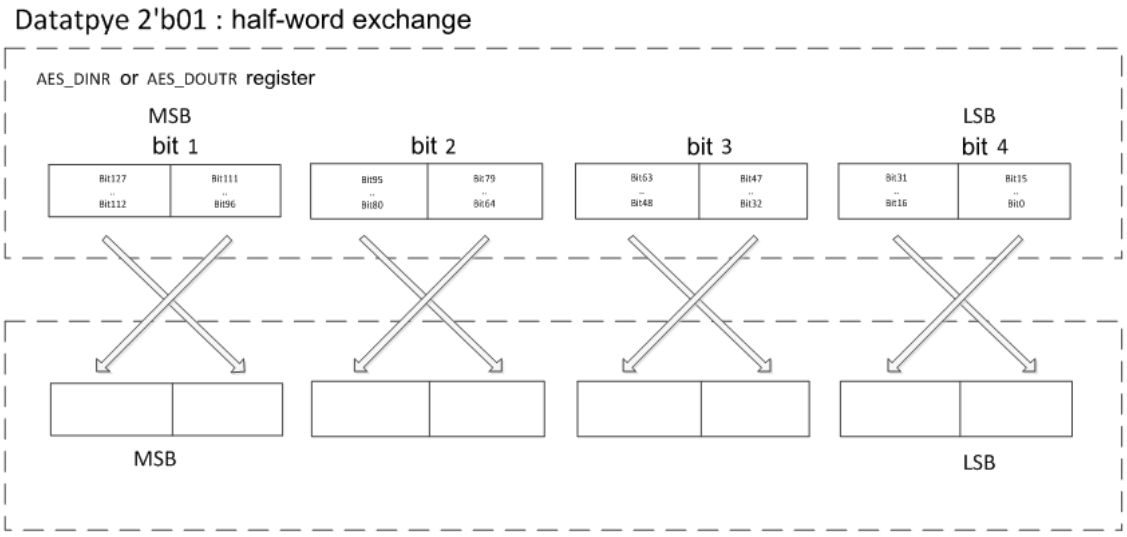
The encryption and decryption process of GCM mode is the same as CTR mode. Only MultH module is used for authentication process.

- The AES module is invoked to calculate H.
- The AES module is invoked to calculate E(K, Y0).
- Use CTR mode to start AES encryption and decryption of continuous data. Initial value of IV register is Y1.
- Perform Continuous calculation of GHASH using multH module
- The value of tag can be calculated by the XOR result of the final GHASH result and E(K, Y0).

16.4 Data Type

AES reads and writes 32bit data at a time, and each 32bit can exchange data in different ways according to the setting of the DATATYPE[1:0] register. As shown in Figure 16-12.





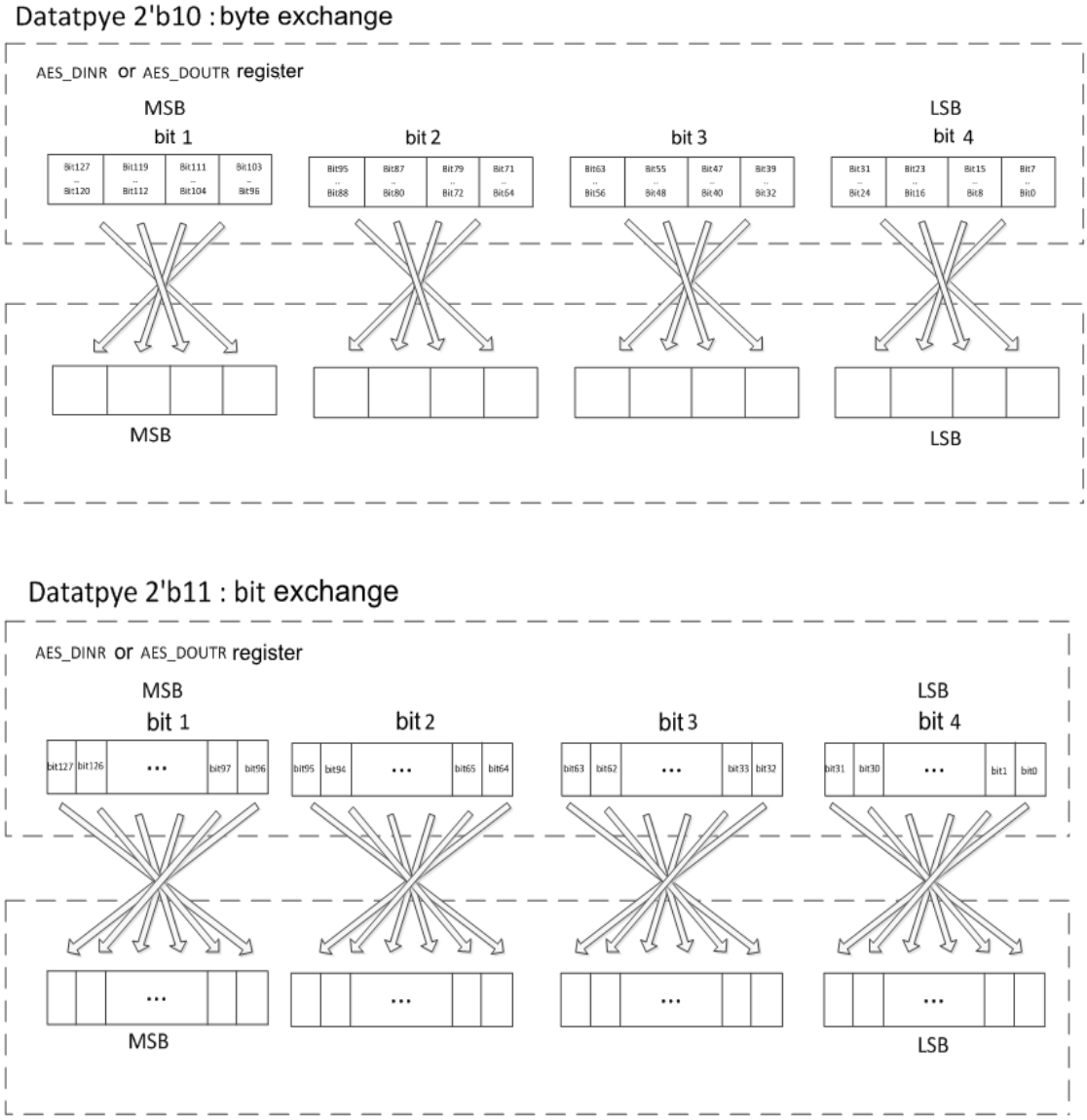


Figure 16-12 Data Storage Format under Different Data Type

16.5 Operation Sequence

16.5.1 Mode 1: Encryption

- Set EN=0 to reset AES
- Set mode register MODE[1:0]=00, set stream data processing mode register CHMOD[1:0]
- Write AES_KEYRx register, and if under CTR and CBC mode user should also write AES_IVRx register
- Set EN=1 to enable AES
- Input the data by writing 4 times to AES_DINR register

- Wait for the CCF flag to set
- Read 4 times from AES_DOUTR for the entire encryption result
- For the same key, repeat steps 5,6,7 to encrypt the next 128bit block

Steps 5-7 are shown in the figure.

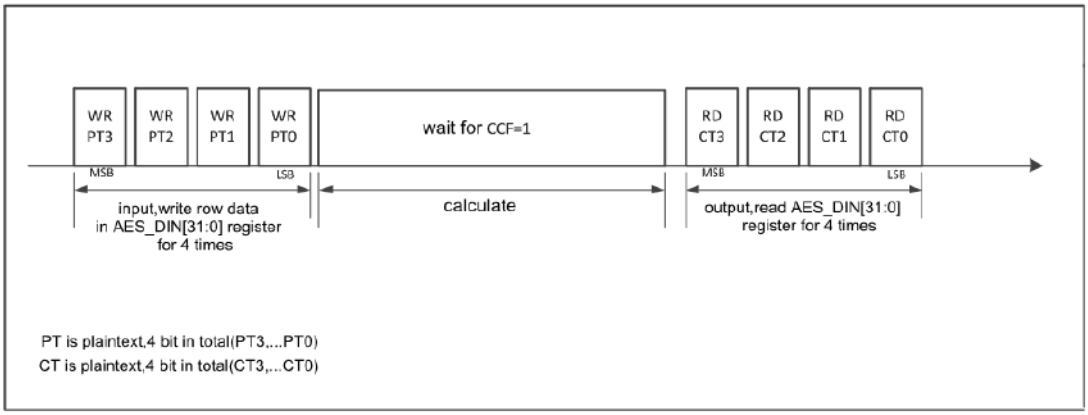


Figure 16-13 Operation Sequence of Mode 1: Encryption

16.5.2 Mode 2: Key Expansion

- Reset EN Reset AES module
- Set mode register mode[1:0]=01, the value of CHMOD[1:0] register is not concerned.
- Write the AES_KEYRx register.
- Write EN=1 to enable AES
- Wait for the CCF logo to be set
- Clear the CCF flag, and the expanded key is automatically written back to the AES_KEYRx register. If necessary, you can read the AES_KEYRx register to get the result. To recalculate the extended key, repeat steps 3, 4, 5, and 6.

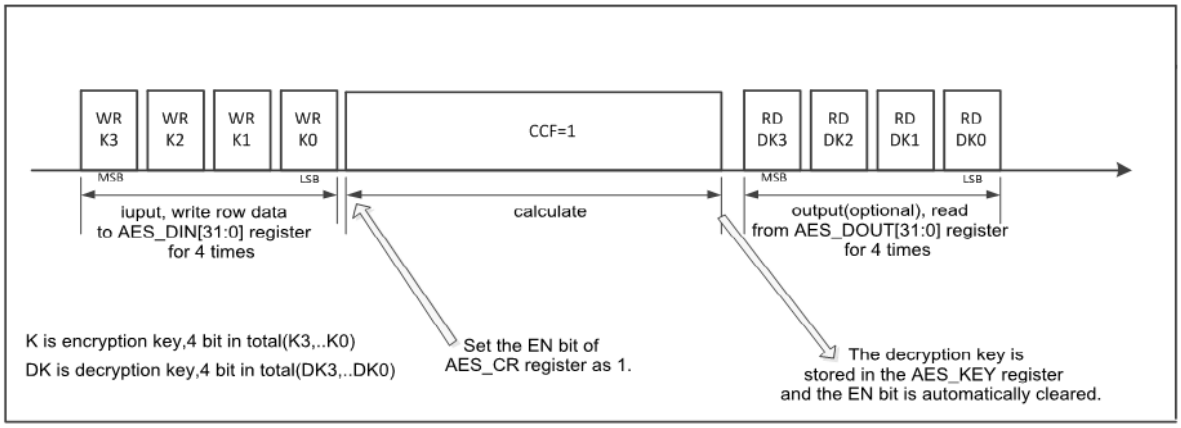


Figure 16-14 Operation Sequence of Mode 2: Key Expansion

16.5.3 Mode 3: Decryption

- Reset EN Reset AES module
- Set the mode register mode[1:0]=10, set the stream data processing mode register CHMOD[1:0]
- Write the AES_KEYRx register (if the extended key has been calculated in mode 2, you can skip this step), and write the AES_IVRx register in CTR and CBC modes.
- Write EN=1 to enable AES
- Write AES_DINR register 4 times
- Wait for the CCF logo to be set
- Read the decryption result from AES_DOUTR 4 times
- For the same key, repeat steps 5, 6, and 7 to decrypt the next 128bit block

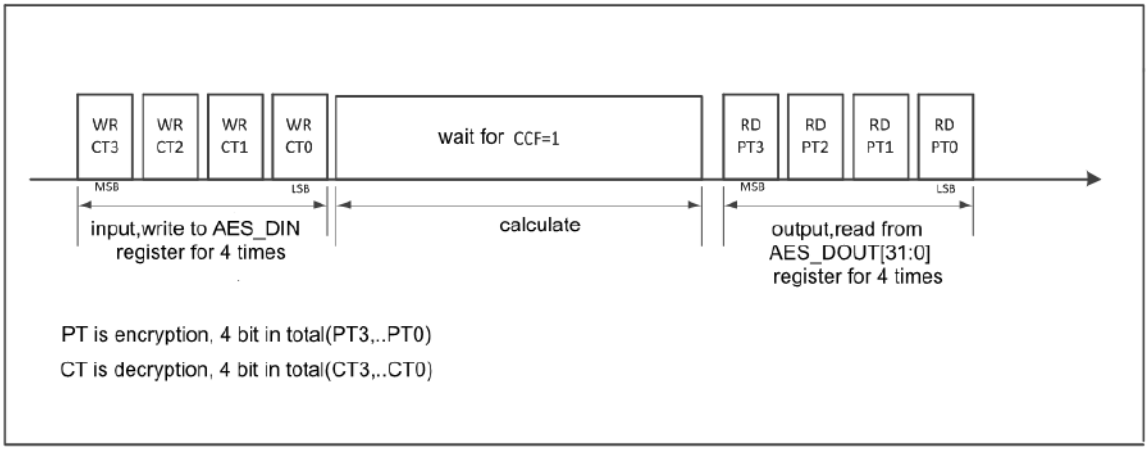


Figure 16-15 Operation Sequence of Mode 3: Decryption

16.5.4 Mode 4: Key Expansion + Decryption

- Reset EN Reset AES module
- Set the mode register mode[1:0]=11, and set the stream data processing mode register CHMOD[1:0]. This mode is forbidden to use in CTR mode. If you set mode[1:0]=11 and CHMOD[1:0]=10, you will be forced to enter the CTR decryption mode.
- Write the AES_KEYRx register, and write the AES_IVRx register in CBC mode.
- Write EN=1 to enable AES
- Write AES_DINR register 4 times
- Wait for the CCF logo to be set
- Read the decryption result from AES_DOUTR 4 times
- For the same key, repeat steps 5, 6, and 7 to decrypt the next 128bit block

Note: In this mode, the encryption key is always stored in the AES_KEYRx register, and the extended key will be recalculated internally each time and will not be stored in the AES_KEYRx register.

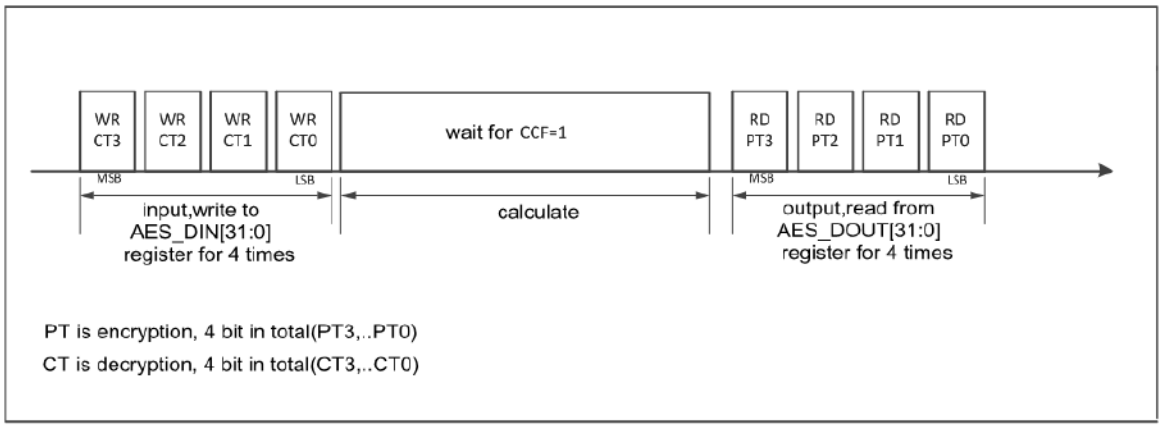


Figure 16-16 Operation Sequence of Mode 4: Decryption

16.5.5 Using the MultH Module

- Reset EN Reset the AES module.
- Set the stream data processing mode register CHMOD[1:0]=11. In this mode, the value of the mode[1:0] register cannot be 01. Configured in mode 2: key extension. At the same time, configuring mode[1:0]=01 and CHMOD[1:0]=11 will perform key extension operation due to the higher priority value of the mode register.
- Write the AESHx register, if it is the first round of calculation, the initial value is 0x00000000.
- Write EN=1 to enable the multH module.
- Write the AES_DINR register 4 times. The MultH module will use the value entered in the

AES_DINR register of the XOR of the last calculation result as a multiplier of the multH module. So assign the calculation result of the previous round to 0x00000000, which realizes the function of directly using the value input in the AES_DINR register as a multiplier of the multH module.

- Wait for the CCF logo to be set
- Read the calculation result from the AES_IVR register, you can control whether the read data is exchanged in order through the IVRSWAP register and then put it on the bus
- For the same H, repeat steps 5 and 6 for continuous calculation. Then the function of a GMAC can be realized.

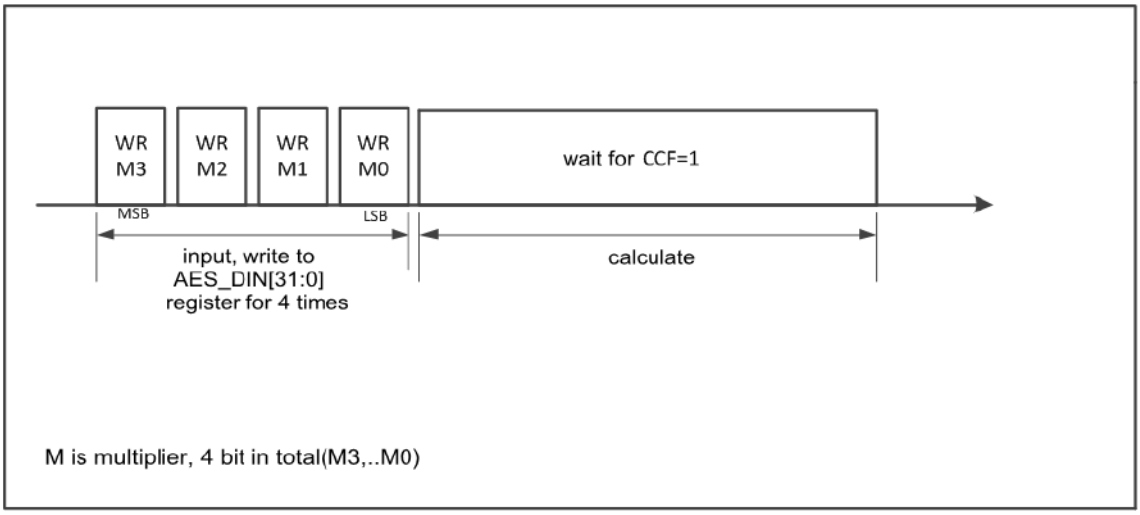


Figure 16-17 Operation Sequence of Using MultH Module

16.6 DMA Interface

- An input request channel: When DMAINEN=1, a DMA request is initiated whenever AES needs input data to be written to the AES_DINR register.
- An output request channel: When DMAOUTEN=1, a DMA request is initiated whenever AES needs to output data from the AES_DOUTR register.

Four DMA requests are generated in each phase, and the requests will generate continuously until the AES module is disabled. 128 bits of data are automatically fetched for the next calculation after the AES calculation.

Note: *The CCF flag may be high when DMAOUTEN=1 in DMA mode.*

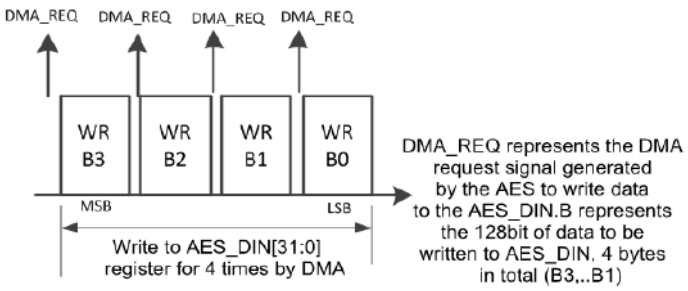


Figure 16-18 DMA Request and Data Transfer Sequence while Input

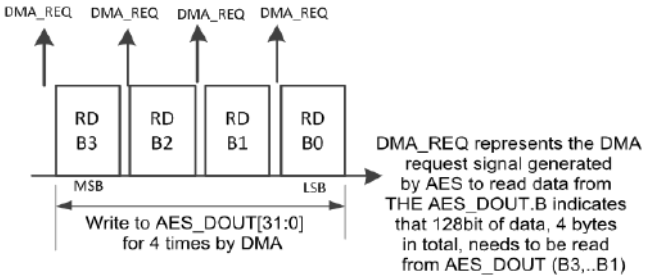


Figure 16-19 DMA Request and Data Transfer Sequence while Output

16.6.1 MultH Module Interfaces with DMA

MultH calculations can also be done via DMA. When DMAINEN=1 and CHMOD[1:0]=11, a DMA request is initiated whenever AES needs input data to be written to the AES_DINR register. Setting DMAOUTEN=1 is invalidate under this mode and AES module will not generate DMA requests.

16.7 Error Flags

A read operation occurs during the compute and input phases will set RDERR.

A write operation occurs during the compute and output phases will set WRERR.

The AES module will not be automatically stopped by the hardware after an error is generated, but it continues to operate as normal.

16.8 Register

Offset	Name	Symbol
AES(Base Address: 0x40013800)		
0x00	AES Control Register	AES_CR
0x04	AES Interrupt Enable Register	AES_IER
0x08	AES Interrupt Status Register	AES_ISR
0x0C	AES Data Input Register	AES_DIR
0x10	AES Data Output Register	AES_DOR
0x14	AES Key Register 0	AES_KEY0
0x18	AES Key Register 1	AES_KEY1
0x1C	AES Key Register 2	AES_KEY2
0x20	AES Key Register 3	AES_KEY3
0x24	AES Key Register 4	AES_KEY4
0x28	AES Key Register 5	AES_KEY5
0x2C	AES Key Register 6	AES_KEY6
0x30	AES Key Register 7	AES_KEY7
0x34	AES Initial Vector Register 0	AES_IVR0
0x38	AES Initial Vector Register 1	AES_IVR1
0x3C	AES Initial Vector Register 2	AES_IVR2
0x40	AES Initial Vector Register 3	AES_IVR3
0x44	AES Control Register	AES_H0
0x48	AES Interrupt Enable Register	AES_H1
0x4C	AES Interrupt Status Register	AES_H2
0x50	AES Data Input Register	AES_H3

16.8.1 AES Control Register (AES_CR)

NAME	AES_CR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-	KEYLEN		DMAOEN	DMAIEN	IVRSWAP		-
access	U-0	R/W-00		R/W-0	R/W-0	R/W-00		U-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	CHMOD		MODE		DATATYP		EN
access	U-0	R/W-00		R/W-00		R/W-00		R/W-0

bit	name	functional description
31:15	--	RFU: **Reserved, read as 0**
14:13	KEYLEN	AES Key Length 00: 128-bit 01: 192-bit 10: 256-bit 11: Reserved Cannot be modified when EN=1.
12	DMAOEN	DMA Output Enable 0: Disable DMA output 1: Enable DMA output The AES module will automatically generate AES->RAM transfer requests under mode 1, mode 3 and mode 4 after setting this bit to 1. Requests will not be generated in mode 2.
11	DMAIEN	DMA Input Enable 0: Disable DMA input 1: Enable DMA input The AES module will automatically generate RAM->AES transfer requests under mode 1, mode 3, mode 4 and MultH mode after setting this bit to 1. Requests will not be generated in mode 2.
10:9	IVRSWAP	IVR register read-out swapping function (IVR register read-out swapping) This register is only read for the IVR register, and the read data undergoes sequential exchange processing before being put on the system bus 00: 32bit data is not exchanged 01: 16bit data half word exchange 10: 8bit data byte exchange 11: 1bit data bit exchange
8:7	--	RFU: **Reserved, read as 0**
6:5	CHMOD	AES data stream processing mode, cannot be modified when AESEN=1. (Cipher Mode) 00: ECB 01: CBC 10: CTR 11: Use MultH module
4:3	MODE	AES working mode, cannot be modified when AESEN=1. (operation MODE) 00: Mode 1: Encryption 01: Mode 2: Key extension 10: Mode 3: Decryption 11: Mode 4: Key expansion + decryption

bit	name	functional description
		When configured as mode 4 in CTR mode, it will automatically enter the decryption mode of CTR. That is, configure MODE=2'b11 when CHMOD=2'b10, and AES will be executed according to MODE=2'b10.
2:1	DATATYP	Select the data type, it cannot be modified when AESEN=1. For specific exchange rules, please refer to the AES data type chapter. (Data type) 00: 32bit data is not exchanged 01: 16bit data half word exchange 10: 8bit data byte exchange 11: 1bit data bit exchange
0	EN	AES enable 0: Disable 1: enable Clearing the AESEN bit at any time can reset the AES module In mode 2, this bit will be automatically cleared by hardware after a calculation is completed

16.8.2 AES Interrupt Enable Register (AES_IER)

NAME	AES_IER							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					WRERR_IE	RDERR_IE	CCF_IE
access	U-0					R/W-0	R/W-0	R/W-0

bit	name	functional description
31:3	-	RFU: **Reserved, read as 0**
2	WRERR_IE	Write error interrupt is enabled, 1 is valid.
1	RDERR_IE	Read error interrupt enable, 1 is valid.
0	CCF_IE	AES calculation completed interrupt enable, 1 is valid.

16.8.3 AES Interrupt Status Register (AES_ISR)

NAME	AES_ISR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					WRERR	RDERR	CCF
access	U-0					R/W1C-0	R/W1C-0	R/W1C-0

bit	name	functional description
31:3	--	RFU: **Reserved, read as 0**
2	WRERR	Write error flag: set when a write operation occurs in the calculation or output stage, and cleared by software writing 1
1	RDERR	Read error flag: set when a read operation occurs during the calculation or input phase, and cleared by software writing 1
0	CCF	AES calculation complete flag, software write 1 to clear 1: Calculation is complete 0: calculation is not completed

16.8.4 AES Data Input Register (AES_DIR)

NAME	AES_DIR							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	DIN[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
name	DIN[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
name	DIN[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
name	DIN[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:0	DIN	Data input register. When AES needs to input encryption and decryption data, write to this register 4 times in succession. (AES Data Input) Mode 1 (encryption): Write the plaintext from MSB to LSB in 4 times. Mode 2 (key expansion): No need to use data input register Mode 3 and Mode 4 (decryption): Write the cipher text from MSB to LSB in 4 times. MultH mode: Write the multiplier A or C from MSB to LSB in 4 times.

16.8.5 AES Data Output Register (AES_DOR)

NAME	AES_DOR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	DOUT[31:24]							
access	R-0000 0000							
bit	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
name	DOUT[23:16]							
access	R-0000 0000							
bit	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
name	DOUT[15:8]							
access	R-0000 0000							
bit	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
name	DOUT[7:0]							
access	R-0000 0000							

bit	name	functional description
31:0	DOUT	Data output register, when the AES calculation is completed, the result of encryption and decryption can be read out four times. (AES Data Output) Mode 1 (encryption): Read the cipher text from MSB to LSB in 4 times. Mode 2 (key expansion): No need to use data input and output registers Mode 3 and Mode 4 (decryption): Output the plaintext from MSB to LSB in 4 times. MultH mode: The result of the operation is stored in the IVR register without reading the AES_DOUTR register.

16.8.6 AES Key Register (AES_KEYx)

NAME	AES_KEYx(x=0,1,2,3,4,5,6,7)							
Offset	0x14 + x*0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	KEYx[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
name	KEYx[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
name	KEYx[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
name	KEYx[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:0	KEYx	AES Key At most 256-bit, AESKEY0 stores the lowest 32bit, AESLKEY7 stores the highest 32bit.

16.8.7 AES Initial Vector Register (AES_IVRx)

NAME	AES_IVRx(x=0,1,2,3)							
Offset	0x34 + x*0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	IVRx[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23	Bit23
name	IVRx[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15	Bit15
name	IVRx[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
name	IVRx[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:0	IVRx	AES Initial Vector AES operation 128-bit initial vector; these registers save the calculation result under MultH mode.

16.8.8 AES MultH Parameter Register (AES_Hx)

NAME	AES_Hx(x=0,1,2,3)							
Offset	0x44 + x*0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	Hx[31:24]							
access	R/W-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	Hx[23:16]							
access	R/W-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	Hx[15:8]							
access	R/W-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	Hx[7:0]							
access	R/W-00000000							

bit	name	functional description
31:0	Hx	MultH operation 128bit input H parameter (H Parameter) H0 saves H[31:0], H3 saves H[127:96]

17 True Random Number Generator (TRNG)

17.1 Introduction

FM33LG0 uses 2 Galois true random noise sources as true random number seeds, combined with simple online detection (32-bit all 0 and all 1 detection), LFSR post-processing, and pseudo-random LFSR to form a random number generator on the chip.

TRNG's startup test and complete online test functions need to be implemented by firmware.

The sampling of Galois noise source and LFSR suggest using 4MHz clock. The interval between taking 32-bit random numbers twice shall not be less than 32 clock cycles.

The true random number generator has passed the FIPS PUB140-2 test with a success rate of 99.9%.

17.2 Functional Description

17.2.1 Random Number Generation

The following figure shows the block diagram of the true random number generator.

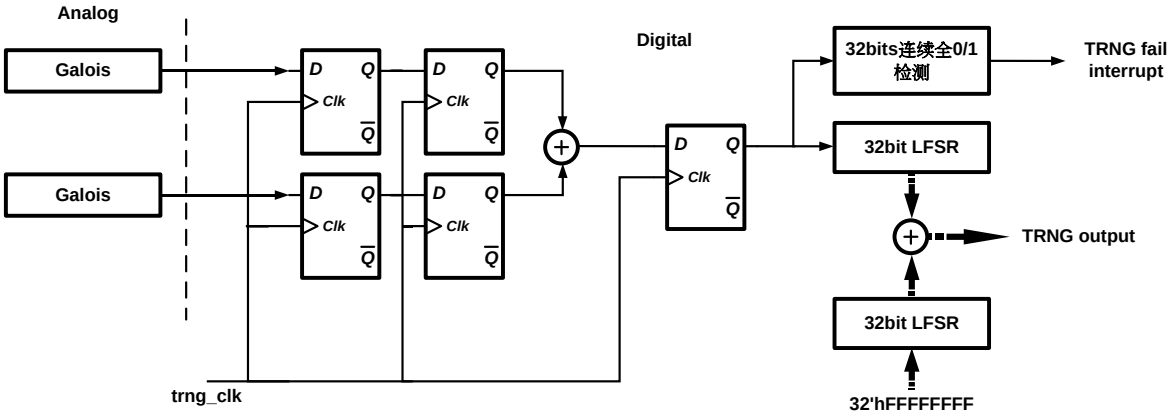


Figure 17-1 True Random Number Generation Module

The true random noise sources are 2 Galois ring oscillators. The output of Galois ring oscillators are XORed in the digital circuit and sampled by the system clock, and then processed by LFSR. Before LFSR post-processing, random number online detection is carried out. If 32 bits continuous 0 or 1 are found, TRNG failure alarm interrupt will be generated. At the same time, in order to avoid the poor randomness, another set of LFSR is used to produce a sequence of pseudo random number. And two sets of LFSR are XORed to generate final random data.

17.2.2 Working Clock

The working clock of the random number generator adopts the frequency division clock of RCHF, which is independent of APBCLK. In order to ensure the quality of random numbers, it is generally recommended to use the 4M clock as the random number working clock, and configure the random number working clock divider register (OPCCON2.RNGPRSC) in the CMU module according to the 4M target frequency.

The register configuration of TRNG uses APBCLK. Before using the TRNG module, you need to set the bus clock enable register TRNG_PCE and the module working clock enable register RNGCKE. See CMU module register description.

The working clock diagram is as follows:

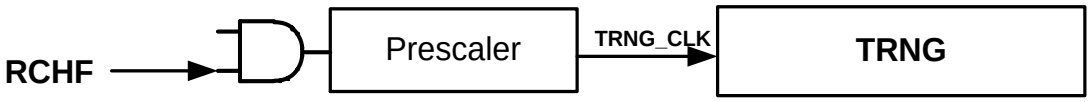


Figure17-2 Clock for TRNG

17.2.3 Random Number Read

When the TRNG module is enabled, the true random noise source and the LFSR post-processing module start to work simultaneously. The software reads out 32 bits of random numbers each time by reading the RNGOUT register. Since the LFSR length is 32 bits, in order to ensure the quality of random numbers, the application should ensure that the interval between two reads of RNGOUT is greater than 32 cycles of TRNG_CLK.

For example, assuming that TRNG_CLK is 4MHz, the interval between two readings to RNGOUT register should not be less than 8us.

17.2.4 CRC Calculation

The LFSR used for post-processing of random numbers can also be used for CRC calculations.

During CRC operation, two sets of 32bit LFSR are respectively used as input data register and CRC operation register, which can calculate the CRC result of 32bit data at one time. Before the CRC calculation, the CPU needs to check whether the current LFSR is occupied. If the LFSR is idle, the CRC function can be used.

Once the CPU starts the CRC operation, the LFSR is automatically set to the reset value, and then 32-bit operation is performed. After the operation is completed, the CRC start register is cleared without interruption; after the software starts the CRC, the status of the start register should be continuously checked until the operation is completed.

CRC polynomial:

CRC32=X32+X26+X23+X22+X16+X12+X11+X10+X8+X7+X5+X4+X2+X1+X0

Software operation process:

- Query LFSR_BUSY to confirm that LFSR is not running
- Write the data to be calculated into CRCDATA0~3
- Set CRC_EN
- Query and wait for CRC_EN to be cleared
- Read calculation results from LFSROUT0~3

17.3 Register

Base address: 0x40013C00

Offset	Name	Symbol
0x00	Random Number Generator Control Register	RNG_CR
0x04	Random Number Generator Data Output Register	RNG_DOR
0x10	Random Number Generator Status Register	RNG_SR
0x14	CRC Control Register	RNG_CRCCR
0x18	CRC Data Input Register	RNG_CRCDIR
0x1C	CRC Status Register	RNG_CRCSR

17.3.1 Random Number Generator Control Register (RNG_CR)

NAME	RNG_CR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							RNGEN
access	U-0							R/W-0

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
0	RNGEN	RNG enable register, software writes 1 to enable (RNG enable) 1: Enable RNG 0: Disable RNG

17.3.2 Random Number Generator Data Output Register (RNG_DOR)

NAME	RNG_DOR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	RNGOUT[31:24]							
access	R							

bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	RNGOUT[23:16]							
access	R							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	RNGOUT[15:8]							
access	R							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RNGOUT[7:0]							
access	R							

bit	name	functional description
31:0	RNGOUT	Random number generation result or CRC operation result register, read only (RNG output)

17.3.3 Random Number Generator Status Register (RNG_SR)

NAME	RNG_SR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						RBUSY	RNF
access	U-0						R-0	R/W-0

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
1	RBUSY	RNG Free Busy Sign 1: RNG is generating random numbers, the software must not use the CRC function at this time 0: RNG is idle
0	RNF	Random number generation failure flag 1: The random number fails the quality test 0: The random number passes the quality inspection

17.3.4 CRC Control Register (RNG_CRCCR)

NAME	RNG_CRCCR							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							CRCEN
access	U-0							R/W-0

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
0	CRCEN	CRC enable control register, software write 1 start CRC, automatically cleared by hardware after CRC calculation is finished (CRC enable) 1: CRC enable 0: CRC disable

17.3.5 CRC Data Input Register (RNG_CRCDIR)

NAME	RNG_CRCDIR							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	CRCIN[31:24]							
access	R/W-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	CRCIN[23:16]							
access	R/W-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CRCIN[15:8]							
access	R/W-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CRCIN[7:0]							
access	R/W-0							

bit	name	functional description
31:0	CRCIN	CRC data input

17.3.6 CRC Status Register (RNG_CRCSR)

NAME	RNG_CRCSR							
Offset	0x1C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							CRCDONE
access	U-0							R/W-0

bit	name	functional description
31:1	-	RFU: **Reserved, read as 0**
0	CRCDONE	CRC calculation done flag, software write 0 to clear 1: CRC calculation was completed 0: CRC calculation was not completed

18 Operational Amplifier (OPA1)

18.1 Introduction

FM33LG0 integrates an operational amplifier, which can be used to amplify weak input signals, or for weak drive signal impedance matching.

The basic characteristics are as follows:

- Input voltage range rail-to-rail
- Typical GBW 2MHz
- Typical power consumption 150uA (normal mode), 2uA (low power consumption mode)
- Maximum drive current 500uA
- Support standalone mode, buffer mode, PGA mode (x2, x4, x8, x16)
- Typical input offset +/-3mv, support user calibration
- OPA output can be connected to ADC for input signal pre-amplification and impedance matching

18.2 Block Diagram

The following figure is a block diagram of a single OPA1:

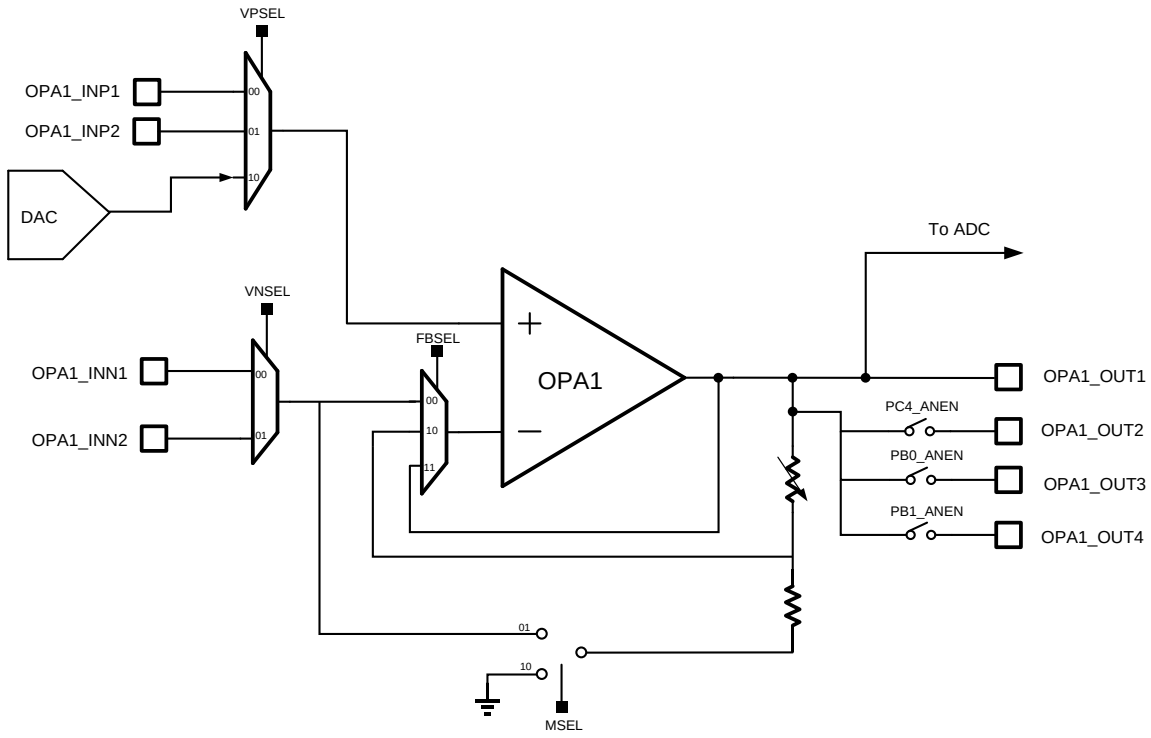


Figure 18-1 OPA1 Circuit Block Diagram

The OPA output is connected to 4 GPIOs, of which OPA1_OUT1 does not pass through an analog switch to optimize the output impedance; the other 3 output channels pass through an analog switch and are closed by default. If you want to use OPA1_OUT2/3/4, you must set the corresponding position in the GPIO module. The ANEN register of the pin.

According to the register configuration, different AMUX channels can be selected to realize different closed-loop applications, such as buffer, PGA (built-in feedback resistor), and independent operational amplifier. The output can be drawn from IO, or connected to ADC, can also generate digital signal or interrupt output.

The following figure is a schematic diagram of the connection relationship when OPA is used as an ADC front-end amplification application:

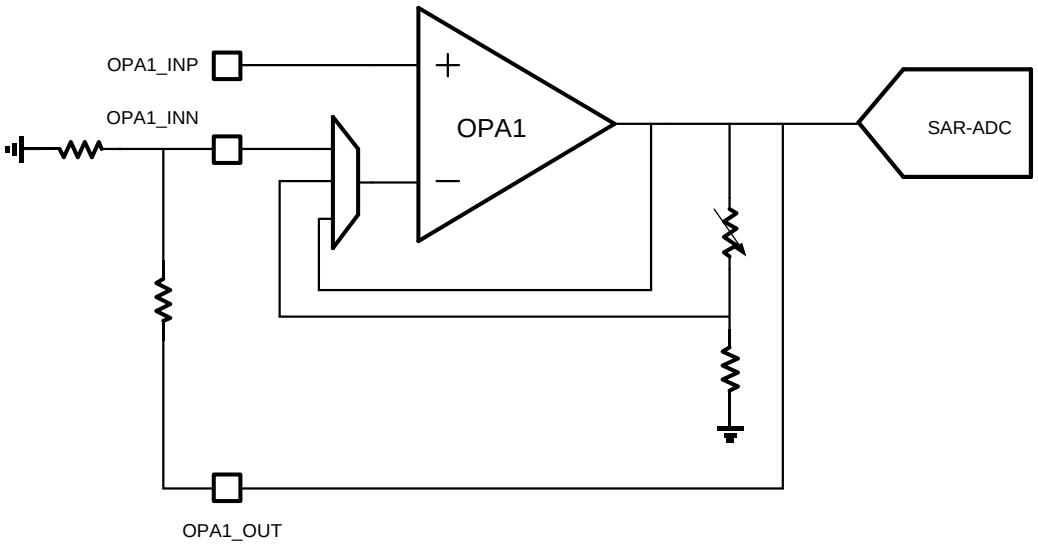


Figure 18-2 OPA Used as ADC Front-End Amplification

18.3 Pin Definition

The OPA module has multiple analog input and output ports, which are multiplexed onto multiple GPIOs.

Pin	OPAx	Symbol	Function
PB10	OPA1	OPA1_INN1	OPA negative input
PB11		OPA1_INP1	OPA positive input
PA6		OPA1_INN2	OPA negative input
PA7		OPA1_INP2	OPA positive input
PB12		OPA1_OUT	OPA output
PC4		OPA1_OUT	OPA output
PB0		OPA1_OUT	OPA output
PB1		OPA1_OUT	OPA output

Table 18-1 OPA Pin Definition

18.4 Functional Description

OPA supports standalone mode, buffer mode and PGA mode (x2, x4, x8, x16)

18.4.1 Clock and Reset

The register clock of the OPA module is provided by the CMU module, and the reset control is provided by the RMU module.

Before operating the OPA module register, the comparator reset must be cleared in the RMU module, and the comparator working clock must be enabled in the CMU.

The OPA itself does not depend on the clock, so it can work in various low-power modes, and the application only needs to complete the register configuration of the OPA before entering the low-power consumption.

18.4.2 Standalone Mode (Non-Inverting Amplifier)

In this mode, the input and output of the OPA are directly connected to the resistance channel of the GPIO of the chip. By connecting the feedback resistor off-chip, the user can flexibly adjust the negative feedback gain of the operational amplifier, as shown in the following figure:

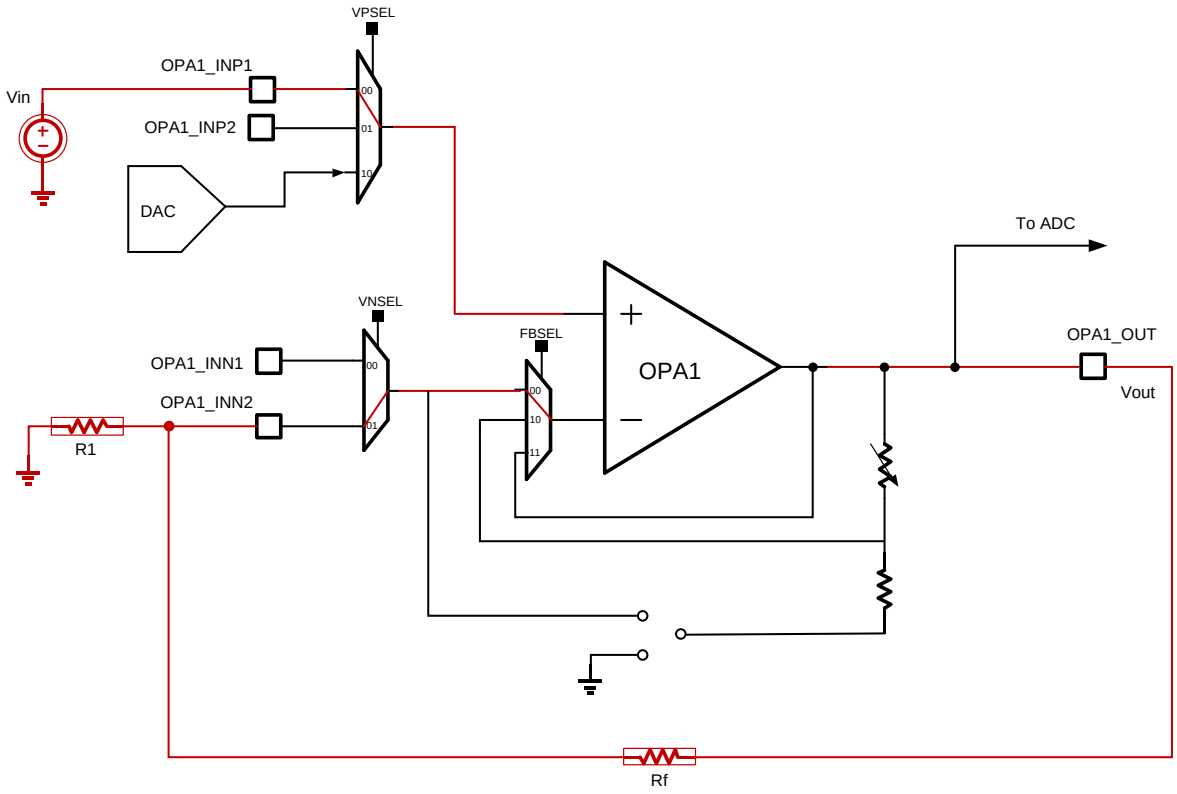


Figure 18-3 OPA Non-Inverting Amplification

The output signal V_{out} is defined by the following formula:

$$V_{out} = (1 + \frac{Rf}{R1}) \times V_{in}$$

Software configuration method:

- Configure OPA1CR.VPSEL and VNSEL to select input IO

- Configure OPA1CR.FBSEL to 00, that is, standalone mode
- Configure OPA1CR.MSEL to 00
- Enable OPA1

18.4.3 Standalone Mode (Inverting Amplifier)

In this mode, the input and output of the OPA are directly connected to the resistance channel of the chip GPIO. By connecting the feedback resistor off-chip, the user can flexibly adjust the negative feedback gain of the op amp, as shown in the following figure:

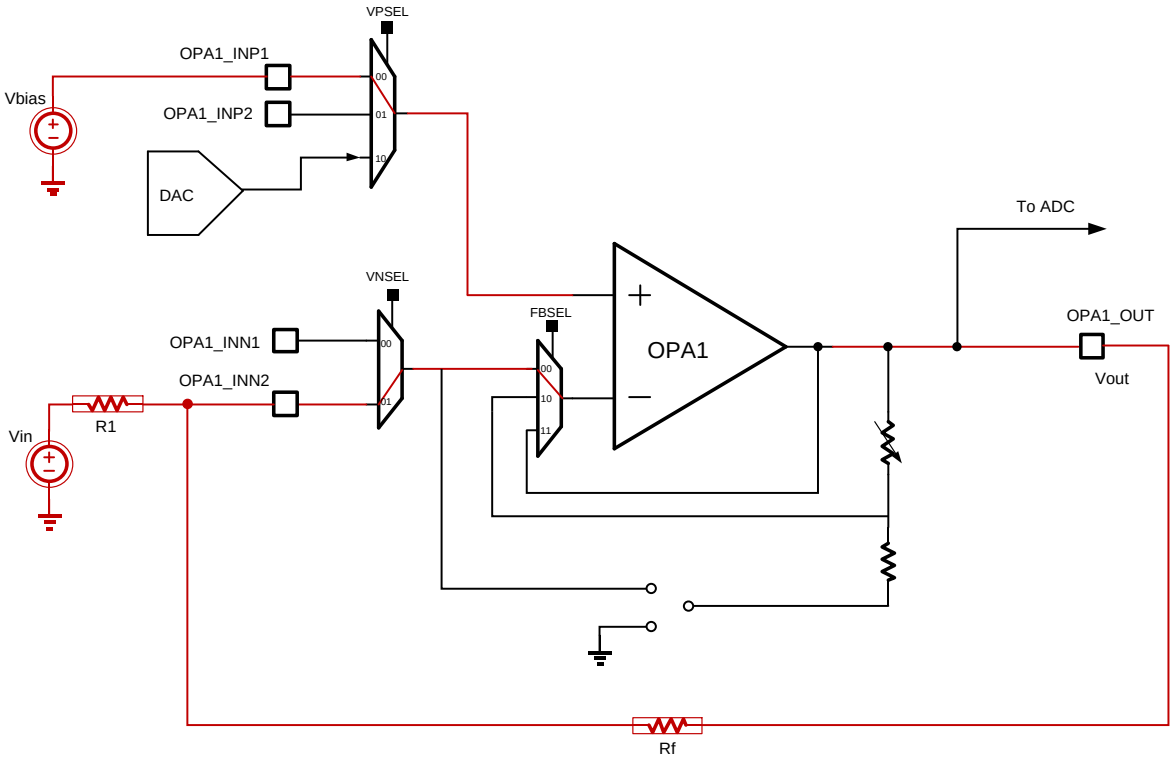


Figure 18-4 OPA Inverted Amplification

The output signal V_{out} is defined by the following formula:

$$V_{out} = V_{bias} + (V_{bias} - V_{in}) \times \frac{Rf}{R1}$$

Software configuration method:

- Configure OPA1CR.VPSEL and VNSEL to select input IO
- Configure OPA1CR.FBSEL to 00, that is, standalone mode
- Configure OPA1CR.MSEL to 00

- Enable OPA1

18.4.4 Buffer Mode

OPA in buffer mode can be used to provide impedance adjustment for ADC input. When the input signal frequency is compatible with OPA's GBW, OPA configured in buffer mode can enhance the drive capability of ADC input signals.

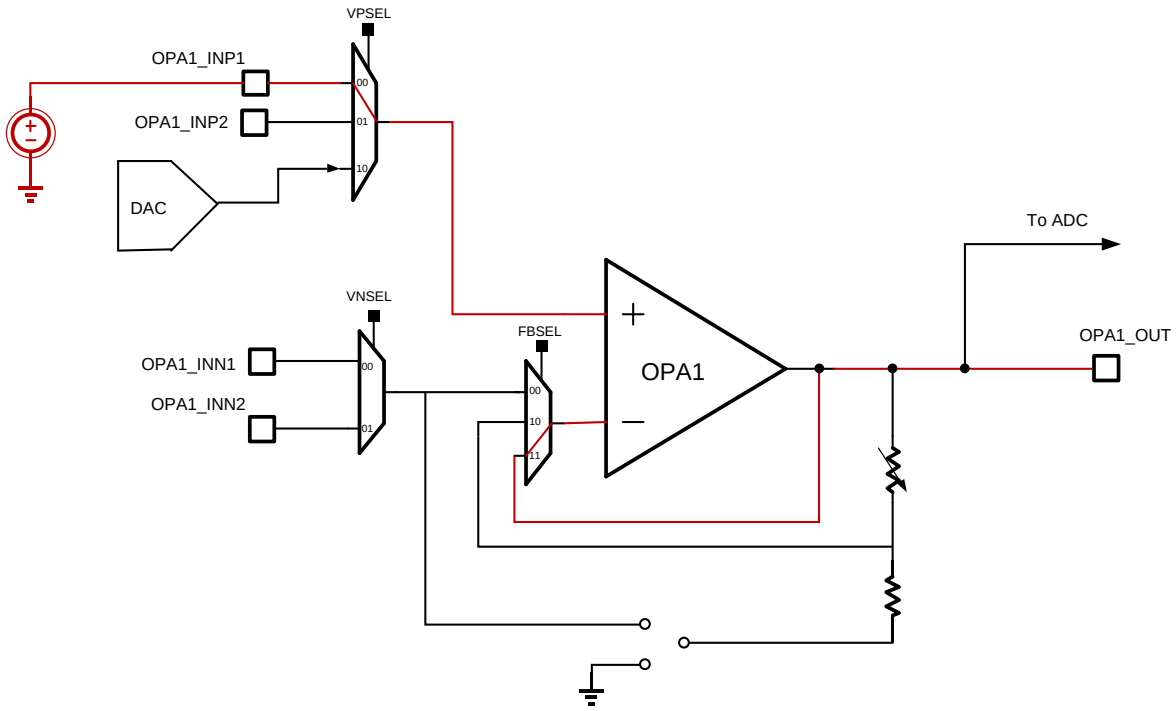


Figure 18-5 OPA Buffer Mode

Software configuration method:

- Configure OPA1CR.VPSEL and VNSEL to select input IO
- Configure OPA1CR.FBSEL to 11, that is, buffer mode
- Configure OPA1CR.MSEL to 00
- Enable OPA1

18.4.5 Non-Inverting PGA Mode

In PGA mode, by adjusting the resistance of the on-chip resistor, a fixed-gain amplification effect can be achieved without connecting an off-chip feedback resistor.

Only OPA1 supports PGA mode, and its supported gains are x2, x4, x8, x16

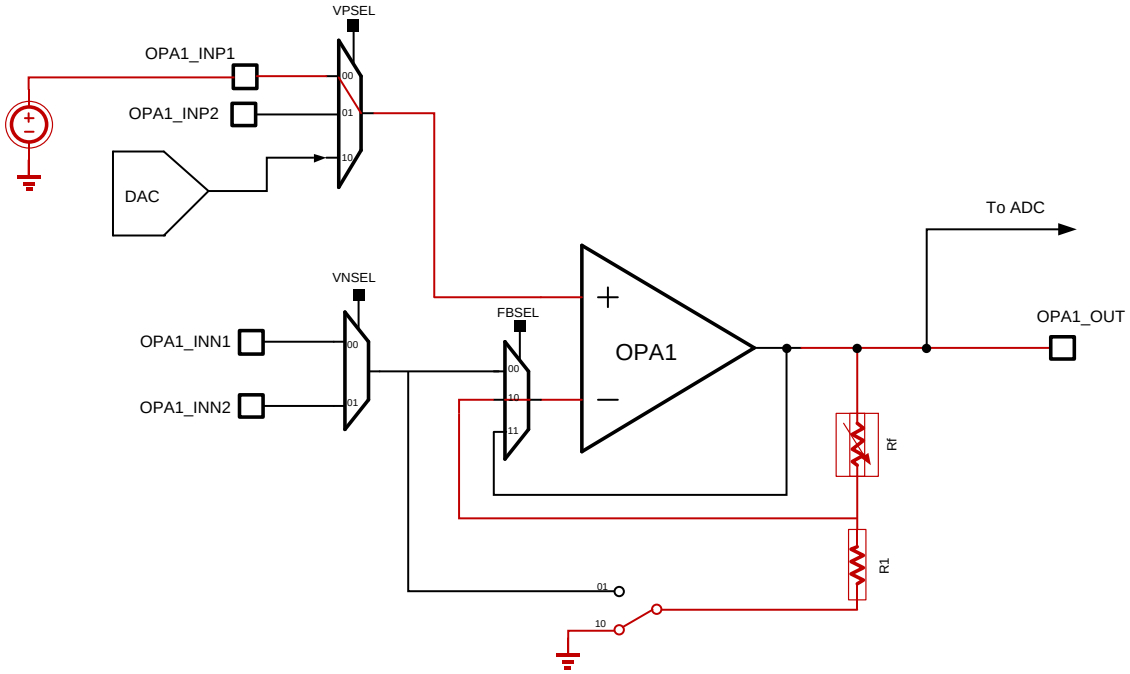


Figure 18-6 Non-Inverting PGA Mode

According to the configuration of PGA_GAIN register, R_f resistance value is 10K, 30K, 70K, 150Kohm, R1 resistance value is fixed to 10Kohm, then the PGA non-inverting gain calculation formula is as follows:

$$Gain_{non-inverting} = \frac{R_f + R_1}{R_1}$$

Therefore, the in-phase magnification of x2, x4, x8, and x16 can be obtained.

By configuring the VN_EXC register and connecting an off-chip capacitor between OPA1_OUT and OPA1_INN, loop filtering can be implemented, as shown in the following figure:

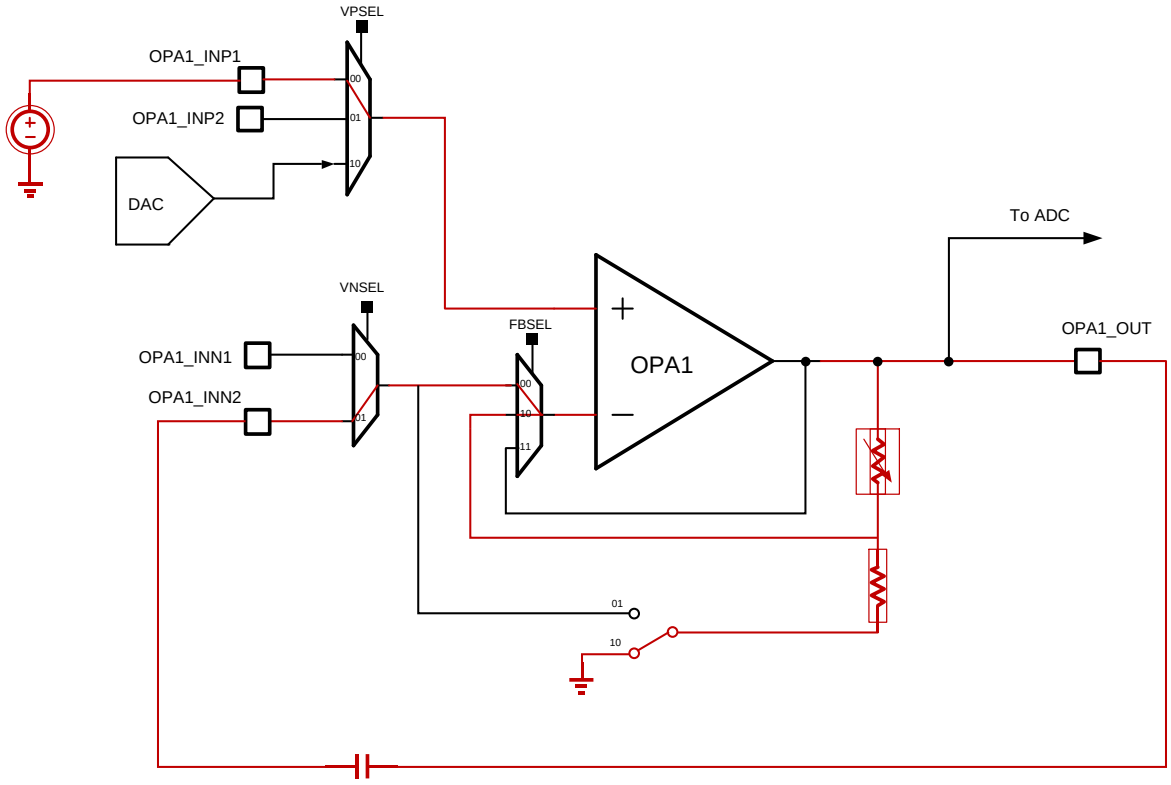


Figure 18-7 OPA Loop Filter

At this time, the PGA gain can be expressed as:

$$Gain_{non-invering} = 1 + \frac{Rf \parallel \frac{1}{2\pi fC}}{R1}$$

Software configuration method:

- Configure OPA1CR.VPSEL and VNSEL to select input IO
- Configure OPA1CR.FBSEL to 10, namely PGA mode
- Configure OPA1CR.MSEL to 10
- Configure OPA1CR.PGA_GAIN to select gain multiple
- If off-chip loop filtering is required, set OPA1CR.VN_EXC
- Enable OPA1

18.4.6 Inverting PGA Mode

In the inverted PGA mode, a DC bias can be applied to the positive terminal of the OPA through the

DAC, as shown in the figure below.

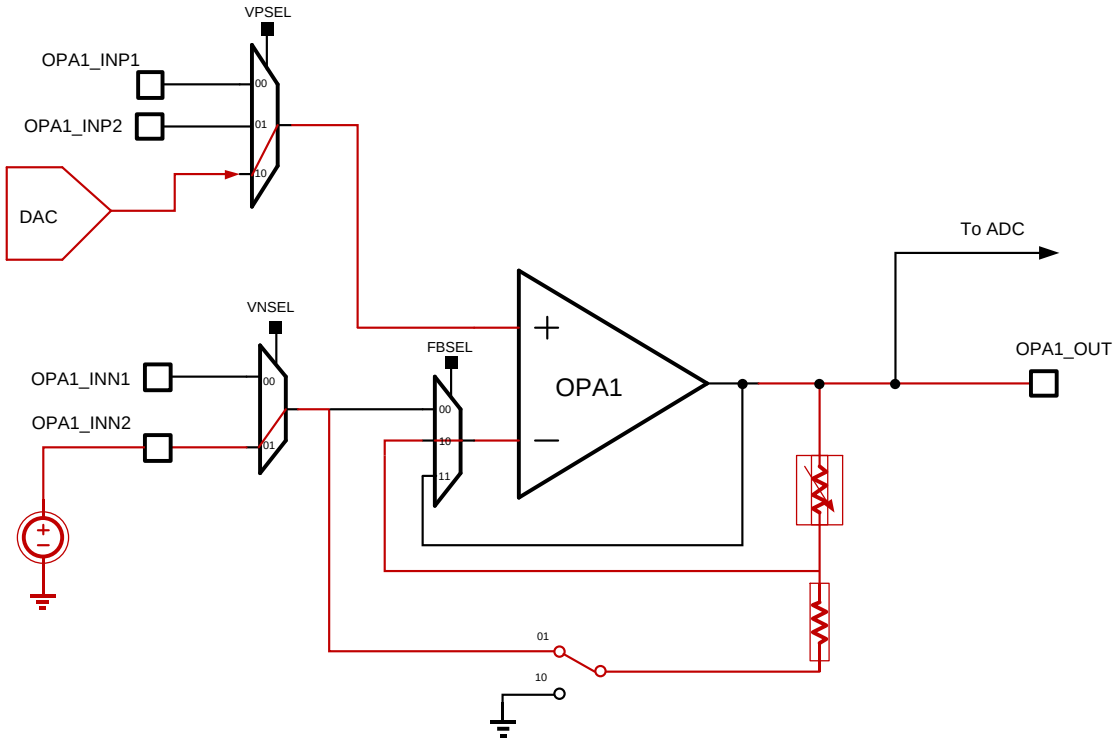


Figure 18-8 Inverted PGA with DAC Bias

According to the configuration of the PGA_GAIN register, the resistance value of R_f is 10K, 30K, 70K, 150Kohm, and the resistance value of R1 is fixed to 10Kohm, then the PGA inverting gain calculation formula is as follows:

$$V_{out} = V_{DAC} + (V_{DAC} - V_{in}) \times \frac{Rf}{R1}$$

Therefore, the inverted magnifications of x1, x3, x7, and x15 can be obtained.

Or you can input the DC bias through an external pin, as shown in the figure below.

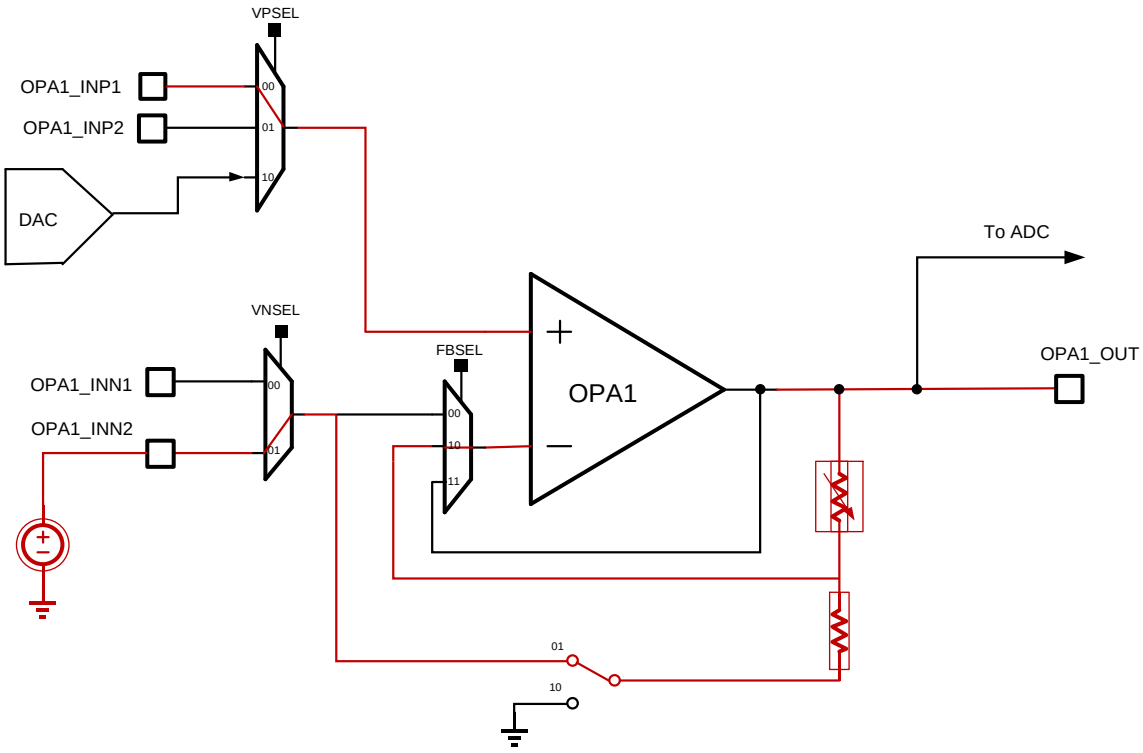


Figure 18-9 Inverted PGA with External Bias

Note: *When using the inverted PGA mode, you need to pay attention to the impedance of the internal switch and the GPIO channel impedance. This impedance is typically around 90 ohms. When there is current flowing from OPA1_OUT through the switch input signal source, the voltage drop on the switch will affect the actual magnification. Among the inverted input channels, only the INN2 channel supports the inverted PGA mode.*

For example, select the inverting magnification x15, then R_f=150Kohm, R1=10Kohm, switch impedance 90Ohm, IO channel impedance 100Ohm, then:

$$V_{out} = V_{DAC} + (V_{DAC} - V_{in}) \times \frac{Rf}{(R1 + R_{sw} + R_{IO})} \approx V_{DAC} + (V_{DAC} - V_{in}) \times 14.72$$

18.4.7 Offset Calibration

The calibration function is used to offset the input offset voltage of the op amp. In order to avoid the impact of packaging and reflow stress, in the occasions that require high op amp offset, it is recommended that users perform board-level calibration after reflow soldering. The calibration is completely realized by software operation. The software compensates the inherent input offset voltage by adjusting the magnitude of the mirror current of the input differential pair, and realizes the calibration by reading the OPA output.

Offset calibration supports two modes: automatic calibration and external calibration. In the automatic

calibration mode, the chip internally generates the input offset voltage, and the external calibration mode requires the user to provide an external offset input.

Auto calibration mode

The user needs to calibrate the N input differential pair and P input differential pair of the op amp respectively according to the following steps.

N differential input calibration steps:

- Configure VPSEL=11, VNSEL=10
- Set TRIM_MODE
- Set NCAL_EN
- Set EN
- Software reads the OUT register
- If OUT=1, write NCAL register as 000000 and gradually increase until OUT changes from 1 to 0
- If OUT=0, write NCAL register as 100000 and gradually increase until OUT changes from 0 to 1
- The calibration is completed and the NCAL data is saved

P differential input calibration steps:

- Configure VPSEL=11, VNSEL=10
- Set TRIM_MODE
- Set PCAL_EN
- Set EN
- Software reads the OUT register
  - If OUT=1, write the PCAL register as 000000 and gradually increase until OUT changes from 1 to 0
  - If OUT=0, write the PCAL register as 100000 and gradually increase until OUT changes from 0 to 1
- The calibration is completed and the PCAL data is saved

Note: *The typical adjustment step size of the calibration circuit is 120uV. After calibration, an input offset voltage of less than +/-100uV can be achieved.*

External calibration mode

The calibration principle is similar to the automatic calibration mode, only TRIM_MODE needs to be cleared, and the required offset voltage is provided on the OPA external input pin. The bias voltage can be selected according to the common-mode voltage in the actual application.

When the applied common mode voltage is less than GND+500mV, short the INN and INP terminals and input the common mode level. At this time, only the PCAL is calibrated.

When the applied common mode voltage is greater than VDDA-500mV, short the INN and INP terminals and input the common mode level. At this time, only NCAL is calibrated.

When the applied common-mode voltage is between GND+500mV and VDDA-500mV, short the INN and INP terminals and input the common-mode level, and calibrate PCAL and NCAL twice.

18.4.8 Low Power Mode

OPA1 can enter a low-power mode to significantly reduce operating power consumption, at this time the bandwidth of the op amp is greatly reduced.

By setting the LPM register, OPA1 enters low power consumption mode. At this time, the unity gain bandwidth of the op amp is reduced to 50Khz, and the power consumption is reduced to about 2uA.

18.4.9 OPA in Sleep Mode

Since the OPA power supply is powered by VDDA, there is no low power consumption limit, and it can keep working in any sleep mode, which is determined by the software configuration.

Note that OPA work requires VREF1p2 to provide bias current, so when entering DeepSleep with OPA enabled, the software cannot turn off VREF1p2.

18.5 Register

Offset	Name	Symbol
OPA(Base address: 0x4001A000)		
0x00	OPA1 Control Register	OPA1_CR
0x04	OPA1 Calibration Register	OPA1_CALR
0x08	OPA1 Calibration Output Register	OPA1_COR

18.5.1 OPA1 Control Register (OPA1_CR)

NAME	OPA1_CR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	MSEL		-		VNSEL		VPSEL	
access	R/W-00		U-0		R/W-111		R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	VN_EXC	PGA_GAIN		FBSEL		LPM	EN
access	U-0	R/W-0	R/W-00		R/W-00		R/W-0	R/W-0

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:14	MSEL	PGA Mode Select 01: The feedback resistor is turned on to the N terminal input switch 10: The feedback resistance to the ground switch is turned on 00,11: RFU
13:12	-	RFU: **Reserved, read as 0**
11:10	VNSEL	OPA1 Negative Input Select 00: OPA1_INN1 01: OPA1_INN2 10,11: RFU Note: When using the inverted PGA function, VNSEL must select channel 01
9:8	VPSEL	OPA1 Positive Input Select 00: OPA1_INP1 01: OPA1_INP2

bit	name	functional description
		10: DAC 11: RFU
7	-	RFU: **Reserved, read as 0**
6	VN_EXC	OPA1 negative terminal is connected to GPIO, only valid when FBSEL=10 (OPA1 Negative Input Connected to GPIO enable) 0: The negative terminal of OPA1 is not connected to GPIO in PGA mode 1: In PGA mode, the negative terminal of OPA1 is connected to GPIO at the same time
5:4	PGA_GAIN	PGA gain select Under the in-phase PGA configuration, the magnification is defined as follows 00: PGA gain x2 01: PGA gain x4 10: PGA gain x8 11: PGA gain x16 In the inverted PGA configuration, the magnification is defined as follows: 00: PGA gain x1 01: PGA gain x3 10: PGA gain x7 11: PGA gain x15
3:2	FBSEL	OPA1 working mode (Feedback Select) 00: standalone mode 01: RFU 10: PGA mode 11: Buffer mode
1	LPM	OPA1 low power control register (OPA1 low power mode) 0: Normal mode 1: Low power consumption mode
0	EN	OPA1 enable register (OPA1 enable) 0: Close OPA1 1: Enable OPA1

18.5.2 OPA1 Calibration Register (OPA1_CALR)

NAME	OPA1_CALR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	NCAL							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PCAL							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					NCAL_EN	PCAL_EN	TRIM_MODE
access	U-0					R/W-0	R/W-0	R/W-0

bit	name	functional description
31:24	-	RFU: **Reserved, read as 0**
23:16	NCAL	OPA1 negative input terminal calibrates trim signal, the highest bit is the Symbol bit (OPA1 negative input calibration) 01111111: Maximum output voltage reduction 00000001: The output voltage decreases the least 00000000: The output voltage remains unchanged 10000000: The output voltage remains unchanged 10000001: Minimum increase in output voltage 11111111: Maximum increase in output voltage Note: NCAL can enter any value from 0 to 11111111
15:8	PCAL	OPA1 positive input terminal calibrates the trim signal, the highest bit is the Symbol bit (OPA1 positive input calibration) 01111111: Maximum output voltage reduction 00000001: The output voltage decreases the least 00000000: The output voltage remains unchanged 10000000: The output voltage remains unchanged 10000001: Minimum increase in output voltage 11111111: Maximum increase in output voltage Note: PCAL can enter any value from 0 to 11111111
7:3	-	RFU: **Reserved, read as 0**
2	NCAL_EN	N differential pair calibration enable 0: Turn off N calibration 1: Enable N calibration
1	PCAL_EN	P differential pair calibration enable 0: Turn off P calibration 1: Enable P calibration
0	TRIM_MODE	Calibration mode selection 0: External calibration mode 1: Automatic calibration mode

18.5.3 OPA1 Calibration Output Register (OPA1_COR)

NAME	OPA1_COR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							OUT
access	U-0							R-x

bit	name	functional description
31:1	-	RFU: **Reserved, read as 0**
0	OUT	OPA output in calibration mode. The software determines whether the offset calibration is successful by polling this signal.

19 Comparator (COMP)

19.1 Introduction

The chip integrates 3 comparators and supports the following features:

- 2 rail-to-rail comparators, support rail-to-rail input, multiple power consumption modes
- 1 low-power comparator, typical power consumption 200nA
- Flexible input options
  - IO pin input
  - Internal reference voltage and its partial pressure and boost
  - DAC output
- Interrupt events can wake up the MCU
- The output signal can be connected to GPIO, or as a trigger source to timer, ADC
- Built-in input reference Buffer
- Window comparator function
- Programmable output digital filter function

19.2 Block Diagram

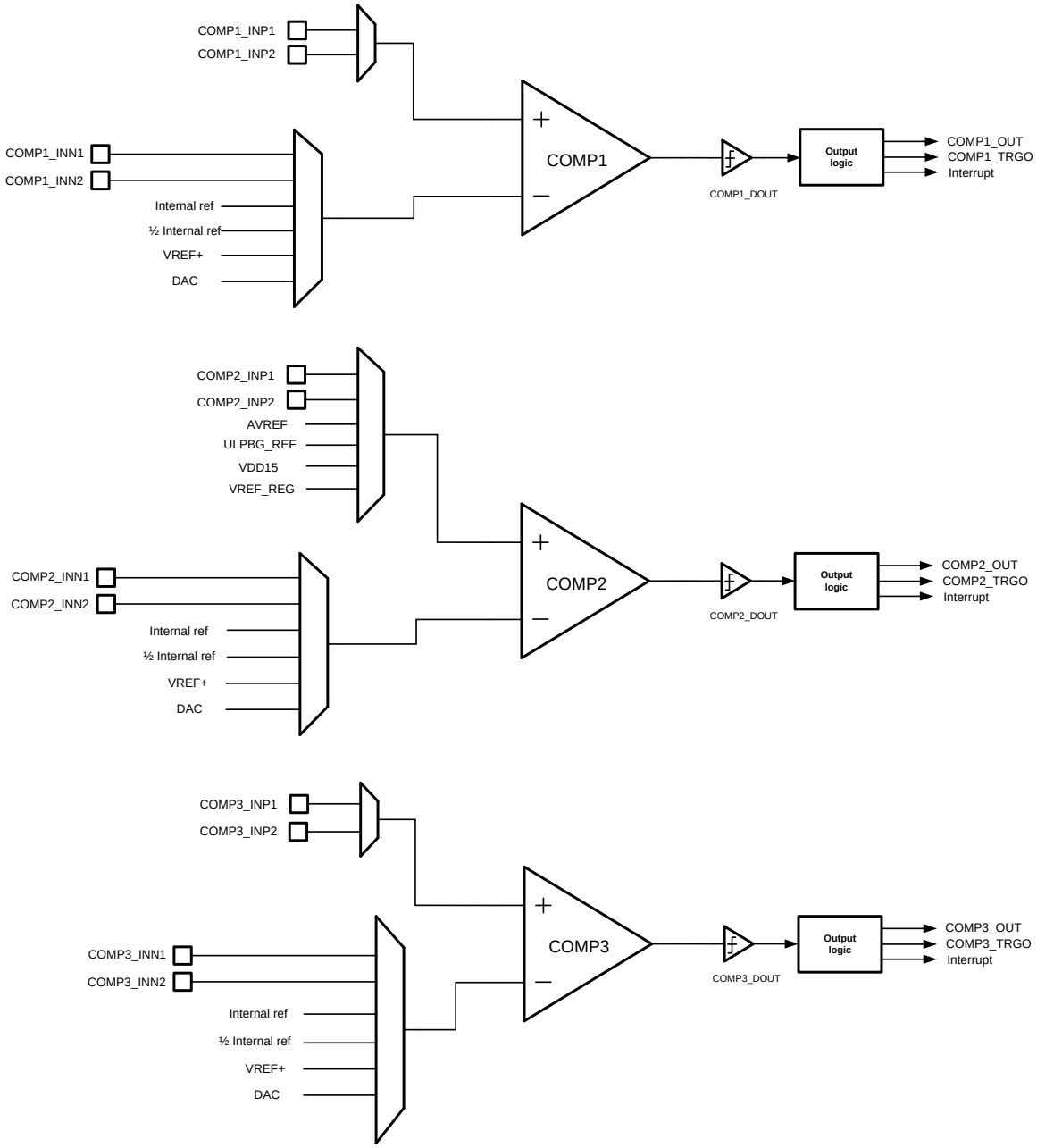


Figure 19-1 Comparator Circuit Block Diagram

The structure of the comparator is shown in the figure above. After the reference voltage passes through the BUFFER module, the reference voltage and the reference voltage are output. The internal ref comes from AVREF or VREF1p2. The software can select the input source of BUFFER. If the comparison base is AVREF or VREF1p2 itself, you can turn off and bypass BUFFER to save power. The comparators 1 and 2 are exactly the same, and the input terminals are connected as shown in the structure diagram. The two comparators generate output signals and output to the digital circuit.

The input voltage range of comparators 1 and 2 is 0~VDD, the setup time is less than 15us, the typical

transmission delay is less than 1us, and the typical power consumption is 2uA.

The input voltage range of comparator 3 is 0~VDD-0.7V, the setup time is less than 10us, the transmission delay is less than 5us, and the typical power consumption is 200nA.

19.3 Functional Description

19.3.1 Basic Function

The comparator compares the positive terminal input voltage and the negative terminal input voltage. When the positive terminal voltage is higher than the negative terminal voltage, it outputs a logic high level, otherwise, it outputs a logic low level.

The logic signal output by the comparator can be output after digital filtering and polarity control, or an interrupt signal can be generated.

The comparator can be configured as a fast mode or a low power consumption mode. The operating current and the transmission delay index of the comparator are different in the two modes, and the appropriate operating mode can be selected according to the application.

19.3.2 Internal Comparison Benchmark Selection

The negative terminal of the comparator can choose to input the internal comparison reference voltage of the chip. The comparison reference comes from VREF1p2 and its partial pressure, AVREF and its partial pressure, and can also choose to use the DAC output voltage as a comparison reference.

When using VREF1p2 and AVREF voltage divider, you must enable the voltage divider buffer and select the buffer input source. Enable the built-in reference source buffer of the comparator module through the BUFSEL and BUFENB registers, and select the input reference.

If BUFENB is set and BUFBYP is set, then the reference voltage bypasses BUFFER and is directly output to the internal ref; note that 1/2 internal ref cannot be used as the comparator input at this time.

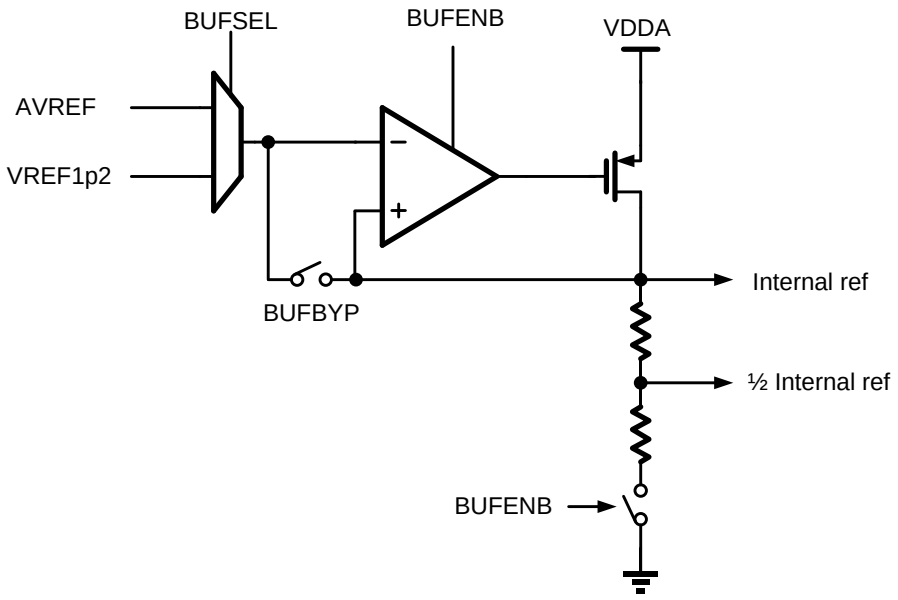


Figure 19-2 Comparator Built-in Reference Buffer

19.3.3 Clock and Reset

The register clock of the comparator module is provided by the CMU module, and the reset control is provided by the RMU module.

Before operating the comparator module registers, the comparator reset must be cleared in the RMU module, and the comparator working clock must be enabled in the CMU.

The comparator itself does not depend on the clock, so it can work in various low-power modes, and the application only needs to complete the register configuration of the comparator before entering the low-power consumption.

19.3.4 Pins and Internal Signal Connections

Comparator input signals can be mapped to the following IO pins. When using external pin input, you need to configure the corresponding GPIO as an analog channel function.

COMP1	Pin	Register
COMP1_INP1	PB8	COMP1_CR.V1PSEL=00
COMP1_INP2	PB9	COMP1_CR.V1PSEL=01
COMP1_INN1	PA10	COMP1_CR.V1NSEL=000
COMP1_INN2	PC6	COMP1_CR.V1NSEL=001
COMP1_OUT	PC4	-

Table 19-1 Comparator 1 Pin List

COMP2	Pin	Register
COMP2_INP1	PA8	COMP2_CR.V2PSEL=000
COMP2_INP2	PA9	COMP2_CR.V2PSEL=001
COMP2_INN1	PA4	COMP2_CR.V2NSEL=000
COMP2_INN2	PA5	COMP2_CR.V2NSEL=001
COMP2_OUT	PC5	-

Table 19-2 Comparator 2 Pin List

COMP3	Pin	Register
COMP3_INP1	PC0	COMP3_CR.V3PSEL=00
COMP3_INP2	PC1	COMP3_CR.V1PSEL=01
COMP3_INN1	PB4	COMP3_CR.V3NSEL=000
COMP3_INN2	PB5	COMP3_CR.V3NSEL=001
COMP3_OUT	PE7	-

Table 19-3 Comparator 3 Pin List

19.3.5 Window Function

The window comparator is used to monitor whether the input analog voltage is within a certain set threshold range. When the input signal level is higher than the high threshold or lower than the low threshold, the output of the comparator is inverted.

The window comparator function needs to use two comparators (COMP1 and COMP2) at the same time. The monitored input voltage is connected to the positive input of the two comparators at the same time, and the negative input of the two comparators is connected to the high threshold and low respectively. Threshold voltage, the output of the two comparators are XORed out. The effective output in window mode is the output signal of comparator 1.

When the WINMODE register is set, the switch in the figure below switches the positive input signal of COMP2 to the positive input signal of COMP1.

The structure of the window comparator is shown in the figure below:

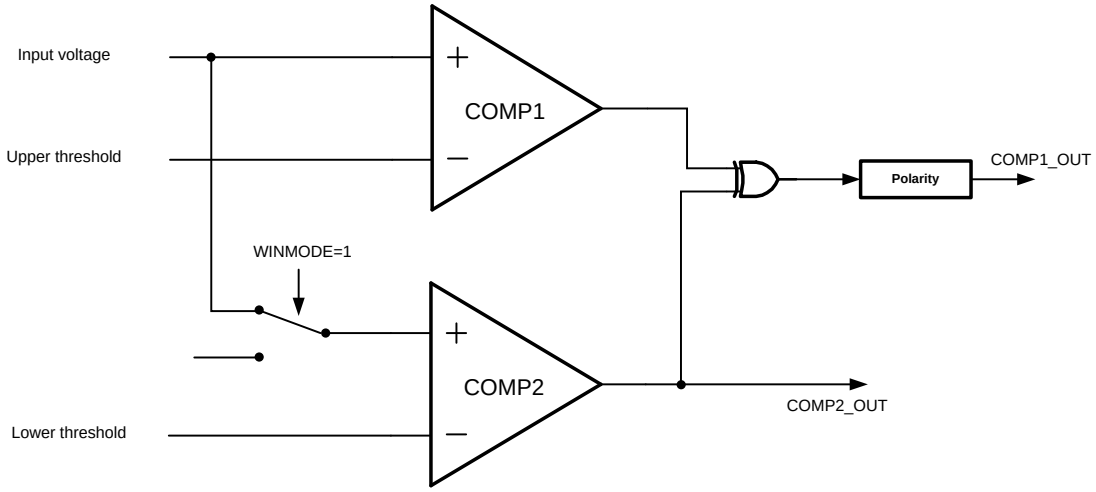


Figure 19-3 Window Comparator Block Diagram

The waveform diagram of the window comparator when it is working is as follows. When the input voltage exceeds the range set by the upper and lower thresholds, the output signal is inverted and an OOW_IF (Out-Of-Window) interrupt is generated. The WIN_IF interrupt is generated every time the input signal crosses the threshold.

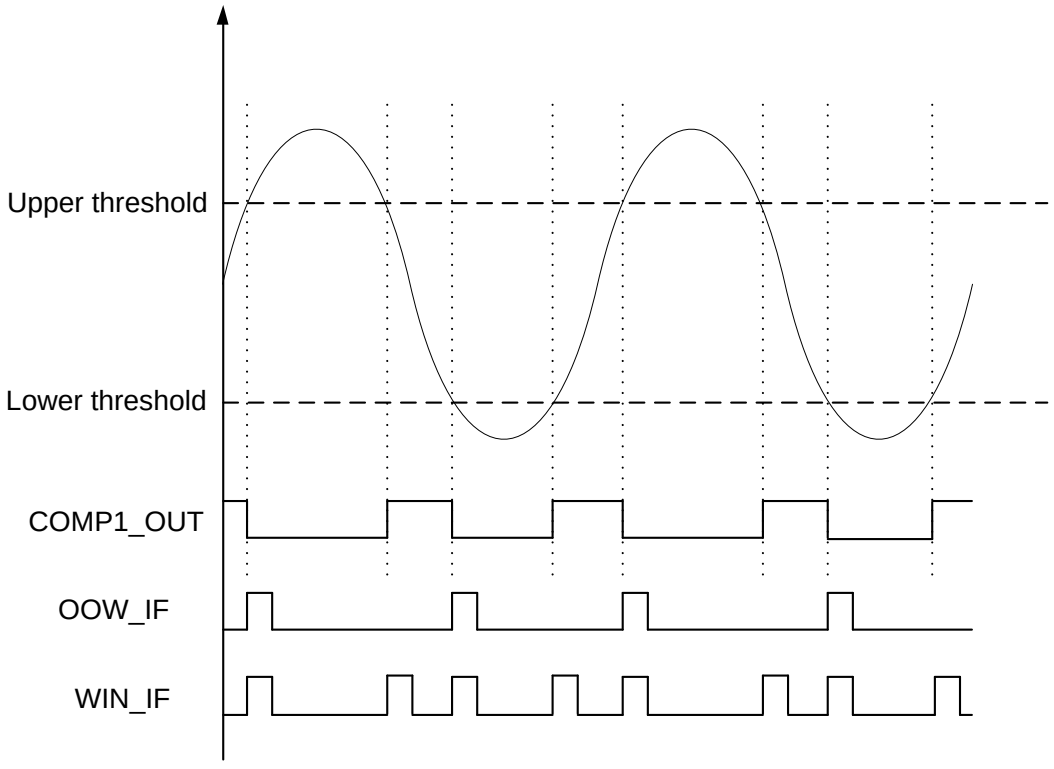


Figure 19-4 Schematic Diagram of Window Comparator Waveform

The setting of the window mode interrupt flag can choose to use the COMP1 output signal after digital filtering or without digital filtering. When the input signal changes very slowly, the comparator output may frequently flip around the comparison threshold. Digital filtering can avoid repeated setting of the OOW_IF and WIN_IF registers.

If you still want to use the window function in sleep mode, you need to turn off the digital filter function of the comparator output.

19.3.6 Power Consumption and Speed Mode

Comparator 1 and 2 support rail-to-rail input, with an input voltage range of 0~VDD, support low-power mode, medium-speed mode, and high-speed mode to balance power consumption and delay in different applications.

After the chip is powered on and reset, the two comparators are in low-power mode by default, and the application can configure their working modes as needed.

Mode	Typical power consumption	Typical propagation delay	Typical setup time	Conditions of use
High speed mode		50ns		Can be used in all chip power modes
Medium speed mode	<2uA	0.6us	5us	
Low power mode	<1uA	2us	15us	

Table 19-4 Comparator 1/2 Working Mode

Comparator 3 is a low-power comparator, with an input voltage range of 0~VDD-0.7V, a setup time of less than 10us, a transmission delay of less than 5us, and a typical power consumption of 200nA.

Mode	Typical power consumption	Typical propagation delay	Typical setup time	Conditions of use
Low power mode	0.2uA	5us	10us	Can be used in all chip power modes

Table 19-5 Comparator 3 Working Mode

19.3.7 Comparator Interrupt

The output of the comparator can generate independent interrupt events on the rising and falling edges. The CMPxIE register can enable or disable interrupt output. The CMPxIF flag register is set when an interrupt event occurs, and is cleared by software writing 1. Software can also directly read

the output value of the comparator through the CMPxO register.

In the window comparator mode, according to the voltage range of the input analog signal, a window threshold interrupt and a window out interrupt can be generated.

The edge detection of the output of the comparator is realized by an asynchronous circuit without a working clock, so an interrupt can be generated when the chip is in sleep mode to wake up the chip.

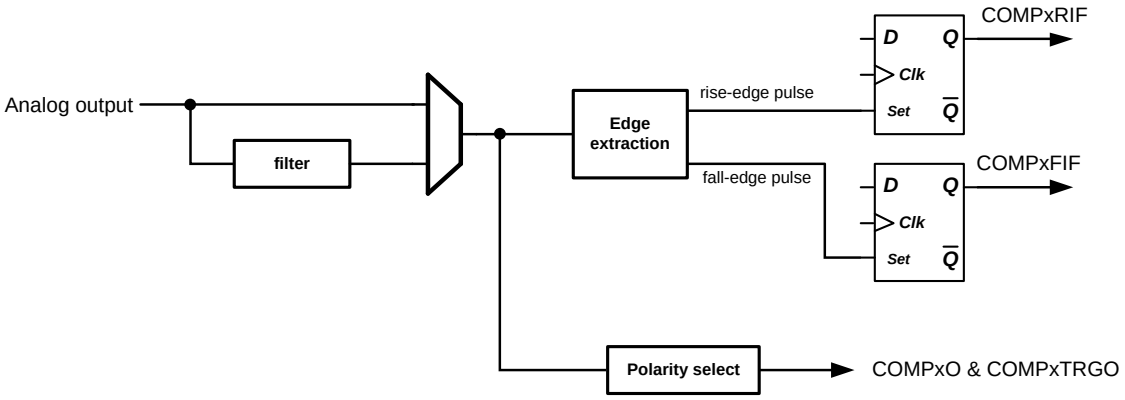


Figure 19-5 Comparator Interrupt Generation

19.3.8 Comparator Output and Trigger Output

The comparator can directly output the comparison result, and can also output trigger signals to other peripheral circuits.

The output logic diagram is as follows.

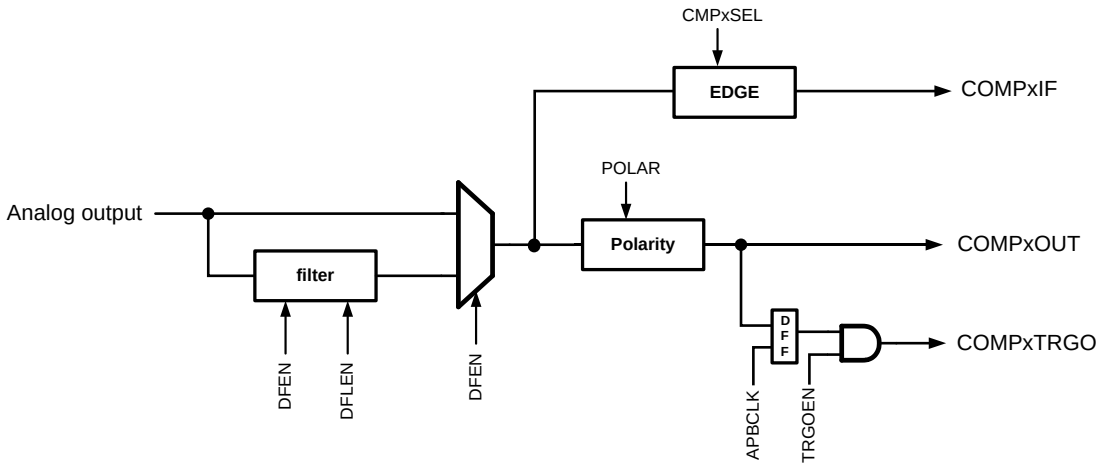


Figure 19-6 Comparator Output Logic

Comparator output

Comparator comparison result (COMPxOUT) can be queried by software in real time, and can also be output to GPIO. The output signal can choose whether to go through digital filtering.

Comparator trigger signal output

The trigger signal output can be generated on the rising edge and falling edge of the comparator output respectively. When the trigger signal needs to be output, the COMP bus clock must be enabled. When a trigger event occurs, the trigger signal is synchronized to the rising edge of APBCLK. The trigger signal can be connected to the internal trigger input of the timer or the internal trigger input of the ADC.

TRGOEN is used to enable or disable the trigger signal output.

The following figure shows an example of a trigger signal generated by the rising edge of the comparator output.

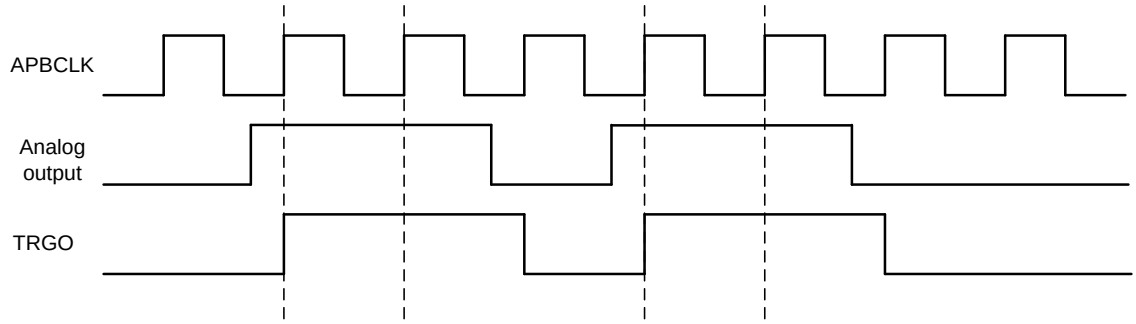


Figure 19-7 The Rising Edge of Comparator Output Generates a Trigger Output Without Filtering

The trigger output of the comparator can be connected to the input of GPTIMx, so that the timer can automatically record the number of times the comparator output flips.

For the specific connection relationship, please refer to 30.4.4 Capture of Internal Trigger Signal (ITRx).

The trigger output of the comparator can also be used as the trigger signal to start ADC conversion, please refer to 38.6.11 Conversion Trigger.

19.3.9 Output Digital Filter

The comparator output supports digital filtering. The filtering method is to use APBCLK to continuously sample the output of the comparator, and only when the number of cycles defined by DFLEN is maintained at the same level, the level is considered valid.

In sleep mode, the digital filter function cannot be used because APBCLK is turned off. If the

comparator needs to be used in sleep mode in the application, the software must turn off the digital filter function of the comparator before sleep.

The following figure is a schematic diagram of DFLEN=2:

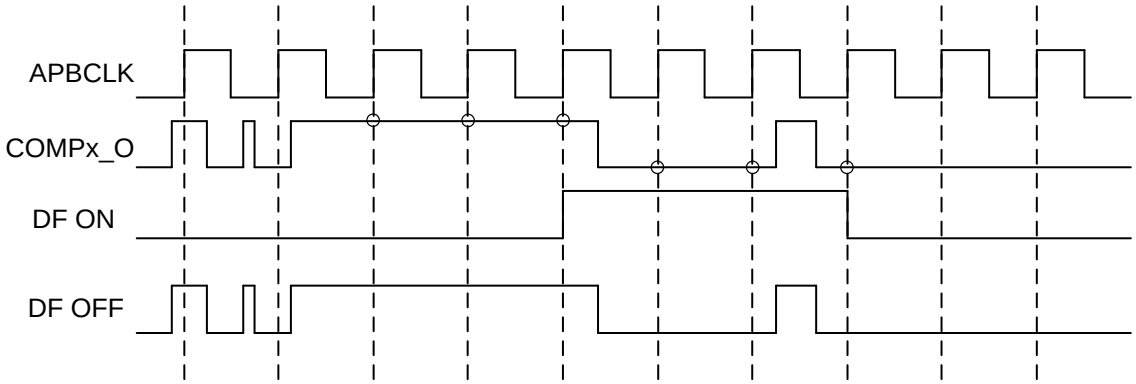


Figure 19-8 Digital Filtering (DFLEN=3) Waveform Diagram

19.4 Register

Offset	Name	Symbol
COMP(Base address: 0x40015400)		
0x00	Comparator Control Register 1	COMP1_CR
0x04	Comparator Control Register 2	COMP2_CR
0x08	Comparator Control Register 3	COMP3_CR
0x0C	Comparator Interrupt Config Register	COMP_ICR
0x10	Comparator Interrupt Status Register	COMP_ISR
0x14	Comparator Buffer Control Register	COMP_BUFCR

19.4.1 COMP1 Control Register (COMP1_CR)

NAME	COMP1_CR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							TRGOEN
access	U-0							R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DFLEN					WINMODE	POLAR	DFEN
access	R/W-00000					R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	MODE		-					CMP1O
access	R/W-00		U-0					R
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		V1PSEL		V1NSEL			CMP1EN
access	U-0		R/W-00		R/W-00			R/W-0

bit	name	functional description
31:25	--	RFU: **Reserved, read as 0**
24	TRGOEN	Comparator 1 trigger signal output enable 0: Disable trigger output 1: Allow trigger output
23:19	DFLEN	Comparator 1 outputs the digital filter length configuration register. The filter length period is DFLEN+1 (the minimum filter length is 3) 00000: 3 APBCLK samples 00001: 3 APBCLK samples 00010: 3 APBCLK samples 00011: 4 APBCLK samples ...

bit	name	functional description
		11111: 32 APBCLK samples
18	WINMODE	Comparator 1&2 window mode control register 0: Disable window mode 1: Enable window mode
17	POLAR	Comparator 1 output polarity control 0: Positive output 1: Inverted output
16	DFEN	Comparator 1 output digital filter enable 0: Disable output digital filter 1: Enable output digital filtering
15:14	MODE	Comparator 1 working mode 00/11: Low power consumption mode 01: Medium speed mode 10: High-speed mode
13:9	--	RFU: **Reserved, read as 0**
8	CMP1O	Comparator 1 output, software read only
7:6	--	RFU: **Reserved, read as 0**
5:4	V1PSEL	Comparator 1 positive input selection 00: COMP1_INP1 01: COMP1_INP2 10: RFU 11: RFU
3:1	V1NSEL	Comparator 1 negative input selection 000: COMP1_INN1 001: COMP1_INN2 010: internal reference 011: 1/2 (internal reference) 100: VREFP 101: DAC 110: RFU 111: RFU
0	CMP1EN	Comparator 1 enable bit 0: Turn off the comparator 1 1: Enable comparator 1

19.4.2 COMP2 Control Register (COMP2_CR)

Name	**COMP2_CR**							
offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							TRGOEN

access	U-0							R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DFLEN					-	POLAR	DFEN
access	R/W-00000					U-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	MODE		-					CMP2O
access	R/W-00		U-0					R
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	V2PSEL			V2NSEL			CMP2EN
access	U-0	R/W-000			R/W-000			R/W-0

bit	name	functional description
31:25	--	RFU: **Reserved, read as 0**
24	TRGOEN	Comparator 2 trigger signal output enable 0: Disable trigger output 1: Allow trigger output
23:19	DFLEN	Comparator 2 outputs the digital filter length configuration register. Filter length period is DFLEN+1 00000: 3 APBCLK samples 00001: 3 APBCLK samples 00010: 3 APBCLK samples 00011: 4 APBCLK samples ... 11111: 32 APBCLK samples
18	--	RFU: **Reserved, read as 0**
17	POLAR	Comparator 2 output polarity control 0: Positive output 1: Inverted output
16	DFEN	Comparator 2 output digital filter enable 0: Disable output digital filter 1: Enable output digital filtering
15:14	MODE	Comparator 2 working mode 00/11: Low power consumption mode 01: Medium speed mode 10: High-speed mode
13:9	--	RFU: **Reserved, read as 0**
8	CMP2O	Comparator 2 output, software read only
7	--	RFU: **Reserved, read as 0**
6:4	V2PSEL	Comparator 2 positive input selection 000: COMP2_INP1 001: COMP2_INP2 010: AVREF 011: ULPBG_VREF(1.16V) 100: VDD15

bit	name	functional description
		101: VREFP
3:1	V2NSEL	Comparator 2 negative input selection 000: COMP2_INN1 001: COMP2_INN2 010: Internal reference 011: 1/2 (internal reference) 100: VREFP 101: DAC 110: RFU 111: RFU
0	CMP2EN	Comparator 2 enable bit 0: Turn off the comparator 2 1: Enable comparator 2

19.4.3 COMP3 Control Register (COMP3_CR)

NAME	COMP3_CR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							TRGOEN
access	U-0							R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DFLEN					-	POLAR	DFEN
access	R/W-00000					U-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							CMP3O
access	U-0							R
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		V3PSEL		V3NSEL			CMP3EN
access	U-0		R/W-00		R/W-000			R/W-0

bit	name	functional description
31:25	--	RFU: **Reserved, read as 0**
24	TRGOEN	Comparator 3 trigger signal output enable 0: Disable trigger output 1: Allow trigger output
23:19	DFLEN	Comparator 3 outputs the digital filter length configuration register. Filter length period is DFLEN+1 00000: 3 APBCLK samples 00001: 3 APBCLK samples 00010: 3 APBCLK samples

bit	name	functional description
		00011: 4 APBCLK samples ... 11111: 32 APBCLK samples
18	--	RFU: **Reserved, read as 0**
17	POLAR	Comparator 3 output polarity control 0: Positive output 1: Inverted output
16	DFEN	Comparator 3 output digital filter enable 0: Disable output digital filter 1: Enable output digital filtering
15:9	--	RFU: **Reserved, read as 0**
8	CMP3O	Comparator 3 output, software read only
7:6	--	RFU: **Reserved, read as 0**
5:4	V3PSEL	Comparator 3 positive input selection 00: COMP3_INP1 01: COMP3_INP2 10,11: RFU
3:1	V3NSEL	Comparator 3 negative input selection 000: COMP3_INN1 001: COMP3_INN2 010: Internal reference 011: 1/2 (internal reference) 100: VREFP 101: DAC 110: RFU 111: RFU
0	CMP3EN	Comparator 3 enable bit 0: Turn off the comparator 3 1: Enable comparator 3

19.4.4 Comparator Interrupt Config Register (COMP_ICR)

NAME	COMP_ICR							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-						OOW_IE	WIN_IE
access	U-0						R/W-0	R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				CMP3SEL		-	CMP3IE
access	U-0				R/W-00		U-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				CMP2SEL		-	CMP2IE

access	U-0				R/W-00		U-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				CMP1SEL		-	CMP1IE
access	U-0				R/W-00		U-0	R/W-0

bit	name	functional description
31:26	--	RFU: **Reserved, read as 0**
25	OOW_IE	Out-Of-Window interrupt enable
24	WIN_IE	Window interrupt enable
23:20	--	RFU: **Reserved, read as 0**
19:18	CMP3SEL	Comparator 3 interrupt source selection 00/11: Comparator 3 output rising or falling edge generates interrupt 01: Comparator 3 generates an interrupt on the rising edge of the output 10: Comparator 3 output falling edge generates an interrupt
17	--	RFU: **Reserved, read as 0**
16	CMP3IE	Comparator 3 interrupt enable, 1 is valid
15:12	--	RFU: **Reserved, read as 0**
11:10	CMP2SEL	Comparator 2 interrupt source selection 00/11: Comparator 2 output rising or falling edge generates an interrupt 01: Comparator 2 generates an interrupt on the rising edge of the output 10: Comparator 2 output falling edge generates an interrupt
9	--	RFU: **Reserved, read as 0**
8	CMP2IE	Comparator 2 interrupt enable, 1 is valid
7:4	--	RFU: **Reserved, read as 0**
3:2	CMP1SEL	Comparator 1 interrupt source selection 00/11: Comparator 1 output rising or falling edge generates interrupt 01: Comparator 1 output rising edge generates interrupt 10: Comparator 1 output falling edge generates an interrupt
1	--	RFU: **Reserved, read as 0**
0	CMP1IE	Comparator 1 interrupt enable, 1 is valid

***Note:** In order to avoid false triggering of the interrupt, the interrupt source selection register should be set when the interrupt enable is turned off.

19.4.5 Comparator Interrupt Status Register (COMP_ISR)

Name	COMP_IF							
offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			OOW_IF	WIN_IF	CMP3IF	CMP2IF	CMP1IF
access	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:5	--	RFU: **Reserved, read as 0**
4	OOW_IF	Out-of-Window interrupt flag，this flag is set by hardware and cleared by software by writing 1
3	WIN_IF	Window interrupt flag，this flag is set by hardware and cleared by software by writing 1
2	CMP3IF	Comparator 3 interrupt flag，this flag is set by hardware and cleared by software by writing 1
1	CMP2IF	Comparator 2 interrupt flag，this flag is set by hardware and cleared by software by writing 1
0	CMP1IF	Comparator 1 interrupt flag，this flag is set by hardware and cleared by software by writing 1

19.4.6 Comparator Buffer Control Register (COMP_BUFCR)

NAME	COMP_BUFCR							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							

bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					BUFBYP	BUFSEL	BUFENB
access	U-0					R/W-0	R/W-0	R/W-1

bit	name	functional description
31:3	--	RFU: **Reserved, read as 0**
2	BUFBYP	Buffer bypass enable 0: Do not bypass 1: Bypass
1	BUFSEL	Buffer input reference selection 0: AVREF 1: VREF1p2
0	BUFENB	Buffer enable 0: Enable reference buffer 1: Turn off the reference buffer

20 Division/Squaring Accelerator (DIVAS)

20.1 Introduction

The DIVAS module is used to help the software accelerate division and square root operations. The hardware circuit includes an integer divider with Symbol number, which can input 32bit dividend and 16bit divisor, output 32bit quotient and 16bit remainder; and a non-Symbol integer square extraction circuit, input 32bit without Symbol number, and output 16bit square extraction result.

Basic characteristics:

- Symbol integer division operation (two's complement format)
- 32bit dividend, 16bit divisor
- 32bit quotient and 16bit remainder
- Divide by 0 warning
- A division calculation requires 16 clock cycles
- Integer square root operation without Symbol
- 32bit square root number, 16bit result
- A square root calculation requires 16 clock cycles

20.2 Clock and Reset

The register clock of the DIVAS module is provided by the CMU module, and the reset control is provided by the RMU module.

Before operating the DIVAS module register, you must clear the DIVAS reset (DIVASRST) in the RMU module and enable the DIVAS clock (DIVAS_PCE) in the CMU.

20.3 Workflow of Hardware Division

The software calls the hardware divider according to the following steps.

- Clear the MODE register
- Write 32bit dividend (twos complement) to OPRD register
- Write a 16bit divisor (two's complement) to the DIVSOR register

- After writing to DIVSOR, the hardware divider automatically starts to work, and the BUSY register is set at the same time
- The software queries the BUSY flag, and BUSY is automatically cleared after the calculation is completed
- Query the DIV_BY_0 flag
- Read the quotient in the QUOT register
- Read the remainder in the REMD register

20.4 Hardware Prescription Workflow

The software calls the hardware squarer according to the following steps.

- Set the MODE register
- Write 32bit square root to OPRD register
- The hardware square dispenser automatically starts to work, and the BUSY register is set at the same time
- The software queries the BUSY flag, and BUSY is automatically cleared after the calculation is completed
- Read the root in the ROOT register

20.5 Register

Base address: 0x40019C00

Offset	Name	Symbol
DIVAS(Base address: 0x40019C00)		
0x00	Operand Register	DIVAS_OPRD
0x04	Divisor Register	DIVAS_DIVSOR
0x08	Quotient Register	DIVAS_QUOT
0x0C	Reminder Register	DIVAS_REMD
0x10	Root Register	DIVAS_ROOT
0x14	Status Register	DIVAS_SR
0x18	Control Register	DIVAS_CR

20.5.1 Operand Register (DIVAS_OPRD)

NAME	DIVAS_OPRD							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	OPRD[31:24]							
access	R/W-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	OPRD[23:16]							
access	R/W-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	OPRD[15:8]							
access	R/W-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	OPRD[7:0]							
access	R/W-00000000							

bit	name	functional description
31:0	OPRD	Operand register In the division operation, save the 32-bit dividend with Symbol During the square root operation, save the 32-bit square root number without Symbol

20.5.2 Divisor Register (DIVAS_DIVSOR)

NAME	DIVAS_DIVSOR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DIVSOR[15:8]							
access	R/W-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DIVSOR[7:0]							
access	R/W-00000001							

bit	name	functional description
31:0	--	RFU: **Reserved, read as 0**
15:0	DIVSOR	16bit divisor (has symbol)

20.5.3 Quotient Register (DIVAS_QUOT)

NAME	DIVAS_QUOT							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	QUOT[31:24]							
access	R							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	QUOT [23:16]							
access	R							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	QUOT [15:8]							
access	R							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	QUOT [7:0]							
access	R							

bit	name	functional description
31:0	QUOT	32bit has Symbol quotient (Address only, no actual register, directly return to the output of the division module when reading)

20.5.4 Reminder Register (DIVAS_REMD)

NAME	DIVAS_REMD							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	REMD[15:8]							
access	R							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	REMD[7:0]							
access	R							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	REMD	16bit remainder (has symbol) (Address only, no actual register, directly return to the output of the division module when reading)

20.5.5 Root Register (DIVAS_ROOT)

NAME	DIVAS_ROOT							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ROOT[15:8]							
access	R							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ROOT[7:0]							
access	R							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	ROOT	16bit square root without Symbol (Only address, no actual register, directly return to the output of the prescribing module when reading)

20.5.6 Status Register (DIVAS_SR)

NAME	DIVAS_SR							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						DIV_BY_0	BUSY
access	U-0						R-0	R-0

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
1	DIV_BY_0	Divide by 0 sign 1 = divide by 0 0 = Divisor is not 0
0	BUSY	Operation instruction 1 = DIVAS is in the process of calculation, the result is not ready 0 = The calculation is complete and the result is ready After the software writes the operand or the divisor, DIVAS starts to calculate, and the software should query the BUSY to be low before reading the quotient, remainder or root register A single calculation requires 16 clock cycles

20.5.7 Control Register (DIVAS_CR)

NAME	DIVAS_CR							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							

access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							MODE
access	U-0							R/W-0

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
0	MODE	Working mode control 0: Hardware divider 1: Hardware square solver

21 Inter-integrated Circuit (I2C) Interface

21.1 Introduction

The I2C module realizes the synchronous communication between the MCU and the external I2C interface device, and the hardware realizes serial-to-parallel conversion. Support I2C master and slave mode, does not support multi-master mode.

Features:

- 1 independent I2C interface
- Supports master and slave mode, does not support multi-master mode
- Support 7-bit or 10-bit slave address
- The transmission speed supports standard mode (100Kbps), fast mode (400Kbps) and Fm+ (1Mbps)
- Support DMA, independent DMA channel of master and slave
- Low-power slave design, which can send and receive data without a system clock
- Support asynchronous slave address matching wake-up, data frame receiving completion wake-up or START detection wake-up

21.2 Block Diagram

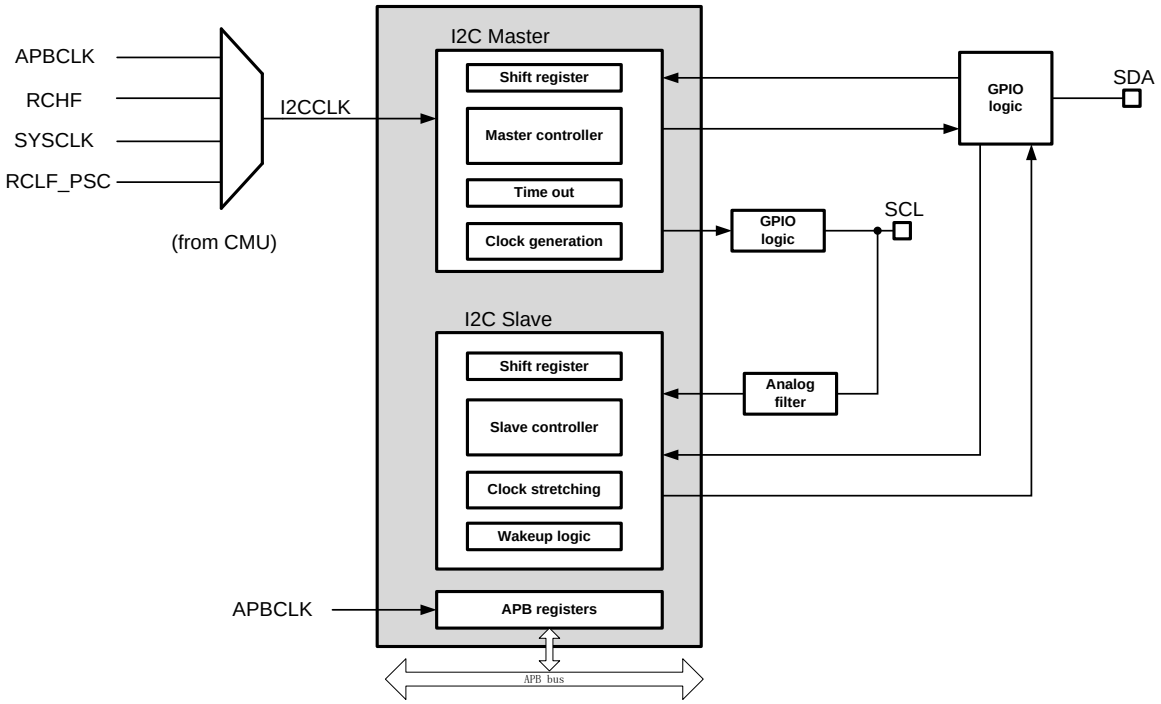


Figure 21-1 I2C Block Diagram

21.3 Pin Definition and Pull-up Resistance Range

Pin	I2Cx	Symbol	Function	Type
PA11	I2C	SCL	I2CClock	High-sink(20mA)
PA12		SDA	I2CData	
PB15		SCL	I2CClock	Normal
PD12		SDA	I2CData	

Table 21-1 I2C Pin Definition

The I2C bus protocol specifies the maximum rise time of standard-mode, fast-mode and fast-mode plus signals, and the minimum sink current that IO can support. See the table below.

Symbol	Parameter	Conditions	Standard-mode		Fast-mode		Fast-mode Plus		Unit
			Min	Max	Min	Max	Min	Max	
V_{IL}	LOW-level input voltage		–0.5	$0.3V_{DD}$	0.5	$0.3V_{DD}$	0.5	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	[2]	$0.7V_{DD}$	[2]	$0.7V_{DD}$	[2]	V
V_{hys}	Hysteresis of Schmitt trigger inputs		-	-	$0.05V_{DD}$	-	$0.05V_{DD}$	-	V
V_{OL1}	LOW-level output voltage 1	(open-drain or open-collector) at 3 mA sink current; V_{DD}>2V	0	0.4	0	0.4	0	0.4	V
V_{OL2}	LOW-level output voltage 2	(open-drain or open-collector) at 2 mA sink current; $V_{DD}\leq$2V	-	-	0	0.2VDD	0	$0.2V_{DD}$	V
I_{OL}	LOW-level output current	V_{OL} = 0.4 V	3	-	3	-	20	-	mA
		V_{OL} = 0.6 V	-	-	6	-	-	-	mA
t_{of}	Output fall time from		-	250	20 x	250	20 x	120	ns

	V_{Ihmin} to V_{ILmax}				(VDD / 5.5 V)		(VDD / 5.5 V)		
t_{SP}	Pulse width of spikes that must be suppressed by the input filter		-	-	0	50	0	50	ns
I_i	Input current each I/O pin	$0.1V_{DD} < V_I < 0.9V_{Ddmax}$	10	+10	10	+10	10	+10	μA
C_i	Capacitance for each I/O pin		-	10	-	10	-	10	pF

[2] $V_{IHmax} = V_{DD(max)} + 0.5V$, the pin's ultimate withstand voltage is 6.5V.

The following table defines the maximum allowable rise time and fall time of the bus signal.

Symbol	Parameter	Conditions	Standard-mode		Fast-mode		Fast-mode Plus		Unit
			Min	Max	Min	Max	Min	Max	
f_{SCL}	SCL clock frequency		0	100	0	400	0	1000	kHz
$t_{HD;STA}$	hold time (repeated) START condition	After this period, the first clock pulse is generated.	4.0	-	0.6	-	0.26	-	μs
t_{LOW}	LOW period of the SCL clock		4.7	-	1.3	-	0.5	-	μs
t_{HIGH}	HIGH period of the SCL clock		4.0	-	0.6	-	0.26	-	μs
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	0.6	-	0.26	-	μs
$t_{HD;DAT}$	data hold time[2]	CBUS compatible masters	5.0	-	-	-	-	-	μs
		I^2C-bus devices	0	-	0	-	0	-	μs
$t_{SU;DAT}$	data set-up time		250	-	100	-	50	-	ns
t_r	rise time of both SDA and SCL signals		-	1000	20	300	-	120	ns
t_f	fall time of both SDA and SCL signals[3][6][7][8]		-	300	20 x (VDD / 5.5 V)	300	20 x (VDD / 5.5 V)	120	ns
$t_{SU;STO}$	set-up time for STOP condition		4.0	-	0.6	-	0.26	-	μs

t_{BUF}	bus free time between a STOP and START condition		4.7	-	1.3	-	0.5	-	μs
C_b	capacitive load for each bus line[10]		-	400	-	400	-	550	pF
$t_{VD;DAT}$	data valid time[11]		-	3.4	-	0.9	-	0.45	μs
$t_{VD;ACK}$	data valid acknowledge time[12]		-	3.45	-	0.9	-	0.45	μs
V_{nL}	noise margin at the LOW level	for each connected device (including hysteresis)	0.1VDD	-	0.1VDD	-	0.1VDD	-	V
V_{nH}	noise margin at the HIGH level	for each connected device (including hysteresis)	0.2VDD	-	0.2VDD	-	0.2VDD	-	V

According to the above protocol specifications, we can calculate the reasonable range of the external pull-up resistor.

Assuming that the bus signal rises from V_{IL}=0.3V_{DD} to V_{IH}=0.7VDD, the charging time can be calculated as:

V(t1) = 0.3 V_{DD} = V_{DD} (1 e $^{t1 / RC}$); t1 = 0.3566749 RC

V(t2) = 0.7 V_{DD} = V_{DD} (1 e $^{t2 / RC}$); t2 = 1.2039729 RC

T = t2- t1 = 0.8473 RC

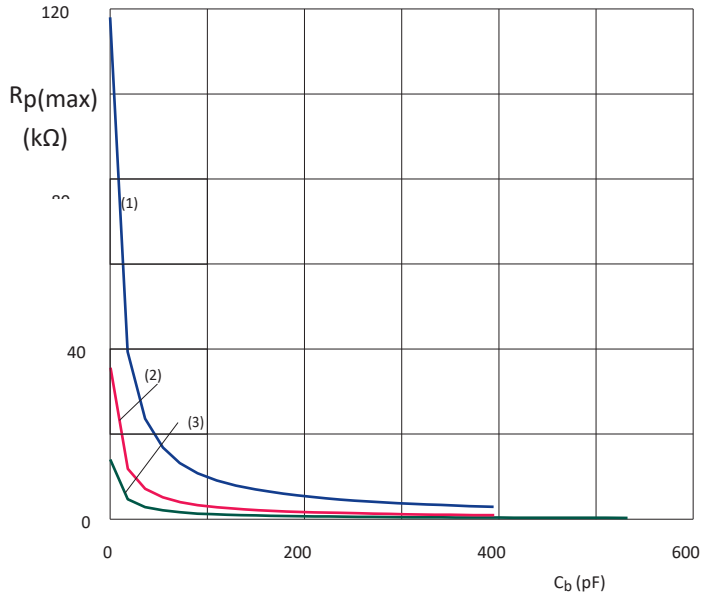
According to the bus capacitive load size and the protocol's requirements for the maximum signal rise time, we can calculate the maximum value of the pull-up resistor:

$$R_{p(\max)} = \frac{t_r}{0.8473 \times C_b}$$

The minimum value of the pull-up resistor is determined by the bus power supply voltage VDD and the IO current sink capability. The I2C pin sink capability of FM33LG0 is 20mA, and the minimum sink capability required by the protocol is 3mA in standard/fast mode and 20mA in Fm+ mode.

$$R_{p(\min)} = \frac{VDD - V_{OL(\max)}}{I_{OL}}$$

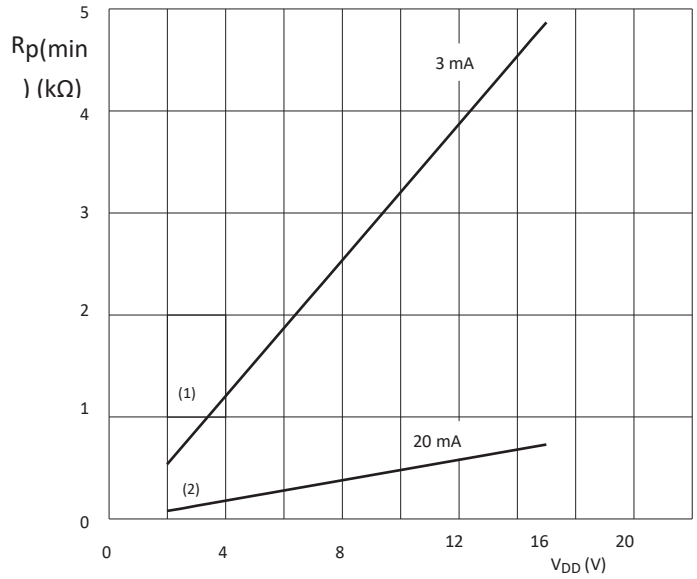
According to the above calculation, the recommended resistance range of the pull-up resistor can be obtained, see the figure below.



(1) Standard-mode

(2) Fast-mode

(3) Fast-modePlus



(1) Fast-mode andStandard-mode

(2) Fast-modePlus

21.4 Clock and Reset

Both the I2C master and slave adopt a dual clock structure:

- The bus register clock of the master and slave is represented by PCLK, which is derived from APBCLK. When the CPU or DMA needs to access I2C internal registers, PCLK must be enabled. See 14.11.15 Peripheral Bus Clock Control Register 3.
- The host's data receiving and sending clock is represented by I2CCLK, which can be derived from APBCLK, RCHF, SYSCLK, RCLF, and can work independently of APBCLK. I2CCLK must be enabled to send and receive data.
- The data transceiver clock of the slave uses the SCL bus clock input, so data can be sent and received without the system clock

The control of PCLK and I2CCLK is completed in the CMU module, and the corresponding CMU control register must be correctly configured before I2C communication.

The dual clock structure can make I2C work not limited to the configuration of APBCLK. When some peripherals need to work at a very high APBCLK frequency, I2C can still work at a reduced frequency; or vice versa, the CPU works At a lower frequency, it does not affect I2C for data communication at a higher baud rate.

In theory, there is no constraint on the relative relationship between PCLK and baud rate clock, and the baud rate clock can be faster or slower than PCLK. But the application needs to pay attention to whether the CPU or DMA has time to carry out data transfer when the frequency difference between the two is large.

The I2C reset register (I2CRST) in the RMU module must be cleared before the module works.

21.5 Communication Flow

21.5.1 Communication Timing Diagram

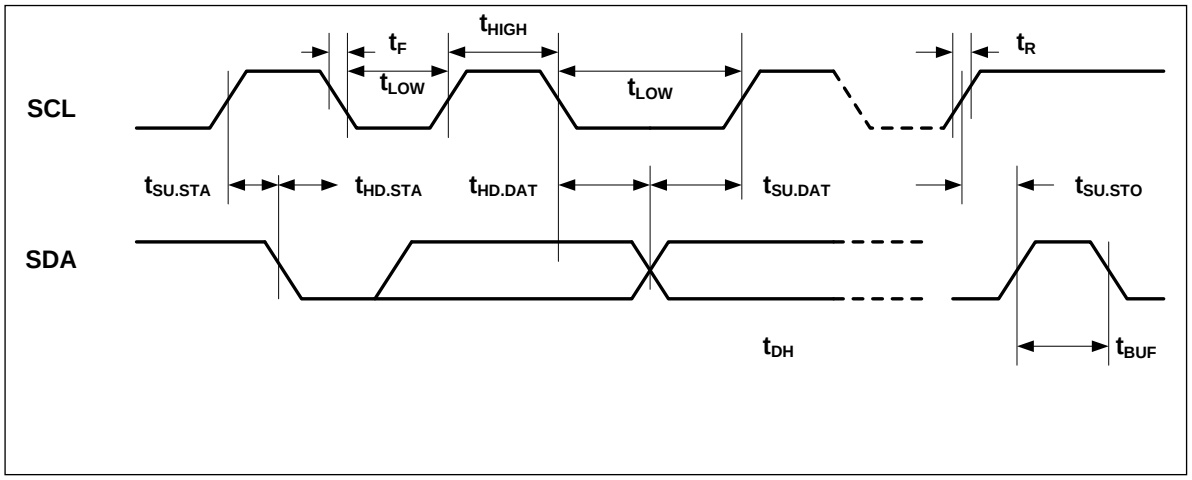


Figure 21-2 I2C Bus Protocol

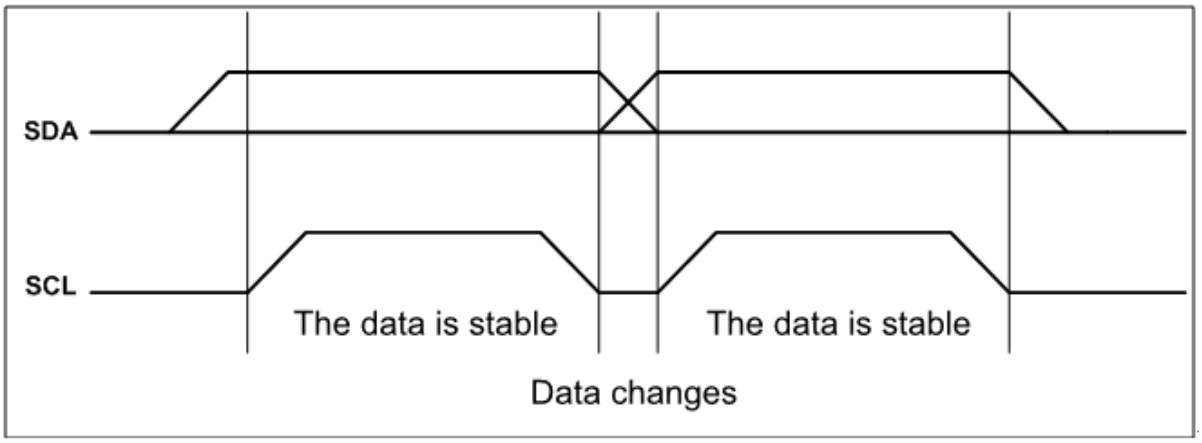


Figure 21-3 Bit Protocol

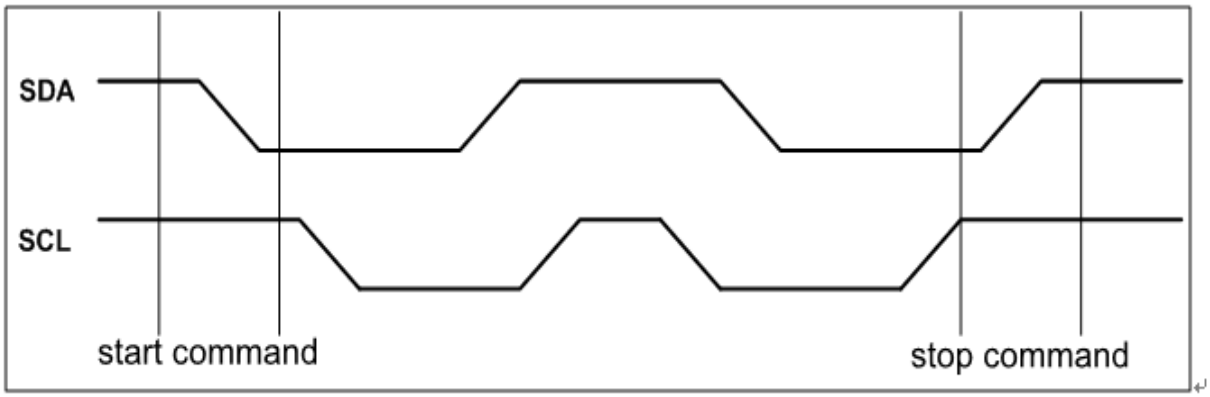


Figure 21-4 Start&Stop Condition Definition

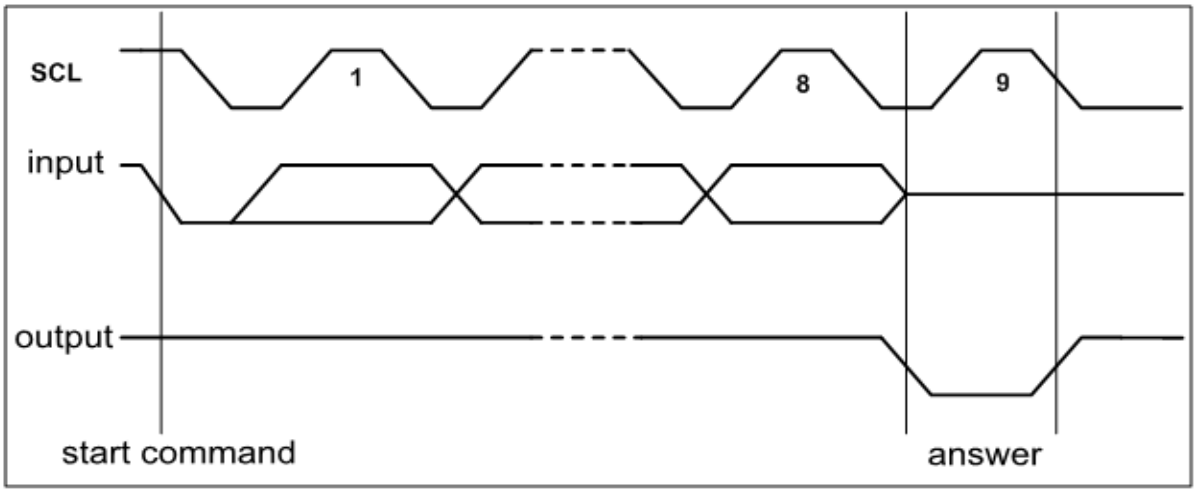


Figure 21-5 ACK

21.5.2 Description

Clock effective timing: SDA pin is usually pulled high by peripheral devices. The data of the SDA pin should change when SCL is low (see Figure 21-3); when the data changes when SCL is high, it will be regarded as a start or stop command as described below.

Start command: When SCL is high, the change of SDA from high to low is regarded as the start command, and the start command must be used as the start of any read/write operation command (see Figure 21-4).

Stop command: When SCL is high, the change of SDA from low to high is regarded as a stop command. After a read operation, the stop command will make the EEPROM enter the wait state low power consumption mode (see Figure 21-4).

Output response: The data on the SDA is serially input and output in a set of 8 bits. The MSB is sent first. After receiving each byte, the receiver should send back an acknowledgement bit in the 9th cycle (below Referred to as ack), the ack clock is provided by the host. The sender suspends SDA during the ack period, and the receiver must pull SDA low to ensure that SDA is low during the high level of the ack clock to form a valid ack signal (see Figure 21-5)

Parameter	Symbol	Standard (100K)		Fast (400K)		Units
		Min	Max	Min	Max	
SCL clock frequency	F_{SCL}	0	100	0	400	kHz
Start condition establishment time	$T_{SU:STA}$	4.7	—	0.6	—	us
Start condition stretching time	$T_{HD:STA}$	4.0	—	0.6	—	us
Clock stretching low time	T_{LOW}	4.7	—	1.3	—	us
Clock stretching high	T_{HIGH}	4.0	—	0.6	—	us

Parameter	Symbol	Standard (100K)		Fast (400K)		Units
		Min	Max	Min	Max	
time						
Data input setup time	$T_{SU:DAT}$	250	—	100(4)	—	ns
Data input stretching time	$T_{HD:DAT}$	5.0 0(2)	— 3.45(3)	— 0(2)	— 0.9(3)	us us
SDA and SCL pull-up times	T_R	—	1000	20+0.1Cb(5)	300	ns
SDA and SCL pull-down time	T_F	—	300	20+0.1Cb(5)	300	ns
Stop condition establishment time	$T_{SU:STO}$	4.0	—	0.6	—	us
Bus idle time	T_{BUF}	4.7	—	1.3	—	us
Capacitive load on the bus	Cb	—	400	—	400	Pf
Min Noise tolerance	V_{nL}	$0.1V_{DD}$	—	$0.1V_{DD}$	—	V
Max Noise tolerance	V_{nH}	$0.2V_{DD}$	—	$0.2V_{DD}$	—	V

Table 21-2 I2C Interface Timing Requirements

21.6 I2C Working Mode

The I2C module supports the following working modes:

- Host receiving
- Host send
- Receive from the machine
- Send from machine

After the chip is powered on, the I2C module is closed by default, and the master and slave do not work. The software needs to select the working mode of the module according to the application, and set MSPEN to enable master communication, or set SSPEN to enable slave communication.

The master and slave cannot work at the same time because they reuse the same IO pins as SCL and SDA. In principle, it is forbidden for software to set MSPEN and SSPEN to 1 at the same time.

21.7 I2C Slave Address Format

The I2C bus protocol defines the following reserved addresses. For most of these reserved addresses, the I2C slave hardware does not make legal judgments, and the software can perform custom processing based on the received addresses.

But for the 10bit slave address application, that is, when SSPCR.A10EN=1, the 1st byte must start with 11110, otherwise the ADDR_ERROR error flag will be triggered. In the case of SSPCR.A10EN=0, if the slave receives the address byte starting with 11110, it will also set the ADDR_ERROR error flag.

Slave address	R/W_ bit	Description
0000 000	0	General Call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purpose
0000 1XX	X	HS-mode master code
1111 1XX	X	Reserved for future purpose
1111 0XX	X	10bit slave addressing

Table 21-3 I2C Slave Reserved Address Definition

21.8 I2C Initialization

The I2C module must be initialized correctly before I2C communication. It is recommended that the software perform the initialization operation according to the following steps:

- Clear the I2CRST register of the RMU module to ensure that the I2C module is not in the reset state
- Set the I2C_PCE register of the CMU module to enable the I2C module register bus interface clock
- Configure the I2C_CKS and I2C_CKE registers of the CMU module, select and enable the I2C working clock (if it is in the slave mode, this step is not required)
- Configure SCL analog filter enable in slave mode as required (input analog filter, >50ns)

21.8.1 IO Config

FM33LG0 has at most two sets of pins for data transmission. Before starting I2C communication, you need to set the FCR register of the corresponding pin to Digital function:

SDA: PA12/PD12

SCL: PA11/PB15

Note that if PA11 and PB15 are configured for SCL function at the same time, PB15 is connected to the I2C module, and PA11 is invalid; if PA12 and PD12 are configured for SDA function at the same time, both pins in master mode will output SDA signals, and in slave mode Only PD12 is connected to the SCL input of the I2C slave.

21.8.2 Master Baud Rate Configuration

The I2C master needs to configure the communication baud rate before enabling, and the slave communication rate is determined by the SCL sent by the master, so no configuration is required.

The MSPBRGH and MSPBRGL baud rate configuration registers are used to generate the communication baud rate. MSPBRGH and MSPBRGL are 9 bit baud rate division coefficients, and the baud rate calculation formula is as follows:

$$\text{SCL cycle } T_{SCL} = T_{BRGH} + T_{BRGL}$$

Among them, MSPBRGH defines the high-level width of SCL, and MSPBRGL defines the low-level width of SCL

$T_{BRGH}=T_{I2CCLK}\times(MSPBRGH+1)$

$T_{BRGL}=T_{I2CCLK}\times(MSPBRGL+1)$

T_{I2CCLK} is the I2C working clock cycle

For example, for a 100k baud rate, T_{SCL}=10us; if the I2C working clock is 8M, then T_{I2CCLK}=125ns. Assuming that the SCL duty cycle is required to be 50%, that is, the SCL high and low levels are both 5us width. According to the above formula, MSPBRGH=MSPBRGL=39 can be calculated

21.8.3 Input Analog Filtering and Output Delay of Slave

The analog filter function is only for the SCL pin, and only the SCLi input signal of the slave can enable the analog filter function.

At the same time, the SDA output delay of the slave machine ensures the output hold time of SDA relative to the falling edge of SCL by adding an analog delay greater than 300ns on SDAo.

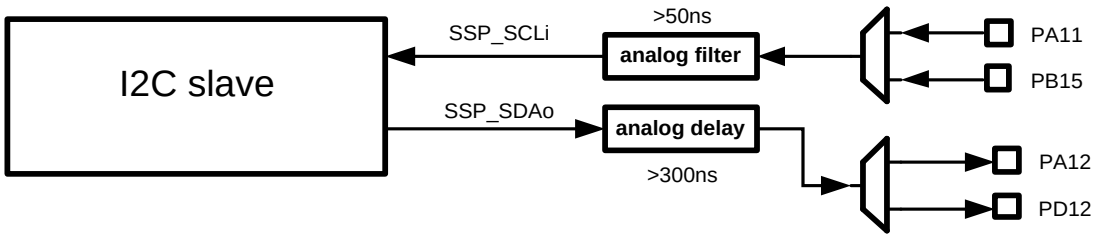


Figure 21-6 Slave Signal Filtering

21.9 I2C Master Function

The I2C master mode of FM33LG0 does not support multi-master bus, so other devices hanging on the bus are slaves. The master always provides the synchronous clock SCL on the bus, and the direction of the SDA data flow can be the master sending and receiving from the machine, or the slave sending and receiving by the host.

I2C bus communication is always initiated by the host, and the host mode supports 7bit or 10bit addressing.

21.9.1 7bit Addressing

In 7bit addressing, the first byte sent by the master contains the slave address and the transfer direction bit ($R/\overline{W}$), depending on $R/\overline{W}$ the subsequent transfer is a master writing data to the slave ($R/\overline{W}$ =0) or a master reading data from the slave ($R/\overline{W}$ =1).

	Slave Address Byte							
bit	7	6	5	4	3	2	1	0
name	address							R/W

Bit description:

bit	name	function
7-1	address	Slave device address
0	R/W	0: Write means sending data (master sends) 1: Read means request data (slave sends back)

Master writes data to slave

A typical frame structure for 7bit addressing, with the master writing data to the slave, is shown in the diagram below.

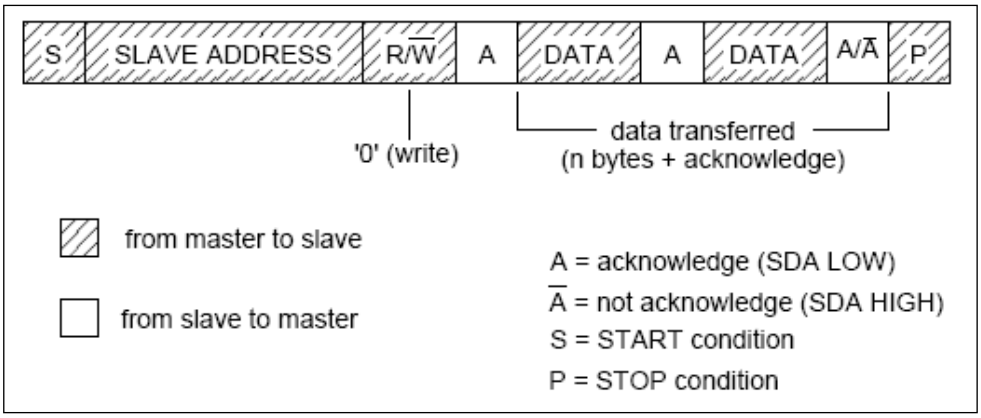


Figure 21-7 Frame Format When a Master Writes Data to a 7-bit Address Slave

1, The master initiates START condition

2, Master sends slave address, slave address contains 7 bits of slave address and 1 bit of R/W flag bit which is 0 when sending data

3, The master sends the first 8-bit data frame

4, the master will determine if a valid ACK is detected at the 9th SCL after each 8-bit data is sent, if the master detects a positive ACK, it will continue to send next byte

5, If the slave cannot reply ACK, the master should send a STOP condition to terminate the transmission after detecting the NACK

6, After the master has finished sending all data, it will send the STOP condition

The software initiates the operation flow of the I2C master send as follows:

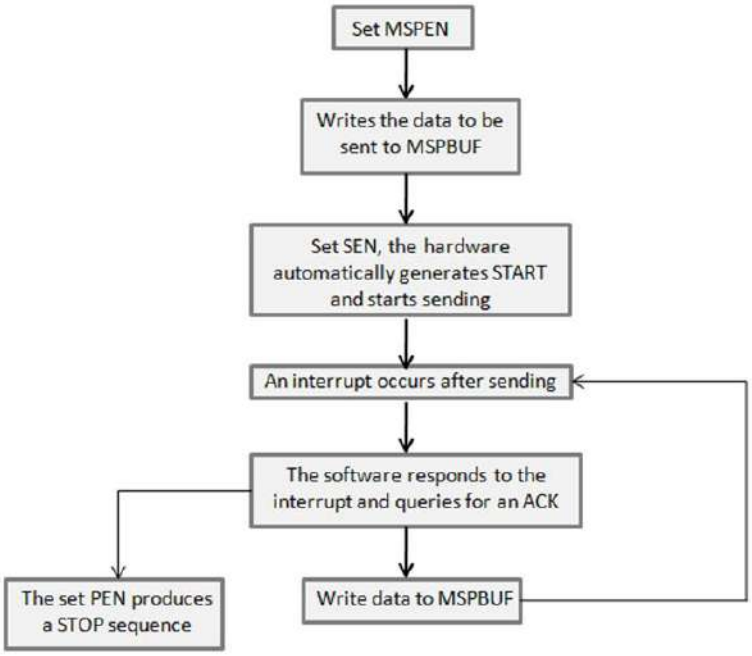


Figure 21-8 I2C Software Sending Data Flow Diagram

The waveform diagram of the I2C master writing data to the 7-bit address slave is as follows:

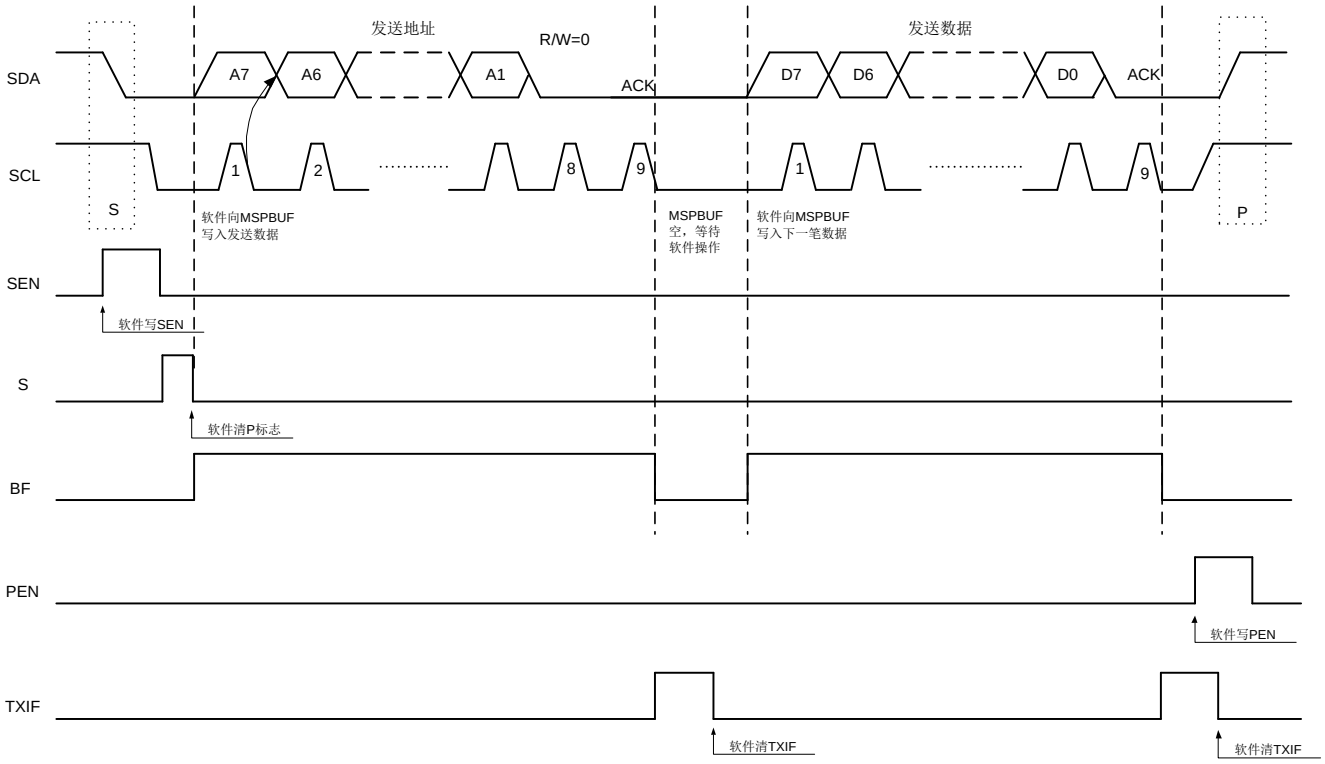


Figure 21-9 I2C Master Sends Data Flow Diagram to 7-bit Address Slave

Master reading data from a slave

A typical frame format for 7bit addressing, where the master reads data from the slave, is shown in the diagram below.

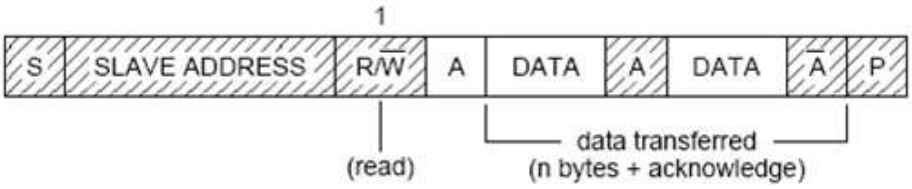


Figure 21-10 Frame Format When a Master Reads Data from a 7-bit Addressing Slave

1, Master initiates START condition

2, The master sends the slave address, the slave address contains 7 bits of the slave address and 1 bit of the R/W flag bit which is 1 when the data is read

3, Set MSPCON.RCEN to 1, the master automatically turn to receive state

4, The master starts to receive the first byte of 8-bit data, and sends a valid ACK to the slave at the 9th SCL, so as to continue to read the next data byte

5, After reading the last byte, the master sends a NACK to the slave at the 9th SCL

6, The master sends STOP bit to terminate the reading

The operational flow of I2C reception is shown in the following diagram:

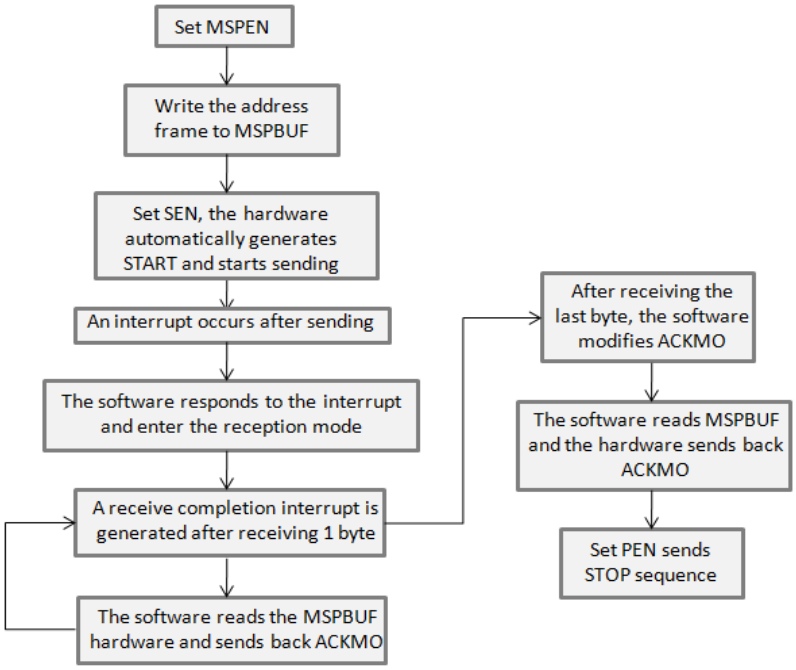


Figure 21-11 I2C Software Receive Data Flow Diagram

After the host receives the data sent by the slave each time, it sends back a response according to the ACKMO register. The reset value of ACKMO is 0, that is, the host sends back an ACK by default. If the software wants the host to send back a NACK after the reception is complete, it needs to rewrite the ACKMO register to 1 in the previous byte reception completion interrupt. When ACKMO is 1, the host will automatically clear ACKMO after sending the response.

The waveform diagram of the I2C master reading data from the 7-bit address slave is as follows:

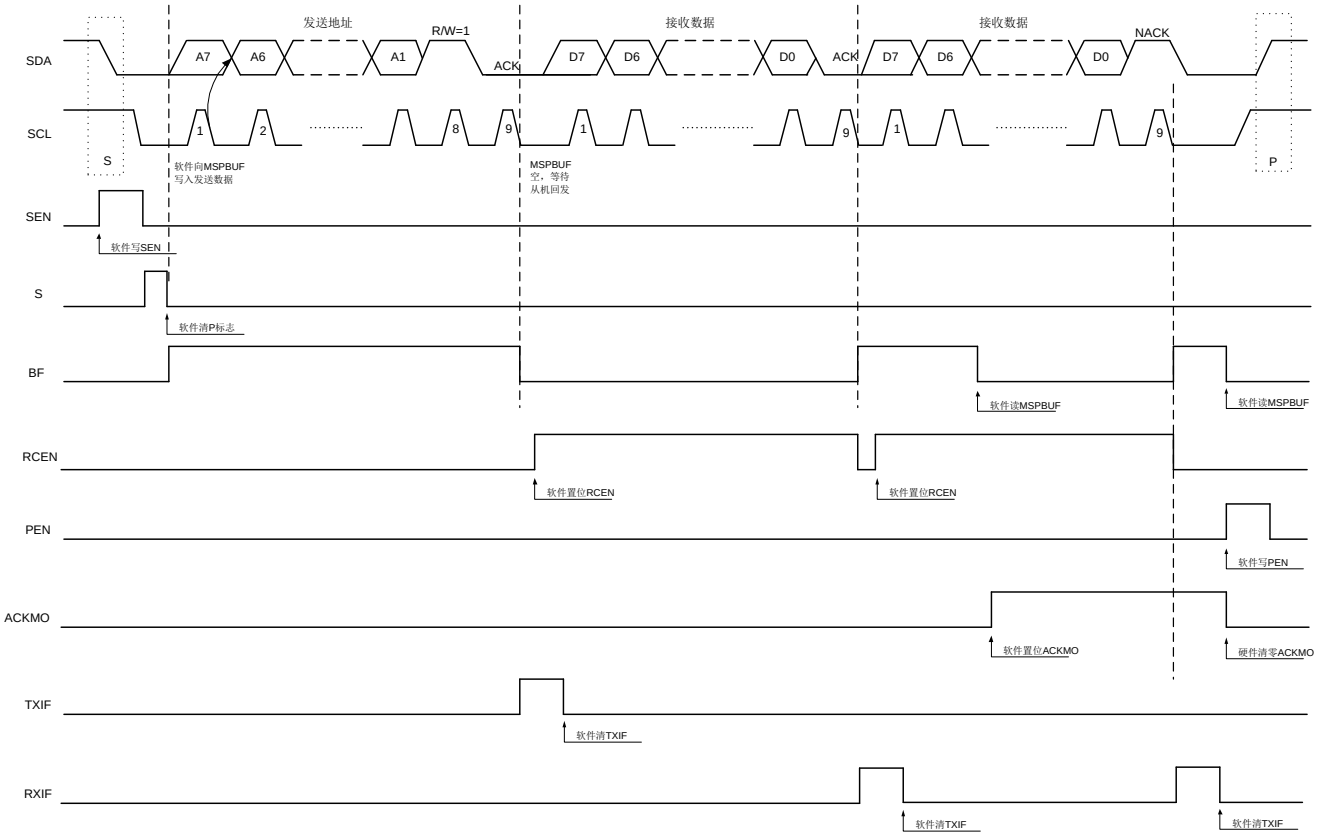


Figure 21-12 I2C Reads the Data Flow Diagram from the 7-bit Address Slave

Bidirectional Data Transfer (Combined Mode)

A typical bidirectional data read and write flow diagram is shown in the figure below. In the process of sending or reading data by the host, the host can restart a new sending or reading communication by sending the Repeated Start sequence. Therefore, the host can send or read data in a communication.

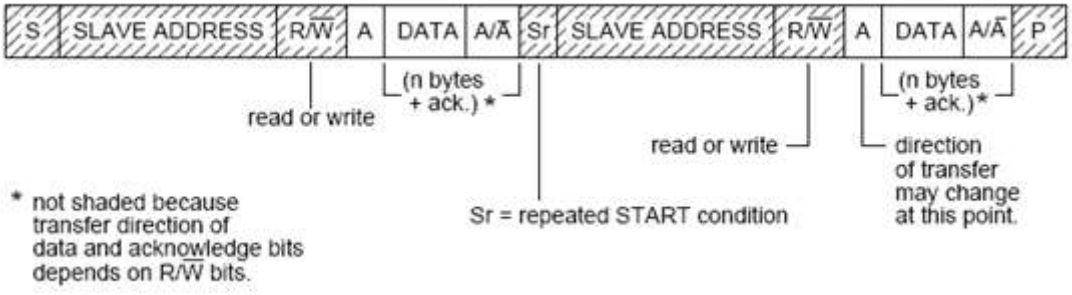


Figure 21-13 Frame Format for Bi-Directional Data Communication

The software operation procedure for bidirectional communication is similar to that for unidirectional communication, except that the direction of transmission is modified by sending the ReSTART condition and slave address bytes.

21.9.2 10bit Addressing

In 10bit addressing, the first byte sent by the master contains part of the slave address (11110_A9_A8) and the transfer direction bit ($R/\overline{W}$), the second byte contains the remaining slave address (A7 to A0). After the two byte addresses have been sent, the data is then transferred.

Master writes data to the slave

A typical data flow for 10bit addressing, where the master writes data to the slave, is shown in the diagram below.

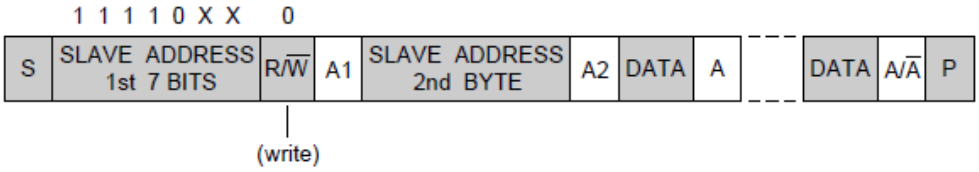


Figure 21-14 10bit Addressing, Master Writes Data to Slave

1. The master initiates START condition
2. The master sends the first slave address byte, starting with 11110, followed by the highest bit of the 2bit slave address, and the R/W flag bit, the R/W bit is 0 when sending data
3. The master checks the ACK sent back by the slave
4. The master sends the second slave address byte, containing the lower 8 bits of the slave address
5. The master checks the ACK replied by the slave
6. The master continues to write data to the slave
7. After the master has finished sending all data, it sends the STOP condition

The software procedure of the I2C master transmitting is as follows:

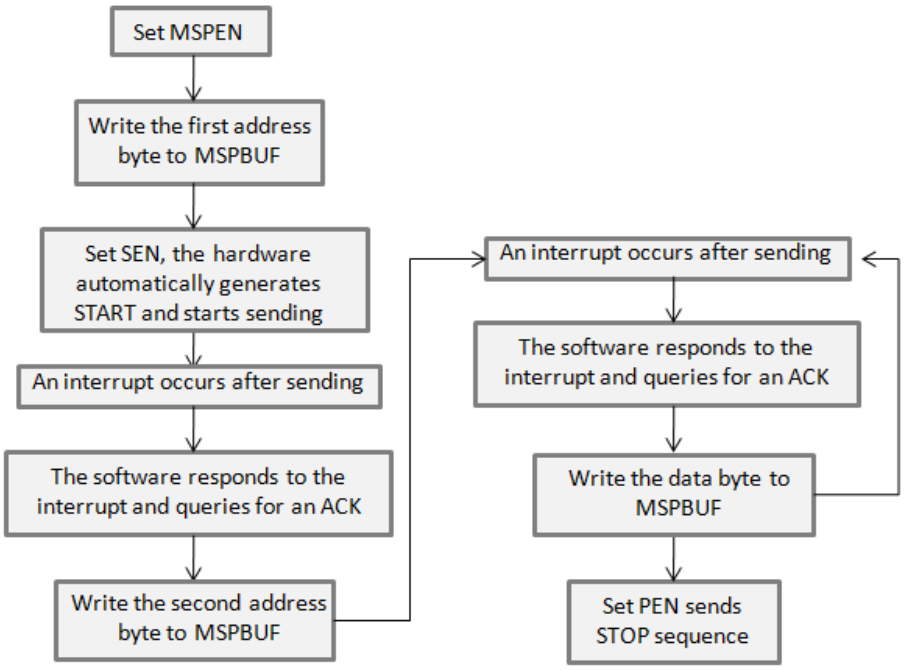


Figure 21-15 I2C Software Transmit Flow Diagram

Master reading data from a slave

A typical data flow for 10bit addressing, where the master reads data from the slave, is shown in the diagram below.

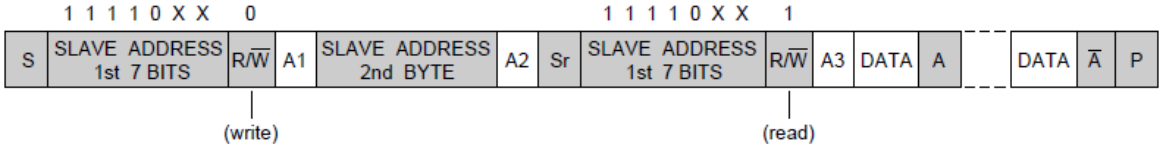


Figure 21-16 10bit Addressing, Master Reads Data from Slave

1, Master initiates START condition

2, The master sends the first byte of the slave address, including 5 bits of the leading code 11110, 2 bits of the highest bit of the slave address and 1 bit of the R/W flag bit, the R/W bit is 1 when the data is read

3, The master sends the second byte of the slave address, including the low 8 bits of the address

4, The master sends ReSTART condition

5, The master sends the first byte of the slave address again, changing R/W to 0

6, Set MSPCON.RCEN to 1, the master goes to receive state

7, The master starts to receive the first byte of 8-bit data, and sends a valid ACK to the slave at the

9th SCL, thus continuing to read the next data byte

8, After reading the last byte, the master sends a NACK to the slave at the 9th SCL

9, The master sends STOP condition

The software procedure for I2C receive flow as follows.

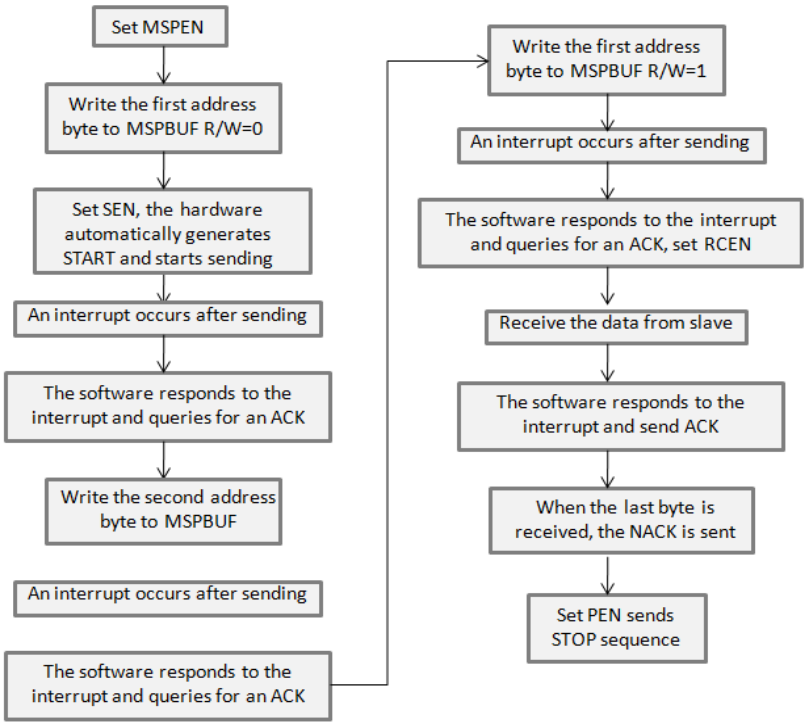


Figure 21-17 I2C Software Send Data Flow Diagram

Bidirectional data transfer (combined mode)

A typical bi-directional data read/write flow is shown in the figure below. During a master write or read, the master can restart a new transaction by sending Repeated Start condition, so the master can have bidirectional communication in one transaction.

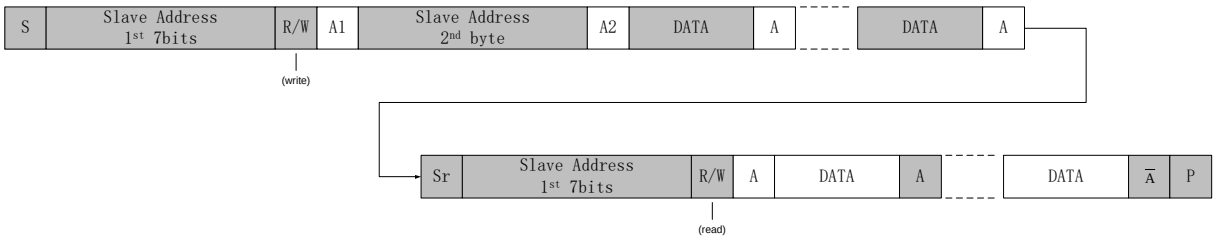


Figure 21-18 I2C Software Sending Data Flow Diagram

The software procedure for combined transfers is similar to that for unidirectional transfers, except that

the transfer direction is modified by sending the ReSTART condition and 1st slave address bytes.

21.9.3 DMA

The I2C master supports DMA. It should be noted that the bus clock (PCLK) of the I2C module must be enabled in order to use the DMA function.

Master using DMA to write data to a slave

When the master uses DMA to send data, all data including the slave address byte and the send data needs to be written to RAM in advance and transmitted via a DMA request. The software should configure the target DMA channel as I2C_TX.

In case DMAEN=1, MSPEN is set and if the data buffer register MSPBUF is empty, the I2C module generates a DMA request and the DMA module responds by writing the data to MSPBUF and the I2C module automatically sets SEN to generate START condition to start data transmission (first byte is the slave address). The I2C does not check the validity of the data, the software must ensure that the data in RAM is correct.

After each byte is sent, the I2C checks the slave ACK and generates a new DMA request if the ACK is correct, or a NACK interrupt if a NACK is received and no further DMA requests are generated.

When the DMA completes sending the specified length of data, the DMA transfer completion interrupt is generated. Then software can set PEN to generate the STOP condition, or the I2C hardware can automatically set PEN to generate the STOP condition according to the DMA transfer completion signal. The desired strategy can be selected by configuring the AUTOEND register.

The flow of the master using DMA for transmitting is shown below:

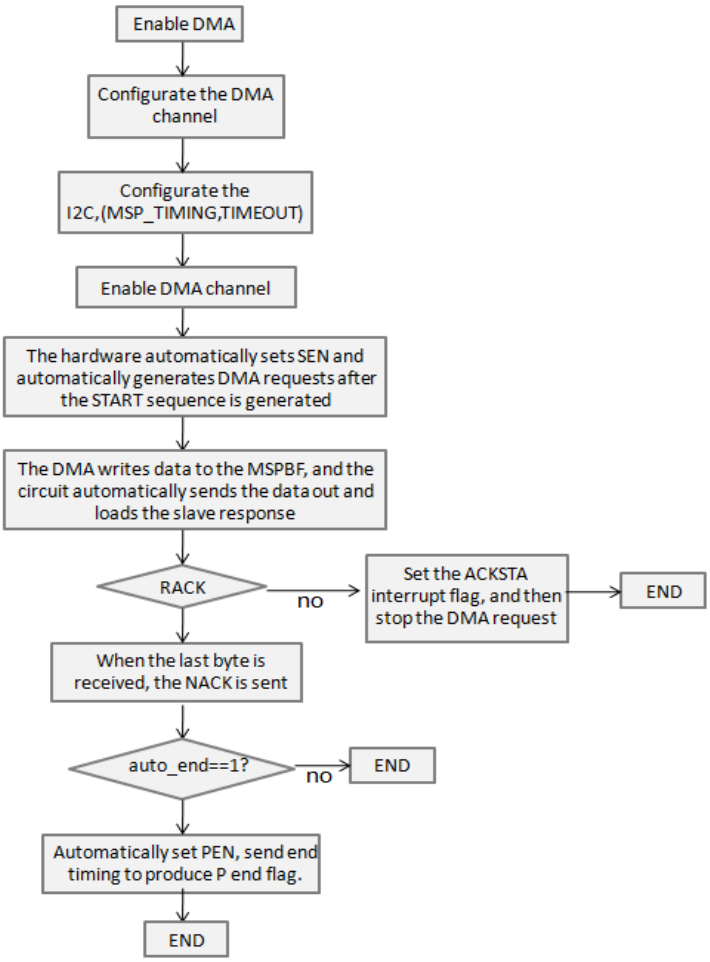


Figure 21-19 I2C Master DMA Transmit Flowchart

The host uses DMA to read data from the slave

In this scenario, the slave address byte must be sent by software. The software should configure the target DMA channel as I2C_RX in advance.

After the software first sends the slave address, set MSP_DMAEN=1, then enable the corresponding DMA channel, I2C automatically enters the receiving mode, and generates a DMA request after each byte is received, and informs the DMA to read the content of MSPBUF. Send ACK back to the slave.

When the DMA transfer reaches the specified length, the DMA transfer completion flag will notify I2C to send back a NACK. Then according to the AUTOEND register configuration, PEN can be set by software or hardware to generate a STOP sequence.

Note: *When the I2C host receives data through DMA, under different AUTOEND configurations and the same DMA transfer length (CHxTSIZE) configuration, the number of bytes received by DMA will be different. When AUTOEND=0, the received byte count is CHxTSIZE+1; when AUTOEND=1, the received byte count is CHxTSIZE.*

The process of receiving by the host using DMA is as follows:

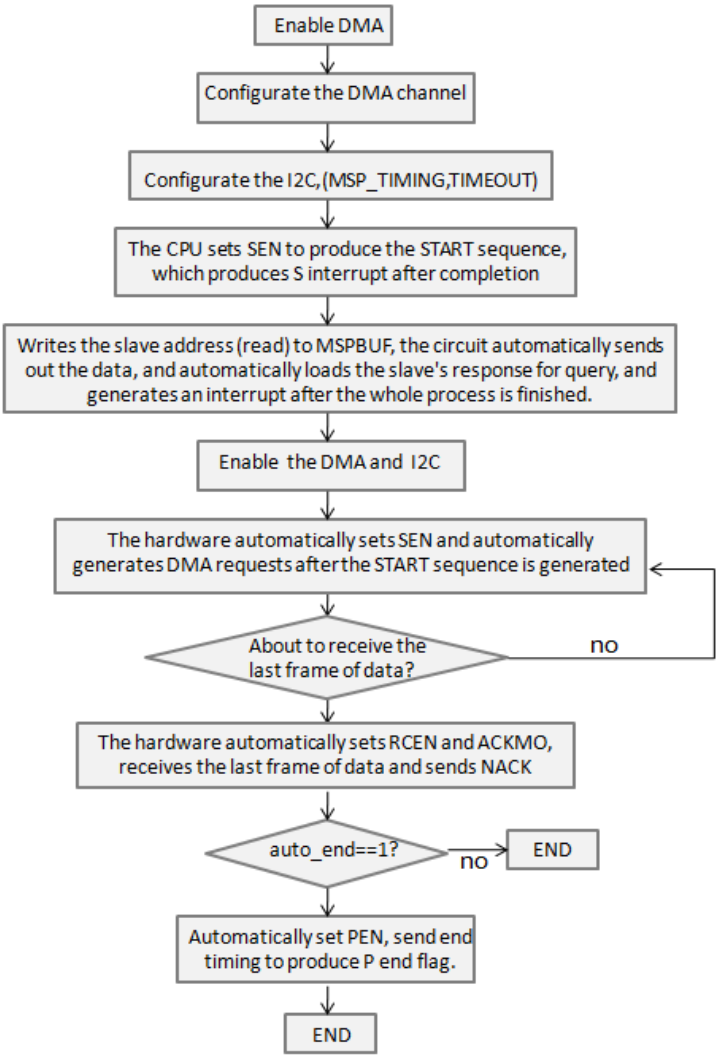


Figure 21-20 I2C Master DMA Receive Flowchart

21.9.4 Slave Clock Stretching

The I2C protocol allows low-speed slaves to suspend data communication by pulling SCL low. I2C masters improve compatibility with low-speed peripherals by supporting this feature. At the start of each byte transfer, the master checks the actual level of SCL on the bus after attempting to drive SCL high. If it is not high, it means that the slave is performing an SCL stretch and the master will continue to monitor the SCL level until SCL is high before starting subsequent operations.

Note: *The master only performs an SCL stretch check at the first SCL rising edge of each byte transfer.*

21.9.5 Timeout Error

The I2C master also implements a timeout function that generates an alarm interrupt and returns to IDLE status if SCL is stretched by slave.

When the master detects an SCL stretching, its internal timer starts counting. The maximum length of the SCL stretching timeout is 4096 SCL cycles. Assuming a baud rate of 100K, the timeout period is approximately 40ms.And if the baud rate is 400K, the timeout period is approximately 10ms.

The timeout period can be set by the software via the 12bit TIMEOUT register. The software must set the TIMEOUT register with MSPEN is 0. This reset value is 0xFFF, which means the maximum 4096*TSCL timeout period. When the SCL stretching is detected, the TIMEOUT register starts to decrement. And when the counter reaches 0, the counter stops and the TIMEOUT register is reset to 0xFFF, and a timeout interrupt is triggered at the same time. Therefore, the timeout period can be set by modifying the initial value of TIMEOUT.

$$T_{SCL\_STRETCHING\_TIMEOUT}=TIMEOUT[11:0] * T_{SCL}$$

When a TIMEOUT interrupt occurs, it is recommended that the I2C module is reset by software.

This timeout function can be disabled. The software can also implement its own timeout function of any length by using the timer in combination with the SCL pin state polling.

21.9.6 Programmable Timing and Baud Rate Generation

The host mode of the I2C module provides flexible timing programming features, allowing users to define the low-level width, high-level width of the SCL clock, and the setup and hold time of SDA data.

The low-level and high-level width of SCL can be set through the MSPBRG register, and the retention of SDA data relative to the SCL clock pulse and the length of the setup retention time can be configured through the SDAHD register.

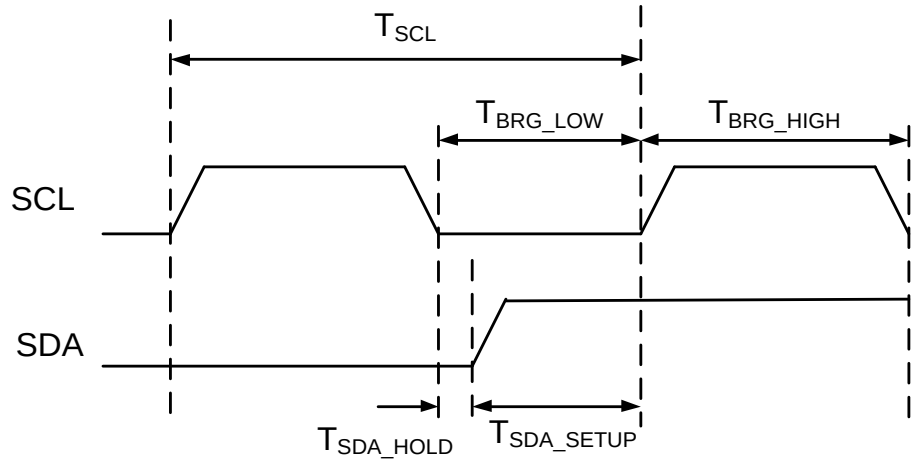


Figure 21-21 Master Timing Control

In the above figure, T_{SCL} is the communication baud rate and parameters can be defined by the following equation.

$$T_{SCL} = T_{BRG\_LOW} + T_{BRG\_HIGH}$$

$$T_{SDA\_SETUP} = T_{BRG\_LOW} - T_{SDA\_HOLD}$$

Note that the configuration of the MSPBGRH, MSPBRGL and SDAHD registers must meet the following requirements, as violation of these requirements will result in abnormal bus timings.

MSPBRGH >= 2

MSPBRGL >= 2

MSPBRGL-1 >= SDAHD >= 1

TIMEOUT >= 1

21.10 I2C Slave Function

1 The I2C slave does not need a system clock to work, so it can send and receive data and wake up when the chip is sleeping.

After the slave receives 1 byte of data, it generates an interrupt to notify the CPU to process the data. Before the CPU takes the data, the hardware can pull down SCL (software control is enabled) to notify the sender that it is busy, and the sender should suspend sending until SCL is released. . If the receiver cannot respond to the ACK, the sender should send P to terminate the communication or send Sr to start a new communication after detecting the ACK failure.

After the slave sends 1 byte of data, an interrupt is generated to notify the CPU, the hardware pulls down SCL to make the master wait, the CPU responds to the interrupt and is ready for the next byte of data, and then releases the SCL, the master continues to send SCL so that the slave continues to send data

21.10.1 Slave Addressing

According to the SR.ASPC10EN register status, the slave can support 7bit or 10bit addressing process. The slave address is defined by the SSPADR register.

For the 10bit slave address application, that is, when SSPCR.A10EN=1, the 1st byte must start with 11110, otherwise the ADDR_ERROR error flag will be triggered. In the case of SSPCR.A10EN=0, if the slave receives the address byte starting with 11110, it will also set the ADDR_ERROR error flag.

21.10.2 Slave Sends Data

Recommended operation process:

- The slave receives the address byte (R/W=1), sends back an ACK, and generates an address match interrupt
- Since R/W=1, the hardware automatically performs SCL extension, and the slave enters the sending state
- The software responds to the interrupt, queries the R/W flag, and confirms that it is sent by the slave
- The software writes the data to be sent into SSPBUF
- Automatically release SCL by hardware
- The new SCL is coming, SSPBUF is shifted and output to the SDA bus
- Receive ACK and generate transmission completion interrupt
- Repeat the data transmission process until the STOP sequence is received, or the host NACK is received

The following figure is a schematic diagram of a typical slave data transmission waveform:

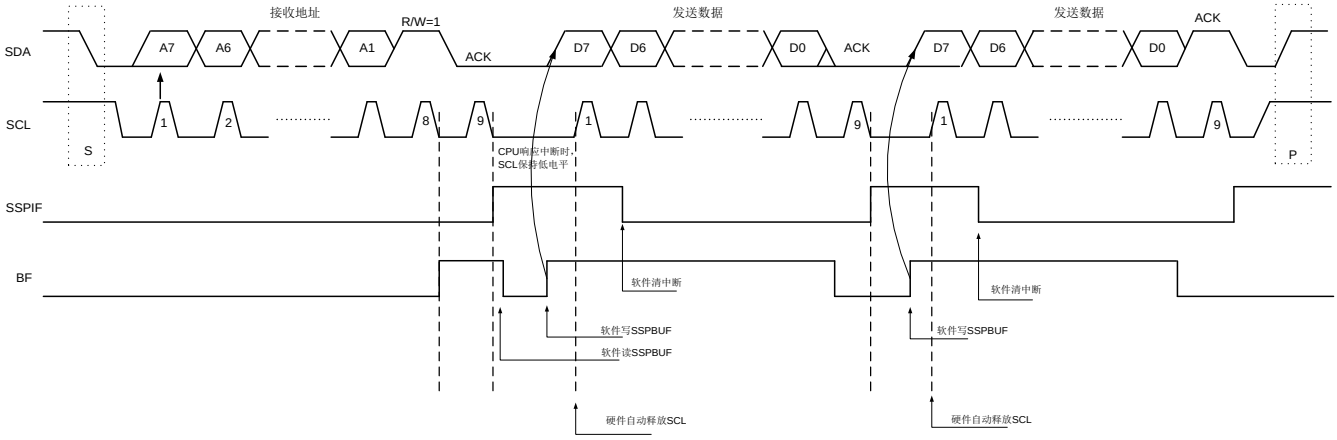


Figure 21-22 Slave Data Sending Waveform

In the slave sending process, when the slave receives the correct address, the ADM flag is set, and the address byte will not be written into SSPBUF, so the BF flag will not be set. The hardware automatically pulls down the SCL signal and waits for the software to write SSPBUF. When the software writes SSPBUF, the BF flag is set, and the hardware releases SCL at the same time.

21.10.3 Slave Receives Data

Recommended operation process:

- The slave receives the address byte (R/W=0), sends back ACK, generates address match interrupt
- Since R/W=0, the hardware automatically performs SCL stretching, and the slave keeps the sending state
- The software responds to the interrupt, queries the R/W flag, and confirms that it is received by the slave
- Software reads SSPBUF, hardware automatically releases SCL and starts to receive data
- The master data byte arrives, the hardware sets the BF flag after the byte reception is completed
- The slave sends back ACK and generates a reception completion interrupt
- The hardware automatically performs SCL stretching (SCLSEN=1)
- The software responds to the interrupt, reads SSPBUF, and the hardware automatically clears the BF flag
- The hardware release SCL automatically
- Repeat the data reception process until the STOP sequence is received, or the software sets ACKEN to 0

The following figure is a typical schematic diagram of slave data receiving waveform (SCLSEN=1):

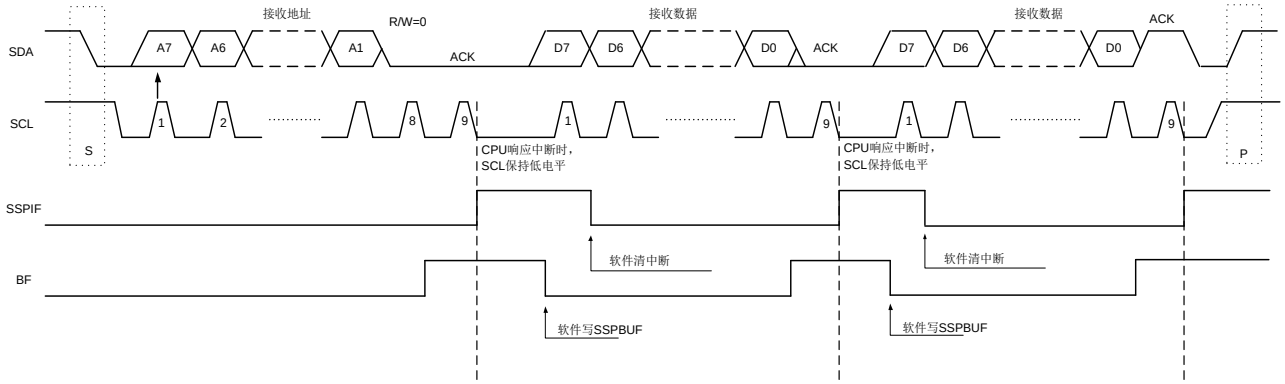


Figure 21-23 Slave Data Receiving Waveform

During the slave receiving process, the slave first receives the address byte. If the address matches, the ADM flag is set, the address byte will be written to SSPBUF and the BF flag is set, and then the hardware pulls down SCL. When the software reads SSPBUF, the BF flag is automatically cleared, the hardware releases SCL, and subsequent data reception can be carried out.

Note: *In the slave receiving process, the address byte will be written into SSPBUF and cause BF to be set. The software needs to read SSPBUF to clear BF and release SCL. In the slave sending process, the address byte will not be written into SSPBUF, so the BF flag will not be set.*

The slave can end the communication passively or actively when receiving data.

If the master actively issues STOP, the slave passively ends this communication. Or the software clears the ACKEN register in the interrupt handler, the slave will send back NACK after receiving the next byte, and the master will issue a STOP to end this communication after receiving the NACK.

Slave SCL stretching

TheI^2C slave enables SCL stretching (slave clock stretching) by default, but the software can turn off this function (SCLSEN register) to adapt to the master that does not support slave SCL stretching.

When the SCL stretching is enabled, after the data reception is completed, the software can clear the BF flag only when the receive buffer is read during the SCL stretching period. If data overflow occurs during reception, the SSPOV flag is set, and the hardware sends back NACK at this time, and SCL is no longer stretched, so that the master can issue a STOP. When SSPOV is set, it is recommended that the software wait for the STOP flag to be set, and then read the receive buffer to clear the BF flag.

Receive data overflow

When the receive buffer of the slave is full (BF=1), if new data is received, a receiving overflow occurs, and the SSPOV flag is set. The old data in the receive buffer will be overwritten by the new data. Only when the SCL stretching function is disabled, the receive data overflow may occur.

The following figure is a schematic diagram of data receiving overflow when SCLSEN=0:

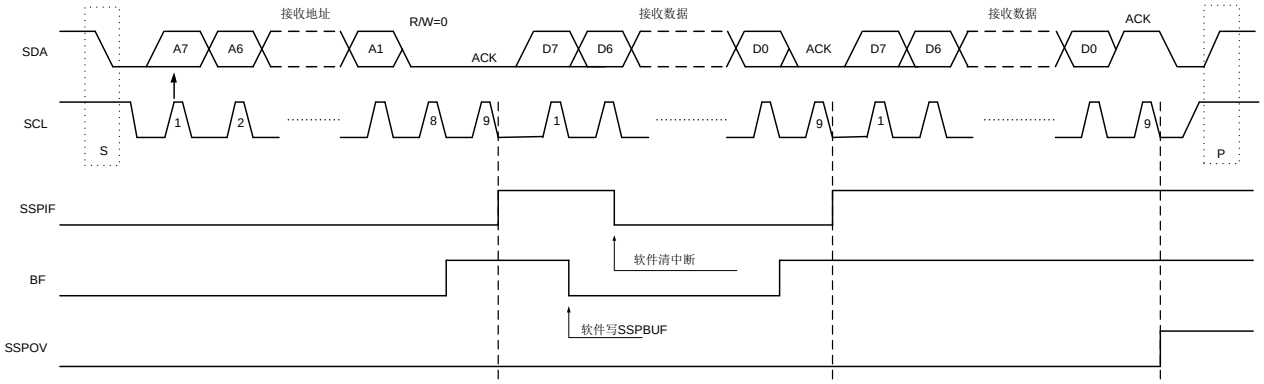


Figure 21-24 Slave Data Receiving Waveform (SCLSEN=0, Receiving Overflow)

21.10.4 Slave Low-power Receiving Wake-up

Since the I2C slave does not need a system clock to work, it can receive data and wake up the MCU in sleep mode.

The I2C slave supports START timing wake-up, address match wake-up and data reception completion wake-up.

Software setup process:

- Turn off the I2C master
- Set slave address
- According to the required wake-up event, set SE, ADME or BFE interrupt enable
- Set the corresponding GPIO to I2C function
- Set SSPEN and start the I2C slave
- Enter sleep mode and wait for data reception
- When the wake-up event comes, the software queries the wake-up source and handles the I2C data transmission

21.10.5 DMA

The I2C slave supports DMA. It should be noted that the bus clock (APBCLK) of the I2C module must be enabled in order to perform the DMA operation. The bus clock is used to generate DMA requests and receive DMA responses.

The slave uses DMA to receive data

When the I2C slave receives the correct address, it generates the ADM interrupt flag. After the software responds to the interrupt, it queries the received R/W bit. If it is 0, the master is ready to write data to the slave. At this time, the software can configure the specific DMA channel as I2C_RX and enable the DMAEN of the I2C slave; then every time the slave completes a byte reception, a DMA request will be generated and the DMA will be notified to read the SSPBUF.

There are two possibilities to end DMA slave reception:

1) The data transfer length has not reached the DMA length configuration, and the master has issued a STOP sequence, the software should respond to the STOP interrupt and actively handle this situation;

2) The data transfer length reaches the DMA length configuration, but because the DMA request is generated after the slave sends back ACK, the software should respond to the DMA transfer completion interrupt and clear ACKEN to zero, so that the slave will send back NACK to end this communication after receiving the next byte.

The flow of receiving by the slave using DMA is as follows:

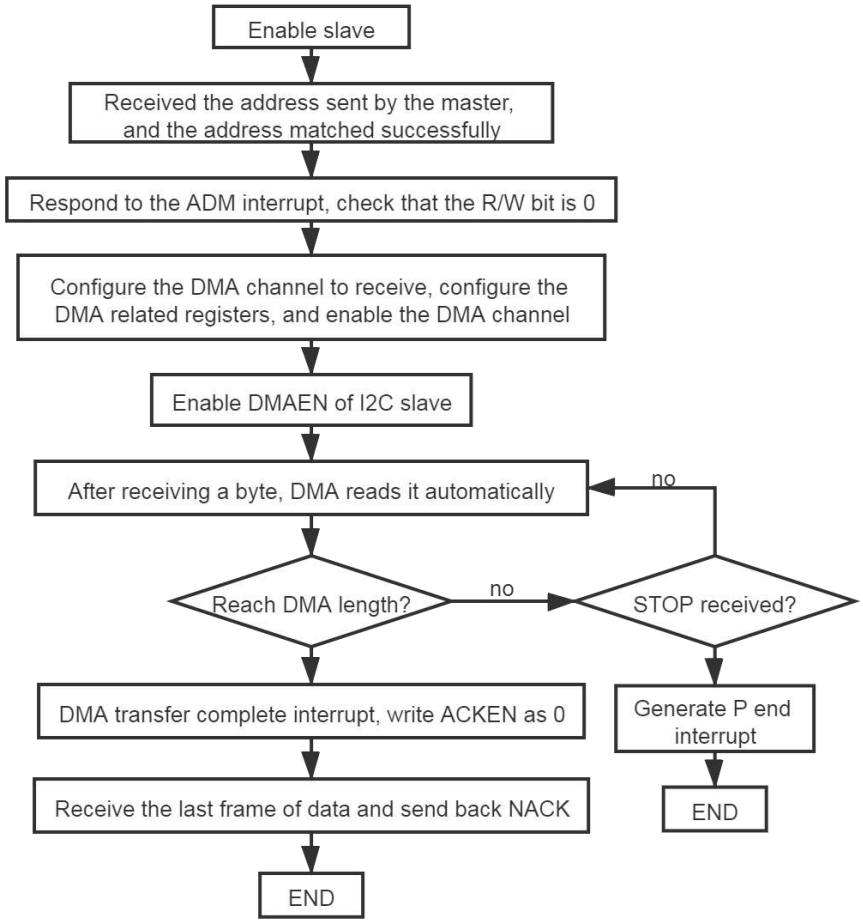


Figure 21-25 I2C Slave DMA Receiving Flowchart

The slave uses DMA to send data

When the I2C slave receives the correct address, it generates the ADM interrupt flag. After the software responds to the interrupt, it queries the received R/W bit. If it is 1, the master is ready to read data from the slave. At this time, the software needs to read the SSPBUF to clear the BF flag, then configure the specific DMA channel as I2C_TX, and enable the DMAEN of the I2C slave; then when the slave data buffer SSPBUF is empty, it will generate a DMA request to notify the DMA to write to the SSPBUF.

Only the master sends back NACK to end the read operation. When the read data length is greater than the transfer length set by DMA, since DMA no longer responds to I2C requests, the slave will pull down SCL until the software disables the I2C slave module.

The flow of the slave using DMA to send is as follows:

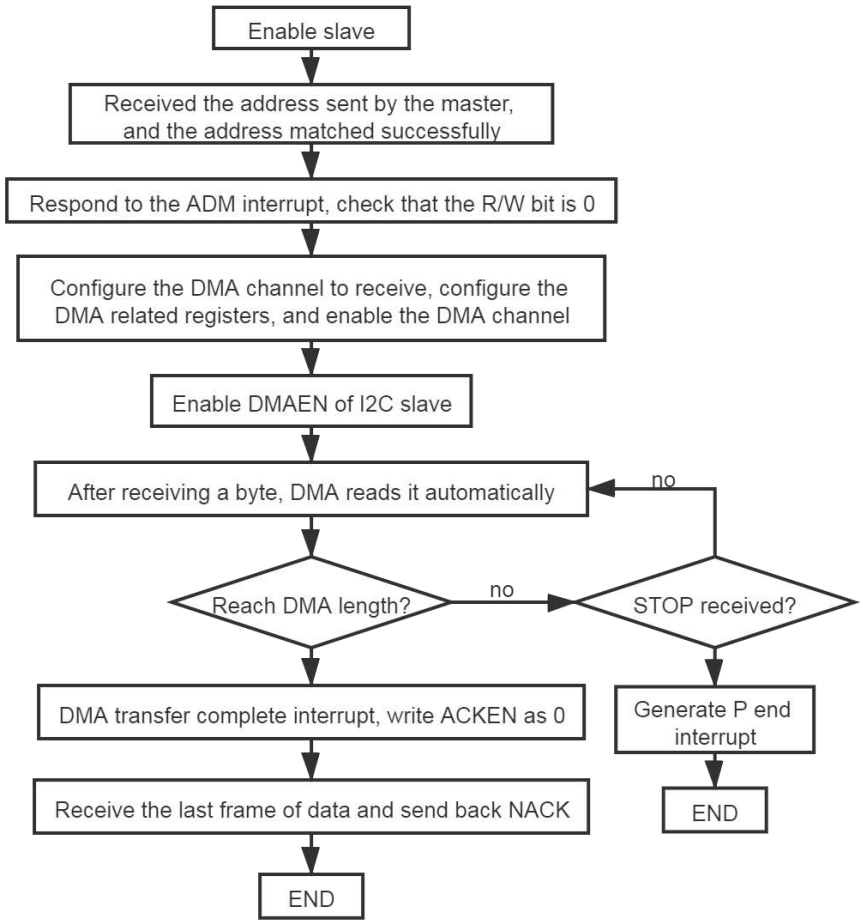


Figure 21-26 I2C Slave DMA Sending Flowchart

21.10.6 Slave Timing

Since the data transmission and reception of the slave only uses SCL, some analog delay is needed to realize the data establishment and retention time control of SDA, and the timing of SCL is completely controlled by the master.

The timing control of the slave is shown in the figure below. According to I2C protocol requirements, the minimum data retention time of SDA relative to the falling edge of SCL is 0ns, that is, the slave can use the falling edge of SCL to send data to meet the requirements. However, considering the actual fall time of the SCL waveform on the bus, in order to better cover the retention time requirements, an extra RC delay greater than 300ns is added to the SDA output. This delay only needs to be applied to the SDA output of the I2C slave (SSP_SDAO).

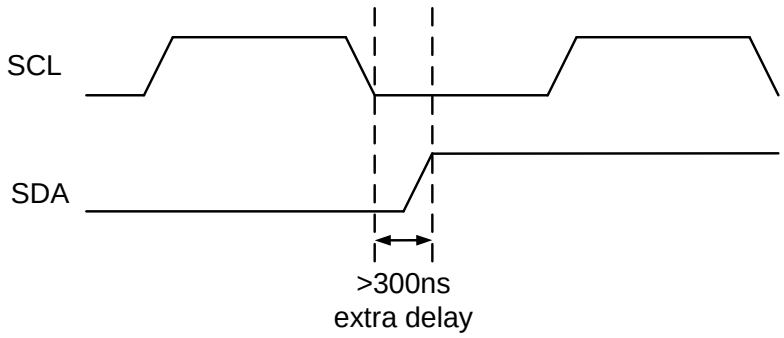


Figure 21-27 SDA Output Delay Waveform

21.11 Register

Offset	Name	Symbol
I2C(Base address: 0x40012400)		
0x00000000	I2C Master Config Register	I2C_MSPCFGR
0x00000004	I2C Master Control Register	I2C_MSPCR
0x00000008	I2C Master Interrupt Enable Register	I2C_MSPIER
0x0000000C	I2C Master Interrupt Status Register	I2C_MSPISR
0x00000010	I2C Master Status Register	I2C_MSPSR
0x00000014	I2C Master Baud Rate Generator Register	I2C_MSPBGR
0x00000018	I2C Master Transfer Buffer Register	I2C_MSPBUF
0x0000001C	I2C Master Timing Control Register	I2C_MSPTCR
0x00000020	I2C Master Time-Out Register	I2C_MSPTOR
0x00000024	I2C Slave Control Register	I2C_SSPCR
0x00000028	I2C Slave Interrupt Enable Register	I2C_SSPIER
0x0000002C	I2C Slave Interrupt Status Register	I2C_SSPISR
0x00000030	I2C Slave Status Register	I2C_SSPSR
0x00000034	I2C Slave Transfer Buffer Register	I2C_SSPBUF
0x00000038	I2C Slave Address Register	I2C_SSPADR

21.11.1 I2C Master Config Register (I2C_MSPCFGR)

NAME	I2C_MSPCFGR							
Offset	0x00000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-						AUTOEND	DMAEN
access	U-0						R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						TOEN	MSPEN
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:15	--	RFU: **Reserved, read as 0**
17	AUTOEND	Master DMA Automatic Ending 1: When the DMA transfer of the specified length data is completed, the STOP condition is sent automatically 0: Wait for software to take over after the DMA transfer of the specified length is completed
16	DMAEN	Master DMA Enable 0: DMA disable 1: DMA enable
15:2	--	RFU: **Reserved, read as 0**
1	TOEN	SCLPull-down Timeout Enable 1: Timeout enable, the timeout period is defined by the MSPTO register 0: Timeout disable
0	MSPEN	I2CMaster Mode Enable 1: I2Cmaster mode enable 0: I2Cmaster mode disable

21.11.2 I2C Master Control Register (I2C_MSPCR)

NAME	I2C_MSPCR							
Offset	0x00000004							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				RCEN	PEN	RSEN	SEN
access	U-0				R/W-0	R/W/Dy-0	R/W/Dy-0	R/W/Dy-0

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3	RCEN	In master receive mode, receive enable 1: Receive enable 0: Receive disable

bit	name	functional description
		In master communication, after the software sends the address byte, it switches the transfer direction to master reception by setting RCEN, and then it can receive data from the slave. RCNE remains 1 during the receiving process until the software sets PEN to send the STOP sequence.
2	PEN	STOP condition generation enable bit, software writes1 to send STOP condition, cleared by hardware (Stop Enable)
1	RSEN	Repeated START timing generation enable bit, software writes 1 to send Repeated START condition, cleared by hardware (Repeated Start Enable)
0	SEN	START condition generation enable bit, software writes 1 to send START condition, cleared by hardware (Start Enable)

21.11.3 I2C Master Interrupt Enable Register (I2C_MSPIER)

NAME	I2C_MSPIER							
Offset	0x00000008							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	WCOLIE	TOIE	SIE	PIE	NACKIE	TXIE	RXIE
access	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:7	--	RFU: **Reserved, read as 0**
6	WCOLIE	Write Collision Interrupt Enable 1: Enable 0: Disable
5	TOIE	SCL Overtime Enable 1: Enable 0: Disable
4	SIE	START Interrupt Enable 1: Enable 0: Disable
3	PIE	STOP Interrupt Enable

bit	name	functional description
		1: Enable 0: Disable
2	NACKIE	Master Mode Non-ACK Interrupt Enable 1: Enable 0: Disable
1	TXIE	Master Mode Trasnmit Done Interrupt Enable 1: Enable 0: Disable
0	RXIE	Master Mode Receive Done Interrupt Enable 1: Enable 0: Disable

21.11.4 I2C Master Interrupt Status Register (I2C_MSPISR)

NAME	I2C_MSPISR							
Offset	0x0000000C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	WCOL	TO	S	P	ACKSTA	TXIF	RXIF
access	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:7	--	RFU: **Reserved, read as 0**
6	WCOL	Write collision detection bit, MCU can only write MSPBUF after completing START timing or sending a completed read/write frame, otherwise write collision occurs; hardware set, software writes 1 to clear (Write Collision Interrupt Flag) 1: Write collision occurred 0: No collision
5	TO	SCL overtime interrupt flag, only works when TOEN is 1 (SCL Over Time Interrupt Flag) 1: SCL overtime occurred 0: No SCL overtime occurred
4	S	START timing sending completion interrupt flag, hardware set, cleared

bit	name	functional description
		after reading (Start Interrupt flag)
3	P	STOP timing sending completion interrupt flag, hardware set, cleared after reading(Stop Interrupt flag)
2	ACKSTA	Response signal from the slave in master sending mode; this flag can generate an interrupt when a NACK is received after a master send; hardware set, software write 1 to clear (Acknowledge Status Flag) 1: Slave responds to NACK 0: Slave responds to ACK
1	TXIF	I2C master mode transmit done interrupt flag, hardware set, software write 1 to clear(Trasnmit Done Interrupt Flag) This flag register is set after the master receives the ACK or NACK sent back from the slave.
0	RXIF	I2C master mode receive done interrupt flag, hardware set, software write 1 to clear (Receive Done Interrupt Flag) This flag register is set after the master sends back an ACK or NACK.

21.11.5 I2C Master Status Register (I2C_MSPSR)

NAME	I2C_MSPSR							
Offset	0x00000010							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		BUSY	RW	-	BF	-	ACKMO
access	U-0		R-0	R-0	U-0	R-0	U-0	R/W-0

bit	name	functional description
31:6	--	RFU: **Reserved, read as 0**
5	BUSY	I2C Communication Status Bit (Busy) 1: The interface is in the read/write state and data transfer is in progress 0: Data transfer has been completed
4	RW	I2C Transfer Direction Status Bits (Read/Write) 1: The master reads data from the slave 0: The master writes data to the slave

bit	name	functional description
3	--	RFU: **Reserved, read as 0**
2	BF	Buffer Full Status Bit (Buffer Full) Receive. 1: Reception complete, MSPBUF full 0: Reception not complete, MSPBUF empty Send. 1: Transmitting, MSPBUF full 0: Sending complete, MSPBUF empty
1	--	RFU: **Reserved, read as 0**
0	ACKMO	Status of the master response signal in master receive mode (Ack Master output) 1: The master sends back a NACK 0: The master sends back a ACK ***Note:*** *The P flag register must be cleared before the software can set ACKMO*

21.11.6 I2C Master Baud rate Generator Register (I2C_MSPBGR)

NAME	I2C_MSPBGR							
Offset	0x00000014							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							MSPBRGH[8]
access	U-0							R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	MSPBRGH[7:0]							
access	R/W-00010011							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							MSPBRGL[8]
access	U-0							R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	MSPBRGL[7:0]							
access	R/W-00010011							

bit	name	functional description
31:25	--	RFU: **Reserved, read as 0**
24:16	MSPBRGH	The width of the SCL clock low level, counted by the I2C operating clock (Master SCL High level length)
15:9	--	RFU: **Reserved, read as 0**

8:0	MSPBRGL	The width of the SCL clock low level, counted by the I2C operating clock. (Master SCL Low level length)

21.11.7 I2C Master Transfer Buffer (I2C_MSPBUF)

NAME	I2C_MSPBUF							
Offset	0x00000018							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SSPBUF							
access	R/W-00000000							

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:0	MSPBUF	MSPBUF[7:0]:The reading and writing of data is accomplished through the operation of MSPBUF. When sending, a write operation is performed to the MSPBUF and the data send/receive shift register (MSPSR) is also loaded; when receiving, the MSPBUF and MSPSR form a double buffer structure and the data is read out to the MSPBUF. After receiving a byte of data, the MSPSR loads the data into the MSPBUF and at the same time sets the I2CIF. MSPSR is not a direct register and has no physical address. (Master data Buffer)

21.11.8 I2C Master Timing Control Register (I2C_MSPTCR)

NAME	I2C_MSPTCR							
Offset	0x0000001C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							

bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							SDAHD[8]
access	U-0							R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SDAHD[7:0]							
access	R/W-00001010							

bit	name	functional description
31:9	--	RFU: **Reserved, read as 0**
8:0	SDAHD	Defines the SDA hold time parameter with respect to the falling edge of SCL, counted by the I2Coperating clock (SDA hold delay) **Note:** The minimum effective value is 1, the maximum effective value is MSPBRGL

21.11.9 I2C Master Time-Out Register (I2C_MSPTOR)

NAME	I2C_MSPTOR							
Offset	0x00000020							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name					TIMEOUT[11:8]			
access					R/W-1111			
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TIMEOUT[7:0]							
access	R/W-1111 1111							

bit	name	functional description
31:12	--	RFU: **Reserved, read as 0**
11:0	TIMEOUT	Defines the slave SCL low stretching timeout period, which can be rewritten by software with MSPEN=0 (SCL stretching Time Out) $T_{SCL\_STRETCHING\_TIMEOUT}$=TIMEOUT[11:0] * T_{SCL}

21.11.10 I2C Slave Control Register (I2C_SSPCR)

NAME	I2C_SSPCR							
Offset	0x00000024							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						SCLSEN	DMAEN
access	U-00000000						R/W-1	RW-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			ACKEN	-	ANFEN	A10EN	SSPEN
access	U-0			RW-1	U-0	RW-0	RW-0	RW-0

bit	name	functional description
31:10	RFU	RFU: **Reserved, read as 0**
9	SCLSEN	I2C slave clock stretching enable (SCL Stretching Enable) 0: Disable slave clock stretching 1: Enable slave clock stretching ***Note:*** *When the slave uses DMA communication, SCLCEN must be set to 1*
8	DMAEN	I2C slave DMA enable 1: Enable DMA function 0: Disable DMA function
7:5	-	RFU: **Reserved, read as 0**
4	ACKEN	ACK enable bit (Slave Ack Enable) 1: Slave will send back ACK after receiving 0: Slave does not send back ACK
3	SDAO_DLYEN	SDA slave output delay enable 0: Bypass slave SDA output delay 1: Enable slave SDA output delay
2	SCLI_ANFEN	SCL slave input analog filter enable 0: Bypass analog filter 1: Enable analog filter
1	A10EN	10bit Slave address enable 1: Slave uses 10bit address 0: Slave uses7bit address
0	SSPEN	I2C slave enable bit 1: Enable I2C slave

bit	name	functional description
		0: Disable I2C slave

21.11.11 I2C Slave Interrupt Enable Register (I2C_SSPIER)

NAME	I2C_SSPIER							
Offset	0x00000028							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	--							
access	U-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	--							
access	U-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	--							
access	U-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ADEIE	SIE	PIE	WCOLIE	SSPOVIE	ADMIE	TXIE	RXIE
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7	ADEIE	Slave Address Error Interrupt Enable, 1 is valid
6	SIE	Start Interrupt Enable, 1 is valid
5	PIE	Stop Interrupt Enable, 1 is valid
4	WCOLIE	Write Collision Interrupt Enable, 1 is valid
3	SSPOVIE	Slave Buffer Overflow Interrupt Enable, 1 is valid
2	ADMIE	Slave address match interrupt enable, 1 is valid
1	TXIE	Transmit Complete Interrupt Enable, 1 is valid
0	RXIE	Receive Complete Interrupt Enable, 1 is valid

21.11.12 I2C Slave Interrupt Status Register (I2C_SSPISR)

NAME	I2C_SSPISR							
Offset	0x0000002C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	--							
access	U-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	--							
access	U-00000000							

NAME	I2C_SSPISR							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	--							
access	U-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ADE	S	P	WCOL	SSPOV	ADM	TXIF	RXIF
access	R/W1C-0	R-0	R-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7	ADE	Address error flag, hardware set, write 1 to clear In the case of a 7bit address, the address byte starting with 11110 is received, or when the first byte does not start with 11110 in the case of a 10bit address, ADEE is triggered.
6	S	The start sequence is detected, hardware set, software is automatically cleared after reading (Start flag)
5	P	The stop sequence is detected, hardware set, software is automatically cleared after reading (Stop flag)
4	WCOL	Write Collision flag, hardware set, write 1 to clear 1: In the case of BF=1, the software writes new data to SSPBUF 0: No write collision When WCOL occurs, new data will be discarded
3	SSPOV	Slave buffer overflow flag, hardware set, write 1 to clear 1: In the case of BF=1, the slave receives new data 0: No receive overflow If the slave enables SCL stretching, the received data will not overflow; therefore, SSPOV can only be set when SCLSEN=0.
2	ADM	Slave address matched flag, hardware set, write 1 to clear 1: The received 7bit or 10bit address is consistent with the contents of the SLAVE_ADDR register 0: The received address is inconsistent with SLAVE_ADDR
1	TXIF	I2C slave transmit interrupt flag, hardware set, write 1 to clear
0	RXIF	I2C slave receive interrupt flag, hardware set, write 1 to clear

21.11.13 I2C Slave Status Register (I2C_SSPSR)

NAME	I2C_SSPSR							
Offset	0x00000030							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	RFU							
access	U-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

NAME	I2C_SSPSR							
name	RFU							
access	U-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	RFU							
access	U-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RFU				BUSY	RW	DA	BF
access	U-0				R-0	R-0	R-0	R-0

bit	name	functional description
31:4	RFU	RFU: **Reserved, read as 0**
3	BUSY	Slave communication flag (Busy) 1: Slave data receiving and sending 0: Slave idle
2	RW	Read/write direction status register (Read/Write) 1: The slave receives R/W=1, and the slave needs to send data to the master 0: Slave is in the state of receiving data
1	DA	Data/address frame indication 1: The last byte received is data 0: The last byte received is the address
0	BF	Slave buffer full flag 1: SSPBUF full 0: SSPBUF empty

21.11.14 I2C Slave Transfer Buffer (I2C_SSPBUF)

NAME	I2C_SSPBUF							
Offset	0x00000034							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	RFU							
access	U-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	RFU							
access	U-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	RFU							
access	U-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SSPBUF							
access	RW-00000000							

bit	name	functional description
31:8	RFU	RFU: **Reserved, read as 0**
7:0	SSPBUF	SSPBUF[7:0]: The reading and writing of data is accomplished through the operation of SSPBUF. When sending, a write operation is performed to the SSPBUF and the data send/receive shift register (SSPSR) is also loader; when receiving, the SSPBUF and SSPSR form a double buffer structure and the data is read out to the SSPBUF. After receiving a byte of data, the SSPSR loads the data into the SSPBUF and at the same time sets the I2CIF. SSPSR is not a direct register and has no physical address. (Slave Buffer)

21.11.15 I2C Slave Address Register (I2C_SSPADR)

NAME	I2C_SSPADR							
Offset	0x00000038							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	RFU							
access	U-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	RFU							
access	U-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	RFU						SSPADDR	
access	U-000000						RW-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SSPADDR							
access	RW-00000000							

bit	name	functional description
31:10	RFU	RFU: **Reserved, read as 0**
9:0	SSPADDR	Slave Address Register A10EN = 1, 10 bits are valid A10EN = 0, only the lower 7 bits are valid

22 Universal Asynchronous Receiver/Transmitter (UART)

22.1 Introduction

The features of UART serial communication module are as follows

- Software configurable baud rate
- 5 independent channels (UART0, UART1, UART3, UART4, UART5)
- Full duplex communication port
- UART has data reception completion/reception error interrupt, and prompts the type of error
- Configurable data length, support 6, 7, 8, 9bits
- Configurable stop bit, support 1 stop bit or 2 stop bit
- It can be configured as infrared modulation output function, and the carrier frequency can be set, and the carrier duty cycle can be set
- Support DMA
- Support receiving timeout mechanism

22.2 Block Diagram

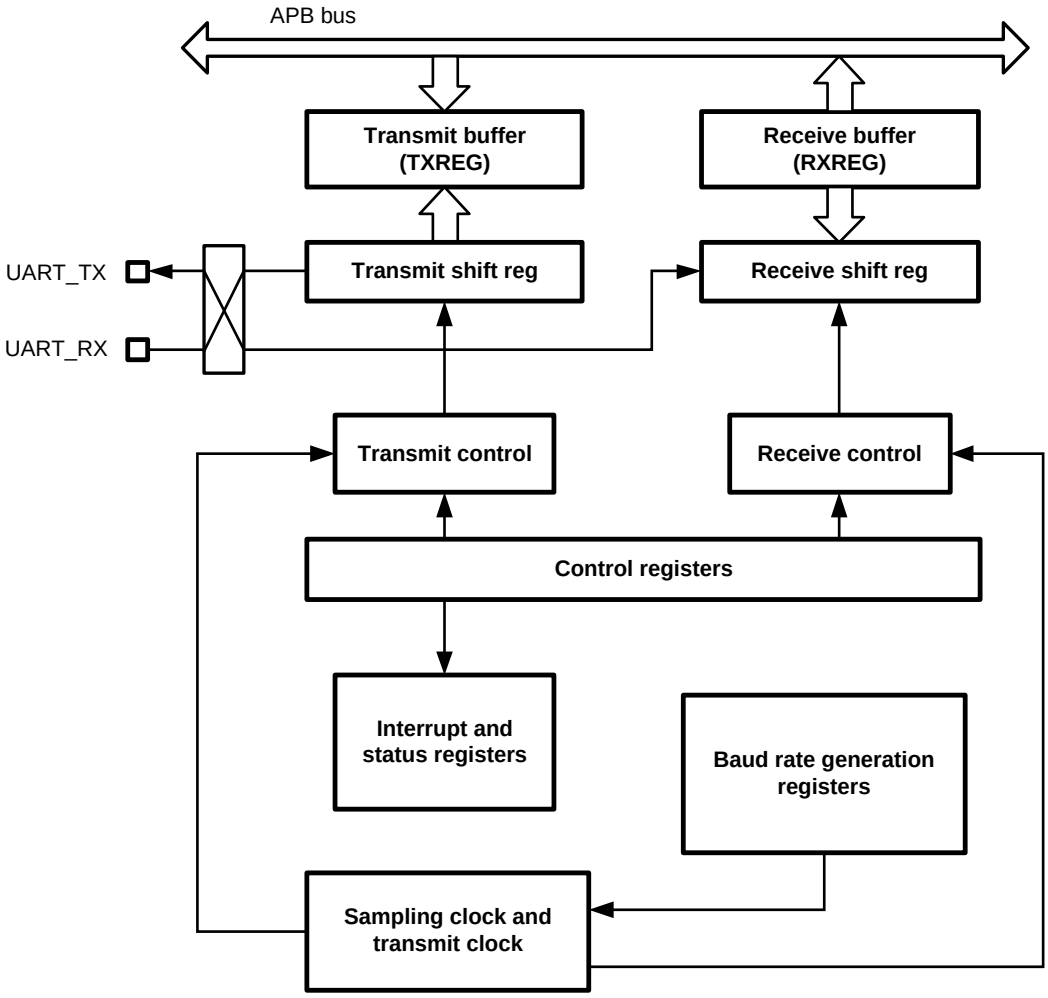


Figure 22-1 UART Block Diagram

22.3 Pin Definition

The UART module uses 2 pins to communicate with external devices, and the receiving and sending signals of each UART may be mapped to different GPIOs.

The following table shows the pin mapping of FM33LG0x8:

Pin		UARTx	Symbol	Function
PA2	PA13	UART0	UART0_RX	Data reception
PA3	PA14		UART0_TX	Data transmission
PC2	PB13	UART1	UART1_RX	Data reception
PC3	PB14		UART1_TX	Data transmission
PB0	PD7	UART3	UART3_RX	Data reception
PB1	PD8		UART3_TX	Data transmission
PA0	PB2	UART4	UART4_RX	Data reception
PA1	PB3		UART4_TX	Data transmission
PC4	PD0	UART5	UART5_RX	Data reception
PC5	PD1		UART5_TX	Data transmission

Table 22-1 UART Pin List

When the UART function is mapped to multiple pins at the same time:

- PA2 and PA13 are configured as digital peripheral functions at the same time
  - Only the RX signal on PA2 will be input into the module
- PC2 and PB13 are configured as digital peripheral functions at the same time
  - Only the RX signal on PA2 will be input into the module
- PC4 and PD0 are configured as digital peripheral functions at the same time
  - Only the RX signal on PC4 will be input into the module
- PA0 and PB2 are configured as digital peripheral functions at the same time
  - Only the RX signal on PA0 will be input into the module
- When the UART sending function is mapped to multiple GPIOs at the same time, these pins will send data at the same time.

22.4 UART Mode

FM33LG0xxintegrates different types of UART (LPUART), and the differences are shown in the following table:

UART character	UART0/1	UART3/4/5	LPUART0/1/2
DMA Supply	Y	Y	Y
Half Duplex/Full Duplex	Y	Y	Y
The infrared emission	Y	Y	-
Dual clock domain (working clock independent of bus)	Y	-	Y
Sleep wake up	Y	-	Y
Receive Timeout	Y	-	-
Transmission Delay	Y	-	-
Data Length	6、7、8、9bits		

Table 22-2 UART Mode

22.5 UART Character Format

The basic timing of UART transmission characters is shown in the figure below. Each character contains at least 1bit START and at least 1bit STOP bits, data lengths can be configured to be 6-9bits, and parity bit can be selected.

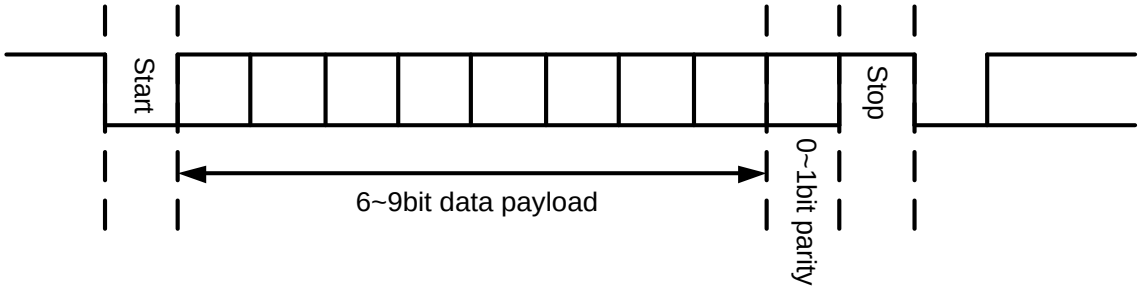


Figure 22-2 UART Character Format

UART supports multiple frame formats controlled by the UARTx_CSR.PDSEL register and UARTx_CSR.PARITY register, which are listed in the table below:

PDSEL	PARITY	Frame Format[1]
00	00	[Start | 7 bits data | Stop]
	01, 10	[Start | 7 bits data | Parity | Stop]
01	00	[Start | 8 bits data | Stop]

	01, 10	[Start | 8 bits data | Parity | Stop]
10	00	[Start | 9 bits data | Stop]
	01, 10	[Start | 9 bits data | Parity | Stop]
11	00	[Start | 6 bits data | Stop]
	01, 10	[Start | 6 bits data | Parity | Stop]

Table 22-3 UART Data Frame Format

[1]: The Stop bit can be either 1bit or 2bits, depending on the STOPCFG register.

Note: *THE PDSEL register is used to configure the data length of the frame. The communication frame length is [start bit + data bit + check bit + stop bit].*

22.6 Functional Description

22.6.1 Clock and Reset

UART0 and UART1 use a dual clock structure:

- The bus register clock is represented by PCLK and is derived from APBCLK. When the CPU or DMA needs to access the UART internal registers, PCLK must be enabled
- The data sending and receiving clock is expressed by UCLK, which can be derived from RCHF, SYSCLK, and XTHF in addition to APBCLK, and can work independently of APBCLK. UCLK must be enabled to send and receive data.

The control of PCLK and UCLK is completed in the CMU module, and the corresponding CMU control register must be correctly configured before UART communication.

The dual clock structure can make the work of UART0 and UART1 not limited to the configuration of APBCLK. When some peripherals need to work at a very high APBCLK frequency, the UART can still work at a reduced frequency; or vice versa, The CPU works at a lower frequency, and it does not affect the UART for data communication at a higher baud rate.

In theory, there is no constraint on the relative relationship between PCLK and UCLK, and UCLK can be faster or slower than PCLK. But the application needs to pay attention to whether the CPU or DMA has time to carry out data transfer when the frequency difference between the two is large.

Different from UART0 and UART1, UART3, UART4 and UART5 adopt a single clock structure. At this time, UCLK=PCLK, and the UART data receiving and sending clock is also derived from APBCLK.

The reset register (UARTxRST) in the RMU must be cleared before the UART module works.

22.6.2 Bit Receiving Sampling

UART oversampling the received data by 8 times or 16 times the baud rate, and performs a majority decision of two out of three in the middle of each bit to improve the ability to suppress signal noise.

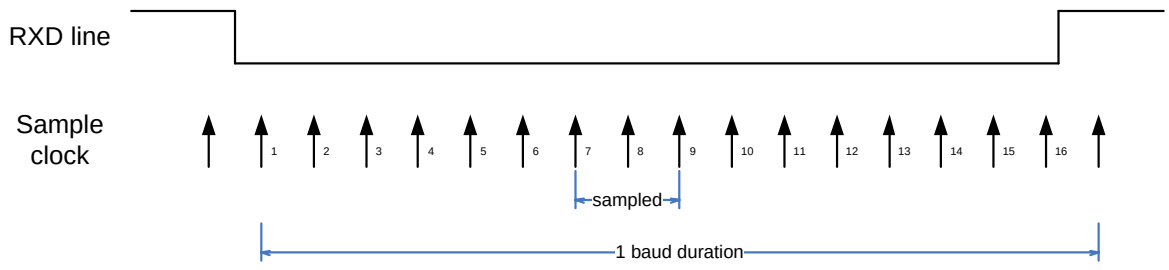


Figure 22-3 Bit Receiving 16 Times Sampling

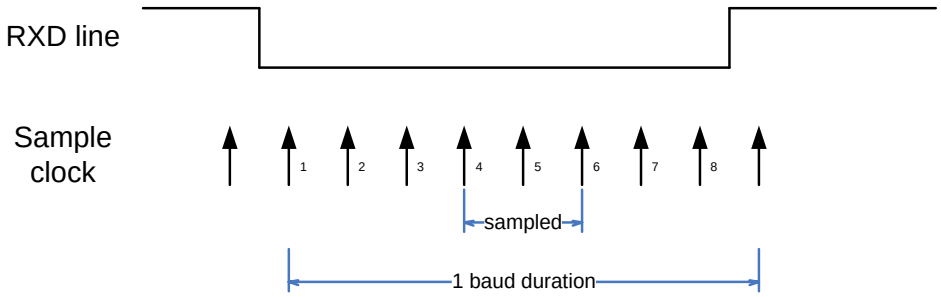


Figure 22-4 Bit Receiving 8 Times Sampling

The bit received by the receiving shift register is the result of the majority decision. For example, if the result of three samplings is 001, the judgment is 0; if it is 011, the judgment is 1.

Through the OVSM register, you can configure the oversampling multiple when the UART receives data. If the UART performs 16 times oversampling of the input signal, the SPBRG configuration must not be less than 16, that is, the UART working clock must be at least 16 times the baud rate. If the UART performs 8 times oversampling of the input signal, the SPBRG configuration must not be less than 8, that is, the UART working clock must be at least 8 times the baud rate.

When selecting a smaller oversampling multiple, a higher communication baud rate can be obtained.

22.6.3 Data Transmission

In the sending mode, the serial data sending circuit of the UART mainly includes a sending shift register (TSR), and the TSR function is to shift the data out one by one. The data to be sent must be written to the sending buffer first. After the software sets the TXEN register, if the transmit buffer is not empty, the UART loads the buffer data into the TSR and starts shifting output.

Note: Since the register operation clock and the baud rate clock are asynchronous, when the transmission starts, it is necessary to wait for the baud rate clock to arrive. Therefore, there is a maximum of 1 baud delay between the TXEN setting and the UART starting to transmit the Start bit.

TXBE and TXSE are the transmit interrupt flag bits, which respectively indicate that the transmit buffer is empty and the TSR is empty. The software can choose to generate a transmit completion interrupt at an appropriate point in time.

Under normal circumstances, the TSR register is empty at the beginning, the data transmission needs to set the baud rate SPBRG, enable the sending module (set TXEN to 1), and then write to the TXBUF register to start sending. You can also write the TXBUF register after setting the baud rate SPBRG, and then set the TXEN enable transmission module to start data transmission. If the transmit module enable bit TXEN is cleared to 0 during data transmission, the data transmission will be interrupted and the transmit module will also be reset.

The figure below is an example of UART asynchronous transmission. In this example, the software

first writes data to TXBUF, and then initiates transmission by setting TXEN.

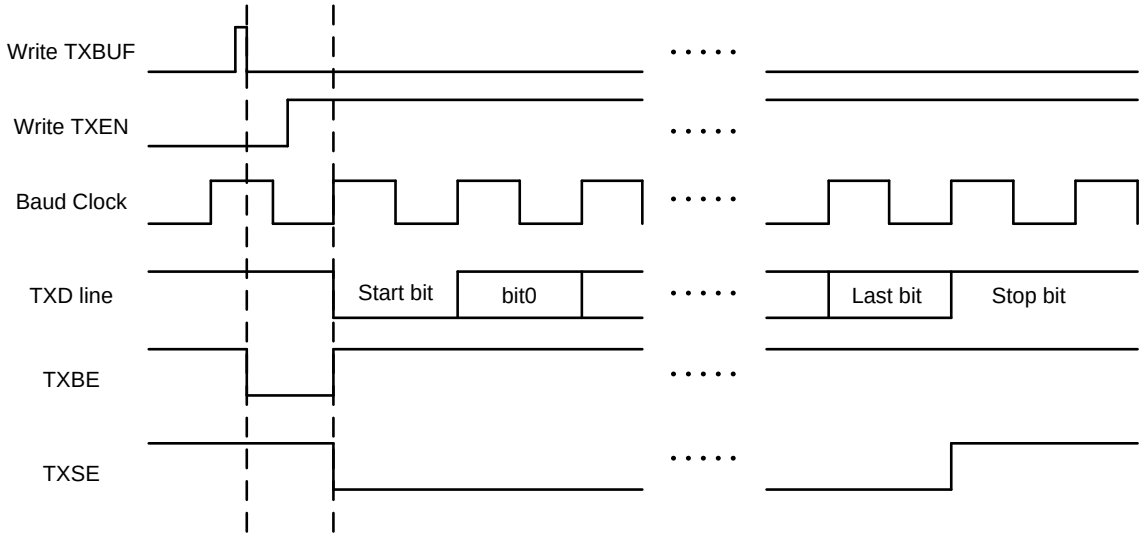


Figure 22-5 UART Asynchronous Transmission Waveform 1

The recommended steps in the figure are as follows:

- Select the appropriate baud rate and initialize SPBRG
- If an interrupt is required, set TXSE_IE or TXBE_IE
- Determine the format of data transmission: Set PDSEL register, determine the length of data transmission; set the PARITY register to choose whether to send a PARITY bit and the type of PARITY, and set the STOPSEL register to decide whether to send a 1-bit or 2-bit stop
- If the data to send is infrared modulated, write the appropriate value to the IRCON register to obtain the corresponding modulation frequency and duty cycle, and set TXIREN
- Write the data to TXBUF register
- Enable transmission: set TXEN

The software can also set TXEN first and then write TXBUF. UART will immediately start the sending process after the data is written to TXBUF.

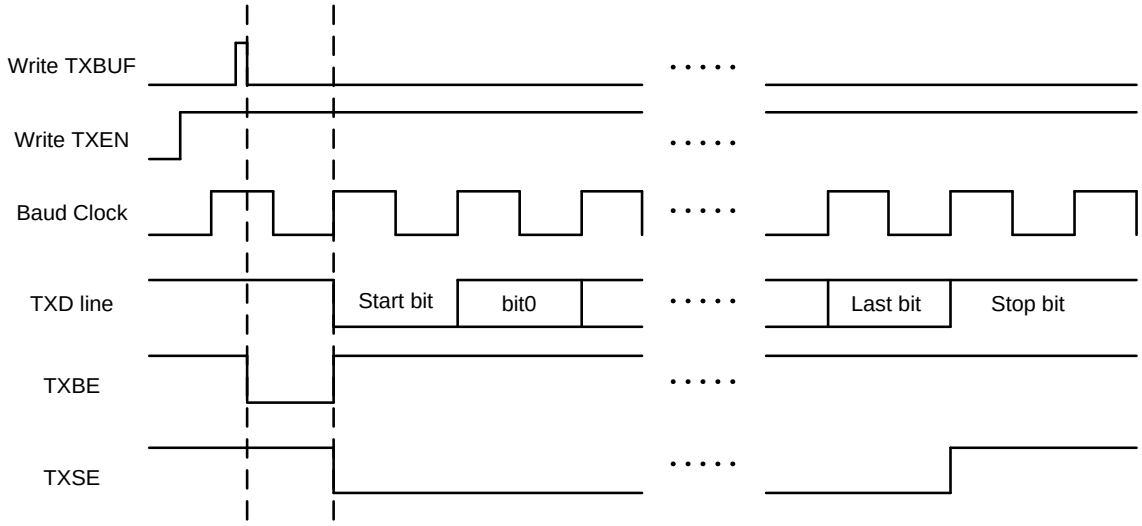


Figure 22-6 UART Asynchronous Transmission Waveform 2

When TXBUF is empty, the software can immediately write the next data to be sent, in order to achieve continuous data transmission without interval.

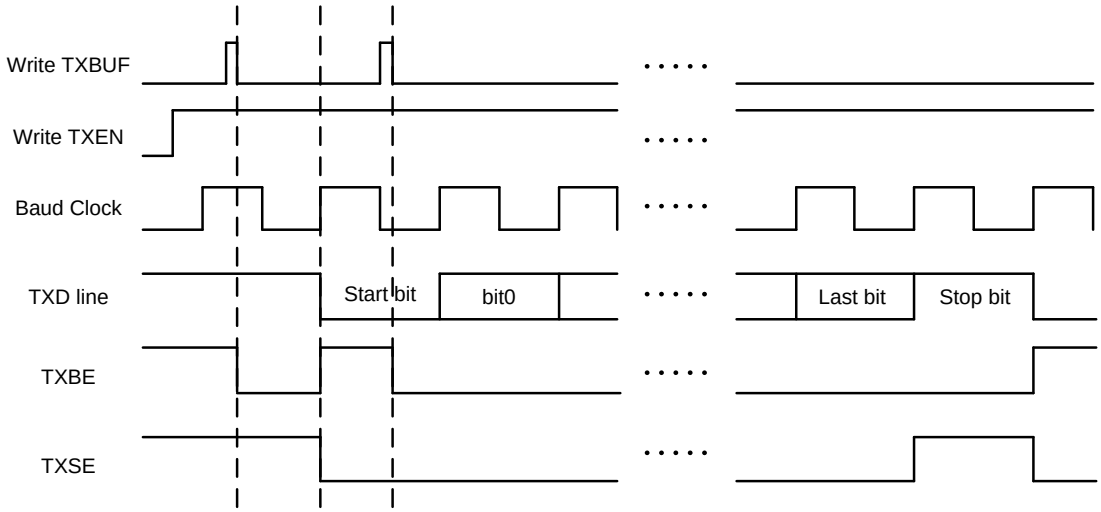


Figure 22-7 UART Asynchronous Transmission Waveform 3

22.6.4 Data Reception

The serial data receiving of UART uses a receive shift register (RSR). When the stop bit is received, RSR feeds the received data into the receive buffer (RXBUF). The interrupt flag RXBF is set to 1 after each received byte is copied into the receive buffer. When new data is received when RXBUF is full, the original data in the RX buffer will be overwritten, and RXBF flag is set again. Meanwhile, receive overflow error occurs, and OERR is set to 1.The OERR flag can be cleared by writing 1by software or reading RXBUF.

During the receiving process, if the correct stop bit is not detected, a frame format error occurs and FERR is set to 1.If a parity error occurs, flag bit PERR is set to 1.

The recommended asynchronous receive procedure is as follows:

- Select the appropriate baud rate and initialize SPBRG
- If an interrupt is required, set RXBF_IE
- Set the format of data receiving: set PDSEL register to determine the length of data sent;
- Set the PARITY register to choose whether to send a PARITY bit and the type of PARITY, and set the STOPSEL register to decide whether to send a 1-bit or 2-bit stop
- Enable receiving: Set RXEN
- At the end of a frame, the RXBF bit is set to 1. If the RXBF_IE bit is set to 1, an interrupt will be generated
- Read PERR, FERR, and OERR registers to determine if there are any data error or overflow
- Read the received data in the RXBUF register

22.6.5 Low-Power Sleep Wake-up (UART0/1)

UART0 and UART1 support chip sleep wake-up triggered by the falling edge of RXD. When the NEWUP register is set, a falling edge event (low-level duration>100ns) on the RXD input will wake up the chip from sleep mode. With this function, UART0/1 can receive data in sleep mode.

The software configuration method is as follows:

- Configure UART register and enable NEWUP
- Configure the UART working clock, configure the baud rate divider register as needed
- Configure the corresponding GPIO as UART data receiving function
- Set RXEN to enable reception
- The software sets the chip to sleep and waits for the UART to receive events

22.6.6 Use DMA for UART Communication

When DMA is enabled, UART will automatically generate the corresponding DMA request when the TX buffer is empty or the RX buffer is full. Application needs to configure DMA channel connections, set the RAM pointer, and enable DMA channels. After that, the DMA will automatically respond to the UART request and complete the data transfer between RAM and UART.

Application example: DMA for UART0 receive

- Configure DMA channel 1 or 3 as RXD0.
- Set corresponding channel parameters: RAM pointer, address increment and decrement, channel priority, transmission length, interrupt settings, etc.
- Enable corresponding DMA channels.
- Configure UART parameters.
- The enabled UART waits for the data to be received.
- UART automatically generates DMA requests when RX buffer is full.
- DMA responds to the request, reads the UART RXBUF, and writes to the specified RAM address.

22.6.7 Transmission Completion Interrupt in DMA Mode

When UART transfers data through DMA, DMA will produce a DMA channel interrupt after a specified length of data has been transferred. But when the channel interrupt occurs, the last frame of data has just been written to the UART TXBF and has not yet been sent. By configuring the DMATXIFCFG register, it is possible to generate a transmit completion interrupt (buffer empty or shift register empty) when the last frame data has been sent. Which will interrupt CPU after all data has been sent.

The software procedure is described as follows:

- Configure DMA channels UART TX
- Disable the DMA channel interrupt.
- Set UART TXBE_IE or TXSE_IE registers to allow interrupts to be generated.
- Set the DMATXIFCFG register, allowing only the last frame of data to produce interrupt output.
- Prepare data to send. Enable the DMA.
- UART transmits continuously until the last frame, no TXBE or TXSE interrupts are generated.
- After the last frame is sent, UART produces a TXBE or TXSE interrupt.

The following table assumes that UART sends N frames via DMA:

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	UART interrupt
0	x	1~N	After each frame is sent, then set.	Inactive
1	0	1~N	After each frame is sent, then set.	Inactive
	1	1~N-1	After each frame is	Inactive

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	UART interrupt
			sent, then set.	
		N	After each frame is sent, then set.	Active

Table 22-4 DMA Transmit Interrupt

22.7 Baud Rate Generation

22.7.1 Baud Rate Generation

The Baud rate register is a 16-bit register whose value X is any integer between 16 and 65535.

Baud rate calculation formula:

$$Baud = F_{CLK} / (SPBRG+1)$$

Note: *FCLK can be different clocks in different UART. For UART2~5, FCLK is APBCLK. For UART0 and UART1, FCLK is a working clock independent of APBCLK.*

To support full duplex communication, the receiving and transmitting baud rates are generated separately.

The following table shows the baud rate at common system clock frequencies:

Baud	F_{CLK}=16MHz			F_{CLK}=8MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1
300	300.0019	0.000625	53333	299.9963	-0.00125	26667
1200	1200.03	0.0025	13333	1199.94	-0.005	6667
2400	2399.88	-0.005	6667	2400.24	0.010001	3333
4800	4800.48	0.010001	3333	4799.04	-0.02	1667
9600	9598.08	-0.02	1667	9603.842	0.040016	833
19200	19207.68	0.040016	833	19184.65	-0.07994	417
38400	38369.3	-0.07994	417	38461.54	0.160256	208
57600	57553.96	-0.07994	278	57553.96	-0.07994	139
115200	115107.9	-0.07994	139	115942	0.644122	69
230400	231884.1	0.644122	69	228571.4	-0.79365	35
460800	457142.9	-0.79365	35	470588.2	2.124183	17

Baud	F_{CLK}=24MHz			F_{CLK}=32MHz		
bps	Actual (bps)	Error%	X+1	Actual (bps)	Error%	X+1

300	300	0	80000	299.9991	-0.00031	106667
1200	1200	0	20000	1199.985	-0.00125	26667
2400	2400	0	10000	2400.06	0.0025	13333
4800	4800	0	5000	4799.76	-0.005	6667
9600	9600	0	2500	9600.96	0.010001	3333
19200	19200	0	1250	19196.16	-0.02	1667
38400	38400	0	625	38415.37	0.040016	833
57600	57553.96	-0.07994	417	57553.96	-0.07994	556
115200	115384.6	0.160256	208	115107.9	-0.07994	278
230400	230769.2	0.160256	104	230215.8	-0.07994	139
460800	461538.5	0.160256	52	463768.1	0.644122	69

Table 22-5 Common Clock Frequency Baud Rate Calculation

22.7.2 Adaptive Baud Rate Generation

Using the Capture function of Timer, the adaptive baud rate can be realized. Generally the external UART device sends a frame according to the negotiated data content (such as 0xF8) at target baud rate, and the timer counts the high level pulse width of the frame data. The MCU reads the timer capture result to calculate the Baud rate, and writes it into the baud rate register. Then the following data can be received with new baud rate. Refer to the Timer section.

22.8 Infrared Modulation

The UART_IRCR register holds an 11-bit frequency divider TZBRG, whose value is any integer between 0 and 2047. All UART share one infrared modulation generator.

Infrared modulation frequency calculation formula:

FIR = FAPBCLK/ (TZBRG+ 1)

The infrared modulation method is: when sending data 0, the infrared frequency is modulated; when sending data 1, the infrared frequency is not modulated.

In order to meet the needs of PNP and NPN infrared optical transistor, register IRFLAG bit controls the polarity of infrared modulation output.

When IRFLAG=0, it is positive polarity output, suitable for PNP.

When IRFLAG=1, it is negative polarity output, suitable for NPN.

The TH register is used to configure the ir modulation duty cycle.

Duty ratio: Y = (TZBRG[10:4] * TH) /(TZBRG+1)

When TH=4’b0000, the duty ratio is Y = (TZBRG[10:1]+1)/(X+1)；

When TZBRG[10:4]=7 'h00, the duty ratio is Y = TH/(TZBRG[3:0]+ 1);

If this time TH>TZBRG [3:0], then ir modulation clock IRCLK is fixed at high level.

When the infrared modulation polarity is reversed (IRFLAG=1), the duty cycle is also 1-y.

The infrared modulation waveform is shown in the following figure:

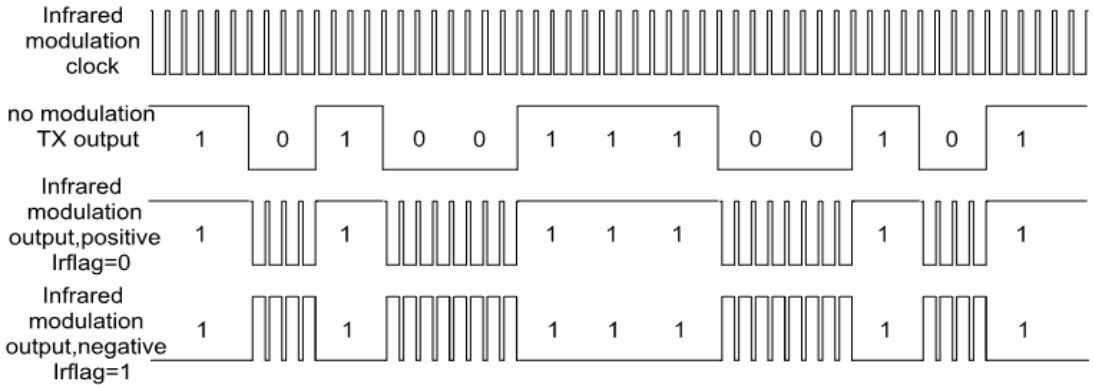


Figure 22-8 The Infrared Modulation Waveform

Duty cycle is defined as the high level length/period regardless of whether the effective level is 0 or 1.

22.9 Receive Timeout

A time-out mechanism is designed for time-sensitive applications such as MODBUS. When the RXTOEN register is enabled, the timeout counter is counted at the baud rate clock. Each time a complete data frame is received, the timeout counter is cleared and the count is restarted. The upper limit of the timeout overflow can be configured by the software with a maximum of 255 baud.

Note: *Receive timeout function is not supported in UART3, USRT4 and UART5.*

22.10 Transmit Delay

Through the TXDLY_LEN register, you can control the time interval between two data frames sent, the unit is Baud. The transmit delay is the interval between the end of the last STOP bit of the previous frame and the start bit of the next frame.

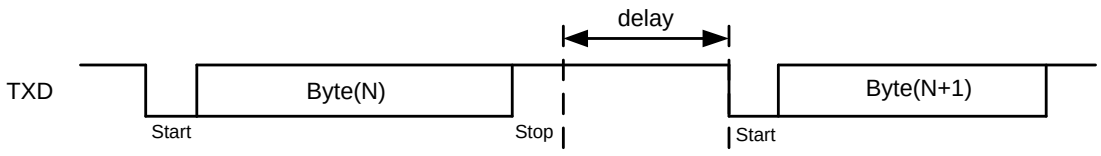


Figure 22-9 UART Transmit Delay

Note*: UART3, UART4 and UART5 does not support sending delay function.*

22.11 Register

Offset	Name	Symbol
UARTI Register (Base address: 0x40017C00)		
0x00	Infrared Modulation Control Register	UART_IRCR
UART0 register (Base address: 0x40012000)		
0x00	UART0 Control Status Register	UART0_CSR
0x04	UART0 Interrupt Enable Register	UART0_IER
0x08	UART0 Interrupt Status Register	UART0_ISR
0x0C	UART0 Time-Out and Delay Register	UART0_TODR
0x10	UART0 Receive Buffer	UART0_RXBUF
0x14	UART0 Transmit Buffer	UART0_TXBUF
0x18	UART0 Baud Rate Generator Register	UART0_BGR
UART1 register (Base address: 0x40016800)		
0x00	UART1 Control Status Register	UART1_CSR
0x04	UART1 Interrupt Enable Register	UART1_IER
0x08	UART1 Interrupt Status Register	UART1_ISR
0x0C	UART1 Time-Out and Delay Register	UART1_TODR
0x10	UART1 Receive Buffer	UART1_RXBUF
0x14	UART1 Transmit Buffer	UART1_TXBUF
0x18	UART1 Baud Rate Generator Register	UART1_BGR
UART3 register (Base address: 0x40017000)		
0x00	UART3 Control Status Register	UART3_CSR
0x04	UART3 Interrupt Enable Register	UART3_IER
0x08	UART3 Interrupt Status Register	UART3_ISR
0x10	UART3 Receive Buffer	UART3_RXBUF
0x14	UART3Transmit Buffer	UART3_TXBUF
0x18	UART3 Baud Rate Generator Register	UART3_BGR
UART4 register (Base address: 0x40017400)		
0x00	UART4 Control Status Register	UART4_CSR
0x04	UART4 Interrupt Enable Register	UART4_IER
0x08	UART4 Interrupt Status Register	UART4_ISR
0x10	UART4 Receive Buffer	UART4_RXBUF
0x14	UART4 Transmit Buffer	UART4_TXBUF
0x18	UART4 Baud Rate Generator Register	UART4_BGR
UART5 register (Base address: 0x40017800)		
0x00	UART5 Control Status Register	UART5_CSR
0x04	UART5 Interrupt Enable Register	UART5_IER
0x08	UART5 Interrupt Status Register	UART5_ISR
0x10	UART5 Receive Buffer	UART5_RXBUF
0x14	UART5 Transmit Buffer	UART5_TXBUF
0x18	UART5 Baud Rate Generator Register	UART5_BGR

22.11.1 Infrared Modulation Control Register (UART_IRCR)

NAME	UART_IRCR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	IRFLAG	TH				TZBRG[10:8]		
access	R/W-0	R/W-0000				R/W-000		
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TZBRG[7:0]							
access	R/W-11010010							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15	**IRFLAG**	Controls the default output polarity when infrared modulation sends data. 0: Positive polarity 1: Negative polarity
14:11	**TH**	Transmission High Duty
10:0	**TZBRG**	Transmission Baud Rate

22.11.2 UARTx Control Status Register (UARTx_CSR)

NAME	UARTx_CSR(x=0,1,3,4,5)							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							BUSY
access	U-0							R-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-						TXIREN	RXTOEN
access	U-0						R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-		OVSM	IOSWAP	NEWUP	DMATXIFCFG	BITORD	STOPCFG
access	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN

access	R/W-00	R/W-00	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:25	-	RFU: **Reserved, read as 0**
24	**BUSY**	UART communication flag, read-only 1: UART is communicating. 0: UARTIDLE
23:18	-	RFU: **Reserved, read as 0**
17	**TXIREN**	Send infrared modulation enablement. 1: Enable infrared modulation transmission. 0: Turn off infrared modulation sending.
16	**RXTOEN**	Receive timeout enablement. 1: Enable the receive timeout function. 0: Turn off receive timeout.
15:14	-	RFU: **Reserved, read as 0**
13	**OVSM**	Exchange of RX and TX pins. 0: The default pin order.(Consistent with package diagram.) 1: Swap pin.
12	**IOSWAP**	RFU: **Reserved, read as 0**
11	**NEWUP**	DMA send completion interrupt, only valid if UART sends through DMA. 1: In the case of IE=1, interrupt signal output is allowed after the last frame is sent in DMA mode; interrupt signal output is not allowed after the data frame before the last frame has been sent. 0: It is up to IE to decide whether to allow interrupt signal output.
10	**DMATXIFCFG**	Bit order in which data is sent/received. 0: LSB first 1: MSB first
9	**BITORD**	Stop bit width configuration, only valid for sending frame format 0: 1stop bit. 1: 2stop bit.
8	**STOPCFG**	Select the data length of each frame; this register is valid for both sending and receiving data. 00: 7 data bit 01: 8 data bit 10: 9 data bit 11: 6 data bit

bit	name	functional description
7:6	**PDSEL**	Parity bit configuration; This register is valid for both sending and receiving data. 00: No parity bit 01: Even 10: Odd 11: RFU
5:4	**PARITY**	Receive data polarity configuration. 0: Standard 1: Inverted
3	**RXPOL**	Transmit data polarity configuration. 0: Standard 1: Inverted
2	**TXPOL**	Receive enable
1	**RXEN**	Transmit enable
0	**TXEN**	RFU: **Reserved, read as 0**

22.11.3 UARTx Interrupt Enable Register (UARTx_IER)

NAME	UARTx_IER(x=0,1,3,4,5)							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				RXTO_IE	RXERR_IE	-	RXBF_IE
access	U-0				R/W-0	R/W-0	U-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	NEWUP_IE	-					TXBE_IE	TXSE_IE
access	R/W-0	U-0					R/W-0	R/W-0

bit	name	functional description
31:12	**-**	RFU: **Reserved, read as 0**
11	**RXTO_IE**	Receive timeout interrupt enable, 1 enable. (Only UART0 and UART1 are valid)
10	**RXERR_IE**	Receive error interrupt enable, 1 enable.
9	**-**	RFU: **Reserved, read as 0**
8	**RXBF_IE**	Receive buffer full interrupt enablement, 1 enable

bit	name	functional description
7	**NEWUP_IE**	RX falling edge asynchronous detection interrupt enable, 1 enable (Only UART0 and UART1 are valid)
6:2	**-**	RFU: **Reserved, read as 0**
1	**TXBE_IE**	Transmit buffer empty and Transmit Register empty interrupt enable, 1 enable
0	**TXSE_IE**	transmit buffer empty and the transmit shift register empty interrupt enabled, 1 enable

22.11.4 UARTx Interrupt Status Register (UARTx_ISR)

NAME	UARTx_ISR(x=0,1,3,4,5)							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-					PERR	FERR	OERR
access	U-0					R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				RXTO	-		RXBF
access	U-0				R/W-0	U-0		R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	NEWKF	-				TX_OERR	TXBE	TXSE
access	R/W-0	U-0				R/W-0	R-0	R/W-0

bit	name	functional description
31:19	**-**	RFU: **Reserved, read as 0**
18	**PERR**	Parity error interrupt flag, hardware set, software write 1 to clear
17	**FERR**	Frame format error interrupt flag, hardware set, software write 1 to clear
16	**OERR**	The receiving buffer overflow error interrupt flag is set when the Receive Buffer is full and the new data is received. When hardware is set and software writes 1 or reads RXBUF, the receiving overflow interrupt will be cleared, and the original data in the receiving buffer will be overwritten by the new data.
15:12	**-**	RFU: **Reserved, read as 0**
11	**RXTO**	Receive timeout interrupt flag, hardware set, software write 1 to clear (UART0 and UART1 only)
10:9	**-**	RFU: **Reserved, read as 0**

bit	name	functional description
8	**RXBF**	Receive Buffer full interrupt flag, hardware set, software writes 1 or reads RXBUF to clear
7	**NEWKF**	RX falling edge asynchronous detection interrupt flag, hardware set, software write 1 to clear (Neg Edge Wakeup Flag write 1 to clear) (Only UART0 and UART1 are valid)
6:3	**-**	RFU: **Reserved, read as 0**
2	**TX_OERR**	Transmit Buffer empty interrupt flag, hardware set, software write TXBUF to clear.
1	**TXBE**	shift register empty interrupt flag. Hardware set, software write 1 or software write tx buffer to clear.
0	**TXSE**	RFU: **Reserved, read as 0**

22.11.5 UARTx Time-Out and Delay Register (UARTx_TODR)

NAME	UARTx_TODR(x=0,1)							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	TXDLY_LEN							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RXTO_LEN							
access	R/W-1111 1111							

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:8	**TXDLY_LEN**	Transmit delay, maximum 255baud
7:0	**RXTO_LEN**	Receive timeout length, maximum 255baud

22.11.6 UARTx Receive Buffer (UARTx_RXBUF)

NAME	UARTx_RXBUF(x=0,1,3,4,5)							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							RXBUF[8]
access	U-0							R-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RXBUF[7:0]							
access	R-0000 0000							

bit	name	functional description
31:9	-	RFU: **Reserved, read as 0**
8:0	**RXBUF**	Receive data buffer

When sending and receiving 7 bits, the received 7 bits data is stored in RXBUF[6:0]

22.11.7 UARTx Transmit Buffer (UARTx_TXBUF)

NAME	UARTx_TXBUF(x=0,1,3,4,5)							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							TXBUF[8]
access	U-0							W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TXBUF[7:0]							
access	W-0000 0000							

bit	name	functional description
31:9	-	RFU: **Reserved, read as 0**
8:0	**TXBUF**	Transmit data buffer register

When sending and receiving 7-bits, the 7bits data sent will be written into TXBUF[6:0]

22.11.8 UARTx Baud Rate Generator Register (UARTx_BGR)

NAME	UARTx_BGR(x=0,1,3,4,5)							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	SPBRG[15:8]							
access	R/W-00000011							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SPBRG[7:0]							
access	R/W-01000001							

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:0	**SPBRG**	Serial Port Baud Rate Generation

Baud rate calculation is detailed in 18.7.1 Baud rate

Note:

When SPBRG <= 0x000F, UARTDIV=16'H000F.

When SPBRG > 0x000F, UARTDIV=SPBR.

23 Low Power UART (LPUART)

23.1 Introduction

LPUART is an enhanced asynchronous serial communication interface. Its working clock can be selected from 32768Hz crystal oscillator clock (XTLF), high frequency ring oscillator clock (RCHF), and low power consumption low frequency ring oscillator clock (RCLF). LPUART can support data reception up to 9600 baud rate. At this time, LPUART has extremely low power consumption and can work in Sleep/DeepSleep mode.

Features:

- Asynchronous data transmission and reception
- 3 independent LPUART (LPUART0, LPUART1, LPUART2)
- Standard UART frame format
  - 1bit start bit
  - Configurable data length, support 6, 7, 8, 9bits
  - Odd parity, even parity or no parity bit
  - 1 or 2bit stop bit
- Programmable data polarity
- When the working clock is XTLF or RCLF, it supports data sending and receiving in Sleep/DeepSleep mode
- Interrupt flag
  - The receiving buffer is full
  - Receive buffer overflow
  - Received frame format error
  - Receive check digit error
  - START detection
  - Data matching
  - Sending completed
- Wake up the chip in sleep mode
  - RXD falling edge wake-up
  - Wake-up from start bit detection

- Wake up after receiving 1 byte
- 1 byte data match wake up

- Support DMA (not supported in Sleep/DeepSleep mode)

23.2 Block Diagram

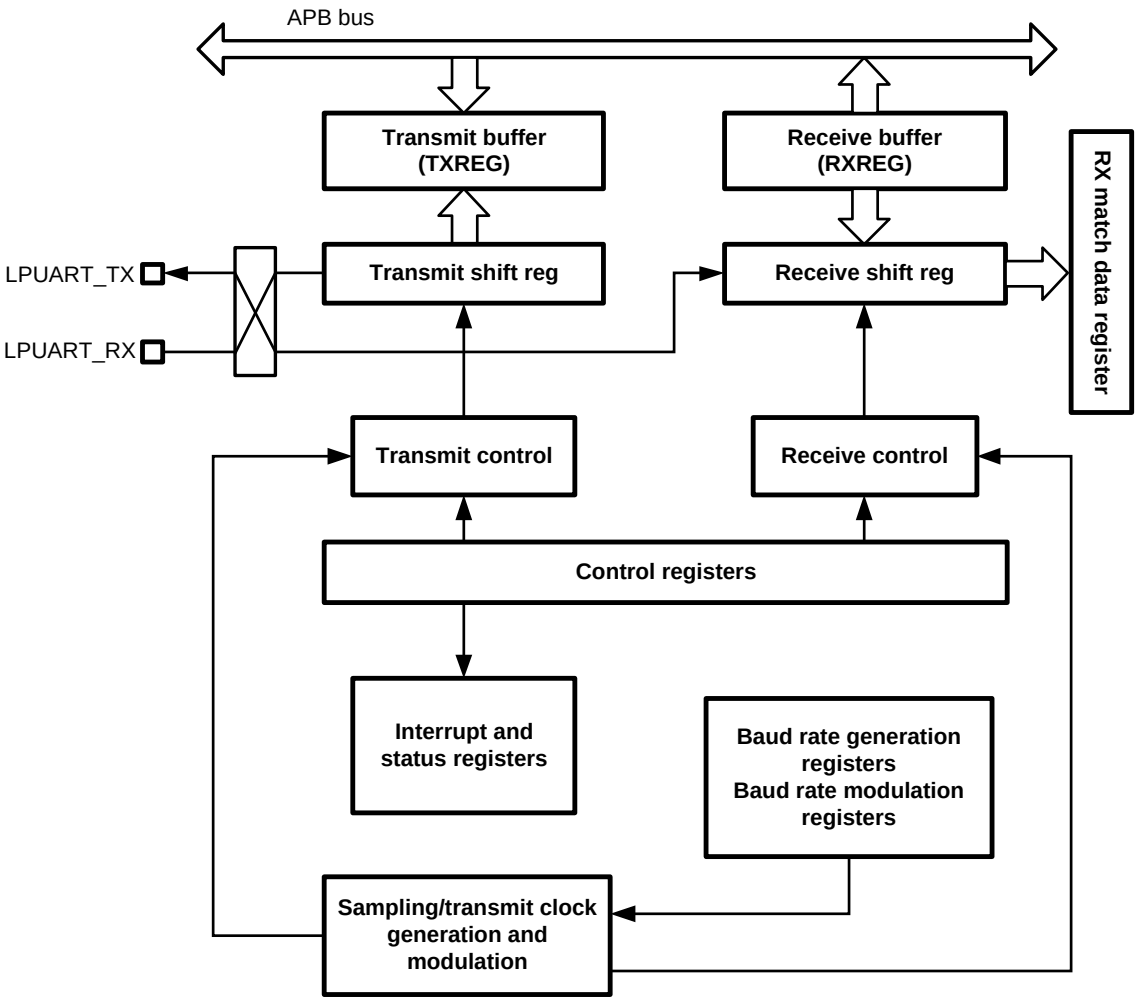


Figure 23-1 LPUART Block Diagram

23.3 Pin Definition

The LPUART module uses 2 pins to communicate with external devices, and the receiving and sending signals of each UART may be mapped to different GPIOs.

Pin		UARTx	Symbol	Description
PA13	PA2	LPUART0	LPUART0_RX	Data Receive
PA14	PA3		LPUART0_TX	Data Transmit
PC2	PB13	LPUART1	LPUART1_RX	Data Receive
PC3	PB14		LPUART1_TX	Data Transmit
PA11	PB4	LPUART2	LPUART2_RX	Data Receive
PA12	PB5		LPUART2_TX	Data Transmit

Table 23-1 LPUART Pin Correspondence Table

When the LPUART function is mapped to multiple pins at the same time:

- PA2 and PA13 are configured as digital peripheral functions at the same time
  - Only the RX signal on PA13 will be input into the module
- PC2 and PB13 are configured as digital peripheral functions at the same time
  - Only the RX signal on PC2 will be input into the module
- PA11 and PB4 are configured as digital peripheral functions at the same time
  - Only the RX signal on PB4 will be input into the module
- When the LPUART transmission function is mapped to multiple GPIOs at the same time, these pins will send data at the same time

23.4 Clock and Reset

LPUART uses a clock independent of APBCLK for data transmission and reception, and relevant registers need to be configured in the CMU module before work.

LPUART can use XTLF or RCLF to work. Due to the low accuracy of RCLF, it is recommended to calibrate the clock before using RCLF for LPUART communication, and calibrate the RCLF to within +/-1%.

In ACTIVE mode, LPUART can also use RCHF to work, and the clock accuracy will be higher than RCLF at this time to obtain better timing fault tolerance. When working with RCHF, the prescaler circuit prescales the RCHF to obtain a clock frequency close to 32768Hz. For example, when the RCHF is 8M/16M/24M, the prescaler frequency division coefficient should be 244/488/732 respectively.

The LPUART working clock structure is shown in the figure below. This part of the functions and

registers are implemented in the CMU module. The bus clock is enabled by setting the LPUARTx_PCE register to operate the register, the working clock source is selected by configuring LPUARTxCKS, and the working clock is enabled by setting LPUARTxCKE.

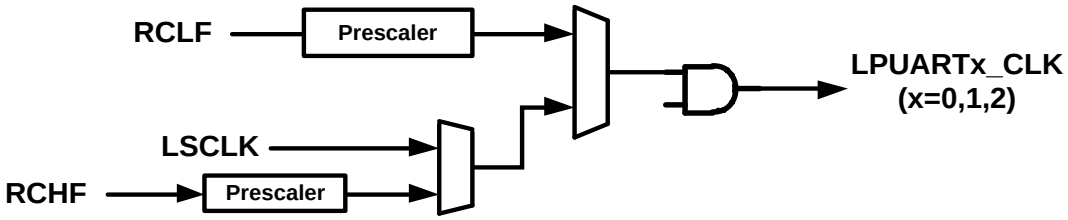


Figure 23-2 LPUART Clock

The RCLF frequency is 614.4Khz. After the internal pre-frequency division of the CMU module, the working clock output to the LPUART is 38.4Khz, which is 4 times the frequency of 9600. Through this clock, more accurate LPUART data transmission and reception can be realized, especially to ensure the stability of the output baud length.

23.5 Character Description

The basic timing of LPUART characters is similar to a standard UART. Each character frame contains at least 1bit START and at least 1bit STOP bits, data lengths can be configured to be 6-9bits, and parity bits can be selected.

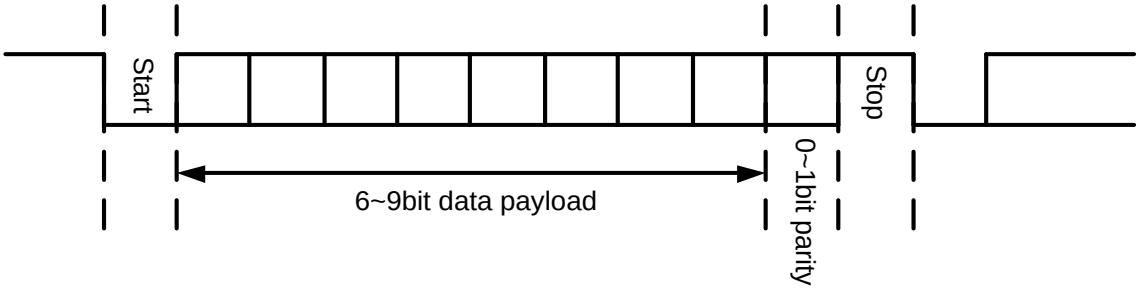


Figure 23-3 Character Format

LPUART supports multiple frame formats controlled by the UARTxCSR.PDSEL register and UARTxCSR.PARITY register, as shown in the following table:

PDSEL	PARITY	Frame Format[1]
00	00	[Start | 7 bits data | Stop]
	01, 10	[Start | 7 bits data | Parity | Stop]

01	00	[Start | 8 bits data | Stop]
	01, 10	[Start | 8 bits data | Parity | Stop]
10	00	[Start | 9 bits data | Stop]
	01, 10	[Start | 9 bits data | Parity | Stop]
11	00	[Start | 6 bits data | Stop]
	01, 10	[Start | 6 bits data | Parity | Stop]

Table 23-2 LPUART Frame Format

[1]: The Stop bit can be either 1bit or 2bit, depending on the STOPCFG register.

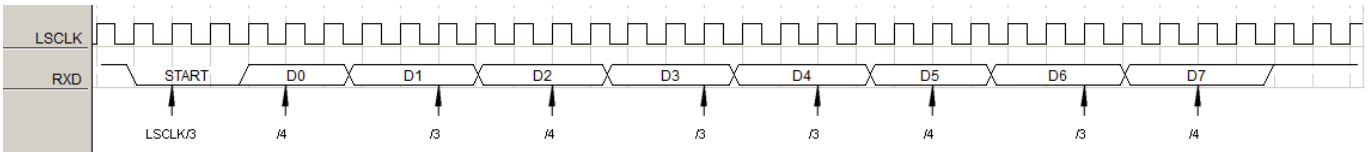
Note: THE PDSEL register is used to configure the data length of the frame. The communication frame length is [start bit + data bit + parity bit + stop bit].

23.6 Functional Description

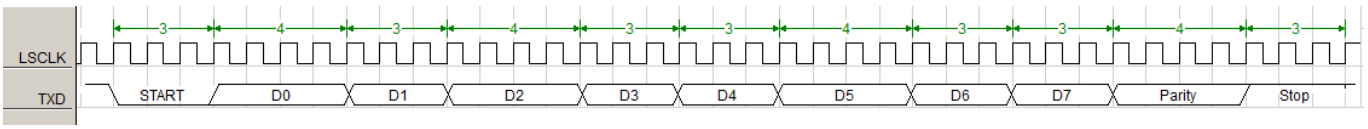
23.6.1 Bit Receive Sampling and Transmission

When the working clock is selected as LSCLK or RCHF frequency division, the frequency is only about 32Khz. At this time, the standard serial port cannot support 9600bps communication, so bit modulation design needs to be introduced.

Since the LPUART working clock is not an integer multiple of the baud rate, a fixed frequency division factor will introduce cumulative errors. When receiving, use 3 and 4 frequency divisions to receive alternately to ensure that sampling is done in the middle of each bit, and each bit is sampled. once. Whether each bit is divided by 3 or 4 is controlled by the MCTL register. For example:



Similar to LPUART reception, the working clock of LPUART is not an integer multiple of the baud rate. The use of a fixed frequency division factor will also introduce cumulative errors. When sending, the frequency division of 3 and 4 is used to transmit alternately. Whether each bit is divided by 3 or Frequency divided by 4 is controlled by the MCTL register. For example:



The software needs to reasonably configure the modulation control register MCTL according to the communication baud rate. When the working clock is 32768Hz, the recommended configuration parameter table is as follows:

Baud	MCTL												
	Bit0(start)	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12
9600	0	1	0	0	1	0	1	0	1	0	1	0	0
4800	1	1	0	1	1	1	1	1	0	1	1	1	1
2400	1	1	0	1	1	0	1	1	0	1	1	0	1
1200	0	1	0	0	1	0	0	1	0	0	1	0	0
600	0	1	1	0	1	0	1	1	0	1	1	0	1
300	0	1	0	0	0	0	1	0	0	0	0	1	0

Table 23-3 LPUART Bit Modulation Factor

When the working clock is selected as RCLF frequency division, the calibrated RCLF frequency is about 614.4Khz, which corresponds to 64 times of 9600. After pre-frequency division, 38.4Khz working clock is obtained. At this time, LPUART communication does not require data bit modulation. The software should clear MCTL_EN to zero.

23.6.2 Data Reception Procedure

- Configure the BAUD register to determine the baud rate
- Choose appropriate modulation parameters according to the baud rate, and configure the MCTL and MCTL_EN registers
- Configure the CSR register, select the frame format, polarity, interrupt parameters, etc.
- Configure the RXEN register to open the receive enable
- Waiting for interrupt event

23.6.3 Data Transmit Procedure

- Configure the BAUD register to determine the baud rate
- Choose appropriate modulation parameters according to the baud rate, and configure the MCTL and MCTL_EN registers
- Configure the CSR register, select the frame format, polarity, interrupt parameters, etc.
- Configure the TXEN register to open the transmit enable
- Waiting for interrupt event

23.6.4 Use DMA for Data Receiving and Transmitting

When the DMA is enabled, the LPUART will automatically generate the corresponding DMA request when the TX buffer is empty or the RX buffer is full. The application needs to configure DMA channel connections, connect specific channels to LPUART peripherals, set the RAM pointer, and enable DMA channels. After that, DMA will automatically respond to the LPUART request and complete the data transfer between RAM and LPUART.

Application example: LPUART1 receive using DMA

- Configure the DMA channel X as LPUART1_RX
- Set corresponding channel parameters: RAM pointer, address increment or decrement, channel priority, transmission length, interrupt, etc
- Enable corresponding DMA channels
- Configure the LPUART1 parameters
- Enable LPUART1by writing RXEN=1, and wait for incoming data
- LPUART1 automatically generates DMA requests upon arrival of data
- DMA responds to the request, reads the LPUART1 receive buffer, and writes to the specified RAM address

23.6.5 Data Receiving and Wakeup in Sleep Mode

LPUART supports data reception and wake up the chip in Sleep and DeepSleep modes. At this time, the chip power consumption is extremely low, and it keeps monitoring the RXD pin until a specific event arrives and wakes up the chip to exit the sleep mode.

- Configure the BAUD register to determine the baud rate
- Choose appropriate modulation parameters according to the baud rate, and configure the MCTL and MCTL_EN registers
- Configure the CSR register, select the frame format and polarity, and select the wake-up event to be START bit through RXEV, one frame receiving completion, one frame data matching or RXD falling edge detection, WKBYTE_CFG is set to 1
- Configure the RXEN register to open the receive enable
- The software enters Sleep/DeepSleep

23.6.6 Use LPRUN for UART Receiving and Transmitting

Through LPUART and DMA, the software can automatically send and receive a certain amount of LPUART data in LPRUN mode without CPU intervention, while ensuring that the power consumption of the whole chip is less than 10uA under typical conditions.

- Configure the BAUD register to determine the baud rate
- Choose appropriate modulation parameters according to the baud rate, and configure the MCTL and MCTL_EN registers
- Configure the CSR register, select the frame format, polarity, interrupt parameters, etc.
- Configure DMA channel control register, select LPUART to send and receive
- If you need to send data, write the data to be sent to the specified location in RAM
- Configure DMA data receiving and sending length and RAM pointer
- Select LSCLK as the main clock of the system
- The software enters LPRUN
- Configure TXEN and RXEN registers to enable sending and receiving

23.6.7 Transmission Completion Interrupt in DMA Mode

When LPUART transmits data through DMA, DMA will generate a DMA channel interrupt after the specified length of data transfer is completed. But when the channel interrupt is generated, the last frame of data has just been written into the LPUART transmit buffer and has not been sent out yet.

By configuring the DMATXIFCFG register, it is possible to generate a transmission completion interrupt (buffer empty or shift register empty) when the DMA transmission is completed and the last frame of data transmission is completed, so that after all the data is sent out, the CPU will be interrupted. Application scenarios.

The software workflow is explained as follows:

- Configure DMA channel as LPUART transmission
- Turn off DMA channel interrupt enable
- Set the LPUART TXBE_IE or TXSE_IE register to allow interrupt generation
- Set the DMATXIFCFG register to allow only the last frame of data to generate an interrupt output
- Prepare data to be sent and enable DMA
- LPUART transmits continuously until the last frame, no TXBE or TXSE interrupt will be generated during transmission
- After the last frame is sent, LPUART generates TXBE or TXSE interrupt

The following table assumes that LPUART sends N frames through DMA:

TXBE_IE TXSE_IE	DMATXIFCFG	Frame No.	TXBE TXSE	LPUART interrupt
0	x	1~N	After each frame is sent, set	Inactive
1	0	1~N	After each frame is sent, set	TXBE active, TXSE active in the last frame
	1	1~N-1	After each frame is sent, set	Inactive
		N	After each frame is sent, set	active

Table 23-4 LPUART DMA Interrupt Description

23.7 Register

Offset	Name	Symbol
LPUART0 register(Base address: 0x40014000)		
0x00	LPUART0 Control Status Register	LPUART0_CSR
0x04	LPUART0 Interrupt Enable Register	LPUART0_IER
0x08	LPUART0 Interrupt Status Register	LPUART0_ISR
0x0C	LPUART0 Baud Rate Modulation Register	LPUART0_BMR
0x10	LPUART0 Receive Buffer Register	LPUART0_RXBUF
0x14	LPUART0 Transmit Buffer Register	LPUART0_TXBUF
0x18	LPUART0 Data Matching Register	LPUART0_DMR
LPUART1 register(Base address: 0x40014400)		
0x00	LPUART1 Control Status Register	LPUART1_CSR
0x04	LPUART1 Interrupt Enable Register	LPUART1_IER
0x08	LPUART1 Interrupt Status Register	LPUART1_ISR
0x0C	LPUART1 Baud Rate Modulation Register	LPUART1_BMR
0x10	LPUART1 Receive Data Register	LPUART1_RXBUF
0x14	LPUART1 Transmit Data Register	LPUART1_TXBUF
0x18	LPUART1 Data Matching Register	LPUART1_DMR
LPUART2 register(Base address: 0x40015000)		
0x00	LPUART2 Control Status Register	LPUART2_CSR
0x04	LPUART2 Interrupt Enable Register	LPUART2_IER
0x08	LPUART2 Interrupt Status Register	LPUART2_ISR
0x0C	LPUART2 Baud Rate Modulation Register	LPUART2_BMR
0x10	LPUART2 Receive Buffer Register	LPUART2_RXBUF
0x14	LPUART2 Transmit Buffer Register	LPUART2_TXBUF
0x18	LPUART2 Data Matching Register	LPUART2_DMR

23.7.1 LPUARTx Control Status Register (LPUARTx_CSR)

NAME	**LPUARTx _CSR(x=0,1,2)**							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							BUSY
access	U-0							R-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				WKBYTE_CFG	-	RXEV	
access	U-0				R/W-0	U-0	R/W-00	
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

name	-				IOSWAP	DMATXIFCFG	BITORD	STOPCFG
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PDSEL		PARITY		RXPOL	TXPOL	RXEN	TXEN
access	R/W-00		R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:25	-	RFU: **Reserved, read as 0**
24	**BUSY**	LPUART communication flag, read-only (Busy) 1: UART is communicating. 0: UART idle
23:20	-	RFU: **Reserved, read as 0**
19	**WKBYTE_CFG**	Wakeup Byte Config 1: A wake-up interrupt is triggered when 1 byte is received and the parity and STOP bits are correct 0: After receiving 1 byte, do not check the check bit and STOP bit, trigger the wake-up interrupt directly
18	-	RFU: **Reserved, read as 0**
17:16	**RXEV**	The wake interrupt event configuration is used to control under which events the wake interrupt is provided to the CPU (Receive Wakeup Event) 00: START bit detects wake up 01: The 1byte data is received 10: Received data match successfully 11: RXD falling edge detection
15:12	-	RFU: **Reserved, read as 0**
11	**IOSWAP**	RX and TX pin swapping (IO swapping) 0: Default pin sequence (consistent with package drawing) 1: Swap the pin sequence
10	**DMATXIFCFG**	DMA transmit completion interrupt enablement is only valid if LPUART is sending through DMA (DMA Transmit Interrupt Config) 1: In the case of IE=1, interrupt is generated after the last frame is sent by DMA; Interrupt is masked before the last frame has been sent 0: It is up to IE to decide whether to mask interrupt or not
9	**BITORD**	Bit order in which data is sent/received (Bit Order) 0: LSB first 1: MSB first
8	**STOPCFG**	Stop bit configuration, only valid for transmitting 0: 1 Stop bit 1: 2 Stop bit

bit	name	functional description
7:6	**PDSEL**	Select the data length of each frame; This register is valid for both sending and receiving data. (Payload Data length Select) 00: 7 bits 01: 8 bits 10: 9 bits 11: 6 bits
5:4	**PARITY**	Parity bit configuration; This register is valid for both transmitting and receiving data (Parity) 00: None 01: Odd 10: Parity 11: RFU
3	**RXPOL**	Receive Polarity Configuration 0: Standard 1: Inverted
2	**TXPOL**	Transmit Polarity Configuration 0: Standard 1: Inverted
1	**RXEN**	Receive enable 1: Enable 0: Disable
0	**TXEN**	Transmit enable 1: Enable 0: Disable

23.7.2 LPUARTx Interrupt Enable Register (LPUARTx_IER)

NAME	**LPUARTx _IER(x=0,1,2)**							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			RXEV_IE	-	RXERR_IE	-	RXBF_IE
access	U-0			R/W-0	U-0	R/W-0	U-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						TXBE_IE	TXSE_IE
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:13	**-**	RFU: **Reserved, read as 0**
12	**RXEV_IE**	Receive Event Interrupt Enable, 1 effective
11	**-**	RFU: **Reserved, read as 0**
10	**RXERR_IE**	Receive Error Interrupt Enable, 1 effective
9	**-**	RFU: **Reserved, read as 0**
8	**RXBF_IE**	Receive Buffer Full Interrupt Enable, 1 effective
7:2	**-**	RFU: **Reserved, read as 0**
1	**TXBE_IE**	Transmit Buffer Empty Interrupt Enable, 1 effective
0	**TXSE_IE**	Transmit Shift register Interrupt Enable, 1 effective

23.7.3 LPUARTx Interrupt Status Register (LPUARTx_ISR)

NAME	LPUARTx _ISR(x=0,1,2)							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							RXEVF
access	U-0							R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				TXOV	PERR	FERR	OERR
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							RXBF
access	U-0							R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						TXBE	TXSE
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:25	**-**	RFU: **Reserved, read as 0**
24	**RXEVF**	Receive Event Interrupt Flag. Set by hardware, write 1 to clear. The interrupt flag trigger source is configured with the LPUxCR.RXEV register.
23:20	**-**	RFU: **Reserved, read as 0**
19	**TXOV**	Transmit Overflow Error. Set by hardware, write 1 to clear. TXOV is triggered when software writes new data into TX buffer when TX buffer is full.
18	**PERR**	Parity Error Interrupt Flag. Set by hardware, write 1 to clear.
17	**FERR**	Frame Error Interrupt Flag. Set by hardware, write 1 to clear.

bit	name	functional description
16	**OERR**	Receive Buffer Overflow Error Interrupt Flag. Set by hardware, write 1 to clear. OERR is triggered when new data has been received when RX buffer is full.
15:9	**-**	RFU: **Reserved, read as 0**
8	**RXBF**	Receive Buffer Full Interrupt Flag. Set by hardware, write 1 to clear.
7:2	**-**	RFU: **Reserved, read as 0**
1	**TXBE**	Transmit Buffer Empty Interrupt Flag. Set by hardware, cleared by writing data into TX buffer
0	**TXSE**	Transmit Shift register Empty Interrupt Flag. Set by hardware, cleared by writing 1 or byte is moved into transmit shift register

23.7.4 LPUARTx Baud Rate Modulation Register (LPUARTx_BMR)

NAME	**LPUARTx _BMR(x=0,1,2)**							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	MCTL_EN	-			MCTL[11:8]			
access	R/W-1	U-0			R/W-0000			
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	MCTL[7:0]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					BAUD		
access	U-0					R/W-000		

bit	name	functional description
31	**MCTL_EN**	Baud rate modulation enable (modulation control enable) 0: Turn off baud rate modulation 1: Enable baud rate modulation When LPUART working clock is RCLF, baud rate modulation is not needed, and MCTL_EN should be cleared by software.
30:28	**-**	RFU: **Reserved, read as 0**
27:16	**MCTL**	Bit width modulation control signal of each bit of LPUART
15:3	**-**	RFU: **Reserved, read as 0**

bit	name	functional description
2:0	**BAUD**	Baud rate control (bps) 000: 9600 001: 4800 010: 2400 011: 1200 100: 600 101/110/111: 300

23.7.5 LPUARTx Receive Buffer Register (LPUARTx_RXBUF)

NAME	LPUARTx _RXBUF(x=0,1,2)							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							RXBUF[8]
access	U-0							R-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RXBUF[7:0]							
access	R-0000 0000							

bit	name	functional description
31:9	**-**	RFU: **Reserved, read as 0**
8:0	**RXBUF**	Receive Buffer

23.7.6 LPUARTx Transmit Buffer Register (LPUARTx_TXBUF)

NAME	LPUARTx _TXBUF(x=0,1)							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

name	-							TXBUF[8]
access	U-0							R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TXBUF[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:9	**-**	RFU: **Reserved, read as 0**
8:0	**TXBUF**	Transmit Buffer

23.7.7 LPUARTxData Matching Register (LPUARTx_TXBUF)

NAME	LPUARTx _DMR(x=0,1)							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							MATD[8]
access	U-0							R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	MATD[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:9	**-**	RFU: **Reserved, read as 0**
8:0	**MATD**	Matching data register. If RXEV=10, the RXEVF interrupt is triggered when the first byte received is matched with MATD, which can be used to wake up the MCU by certain data receiving in the sleep mode.

24 Serial Peripheral Interface (SPI)

24.1 Introduction

The serial peripheral interface (Serial Peripheral Interface, SPI) is a serial synchronous communication method for external devices to exchange data through 4-wires. The chip provides 3 SPI interface modules, which can be configured as a master device or a slave device to realize SPI communication with the outside.

Features:

- Full-duplex 4-wire serial synchronous transmission and reception (SCLK, MOSI, MISO, SSN)
- MISO and MOSI can exchange pin order
- Half-duplex 4-wire serial synchronous transmission and reception (SCLK, SDATA, SSN, DCN), used for TFT screen driver
- 3 independent channels
- Master-slave mode
- Programmable clock polarity and phase
- Programmable bit rate
- Programmable data word length (8/16/24/32bits)
- The maximum baud rate is FAPBCLK/2
- End of transmission interrupt flag
- Write conflict error flag
- Main mode error detection, protection and interrupt flag
- Support DMA

24.2 Block Diagram

The following figure shows the structure diagram of the SPI module.

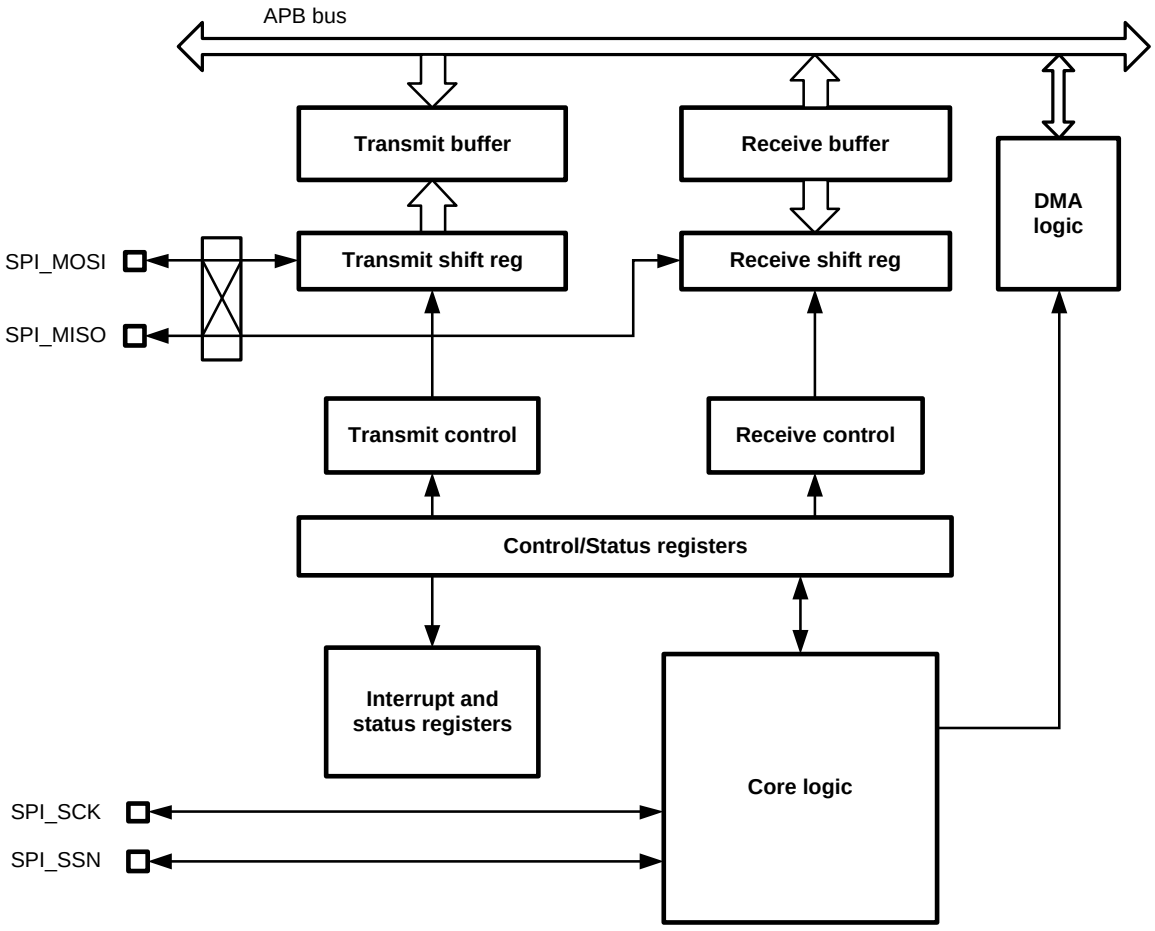


Figure 24-1 Block Diagram of SPI

24.3 Pin Definition

The SPI module uses 4 pins to communicate with external devices. In full-duplex and half-duplex modes, the function definitions of these four pins are different, as shown in the following table:

Pin	SPIx	Full Duplex	Function	Half Duplex	Function
PB8/PE0	SPI0	SSN	Chip selection signal	SSN	Chip selection signal
PB9/PE1		SCLK	Clock	SCLK	Clock
PB10/PE2		MISO	Master Input Slave Output	DCN	Command/Data Flag
PB11/PE3		MOSI	Master Output Slave Input	SDATA	Data
PA8/PD2	SPI1	SSN	Chip selection signal	SSN	Chip selection signal
PA9/PD3		SCLK	Clock	SCLK	Clock
PB0/PD4		MISO	Master Input Slave Output	DCN	Command/Data Flag
PB1/PD5		MOSI	Master Output Slave Input	SDATA	Data
PB6/PC7	SPI2	SSN	Chip selection signal	SSN	Chip selection signal
PB7/PC8		SCLK	Clock	SCLK	Clock
PB15/PC9		MISO	Master Input Slave Output	DCN	Command/Data Flag
PD12/PC10		MOSI	Master Output Slave Input	SDATA	Data

Table 24-1 SPI Pin Correspondence Table

24.4 Clock and Reset

SPI module working clock and bus register clock are both APBCLK.

Before using the SPI module, you must first clear the SPIxRST register in the RMU to release the reset, and set the SPIx_PCE in the CMU to turn on the clock.

24.5 Interface Timing

In order to be compatible with different SPI protocols, the timing of the SPI serial clock can be set by the clock phase selector bit (CPHA) and the clock polarity selector bit (CPOL) to produce four different combinations. To ensure correct data transfer, the timing configuration of the master and slave devices

must be the same.

When in slave mode or when the SPI enable bit (SPE) is 0, there is no serial clock output on the SCK pin.

24.5.1 CPHA=0

If CPHA=0, the SPI module samples data on the first edge of the serial clock, i.e.

If CPOL=1, SCK stays high at bus IDLE, the SPI samples data on the falling edge of the serial clock and sends data on the rising edge of the serial clock.

If CPOL=0, SCK stays low at bus IDLE, the SPI samples data on the rising edge of the serial clock and sends data on the falling edge of the serial clock.

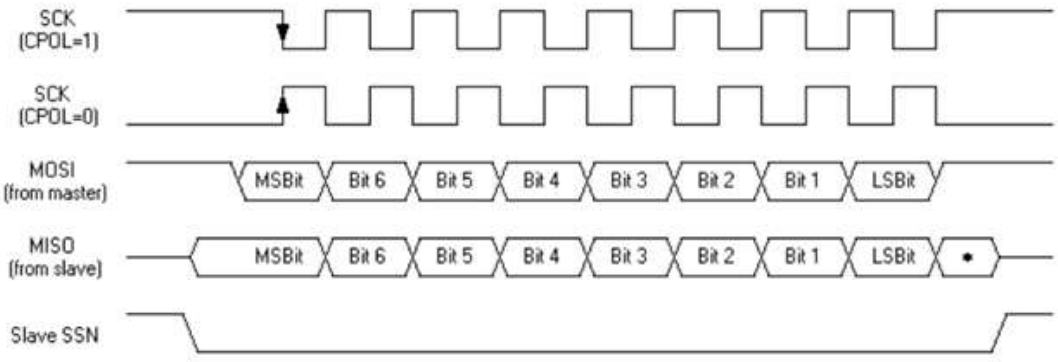


Figure 24-2 SPI Data/Clock Timing Diagram (CPHA=0)

24.5.2 CPHA=1

With CPHA=1, the SPI module samples data on the second edge of the serial clock, i.e.

If CPOL=1, SCK stays high at bus IDLE, samples data on the rising edge of the serial clock and sends data on the falling edge of the serial clock.

If CPOL=0, SCK stays low at bus IDLE, samples data on the falling edge of the serial clock, and sends data on the rising edge of the serial clock.

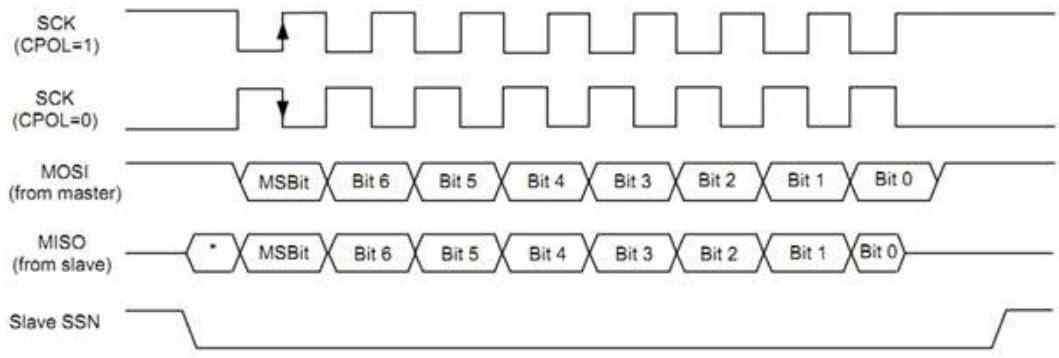


Figure 24-3 SPI Data/Clock Timing Diagram (CPHA=1)

24.5.3 4-wire Half-Duplex Mode (Master)

The 4-wire half-duplex mode can support interactive communication with dot matrix LCD or TFT screen. In this mode, the DCN signal is used to distinguish whether it is a command frame or a data frame that is currently being sent. Two-way data is sent and received through the SDATA (MOSI) pin, and the data direction is switched automatically by the hardware. The SPI of FM33LG0 only supports 4-wire half-duplex master mode, not slave mode.

All communications are initiated by the host, the host first sends the command frame, and then the data frame transmission. Command frame and data frame are distinguished by DCN signal line. The host can write data to or read data from the slave through the 4-wire half-duplex interface.

4-wire half-duplex write operation

The software indicates that the current host wants to initiate a write operation by clearing the HD_RW register.

Before the host initiates a write operation, it first sends a write command frame. After the write command frame is sent, if the sending buffer is empty, the hardware will pull up SSN and stop SCLK sending; if the sending buffer has been written with new data, the hardware will continuously send subsequent data frames.

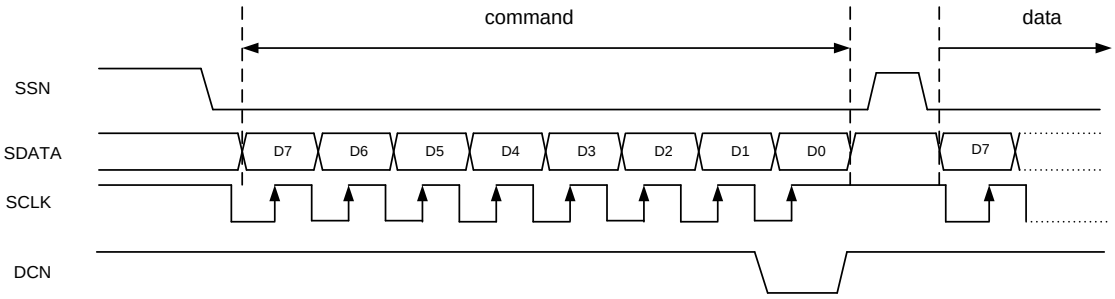


Figure 24-4 Wire Half-Duplex Write Operation

DCN samples the judgment on the rising edge of the 8th clock. If it is 0, it means that the current frame is a command frame. Before sending the command frame, the software needs to write 0 to the DCN_TX register. After the command frame is sent, the hardware automatically sets the DCN register.

4-wire half-duplex read operation

The software indicates that the current host wants to initiate a read operation by setting the HD_RW register.

The 4-wire half-duplex read operation supports 8-bit, 16-bit, 24-bit and 32-bit reads. When the host initiates a read operation, it first sends a read command frame. After the read command frame is sent, a dummy cycle can be sent according to the register configuration. During the dummy cycle, the SCLK

clock is sent normally, but the host does not drive SDATA or accept SDATA input.

After completing the command frame and dummy cycle (optional), the 4-wire half-duplex SPI automatically enters the receiving state, the SDATA signal is driven by the slave, and the data frame received by the host will be written into the receive buffer. After each data frame is received, the RXBF interrupt flag register will be set. The software should read the data in the receive buffer in time. If the receive buffer and the receive shift register are both full, the hardware will stop SCLK sending and read data from the slave until the software or DMA reads the receive buffer.

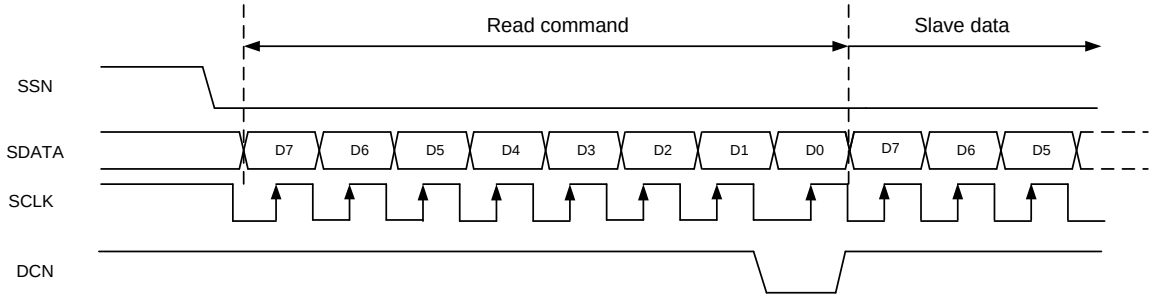


Figure 24-5 Wire Half-Duplex Read Operation (No Dummy Cycle)

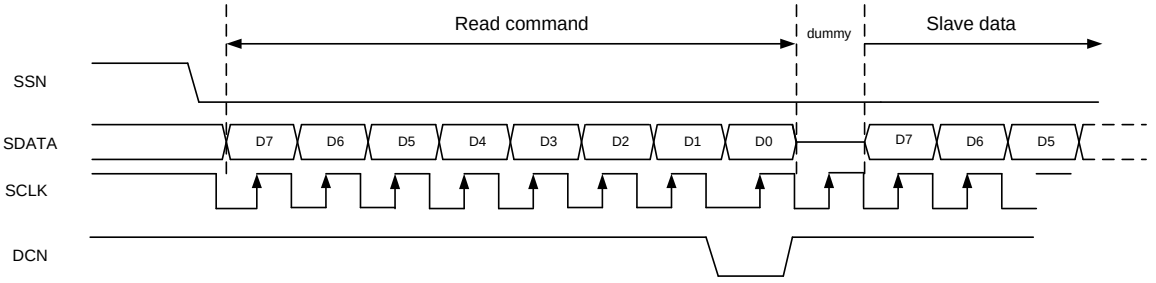


Figure 24-6 Wire Half-Duplex Read Operation (With Dummy Cycle)

24.6 Functional Description

24.6.1 I/O Configuration

Master output, slave input (MOSI)

The MOSI pin is the output of the master device and the input of the slave device, and is used for serial data transmission from the master device to the slave device. When SPI is configured as a master device, this pin is an output, when SPI is configured as a slave device, this pin is an input. The MSB comes first during data transmission.

Master input, slave output (MISO)

The master input slave output (MISO) pin is the output of the slave device and the input of the master device, and is used for serial data transmission from the device to the master device. When SPI is configured as a master device, this pin is an input; when SPI is configured as a slave device, this pin is

an output. The MSB comes first during data transmission.

Serial clock (SCK)

The serial clock (SCK) pin is the output of the master device and the input of the slave device. It is used to synchronize the serial data transmission between the master device and the slave device on the MOSI and MISO lines. When SPI is configured as a master device, this pin outputs the clock. When SPI is configured as a slave device, this pin is an input.

Select from (SSN)

The slave selection (SSN) pin is used to control the selection of the slave device, usually the master SSN output is directly connected to the slave SSN input. If there is only one slave on the bus, the slave SSN can also be grounded to make the slave always enabled. At this time, the SSN output of the master can be used as GPIO.

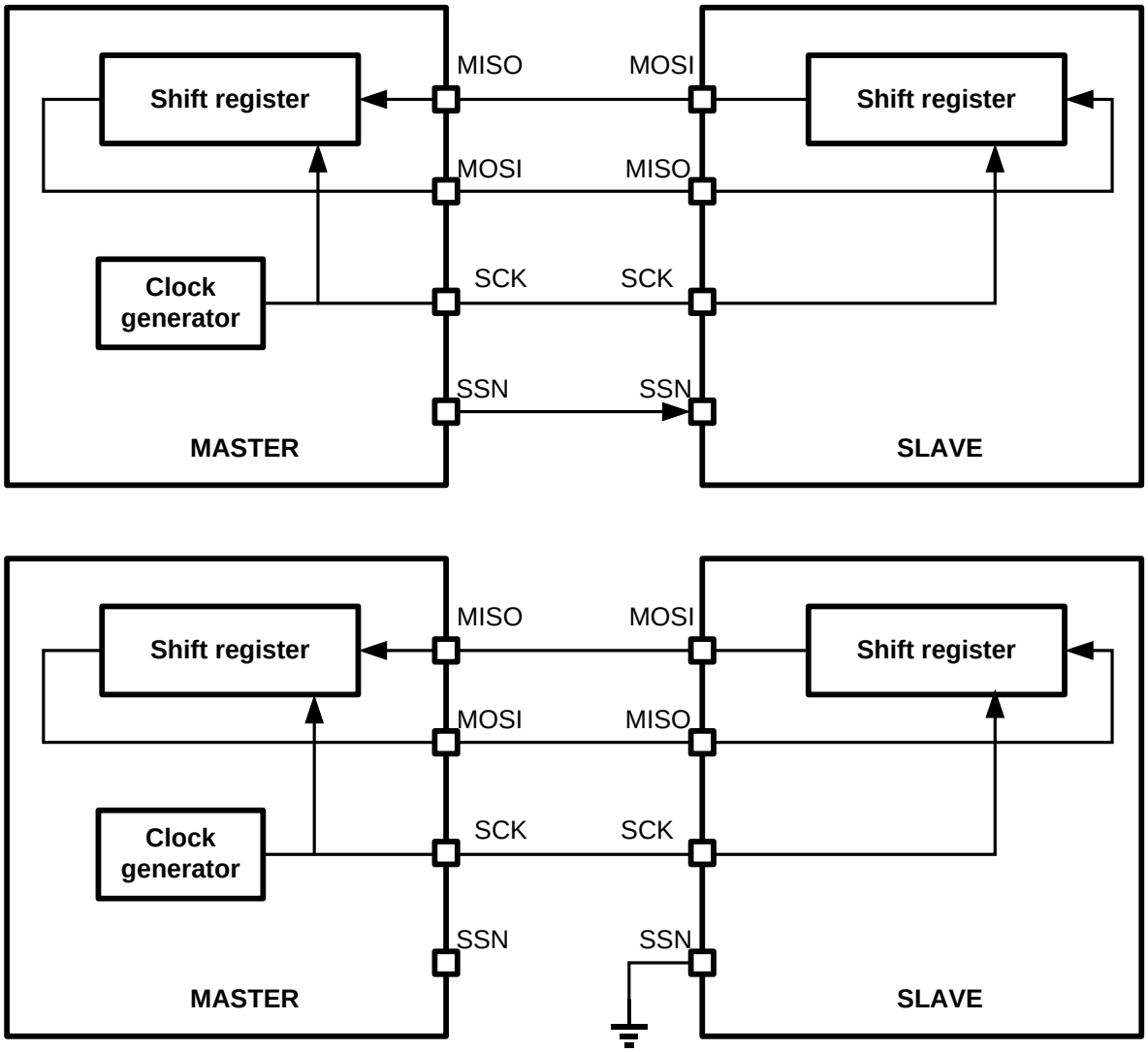


Figure 24-7 SPI Master/SPI Slave Interconnection

24.6.2 Full-Duplex Data Communication

The SPI module defaults to full-duplex communication. If you need to achieve continuous and uninterrupted data communication, the software needs to ensure that TXBUF is not empty. Even if the software only uses SPI for data reception, due to the full-duplex nature of SPI, the software still needs to write to TXBUF. At this time, invalid data is written. You can write all 0s or all Fs according to the MOSI invalid state configuration.

Send buffer

The software or DMA writes the data to be sent into the sending buffer (TXBUF register). When the sending starts, the hardware copies the data from the sending buffer to the shift register and starts sending. After the data is transferred from the transmit buffer to the shift register, the transmit buffer empty flag (TXBE) is set, indicating that new data can be written to TXBUF; if the TXIE register is set, an interrupt is generated. By writing data to TXBUF, the TXBE register can be cleared.

If the new data is written into the transmit buffer before the shift register shift is completed, continuous data transmission can be guaranteed. Writing TXBUF when TXBE is 0 will cause a data conflict, see 24.6.6 Data conflict.

Receive buffer

When the SPI completes a frame of data reception, the received data will be copied from the shift register to the receive buffer (RXBUF register), and the RXBF flag is set, indicating that there is data to be processed in RXBUF. If the RXIE register is set, an interrupt is generated. The RXBF flag can be cleared by reading RXBUF.

Reading RXBUF when RXBF is not set will return the last received data; if the application does not process RXBF in time, and the new data is received when RXBF is set, a data conflict will occur, see 24.6.6 Data conflict.

BUSY logo

When the SPI is sending and receiving data, the BUSY register is set. This register can be used to determine whether the last frame of data has been transmitted in some scenarios. For example, TXBE just means that the data has been shifted and sent, but the actual sending is completed, and you need to wait for the BUSY flag to be cleared.

How to start SPI communication

In host mode, it is recommended to follow the steps below to start SPI communication:

- Application configuration SPI module
- Set SPIEN

- Write data to TXBUF, and the SPI module will automatically start sending SCK and send and receive data

In slave mode, it is recommended that the application complete the configuration and enable before the host starts to send SCK, and write the first frame of data to be sent into TXBUF, and wait for the host to send SCK to start communication.

How to end SPI communication

In host mode, it is recommended to follow the steps below to end SPI communication:

- Waiting for the RXBF and TXBE flags to be set, at this time there is still the last frame of data in the shift register being sent
- Query the BUSY flag, until BUSY is 0, the last frame of data is sent and received
- Close the SPI module, if necessary, read the last frame of received data

In slave mode, the application can close the SPI module after reading any frame of data, and the data that has been moved into the shift register before closing will be ignored.

24.6.3 TX-ONLY Mode

In some cases, the SPI communication is half-duplex. When the master only needs to transmit, the TX-ONLY mode can be used by setting the TXO register. Under TX-only the data received by the MISO will not be written to the RX Buffer, and accordingly the RXBF interrupt flag will not be set.

The TXO auto-clear function can be implemented by setting TXO_AC. In TX-ONLY mode, if the TX buffer is empty (TXBE is set) and the transmit shift register is empty, the TXO register is automatically cleared and the TX-ONLY state is exited.

24.6.4 RX-ONLY Mode

If the SPI master only needs to perform reception, it enters RX-ONLY mode by setting the RXO register. The SPI module can perform continuous data reception without software writing to the TX Buffer, and the MOSI will hold the IDLE level and TXBE interrupt register will not be set.

24.6.5 Master SSN Control

The SPI master supports hardware or software controlled SSN signal.

SSN is controlled by hardware when the SSNSEN register is cleared to zero; if the SSNM register is set, the SPI will pull SSN high after each data frame sent, and the SSN high time is configured by the WAIT register (several SCK clock cycles).

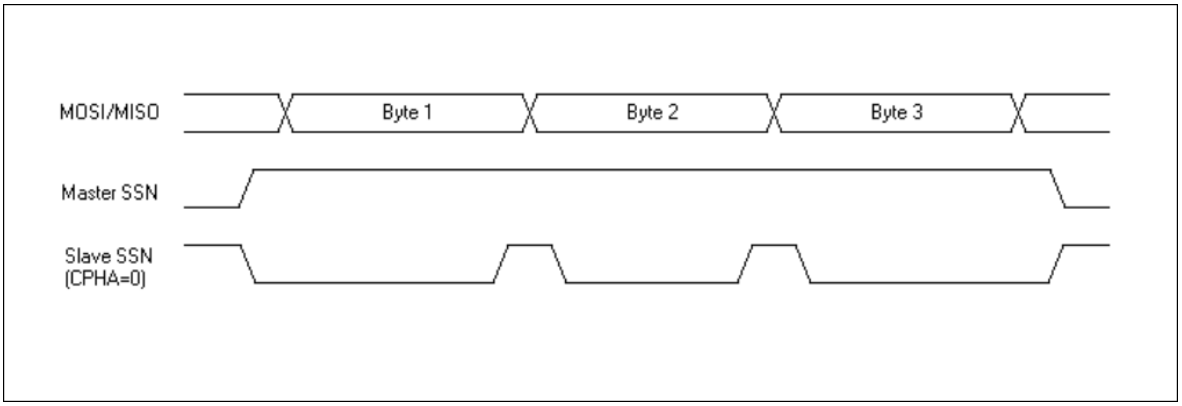


Figure 24-8 SPI SSN Timing Diagram (SSNM=1, CPHA=0)

If the SSNM register is reset, the SPI does not pull up the SSN after each frame of data sent, but goes directly to the next frame transmitting.

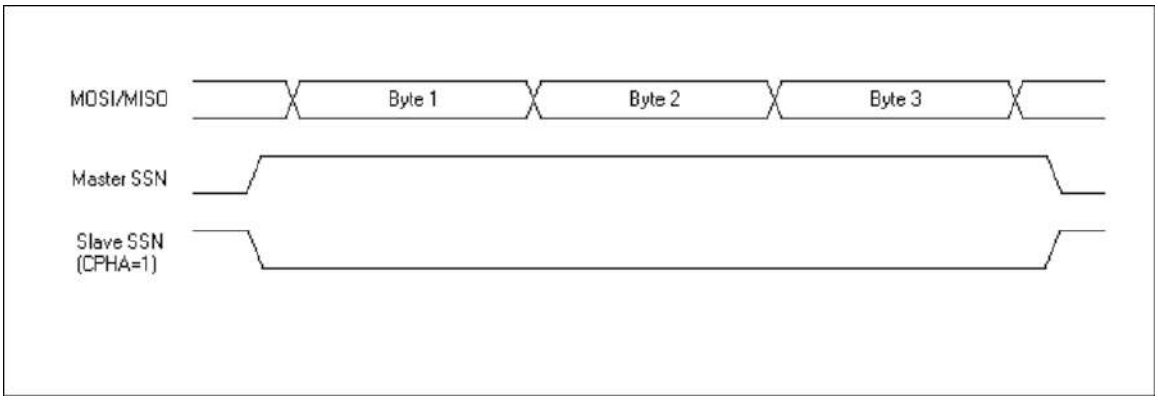


Figure 24-9 SPI SSN Timing Diagram (SSNM=0)

When the SSNSEN register is set, SSN is controlled by software. Software can directly manipulate the SSN level by writing the SPIx_CR2.SSN register bits.

24.6.6 Data Conflicts

When the TX Buffer data of the SPI has not been read into the shift register, or when the data in the RX Buffer of the SPI has not been read by software or DMA, a write operation to the TX Buffer or RX Buffer will generate a corresponding conflict error and the TXCOL/RXCOL bits will be set up, generating an interrupt. The write data that causes the conflict will be ignored. Data conflict errors are generated in both master and slave modes.

Write operations to the TX Buffer are initiated by the Master module inside the chip, including the CPU, DMA, etc. The write operation to RX Buffer is initiated by the external SPI device.

When a data conflict occurs, the original data in TX Buffer and RX Buffer is not refreshed and the newly written data is lost.

24.6.7 SPI Transceiver via DMA

When the SPI module is enabled, the SPI module automatically generates the corresponding DMA requests when both the transmit buffer is empty and the receive buffer is full. The application software needs to configure the DMA channel connection in advance, connects the specific channel to the SPI peripheral, set the RAM pointer, and enable the DMA channel. Thereafter the DMA will automatically respond to the SPI request and complete the data handling between RAM and SPI.

Note: *If DMA is used for transmitting and receiving full-duplex communication, the software should enable the DMA transmit channel first, and then the DMA receive channel; the opposite may cause the SPI to send an extra byte of dummy data.*

Using DMA for SPI reception

- Configure DMA channel 3 or 5 as SPI_RX
- Set RAM pointer address, address increment/decrement, channel priority, transfer length, interrupt settings, etc.
- Enabling the corresponding DMA channel
- Configure SPI module parameters
- Enable the SPI module and wait for data reception
- SPI automatically generates DMA request after receiving data
- DMA responds to the request, reads the SPI receive cache register, and writes to the specified RAM address
- When the specified length of DMA transfer is completed, the DMA will ignore subsequent requests and generate a transfer completion interrupt, and the software should handle the interrupt and shut down the SPI
- If data is received again before closing SPI, software can clear RXBUF by writing RXBFC

Using DMA for SPI transmission

The DMA transmit process is similar to the receive process described above, with the main difference being that the software cannot shut down the SPI immediately after the specified length of DMA transmission is completed, because the last frame of data is still being shifted and sent at this time, so

the software needs to query the BUSY flag until the end of shifting and sending, and then shut down the SPI module.

Data frame length and RAM data organization

The SPI transmission frame length can be configured to 8, 16, 24, or 32 bits.

When the data frame length is 8bit, DMA carries 1byte at a time, 4 carries fill one address of RAM, and small end storage is used within the word:.

RAM word: { data3, data2, data1, data0 }

When the length of the data frame is 16 bits, the DMA carries 2 bytes at a time, 2 carries fill one RAM address, and the word is stored in the small end: { data3, data2, data1, data0 }

RAM word: { data1, data0 }

When the data frame length is 24 bits, the DMA carries 1 word at a time, filling one RAM address with one carry, but the valid data occupies only the lower 24 bits of the RAM word: { data1, data0 }

RAM word: { 8'h0, data0 }

When the data frame length is 32 bits, the DMA carries 1 word at a time, filling one RAM address in one carry: { 8'h0, data0 }

RAM word: { data0 }

24.7 Register

Offset	Name	Symbol
SPI0 register (Base address: 0x40010400)		
0x00	SPI0 Control Register1	SPI0_CR1
0x04	SPI0 Control Register2	SPI0_CR2
0x08	SPI0 Control Register3	SPI0_CR3
0x0C	SPI0 Interrupt Enable Register	SPI0_IER
0x10	SPI0 Status Register	SPI0_ISR
0x14	SPI0 Transmit Buffer	SPI0_TXBUF
0x18	SPI0 Receive Buffer	SPI0_RXBUF
SPI1 register (Base address: 0x40010800)		
0x00	SPI1 Control Register1	SPI1_CR1
0x04	SPI1 Control Register2	SPI1_CR2
0x08	SPI1 Control Register3	SPI1_CR3
0x0C	SPI1 Interrupt Enable Register	SPI1_IER
0x10	SPI1 Status Register	SPI1_ISR
0x14	SPI1 Transmit Buffer	SPI1_TXBUF
0x18	SPI1 Receive Buffer	SPI1_RXBUF
SPI2 register (Base address: 0x40014800)		
0x00	SPI2 Control Register1	SPI2_CR1
0x04	SPI2 Control Register2	SPI2_CR2
0x08	SPI2 Control Register3	SPI2_CR3
0x0C	SPI2 Interrupt Enable Register	SPI2_IER
0x10	SPI2 Status Register	SPI2_ISR
0x14	SPI2 Transmit Buffer	SPI2_TXBUF
0x18	SPI2 Receive Buffer	SPI2_RXBUF

24.7.1 SPI Control Register 1 (SPIx_CR1)

NAME	**SPIx_CR1 (x=0,1,2)**							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				IOSWAP	MSPA	SSPA	MM
access	U-0				R/W-0	R/W-0	R/W-0	R/W-1
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	WAIT		BAUD			LSBF	CPOL	CPHA

access	R/W-00	R/W-000	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:11	--	RFU: **Reserved, read as 0**
11	**IOSWAP**	MOSI and MISO pin swapping (IO swapping) 0: Default pin order 1: Swapping pin order
10	**MSPA**	Master Sampling Position Adjustment, the Master's sampling position adjustment for MISO signals, used to compensate for PCB alignment delays when communicating at high speed 1: Sampling point delayed by half SCK cycle 0: No adjustment
9	**SSPA**	Slave Sending Position Adjustment, Slave MISO sending position adjustment 1: Sending half SCK cycle ahead 0: No adjustment
8	**MM**	Master/Slave mode selection 1: Master mode 0: Slave mode
7:6	**WAIT**	In Master mode, at least (1+WAIT) SCK cycle wait time is added after each frame is sent before transmitting the data of the next frame. If SSN is controlled by hardware and SSNM=1, hardware will pull up SSN automatically.
5:3	**BAUD**	Master mode baud rate configuration bits: 000: $f_{APBCLK}/2$ 001: $f_{APBCLK}/4$ 010: $f_{APBCLK}/8$ 011: $f_{APBCLK}/16$ 100: $f_{APBCLK}/32$ 101: $f_{APBCLK}/64$ 110: $f_{APBCLK}/128$ 111: $f_{APBCLK}/256$ These bits cannot be modified while communication is in progress.
2	**LSBF**	Frame format (LSB First) 0: MSB is sent first 1: LSB is sent first **Note:** The value of this bit cannot be changed while communication is in progress.
1	**CPHOL**	Clock Polarity Selection (Clock Polarity) 1: Serial clock stops at high level 0: Serial clock is stopped at low level **Note:** The value of this bit cannot be changed while communication is in progress

bit	name	functional description
		Note: The value of this bit cannot be changed when SSN is low
0	**CPHA**	Clock Phase Selection (Clock Phase) 1: The second clock edge is the first capture edge 0: The first clock edge is the first capture edge **Note:** The value of this bit cannot be changed while communication is in progress.

24.7.2 SPI Control Register 2 (SPIx_CR2)

NAME	SPIx_CR2 (x=0,1,2)							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DUMMY_EN	-			RXO	DLEN		HALFDUPLEX
access	R/W-0	U-0			R/W-0	R/W-00		R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	HD_RW	CMD8b	SSNM	TXO_AC	TXO	SSN	SSNSEN	SPIEN
access	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0

bit	name	functional description
31:16	**--**	RFU: **Reserved, read as 0**
15	**DUMMY_EN**	Whether to insert a dummy cycle in the read operation under 4-wire half-duplex protocol (Dummy cycle Enable) 0: no dummy cycle is inserted 1: insert a dummy cycle after the read command
14:12	**--**	RFU: **Reserved, read as 0**
11	**RXO**	RXONLY control bit, when this register is set, the SPI can receive continuously without software writing TXBUF (Receive Only mode) 1: Enable the single receive mode of Master 0: turn off the single receive mode (send/receive full duplex)
10:9	**DLEN**	Communication Data Length 00: 8bit 01:16bit

bit	name	functional description
		10: 24bit 11: 32bit
8	**HALFDUPLEX**	Communication mode selection (Half-Duplex mode) 0: Standard SPI mode, 4-wire full duplex 1: DCN mode, 4-wire half-duplex
7	**HD_RW**	Read/Write config for Half-Duplex mode 0: Master write to slave in 4-wire half-duplex protocol 1: Master read slave in 4-wire half-duplex protocol
6	**CMD8b**	Define the command frame length in half duplex mode (Command 8 bits) 1: Command frame length is fixed to 8 bits 0: Command frame length is defined by DLEN
5	**SSNM**	SSN control mode selection in Master mode (SSN mode) 1: Master pull SSN high after each frame is sent, the time to maintain high level is controlled by WAIT register 0: Master keeps SSN low after each frame is sent
4	**TXO_AC**	TXONLY auto-clear enable 1: TXONLY hardware auto-clear is valid, after the software enables TXO, wait for the hardware to clear after the transmission is finished 0: TXONLY hardware auto-clear enable is disabled
3	**TXO**	TXONLY control bit (Transmit Only mode enable) 1: Enable the Master's Transmit Only mode 0: Disable single transmit mode (send and receive full duplex)
2	**SSN**	In Master mode, if SSNSEN is 1, software can control SSN output level by this bit 1: SSN output high level 0: SSN output low level
1	**SSNSEN**	Master mode, software control SSN enable (SSN Software Enable) 1: SSN output in Master mode is controlled by software 0: SSN output in Master mode is automatically controlled by hardware
0	**SPIEN**	SPI enable 1: Enable SPI 0: Turn off SPI and clear the transmit/receive cache

24.7.3 SPI Control Register 3 (SPIx_CR3)

NAME	SPIx_CR3 (x=0,1,2)							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				TXBFC	RXBFC	MERRC	SERRC
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3	**TXBFC**	Transmit Buffer Clear, software write 1 to clear transmit buffer
2	**RXBFC**	Receive Buffer Clear, software write 1 to clear receive buffer
1	**MERRC**	Master Error Clear, software write 1 to clear MERR
0	**SERRC**	Slave Error Clear, software write 1 to clear SERR

24.7.4 SPI Interrupt Enable Register (SPIx_IER)

NAME	SPIx_IER (x=0,1,2)							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					ERRIE	TXIE	RXIE
access	U-0					R/W-0	R/W-0	R/W-0

bit	name	description
31:3	--	RFU: **Reserved, read as 0**
2	**ERRIE**	SPI Error Interrupt Enable
1	**TXIE**	Transmit Interrupt Enable
0	**RXIE**	Receive Interrupt Enable

24.7.5 SPI Interrupt Status Register (SPIx_ISR)

NAME	SPIx_ISR (x=0,1,2)							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			DCN_TX	-	RXCOL	TXCOL	BUSY
access	U-0			R/W/Dy-1	U-0	R/W/Dy-0	R/W/Dy-0	R-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	MERR	SERR	-			TXBE	RXBF
access	U-0	R-0	R-0	U-0			R-1	R-0

bit	name	functional description
31:13	--	RFU: **Reserved, read as 0**
12	**DCN_TX**	In half-duplex mode (HALFDUPLEX=1), the DCN signal level is configured to be sent at the last bit of each data frame (Data/Command transmit config) 0: DCN=0 for command frames 1: DCN=1 for data frames The software should set DCN_TX register before transmitting. If DCN_TX=0, the hardware will automatically set DCN_TX to 1 after a frame is sent, i.e. only one command frame will be sent by default, and all subsequent frames will be data frames.
11	--	RFU: **Reserved, read as 0**
10	**RXCOL**	Receive Collision flag, write 1 to clear
9	**TXCOL**	Transmit Collision flag, write 1 to clear
8	**BUSY**	SPI idle flag, read-only (busy flag) 1: SPI transfer in progress 0: SPI transfer idle
7	--	RFU: **Reserved, read as 0**

bit	name	functional description
6	**MERR**	Master Error flag (Master Error flag) MERR is set when the SSN is pulled high before the 8-bit transfer under the master
5	**SERR**	Slave Error flag SERR is set when the SSN is pulled high before 8 bits are transferred under the Slave
4:2	**--**	RFU: **Reserved, read as 0**
1	**TXBE**	TX Buffer Empty flag 1: Tx buffer empty, software write TXBUF to clear 0: Tx buffer full
0	**RXBF**	RX Buffer Full flag 1: Receive buffer full, software reads RXBUF to clear 0: Receive buffer empty

24.7.6 SPI Transmit Buffer (SPIx_TXBUF)

NAME	SPIx_TXBUF (x=0,1,2)							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	TXBUF							
access	W-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	TXBUF							
access	W-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	TXBUF							
access	W-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TXBUF							
access	W-00000000							

bit	name	functional description
31:0	**TXBUF**	Transmit Buffer

24.7.7 SPI Receive Buffer (SPIx_RXBUF)

NAME	SPIx_RXBUF (x=0,1,2)							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	RXBUF							
access	R-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

name	RXBUF							
access	R-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	RXBUF							
access	R-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RXBUF							
access	R-00000000							

bit	name	functional description
31:0	**RXBUF**	Receive Buffer

25 Smart Card Interface (ISO7816)

25.1 Introduction

Smart card interface (7816) is a serial and synchronous communication interface to exchange 8-bit data over 2-wire with external smart card. The chip implements two 7816 master interface function.

- 1-channel 7816-3 interface (mapped to two sets of GPIO)
- Card clock output port, configurable output frequency (1MHz~5MHz)
- configurable bit order (MSB First / LSB First)
- The error signal width can be configured as 1/1.5/2 ETU
- Supports error triggered re-transmission function, and the number of re-transmission times can be configured as 0~3 times
- EGT can be set to 0~256 with support for multiple timeout interrupts
- Data reception complete interrupt and error interrupts
- Configurable interrupt generation condition(buffer empty/ shift register empty)
- Support DMA interface

25.2 Block Diagram

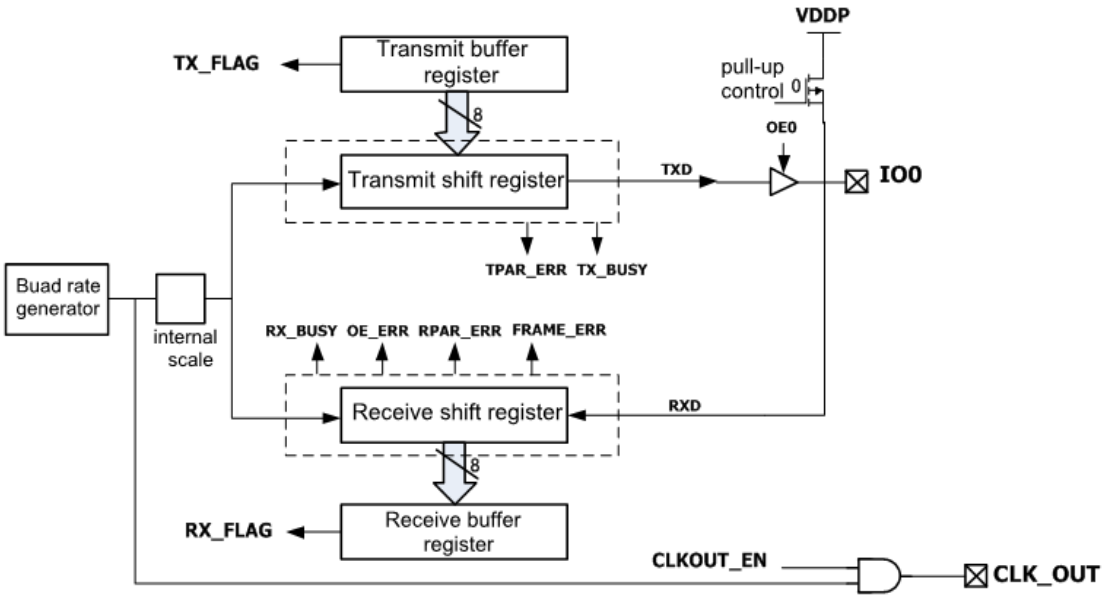


Figure 25-1 ISO7816 Block Diagram

25.3 Clock and Reset

7816 module work clock and bus register clock are APBCLK.

Before using the 7816 module, you must first clear the U7816RST register in the RMU to release the complex bit, and set the U7816_PCE in the CMU to turn on the clock.

25.4 Bus Timing

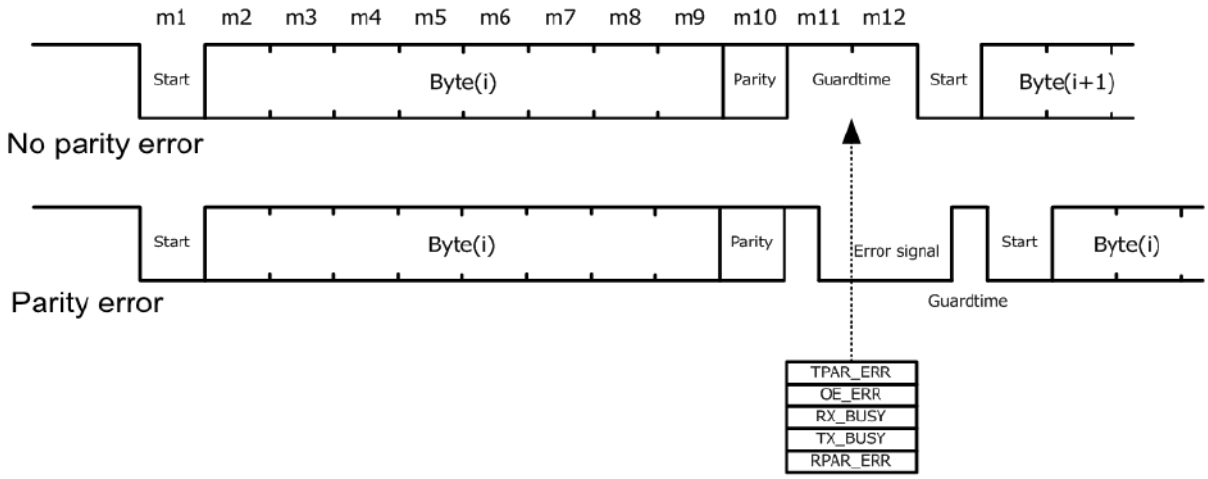


Figure 25-2 ISO7816 Data Frame Structure

According to the protocol of ISO7816, the basic interface sequence of 7816 is as follows:

- A start bit followed by 8 data bits and 1 parity bit ends with GUARDTIME of 1ETU or 2ETU.
- The minimum single-byte data length is 11ETU or 12ETU.
- If the parity is correct, the GUARDTIME of two ETU is inserted to ensure that the data length is 12 ETU. In the 11th ETU, RX_BUSY is invalid and possible OE_ERR flags are generated to complete the data transmission. If there is an ERROR in the receiving parity, IO is pulled down at 10.5ETU to produce an ERROR SIGNAL. The ERROR SIGNAL has a minimum of 1 ETU and a maximum of 2 ETU. The RPAR_ERR flag is generated at the 11th ETU as required.
- At the 11th ETU, if the ERROR SIGNAL is not detected, the transmitted data was correct. The data transmission is finished and TX_BUSY is cleared.
- If ERROR SIGNAL is detected at the 11th ETU, it means the data is not correctly received, and the TPAR_ERR flag is generated and data is re-transmitted after waiting for 2 ETU.

25.5 Functional Description

25.5.1 Data Receive

7816 data receive flow:

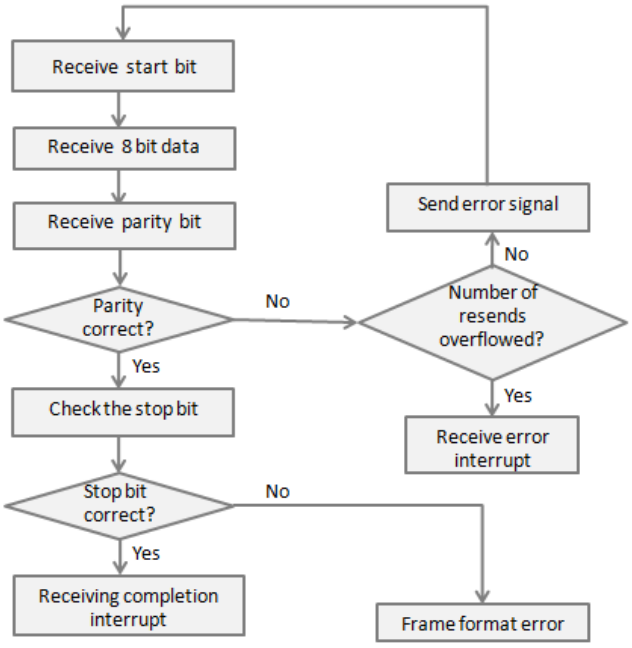


Figure 25-3 ISO7816 Data Receive Process

25.5.2 Data Transmission

When TXEN is turned on, the software only needs to write data to TXBUF, and the hardware will automatically send data when the corresponding IO port is free. The software can write data to TXBUF during the sending process, and the hardware will continue after the previous frame is sent. Send the next frame. When sending data, the internal input port is automatically closed, that is, the data sent by itself cannot be received in the normal application mode of the circuit. It should be noted that since there is only the first level cache in this design, the interval between two TXBUF writes by the software cannot be too short. If TXBUF is written before the state machine loads the data into the shift register and starts sending, the previous data will be washed out. . Note that when sending, the software must at least wait for the hardware to move the previous data into the shift register before writing down a data. The software can monitor TXIF. TXIF is 1 means that the transmit buffer register is empty and the data has entered the shift register for transmission. Write the next data to TXBUF.

7816 data sending process:

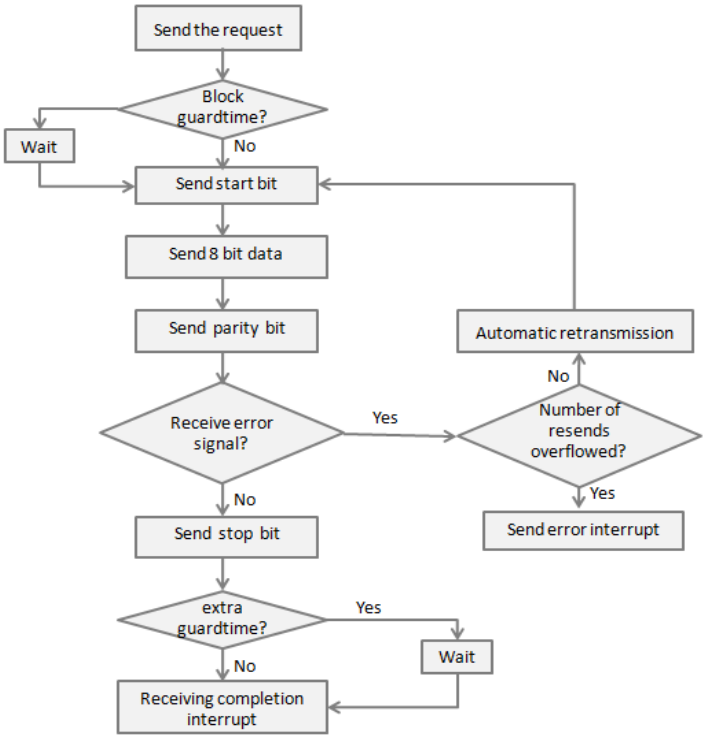


Figure 25-4 ISO7816 Data Transmission Process

25.5.3 Use DMA to Control 7816 for Sending and Receiving Data

When the 7816 module is enabled, the 7816 module will automatically generate the corresponding DMA request when the TX buffer is empty or the RX buffer is full. The application needs to configure DMA channel connections, connect specific channels to 7816 peripherals, set the RAM address pointer, and enable DMA channels. The DMA will then automatically respond to the 7816 request and complete the data transfer between RAM and 7816.

Application: Use DMA to control 7816 for sending and receiving data

- Configure DMA channel 5 as U7816_RX
- Set the RAM pointer address, address increment or decrement, channel priority, transmission length, interrupt, and so on
- Enable the corresponding DMA channel
- Configure the 7816 module parameters
- Enable 7816 module, waiting for data reception
- 7816 automatically generates DMA request after receiving data
- Configure DMA channel 0 to U7816_RX
- DMA responds to the request, reads the 7816 receive buffer, and writes to the specified RAM address

25.6 Register

Offset	Name	Symbol
U7816(Base address: 0x40011C00)		
0x00	U7816 Control Register	U7816_CR
0x04	U7816 Frame Format Register	U7816_FFR
0x08	U7816 Extra Guard Time Register	U7816_EGTR
0x0C	U7816 Prescaler Register	U7816_PSC
0x10	U7816 Baud Rate Generator Register	U7816_BGR
0x14	U7816 Data RX Buffer	U7816_RXBUF
0x18	U7816 Data TX Buffer	U7816_TXBUF
0x1C	U7816 Interrupt Enable Register	U7816_IER
0x20	U7816 Interrupt Status Register	U7816_ISR

25.6.1 U7816 Control Register (U7816_CR)

NAME	U7816_CR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		TXEN	RXEN	CKOEN	-	-	RFU
access	U-0		R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0

bit	name	functional description
31:6	--	RFU: **Reserved, read as 0**
5	TXEN	U7816 channel Transmit Enable Bit (Transmit Enable) 1: Channel TX is enabled 0: Channel TX is disabled
4	RXEN	U7816 channel Receive Enable bit 1: Channel receive enabled 0: Channel receive disabled
3	CKOEN	U7816 Clock CLK output Enable bit (Clock output Enable) 1: 7816 clock output enable 0: 7816 clock output disable
2	--	RFU: **Reserved, read as 0**

bit	name	functional description
1	--	RFU: **Reserved, read as 0**
0	RFU	Reserved

25.6.2 U7816 Frame Format Register (U7816_FFR)

NAME	U7816_FFR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-					ERSW		ERSGD
access	U-0					R/W-00		R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	BGTEN	REP_T	PAR		FREN	TREPEN	RREPEN	DICONV
access	R/W-0	R/W-0	R/W-00		R/W-0	R/W-1	R/W-1	R/W-0

bit	name	functional description
31:13	--	RFU: **Reserved, read as 0**
12:11	DUMMY	Can read and write, no function
10:9	ERSW	ERROR SIGNAL length selection 11: The length of ERROR SIGNAL is 1ETU; 10: The length of ERROR SIGNAL is 1.5ETU; 01: The length of ERROR SIGNAL is 2ETU; 00: The length of ERROR SIGNAL is 2ETU;
8	ERSGD	GUARDTIME length after ERROR SIGNAL (valid only when sending) (Error Signal Guard Time) 1: GUARDTIME after ERROR SIGNAL is 1~1.5ETU. 0: GUARDTIME after ERROR SIGNAL is 2~2.5ETU. When the length of ERROR SIGNAL is an integer ETU, GUARDTIME is 1.5 or 2.5ETU; When the length of ERROR SIGNAL is 1.5ETU, GUARDTIME is 1 or 2ETU
7	BGTEN	BGT control bit. Whether BGT is inserted between receiving and sending. BGT is the minimum time required between receiving and sending (Block Guard Time enable) 1: BGT enable, insert Block Guard Time (12 ETU);

bit	name	functional description
		0: BGT disable, Block Guard Time (12 ETU) is not inserted;
6	REP_T	Number of automatic retransmission time when receiving data parity error (Repeated Times) 1: 3 times 0: 1 time
5:4	PAR	Parity type selection 00: Even 01: Odd 10: Always 1 11: No parity
3	FREN	Receive Guard Time config 1: Guard Time is 1 ETU 0: Guard Time is 2 ETU
2	TREPEN	Transmit Repeat Enable 1: Auto re-transmission enabled on error signal. Tx_perity_err flag is set when re-transmission time exceeds REP_T value 0: No re-transmission enabled. Tx_perity_err flag is set when error signal is detected.
1	RREPEN	Receiving Repeat Enable 1: Reply error signal on parity error. When single byte receive has been repeated for more than REP_T times, rx_parity_err flag is set 0: No error signal reply on parity error. rx_parity_err flag is set
0	DICONV	Transfer order, bit Direction Conversion 1: Reverse coding, sending and receiving MSB first; Reverse logic level 0: Forward coding, sending and receiving LSB first; Positive logic level

25.6.3 U7816 Extra Guard Time Register (U7816_EGTR)

NAME	U7816_EGTR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							

bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TXEGT							
access	R/W-00000000							

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:0	TXEGT	Transmit Extra Guard Time

25.6.4 U7816 Prescaler Register (U7816_PSC)

NAME	U7816_PSC							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			CLKDIV				
access	U-0			R/W-00011				

bit	name	functional description
31:5	--	RFU: **Reserved, read as 0**
4:0	CLKDIV	U7816 Clock output Divider $F_{7816}=F_{APBCLK}/(CLKDIV+1)$ Special case: when CLK_DIV is set to 0 or 1,$F_{7816}=F_{APBCLK}/2$ **Note:** The working clock range specified in the 7816 protocol is 1~5MHz.

25.6.5 U7816 Baud rate Generator Register (U7816_BGR)

NAME	U7816_BGR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							

bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				PDIV			
access	U-0				R/W-0001			
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PDIV							
access	R/W-01110011							

bit	name	functional description
31:12	--	RFU: **Reserved, read as 0**
11:0	PDIV	U7816 Pre-divider control register controlling the Divider of 7816 communications (baud rate) Baud ＝ F_{7816}/(PDIV ＋ 1)

25.6.6 U7816 Data RX Buffer (U7816_RXBUF)

NAME	U7816_RXBUF							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RXBUF							
access	R/W-00000000							

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:0	RXBUF	U7816 Data Receive Buffer

25.6.7 U7816 Data TX Buffer (U7816_TXBUF)

NAME	U7816_TXBUF							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							

access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TXBUF							
access	R/W-00000000							

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:0	TXBUF	U7816 Data Transmit buffer

25.6.8 U7816 Interrupt Enable Register (U7816_IER)

NAME	**U7816_IER**							
Offset	0x1C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name						RXIE	TXIE	LSIE
access						R/W-0	R/W-0	R/W-0

bit	name	functional description
31:3	--	RFU: **Reserved, read as 0**
2	RXIE	Data receiving interrupt enabled bit. RXIF Interrupt flag bit (Receive Interrupt Enable) 1: A receive completion interrupt occurs when the RXIF register is set 0: Abort receiving completion
1	TXIE	Data transmit interrupt enable bit. Corresponding TXIF Interrupt flag bit 1: When the TXIF register setting produces the interrupt that sends the completion 0: Interrupt to disable sending completion
0	LSIE	Interruption of line state enabled bit. Line Status Interrupt Enable (ERRIF)

bit	name	functional description
		1: A line error interrupt occurred while the ERRIF register was set 0: Disallow line error

25.6.9 U7816 Interrupt Status Register (U7816_ISR)

NAME	U7816_ISR							
Offset	0x20							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-					WAIT_RPT	TXBUSY	RXBUSY
access	U-0					R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				TPARERR	RPARERR	FRERR	OVERR
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					RXIF	TXIF	ERRIF
access	U-0					R/W-0	R/W-0	R/W-0

bit	name	functional description
31:19	--	RFU: **Reserved, read as 0**
18	WAIT_RPT	The U7816 interface sends an error signal and is waiting for the data re-transmission.(Waiting for Repeat flag) Set by hardware, software read-only
17	TXBUSY	Transmission busy flag, Automatically reset after transmission complete 1: Set when transmission starts, cleared by hardware at middle of STOP bit 0: Data transmission idle
16	RXBUSY	Receiving busy flag, Automatically reset after receiving complete 1: Data receiving on-going 0: Data receiving idle
15:12	--	RFU: **Reserved, read as 0**
11	TPARERR	Transmit Parity Error, hardware set, write 1 to clear
10	RPARERR	Receive Parity Error flag, hardware set, write 1 to clear
9	FRERR	Frame Error flag, hardware set, write 1 to clear 1: The frame format is incorrect, the length of the frame byte

bit	name	functional description
		received is incorrect, or the frame bit or stop bit received is incorrect 0: No parity error while receiving data
8	OVERR	Receive Overflow Error, hardware set, write 1 to clear 1: The receive buffer register has not been read out while new data has been received. Previous data in the receive buffer register is overwritten 0: No overflow error
7:3	--	RFU: **Reserved, read as 0**
2	RXIF	Receive interrupt flag, hardware set, read data receive buffer to clear 1: 1byte data received, data receiving buffer full 0: No data received. Data receiving buffer is empty
1	TXIF	Transmit interrupt flag, After power-on reset, this flag is automatically set, indicating that the buffer is empty and data can be written. The flag is automatically cleared after the software writes data into TX buffer, and it is set again when data is moved to shift register 1: Tx buffer is empty 0: Tx buffer is not empty
0	ERRIF	Error interrupt flag The bit is ORed from TPARERR, RPARERR, FRERR, OVERR. The software clears the bit by clearing the error flag register above.

26 Controller Area Network (CAN)

26.1 Introduction

CAN module is used for CAN bus data transmission and reception, supports CAN2.0A and 2.0B protocols

The main features of CAN module are as follows:

- Comply with ISO11898-1 protocol, support CAN2.0A and CAN2.0B standards
- Support standard (11bit ID) and extended (29bit ID) frames
- Support the highest baud rate 1Mbps
- 2 messages sending FIFO (32 bytes), 2 messages receiving FIFO (32 bytes)
- 1 high priority sending buffer
- Support automatic retransmission under conditions of error or arbitration failure
- 4 receiving filters
- Support loopback mode
- Support automatic wake-up
- Error counter

26.2 Block Diagram

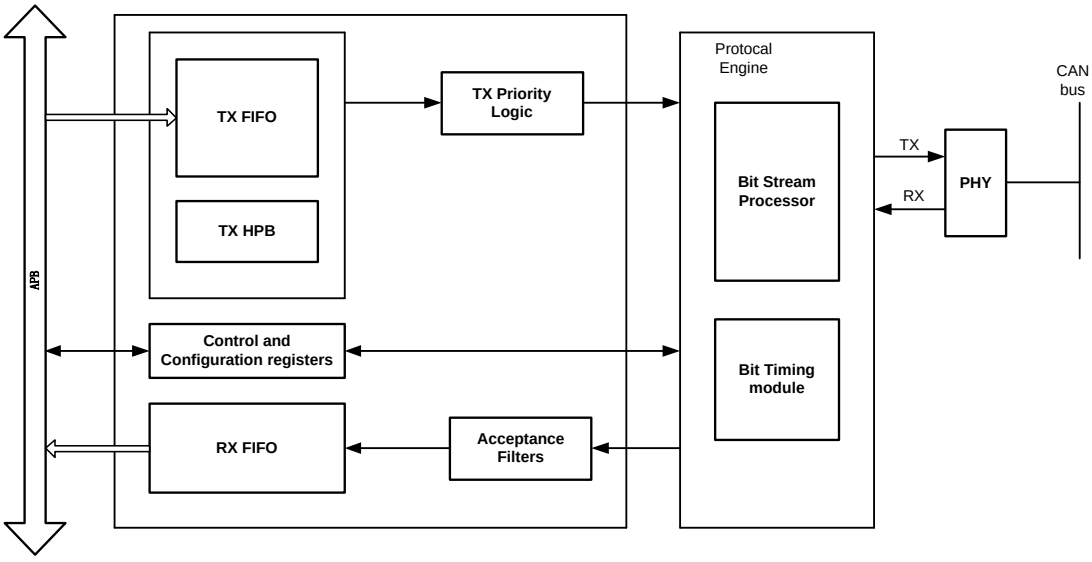


Figure 26-1 CAN Module Structure Block Diagram

The CAN module requires an external CAN PHY chip to implement the physical layer data transmission and reception.

26.3 Pin Definition

The CAN module uses 2 pins to communicate with external devices, and the sending and receiving signals may be mapped to different GPIOs.

The following table shows the pin mapping of FM33LG0x8:

Pin			Symbol	Description
PA6	PC15	PD10	CAN_RX	CAN Receive pin
PA7	PE5	PE9	CAN_TX	CAN Transmit pin

Table 26-1CAN Pin List

Note: *PA16, PC15, PD10 can all be used as CAN input pins. If these three pins are configured as CAN function, the priority is PD10>PC15>PA6*

26.4 Functional Description

26.4.1 Clock and Reset

The CAN module has two clocks: CAN_CLK and SYS_CLK.

CAN bus data transmission and reception use CAN_CLK, and APB bus register access uses SYS_CLK. There is no frequency constraint relationship between CAN_CLK and SYS_CLK. SYS_CLK is connected to APBCLK on the system.

- CAN_CLK frequency range: 8~24Mhz
- SYS_CLK frequency range: 8~64Mhz

CAN_CLK can choose to use RCHF, XTHF, PLL and APBCLK, and the frequency accuracy of the clock used must meet the tolerance range specified by ISO11898-1.

The reset of CAN module includes system reset, RMU software reset, and internal software reset of the module. Among them, by writing 1 to the internal register SRST of the module, the internal self-reset operation of the module can be realized. At this time, all configuration and control registers in the module, including SRST itself, will be reset to the reset value.

26.4.2 Bit Timing

CAN data bit timing can be divided into 4 parts:

- Synchronization segment
- Transmission section
- Phase segment 1
- Phase segment 2

Figure 26-2 CAN Bit Timing

Each segment is composed of a certain number of time slices, called time quanta (tq), the quantum clock period determines the length of tq time, and the quantum clock is obtained by dividing the CAN_CLK internally by the CAN module. You can configure the frequency division coefficient of the quantum clock relative to CAN_CLK by setting the baud rate prescaler register BRPR.

The transmission section and the phase section 1 form the time section 1 (TS1), and the phase section 2 forms the time section 2 (TS2); the number of tq in TS1 and TS2 can be set through the Bit-Timing register BTR. The synchronization segment is always 1 tq in length.

26.4.3 Bit Stream Processor

The BSP module is used to implement the MAC/LLC function in the CAN bus protocol. In the data transmission, the BSP load completes the following tasks:

- Parallel data serialization
- Insert stuff bits, CRC, and other protocol data fields according to protocol requirements

When sending, the BSP monitors the received data at the same time to perform bus arbitration, and when the arbitration fails, it will perform the retransmission task.

When receiving, the BSP removes stuff bits, CRC bits and other protocol data fields, stores the payload data in the RX FIFO, and monitors data errors.

According to the bus error state, the BSP controls the CAN controller to enter the corresponding error state: Error Active, Error Passive, Bus Off. When a transmission error event is found on the TX or RX data signal, the BSP module updates the bus error counter (CAN_ECR) according to the rules specified by the ISO11898-1 standard, and makes the CAN controller enter various error states according to the value of ECR.

26.4.4 Controller Working Mode

The CAN controller module supports the following working modes:

- Configuration
- Normal
- Loop Back

Configuration mode

When any of the following events occur, the CAN controller will enter the Configuration mode:

- Write 0 to the CEN register
- Write 1 to SRST register
- The CAN controller enters the configuration mode by default after reset

In this mode, the CAN controller is in the following state:

- Stop synchronizing with the bus and drive fixed recessive bit output
- ECR and ESR register reset
- BTR and BRPR registers can be rewritten
- CAN controller no longer receives any messages
- The messages in TX FIFO and TX HPB will not be sent, but will be retained; when entering Normal mode, these messages will be sent
- Can read messages in RX FIFO
- Can write messages to TX FIFO and TX HPB
- All configuration registers can be accessed

In configuration mode, setting the CEN register will make the CAN controller exit this mode after waiting for 11 consecutive recessive bits.

After exiting the configuration mode, the CAN controller will enter Normal or LoopBack mode according to the status of the LBACK and SLEEP registers.

Normal mode

In normal mode, the CAN controller participates in bus communication and sends and receives messages normally. The Configuration mode can be entered from the normal mode controller.

Loop Back mode

In the loop back mode, the CAN controller does not participate in bus communication, only continuously sends recessive bits to the bus, and does not receive any messages on the CAN bus. At the same time, the internally sent bit stream is directly looped back to the receiving end to realize spontaneous transmission and self-reception for testing and diagnosis.

The controller can only enter the loop back mode from the configuration mode by setting the LPBACK register to 1, and the CEN register to 1.

26.4.5 Message Storage and Structure

Message storage

CAN bus communication takes Message Frame as the basic organization form. The message to be sent is stored in TX FIFO or HPB, where HPB is a high-priority Buffer, and the message frame in HPB is always sent prior to the message frame in TX FIFO. HPB can only store one message, while TX FIFO can store 2 messages.

The message received by the CAN controller is filtered by the acceptance filter first, and the message that meets the filter rules will be stored in the RX FIFO, otherwise it will be discarded. Two messages can be stored in the RX FIFO.

The sending and receiving of messages follows the following rules:

- The priority of messages in TX HPB is higher than TX FIFO
- If there is an arbitration failure or error during the sending process, the CAN controller will try to resend the current message; only after the current message is successfully sent will subsequent messages be sent, even the message in the HPB cannot interrupt the retransmission process of the current message
- The message data in TX FIFO, HPB, RX FIFO will still be retained when the CAN controller is in Bus Off state or Configuration mode

Message structure

The length of each message on the CAN bus is 16 bytes, including message ID (4 bytes), data length code DLC (4 bytes), Data Word 1 (4 bytes) and Data Word 2 (4 bytes).

Message ID (Identifier)

bit	31	30:13	12	11	10:0
field	RTR	ID[17:0]	IDE	SRR/RTR	ID[28:18]

Data Length Code – DLC

bit	31:4	3:0
field	RFU	DLC

Data Word 1

bit	31:24	23:16	15:8	7:0
field	DB3[7:0]	DB2[7:0]	DB1[7:0]	DB0[7:0]

Data Word 2

bit	31:24	23:16	15:8	7:0
field	DB7[7:0]	DB6[7:0]	DB5[7:0]	DB4[7:0]

When receiving a message, the software must read the complete 16 bytes from the RX FIFO, even if the length of the received message data is less than 8 bytes; each message is read in 4 times and the 32bit read operation is completed, in the order of ID, DLC, DW1, DW2 Read sequentially from the FIFO.

When sending a message, the software must write a complete 16 bytes to the TX FIFO or HPB, and must write 4 words in the order of ID, DLC, DW1, DW2, even if the effective data length is less than 8 bytes, it must be completed 4 words are written, and the vacant bits in DW should be filled with 0. When sending data, the MSB of each byte is sent first, and the data bytes are sent in the order of DB0~DB7.

Note: *CAN_TXF_IDR, CAN_TXF_DLCR, CAN_TXF_DW1R, CAN_TXF_DW2R registers are in the form of FIFO, and the writes to these four addresses will be sequentially mapped to the FIFO push operation; for example, if the software writes CAN_TXF_DLCR 4 times in a row, the hardware circuit will also do the same ID, DLC, DW1, DW2 are written 4 times. To avoid confusion, the software is required to perform continuous writes to the transmit FIFO in strict order.*

Message identification code (Identifier)

The CAN controller supports two formats of message identification codes:

- Standard frame: ID length is 11bit, only ID[28:18], SRR/RTR and IDE are valid bits, where IDE is 0, SRR/RTR is selected according to data frame and remote frame.
- Extended frame: ID length is 29bit, all bits are valid, SRR/RTR and IDE are 1

Bits	Name	Descriptions
31	RTR	Remote Transmission Request Only extended frames are valid 1: Remote frame 0: Data frame
30:13	ID[18:0]	Extended Message ID Only extended frames are valid Standard frame should be all 0
12	IDE	Identifier Extension 1: Use extended message identification code, 29-digit ID 0: Use standard message identification code, 11-digit ID
11	SRR/RTR	Substitute Remote Transmission Request Only the standard frame is valid, and the extended frame should remain 1 1: Indicates that the current message frame is a remote frame 0: Indicates that the current message frame is a data frame
10:0	ID[28:18]	Standard Message ID Both standard frame and extended frame are valid

26.4.6 Acceptance Filter

The CAN controller has 4 message filters, and each message filter contains a set of Mask registers and a set of ID registers.

The working steps of message filtering are as follows:

- The received message ID is logically ANDed with the Mask register
- The contents of the filter ID register are also logically ANDed with the Mask register
- Compare two logical AND operation results
- If the two results are equal, the message passes the filter and is stored in the RX FIFO
- 4 filters can work at the same time, as long as the received message passes through any filter, it will be stored in the RX FIFO
- Message frames that cannot pass the filter are discarded

Note: *The software can choose to enable any number of message filters or close all message filters. If none of the message filtering is enabled, all received messages will be stored in the RX FIFO.*

26.4.7 Error Management

26.4.7.1 Node Strategy

CAN contains a transmission error counter and a reception error counter. When the message frame is correctly sent or received, the corresponding error counter should be decremented, and when an error is encountered during transmission or reception, the corresponding error counter should be incremented.

The detailed rules for error counting are listed below:

- When the receiving node detects an error, the receiving error counter is +1 (exception: bit errors detected when the active error flag or overload flag is sent will not cause the receiving error counter to increase)
- After the receiving node sends the active error flag, it detects that the subsequent first bit is a dominant bit, and the receiving error counter +8
- The sending node sends an error flag, and the sending error counter is +8; the sending error counter remains unchanged under the following exceptions
  - The sending node is in a passive error state, an ACK error is detected, and there is no dominant bit in the passive error flag
  - The stuffing bit error is detected in the arbitration stage, and the recessive stuffing bit is sent, but the dominant stuffing bit is actually detected
- The sending node detects a bit error when sending an active error flag or an overload flag, and sends an error counter +8
- When 14 consecutive dominant bits on the bus are detected, or 8 consecutive dominant bits are detected after the passive error flag, the sending node should set the sending error counter +8, and the receiving node should set the receiving error counter +8
- When a message frame is sent correctly, send error counter -1, reduce to 0 to stop
- When a message frame is received correctly, if the reception error counter is between 1 and 127, it should be -1, if it is 0, it will remain unchanged, and if it is greater than 127, it will be set to a number between 119 and 127.

26.4.7.2 Error State Transition

After CAN exits the configuration mode, it enters the error active state. In this state, the CAN controller will actively send an error frame after detecting an error.

When the count value of the receiving error counter or the sending error counter is greater than 127,

CAN will enter the error passive state. In this state, CAN no longer sends error frames.

When the count values of the receiving error counter and the sending error counter of the passive error node are less than or equal to 127, CAN returns to the active error state.

When the sending error counter value of the node is greater than 255, the node should enter the bus-off state (bus-off).

The node in the bus off state cannot send any frame, nor can it send ACK, which has no effect on the bus.

The node error state transition can conform to the ISO11898-1 protocol specification shown in the figure below:

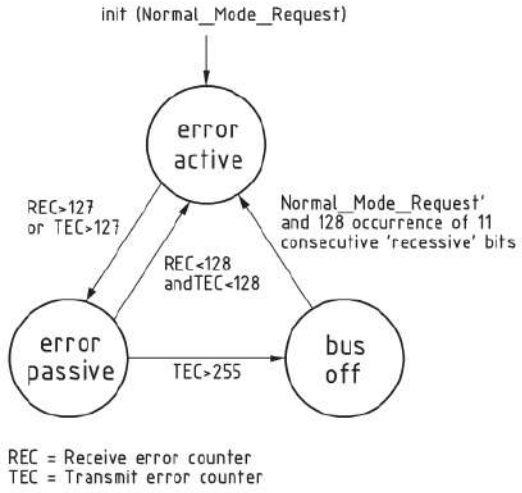


Figure 26-3 Node Error State Transition

CAN defines the following error states:

- Normal reception and normal transmission: TEC and REC are both less than 96
- Receive warning: REC is greater than or equal to 96 and less than 128
- Receive error: REC is greater than or equal to 128
- Send warning: TEC is greater than or equal to 96 and less than 128
- Send error: TEC is greater than or equal to 128 and less than 256
- Bus-off: TEC is greater than or equal to 256

The ESTAT register is used to indicate the current state of the CAN controller, and the software can also determine the state of the controller by querying the REC and TEC registers.

26.4.7.3 Bus-Off Recovery

CAN can support the use of the following strategies to recover from the bus off state.

- Automatic recovery mode

CAN adopts a bus shutdown recovery method that conforms to the ISO11898 protocol specification. When CAN continuously detects 11bit recessive bits more than 128 times, the controller automatically returns to the error active state. The send and receive error counters TEC and REC are automatically cleared, and the BORF interrupt flag is set.

26.4.7.4 Error Management Interrupts and Flags

When the CAN internal state machine enters the bus-off, the BSOFFIF interrupt flag is set, and an interrupt event is generated at the same time when the BSOFFIE register is enabled.

When the TEC or REC count value is greater than 96, the ERRWRN flag register is set. This flag is only used to indicate that the error count value is high and give a software warning, and no interrupt can be generated.

The increment of TEC or REC is because the CAN controller detects an error, so when the ERROR interrupt is set, the software can query the TEC and REC count values to determine whether the internal error status of the CAN controller has shifted.

26.5 Programming Model

The CAN bus module programming model is used to guide software programming and application development, including register configuration methods and message transmission.

26.5.1 Register Configuration

After chip power-on reset or system reset, before CAN bus communication, you need to configure CAN bus registers according to the following instructions.

- Select operation mode

For loopback mode, set the LPBACK bit in the MSR.

For normal mode, clear the LPBACK bit in the MSR.

- Configure the transport layer configuration register

Program the baud rate prescaler register BRPR and the bit timing register BTR to correspond to the network timing parameters and the network characteristics of the system.

- Configure receiving filter register

Clear the UAF bit in the CAN_AFR register

Query the ACFBSY bit in CAN_SR until it returns 0

Write the appropriate mask information into the receive filter mask register AFMRx

Write the appropriate ID information to the receive filter register AFIRx

Set the UAF bit in the CAN_AFR register

Repeat the above steps for each group of receive filter mask register AFMRx and receive filter register AFIRx that need to be enabled

Configure the interrupt enable register IER to select the bits in the interrupt status register ISR that can generate interrupts

Set the CEN bit in the CAN_CR register to enable CAN

26.5.2 Message Transmission

The message to be sent can be written to TX FIFO or TX HPB. The message in TX HPB takes precedence over the message in TX FIFO. The TXOKIF bit in CAN_ISR is automatically set after CAN successfully transmits a message.

After the message frame is completely written into the TX FIFO or TX HPB, the data in the FIFO or HPB is in a sending waiting state. If the bus is IDLE, the sending process is automatically started.

Note: *CAN controller does not support send cancel function.*

- Write messages to TX FIFO

All messages written into the TX FIFO should follow the format defined in the previous "message frame storage and message frame structure".

Perform message writing:

1. Polling the TXFLL bit in CAN_SR. When the TXFLL bit is "0", the message can be written into the TX FIFO.

2. Write the ID of the message into the send FIFO ID register

3. Write the data length of the message into the send FIFO DLC register

4. Write the message DW1 into the send FIFO DataWord1 register

5. Write the message DW2 into the send FIFO DataWord2 register

Messages can be continuously written to the TX FIFO until the TX FIFO is full. When the TX FIFO is full, the hardware sets the TXFLLIF bit in CAN_ISR and the TXFLL bit in CAN_SR. If the software adopts the polling method, it should poll the TXFLL bit in the status register after each write. If the software uses the interrupt mode, it can be written continuously until the TXFLLIF bit in the ISR generates an interrupt.

- Write messages to TX HPB

All messages written into the TX FIFO should follow the format defined in the previous "message frame storage and message frame structure".

Write a message to TX HPB:

1. Polling the TXBFLL bit in CAN_SR. When the TXBFLL bit is "0", the message can be written to TX HPB.

2. Write the ID of the message into the high priority send buffer ID register

3. Write the data length of the message into the high priority transmit buffer DLC register

4. Write the message DW1 into the high-priority transmit buffer DataWord1 register

5. Write the message DW2 into the high priority send buffer DataWord2 register

After each write to TX HPB, the TXBFLL bit in the status register CAN_SR and the TXBFLLIF bit in the interrupt status register CAN_ISR are set to 1.

- Receive messages

Whenever a new message is received and written into the RX FIFO, the RXNEMPIF and RXOKIF bits in CAN_ISR will be set. If a read operation is performed on an empty RX FIFO, the RCUFLW bit in

CAN_ISR is set to 1.

Read messages from RX FIFO:

1. Polling the RXOKIF or RXNEMPIF bit in CAN_ISR. In interrupt mode, the read FIFO can be executed after the RXOKIF or RXNEMPIF bit in CAN_ISR generates an interrupt.

-Read the ID of the message from the receive FIFO ID register

-Read the data length of the message from the receive FIFO DLC register

-Read message DW1 from the receiving FIFO DW1 position

-Read message DW2 from the receiving FIFO DW2 position

After reading, if there are one or more messages in the RX FIFO, set the RXNEMPIF bit in CAN_ISR. This bit can either be polled or an interrupt can be generated.

Repeat this operation until the FIFO is empty.

26.6 Register

Offset	Name	Symbol
CAN(Base address: 0x40019400)		
0x00	CAN Control Register	CAN_CR
0x04	CAN Mode Select Register	CAN_MSR
0x08	CAN Baud Rate Prescaler Register	CAN_BRPR
0x0C	CAN Bit Timing Register	CAN_BTR
0x10	CAN Error Counter Register	CAN_ECR
0x14	CAN Error Status Register	CAN_ESR
0x18	CAN Status Register	CAN_SR
0x1C	CAN Interrupt Status Register	CAN_ISR
0x20	CAN Interrupt Enable Register	CAN_IER
0x24	CAN Interrupt Clear Register	CAN_ICR
0x28	-	-
0x2C	-	-
0x30	CAN TX FIFO ID Register	CAN_TXF_IDR
0x34	CAN TX FIFO DLC Register	CAN_TXF_DLCR
0x38	CAN TX FIFO Data Word1 Register	CAN_TXF_DW1R
0x3C	CAN TX FIFO Data Word2 Register	CAN_TXF_DW2R
0x40	CAN TX HPB ID Register	CAN_HPB_IDR
0x44	CAN TX HPB DLC Register	CAN_HPB_DLCR
0x48	CAN TX HPB Data Word1 Register	CAN_HPB_DW1R
0x4C	CAN TX HPB Data Word2 Register	CAN_HPB_DW2R
0x50	CAN RX FIFO ID Register	CAN_RXF_IDR
0x54	CAN RX FIFO DLC Register	CAN_RXF_DLCR
0x58	CAN RX FIFO Data Word1 Register	CAN_RXF_DW1R
0x5C	CAN RX FIFO Data Word2 Register	CAN_RXF_DW2R
0x60	Acceptance Filter Register	CAN_AFR
0x64	Acceptance Filter Mask Register1	CAN_AFMR1
0x68	Acceptance Filter ID Register1	CAN_AFIR1
0x6C	Acceptance Filter Mask Register2	CAN_AFMR2
0x70	Acceptance Filter ID Register2	CAN_AFIR2
0x74	Acceptance Filter Mask Register3	CAN_AFMR3
0x78	Acceptance Filter ID Register3	CAN_AFIR3
0x7C	Acceptance Filter Mask Register4	CAN_AFMR4
0x80	Acceptance Filter ID Register4	CAN_AFIR4

26.6.1 CAN Control Register (CAN_CR)

NAME	CAN_CR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						CEN	SRST
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
1	CEN	CAN controller enable 0: CAN controller is in configuration mode 1: CAN controller enters LoopBack or Normal mode according to the status of the LPBACK register
0	SRST	Software reset register Write 1 to reset CAN controller, read out always 0

26.6.2 CAN Mode Select Register (CAN_MSR)

NAME	**CAN_MSR**							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name			-					
access			U-0					
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						LPBACK	-
access	U-0						R/W-0	U-0

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
1	LPBACK	LoopBack mode 1: CAN controller is in LoopBack mode 0: CAN controller is in Normal or Configuration mode

bit	name	functional description
0	--	RFU: Need to keep 0, write 1 is prohibited

26.6.3 CAN Baud Rate Prescaler Register (CAN_BRPR)

NAME	CAN_BRPR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	BPR							
access	R/W-0000 0000							

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:0	BRP	Baud Rate Prescaler, prescale CAN_CLK according to the value of this register to get the quantum clock frequency The actual frequency value is BPR+1 Tq=tCAN_CLK * (BPR+1)

26.6.4 CAN Bit Timing Register (CAN_BTR)

NAME	CAN_BTR							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							SJW[1]
access	U-0							R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SJW[0]	TS2			TS1			
access	R/W-0	R/W-000			R/W-0000			

bit	name	functional description
31:9	--	RFU: **Reserved, read as 0**
8:7	SJW	Synchronization Jump Width, see CAN2.0 protocol tSJW = tq * (SJW + 1)
6:4	TS2	Time Segment2, which defines the length of Phase Segment2 tTS2 = tq * (TS2 + 1)
3:0	TS1	Time Segment1, which defines the length of Propagation Segment + Phase Segment1 tTS1 = tq * (TS1 + 1)

26.6.5 CAN Error Counter Register (CAN_ECR)

NAME	CAN_ECR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	REC							
access	R-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TEC							
access	R-0000 0000							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:8	REC	Receive Error Counter The software is read-only, REC increases by 1 every time a receiving error occurs
7:0	TEC	Transmit Error Counter The software is read-only, TEC increases by 1 every time a sending error occurs

Note: The ECR register is cleared in the following cases

- Write 1 to SRST register
- Clear the CEN register
- CAN controller enters Bus Off
- During the Bus off recovery phase, the CAN controller enters Error Active after receiving 128 L1

recessive bits

26.6.6 CAN Error Status Register (CAN_ESR)

NAME	CAN_ESR							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			ACKER	BERR	STER	FMER	CRCER
access	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:5	--	RFU: **Reserved, read as 0**
4	ACKER	Acknowledge Error 1: ACK error detected 0: No ACK error Hardware set, software write 1 to clear
3	BERR	Bit Error 1: The received bit is different from the bit being sent 0: No bit error Hardware set, software write 1 to clear
2	STER	Stuffing Error 1: Data stuffing bit error 0: No filling error Hardware set, software write 1 to clear
1	FMER	Form Error 1: There is an error in the fixed format field in the message frame 0: No error Hardware set, software write 1 to clear
0	CRCER	CRC Error 1: CRC check error 0: CRC check is correct Hardware set, software write 1 to clear

26.6.7 CAN Status Register (CAN_SR)

NAME	CAN_SR							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				ACFBSY	TXFLL	TXBFLL	ESTAT[1]
access	U-0				R-0	R-0	R-0	R-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ESTAT[0]	ERRWRN	BBSY	BIDLE	NORMAL	-	LPBACK	CONFIG
access	R-0	R-0	R-0	R-0	R-0	U-0	R-0	R-1

bit	name	functional description
31:12	--	RFU: **Reserved, read as 0**
11	ACFBSY	Acceptance Filter Busy 1: Acceptance Filter works, Acceptance Filter Mask and ID registers cannot be rewritten 0: Acceptance Filter is idle, Mask and ID registers can be rewritten
10	TXFLL	Transmit FIFO is FULL 1: TX FIFO is full 0: TX FIFO is not full
9	TXBFLL	High Priority Transmit Buffer is FULL 1: HPB full 0: HPB dissatisfaction
8:7	ESTAT	Error Status, read only 00: CAN controller is in Configuration mode 01: Error Active state 10: Bus off state 11: Error Passive state
6	ERRWRN	Error Warning, read only 1: TEC or REC count value is greater than or equal to 96 0: TEC and REC count values are both less than 96
5	BBSY	Bus Busy logo, read only 1: CAN controller is sending and receiving data 0: CAN controller is in configuration mode, or bus IDLE
4	BIDLE	Bus IDLE logo, read only

bit	name	functional description
		1: There is currently no bus communication 0: The current bus is communicating, or the CAN controller is in configuration mode
3	NORMAL	Normal mode flag, read only 1: CAN controller is in normal mode 0: CAN controller is not in normal mode
2	--	RFU: **Reserved, read as 0**
1	LBACK	Loop Back mode flag, read only 1: CAN controller is in loop back mode 0: CAN controller is not in loop back mode
0	CONFIG	Configuration mode flag, read only 1: CAN controller is in configuration mode 0: CAN controller is not in configuration mode

26.6.8 CAN Interrupt Status Register (CAN_ISR)

NAME	CAN_ISR							
Offset	0x1C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						BSOFFIF	ERRORIF
access	U-0						R-0	R-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RXNEMPIF	RXOFLWIF	RXUFLWIF	RXOKIF	TXBFLLIF	TXFLLIF	TXOKIF	ARBLSTIF
access	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

bit	name	functional description
31:10	--	RFU: **Reserved, read as 0**
9	BSOFFIF	Bus Off interrupt flag, set by hardware, cleared by software by writing ICR register 1: CAN controller enters Bus Off state
8	ERRORIF	Error interrupt flag, set by hardware, cleared by software by writing ICR register 1: An error occurred during messaging
7	RXNEMPIF	Receive FIFO Not Empty interrupt flag, set by hardware,

bit	name	functional description
		cleared by software by writing ICR register 1: The receive FIFO is not empty
6	RXOFLWIF	Receive FIFO Overflow interrupt flag, set by hardware, cleared by software by writing ICR register 1: The receive FIFO overflows, that is, a new message is received when the receive FIFO is full
5	RXUFLW	Receive FIFO Underflow interrupt flag, set by hardware, cleared by software by writing ICR register 1: The receive FIFO underflows, that is, the read operation is performed when the receive FIFO is empty
4	RXOKIF	Receive OK interrupt flag, set by hardware, cleared by software by writing ICR register 1: Indicates that 1 message frame was successfully received
3	TXBFLLIF	High Priority Transmit Buffer FULL interrupt flag, set by hardware, cleared by software by writing ICR register 1: Indicates that the high priority sending BUFFER is full
2	TXFLLIF	Transmit FIFO FULL interrupt flag, set by hardware, cleared by software by writing ICR register 1: Indicates that the sending FIFO is full
1	TXOKIF	Transmission OK interrupt flag, set by hardware, cleared by software writing ICR register 1: Indicates that 1 message frame was successfully sent
0	ARBLSTIF	Arbitration Lost interrupt flag, set by hardware, cleared by software by writing ICR register 1: Arbitration failed

26.6.9 CAN Interrupt Enable Register (CAN_IER)

NAME	CAN_IER							
Offset	0x20							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						EBSOFFIE	EERRORIE
access	U-0						R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ERXNE	ERXOFL	ERXUFL	ERXOKI	ETXBFL	ETXFLLI	ETXOKI	EARBLS

	MPIE	WIE	WIE	E	LIE	E	E	TIE
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:10	--	RFU: **Reserved, read as 0**
9	BSOFFIE	Bus Off interrupt enable, 1 allows interrupt, 0 disables interrupt
8	ERRORIE	Error interrupt enable, 1 enables interrupt, 0 disables interrupt
7	RXNEMPIE	Receive FIFO Not Empty interrupt enable, 1 enables interrupt, 0 disables interrupt
6	RXOFLWIE	Receive FIFO Overflow interrupt enable, 1 enables interrupt, 0 disables interrupt
5	RXUFLWIE	Receive FIFO Underflow interrupt enable, 1 enables interrupt, 0 disables interrupt
4	RXOKIE	Receive OK interrupt enable, 1 allows interrupt, 0 disables interrupt
3	TXBFLLIE	High Priority Transmit Buffer FULL interrupt is enabled, 1 allows interrupts, 0 prohibits interrupts
2	TXFLLIE	Transmit FIFO FULL interrupt enable, 1 enables interrupt, 0 disables interrupt
1	TXOKIE	Transmission OK interrupt enable, 1 allows interrupt, 0 disables interrupt
0	ARBLSTIE	Arbitration Lost interrupt enable, 1 enables interrupt, 0 disables interrupt

26.6.10 CAN Interrupt Clear Register (CAN_ICR)

NAME	**CAN_ICR**							
Offset	0x24							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						CBSOFF	CERROR
access	U-0						R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CRXNEMP	CRXOFLW	CRXUFLW	CRXOK	CTXBFLL	CTXFLL	CTXOK	CARBLST
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:10	--	RFU: **Reserved, read as 0**
9	CBSOFF	Clear Bus Off, write 1 to clear BSOFF
8	CERROR	Clear ERROR, write 1 to clear ERROR
7	CRXNEMP	Clear Receive FIFO Not Empty, write 1 to clear RXNEMP
6	CRXOFLW	Clear Receive FIFO Overflow, write 1 to clear RXOFLW
5	CRXUFLW	Clear Receive FIFO Underflow, write 1 to clear RCUFLW
4	CRXOK	Clear Receive OK, write 1 to clear RXOK
3	CTXBFLL	Clear High Priority Transmit Buffer FULL, write 1 to clear TXBFLL
2	CTXFLL	Clear Transmit FIFO FULL, write 1 to clear TXFLL
1	CTXOK	Clear Transmission OK, write 1 to clear TXOK
0	CARBLST	Clear Arbitration Lost, write 1 to clear ARBLST

26.6.11 CANTX FIFO ID Register (CAN_TXF_IDR)

NAME	CAN_TXF_IDR							
Offset	0x30							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	IDR[31:24]							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	IDR[23:16]							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	IDR[15:8]							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	IDR[7:0]							
access	W-0000 0000							

bit	name	functional description
31:0	IDR	Send message identification code (Identifier Register) For the data format, see 26.4.5 Message storage and structure (Message storage and structure)

26.6.12 CANTX FIFO DLC Register (CAN_TXF_DLCR)

NAME	CAN_TXF_DLCR							
Offset	0x34							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							

bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				DLC			
access	U-0				W-0000			

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3:0	DLC	Data Length Code Define the data byte length in the message frame

26.6.13 CANTX FIFO Data Word1 Register (CAN_TXF_DW1R)

Name	CAN_TXF_DW1R							
offset	0x38							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	DB3							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DB2							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DB1							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DB0							
access	W-0000 0000							

bit	name	functional description
31:24	DB3	Data Byte 3
23:16	DB2	Data Byte 2
15:8	DB1	Date Byte 1
7:0	DB0	Data Byte 0

26.6.14 CANTX FIFO Data Word2 Register (CAN_TXF_DW2R)

NAME	CAN_TXF_DW2R							
Offset	0x3C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

name	DB7							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DB6							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DB5							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DB4							
access	W-0000 0000							

bit	name	functional description
31:24	DB7	Data Byte 7
23:16	DB6	Data Byte 6
15:8	DB5	Date Byte 5
7:0	DB4	Data Byte 4

26.6.15 CANTX HPB ID Register (CAN_HPB_IDR)

NAME	CAN_HPB_IDR							
Offset	0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	IDR[31:24]							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	IDR[23:16]							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	IDR[15:8]							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	IDR[7:0]							
access	W-0000 0000							

bit	name	functional description
31:0	IDR	Send message identification code (Identifier Register) For the data format, see 26.4.5 Message storage and structure (Message storage and structure)

26.6.16 CANTX HPB DLC Register (CAN_HPB_DLCR)

NAME	CAN_HPB_DLCR							
Offset	0x44							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				DLC			
access	U-0				W-0000			

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3:0	DLC	Data Length Code Define the data byte length in the message frame

26.6.17 CANTX HPB Data Word1 Register (CAN_HPB_DW1R)

NAME	CAN_HPB_DW1R							
Offset	0x48							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	DB3							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DB2							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DB1							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DB0							
access	W-0000 0000							

bit	name	functional description
31:24	DB3	Data Byte 3
23:16	DB2	Data Byte 2
15:8	DB1	Date Byte 1

bit	name	functional description
7:0	DB0	Data Byte 0

26.6.18 CANTX HPB Data Word2 Register (CAN_HPB_DW2R)

NAME	CAN_HPB_DW2R							
Offset	0x4C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	DB7							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DB6							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DB5							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DB4							
access	W-0000 0000							

bit	name	functional description
31:24	DB7	Data Byte 7
23:16	DB6	Data Byte 6
15:8	DB5	Date Byte 5
7:0	DB4	Data Byte 4

26.6.19 CAN RXFIFO ID Register (CAN_RXF_IDR)

NAME	CAN_RXF_IDR							
Offset	0x50							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	IDR[31:24]							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	IDR[23:16]							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	IDR[15:8]							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	IDR[7:0]							
access	W-0000 0000							

bit	name	functional description
31:0	IDR	Send message identification code (Identifier Register) For the data format, see 26.4.5 Message storage and structure (Message storage and structure)

26.6.20 CAN RXFIFO DLC Register (CAN_RXF_DLCR)

NAME	CAN_RXF_DLCR							
Offset	0x54							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				DLC			
access	U-0				W-0000			

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3:0	DLC	Data Length Code Define the data byte length in the message frame

26.6.21 CAN RXFIFO Data Word1 Register (CAN_RXF_DW1R)

NAME	CAN_RXF_DW1R							
Offset	0x58							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	DB3							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DB2							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DB1							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DB0							
access	W-0000 0000							

bit	name	functional description
31:24	DB3	Data Byte 3
23:16	DB2	Data Byte 2
15:8	DB1	Date Byte 1
7:0	DB0	Data Byte 0

26.6.22 CAN RXFIFO Data Word2 Register (CAN_RXF_DW2R)

NAME	CAN_RXF_DW2R							
Offset	0x5C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	DB7							
access	W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DB6							
access	W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DB5							
access	W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DB4							
access	W-0000 0000							

bit	name	functional description
31:24	DB7	Data Byte 7
23:16	DB6	Data Byte 6
15:8	DB5	Date Byte 5
7:0	DB4	Data Byte 4

26.6.23 Acceptance Filter Register (CAN_AFR)

NAME	CAN_AFR							
Offset	0x60							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							

bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				UAF4	UAF3	UAF2	UAF1
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3	UAF4	Use Acceptance Filter 4 1: Enable No. 4 filter 0: Turn off filter No. 4
2	UAF3	Use Acceptance Filter 3 1: Enable filter 3 0: Turn off filter 3
1	UAF2	Use Acceptance Filter 2 1: Enable filter 2 0: Turn off filter 2
0	UAF1	Use Acceptance Filter 1 1: Enable filter 1 0: Turn off filter No. 1

26.6.24 Acceptance Filter Mask Register x (CAN_AFMRx)

NAME	CAN_AFMRx (x=1,2,3,4)							
Offset	0x64 + (x-1)*0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	AMRTR	AMID[17:11]						
access	R/W-0	R/W-000 0000						
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	AMID[10:3]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	AMID[2:0]			AMIDE	AMSRR	AMID[28:26]		
access	R/W-000			R/W-0	R/W-0	R/W-000		
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	AMID[25:18]							
access	R/W-0000 0000							

bit	name	functional description
31	AMRTR	RTR Mask 1: RTR bit participates in filter comparison 0: RTR bit does not participate in filter comparison
30:13	AMID[17:0]	ID Mask bit 17-0 1: Corresponding bit participates in filter comparison 0: Corresponding bit does not participate in filter comparison

bit	name	functional description
12	AMIDE	IDE Mask 1: IDE bit participates in filter comparison 0: IDE bit does not participate in filter comparison
11	AMSRR	SRR Mask 1: SRR bit participates in filter comparison 0: SRR bit does not participate in filter comparison
10:0	AMID[28:18]	ID Mask bit 28-18 1: Corresponding bit participates in filter comparison 0: Corresponding bit does not participate in filter comparison

26.6.25 Acceptance Filter ID Register x (CAN_AFIRx)

NAME	CAN_AFIRx (x=1,2,3,4)							
Offset	0x68 + (x-1)*0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	AIRTR	AIID[17:11]						
access	R/W-0	R/W-000 0000						
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	AIID[10:3]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	AIID[2:0]			AIIDE	AISRR	AIID[28:26]		
access	R/W-000			R/W-0	R/W-0	R/W-000		
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	AIID[25:18]							
access	R/W-0000 0000							

bit	name	functional description
31	AIRTR	Filter RTR (Acceptance ID RTR)
30:13	AIID[17:0]	Filter ID bit 17-0 (Acceptance ID IDR)
12	AIIDE	Filter IDE (Acceptance ID IDE)
11	AISRR	Filter SRR (Acceptance ID SRR)
10:0	AIID[28:18]	Filter ID bit 28-18 (Acceptance ID IDR)

27 Direct Memory Access Controller (DMA)

27.1 Introduction

- 7-channel peripheral PDMA, support Peripherals<>RAM transfer
- 1 channel memory MDMA, support Flash<>RAM transfer
- Peripheral DMA transfer is triggered by the request of the peripheral, and the CPU operation is not affected during the DMA operation
- The maximum transmission length of the peripheral channel is 65536 bytes (64KB), and supports byte/half-word/word transmission
- The maximum transmission length of Flash->RAM channel is 4096 bytes, only word transmission is supported
- Support Flash continuous programming (RAM->Flash), need to be erased in advance, one time programming is fixed at 256 bytes
- RAM pointer increment and decrement
- Can generate half-process interrupt and full-process interrupt
- The channel priority can be configured (4-level priority)

27.2 Principle of Operation

Peripheral DMA is Peripheral<>RAM channel, which uses peripheral request trigger mode for data transmission. Each peripheral channel can support peripheral->RAM or RAM->peripheral data transmission, and according to the target peripheral type Different, adaptively select byte/half-word/word transmission mode. As the Master, DMA will initiate AHB transactions to perform data operations after receiving the request. The peripheral target address is automatically located according to the channel access selection, and the RAM target address is located according to the register configuration.

Each channel can select one of multiple peripherals as the source or destination, and the software can set the channel priority. When two channels want to access RAM at the same time, the priority determines who will access first, and the other channel will be suspended Until the priority channel access is completed.

Peripheral request can be ready to send (RAM/Flash->Peripheral) or receive completion (Peripheral->RAM). The data transmission is completed through the AHB bus. When the DMA accesses the peripheral, the CPU access to the same peripheral will cause conflicts , Which Master access is suspended depends on the arbitration priority set by BusMatrix. It should be noted here that since most of the peripherals are hung on the APB bus, the mapping of APB to AHB is only one slave, so when DMA accesses any peripheral in the APB, even if the CPU accesses other peripherals under the APB, It will also cause bus arbitration. The transmission direction of each channel can be configured through the DIR register, and the software must ensure that the transmission direction configuration is consistent with the peripheral request actually mounted on this channel. For example, if the peripheral request currently mounted on channel 1 is received by UART0, the DIR register must be configured as 0 (data is read from the peripheral and written to RAM). Each time UART0 receives a frame of data, it will send a RXD0 request to the DMA After the DMA responds to the request, it reads the data from the UART0 receive buffer register. If DIR is incorrectly configured as 1, the DMA write operation to the UART0 receive buffer register will be ignored by UART0.

The software can set the memory pointer of the DMA, which is used to configure the starting address of the DMA transfer. You can choose the pointer increment or decrement mode. In addition, the TSIZE register configures the number of transmissions. According to the start address and the number of transmissions, the termination address is calculated. When the memory pointer points to the termination address, the current transmission ends and the channel is closed.

When the channel is enabled, the DMA is ready to accept the peripheral request selected by the channel. When half of the configured transmission length is transmitted, a CHHT interrupt is set; when the configured transmission length is completed, the CHHT interrupt is set. The above interrupts can be masked by the corresponding interrupt enable register.

Before the completion of a complete DMA transfer block, the software can close the channel enable at

any time. At this time, the DMA will be suspended. If the software re-enables the channel afterwards, the DMA will continue to perform the previously suspended operation. But TSIZE is still the original setting value and has not been updated, so you need to pay attention.

27.3 Block Diagram

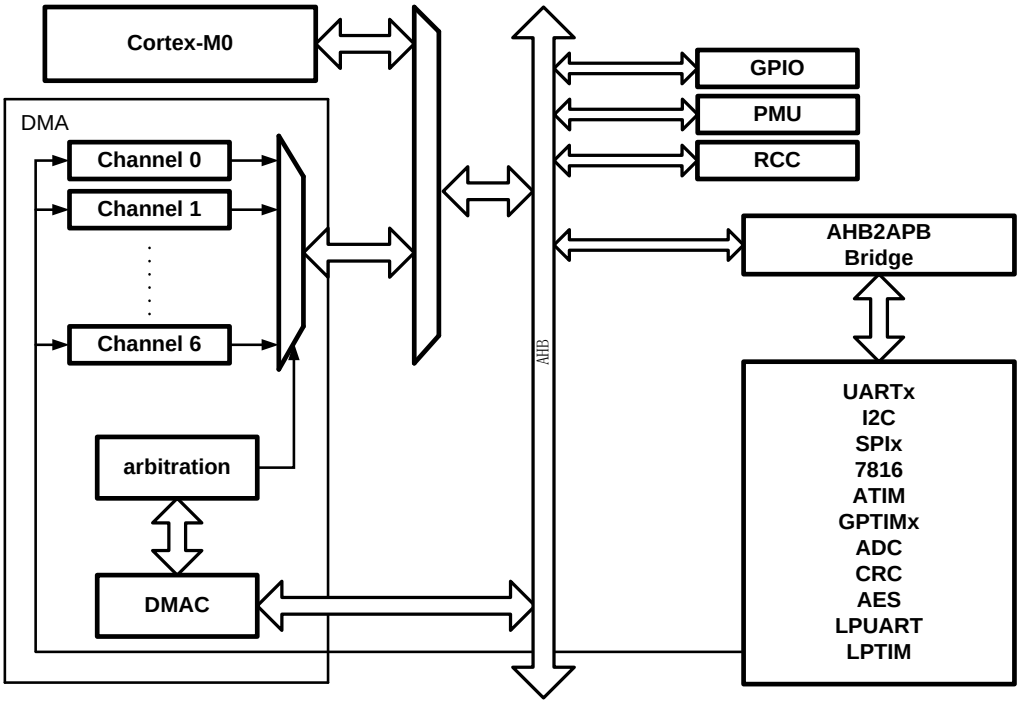


Figure 27-1 DMA Block Diagram

27.4 Workflow

DMA register configuration:

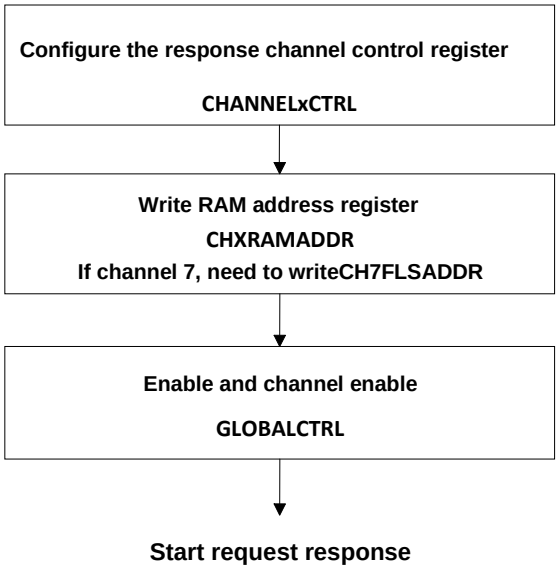


Figure 27-2 DMA Register Configuration

DMA divides the request response into two parts: the channel request processing process and the data transfer process.

- Channel request processing

  a) DMA receives the request, skip to step b

  b) Judge whether there are other channels being transported. If so, stay in step b until the other channels are completed at the same time; if not, further judge whether there are other request signals that are set at the same time. If so, judge that the current channel has priority Whether the level is higher than other channels, if yes, skip to step c and initiate a request to the data transfer process, if not, stay in step b until the other channels are completed for the current transfer

  c) And wait for the data transfer complete response signal, if you get a response, skip to step d, otherwise stop at step c

  d) Data transfer length +1, judge whether it reaches the set length, if it is, turn off the channel enable; judge whether the request is released, if yes, skip to step a, if not, stay at step d to judge that the data transmission reaches the set length , Otherwise skip to step a

- Data handling

  a) Waiting for the channel request processing process to initiate the request

  b) Read data from the source address

  c) Write the read data to the target address

d) Send a handling completion response to the channel request processing process, and skip to step a

The flow of DMA work is shown in the figure below:

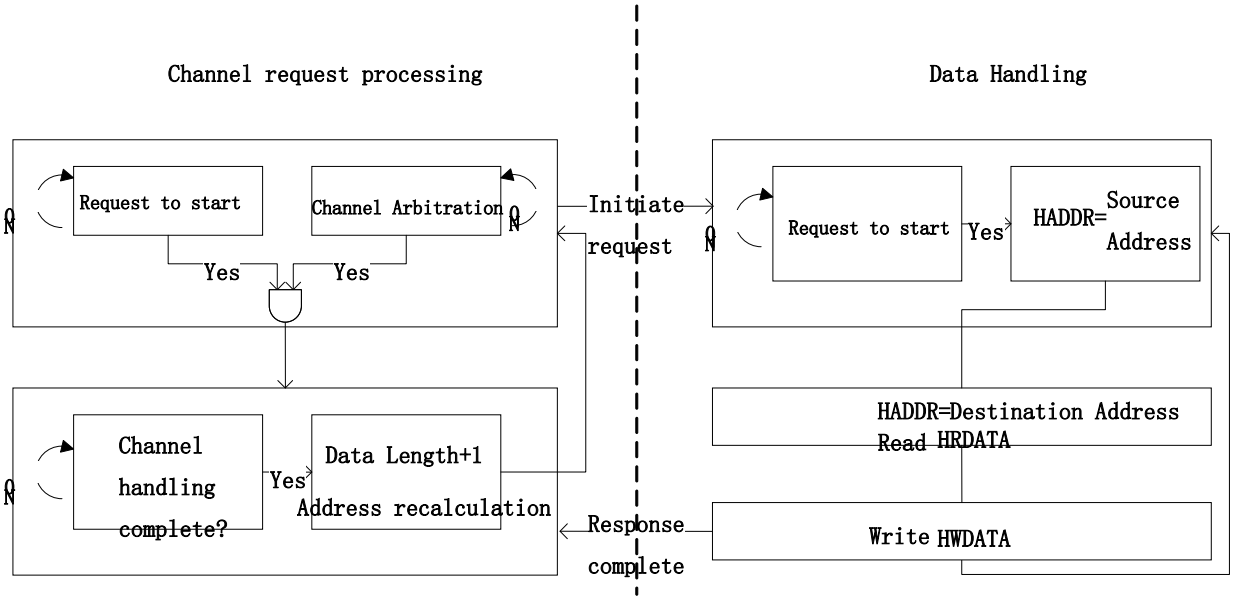


Figure 27-3 DMA Workflow

27.5 Access Bandwidth

The DMA peripheral channel supports byte/halfword/word access, and each channel can configure the transmission bandwidth through the BDW bit in the channel control register.

27.6 Channel Control

27.6.1 DMA Request Mapping

DMA has 7 priority configurable peripheral channels. Each channel can accept 8 request responses. According to the configuration register of each channel, one of the requests is selected and sent to the channel controller. The channel controller is based on the busy status and The priority is to select one of the channel requests for response processing, and the peripheral request mapping is as follows.

Number	Peripherals Request	Channel0	Channel1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6
0	***ADC***	ADC				ADC		
1	***SPI0***		SPI0_RX	SPI0_TX				
2	***SPI1***				SPI1_RX	SPI1_TX	SPI1_RX	SPI1_TX

Number	Peripherals Request	Channel0	Channel1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6
3	***SPI2***			SPI2_RX		SPI2_TX	SPI2_RX	SPI2_TX
4	***UART0***		RXD0	TXD0	RXD0	TXD0		
5	***UART1***				RXD1	TXD1	RXD1	TXD1
6	***UART3***						RXD3	TXD3
7	***UART4***			RXD4	TXD4			
8	***UART5***					RXD5		TXD5
9	***LPUART0***	LPUART0_RX	LPUART0_TX					
10	***LPUART1***	LPUART1_TX		LPUART1_RX				
11	***LPUART2***				LPUART2_RX		LPUART2_TX	
12	***U7816***						U7816RX	U7816TX
13	***I2C***					I2C_RX		I2C_TX
14	***AES***	AES_IN	AES_OUT					
15	***CRC***	CRC						
16	***ATIM***	ATIM_CH1	ATIM_CH2	ATIM_CH3	ATIM_CH4	ATIM_TRIG ATIM_COM ATIM_UEV		
17	***GTIM1***	GTIM1_CH1	GTIM1_CH2	GTIM1_CH3	GTIM1_CH4			GTIM_TRIG GTIM1_UEV
18	***GTIM2***	GTIM2_CH1	GTIM2_CH2	GTIM2_CH3	GTIM2_CH4		GTIM2_TRIG GTIM2_UEV	
19	***DAC***		DAC				DAC	
		8	8	8	8	8	8	8

Note that the ATIM_TRIG, ATIM_COM, and ATIM_UEV requests are only for the DMA Burst mode of the advanced timer, that is, when these requests come, the DMA needs to access the DMAR register of ATIM, so these three requests can be combined into one channel to complete; the same Therefore, the GTIMx_TRIG and GTIMx_UEV requests of the general timer can also be combined into one channel.

Peripheral request mapping is configured through the SSEL register. From top to bottom in the above table, the valid peripheral request signals under the conditions of SSEL=0~7 are respectively shown. For example, for channel 0, when SSEL=2, the selected peripheral request is LPUART1_TX, that is, the data transmission DMA request of LPUART1 is connected to the request input of DMA channel 0.

27.6.2 Channel Priority

DMA has 7 peripheral channels in total, and the priority level of each channel can be configured as: very high, high, medium, low through registers. When multiple channels are configured with the same priority level, the greater the channel number, the lower the priority level.

DMA will re-select the channel request every time the data is transferred. Assume that the transmission length of channel 0 is 3 and the transmission length of channel 1 is 2. When channel 0 completes the second transmission and prepares for the third data transfer, the channel 1 request response is set. At this time, the channel controller switches to channel 1 data transfer according to the channel priority, until the channel 1 data transfer is completed, the channel register Then switch back to channel 0 to complete the remaining data transfer.

27.6.3 Definition of Transmission Direction

In the DMA channel definition rules, RX means DMA reads data from peripherals and writes to RAM, and TX means DMA reads data from RAM/Flash and writes to peripherals.

After the software configures the peripheral allocation of each channel, it also needs to configure the DIR register to set the channel transmission direction. The wrong direction setting will cause the DMA to fail to work normally.

27.6.4 Loop Mode

Peripheral DMA channel supports circular mode (Circular mode). In the loop mode, when the transfer length defined by the TSIZE register is completed, the DMA will not automatically stop, but will return to the starting address defined by the RAM pointer register to continue the transfer. DMA's half-range interrupt and full-range interrupt will still be set normally, DMA will not terminate the transmission until the software closes the channel.

The cyclic mode is enabled by setting the CHxCR.CIRC register.

The storage DMA channel does not support circular mode.

27.7 Register

Offset	Name	Symbol
DMA(Base address: 0x40000400)		
0x00	DMA Global Control Register	DMA_GCR
0x04	Channel 0 Control Register	DMA_CH0CR
0x08	Channel 0 Memory Address Register	DMA_CH0MAR
0x0C	Channel 1 Control Register	DMA_CH1CR
0x10	Channel 1 Memory Address Register	DMA_CH1MAR
0x14	Channel 2 Control Register	DMA_CH2CR
0x18	Channel 2 Memory Address Register	DMA_CH2MAR
0x1C	Channel 3 Control Register	DMA_CH3CR
0x20	Channel 3 Memory Address Register	DMA_CH3MAR
0x24	Channel 4 Control Register	DMA_CH4CR
0x28	Channel 4 Memory Address Register	DMA_CH4MAR
0x2C	Channel 5 Control Register	DMA_CH5CR
0x30	Channel 5 Memory Address Register	DMA_CH5MAR
0x34	Channel 6 Control Register	DMA_CH6CR
0x38	Channel 6 Memory Address Register	DMA_CH6MAD
0x3C	Channel 7 Control Register	DMA_CH7CR
0x40	Channel 7 Flash Address Register	DMA_CH7FLSAD
0x44	Channel 7 RAM Address Register	DMA_CH7RAMAD
0x48	DMA Interrupt Status Register	DMA_ISR

27.7.1 DMA Global Control Register (DMA_GCR)

NAME	DMA_GCR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						ADDRERR_IE	DMAEN
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
1	**ADDRERR_IE**	DMA error address interrupt enable 1: Error address interrupt enable 0: Error address interrupt disable
0	**DMAEN**	DMA enable 1: DMA enable 0: DMA disable

27.7.2 Channel x Control Register (DMA_CHxCR)

NAME	DMA_CHxCR (x=0~6)							
Offset	x*0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	TSIZE							
access	R/W-00000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	TSIZE							
access	R/W-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-		PRI		INC	SSEL		
access	U-0		R/W-00		R/W-0	R/W-000		
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	DIR	BDW		CIRC	FTIE	HTIE	EN
access	U-0	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:16	**TSIZE**	Channelx transfer length, 1-65336 transmissions
15:14	--	RFU: **Reserved, read as 0**
13:12	**PRI**	Channelx priority 00: Low 01: Medium 10: High 11: Very High
11	**INC**	RAM address increase and decrease settings 1: RAM address Increment 0: RAM address decrement
10:8	**SSEL**	Channelx peripheral request mapping Each channel can accept 8 peripheral requests. For the mapping of peripheral requests, see 24.6.1 DMA request mapping
7	--	RFU: **Reserved, read as 0**
6	**DIR**	Channel transmission direction 0: Read data from peripheral to RAM

bit	name	functional description
		1: Read data from RAM and write to peripheral
5:4	**BDW**	Peripheral size 00: 8bit 01: 16bit 10: 32bit 11: RFU
3	**CHxCIRC**	Circular mode 0: Disable Circular mode 1: Enable Circular mode
2	**FTIE**	Channelx Transfer complete interrupt enable 1: Transfer complete interrupt enable 0: Transfer complete interrupt disable
1	**HTIE**	Channelx Half transfer complete interrupt enable 1: Half transfer complete interrupt enable 0: Half transfer complete interrupt disable
0	**EN**	Channelx enable 1: Enable channel 0: Disable channel

27.7.3 Channel x Memory Address Register (DMA_CHxMAR)

NAME	**DMA_CHxMAR (x=0~6)**							
Offset	x*0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	MEMAD[31:24]							
access	R/W-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	MEMAD[23:16]							
access	R/W-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	MEMAD[15:8]							
access	R/W-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	MEMAD[7:0]							
access	R/W-00000000							

bit	name	functional description
31:0	**MEMAD**	Channelx memory pointer address, the software writes the memory target address to this register before the DMA transfer is started. DMA access will trigger a hardfault when the pointer points to a null address

bit	name	functional description
		When the pointer points to Flash, writing data to Flash is prohibited. The software can query the destination memory address of the current DMA transfer.

27.7.4 Channel 7 Control Register (DMA_CH7CR)

NAME	DMA_CH7CR							
Offset	0x3C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-				TSIZE			
access	U-0				R/W-00000			
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	TSIZE							
access	R/W-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-		PRI		-	DIR	RI	FI
access	U-0		R/W-00		U-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					FTIE	HTIE	EN
access	U-0					R/W-0	R/W-0	R/W-0

bit	name	functional description
31:28	**--**	RFU: **Reserved, read as 0**
27:16	**SIZE**	Channel7 transfer length, 1-8192 transfers, only valid for Flash->RAM transfers, RAM->Flash transfers are fixed length 64 transfers
15:14	**--**	RFU: **Reserved, read as 0**
13:12	**PRI**	Channel7 priority 00: Low 01: Medium 10: High 11: Very High
11	**--**	RFU: **Reserved, read as 0**
10	**DIR**	Channel7 transmission direction 1: Flash->RAM transfer 0: RAM->Flash transfer
9	**RI**	Channel7 RAM address increment/decrement setting, valid only in Flash->RAM transfer 1: RAM address Increment 0: RAM address decrement
8	**FI**	Channel7 Flash address increment/decrement setting, valid only

bit	name	functional description
		in Flash->RAM transfer 1: Flash address Increment 0: Flash address decrement
7:3	--	RFU: **Reserved, read as 0**
2	**FTIE**	Channel7 Transfer complete interrupt enable Transfer complete interrupt enable 1: Transfer complete interrupt enable 0: Transfer complete interrupt disable
1	**HTIE**	Channel7 Half transfer complete interrupt enable 1: Half transfer complete interrupt enable 0: Half transfer complete interrupt disable
0	**EN**	Channel7 enable 1: Enable channel 0: Disable channel

27.7.5 Channel 7 Flash Address Register (DMA_CH7FLSAD)

NAME	DMA_CH7FLSAD							
Offset	0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-	FLSAD[14:8]						
access	U-0	R/W-00000000						
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	FLSAD[7:0]							
access	R/W-00000000							

bit	name	functional description
31:15	--	RFU: **Reserved, read as 0**
14:0	**FLSAD**	Channel7 Flash pointer address, the software writes Flash target address to this register before DMA transfer starts, after DMA starts, this register is incremental or decremental with DMA transfer Software can query the target Flash address of the current DMA transfer The low bit of this register (bit5-0) is valid only in Flash->RAM transfer, and the half-sector starting address of Flash is aligned by

bit	name	functional description
		default in RAM->Flash transfer.

27.7.6 Channel 7 RAM Address Register (DMA_CH7RAMAD)

NAME	DMA_CH7RAMAD							
Offset	0x44							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				RAMAD[13:8]			
access	U-0				R/W-00000000			
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RAMAD[7:0]							
access	R/W-00000000							

bit	name	functional description
31:12	**--**	RFU: **Reserved, read as 0**
11:0	**RAMAD**	Channel7 RAM word pointer address, the software writes the RAM target address (word address) to this register before the DMA transfer starts, after the DMA starts this register is incremental or decremental with the DMA transfer Software can query the current DMA transfer target RAM address

27.7.7 DMA Interrupt Status Register (DMA_ISR)

NAME	DMA_ISR							
Offset	0x48							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							ADDR ERR
access	U-0							R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CHFT[7:0]							
access	R/W-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

name	CHHT[7:0]
access	R/W-00000000

bit	name	functional description
31:17	--	RFU: **Reserved, read as 0**
16	**ADDRERR**	DMA transfer address error flag, set when the memory pointer exceeds the legal address range of RAM and Flash
15:8	**CHFT[7:0]**	DMA channel x transfer completion flag，This flag is set by hardware. It is cleared by software by writing 1 to this bit. 1: Corresponding channel transmission completed 0: Corresponding channel transmission is not completed
7:0	**CHHT[7:0]**	DMA channel x transfer halfway flag，This flag is set by hardware. It is cleared by software by writing 1 to this bit.

28 Cyclic Redundancy Check Calculation Unit (CRC)

28.1 Introduction

Cyclic Redundancy Check is the most commonly used checksum method for computer and instrument data communication. The CRC calculation unit in FM33A0xxEV is a completely independent module.CRC calculation and verification can be carried out for data communication such as 7816, I2C, UART and SPI through software control.

In addition, CRC can also perform Flash content integrity verification. With DMA, you can compute the CRC result of program content in Flash in real time and generate an integrity signature that is stored with the program in Flash. By verifying this CRC signature, you can verify whether the Flash content is correct and complete.

- Uses fully programmable polynomial with programmable size(7,8,16,32bits)
- Programmable CRC initial value
- CRC fast algorithm: 8bit CRC operation was completed in 1 clock cycle, and 32bit CRC operation was completed in 4 clock cycles
- Supports automatic input/output data order adjustment (byte, half-word, or full-word)
- Supports XOR for output results

28.2 Operation Flow

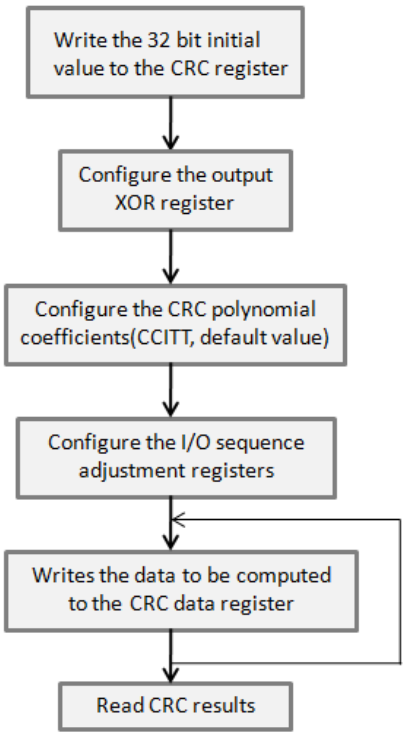


Figure 28-1 CRC Operation Flow

The CRC configuration and calculation procedure are as follows:

- Configure the initial value of the shift register with a range of 0x0000_0000~0xFFFF_FFFF.
- Configure the output XOR register CRC_XOR
- The software needs to configure input REFLECTIN enable, output REFLECTOUT enable and XOROUT enable.
- The software writes data into the data register (CRC_DR) and then automatically computes successive shifts.
- After the computation, the result is written back to the data register, and the software determines whether the result can be retrieved according to the BUSY bit
- If the polynomial is 7bit, the result is CRC_DR[6:0]; if the polynomial is 8bit, the result is CRC[7:0]; if the polynomial is 16bit, the result is CRC_DR[15:0]; if the polynomial is 32bit, the result is CRC_DR[31:0]; if the polynomial is 3bit, the result is CRC_DR[31:0].
- After the calculation of the previous CRC, the result of the previous CRC will be retained in the data register as the initial value of the shift register of the subsequent data. After the CRC calculation is triggered several times in a row, the software finally reads the CRC value of the cumulative calculated result.

28.3 Golden Data

Golden data are available for testing and validation.

Polynomial	Input	Initial value(hexadecimal)		
		All 0	All F	6363
		CRC result（hexadecimal）		
CRC-8	5A5A	0F	D8	C5
	1223344	F9	28	96
CRC-16	5A5A	5DD9	DDD4	9696
	11223344	7D35	7D11	4698
CRC-CCITT	5A5A	1ACB	07C4	1877
	11223344	DD33	59F3	DD06

28.4 DMA Interface

The channel between CRC and DMA is unidirectional (RAM->CRC).The CRC module can read and verify RAM data through the DMA module, and its workflow is shown in the figure. The CRC made a request to DMA, which received the request, read RAM and wrote the data to the CRCDR register of the CRC module. After receiving the data, the CRC module cancels the DMA request and starts to calculate the verification value. After the verification is completed, the CRC module resets the DMA request.

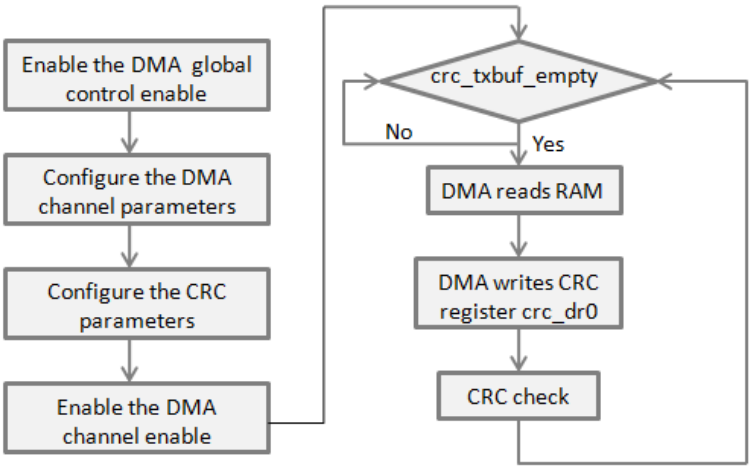


Figure 28-2 CRC the Data in RAM by DMA

28.5 Flash Data Integrity Check

By using DMA, CRC can be used to check the integrity of the Flash content. Select the CRC channel of the DMA, point the DMA address pointer to the Flash address, configure the DMA length, and start the DMA to realize the continuous transfer of the Flash content to the CRC module for CRC. After calculation, the software can read the CRC result after the DMA transfer is completed, and compare it with the expected value to confirm whether the Flash content is correct and complete.

28.6 Register

Offset	Name	Symbol
CRC(Base address: 0x40010000)		
0x00	CRC Data Register	CRC_DR
0x04	CRC Control Register	CRC_CR
0x08	CRC Linear Feedback Shift Register	CRC_LFSR
0x0C	CRC Output XOR Register	CRC_XOR
0x1C	CRC Polynomial Register	CRC_POLY

28.6.1 CRC Data Register (CRC_DR)

NAME	CRC_DR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	DR[31:24]							
access	R/W-11111111							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DR[23:16]							
access	R/W-11111111							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DR[15:8]							
access	R/W-11111111							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DR[7:0]							
access	R/W-11111111							

bit	name	functional description
31:0	**DR**	CRCDR is used as a data input register and saves the CRC calculation result after the operation is completed. As input: if Flash CRC calculation or word operation is enabled, CRCDR[31:0] is calculated, a total of 4 byte operations (from low to high); otherwise, CRCDR[7:0] is calculated, a total of 1 byte operation. When saving the result: if it is a 7-bit polynomial result, save it in CRCDR[6:0], if it is an 8-bit polynomial result, save it in CRCDR[7:0], if it is a 16-bit polynomial result, save it in CRCDR[15:0], if The result of the 32-bit polynomial is stored in CRCDR[31:0].

28.6.2 CRC Control Register (CRC_CR)

NAME	CRC_CR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						OPWD	PARA
access	U-0						R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RFLTIN		RFLTO	RES	BUSY	XOR	SEL	
access	R/W-00		R/W-0	R-0	R-0	R/W-0	R/W-10	

bit	name	functional description
31:10	--	RFU: **Reserved, read as 0**
9	**OPWD**	Operation by Word 0: Byte operation, CRC calculation only for CRCDR the lowest byte 1: Word operation, CRC calculation is performed for all 4 bytes of CRCDR
8	**PARA**	CRC Parallel Calculation 0: Serial operation. It takes 8 clock cycles to compute 1 byte 1: Parallel computing. It takes 1 clock cycle to compute 1 byte
7:6	**RFLTIN**	CRC Reflected Input 00: Input unreflected 01: Input reflected by byte 10: Input reflected by half-word 11: Input reflected by word For example, the calculated data is 0x11223344, If RFLTIN==01, the data is changed to 0x8844CC22 and then calculated If RFLTIN==10, the data is changed to 0x448822CC before calculation If RFLTIN==11, the data is changed to 0x22CC4488 and then calculated
5	**RFLTO**	CRC Reflected Output 0: Output unreflected 1: Output reflected by byte Such as:

bit	name	functional description
		If RFLTO==1, the output result is 0x2C48 if the currently computed CRC result is 0x1234 If RFLTO==0, output 0x1234 directly Note: This result is not the final output. It needs to see if XOR is 1. See bit2 for details
4	**RES**	CRC Result Flag, read only 0: The CRC result is 0 1: CRC results were not all 0
3	**BUSY**	CRC Busy Flag, read only 0: CRC operation ends 1: CRC operation is in progress
2	**XOR**	Output XORed with CRC_XOR register enable 0: Output no xor CRC_XOR register 1: Output xor CRC_XOR register
1:0	**SEL**	CRC Polynomial width Selection 00: 32 bit 01: 16 bit 10: 8 bit 11: 7 bit

28.6.3 CRC Linear Feedback Shift Register (CRC_LFSR)

NAME	**CRC_LFSR**							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	LFSR[31:24]							
access	R/W-11111111							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	LFSR[23:16]							
access	R/W-11111111							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	LFSR[15:8]							
access	R/W-11111111							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	LFSR[7:0]							
access	R/W-11111111							

bit	name	functional description
31:0	**LFSR**	CRC Linear Feedback Shift Register The CRC initial value can be written by the software before the operation begins

28.6.4 CRC Output XOR Register (CRC_XOR)

NAME	CRC_XOR							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	XOR[31:24]							
access	R/W-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	XOR[23:16]							
access	R/W-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	XOR[15:8]							
access	R/W-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	XOR[7:0]							
access	R/W-00000000							

bit	name	functional description
31:0	**XOR**	CRC exclusive XOR register When CRC_CR.xor is 1, the CRC result will XOR the register data before output.

28.6.5 CRC Polynomial Register (CRC_POLY)

NAME	CRC_POLY							
Offset	0x1C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	POLY[31:24]							
access	R/W-00000000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	POLY[23:16]							
access	R/W-00000000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	POLY[15:8]							
access	R/W-00010000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	POLY[7:0]							
access	R/W-00100001							

bit	name	functional description
31:0	**POLY**	CRC Polynomials

29 Advanced-Control Timer (ATIM)

29.1 Introduction

FM33LG0 contains one advanced-control timer.

The advanced-control timer consists of a 16-bit auto-reload counter driven by a programmable prescaler.

It may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare, PWM, complementary PWM with dead-time insertion).

29.2 Features

- 16-bit up, down, up/down auto-reload counter
- 16-bit programmable prescaler allowing dividing，Supports real-time adjustment of counting clock frequency division
- 4 independent channels for Input Capture, Output Compare, PWM generation and One-pulse mode output
- Complementary outputs with programmable dead-time
- Repetition counter, supports timer to update status after multiple cycles
- Two brake input pins, comparator to brake, SVD to brake, brake signal supports filtering, polarity selection and combination configuration
- Interrupt/DMA generation on the following events:

  – Update: counter overflow/underflow, counter initialization (by software or internal/external trigger)

  – Trigger event (counter start, stop, initialization or count by internal/external trigger)

  – Input capture

  – Output compare

  – Break input
- Supports incremental (quadrature) encoder and hall-sensor circuitry for positioning purposes

Terms：

- OCxREF signal high is called the active, low is called the inactive
- IDLE mode: Relative to the operating mode, it refers to when a braking event occurs MOE=0
- Output disables: GPIO output is disable, not driven by TIMER
- Off-state: GPIO output is enable, but TIMER output is inactive
- Inactive state: One or two of the complementary channels output is inactive

29.3 Block Diagram

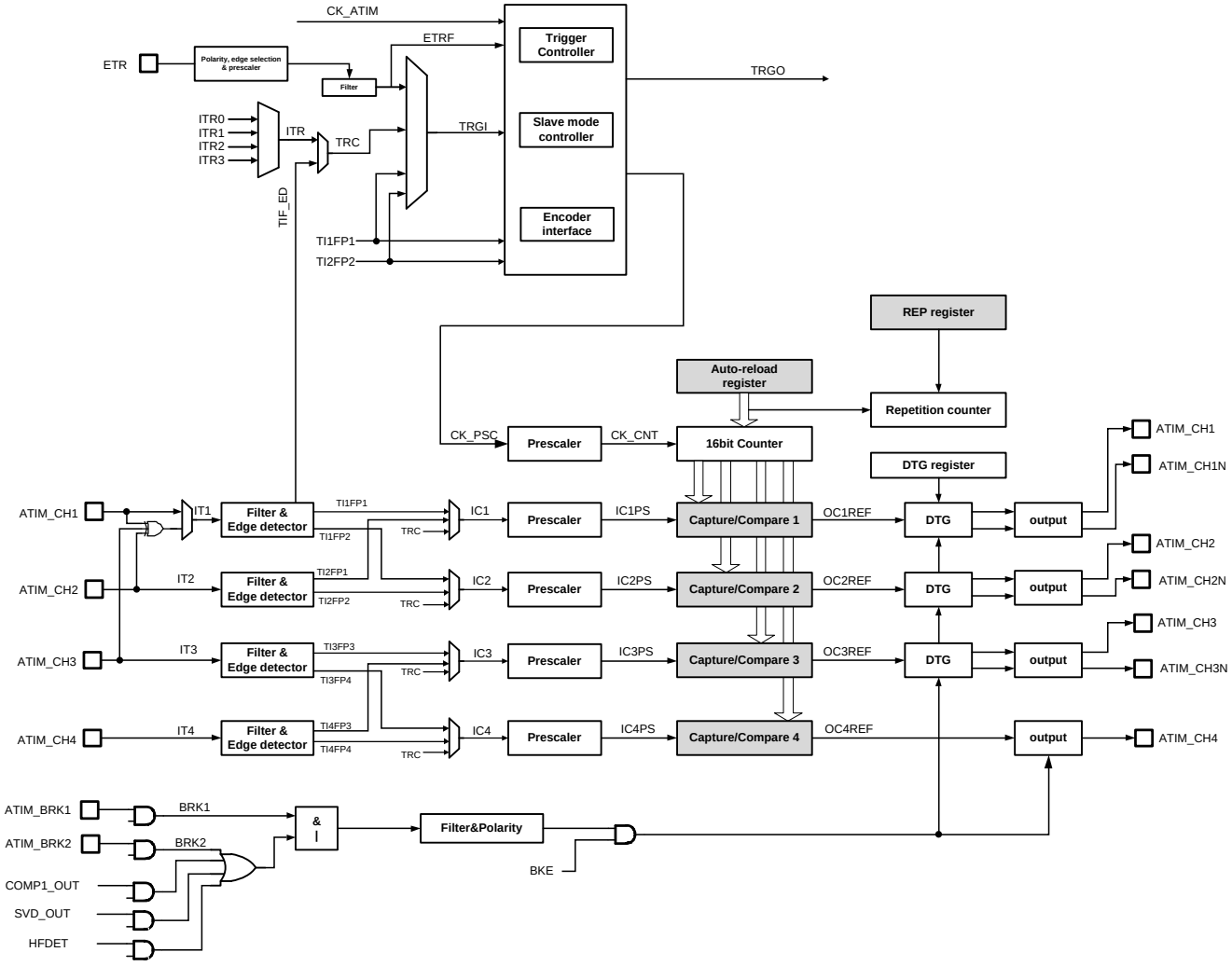


Figure 29-1 Advanced-Control Timer Block Diagram

29.4 Functional Description

29.4.1 Time-Base Unit

The main block of the programmable advanced-control timer is a 16-bit counter with itsrelated auto-reload register. The counter can count up, down or both up and down. Thecounter clock can be divided by AHBCLK.

The counter, the auto-reload register and the prescaler register can be written or read bysoftware. This is true even when the counter is running.

The time-base unit includes:

- Counter register (ATIM_CNT)
- Prescaler register (ATIM_PSC)
- Auto-reload register (ATIM_ARR)
- Repetition counter register (ATIM_RCR)

ARR can be preloaded, depending on ARPE (Auto Reload Preload Enable) register. When ARPE=0, the data write to ARR register is transferred into the shadow register permanently. When ARPE=1, the data write to ARR register is transferred into the shadow register at each update event (UEV). The update event (UEV)can also be generated by software to load ARR.

ATIM_CNT is clocked by the prescaler output CK_CNT, which is enabled only when the counter enable bit (CEN) is set. When CNT=ARR, this round of counting is end and send update event.

The prescaler can divide the counter clock frequency by any factor between 1 and 65536. It is based on a 16-bit counter controlled through a 16-bit register (in the TIMx_PSC register).It can be changed on the fly as this control register is buffered. The new prescaler ratio is taken into account at the next update event.

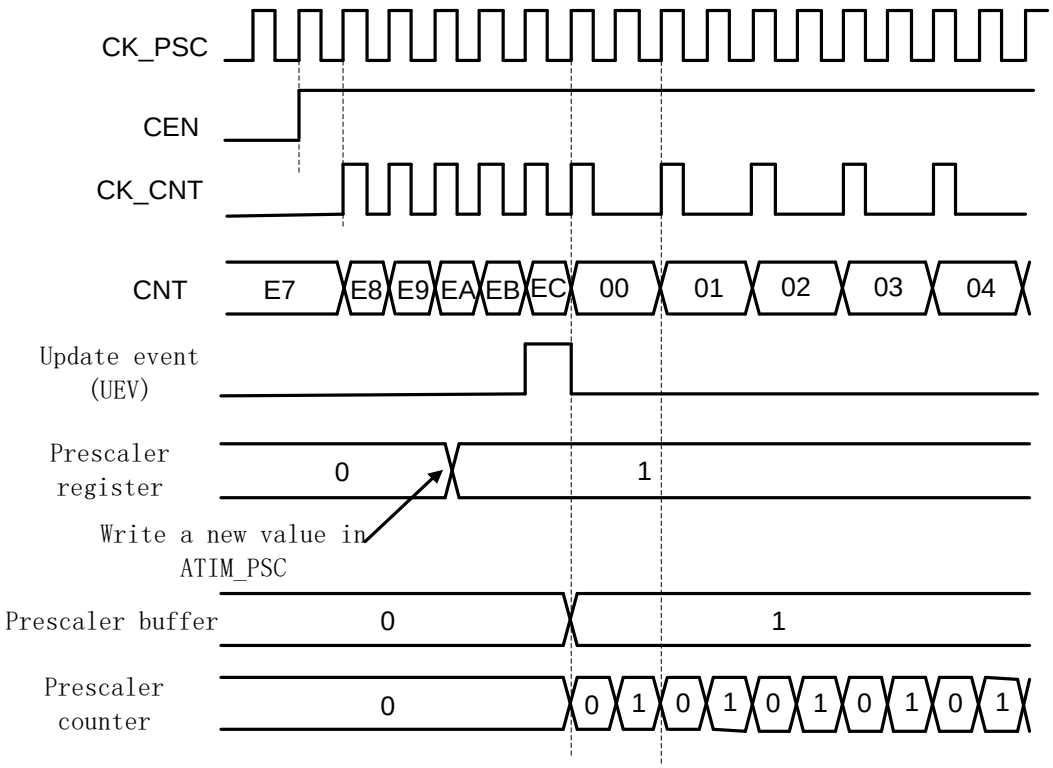


Figure 29-2 Counter Timing Diagram with Prescaler Division Change from 1 to 2

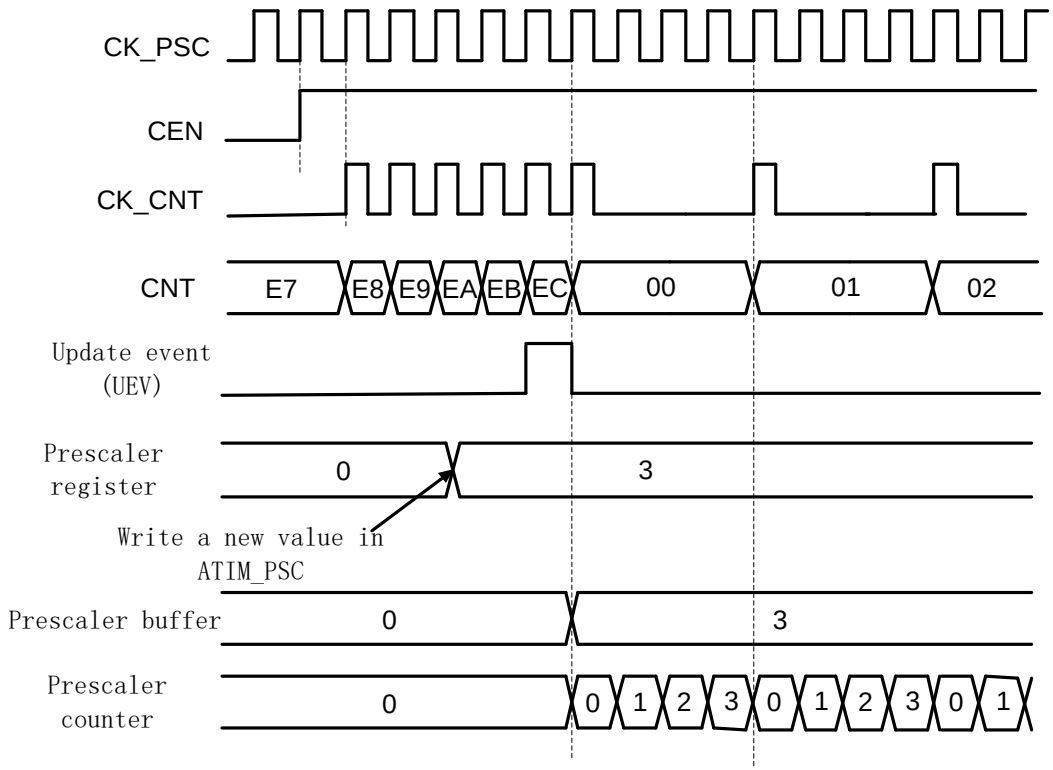


Figure 29-3 Counter Timing Diagram with Prescaler Division Change from 1 to 4

29.4.2 Counter Modes

Supports upcounting mode, downcounting mode and center-aligned mode.

Upcounting mode

In downcounting mode, the counter counts from the auto-reload value (content of the TIMx_ARR register) down to 0, then restarts from the auto-reload value and generates acounter underflow event.

If the repetition counter is used, the update event (UEV) is generated after downcounting is repeated for the number of times programmed in the repetition counter register(TIMx_RCR) + 1.

Software can also setting the UG bit to generate update event, the counts and counter of the prescaler restarts from 0. If the URS bit (update request selection) in TIMx_CR1 register is set, setting the UG bit generates an update event UEV but without setting the UIF flag.

Setting UDIS register can disable update event, to avoid updating the shadow registers while writing new values in the preload registers.

When an update event occurs, all the registers are updated and the update flag:

- The repetition counter is reloaded with the content of ATIM_RCR register
- The auto-reload active register is updated with the preload value (content of the ATIM_ARR register).
- The buffer of the prescaler is reloaded with the preload value (content of the ATIM_PSC register)

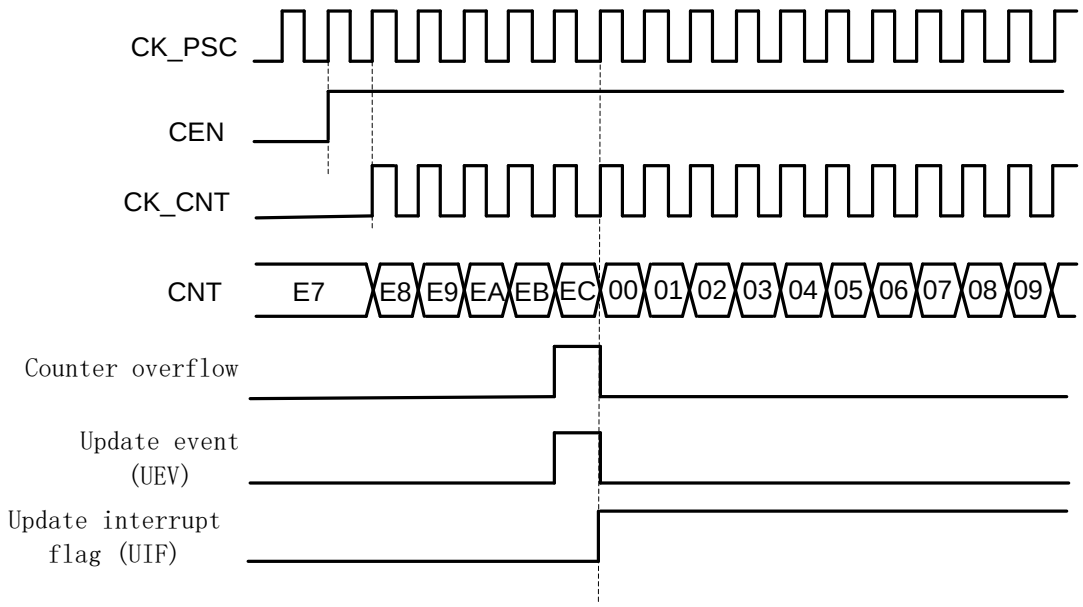


Figure 29-4 Counter Timing Diagram, Internal Clock is Not Divided

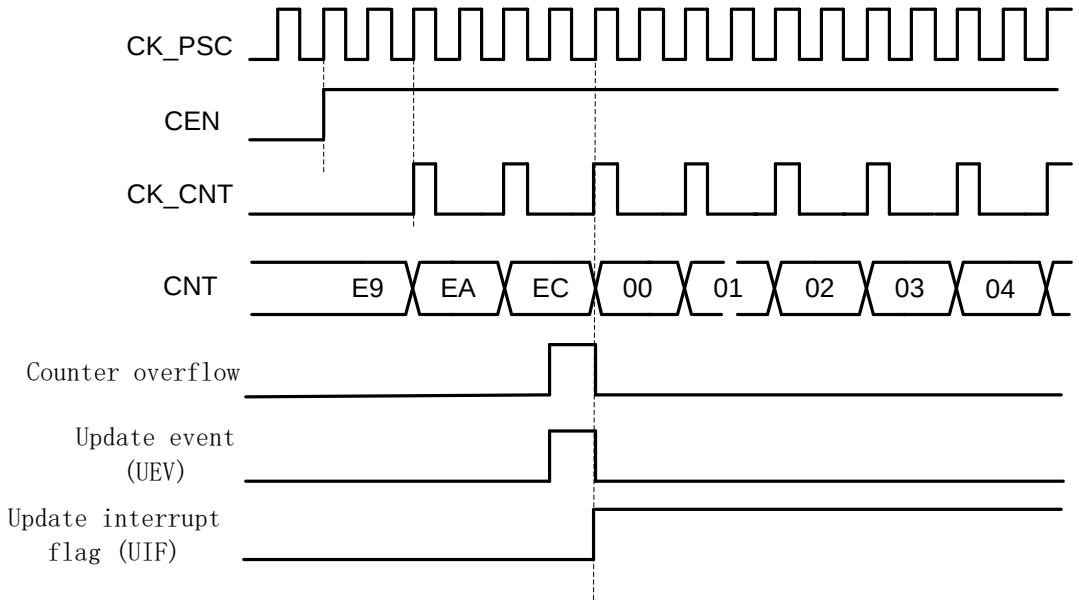


Figure29-5 Counter Timing Diagram, Internal Clock Divided by 2

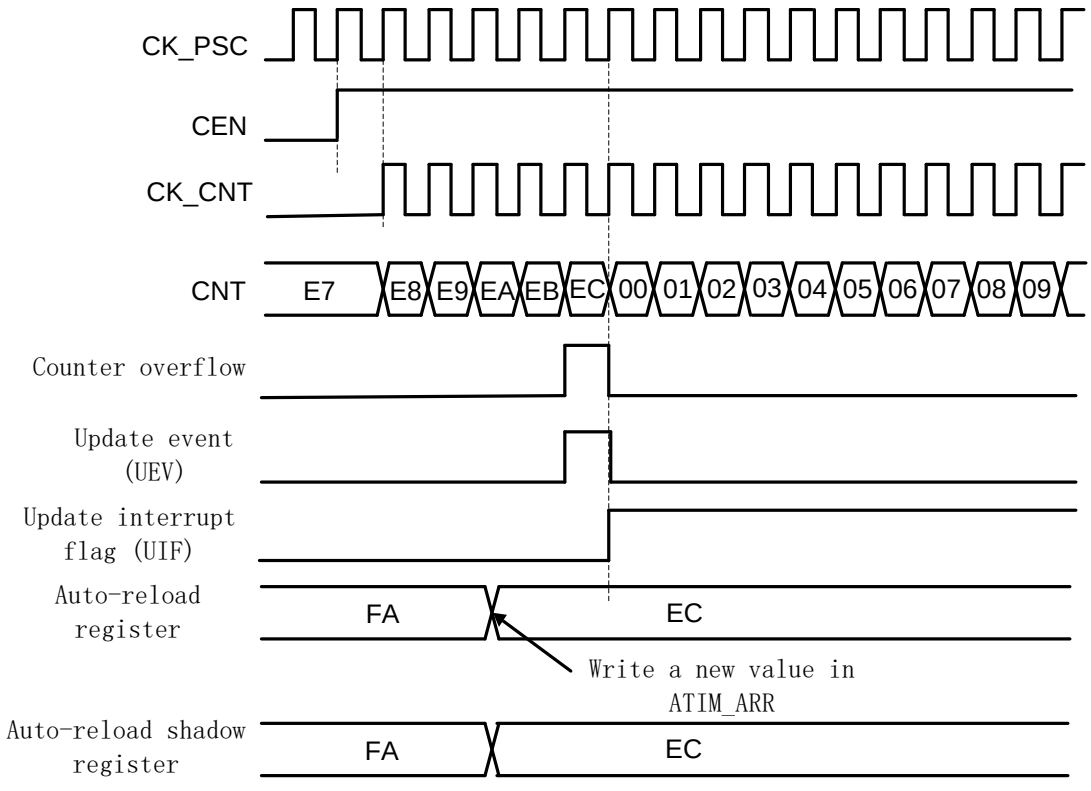


Figure 29-6 ARPE=0 (ATIM_ARR is Not Preloaded) Update Event

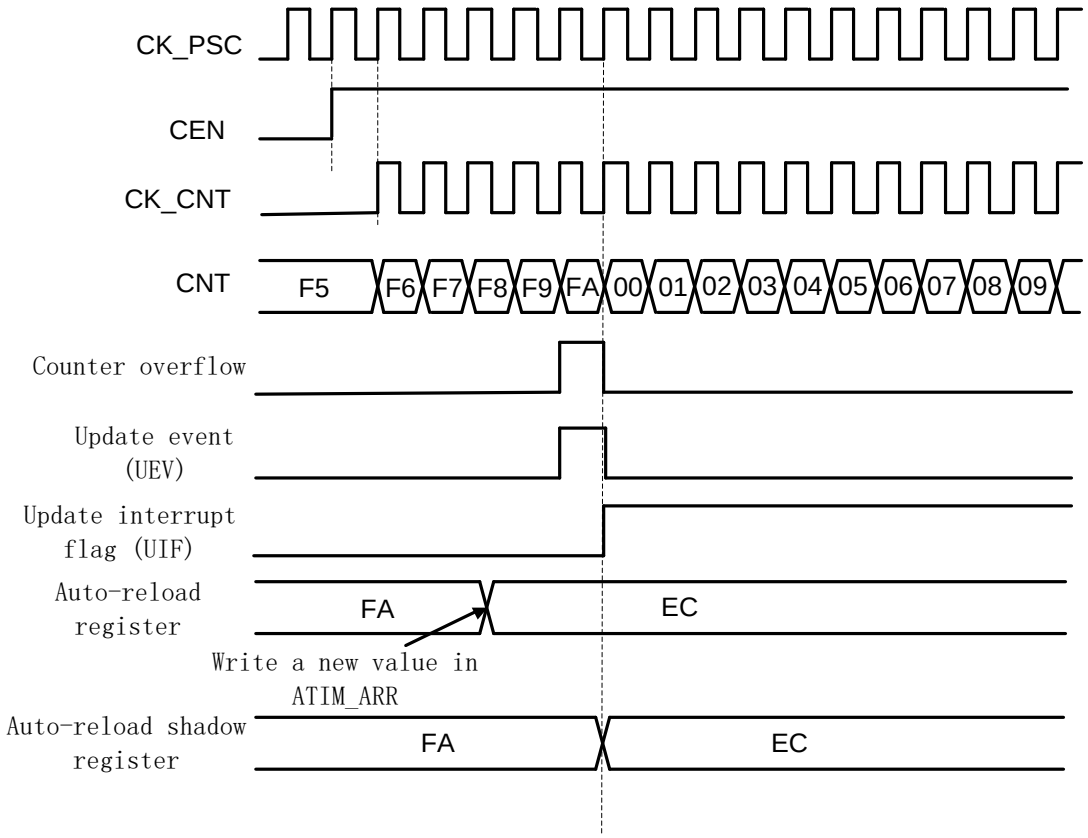


Figure 29-7 ARPE=1 (ATIM_ARR is Preloaded) Update Event

Downcounting mode

In downcounting mode, the counter counts from the auto-reload value (content of the TIMx_ARR register) down to 0, then restarts from the auto-reload value and generates acounter underflow event.

Software can also setting the UG bit to generate update event，the counts and counter of the prescaler restarts from 0。if the URS bit (update request selection) in TIMx_CR1 register is set, setting the UG bit generates an update event UEV but without setting the UIF flag.

Setting UDIS register can disable update event，to avoid updating the shadow registers while writing new values in the preload registers.

When an update event occurs, all the registers are updated and the update flag:

- The repetition counter is reloaded with the content of ATIM_RCR register
- The auto-reload active register is updated with the preload value (content of the ATIM_ARR register).
- The buffer of the prescaler is reloaded with the preload value (content of the ATIM_PSC register)

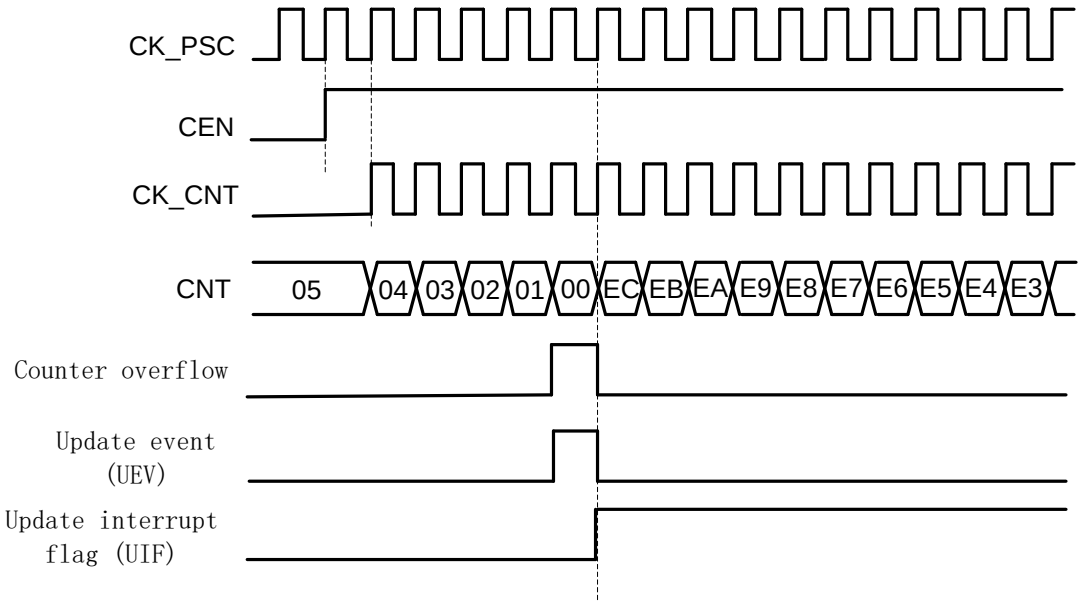


Figure 29-8 Downcounting Mode, Internal Clock is not Divided

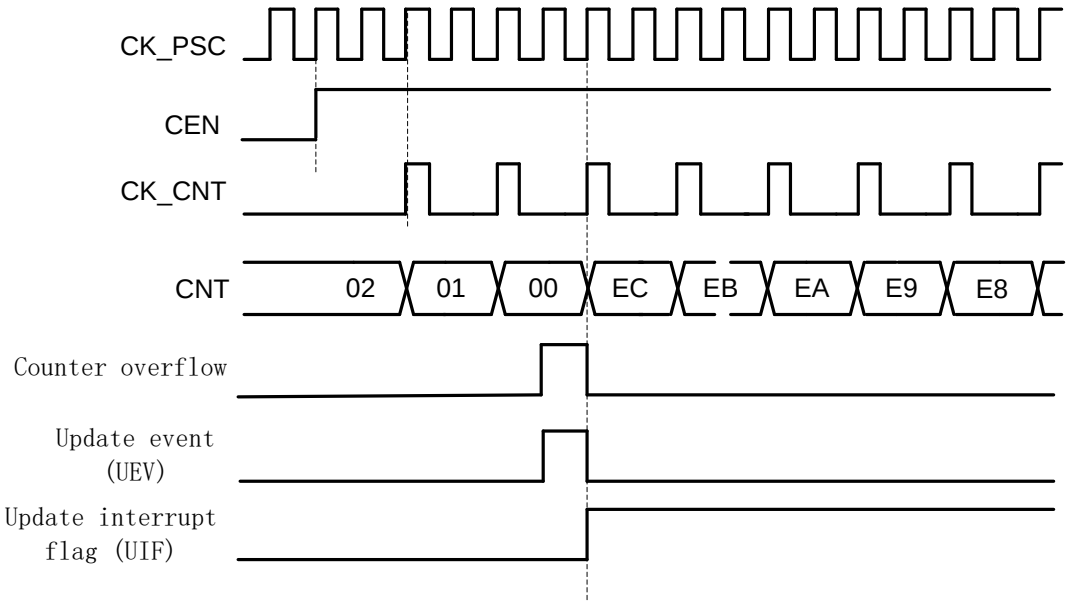


Figure 29-9 Downcounting Mode, Internal Clock Divided by 2

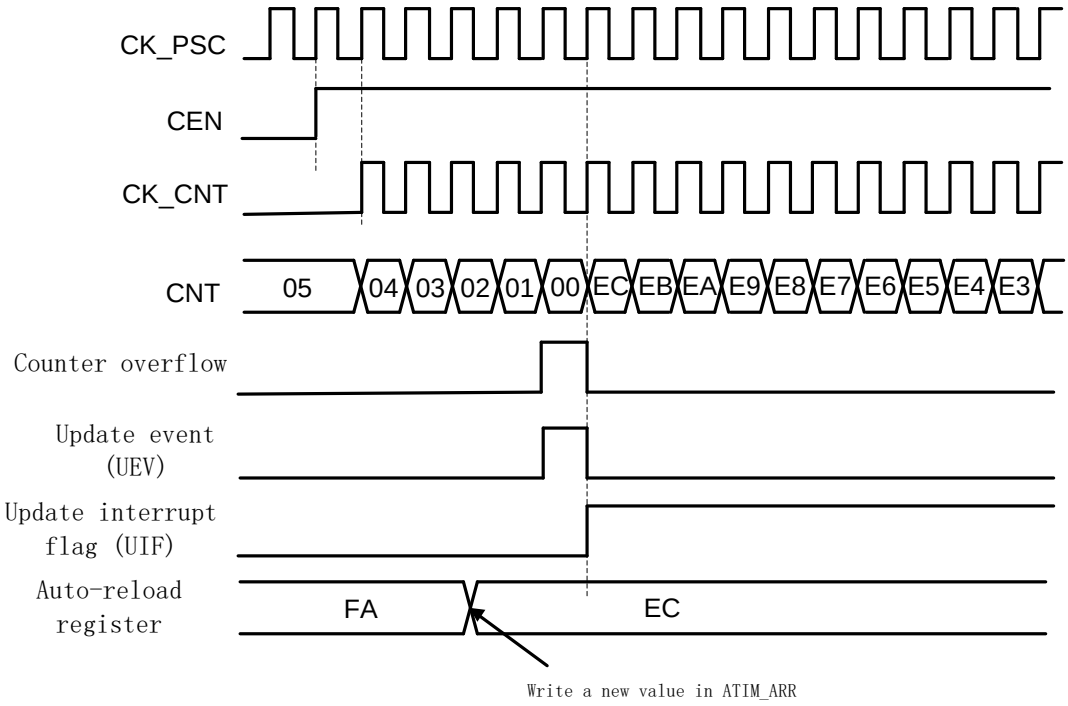


Figure 29-10 Downcounting Mode, Internal Clock Divided by 2

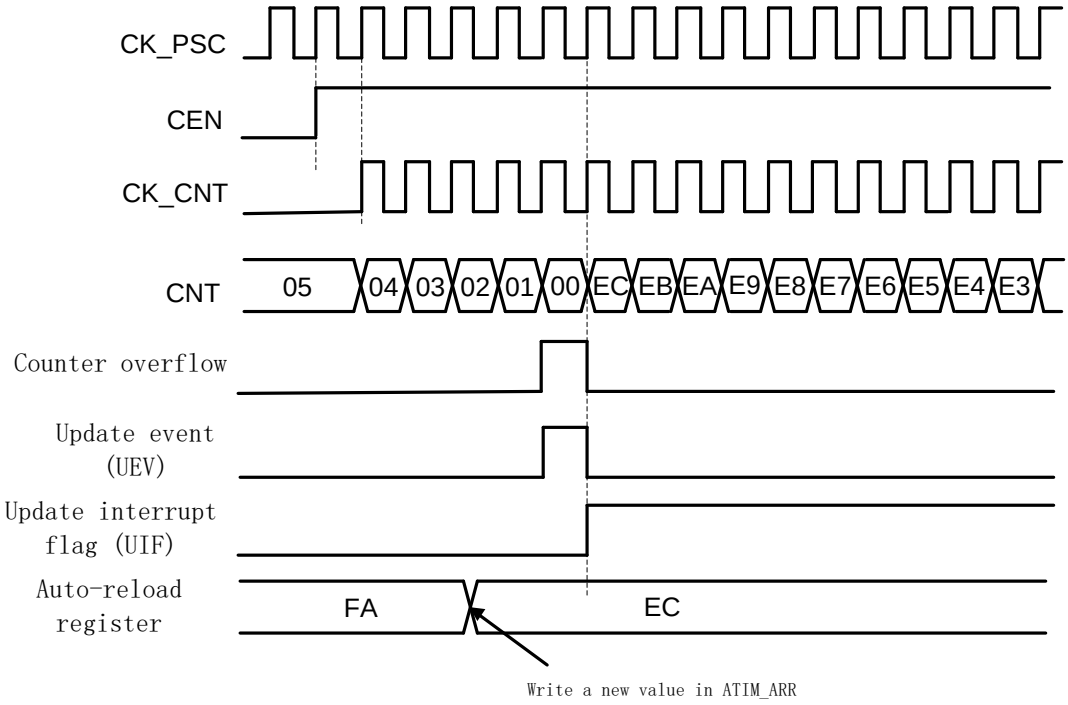


Figure 29-11 Downcounting Mode, ATIM_ARR is Not Preloaded

Center-aligned mode

In center-aligned mode, the counter counts from 0 to the auto-reload value (content of the TIMx_ARR register) – 1, generates a counter overflow event, then counts from the autoreload value down to 1 and generates a counter underflow event. Then it restarts counting from 0.

Center-aligned mode is active when the CMS register are not equal to'00'. The Output compare interrupt flag of channels configured in output is set when: the counter counts down (Center aligned mode 1, CMS = "01"), the counter counts up (Centeraligned mode 2, CMS = "10") the counter counts up and down (Center aligned mode 3,CMS = "11").In this mode, the DIR direction register cannot be written. It is updated by hardware and gives the current direction of the counter.

When an update event occurs at overflow and underflow, the shadow register of ARR, PSC and RCR is updated.

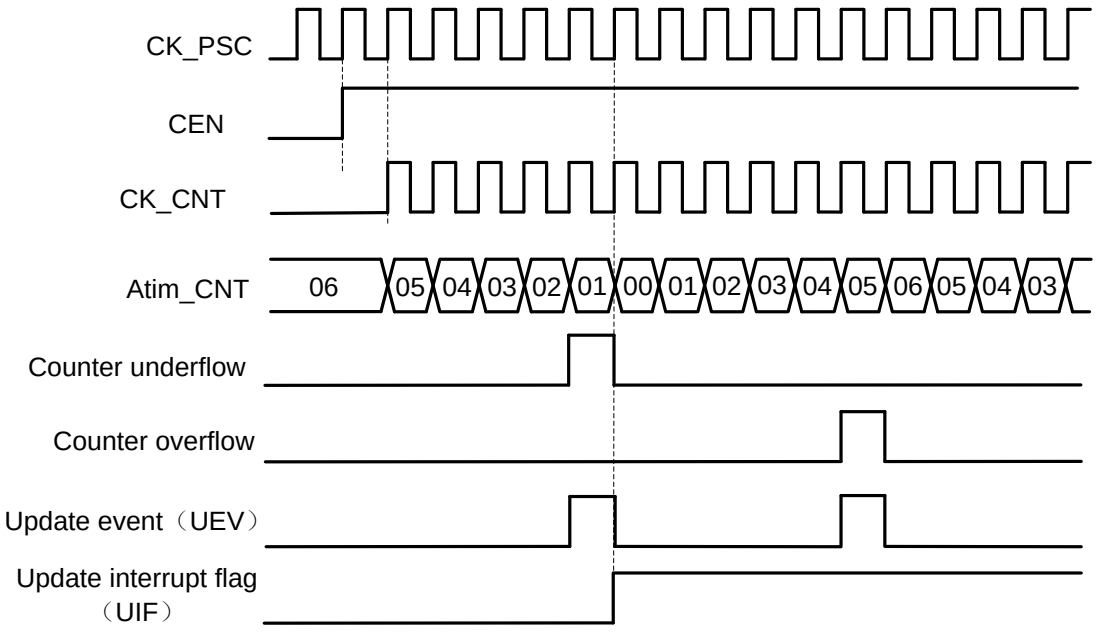


Figure 29-12 Counter Timing Diagram, ATIM_PCS=0, ATIM_ARR=0x6

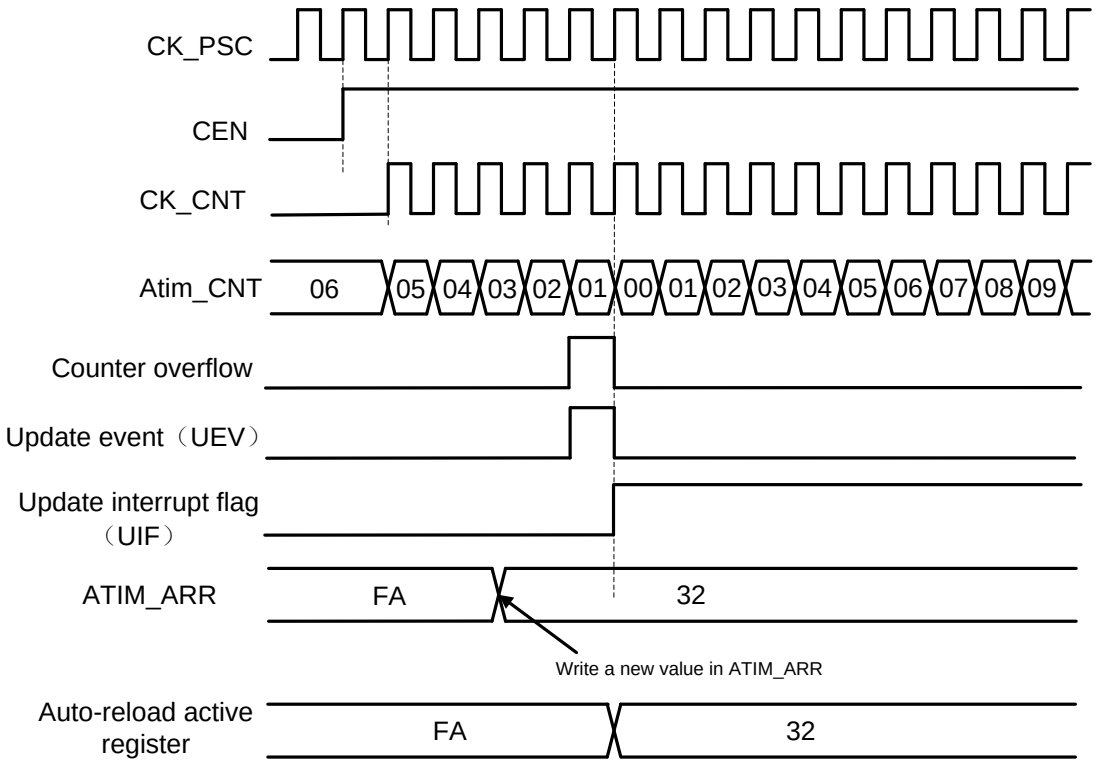


Figure 29-13 Counter Timing Diagram, Update Event When ARPE=1 (Underflow)

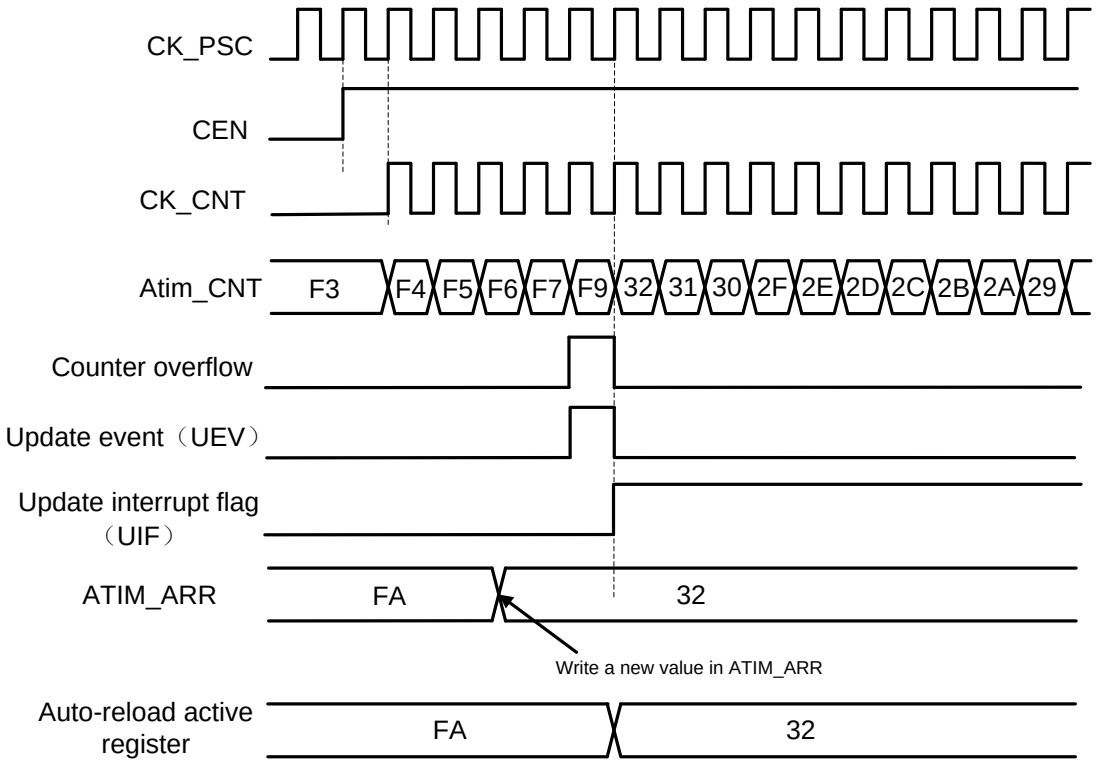


Figure 29-14 Counter Timing Diagram, Update Event When ARPE=1 (Overflow)

29.4.3 Repetition Counter

Update event is generated with respect to the counter overflows/underflows, It is actually generated only when the repetition counter has reached zero. This means that data are transferred from the preload registers to the shadow registers(ARR auto-reload register, PSC prescaler register, but also CCRx capture/compare registers in compare mode) every N+1 counter overflows or underflows, where N is the value in the RCR repetition counter register.

The repetition counter is decremented:

- At each counter overflow in upcounting mode
- At each counter underflow in downcounting mode
- At each counter overflow and at each counter underflow in center-aligned mode

When the update event is generated by software (by setting the UG bit in EGR register) or by hardware through the slave mode controller, it occurs immediately whatever the value of the repetition counter is and the repetition counter is reloaded with the content of the RCR register.

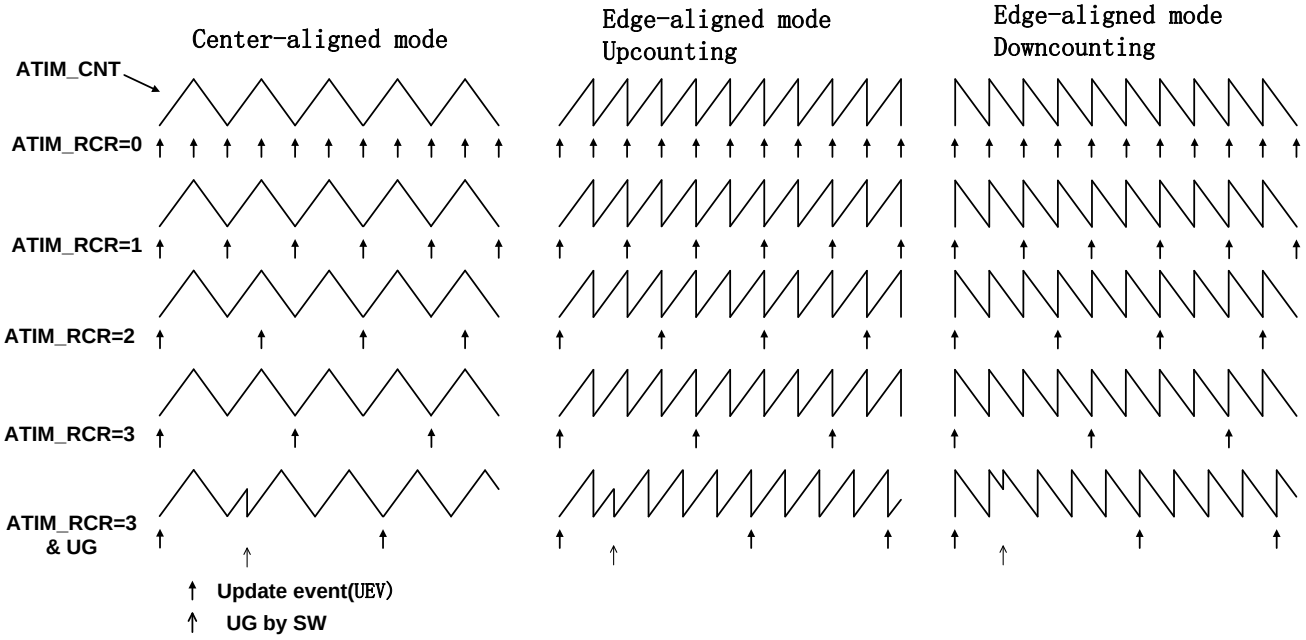


Figure 29-15 Update Rate Examples Depending on Mode and RCR Register Settings

29.4.4 Preload Register

The below register supports preload function:

- ARR
- RCR
- PSC (preload function is always on)
- CCR
- CcxE and CcxNE
- OcxM

Except PSC register, the preload function can be enabled or disabled by the software.

The register with preload function contains two groups of physical entities:

- Shadow register: register in use by the actual timer
- Preload register: register in use by the actual timer

When preload is disabled, the register features with preload function are as follows:

- The preload register can be accessed and rewritten by software in real time
- The shadow register is updated synchronously with the preload register

If preload is enabled, then:

- All software operations access the preload register
- When the update event occurs, the contents of all the preload registers will be synchronously transferred to the corresponding shadow register

29.4.5 Working Clock

The counter can work with the following clock:

- APBCLK-Internal clock mode
- External input clock (Tix) - External clock mode 1
- External trigger input (ETR) - External clock mode 2
- Internal trigger inputs (ITRx) - Using one timer TRGO to be clock

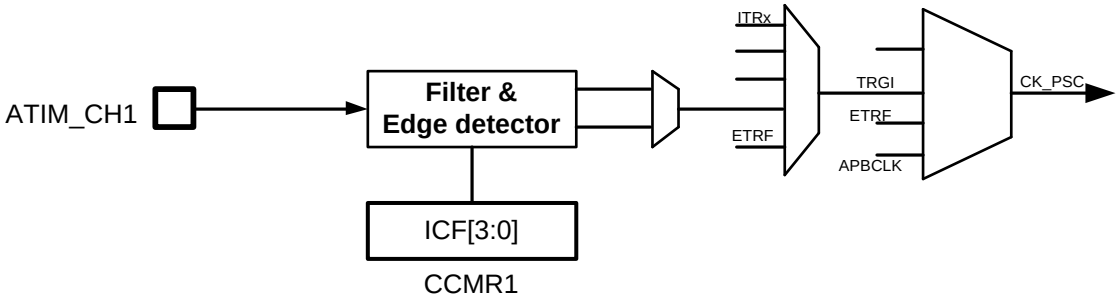


Figure 29-16 ATIM Clock Source Figure

29.4.5.1 Internal Clock Source

If the slave mode controller is disabled (SMS=000), then the CEN, DIR (in the CR1register) and UG bits (in the EGR register) are actual control bits and can be changed only by software.

When the UG register is set, the update signal is synchronized by PSC counter, the counter value will be reinitialized.

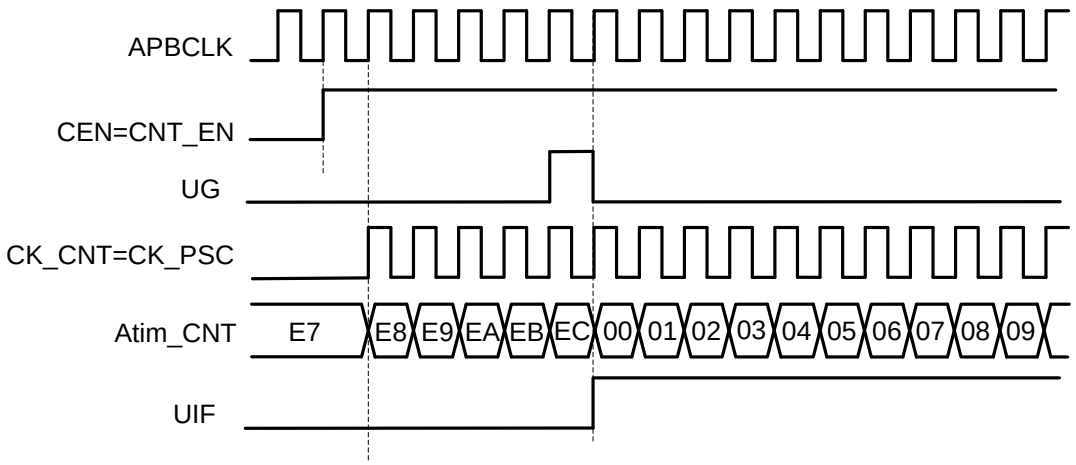


Figure 29-17 Control Circuit in Normal Mode, Internal Clock Divided by 1

29.4.5.2 External Clock Source Mode

This mode is selected when SMS=111 in the SMCR register. The counter can count at each rising or falling edge on a selected input.

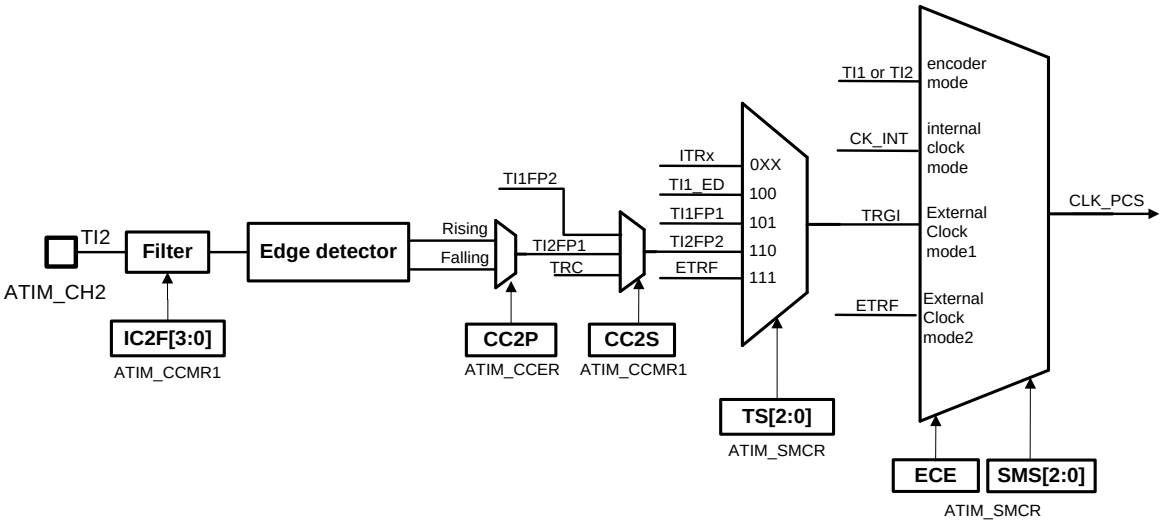


Figure 29-18 TI2 External Clock Connection Example

Before triggering the counter counting, the external input signal will synchronized by internal clock, and the effective edge of the input signal will trigger the TIF flag.

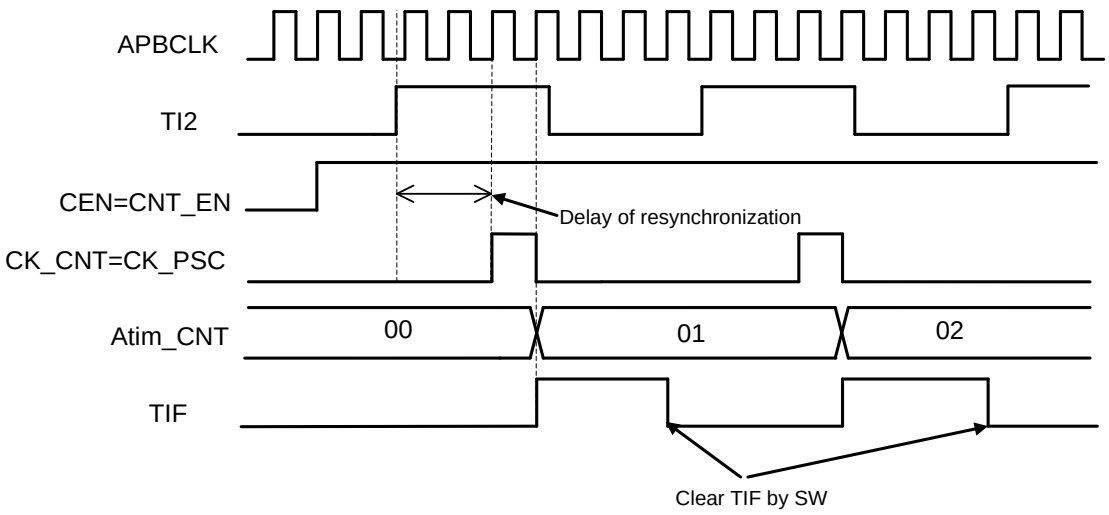


Figure 29-19 Control Circuit in External Clock Mode 1

When using external clock counting, still should enable the internal clock of ATIM (APBCLK), because ATIM uses APB_ CLK to synchronize and filter the external input clock. In external clock mode 1, the external input clock is filtered and edge selected to obtain an effective counting edge, which is input to the prescaler module as an effective working clock (CLK_PSC).

The external clock synchronization adopts a simple two-stage trigger structure. Therefore, in order to avoid metastability, the external input clock width is required to be at least greater than 2 APB_ CLK cycle.

In this mode, only the inputs of channels 1 and 2 can be used as clock input. The required configuration is as follows:

- configure the corresponding pin as ATIM_ CH2 function in the GPIO module
- Turn off channel enable and configure ATIM_CCER.CC2E = 0, ensure that the subsequent channel configuration is successful
- Select input channel and configureATIM_CCMR1.CC2S=01, IC2 choose TI2
- Select the effective edge of the count and configure ATIM_CCER.CC2P=0, select pos or neg
- Configure input filtering time, configureATIM_CCMR1.IC2F[3:0](IC2F=0000, No input filtering)
- Enable external clock mode 1, configuration ATIM_SMCR.SMCR=111
- Select trigger input source and configure ATIM_SMCR.TS=110,Ti2 is selected as the trigger input source
- Open channel enable, configureATIM_CCER.CC2E=1
- Enable counter, configure ATIM_CR1.CEN=1

The following figure is an example of a typical external clock counting mode 1:

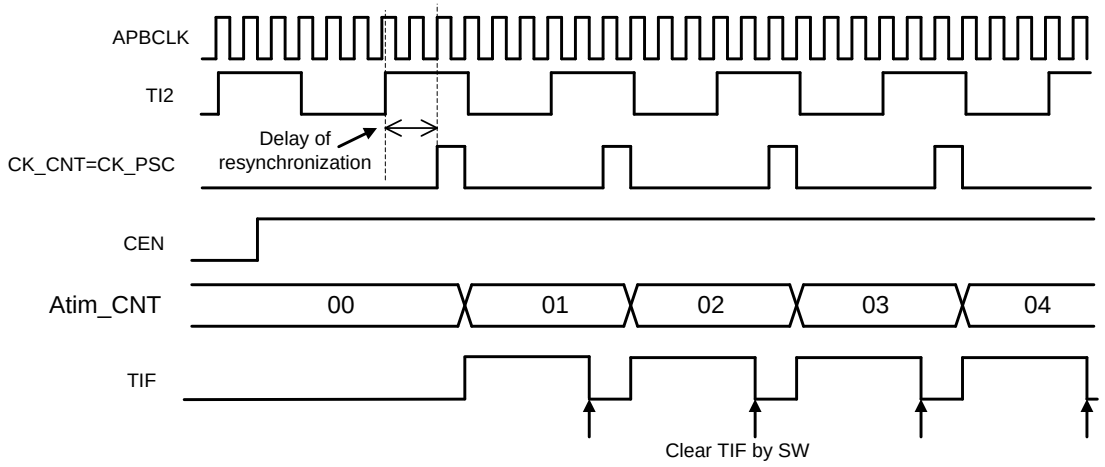


Figure 29-20 Control Circuit in External Clock Mode 1

29.4.5.3 External Clock Source Mode 2

This mode is selected by writing ECE=1 in the TIMx_SMCR register. The counter can count at each rising or falling edge on the external trigger input ETR.

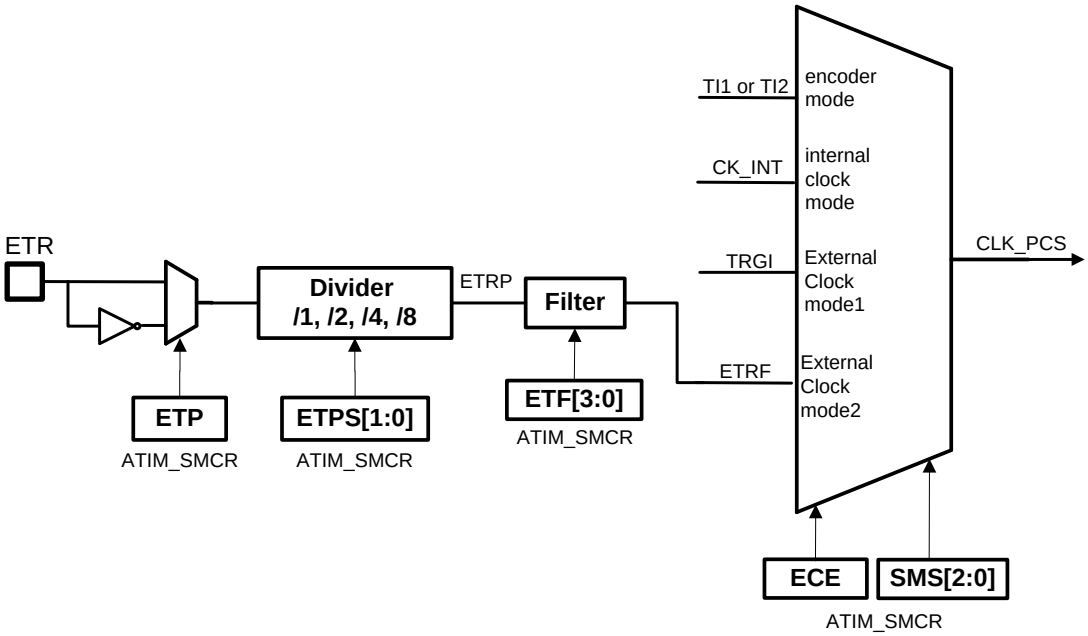


Figure 29-21 External Trigger Input Block

The following figure uses the rising edge of ETR frequency division for counting, in which the actual counting time is delayed from the rising edge of ETR input due to the synchronization process of internal clock.

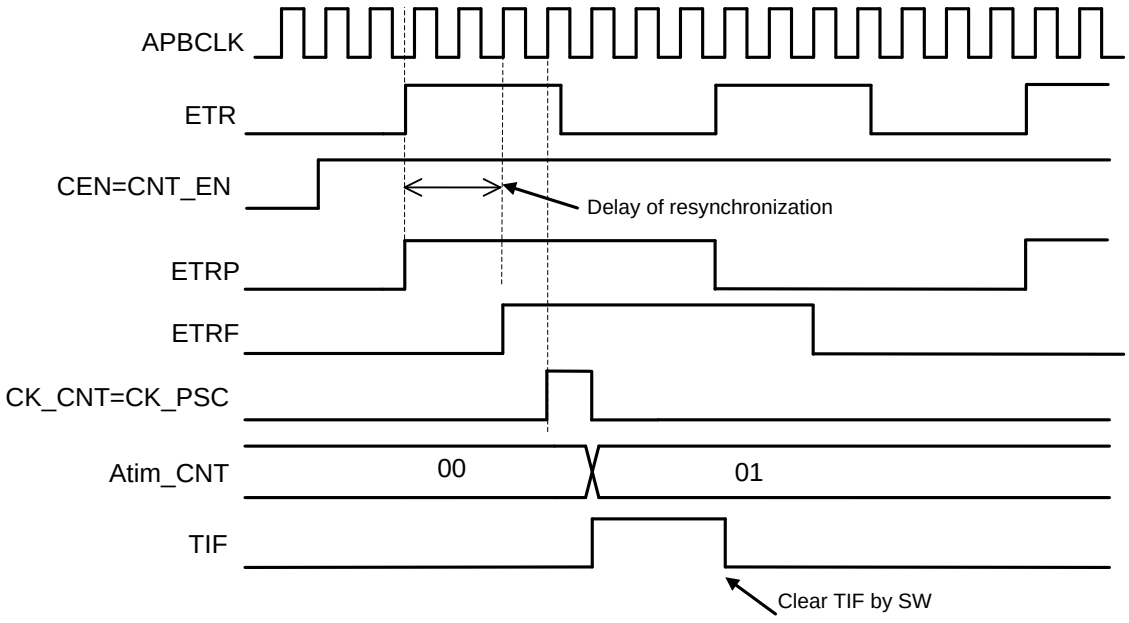


Figure 29-22 Control Circuit in External Clock Mode 2

The main difference from external clock mode 1 is that the ETR input is directly divided and then filtered to generate CK_ PSC clock, which means that ETR input frequency can be supported higher

than APB_ CLK application scenario. In this case, the ETR input needs to be prescaled first, and then used to drive the counter.

The configuration required for this mode is as follows：

- configure the corresponding pin as ATIM_ ETR function in the GPIO module
- Set ETP for edge selection，ATIM_SMCR.ETP=0
- Set ETR frequency division and configure ATIM_SMCR.ETPS[1:0]=01
- Configure input filter time，ATIM_SMCR.ETF[3:0]=0000
- Set the ECE register to enable external clock mode 2,ATIM_SMCR.ECE=1, ATIM_SMCR.SMS=000
- Enable counter, configure ATIM_CR1.CEN=1

The following figure is an example of a typical external clock mode 2:

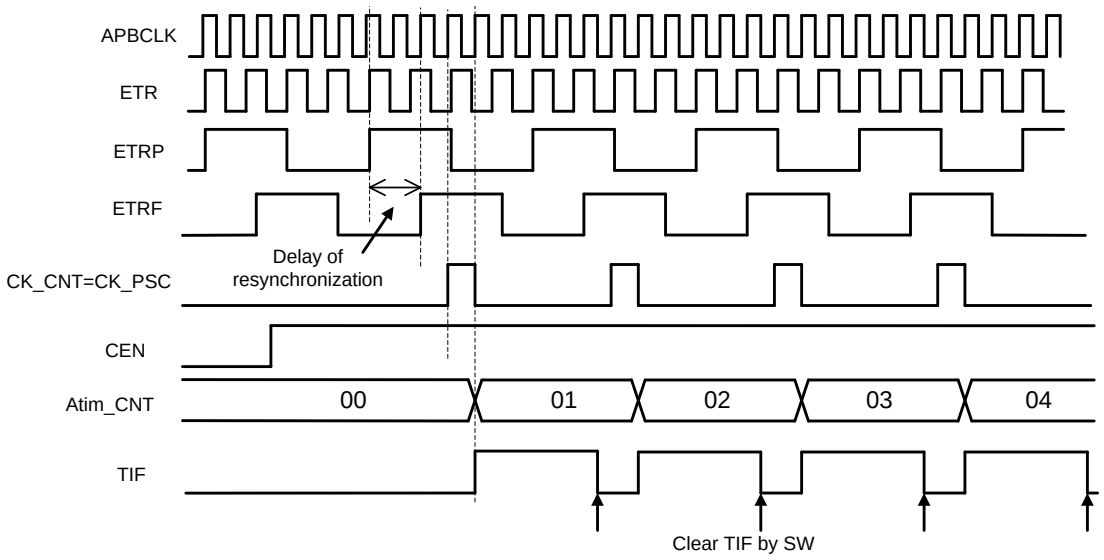


Figure 29-23 Control Circuit in External Clock Mode 2

When using external clock mode 2, ATIM can still be configured as slave mode: For example, ETR input counting is used, and TRGO of another timer is used as trigger signal. When the trigger event comes, reset the counter and restart counting.

29.4.6 Internal Trigger Signal (ITRx)

ATIM supports 4 ITR inputs, which can be used for counting trigger or internal signal capture. When used for internal signal capture, it is necessary to configure TS as 000 ~ 011 to select ITR0 ~ ITR3, and CCxS should set 11, then TRC is as the capture signal.

Each ITR input supports 4 internal signal extensions configured by the ITRxSEL register. Refer to the following table for input signal source:

Slave	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
ATIM	GPTIM1_TRGO	GPTIM2_TRGO	COMP1	COMP2

29.4.7 Capture/Compare Channels

ATIM contains 4 capture/compare channels, each Capture/Compare channel is built around a capture/compare register (including a shadow register), a input stage for capture and an output stage.

The input stage samples the corresponding Tix input to generate a filtered signal TixF, then, an edge detector with polarity selection generates a signal (TixFPx) which can be used as trigger input by the slave mode controller or as the capture command. It is prescaled before the capture register (ICxPS).

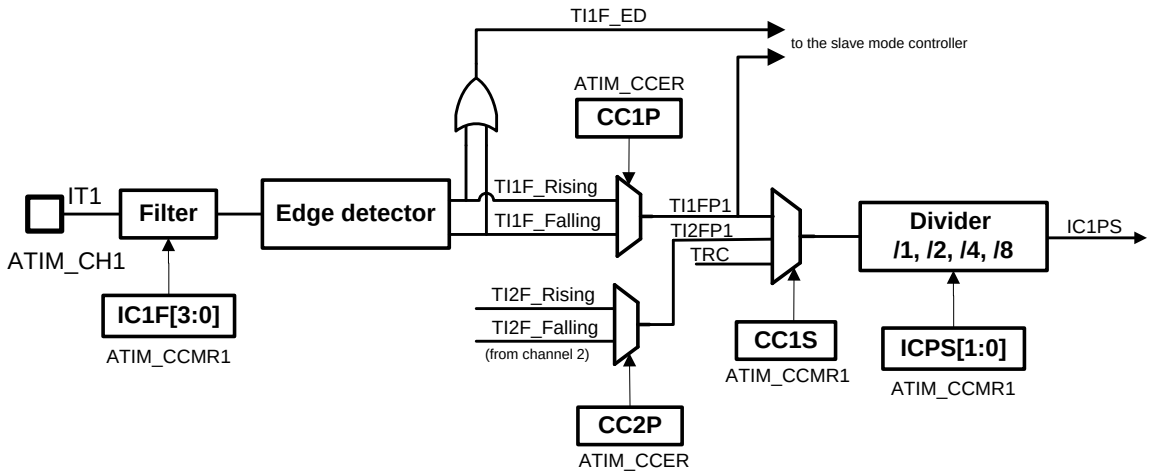


Figure 29-24 Capture/Compare Channel (Example: Channel 1 Input Stage)

The output stage generates an intermediate waveform which is then used for reference: OCxREF (active high). The polarity acts at the end of the chain. Channels 1~3 support complementary output and dead-time insertion, channel 4 is relatively simple and does not support complementary output.

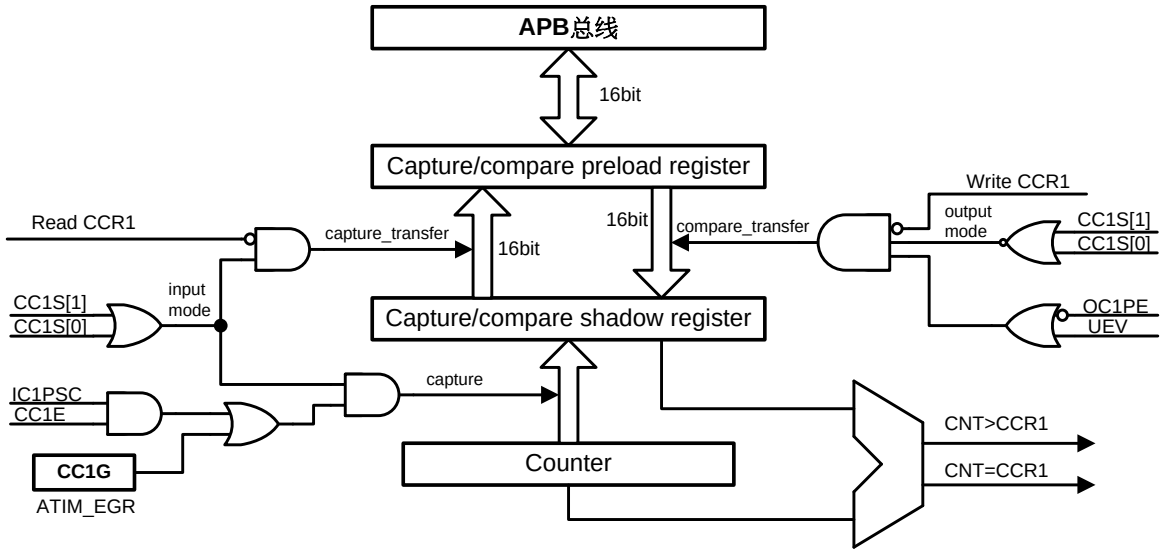


Figure 29-25 Capture/Compare Channel 1 Main Circuit

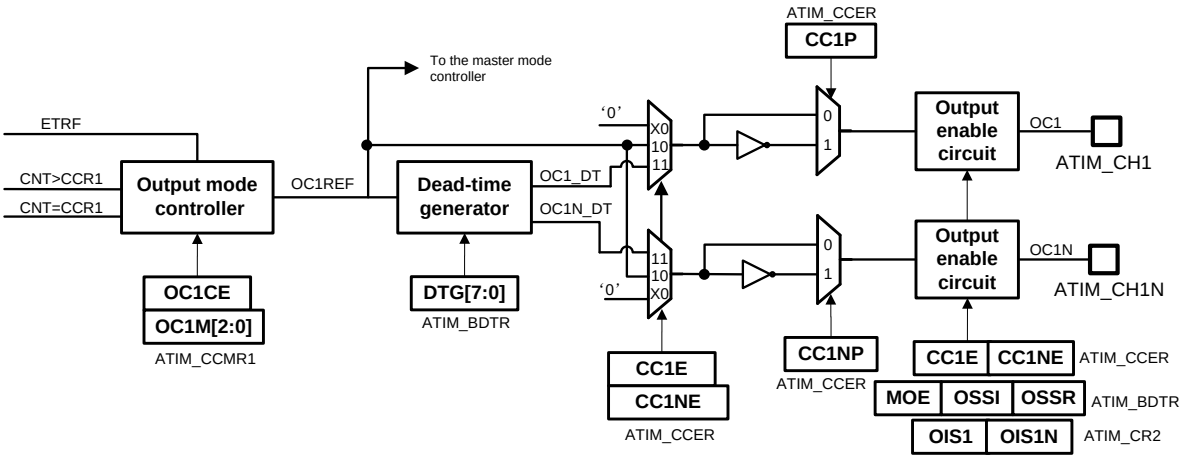


Figure 29-26 Output Stage of Capture/Compare Channel (Channel 1 to 3)

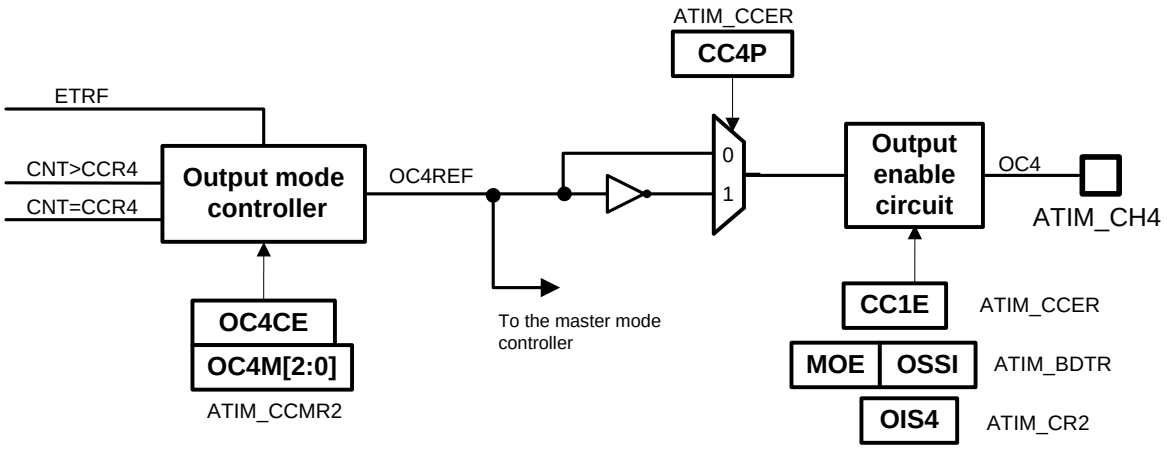


Figure 29-27 Output Stage of Capture/Compare Channel (Channel 4)

The capture/compare block is made of one preload register and one shadow register. Write and read always access the preload register. In capture mode, captures are actually done in the shadow register, which is copied into the preload register. In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter.

29.4.8 Input Capture Mode

When the expected level transition occurs on the Icx signal, a capture will be triggered, and the current counter value is latched into the CCR. At the same time, the CcxIF interrupt flag is set, and the corresponding interrupt or DMA request can be triggered. If a capture event occurs while CcxIF is high, the capture data conflict flag (CcxOF, Over-Capture) is set (the last captured value in CCR is overwritten). CcxIF can be cleared by software or automatically by reading the CCR register. The CcxOF flag is cleared by software writing 1.

Through the cooperation of two or more channels, the input capture of the PWM signal can be realized. For example, to calculate the period and duty cycle of an input signal, this signal can be input from the TI1 pin, the chip will take the rising edge of the filtered signal to obtain TI1FP1, and take the falling edge of the filtered signal to obtain TI1FP2, and input TI1FP1 To capture channel 1, input TI1FP2 to capture channel 2, then channel 1 can capture the rising edge of the input signal, while channel 2 captures the falling edge of the input signal; after the capture interrupt occurs regularly, the software passes the CCR1 and CCR2 register values. The period and duty cycle of the input signal can be calculated.

The rising edge of TI1 input captures the value of the counter to ATIM_CCR1 register, configuration steps are as follows:

- Configure the corresponding pin as ATIM_ CH1 function in the GPIO module
- Close channel enable, configureATIM_CCER.CC1E=0, ensure that the subsequent channel configuration is successful
- Select input channel and configureATIM_CCMR1.CC1S=01, IC1 choose to TI1
- Select the effective edge of the count and configure ATIM_CCER.CC1P, choose pos or neg
- Configure input filtering time, configureATIM_CCMR1.IC1F[3:0]
- Configure input prescaler, configureATIM_CCMR1.IC1PS[1:0]
- Open channel enable, configureATIM_CCER.CC1E=1

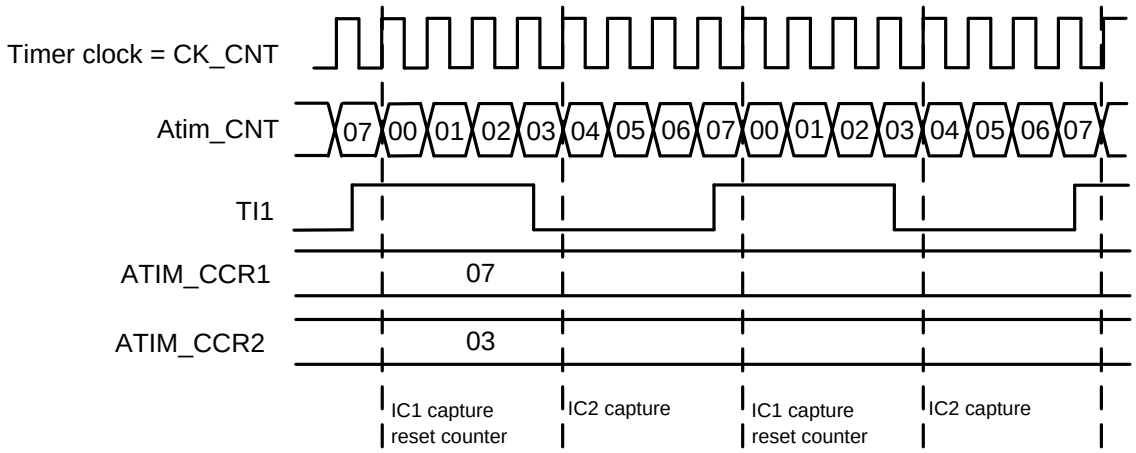


Figure 29-28 PWM Input Capture Mode Timing

To realize the PWM input capture function, the following settings are required：

- configure the corresponding pin as ATIM_ CH1 function in the GPIO module
- Close channel enable, configureATIM_CCER.CC1E=0, ATIM_CCER.CC2E=0ensure that the subsequent channel configuration is successful
- Select input channel,IC1 and IC2 are chosen to the same TI1 input, set ATIM_CCMR1.CC1S=01, ATIM_CCMR1.CC2S=10
- Select the effective edge of the count, the two channels IC1,IC2 have opposite polarities, set ATIM_CCER.CC1P=0, ATIM_CCER.CC2P=1
- Configure input filtering time, configureATIM_CCMR1.IC1F[3:0], ATIM_CCMR1.IC2F[3:0]
- Configure input prescaler, configureATIM_CCMR1.IC1PS[1:0], ATIM_CCMR1.IC2PS[1:0]
- Select trigger input signal and configure ATIM_SMCR.TS[2:0]=101
- Set the slave mode controller to reset mode and configureATIM_SMCR.SMS[2:0]=100
- Open channel enable, configureATIM_CCER.CC1E=1, ATIM_CCER.CC2E=1

29.4.9 Forced Output Mode

In the compare output mode, software can directly force OCxREF to a specific level, independent of the CCR and counter comparison results.

The software can directly force OCxREF to be valid by writing the OcxM=101 register (OCxREF is fixed to be active high), and can directly force OCxREF to be invalid (low level) by writing OcxM=100. However, the software force operation will not cancel the comparison process, and the comparison between the CCR and the counter will continue.

29.4.10 Output Compare Mode

In the output comparison mode, when the CCR is equal to the counter value, OCxREF can be set to valid, invalid, or level reversal. At the same time, the interrupt flag will also be set and DMA requests can be sent.

The output comparison can also be used to output a pulse signal of a specific width (single output).

Steps:

- Select counting clock (internal, external, prescaled, etc.)
- Write the desired data to the ARR and CCR registers
- Set interrupt enable and DMA enable as needed
- Select output mode
- Enable counter

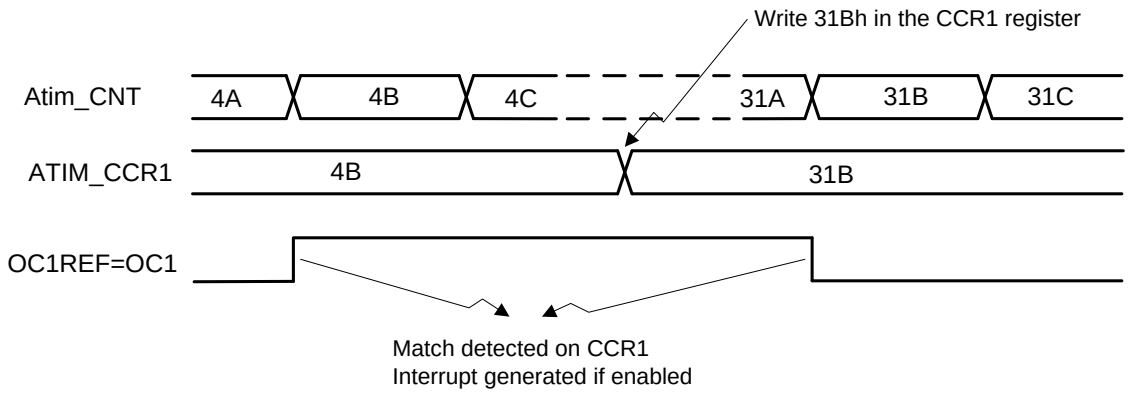


Figure 29-29 Output Comparison Mode, Trigger OC1

Without enabling preload, the software can rewrite the CCR register at any time to realize the real-time control of the output waveform. If preload is enabled, the CCR shadow register is updated to the contents of the preload register only when the next update event occurs.

29.4.11 PWM Output

Pulse Width Modulation mode allows you to generate a signal with a frequency determined by the value of the ARR register and a duty cycle determined by the value of the CCR register.

The polarity of the output signal can be configured by the CCxP register. During PWM mode operation, CNT and CCR are compared in real time. Since the counter supports edge-aligned and center-aligned counting modes, the PWM output also supports edge-aligned and center-aligned modes.

PWM edge alignment mode

In upcounting mode, when configured as PWM mode 1, OCxREF is high as long as CNT < CCRx else it becomes low. If the compare value in CCR is greater than the auto-reload value (in ARR), then OCxREF is held at 1; if the compare value is 0, then OCxREF is held at 0.

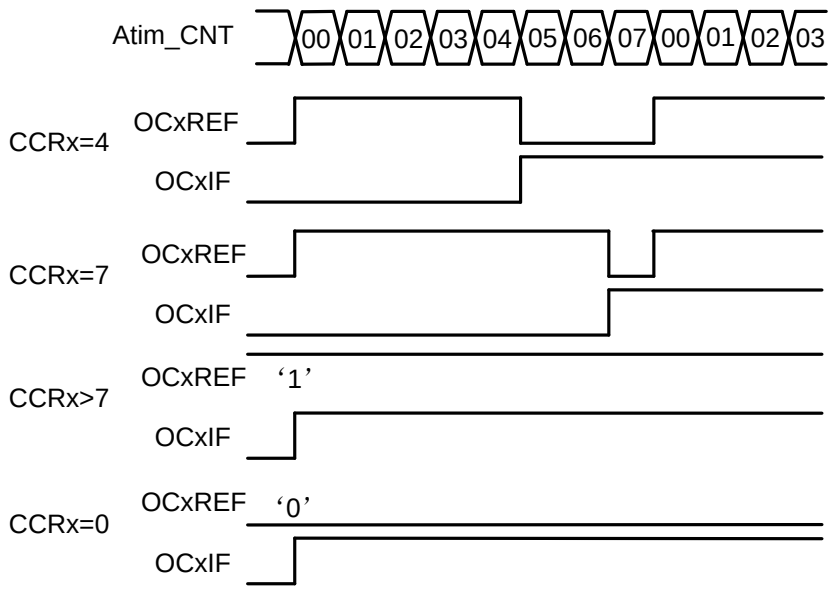


Figure29-30Edge-Aligned PWM Waveforms(ARR=7)

OCxREF value configuration is same at downcounting and upcounting.

PWM center-aligned mode

OCxREF value configuration is same as edge alignment mode:

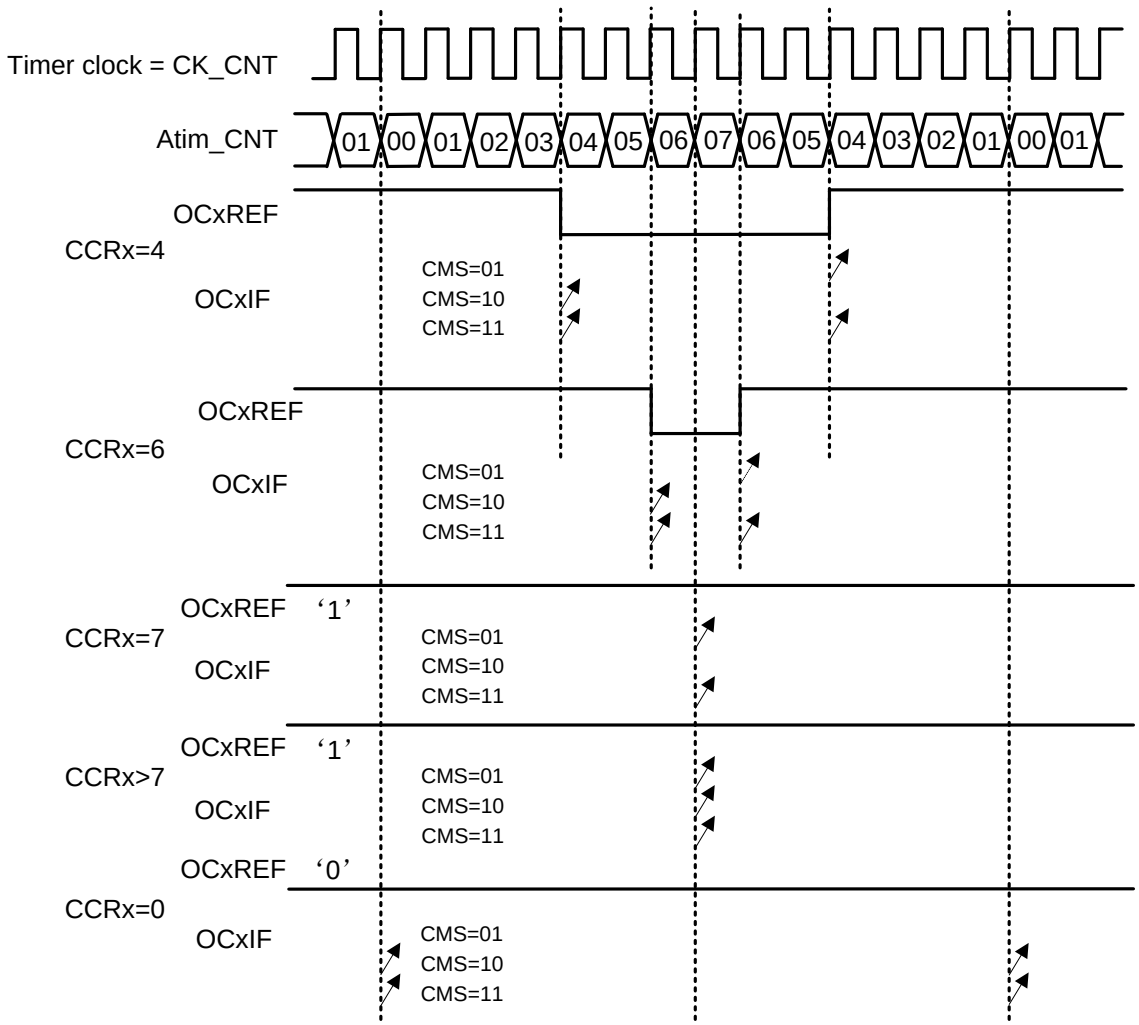


Figure 29-31 Center-Aligned PWM Waveforms(APR=7)

When the center-aligned counting is started, the initial counting direction is determined by the DIR register; then during the counting process, the state of the DIR register is directly controlled by the hardware. For the sake of safety, it is recommended that the user program do an update through the UG register before starting the counter, and do not rewrite the counter during the counting process.

29.4.12 Complementary Outputs and Dead-Time Insertion

ATIM channel 1~3 support complementary outputs and dead-time insertion. DTG[7:0] register used to set dead-time insertion (active to all channels). The OCx output signal is the same as the reference signal except for the rising edge, which is delayed relative to the reference rising edge. The OCxN output signal is the opposite of the reference signal except for the rising edge, which is delayed relative to the reference falling edge.

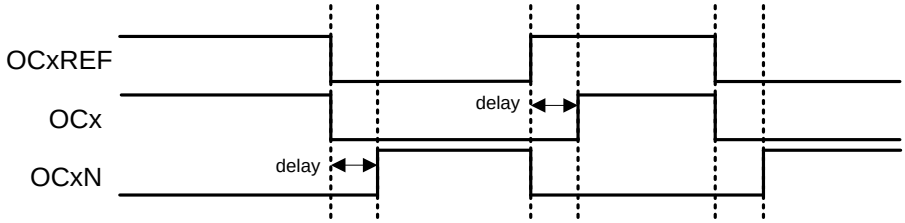


Figure 29-32 Complementary Outputs with Dead-Time Insertion

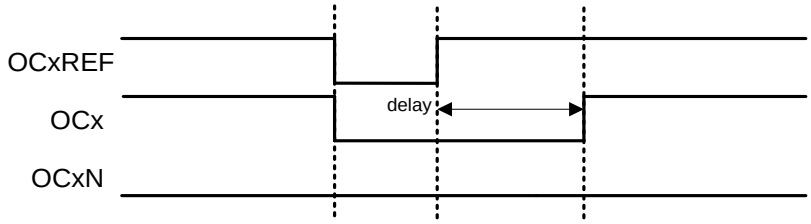


Figure 29-33 Dead-Time Waveforms with Delay Greater than the Negative Pulse

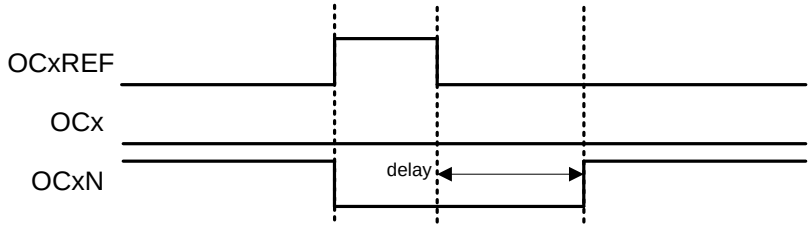
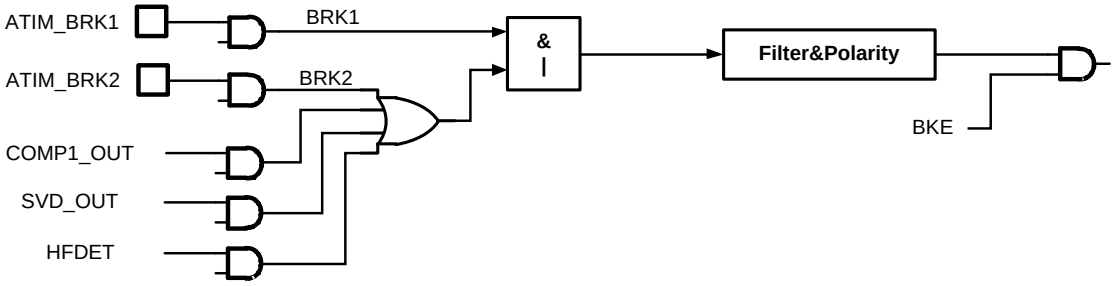


Figure 29-34 Dead-Time Waveforms with Delay Greater than the Positive Pulse

29.4.13 Braking Function

The braking function is by the 2-channel breaking signals input by the external BRK pin, or the effective output generated by comparator, SVD and the vibration stop detection of the XTHF. After power on reset, the braking circuit is prohibited, and the user enables the braking function by setting the BKE register. The 2-way brake input can be configured as and operation, or operation. The combined brake signal can be configured with effective polarity and digital filtering.

The brake input control logic is shown in figure below:



When GPIO is set as digital peripheral function, its input signal is directly connected to the brake input of ATIM. When GPIO is set to other functions, the brake input port of ATIM is fixed to 1. Through the BRKxGATE register, the actual level of the gated BRKx signal can be controlled. The software can flexibly set the unused BRKx to 0 or 1 level to meet the needs of the later logic circuit.

When a breaking event occurs：

- The output enable register is asynchronously cleared, and can be selected through the OSSI register. The output is forced to the inactive / idle / reset state
- Each output channel is driven to the level defined by the OISx register
- When the complementary output is enabled, the output is asynchronously set to the inactive and reset states, the dead band insertion circuit starts to work, and the drive output is the level defined by OISx and OISxN after the dead band time
- The brake flag register is set, and the interrupt or DMA can be triggered according to the configuration
- If automatic output is enabled (AOE=1), the output enable bit (MOE) will be automatically set when the next update event occurs; otherwise, the MOE will remain 0 until it is reset by the software

Note: The BRK signal is level effective, so MOE cannot be enabled and the brake mark BIF cannot be cleared when BRK remains valid.

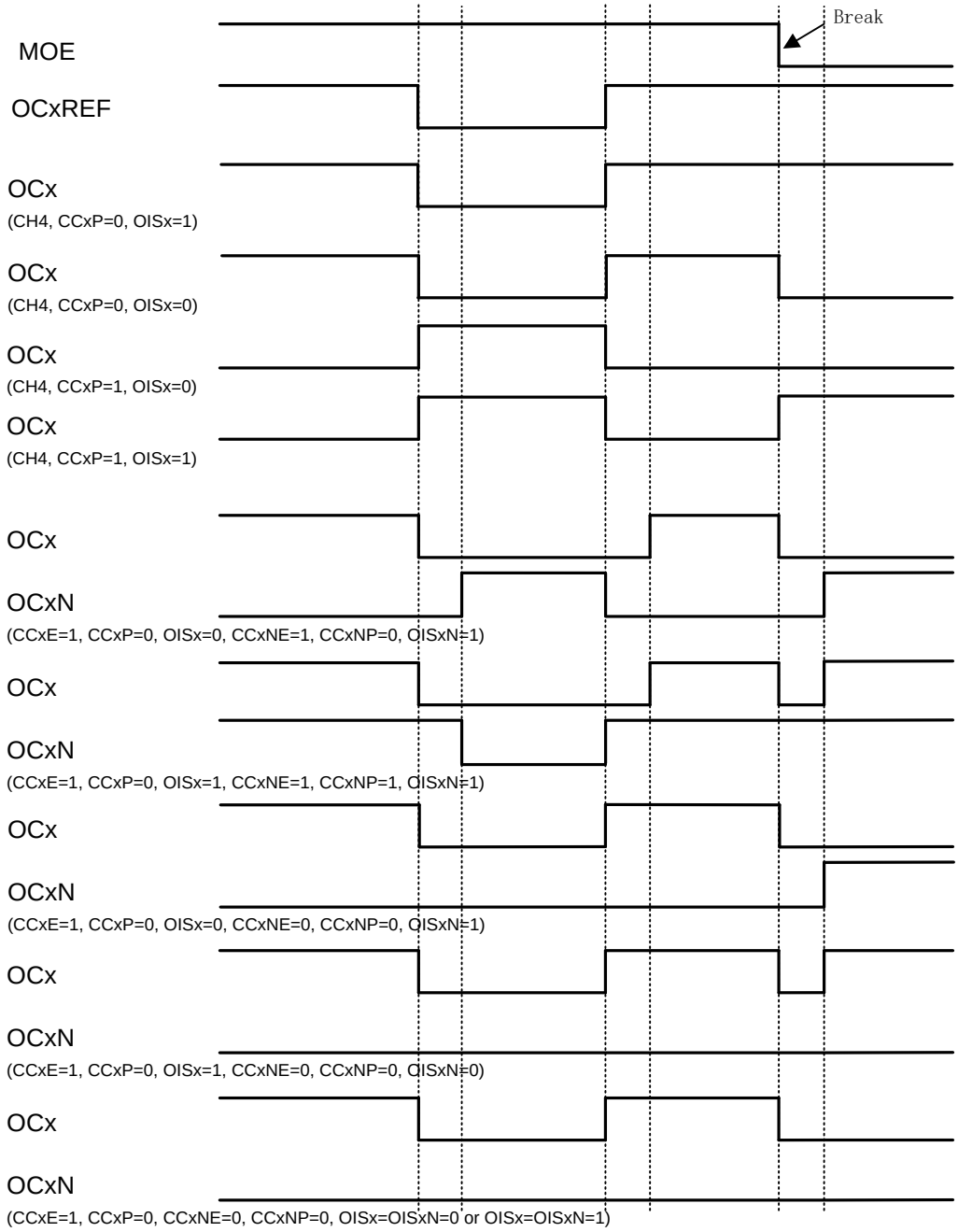


Figure 29-35 Output Behavior in Response to a Break

29.4.14 Status Logic Table of Complementary Output Channel Signal

The following is the status logic table of complementary output channel with different control register, MOE is the total output enable bit of timer, OSSI defines whether to close IO output or enter off state under idle state (MOE=0), and OSSR defines whether to close IO output or enter off state under run state (MOE=1).

Control register					Output states	
MOE	**OSSI**	**OSSR**	**CcxE**	**CcxNE**	**Ocx output state**	**OcxN output state**
1	X	0	0	0	Output disabled (not driven by the timer), Ocx=0, Ocx_EN=0	Output disabled (not driven by the timer), OcxN=0, OcxN_EN=0
		0	0	1	Output disabled (not driven by the timer), Ocx=0, Ocx_EN=0	OCxREF + Polarity OcxN=OCxREF xorCCxNP, OcxN_EN=1
		0	1	0	OCxREF + Polarity Ocx=OCxREFx or CCxP, Ocx_EN=1	Output Disabled (not driven by the timer) OcxN=0, OcxN_EN=0
		0	1	1	OCREF + Polarity + dead-time Ocx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OcxN_EN=1
		1	0	0	Output Disabled (not driven by the timer) Ocx=CCxP, Ocx_EN=0	Output Disabled (not driven by the timer) OcxN=CCxNP, OcxN_EN=0
		1	0	1	Off-State (output enabled with inactive state) Ocx=CCxP, Ocx_EN=1	OCxREF + Polarity OcxN=OCxREFx or CCxNP, OcxN_EN=1
		1	1	0	OCxREF + Polarity Ocx=OCxREFx or CCxP, Ocx_EN=1	Off-State (output enabled with inactive state) OcxN=CCxNP, OcxN_EN=1
		1	1	1	OCREF + Polarity + dead-time Ocx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OcxN_EN=1
0	0	X	0	0	Output disabled (not driven by the timer)	Output disabled (not driven by the timer)
	0		0	1	Output disabled (not driven by the timer）	
	0		1	0		
	0		1	1		
	1		0	0	Output disabled (not driven by the timer)	Output disabled (not driven by the timer)

Control register					Output states
	1		0	1	Off-state (output enabled with inactive state)
	1		1	0	Asynchronously: Ocx=CCxP, Ocx_EN=1,
	1		1	1	OcxN=CCxNP, OcxN_EN=1 If the clock is present: Ocx=OISx, OcxN=OISxN after dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state

29.4.15 6-Step PWM Generation

When complementary outputs are used on a channel, preload bits are available on the OCxM, CCxE and CCxNE bits. The preload bits are transferred to the shadow bits at the COM commutation event. Thus you can program in advance the configuration for the next step and change the configuration of all the channels at the same time. COM can be generated by software by setting the COM bit in the TIMx_EGR register or by hardware (on TRGI rising edge).

A flag is set when the COM event occurs, which can generate an interrupt or a DMA request.

Following figure describes the behavior of the OCx and OCxN outputs when a COM event occurs, in 3 different examples of programmed configurations.

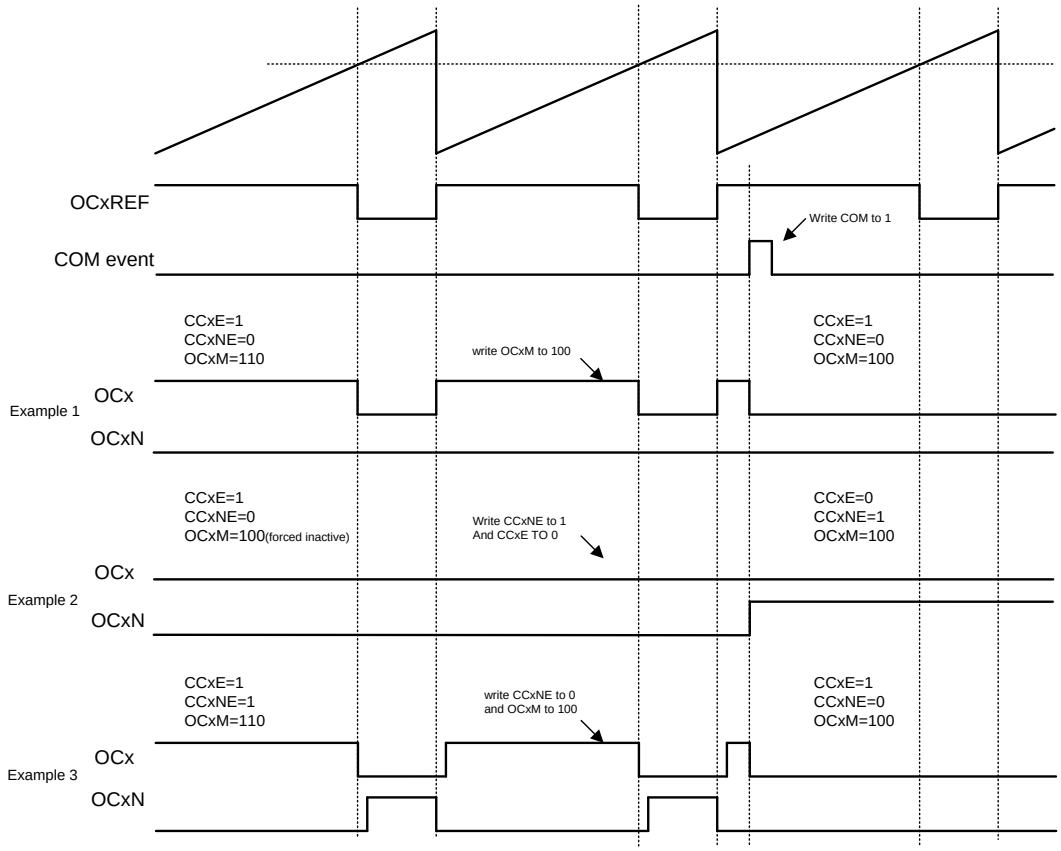


Figure 29-36 6-Step Generation, COM Example (OSSR=1)

29.4.16 One-Pulse Mode

One-pulse mode (OPM) is a particular case of the previous modes. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

Unlike other output modes, the counter stops automatically when the next update event arrives. Only when the CCR and the initial value of the counter are different, the pulse can be output correctly. When counting up, CNT<CCR<=ARR is required, and when counting down, CNT>CCR is required.

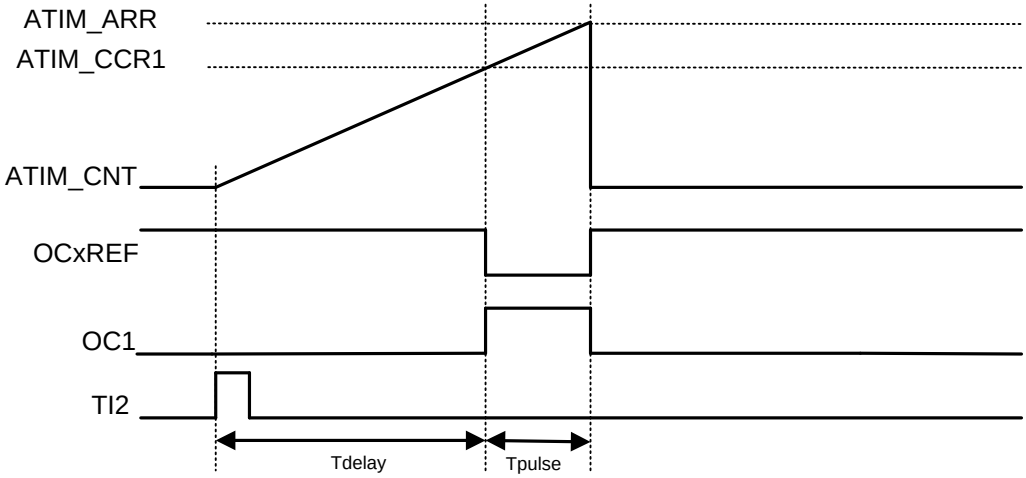


Figure 29-37 Example of Single Pulse Mode

The upper figure takes TI2 input as the trigger signal of the counter. OCxREF output low level when the count value is equal to CCR. After counting to ARR, OCxREF returns to high level, and the counter rolls back to 0 to stop counting.

The configuration of TI2 as input trigger to realize the above functions is as follows:

- Configure the corresponding pin as ATIM_ CH2 function in the GPIO module
- Close channel enable, configure ATIM_CCER.CC2E=0, ensure that the subsequent channel configuration is successful
- Select input channel and configure ATIM_CCMR1.CC2S=01
- Select the effective edge of the count and configure ATIM_CCER.CC2P=0
- Select trigger input signal and configure ATIM_SMCR.TS[2:0]=110, TI2FP2 as TRGI
- Set the slave mode controller to trigger mode and configure ATIM_SMCR.SMS[2:0]=110, TI2FP2 is used to start the counter
- Open channel enable, configure ATIM_CCER.CC2E=1

To realize the above functions, OC1 is configured as an output as follows:

- Configure the corresponding pin as ATIM_ CH1 function in the GPIO module
- Close channel enable, configure ATIM_CCER.CC1E=0, ensure that the subsequent channel configuration is successful
- Output channel, configure ATIM_CCMR1.CC1S=00
- Select the effective edge of the count and configure ATIM_ CCMR1.OC1M=111, PWM mode2
- Open channel enable, configure ATIM_CCER.CC1E=1

Special setting of OPM waveform generation:

- ATIM_CCR1 value determines Tdelay
- The difference between ATIM_ARR and ATIM_CCR1 determines Tpulse (ATIM_ARR - ATIM_ CCR1)
- Set to single pulse mode, configure ATIM_CR1.OPM=1

29.4.17 External Event Clearing OCxREF

The effective state of OCxREF is high level. By applying high level to the external ETR pin, OCxREF can be directly pulled down until the next update event. This function is only valid in output comparison and PWM mode. To enable this function, OcxCE needs to be set to 1.

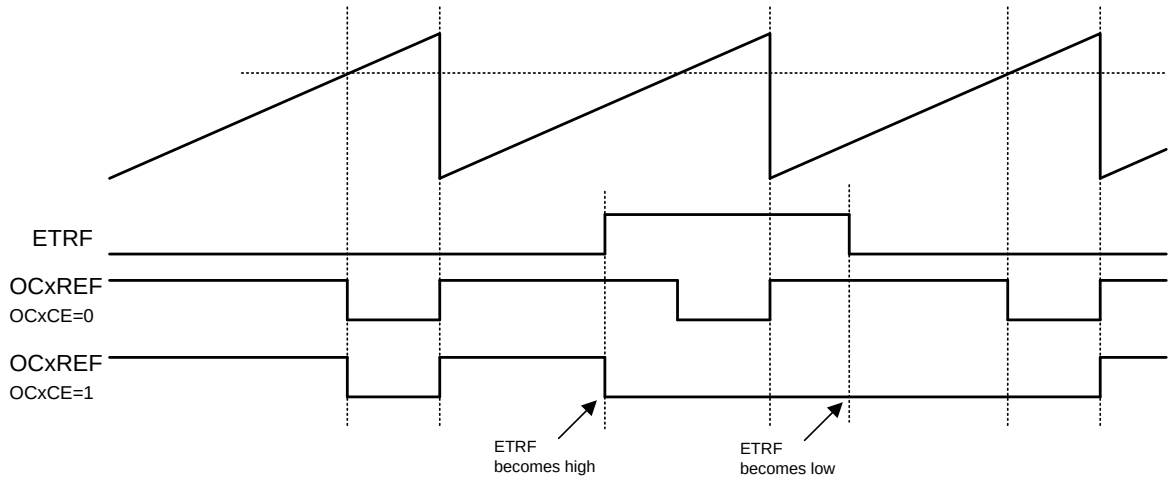


Figure 29-38 ETR Event Clear OCxREF

29.4.18 Encoder Interface Mode

The encoder interface mode involves two external input signals. ATIM determines whether to increase or decrease the count value according to the level of the edge of one signal relative to the other signal. The following table shows the relationship between counting mode and two input signals:

Active edge	Level on opposite signal (TI1 to TI2, TI2 to TI1)	TI1 signal		TI2 signal	
		Rising	Falling	Rising	Falling
Counting on TI1 only	High	Down	Up	No Count	No Count
	Low	Up	Down	No Count	No Count
Counting on TIW only	High	No Count	No Count	Up	Down
	Low	No Count	No Count	Down	Up
Counting on TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

Table 29-1 Encoder Interface Counting Method

For example, when the counter counts with the TI1 signal as the clock, if the rising edge of TI1 samples that TI2 is high, the counter decreases; if the falling edge of TI1 samples that TI2 is high, the counter is incremented.

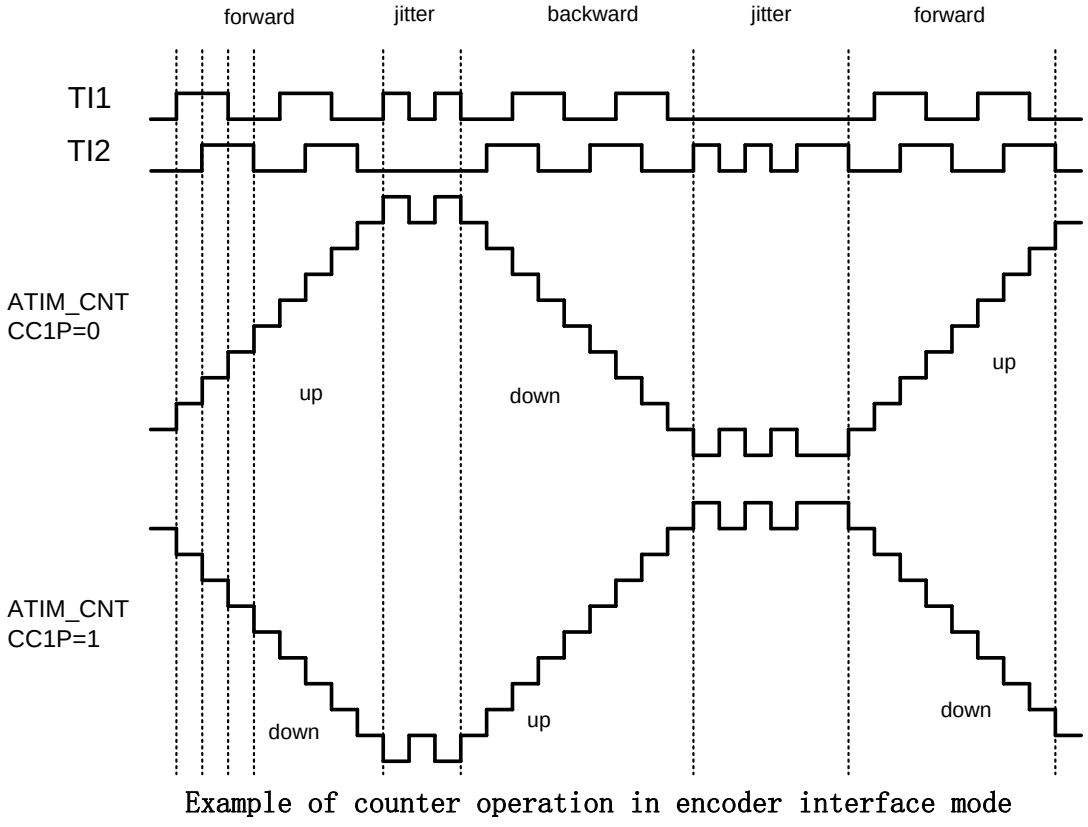


Figure 29-39 Example of Counter Operation in Encoder Interface Mode

The encoding mode input channel needs to be set as follows:

- Configure the corresponding pin as ATIM_ CH1, ATIM_ CH2 function in the GPIO module
- Close channel enable, configure ATIM_CCER.CC1E=0, ATIM_CCER.CC2E=0, ensure that the subsequent channel configuration is successful
- Select input channel and configure ATIM_CCMR1.CC1S=01, ATIM_CCMR1.CC2S=01
- Select the effective edge of the count and configure ATIM_CCER.CC1P=0, ATIM_CCER. CC2P=0
- Set the slave mode controller to coding mode 3 and configure ATIM_SMCR.SMS[2:0]=011
- Open channel enable, configure ATIM_CCER.CC1E=1, ATIM_CCER.CC2E=1

29.4.19 TIM Slave Mode

The ATIM can be synchronized with an external trigger in several modes: Reset mode, Gated mode and Trigger mode.

Reset mode

In this mode, external input events will cause all preload registers in TIM to be reinitialized, and CNT will return to 0 to start counting. The following figure is an example. The counter counts normally. When the rising edge of external TI1 input is triggered, the counter is cleared and the counting is restarted.

The configuration is as follows:

- Configure the corresponding pin as ATIM_ CH1 function in the GPIO module
- Close channel enable, configure ATIM_CCER.CC1E=0 ensure that the subsequent channel configuration is successful
- Select input channel and configure ATIM_CCMR1.CC1S=01
- Select the effective edge of the count and configure ATIM_CCER.CC1P=0
- Select trigger input signal and configure ATIM_SMCR.TS[2:0]=101, TI1FP1 as TRGI
- Set the slave mode controller to reset mode and configure ATIM_SMCR.SMS[2:0]=100
- Open channel enable, configure ATIM_CCER.CC1E=1
- Enable counter, configure ATIM_CR1.CEN=1

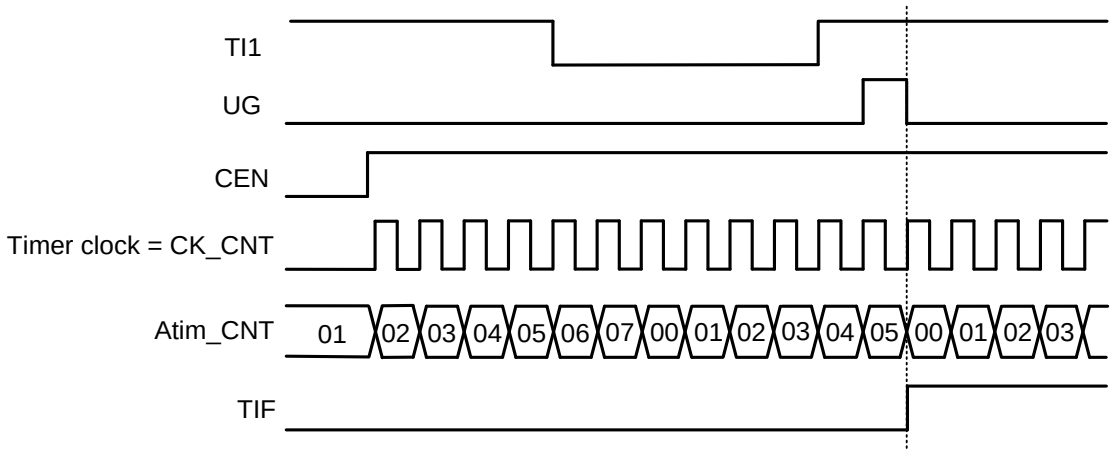


Figure 29-40 Control Circuit in Reset Mode

Gated mode

In this mode, the counter only works when the input signal is at a specific level. When the level change causes the counter to start or stop counting, the interrupt flag will be triggered.

The configuration is as follows:

- Configure the corresponding pin as ATIM_ CH1 function in the GPIO module
- Close channel enable, configure ATIM_CCER.CC1E=0 ensure that the subsequent channel configuration is successful
- Select input channel and configure ATIM_CCMR1.CC1S=01
- Select the effective edge of the count and configure ATIM_CCER.CC1P=0
- Select trigger input signal and configure ATIM_SMCR.TS[2:0]=101, TI1FP1 as TRGI
- Set the slave mode controller to gated mode and configure ATIM_SMCR.SMS[2:0]=101
- Open channel enable, configure ATIM_CCER.CC1E=1
- Enable counter, configure ATIM_CR1.CEN=1

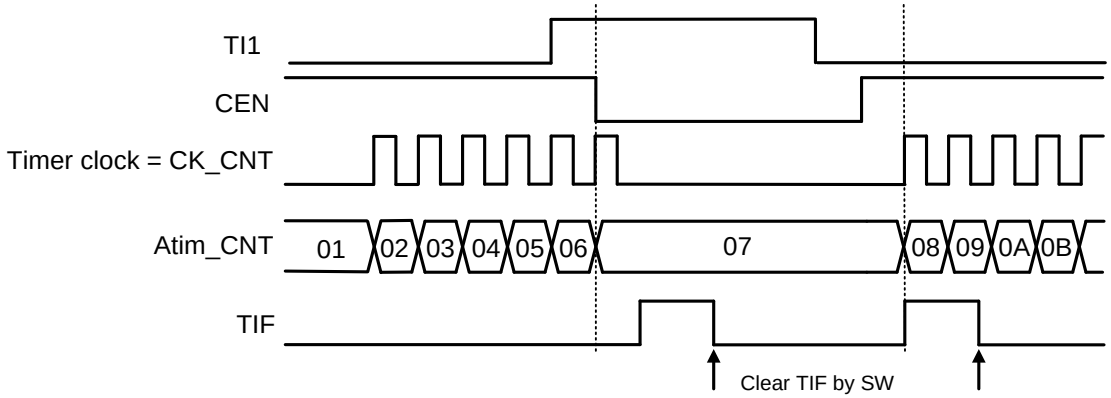


Figure 29-41 Control Circuit in Gated Mode

Trigger mode

The counter can start in response to an event on a selected input.

The configuration is as follows:

- Configure the corresponding pin as ATIM_ CH1 function in the GPIO module
- Close channel enable, configure ATIM_CCER.CC1E=0 ensure that the subsequent channel configuration is successful
- Select input channel and configure ATIM_CCMR1.CC1S=01
- Select the effective edge of the count and configure ATIM_CCER.CC1P=0
- Select trigger input signal and configure ATIM_SMCR.TS[2:0]=101, TI1FP1 as TRGI
- Set the slave mode controller to trigger mode and configure ATIM_SMCR.SMS[2:0]=110
- Open channel enable, configure ATIM_CCER.CC1E=1

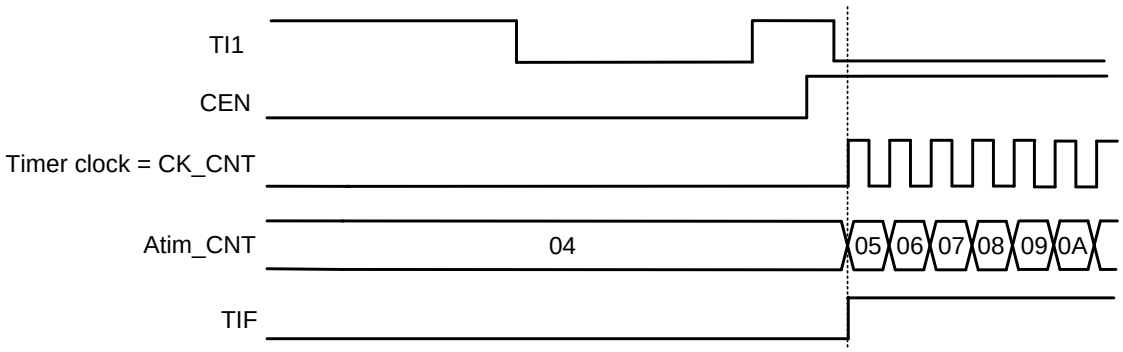


Figure 29-42 Control Circuit in Trigger Mode

External clock mode 2 + Trigger mode

ETR can be set as the counting clock and another external input can be used as the counter start trigger signal. For example, after detecting the rising edge of TI1, the counter starts counting with the rising edge of ETR input

The configuration is as follows:

- Configure the corresponding pin as ATIM_ CH1, ATIM_ETR function in the GPIO module
- Set ETP for edge selection, ATIM_SMCR.ETP=0
- Set ETR frequency division ratio and configure ATIM_SMCR.ETPS[1:0]=01
- Configure input filter time, ATIM_SMCR.ETF[3:0]=0000
- Set the ECE register to enable external clock mode 2, ATIM_SMCR.ECE=1
- Close channel enable, configure ATIM_CCER.CC1E=0 ensure that the subsequent channel configuration is successful
- Select input channel and configure ATIM_CCMR1.CC1S=01
- Select the effective edge of the count and configure ATIM_CCER.CC1P=0
- Select trigger input signal and configure ATIM_SMCR.TS[2:0]=101, TI1FP1 as TRGI
- Set the slave mode controller to trigger mode and configure ATIM_SMCR.SMS[2:0]=110
- Open channel enable, configure ATIM_CCER.CC1E=1

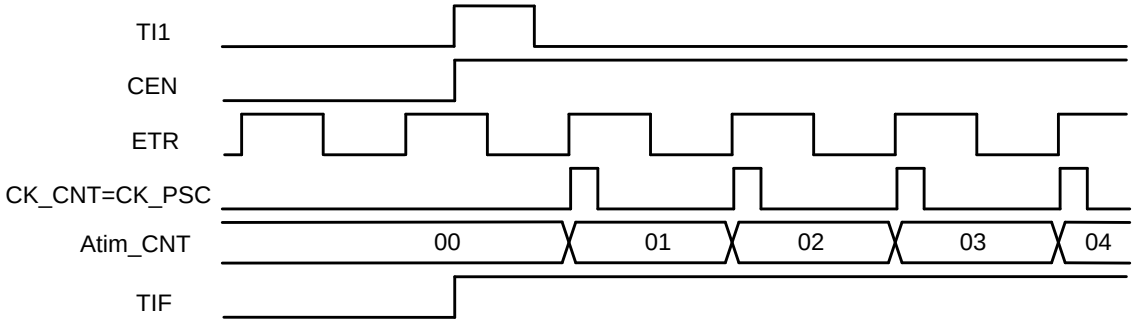


Figure 29-43 Control Circuit in External Clock Mode 2 + Trigger Mode

29.4.20 DMA Access

ATIM supports 7 kinds of DMA requests, including 4 CC channel requests, external trigger requests, user software trigger requests and COM trigger requests.

Each CC channel generates a DMA request, which is used to transfer the content in CCRx to RAM in capture mode and write the data in RAM to CCRx in comparison mode; the DMA request of CC channel can be configured as single transmission or burst transmission (CCxBURSTEN). A single transmission only accesses the CCRx register, and Burst transmission accesses a specific set of registers according to the DCR register configuration.

In addition, external trigger events, software trigger events and com events can also generate DMA requests. When these requests occur, DMA burst transmission will be started, data will be written to one or more registers within ATIM, or one or more register values will be read from ATIM.

DMA request	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA access object	Once transmission length
ATIM_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_COM	N/A	0	Read DMAR	DBL
		1	Write DMAR	

29.4.21 DMA Burst

ATIM supports DMA and DMA burst access. ATIM can be configured to trigger DMA requests when specific events occur. The capture results in CCR can be written to ram, or the contents of one or more registers can be written to ATIM's preload register from RAM.

DMA burst supports multiple DMA requests triggered by one event. Its main function is to continuously update the contents of multiple registers after the event. Therefore, it can realize the functions of dynamically adjusting the output waveform in real time.

The DMA controller needs to point the peripheral target address to a virtual register ATIM_ DMAR。When a specific timer event occurs, ATIM will continuously transmit multiple DMA requests. The writes to ATIM_DMAR by per DMA are redirected to the actual function register of ATIM.

The DBL register is used to set the DMA burst length, and the DBA register is used to set the base address (offset relative to ATIM_CR) for DMA access to ATIM.

29.4.22 Timer Input XOR Function

The TI1Sbit in the ATIM_CR2register, allows the input filter of channel 1 to be connected to the output of a XOR gate, combining the three input pins CH1, CH2 andCH3,used with all the timer input functions such as trigger or input capture.

29.4.23 Hall Sensor Interface

While the advanced timer drives the motor, the output of the hall sensor can be connected by using another general purpose timer as the interface timer.

The interface timer XORs the three input signals and sends them to channel 1 to capture. The slave mode controller of the interface timer is set to reset mode, and the slave input is selected as TI1F_ED, so whenever any one of the three input signals is flipped, the timer will start counting from 0 again. At the same time, configure channel 1 as capture mode, and the capture signal is TRC, so that when the input signal flips, the current count value is captured by the CC1 channel. This capture value records the interval time between the Hall sensor signal flips and can be used for Calculate motor speed.

Other channels of the interface timer can also be configured as PWM output mode, and output OCREF through TRGO to the advanced timer as a COM trigger signal, which can be used to trigger the PWM of the advanced timer to update the configuration parameters in real time.

The figure below shows that CC1 of GPTIM is used for Hall signal timing capture, and CC2 is used to output trigger signal to advanced timer to realize PWM parameters (COM event).

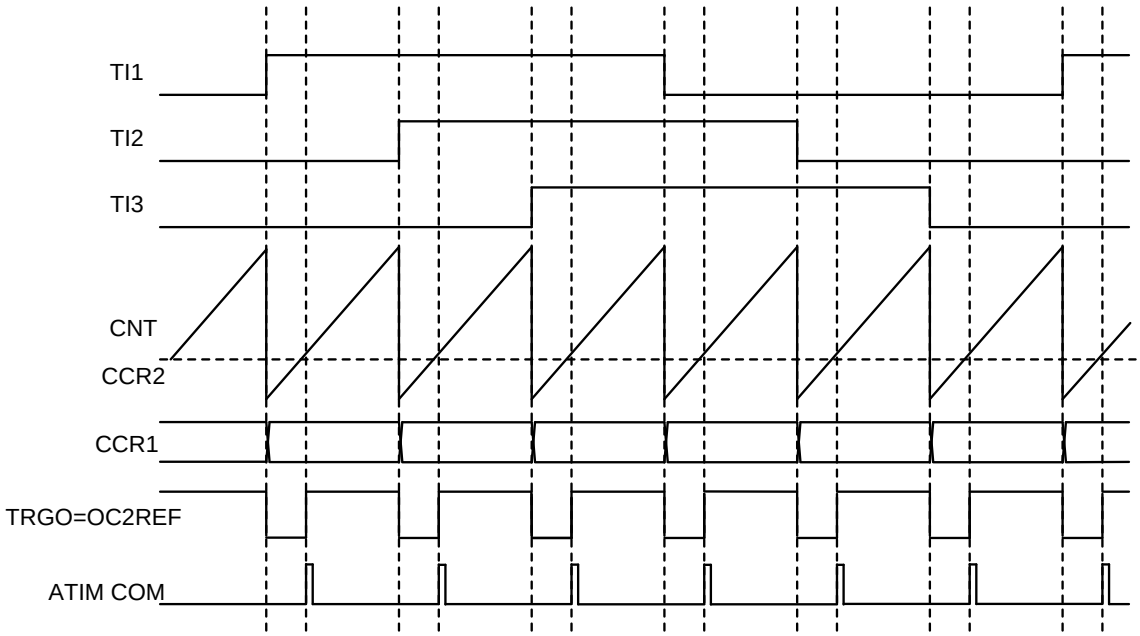


Figure 29-44 Hall Sensor Interface

29.4.24 Debug Mode

When the Cortex-M0 enters the debug mode, the timer can stop or continue to work, and its behavior is defined by the DBG_AT_STOP register of the DBG module.

During debugging, when the timer is stopped, its output will be disabled (MOE is cleared). According to the register configuration, the output signal at this time can be forced to inactive or controlled by the GPIO module.

29.5 Register

Offset	Name	Symbol
ATIM(Base address: 0x40013000)		
0x00000000	ATIM Control Register1	ATIM_CR1
0x00000004	ATIM Control Register2	ATIM_CR2
0x00000008	ATIM Slave Mode Control Register	ATIM_SMCR
0x0000000C	ATIM DMA and Interrupt Enable Register	ATIM_DIER
0x00000010	ATIM Interrupt Status Register	ATIM_ISR
0x00000014	ATIM Event Generation Register	ATIM_EGR
0x00000018	ATIM Capture/Compare Mode Register1	ATIM_CCMR1
0x0000001C	ATIM Capture/Compare Mode Register2	ATIM_CCMR2
0x00000020	ATIM Capture/Compare Enable Register	ATIM_CCER
0x00000024	ATIM Counter Register	ATIM_CNT
0x00000028	ATIM Prescaler Register	ATIM_PSC
0x0000002C	ATIM Auto-Reload Register	ATIM_ARR
0x00000030	ATIM Repetition Counter Register	ATIM_RCR
0x00000034	ATIM Capture/Compare Register1	ATIM_CCR1
0x00000038	ATIM Capture/Compare Register2	ATIM_CCR2
0x0000003C	ATIM Capture/Compare Register3	ATIM_CCR3
0x00000040	ATIM Capture/Compare Register4	ATIM_CCR4
0x00000044	ATIM Brake and Deadtime Register	ATIM_BDTR
0x00000048	ATIM DMA Control Register	ATIM_DCR
0x0000004C	ATIM DMA Access Register	ATIM_DMAR
0x00000060	ATIM Brake Control Register	ATIM_BKCR

29.5.1 ATIM Control Register1 (ATIM_CR1)

NAME	ATIM_CR1							
Offset	0x00000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						CKD	
access	U-0						R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ARPE	CMS		DIR	OPM	URS	UDIS	CEN
access	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:10	-	RFU: **Reserved, read as 0**
9:8	**CKD**	Dead Time and Digital Filter clock frequency Divider register (Divider ratio against CK_INT) (Counter Clock Divider) 00: $t_{DTS}=t_{CK\_INT}$ 01: $t_{DTS}=2*t_{CK\_INT}$ 10: $t_{DTS}=4*t_{CK\_INT}$ 11: RFU, do not program this value
7	**ARPE**	Auto-Reload Preload Enable 0: ARR preload disable 1: ARR preload enable
6:5	**CMS**	Center-Aligned Mode Select 00: Edge-aligned mode 01: Center-aligned mode 1, Output compare interrupt flags of channels configured in output are set only when the counter is counting down. 10: Center-aligned mode 2, Output compare interrupt flags of channels configured in output are set only when the counter is counting up. 11: Center-aligned mode 3, Output compare interrupt flags of channels configured in output are set both when the counter is counting up or down.
4	**DIR**	Counter Direction 0: Counter used as upcounter 1: Counter used as downcounter **Note:** This bit is read only when the timer is configured in Center-aligned mode or Encoder mode.
3	**OPM**	One Pulse Mode 0: Counter is not stopped at update event 1: Counter stops counting at the next update event (clearing the bit CEN)
2	**URS**	Update Request Selection 0: Any of the following events generate an update interrupt or DMA request if enabled. These events can be: – Counter overflow/underflow – Setting the UG bit – Update generation through the slave mode controller 1: Only counter overflow/underflow generates an update interrupt or DMA request if enabled.
1	**UDIS**	Update Disable UEV enabled. The Update (UEV) event is generated by one of the following events: – Counter overflow/underflow

bit	name	functional description
		– Setting the UG bit – Update generation through the slave mode controller UEV disabled. The Update event is not generated, shadow registers keep their value(ARR, PSC, CCRx). However the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.
0	**CEN**	Counter Enable 0: Counter disabled 1: Counter enabled **Note:** However trigger mode can set the CEN bit automatically by hardware.

29.5.2 ATIM Control Register2 (ATIM_CR2)

NAME	ATIM_CR2							
Offset	0x00000004							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1
access	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TI1S	MMS			CCDS	CCUS	-	CCPC
access	R/W-0	R/W-000			R/W-0	R/W-0	U-0	R/W-0

bit	name	functional description
31:15	**-**	RFU: **Reserved, read as 0**
14	**OIS4**	Refer to OIS1 bit
13	**OIS3N**	Refer to OIS1N bit
12	**OIS3**	Refer to OIS1 bit
11	**OIS2N**	Refer to OIS1N bit
10	**OIS2**	Refer to OIS1 bit
9	**OIS1N**	Output Idle State for OC1N 0: OC1N=0 after a dead-time when MOE=0 1: OC1N=1 after a dead-time when MOE=0
8	**OIS1**	Output Idle State for OC1 0: OC1=0 (after a dead-time if OC1N is implemented) when MOE=0

bit	name	functional description
		1: OC1=1 (after a dead-time if OC1N is implemented) when MOE=0
7	**TI1S**	Timer Input 1 Select 0: ATIM_CH1 pin is connected to TI1 input 1: ATIM_CH1, CH2, CH3 pins are connected to the TI1 input (XOR combination)
6:4	**MMS**	Master mode select, These bits allow to select the information to be sent in master mode to slave timers for synchronization (TRGO)(Master Mode Selection) 000: The UG bit from the ATIM_EGR register is used as trigger output TRGO 001: The Counter Enable signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time 010: UE The update event is selected as trigger output TRGO 011: Compare Pulse - The trigger output send a positive pulse when the CC1IF flag is to beset (even if it was already high), as soon as a capture or a compare match occurred. (TRGO). 100: Compare - OC1REF signal is used as trigger output (TRGO) 101: Compare - OC2REF signal is used as trigger output (TRGO) 110: Compare - OC3REF signal is used as trigger output (TRGO) 111: Compare - OC4REF signal is used as trigger output (TRGO) **Note:** The clock of the slave timer and ADC must be enabled prior to receiving events from the master timer, and must not be changed on-the-fly while triggers are received from the master timer.
3	**CCDS**	Capture/Compare DMA Select 0: DMA request sent when Capture/compare event occurs 1: DMA requests sent when update event occurs
2	**CCUS**	Capture/Compare Update Selection 0: When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COMG bit only 1: When capture/compare control bits are preloaded (CCPC=1), they are updated by setting the COMG bit or when an rising edge occurs on TRGI
1	**-**	RFU: **Reserved, read as 0**

bit	name	functional description
0	**CCPC**	Capture/Compare Preload Control enable 0: CCxE, CCxNE and OCxM bits are not preloaded 1: CCxE, CCxNE and OCxM bits are preloaded **Note:** This bit acts only on channels that have a complementary output.

29.5.3 ATIM Slave Mode Control Register (ATIM_SMCR)

NAME	ATIM_SMCR							
Offset	0x00000008							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ETP	ECE	ETPS		ETF			
access	R/W-0	R/W-0	R/W-00		R/W-0000			
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	MSM	TS			-	SMS		
access	R/W-0	R/W-000			U-0	R/W-000		

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15	**ETP**	External Trigger Polarity 0: Active at high level or rising edge. 1: Active at low level or falling edge.
14	**ECE**	External Clock Enable 0: External clock mode 2 disabled 1: External clock mode 2 enabled. The counter is clocked by any active edge on the ETRF signal.
13:12	**ETPS**	External Trigger Prescaler External trigger signal ETRP frequency must be at most 1/4 of TIMxCLK frequency. A prescaler can be enabled to reduce ETRP frequency. It is useful when inputting fast external clocks. 00: Prescaler OFF 01: ETRP frequency divided by 2 10: ETRP frequency divided by 4 11: ETRP frequency divided by 8
11:8	**ETF**	External Trigger Filter 0000: No filter

bit	name	functional description
		0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS/2}$, N=6 0101: $f_{SAMPLING}=f_{DTS/2}$, N=8 0110: $f_{SAMPLING}=f_{DTS/4}$, N=6 0111: $f_{SAMPLING}=f_{DTS/4}$, N=8 1000: $f_{SAMPLING}=f_{DTS/8}$, N=6 1001: $f_{SAMPLING}=f_{DTS/8}$, N=8 1010: $f_{SAMPLING}=f_{DTS/16}$, N=5 1011: $f_{SAMPLING}=f_{DTS/16}$, N=6 1100: $f_{SAMPLING}=f_{DTS/16}$, N=8 1101: $f_{SAMPLING}=f_{DTS/32}$, N=5 1110: $f_{SAMPLING}=f_{DTS/32}$, N=6 1111: $f_{SAMPLING}=f_{DTS/32}$, N=8
7	**MSM**	Master Slave Mode 0: No action 1: The effect of an event on the trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if you want to synchronize several timers on a single external event.
6:4	**TS**	Trigger Source 000: Internal Trigger 0 (ITR0) 001: Internal Trigger 1 (ITR1) 010: Internal Trigger 2 (ITR2) 011: Internal Trigger 3 (ITR3) 100: TI1 Edge Detector (TI1F_ED) 101: Filtered Timer Input 1 (TI1FP1) 110: Filtered Timer Input 2 (TI2FP2) 111: External Trigger input (ETRF) **Note:** These bits must be changed only when they are not used (e.g. when SMS=000) to avoid wrong edge detections at the transition.
3	**-**	RFU: **Reserved, read as 0**
2:0	**SMS**	Slave Mode Selection 000: Slave mode disabled - if CEN = ‘1’ then the prescaler is clocked directly by the internal clock. 001: Encoder mode 1 - Counter counts up/down on TI2FP1 edge depending on TI1 level. 010: Encoder mode 2 - Counter counts up/down on TI1FP2 edge depending on TI2level. 011: Encoder mode 3 - Counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the level of the other

bit	name	functional description
		input 100: Reset Mode - Rising edge of the selected trigger input (TRGI) reinitializes the counter and generates an update of the registers. 101: Gated Mode - The counter clock is enabled when the trigger input (TRGI) is high. The counter stops (but is not reset) as soon as the trigger becomes low. Both start and stop of the counter are controlled. 110: Trigger Mode - The counter starts at a rising edge of the trigger TRGI (but it is not reset). Only the start of the counter is controlled 111: External Clock Mode 1 - Rising edges of the selected trigger (TRGI) clock the counter.

29.5.4 ATIM DMA and Interrupt Enable Register (ATIM_DIER)

NAME	ATIM_DIER							
Offset	0x0000000C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE
access	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:20	**-**	RFU: **Reserved, read as 0**
19	**CC4BURSTEN**	DMA mode configuration Capture/compare channel 4 (CC4 Burst Enable) 0: Single mode, only access CCR 1: Burst mode, configure DCR to access address and length
18	**CC3BURSTEN**	DMA mode configuration Capture/compare channel 3 (CC3 Burst Enable) 0: Single mode, only access CCR 1: Burst mode, configure DCR to access address and length

bit	name	functional description
17	**CC2BURSTEN**	DMA mode configuration Capture/compare channel 2 (CC2 Burst Enable) 0: Single mode, only access CCR 1: Burst mode, configure DCR to access address and length
16	**CC1BURSTEN**	DMA mode configuration Capture/compare channel 1 (CC1 Burst Enable) 0: Single mode, only access CCR 1: Burst mode, configure DCR to access address and length
15	**-**	RFU: **Reserved, read as 0**
14	**TDE**	Triggered DMA Enable 0: Slaver mode, Trigger DMA request disabled 1: Slaver mode, Trigger DMA request enabled (automatically update preload register)
13	**COMDE**	COM event DMA Enable 0: COM event occurs, COM DMA request disabled 1: COM event occurs, COM DMA request enabled
12	**CC4DE**	Capture/Compare 4 DMA request enable (CC4 DMA Enable) 0: CC4 DMA request disabled 1: CC4 DMA request enabled
11	**CC3DE**	Capture/Compare 3 DMA request enable (CC3 DMA Enable) 0: CC3 DMA request disabled 1: CC3 DMA request enabled
10	**CC2DE**	Capture/Compare 2 DMA request enable (CC2 DMA Enable) 0: CC2 DMA request disabled 1: CC2 DMA request enabled
9	**CC1DE**	Capture/Compare 1 DMA request enable (CC1 DMA Enable) 0: CC1 DMA request disabled 1: CC1 DMA request enabled
8	**UDE**	Update Event DMA Enable 0: Update Event occurs, Update DMA request disabled 1: Update Event occurs, Update DMA request enabled
7	**BIE**	Break event Interrupt Enable 0: Break interrupt disabled 1: Break interrupt enabled
6	**TIE**	Trigger event Interrupt Enable 0: Trigger interrupt disabled 1: Trigger interrupt enabled
5	**COMIE**	COM event Interrupt Enable 0: COM interrupt disabled 1: COM interrupt enabled
4	**CC4IE**	Capture/Compare 4 interrupt enable (CC4 Interrupt Enable) 0: CC4 interrupt disabled 1: CC4 interrupt enabled

bit	name	functional description
3	**CC3IE**	Capture/Compare 3 interrupt enable (CC3 Interrupt Enable) 0: CC3 interrupt disabled 1: CC3 interrupt enabled
2	**CC2IE**	Capture/Compare 2 interrupt enable (CC2 Interrupt Enable) 0: CC2 interrupt disabled 1: CC2 interrupt enabled
1	**CC1IE**	Capture/Compare 1 interrupt enable (CC1 Interrupt Enable) 0: CC1 interrupt disabled 1: CC1 interrupt enabled
0	**UIE**	Update event Interrupt Enable 0: Update interrupt disabled 1: Update interrupt enabled

29.5.5 ATIM Interrupt Status Register(ATIM_ISR)

NAME	**ATIM_ISR**							
Offset	0x00000010							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			CC4OF	CC3OF	CC2OF	CC1OF	-
access	U-0			R/W-0	R/W-0	R/W-0	R/W-0	U-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:13	**-**	RFU: **Reserved, read as 0**
12	**CC4OF**	Capture/Compare 4 overcapture flag (Over-Capture Interrupt Flag for CC4, write 1 to clear) Refer to CC1OF description
11	**CC3OF**	Capture/Compare 3 overcapture flag (Over-Capture Interrupt Flag for CC3, write 1 to clear) Refer to CC1OF description
10	**CC2OF**	Capture/Compare 2 overcapture flag (Over-Capture Interrupt Flag for CC2, write 1 to clear) Refer to CC1OF description
9	**CC1OF**	Capture/Compare 1 overcapture flag (Over-Capture Interrupt

bit	name	functional description
		Flag for CC1, write 1 to clear) This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to '0'. 0: No overcapture has been detected. 1: The counter value has been captured while CC1IF flag was already set
8	**-**	RFU: **Reserved, read as 0**
7	**BIF**	Break Interrupt Flag, write 1 to clear
6	**TIF**	Trigger Interrupt Flag, write 1 to clear
5	**COMIF**	COM Interrupt Flag, write 1 to clear
4	**CC4IF**	CC4 Interrupt Flag, write 1 to clear Refer to CC1IF description
3	**CC3IF**	CC3 Interrupt Flag, write 1 to clear Refer to CC1IF description
2	**CC2IF**	CC2 Interrupt Flag, write 1 to clear Refer to CC1IF description
1	**CC1IF**	Capture/Compare 1 interrupt flag (CC1 Interrupt Flag, write 1 to clear) If channel CC1 is configured as output: This flag is set by hardware when the counter matches the compare value, it is cleared by software. 0: No match. 1: The content of the counter TIMx_CNT matches the content of the TIMx_CCR1 register. If channel CC1 is configured as input: This bit is set by hardware on a capture. It is cleared by software or by reading theTIMx_CCR1 register. 0: No input capture occurred 1: The counter value has been captured in TIMx_CCR1 register (An edge has been detected on IC1 which matches the selected polarity)
0	**UIF**	Update event Interrupt Flag, write 1 to clear 0: No update occurred. 1: Update interrupt pending. This bit is set by hardware when the registers are updated: – At overflow or underflow regarding the repetition counter value (update if repetition counter= 0) and if the UDIS=0. – When CNT is reinitialized by software using the UG bit in TIMx_EGR register, if URS=0 and UDIS=0

29.5.6 ATIM Event Generation Register (ATIM_EGR)

NAME	ATIM_EGR							
Offset	0x00000014							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
access	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

bit	name	functional description
31:8	**-**	RFU: **Reserved, read as 0**
7	**BG**	Break Generate, This bit is set by software in order to generate an event, it is automatically cleared by hardware.
6	**TG**	Trigger Interrupt Flag This bit is set by software in order to generate an event, it is automatically cleared by hardware.
5	**COMG**	Capture/Compare control update generation This bit can be set by software, it is automatically cleared by hardware
4	**CC4G**	Capture/Compare 4 generation Refer to CC1G description
3	**CC3G**	Capture/Compare 3 generation Refer to CC1G description
2	**CC2G**	Capture/Compare 2 generation Refer to CC1G description
1	**CC1G**	Capture/Compare 1 generation This bit is set by software in order to generate an event, it is automatically cleared by hardware. If channel CC1 is configured as output: CC1IF flag is set, Corresponding interrupt or DMA request is sent if enabled. If channel CC1 is configured as input: The current value of the counter is captured in ATIM_CCR1 register. The CC1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. The CC1OF flag is set if the CC1IF flag was already high.

bit	name	functional description
0	**UG**	Update generation This bit can be set by software, it is automatically cleared by hardware. Reinitialize the counter and generates an update of the registers. Note that the prescaler counter is cleared too (anyway the prescaler ratio is not affected). The counter is cleared if the center-aligned mode is selected or if DIR=0 (upcounting), else it takes the auto-reload value (ATIM_ARR) if DIR=1 (downcounting)

29.5.7 ATIM Capture/Compare Mode Register1 (ATIM_CCMR1)

The channels can be used in input (capture mode) or in output (compare mode).

NAME	ATIM_CCMR1							
Offset	0x00000018							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	OC2CE	OC2M			OC2PE	OC2FE	CC2S	
	IC2F				IC2PSC		CC2S	
bit	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	
name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
access	OC1CE	OC1M			OC1PE	OC1FE	CC1S	
bit	IC1F				IC1PSC		CC1S	
name	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	

Output compare mode

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15	**OC2CE**	Output Compare 2 clear enable, refer to OC1CE (OC2 Clear Enable)
14:12	**OC2M**	Output Compare 2 mode, refer to OC1M (OC2 Mode)
11	**OC2PE**	Output Compare 2 preload enable, refer to OC1PE (OC2 Preload Enable)
10	**OC2FE**	Output Compare 2 fast enable, refer to OC1FE (OC2 Fast Enable)
9:8	**CC2S**	Output Compare 2 fast enable(CC2 Channel Selection)

bit	name	functional description
		00: CC2 channel is configured as output 01: CC2 channel is configured as input, IC2 is mapped on TI2 10: CC2 channel is configured as input, IC2 is mapped on TI1 11: Channel is configured as input, IC2 is mapped on TRC. Note: CC2S bits are writable only when the channel is OFF (CC2E = '0' in ATIM_CCER)
7	**OC1CE**	Output Compare 1 clear enable 0: OC1Ref is not affected by the ETRF Input 1: OC1Ref is cleared as soon as a High level is detected on ETRF input
6:4	**OC1M**	Output Compare 1 mode, These bits define the behavior of the output reference signal OC1REF from OC1 Mode 000: The comparison between the output compare register ATIM_CCR1 and the counter ATIM_CNT has no effect on the outputs 001: When CCR1 matches CNT, Set channel 1 to active level on match 010: When CCR1matches CNT, Set channel 1 to inactive level on match. 011: When CCR1= matches CNT, ToggleOC1REF 100: OC1REF is forced low. 101: OC1REF is forced high 110: PWM mode 1 - In upcounting, OC1REF is active as long as CNT<CCR1else inactive. In downcounting, OC1REF is inactive (OC1REF='0') as long as CNT>CCR1 else active (OC1REF='1'). 111: PWM mode 2 - In upcounting,OC1REF is inactive as long as CNT<CCR1else active. In downcounting, OC1REF is active as long as CNT>CCR1 else inactive.
3	**OC1PE**	Output Compare 1 preload enable 0: Preload register on CCR1 disabled. CCR1 can be written at anytime 1: Preload register on CCR1 enabled. Read/Write operations access the preload register. CCR1 preload value is loaded in the active register at each update event
2	**OC1FE**	Output Compare 1 fast enable 0: CC1 behaves normally depending on counter and CCR1 values even when the trigger is ON. 1: An active edge on the trigger input acts like a compare match on CC1 output. Then, OC is set to the compare level independently from the result of the comparison. OCFE acts only if the channel is configured in PWM1 or PWM2 mode.
1:0	**CC1S**	Capture/Compare 1 selection

bit	name	functional description
		This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIMx_SMCR register) Note: CC1S bits are writable only when the channel is OFF (CC1E = '0' in ATIM_CCER).

Input capture mode

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:12	**IC2F**	Input capture 2 filter
11:10	**IC2PSC**	Input capture 2 prescaler
9:8	**CC2S**	Capture/Compare 2 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC2 channel is configured as output 01: CC2 channel is configured as input, IC3 is mapped on TI2 10: CC2 channel is configured as input, IC3 is mapped on TI1 11: CC2 channel is configured as input, IC3 is mapped on TRC. **Note:** CC2S bits are writable only when the channel is OFF (CC2E = '0' in ATIM_CCER).
7:4	**IC1F**	Input capture 1 filter This bit-field defines the frequency used to sample TI1 input and the length of the digital filter appliedto TI1. 0000: No filter, sampling is done at fDTS 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS/2}$, N=6 0101: $f_{SAMPLING}=f_{DTS/2}$, N=8 0110: $f_{SAMPLING}=f_{DTS/4}$, N=6 0111: $f_{SAMPLING}=f_{DTS/4}$, N=8 1000: $f_{SAMPLING}=f_{DTS/8}$, N=6 1001: $f_{SAMPLING}=f_{DTS/8}$, N=8 1010: $f_{SAMPLING}=f_{DTS/16}$, N=5 1011: $f_{SAMPLING}=f_{DTS/16}$, N=6 1100: $f_{SAMPLING}=f_{DTS/16}$, N=8 1101: $f_{SAMPLING}=f_{DTS/32}$, N=5

bit	name	functional description
		1110: $f_{SAMPLING}=f_{DTS/32}$, N=6 1111: $f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC1PSC**	Input capture 1 prescaler 00: no prescaler, capture is done each time an edge is detected on the capture input 01: capture is done once every 2 events 10: capture is done once every 4 events 11: capture is done once every 8 events The prescaler is reset as soon as CC1E='0'
1:0	**CC1S**	Capture/Compare 1 Selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channel is configured as input, IC1 is mapped on TRC. **Note:** CC1S bits are writable only when the channel is OFF(*CC1E = '0'*)

29.5.8 ATIM Capture/Compare Mode Register2(ATIM_CCMR2)

This register is reused for two different sets of functions in output comparison and input capture configurations.

NAME	ATIM_CCMR2							
Offset	0x0000001C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	OC4CE	OC4M			OC4PE	OC4FE	CC4S	
	IC4F				IC4PSC		CC4S	
bit	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	
name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
access	OC3CE	OC3M			OC3PE	OC3FE	CC3S	
bit	IC3F				IC3PSC		CC3S	
name	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	

Output compare mode

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15	**OC4CE**	Output compare 4 clear enable, refer to OC1CE (OC4 Clear Enable)
14:12	**OC4M**	Output compare 4 preload enable, refer to OC1M (OC4 Mode)
11	**OC4PE**	Output compare 4 preload enable, refer to OC1PE (OC4 Preload Enable)
10	**OC4FE**	Output compare 4 fast enable, refer to OC1FE (OC4 Fast Enable)
9:8	**CC4S**	Capture/Compare 4 selection(CC4 Channel Selection) 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. **Note:** CC4S bits are writable only when the channel is OFF (CC4E = '0')
7	**OC3CE**	Output compare 3 clear enable(OC3 Clear Enable) 0: OC1REF is not affected by ETRF 1: Detect ETRF high voltage level, automatically clear OC1REF
6:4	**OC3M**	Output compare 3 mode, This register defines the behavior of the OC3REF signal 000: The result of the comparison between the output comparison register CCR3 and the counter CNT doesn't affect the output 001: When CCR3 matches CNT, OC3REF signal is forced high 010: When CCR3 matches CNT, OC3REF signal is forced low 011: When CCR3 matches CNT, OC3REF toggles 100: Force inactive level – OC3REF is forced low 101: Force active level - OC1REF is forced high. 110: PWM mode 1 - In upcounting, channel 3 is active as long as ATIM_CNT<ATIM_CCR3 else inactive. In downcounting, channel 3 is inactive (OC3REF='0') as long as ATIM_CNT>ATIM_CCR1 else active (OC3REF='1'). 111: PWM mode 2 - In upcounting, channel 1 is inactive as long as ATIM_CNT<ATIM_CCR1else active. In downcounting, channel 1 is active as long as ATIM_CNT>ATIM_CCR1 elseinactive.
3	**OC3PE**	Output Compare 3 preload enable 0: Preload register on ATIM_CCR3 disabled. ATIM_CCR3 can be written at anytime, thenew value is taken in account immediately. 1:Preload register on ATIM_CCR3 enabled. Read/Write

bit	name	functional description
		operations access the preloadregister. ATIM_CCR3 preload value is loaded in the shadowregister at each update event.
2	**OC3FE**	Output Compare 1 fast enable 0: OC3FE disable, trigger input doesn't affect compare output 1: OC3FE enable, The trigger input immediately changes the OC3REF to the output when the comparison values match, regardless of the actual current comparison This function is available only when the current channel is configured in PWM1 or PWM2 mode
1:0	**CC3S**	Capture/compare 3 selection (CC4 Channel Selection) 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped on TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped on TRC **Note:** CC3S bits are writable only when the channel is OFF (CC3E = '0')

Input capture mode

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:12	**IC4F**	Input capture 4 filter
11:10	**IC4PSC**	Input capture 4 prescaler
9:8	**CC4S**	Capture/Compare 4 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. **Note:** CC4S bits are writable only when the channel is OFF (CC4E = '0')
7:4	**IC3F**	Input capture 3 filter This bit-field defines sampling frequency and filter length of TI3 0000: No filter, sampling is done at fDTS 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS/2}$, N=6 0101: $f_{SAMPLING}=f_{DTS/2}$, N=8 0110: $f_{SAMPLING}=f_{DTS/4}$, N=6 0111: $f_{SAMPLING}=f_{DTS/4}$, N=8 1000: $f_{SAMPLING}=f_{DTS/8}$, N=6

bit	name	functional description
		1001: $f_{SAMPLING}=f_{DTS/8}$, N=8 1010: $f_{SAMPLING}=f_{DTS/16}$, N=5 1011: $f_{SAMPLING}=f_{DTS/16}$, N=6 1100: $f_{SAMPLING}=f_{DTS/16}$, N=8 1101: $f_{SAMPLING}=f_{DTS/32}$, N=5 1110: $f_{SAMPLING}=f_{DTS/32}$, N=6 1111: $f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC3PSC**	Input capture 3prescaler 00: No prescaler 01: Capture is done once every 2 events 10: Capture is done once every 4 events 11: Capture is done once every 8 events The prescaler is reset as soon as CC1E='0'
1:0	**CC3S**	Capture/Compare 3 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped on TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped on TRC **Note:** CC1S bits are writable only when the channel is OFF (CC1E = '0')

29.5.9 ATIM Capture/Compare Enable Register (ATIM_CCER)

NAME	ATIM_CCER							
Offset	0x00000020							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E
access	U-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:14	-	RFU: **Reserved, read as 0**
13	**CC4P**	Capture/Compare 4 output polarity, refer to CC1P
12	**CC4E**	Capture/Compare 4 output enable, refer to CC1E
11	**CC3NP**	Capture/Compare 3 complementary output polarity, refer to CC1NP
10	**CC3NE**	Capture/Compare 3 complementary output enable, refer to CC1NE
9	**CC3P**	Capture/Compare 3 output polarity, refer to CC1P
8	**CC3E**	Capture/Compare 3 output enable, refer to CC1E
7	**CC2NP**	Capture/Compare 2 complementary output polarity, refer to CC1NP
6	**CC2NE**	Capture/Compare 2 complementary output enable, refer to CC1NE
5	**CC2P**	Capture/Compare 2 output polarity, refer to CC1P
4	**CC2E**	Capture/Compare 2 output enable, refer to CC1E
3	**CC1NP**	Capture/Compare 1 complementary output polarity 0: OC1N active high. 1: OC1N active low.
2	**CC1NE**	Capture/Compare 1 complementary output enable 0: Off - OC1N is not active. OC1N level is then function of MOE, OSSI, OSSR, OIS1, OIS1Nand CC1E bits.
1	**CC1P**	Capture/Compare 1 output polarity CC1 channel configured as output: 0: OC1 active high 1: OC1 active low CC1 channel configured as input: 0: Non-inverted/rising edge 1: Inverted/falling edge
0	**CC1E**	Capture/Compare 1 output enable CC1 channel configured as output: 0: Off - OC1 is not active. 1: OC1 active CC1 channel configured as input: 0: Capture disabled. 1: Capture enabled.

29.5.10 ATIM Counter Register (ATIM_CNT)

NAME	ATIM_CNT							
Offset	0x00000024							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CNT[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CNT[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:0	**CNT**	Counter value

29.5.11 ATIM Prescaler Register (ATIM_PSC)

NAME	**ATIM_PSC**							
Offset	0x00000028							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PSC[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PSC[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:0	**PSC**	Prescaler value $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[15:0]+1)$ This is a preload register whose contents are loaded into the Shadow register when the update event occurs

29.5.12 ATIM Auto-Reload Register (ATIM_PRR)

NAME	ATIM_ARR							
Offset	0x0000002C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ARR[15:8]							
access	R/W-1111 1111							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ARR[7:0]							
access	R/W-1111 1111							

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:0	**ARR**	Auto-reload value ARR is the value to be loaded in the shadow register when update event occurs

29.5.13 ATIM Repetition Counter Register (ATIM_RCR)

NAME	ATIM_RCR							
Offset	0x00000030							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	REP[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:8	**-**	RFU: **Reserved, read as 0**
7:0	**REP**	Repetition counter value

bit	name	functional description
		Each time the REP_CNT related downcounter reaches zero, an update event is generated and it restarts counting from REP value, triggering update event when REP=0

29.5.14 ATIM Capture/Compare Register1 (ATIM_CCR1)

NAME	ATIM_CCR1							
Offset	0x00000034							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR1[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR1[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:0	**CCR1**	Capture/Compare channel 1 Register **If channel CC1 is configured as output:** CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value). **If channel CC1 is configured as input:** CCR1 is the counter value transferred by the last input capture 1 event (IC1).CCR1 is read only

29.5.15 ATIM Capture/Compare Register2 (ATIM_CCR2)

NAME	ATIM_CCR2							
Offset	0x00000038							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

name	CCR2[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR2[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**CCR2**	Capture/Compare channel 2 Register **If channel CC2 is configured as output:** CCR2 is the value to be loaded in the actual capture/compare 2 register (preload value) **If channel CC2 is configured as input:** CCR2 is the counter value transferred by the last input capture 2 event (IC2). CCR2 is read only

29.5.16 ATIM Capture/Compare Register3 (ATIM_CCR3)

NAME	ATIM_CCR3							
Offset	0x0000003C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR3[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR3[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**CCR3**	Capture/Compare channel 3 Register **If channel CC3 is configured as output:** CCR3 is the value to be loaded in the actual capture/compare 3 register (preload value).The active capture/compare register contains the value to be compared to the counter ATIM_CNT and signaled on OC3 output. **If channel CC3 is configured as input:** CCR3 is the counter value transferred by the last input capture

bit	name	functional description
		3 event (IC3). CCR3 is read only

29.5.17 ATIM Capture/Compare Register4 (ATIM_CCR4)

NAME	ATIM_CCR4							
Offset	0x00000040							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR4[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR4[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**CCR4**	Capture/Compare channel 4 Register **If channel CC4 is configured as output:** CCR4 is the value to be loaded in the actual capture/compare 4 register (preload value).The active capture/compare register contains the value to be compared to the counter ATIM_CNT and signaled on OC4 output. **If channel CC4 is configured as input:** CCR4 is the counter value transferred by the last input capture 4 event (IC4)，CCR4 is read only

29.5.18 ATIM Brake and Deadtime Register (ATIM_BDTR)

NAME	ATIM_BDTR							
Offset	0x00000044							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

name	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK	
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DTG							
access	R/W-0000 0000							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15	**MOE**	Master Output Enable 0: OC and OCN outputs are disabled or forced to idle state. 1: OC and OCN outputs are enabled if their respective enable bits are set (CCxE, CCxNE in ATIM_CCER register)
14	**AOE**	Automatic Output Enable 0: MOE can be set only by software 1: MOE can be set by software or automatically at the next update event (if the break input is not be active)
13	**BKP**	Break Polarity 0: Break input BRK is active low 1: Break input BRK is active high
12	**BKE**	Break enable 0: Break inputs (BRK and CSS clock failure event) disabled 1; Break inputs (BRK and CSS clock failure event) enabled
11	**OSSR**	Off-state selection for Run mode This bit is used when MOE=1 on channels having a complementary output which are configured as outputs. OSSR is not implemented if no complementary output is implemented in the timer. 0: When inactive, OC/OCN outputs are disabled, OC/OCN don't driver GPIO 1: When inactive, OC/OCN outputs are enabled with their inactive level as soon as CCxE=1or CCxNE=1. Then, OC/OCN driver GPIO inactive
10	**OSSI**	Off-State Select in IDLE mode This bit is used when MOE=0 on channels configured as outputs 0: When inactive, OC/OCN outputs are disabled ,OC/OCN don't driver GPIO 1: When inactive, OC/OCN outputs are forced first with their idle level as soon as CCxE=1 or CCxNE=1. OC/OCN driver GPIO idle, After the dead zone time ends, the system starts invalid
9:8	**LOCK**	register write LOCK 00: LOCK OFF - No bit is write protected 01: LOCK Level 1 = DTG bits in ATIM_BDTR register, OISx and

bit	name	functional description
		OISxN bits in ATIM_CR2register and BKE/BKP/AOE bits in ATIM_BDTR register can no longer be written. 10: LOCK Level 2 = LOCK Level 1 + CC Polarity bits，CCxP, CCxNP, OSSR, OSSI can no longer be written. 11: can no longer be written，OcxM, OcxPE can no longer be written. **Note:** The LOCK bits can be written only once after the reset. Once the ATIM_BDTR register has been written, their content is frozen until the next reset.
7:0	**DTG**	Dead-time generator setup This bit-field defines the duration of the dead-time inserted between the complementary outputs. DT correspond to this duration. 000/001/010/011: DT=DTG[7:0] * t_{DTS} 100/101: DT=(64+DTG[5:0]) * 2 * t_{DTS} 110: DT=(32+DTG[4:0]) * 8 * t_{DTS} 111: DT=(32+DTG[4:0]) * 16 * t_{DTS}

29.5.19 ATIM DMA Control Register (ATIM_DCR)

NAME	ATIM_DCR							
Offset	0x00000048							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			DBL				
access	U-0			R/W-0 0000				
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			DBA				
access	U-0			R/W-0 0000				

bit	name	functional description
31:13	**-**	RFU: **Reserved, read as 0**
12:8	**DBL**	DMA Burst Length This 5-bit vector defines the number of DMA transfers 00000: 1 transfer 00001: 2 transfers 00010: 3 transfers

bit	name	functional description
		00011: 4 transfers 00100: 5 transfers 00101: 6 transfers 00110: 7 transfers 00111: 8 transfers 01000: 9 transfers 01001: 10 transfers 01010: 11 transfers 01011: 12 transfers 01100:13 transfers 01101: 14transfers 01110: 15 transfers 01111: 16 transfers 10000: 17 transfers 10001: 18 transfers Other: Invalid value, forbidden to write
7:5	-	RFU: **Reserved, read as 0**
4:0	**DBA**	DMA base address, This 5-bits vector defines the base-address for DMA transfers 00000: ATIM_CR1 00001: ATIM_CR2 00010: ATIM_SMCR **……** **Note:** When DBA+DBL exceeds the ATIM register address range, the actual burst will stop automatically after transmission to the ATIM highest register address, that is, the burst length will shorten.

29.5.20 ATIM DMA Access Register (ATIM_DMAR)

NAME	ATIM_DMAR							
Offset	0x0000004C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	DMAR[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DMAR[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DMAR[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

name	DMAR[7:0]
access	R/W-0000 0000

bit	name	functional description
31:0	**DMAR**	DMA burst access Register When using DMA Burst transport, set the DMA channel peripheral address to ATIM_DMAR and ATIM will make multiple DMA requests based on the DBL value

29.5.21 ATIM Brake Control Register (ATIM_BKCR)

NAME	ATIM_BKCR							
Offset	0x00000060							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						BRK2GATE	BRK1GATE
access	U-0						R/W-1	R/W-1
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	BRKF				BRKCOMB	HFDET_BRKEN	SVD_BRKEN	COMP_BRKEN
access	R/W-0000				R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:10	-	RFU: **Reserved, read as 0**
9	**BRK2GATE**	Break 2 Gate 0: ATIM_BRK2 is set 0 1: No gating
8	**BRK1GATE**	Break 1 Gate 0: ATIM_BRK1 is set 0 1: No gating
7:4	**BRKF**	Break Filter 0000: No filter 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS/2}$, N=6 0101: $f_{SAMPLING}=f_{DTS/2}$, N=8

bit	name	functional description
		0110: $f_{SAMPLING}=f_{DTS/4}$, N=6 0111: $f_{SAMPLING}=f_{DTS/4}$, N=8 1000: $f_{SAMPLING}=f_{DTS/8}$, N=6 1001: $f_{SAMPLING}=f_{DTS/8}$, N=8 1010: $f_{SAMPLING}=f_{DTS/16}$, N=5 1011: $f_{SAMPLING}=f_{DTS/16}$, N=6 1100: $f_{SAMPLING}=f_{DTS/16}$, N=8 1101: $f_{SAMPLING}=f_{DTS/32}$, N=5 1110: $f_{SAMPLING}=f_{DTS/32}$, N=6 1111: $f_{SAMPLING}=f_{DTS/32}$, N=8
3	**BRKCOMB**	Break Combination 0: Two break signal 'xor' 1: Two break signal 'and'
2	**HFDET_BRKEN**	HFDET Break Enable 0: HFDET_BRKEN disable 1: HFDET_BRKEN enable
1	**SVD_BRKEN**	SVD Break Enable 0: SVD_BRKEN disable 1: SVD_BRKEN enable
0	**COMP_BRKEN**	Comparator Break Enable 0: COMP_BRKEN disable 1: COMP_BRKEN enable

30 General Purpose Timers (GPTIM0,1,2)

30.1 Introduction

FM33LG0 contains 3 general purpose timers.

The general-purpose timer includes a 16bit auto-reload counter and a programmable prescaler.

General-purpose timers can support a variety of applications, including capture, output compare, and PWM.

30.2 Features

- 16bit up, down, up/down auto-reload counter
- 16bit programmable prescaler, support real-time adjustment of counting clock frequency division
- 4 independent channels can be used for input capture, output compare, PWM (edge or center-aligned mode), one pulse output
- Support cascading with other timers
- Support to generate interrupts when the following events occur:
  – Counter overflow, counter initialization (software or hardware trigger)
  – Trigger events (counter start, stop, initialization, internal and external triggers)
  – Input capture
  – Output compare

30.3 Block Diagram

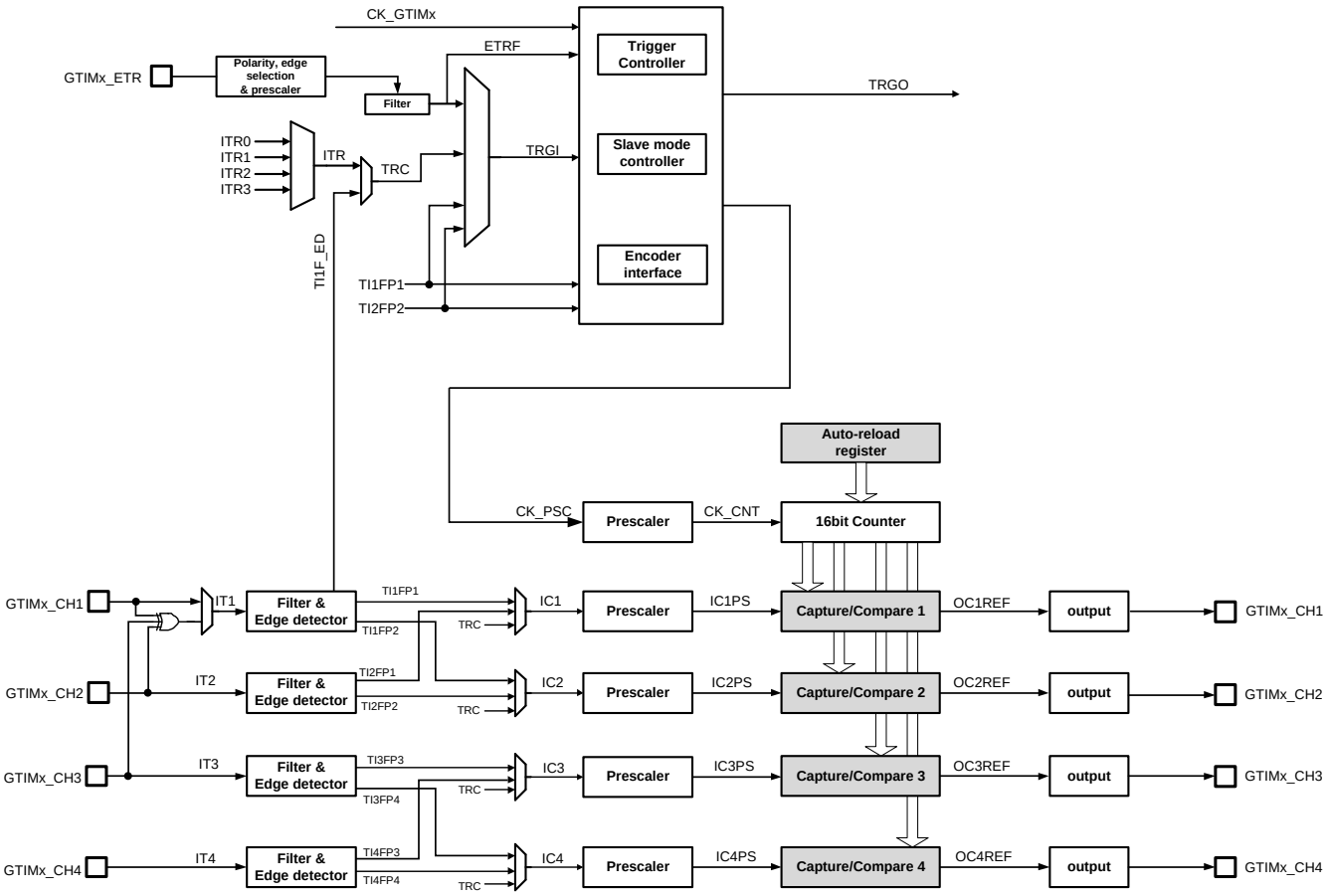


Figure 30-1 General Timer Architecture Schematic Diagram

30.4 Functional Description

30.4.1 Time-Base Unit

The main block of the programmable timer is a 16-bit counter with its related auto-reload register. The counter can count up, down or both up and down. The counter clock used APBCLK can be divided by a prescaler.

The counter, the auto-reload register and the prescaler register can be written or read by software. This is true even when the counter is running.

The time-base unit includes:

- Counter Register (GPTIM_CNT)
- Prescaler Register (GPTIM_PSC)
- Auto-Reload Register (GPTIM_ARR)

The auto-reload register (GPTIM_ARR)is preloaded. Writing to or reading from the auto-reload register(ARPE) accesses the preload register. When ARPE equals 0, the ARR register will be written and the written data will be directly passed to the shadow register. When ARPE equals1, writing event will be performed to the ARR register and the data will be transferred to the shadow register when the update event (GPTIM_CNT up overflow or down overflow) occurs. Software can also trigger ARR update (UEV) through operation of register.

The operating clock of the GPTIM_CNT is driven by the divider clock generated by GPTIM_PSC, and the CNT starts counting only when the Counter Enable Register (CEN) is set. When CNT equals ARR, the current round of counting ends and the update event is triggered.

The prescaler can divide the counter clock (APBCLK) frequency by any factor between 1 and 65536.The PSC register is also cached, and rewriting the PSC does not actually rewrite the shadow register; it is updated from the PSC to the shadow register only when a new update event arrives. Therefore, during CNT counting, software can rewrite the PSC in real time, and the new prescale ratio will be adopted when the next update event occurs.

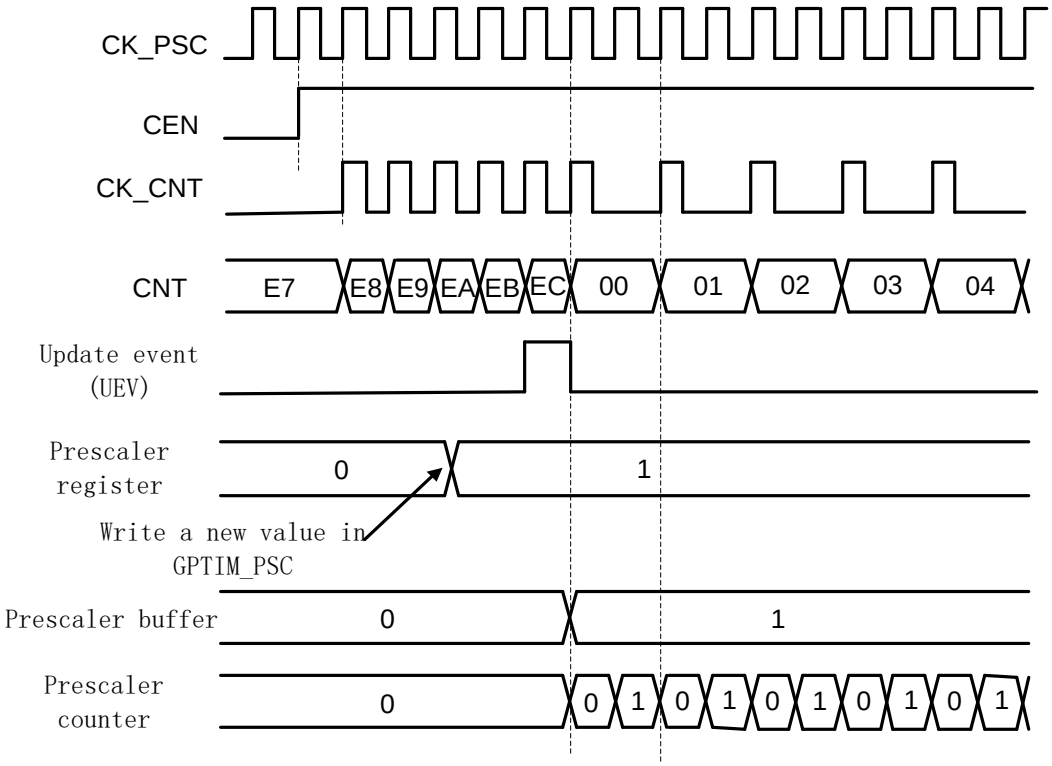


Figure 30-2 Counter Timing Diagram with Prescaler Division Change from 1 to 2

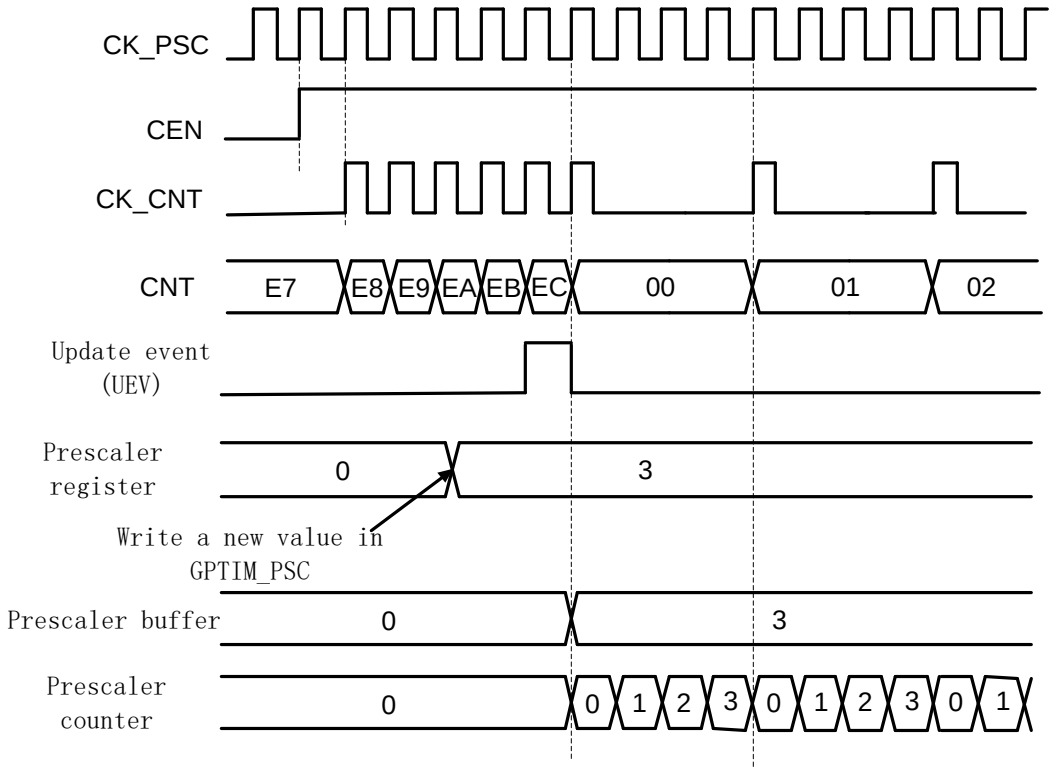


Figure 30-3 Counter Timing Diagram with Prescaler Division Change from 1 to 4

30.4.2 Counter Mode

The timer supports Upcounting mode, Downcounting mode and Center-aligned mode (up/down counting).

Upcounting mode

In upcounting mode, the counter counts from 0 to the auto-reload value (content of the ARR register), then restarts from 0 and generates a counter overflow event.

An Update event can be generated at each counter overflow or by setting the UG bit. The CNT and prescaler counters are automatically cleared to zero. Setting the UG register to trigger the UIF (Update Interrupt Flag) interrupt flag setting is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS register so that the value in the preload register is not updated to the working register.

When an update event occurs, the following registers are updated and the UIF is set:

- ARR shadow register is updated to the content of GPTIM_ARR
- PSC shadow register is updated to the content of GPTIM_PSC

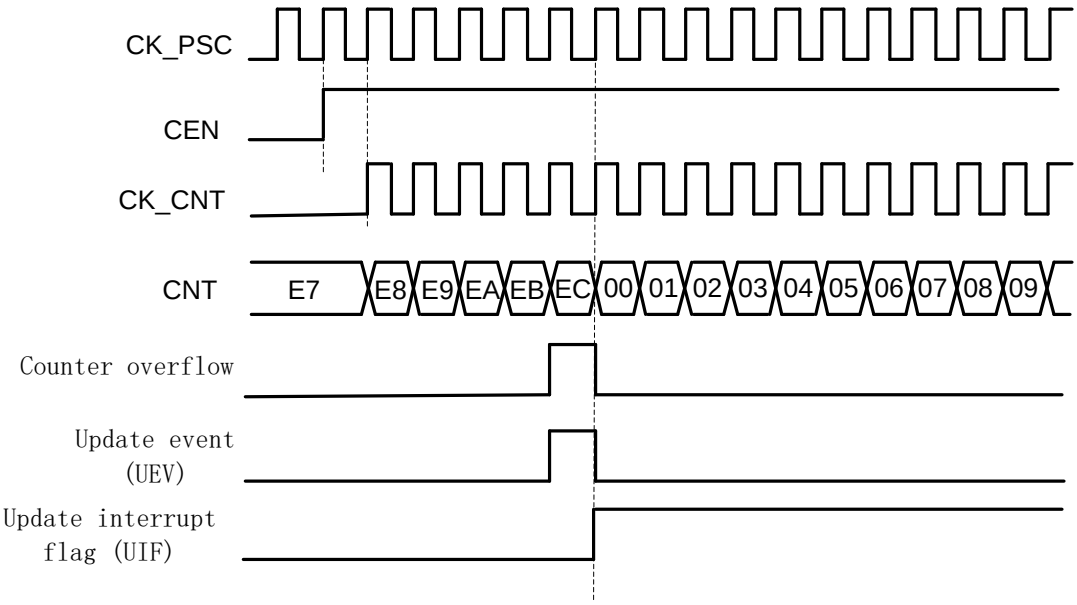


Figure 30-4 Upcounting, Internal Clock Divided by 1

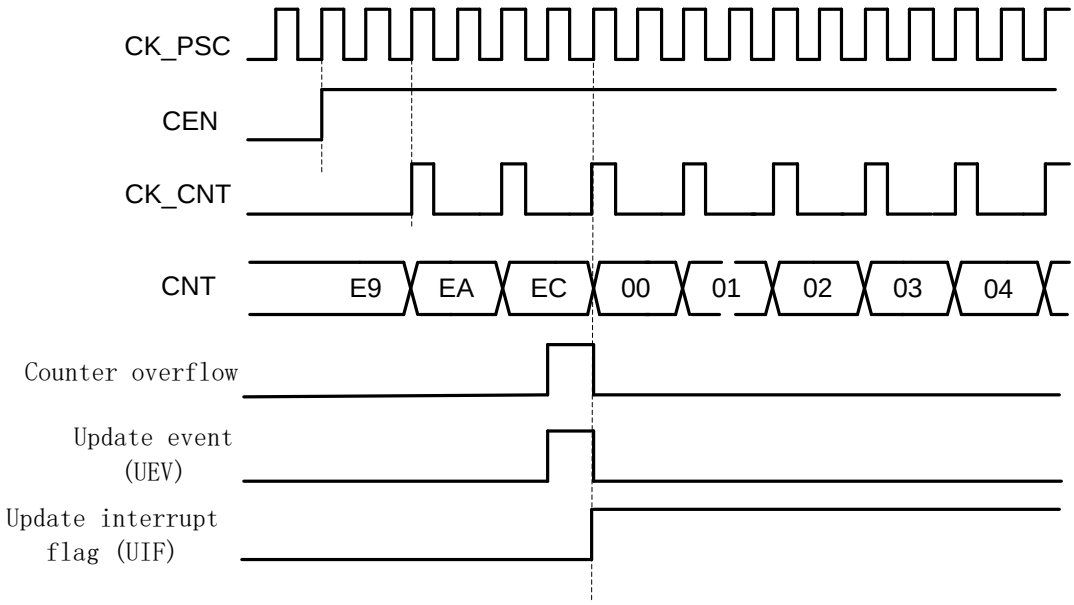


Figure 30-5 Upcounting, Internal Clock Divided by 2

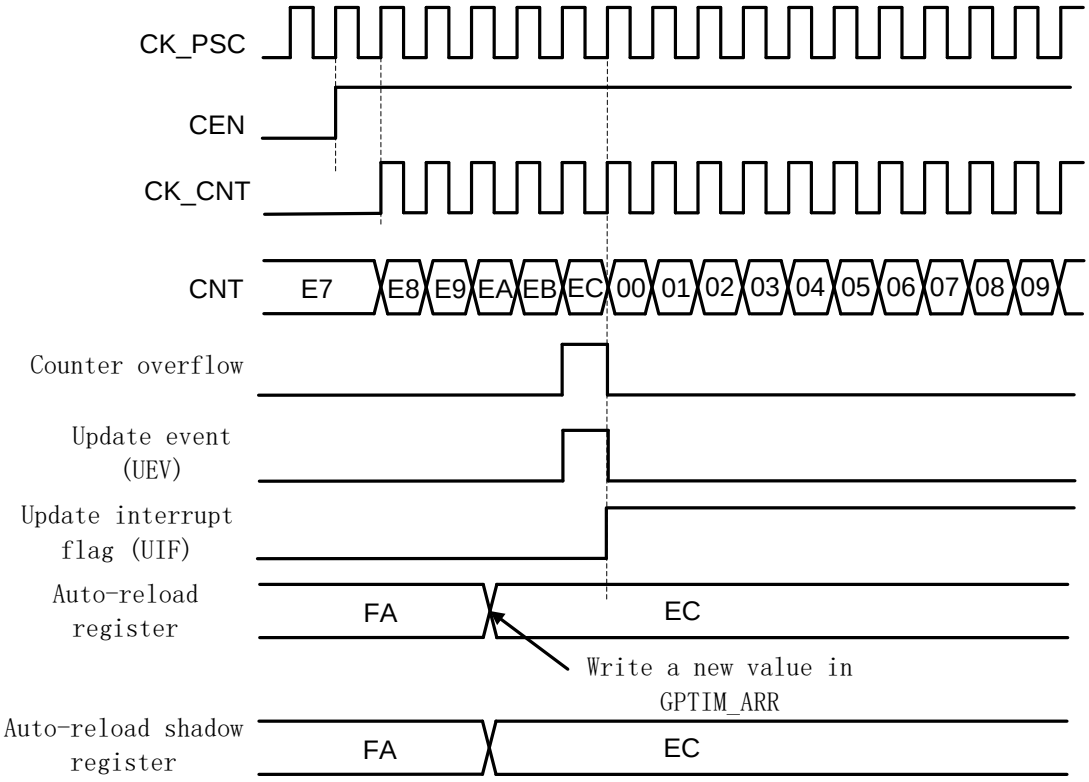


Figure 30-6 Counter Timing Diagram, Update Event When ARPE=0 (ARR is not Preloaded)

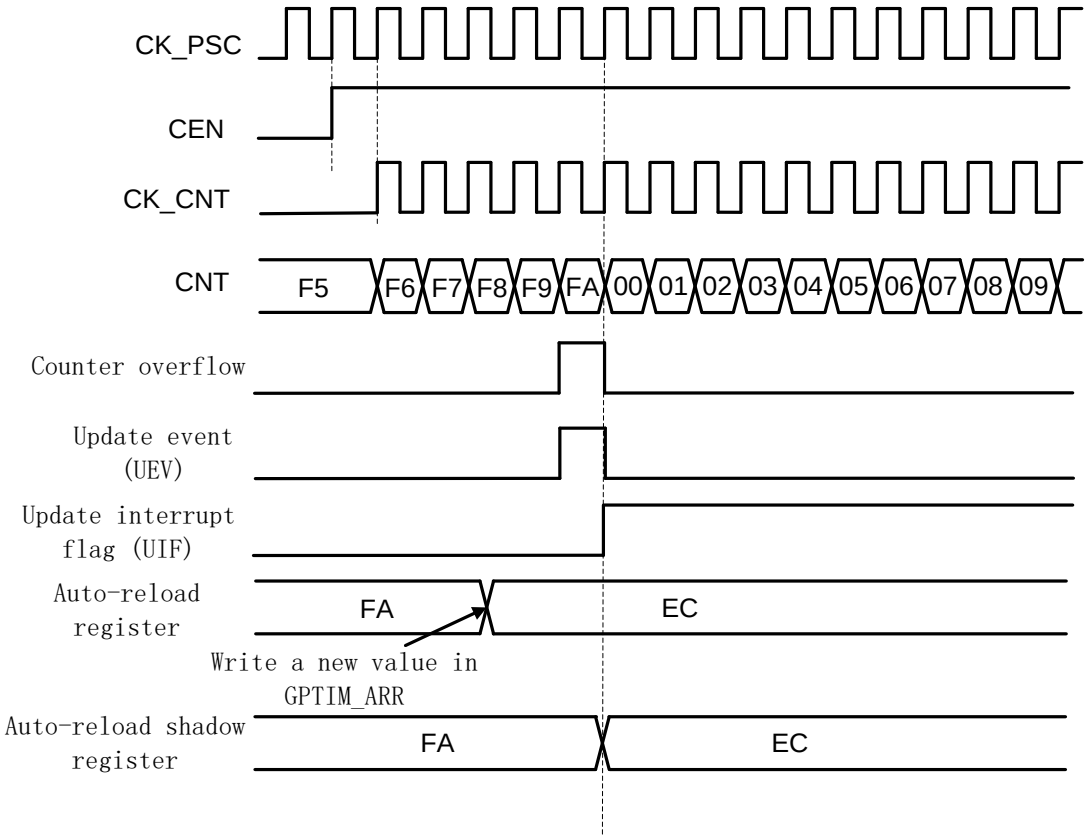


Figure 30-7 Counter Timing Diagram, Update Event When ARPE=1 (ARR is preloaded)

Downcounting mode

In downcounting mode, the counter counts from the auto-reload value (content of the TIMx_ARR register) down to 0, then restarts from the auto-reload value and generates a counter underflow event. An Update event can be generated at each counter overflow or by setting the UG bit. The CNT and prescaler counters are automatically cleared to zero. Setting the UG register to trigger the UIF (Update Interrupt Flag) interrupt flag setting is determined by the setting of the URS register.

The update event can be disabled by setting the UDIS register so that the value in the preload register is not updated to the working register.

When an update event occurs, the following registers are updated and the UIF is set:

- ARR shadow register is updated to the content of GPTIM_ARR
- PSC shadow register is updated to the content of GPTIM_PSC

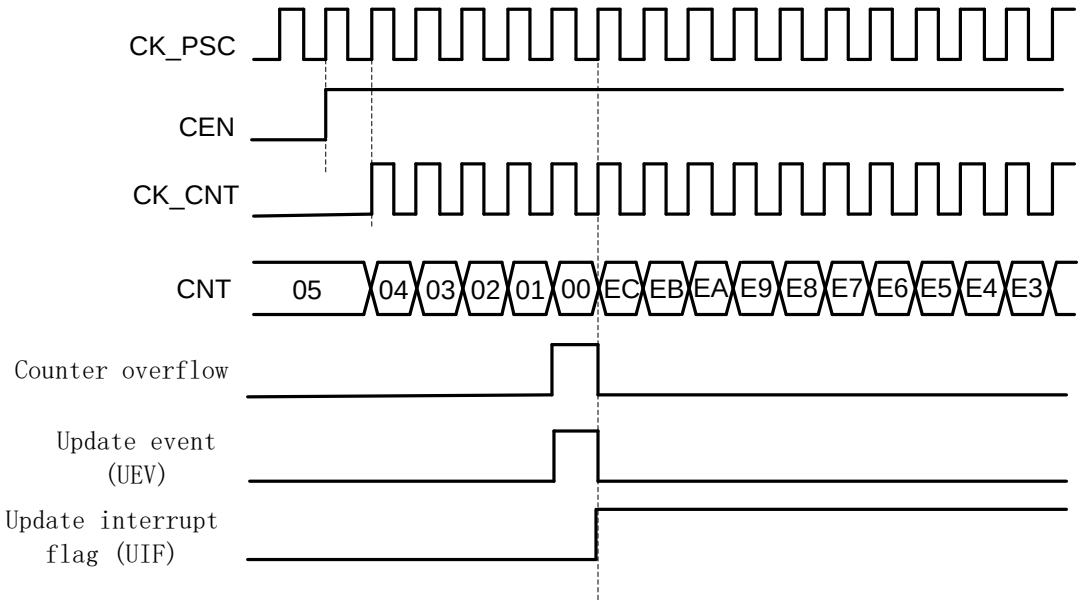


Figure 30-8 Downcounting, Internal Clock Divided by 1

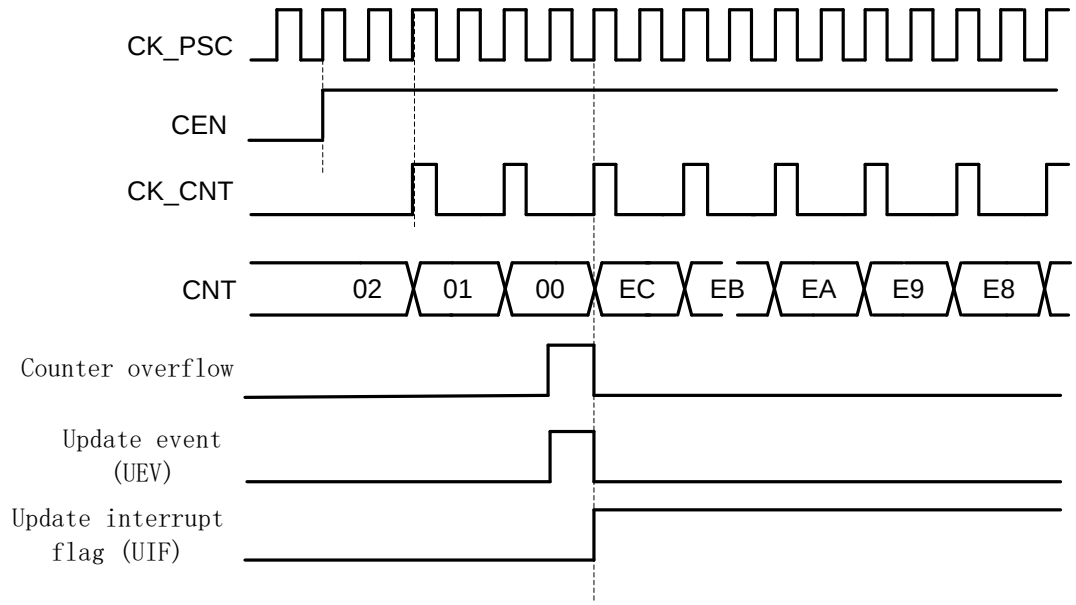


Figure 30-9 Downcounting, Internal Clock Divided by 2

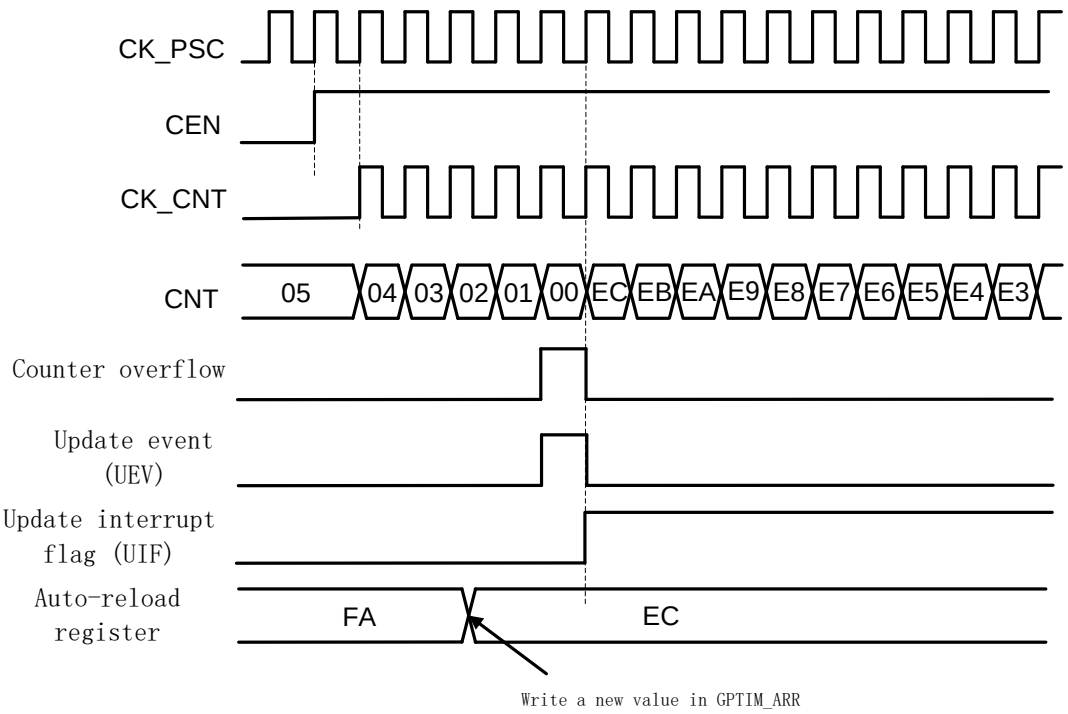


Figure 30-10 Downcounting, Update Event

Center-aligned mode (Up/Down counting)

In center-aligned mode, the counter counts from 0 to the auto-reload value (content of the ARR register) – 1, generates a counter overflow event, then counts from the auto-reload value down to 1 and generates a counter underflow event. Then it restarts counting from 0.

Center-aligned mode is active when the CMS bits are not equal to00. The Output compare interrupt flag of channels configured in output is set when CMS=01; the counter counts up when CMS=10; the counter counts up and down when CMS=11.

In this mode, the direction bit (DIR) cannot be written. It is updated by hardware and gives the current direction of the counter.

Counters update the shadow registers of ARR, PSC on both overflow and underflow events.

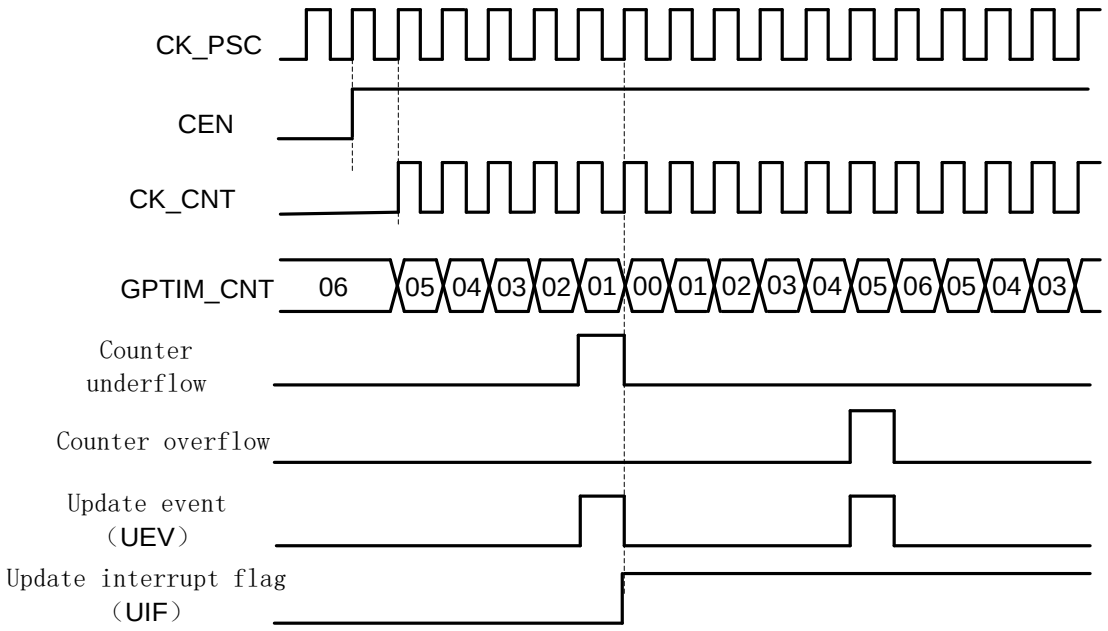


Figure 30-11 Center-Aligned Mode, GPTIM_PCS=0, GPTIM_ARR=0x6

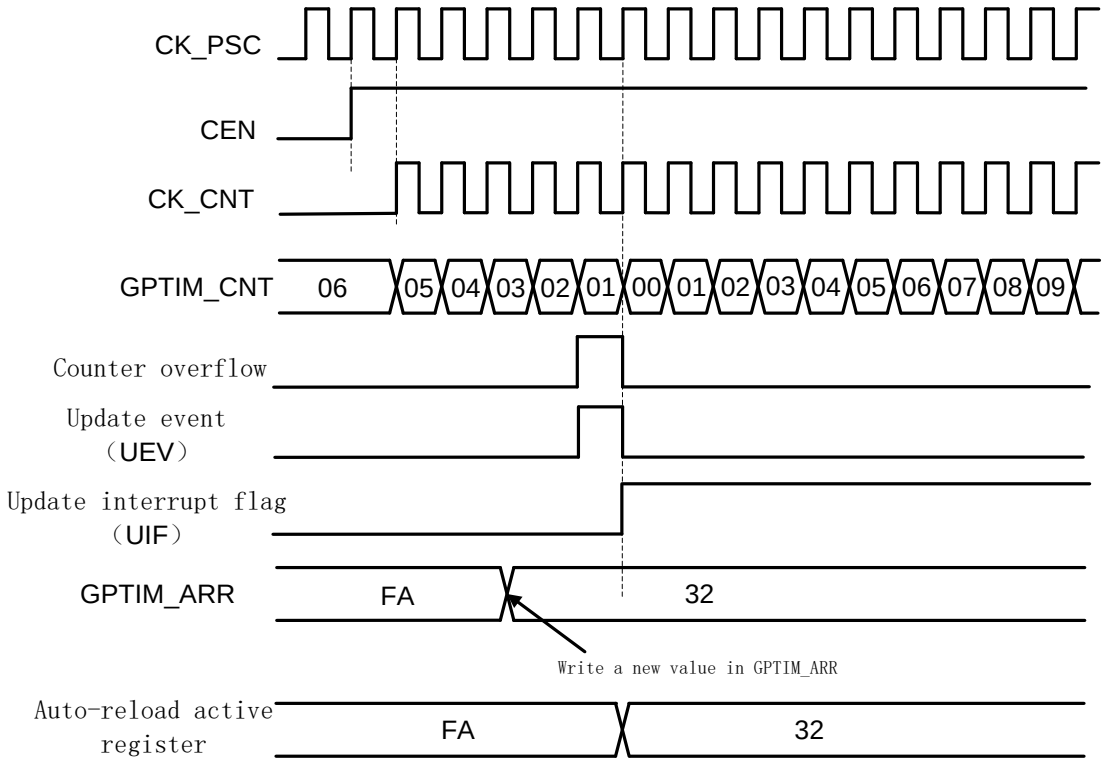


Figure 30-12 Center-Aligned Mode, Update Event with ARPE=1 (Counter Underflow)

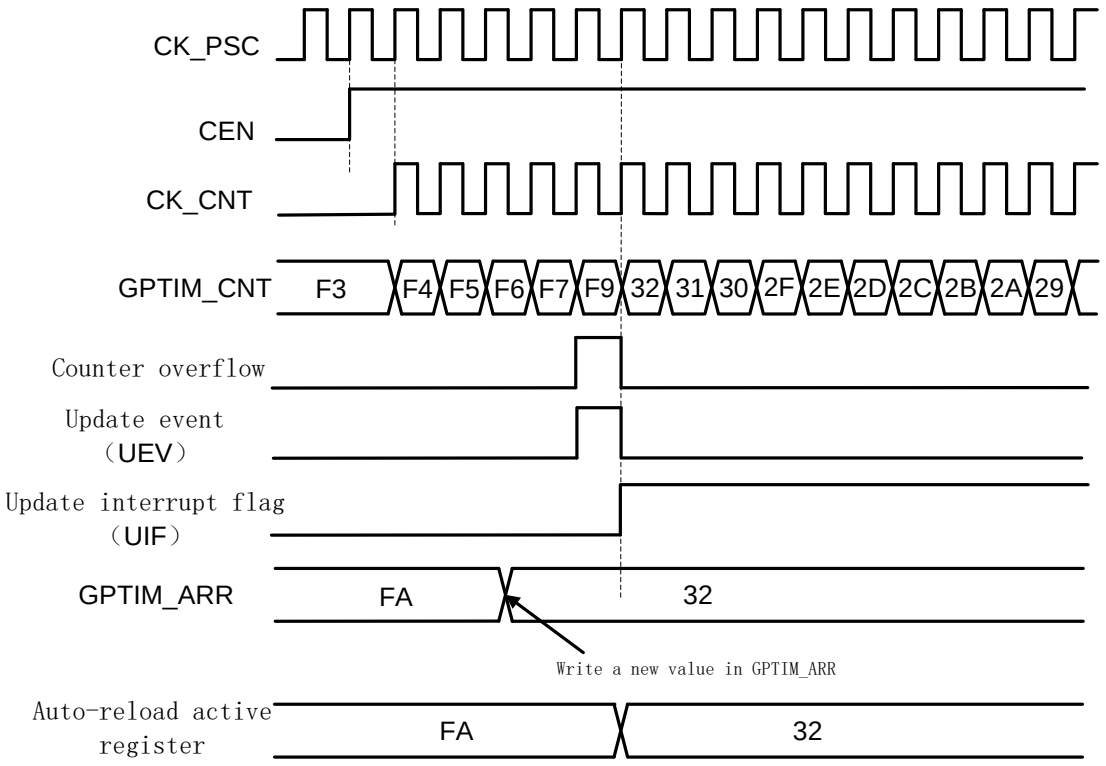


Figure 30-13 Center-Aligned Mode, Update Event with ARPE=1 (Counter Overflow)

30.4.3 Clock Select

The counter clock can be provided by the following clock sources:

- Internal clock (APBCLK)
- External clock mode1: External input pin (Tix)
- External clock mode2: External trigger input (ETR)
- Internal trigger inputs (ITRx): Using TGO of one timer

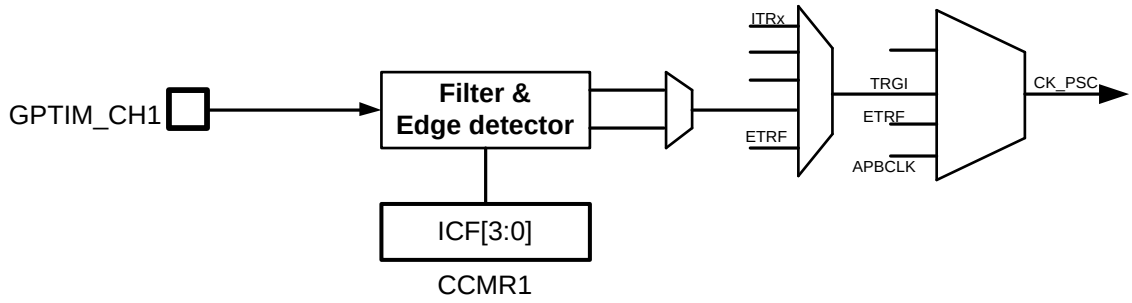


Figure 30-14 GPTIM Clock Select

30.4.3.1 Internal Clock Mode

In the internal clock mode, the slave mode is disabled (SMS=000), and the register bits such as CEN, DIR, and UG are all controlled by software.

After software operation of UG register, the counter value will be reinitialized after the update signal is synchronized by CLK_PSC.

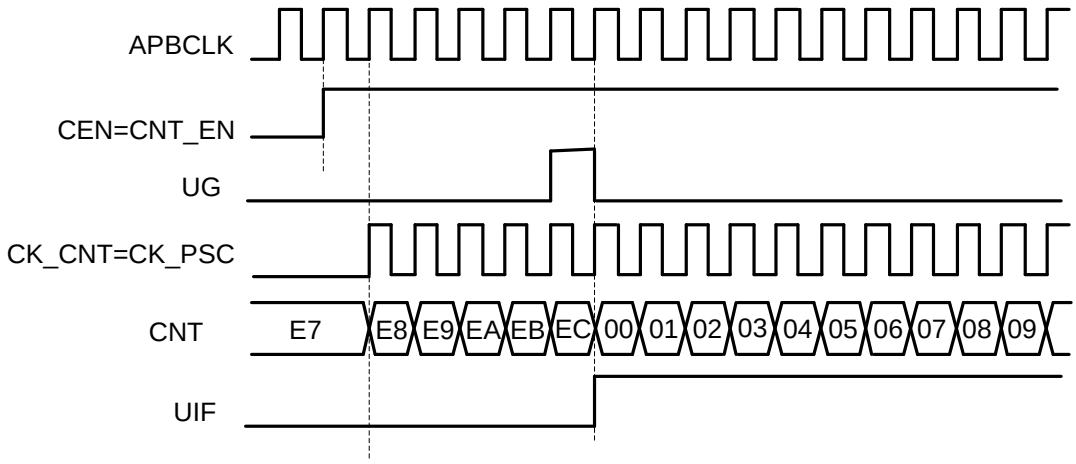


Figure 30-15 Internal Clock Divided by 1

30.4.3.2 External Clock Mode 1

This mode is selected when SMS=111. The counter can count at each rising or falling edge on a selected input.

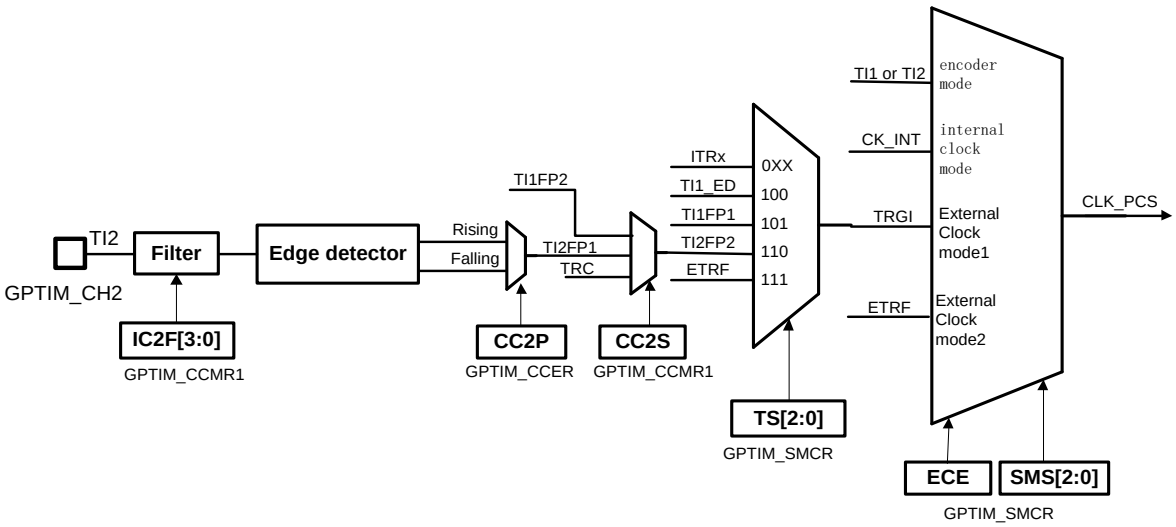


Figure 30-16 TI2 External Clock Connection Example

The external input signal will go through the synchronization process of the internal clock before triggering the counter count, while the valid edge of the input signal will trigger the TIF flag

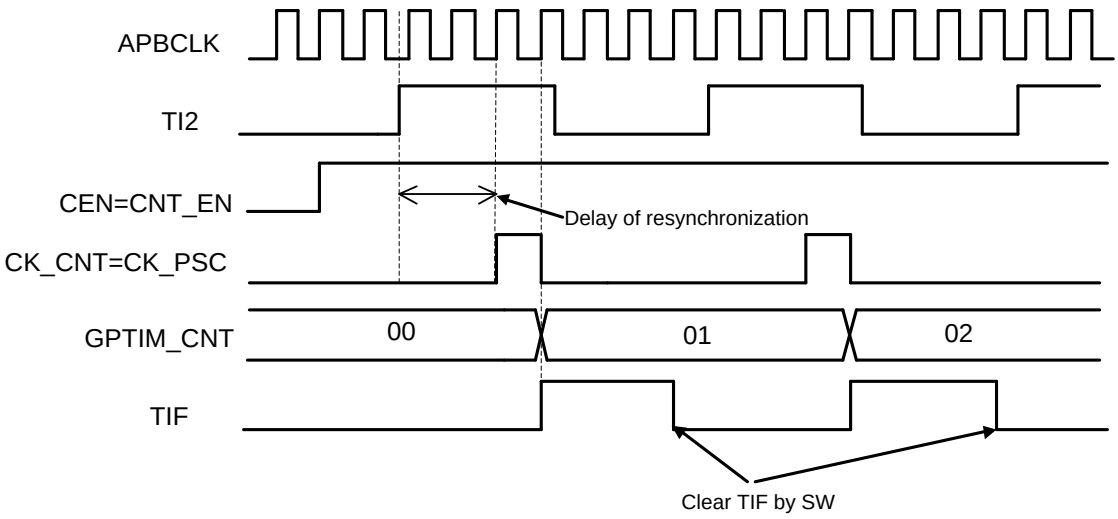


Figure 30-17Control Circuit in External Clock Mode 1

When using external clock counting, the GPTIM's internal clock (APBCLK) still keeps enabled, because the APB_CLK is used to synchronize and filter the external input clock. In external clock mode 1, the external input clock is first filtered and edge-selected to get a valid counting edge, which is input to the prescaler module as a valid operating clock (CLK_PSC).

The external clock synchronization uses a simple 2-stage flip-flop structure. And in order to avoid sub-stability it is required that an external input clock width is greater than at least 2 APB_CLK cycles.

Only the inputs of channels 1 and 2 can be used as clock inputs in this mode, and the required configuration is as follows:

- In the GPIO module, configure the corresponding pin as GPTIM_CH2
- Turn off the channel enable, configure GPTIM_CCER.CC2E=0, make sure than the channel configuration can be successful
- Select input channel, configure GPTIM_CCMR1.CC2S=01, IC2 is mapped to TI2
- Select the count valid edge, configure GPTIM_CCER.CC2P=0, select the upper or lower edge
- Configure input filtering time, configure GPTIM_CCMR1.IC2F[3:0] (IC2F=0000, no input filtering)
- Enable external clock mode 1, configure GPTIM_SMCR.SMCR=111
- Select trigger input source, configure GPTIM_SMCR.TS=110, select TI2 as trigger input

source

- Turn on channel enable, configure GPTIM_CCER.CC2E=1
- Enable counter, configure GPTIM_CR1.CEN=1

The following figure shows an example of a typical external clock mode 1:

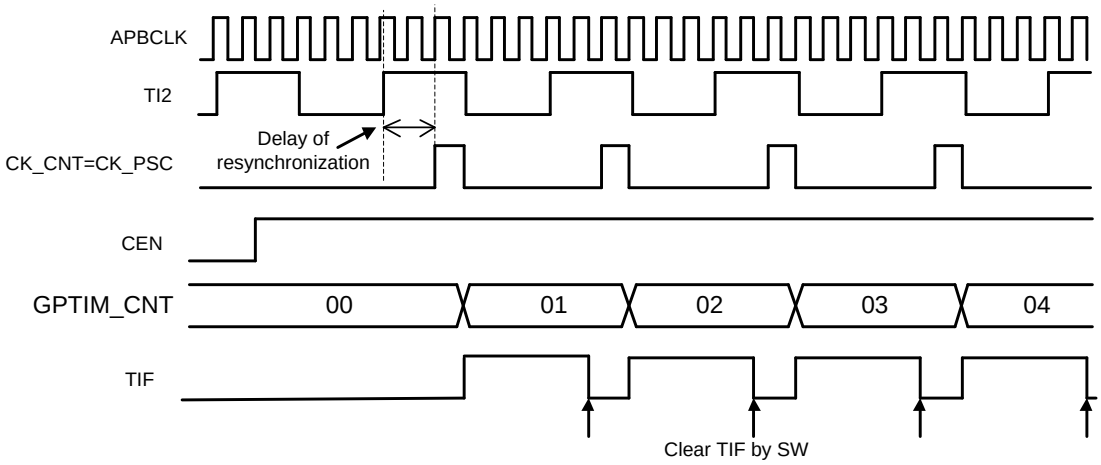


Figure 30-18 Control Circuit in External Clock Mode 1

30.4.3.3 External Clock Mode 2

This mode uses the rising or falling edge (double edge is not supported) of the input signal of GPTIM_ETR to count.

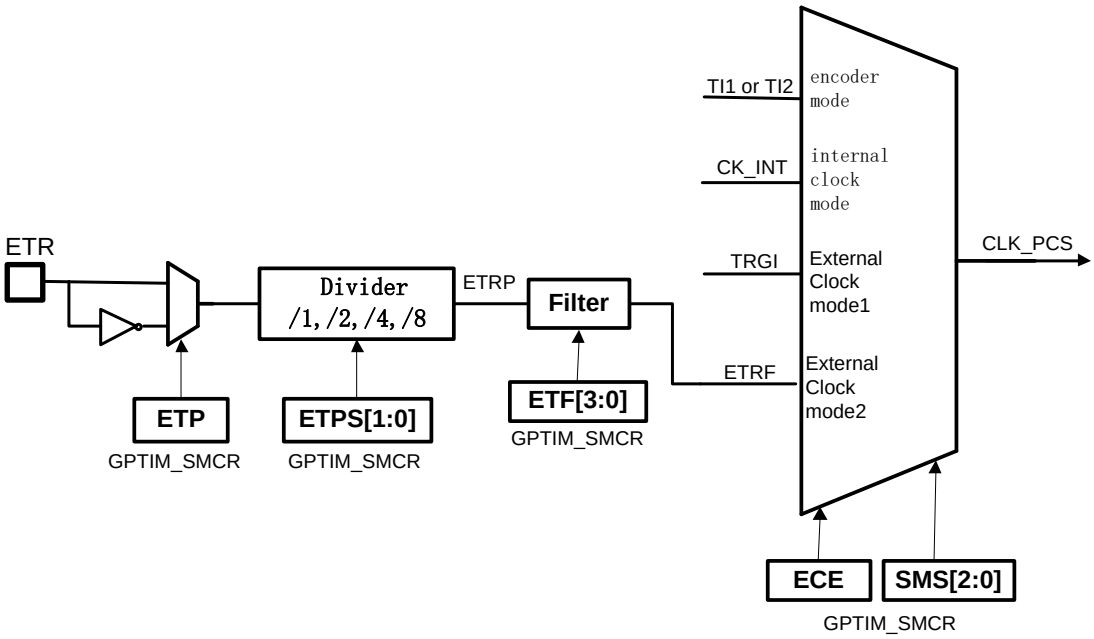


Figure 30-19 External Trigger Input Block

The figure below shows counting using the rising edge of the ETR divided by 2, where the actual counting occurs at a time delayed from the rising edge of the ETR input due to the synchronization process of the internal clock.

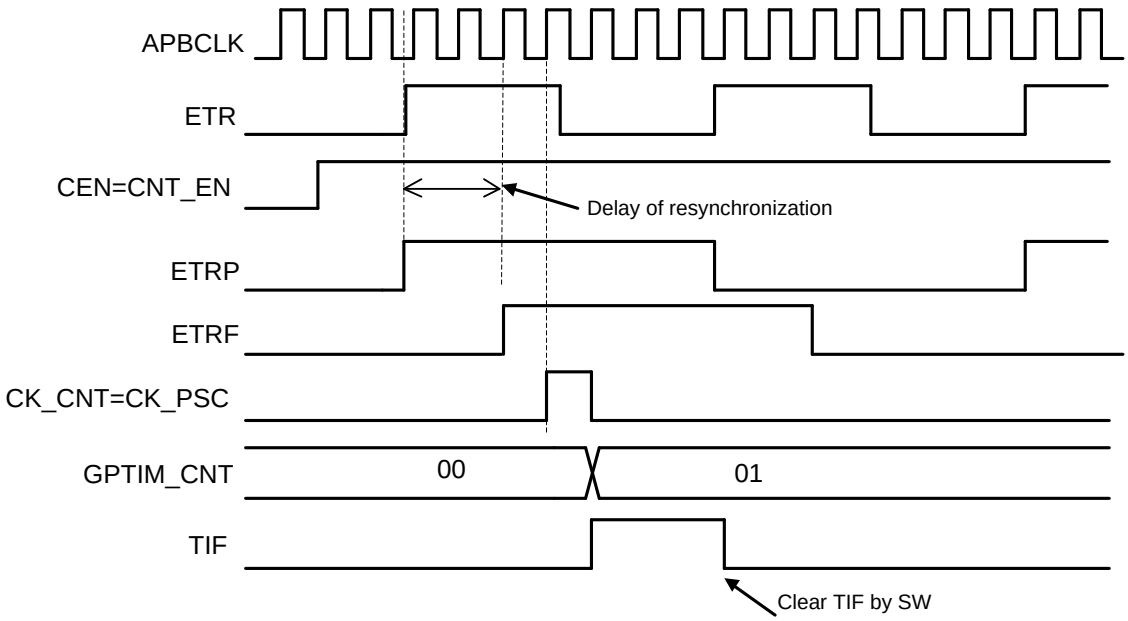


Figure 30-20 Control Circuit 1 in External Clock Mode 2

The main difference from external clock mode 1 is that the ETR input is directly divided and then filtered to generate the CK_PSC clock, which means that it can support application scenarios where the ETR input frequency is higher than APB_CLK, in which case the ETR input needs to be pre-divided first before it can be used to drive the counter.

The configuration required for this mode is as follows:

- In the GPIO module, configure the corresponding pin as GPTIM_ETR
- Set ETP for edge select, GPTIM_SMCR.ETP=0
- Set the ETR dividing ratio, configure GPTIM_SMCR.ETPS[1:0]=01
- Configure input filtering time, GPTIM_SMCR.ETF[3:0]=0000
- Set ECE register to enable external clock mode 2, GPTIM_SMCR.ECE=1, GPTIM_SMCR.SMS=000
- Enable counter, configure GPTIM_CR1.CEN=1

The following figure shows an example of a typical external clock mode 2:

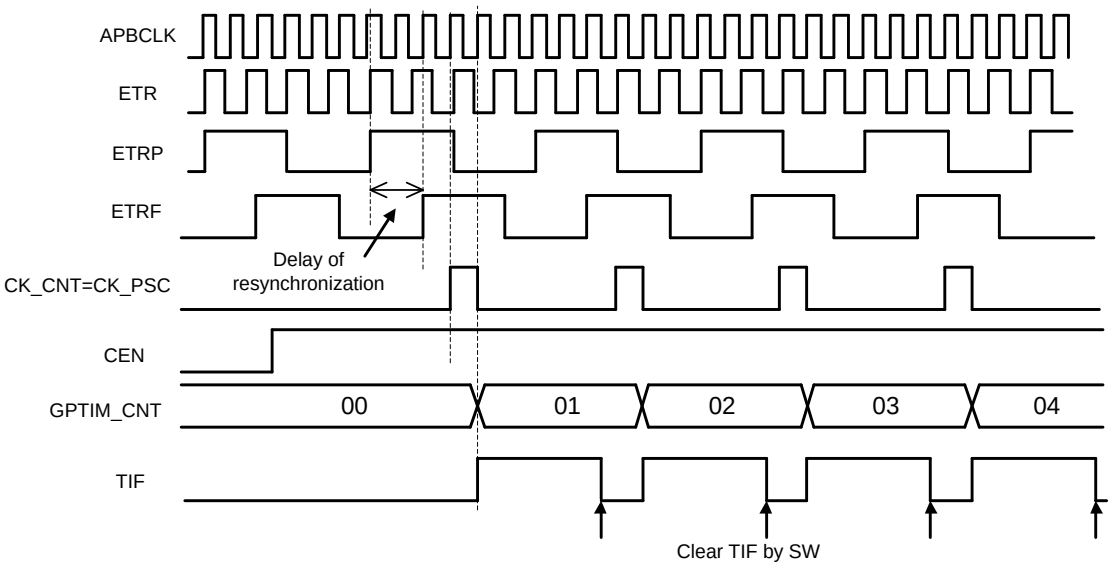
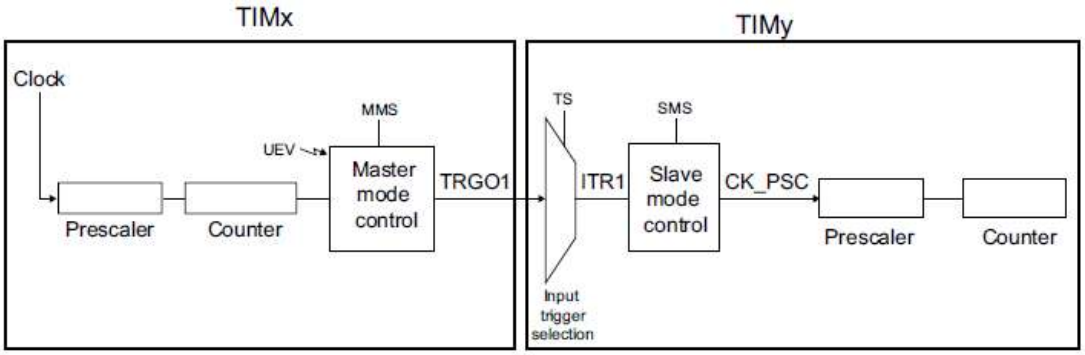


Figure 30-21Control Circuit 2 in External Clock Mode 2

When using external clock mode 2, it is still possible to configure GPTIM as slave mode: for example, you can use the ETR input to count, while using another Timer's TRGO as the trigger signal, and resetting the counter to start counting again when the trigger event arrives.

30.4.3.4 Internal Trigger Inputs Mode (ITR)

Each GPTIM supports 4 ITR inputs, which can be used for counting trigger or capturing internal signal. When ITR is selected as Count Trigger signal, GPTIM counter will count at the high level of each ITR signal. Cascading timer is possible through the internal trigger mode, an example shown in the following figure:



Configuring TIMx as master mode and periodically output TRGO pulse signal, and TIMy as Slave mode and setting TIMx's TRGO to ITR, TIMy counts once when TIMx.TRGO pulse arrives.

The timer cascaded based on the internal trigger mode has the following requirements:

- The TRGO signal is designed as an APBCLK single-cycle pulse
- TIMx and TIMy both operate in the APBCLK clock domain
- TRGO is a synchronous pulse for the receiver
- The operating clocks of both Master and Slave must be enabled

The trigger signal that can be used for internal trigger mode can be ADC_EOC or comparator output in addition to other timer outputs. In order to meet the above requirements, the trigger signal from ADC and COMP output needs to be processed into APBCLK synchronous pulse.

30.4.4 Capture of Internal Trigger Signal (ITRx)

Each GPTIM supports 4 ITR inputs, which can be used for count trigger or internal signal capture. When used for internal signal capture, TS needs to be configured as 000~011 for selecting ITR0~ITR3, and CCxS needs to be configured as 11, i.e. TRC is selected as the capture signal. By this method, Timer can capture the period or level width of various chip internal signals.

Each ITR input supports 4 internal signal extensions, which are configured by the ITRxSEL register. Refer to the following table for input signal sources:

GPTIM0			Function
ITR0SEL	00	ATIM_TRGO	Counting Trigger
	01	UART0_RX	Width capture
	10	UART1_RX	Width capture
	11	UART3_RX	Width capture
ITR1SEL	00	GPTIM2_TRGO	Counting Trigger
	01	XTHF	Cycle Capture
	10	RCHF	Cycle Capture
	11	LPUART1_RX	Cycle Capture
ITR2SEL	00	BSTIM32_TRGO	Counting Trigger
	01	LPUART2_RX	Width capture
	10	RCLP	Cycle Capture
	11	XTLF	Cycle Capture
ITR3SEL	00	COMP1_TRGO	Counting Trigger
	01	RCLF	Cycle Capture
	10	COMP2_TRGO	Counting Trigger
	11	LPT32_TRGO	Counting Trigger
GPTIM1			**Function**
ITR0SEL	00	ATIM_TRGO	Counting Trigger
	01	UART0_RX	Width capture
	10	UART1_RX	Width capture
	11	UART3_RX	Width capture
ITR1SEL	00	GPTIM0_TRGO	Counting Trigger
	01	LUT1_TRGO	Cycle Capture
	10	RCHF	Cycle Capture
	11	ADC_EOC_TRGO	Counting Trigger
ITR2SEL	00	BSTIM32_TRGO	Counting Trigger
	01	LSCLK	Cycle Capture
	10	RCLP	Cycle Capture
	11	XTLF	Cycle Capture
ITR3SEL	00	COMP1_TRGO	Counting Trigger
	01	LUT3_TRGO	Cycle Capture
	10	COMP2_TRGO	Counting Trigger
	11	LPT32_OUT	Counting Trigger
GPTIM2			**Function**

ITR0SEL	00	ATIM_TRGO	Counting Trigger
	01	UART3_RX	Width capture
	10	UART4_RX	Width capture
	11	LUT0_TRGO	Width capture
ITR1SEL	00	GPTIM1_TRGO	Counting Trigger
	01	XTHF	Cycle Capture
	10	RCHF	Cycle Capture
	11	ADC_EOC_TRGO	Counting Trigger
ITR2SEL	00	BSTIM16_TRGO	Counting Trigger
	01	LSCLK	Cycle Capture
	10	RCLP	Cycle Capture
	11	XTLF	Cycle Capture
ITR3SEL	00	COMP1_TRGO	Counting Trigger
	01	LUT2_TRGO	Cycle Capture
	10	COMP2_TRGO	Counting Trigger
	11	LPT16_TRGO	Counting Trigger

The TRGO signal, which is used as a count trigger, is processed as an enable signal which is one APBCLK widthuntil it reaches the ITRx of the GPTIM. The signals used as signal of period or width capture don't need to be processed and are captured directly by GPTIM.

The software should ensure that the correct signal is selected for the correct function, a wrong configuration will lead to completely wrong results. For example, if ATIM_TRGO is used for width capture, the result will be meaningless.

30.4.5 Capture/Compare Channels

Each GPTIM contains 4 capture/compare channels,each consisting of a capture compare register (CCR) (including shadow registers), a capture input stage, and a compare output stage.

The input stage circuitry samples the Tix input to generatea filtered signal TixF.Then, an edge detector with polarity selection generates a signal (TIxFPx) which can beused as trigger input by the slave mode controller or as the capture command. It isprescaled before the capture register

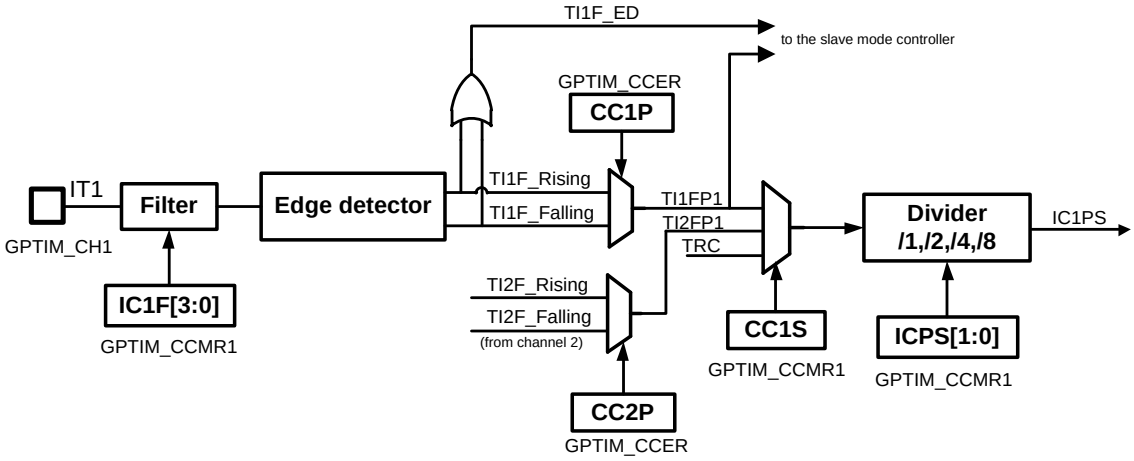


Figure 30-22 Capture/Compare Channel (Example: Channel 1 Input Stage)

The output stage circuit will generate an output reference signal OCxREF, which is fixed to be active high as the reference input of the final output circuit. Complementary outputs are not supported for GPTIM output channels.

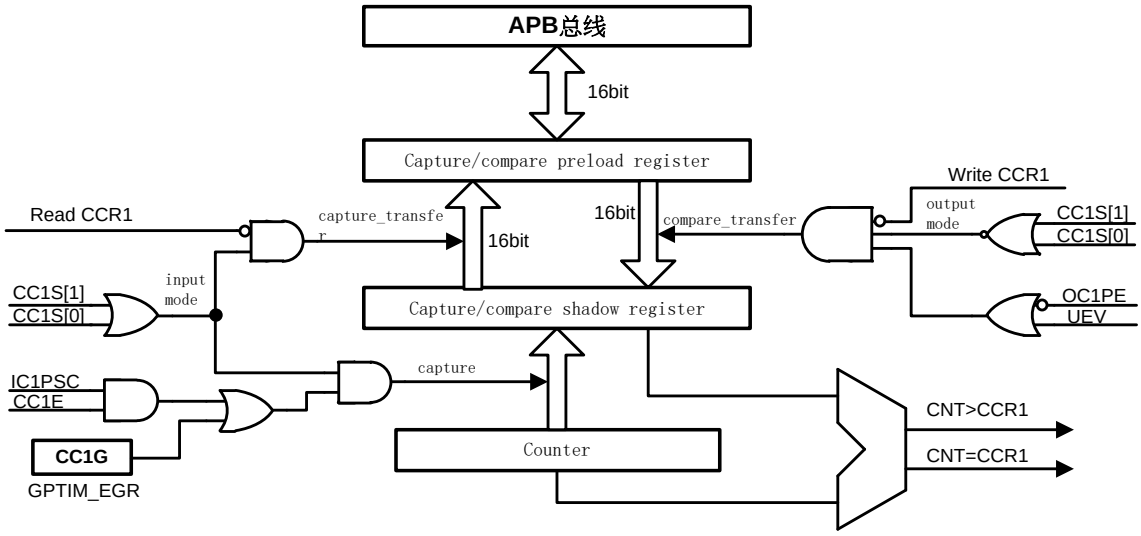


Figure 30-23 Capture/Compare Channel 1 Main Circuit

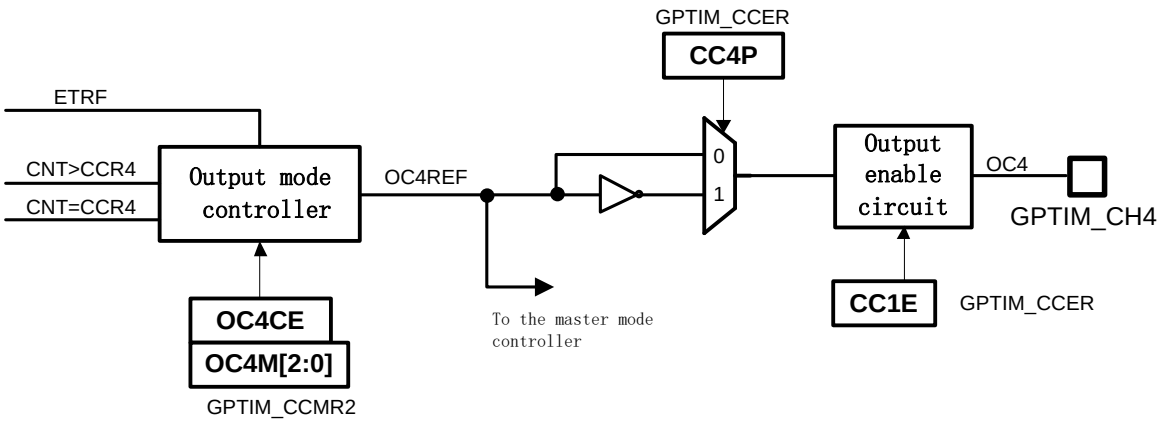


Figure 30-24 Output Stage of Capture/Compare Channel

The capture/compare register (CCR) contains the preload register and the shadow register. Software reads and writes always access the preload register. In capture mode, the captured value is saved in the shadow register and copied to the preload register. In compare mode, the value of the preload register is copied to the shadow register for comparison with the counter.

30.4.6 Input Capture Mode

In Input capture mode, the Capture/Compare Registers (CCR) are used to latch the value of the counter after a transition detected by the corresponding ICx signal. When a capture occurs, the corresponding CCXIF flag is set and an interrupt or a DMA request can be sent if they are enabled. If a capture occurs while the CCxIF flag was already high, then the over-capture flag CCxOFis set. CCxIF can be cleared by software by writing it to 1 or by reading the captured data stored in the CCRx register. CCxOF is cleared when you write it to 1.

The input capture of PWM signals can be achieved by matching two or more channels. For example, to calculate the period and duty cycle of an input signal, you can set TI1 pin as input signal, and the chip internally takes the filtered signal to get TI1FP1 by taking the rising edge and TI1FP2 by taking the filtered signal to get the falling edge, and input TI1FP1 to capture channel 1 and TI1FP2 to capture channel 2, so that channel 1 can capture the rising edge of the input signal, while after the capture interrupt occurs periodically, the software can calculate the period and duty cycle of the input signal according to the values of CCR1 and CCR2 registers.

To implement capturing the counter value to GPTIM_CCR1 register at the rising edge of TI1 input, the steps of configuration as follows:

- In the GPIO module, configure the corresponding pin for GPTIM_CH1 function
- Turn off the channel enable register and configure GPTIM_CCER.CC1E=0 to ensure that the channel configuration is successful afterwards
- Select the input channel, configure GPTIM_CCMR1.CC1S=01, IC1 is mapped to TI1

- Select count valid edge, configure GPTIM_CCER.CC1P, select pos edge or neg edge
- Configure input filtering time, configure GPTIM_CCMR1.IC1F[3:0]
- Configure input prescaler, configure GPTIM_CCMR1.IC1PS[1:0]
- Turn on channel enable, configure GPTIM_CCER.CC1E=1

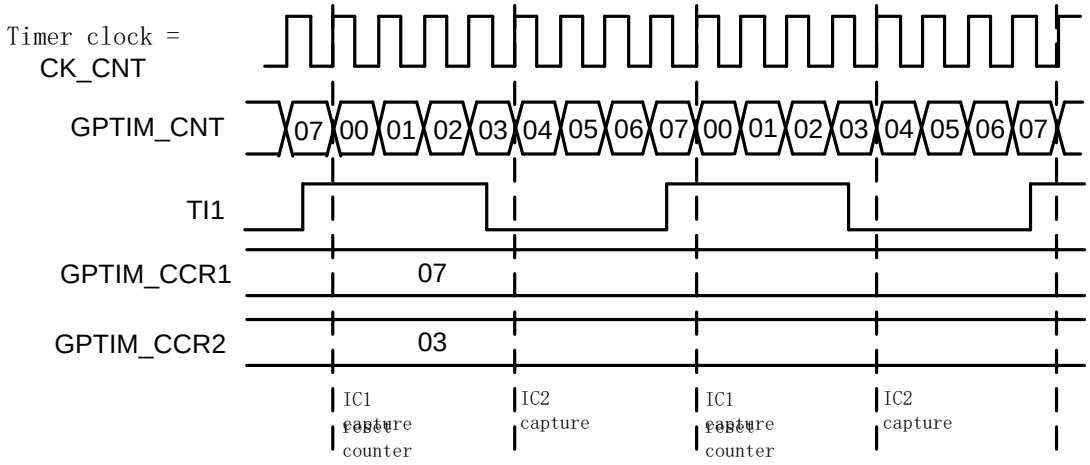


Figure 30-25 PWM Input Capture Mode Timing

If you want to implement the PWM input capture function, you need to make the following settings:

- In the GPIO module, configure the corresponding pin as GPTIM_CH1 function
- Turn off the channel enable, configure GPTIM_CCER.CC1E=0, GPTIM_CCER.CC2E=0 to ensure the channel configuration is successful afterwards
- Select input channel, two channels IC1,IC2 are mapped to the same TI1 input port, configure GPTIM_CCMR1.CC1S=01, GPTIM_CCMR1.CC2S=10
- Select count active edge, two channels IC1,IC2 active edge polarity opposite, configurationGPTIM_CCER.CC1P=0, GPTIM_CCER.CC2P=1
- Configure input filtering time, configure GPTIM_CCMR1.IC1F[3:0], GPTIM_CCMR1.IC2F[3:0]
- Configure input prescaler, configure GPTIM_CCMR1.IC1PS[1:0], GPTIM_CCMR1.IC2PS[1:0]
- Select trigger input signal, configure GPTIM_SMCR.TS[2:0]=101
- Set the slave mode controller to reset mode, configure GPTIM_SMCR.SMS[2:0]=100
- Turn on channel enable, configure GPTIM_CCER.CC1E=1, GPTIM_CCER.CC2E=1

30.4.7 Forced Output Mode

In compare output mode, software can directly force OCxREF to a specific level independent of the CCR and counter comparison results.

Software can directly force OCxREF to valid (OCxREF is fixed to high valid) by setting OcxM=101, and OCxREF can be directly forced to invalid (low) by setting OcxM=100. However, the software force operation will not cancel the comparison process, and the comparison between CCR and counter will continue.

30.4.8 Output Compare Mode

In output compare mode, when CCR is equal to the counter value, OCxREF can be set to valid, invalid, or level flipped. At the same time, the interrupt flag is set and the DMA request can be sent.

Output compare can also be used to output a pulse signal of a specific width (single output).

Steps:

1, Select the counting clock (internal, external, pre-divided, etc.)

2, Write desired data to ARR and CCR registers

3, Set interrupt enable and DMA enable as needed

4, Select output mode

5, Enable the counter

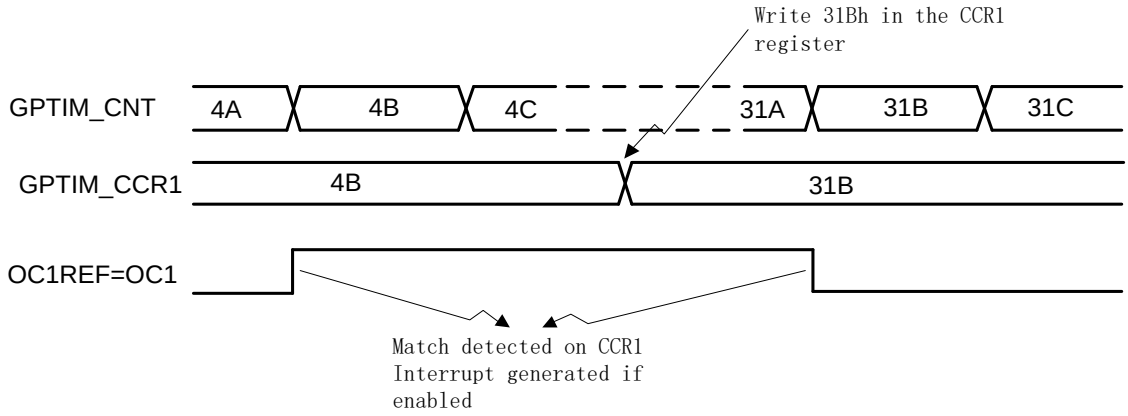


Figure 30-26 Output Compare Mode, Toggle on OC1

Without enabling preload, the software can rewrite the CCR register at any time to achieve real-time control of the output waveform. If preload is enabled, the CCR shadow register will only be updated to the contents of the preload register when the next update event occurs.

30.4.9 PWM Mode

The PWM mode can output a pulse width modulated signal, whose period is determined by the ARR register and duty cycle is determined by the CCR register.

The polarity of the output signal can be configured by the CCxP register, and the CNT and CCR are compared in real time during PWM mode operation. Since the counter supports edge-aligned and center-aligned counting modes, the PWM output also supports edge-aligned and center-aligned modes.

PWM edge-aligned mode

In the case of counting up, when configured to PWM mode 1, the OCxREF signal is high when CNT < CCR, otherwise it is low. If the CCR value is greater than the ARR value, OCxREF is fixed to 1; if CCR is 0 then OCxREF is fixed to 0.

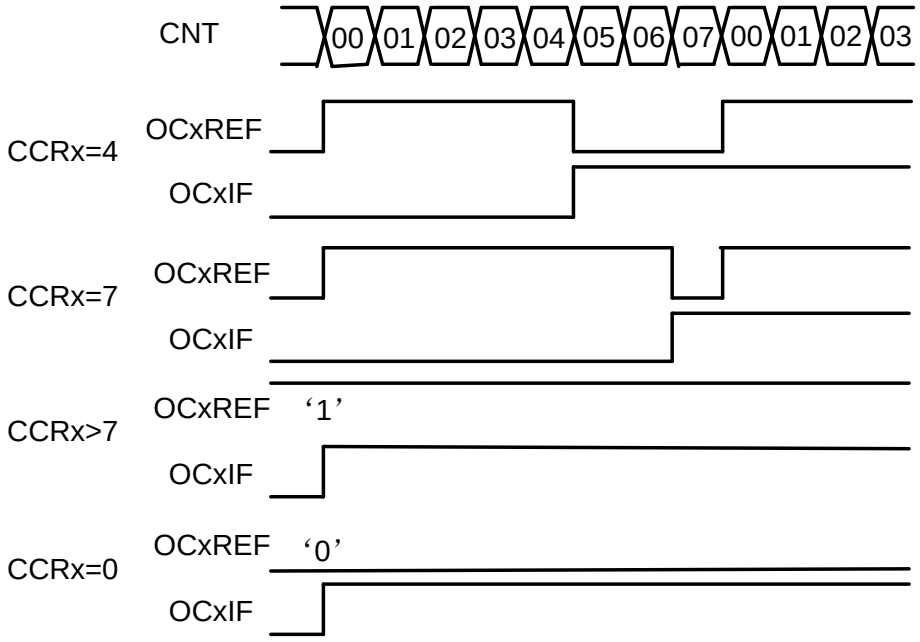


Figure 30-27 Edge-Aligned PWM Waveforms (ARR=7)

PWM center-aligned mode

The OCxREF level definition is the same as the edge alignment mode. The following figure shows an example.

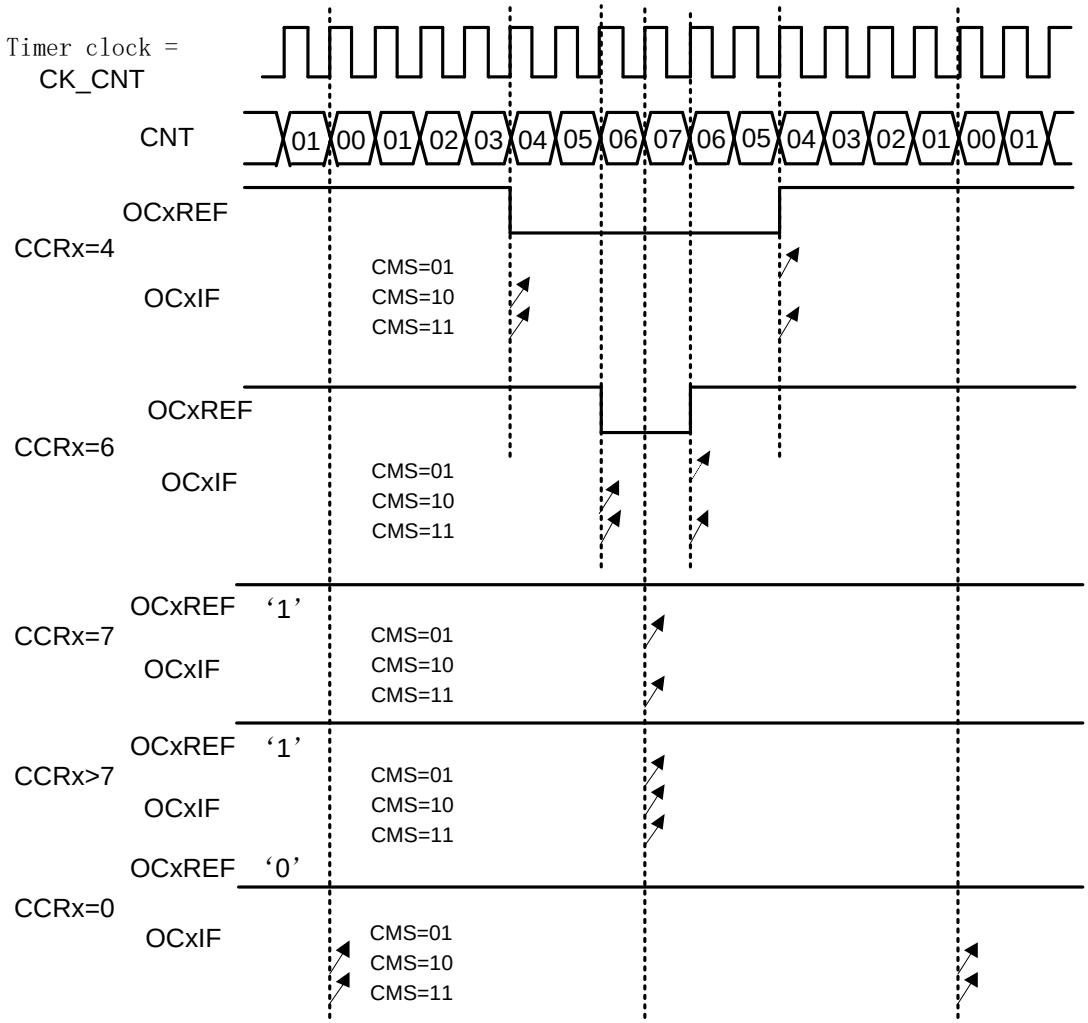


Figure 30-28 Center-Aligned PWM Waveforms (ARR=7)

When the central alignment counter is started, the counting direction is determined by the DIR register at the beginning; subsequently, the state of the DIR register is controlled directly by the hardware during the counting process. For safety, it is recommended that the user program do an update through the UG register before starting the counter, and do not rewrite the counter during the counting process.

30.4.10 One-Pulse Mode

One-pulse mode (OPM) is a particular case of the previous modes. It allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay.

Unlike other output modes, the counter is automatically stopped when the next update event arrives. Only when the CCR and the initial value of the counter are different, the pulse can be output correctly. When counting up, CNT<CCR<=ARR is required, and when counting down, CNT>CCR is required.

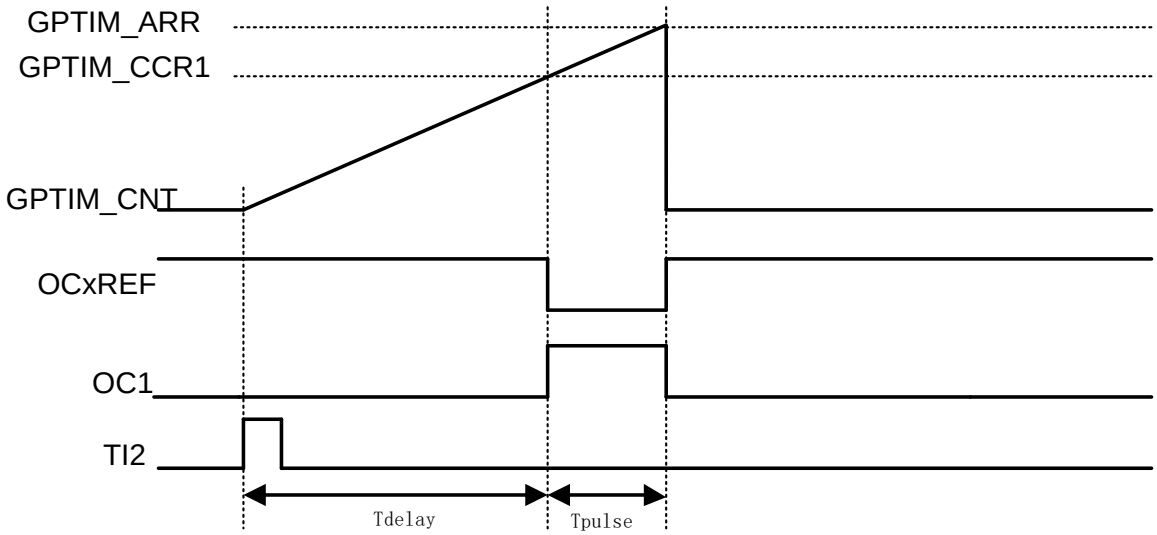


Figure 30-29 Example of One-Pulse Mode

The above diagram is to use TI2 input as the counter trigger signal, the count value is equal to CCR after OCxREF output low, count to ARR after OCxREF back to high, and the counter rolls back to 0, stop counting.

The configuration to achieve the above function TI2 as input trigger is as follows:

- In the GPIO module, configure the corresponding pin as GPTIM_CH2 function
- Turn off the channel enable, configure GPTIM_CCER.CC2E=0, make sure the channel configuration is successful afterwards
- Select input channel, configure GPTIM_CCMR1.CC2S=01
- Select count valid edge, configure GPTIM_CCER.CC2P=0
- Select the trigger input signal, configure GPTIM_SMCR.TS[2:0]=110, TI2FP2 as TRGI
- Set the slave mode controller to trigger mode, configure GPTIM_SMCR.SMS[2:0]=110, TI2FP2 is used to start the counter
- Turn on channel enable, configure GPTIM_CCER.CC2E=1

The configuration to achieve the above function OC1 as an output is as follows:

- In the GPIO module, configure the corresponding pin for GPTIM_CH1 function

- Close the channel enable, configure GPTIM_CCER.CC1E=0, ensure the channel configuration is successful afterwards
- Output channel, configure GPTIM_CCMR1.CC1S=00
- OC1M=111,PWM mode 2
- Turn on the channel enable, configure GPTIM_CCER.CC1E=1

Special settings for the OPM waveform generation time base.

- The value of GPTIM_CCR1 determines Tdelay
- The difference between GPTIM_ARR and GPTIM_CCR1 determines Tpulse (GPTIM_ARR - GPTIM_CCR1)
- Set to single pulse mode, configure GPTIM_CR1.OPM=1

30.4.11 External Event Clears OCxREF

OCxREF is active high, by applying high level to external ETR pin, OCxREF can be pulled low directly until the next update event. This function is only valid in output compare and PWM mode, not in software force mode. Enabling this function requires setting OcxCE to 1.

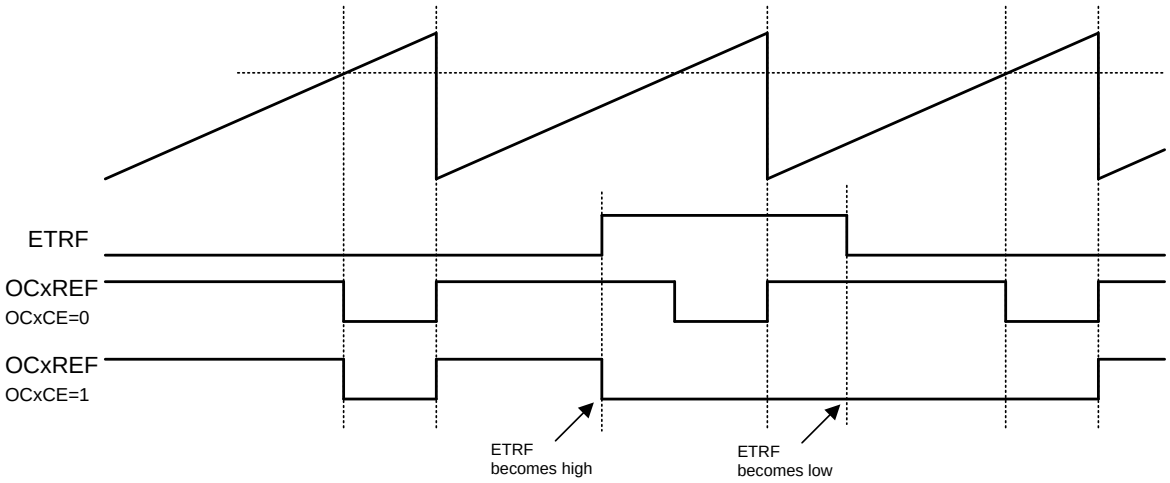


Figure 30-30ETR Signal Clears OCxREF of GPTIM

30.4.12 Encoder Interface Mode

The encoder interface mode involves two external input signals, and GPTIM determines whether to increment or decrement the count value based on the edge level of one signal relative to the level of the other. The following table shows the relationship between the counting mode and the two input signals:

Active edge	Level on opposite signal (TI1 for TI2, TI2 for TI1)	TI1 signal		TI2 signal	
		Rising	Falling	Rising	Falling
Count only at TI1	High	Down	Up	No Count	No Count
	Low	Up	Down	No Count	No Count
Count only at TI2	High	No Count	No Count	Up	Down
	Low	No Count	No Count	Down	Up
Counts at both TI1 and TI2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

Table 30-1 Encoder Interface Counting Method

For example, when the counter is clocked with the TI1 signal, the counter decreases if the rising edge of TI1 is sampled high to TI2, and increments if the falling edge of TI1 is sampled high to TI2.

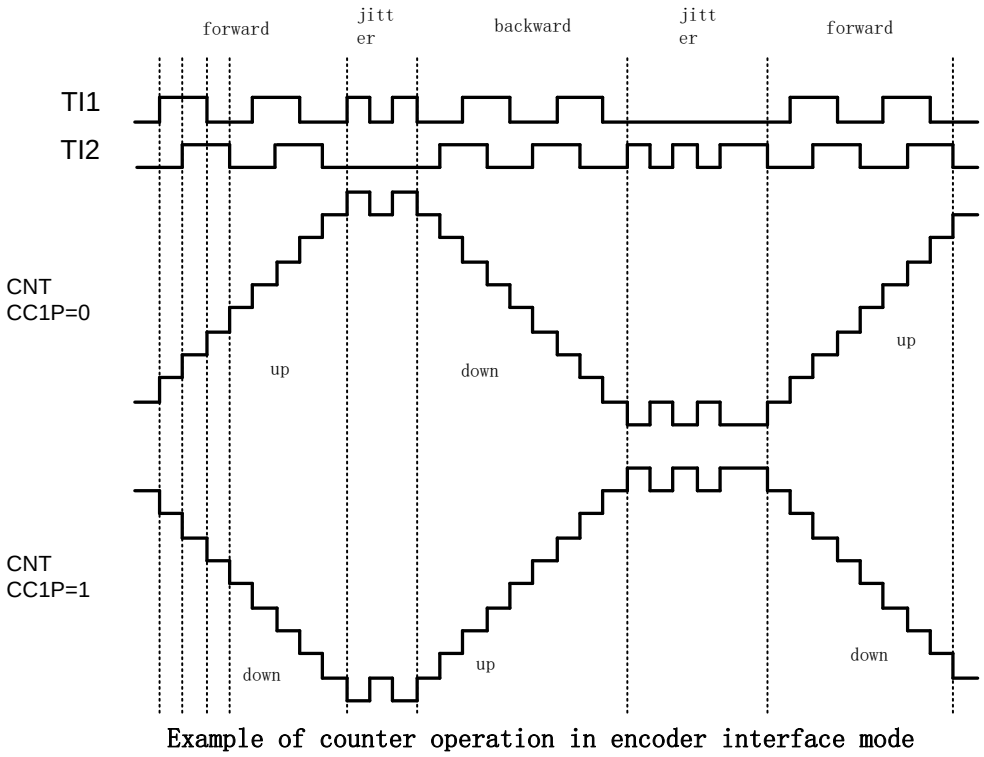


Figure 30-31 Example of Counter Operation in Encoder Interface Mode

The following settings are required for the coding mode input channels:

- In the GPIO module, configure the corresponding pins as GPTIM_CH1, GPTIM_CH2 function
- Turn off the channel enable, configure GPTIM_CCER.CC1E=0, GPTIM_CCER.CC2E=0, make sure the channel configuration is successful afterwards.

- Select the input channel, configure GPTIM_CCMR1.CC1S=01, GPTIM_CCMR1.CC2S=01
- Select count valid edge, configure GPTIM_CCER.CC1P=0, GPTIM_CCER.CC2P=0
- Set the slave mode controller to encode mode 3, configure GPTIM_SMCR.SMS[2:0]=011
- Turn on channel enable, configure GPTIM_CCER.CC1E=1, GPTIM_CCER.CC2E=1

30.4.13 GPTIM Slave Mode

When GPTIM is used as slave (triggered by external events), it can be configured to work in three modes: reset mode, gated mode, and trigger mode.

Reset mode

In this mode, the event of external input will cause all TIM internal preload registers to reinitialize and CNT to return to 0 to start counting. As an example in the figure below, the counter counts normally and the rising edge of external TI1 input triggers the counter to clear to zero and start counting again.

The configuration in the following example is as follows:

- In the GPIO module, configure the corresponding pin as GPTIM_CH1 function
- Turn off the channel enable, configure GPTIM_CCER.CC1E=0 to ensure the channel configuration is successful afterwards
- Select input channel, configure GPTIM_CCMR1.CC1S=01
- Select the count valid edge, configure GPTIM_CCER.CC1P=0
- Select the trigger input signal, configure GPTIM_SMCR.TS[2:0]=101, TI1FP1 as TRGI
- Set the slave mode controller to reset mode, configure GPTIM_SMCR.SMS[2:0]=100
- Turn on channel enable, configure GPTIM_CCER.CC1E=1
- Enable counter, configure GPTIM_CR1.CEN=1

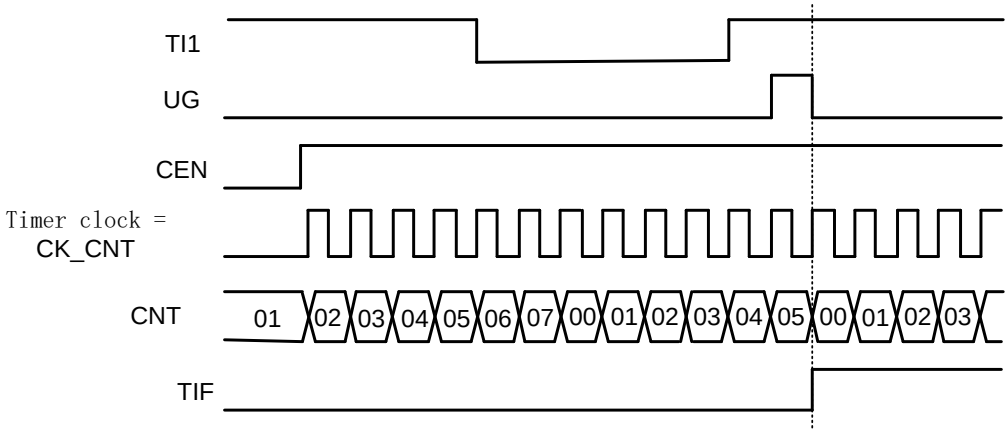


Figure 30-32 Control Circuit in Reset Mode

Gated mode

In this mode, the counter operates only when the input signal is a specific level. The interrupt flag is triggered when the level shift causes the counter to start or stop counting.

The configuration in the following example is as follows:

- In the GPIO module, configure the corresponding pin as GPTIM_CH1 function
- Turn off the channel enable, configure GPTIM_CCER.CC1E=0 to ensure the channel configuration is successful afterwards
- Select input channel, configure GPTIM_CCMR1.CC1S=01
- Select the count valid edge, configure GPTIM_CCER.CC1P=0
- Select the trigger input signal, configure GPTIM_SMCR.TS[2:0]=101, TI1FP1 as TRGI
- Set the slave mode controller to gated mode, configure GPTIM_SMCR.SMS[2:0]=101
- Turn on channel enable, configure GPTIM_CCER.CC1E=1
- Enable counter, configure GPTIM_CR1.CEN=1

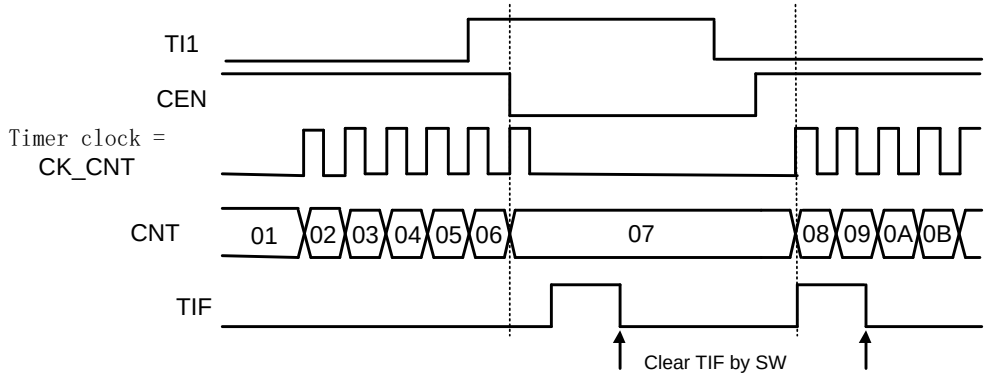


Figure 30-33 Control Circuit in Gated Mode

Trigger mode

The counter starts counting only after the arrival of an event from an external input.

The configuration in the following example is as follows:

- In the GPIO module, configure the corresponding pin as GPTIM_CH1 function
- Turn off the channel enable, configure GPTIM_CCER.CC1E=0 to ensure the channel configuration is successful afterwards
- Select input channel, configure GPTIM_CCMR1.CC1S=01
- Select the count valid edge, configure GPTIM_CCER.CC1P=0

- Select the trigger input signal, configure GPTIM_SMCR.TS[2:0]=101, TI1FP1 as TRGI
- Set slave mode controller to trigger mode, configure GPTIM_SMCR.SMS[2:0]=110
- Turn on channel enable, configure GPTIM_CCER.CC1E=1

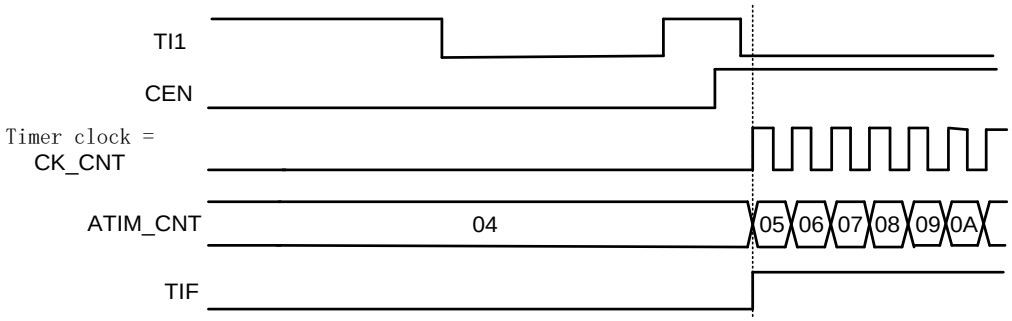


Figure 30-34 Control Circuit in Trigger Mode

External Clock mode 2 + trigger mode

ETR can be set as the count clock while using another external input as the counter start trigger signal. For example, after the rising edge of TI1 is detected, the counter starts counting with the rising edge of the ETR input.

The configuration in the following example is as follows:

- In the GPIO module, configure the corresponding pin as GPTIM_CH1, ATIM_ETR function
- Set ETP for edge selection, GPTIM_SMCR.ETP=0
- Set ETR dividing ratio, configure GPTIM_SMCR.ETPS[1:0]=01
- Configure input filtering time, GPTIM_SMCR.ETF[3:0]=0000
- Set ECE register to enable external clock mode 2, GPTIM_SMCR.ECE=1
- Turn off the channel enable, configure GPTIM_CCER.CC1E=0 to ensure the channel configuration is successful afterwards
- Select the input channel, configure GPTIM_CCMR1.CC1S=01
- Select the count valid edge, configure GPTIM_CCER.CC1P=0
- Select trigger input signal, configure GPTIM_SMCR.TS[2:0]=101, TI1FP1 as TRGI
- Set slave mode controller to trigger mode, configure GPTIM_SMCR.SMS[2:0]=110
- Turn on channel enable, configure GPTIM_CCER.CC1E=1

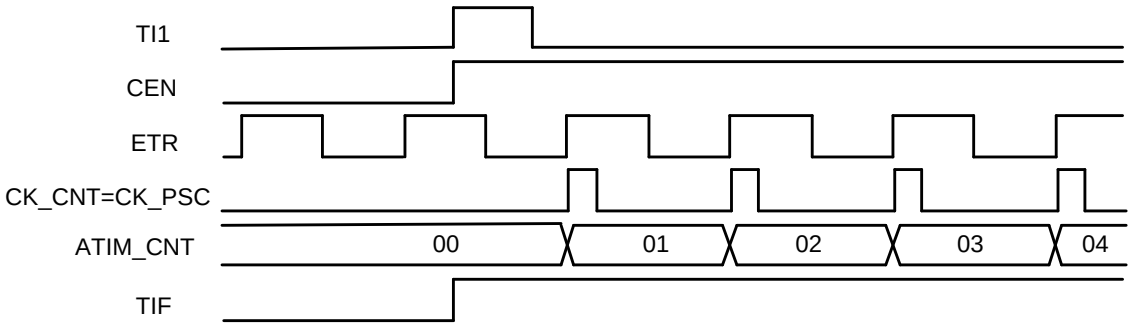


Figure 30-35 Control Circuit in External Clock Mode 2 + Trigger Mode

30.4.14 DMA Access

GPTIM supports 6 types of DMA requests, which are 4 CC channel requests, external trigger requests and user software trigger requests.

Each CC channel generates one DMA request, which is used to transfer the contents of CCRx to RAM in capture mode or to write the data in RAM to CCRx in compare mode; the DMA requests of CC channels can be configured as single transfer or Burst transfer (CCxBURSTEN), single transfer only accesses CCRx registers, Burst transfer and Burst transfer accesses a specific set of registers according to the DCR register configuration.

In addition, external trigger events and software trigger events can also generate DMA requests, and when both of these requests occur, a DMA Burst transfer is initiated to write data to one or more registers inside the GPTIM or to read one or more register values from the GPTIM.

DMA requests	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA Access Objects	Length of one transmission
GTIMx_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	

DMA requests	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA Access Objects	Length of one transmission
GTIMx_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
GTIMx_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	

30.4.15 DMA Burst

DMA-Burst supports a single event triggering multiple consecutive DMA requests. The main function is to continuously update the contents of multiple registers after the event occurs, thus enabling functions such as dynamic real-time adjustment of output waveforms.

The DMA controller needs to point the peripheral target address to a virtual register GPTIM_DMAR. On the occurrence of a specific timer event, GPTIM will continuously send multiple DMA requests. Each DMA write operation to GPTIM_DMAR is redirected to the actual function register by GPTIM.

The DBL register is used to set the DMA burst length, and the DBA register is used to set the base address (relative to the offset of GPTIM_CR) for DMA access to the GPTIM internals.

30.4.16 Input Isochronous Function

The input signals of channels 1 to 3 can be heterodyned up and then connected to the filter and edge circuit inputs of channel 1 for input capture or trigger of channel 1.

The TI1S bit of the GPTIM_CR2 register is used to select whether the input of channel 1 comes from the heterodyning of the three channel inputs.

30.4.17 Debug Mode

When Cortex-M0 enters debug mode, the timer can stop or continue its operation, and its behavior is defined by the DBG_TIMx_STOP register of the DCU module.

30.5 Register

Offset	Name	Symbol
GPTIM0(Base address: 0x40014C00)		
0x00000000	GPTIM0 Control Register1	GPTIM0_CR1
0x00000004	GPTIM0 Control Register2	GPTIM0_CR2
0x00000008	GPTIM0 Slave Mode Control Register	GPTIM0_SMCR
0x0000000C	GPTIM0 DMA and Interrupt Enable Register	GPTIM0_DIER
0x00000010	GPTIM0 Interrupt Status Register	GPTIM0_ISR
0x00000014	GPTIM0 Event Generation Register	GPTIM0_EGR
0x00000018	GPTIM0 Capture/Compare Mode Register1	GPTIM0_CCMR1
0x0000001C	GPTIM0 Capture/Compare Mode Register2	GPTIM0_CCMR2
0x00000020	GPTIM0 Capture/Compare Enable Register	GPTIM0_CCER
0x00000024	GPTIM0 Counter Register	GPTIM0_CNT
0x00000028	GPTIM0 Prescaler Register	GPTIM0_PSC
0x0000002C	GPTIM0 Auto-Reload Register	GPTIM0_ARR
0x00000034	GPTIM0 Capture/Compare Register1	GPTIM0_CCR1
0x00000038	GPTIM0 Capture/Compare Register2	GPTIM0_CCR2
0x0000003C	GPTIM0 Capture/Compare Register3	GPTIM0_CCR3
0x00000040	GPTIM0 Capture/Compare Register4	GPTIM0_CCR4
0x00000048	GPTIM0 DMA Control Register	GPTIM0_DCR
0x0000004C	GPTIM0 DMA Access Register	GPTIM0_DMAR
0x00000060	GPTIM0 Internal Trigger Select Register	GPTIM0_ITRSEL
GPTIM1(Base address: 0x40016400)		
0x00000000	GPTIM1 Control Register1	GPTIM1_CR1
0x00000004	GPTIM1 Control Register2	GPTIM1_CR2
0x00000008	GPTIM1 Slave Mode Control Register	GPTIM1_SMCR
0x0000000C	GPTIM1 DMA and Interrupt Enable Register	GPTIM1_DIER
0x00000010	GPTIM1 Interrupt Status Register	GPTIM1_ISR
0x00000014	GPTIM1 Event Generation Register	GPTIM1_EGR
0x00000018	GPTIM1 Capture/Compare Mode Register1	GPTIM1_CCMR1
0x0000001C	GPTIM1 Capture/Compare Mode Register2	GPTIM1_CCMR2
0x00000020	GPTIM1 Capture/Compare Enable Register	GPTIM1_CCER
0x00000024	GPTIM1 Counter Register	GPTIM1_CNT
0x00000028	GPTIM1 Prescaler Register	GPTIM1_PSC
0x0000002C	GPTIM1 Auto-Reload Register	GPTIM1_ARR
0x00000034	GPTIM1 Capture/Compare Register1	GPTIM1_CCR1
0x00000038	GPTIM1 Capture/Compare Register2	GPTIM1_CCR2
0x0000003C	GPTIM1 Capture/Compare Register3	GPTIM1_CCR3
0x00000040	GPTIM1 Capture/Compare Register4	GPTIM1_CCR4
0x00000048	GPTIM1 DMA Control Register	GPTIM1_DCR
0x0000004C	GPTIM1 DMA Access Register	GPTIM1_DMAR
0x00000060	GPTIM1 Internal Trigger Select Register	GPTIM1_ITRSEL

Offset	Name	Symbol
GPTIM2(Base address: 0x40018000)		
0x00000000	GPTIM2 Control Register1	GPTIM2_CR1
0x00000004	GPTIM2 Control Register2	GPTIM2_CR2
0x00000008	GPTIM2 Slave Mode Control Register	GPTIM2_SMCR
0x0000000C	GPTIM2 DMA and Interrupt Enable Register	GPTIM2_DIER
0x00000010	GPTIM2 Interrupt Status Register	GPTIM2_ISR
0x00000014	GPTIM2 Event Generation Register	GPTIM2_EGR
0x00000018	GPTIM2 Capture/Compare Mode Register1	GPTIM2_CCMR1
0x0000001C	GPTIM2 Capture/Compare Mode Register2	GPTIM2_CCMR2
0x00000020	GPTIM2 Capture/Compare Enable Register	GPTIM2_CCER
0x00000024	GPTIM2 Counter Register	GPTIM2_CNT
0x00000028	GPTIM2 Prescaler Register	GPTIM2_PSC
0x0000002C	GPTIM2 Auto-Reload Register	GPTIM2_ARR
0x00000034	GPTIM2 Capture/Compare Register1	GPTIM2_CCR1
0x00000038	GPTIM2 Capture/Compare Register2	GPTIM2_CCR2
0x0000003C	GPTIM2 Capture/Compare Register3	GPTIM2_CCR3
0x00000040	GPTIM2 Capture/Compare Register4	GPTIM2_CCR4
0x00000048	GPTIM2 DMA Control Register	GPTIM2_DCR
0x0000004C	GPTIM2 DMA Access Register	GPTIM2_DMAR
0x00000060	GPTIM2 Internal Trigger Select Register	GPTIM2_ITRSEL

30.5.1 GPTIMx Control Register1 (GPTIMx_CR1)

NAME	GPTIMx_CR1(x=0,1,2)							
Offset	0x00000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						CKD	
access	U-0						R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ARPE	CMS		DIR	OPM	URS	UDIS	CEN
access	R/W-0	R/W-00		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:10	-	RFU: **Reserved, read as 0**
9:8	CKD	**CKD**: Clock division This bit-field indicates the division ratio between the timer clock

bit	name	functional description
		(CK_INT) frequency and sampling clock used by the digital filters (ETR, TIx) 00: tDTS = tCK_INT 01: tDTS = 2 × tCK_INT 10: tDTS = 4 × tCK_INT 11: Reserved
7	ARPE	**ARPE**: Auto-reload preload enable 0: TIMx_ARR register is not buffered 1: TIMx_ARR register is buffered
6:5	CMS	**CMS**: Center-aligned mode selection 00: Edge-aligned mode. The counter counts up or down depending on the direction bit(DIR). 01: Center-aligned mode 1. The counter counts up and down alternatively. 10: Center-aligned mode 2. The counter counts up and down alternatively. 11: Center-aligned mode 3. The counter counts up and down alternatively.
4	DIR	**DIR**: Direction 0: Counter used as upcounter 1: Counter used as downcounter ***Note:*** *This bit is read only when the timer is configured in Center-aligned mode or Encoder mode.*
3	OPM	**OPM**: One-pulse mode 0: Counter is not stopped at update event 1: Counter stops counting at the next update event (clearing the bit CEN)
2	URS	**URS**: Update request source 0: Any of the following events generate an update interrupt or DMA request if enabled. These events can be: – Counter overflow/underflow – Setting the UG bit – Update generation through the slave mode controller 1: Only counter overflow/underflow generates an update interrupt or DMA request if enabled.
1	UDIS	**UDIS**: Update disable 0: UEV enabled. The Update (UEV) event is generated by one of the following events: – Counter overflow/underflow – Setting the UG bit – Update generation through the slave mode controller

bit	name	functional description
		1: UEV disabled. The Update event is not generated, shadow registers keep their value(ARR, PSC, CCRx). However the counter and the prescaler are reinitiali
0	CEN	**CEN**: Counter enable 0: Counter disabled 1: Counter enabled **Note:** External trigger mode can automatically set the CEN

30.5.2 GPTIMx Control Register2 (GPTIMx_CR2)

NAME	GPTIMx_CR2(x=0,1,2)							
Offset	0x00000004							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TI1S	MMS			CCDS	-		
access	R/W-0	R/W-000			R/W-0	U-0		

bit	name	functional description
31:8	-	RFU: **Reserved, read as 0**
7	TI1S	**TI1S**: TI1 selection 0: The TIMx_CH1 pin is connected to TI1 input 1: The TIMx_CH1, CH2 and CH3 pins are connected to the TI1 input
6:4	MMS	Master Mode Selection, used to configure the TRGO source sent to the slave in master mode 000: UG register of GPTIM_EGR is used as TRGO 001: The counter enable signal CNT_EN is used as TRGO, which can be used to start multiple timers at the same time 010: UE (update event) signal is used as TRGO 011: Compare pulse, if the CC1IF flag is about to be set, TRGO outputs a positive pulse 100: OC1REF is used as TRGO 101: OC2REF as TRGO 110: OC3REF is used as TRGO

bit	name	functional description
		111: OC4REF used as TRGO ***Note:*** *The slave timer or ADC must enable the working clock in advance to receive the TRGO sent by the master timer*
3	CCDS	Capture/compare DMA selection 0: DMA request sent when CCx event occurs 1: DMA requests sent when update event occurs
2:0	-	RFU: **Reserved, read as 0**

30.5.3 GPTIMx Slave Mode Control Register (GPTIMx_SMCR)

NAME	GPTIMx_SMCR(x=0,1,2)							
Offset	0x00000008							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ETP	ECE	ETPS		ETF			
access	R/W-0	R/W-0	R/W-00		R/W-0000			
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	MSM	TS			-	SMS		
access	R/W-0	R/W-000			U-0	R/W-000		

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15	**ETP**	External trigger polarity 0: ETR is non-inverted, active at high level or rising edge 1: ETR is inverted, active at low level or falling edge
14	**ECE**	External clock enable 0: External clock mode 2 disabled 1: External clock mode 2 enabled. The counter is clocked by any active edge on the ETRF signal.
13:12	**ETPS**	External trigger prescaler External trigger signal ETRP frequency must be at most 1/4 of CK_INT frequency. A prescaler can be enabled to reduce ETRP frequency. It is useful when inputting fast external clocks. 00: Prescaler OFF 01: ETRP frequency divided by 2 10: ETRP frequency divided by 4

bit	name	functional description
		11: ETRP frequency divided by 8
11:8	**ETF**	External trigger filter 0000: No filter 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS/2}$, N=6 0101: $f_{SAMPLING}=f_{DTS/2}$, N=8 0110: $f_{SAMPLING}=f_{DTS/4}$, N=6 0111: $f_{SAMPLING}=f_{DTS/4}$, N=8 1000: $f_{SAMPLING}=f_{DTS/8}$, N=6 1001: $f_{SAMPLING}=f_{DTS/8}$, N=8 1010: $f_{SAMPLING}=f_{DTS/16}$, N=5 1011: $f_{SAMPLING}=f_{DTS/16}$, N=6 1100: $f_{SAMPLING}=f_{DTS/16}$, N=8 1101: $f_{SAMPLING}=f_{DTS/32}$, N=5 1110: $f_{SAMPLING}=f_{DTS/32}$, N=6 1111: $f_{SAMPLING}=f_{DTS/32}$, N=8
7	**MSM**	Master/Slave mode 0: No action 1: The effect of an event on the trigger input (TRGI) is delayed to allow a perfectsynchronization between the current timer and its slaves (through TRGO). It is useful if you want to synchronize several timers on a single external event.
6:4	**TS**	Trigger selection This bit-field selects the trigger input to be used to synchronize the counter 000: Internal Trigger 0 (ITR0). 001: Internal Trigger 1 (ITR1). 010: Internal Trigger 2 (ITR2). 011: Internal Trigger 3 (ITR3). 100: TI1 Edge Detector (TI1F_ED) 101: Filtered Timer Input 1 (TI1FP1) 110: Filtered Timer Input 2 (TI2FP2) 111: External Trigger input (ETRF) ***Note:*** *These bits must be changed only when they are not used (e.g. when SMS=000) to avoid wrong edge detections at the transition.*
3	**-**	RFU: **Reserved, read as 0**
2:0	**SMS**	Slave mode selection 000: Slave mode disabled - if CEN = ‘1 then the prescaler is clocked directly by the internal clock. 001: Encoder mode 1 - Counter counts up/down on TI2FP1

bit	name	functional description
		edge depending on TI1FP2 level. 010: Encoder mode 2 - Counter counts up/down on TI1FP2 edge depending on TI2FP1 level. 011: Encoder mode 3 - Counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the level of the other input. 100: Reset Mode - Rising edge of the selected trigger input (TRGI) reinitializes the counter and generates an update of the registers. 101: Gated Mode - The counter clock is enabled when the trigger input (TRGI) is high. The counter stops (but is not reset) as soon as the trigger becomes low. 110: Trigger Mode - The counter starts at a rising edge of the trigger TRGI (but it is notreset). 111: External Clock Mode 1 - Rising edges of the selected trigger (TRGI) clock the counter.

30.5.4 GPTIMx DMA and Interrupt Enable Register (GPTIMx_DIER)

NAME	GPTIMx_DIER(x=0,1,2)							
Offset	0x0000000C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				CC4BURSTEN	CC3BURSTEN	CC2BURSTEN	CC1BURSTEN
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-	TDE	-	CC4DE	CC3DE	CC2DE	CC1DE	UDE
access	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	TIE	-			CC2IE	CC1IE	UIE
access	U-0	R/W-0	U-0			R/W-0	R/W-0	R/W-0

bit	name	functional description
31:20	**-**	RFU: **Reserved, read as 0**
19	**CC4BURSTEN**	CC4 Burst Enable 0: Single mode, access to CCR only 1: Burst mode, address and length of accesses configured via DCR
18	**CC3BURSTEN**	CC3 Burst Enable

bit	name	functional description
		0: Single mode, access to CCR only 1: Burst mode, address and length of accesses configured via DCR
17	**CC2BURSTEN**	CC2 Burst Enable 0: Single mode, access to CCR only 1: Burst mode, address and length of accesses configured via DCR
16	**CC1BURSTEN**	CC1 Burst Enable 0: Single mode, access to CCR only 1: Burst mode, address and length of accesses configured via DCR
15	**-**	RFU: **Reserved, read as 0**
14	**TDE**	Triggered DMA Enable 0: Trigger DMA request disabled. 1: Trigger DMA request enabled.
13	**-**	RFU: **Reserved, read as 0**
12	**CC4DE**	Capture/Compare 4 DMA request enable 0: CC4 DMA request disabled. 1: CC4 DMA request enabled.
11	**CC3DE**	Capture/Compare 3 DMA request enable 0: CC3 DMA request disabled. 1: CC3 DMA request enabled.
10	**CC2DE**	Capture/Compare 2 DMA request enable 0: CC2 DMA request disabled. 1: CC2 DMA request enabled.
9	**CC1DE**	Capture/Compare 1 DMA request enable 0: CC1 DMA request disabled. 1: CC1 DMA request enabled.
8	**UDE**	Update DMA request enable 0: Update DMA request disabled. 1: Update DMA request enabled.
7	**-**	RFU: **Reserved, read as 0**
6	**TIE**	Trigger interrupt enable 0: Trigger interrupt disabled. 1: Trigger interrupt enabled.
5	**-**	RFU: **Reserved, read as 0**
4	**CC4IE**	Capture/Compare 4 interrupt enable 0: CC4 interrupt disabled. 1: CC4 interrupt enabled.
3	**CC3IE**	Capture/Compare 3 interrupt enable 0: CC3 interrupt disabled. 1: CC3 interrupt enabled.
2	**CC2IE**	Capture/Compare 2 interrupt enable

bit	name	functional description
		0: CC2 interrupt disabled. 1: CC2 interrupt enabled.
1	**CC1IE**	Capture/Compare 1 interrupt enable 0: CC1 interrupt disabled. 1: CC1 interrupt enabled.
0	**UIE**	Update interrupt enable 0: Update interrupt disabled. 1: Update interrupt enabled.

30.5.5 GPTIMx Interrupt Status Register (GPTIMx_ISR)

NAME	GPTIMx_ISR(x=0,1,2)							
Offset	0x00000010							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			CC4OF	CC3OF	CC2OF	CC1OF	-
access	U-0			R/W-0	R/W-0	R/W-0	R/W-0	U-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	TIF	-	CC4IF	CC3IF	CC2IF	CC1IF	UIF
access	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:13	**-**	RFU: **Reserved, read as 0**
12	**CC4OF**	Capture/Compare 4 overcapture flag Refer to CC1OF description
11	**CC3OF**	Capture/Compare 3 overcapture flag Refer to CC1OF description
10	**CC2OF**	Capture/compare 2 overcapture flag Refer to CC1OF description
9	**CC1OF**	Capture/Compare 1 overcapture flag This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to 1. 0: No overcapture has been detected. 1: The counter value has been captured while CC1IF flag was already set
8:7	**-**	RFU: **Reserved, read as 0**

bit	name	functional description
6	**TIF**	Trigger event Interrupt Flag, write 1 to clear
5	**-**	RFU: **Reserved, read as 0**
4	**CC4IF**	CC4 Interrupt Flag, write 1 to clear
3	**CC3IF**	CC3 Interrupt Flag, write 1 to clear
2	**CC2IF**	CC2 Interrupt Flag, write 1 to clear
1	**CC1IF**	Capture/compare 1 interrupt flag **If channel CC1 is configured as output:** This flag is set by hardware when the counter matches the compare value **If channel CC1 is configured as input:** This bit is set by hardware on a capture. It is cleared by software or by reading the TIMx_CCR1 register.
0	**UIF**	Update interrupt flag – This bit is set by hardware on an update event. It is cleared by software writing 1. 0: No update occurred. 1: Update interrupt pending. This bit is set by hardware when the registers are updated: – At overflow or underflow and if the UDIS=0 in the GPTIMx_CR1 register. – When CNT is reinitialized by software using the UG bit in TIMx_EGR register, if URS=0 and UDIS=0 in the GPTIMx_CR1 register. – When CNT is reinitialized by a trigger event (refer to the synchro control register description), if URS=0 and UDIS=0 in the TIMx_CR1 register.

30.5.6 GPTIMx Event Generation Register (GPTIMx_EGR)

NAME	**GPTIMx_EGR(x=0,1,2)**							
Offset	0x00000014							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

name	-	TG	-	CC2G	CC1G	UG
access	U-0	W-0	U-0	W-0	W-0	W-0

bit	name	functional description
31:7	-	RFU: **Reserved, read as 0**
6	**TG**	Trigger generation This bit is set by software in order to generate an event, it is automatically cleared by hardware.
5:3	-	RFU: **Reserved, read as 0**
2	**CC2G**	Capture/compare 2 generation
1	**CC1G**	Capture/compare 1 generation **If channel CC1 is configured as output:** CC1IF flag is set, Corresponding interrupt or DMA request is sent if enabled. **If channel CC1 is configured as input:** The current value of the counter is captured in GPTIMx_CCR1 register. The CC1IF flag is set, the corresponding interrupt or DMA request is sent if enabled.
0	**UG**	Update generation This bit can be set by software, it is automatically cleared by hardware. When the software sets UG, the counter is reinitialized and the shadow register is updated, and the prescaler counter is cleared to zero.

30.5.7 GPTIMx Capture/Compare Mode Register1 (GPTIMx_CCMR1)

This register is multiplexed into two different sets of functions in the output compare and input capture configurations.

NAME	GPTIMx_CCMR1(x=0,1,2)							
Offset	0x00000018							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	OC2CE	OC2M			OC2PE	OC2FE	CC2S	
	IC2F				IC2PSC		CC2S	
access	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	OC1CE	OC1M			OC1PE	OC1FE	CC1S	

	IC1F				IC1PSC		CC1S
access	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-00

Output compare mode

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15	**OC2CE**	Output compare 2 clear enable
14:12	**OC2M**	Output compare 2 mode
11	**OC2PE**	Output compare 2 preload enable
10	**OC2FE**	Output compare 2 fast enable
9:8	**CC2S**	Capture/Compare 2 selection 00: CC2 channel is configured as output 01: CC2 channel is configured as input, IC2 is mapped on TI2 10: CC2 channel is configured as input, IC2 is mapped on TI1 11: CC2 channel is configured as input, IC2 is mapped on TRC. ***Note:*** *CC2S bits are writable only when the channel is OFF (CC2E = 0 in GPTIMx_CCER).*
7	**OC1CE**	Output Compare 1 Clear Enable 0: OC1Ref is not affected by the ETRF input 1: OC1Ref is cleared as soon as a High level is detected on ETRF input
6:4	**OC1M**	Output compare 1 mode，These bits define the behavior of the output reference signal OC1REF 000: Frozen - The comparison between the output compare register GPTIMx_CCR1 and the counter GPTIMx_CNT has no effect on the outputs. 001: Set channel 1 to active level on match. OC1REF signal is forced high when the counter GPTIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 010: Set channel 1 to inactive level on match. OC1REF signal is forced low when the counter TIMx_CNT matches the capture/compare register 1 (TIMx_CCR1). 011: Toggle - OC1REF toggles when GPTIMx_CNT=GPTIMx_CCR1. 100: Force inactive level - OC1REF is forced low. 101: Force active level - OC1REF is forced high. 110: PWM mode 1 - In upcounting, channel 1 is active as long as GPTIMx_CNT<GPTIMx_CCR1 else inactive. In downcounting, channel 1 is inactive (OC1REF=0) as long as GPTIMx_CNT>GPTIMx_CCR1 else active (OC1REF=1). 111: PWM mode 2 - In upcounting, channel 1 is inactive as long as GPTIMx_CNT<GPTIMx_CCR1 else active. In downcounting, channel 1 is active as long as

bit	name	functional description
		GPTIMx_CNT>GPTIMx_CCR1 else inactive.
3	**OC1PE**	Output compare 1 preload enable 0: Preload register on GPTIMx_CCR1 disabled. GPTIMx_CCR1 can be written at anytime, 1: Preload register on GPTIMx_CCR1 enabled. Read/Write operations access the preload register. GPTIMx_CCR1 preload value is loaded in the active register at each update event.
2	**OC1FE**	Output compare 1 fast enable 0: Turn off fast enable, the trigger input will not affect the comparison output 1: Turn on fast enable, the trigger input will immediately change the OC1REF to the output when the comparison value matches, regardless of the actual current comparison **Note:**This function is only valid when the current channel is configured in PWM1 or PWM2 mode
1:0	**CC1S**	CC1 channel Selection 00: CC1 channel is configured as output. 01: CC1 channel is configured as input, IC1 is mapped on TI1. 10: CC1 channel is configured as input, IC1 is mapped on TI2. 11: CC1 channel is configured as input, IC1 is mapped on TRC. ***Note:*** *CC1S bits are writable only when the channel is OFF (CC1E = 0 in GPTIMx_CCER).*

Input capture mode

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:12	**IC2F**	IC2 Filter
11:10	**IC2PSC**	IC2 Prescaler
9:8	**CC2S**	Capture/Compare2 channel Selection 00: CC2 channel is configured as output. 01: CC2 channel is configured as input, IC2 is mapped on TI2. 10: CC2 channel is configured as input, IC2 is mapped on TI1. 11: CC2 channel is configured as input, IC2 is mapped on TRC. ***Note:*** *CC2S bits are writable only when the channel is OFF (CC2E = 0 in TIMx_CCER).*
7:4	**IC1F**	Input capture 1 filter This bit-field defines the frequency used to sample TI1 input and the length of the digital filterapplied to TI1. 0000: No filter, sampling is done at fDTS 0001: $f_{SAMPLING}=f_{CK\_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK\_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK\_INT}$, N=8

bit	name	functional description
		0100: $f_{SAMPLING}=f_{DTS/2}$, N=6 0101: $f_{SAMPLING}=f_{DTS/2}$, N=8 0110: $f_{SAMPLING}=f_{DTS/4}$, N=6 0111: $f_{SAMPLING}=f_{DTS/4}$, N=8 1000: $f_{SAMPLING}=f_{DTS/8}$, N=6 1001: $f_{SAMPLING}=f_{DTS/8}$, N=8 1010: $f_{SAMPLING}=f_{DTS/16}$, N=5 1011: $f_{SAMPLING}=f_{DTS/16}$, N=6 1100: $f_{SAMPLING}=f_{DTS/16}$, N=8 1101: $f_{SAMPLING}=f_{DTS/32}$, N=5 1110: $f_{SAMPLING}=f_{DTS/32}$, N=6 1111: $f_{SAMPLING}=f_{DTS/32}$, N=8
3:2	**IC1PSC**	Input capture 1 prescaler The prescaler is reset as soon as CC1E=0 (TIMx_CCER register). 00: No prescaler, capture is done each time an edge is detected on the capture input 01: Capture is done once every 2 events 10: Capture is done once every 4 events 11: Capture is done once every 8 events
1:0	**CC1S**	Capture/Compare1 channel Selection 00: CC1 channel is configured as output 01: CC1 channel is configured as input, IC1 is mapped on TI1 10: CC1 channel is configured as input, IC1 is mapped on TI2 11: CC1 channel is configured as input, IC1 is mapped on TRC. This mode is working only if an internal trigger input is selected through TS bit (TIMx_SMCR register) ***Note:*** *CC1S bits are writable only when the channel is OFF (CC1E = 0 in GPTIMx_CCER).*

30.5.8 GPTIMx Capture/Compare Mode Register2 (GPTIMx_CCMR2)

This register is multiplexed into two different sets of functions in the output compare and input capture configurations.

NAME	GPTIMx_CCMR2(x=0,1,2)							
Offset	0x0000001C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							

bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	OC4CE	OC4M			OC4PE	OC4FE	CC4S	
	IC2F				IC2PSC		CC4S	
access	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	OC3CE	OC3M			OC3PE	OC3FE	CC3S	
	IC3F				IC3PSC		CC3S	
access	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Output compare mode

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15	**OC4CE**	Output compare 4 clear enable
14:12	**OC4M**	Output compare 4 mode
11	**OC4PE**	Output compare 4 preload enable
10	**OC4FE**	Output compare 4 fast enable
9:8	**CC4S**	Capture/Compare 4 selection 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. ***Note:*** *CC4S bits are writable only when the channel is OFF (CC4E = 0 in GPTIMx_CCER).*
7	**OC3CE**	Output compare 3 clear enable
6:4	**OC3M**	Output compare 3 mode
3	**OC3PE**	Output compare 3 preload enable
2	**OC3FE**	Output compare 3 fast enable
1:0	**CC3S**	Capture/Compare 3 selection This bit-field defines the direction of the channel (input/output) as well as the used input. 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped on TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped on TRC. ***Note:*** *CC3S bits are writable only when the channel is OFF (CC3E = 0 in GPTIMx_CCER).*

Input capture mode

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:12	**IC4F**	Input capture 4 filter
11:10	**IC4PSC**	Input capture 4 prescaler

bit	name	functional description
9:8	**CC4S**	Capture/Compare 4 selection 00: CC4 channel is configured as output 01: CC4 channel is configured as input, IC4 is mapped on TI4 10: CC4 channel is configured as input, IC4 is mapped on TI3 11: CC4 channel is configured as input, IC4 is mapped on TRC. ***Note:*** *CC4S bits are writable only when the channel is OFF (CC4E = 0 in GPTIMx_CCER).*
7:4	**IC3F**	Input capture 3 filter
3:2	**IC3PSC**	Input capture 3 prescaler
1:0	**CC3S**	Capture/Compare 3 selection 00: CC3 channel is configured as output 01: CC3 channel is configured as input, IC3 is mapped on TI3 10: CC3 channel is configured as input, IC3 is mapped on TI4 11: CC3 channel is configured as input, IC3 is mapped on TRC. ***Note:*** *CC3S bits are writable only when the channel is OFF (CC3E = 0 in GPTIMx_CCER).*

30.5.9 GPTIMx Capture/Compare Enable Register (GPTIMx_CCER)

NAME	GPTIMx_CCER(x=0,1,2)							
Offset	0x00000020							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-		CC4P	CC4E	-		CC3P	CC3E
access	U-0		R/W-0	R/W-0	U-0		R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		CC2P	CC2E	-		CC1P	CC1E
access	U-0		R/W-0	R/W-0	U-0		R/W-0	R/W-0

bit	name	functional description
31:14	**-**	RFU: **Reserved, read as 0**
13	**CC4P**	Capture/Compare 4 output polarity Refer to CC1P description
12	**CC4E**	Capture/Compare 4 output enable Refer to CC1E description
11:10	**-**	RFU: **Reserved, read as 0**
9	**CC3P**	Capture/Compare

bit	name	functional description
		Refer to CC1P description
8	**CC3E**	Capture/Compare 3 output enable Refer to CC1E description
7:6	**-**	RFU: **Reserved, read as 0**
5	**CC2P**	Capture/Compare 2 output Refer to CC1P description
4	**CC2E**	Capture/Compare 2 output enable Refer to CC1E description
3:2	**-**	RFU: **Reserved, read as 0**
1	**CC1P**	Capture/Compare 1 output polarity **CC1 channel configured as output:** 0: OC1 active high 1: OC1 active low. **CC1 channel configured as input:** CC1NP/CC1P is used to select the polarity of TI1FP1 and TI2FP1 00: Non-inverting/rising edge 01: Inverse/falling edge 10: Reserved, do not use 11: Non-inverse, both up and down edges are valid
0	**CC1E**	Capture/Compare 1 output enable **CC1 channel configured as output:** 0: Off - OC1 is not active. 1: On - OC1 signal is output on the corresponding output pin. **CC1 channel configured as input:** This bit determines if a capture of the counter value can actually be done into the input capture/compare register 1 (TIMx_CCR1) or not. 0: Capture disabled. 1: Capture enabled.

Output control bit for standard Ocx channels

CcxE bit	Ocx output status
0	Output Disabled (OCx=0, OCx_EN=0)
1	OCx=OCxREF + Polarity, OCx_EN=1

30.5.10 GPTIMx Counter Register (GPTIMx_CNT)

NAME	GPTIMx_CNT(x=0,1,2)							
Offset	0x00000024							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CNT[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CNT[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**CNT**	Counter value

30.5.11 GPTIMx Prescaler Register (GPTIMx_PSC)

NAME	GPTIMx_PSC(x=0,1,2)							
Offset	0x00000028 + x*0x400							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PSC[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PSC[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**PSC**	Prescaler value The counter clock frequency CK_CNT is equal to fCK_PSC / (PSC[15:0] + 1). PSC contains the value to be loaded in the active prescaler register at each update event.

30.5.12 GPTIMx Auto-Reload Register (GPTIMx_ARR)

NAME	GPTIMx_ARR(x=0,1,2)							
Offset	0x0000002C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ARR[15:8]							
access	R/W-1111 1111							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ARR[7:0]							
access	R/W-1111 1111							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**ARR**	Prescaler value ARR is the value to be loaded in the actual auto-reload register.

30.5.13 GPTIMx Capture/Compare Register1 (GPTIMx_CCR1)

NAME	GPTIMx_CCR1(x=0,1,2)							
Offset	0x00000034							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR1[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR1[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**CCR1**	Capture/Compare 1 value **If channel CC1 is configured as output**:

bit	name	functional description
		CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value). **If channel CC1is configured as input**: CCR1 is the counter value transferred by the last input capture 1 event (IC1).

30.5.14 GPTIMx Capture/Compare Register2 (GPTIMx_CCR2)

NAME	GPTIMx_CCR2(x=0,1,2)							
Offset	0x00000038							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR2[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR2[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:0	**CCR2**	**If channel CC2 is configured as output:** CCR2 is the value to be loaded in the actual capture/compare 2 register (preload value). **If channel CC2 is configured as input** CCR2 is the counter value transferred by the last input capture 2 event (IC2).

30.5.15 GPTIMx Capture/Compare Register3 (GPTIMx_CCR3)

NAME	GPTIMx_CCR3(x=0,1,2)							
Offset	0x0000003C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							

bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR3[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR3[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**CCR3**	**If channel CC3 is configured as output:** CCR3 is the value to be loaded in the actual capture/compare 3 register (preload value). **If channel CC3 is configured as input** CCR3 is the counter value transferred by the last input capture 3 event (IC3).

30.5.16 GPTIMx Capture/Compare Register4 (GPTIMx_CCR4)

NAME	GPTIMx_CCR4(x=0,1,2)							
Offset	0x00000040							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR4[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR4[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**CCR4**	**If channel CC4 is configured as output:** CCR4 is the value to be loaded in the actual capture/compare 4 register (preload value). **If channel CC4 is configured as input** CCR4 is the counter value transferred by the last input capture 4 event (IC4).

30.5.17 GPTIMx DMA Control Register (GPTIMx_DCR)

NAME	GPTIMx_DCR(x=0,1,2)							
Offset	0x00000048							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			DBL				
access	U-0			R/W-0 0000				
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			DBA				
access	U-0			R/W-0 0000				

bit	name	functional description
31:13	-	RFU: **Reserved, read as 0**
12:8	**DBL**	DMA burst length The timer recognizes a burst transfer when a read or a write access is done to the GPTIMx_DMAR address 00000: 1 transfer 00001: 2 transfers 00010: 3 transfers ... 10001: 18 transfers.
7:5	-	RFU: **Reserved, read as 0**
4:0	**DBA**	DMA Burst offset Address 00000: GPTIM_CR1 00001: GPTIM_CR2 00010: GPTIM_SMCR …… **Note:** When the DBA + DBL exceeds the GPTIM register address range, the actual BURST is transmitted automatically after the GPTIM maximum register address, that is, the BURST length will be shortened

30.5.18 GPTIMx DMA Access Register (GPTIMx_DMAR)

NAME	GPTIMx_DMAR(x=0,1,2)							
Offset	0x0000004C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DMAR[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DMAR[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:0	**DMAR**	DMA register for burst accesses When using DMA BURST transmission, set the DMA channel peripheral address to GPTIM_DMAR, and GPTIM generates multiple DMA requests according to the value of DBL.

30.5.19 GPTIMx Internal Trigger Select Register (GPTIMx_ITRSEL)

NAME	**GPTIMx_ITRSEL(x=0,1,2)**							
Offset	0x00000060							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ITR3SEL		ITR2SEL		ITR1SEL		ITR0SEL	
access	R/W-00		R/W-00		R/W-00		R/W-00	

bit	name	functional description
31:8	-	RFU: **Reserved, read as 0**
7:6	ITR3SEL	Internal Trigger Source Selection Internal trigger signal (ITRX) capture For details, see 30.4.4 Capture of Internal Trigger Signal (ITRX)
5:4	ITR2SEL	
3:2	ITR1SEL	
1:0	ITR0SEL	

31 32bits Basic Timer (BSTIM32)

31.1 Introduction

The FM33LG0 contains of a 32bits basic timer.

The basic timer contains a 32bits auto-reload counter and a programmable prescaler.

The basic timer is mainly used as generic timers for time-base generation, and can also generate trigger events to drive ADC sampling.

31.2 Features

- 32bits auto-reload upcounter
- 32bits programmable prescaler used to divide the counter clock frequency by real-time adjustment
- ADC timing trigger
- Interrupt generation on the update event: counter overflow

31.3 Block Diagram

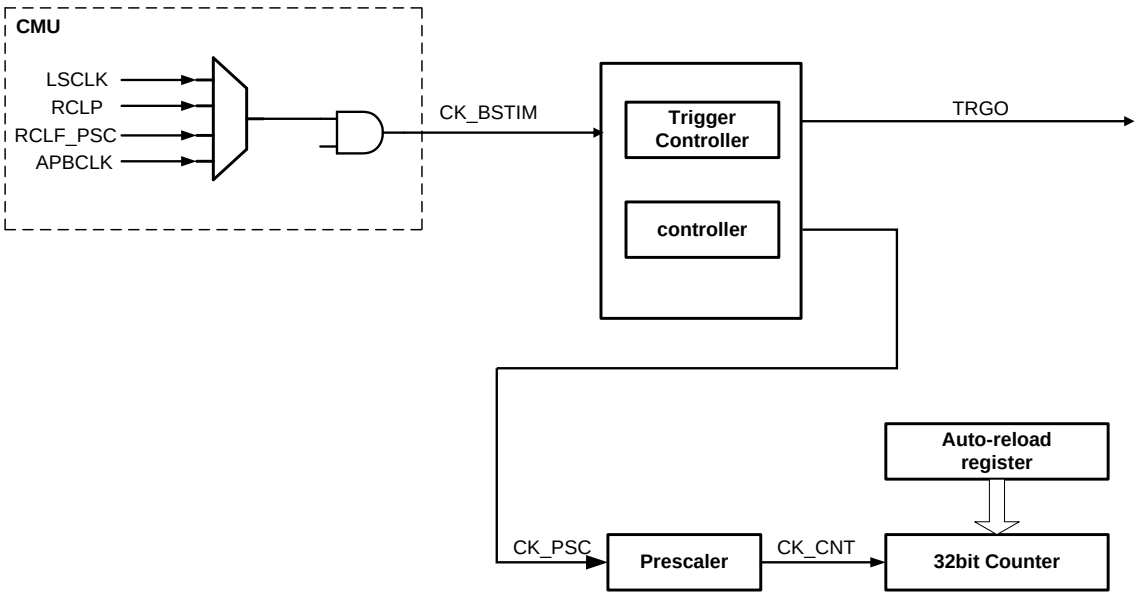


Figure 31–1 32bits Basic Timer Block Diagram

31.4 Functional Description

31.4.1 Time-Base Unit

The timing unit of the basic timer consists of a 32-bit counter and an auto-reload register. The counter counts up. The count clock can be obtained by dividing the frequency of APBCLK, LSCLK, RCLP, and RCLF_PSC by a 32-bit prescaler.

Counter, Auto-Reload Register Prescaler registers can be overwritten or read by software, even while the counter is running.

The time-base unit includes:

- Counter Register (BSTIM32_CNT)
- Prescaler Register (BSTIM32_PSC)
- Auto-Reload Register (BSTIM32_ARR)

The ARR contains the preload function. Software can read or write ARR directly or just access its cache controlled by ARPE (Auto Reload Preload Enable) register. When ARPE=1, software read or write ARR is accessing its cache register, and when update event (BSTIM32_CNT up overflow or down overflow) occurs, it will update the data in the cache register to ARR. Software can also trigger ARR update actively through register operation.

The BSTIM32_CNT is driven by the divider clock generated by BSTIM32_PSC, which is enabled only when the counter enable bit (CEN) is set. When CNT=ARR, the current round of counting ends and the update event is sent.

The BSTIM32_PSC is a synchronous prescaler capable of dividing APBCLK, LSCLK, RCLP and RCLF_PSC by any factor between 1 and 2^{32}. The PSC register is also cached, and rewriting the PSC is actually rewriting the cache register. The PSC is only updated from the cache register when a new update event occurs. Therefore, the software can rewrite the PSC in real time during the CNT counting process.

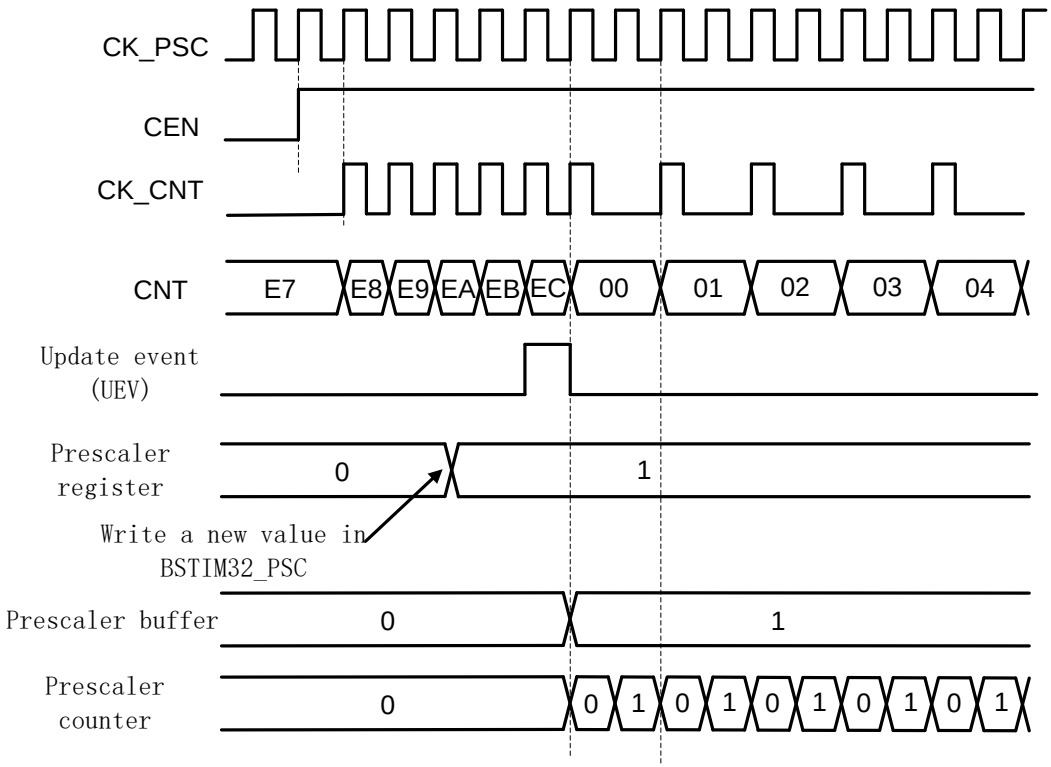


Figure 31–2 Counter Timing Diagram with Prescaler Division Change from 1 to 2

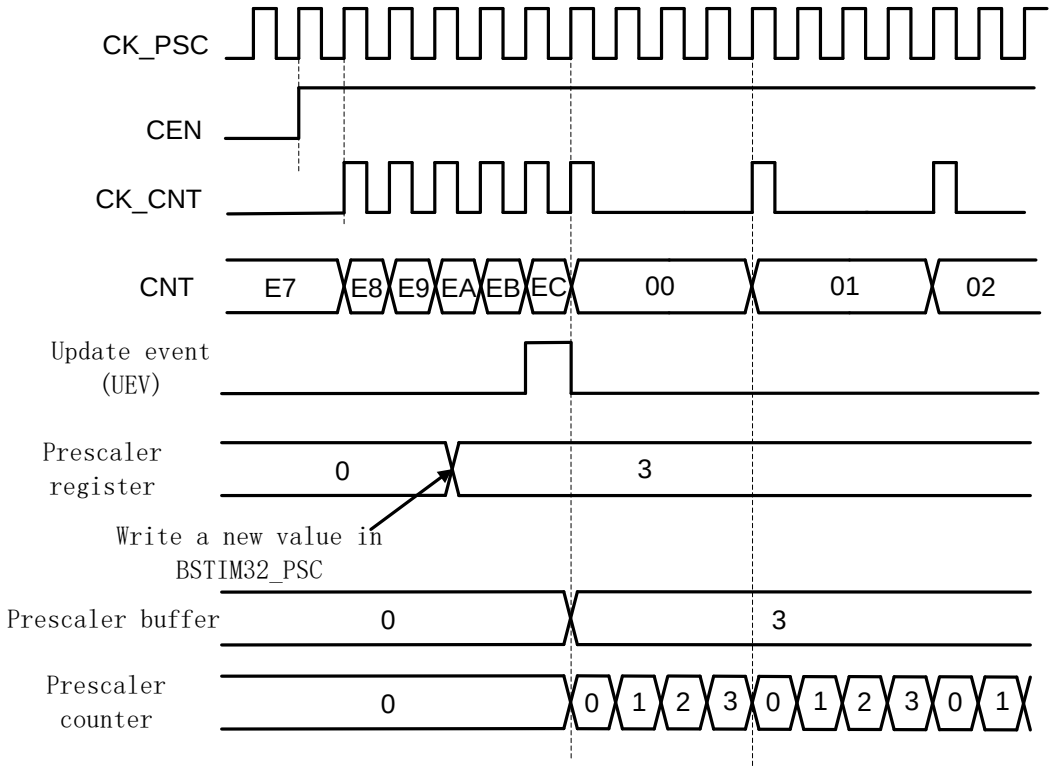


Figure 31–3 Counter Timing Diagram with Prescaler Division Change from 1 to 4

31.4.2 Counter Mode

The general purpose timer only supports up counting mode.

Upward counting

In this mode, the counter counts from 0 until CNT=ARR, then restarts from 0 and generates a counter overflow event.

The software can trigger the update event directly by setting the UG register, at which time the CNT and prescaler counter will be cleared automatically. Setting the UG register will not trigger setting the UIF (Update Interrupt Flag).

When an update event occurs, the following registers are updated and the UIF is set:

- BSTIM32_ARR is updated to the value in the cache
- BSTIM32_PSC is updated to the value in the cache

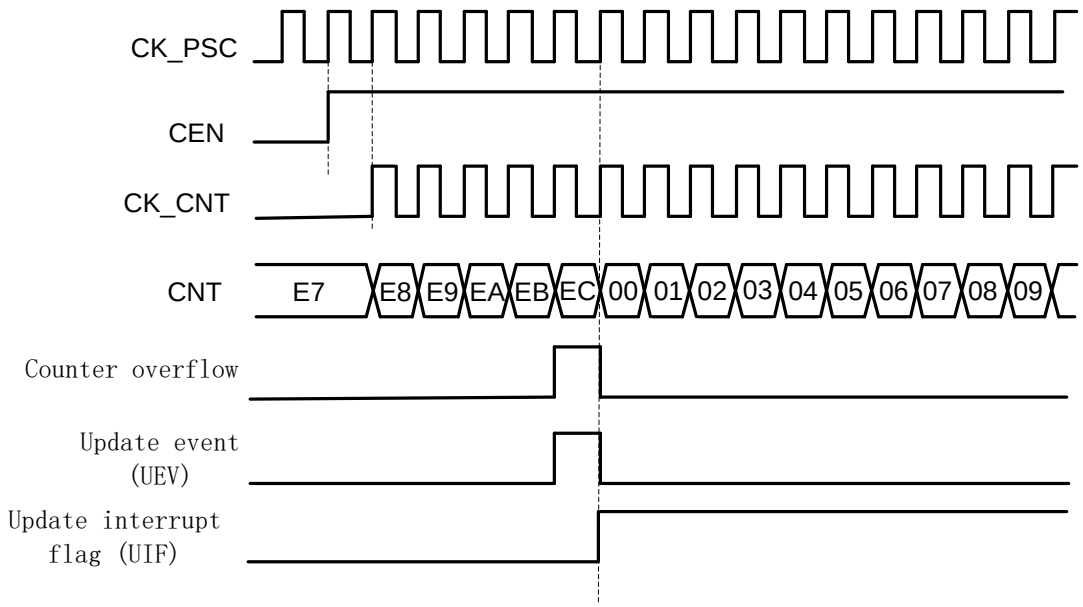


Figure 31–4 Upcounting, Internal Clock Divided by 1

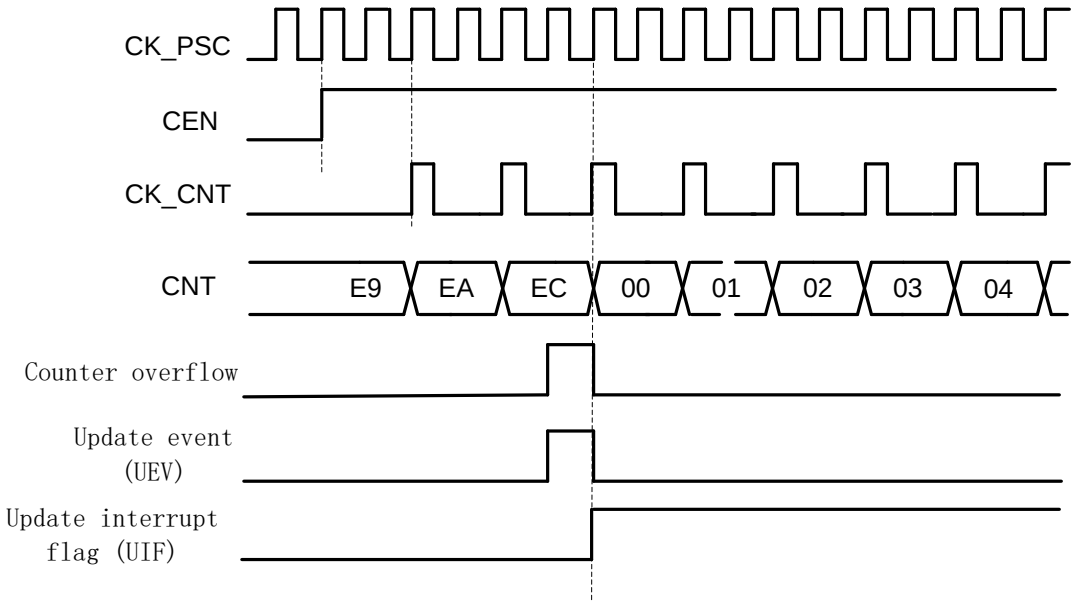


Figure 31–5 Upcounting, Internal Clock Divided by 2

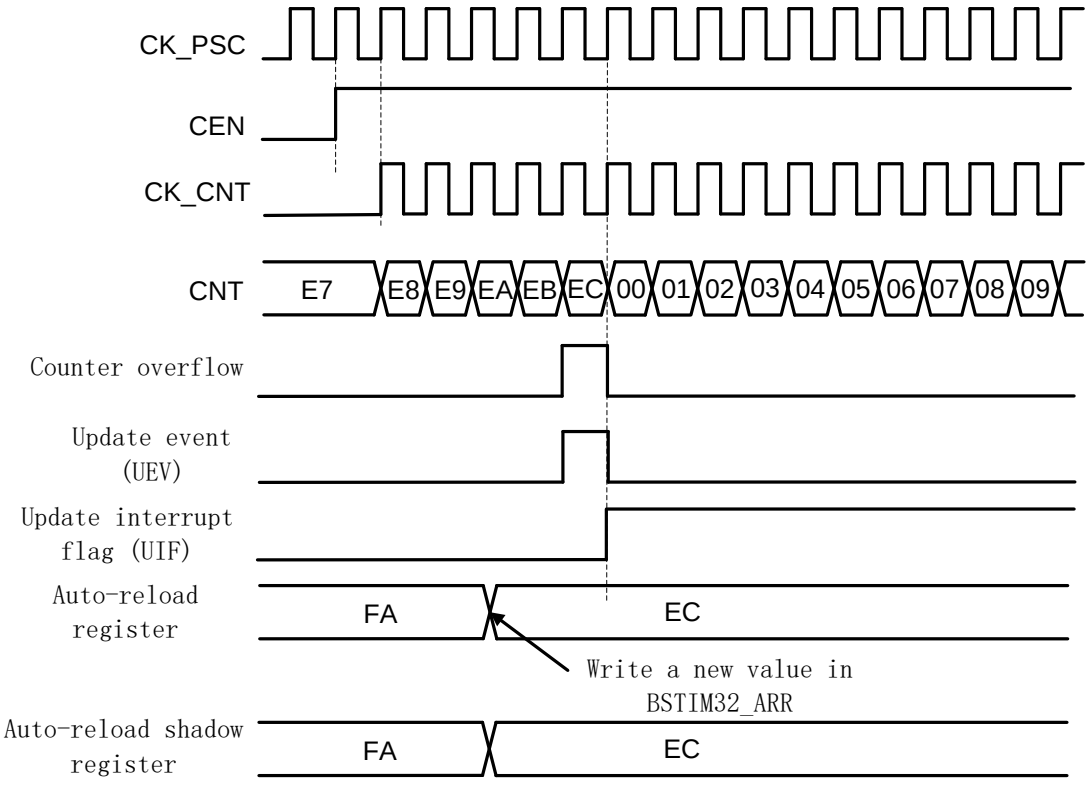


Figure 31–6 Counter Timing Diagram, Update Event When ARPE=0 (ARR is not Preloaded)

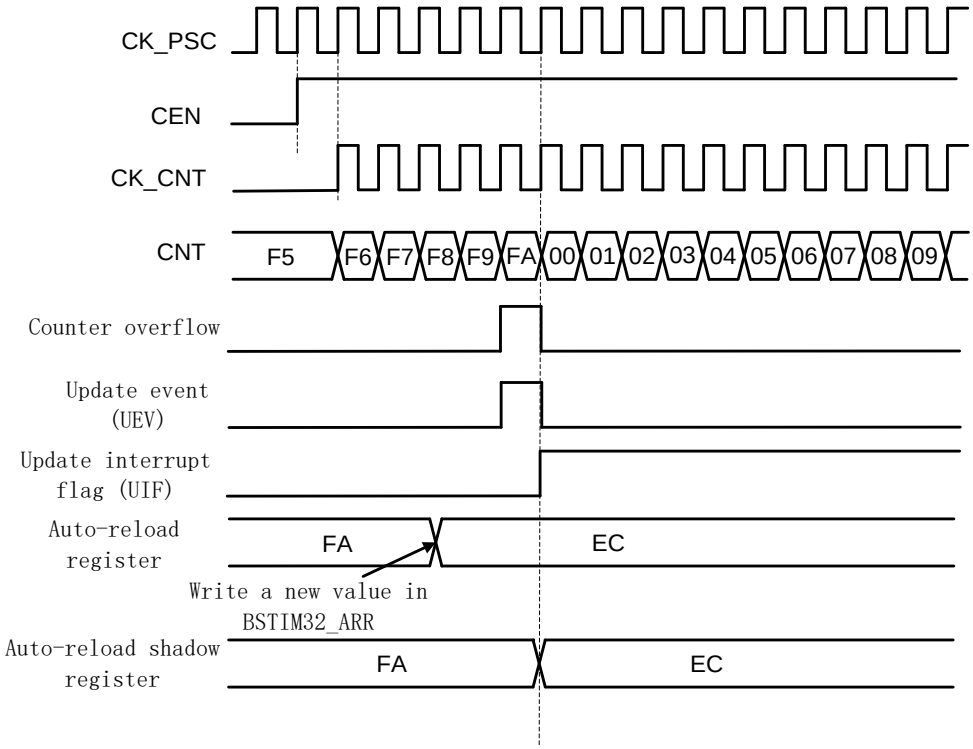


Figure 31–7 Counter Timing Diagram, Update Event When ARPE=1 (ARR is Preloaded)

31.4.3 Clock Source

The BSTIM clock is provided by the Internal clock source, and the register bits such as CEN and UG are software controlled.

After the software operates the UG register, the counter value will be reinitialized after the update signal is synchronized by CLK_PSC.

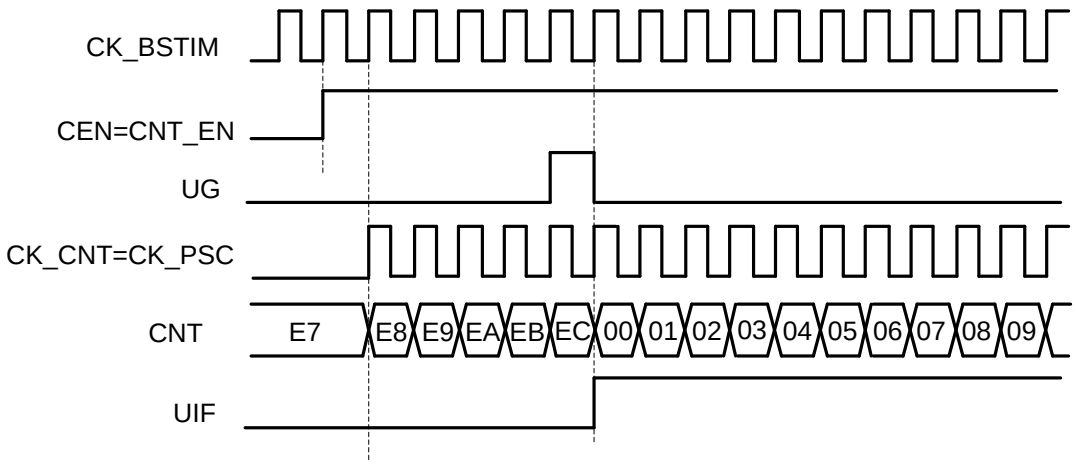


Figure 31–8 Internal Clock Divided by 1

31.4.4 Debug Mode

When the Cortex-M0 enters the debug mode, the timer can stop or continue to work, and its behavior is defined by the DBG_BT32_STOP register of the DBG module.

31.5 Register

Offset	Name	Symbol
BSTIM32(Base address: 0x40016000)		
0x00000000	BSTIM32 Control Register1	BSTIM32_CR1
0x00000004	BSTIM32 Control Register2	BSTIM32_CR2
0x0000000C	BSTIM32 Interrupt Enable Register	BSTIM32_IER
0x00000010	BSTIM32 Interrupt Status Register	BSTIM32_ISR
0x00000014	BSTIM32 Event Generation Register	BSTIM32_EGR
0x00000024	BSTIM32 Counter Register	BSTIM32_CNT
0x00000028	BSTIM32 Prescaler Register	BSTIM32_PSC
0x0000002C	BSTIM32 Auto-Reload Register	BSTIM32_ARR

31.5.1 BSTIM32 Control Register1 (BSTIM32_CR1)

NAME	BSTIM32_CR1							
Offset	0x00000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ARPE	-			OPM	URS	UDIS	CEN
access	R/W-0	U-0			R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:8	-	RFU: **Reserved, read as 0**
7	ARPE	Auto-Reload Preload Enable 0: ARR register is not enabled for preload 1: ARR register is enabled for preload
6:4	-	RFU: **Reserved, read as 0**
3	OPM	One Pulse Mode 0: Counter is not stopped at update event 1: Counter stops counting at the next update event (clearing the CEN bit).
2	URS	Update Request Select 0: Any of the following events generates an update interrupt: - Counter overflow/underflow

bit	name	functional description
		- Setting the UG bit 1: Only counter overflow/underflow generates an update interrupt or DMA request if enabled.
1	UDIS	Update Disable 0: UEV enabled. The Update (UEV) event is generated by one of the following events: - Counter overflow/underflow - Setting the UG bit 1: UEV disabled. The Update event is not generated, shadow registers keep their value(ARR, PSC). However the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.
0	CEN	Counter Enable 0: Counter disabled 1: Counter enabled **Note:** Trigger mode can set the CEN bit automatically by hardware.

31.5.2 BSTIM32 Control Register2 (BSTIM32_CR2)

NAME	BSTIM32_CR2							
Offset	0x00000004							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	MMS			-			
access	U-0	R/W-000			U-0			

bit	name	functional description
31:7	-	RFU: **Reserved, read as 0**
6:4	MMS	Master mode selection, these bits are used to select the information to be sent in master mode to slave timers for synchronization (TRGO). The combination is as follows: 000: the UG bit from the BSTIM_EGR register is used as a trigger output (TRGO).

bit	name	functional description
		001: the Counter enable signal, CNT_EN, is used as a trigger output (TRGO). 010: The update event is selected as a trigger output (TRGO). 011/100/111: RFU **Note:** The clock of the slave timer and ADC must be enabled prior to receiving events from the master timer.
3:0	-	RFU: **Reserved, read as 0**

31.5.3 BSTIM32 Interrupt Enable Register (BSTIM32_IER)

NAME	BSTIM32_IER							
Offset	0x0000000C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							UIE
access	U-0							R/W-0

bit	name	functional description
31:1	**-**	RFU: **Reserved, read as 0**
0	**UIE**	Update event Interrupt Enable 0: Update interrupt disabled. 1: Update interrupt disabled.

31.5.4 BSTIM32 Interrupt Status Register (BSTIM32_ISR)

NAME	BSTIM32_ISR							
Offset	0x00000010							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							UIF
access	U-0							R/W-0

bit	name	functional description
31:1	**-**	RFU: **Reserved, read as 0**
0	**UIF**	Update event Interrupt Flag, write 1 to clear. The UIF is set and the shadow register is updated when the following events occur -At overflow or underflow and if repetition counter= 0 and UDIS = 0. -When CNT is reinitialized by software using the UG bit, if URS = 0and UDIS = 0. -The event initialization counter is triggered, if URS = 0and UDIS = 0.

31.5.5 BSTIM32 Event Generation Register (BSTIM32_EGR)

NAME	**BSTIM32_EGR**							
Offset	0x00000014							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							UG
access	U-0							W-0

bit	name	functional description
31:1	**-**	RFU: **Reserved, read as 0**
0	**UG**	User Generate, this bit can be set by software, it is automatically cleared by hardware. When the software sets UG, the counter is reinitialized and the shadow register is updated, and the prescaler counter is cleared. ***Note:*** *Since the BSTIM counter operating clock and the system*

bit	name	functional description
		bus clock APBCLK are independent of each other, when the software sets the UG register, the UIF is not set immediately, but needs to be synchronized by CK_BSTIM before it is set, and the delay between the two is related to the frequency and phase relationship between APBCLK and CK_BSTIM, which is not a definite value.

31.5.6 BSTIM32 Counter Register (BSTIM32_CNT)

NAME	BSTIM32_CNT							
Offset	0x00000024							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	CNT[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	CNT[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CNT[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CNT[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:0	**CNT**	Counter value.

31.5.7 BSTIM32 Prescaler Register (BSTIM32_PSC)

NAME	BSTIM32_PSC							
Offset	0x00000028							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	PSC[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	PSC[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PSC[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PSC[7:0]							

access	R/W-0000 0000

bit	name	functional description
31:0	**PSC**	Prescaler value, the counter clock frequency CK_CNT is equal to $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[31:0]+1)$ This is a preload register whose contents are loaded into the shadow register when the update event occurs.

31.5.8 BSTIM32 Auto-Reload Register (BSTIM32_ARR)

NAME	BSTIM32_ARR							
Offset	0x0000002C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	ARR[31:24]							
access	R/W-1111 1111							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	ARR[23:16]							
access	R/W-1111 1111							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ARR[15:8]							
access	R/W-1111 1111							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ARR[7:0]							
access	R/W-1111 1111							

bit	name	functional description
31:0	**ARR**	Auto-Reload Register This is a preload register whose contents are loaded into the shadow register when the update event occurs.

32 16bits Basic Timer (BSTIM16)

32.1 Introduction

The FM33LG0 contains of a 16-bit basic timer.

The basic timer contains a 16-bit auto-reload counter and a programmable prescaler.

The basic timer is mainly used as generic timers for time-base generation, and can also generate trigger events to drive ADC sampling.

32.2 Features

- 16-bit auto-reload upcounter
- 16-bit programmable prescaler used to divide the counter clock frequency by real-time adjustment
- ADC timing trigger
- Interrupt generation on the update event: counter overflow

32.3 Block Diagram

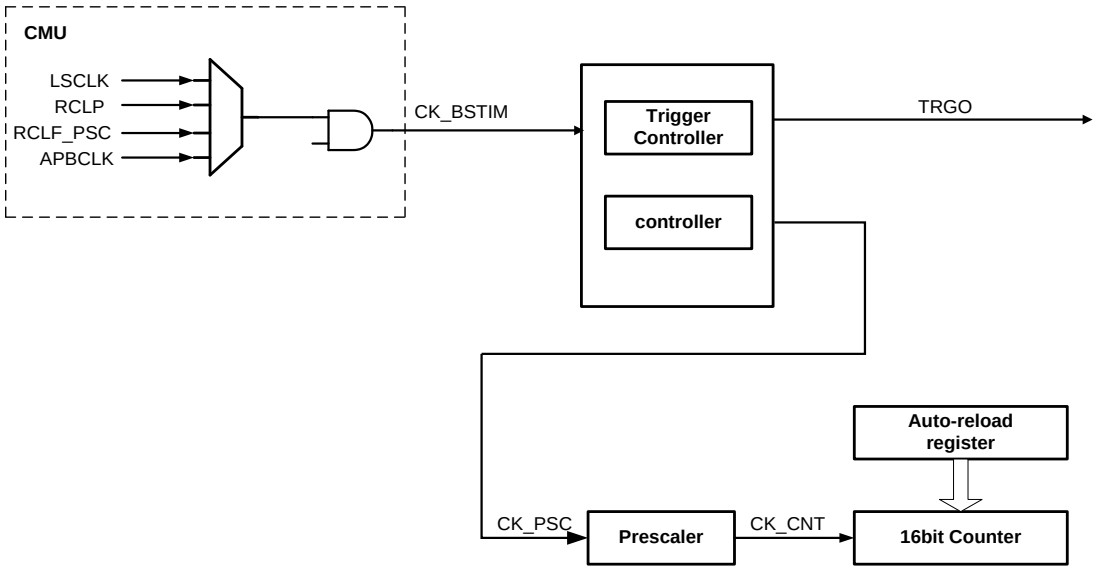


Figure 32–1 16bits Basic Timer Block Diagram

32.4 Functional Description

32.4.1 Time-Base Unit

The timing unit of the basic timer consists of a 16-bit counter and an auto-reload register. The counter counts up. The count clock can be obtained by dividing APBCLK, LSCLK, RCLP, and RCLF_PSC by a 16-bit prescaler.

The counter, auto-reload register, prescaler register can all be overwritten or read by software, even while the counter is running.

The time-base unit includes:

- Counter Register (BSTIM16_CNT)
- Prescaler Register (BSTIM16_PSC)
- Auto-Reload Register (BSTIM16_ARR)

The ARR contains the preload function. Software can read or write ARR directly or just access its cache controlled by ARPE (Auto Reload Preload Enable) register. When ARPE=1, software read or write ARR is accessing its cache register, and when update event (BSTIM16_CNT up overflow or down overflow) occurs, it will update the data in the cache register to ARR. Software can also trigger ARR update actively through register operation.

The BSTIM16_CNT is driven by the divider clock generated by BSTIM16_PSC, which is enabled only when the counter enable bit (CEN) is set. When CNT=ARR, the current round of counting ends and the update event is sent.

The BSTIM16_PSC is a synchronous prescaler capable of dividing APBCLK, LSCLK, RCLP and RCLF_PSC by any factor between 1 and 65536. The PSC register is also cached, and rewriting the PSC is actually rewriting the cache register. The PSC is only updated from the cache register when a new update event occurs. Therefore, the software can rewrite the PSC in real time during the CNT counting process.

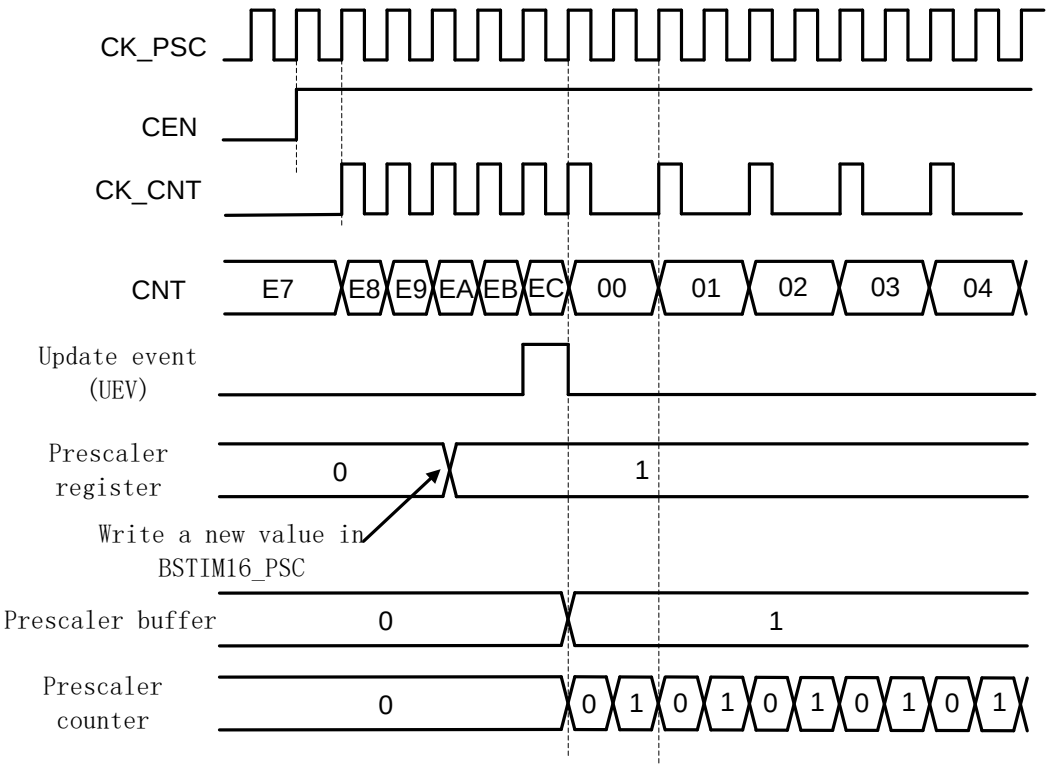


Figure 32–2 Counter Timing Diagram with Prescaler Division Change from 1 to 2

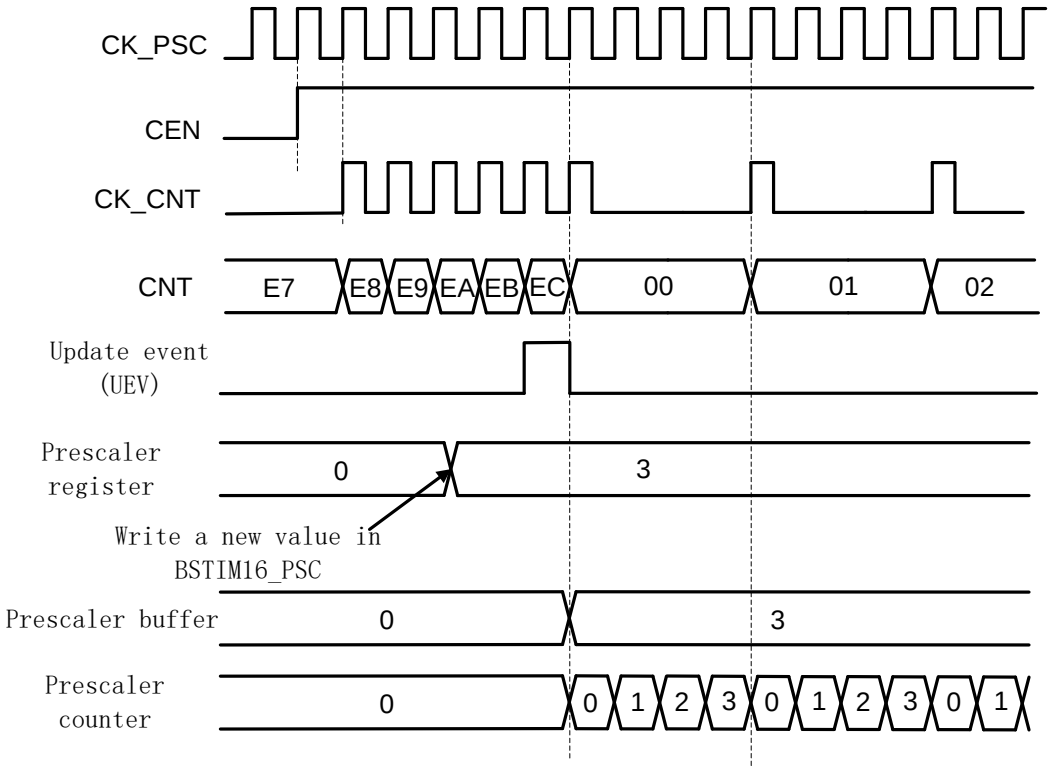


Figure 32–3 Counter Timing Diagram with Prescaler Division Change from 1 to 4

32.4.2 Counter Mode

The general purpose timer only supports up counting mode.

Upward counting

In this mode, the counter counts from 0 until CNT=ARR, then restarts from 0 and generates a counter overflow event.

The software can trigger the update event directly by setting the UG register, at which time the CNT and prescaler counter will be cleared automatically. Setting the UG register will not trigger setting the UIF (Update Interrupt Flag).

The update event can be disabled by setting the UDIS register, which avoids updating the value in the preload register to the working register.

When an update event occurs, the following registers are updated and the UIF is set:

- BSTIM16_ARR is updated to the value in the cache
- BSTIM16_PSC is updated to the value in the cache

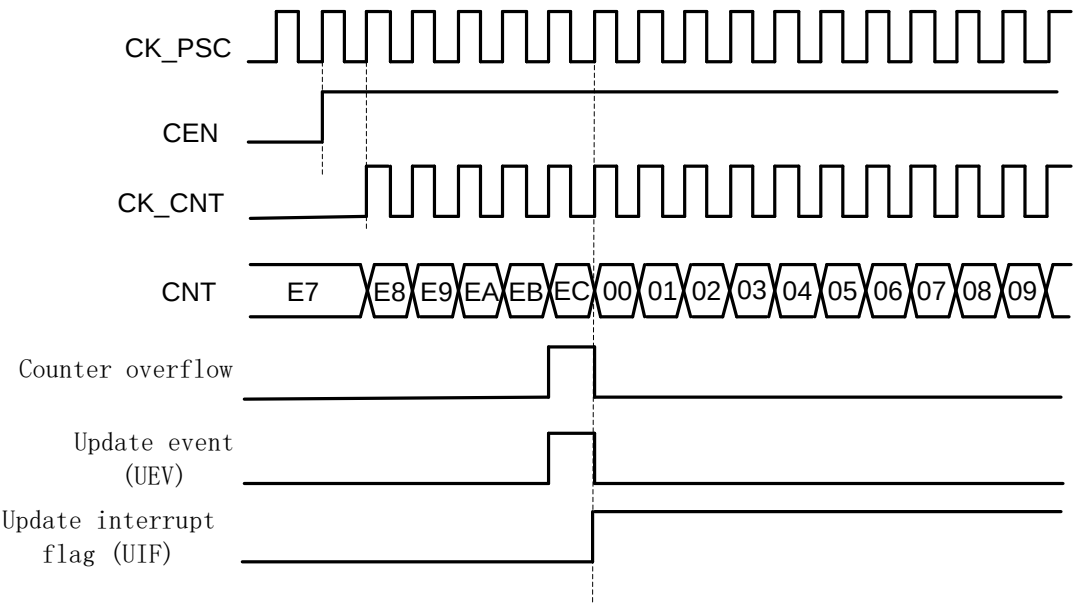


Figure 32–4 Upcounting, Internal Clock Divided by 1

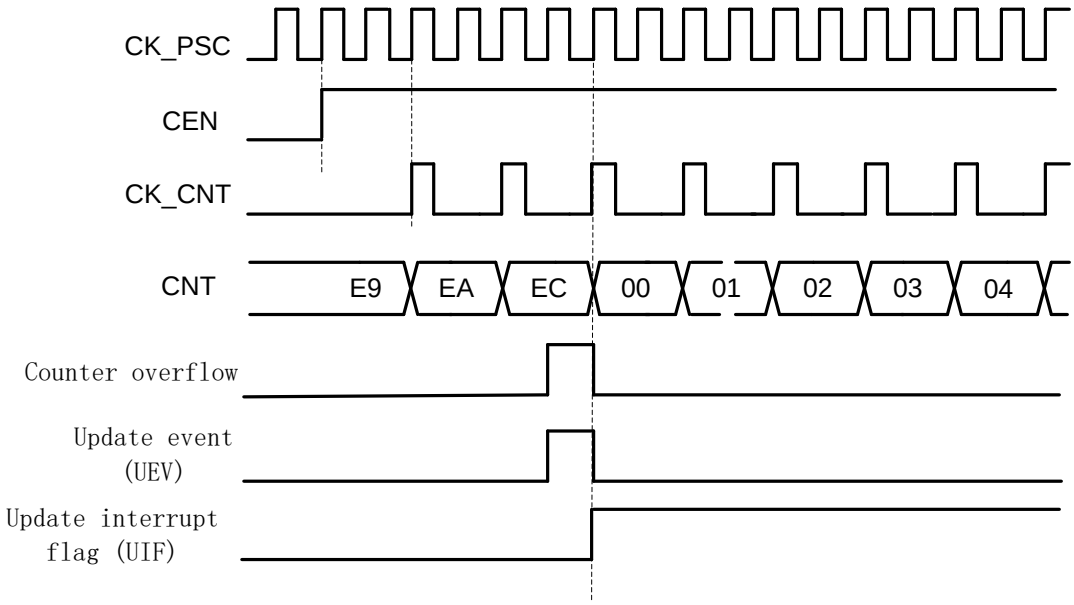


Figure 32–5 Upcounting, Internal Clock Divided by 2

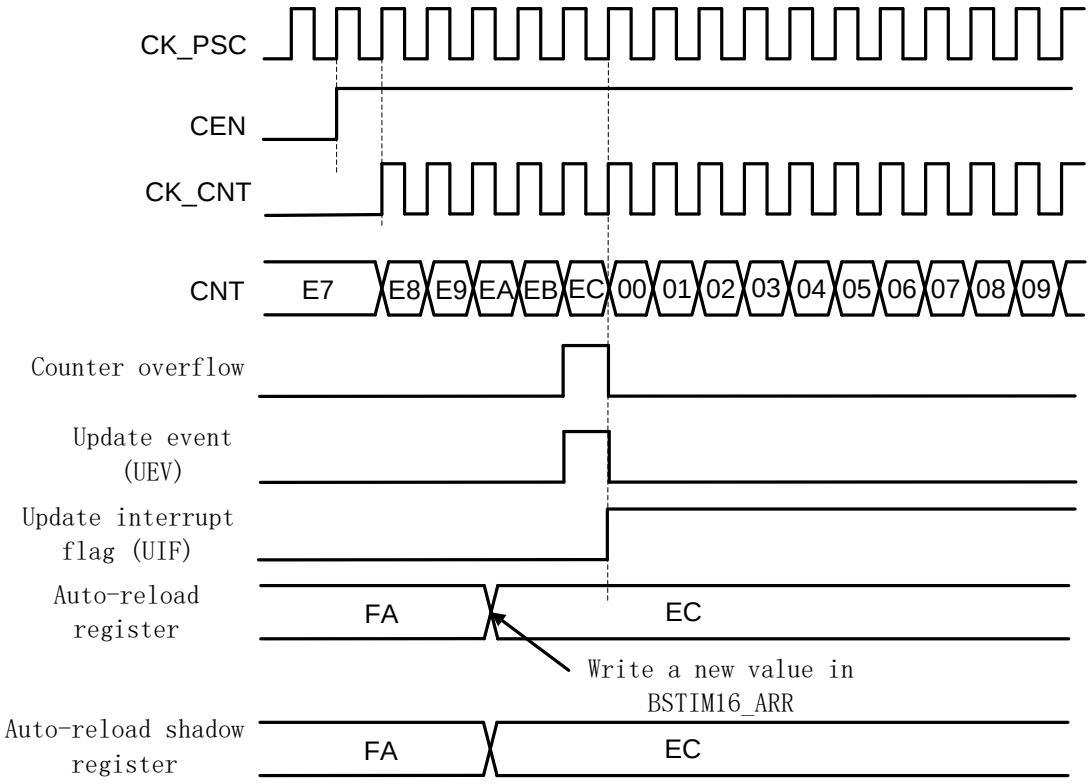


Figure 32–6 Counter Timing Diagram, Update Event When ARPE=0 (ARR is not Preloaded)

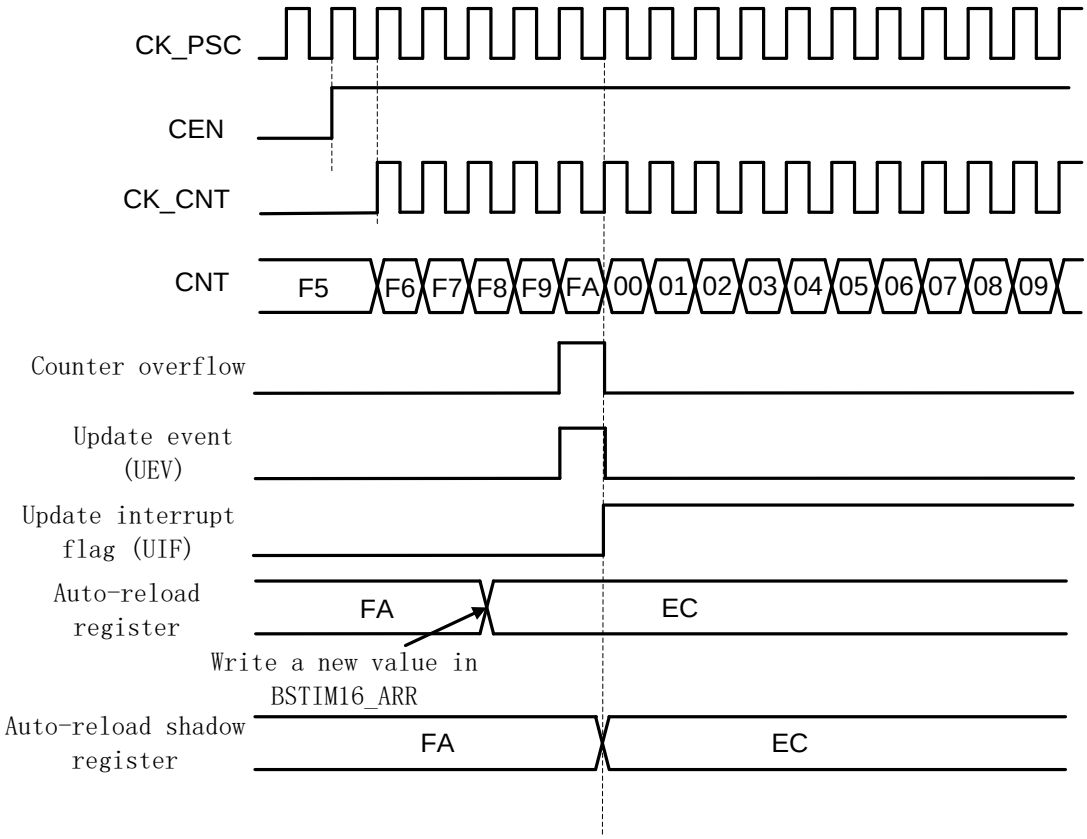


Figure 32–7 Counter Timing Diagram, Update Event When ARPE=1 (ARR is Preloaded)

32.4.3 Clock Source

The BSTIM clock is provided by the Internal clock source, and the register bits such as CEN and UG are software controlled.

After the software operates the UG register, the counter value will be reinitialized after the update signal is synchronized by CLK_PSC.

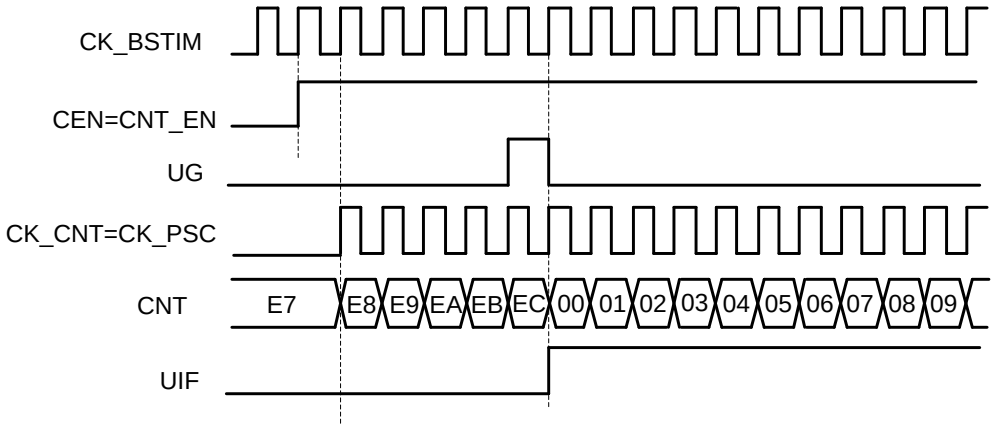


Figure 32–8 Internal Clock Divided by 1

32.4.4 Debug Mode

When the Cortex-M0 enters the debug mode, the timer can stop or continue to work, and its behavior is defined by the DBG_BT16_STOP register of the DBG module.

32.5 Register

Offset	Name	Symbol
BSTIM16(Base address: 0x40018C00)		
0x00000000	BSTIM16 Control Register1	BSTIM16_CR1
0x00000004	BSTIM16 Control Register2	BSTIM16_CR2
0x0000000C	BSTIM16 Interrupt Enable Register	BSTIM16_IER
0x00000010	BSTIM16 Interrupt Status Register	BSTIM16_ISR
0x00000014	BSTIM16 Event Generation Register	BSTIM16_EGR
0x00000024	BSTIM16 Counter Register	BSTIM16_CNT
0x00000028	BSTIM16 Prescaler Register	BSTIM16_PSC
0x0000002C	BSTIM16 Auto-Reload Register	BSTIM16_ARR

32.5.1 BSTIM16 Control Register1 (BSTIM16_CR1)

NAME	BSTIM16_CR1							
Offset	0x00000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ARPE	-			OPM	URS	UDIS	CEN
access	R/W-0	U-0			R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:8	-	RFU: **Reserved, read as 0**
7	ARPE	Auto-Reload Preload Enable 0: ARR register is not enabled for preload 1: ARR register is enabled for preload
6:4	-	RFU: **Reserved, read as 0**
3	OPM	One Pulse Mode 0: Counter is not stopped at update event 1: Counter stops counting at the next update event (clearing the CEN bit).
2	URS	Update Request Select 0: Any of the following events generates an update interrupt: - Counter overflow/underflow

bit	name	functional description
		- Setting the UG bit 1: Only counter overflow/underflow generates an update interrupt or DMA request if enabled.
1	UDIS	Update Disable 0: UEV enabled. The Update (UEV) event is generated by one of the following events: - Counter overflow/underflow - Setting the UG bit 1: UEV disabled. The Update event is not generated, shadow registers keep their value(ARR, PSC). However the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.
0	CEN	Counter Enable 0: Counter disabled 1: Counter enabled **Note:** Trigger mode can set the CEN bit automatically by hardware.

32.5.2 BSTIM16 Control Register2 (BSTIM16_CR2)

NAME	BSTIM16_CR2							
Offset	0x00000004							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	MMS			-			
access	U-0	R/W-000			U-0			

bit	name	functional description
31:7	-	RFU: **Reserved, read as 0**
6:4	MMS	Master mode selection, these bits are used to select the information to be sent in master mode to slave timers for synchronization (TRGO). The combination is as follows: 000: the UG bit from the BSTIM_EGR register is used as a trigger output (TRGO).

bit	name	functional description
		001: The Counter enable signal, CNT_EN, is used as a trigger output (TRGO). 010: The update event is selected as a trigger output (TRGO). 011/100/111: RFU **Note:** The clock of the slave timer and ADC must be enabled prior to receiving events from the master timer.
3:0	-	RFU: **Reserved, read as 0**

32.5.3 BSTIM16 Interrupt Enable Register (BSTIM16_IER)

NAME	BSTIM16_IER							
Offset	0x0000000C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							UIE
access	U-0							R/W-0

bit	name	functional description
31:1	**-**	RFU: **Reserved, read as 0**
0	**UIE**	Update event Interrupt Enable 0: Update interrupt disabled. 1: Update interrupt disabled.

32.5.4 BSTIM16 Interrupt Status Register (BSTIM16_ISR)

NAME	BSTIM16_ISR							
Offset	0x00000010							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							

access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							UIF
access	U-0							R/W-0

bit	name	functional description
31:1	**-**	RFU: **Reserved, read as 0**
0	**UIF**	Update event Interrupt Flag, write 1 to clear flag. The UIF is set and the shadow register is updated when the following events occur -At overflow or underflow and if repetition counter= 0 and UDIS = 0. -When CNT is reinitialized by software using the UG bit, if URS = 0and UDIS = 0. -The event initialization counter is triggered, if URS = 0and UDIS = 0.

32.5.5 BSTIM16 Event Generation Register (BSTIM16_EGR)

NAME	**BSTIM16_EGR**							
Offset	0x00000014							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							UG
access	U-0							W-0

bit	name	functional description
31:1	**-**	RFU: **Reserved, read as 0**
0	**UG**	User Generate, this bit can be set by software, it is automatically cleared by hardware. When the software sets UG, the counter is reinitialized and the shadow register is updated, and the prescaler counter is

		cleared. ***Note:*** *Since the BSTIM counter operating clock and the system bus clock APBCLK are independent of each other, when the software sets the UG register, the UIF is not set immediately, but needs to be synchronized by CK_BSTIM before it is set, and the delay between the two is related to the frequency and phase relationship between APBCLK and CK_BSTIM, which is not a definite value.*

32.5.6 BSTIM16 Counter Register (BSTIM16_CNT)

NAME	BSTIM16_CNT							
Offset	0x00000024							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CNT[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CNT[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	**-**	RFU: **Reserved, read as 0**
15:0	**CNT**	Counter value.

32.5.7 BSTIM16 Prescaler Register (BSTIM16_PSC)

NAME	BSTIM16_PSC							
Offset	0x00000028							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PSC[15:8]							
access	R/W-0000 0000							

bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PSC[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**PSC**	Prescaler value, the counter clock frequency CK_CNT is equal to $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[15:0]+1)$ This is a preload register whose contents are loaded into the shadow register when the update event occurs.

32.5.8 BSTIM16 Auto-Reload Register (BSTIM16_ARR)

NAME	BSTIM16_ARR							
Offset	0x0000002C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ARR[15:8]							
access	R/W-1111 1111							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ARR[7:0]							
access	R/W-1111 1111							

bit	name	functional description
31:16	-	RFU: **Reserved, read as 0**
15:0	**ARR**	Auto-Reload Register This is a preload register whose contents are loaded into the shadow register when the update event occurs.

33 32bits Low Power Timer (LPTIM32)

33.1 Introduction

LPTIM32 is a 32bits low power timer/counter module. By selecting the appropriate operating clock, LPTIM32 keeps running in various low-power modes and consumes very low power. LPTIM32 can even operate without an internal clock. Therefore, the function of external pulse counting in sleep mode can be realized. In addition, LPTIM32 in combination with an external input trigger signal, a low-power timeout wake-up function can be implemented.

The main features of LPTIM32:

- 1 independent 32-bit upcounter
- 3bit asynchronous clock prescaler with 8 dividing factors(1, 2, 4, 8, 16, 32, 64, 128)
- Optional operating clock:
  - Internal clock source: LSCLK,RCLP, APBCLK, RCLF_PSC
  - External clock source: LPT32_ETR(With analog filtering)
- Four-channel 32bit capture/compare register
- 32bit Auto Reload Register
- Input polarity selection
- Clockless external pulse counting
- Externally triggered wakeup from sleep timeout
- 32bit PWM output
- 32bit input signal capture
- Trigger signal output

33.2 Block Diagram

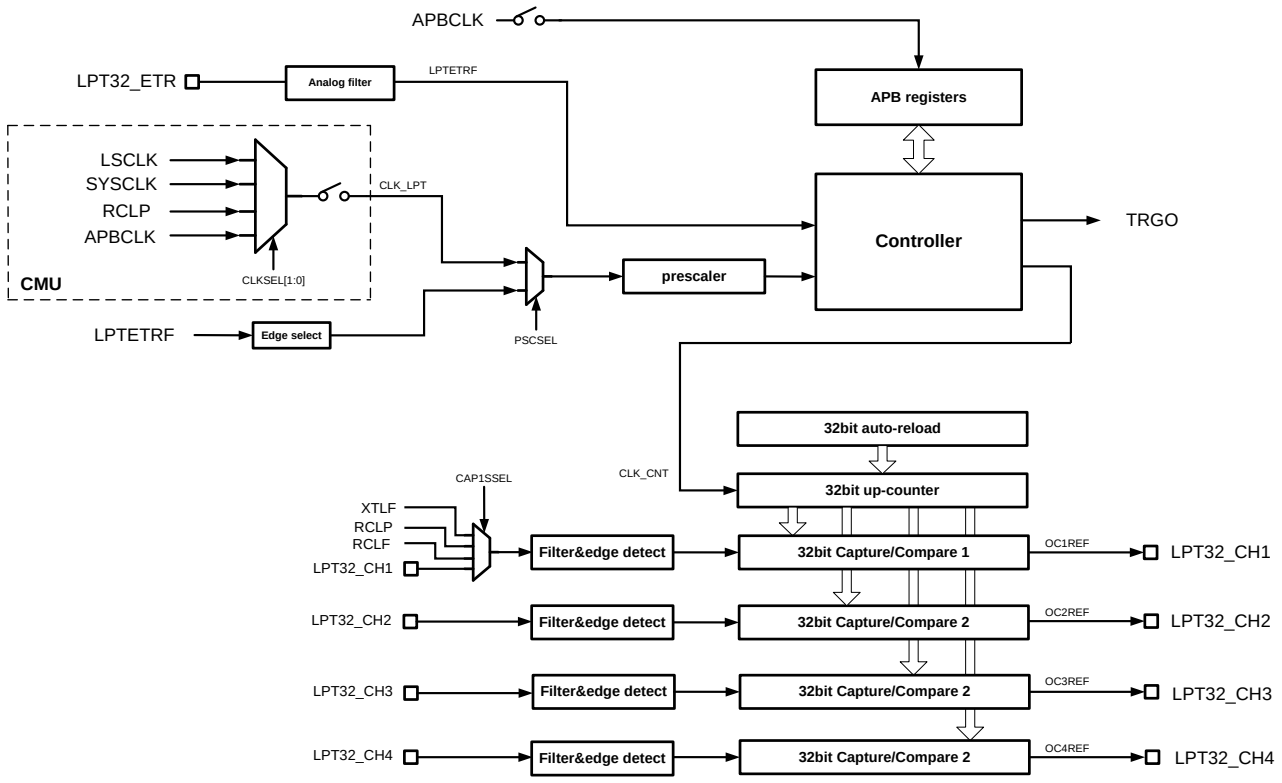


Figure 33–1 LPTIM32 Block Diagram

33.3 Clock and Reset

The control and status registers of LPTIM32 are located on the APB bus, so the peripheral bus clock must be enabled before the software can access the registers. For details, refer toChapter14 Clock Management Unit (CMU).

The working clock of the LPTIM32 timer is independent of the system bus clock and can be selected from multiple independent clock sources. The count clock source of LPTIM32 can be selected by configuring the CMU register. In order to ensure the stable and reliable operation of the timer, it is forbidden to modify the counting clock when EN is 1.

After selecting a suitable counting clock, it can also be prescaled through the DIVSEL register to obtain a lower operating clock frequency. Similarly, DIVSEL must be modified when EN is 0.

The LPTIM32 module can reset and cancel the reset by operating the LPT32RST register. For details, refer to the RMU chapter.

33.4 Pin Definition

Function	Pin mapping LQFP80
LPT32_ETR	PA10
LPT32_CH1	PA8
LPT32_CH2	PA9
LPT32_CH3	PC15
LPT32_CH4	PE5

Table 33-1 LPTIM32 Pin Mapping

33.5 Pin Definition

LPTIM32 supports 4 timer operating modes: general timer, external pulse triggered counting, external asynchronous pulse counting, and Timeout mode.

33.5.1 General Timer

When LPTIM32_CFGR.TMODE=00, LPTIM32 is general timer operation mode.

- CLK_LPT clock counting after using multiplexed selection
- Configure the OPCCR1.LPT32CKS register in the CMU module to select the appropriate count clock
- There is a synchronization process of two count clocks after the LPTIM32_CR.EN enable is set
- When enabled, the timer starts counting up until the count value is equal to ARR

Single count and continuous count

LPTIM32 has two counting modes - single count and continuous count.

Continuous counting mode: The counter starts and stays running until it is turned off. The counter reaches the target value (ARR) and then returns to 0 to restart counting and generates an overflow interrupt OVIF.

Single counting mode: When the counter is triggered, it counts to the target value (ARR) and then returns to 0 and stops automatically, generating an overflow interrupt OVIF while the hardware automatically clears the LPTIM32_CR.EN.

Note: *Since the count clock used by LPTIM32 is asynchronous to APBCLK, when the CPU clears the OVIF register, the clearing action is synchronized to the LPTIM32 count clock, requiring 2 cycles. When ARR is configured as 0 or 1, the synchronization process will cause 1 OVIF event to be lost.*

Therefore, it is not recommended to set ARR to 0 or 1.

33.5.2 External Pulse Trigger Counting

In external pulse trigger counting mode (LPTIM32_CFGR.TMODE=01), LPTIM32 uses the signal input from LPT32_ETR pin as the trigger signal. LPT32_ETR signal is first sampled and synchronized by LPTIM32 operating clock, and then can trigger the timer increment on its rising edge, falling edge or rising falling edge. Since it is necessary to use CLK_LPT to sample and identify the changing edge of the LPT32_ETR signal, it is required that the effective level width of the ETR input signal must be greater than two times the CLK_LPT period. The software can set which edge of LPT32_ETR is counted by LPTIM32 through LPTIM32_CFG.TRIGCFG register.

The following figure shows an example of LPT32_ETR triggered counting on rising edge.

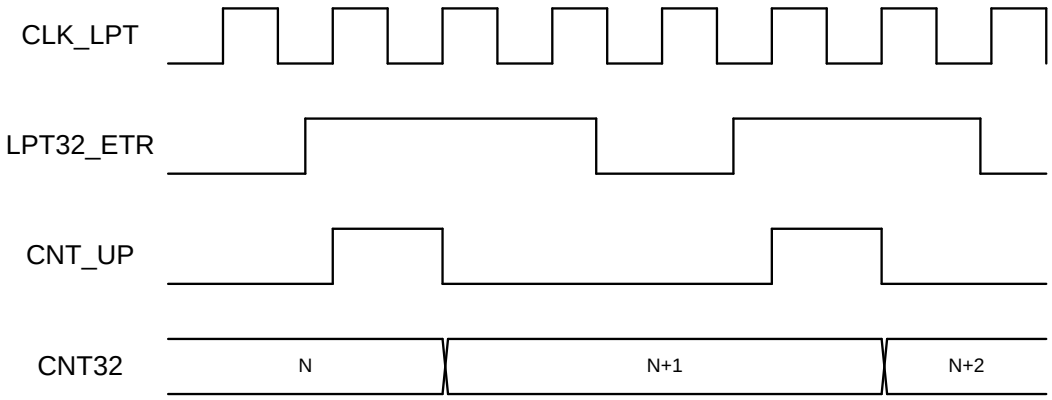


Figure 33–2 External ETR pulse rising edge triggers counting

33.5.3 External Asynchronous Pulse Counting

In external asynchronous pulse counting mode (LPTIM32_CFG.TMODE=10), the LPTIM32 uses the signal input from the LPT32_ETR pin directly as the counting clock. In this case, the LPTIM32 works fully asynchronously and does not need to enable any internal clock. The software can select whether the timer uses ETR rising or falling edge counting via LPTIM32_CFG.EDGESEL. Since any disturbing signal on the LPT32_ETR pin in this mode may cause the timer to malfunction, it is recommended to enable the ETR input analog filtering function, which is able to filter out glitch signals within about 100ns.

The following figure shows an example of external asynchronous pulse counting on falling edge.

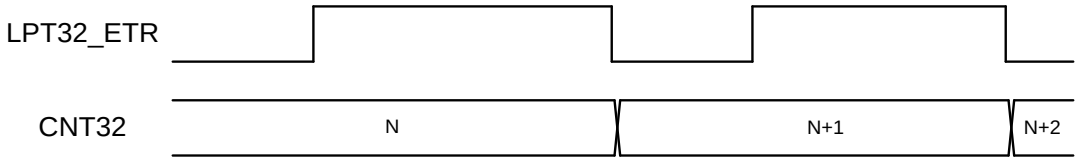


Figure 33–3 External ETR pulse asynchronous counting (Falling edge)

33.5.4 Timeout Mode

In Timeout mode (LPTIM32_CFG.TMODE=11), the LPTIM32 uses the signal input from the LPT32_ETR pin as the trigger signal and the timer works with the internal clock CLK_LPT. After the timer starts in Timeout mode, it does not start counting immediately, but waits for the first valid edge of the LPT32_ETR signal to arrive. When the first valid edge arrives, the timer is triggered to start free counting, and thereafter each new valid edge of ETR clears the counter and starts counting again. According to the actual frequency of the external input ETR signal, the counter operating clock and overflow limit (ARR) shall be reasonably configured to keep the timer from overflowing. If the timer overflows, it means no expected ETR event arrives within the specified time interval, then the timer generates an overflow interrupt, the count value returns to 0, and the LPTIM_CR.EN is automatically cleared to end the counting process.

After the LPT32_ETR signal is sampled and synchronized by the LPTIM32 operating clock, it can trigger the counter to clear and restart at its rising edge, falling edge or both edges. Since it is necessary to sample and identify the changing edge of the LPT32_ETR signal using CLK_LPT, it is required that the effective level width of the ETR input signal must be greater than two times the CLK_LPT period. The software can set which edge of LPT32_ETR is counted by LPTIM32 through LPTIM_CFG.TRIGCFG register.

The following figure is an example of using LPT32_ETR rising edge clearing in timeout mode and eventually overflowing.

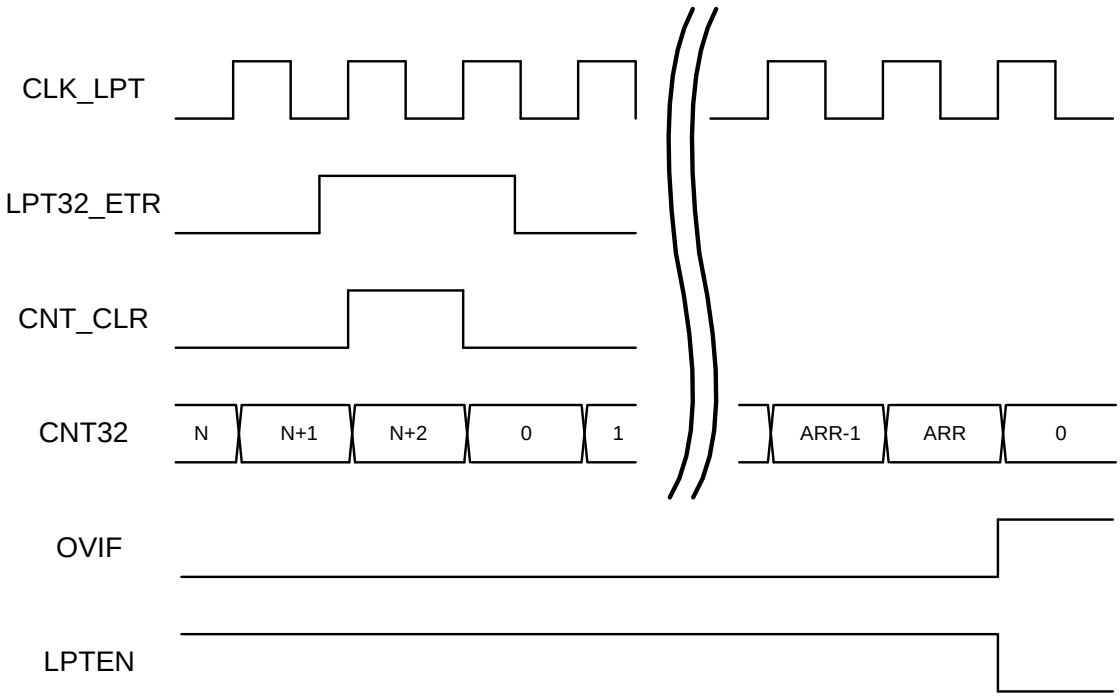


Figure 33–4 Timeout Mode

Using Timeout mode and enabling the LPTIM32 interrupt, the timeout wakeup function triggered by external signals can be realized when the chip is in sleep mode. At this time, as long as there is a periodic signal input on the LPT32_ETR pin, it can keep the chip in sleep. If no trigger signal arrives within the specified time, the LPTIM32 timeout overflow interrupt will wake up the chip.

33.6 Capture/Compare Function

LPTIM32 comes with four independent 32bit capture/compare channels, with 32bit timer as time base, combined with CCRx register,LPTIM32 supports four channels of 32bit PWM output, or 32bit input capture function.

33.6.1 32bit PWM

Both independent capture/compare channels of LPTIM32 can output 32bit PWM waveforms. The PWM function needs to configure the capture/compare channel as a compare output.

After PWM function is enabled, LPTIM32 starts counting from 0x0000_0000. Take the positive polarity waveform as an example, when the count value is equal to the comparison value (CCRx), the output is set high, when the count value is equal to the target value (ARR), the output is low. The PWM period is determined by the ARR register and the duty cycle is determined by the CCRx register. The LPTIM32_CCSR.POLAR register can configure the polarity of the output waveform.

To implement PWM output function, LPTIM32_CCSR.CCxS needs to be configured to 10, at which time LPT_CHx becomes the output channel and the corresponding GPIO automatically enables the output function (the software needs to configure the GPIO as a digital peripheral function).

The following figure shows an example of PWM output with POLAR=1.

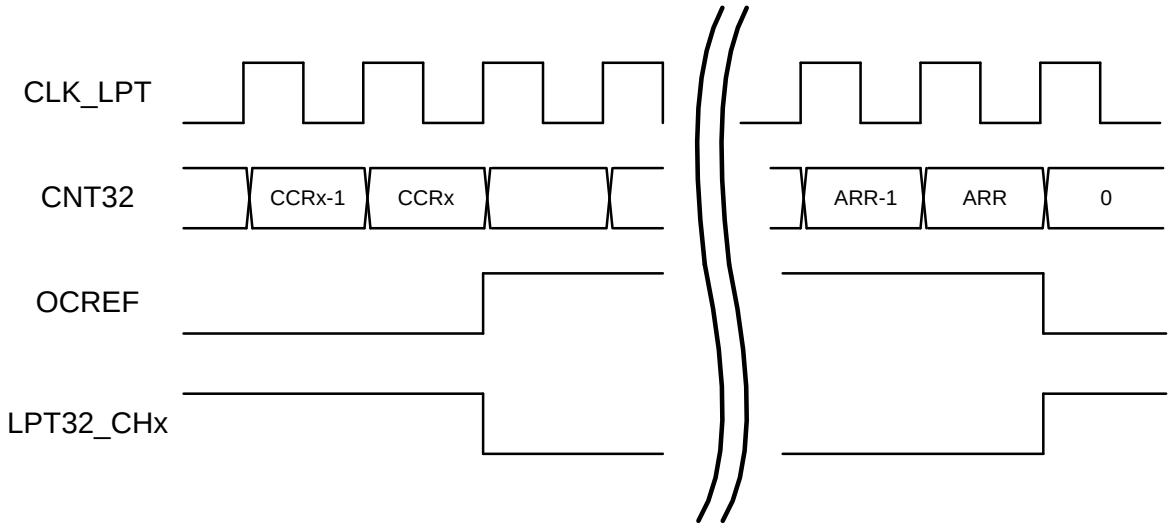


Figure 33–5 PWM Output

33.6.2 Input Capture

The LPTIM32's four capture/compare channels enable two independent input signal period or level width capture functions.

The input capture can be configured to capture on the rising edge, falling edge, or both edge of the input signal. Each time a capture occurs, the CAPxEDGE register indicates whether the current capture is a rising or falling edge.

Channel 1 of LPTIM32 can capture the external pin input or chip internal clock signal (XTLF, RCLP, RCLF). The period capture of internal clock signal can be used for clock frequency correction with software; while Channel 2, 3 and 4 can only capture the external pin input signal.

After enabling the input mode, the 32bit counter is free to count as a time base. When a valid edge of the captured signal arrives, the current count value is latched into the CCRx register and a capture interrupt is generated; the capture interrupt flag is automatically cleared by hardware when the CCRx register is read by software, the capture interrupt flag can also be cleared by software by writing 1 in addition. When the capture interrupt flag is not cleared and a new capture event arrives, the capture conflict interrupt flag (CAPxOVR) will be set.

The following figure shows an example of capturing the rising and falling edges of the input signal.

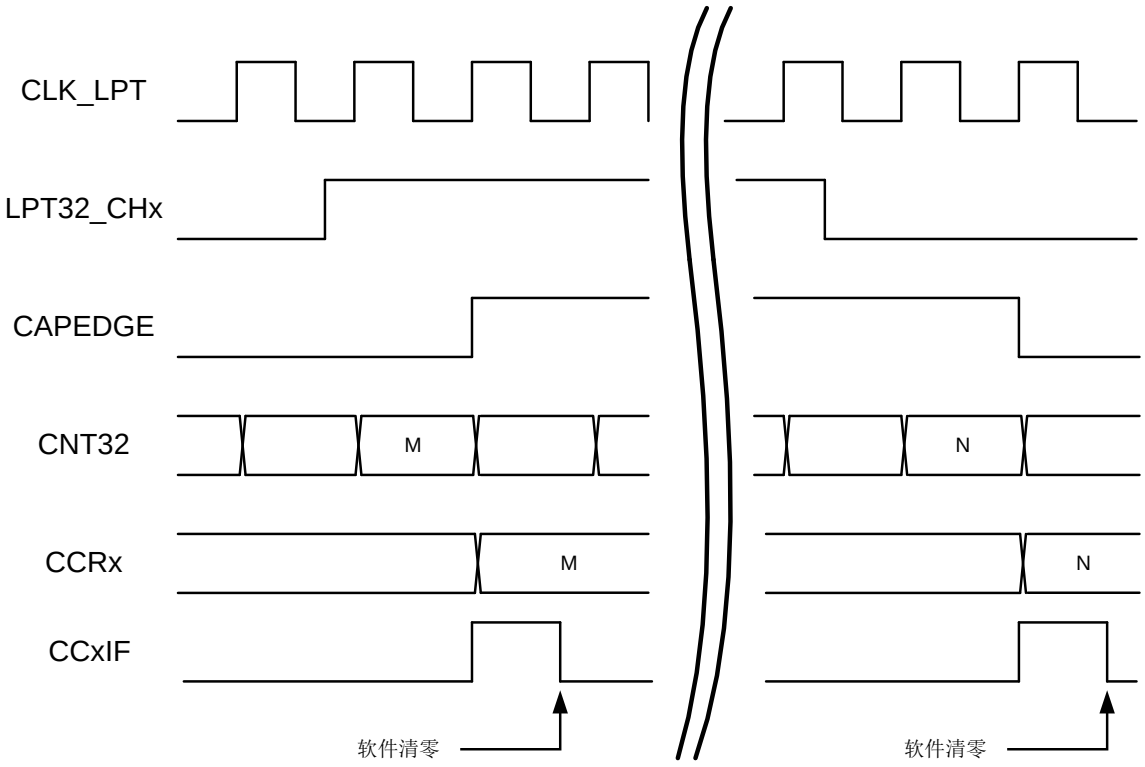


Figure 33–6 Input Signal Edge Capture

33.7 Trigger Signal Output

LPTIM32 can output trigger signals to other peripherals under certain conditions. Trigger signal sources include:

- LPTIM32 enable
- Update event: Counter overflow, count value equal to CCR1 or CCR2 or CCR3 or CCR4
- CC1 channel comparison pulse: the count value is equal to CCR1
- CC1 channel capture event
- CC2 channel capture event
- CC3 channel capture event
- CC4 channel capture event

The trigger signal is synchronized with APBCLK, so before using this function, you must turn on the LPTIM32 bus clock, that is, set the LPT32_PCE register.

33.8 Register

Offset	Name	Symbol
LPTIM32(Base Address: 0x40013400)		
0x00000000	LPTIM32 Config Register	LPTIM32_CFGR
0x00000004	LPTIM32 Counter Register	LPTIM32_CNT
0x00000008	LPTIM32 Capture/Compare Control and Status Register	LPTIM32_CCSR
0x0000000C	LPTIM32 Auto-Reload Register	LPTIM32_ARR
0x00000010	LPTIM32 Interrupt Enable Register	LPTIM32_IER
0x00000014	LPTIM32 Interrupt Status Register	LPTIM32_ISR
0x00000018	LPTIM32 Control Register	LPTIM32_CR
0x00000020	LPTIM32 Capture/Compare Register1	LPTIM32_CCR1
0x00000024	LPTIM32 Capture/Compare Register2	LPTIM32_CCR2
0x00000028	LPTIM32 Capture/Compare Register3	LPTIM32_CCR3
0x0000002C	LPTIM32 Capture/Compare Register4	LPTIM32_CCR4

33.8.1 LPTIM32 Config Register (LPTIM32_CFGR)

NAME	LPTIM32_CFGR							
Offset	0x00000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							ETR_AFEN
access	U-0							R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-					MMS		
access	U-0					R/W-000		
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-	PSCSEL	-	DIVSEL			-	
access	U-0	R/W-0	U-0	R/W-000			U-0	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	EDGESEL	TRIGCFG		-		ONST	TMOD	
access	R/W-0	R/W-00		U-0		R/W-0	R/W-00	

bit	name	functional description
31:25	**--**	RFU: **Reserved, read as 0**
24	**ETR_AFEN**	External Trigger input Analog Filter Enable 0: Disable analog filtering 1: Enable analog filtering, the filter width is about 100ns
23:19	**--**	RFU: **Reserved, read as 0**

bit	name	functional description
18:16	**MMS**	Master Mode Select, used to configure the source of the synchronous trigger signal (TRGO) sent to the slave in the master mode 000: RFU 001: Counter enable signal EN is used as TRGO 010: UE (update event) signal is used as TRGO 011: CC1 compare pulse, if the CC1IF flag is about to be set, TRGO outputs a positive pulse 100: CC1 capture event used as TRGO 101: CC2 capture event used as TRGO 110: CC3 capture event used as TRGO 111: CC4 capture event used as TRGO Note: The slave must enable the working clock in advance to receive the TRGO sent by the master timer
15	**--**	RFU: **Reserved, read as 0**
14	**PSCSEL**	Prescaler Input Select 0: CLKSEL 1: LPTETR
13	**--**	RFU: **Reserved, read as 0**
12:10	**DIVSEL**	Counter Clock Divider Select 000: Divided-by-1 001: Divided-by-2 010: Divided-by-4 011: Divided-by-8 100: Divided-by-16 101: Divided-by-32 110: Divided-by-64 111: Divided-by-128
9:8	**--**	RFU: **Reserved, read as 0**
7	**EDGESEL**	ETR Clock Edge Select 0: LPT_ETR rising edge count 1: LPT_ETR falling edge count
6:5	**TRIGCFG**	ETR Trigger Configuration (Need to use the internal clock to synchronously sample LPT_ETR) 00: Trigger on rising edge of LPT_ETR input signal 01: Trigger on falling edge of LPT_ETR input signal 10/11: External input signal rising and falling edge trigger
4:3	**--**	RFU: **Reserved, read as 0**

bit	name	functional description
2	**ONST**	One State Timer 0: Continuous counting mode: The counter keeps running after being triggered until it is turned off. After the counter reaches the target value, it returns to 0 and restarts counting, and an overflow interrupt is generated. 1: Single counting mode: After the counter is triggered, it counts to the target value and then returns to 0, and stops automatically, generating an overflow interrupt.
1:0	**TMODE**	Timer Operation Mode 00: General timer mode 01: External pulse trigger counting mode 10: External asynchronous pulse counting mode 11: Timeout mode

33.8.2 LPTIM32 Counter Register (LPTIM32_CNT)

NAME	**LPTIM32_CNT**							
Offset	0x00000004							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	CNT32[31:24]							
access	R-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	CNT32[23:16]							
access	R-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CNT32[15:8]							
access	R-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CNT32[7:0]							
access	R-0000 0000							

bit	name	functional description
31:0	**CNT32**	32bit counter (Counter 32bits-wide)

33.8.3 LPTIM32 Capture/Compare Control and Status Register (LPTIM32_CCSR)

NAME	**LPTIM32_CCSR**							
Offset	0x00000008							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-						CAP1SSEL	
access	U-0						R/W-00	

bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	CAP4EDGE	CAP3EDGE	CAP2EDGE	CAP1EDGE	POLAR4	POLAR3	POLAR2	POLAR1
access	R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CAPCFG4		CAPCFG3		CAPCFG2		CAPCFG1	
access	R/W-00		R/W-00		R/W-00		R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CC4S		CC3S		CC2S		CC1S	
access	R/W-00		R/W-00		R/W-00		R/W-00	

bit	name	functional description
31:26	--	RFU: **Reserved, read as 0**
25:24	**CAP1SSEL**	Capture Channel 1 Source Select, only valid when CH1 channel is configured as input capture 00: LPT32_CH1 input 01: XTLF 10: RCLP 11: RCLF
23	**CAP4EDGE**	Channel 4 Captured Edge, update when CC4IF is set 0: Rising edge 1: Falling edge
22	**CAP3EDGE**	Channel 3 Captured Edge, update when CC3IF is set 0: Rising edge 1: Falling edge
21	**CAP2EDGE**	Channel 2 Captured Edge, update when CC2IF is set 0: Rising edge 1: Falling edge
20	**CAP1EDGE**	Channel 1 Captured Edge, update when CC1IF is set 0: Rising edge 1: Falling edge
19	**POLAR4**	Channel 4 Compare Output Polarity 0: Positive polarity waveform, starts at low, set high when the count value == compare value, restored to low when the count value == ARR 1: Negative polarity waveform, positive polarity waveform is reversed
18	**POLAR3**	Channel 3 Compare Output Polarity 0: Positive polarity waveform, starts at low, set high when the count value == compare value, restored to low when the count value == ARR 1: Negative polarity waveform, positive polarity waveform is reversed

bit	name	functional description
17	**POLAR2**	Channel 2 Compare Output Polarity 0: Positive polarity waveform, starts at low, set high when the count value == compare value, restored to low when the count value == ARR 1: Negative polarity waveform, positive polarity waveform is reversed
16	**POLAR1**	Channel 1 Compare Output Polarity 0: Positive polarity waveform, starts at low, set high when the count value == compare value, restored to low when the count value == ARR 1: Negative polarity waveform, positive polarity waveform is reversed
15:14	**CAPCFG4**	Channel 4 Capture Edge Config 00: Rising edge capture 01: Falling edge capture 10: Rising and falling edge capture 11: RFU
13:12	**CAPCFG3**	Channel 3 Capture Edge Config 00: Rising edge capture 01: Falling edge capture 10: Rising and falling edge capture 11: RFU
11:10	**CAPCFG2**	Channel 2 Capture Edge Config 00: Rising edge capture 01: Falling edge capture 10: Rising and falling edge capture 11: RFU
9:8	**CAPCFG1**	Channel 1 Capture Edge Config 00: Rising edge capture 01: Falling edge capture 10: Rising and falling edge capture 11: RFU
7:6	**CC4S**	Channel 4 Capture/Compare Select 00,11: Disable channel 4 capture/compare function 01: Enable channel 4 capture function (LPT32_CH4 is input) 10: Enable channel 4 compare function (LPT32_CH4 is output)
5:4	**CC3S**	Channel 3 Capture/Compare Select 00,11: Disable channel 3 capture/compare function 01: Enable channel 3 capture function (LPT32_CH3 is input) 10: Enable channel 3 compare function (LPT32_CH3 is output)

bit	name	functional description
3:2	**CC2S**	Channel 2 Capture/Compare Select 00,11: Disable channel 2 capture/compare function 01: Enable channel 2 capture function (LPT32_CH2 is input) 10: Enable channel 2 compare function (LPT32_CH2 is output)
1:0	**CC1S**	Channel 1 Capture/Compare Select 00,11: Disable channel 1 capture/compare function 01: Enable channel 1 capture function (LPT32_CH1 is input) 10: Enable channel 1 compare function (LPT32_CH1 is output)

33.8.4 LPTIM32 Auto-Reload Register (LPTIM32_ARR)

NAME	LPTIM32_ARR							
Offset	0x0000000C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	ARR[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	ARR[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ARR[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ARR[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:0	**ARR**	Auto-Reload Register When the counter count is equal to ARR, the counter returns to its initial value and starts counting up again

33.8.5 LPTIM32 Interrupt Enable Register (LPTIM32_IER)

NAME	LPTIM_IER							
Offset	0x00000010							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							

bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				OVR4IE	OVR3IE	OVR2IE	OVR1IE
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TRIGIE	OVIE	-		CC4IE	CC3IE	CC2IE	CC1IE
access	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:12	--	RFU: **Reserved, read as 0**
11	**OVR4IE**	Channel 4 Over-Capture Interrupt Enable 1: Enable interrupt 0: Disable interrupt
10	**OVR3IE**	Channel 3 Over-Capture Interrupt Enable 1: Enable interrupt 0: Disable interrupt
9	**OVR2IE**	Channel 2 Over-Capture Interrupt Enable 1: Enable interrupt 0: Disable interrupt
8	**OVR1IE**	Channel 1 Over-Capture Interrupt Enable 1: Enable interrupt 0: Disable interrupt
7	**TRIGIE**	External Trigger Interrupt Enable 1: External trigger interrupt enable 0: External trigger interrupt disable
6	**OVIE**	Counter Over-Flow Interrupt Enable 1: Counter over-flow interrupt enable 0: Counter over-flow interrupt disable
5:4	--	RFU: **Reserved, read as 0**
3	**CC4IE**	Capture/Compare channel 4 Interrupt Enable 1: Capture/Compare channel 4 interrupt enable 0: Capture/Compare channel 4 interrupt disable
2	**CC3IE**	Capture/Compare channel 3 Interrupt Enable 1: Capture/Compare channel 3 interrupt enable 0: Capture/Compare channel 3 interrupt disable
1	**CC2IE**	Capture/Compare channel 2 Interrupt Enable 1: Capture/Compare channel 2 interrupt enable 0: Capture/Compare channel 2 interrupt disable
0	**CC1IE**	Capture/Compare channel 1 Interrupt Enable 1: Capture/Compare channel 1 interrupt enable 0: Capture/Compare channel 1 interrupt disable

33.8.6 LPTIM32 Interrupt Status Register (LPTIM32_ISR)

NAME	LPTIM32_ISR							
Offset	0x00000014							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				CAP4OVR	CAP3OVR	CAP2OVR	CAP1OVR
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TRIGIF	OVIF	-		CC4IF	CC3IF	CC2IF	CC1IF
access	R/W-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:12	--	RFU: **Reserved, read as 0**
11	**CAP4OVR**	Channel 4 Over-Capture Interrupt Flag, hardware set, write 1 to clear by software 1: In input capture mode, a new capture occurs when CC4IF is 1, and overrun occurs 0: No overrun occurs
10	**CAP3OVR**	Channel 3 Over-Capture Interrupt Flag, hardware set, write 1 to clear by software 1: In input capture mode, a new capture occurs when CC3IF is 1, and overrun occurs 0: No overrun occurs
9	**CAP2OVR**	Channel 2 Over-Capture Interrupt Flag, hardware set, write 1 to clear by software 1: In input capture mode, a new capture occurs when CC2IF is 1, and overrun occurs 0: No overrun occurs
8	**CAP1OVR**	Channel 1 Over-Capture Interrupt Flag, hardware set, write 1 to clear by software 1: In input capture mode, a new capture occurs when CC2IF is 1, and overrun occurs 0: No overrun occurs
7	**TRIGIF**	External Trigger Interrupt Flag, write 1 to clear 1: External trigger arrival interrupt generates 0: No interrupt generates

bit	name	functional description
6	**OVIF**	Counter Over-Flow Interrupt Flag, write 1 to clear 1: Counter over-flow interrupt generates 0: No interrupt generates
5:4	**--**	RFU: **Reserved, read as 0**
3	**CC4IF**	Capture/Compare Channel 4 Interrupt Flag, hardware set, write 1 to clear by software 1: Counter value and comparison value 4 match, or a capture event occurs 0: No interrupt generates
2	**CC3IF**	Capture/Compare Channel 3 Interrupt Flag, hardware set, write 1 to clear by software 1: Counter value and comparison value 3 match, or a capture event occurs 0: No interrupt generates
1	**CC2IF**	Capture/Compare Channel 2 Interrupt Flag, hardware set, write 1 to clear by software 1: Counter value and comparison value 2 match, or a capture event occurs 0: No interrupt generates ***Note:*** *In the capture mode, this flag is cleared by writing 1 or reading CCR2.*
0	**CC1IF**	Capture/Compare Channel 1 Interrupt Flag, hardware set, write 1 to clear by software 1: Counter value and comparison value 1 match, or a capture event occurs 0: No interrupt generates ***Note:*** *In the capture mode, this flag is cleared by writing 1 or reading CCR1.*

33.8.7 LPTIM32 Control Register (LPTIM32_CR)

NAME	LPTIM32_CR							
Offset	0x00000018							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							

bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							EN
access	U-0							R/W-0

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
0	**EN**	LPTIM Enable 1: Enable counter 0: Disable counter

33.8.8 LPTIM32 Capture/Compare Register1 (LPTIM32_CCR1)

NAME	LPTIM32_CCR1							
Offset	0x00000020							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	CCR1[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	CCR1[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR1[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR1[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:0	**CCR1**	Channel1 Capture/Compare Register

33.8.9 LPTIM32 Capture/Compare Register2 (LPTIM32_CCR2)

NAME	LPTIN32_CCR2							
Offset	0x00000024							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	CCR2[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	CCR2[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR2[15:8]							
access	R/W-0000 0000							

bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR2[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:0	**CCR2**	Channel2 Capture/Compare Register

33.8.10 LPTIM32 Capture/Compare Register3 (LPTIM32_CCR3)

NAME	LPTIM32_CCR3							
Offset	0x00000028							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	CCR3[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	CCR3[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR3[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR3[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:0	**CCR3**	Channel3 Capture/Compare Register

33.8.11 LPTIM32 Capture/Compare Register4 (LPTIM32_CCR4)

NAME	LPTIM32_CCR4							
Offset	0x0000002C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	CCR4[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	CCR4[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR4[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR4[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:0	**CCR4**	Channel4 Capture/Compare Register

34 16bits Low Power Timer (LPTIM16)

34.1 Introduction

LPTIM16 is a 16bits low power timer/counter module. By selecting the appropriate operating clock, LPTIM16 keeps running in various low-power modes and consumes very low power. LPTIM16 can even operate without an internal clock. Therefore, the function of external pulse counting in sleep mode can be realized. In addition, LPTIM16 in combination with an external input trigger signal, a low-power timeout wake-up function can be implemented.

The main features of LPTIM16:

- 1 independent 16-bit upcounter
- 3bit asynchronous clock prescaler with 8 dividing factors (1, 2, 4, 8, 16, 32, 64, 128)
- Optional operating clock:
  - Internal clock source: LSCLK, RCLP, APBCLK, RCLF
  - External clock source: LPT16_ETR (With analog filtering)
- Two-channel 16bit capture/compare register
- 16bit Auto Reload Register
- Input polarity selection
- Clockless external pulse counting
- Externally triggered wakeup from sleep timeout
- 16bit PWM output
- 16bit input signal capture
- Trigger signal output
- Two-channel quadrature encoder

34.2 Block Diagram

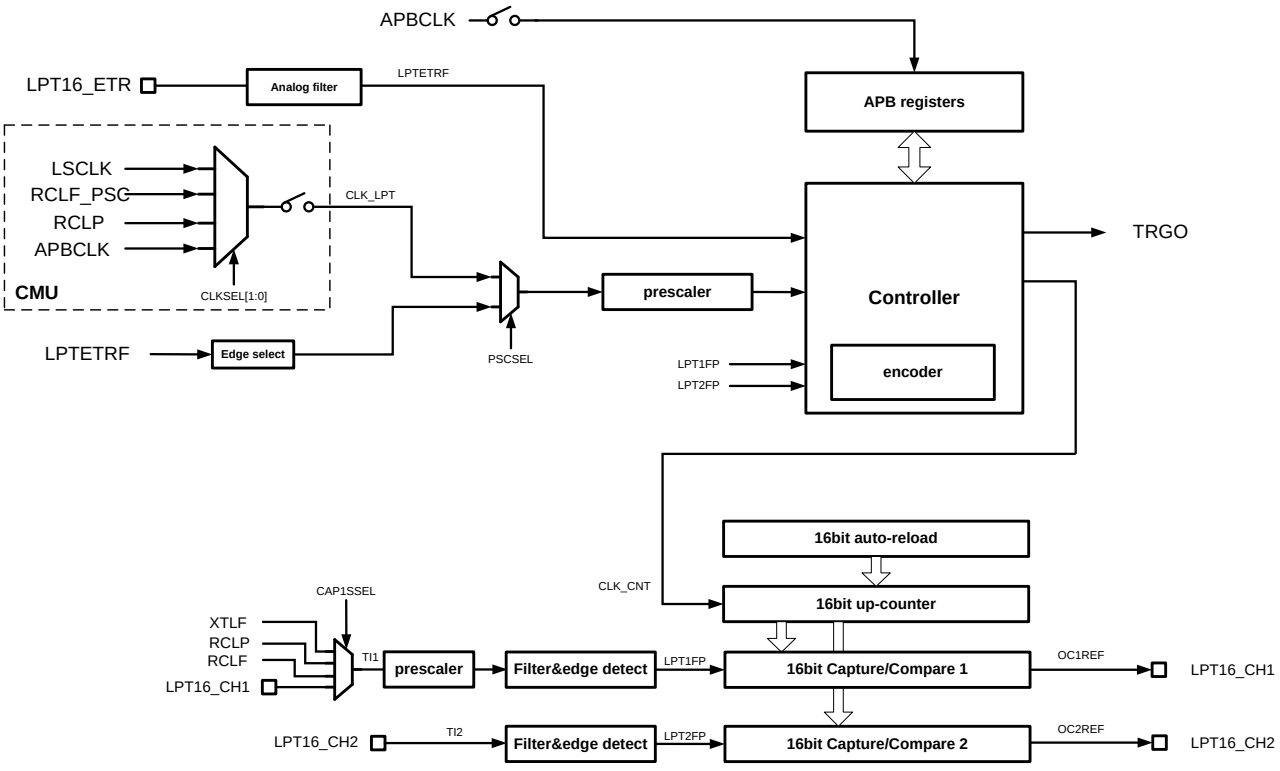


Figure 34–1 LPTIM16 Block Diagram

34.3 Clock and Reset

The control and status registers of LPTIM16 are located on the APB bus, so the peripheral bus clock must be enabled before the software can access the registers. For details, refer to Chapter14 Clock Management Unit (CMU).

The working clock of the LPTIM16 timer is independent of the system bus clock and can be selected from multiple independent clock sources. The count clock source of LPTIM16 can be selected by configuring the independent working clock of the peripherals in the CMU. In order to ensure the stable and reliable operation of the timer, it is forbidden to modify the counting clock when EN is 1.

After selecting a suitable counting clock, it can also be prescaled through the DIVSEL register to obtain a lower operating clock frequency. Similarly, DIVSEL must be modified when ENis 0.

The LPTIM16 module can reset and cancel the reset by operating the LPT16RST register. For details, refer to the RMU peripheral reset register.

34.4 Pin Definition

Function	Pin mapping LQFP80
LPT16_ETR	PD5
LPT16_CH1	PC13
LPT16_CH2	PC14

Table 34-1 LPTIM16 Pin Mapping

34.5 Timer Function

LPTIM16 supports 4 timer operating modes: general timer, external pulse triggered counting, external asynchronous pulse counting, and Timeout mode.

34.5.1 General Timer

When LPTIM16_CFGR.TMODE=00, LPTIM16 is general timer operation mode.

- CLK_LPT clock counting after using multiplexed selection
- Configure the OPCCR1.LPT16CKS register in the CMU module to select the appropriate count clock
- There is a synchronization process of two count clocks after the LPTIM16_CR.EN enable is set
- When enabled, the timer starts counting up until the count value is equal to ARR

Single count and continuous count

LPTIM16 has two counting modes - single count and continuous count.

Continuous counting mode: The counter starts and stays running until it is turned off. The counter reaches the target value (ARR) and then returns to 0 to restart counting and generates an overflow interrupt OVIF.

Single counting mode: When the counter is triggered, it counts to the target value (ARR) and then returns to 0 and stops automatically, generating an overflow interrupt OVIF while the hardware automatically clears the LPTIM16_CR.EN.

Note: *Since the count clock used by LPTIM16 is asynchronous to APBCLK, when the CPU clears the OVIF register, the clearing action is synchronized to the LPTIM16 count clock, requiring 2 cycles. When ARR is configured as 0 or 1, the synchronization process will cause 1 OVIF event to be lost. Therefore, it is not recommended to set ARR to 0 or 1.*

34.5.2 External Pulse Trigger Counting

In external pulse trigger counting mode (LPTIM16_CFGR.TMODE=01), LPTIM16uses the signal input from LPT16_ETR pin as the trigger signal. LPT16_ETR signal is first sampled and synchronized by LPTIM16 operating clock, and then can trigger the timer increment on its rising edge, falling edge or rising falling edge. Since it is necessary to use CLK_LPT to sample and identify the changing edge of the LPT16_ETR signal, it is required that the effective level width of the ETR input signal must be greater than two times the CLK_LPT period. The software can set which edge of LPT16_ETR is counted by LPTIM16 through LPTCFG.TRIGCFG register.

The following figure shows an example of LPT16_ETR triggered counting on rising edge.

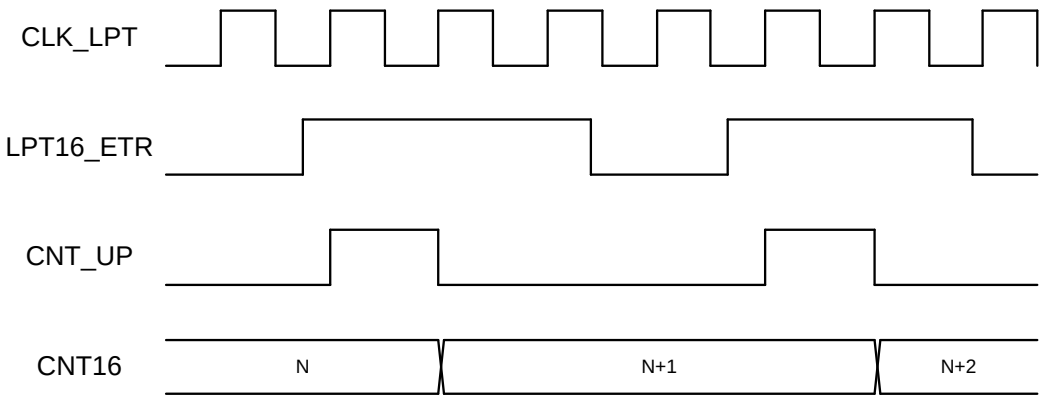


Figure 34–2 External ETR pulse rising edge triggers counting

34.5.3 External Asynchronous Pulse Counting

In external asynchronous pulse counting mode (LPTIM16_CFGR.TMODE=10), the LPTIM16 uses the signal input from the LPT16_ETR pin directly as the counting clock. In this case, the LPTIM16 works fully asynchronously and does not need to enable any internal clock. The software can select whether the timer uses ETR rising or falling edge counting via LPTIM16_CFGR.EDGESEL. Since any disturbing signal on the LPT16_ETR pin in this mode may cause the timer to malfunction, it is recommended to enable the ETR input analog filtering function, which is able to filter out glitch signals within about 100ns.

The following figure shows an example of external asynchronous pulse counting on falling edge.

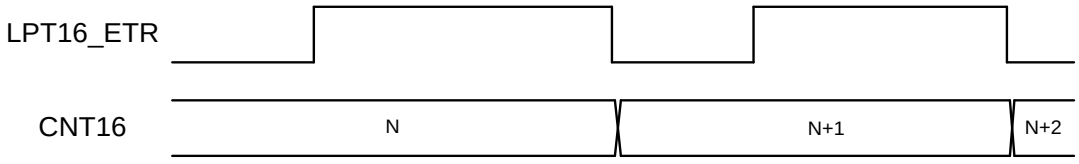


Figure 34–3 External ETR pulse asynchronous counting (Falling edge)

34.5.4 Timeout Mode

In Timeout mode (LPTIM16_CFGR.TMODE=11), the LPTIM16 uses the signal input from the LPT16_ETR pin as the trigger signal and the timer works with the internal clock CLK_LPT.After the timer starts in Timeout mode, it does not start counting immediately, but waits for the first valid edge of the LPT16_ETR signal to arrive. When the first valid edge arrives, the timer is triggered to start free counting, and thereafter each new valid edge of ETR clears the counter and starts counting again. According to the actual frequency of the external input ETR signal, the counter operating clock and overflow limit (ARR) shall be reasonably configured to keep the timer from overflowing. If the timer overflows, it means no expected ETR event arrives within the specified time interval, then the timer generates an overflow interrupt, the count value returns to 0, and the LPTIM16_CR.EN is automatically cleared to end the counting process.

After the LPT16_ETR signal is sampled and synchronized by the LPTIM16 operating clock, it can trigger the counter to clear and restart at its rising edge, falling edge or both edges. Since it is necessary to sample and identify the changing edge of the LPT16_ETR signal using CLK_LPT, it is required that the effective level width of the ETR input signal must be greater than two times the CLK_LPT period. The software can set which edge of LPT16_ETR is counted by LPTIM16 through LPTIM16_CFGR.TRIGCFG register.

The following figure is an example of using LPT16_ETR rising edge clearing in timeout mode and eventually overflowing.

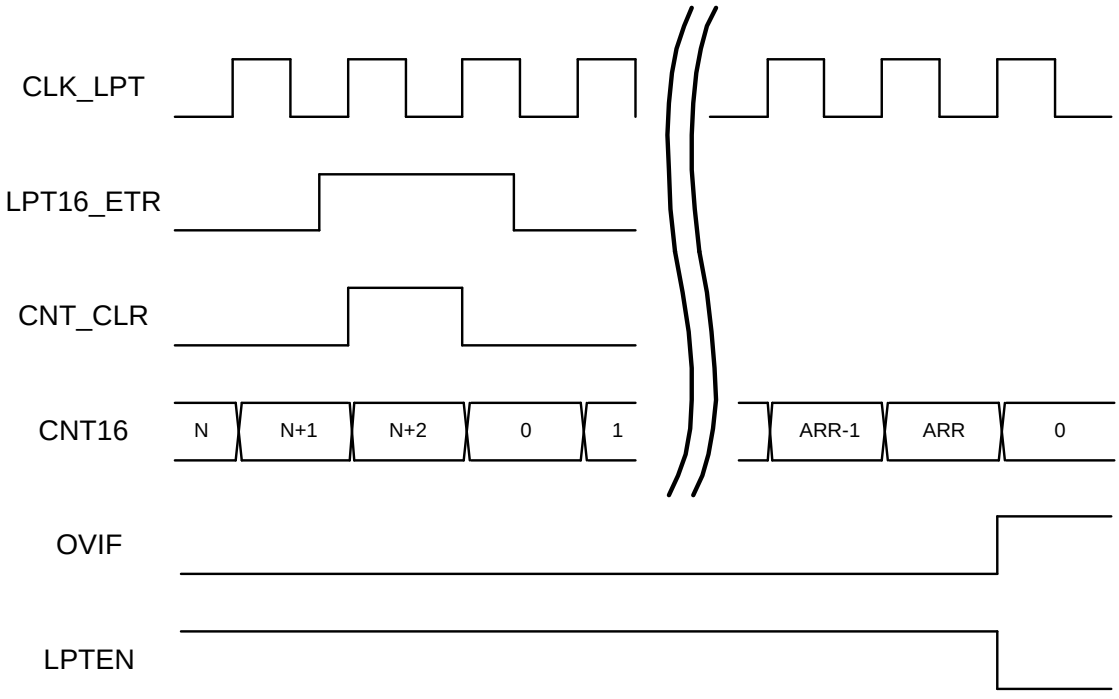


Figure 34–4 Timeout Mode

Using Timeout mode and enabling the LPTIM16 interrupt, the timeout wakeup function triggered by

external signals can be realized when the chip is in sleep mode. At this time, as long as there is a periodic signal input on the LPT16_ETR pin, it can keep the chip in sleep. If no trigger signal arrives within the specified time, the LPTIM16 timeout overflow interrupt will wake up the chip.

34.6 Capture/Compare Function

LPTIM16 comes with two independent 16bit capture/compare channels, with 16bit timer as time base, combined with CCRx register, LPTIM16 supports two channels of 16bit PWM output, or 16bit input capture function.

34.6.1 16bit PWM

Both independent capture/compare channels of LPTIM16 can output 16bit PWM waveforms. The PWM function needs to configure the capture/compare channel as a compare output.

After PWM function is enabled, LPTIM16 starts counting from 0x0000_0000. Take the positive polarity waveform as an example, when the count value is equal to the comparison value (CCRx), the output is set high, when the count value is equal to the target value (ARR), the output is low. The PWM period is determined by the ARR register and the duty cycle is determined by the CCRx register. The LPTIM16_CCSR.CCxP register can configure the polarity of the output waveform.

To implement PWM output function, LPTIM16_CCSR.CCxS needs to be configured to 10, at which time LPT_CHx becomes the output channel and the corresponding GPIO automatically enables the output function (the software needs to configure the GPIO as a digital peripheral function).

The following figure shows an example of PWM output with POLAR=1.

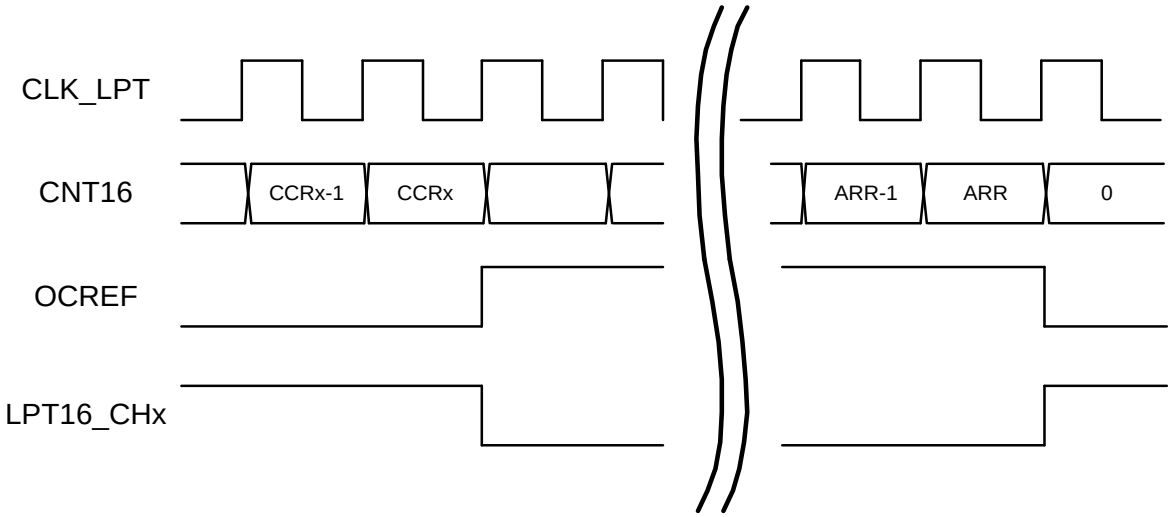


Figure 34–5 PWM Output

34.6.2 Input Capture

The LPTIM16's two capture/compare channels enable two independent input signal period or level width capture functions.The input signal capture function can be used with DMA to realize automatic handling of multiple consecutive capture results.

The input capture can be configured to capture on the rising edge, falling edge, or both edge of the input signal. Each time a capture occurs, the CAPxEDGE register indicates whether the current capture is a rising or falling edge.

Channel 1 of LPTIM16 can capture the external pin input or chip internal clock signal (XTLF, RCLP, RCLF). The period capture of internal clock signal can be used for clock frequency correction with software; while Channel 2 can only capture the external pin input signal.Channel 1 has a built-in prescaler. After dividing the frequency of the input signal, it can capture and correct the period of the high-frequency clock after frequency division.

After enabling the input mode, the 16bit counter is free to count as a time base. When a valid edge of the captured signal arrives, the current count value is latched into the CCRx register and a capture interrupt is generated; the capture interrupt flag is automatically cleared by hardware when the CCRx register is read by software, the capture interrupt flag can also be cleared by software by writing 1 in addition. When the capture interrupt flag is not cleared and a new capture event arrives, the capture conflict interrupt flag (CAPxOVR) will be set.

The following figure shows an example of capturing the rising and falling edges of the input signal.

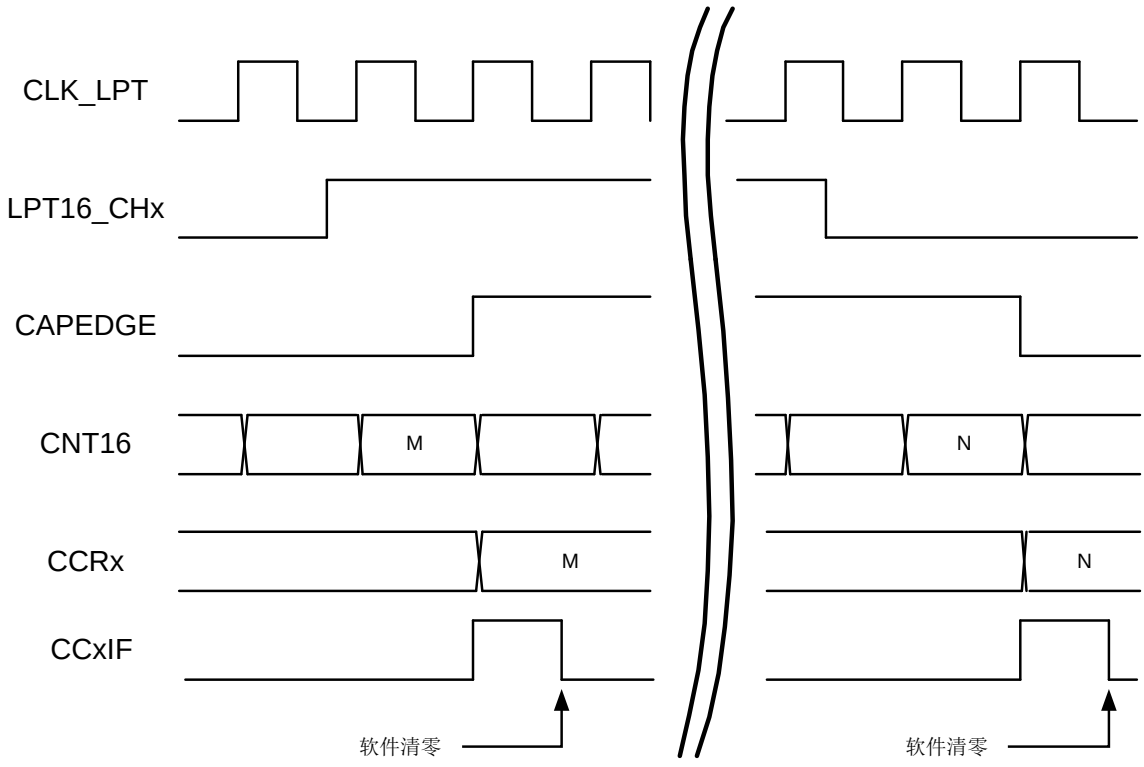


Figure 34–6 Input Signal Edge Capture

34.6.3 Input Digital Filter

The two input channels can independently enable or disable the input digital filter function. When digital filtering is enabled, use the CLK_LPT count clock to sample the input signal three times continuously, and the same level is considered valid.

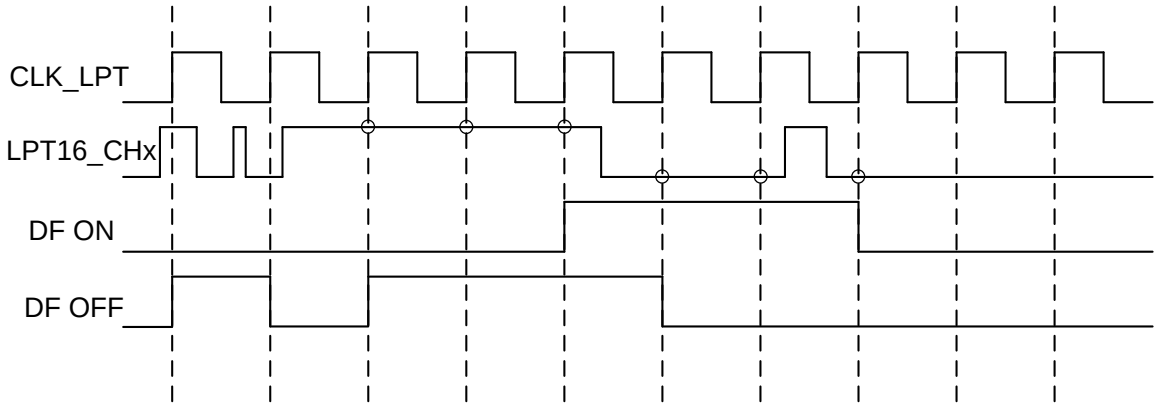


Figure 34–7 Channel Input Digital Filter

34.7 Quadrature Encoder

The encoder interface mode involves two external input signals. The LPTIM16 determines whether to increase or decrease the count value according to the edge of one signal relative to the level of the other signal. The following table shows the relationship between the counting method and the two input signals:

Valid edge	Corresponding signal level (LPT1FP corresponds to LPT2FP, LPT2FP corresponds to LPT1FP)	LPT1FP		LPT2FP	
		Rising	Falling	Rising	Falling
Only count at LPT1FP	High	Decrease	Increase	Not counted	Not counted
	Low	Increase	Decrease	Not counted	Not counted
Only count at LPT2FP	High	Not counted	Not counted	Increase	Decrease
	Low	Not counted	Not counted	Decrease	Increase
Count both at LPT1FP and LPT2FP	High	Decrease	Increase	Increase	Decrease
	Low	Increase	Decrease	Decrease	Increase

Table 34-2 Encoder Interface Counting Method

For example, when the counter counts with the LPT1FP signal as the clock, if the rising edge of LPT1FP is sampled that LPT2FP is high, the counter is decremented; if the falling edge of LPT1FP is sampled that LPT2FP is high, the counter is incremented.

The following figure shows an example of the encoder simultaneously counting on the edges of LPT1FP and LPT2FP.

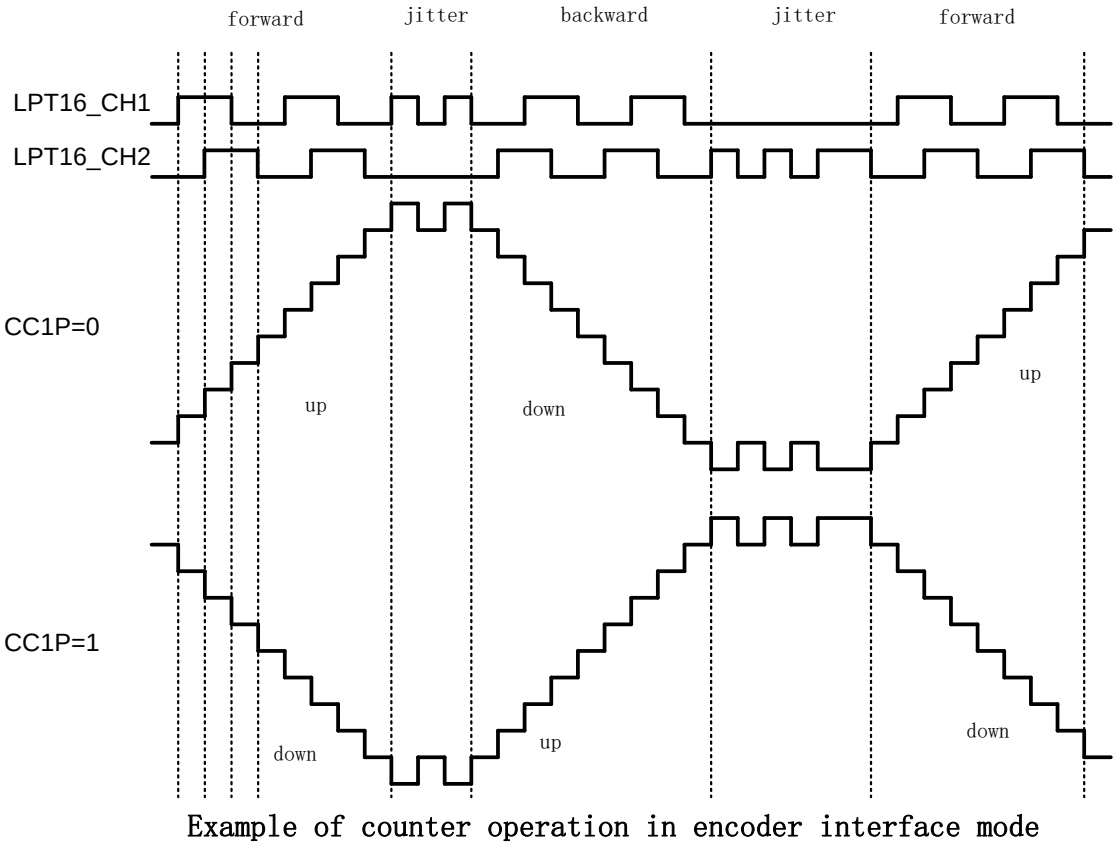


Figure 34–8 Example of Counter Operation in Encoder Mode

The quadrature coded signal uses LPT16_CHx input, you can choose whether to digitally filter the input signal. Note that there is also a prescaler circuit on the LPT16_CH1 channel, and the prescaler should be turned off when performing quadrature encoding counting.

The QECD register is used to configure the direction of the quadrature encoding count.

The encoding mode input channel needs to be set as follows:

- In the GPIO module, configure the corresponding pins as LPT16_CH1, LPT16_CH2 functions
- Select the input channel, configure LPTIM16_CCSR.CC1S=01, LPTIM16_CCSR.CC2S=01
- Select counting valid edge (input signal polarity), configure LPTIM16_CCSR.CC1P and CC2P
- Set the encoder mode, configure LPTIM16_CFGR.QEMD register
- Enable LPTIM16, set LPTIM16_CR.EN

34.8 Trigger Signal Output

LPTIM16 can output trigger signals to other peripherals under certain conditions. Trigger signal sources include:

- Enable LPTIM16
- Update event: counter overflow, count value equal to CRR1 or CRR2
- CC1 channel comparison pulse: the count value is equal to CCR1
- CC1 channel capture event
- CC2 channel capture event

The trigger signal is synchronized with APBCLK, so before using this function, you must turn on the LPTIM bus clock, that is, set the LPT16_PCE register.

34.9 Register

Offset	Name	Symbol
LPTIM16(Base Address: 0x40018800)		
0x00000000	LPTIM16 Config Register	LPTIM16_CFGR
0x00000004	LPTIM16 Counter Register	LPTIM16_CNT
0x00000008	LPTIM16 Capture/Compare Control and Status Register	LPTIM16_CCSR
0x0000000C	LPTIM16 Auto-Reload Register	LPTIM16_ARR
0x00000010	LPTIM16 Interrupt Enable Register	LPTIM16_IER
0x00000014	LPTIM16 Interrupt Status Register	LPTIM16_ISR
0x00000018	LPTIM16 Control Register	LPTIM16_CR
0x00000020	LPTIM16 Capture/Compare Register1	LPTIM16_CCR1
0x00000024	LPTIM16 Capture/Compare Register2	LPTIM16_CCR2

34.9.1 LPTIM16 Config Register (LPTIM16_CFGR)

NAME	LPTIM16_CFGR							
Offset	0x00000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							ETR_AFEN
access	U-0							R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-					MMS		
access	U-0					R/W-000		
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-	PSCSEL	-	DIVSEL			-	
access	U-0	R/W-0	U-0	R/W-000			U-0	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	EDGESEL	TRIGCFG		QEMD		ONST	TMOD	
access	R/W-0	R/W-00		R/W-00		R/W-0	R/W-00	

bit	name	functional description
31:25	--	RFU: **Reserved, read as 0**
24	**ETR_AFEN**	External Trigger input Analog Filter Enable 0: Disable analog filtering 1: Enable analog filtering, the filter width is about 100ns
23:19	--	RFU: **Reserved, read as 0**

bit	name	functional description
18:16	**MMS**	Master mode selection, used to configure the source of the synchronous trigger signal (TRGO) sent to the slave in the master mode. 000: RFU 001: Counter enable signal EN is used as TRFO 010: UE (update event) signal is used as TRGO 011: CC1 compare pulse, if the CC1IF flag is about to be set, TRGO outputs a positive pulse 100: CC1 channel capture event 101: CC2 channel capture event 110: RFU 111: RFU Note: The slave must enable the working clock in advance to receive the TRGO sent by the master timer
15	**--**	RFU: **Reserved, read as 0**
14	**PSCSEL**	Prescaler Input Select 0: CLKSEL selected clock 1: LPTETR
13	**--**	RFU: **Reserved, read as 0**
12:10	**DIVSEL**	Counter Clock Divider Select 000: Divided-by-1 001: Divided-by-2 010: Divided-by-4 011: Divided-by-8 100: Divided-by-16 101: Divided-by-32 110: Divided-by-64 111: Divided-by-128
9:8	**--**	RFU: **Reserved, read as 0**
7	**EDGESEL**	ETR Clock Edge Select 0: LPT_ETR rising edge count 1: LPT_ETR falling edge count
6:5	**TRIGCFG**	ETR Trigger Configuration (Need to use the internal clock to synchronously sample LPT_ETR) 00: LPT_ETR input signal rising edge trigger 01: LPT_ETR input signal falling edge trigger 10/11: LPT_ETR external input signal rising and falling edge trigger

bit	name	functional description
4:3	**QEMD**	Quad encoder mode, only valid when TMODE=00 00: Disable quadrature encoder 01: Encoder mode 1; the counter uses the LPT2FP edge and counts according to the level of LPT1FP 10: Encoder mode 2; the counter uses the LPT1FP edge and counts according to the level of LPT2FP 11: Encoder mode 3; the counter uses both LPT1FP and LPT2FP edges at the same time, counting according to other input signal levels
2	**ONST**	One State Timer 0: Continuous counting mode: The counter keeps running after being triggered until it is disabled. After the counter reaches the target value, it returns to 0 and restarts counting, and an overflow interrupt is generated. 1: Single counting mode: After the counter is triggered, it counts to the target value and then returns to 0, and stops automatically, generating an overflow interrupt.
1:0	**TMODE**	Timer Operation Mode 00: General timer mode 01: External pulse trigger counting 10: External asynchronous pulse counting 11: Timeout mode

34.9.2 LPTIM16 Counter Register (LPTIM16_CNT)

NAME	**LPTIM16_CNT**							
Offset	0x00000004							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CNT16[15:8]							
access	R-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CNT16[7:0]							
access	R-0000 0000							

bit	name	functional description
31:16	**--**	RFU: **Reserved, read as 0**

bit	name	functional description
15:0	**CNT16**	Counter current count value(Counter 16bits-wide)

34.9.3 LPTIM16 Capture/Compare Control and Status Register (LPTIM16_CCSR)

NAME	LPTIM16_CCSR							
Offset	0x00000008							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	CAP1PSC						CAP1SSEL	
access	R/W-00 0000						R/W-00	
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-		CAP2EDGE	CAP1EDGE	-	-	CC2P	CC1P
access	U-0		R-0	R-0	U-0	U-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-		CC2DF	CC1DF	CAPCFG2		CAPCFG1	
access	U-0		R/W-0	R/W-0	R/W-00		R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				CC2S		CC1S	
access	U-0				R/W-00		R/W-00	

bit	name	functional description
31:26	**CAP1PSC**	Capture channel 1 prescaler 0x00: Divided-by-1 0x01: Divided-by-2 0x02: Divided-by-3 …… 0x3F: Divided-by-64
25:24	**CAP1SSEL**	Capture channel 1 source select, only valid when CH1 channel is configured as input capture 00: LPT16_CH1 input 01: XTLF 10: RCLP 11: RCLF
23	**--**	RFU: **Reserved, read as 0**
22	**--**	RFU: **Reserved, read as 0**
21	**CAP2EDGE**	Channel2 Captured Edge, update when CC2IF is set 0: Rising edge 1: Falling edge
20	**CAP1EDGE**	Channel 1 Captured Edge, update when CC1IF is set 0: Rising edge 1: Falling edge
19	**--**	RFU: **Reserved, read as 0**

bit	name	functional description
18	--	RFU: **Reserved, read as 0**
17	**CC2P**	When channel 2 is configured as comparison function: Channel 2 Compare Output Polarity 0: Positive polarity waveform, start low, set high when the count value is equal to the comparison value, and return to low when the count value is equal to ARR 1: Negative polarity waveform, inverted positive waveform
		When channel 2 is configured as input capture function: Channel 2 Input Polarity 0: Input is not inverted 1: Input inverts
16	**CC1P**	When channel 1 is configured as comparison function: Channel 1 Compare Output Polarity 0: Positive polarity waveform, start low, set high when the count value is equal to the comparison value, and return to low when the count value is equal to ARR 1: Negative polarity waveform, positive polarity waveform inverted
		When channel 1 is configured as input capture function: Channel 1 Input Polarity 0: Input is not inverted 1: Input inverts
15:14	--	RFU: **Reserved, read as 0**
13	**CC2DF**	Channel 2 Input Digital Filter 1: Enable digital filter 0: Disable digital filter
12	**CC1DF**	Channel 1Input Digital Filter 1: Enable digital filter 0: Disable digital filter
11:10	**CAPCFG2**	Channel 2 Capture Edge Config 00: Rising edge capture 01: Falling edge capture 10: Rising and falling edges capture 11: RFU
9:8	**CAPCFG1**	Channel 1 Capture Edge Config 00: Rising edge capture 01: Falling edge capture 10: Rising and falling edges capture 11: RFU
7:6	--	RFU: **Reserved, read as 0**
5:4	--	RFU: **Reserved, read as 0**

bit	name	functional description
3:2	**CC2S**	Channel 2 Capture/Compare Select 00,11: Disable channel 2 capture/compare function 01: Enable channel 2 capture function (LPT16_CH2 is input) 10: Enable channel 2 compare function (LPT16_CH2 is output)
1:0	**CC1S**	Channel 1 Capture/Compare Select 00,11: Disable channel 1 capture/compare function 01: Enable channel 1 capture function (LPT16_CH1 is input) 10: Enable channel 1 compare function (LPT16_CH1 is output)

34.9.4 LPTIM16 Auto-Reload Register (LPTIM16_ARR)

NAME	LPTIM16_ARR							
Offset	0x0000000C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ARR[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ARR[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	**--**	RFU: **Reserved, read as 0**
15:0	**ARR**	Auto-Reload Register When the count value of the counter is equal to ARR, the counter returns to the initial value and starts counting up again

34.9.5 LPTIM16 Interrupt Enable Register (LPTIM16_IER)

NAME	LPTIM16_IER							
Offset	0x00000010							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							

access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						OVR2IE	OVR1IE
access	U-0						R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TRIGIE	OVIE	-				CC2IE	CC1IE
access	R/W-0	R/W-0	U-0				R/W-0	R/W-0

bit	name	functional description
31:10	--	RFU: **Reserved, read as 0**
9	**OVR2IE**	Channel 2 Over-Capture Interrupt Enable 1: Allow interrupt 0: Prohibit interrupt
8	**OVR1IE**	Channel 1 Over-Capture Interrupt Enable 1: Allow interrupt 0: Prohibit interrupt
7	**TRIGIE**	External Trigger Interrupt Enable 1: External trigger arrival interrupt enable 0: External trigger arrival interrupt disable
6	**OVIE**	Counter Over-Flow Interrupt Enable 1: Counter overflow interrupt enable 0: Counter overflow interrupt disable
5:2	--	RFU: **Reserved, read as 0**
1	**CC2IE**	Capture/Compare Channel 2 Interrupt Enable 1: Capture/Compare channel 2 interrupt enable 0: Capture/Compare channel 2 interrupt disable
0	**CC1IE**	Capture/Compare channel 1 Interrupt Enable 1: Capture/Compare channel 1 interrupt enable 0: Capture/Compare channel 1 interrupt disable

34.9.6 LPTIM16 Interrupt Status Register (LPTIM16_ISR)

NAME	LPTIM16_ISR							
Offset	0x00000014							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						CAP2OVR	CAP1OVR

access	U-0						R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TRIGIF	OVIF	-				CC2IF	CC1IF
access	R/W-0	R/W-0	U-0				R/W-0	R/W-0

bit	name	functional description
31:10	--	RFU: **Reserved, read as 0**
9	**CAP2OVR**	Channel 2 Over-Capture Interrupt Flag, hardware set, write 1 to clear by software 1: In input capture mode, a new capture occurs whenCC2IF is 1, and overrun occurs 0: No overrun occurs
8	**CAP1OVR**	Channel 1 Over-Capture Interrupt Flag, hardware set, write 1 to clear by software 1: In input capture mode, a new capture occurs whenCC1IF is 1, and overrun occurs 0: No overrun occurs
7	**TRIGIF**	External Trigger Interrupt Flag, write 1 to clear 1: External trigger arrival interrupt generates 0: No interrupt generates
6	**OVIF**	Counter Over-Flow Interrupt Flag, write 1 to clear 1: Non-encoder mode – Counter equal to ARR Encoder mode – Counter overflow (equal to ARR), or underflow 0: No interrupt generates
5:2	--	RFU: **Reserved, read as 0**
1	**CC2IF**	Capture/Compare Channel 2 Interrupt Flag, hardware set, write 1 to clear by software 1: Counter value and comparison value 2 match, or a capture event occurs 0: No interrupt generates ***Note:*** *In capture mode, this flag is cleared by writing 1 or reading CCR2.*
0	**CC1IF**	Capture/Compare channel 1 Interrupt Flag, hardware set, write 1 to clear by software 1: Counter value and comparison value 1 match, or a capture event occurs 0: No interrupt generates ***Note:*** *In capture mode, this flag is cleared by writing 1 or reading CCR1.*

34.9.7 LPTIM16 Control Register (LPTIM16_CR)

NAME	LPTIM16_CR							
Offset	0x00000018							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							EN
access	U-0							R/W-0

bit	name	functional description
31:1	**--**	RFU: **Reserved, read as 0**
0	**EN**	LPTIM Enable 1: Enable counter count 0: Disable counter count

34.9.8 LPTIM16 Capture/Compare Register1 (LPTIM16_CCR1)

NAME	LPTIM16_CCR1							
Offset	0x0000001C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR1[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR1[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	**--**	RFU: **Reserved, read as 0**
15:0	**CCR1**	Channel1 Capture/Compare Register

34.9.9 LPTIM16 Capture/Compare Register2 (LPTIM16_CCR2)

NAME	LPTIM16_CCR2							
Offset	0x00000020							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	CCR2[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	CCR2[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	**--**	RFU: **Reserved, read as 0**
15:0	**CCR2**	Channel2 Capture/Compare Register

35 Real-time Clock (RTCA)

35.1 Introduction

The Real-time Clock (RTCA) module maintains accurate timing for long periods of time, consumes very low power, and works in all power consumption modes.

The main features are as follows:

- BCD time format, complete perpetual calendar (00~99 years)
- Periodic wake-up interrupt
- Alarm interrupt
- Configurable periodic timing signal output
- Digital correction, accuracy +/-0.477ppm
- Feedback resistor integration
- RTC timer part does not reset

35.2 Block Diagram

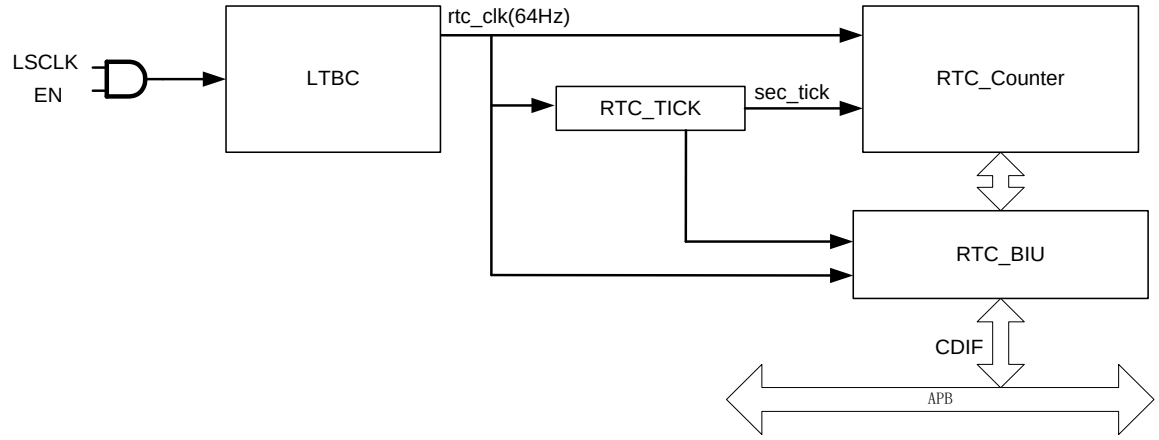


Figure 35–1 RTC Block Diagram

The LTBC module is a low power time base counter module used to generate the low speed operating clock required by the system, see section 35.3.1 LTBC function introduction for specific description.

The RTC_TICK module is mainly used to generate the second pulse signal sec_tick that jumps every second and the 2Hz,4Hz,8Hz,16Hz and other signals needed to generate interrupts. The sec_tick signal is output to the RTC_Counter module to realize the counter synchronization of the perpetual

calendar.

The RTC_Counter module is the perpetual calendar implementation module of RTC, including a second counter, a minute counter, an hour counter, a day counter, a week counter, a month counter, and a year counter.The module enables automatic recognition of leap years.

35.3 Working Principle

The RTCA will reset after power on, so the software needs to set the current time before normal operation. The time clock uses 32.768KHz crystal oscillator. Since the crystal oscillator may fail, in order to ensure reliability, the failure detection circuit is enabled to continuously detect the 32.768KHz oscillator output and generate an alarm interrupt if it is found to fail.Meanwhile, the software can configure whether to automatically switch the RTCA clock to RCLP when XTLF stops oscillation. If this function is enabled, the RTCA time will have a certain error, but it will not stop; if the automatic switching is not enabled, it can also be handled by the software after responding to the stop oscillation interrupt.

35.3.1 Low-power Time Base Counter (LTBC)

The low-power time base counter (LTBC) module is used to generate the low speed operating clock required by the system. Its functions include：

- Get 64Hz RTCA and WDT working clock by prescaling LSCLK
- Digital correction of the RTC clock can be achieved by adjusting the count period, the minimum step size can be achieved by adjusting once every 32s is 0.952ppm, and the theoretical accuracy after adjustment is +/-0.477ppm
- Precise second time mark can be obtained by using PLL virtual correction
- Generate 1KHz, 256Hz, 64Hz, 16Hz, 4Hz, 1Hz periodic interrupts, of which 1K and 256Hz are uncorrected and the others are digitally corrected (if digital correction is enabled)
- The 64Hz prescaler circuit is not affected by chip reset
- 1/256s precision timing

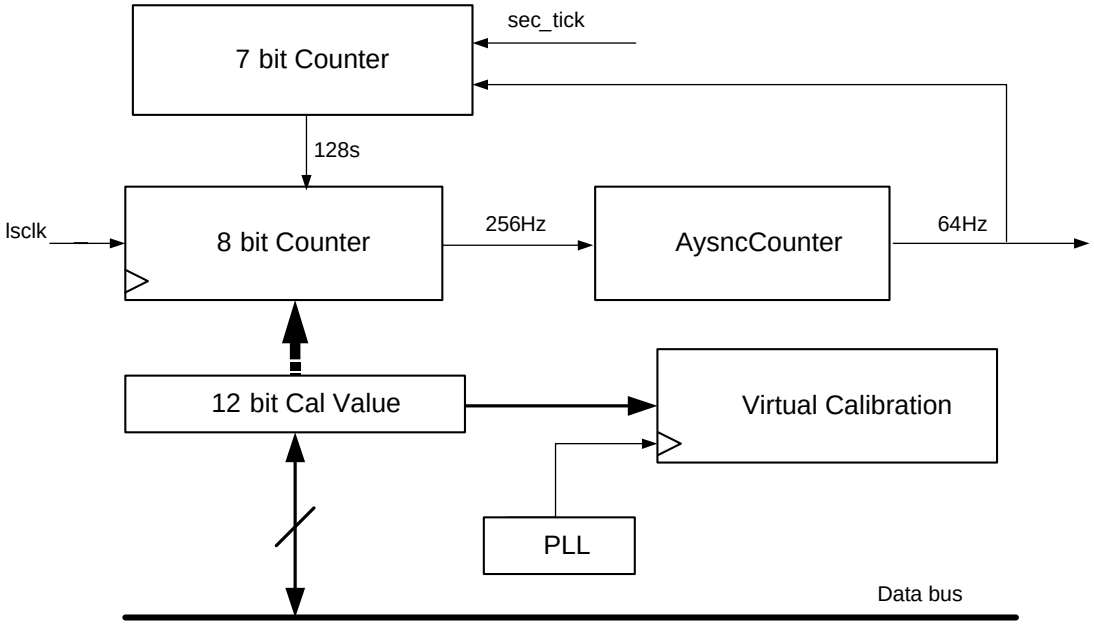


Figure 35–2 LTBC Block Diagram

35.3.2 LTBC Digital Correction

LTBC is mainly composed of a synchronous prescaler counter, an asynchronous divider counter, a clock correction value register, a virtual correction circuit and a control register.

The purpose of digital correction is to enable the RTC to obtain averagely accurate timing over a longer period of time.Since the clock source of the RTC is 32768Hz, the minimum step of digital correction is 30.5us, and if it is adjusted once in 1 second, the maximum accuracy can only reach 30.517ppm. To get higher accuracy, the adjustment must be made in longer period.FM33LG0 takes 32s as a correction period, and each period can be adjusted from 0 to +/-511 32768Hz clock period, so the highest accuracy is 30.5us/32s = 0.952ppm, the maximum adjustment range is +/-(511*30.517us/32s) = +/-487ppm, and the average minimum clock error after adjustment is +/-0.476ppm.

The correction value consists of 10bit registers, where the highest bit is the sign bit, indicating the increase or decrease of the count value, and the remaining 9bit indicates the absolute value of the increase or decrease. In order to improve the average accuracy per second and avoid large second-to-second jitter, the 32s correction value is distributed equally within each second, which is implemented as follows.

In addition to the highest sign bit, the remaining 9 bits can be divided into a high 4-bit public value and a 5-bit private value, where the public value indicates the value to be adjusted every second within 32s, and the private value indicates the need to add or subtract 1 in some seconds within 32s.

Bit9	Bit[8:5]	Bit[4:0]
Sign	Common Value I	Differential Value (D)

The correction value formula can be expressed as follows: Correction(ppm) = (C*32 + D)*30.517/32000000

Assuming that the clock increase by 0.953ppm, which is equivalent to only 30.5us in 32s, the correction value is written as 0_0000_00001, so the public value is 0 and the private value is 1, only need to add 1 to a second period within 32s. And assuming that the clock increase by 487ppm, which is equivalent to 511 30.5us in 32s, the correction value is written as 0_1111_11111, the public value is 15, the private value is 31, which means that 15 is added every second in 32s, and there are 31swhich need to add 1 extra.

Example of correction value:

ppm	ADJUST[1]	Common	Differential	Expression
0.953	0_0000_00001	0	1	1*30.517/32000000
-125.88	1_0100_00100	4	4	(4*32+4)*30.517/32000000
32.42	0_0001_00010	1	2	(1*32+2)*30.517/32000000
487.32	0_1111_11111	15	31	(15*32+31)*30.517/32000000

Note:

[1] ADJUST: Clock Error Adjustment Register

Through the above method, a smoothly adjusted second period can be obtained, and the maximum difference between second and second is only 30.5ppm, which can avoid the second period jitter introduced by centralized adjustment.

To avoid timing conflicts, the software should update ADJUST and start clock correction after the second interrupt.

35.3.3 BCD Clock

Second Count

The second count only needs 7 bits, counting from 0 to 59, where bit[3:0] is 1 second unit, counting range 0-9; bit[6:4] is 10 seconds unit, counting range 0-5. When the count reaches 60s, the second incoming signal is triggered to add 1 to the minute counter.

Bit6-4	Bit3-0
0-5	0-9

Minute Count

The minute count also only needs 7 bits, and the counting range is the same as second count, so the implementation method is same.

Bit6-4	Bit3-0
0-5	0-9

Hour Count

Thehour countrangeis 0 to 24 with only 6 bits.

Bit5-4	Bit3-0
0-2	0-9

Day Count

The day count range is 1 to 31, only 6 bits, starting from 1, counting up to 28/29/30/31 according to the month and leap year, and triggering the day-in signal to add 1 to the month counter when the count is full.

Bit5-4	Bit3-0
0-3	0-9

Week Count

The week count range is 0 to 6 with only 3 bits and cycle count from 0 to 6.

Bit2-0
0-6

Month Count

The month count range is 1 to 12, only 5 bits, triggering the month-in signal to add 1 to the year counter when the count is full.

Bit4	Bit3-0
0-1	0-9

Year Count

The year count range is 0 to 99 with 8 bits and cycle count from 0 to 99.

Bit7-4	Bit3-0
0-9	0-9

35.3.4 RTC Enable and Disable

RTCA can be enabled or disabled by software. RTCA is disabled by default after power-on, and the software enables RTCA by operating the EN register after the XTLF starts to oscillate.

When the EN register is cleared, the RTCA internal clock is gated to save power.

35.3.5 RTC Time Setting

Software can set the RTC BCD registers directly at any time, usually it is recommended to set the time after XTLF is fully started; since the operation of setting the time register is asynchronous with the RTC timing, it is recommended that the software set the time after the second interrupt event and read out the time value for verification after the time setting.

Note that the hardware does not check the time validity, the software must ensure that the written BCD time is correct.

The FM33LG0 also supports ms-level timing, i.e. the time can be set to 3.9ms level accuracy (1/256s). In addition, when the software writes the second time, the hardware automatically clears the 64Hz->1Hz intersecond counter to achieve second alignment.

In order to improve the anti-interference capability, FM33LG0 provides time write protection function. 0xACACACAC must be written to the write protection register before the time register can be rewritten. After the time setting is completed, the software can prohibit the writing of time registers by writing any other value to restore the write protection.

35.3.6 RTC Time Reading

Time reading mode 1:

- Read the current time register value
- Read the current time register value again
- If the contents of two readings are the same, it is the correct current time; otherwise, repeat the first two steps

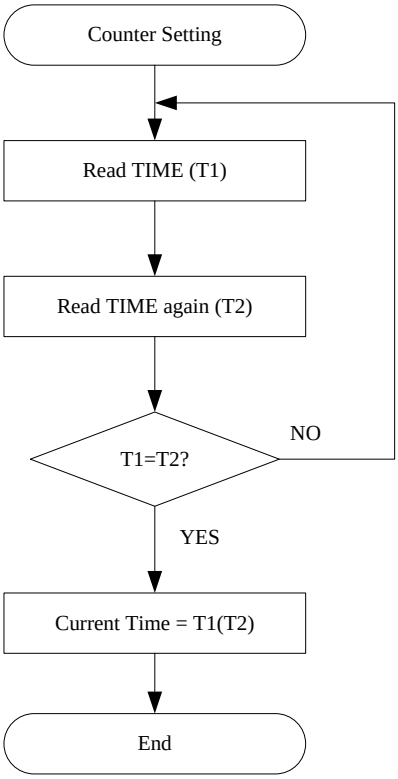


Figure 35–3 RTC Time Reading Flow Chart

Time reading mode 2:

Reading the time register immediately by software after 1s interrupt occurs, which ensures that the correct current time value is read.

35.3.7 Leap Year Determination

The RTCA module will automatically determine the leap year within the range of 2000-2099 and process the BCD time.

35.4 Register

Offset	Name	Symbol
RTCA(Base Address: 0x40011000)		
0x00000000	RTC Write Enable Register	RTCA_WER
0x00000004	RTC Interrupt Enable Register	RTCA _IER
0x00000008	RTC Interrupt Status Register	RTCA _ISR
0x0000000C	BCD Format Time Second Registers	RTCA _BCDSEC
0x00000010	BCD Format Time Minute Registers	RTCA _BCDMIN
0x00000014	BCD Format Time Hour Registers	RTCA _BCDHOUR
0x00000018	BCD Format Time Day Registers	RTCA _BCDDAY
0x0000001C	BCD Format Time Week Registers	RTCA _BCDWEEK
0x00000020	BCD Format Time Month Registers	RTCA _BCDMONTH
0x00000024	BCD Format Time Year Registers	RTCA _BCDYEAR
0x00000028	RTCA Alarm Register	RTCA _ALARM
0x0000002C	RTCA Time Mark Select	RTCA _TMSEL
0x00000030	RTCA Time Adjust Register	RTCA _ADJUST
0x00000034	RTCA Time Adjust Sign Register	RTCA _ADSIGN
0x0000003C	RTCA Sub-Second Counter	RTCA _SBSCNT
0x00000040	RTCA Control Register	RTCA _CR

35.4.1 RTC Write Enable Register (RTCA_WER)

NAME	RTCA_WER							
Offset	0x00000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							WE
access	U-0							R/W-0

bit	name	functional description
31:1	-	RFU: **Reserved, read as 0**
0	WE	RTC Write Enable When the CPU writes 0xACACACAC to RTCA_WER, the CPU

bit	name	functional description
		is allowed to write the initial value to the BCD time register of RTC, and then RTCWE is set to 1. When the CPU writes any value not 0xACACACAC to RTCWE, the write protection is restored, and then RTCA_WER is cleared to 0.

35.4.2 RTC Interrupt Enable Register (RTCA_IER)

NAME	RTCA_IER							
Offset	0x00000004							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			ADJ_IE	ALARM_IE	1KHZ_IE	256HZ_IE	64HZ_IE
access	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
name	16HZ_IE	8HZ_IE	4HZ_IE	2HZ_IE	SEC_IE	MIN_IE	HOUR_IE	DAY_IE
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:13	-	RFU: **Reserved, read as 0**
12	ADJ_IE	Time Adjust Interrupt Enable 1: Interrupt enable 0: Interrupt disable
11	ALARM_IE	Alarm Interrupt Enable 1: Interrupt enable 0: Interrupt disable
10	1KHZ_IE	1KHz Periodic Interrupt Enable 1: Interrupt enable 0: Interrupt disable
9	256HZ_IE	256HzPeriodic Interrupt Enable 1: Interrupt enable 0: Interrupt disable
8	64HZ_IE	64hz Periodic Interrupt Enable 1: Interrupt enable 0: Interrupt disable
7	16HZ_IE	16Hz Periodic Interrupt Enable

bit	name	functional description
		1: Interrupt enable 0: Interrupt disable
6	8HZ_IE	8Hz Periodic Interrupt Enable 1: Interrupt enable 0: Interrupt disable
5	4HZ_IE	4Hz Periodic Interrupt Enable 1: Interrupt enable 0: Interrupt disable
4	2HZ_IE	2Hz Periodic Interrupt Enable 1: Interrupt enable 0: Interrupt disable
3	SEC_IE	1Hz Periodic Interrupt Enable 1: Interrupt enable 0: Interrupt disable
2	MIN_IE	Minute Interrupt Enable 1: Interrupt enable 0: Interrupt disable
1	HOUR_IE	Hour Interrupt Enable 1: Interrupt enable 0: Interrupt disable
0	DAY_IE	Day Interrupt Enable 1: Interrupt enable 0: Interrupt disable

35.4.3 RTC Interrupt Status Register (RTCA_ISR)

NAME	RTCA_ISR							
Offset	0x00000008							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			ADJ_IF	ALARM_IF	1KHZ_IF	256HZ_IF	64HZ_IF
access	U-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
name	16HZ_IF	8HZ_IF	4HZ_IF	2HZ_IF	SEC_IF	MIN_IF	HOUR_IF	DAY_IF
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:13	-	RFU: **Reserved, read as 0**
12	ADJ_IF	Time Adjust Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
11	ALARM_IF	Alarm Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
10	1KHZ_IF	1KHz Periodic Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
9	256HZ_IF	256Hz Periodic Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
8	64HZ_IF	64Hz Periodic Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
7	16HZ_IF	16Hz Periodic Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
6	8HZ_IF	8Hz Periodic Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
5	4HZ_IF	4Hz Periodic Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
4	2HZ_IF	2Hz Periodic Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
3	SEC_IF	1Hz Periodic Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
2	MIN_IF	Minute Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
1	HOUR_IF	Hour Interrupt Flag,write 1 to clear 1: Interrupt set 0: No interrupt generates
0	DAY_IF	Day Interrupt Flag,write 1 to clear 1: Interrupt set 0: No interrupt generates

35.4.4 BCD Format Time Second Registers (RTCA_BCDSEC)

NAME	RTCA_BCDSEC							
Offset	0x0000000C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	SEC						
access	U-0	R/W-xxx xxxx						

bit	name	functional description
31:7	-	RFU: **Reserved, read as 0**
6:0	SEC	Binary-Coded Decimal Format Seconds Register

35.4.5 BCD Format Time Minute Registers (RTCA_BCDMIN)

NAME	RTCA_BCDMIN							
Offset	0x00000010							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	MIN						
access	U-0	R/W-xxx xxxx						

bit	name	functional description
31:7	-	RFU: **Reserved, read as 0**
6:0	MIN	Binary-Coded Decimal Format Minutes Register

35.4.6 BCD Format Time Hour Registers (RTCA_BCDHOUR)

NAME	RTCA_BCDHOUR							
Offset	0x00000014							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		HOUR					
access	U-0		R/W-xx xxxx					

bit	name	functional description
31:6	-	RFU: **Reserved, read as 0**
5:0	HOUR	Binary-Coded Decimal Format Hours Register

35.4.7 BCD Format Time Day Registers (RTCA_BCDDAY)

NAME	RTCA_BCDDAY							
Offset	0x00000018							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		DAY					
access	U-0		R/W-xx xxxx					

bit	name	functional description
31:6	-	RFU: **Reserved, read as 0**
5:0	DAY	Binary-Coded Decimal Format Date Register

35.4.8 BCD Format Time Week Registers (RTCA_BCDWEEK)

NAME	RTCA_BCDWEEK							
Offset	0x0000001C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					WEEK		
access	U-0					R/W-xxx		

bit	name	functional description
31:3	-	RFU: **Reserved, read as 0**
2:0	WEEK	Binary-Coded Decimal Format Week Register

35.4.9 BCD Format Time Month Registers (RTCA_BCDMONTH)

NAME	RTCA_BCDMONTH							
Offset	0x00000020							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			MONTH				
access	U-0			R/W-x xxxx				

bit	name	functional description
31:5	-	RFU: **Reserved, read as 0**
4:0	MONTH	Binary-Coded Decimal Format Month Register

35.4.10 BCD Format Time Year Registers (RTCA_BCDYEAR)

NAME	RTCA_BCDYEAR							
Offset	0x00000024							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	YEAR							
access	R/W-xxxx xxxx							

bit	name	functional description
31:8	-	RFU: **Reserved, read as 0**
7:0	YEAR	Binary-Coded Decimal Format Year Register

35.4.11 RTCA Alarm Register (RTCA_ALARM)

NAME	RTCA_ALARM							
Offset	0x00000028							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-		HOUR					
access	U-0		R/W-00 0000					
bit	Bit15	Bit14	BIT13	BIT12	BIT11	BIT10	Bit9	Bit8
name	-	MIN						
access	U-0	R/W-000 0000						
bit	Bit7	Bit6	BIT5	BIT4	BIT3	BIT2	Bit1	Bit0
name	-	SEC						
access	U-0	R/W-000 0000						

bit	name	functional description
31:22	-	RFU: **Reserved, read as 0**
21:16	HOUR	Alarm Hour Register
15	-	RFU: **Reserved, read as 0**

bit	name	functional description
14:8	MIN	Alarm Minute Register
7	-	RFU: **Reserved, read as 0**
6:0	SEC	Alarm Second Register

35.4.12 RTCA Time Mark Select (RTCA_TMSEL)

NAME	RTCA_TMSEL							
Offset	0x0000002C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				TMSEL			
access	U-0				R/W-0000			

bit	name	functional description
31:4	-	RFU: **Reserved, read as 0**
3:0	TMSEL	Time Mark Select: 0000: RFU 0001: RFU 0010: Output second counter carry signal, high level width 1s 0011: Output minute counter carry signal, high level width 1s 0100: Output hour counter carry signal, high level width 1s 0101: Output day counter carry signal, high level width 1s 0110: Output alarm matching signal 0111: Output 32 second square wave signal 1000: RFU 1001: Reverse output second counter carry signal 1010: Reverse output minute counter carry signal 1011: Reverse output hour counter carry signal 1100: Reverse output day counter carry signal

bit	name	functional description
		1101: Reverse output alarm matching signal 1110: RFU 1111: Output RTC internal second time scale square wave

35.4.13 RTCA Time Adjust Register (RTCA_ADJUST)

NAME	RTCA_ADJUST							
Offset	0x00000030							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						ADSIGN	ADJUST[8]
access	U-0						R/W-x	R/W-x
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ADJUST[7:0]							
access	R/W-xxxx xxxx							

bit	name	functional description
31:10	-	RFU: **Reserved, read as 0**
9	ADSIGN	Adjust Sign 0: Indicates to increase the initial value of the count 1: Indicates to decrease the initial value of the count
8:0	ADJUST	Time Adjust

35.4.14 RTCA Sub-Second Counter (RTCA_SBSCNT)

NAME	RTCA_SBSCNT							
Offset	0x0000003C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							

access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	MSCNT							
access	R/W-xxxx xxxx							

bit	name	functional description
31:8	-	RFU: **Reserved, read as 0**
7:0	MSCNT	Milli-Second Counter, effective bit 8bit, precision 3.9ms

35.4.15 RTCA Control Register (RTCA_CR)

NAME	RTCA_CR							
Offset	0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							EN
access	U-0							R/W-0

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
0	EN	RTC enable 0: Disable RTCA 1: Enable RTCA

36 Real-time Clock (RTCB)

36.1 Introduction

The Real-time Clock (RTCB) module maintains accurate timing for long periods of time. It is in the VAO power domain and can still use the VBTA power supply to maintain time when VDD is powered off.

The main features are as follows：

- BCD time format, complete perpetual calendar (00~99 years)
- Periodic wake-up interrupt
- Alarm interrupt
- Configurable periodic timing signal output
- Digital correction, accuracy +/-0.477ppm
- RTC timer part does not reset
- Five 32bit backup registers

36.2 Block Diagram

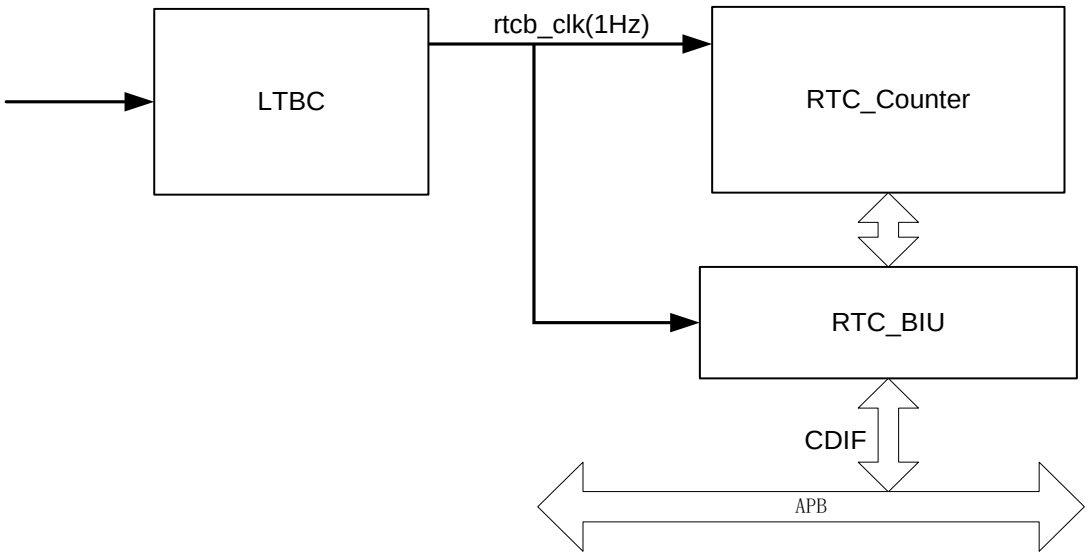


Figure 36–1 RTCB Block Diagram

The LTBC module is a low power time base counter module used to generate the low speed operating

clock required by the system, see LTBC function introduction for specific description.

The RTC_Counter module is the perpetual calendar implementation module of RTC, including a second counter, a minute counter, an hour counter, a day counter, a week counter, a month counter, and a year counter. The module enables automatic recognition of leap years.

36.3 Working Principle

The RTCB will reset after power on, so the software needs to set the current time before normal operation. The time clock uses 32.768KHz crystal oscillator. Since the crystal oscillator may fail, in order to ensure reliability, the failure detection circuit is enabled to continuously detect the 32.768KHz oscillator output and generate an alarm interrupt if it is found to fail.

36.3.1 Working Clock

RTCB only uses XTLF to work. If XTLF stops oscillating, RTCB stops running time.

36.3.2 Low-power Time Base Counter (LTBC)

The low-power time base counter (LTBC) module is used to generate the low speed operating clock required by the system. Its functions include：

- Get 1Hz RTC working clock by prescaling XTLF
- Digital correction of the RTC clock can be achieved by adjusting the count period, the minimum step size can be achieved by adjusting once every 32s is 0.952ppm, and the theoretical accuracy after adjustment is +/-0.477ppm
- Generate 1Hz periodic interrupt and alarm timer interrupt
- Prescaler circuit and BCD clockare not affected by chip reset

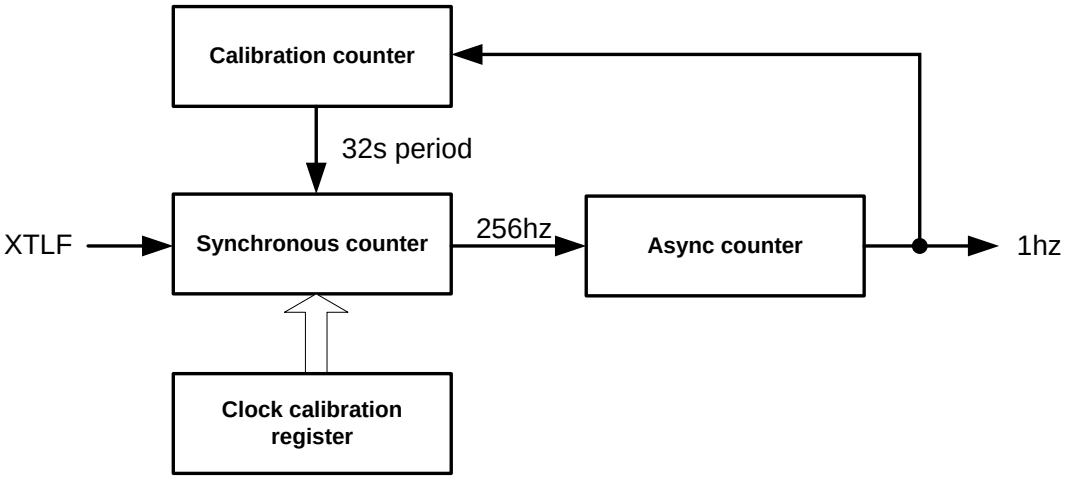


Figure 36–2 LTBC Block Diagram

36.3.3 Low-power Time Base Counter (LTBC)

LTBC is mainly composed of a synchronous prescaler counter, an asynchronous divider counter, a clock correction value register, a virtual correction circuit and a control register.

The purpose of digital correction is to enable the RTC to obtain averagely accurate timing over a longer period of time. Since the clock source of the RTC is 32768Hz, the minimum step of digital correction is 30.5us, and if it is adjusted once in 1 second, the maximum accuracy can only reach 30.517ppm. To get higher accuracy, the adjustment must be made in longer period.FM33LG0 takes 32s as a correction period, and each period can be adjusted from 0 to +/-511 32768Hz clock period, so the highest accuracy is 30.5us/32s = 0.952ppm, the maximum adjustment range is +/-(511*30.517us/32s) = +/-487ppm, and the average minimum clock error after adjustment is +/-0.476ppm.

The correction value consists of 10bit registers, where the highest bit is the sign bit, indicating the increase or decrease of the count value, and the remaining 9bit indicates the absolute value of the increase or decrease. In order to improve the average accuracy per second and avoid large second-to-second jitter, the 32s correction value is distributed equally within each second, which is implemented as follows.

In addition to the highest sign bit, the remaining 9 bits can be divided into a high 4-bit public value and a 5-bit private value, where the public value indicates the value to be adjusted every second within 32s, and the private value indicates the need to add or subtract 1 in some seconds within 32s.

Bit9	Bit[8:5]	Bit[4:0]
Sign	Common Value I	Differential Value (D)

The correction value formula can be expressed as follows: Correction(ppm) = (C*32 + D)*30.517/32000000

Assuming that the clock increase by 0.953ppm, which is equivalent to only 30.5us in 32s, the correction value is written as 0_0000_00001, so the public value is 0 and the private value is 1, only need to add 1 to a second period within 32s. And assuming that the clock increase by 487ppm, which is equivalent to 511 30.5us in 32s, the correction value is written as 0_1111_11111, the public value is 15, the private value is 31, which means that 15 is added every second in 32s, and there are 31swhich need to add 1 extra.

Example of correction value:

ppm	ADJUST[1]	Common	Differential	Expression
0.953	0_0000_00001	0	1	1*30.517/32000000
-125.88	1_0100_00100	4	4	(4*32+4)*30.517/32000000
32.42	0_0001_00010	1	2	(1*32+2)*30.517/32000000
487.32	0_1111_11111	15	31	(15*32+31)*30.517/32000000

Note:

[1] ADJUST: Clock Error Adjustment Register

Through the above method, a smoothly adjusted second period can be obtained, and the maximum difference between second and second is only 30.5ppm, which can avoid the second period jitter introduced by centralized adjustment.

To avoid timing conflicts, the software should update ADJUST and start clock correction after the second interrupt.

36.3.4 BCD Clock

Second Count

The second count only needs 7 bits, counting from 0 to 59, where bit[3:0] is 1 second unit, counting range 0-9; bit[6:4] is 10 seconds unit, counting range 0-5. When the count reaches 60s, the second incoming signal is triggered to add 1 to the minute counter.

Bit6-4	Bit3-0
0-5	0-9

Minute Count

The minute count also only needs 7 bits, and the counting range is the same as second count, so the implementation method is same.

Bit6-4	Bit3-0
0-5	0-9

Hour Count

Thehour countrangeis 0 to 24 with only 6 bits.

Bit5-4	Bit3-0
0-2	0-9

Day Count

The day count range is 1 to 31, only 6 bits, starting from 1, counting up to 28/29/30/31 according to the month and leap year, and triggering the day-in signal to add 1 to the month counter when the count is full.

Bit5-4	Bit3-0
0-3	0-9

Month Count

The month count range is 1 to 12, only 5 bits, triggering the month-in signal to add 1 to the year counter when the count is full.

Bit4	Bit3-0
0-1	0-9

Year Count

The year count range is 0 to 99 with 8 bits and cycle count from 0 to 99.

Bit7-4	Bit3-0
0-9	0-9

36.3.5 RTC Enable and Disable

RTCB can be enabled or disabled by software. RTCB is enabled by default after power-on.

36.3.6 RTC Time Setting

Software can set the RTCB registers directly at any time, usually it is recommended to set the time after XTLF is fully started; since the operation of setting the time register is asynchronous with the

RTCB timing, it is recommended that the software set the time after the second interrupt event and read out the time value for verification after the time setting.

Note that the hardware does not check the time validity, the software must ensure that the written BCD time is correct.

In order to improve the anti-interference capability, RTCB provides time write protection function. 0xACACACAC must be written to the write protection register before the time register can be rewritten. After the time setting is completed, the software can prohibit the writing of time registers by writing any other value to restore the write protection.

36.3.7 RTC Time Reading

Time reading mode 1:

- Read the current time register value.
- Read the current time register value again.
- If the contents of the 2 readings are same, it is the correct current time; if the contents of the two readings are not same, repeat the first two steps.

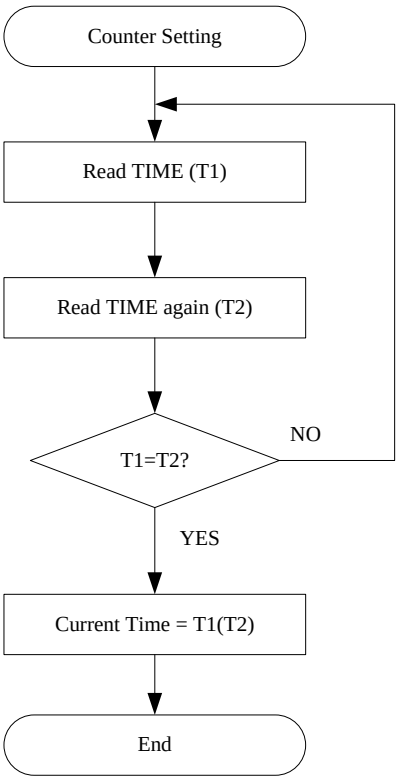


Figure 36–3 RTC Time Reading Flow Chart

Time reading mode 2:

Reading the time register immediately by software after 1s interrupt occurs, which ensures that the

correct current time value is read.

36.3.8 Leap Year Determination

The RTCB module will automatically determine the leap year within the range of 2000-2099 and process the BCD time.

36.3.9 RTC Timestamp

In order to support Tamper Detection, RTCB supports the time stamp function triggered by external IO events. The external IO trigger source is the input level change of PH15. To ensure the reliability of input detection, it is recommended to enable the Tamper digital filter. When using this function, configure the PH15 as a GPIO function, enable the RTC_STPCR.TAMPEN register, when any rising or falling edge set after filtering appears on the PH15, the RTC will automatically record the current time to the STPCLKRR and STPCALRR register groups, and at the same time generate corresponding flags, which can be used to generate interrupts or for software to query.

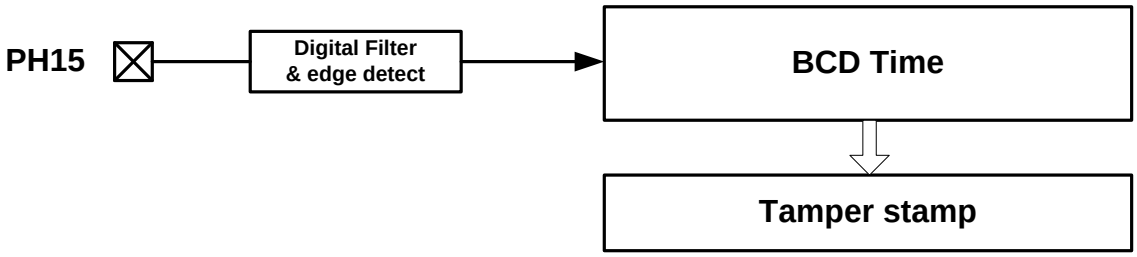


Figure 36–4 Timestamp

The timestamp records the event occurrence time only when the corresponding status register is 0. If the corresponding status is already 1, the corresponding event is ignored. Therefore, if multiple events occur, the timestamp only records the time when the first event occurred, unless the software clears the status register after the event occurs.

The tamper input signal is digitally filtered in the RTCB module. The filtering method is to use XTLF to sample continuously for 3 beats before it is considered legal. The digital filtering function can be enabled or disabled.

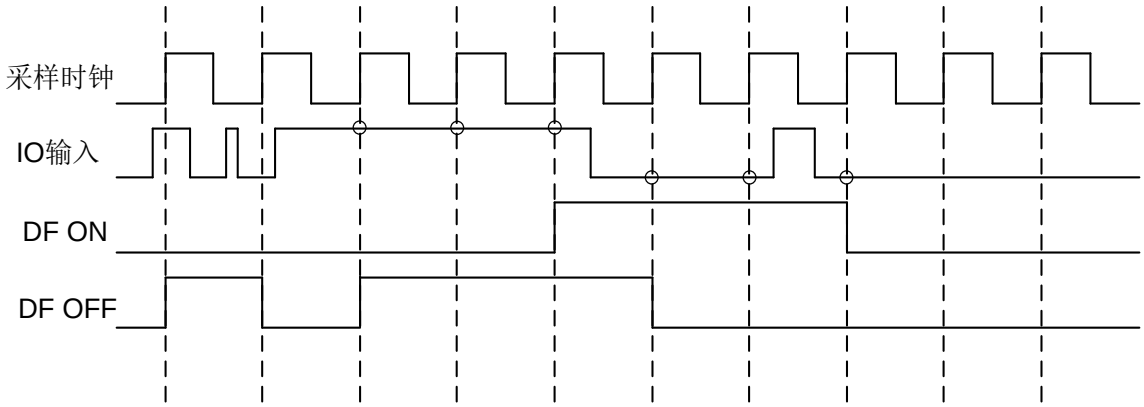


Figure 36–5 Tamper Input Digital Filter

36.3.10 Backup Register

RTCB contains five 32bit backup registers, a total of 20 bytes. In case of VDD power failure, the backup register is powered by VBAT and can still be used to maintain critical data.

When the Tamper event is detected, the chip can automatically erase the contents of the backup register. This function can be enabled or disabled through the TAMPEN register configuration.

The backup register supports tamper event clearing function. When a specific event occurs, the hardware automatically clears the contents of the backup register. The supported reset events include:

- Tamper pin event (Same as timestamp event)
- XTLF stop oscillation event

Whether to clear the register when the above event occurs can be enabled by the TAMPEN register, and the event source that triggers the backup register to be cleared can be selected through the TAMPSEL register.

36.4 Register

Offset	Name	Symbol
RTCB(Base Address: 0x4001F000)		
0x00000000	RTCBWrite Enable Register	RTCB_WER
0x00000004	RTCBInterrupt Enable Register	RTCB_IER
0x00000008	RTCBInterrupt Status Register	RTCB_ISR
0x0000000C	BCD Format Time Second Registers	RTCB_BCDSEC
0x00000010	BCD Format Time Minute Registers	RTCB_BCDMIN
0x00000014	BCD Format Time Hour Registers	RTCB_BCDHOUR
0x00000018	BCD Format Time Day Registers	RTCB_BCDDAY
0x0000001C	BCD Format Time Week Registers	RTCB_BCDWEEK
0x00000020	BCD Format Time Month Registers	RTCB_BCDMONTH
0x00000024	BCD Format Time Year Registers	RTCB_BCDYEAR
0x00000028	RTCB Alarm Register	RTCB_ALARM
0x0000002C	RTCB Time Mark Select	RTCB_TMSEL
0x00000030	RTCB Time Adjust Register	RTCB_ADJR
0x00000040	RTCB Control Register	RTCB_CR
0x00000048	RTCB Time Stamp Control Register	RTCB_STPCR
0x0000004C	RTCB Time Stamp Clock Record Register	RTCB_STPCLKRR
0x00000050	RTCB Time Stamp Calendar Record Register	RTCB_STPCALRR
0x00000070	RTCBBackup Register0	RTCBKR0
0x00000074	RTCB Backup Register1	RTCB_BKR1
0x00000078	RTCB Backup Register2	RTCB_BKR2
0x0000007C	RTCB Backup Register3	RTCB_BKR3
0x00000080	RTCB Backup Register4	RTCB_BKR4

36.4.1 RTCB Write Enable Register (RTCB_WER)

NAME	RTCB_WER							
Offset	0x00000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							WE
access	U-0							R/W-0

bit	name	functional description
31:1	-	RFU: **Reserved, read as 0**
0	WE	RTC Write Enable When the CPU writes 0xACACACAC to RTCWE, the CPU is allowed to write the initial value to the BCD time register of RTC, and then RTCWE is set to 1; when the CPU writes any value other than 0xACACACAC to RTCWE, the write protection is restored, and RTCWE is cleared to 0 at this time.

36.4.2 RTCB Interrupt Enable Register (RTCB_IER)

NAME	RTCB_IER							
Offset	0x00000004							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-		STP0IE	ADJ_IE	-			
access	U-0		R/W-0	R/W-0	U-0			
bit	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
name	-				SEC_IE	MIN_IE	HOUR_IE	DAY_IE
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:14	-	RFU: **Reserved, read as 0**
13	STP0IE	Time Stamp Interrupt Enable 1: Interrupt enable 0: Interrupt disable
12	ADJ_IE	Time Adjust Interrupt Enable 1: Interrupt enable 0: Interrupt disable
11:4	-	RFU: **Reserved, read as 0**
3	SEC_IE	1Hz Periodic Interrupt Enable 1: Interrupt enable 0: Interrupt disable
2	MIN_IE	Minute Interrupt Enable 1: Interrupt enable 0: Interrupt disable

bit	name	functional description
1	HOUR_IE	Hour Interrupt Enable 1: Interrupt enable 0: Interrupt disable
0	DAY_IE	Day Interrupt Enable 1: Interrupt enable 0: Interrupt disable

36.4.3 RTCB Interrupt Status Register (RTCB_ISR)

NAME	RTCB_ISR							
Offset	0x00000008							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-		STP_IF	ADJ_IF	-			
access	U-0		R/W-0	R/W-0	U-0			
bit	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
name	-				SEC_IF	MIN_IF	HOUR_IF	DAY_IF
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:14	-	RFU: **Reserved, read as 0**
13	STP_IF	Time Stamp Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
12	ADJ_IF	Time Adjust Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
11:4	-	RFU: **Reserved, read as 0**
3	SEC_IF	1Hz Periodic Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
2	MIN_IF	Minute Interrupt Flag, write 1 to clear 1: Interrupt set 0: No interrupt generates
1	HOUR_IF	Hour Interrupt Flag,write 1 to clear 1: Interrupt set

bit	name	functional description
		0: No interrupt generates
0	DAY_IF	Day Interrupt Flag,write 1 to clear 1: Interrupt set 0: No interrupt generates

36.4.4 BCD Format Time Second Registers (RTCB_BCDSEC)

NAME	RTCB_BCDSEC							
Offset	0x0000000C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	SEC						
access	U-0	R/W-xxx xxxx						

bit	name	functional description
31:7	-	RFU: **Reserved, read as 0**
6:0	SEC	Binary-Coded Decimal Format Seconds Register

36.4.5 BCD Format Time Minute Registers (RTCB_BCDMIN)

NAME	RTCB_BCDMIN							
Offset	0x00000010							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	MIN						
access	U-0	R/W-xxx xxxx						

bit	name	functional description
31:7	-	RFU: **Reserved, read as 0**
6:0	MIN	Binary-Coded Decimal Format Minutes Register

36.4.6 BCD Format Time Hour Registers (RTCB_BCDHOUR)

NAME	RTCB_BCDHOUR							
Offset	0x00000014							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		HOUR					
access	U-0		R/W-xx xxxx					

bit	name	functional description
31:6	-	RFU: **Reserved, read as 0**
5:0	HOUR	Binary-Coded Decimal Format Hours Register

36.4.7 BCD Format Time DAY Registers (RTCB_BCDDAY)

NAME	RTCB_BCDDAY							
Offset	0x00000018							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		DAY					
access	U-0		R/W-xx xxxx					

bit	name	functional description
31:6	-	RFU: **Reserved, read as 0**
5:0	DAY	Binary-Coded Decimal Format Date Register

36.4.8 BCD Format Time Week Registers (RTCB_BCDWEEK)

NAME	RTCB_BCDWEEK							
Offset	0x0000001C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-					WEEK		
access	U-0					R/W-xxx		

bit	name	functional description
31:3	-	RFU: **Reserved, read as 0**
2:0	WEEK	Binary-Coded Decimal Format Week Register

36.4.9 BCD Format Time Month Registers (RTCB_BCDMONTH)

NAME	RTCB_BCDMONTH							
Offset	0x00000020							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			MONTH				
access	U-0			R/W-x xxxx				

bit	name	functional description
31:5	-	RFU: **Reserved, read as 0**
4:0	MONTH	Binary-Coded Decimal Format Month Register

36.4.10 BCD Format Time Year Registers (RTCB_BCDYEAR)

NAME	RTCB_BCDYEAR							
Offset	0x00000024							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	YEAR							
access	R/W-xxxx xxxx							

bit	name	functional description
31:8	-	RFU: **Reserved, read as 0**
7:0	YEAR	Binary-Coded Decimal Format Year Register

36.4.11 RTCB Time Mark Select (RTCB_TMSEL)

NAME	RTCB_TMSEL							
Offset	0x0000002C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				TMSEL			
access	U-0				R/W-0000			

bit	name	functional description
31:4	-	RFU: **Reserved, read as 0**
3:0	TMSEL	This register configures the RTCB internal frequency signal output from the RTCOUT (PH15) pin. Frequency output select signal: 4'b0000: Disable output 4'b0001: RFU 4'b0010: Output second counter carry signal, high level width 1s 4'b0011: Output minute counter carry signal, high level width 1s 4'b0100: Output hour counter carry signal, high level width 1s 4'b0101: Output day counter carry signal, high level width 1s 4'b0110: RFU 4'b0111: Output 32 second square wave signal 4'b1000: Second time mark signal Others: Disable output

36.4.12 RTCB Time Adjust Register (RTCB_ADJR)

NAME	RTCB_ADJR							
Offset	0x00000030							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						ADJUST[9:8]	
access	U-0						R/W-xxx	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ADJUST[7:0]							
access	R/W-xxxxxxxx							

bit	name	functional description
31:12	--	RFU: **Reserved, read as 0**
9:0	ADJUST	LTBC compensation adjustment value, the highest bit is the sign bit, indicating the adjustment direction; the lower 9 bits are the adjustment value.

36.4.13 RTCB Control Register (RTCB_CR)

NAME	RTCB_CR							
Offset	0x00000040							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							EN
access	U-0							R/W-1

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
0	EN	RTC Enable Register, enabled by default 0: Disable RTCB 1: Enable RTCB

36.4.14 RTCB Time Stamp Control Register (RTCB_STPCR)

NAME	RTCB_STPCR							
Offset	0x00000048							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-						TAMPSEL	
access	U-0						R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DF	-				TSEDGE	TAMPEN	TSEN
access	R/W-0	U-0				R/W-0	R/W-0	R/W-0

bit	name	functional description
31:10	--	RFU: **Reserved, read as 0**

bit	name	functional description
9:8	TAMPSEL	Tamper clear backup register event source select 00: RFU 01: Tamper pin event, valid edge selected by TSEDGE 10: XTLF stop oscillation event 11: Tamper pin and XTLF stop oscillation are both effective ***Note:*** *If XTLF is selected to stop oscillation, the backup register will always be reset asynchronously when the oscillation stop occurs, and the software can no longer write data until XTLF is restored*
7	DF	Tamper PinDigital Filter enable 0: Disable digital filter 1: Enable digital filter
6:3	--	RFU: **Reserved, read as 0**
2	TSEDGE	Time Stamp Edge Select 0: Time stamp event and interrupt are generated on the rising edge of the input signal 1: Time stamp event and interrupt are generated on the falling edge of the input signal
1	TAMPEN	Tamper Event Clear Backup Register Enable 1: When a tamper event occurs, the backup register is automatically cleared 0: When a tamper event occurs, keep the backup register
0	TSEN	Time Stamp Triggered by PH15 Enable 1: Enable time stamp 0: Disable time stamp

36.4.15 RTCB Time Stamp Clock Record Register (RTCB_STPCLKRR)

NAME	RTCB_STPCLKRR							
Offset	0x0000004C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-		HRSTP0R					
access	U-0		RW-X					
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-	MINSTP0R						
access	U-0	RW-X						
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	SECSTP0R						

access	U-0	RW-X

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
21:16	HRSTP0R	Store the value of the BCD hour register after detecting the rising edge of PH15.
15	--	RFU: **Reserved, read as 0**
14:8	MINSTP0R	Store the value of the BCD minute register after detecting the rising edge of PH15.
7	--	RFU: **Reserved, read as 0**
6:0	SECSTP0R	Store the value of the BCD second register after detecting the rising edge of PH15.

36.4.16 RTCB Time Stamp Clock Record Register (RTCB_STPCALRR)

NAME	RTCB_STPCALRR							
Offset	0x00000050							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	YRSTP0R							
access	RW-X							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name				MONSTP0R				
access				RW-X				
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		DAYSTP0R					
access	U-0		RW-X					

bit	name	functional description
31:24	YRSTP0R	Store the value of the BCD year register after detecting the rising edge of PH15.
23:21	--	RFU: **Reserved, read as 0**
20:16	MONSTP0R	Store the value of the BCD month register after detecting the rising edge of PH15.
15:6	--	RFU: **Reserved, read as 0**
5:0	DAYSTP0R	Store the value of the BCD day register after detecting the rising edge of PH15.

36.4.17 RTCB Backup Register x (RTCB_BKRx)

A total of 20 bytes can be read and written without reset value.

NAME	RTCB_BKRx(x=0,1,2,3,4)							
Offset	0x00000070~80							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	BKP[31:24]							
access	R/W-xxxx xxxx							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	BKP[23:16]							
access	R/W-xxxx xxxx							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	BKP[15:8]							
access	R/W-xxxx xxxx							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	BKP[7:0]							
access	R/W-xxxx xxxx							

bit	name	functional description
31:0	BKP	RTC Backup Registers

37 LCD Display

37.1 Introduction

The LCD display driver module is used to drive segmented LCDs, capable of supporting 4, 6 and 8COM, with the maximum number of display segments being 176 segments (4COM), 252 segments (6COM) and 320 segments (8COM) respectively.

Main features:

- Maximum support for 8×40, 6×42, 4×44 display segments
- 1/3bias, 1/4bias
- 16 levels of gray scale adjustable
- LCD driver supports two modes: on-chip resistance type and off-chip capacitive type
- Support blinking function, and the blinking frequency is adjustable
- Support intermittent light-up function, lighton and off time can be configured
- Support all on and off functions
- Low power consumption, LCD driver can work in Active mode, Sleep mode and DeepSleep mode
- Support both Type A and Type B LCD driver waveforms (Configurable)
- Typical frame refresh rate 64Hz

37.2 Block Diagram

The block diagram of the LCD display drive module is shown in Figure 37-1:

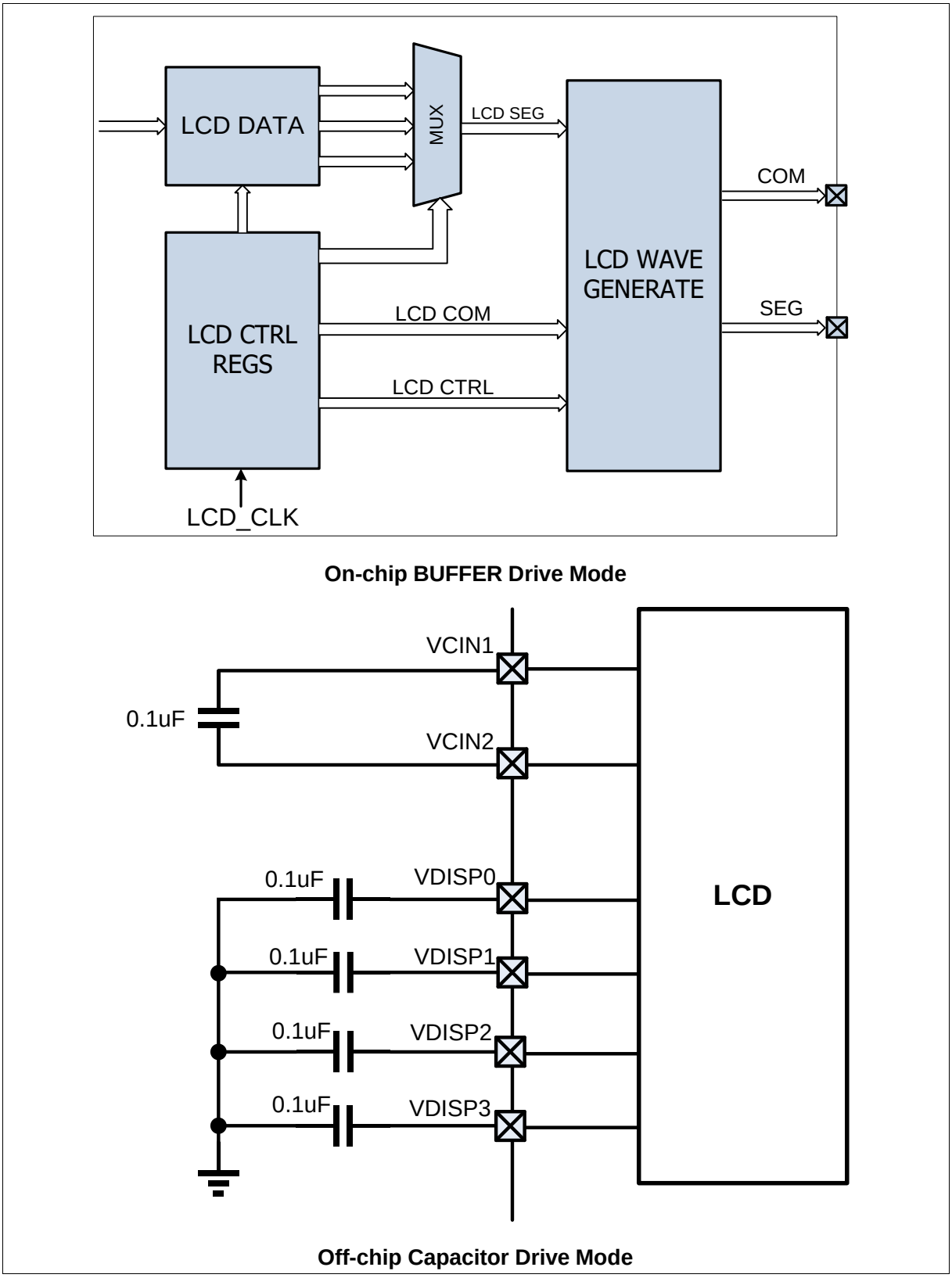


Figure 37–1 LCD Display Control Module Block Diagram

37.3 IO Configuration

The LCD driver circuit will occupy up to 48 GPIOs when operating. Before using the LCD, you need to set the pins used to the analog function (GPIOx_FCR=11) and enable the corresponding COMEN or SEGEN. If other analog functions are multiplexed on the same pin, software must guarantee all analog except for LCD will not use this pin.

37.4 Function Description

37.4.1 Operating Clock and Display Frame Rate

LCD driver circuit uses LSCLK to work, and its typical frequency is around 32KHz. By configuring the DF register, the frame rate of the LCD display can be set. The frame frequency is calculated by the following formula (note that DF cannot be 0).

COM number	Frame Rate (Hz)	
	A type waveform	B type waveform
4	LCD operating frequency/(4 ×DF[7:0] × 2)	LCD operating frequency/(4×DF[7:0] × 4)
6	LCD operating frequency/(6 ×DF[7:0] × 2)	LCD operating frequency/(6×DF[7:0] × 4)
8	LCD operating frequency/(8 ×DF[7:0] × 2)	LCD operating frequency/(8×DF[7:0] × 4)

Table 37-1 Frame Rate Calculation Formula

The following table gives an example of the relationship between the DF register values and frame rate. Frame rate is set to be around 60Hz typically.

Frame rate (Hz)	Operating clock (Hz)	4COM		6COM		8COM	
		A type	B type	A type	B type	A type	B type
50	32768	82	41	54	27	41	20
58	32768	70	35	47	24	35	17
64	32768	64	32	42	21	32	16
70	32768	58	29	39	20	29	14
75	32768	54	27	36	18	27	13

Table 37-2 Relationship Between Typical Frame Rate and DF

37.4.2 LCD Type A Scan Waveform

1/3 bias means that the LCD drive circuit can output 4 drive levels, which are obtained by evenly distributing the bias voltage. 1/4 bias means that the LCD drive circuit can output 5 drive levels, which are obtained by evenly distributing the bias voltage.

The VLCD bias voltage can be configured through the LCDBIAS register, and the maximum does not exceed the power supply voltage.

The following figure shows the type A drive waveforms of 4COM 1/3 bias, 6COM 1/3 bias and 1/4 bias, and 8COM 1/4 bias.

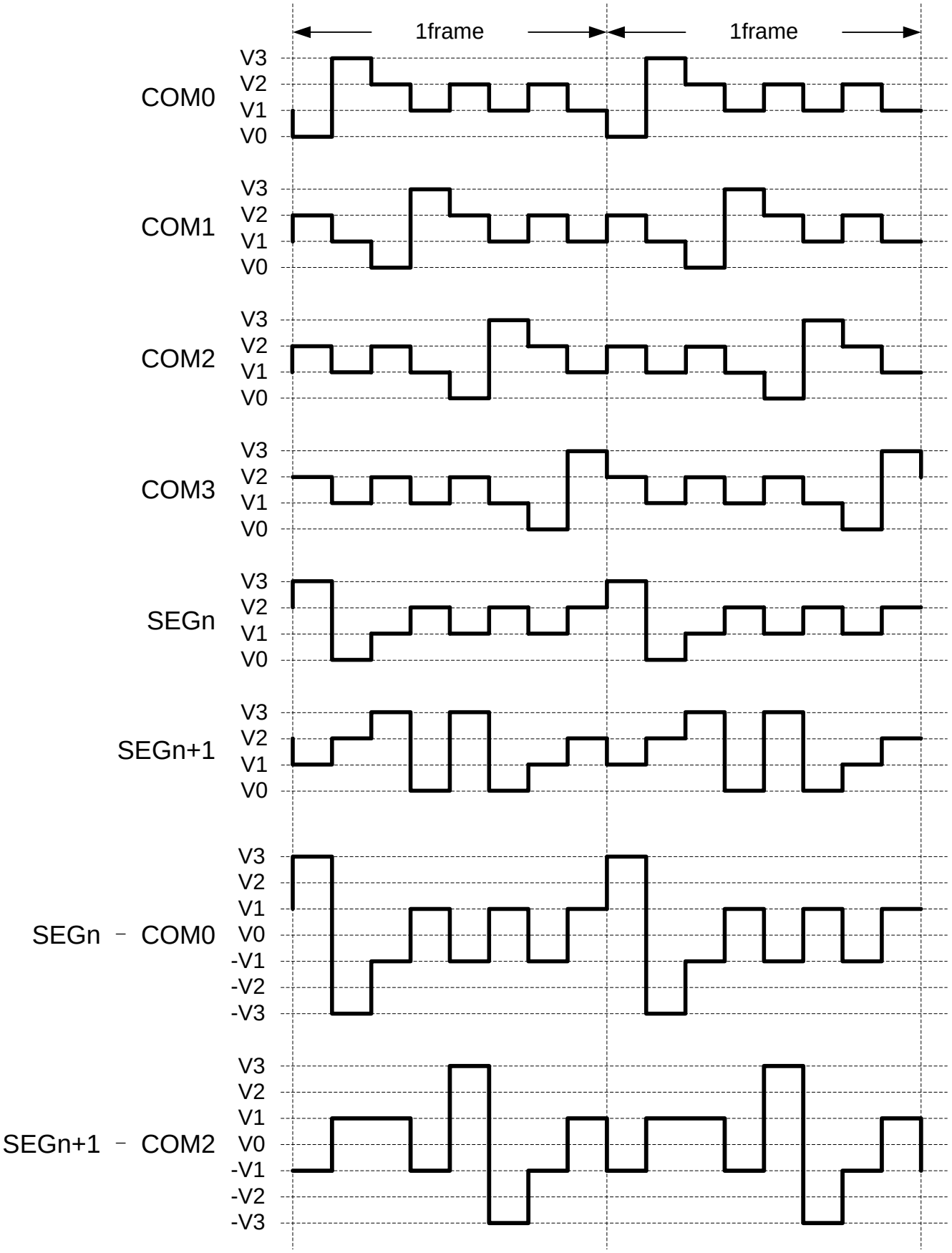


Figure 37–2 LCD Drive Waveform (1/4 duty, 1/3bias, type A)

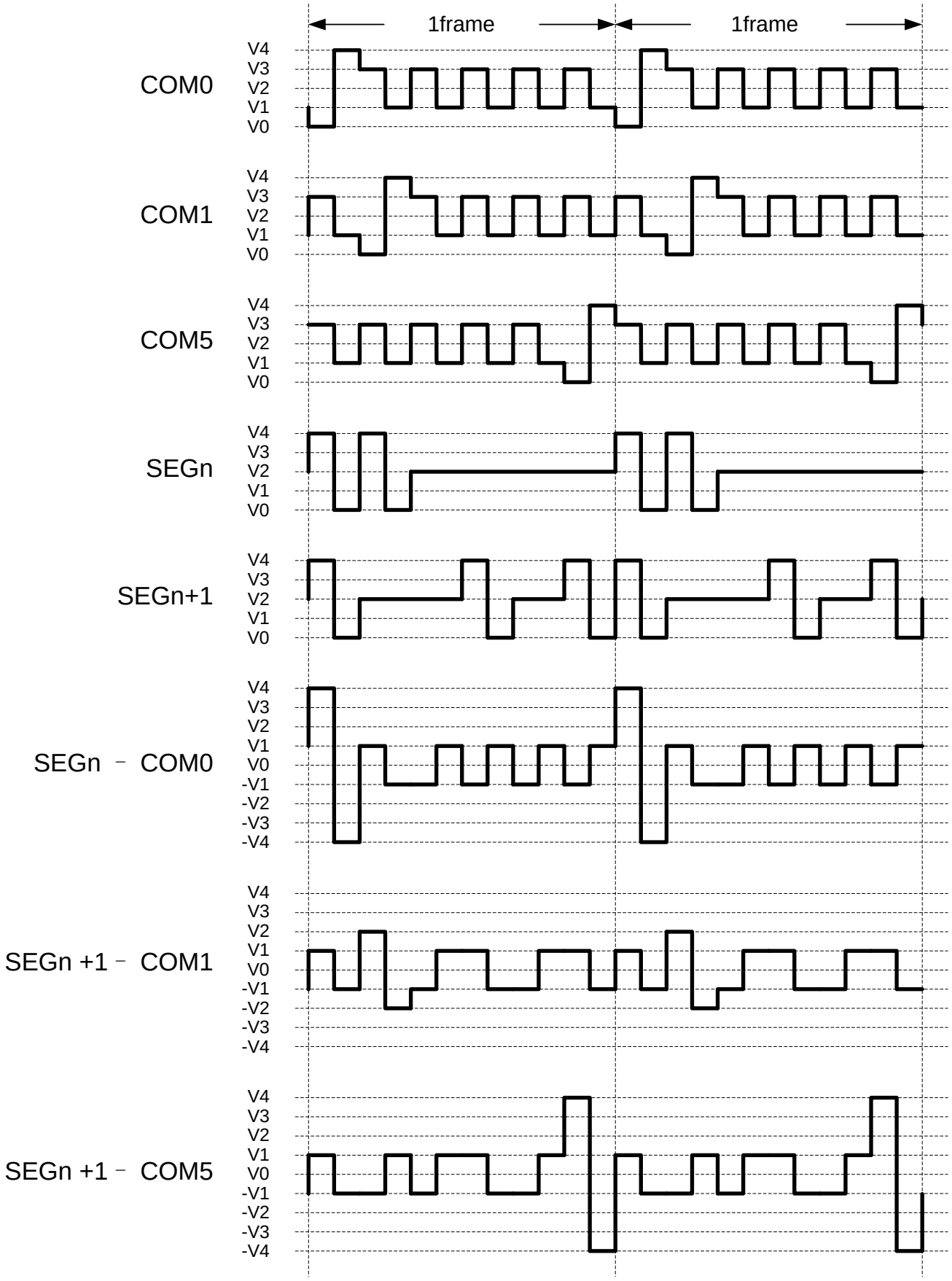


Figure 37–3 LCD Drive Waveform (1/6 duty, 1/4bias, type A)

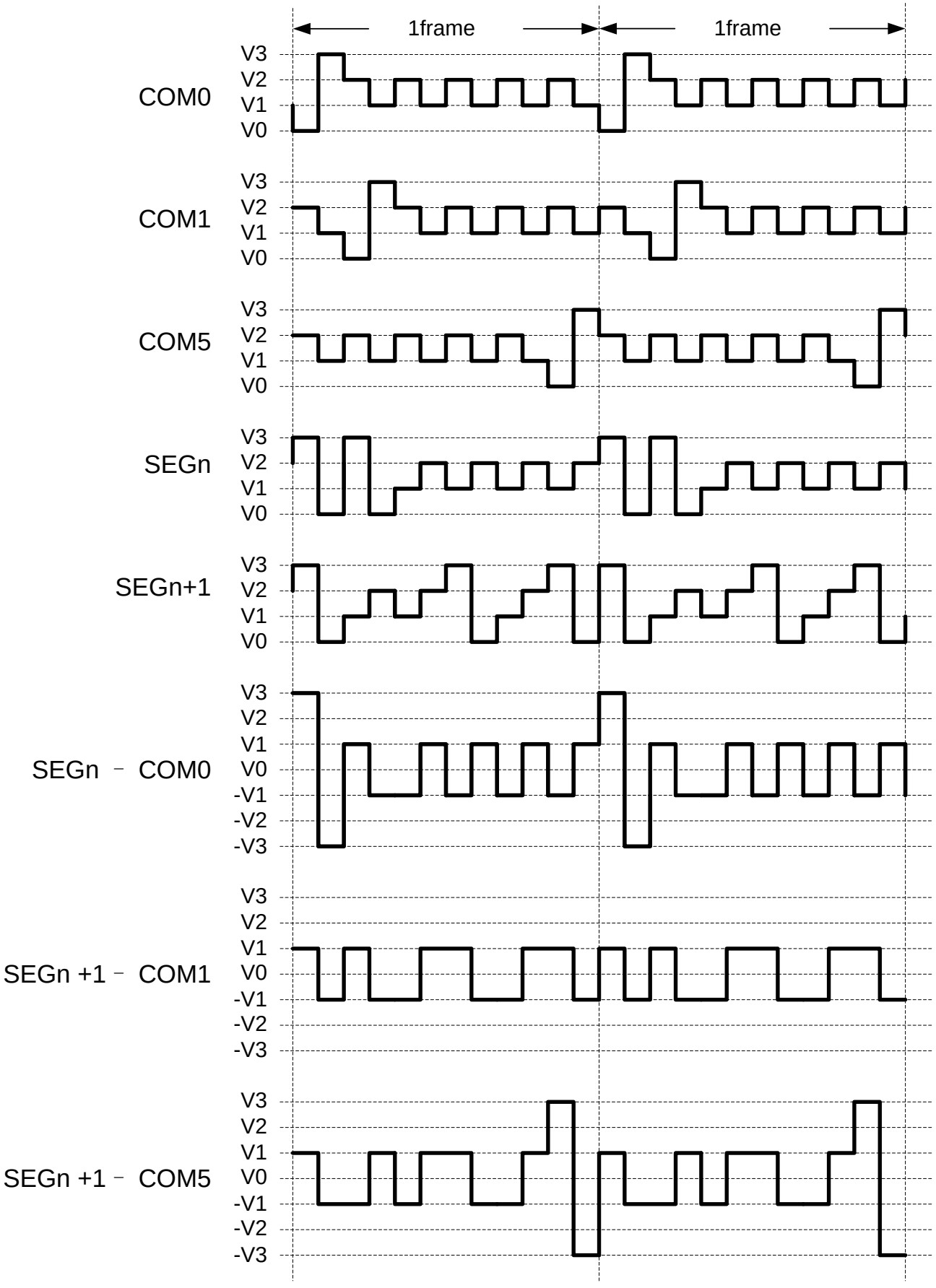


Figure 37–4 LCD Drive Waveform (1/6 duty, 1/3 bias, type A)

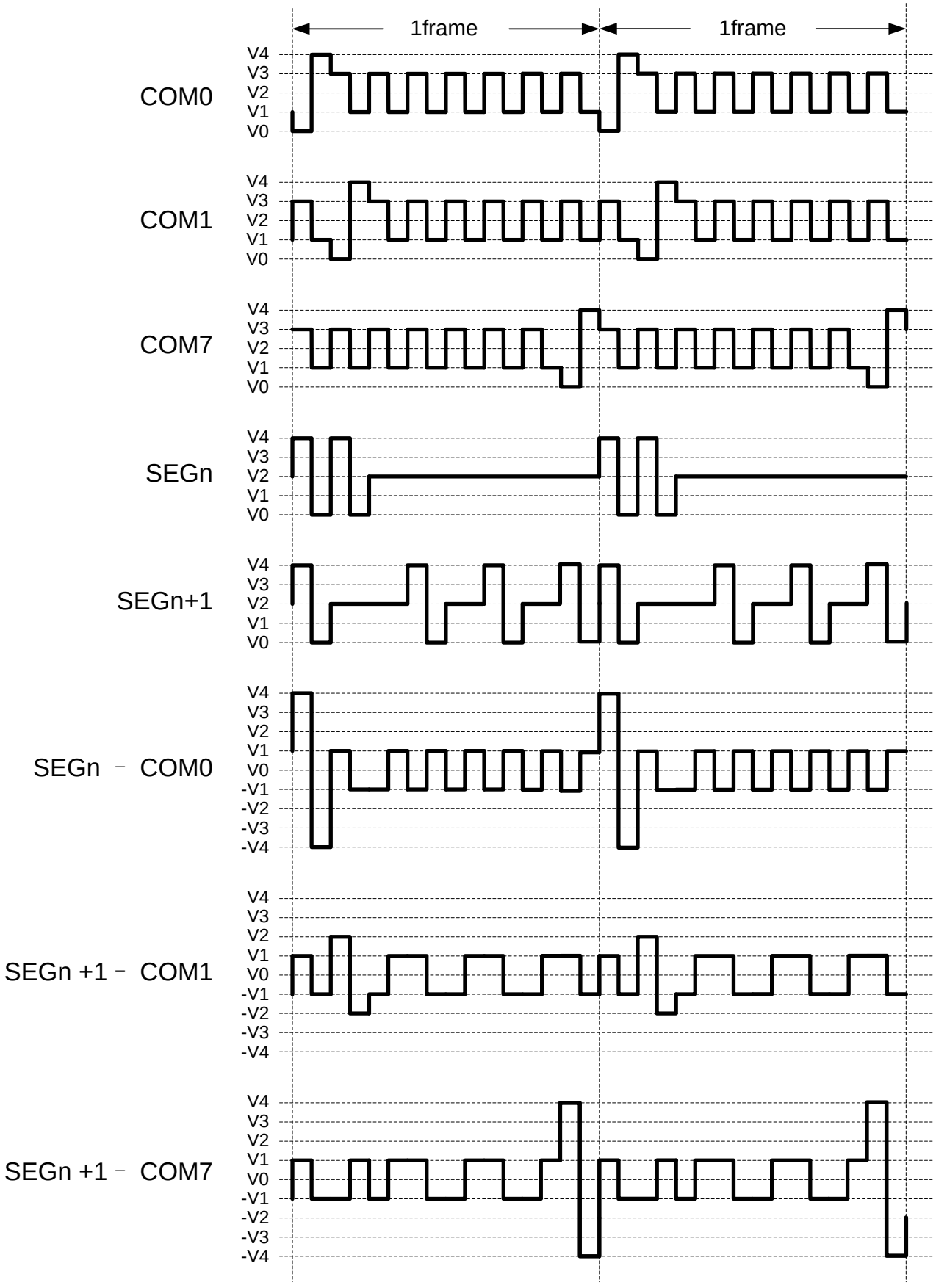


Figure 37–5 LCD Drive Waveform (1/8 duty, 1/4 bias, type A)

37.4.3 LCD Type B Scan Waveform

The following figure shows the type B driving waveforms of 4COM 1/3 bias, 6COM 1/3 bias and 1/4 bias, and 8COM 1/4 bias.

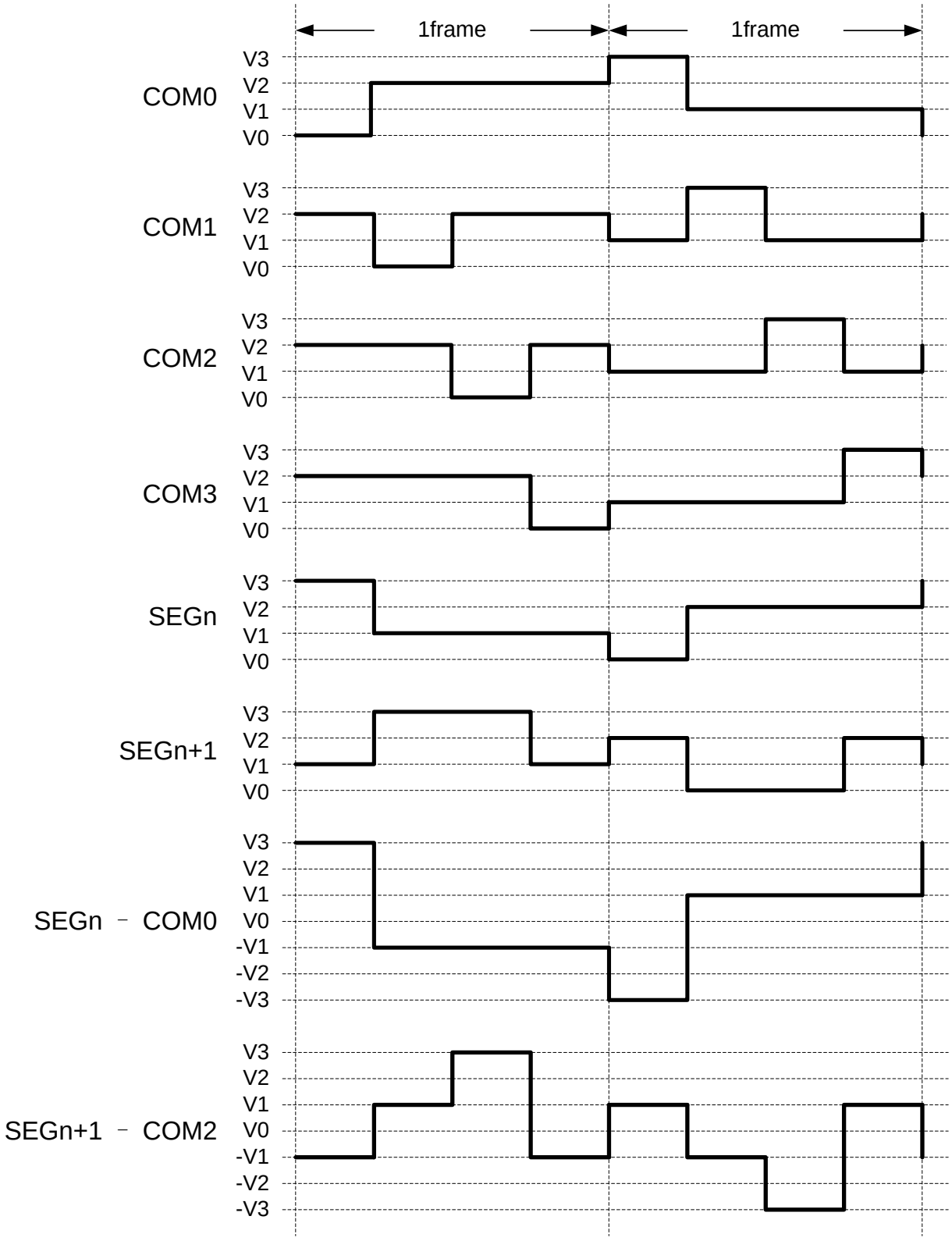


Figure 37–6 LCD Drive Waveform (1/4 duty, 1/3 bias, type B)

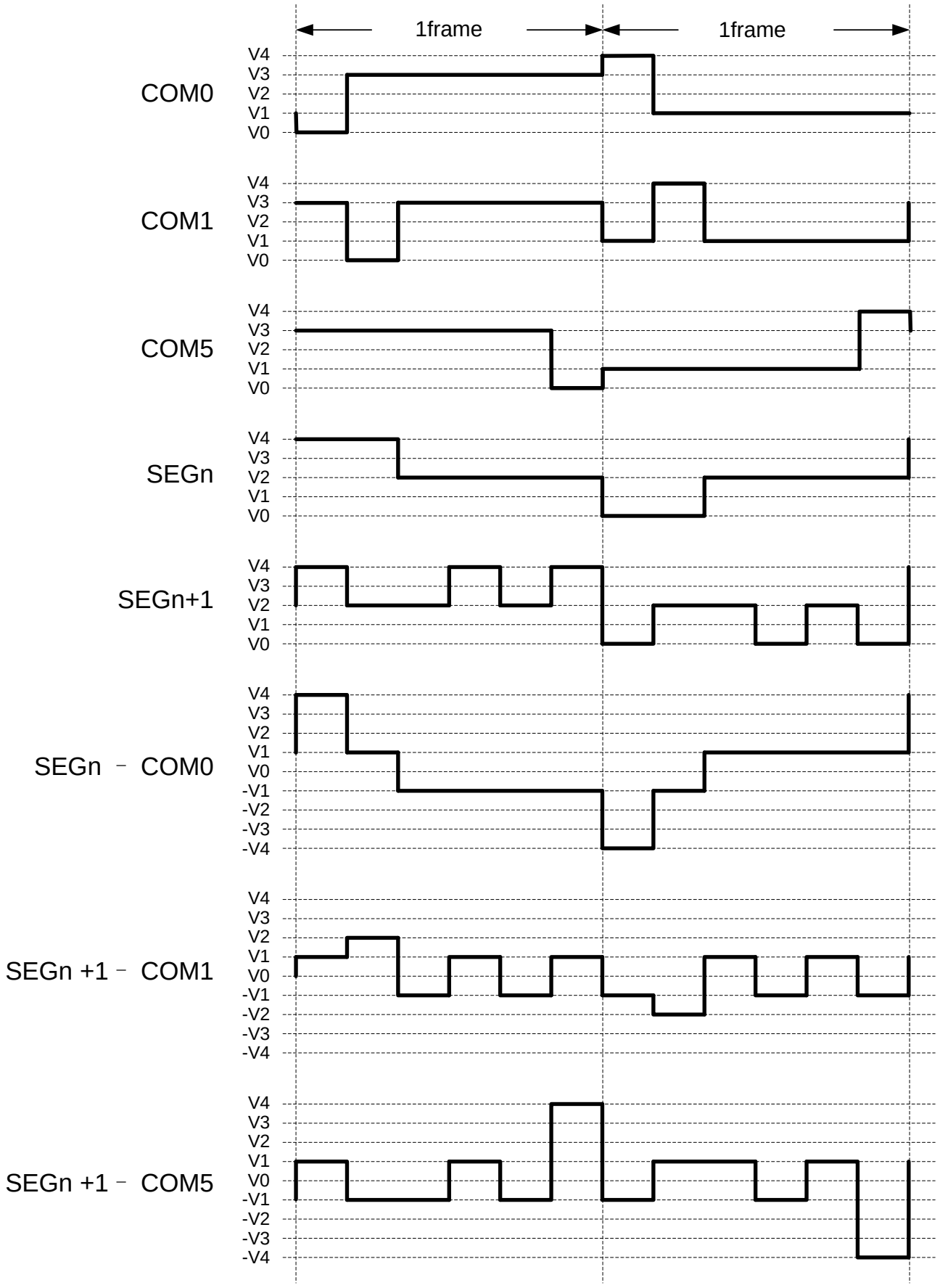


Figure 37–7 LCD Drive Waveform (1/6 duty, 1/4 bias, type B)

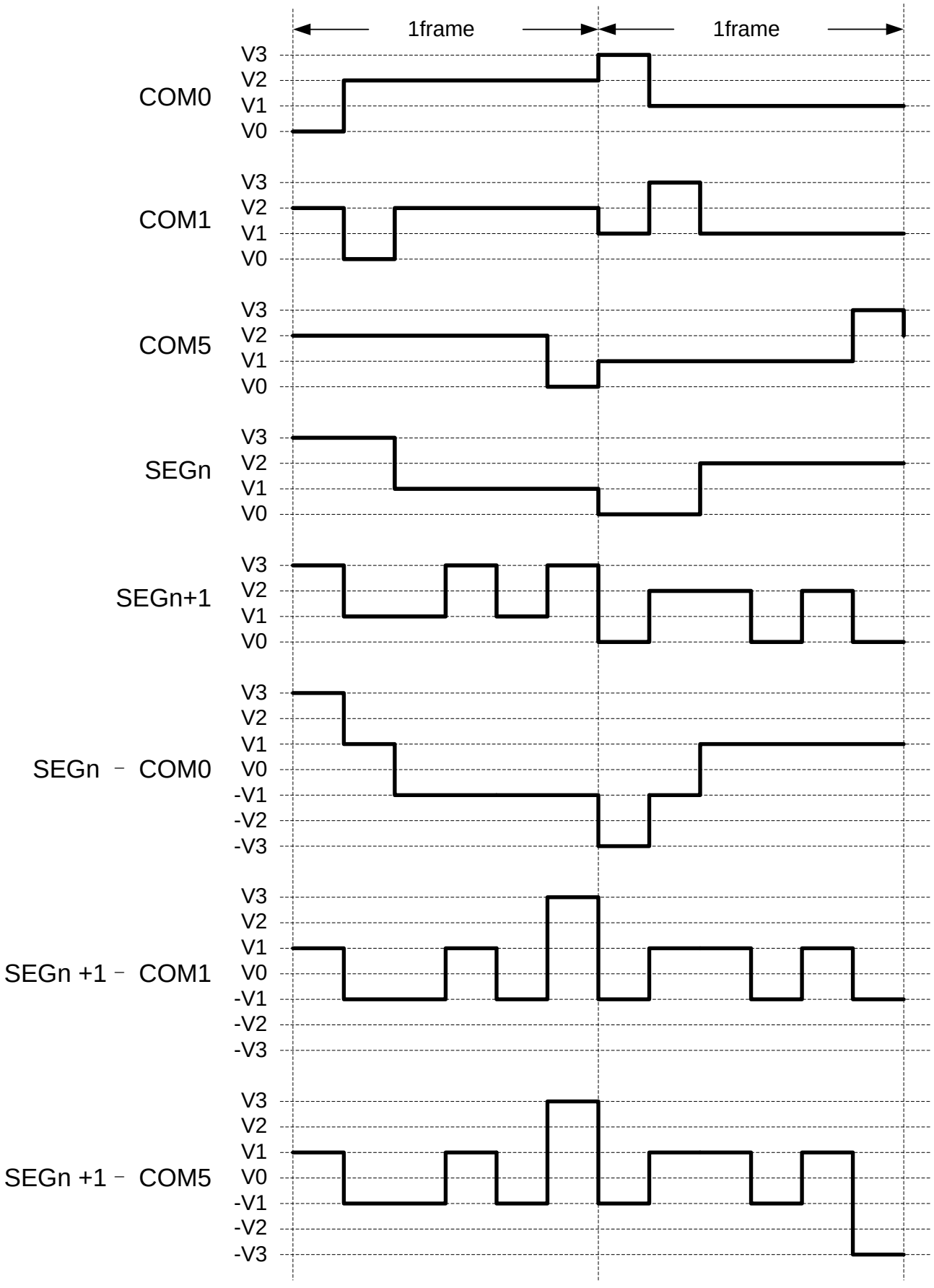


Figure 37–8 LCD Drive Waveform (1/6 duty, 1/3 bias, type B)

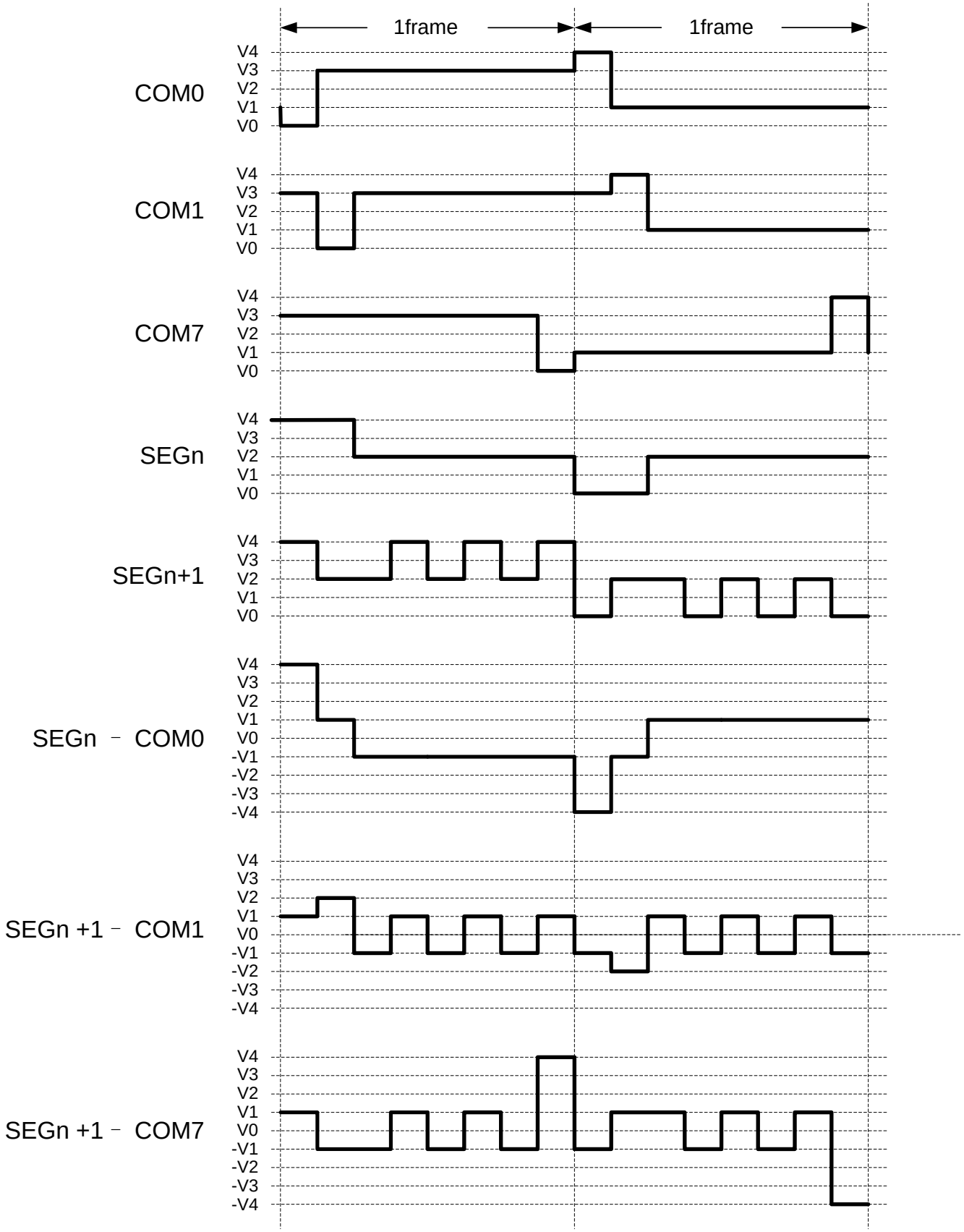


Figure 37–9 LCD Drive Waveform (1/8 duty, 1/4 bias, type B)

37.4.4 On-chip Buffer Drive Mode

The on-chip buffer drive mode generates equal division voltage from the power supply voltage through a voltage divider resistor, the divided voltage is buffered to enhance the drive strength, and the buffer output is connected to the waveform generation module to generate COM and SEG signals. This mode does not require off-chip equipment and has low power consumption.

The structure schematic is as follows.

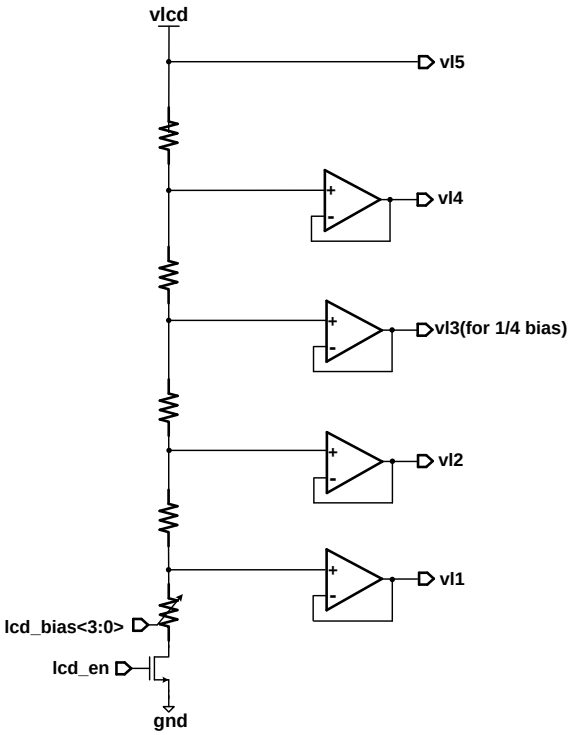


Figure 37–10 LCD On-chip Resistor Buffer Type Drive Circuit

The output impedance of the driver buffer is approximately 5Kohm.

The LCD drive voltage can be adjusted by configuring the LCDBIAS register, see section 37.4.7 "Bias Voltage Adjustment".

37.4.5 Off-chip Capacitor Drive Mode

The off-chip capacitor drive mode is driven by a switch and a capacitor. The capacitor is provided by an off-chip device. It is necessary to connect 0.1uF capacitors between the VCIN1 and VCIN2 pins, and between the VDISP0/1/2/3 pins and the ground. The output of the power supply voltage is equally divided through the off-chip capacitor and the on-chip switch. The off-chip capacitor drive mode has lower power consumption than the buffer drive mode, and requires the use of off-chip capacitors.

When using the off-chip capacitor drive mode, you need to set the ENMODE in the LCD drive mode register LCD_CR to 1, and configure the drive times and frequency through SC_CTRL and SCFSEL. Choosing a higher driving frequency and more driving times will obtain a better display effect, but the corresponding power consumption will also increase.

Note: *When using off-chip capacitor mode, the pins where VDISPx and VCINx are located must be configured as analog functions (FCR=11), and the GPIOx_ANEN register of the pin where VDISPx is located must be set, see 41.9.10 GPIO Analog Channel Enable Register.*

37.4.6 Display Flick Function

The software can set the FLICK bit in the display control register LCD_CR to 1 to enable the display flicking. After FLICK is enabled, the flicking frequency is determined according to the TON and TOFF register values. Before enabling the FLICK function, TON/TOFF should be set and DF should be set to turn on the display. If TON/TOFF is not set, its reset value is 0 and the display will flick at 64Hz. If the display is not turned on first, the FLICK setting is invalid and there will be no display.

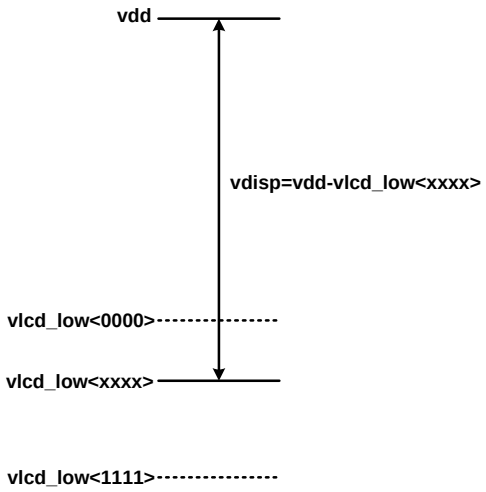
The minimum step of TON/TOFF is T_{step} = COM*DF[7:0]*2*16/32768Hz, the actual ON/OFF time is TON/TOFF * T_{step}. Display and off are synchronized with frame scan, i.e. off after a frame scan, or light up at the beginning of a frame, and the corresponding interrupt is given after off or light up. Since the end-of-frame signal is 64HZ, the count value of TON/TOFF should be the register setting value x16.

37.4.7 Bias Voltage Adjustment

The display voltage range of LCD output can be adjusted to suit different specifications of the LCD panel, and the output voltage range can be expressed as：

VDISP=VDD-VLCD_LOW

VLCD_LOW can be adjusted by LCDBIAS[3:0], LCDBIAS=0000 corresponds to the highest VLCD_LOW voltage, output voltage range VDISP=VDD-VLCD_LOW is minimum; LCDBIAS=1111 corresponds to the lowest VLCD_LOW voltage, output voltage range VDISP=VDD-VLCD_LOW is maximum, as shown in the following figure:



The application should be based on the actual LCD panel characteristics to select the appropriate VDISP voltage, you can refer to the table in 37.6.1 for configuration.

37.5 Low Power Mode

When the chip is in low power mode, the LCD can still work normally. When the main power is off, the LCD stops working.

Mode	Description
LPRUN	Display normally
Sleep/DeepSleep	Display normally
VBAT	Cannot display normally

Table 37-3 LCD and Low Power Mode

37.6 Register

Offset	Name	Symbol
LCD(Base Address: 0x40010C00)		
0x00000000	LCD Control Register	LCD_CR
0x00000004	LCD Test Register	LCD_TEST
0x00000008	LCD Frequency Control Register	LCD_FCR
0x0000000C	LCD Flick Time Register	LCD_FLKT
0x00000014	LCD Interrupt Enable Register	LCD_IER
0x00000018	LCD Interrupt Status Register	LCD_ISR
0x00000024	LCD Data Buffer Registers 0	LCD_DATA0
0x00000028	LCD Data Buffer Registers 1	LCD_DATA1
0x0000002C	LCD Data Buffer Registers 2	LCD_DATA2
0x00000030	LCD Data Buffer Registers 3	LCD_DATA3
0x00000034	LCD Data Buffer Registers 4	LCD_DATA4
0x00000038	LCD Data Buffer Registers 5	LCD_DATA5
0x0000003C	LCD Data Buffer Registers 6	LCD_DATA6
0x00000040	LCD Data Buffer Registers 7	LCD_DATA7
0x00000044	LCD Data Buffer Registers 8	LCD_DATA8
0x00000048	LCD Data Buffer Registers 9	LCD_DATA9
0x00000050	LCD COM Enable Register	LCD_COMEN
0x00000054	LCD SEG Enable Register0	LCD_SEGEN0
0x00000058	LCD SEG Enable Register 1	LCD_SEGEN1

37.6.1 LCD Control Register (LCD_CR)

NAME	LCD_CR							
Offset	0x00000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name		SCFSEL			SC_CTRL		IC_CTRL	
access		R/W-000			R/W-00		R/W-01	
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ENMODE	FLICK	-		LCDBIAS			
access	R/W-1	R/W-0	U-0		R/W-1110			
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		BIASMD	ANTIPOLAR	WFT	LMUX		LCDEN
access	U-0		R/W-0	R/W-0	R/W-0	R/W-00		R/W-0

bit	name	functional description
30:23	--	RFU: **Reserved, read as 0**
22:20	**SCFSEL**	Capacitor drive frequency select 000 = Frequency is frame rate *COM number 001 = Frequency isLSCLK/8 010 = Frequency isLSCLK/16 011 = Frequency isLSCLK/32 100 = Frequency isLSCLK/64 101 = Frequency isLSCLK/128 110 = Frequency isLSCLK/256 111= Frequency isLSCLK /512 **Note:** When the 110 or 111 gear is selected, if the frequency is lower than the frame rate*2*COM, the output result is the same as the 000 gear; when the frame frequency is set to a high level, using the 000 gear may cause the drive pulse frequency to be too high and not normal Output.
19:18	**SC_CTRL**	Drive Mode Control, in off-chip capacitor drive mode 00 = Single drive 01 = Drive 2 times continuously 10 = Drive 4 times continuously, when the SC frequency is greater than or equal to 4KHz, this option is also multidrive 11 = Multidrive
17:16	**IC_CTRL**	Input Bias Current Control 00 =Maximum current 01 = Second largest current 10 = Second smallest current 11 = Minimum Current
15	**ENMODE**	LCD Enabling Mode 0= Off-chip capacitor drive 1 = On-chip resistive drive
14	**FLICK**	LCD Flick Enable 1: Display flick, flick frequency is set by TON and TOFF registers 0: Disable flick
13:12	**--**	RFU: **Reserved, read as 0**
11:8	**LCDBIAS**	LCD Bias Voltage Select, for displaying grayscale control
7:6	**--**	RFU: **Reserved, read as 0**
5	**BIASMD**	Bias Mode 1: 1/3 Bias 0: 1/4 Bias
4	**ANTIPOLAR**	Anti-Polarization 1: COM and SEG grounded with LCD off 0: COM and SEG floating with LCD off
3	**WFT**	Waveform Format 1: B type waveform

bit	name	functional description
		0: A type waveform
2:1	**LMUX**	Segment Line Mux 00: 4COM 01: 6COM 10/11: 8COM
0	**LCDEN**	LCD Enable 1: LCD enable 0: LCD disable

LCDBIAS[3:0]	Internal bias voltage at different VDD(V)			
	5	4.5	3.6	3.0
0000	2.74	2.47	1.97	1.64
0001	2.83	2.54	2.03	1.69
0010	2.92	2.62	2.10	1.75
0011	3.01	2.71	2.17	1.81
0100	3.12	2.80	2.24	1.87
0101	3.23	2.90	2.32	1.94
0110	3.35	3.01	2.41	2.01
0111	3.47	3.13	2.50	2.08
1000	3.61	3.25	2.60	2.17
1001	3.76	3.39	2.71	2.26
1010	3.93	3.53	2.83	2.35
1011	4.10	3.69	2.95	2.46
1100	4.30	3.87	3.09	2.58
1101	4.51	4.06	3.25	2.71
1110	4.75	4.27	3.42	2.85
1111	5.00	4.50	3.60	3.00

37.6.2 LCD Test Register (LCD_TEST)

NAME	LCD_TEST							
Offset	0x00000004							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

name	LCCTRL	-	TESTEN
access	R/W-0	U-0	R/W-0

bit	name	functional description
31:8	---	RFU: **Reserved, read as 0**
7	**LCCTRL**	LCDtest control bit, valid only in test mode (Line Constant Control) COMand SEG output levels are determined by the pin output data register in test mode. See the table below for the results of SEGor COM output under different settings.
6:0	---	RFU: **Reserved, read as 0**
0	**TESTEN**	Test mode Enable 1 = LCDtest mode enable. In LCD test mode, the LCD pin statically outputs analog DC level, and all register settings related to dynamic scan time and scan waveform are invalid. 0 = Normal operation mode, test mode is invalid, the relevant test register control is invalid.

Pin output level in test mode:

LCCTRL	DISPDATA	COM pin output voltage	SEG pin output voltage
		1/3bias	1/3bias
0	0	V3	V2
0	1	V1	V4
1	0	V2	V3
1	1	V4	V1

37.6.3 LCD Frequency Control Register (LCD_FCR)

NAME	LCD_FCR							
Offset	0x00000008							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DF[7:0]							
access	R/W-00000000							

bit	name	functional description

bit	name	functional description
31:8	---	RFU: **Reserved, read as 0**
7:0	**DF**	Display Frequency

37.6.4 LCD Flick Time Register (LCD_FLKT)

NAME	LCD_FLKT							
Offset	0x0000000C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	TOFF[7:0]							
access	R/W-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TON[7:0]							
access	R/W-00000000							

bit	name	functional description
31:16	---	RFU: **Reserved, read as 0**
15:8	**TOFF**	Display-Off Time TOFF minimum step is T_{step} = COM*DF[7:0]*2*16/32768Hz, the actual OFF time is TOFF * T_{step}
7:0	**TON**	Display-On Time TON minimum step isT_{step} = COM*DF[7:0]*2*16/32768Hz, the actual ON time is TON * T_{step}

37.6.5 LCD Interrupt Enable Register (LCD_IER)

NAME	LCD_IER							
Offset	0x00000014							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							

access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						DONIE	DOFFIE
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:2	-	RFU: **Reserved, read as 0**
1	**DONIE**	Display-On Interrupt Enable 1: Display-on interrupt enable 0: Display-on interrupt disable
0	**DOFFIE**	Display-OFF Interrupt Enable 1: Display-offinterrupt enable 0: Display-offinterrupt disable

37.6.6 LCD Interrupt Status Register (LCD_ISR)

名称	LCD_ISR							
Offset	0x00000018							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						DONF	DOFFIF
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:2	-	RFU: **Reserved, read as 0**
1	**DONIF**	Display-On Interrupt Flag, hardware set, write 1 to clear
0	**DOFFIF**	Display-OFF Interrupt Flag, hardware set, write 1 to clear

37.6.7 LCD Data Buffer Registers x (LCD_DATAx)

There are 8 32bit display data registers in the LCD display module. All are readable and writable, and the reset value is 0.

NAME	LCD_DATAx(x=0,1,2,3,4,5,6,7,8,9)							
Offset	0x00000024 + x*0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	DSDA[31:24]							
access	R/W-0000 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	DSDA[23:16]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DSDA[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DSDA[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:0	DSDA	Display Data

37.6.7.1 4COM Data Buffer Register

NAME	4COM Data Buffer Register							
LCD_DATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM0	SEG30 COM0	SEG29 COM0	SEG28 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
LCD_DATA 1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

NAME	4COM Data Buffer Register							
LCD_DATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG31 COM1	SEG30 COM1	SEG29 COM1	SEG28 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
LCD_DATA 2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM2	SEG30 COM2	SEG29 COM2	SEG28 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
LCD_DATA 3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM3	SEG30 COM3	SEG29 COM3	SEG28 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
LCD_DATA 4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM0	SEG38 COM0	SEG37 COM0	SEG36 COM0	SEG35 COM0	SEG34 COM0	SEG33 COM0	SEG32 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG35 COM1	SEG34 COM1	SEG33 COM1	SEG32 COM1	SEG43 COM0	SEG42 COM0	SEG41 COM0	SEG40 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG43 COM1	SEG42 COM1	SEG41 COM1	SEG40 COM1	SEG39 COM1	SEG38 COM1	SEG37 COM1	SEG36 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2
LCD_DATA 5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG35 COM3	SEG34 COM3	SEG33 COM3	SEG32 COM3	SEG43 COM2	SEG42 COM2	SEG41 COM2	SEG40 COM2

NAME	4COM Data Buffer Register							
LCD_DATA0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG43 COM3	SEG42 COM3	SEG41 COM3	SEG40 COM3	SEG39 COM3	SEG38 COM3	SEG37 COM3	SEG36 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

37.6.7.2 6COM Data Buffer Register

NAME	6COM Data Buffer Register							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM0	SEG30 COM0	SEG43 COM0	SEG42 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
LCD_DATA 1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM1	SEG30 COM1	SEG43 COM1	SEG42 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
LCD_DATA 2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2

NAME	6COM Data Buffer Register							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM2	SEG30 COM2	SEG43 COM2	SEG42 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
LCD_DATA 3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM3	SEG30 COM3	SEG43 COM3	SEG42 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
LCD_DATA 4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM4	SEG6 COM4	SEG5 COM4	SEG4 COM4	SEG3 COM4	SEG2 COM4	SEG1 COM4	SEG0 COM4
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM4	SEG14 COM4	SEG13 COM4	SEG12 COM4	SEG11 COM4	SEG10 COM4	SEG9 COM4	SEG8 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM4	SEG22 COM4	SEG21 COM4	SEG20 COM4	SEG19 COM4	SEG18 COM4	SEG17 COM4	SEG16 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM4	SEG30 COM4	SEG43 COM4	SEG42 COM4	SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
LCD_DATA 5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM5	SEG6 COM5	SEG5 COM5	SEG4 COM5	SEG3 COM5	SEG2 COM5	SEG1 COM5	SEG0 COM5
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM5	SEG14 COM5	SEG13 COM5	SEG12 COM5	SEG11 COM5	SEG10 COM5	SEG9 COM5	SEG8 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM5	SEG22 COM5	SEG21 COM5	SEG20 COM5	SEG19 COM5	SEG18 COM5	SEG17 COM5	SEG16 COM5
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

NAME	6COM Data Buffer Register							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG31 COM5	SEG30 COM5	SEG43 COM5	SEG42 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
LCD_DATA 6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM0	SEG38 COM0	SEG37 COM0	SEG36 COM0	SEG35 COM0	SEG34 COM0	SEG33 COM0	SEG32 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG37 COM1	SEG36 COM1	SEG35 COM1	SEG34 COM1	SEG33 COM1	SEG32 COM1	SEG41 COM0	SEG40 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2	SEG41 COM1	SEG40 COM1	SEG39 COM1	SEG38 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG33 COM3	SEG32 COM3	SEG41 COM2	SEG40 COM2	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2
LCD_DATA 7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG41 COM3	SEG40 COM3	SEG39 COM3	SEG38 COM3	SEG37 COM3	SEG36 COM3	SEG35 COM3	SEG34 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG39 COM4	SEG38 COM4	SEG37 COM4	SEG36 COM4	SEG35 COM4	SEG34 COM4	SEG33 COM4	SEG32 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG37 COM5	SEG36 COM5	SEG35 COM5	SEG34 COM5	SEG33 COM5	SEG32 COM5	SEG41 COM4	SEG40 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
					SEG41 COM5	SEG40 COM5	SEG39 COM5	SEG38 COM5

37.6.7.3 8COM Data Buffer Register

NAME	8COM Data Buffer Register							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

NAME	8COM Data Buffer Register							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG43 COM0	SEG42 COM0	SEG41 COM0	SEG40 COM0	SEG27 COM0	SEG26 COM0	SEG25 COM0	SEG24 COM0
LCD_DATA 1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM1	SEG42 COM1	SEG41 COM1	SEG40 COM1	SEG27 COM1	SEG26 COM1	SEG25 COM1	SEG24 COM1
LCD_DATA 2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM2	SEG22 COM2	SEG21 COM2	SEG20 COM2	SEG19 COM2	SEG18 COM2	SEG17 COM2	SEG16 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM2	SEG42 COM2	SEG41 COM2	SEG40 COM2	SEG27 COM2	SEG26 COM2	SEG25 COM2	SEG24 COM2
LCD_DATA 3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM3	SEG22 COM3	SEG21 COM3	SEG20 COM3	SEG19 COM3	SEG18 COM3	SEG17 COM3	SEG16 COM3
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM3	SEG42 COM3	SEG41 COM3	SEG40 COM3	SEG27 COM3	SEG26 COM3	SEG25 COM3	SEG24 COM3
LCD_DATA 4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM4	SEG6 COM4	SEG5 COM4	SEG4 COM4	SEG3 COM4	SEG2 COM4	SEG1 COM4	SEG0 COM4

NAME	8COM Data Buffer Register							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM4	SEG14 COM4	SEG13 COM4	SEG12 COM4	SEG11 COM4	SEG10 COM4	SEG9 COM4	SEG8 COM4
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM4	SEG22 COM4	SEG21 COM4	SEG20 COM4	SEG19 COM4	SEG18 COM4	SEG17 COM4	SEG16 COM4
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM4	SEG42 COM4	SEG41 COM4	SEG40 COM4	SEG27 COM4	SEG26 COM4	SEG25 COM4	SEG24 COM4
LCD_DATA 5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM5	SEG6 COM5	SEG5 COM5	SEG4 COM5	SEG3 COM5	SEG2 COM5	SEG1 COM5	SEG0 COM5
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM5	SEG14 COM5	SEG13 COM5	SEG12 COM5	SEG11 COM5	SEG10 COM5	SEG9 COM5	SEG8 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM5	SEG22 COM5	SEG21 COM5	SEG20 COM5	SEG19 COM5	SEG18 COM5	SEG17 COM5	SEG16 COM5
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM5	SEG42 COM5	SEG41 COM5	SEG40 COM5	SEG27 COM5	SEG26 COM5	SEG25 COM5	SEG24 COM5
LCD_DATA 6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM6	SEG6 COM6	SEG5 COM6	SEG4 COM6	SEG3 COM6	SEG2 COM6	SEG1 COM6	SEG0 COM6
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM6	SEG14 COM6	SEG13 COM6	SEG12 COM6	SEG11 COM6	SEG10 COM6	SEG9 COM6	SEG8 COM6
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG23 COM6	SEG22 COM6	SEG21 COM6	SEG20 COM6	SEG19 COM6	SEG18 COM6	SEG17 COM6	SEG16 COM6
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG43 COM6	SEG42 COM6	SEG41 COM6	SEG40 COM6	SEG27 COM6	SEG26 COM6	SEG25 COM6	SEG24 COM6
LCD_DATA 7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG7 COM7	SEG6 COM7	SEG5 COM7	SEG4 COM7	SEG3 COM7	SEG2 COM7	SEG1 COM7	SEG0 COM7
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG15 COM7	SEG14 COM7	SEG13 COM7	SEG12 COM7	SEG11 COM7	SEG10 COM7	SEG9 COM7	SEG8 COM7
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

NAME	8COM Data Buffer Register							
LCD_DATA 0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG23 COM7	SEG22 COM7	SEG21 COM7	SEG20 COM7	SEG19 COM7	SEG18 COM7	SEG17 COM7	SEG16 COM7
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG31 COM7	SEG30 COM7	SEG29 COM7	SEG28 COM7	SEG27 COM7	SEG26 COM7	SEG25 COM7	SEG24 COM7
LCD_DATA 8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM0	SEG38 COM0	SEG37 COM0	SEG36 COM0	SEG35 COM0	SEG34 COM0	SEG33 COM0	SEG32 COM0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG39 COM1	SEG38 COM1	SEG37 COM1	SEG36 COM1	SEG35 COM1	SEG34 COM1	SEG33 COM1	SEG32 COM1
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG39 COM2	SEG38 COM2	SEG37 COM2	SEG36 COM2	SEG35 COM2	SEG34 COM2	SEG33 COM2	SEG32 COM2
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG39 COM3	SEG38 COM3	SEG37 COM3	SEG36 COM3	SEG35 COM3	SEG34 COM3	SEG33 COM3	SEG32 COM3
LCD_DATA 9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SEG39 COM4	SEG38 COM4	SEG37 COM4	SEG36 COM4	SEG35 COM4	SEG34 COM4	SEG33 COM4	SEG32 COM4
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	SEG39 COM5	SEG38 COM5	SEG37 COM5	SEG36 COM5	SEG35 COM5	SEG34 COM5	SEG33 COM5	SEG32 COM5
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
	SEG39 COM6	SEG38 COM6	SEG37 COM6	SEG36 COM6	SEG35 COM6	SEG34 COM6	SEG33 COM6	SEG32 COM6
	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
	SEG39 COM7	SEG38 COM7	SEG37 COM7	SEG36 COM7	SEG35 COM7	SEG34 COM7	SEG33 COM7	SEG32 COM7

37.6.8 LCD COM Enable Register (LCD_COMEN)

NAME	LCD_COMEN							
Offset	0x00000050							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				COMEN[3:0]			
access	U-0				R/W-0000			

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3:0	COMEN	COM Enable 1: COM output enable 0: COM output disable ***Note:*** *Control COM0~3 enable*

37.6.9 LCD SEG Enable Register 0 (LCD_SEGEN0)

NAME	LCD_SEGEN0							
Offset	0x00000054							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	SEGEN31	SEGEN30	SEGEN29	SEGEN28	SEGEN27	SEGEN26	SEGEN25	SEGEN24
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	SEGEN23	SEGEN22	SEGEN21	SEGEN20	SEGEN19	SEGEN18	SEGEN17	SEGEN16
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	SEGEN15	SEGEN14	SEGEN13	SEGEN12	SEGEN11	SEGEN10	SEGEN9	SEGEN8
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SEGEN7	SEGEN6	SEGEN5	SEGEN4	SEGEN3	SEGEN2	SEGEN1	SEGEN0
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:0	SEGENx	LCD SEG output enable control 1: SEG output enable 0: SEG output disable ***Note:*** *SEGEN28~31 simultaneously control COM4~7 enable*

37.6.10 LCD SEG Enable Register 1 (LCD_SEGEN1)

NAME	LCD_SEGEN1							
Offset	0x00000058							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-	-	-	-	SEGEN43	SEGEN42	SEGEN41	SEGEN40
access	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SEGEN39	SEGEN38	SEGEN37	SEGEN36	SEGEN35	SEGEN34	SEGEN33	SEGEN32
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:12	--	RFU: **Reserved, read as 0**
11:0	SEGENx	LCD SEG output enable control 1: SEG output enable 0: SEG output disable

38 Analog to Digital Converter (ADC)

38.1 Introduction

The FM33LG0 has a built-in 2Msps 12bit SAR-ADC, which can measure temperature, battery voltage or other DC signals. The main features are:

- Operating voltage 1.6 to 5.5V
- Input voltage range 0 to VREF+
- Flexible choice of reference source
- Maximum sampling rate 2Msps (F_{ADC}=32Mhz)
- Maximum 20 external input channels:
  - 14 fast channels, which can form 7 differential input pairs
  - 6 slow channels, only support single-ended input
  - The slow channels have built-in buffers, which can be used for weak drive signal measurement
- 7 internal sampling channels
  - Temperature sensor
  - Internal reference measurement
  - VBAT/3
  - VDD/3
  - DAC output
  - OPA output
- Configurable sample-hold time
- Support single conversion and continuous conversion
- Support DMA
- Support over-sampling hardware average, up to 16bit output (256 times average)
- Ultralow power architecture

38.2 Block Diagram

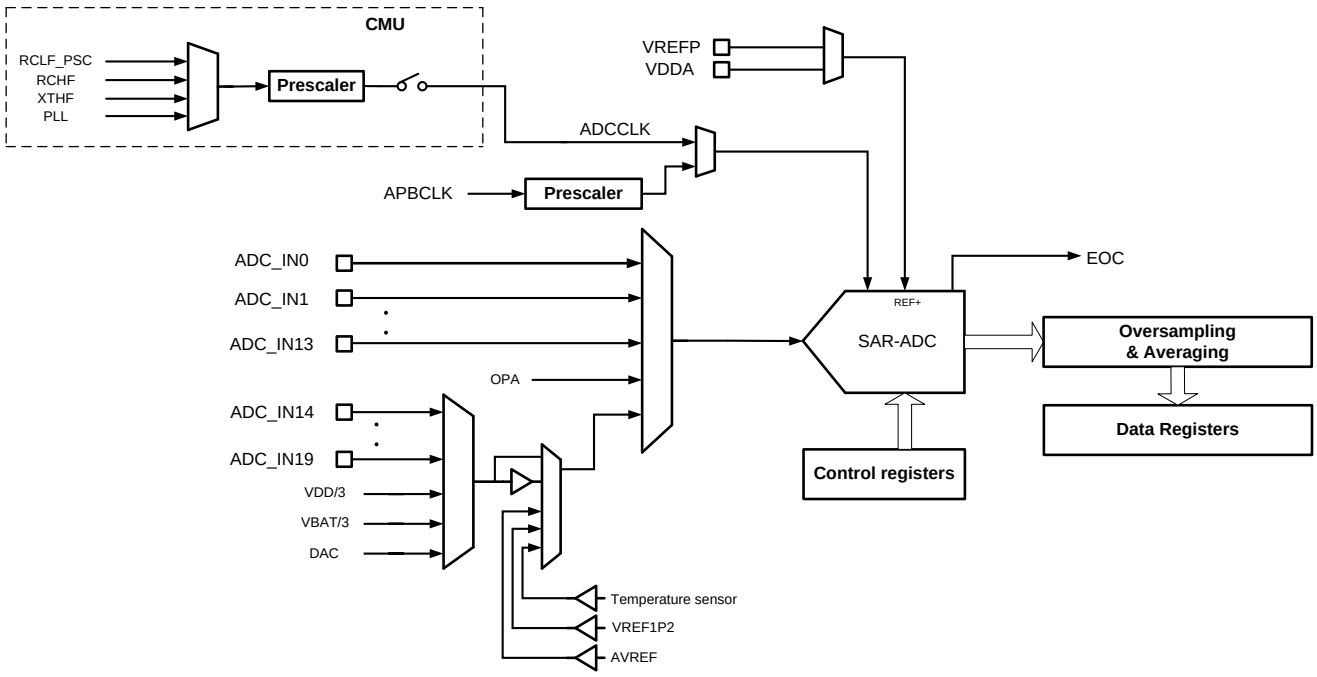


Figure 38–1 ADC Block Diagram

38.3 Input Channel

The ADC supports 7 internal channels and 20 external channels.

Channel	IO	Description
ADC_IN0	PD11	External fast channel, single-ended or differential input Among them, the differential pair combination relationship is as follows: ADC_IN0 – ADC_IN7 ADC_IN1 – ADC_IN8 ADC_IN2 – ADC_IN9 ADC_IN3 – ADC_IN10 ADC_IN4 – ADC_IN11 ADC_IN5 – ADC_IN12 ADC_IN6 – ADC_IN13
ADC_IN1	PD1	
ADC_IN2	PD3	
ADC_IN3	PD5	
ADC_IN4	PA13	
ADC_IN5	PA0	
ADC_IN6	PC7	
ADC_IN7	PD0	
ADC_IN8	PD2	
ADC_IN9	PD4	
ADC_IN10	PD6	
ADC_IN11	PA14	
ADC_IN12	PA1	
ADC_IN13	PC8	
ADC_IN14	PC9	External slow channel, single-ended input
ADC_IN15	PC10	
ADC_IN16	PC11	
ADC_IN17	PC12	
ADC_IN18	PD10	
ADC_IN19	PE9	
VBAT/3	N/A	VBAT sampling channel
VDD/3		VDD sampling channel
VREF1P2		Internal 1.2V reference source sampling channel
TS		Temperature sensor sampling channel
AVREF		Internal 1.0V reference source sampling channel
DAC		DAC output sampling channel
OPA		OPA output sampling channel

Table 38-1 ADC Input Channel Allocation

Note that not all package types can support the maximum number of channels. The corresponding relationship between the product model and the number of ADC channels is shown in the table below.

Product number	Package	External channel	Internal channel
FM33LG0x8	LQFP80	20	7
FM33LG0x6	LQFP64	18	7

38.4 Single-ended and Differential Inputs

ADC supports single-ended input and differential input modes.

In single-ended mode, ADC converts the voltage value of a single input pin to ground, and the input amplitude range is 0~+VREF+. In order to avoid possible input distortion caused by the risk of waveform clipping, it is generally recommended that the maximum amplitude of the input signal does not exceed 0.95*VREF+.

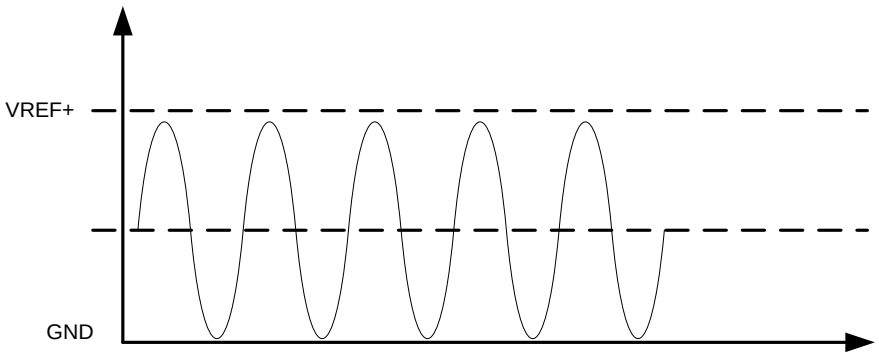


Figure 38–2 Single-ended Input

In the differential input mode, the ADC converts the difference between the differential input pin pair VIN+ and VIN-, the input signal is (VIN+)-(VIN-), and the input range is -VREF~+VREF. Using the differential input method can obtain a better common-mode noise suppression effect. Therefore, when the sampled signal source is far away from the ADC input pins, it is recommended to use the differential pair method to improve the signal-to-noise ratio.

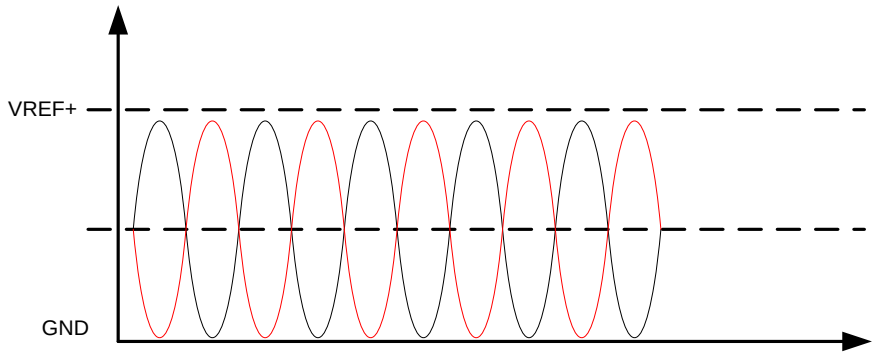


Figure 38–3 Differential Input

Correspondence between differential mode code word and input signal level:

(ADC conversion value-2048)*LSB*2 = (VIN+)–(VIN-)

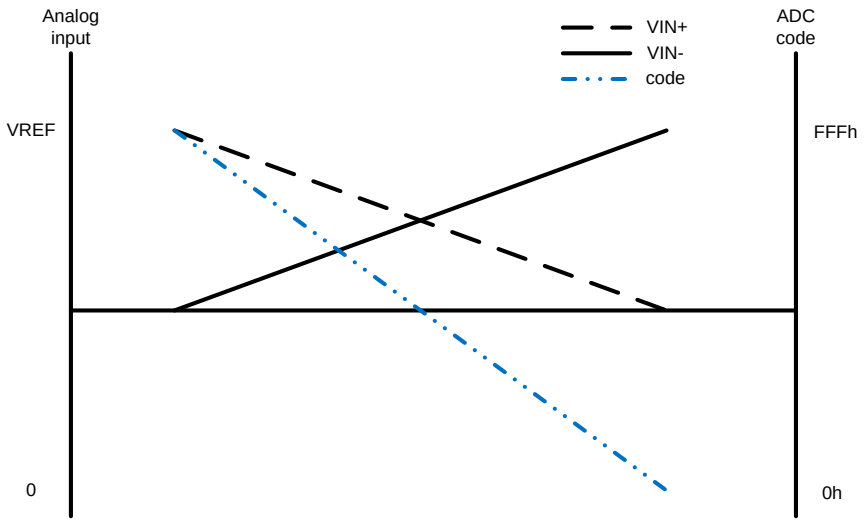


Figure 38–4 The Relationship Between Differential Input Signal and Codeword

When multiple channels are automatically converted, the interleaving of differential channels and single-ended channels can be realized. For example, if you want to continuously sample the differential channel composed of ADC_IN0 and ADC_IN7 and the single-ended channel of ADC_IN2, the software should adopt the following configuration method:

- Configure ADC_CHER register, ECH0, ECH2 are set to 1
- Configure ADC_DCR register, AINS[0] is set to 1
- Choose single conversion or continuous conversion
- Start ADC to start conversion

After the ADC is started, the ADC_IN0 and ADC_IN7 channels are enabled at the same time, and the ADC samples and converts the differential input signal; after completion, the ADC_IN0 and ADC_IN7 channels are closed, and the ADC_IN2 channel is enabled, and the ADC samples and converts the single-ended input signal.

Note: *The differential pair only needs to enable the low-order numbered channel. In the above example, the differential pair IN0 and IN7 only need to enable the ADC_IN0 channel and set AINS[0]*

38.5 Working Timing

ADC has two timings: offset calibration and normal operation.

It is recommended to perform a calibration first after power-on to obtain better performance. There is no need to re-calibrate after calibration, unless the chip is reset globally, or the power supply voltage and temperature have changed significantly. Offset calibration is started by software setting CALEN. The calibration process includes multiple sampling conversion cycles. The number of cycles is configured by the OSCAL_CYCLE register. Generally, it is recommended to use the default configuration, that is, 64 ADCCLK cycles are required to complete the calibration. After the calibration is completed, the EOCAL flag register is set.

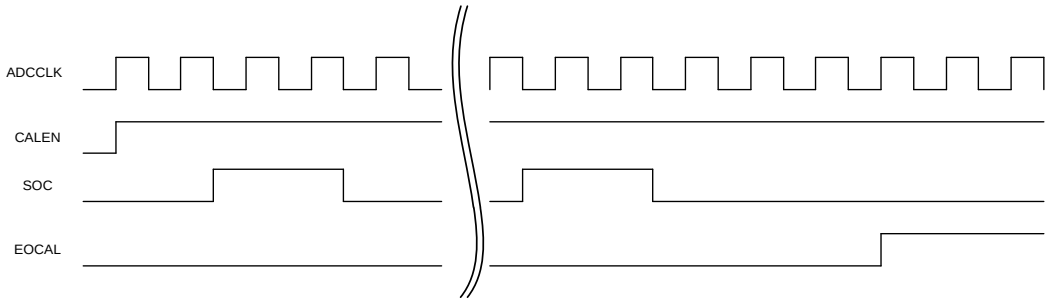


Figure 38–5 ADC Calibration Timing

During sampling conversion, ADC sampling is started through the SOC signal,the SOC high level width controls the ADC sampling time. After the SOC becomes low, the conversion is started, the conversion cycle is 14 ADC_CLK cycles. The sampling time is configurable,the shortest is 2 ADC_CLK and the longest is 512. ADC_CLK. The EOC (End of Conversion) signal is generated after the ADC sampling conversion is completed. The ADC controller sets the EOC flag register after sampling the EOC, and can generate an interrupt event or DMA request according to the configuration.

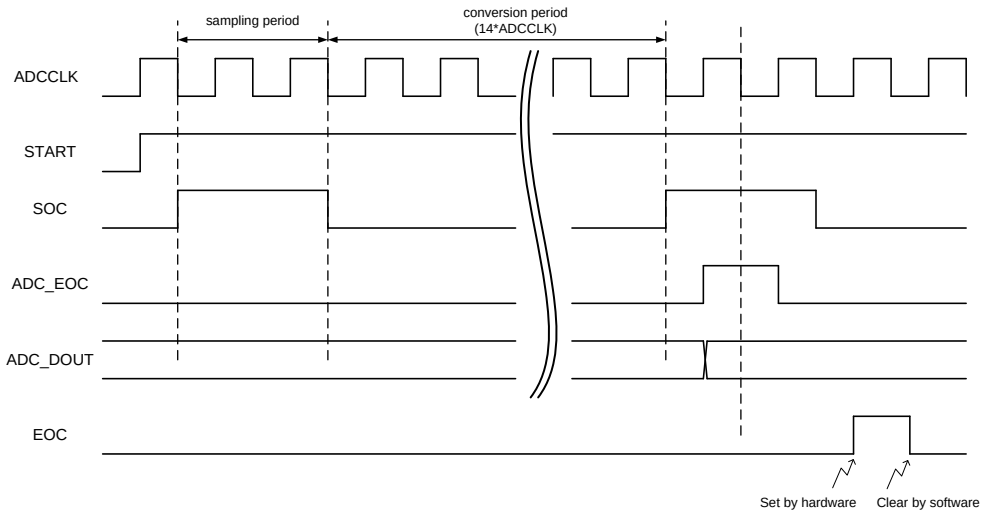


Figure 38–6 ADC Sampling Conversion Timing

The timing diagram of the multi-channel conversion sequence is as follows. The ADC completes a round of sampling conversion for all enabled input channels in turn to become a conversion sequence. After all the enabled input channels are sampled and converted sequentially, the ADC controller sets the EOS (End of Sequence) flag register and can generate interrupt events according to the configuration.

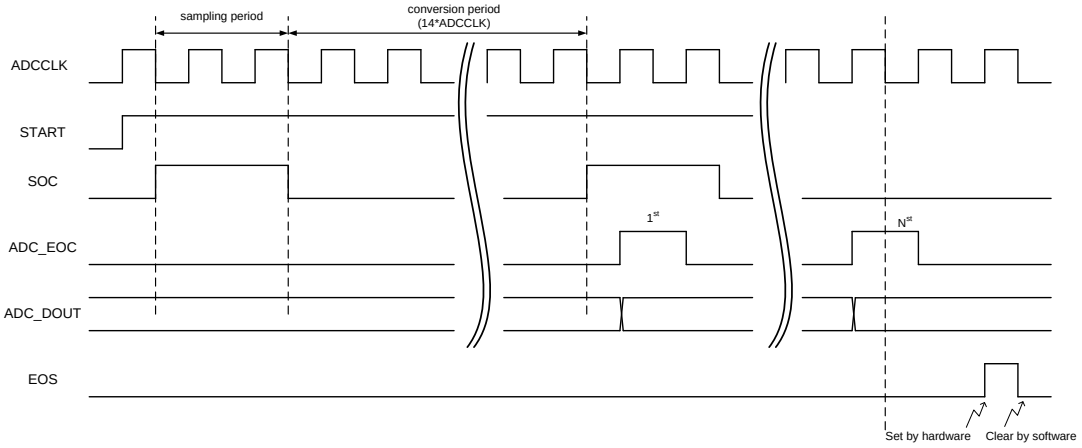


Figure 38–7 ADC Sampling Sequence Timing

The SOC signal is aligned with the falling edge of ADC_CLK, and the EOC signal generated by the ADC is aligned with the rising edge of ADC_CLK.

When ADC_CLK is 16Mhz and the sampling time is configured as 2*ADCCLK, the conversion rate is 1Msps.

When ADC_CLK is 32Mhz and the sampling time is configured as 2*ADCCLK, the conversion rate is 2Msps.

38.6 Functional Description

38.6.1 Use VDDA as a Reference

ADC generally uses the power supply voltage as the reference voltage. When the power supply voltage changes, the conversion value corresponding to a specific input signal level will also change. In order to get accurate absolute voltage, there are two solutions.

Use high-precision internal reference as reference signal

- First, the converted value obtained by measuring the voltage of VREF1P2 under the condition of VDDA=3.0V at 30°C ambient temperature is saved in Flash
- In the actual application of the chip, since the current VDDA voltage is not known, the ADC first measures VREF1P2 to obtain the converted value VREFINT_DATA; the current actual VDDA can be obtained by the following formula:

$$VDDA = \frac{VREFINT\_CAL}{VREFINT\_DATA} \times 3V$$

- Assuming that the ADC sampling value for a certain input channel is ADC_DATA, the actual voltage of a certain input channel (12bit output) can be obtained by the following formula:

$$V_{CHANNEL} = \frac{VREFINT\_CAL \times ADC\_DATA}{VREFINT\_DATA \times 4095} \times 3V$$

- In this way, there is no need to know the actual voltage value of each chip VREFINT, only the ratio of the current VREFINT sampled value to the factory test value needs to be calculated, but the voltage accuracy and temperature accuracy of VDDA during the test need to be guaranteed to minimize the error

Note that VREFIN_CAL is the conversion value of 1.22V under the 3V reference voltage, then this value should be around 1665, that is, 11bit unsigned effective value, ADC_DATA is 12bit unsigned number, VREFINT_CAL*ADC_DATA*3 maximum is 25bit; VREFINT_DATA maximum conversion value 1.22V under 1.8V is about 2775, which is equivalent to a 12bit unsigned number, so VREFINT_DATA*4095 is up to 24bit, and calculation can be achieved through 32bit division.

Use fast internal reference as reference signal

There is also a quickly established internal reference source inside the chip, its output voltage

is 1.0V. Each chip is calibrated when it leaves the factory, and the accuracy is 1.0V+/-0.5%. Compared with VREF1P2, the fast reference source has a fast start-up speed. Under typical conditions, start-up and output voltage establishment can be completed within 5us, while VREF1P2 needs more than 1ms. However, the accuracy of the fast reference source is relatively greatly affected by temperature, and the change in the whole temperature range can reach +/-4%, so it is suitable for occasions where absolute accuracy is not required.

- ADC samples and converts the fast reference output, and converts the VDDA voltage through the following formula (assuming that the 12-bit conversion result of the ADC sampling the fast reference source is BGQS_DATA):

$$VDDA = \frac{4095}{BGQS\_DATA} \times 1.0V$$

- Assuming that the ADC sampling value for a certain input channel is ADC_DATA, the actual voltage of a certain input channel (12bit output) can be obtained by the following formula:

$$V_{CHANNEL} = \frac{ADC\_DATA}{BGQS\_DATA} \times 1.0V$$

38.6.2 Use VREFP as a Reference

If VREFP is used as the reference source, and the VREFP pin is driven by the internal boost Buffer, the working reference voltage of the ADC is determined, and there is no need to reverse the ADC reference voltage by sampling the internal reference signal.

At this time, when the external channel is sampled, the actual amplitude of the input voltage can be determined by the following formula:

$$V_{CHANNEL} = \frac{ADC\_DATA}{4095} \times VREFP$$

38.6.3 Temperature sensor

Calculate the current temperature based on the temperature sensor sampling value

ADC uses internal channel to measure PTAT output voltage to obtain conversion data TS_DATA, when the ADC working reference voltage is VDDA, the current actual temperature can be calculated according to the following steps:

1) Calculate the absolute voltage value of the current temperature sensor output by the following formula

$$\mathrm{VPTAT} = \frac{VREFIN\_CAL \times TS\_DATA}{VREFINT\_DATA \times 4095} \times 3V$$

2) Calculate the absolute voltage value of the temperature sensor output during temperature calibration (30°C) by the following formula

$$\mathrm{VPTAT\_}30C = \frac{TS\_CAL1}{4095} \times 3V$$

3) Calculate the current absolute temperature based on the output slope of the temperature sensor

$$\mathrm{Temperature} = \frac{VPTAT - VPTAT\_30C}{slope} + 30C$$

Among them, TS_DATA is the conversion value of ADC sampling the current temperature sensor output. Since the accurate level of the current VDDA is not known, the conversion value needs to be scaled according to the conversion result of VREFINT. TS_CAL1 is the conversion result of temperature calibration under the condition of 30°C+/-1°C and VDDA=3.0V during chip production, and this data is saved in Flash.

Slope represents the output slope of the temperature sensor, and the typical value is 2.6mV/°C.

If the current working reference voltage of ADC is VREFP, the current temperature can be calculated by the following formula:

$$Temperature = \frac{TS\_DATA \times \frac{VREFP}{3.0V} - TS\_CAL30}{slope} + 30C$$

RTC temperature compensation according to the sampling value of temperature sensor

If the temperature sampling value is only used for RTC temperature compensation, it is not necessary to calculate the actual temperature value (-40 ~ 85, decimal), but only according to the 12bit result of

$TS\_DATA \times \frac{VREFINT\_CAL}{VREFINT\_DATA}$ (From the physical principle, the result is 12bit), take TS_CAL1 as the center point to look up the address table. Because the calculation result of the above formula represents the 12-bit output result of the temperature sensor converted to VDDA=3V, the difference between it and TS_CAL1 is the number of LSB that deviate from 30°C. Use this information as the address to look up the temperature compensation table to get the correction value at the

corresponding temperature.

When using the temperature sensor function, you need to enable the temperature sensor output, that is, the PTAT_EN register, and set the VPTAT_BUFFER_EN register to enable the PTAT buffer. After 5us establishment time, enable the ADC to sample the temperature sensor channel.

38.6.4 Slope and Calibration of Temperature Sensor

The operating voltage range of the temperature sensor is 1.8~5.5V, the temperature measurement range is not less than -40~+85°C, and the span is 125°C. The output slope of PTAT is 2.6mV/°C. When VDDA=5V, the resolution is 1.22mV/LSB, which is 2.13LSB/°C; when VDDA=3V, the resolution is 0.73mV/LSB, which is 3.56LSB/°C.

Power supply voltage	VPTAT slope	ADC mV/LSB	LSB/°C
1.8~5.5V	2.6mV/°C	1.22@5V	2.13
		0.73@3V	3.56

Table 38-2 Temperature Sensor Slope

38.6.5 Programmable Sampling Time

By adjusting the sampling time, the internal resistance of different input signal sources can be adapted. The sampling time can be selected through the SMTS1 and SMTS2 registers:

SMTSx	Sampling cycles (T_{ADCCLK})
0000	2
0001	4
0010	8
0011	12
0100	16
0101	32
0110	64
0111	80
1000	96
1001	128
1010	160
1011	192
1100	256
1101	320
1110	384
1111	512

Table 38-3 ADC Sampling Time

The actual ADC sampling conversion time:

T_{CONV} = (Sampling Cycles + 14) * T_{ADCCLK}

The ADC sampling time is mainly determined by the sampling capacitor, the output impedance of the sampled signal, the internal input channel impedance of the chip, and the required sampling accuracy.

The following figure is a schematic diagram of the circuit structure of a single-ended input channel:

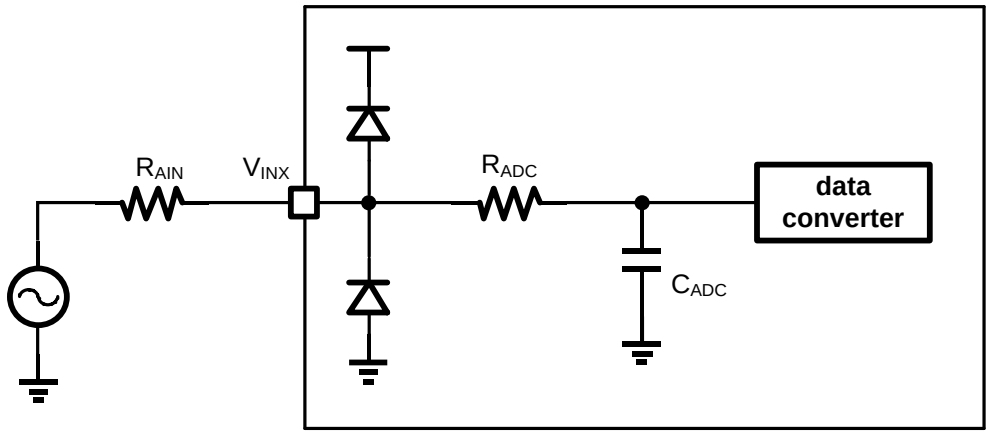


Figure 38–8 ADC Input Channel Diagram

The required sampling time can be estimated according to the following formula:

$$T_{samp} = \ln\left(\frac{2^n}{SA}\right) \times \left(R_{AIN} + R_{ADC}\right) \times C_{ADC}$$

Among them, n = 12, SA represents the allowable sampling error, for example, 0.25 represents 1/4 LSB.

The acceptable sampling time should be calculated and determined according to the relevant parameters and system parameters in the chip manual, and the working clock, sampling period, etc. of the ADC should be configured according to this result.

38.6.6 Output Bit Width Selection

SAR-ADC supports output width setting settings of the highest 12 bits and the lowest 6 bits. The output data bitwidth can be configured through the BITSEL register to obtain a compromise between accuracy and speed.

When different output bitwidths are selected, ADC successively approximates different number of conversion cycles, independent of ADC sampling time.

Assuming the ADC sampling time is set to 2*T_{ADCCLK}, the conversion time is 14*T_{ADCCLK} with 12bit

precision and the total sampling conversion time is 16*T_{ADCCLK}. At 10bit precision, the conversion time corresponds to 12*T_{ADCCLK}.

BITSEL	resolution	Conversion time (T_{ADCCLK})	Max speed @ADCCLK=16Mhz
00	12	14	1Msps
01	10	12	1.14Msps
10	8	10	1.33Msps
11	6	8	1.6Msps
BITSEL	**resolution**	**Conversion time (T_{ADCCLK})**	**Max speed @ADCCLK=32Mhz**
00	12	14	2Msps
01	10	12	2.28Msps
10	8	10	2.66Msps
11	6	8	3.2Msps

Table 38-4 ADC Output Bit Width and Speed

38.6.7 Input Buffer

An analog input buffer is built into the ADC low-speed channel for impedance matching. When sampling very weak external or internal signals, it is recommended to enable input buffer (set BUFEN register) to reduce ADC sampling time and obtain more accurate sampling results.

The input buffer cannot follow the high frequency AC signal, so it is only used to drive the DC input signal. If the AC signal is sampled or the input signal is sufficiently powerful, the input buffer must be bypassed.

The input buffer cannot achieve rail-to-rail input and output. The recommended input signal range is 0.1V~VREF-0.1V, and the operating voltage range is VDDA=2.0~5.5V. Buffer supports low power mode, typical power consumption is about 40uA in normal mode and 20uA in low power mode. In normal mode, the drive capacity can meet the maximum ADC sampling time of 1us, while in low power mode, the sampling time needs to be extended to 10us.

Note:

1) The ADC high-speed channel does not have a built-in buffer to ensure sampling capability for high frequency AC signals. Therefore, it is not recommended to use high-speed channel to sample very weak DC signals.

2) When measuring internal reference VREF1P2 and temperature sensor, ADC should use VREF1P2 and output buffer of temperature sensor.

38.6.7.1 Elimination of Buffer Offset

The offset of the input buffer affects the sampling of small signals. In order to eliminate the influence of offset as much as possible, the buffer implements the chopper function. When the BUFCHP_EN register is set, the hardware oversampling must be enabled with the chopper function to average out the offset error. The number of oversampling is greater than or equal to 2 times.Therefore, when buffer is used and chopper is enabled, the sampling rate will be at least reduced by half compared to the case where chopper is not enabled.

38.6.8 VBAT and VDD Power Voltage Sampling

ADC can be used for VBAT and VDD sampling. Before sampling, VBAT is divided by resistors to obtain VBAT/3 or VDD/3 level and send it to ADC for conversion.

When the VBAT/3 or VDD/3 channel is enabled, the voltage divider resistor string is automatically enabled. The resistance of the voltage divider resistor string is about 100Kohm, and when the input is 3V, the current consumed by the resistor string is about 30uA.

The output impedance of the voltage divider string is relatively large, and the internal buffer needs to be enabled during ADC sampling to strengthen the drive.

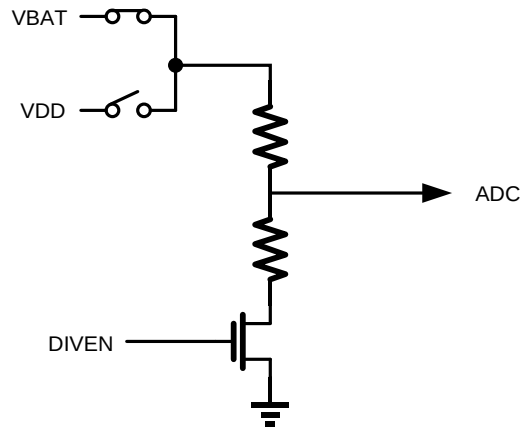


Figure 38–9 VBAT/VDD Divider Circuit Diagram

38.6.9 VBAT and VDD Power Voltage Sampling

ADC can sample the following internal signals of the chip:

- DAC output
- Temperature sensor output
- 1.2V internal reference source
- 1.0V internal reference source
- OPA output

Among them, 1.2V reference, 1.0V reference, and temperature sensor all have independent output buffers. These analog buffers need to be enabled before ADC sampling to strengthen the drive.

The DAC output connected inside the chip is not driven by the internal buffer of the ADC, so it is necessary to enable the internal buffer of the ADC during sampling.

38.6.10 Conversion Mode

ADC supports the following conversion modes:

- Single conversion
  - Semi-automatic trigger (SEMI-AUTOMATIC)
  - Automatic trigger (AUTOMATIC)
- Continuous conversion

Conversion start can be triggered by software or events, and multiple event trigger sources can be selected through registers.

38.6.10.1 Single Conversion Mode

Single conversion means that the ADC automatically stops sampling after completing a conversion sequence (but the ADC remains enabled) and waits for a new trigger event to occur.

Single conversion supports two modes: semi-automatic trigger and automatic trigger.

Automatic trigger mode: After software or hardware trigger event starts ADC conversion, the ADC will sequentially sample all enabled channels. After a single channel is sampled, the EOC (End of Conversion) flag is set, and after all channels are sampled, the EOS (End of Sequence) flag is set, and the conversion ends. Assuming that channels 0, 3 and 5 are enabled

- 1st trigger event:Channels 0, 3, and 5 are sampled sequentially, three EOCs are generated in the process, and EOS is finally generated
- 2nd trigger event:Repeat the above process

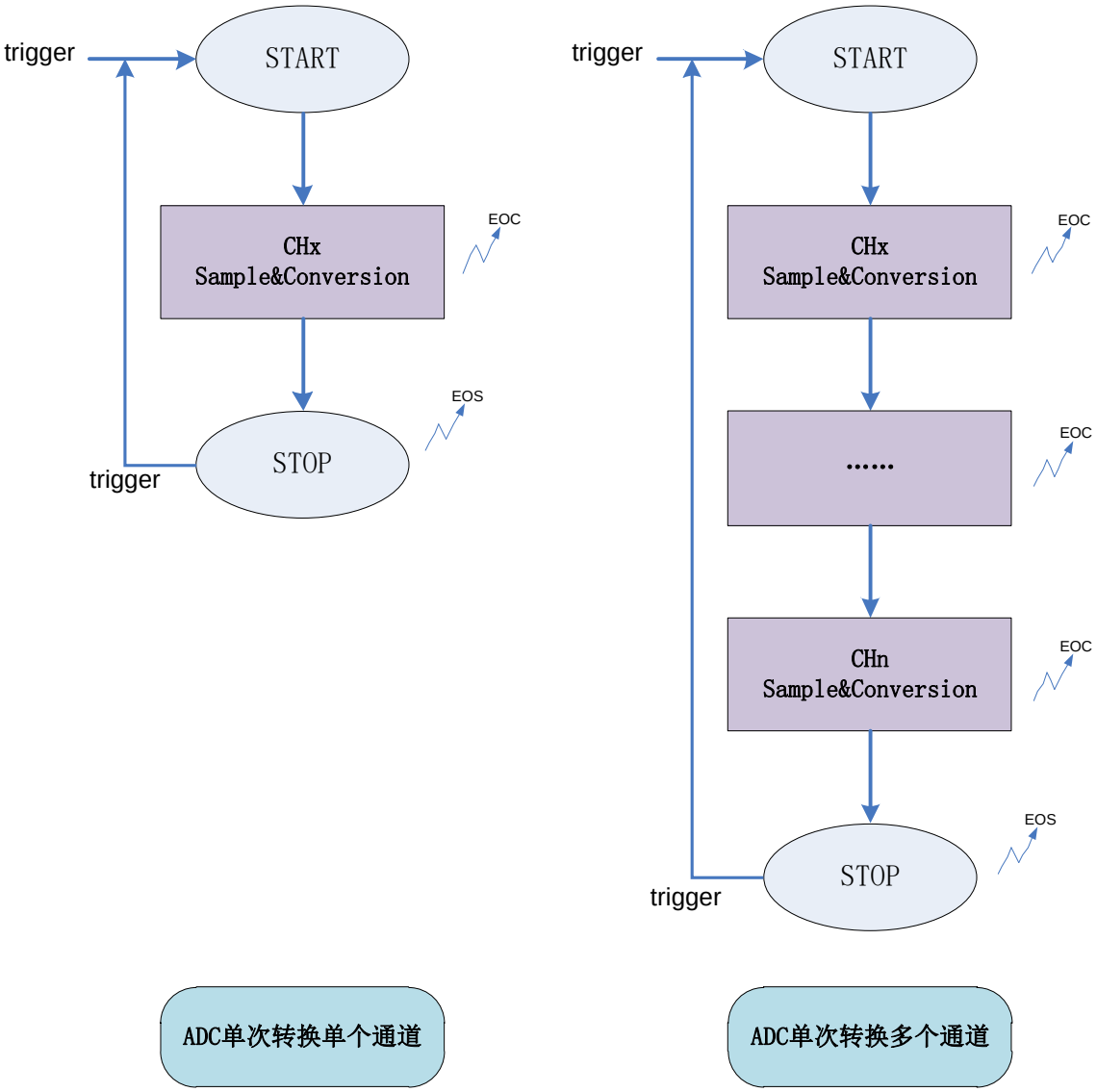


Figure 38–10 ADC Single Conversion Automatic Trigger Mode

Semi-automatic trigger mode: Software or hardware trigger event will only start the ADC once to convert an enable channel. For example, channels 0, 3 and 5 are enabled

- 1st trigger event: Channel 0 is sampled to generate EOC
- 2nd trigger event: Channel 3 is sampled to generate EOC
- 3rd trigger event: Channel 5 is sampled to generate EOC and EOS
- 4th trigger event: Channel 0 is sampled to generate EOC
- 5th trigger event: Channel 3 is sampled to generate EOC

  ……

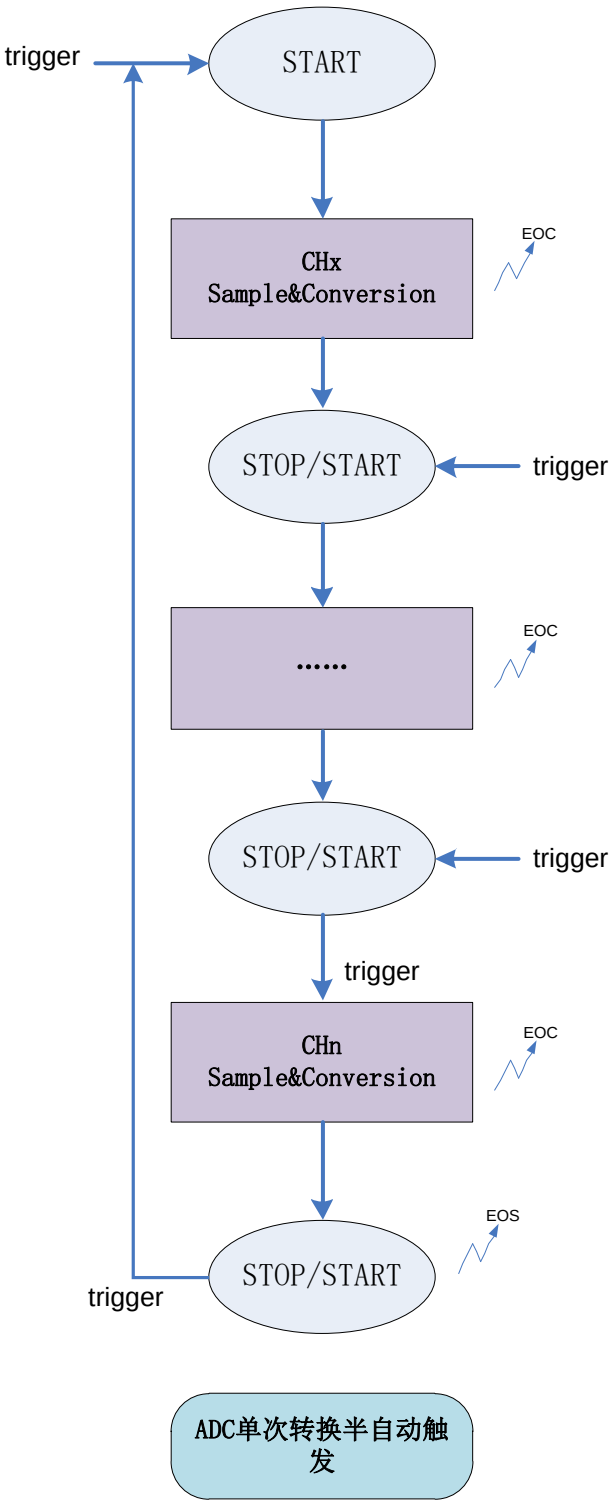


Figure 38–11 ADC Single Conversion Semi-automatic Trigger Mode

38.6.10.2 Continuous Conversion Mode

After the trigger event arrives, all enabled channels are sampled, and the ADC will not stop automatically, but will sample cyclically until the software stops the ADC.

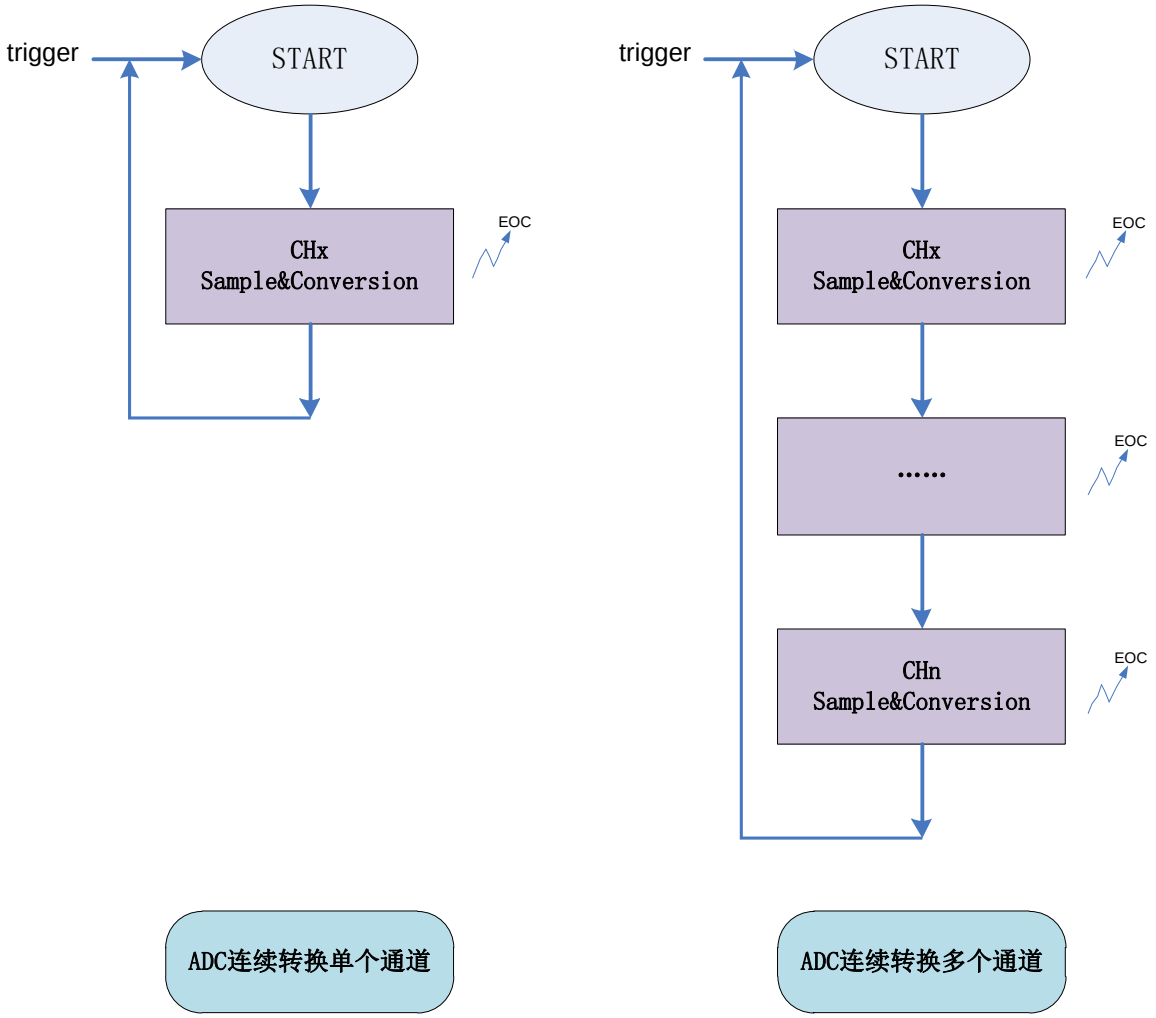


Figure 38–12 ADC Continuous Trigger Mode

After each channel is sampled, the data is stored in the ADC_DATA register, and the software should read the data in time before the next conversion, or move the data through DMA. If the data cannot be taken away in time, an overrun will be caused, the overrun flag is set, and an interrupt can be issued.

38.6.11 Conversion Mode

After ADC is enabled, the conversion trigger supports software or hardware event trigger.

Software Trigger

The software initiates the conversion by setting the SWTRIG register.

Hardware Trigger

ADC has the following hardware trigger sources: LUTx_TRGO, RTCA_TRGO, ATIM_TRGO,

GPTIM0_TRGO, GPTIM1_TRGO, GPTIM2_TRGO, BSTIM16_TRGO, LPTIM16_TRGO, comparator output; if the ADC is in the conversion process, the trigger signal that comes at this time will be ignored.

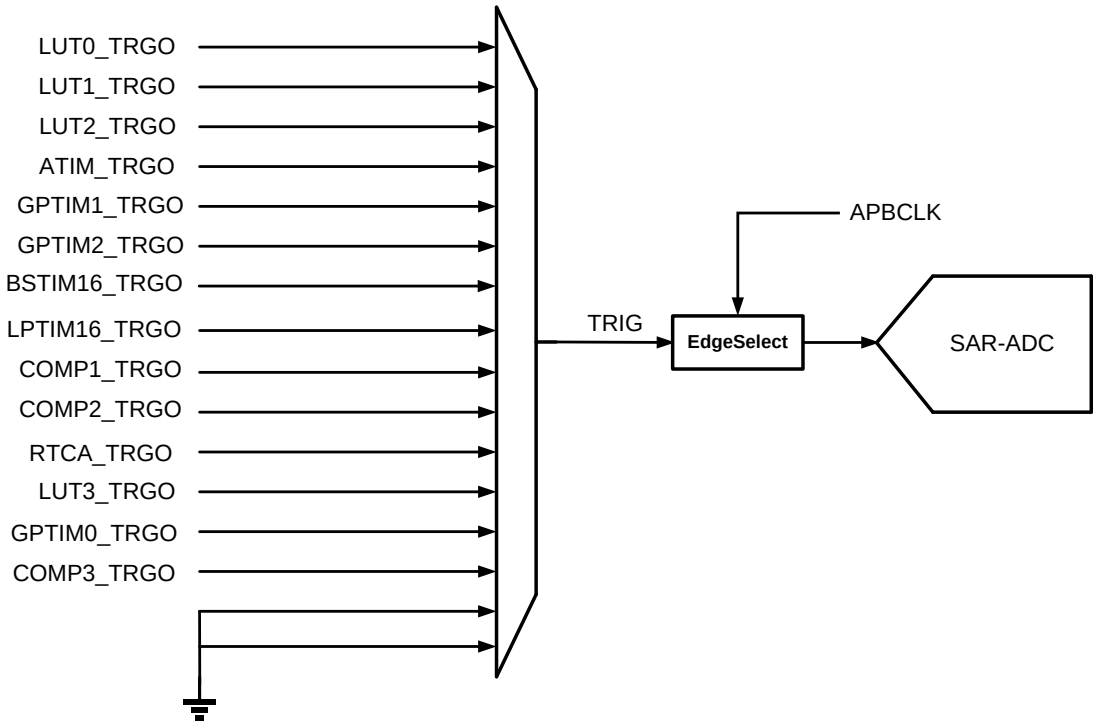


Figure 38–13 ADC Hardware Trigger Source

Note: The hardware trigger source input is synchronously sampled by APBCLK, so the ADC bus clock must be enabled (set the ADC_PCE register in the CMU module) when using the hardware trigger function.

Trigger source timing characteristics classification:

Trigger source	Clock domain	Trigger delay when ADC working clock is selected as APBCLK (T_{APBCLK})	Description
Software trigger	APBCLK	3.5	Can achieve synchronous trigger, determine delay
LUTx_TRGO		4.5	
GPTIMx_TRGO		4.5	
COMPx_TRGO		4.5	
ATIM_TRGO	Others	2~3	Asynchronous clock trigger
BSTIM16_TRGO			
LPTIM16_TRGO			
RTCA_TRGO			

38.6.12 Oversampling and Hardware Averaging

ADC supports hardware oversampling and averaging, which can improve the resolution to a certain extent. The principle is that for low-speed input signals, ENOB can be increased by averaging after multiple consecutive sampling. The oversampling formula is as follows:

$$result = \frac{\sum_{n=1}^{N} CONVERSION_n}{M}$$

Where N is the oversampling multiple, which can be configured as 2/4/8/16/32/64/128/256, M is the result right shift number, the maximum right shift is 8bit; since each conversion result is 12bit,the result of the maximum accumulation of 256 times is 20bit. After shifting, the final result of 12~16bit can be obtained. The ADC output result is only 16 bits at most. If the result exceeds 16 bits after right shifting, the high bits will also be discarded.

When oversampling is enabled, the EOC signal is set after N consecutive samples, which is like only one sample conversion for applications and DMA.

38.6.13 ADC Working Clock

The ADC adopts a dual clock structure, in which APBCLK is used for bus register access; ADCCLK is the ADC working clock, which can be APBCLK or other independent clocks.

When the ADC sampling conversion works under an independent clock, the ADC working frequency can be decoupled from the system clock frequency. For example, the CPU and bus can work at a lower frequency while keeping the ADC working at a higher sampling rate to achieve higher applications flexibility. However, in this case, there is a cross-clock domain resynchronization delay between the trigger event and the ADC sampling start.The size of this delay is related to the current frequency and phase relationship between APBCLK and ADCCLK, and has a certain range of variation.

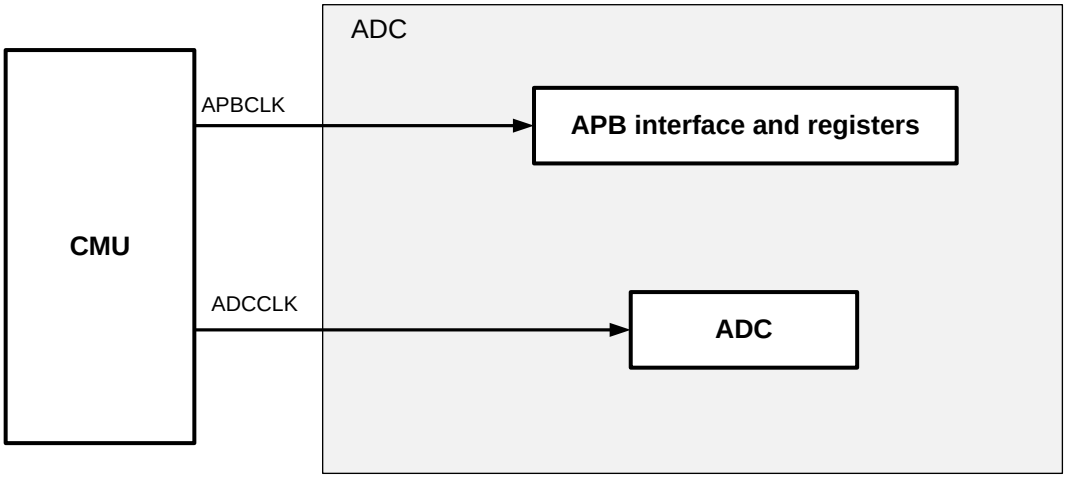


Figure 38–14 ADC Clock Diagram

38.6.14 Data Conflict and Automatic Waiting

The EOC flag will be set after each conversion. Software or DMA will automatically clear the EOC after reading the ADC_DATA register. It can also be cleared by software by writing 1 to it. When the EOC flag is not cleared, the arrival of new conversion data will cause data overrun; there are two overrun modes:

OVRM=0: Keep old data, discard new data

OVRM=1: New data is written to overwrite old data

When overrun occurs with DMA, a new DMA request will not be initiated until the OVR flag is cleared by software.

The ADC controller also supports automatic waiting. If the WAIT register is set by software, the ADC controller will not initiate a new conversion before the ADC_DATA register is read.Hardware trigger events that arrive in the waiting state will also be ignored.

The following figure is a schematic diagram of enabling automatic waiting when the software triggers the continuous mode:

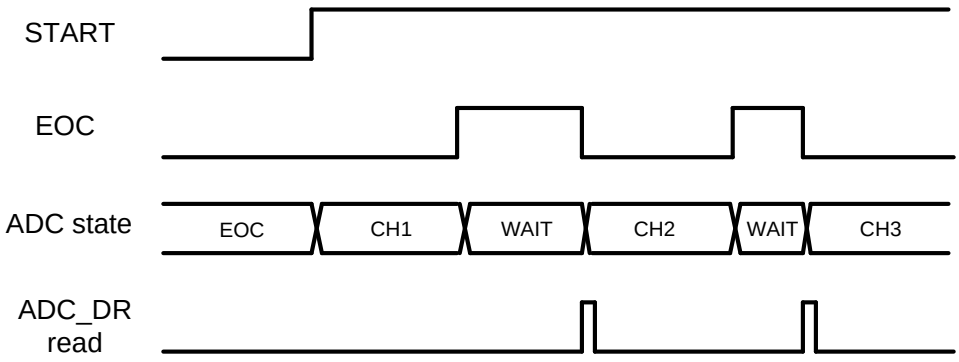


Figure 38–15 ADC Automatic Waiting

38.6.15 DMA

In multi-channel conversion or continuous conversion, using DMA to move the conversion result is an efficient solution. With DMAEN enabled, when each conversion is completed (EOC), the ADC controller will generate a DMA request to notify the DMA to move the result in the data register to the specified SRAM address.ADC conversion is initiated by trigger events, which can be triggered by software or hardware.DMA mode supports multiple conversions after one trigger (automatic mode), or one conversion per trigger (semi-automatic mode).

In both automatic mode (ADC_CFGR2.SEMI=0) and semi-automatic mode (ADC_CFGR2.SEMI=1), the DMA interface of ADC can support single mode and loop mode:

Single Mode

After the conversion is completed, the data transfer is initiated. This process will be repeated until the DMA transfer length configured by the software is completed, and then the ADC controller will automatically stop the conversion (by receiving the DMA transfer completion interrupt flag signal), turn off the ADC, and no longer initiate requests to the DMA.This mode is mainly used to sample a certain length of a specific analog signal.

Loop Mode

Cooperating with the DMA loop mode, ADC continuously cyclically converts and initiates DMA requests until the software stops the conversion. This mode can be used to process continuous analog signal sampling.

The ADC_CFGR2.SEMI register can be used to configure the automatic or semi-automatic conversion in DMA mode. Configure the loop mode or single mode through the CHxCIRC, and configure the transmission length through the CHxTSIZE.

The effect of register configuration combinations is shown in the following table.

ADC_CFGR2.CONT	ADC_CFGR2.SEMI	DMA CHxCIRC	DMA CHxTSIZE	Description
0	0	0	N	Single Automatic Mode, after triggering events, converts input channels N times and carries data to SRAM in turn; all enabled input channels are sampled, and ADC automatically stops and waits for the next trigger; if the number of data moves reaches N, DMA is automatically turned off.
0	0	1	N	Single Automatic Mode (Loop storage), after triggering events, the enabled input channels are continuously converted and the data is transferred in sequence, until all enabled input channels have been sampled, ADC automatically stops and waits for the next trigger; RAM data space length is N, when the length of the transferred data exceeds N, it returns to the starting address pointed to by the DMA channel pointer and overwrites the original data.
0	1	0	N	Single Semi-automatic Mode, each trigger event starts a conversion, sequentially sampling all enabled input channels; the number of data transfers reaches N, DMA is automatically turned off.

0	1	1	N	Single Semi-automatic Mode (Loop storage), each trigger event starts a conversion, each trigger event starts a conversion, sequentially sampling all enabled input channels; RAM data space length is N, when the length of the transferred data exceeds N, it returns to the starting address pointed to by the DMA channel pointer and overwrites the original data.
1	x	0	N	Continuous Mode, after triggering, ADC keeps sampling enabled channels, DMA keeps moving data until N times of data transfers are completed, and DMA is automatically turned off.
1	x	1	N	Continuous Mode (Loop storage), after triggering, ADC keeps sampling enabled channels, DMA keeps moving data; RAM data space length is N, when the length of the transferred data exceeds N, it returns to the starting address and overwrites the original data until the software turns off DMA.

Table 38-5 DMA Configuration and Function

With DMA enabled, if an overrun occurs, the ADC controller will no longer send DMA requests until the OVR flag is cleared.

Note that both single and continuous conversion modes can support DMA transfer; DMA transfer length is defined by the number of EOC, not EOS, that is, DMA only cares about how many ADC_DR are transported.

38.6.15.1 Use Cases of DMA

Case 1:

ADC is configured as singleautomatic mode, 3 ADC input channels are enabled, and the DMA TSIZE is configured as 6.After the first trigger, ADC samples 3 channels in turn, DMA transfers data 3 times, then ADC and DMA suspend work, waiting for the second trigger. After the second trigger, ADC samples 3 channels in turn, and DMA carries data 3 times. After 6 DMA transfers are completed, DMA channel is turned off.

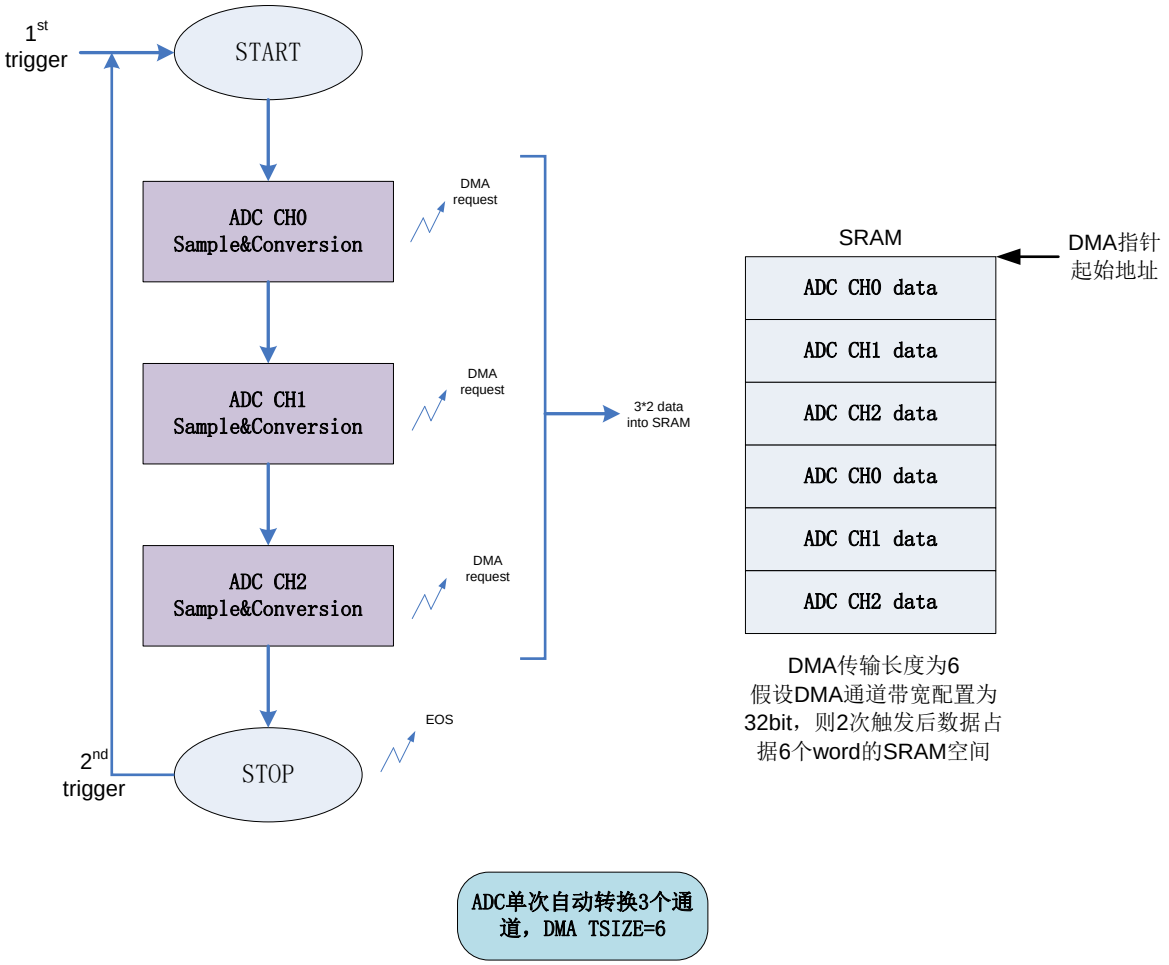


Figure 38–16 ADC Single Automatic Trigger & DMA Case 1

Case 2:

ADC is configured as single automatic mode, 3 ADC input channels are enabled, and the DMA TSIZE is configured as 2. After the first trigger, ADC samples 3 channels in turn, DMA transfers data 2 times, then DMA and ADC are automatically turned off, and the third input channel will not be sampled.

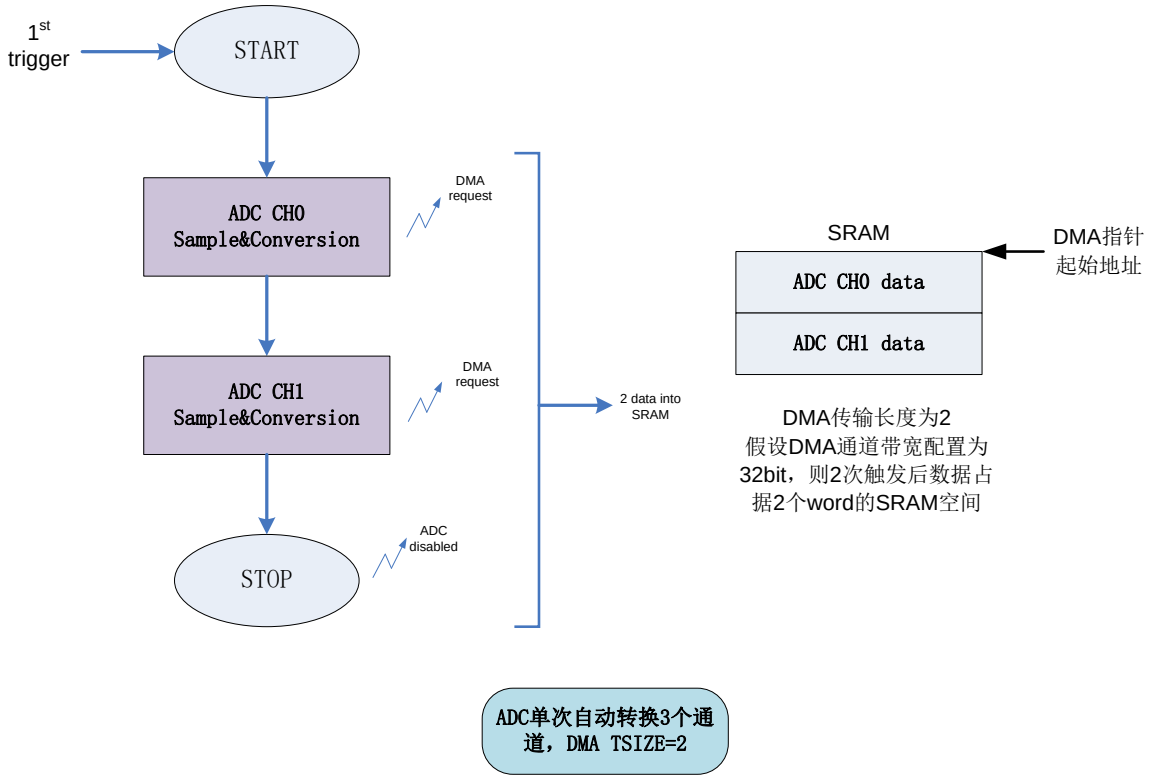


Figure 38–17 ADC Single Automatic Trigger & DMA Case 2

Case 3:

ADC is configured as single semi-automatic mode, 3 ADC input channels are enabled, and the DMA TSIZE is configured as 6.After each trigger, DMA transfers one conversion data. After a total of 6 data transfers are completed, ADC and DMA are automatically turned off.

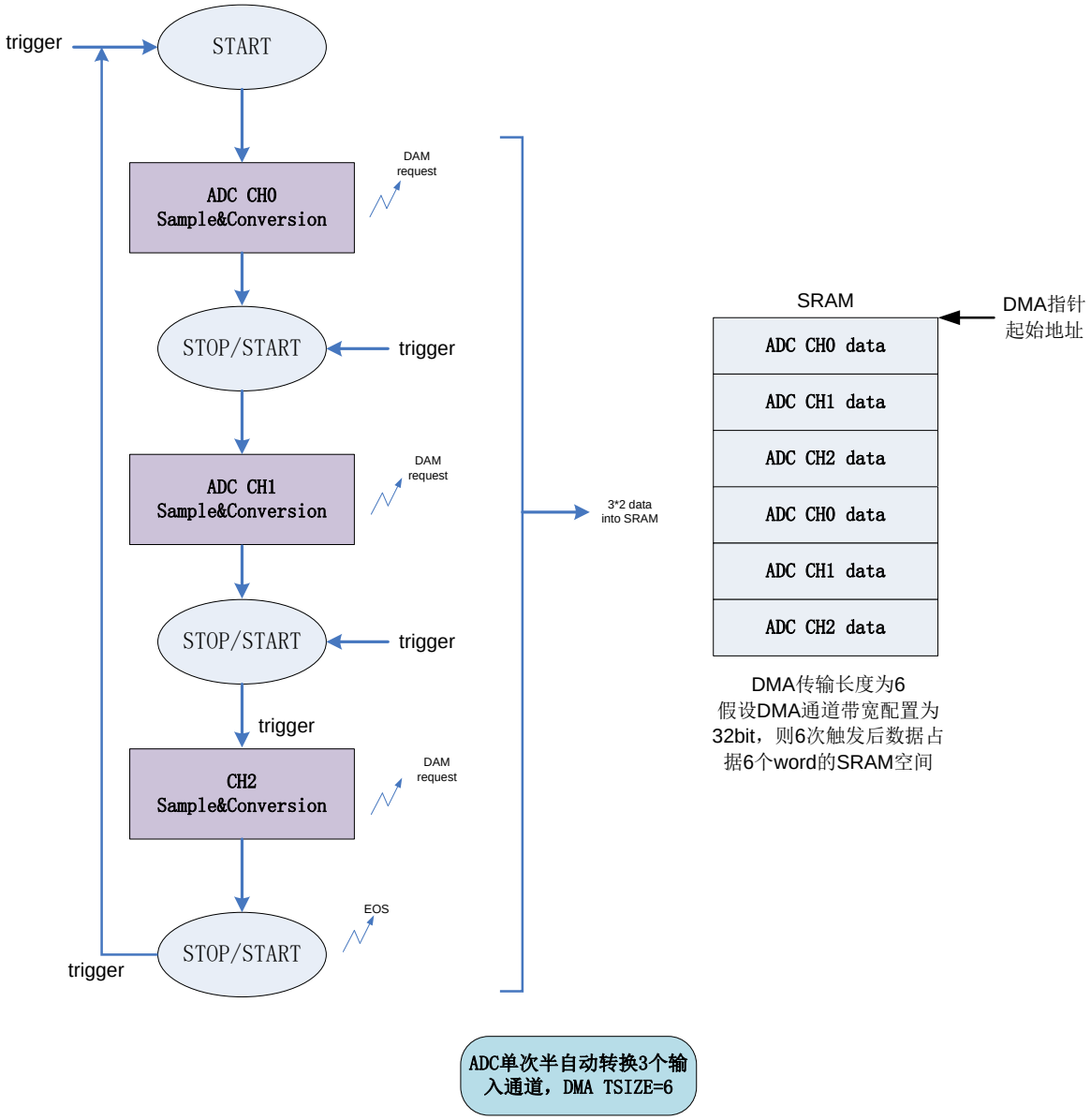


Figure 38–18 ADC Single Semi-automatic Trigger & DMA Case 3

Case 4:

ADC is configured as automatic mode (SEMI=0), 3 ADC input channels are enabled (ADC_IN0/1/2), the DMA TSIZE is configured as 6, the DMA pointer points to RAM address x, and the DMA is configured in loop mode (CICR=1). After the first trigger,ADC performs 3 sampling conversions, DMA transfers data 3 times to RAM address x/x+1/x+2; after the second trigger, ADC performs 3 samplings, DMA transfers data 3 times to RAM address x+3/x+4/x+5; after the third trigger, ADC performs 3 samplings, DMA transfers 3 data to RAM address x/x+1/x+2, cyclically until the software turns offDMA.

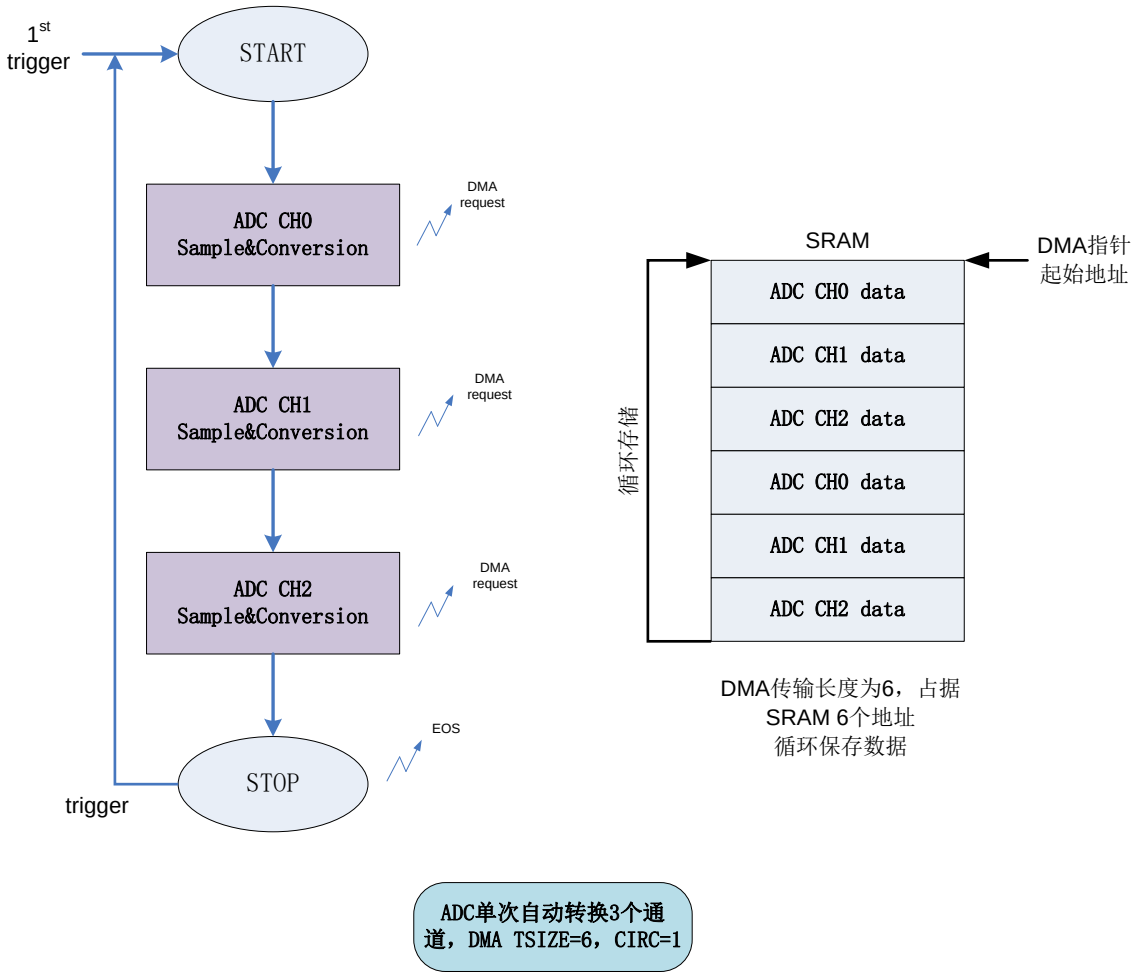


Figure 38–19 ADC Automatic Trigger & DMA Loop Mode

Case 5:

ADC is configured as continuous mode (CONT=1),3 ADC input channels are enabled (ADC_IN0/1/2), the DMA TSIZE is configured as 6, the DMA pointer points to RAM address x, and the DMA is configured in loop mode (CICR=1). After the first trigger, ADC continuously samples and converts the enabled channels, and DMA continuously transfers the data to RAM address x/x+1/x+2/x+3/x+4/x+5, cyclically until the software turns off DMA.

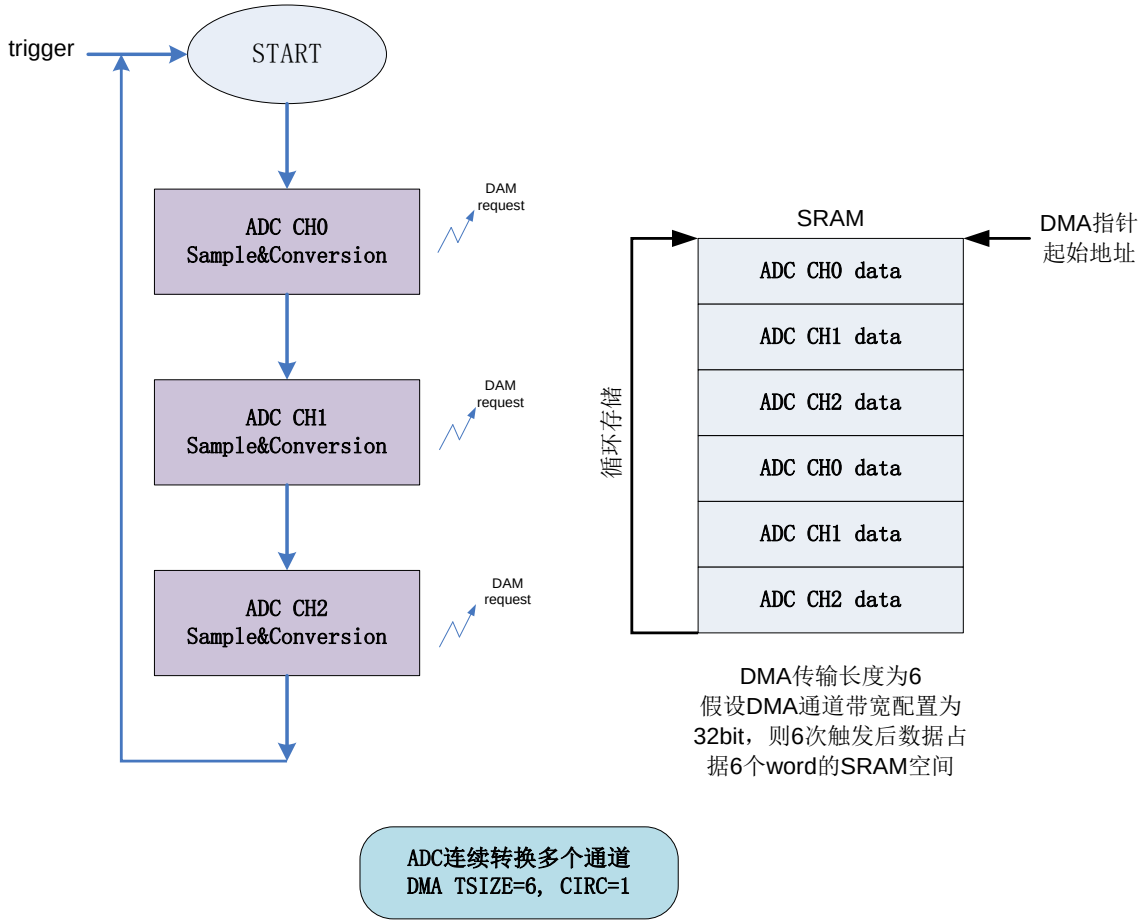


Figure 38–20 ADC Continuous Mode & DMA Loop Mode

38.6.16 Analog Window Watchdog (AWD)

The AWD function is used to monitor whether the input signal level of an analog input channel or all input channels is within the amplitude range set by the register. When the ADC conversion value is higher than AWD_HT or lower than AWD_LT, the interrupt flag register will be set. The flag register is cleared by writing 1 by software.

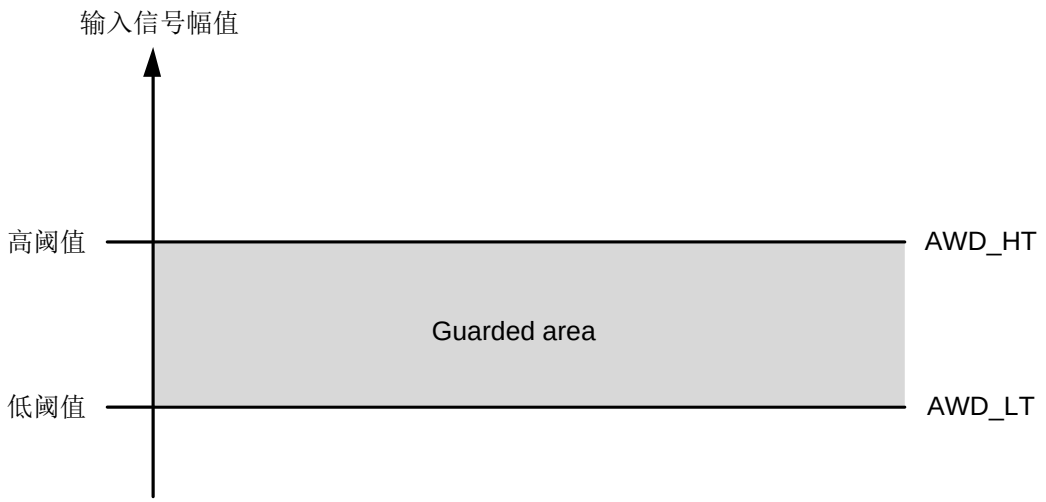


Figure 38–21 ADC Analog Watchdog Threshold Diagram

The analog window watchdog function is enabled through the AWDEN register, and single-channel monitoring or all-channel monitoring is configured through the AWDSC register.

The bit width of the AWD threshold setting register is 16bit.

38.6.17 ADC Calibration

ADC supports offset self-calibration. It is recommended to perform a calibration operation first after the chip is powered on to obtain better accuracy.

The software sets CALEN to start the comparator calibration. After the hardware completes the ADC self-calibration operation, the CALEN register is automatically cleared. After the calibration operation is over, the EOCAL interrupt flag is set. If the interrupt enable EOCALIE is 1, an interrupt is generated to notify the CPU.

Note: *After ADC reset, before the first conversion, it is recommended to start a calibration operation. After the calibration is complete, re-enable the ADC without re-calibration to perform conversion. Therefore, it is recommended that the user perform a calibration operation after the chip power-on reset is completed. When the ADC working environment (temperature, voltage) changes significantly, it is also recommended to calibrate to obtain higher accuracy.*

After the calibration operation, the calibration parameters are saved in the ADC internal registers. ADC module reset or ADC controller reset will not clear the calibration register content, only power-on and power-off reset will clear the calibration register.

38.7 Low Power Mode

When the chip enters the low power mode, the ADC is still allowed to work. But in low power mode, the chip automatically disables all high-speed clock sources, so the highest working clock of ADC is only RCLF, and the corresponding highest sampling rate is 38.4Ksps.

Mode	Description
LPRUN	Allow to work, allow to use DMA
Sleep/DeepSleep	Allow to work, only use RCLF as ADC clock; and can wake up MCU
VBAT	Cannot work

Table 38-6 DMA Configuration and Function

Note: *Because the peripheral trigger requires APBCLK, APBCLK is turned off in Sleep/DeepSleep sleep mode, so it is not possible to start ADC conversion by peripheral trigger in sleep mode. But if ADC uses the RCLF clock, and software starts ADC conversion before entering sleep, ADC will still work after chip sleeps.*

38.8 Register

Offset	Name	Symbol
ADC(Base Address: 0x40015C00)		
0x00	ADC Interrupt and Status Register	ADC_ISR
0x04	ADC Interrupt Enable Register	ADC_IER
0x08	ADC Control Register1	ADC_CR1
0x0C	ADC Control Register2	ADC_CR2
0x10	ADC Calibration Register	ADC_CALR
0x14	ADC Config Register1	ADC_CFGR1
0x18	ADC Config Register2	ADC_CFGR2
0x1C	ADC Sampling Time Register	ADC_SMTR
0x20	ADC Channel Enable Register	ADC_CHER
0x24	ADC Differential Channel Control Register	ADC_DCR
0x28	ADC Data Register	ADC_DR
0x2C	ADC Analog Watchdog Threshold Register	ADC_HLTR

38.8.1 ADC Interrupt and Status Register (ADC_ISR)

NAME	ADC_ISR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	AWD_AH	AWD_UL	EOCAL	BUSY	OVR	EOS	EOC
access	U-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:7	--	RFU: **Reserved, read as 0**
6	AWD_AH	Analog watchdog exceeds the higher limit interrupt flag (Analog Watchdog Above High), when the sampled value is higher than AWD_HT, hardware set, and software writes 1 to clear
5	AWD_UL	Analog watchdog is below the lower limit interrupt flag (Analog Watchdog Under Low), when the sampled value is lower than

bit	name	functional description
		AWD_LT, hardware set, and software writes 1 to clear
4	EOCAL	End Of Calibration, hardware set, and software writes 1 to clear 1: End of calibration process 0: No calibration process The EOCAL flag will be set after the quasi-end.
3	BUSY	ADC Busy 1: ADC is in the process of calibration, sampling or conversion 0: ADC is idle
2	OVR	Data conflict flag, hardware set, software write 1 to clear(Over-Run) When the last conversion result in the ADC_DATA register has not been read, and the new conversion result comes again, the hardware sets the OVR flag 0: No data conflict 1: Data conflict
1	EOS	End Of Sequence After all the enabled channels are converted, EOS is set, and software writes 1 to clear it
0	EOC	End Of Conversion After the conversion of each channel is completed, EOC is set, and software writes 1 to clear it

38.8.2 ADC Interrupt Enable Register (ADC_IER)

NAME	ADC_IER							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	AWD_AHIE	AWD_ULIE	EOCALIE	-	OVRIE	EOSIE	EOCIE
access	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:7	--	RFU: **Reserved, read as 0**
6	AWD_AHIE	Analog watchdog sampling value is higher than the higher limit interrupt enable, 1 is valid
5	AWD_ULIE	Analog watchdog sampling value is lower than the lower limit interrupt enable, 1 is valid
4	EOCALIE	Calibration end interrupt enable register 0: Disable EOCAL interrupt 1: Enable EOCAL interrupt
3	--	RFU: **Reserved, read as 0**
2	OVRIE	Data conflict interrupt enable register 0: Disable data conflict interrupt 1: Enable data conflict interrupt
1	EOSIE	End Of Sequence 0: Disable EOS interrupt 1: Enable EOS interrupt
0	EOCIE	End Of Conversion 0: Disable EOC interrupt 1: Enable EOC interrupt

38.8.3 ADC Control Register1 (ADC_CR1)

NAME	ADC_CR1							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						SWTRIG	ADEN
access	U-0						R/W-0	R/W-0

bit	name	functional description
31:2	--	RFU: **Reserved, read as 0**
1	SWTRIG	ADC start conversion register (software trigger), software writes 1 to start, hardware automatically clears.
0	ADEN	ADC enable register. Set ADEN before starting the conversion. 0: Disable ADC

bit	name	functional description
		1: Enable ADC

38.8.4 ADC Control Register2 (ADC_CR2)

NAME	ADC_CR2							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						TRGCFG	
access	U-0						R/W-00	

bit	name	functional description
31:1	--	RFU: **Reserved, read as 0**
1:0	TRGCFG	Trigger signal enable and polarity selection(Trigger Config).When configured as a non-software trigger, disable software to write to the START register. 00: Software trigger 01: Hardware rising edge trigger 10: Hardware falling edge trigger 11: Hardware rising and falling edges trigger

38.8.5 ADC Calibration Register (ADC_CALR)

NAME	ADC_CALR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	OSCAL_CYCLE							
access	R/W-0000 0111							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-			OFFL_EN				

access	U-0			R/W-0				
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	VCM_CTRL	VCM_MODE	CMPRDY_DELAY		-			CALEN
access	R/W-0	R/W-0	R/W-00		U-0			R/W-0

bit	name	functional description
31:24	-	RFU: **Reserved, read as 0**
23:16	OSCAL_CYCLE	Offset calibration cycle number configuration
15:13	-	RFU: **Reserved, read as 0**
12	OFFL_EN	Offline Calibration Enable 0: Automatic calibration 1: Offline calibration
11:8	-	RFU: **Reserved, read as 0**
7	VCM_CTRL	Common mode current configuration
6	VCM_MODE	VCM control mode 0: Normal 1: Continuous
5:4	CMPRDY_DELAY	Comparator delay control 00: Minimum delay 11: Maximum delay
3:1	-	RFU: **Reserved, read as 0**
0	CALEN	Offset Calibration Enable Software writes 1 to start the calibration cycle, and automatically clears and sets the EOCAL register after calibration. Offset is started by the CALEN register. ***Note:*** *When starting calibration, must ensure that OVSEN = 0 (oversampling cannot be enabled)*

38.8.6 ADC Config Register1 (ADC_CFGR1)

NAME	ADC_CFGR1							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				BUFCHP_EN	BUFLPF	BUFMOD	BUFEN

access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	APBCLK_PSC		EXSOC	BITSEL		-	REFSEL	
access	R/W-00		R/W-0	R/W-00		U-0	R/W-00	

bit	name	functional description
31:12	--	RFU: **Reserved, read as 0**
11	BUFCHP_EN	Buffer chopper enable 0: Disable chopper 1: Enable chopper ***Note:*** *When enabling chopper, hardware oversampling must be enabled.*
10	BUFLPF	Buffer low-pass-filter mode 0: Normal mode 1: Low pass filter mode
9	BUFMOD	Buffer Mode 0: Normal mode 1: Low power mode
8	BUFEN	Buffer Enable 0: Disable and bypass buffer 1: Enable buffer
7:6	APBCLK_PSC	APBCLK prescaler 00: Divided-by-1 01: Divided-by-2 10: Divided-by-4 11: Divided-by-8
5	EXSOC	External Start-of-conversion 0: Disable external input SOC 1: Enable external input SOC
4:3	BITSEL	ADC Output Bit-width Select 00: 12bit 01: 10bit 10: 8bit 11: 6bit
2	CLKSEL	Clock Select 0: Use ADCCLK 1: Use APBCLK
1:0	REFSEL	Reference Select 00,11: Use VDDA 01: Use VREFP 10: Use VDD15

38.8.7 ADC Config Register2 (ADC_CFGR2)

NAME	ADC_CFGR2							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-	AWDCH					AWDSC	AWDEN
access	U-0	R/W-00000					R/W-0	R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	OVSS				OVSR			OVSEN
access	R/W-0000				R/W-000			R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-	IOTRFEN	-		SEMI	WAIT	CONT	OVRM
access	U-0	R/W-0	U-0		R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	EXTS				-	SCANDIR	-	DMAEN
access	R/W-0000				U-0	R/W-0	U-0	R/W-0

bit	name	functional description
31	--	RFU: **Reserved, read as 0**
30:26	AWDCH	Analog Watchdog Channel Select, only valid when AWDSC=1 00000: AWD monitors ADC_IN0 00001: AWD monitors ADC_IN1 00010: AWD monitors ADC_IN2 00011: AWD monitors ADC_IN3 00100: AWD monitors ADC_IN4 00101: AWD monitors ADC_IN5 00111: AWD monitors ADC_IN6 01000: AWD monitors ADC_IN7 01001: AWD monitors ADC_IN8 01010: AWD monitors ADC_IN9 01011: AWD monitors ADC_IN10 01100: AWD monitors ADC_IN11 01101: AWD monitors ADC_IN12 01110: AWD monitors ADC_IN13 01111: AWD monitors ADC_IN14 10000: AWD monitors ADC_IN15 10001: AWD monitors ADC_IN16 10010: AWD monitors ADC_IN17 10011: AWD monitors ADC_IN18 10100: AWD monitors ADC_IN19 10101: AWD monitors ADC_IN20 Others: Preserve

bit	name	functional description
25	AWDSC	Analog Watchdog Single Channel or Full Channel Select 0: AWD monitors all enabled external input channels 1: AWD monitors a single channel specified by AWDCH
24	AWDEN	Analog Watchdog Enable Register 0: Disable AWD 1: Enable AWD This register can only be configured when START=0
23:20	OVSS	Oversampling Shift Control Register 0000: Does not shift 0001: Shift 1bit to the right 0010: Shift 2bit to the right 0011: Shift 3bit to the right 0100: Shift 4bit to the right 0101: Shift 5bit to the right 0110: Shift 6bit to the right 0111: Shift 7bit to the right 1000: Shift 8bit to the right Others: RFU
19:17	OVSR	Oversampling Rate Control 000: 2x 001: 4x 010: 8x 011: 16x 100: 32x 101: 64x 110: 128x 111: 256x
16	OVSEN	Oversampling Enable 0: Disable oversampling 1: Enable oversampling
15	--	RFU: **Reserved, read as 0**
14	IOTRFEN	GPIO Trigger Filter Enable 0: Disable digital filter 1: Enable digital filter
13:12	--	RFU: **Reserved, read as 0**
11	SEMI	Single Conversion Semi-automaticMode, only valid in single conversion (CONT=0), see "Conversion Mode" chapter 0: Automatic mode 1: Semi-automatic mode
10	WAIT	Wait Mode Control 0: No waiting, if the last converted data is not read in time, Overrun may occur 1: Waiting mode, before the last conversion data is read, the

bit	name	functional description
		next conversion will not be started
9	CONT	Continuous Conversion Mode Enable 0: Single conversion 1: Continuous conversion
8	OVRM	Overrun Mode Control 0: Whenoverrun occurs, keep the last data and discard the converted value 1: When overrun occurs, overwrite the last data
7:4	EXTS	Hardware Trigger Source Select 0000: LUT0_TRGO 0001: LUT1_TRGO 0010: LUT2_TRGO 0011: ATIM_TRGO 0100: GPTIM1_TRGO 0101: GPTIM2_TRGO 0110: BSTIM16_TRGO 0111: LPTIM16_TRGO 1000: COMP1_TRGO 1001: COMP2_TRGO 1010: RTCA_TRGO 1011: LUT3_TRGO 1100: GPTIM0_TRGO 1101: COMP3_TRGO 1110: RFU 1111: RFU ***Note:*** *The EXTS register must be modified when TRGCFG=00*
3	--	RFU: **Reserved, read as 0**
2	SCANDIR	External Channel Scan Direction Control (A total of 26 channels, actually only the enabled channels will be sampled) 0: Forward scan, ADC_IN0->ADC_IN19->Internal channel 1: Reverse scan, Internal channel->ADC_IN19->ADC_IN0
1	--	RFU: **Reserved, read as 0**
0	DMAEN	DMA Enable 0: Disable DMA 1: Enable DMA

38.8.8 ADC Sampling Time Register (ADC_SMTR)

NAME	ADC_SMTR							
Offset	0x1C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24

name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SMTS2				SMTS1			
access	R/W-0000				R/W-0000			

<table>
<tr><th>bit</th><th>name</th><th>functional description</th></tr>
<tr><td>31:8</td><td>--</td><td>RFU: Reserved, read as 0</td></tr>
<tr><td>7:4</td><td>SMTS2</td><td>Fast Sampling Time Control 2 (*ADC working clock cycle), applicable to all fast channels
<table>
<tr><th>SMTSx</th><th>Sampling cycles</th></tr>
<tr><td>0000</td><td>2</td></tr>
<tr><td>0001</td><td>4</td></tr>
<tr><td>0010</td><td>8</td></tr>
<tr><td>0011</td><td>12</td></tr>
<tr><td>0100</td><td>16</td></tr>
<tr><td>0101</td><td>32</td></tr>
<tr><td>0110</td><td>64</td></tr>
<tr><td>0111</td><td>80</td></tr>
<tr><td>1000</td><td>96</td></tr>
<tr><td>1001</td><td>128</td></tr>
<tr><td>1010</td><td>160</td></tr>
<tr><td>1011</td><td>192</td></tr>
<tr><td>1100</td><td>256</td></tr>
<tr><td>1101</td><td>320</td></tr>
<tr><td>1110</td><td>384</td></tr>
<tr><td>1111</td><td>512</td></tr>
</table></td></tr>
<tr><td>3:0</td><td>SMTS1</td><td>Slow Sampling Time Control 1 (*ADC working clock cycle), applicable to all slow channels
<table>
<tr><th>SMTSx</th><th>Sampling cycles</th></tr>
<tr><td>0000</td><td>2</td></tr>
<tr><td>0001</td><td>4</td></tr>
<tr><td>0010</td><td>8</td></tr>
<tr><td>0011</td><td>12</td></tr>
<tr><td>0100</td><td>16</td></tr>
<tr><td>0101</td><td>32</td></tr>
<tr><td>0110</td><td>64</td></tr>
</table></td></tr>
</table>

bit	name	functional description	
		0111	80
		1000	96
		1001	128
		1010	160
		1011	192
		1100	256
		1101	320
		1110	384
		1111	512

38.8.9 ADC Channel Enable Register (ADC_CHER)

NAME	ADC_CHER							
Offset	0x20							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-	OPA	DAC	VDD/3	VBAT/3	AVREF	TS	VREF1P2
access	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				ECH19	ECH18	ECH17	ECH16
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ECH15	ECH14	ECH13	ECH12	ECH11	ECH10	ECH9	ECH8
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ECH7	ECH6	ECH5	ECH4	ECH3	ECH2	ECH1	ECH0
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31	--	RFU: **Reserved, read as 0**
30	OPA	OPA Output Measurement, write 1 to enable
29	DAC	DAC Output Measurement, write 1 to enable
28	VDD/3	VDD Voltage Division Measurement, write 1 to enable
27	VBAT/3	VBAT Voltage Division Measurement, write 1 to enable ***Note:*** *When the VBAT measurement channel is enabled, the VBAT divider resistor string is automatically enabled, and the power consumption will increase by one channel under the VBAT power supply.*
26	AVREF	Fast Reference Source Output Measurement, write 1 to enable
25	TS	Temperature Sensor Measurement Channel, write 1 to enable
24	VREF1P2	Internal Reference Voltage Measurement Channel, write 1 to

bit	name	functional description
		enable
23:20	--	RFU: **Reserved, read as 0**
19	ECH19	ADC_IN0~19 Measurement Channels, write 1 to enable Among them, ADC_IN0~13 can form 7 pairs of differential input pairs, and can also be used as 14 single-ended inputs ADC_IN14~19 can only be used as single-ended input
18	ECH18	
17	ECH17	
16	ECH16	
15	ECH15	
14	ECH14	
13	ECH13	
12	ECH12	
11	ECH11	
10	ECH10	
9	ECH9	
8	ECH8	
7	ECH7	
6	ECH6	
5	ECH5	
4	ECH4	
3	ECH3	
2	ECH2	
1	ECH1	
0	ECH0	

38.8.10 ADC Differential Channel Control Register (ADC_DCR)

NAME	ADC_DCR							
Offset	0x24							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	AINS						
access	U-0	R/W-000 0000						

bit	name	functional description
31:7	--	RFU: **Reserved, read as 0**

bit	name	functional description
6:0	AINS	Differential Input Pair Enable Register, used to configured whether ADC_IN0~ADC_IN13 form 7 differential pairs. 0 means single-ended input, 1 means differential input. (Analog Input Select) AINS[0]: 1 means that ADC_IN0 and ADC_IN7 are configured as differential inputs AINS[1]: 1 means that ADC_IN1 and ADC_IN8 are configured as differential inputs AINS[2]: 1 means that ADC_IN2 and ADC_IN9 are configured as differential inputs AINS[3]: 1 means that ADC_IN3 and ADC_IN10 are configured as differential inputs AINS[4]: 1 means that ADC_IN4 and ADC_IN11 are configured as differential inputs AINS[5]: 1 means that ADC_IN5 and ADC_IN12 are configured as differential inputs AINS[6]: 1 means that ADC_IN6 and ADC_IN13 are configured as differential inputs

38.8.11 ADC Data Register(ADC_DR)

NAME	ADC_DR							
Offset	0x28							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DATA[15:8]							
access	R-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DATA[7:0]							
access	R-0							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	DATA	ADCConversion Result When over-sampling averaging is not enabled, the result is 12 bits lower; when over-sampling averaging is enabled, the result

bit	name	functional description
		is 12~16 bits

38.8.12 AWD ThresholdRegister(ADC_HLTR)

NAME	ADC_HLTR							
Offset	0x2C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	AWD_HT[15:8]							
access	R/W-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	AWD_HT[7:0]							
access	R/W-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	AWD_LT[15:8]							
access	R/W-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	AWD_LT[7:0]							
access	R/W-0							

bit	name	functional description
31:16	AWD_HT	AWD monitors the high threshold, the maximum is 16bit, the software is set according to the actual number of bits required, and the high vacant number is kept at 0
15:0	AWD_LT	AWD monitors the low threshold, the maximum is 16bit, the software is set according to the actual number of bits required, and the high vacant number is kept at 0

39 Digital to Analog Converter (DAC)

39.1 Introduction

FM33LG0 integrates a 12-bit voltage output digitaltoanalog converter with a maximum output rate of 1Msps and supports DMA transmission. The DAC uses the VREF+ pin voltage as thereference, and the pin voltage can come frominternal reference source or external input.

DAC has an output drive buffer, which can provide greater drive capacity when the buffer is enabled; DAC output supports sample-and-hold mode to reduce the average operating current.

The main features of DAC are as follows:

- Operating voltage 1.8～5.5V
- Output signal amplitude 0~VDDA
- The highest output conversion rate is 1Msps (F_{DAC}=16Mhz)
- Support sample-and-hold mode
- Support DMA
- DAC output can be connected to comparator input

39.2 Block Diagram

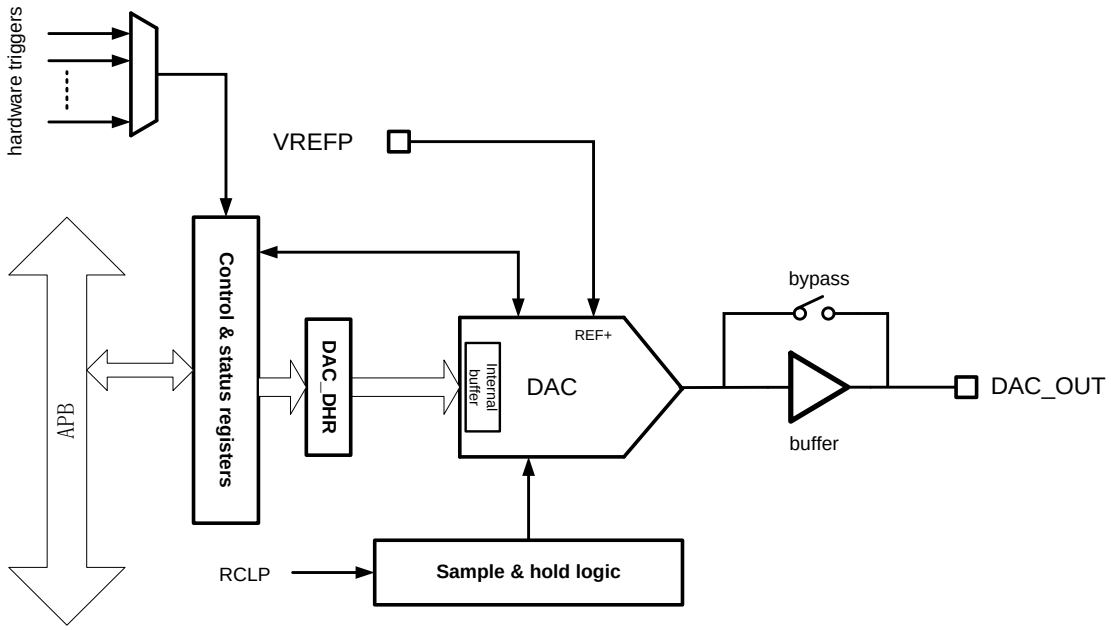


Figure 39–1 DAC Block Diagram

39.3 Pin Definition

The DAC output pins are multiplexed with ordinary GPIO, and the relevant pins need to be configured as analog functions before using the DAC output function.

Pin	Type	Description
VREFP	Input Reference Voltage	DAC working reference, equal to or lower than VDDA
VDDA	Analog Power	DAC power supply
VSSA	Analog Ground	DAC reference ground
DAC_OUT	Analog Output	DAC output signal

Table 39-1 DAC Related Pins

39.4 Connectivity of DAC Output to Pins and Other Modules

The DAC output can be directly output from the PC5 pin or output to the internal module.In order to avoid affecting the PC5 pin function when output to other modules, the feedback path of the DAC buffer is controlled by the switch channel of the PAD, as shown in the figure below.

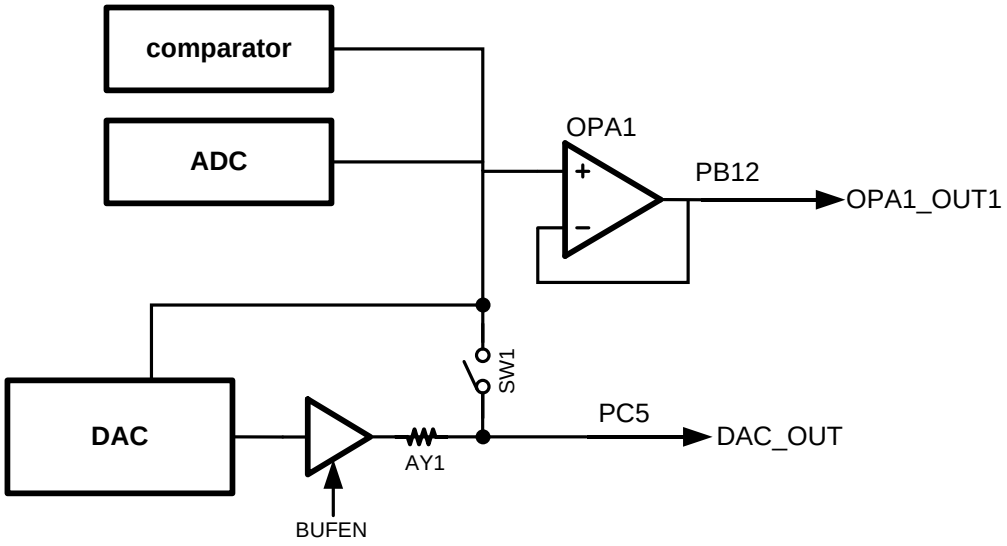


Figure 39–2 DAC Block Diagram

When the DAC_OUT output is connected to an off-chip load, BUFEN=1 and SW1 is closed (SWIEN=1).

When the DAC output is only connected to the on-chip module and not output from PC5, BUFEN=0 and SW1 is open (SWIEN=0).

39.5 Functional Description

39.5.1 Working Clock and Signal Timing

DAC uses the APB clock to work, and the update frequency of the digital code cannot exceed 1MHz, otherwise it cannot guarantee a good output dynamic performance. The DAC working clock comes from the CMU module, and the clock must be configured correctly before using the DAC.

When DAC is enabled, DAC has a start-up time. After the t_{WAKEUP} time has elapsed, the DAC output voltage will begin to establish. For the specific indicators of t_{WAKEUP}, refer to the Chapter3 Electrical Parameters.

Software or DMA writes the digital code that needs to be output into the DHR register.In the continuous output mode, the digital code in the DHR register will be directly copied to the internal buffer of the DAC and drive the DAC output level.In the trigger output mode, the digital code in the DHR register waits for the trigger event to arrive, and then updates it to the internal buffer of the DAC.Once the digital code update is completed, DAC work no longer requires a clock.

The DAC working timing diagram is as follows:

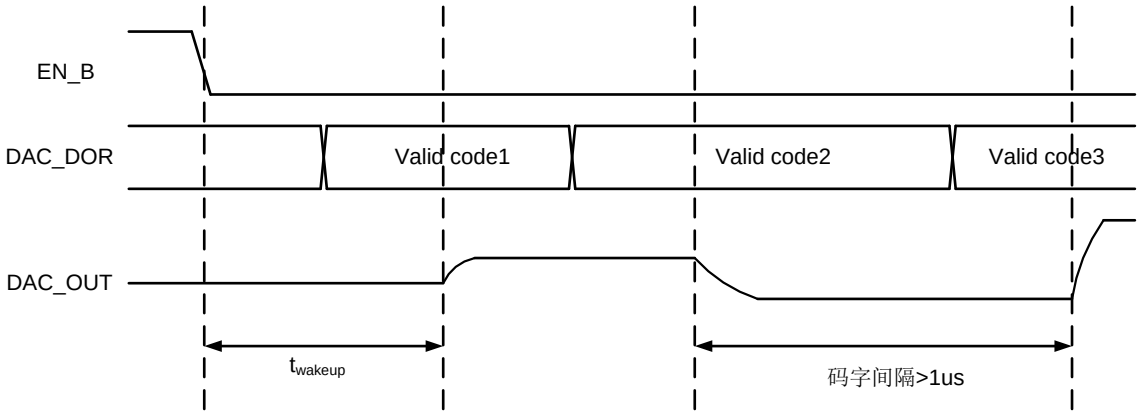


Figure 39–3 DAC Working Timing

39.5.2 DAC Output Mode

DAC output can adopt trigger mode or continuous mode.

If TRGEN (Trigger Enable) = 1, when the trigger signal arrives, the data in the DAC_DHR register is transferred to the internal buffer of DAC, and DAC will update the DAC_OUT output voltage.

If TRGEN=0, the software write operation to the DAC_DHR register will be mapped to the internal buffer of DAC at the same time, that is, the output level of DAC_OUT will be changed in real time.

The figure below shows the DAC continuous output mode. In continuous mode, the frequency of software updating the DAC_DHR register cannot exceed 1Mhz, otherwise the output cannot be guaranteed to be established correctly.

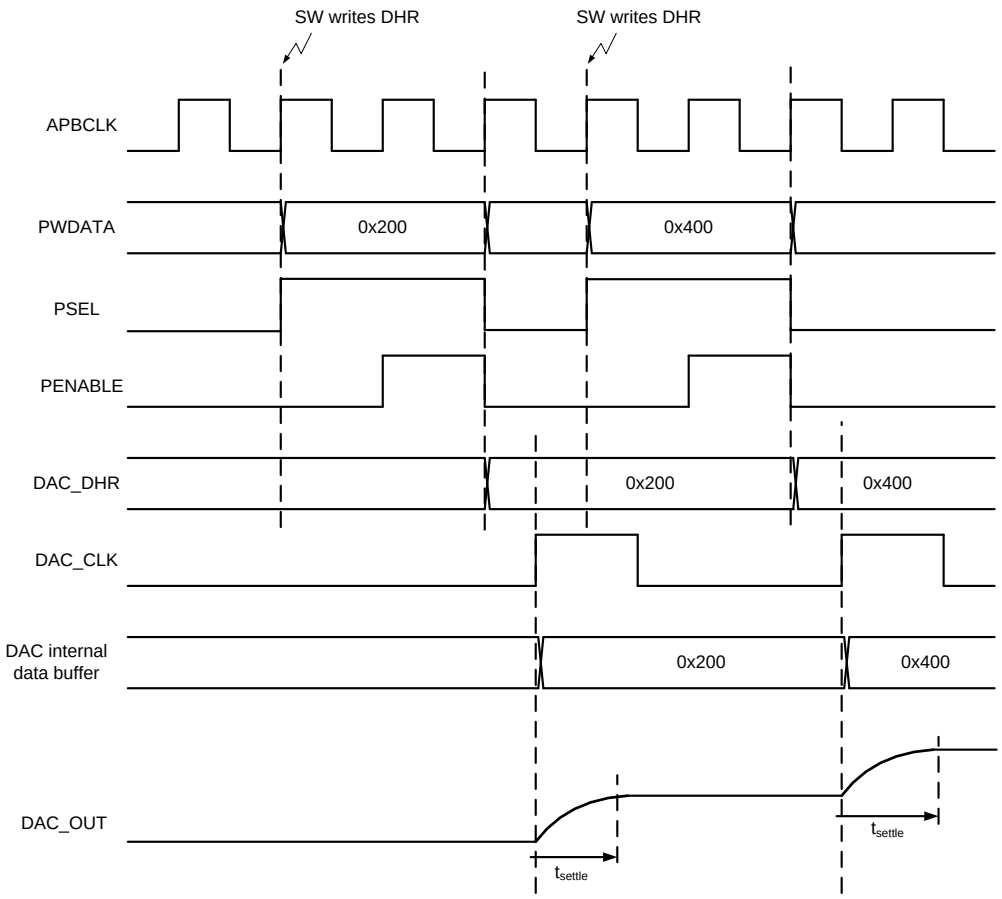


Figure 39–4 DAC Continuous Output Mode

The figure below shows the DAC trigger output mode. In the trigger output mode, the frequency of the trigger event cannot exceed 1Mhz, otherwise the output cannot be guaranteed to be established correctly.

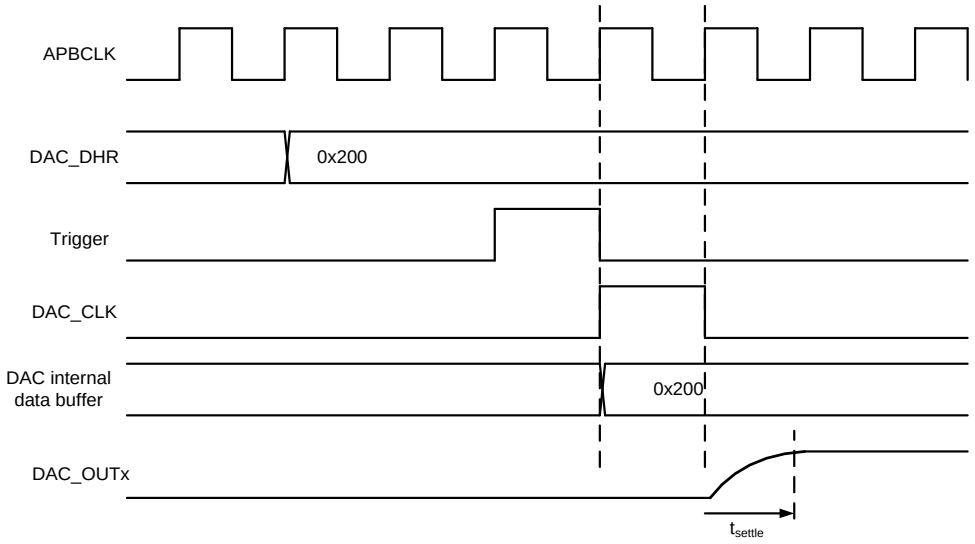


Figure 39–5 DAC Trigger Output Mode

After the internal data buffer of the DAC is updated, until the output voltage of DAC_OUT is stable,

there is a settling time. This time is related to the power supply voltage, output load and DAC output drive capability.

39.5.3 DAC Trigger Source Selection

DAC trigger mode supports hardware trigger or software trigger.

Through the TRGSEL register, you can select the trigger source used by the DAC. The software trigger is used by default. Software can realize the DAC trigger update by writing 1 to the SWTRG register.

When other trigger signal sources are selected, DAC will detect the rising edge of the input trigger signal and update the data in DAC_DHR to the internal buffer of the DAC.

The following table defines the trigger sources corresponding to different configurations of TRGSEL.

Trigger source	Type	TRGSEL[3:0]
SWTRG	Software trigger	0000
ATIM_TRGO	Internal timer trigger signal	0001
GPTIM1_TRGO		0010
GPTIM2_TRGO		0011
BSTIM16_TRGO		0100
LPTIM16_TRGO		0101
RFU	-	0110
		0111
		1000
		1001
		1010
		1011
EXTI[0]	Pin interrupt trigger	1100
EXTI[4]		1101
EXTI[8]		1110
EXTI[12]		1111

Table 39-2 DAC Trigger Source

39.5.4 DAC Output Voltage

The DAC output voltage is a linear conversion of the input digital code in the range of 0~VREF+.

The theoretical value of DAC output voltage is determined by the following formula:

$$DAC\_OUT = VREF \times \frac{DAC\_DHR}{4096}$$

39.5.5 DMA

DAC supports DMA function, which can only be used in trigger mode.

When the DMAEN register is set, the external hardware trigger signal (without software trigger) will cause the internal buffer of DAC to be updated to the content of DAC_DHR, and DAC will generate a DMA request. After the DMA controller receives the DAC request, it will automatically update the DAC_DHR register.

Before the current DMA request is completed, the new hardware trigger signal will be ignored, and the DMAERR flag register will be set.This situation shows that the trigger signal is too dense, and the system is too late to refresh the DAC output. At this time, the software should reduce the frequency of the trigger signal.

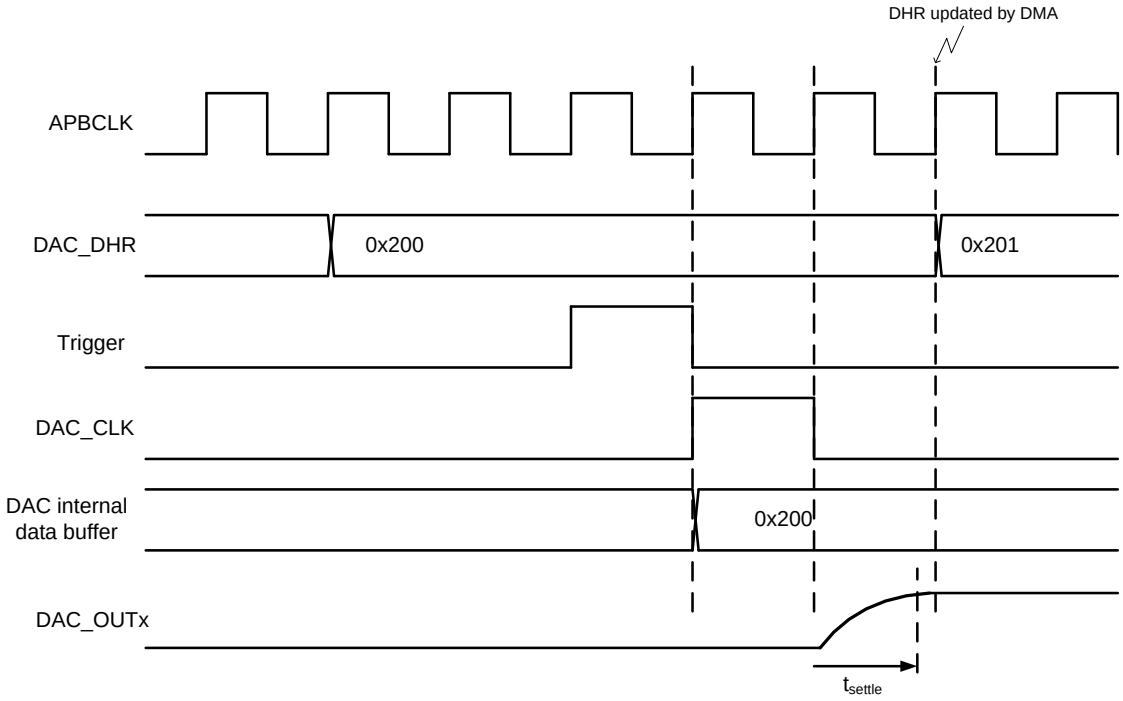


Figure 39–6 Update Data via DMA in Trigger Mode

39.5.6 Sample Hold

The DAC output supports normal mode and sample-and-hold mode. By enabling the DAC output buffer, a strong output drive capability can be obtained.

In the normal mode, the DAC keeps working continuously, that is, the DAC_OUT signal will be continuously driven by the DAC; this is the most common way of working, and there will be a stable current consumption during the DAC working period.

In the sample-and-hold mode, the DAC will only drive the DAC_OUT output for a period of time after receiving the trigger signal, and then automatically turn off the DAC, the DAC_OUT signal becomes a high-impedance state, and an off-chip capacitor is required to maintain the previous voltage state; internal timer After a programmable time interval, the DAC is re-enabled and DAC_OUT is driven;

through this intermittent driving method, with the off-chip sample and hold capacitor, the average current of the DAC can be greatly reduced.

The sample-and-hold mode can be divided into 2 stages:

- **Sampling phase**: DAC output drive is enabled, and the external sampling capacitor is charged to the target level. The charging time mainly depends on the DAC drive capability and the capacitance of the sampling capacitor. Through the TSMPL register, you can configure the time length of the sampling phase (based on the RCLP clock timing).
- **Holding phase**: In this phase, the DAC is turned off, the output is in a high-impedance state, and the sampling capacitor is kept in a voltage state. The length of the hold phase is configured by the THLD register (based on the RCLP clock timing).

Steps to enter sample and hold mode:

- Configure T_{SMPL} and T_{HLD} registers
- Enable DAC
- Set SHEN to start sample and hold mode
- The internal counter starts to count, when the count value is equal to TSMPL, the hardware automatically turns off the DAC enable
- The counter continues to count. When the count value is equal to TSMPL + THLD, the hardware re-enables the DAC and clears the counter to count again

Exit sample and hold mode:

- Clear the SHEN register (turn off the sample and hold and return to the always-enable mode)
- Clear the EN register (turn off the DAC, optional)

SHEN controls the counter logic. When SHEN=0, periodic_en=1, that is, the DAC is directly controlled by the DAC_CR.EN register.

When SHEN=1, the periodic_en signal can turn off the DAC periodically, while the DAC_CR.EN register remains set.

So the software sets DAC_CR.EN first, and sets SHEN to enter the sample and hold when needed.

After that, if the software clears SHEN, it will directly return to the always-enabled state.

If the software clears DAC_CR.EN, the DAC is completely turned off.

The sample-and-hold control circuit uses the RCLP clock to reduce operating power consumption. Before enabling the sample-and-hold mode, the software must ensure that the RCLP clock is enabled.

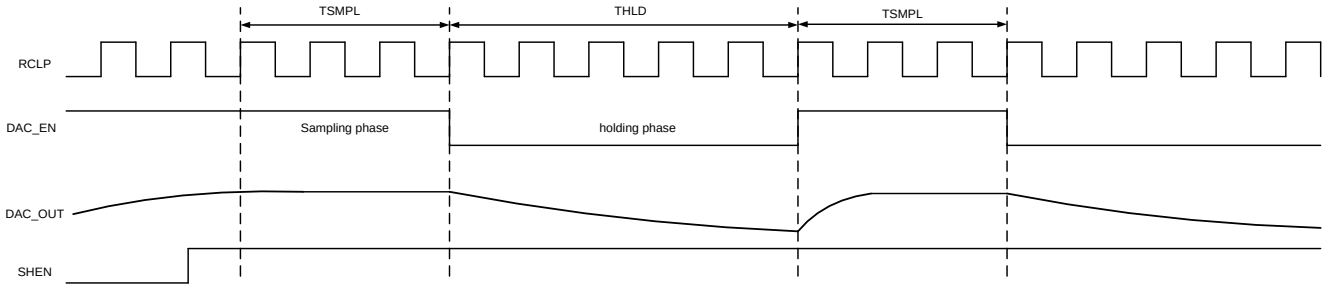


Figure 39–7 DAC Sample and Hold Output

The purpose of the sample-and-hold mode is to maintain a basically constant voltage on the external sampling capacitor with very little power consumption. In the continuous output mode, when any write operation to the DAC_DHR register occurs, the DAC internal buffer DAC_DOR data will also be updated in real time. At this time, no matter which stage the current sample and hold control circuit is in, a new sample will be triggered immediately. (That is, clear the sample and hold counter and restart the sampling phase)

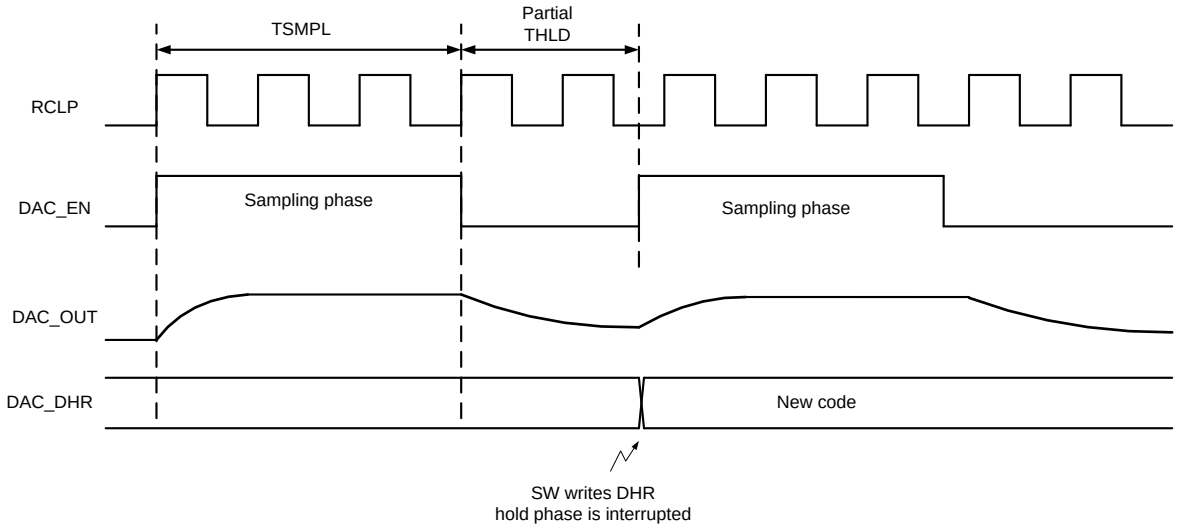


Figure 39–8 Update DHR During DAC Sample-and-Hold Output Process (TRGEN=0)

In the trigger output mode, the software updates the DHR register and does not immediately update the DAC output, but waits for the trigger event to arrive before clearing the counter and updating the DHR data to the DAC output buffer. If the application wants to update the output level immediately, it should write the SWTRIG register once after updating the DHR register.

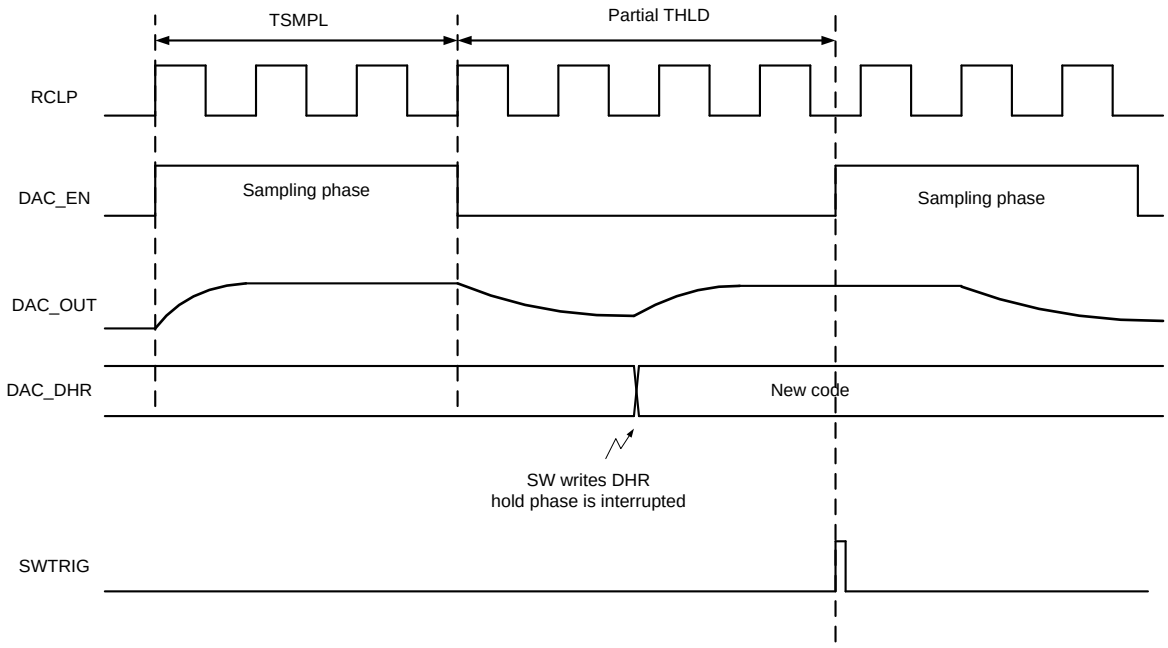


Figure 39–9 Update DHR During DAC Sample-and-Hold OutputProcess (TRGEN=1)

39.5.7 DAC Output Buffer

The DAC output buffer can be used to enhance the DAC output drive capability. For specific drive capability parameters, see Chapter 3 Electrical Parameters.

When outputting through Buffer, the rail-to-rail output range of 0~VREF+ cannot be realized, and the maximum output range can only support 0.2V~VREF-0.2V.

39.5.8 DAC in Low Power Mode

Mode	Description
LPRUN	Can work, can use DMA
Sleep/DeepSleep	Can work, but can only maintain a fixed output level
VBAT	Can not work

Table 39-3 DAC andLow Power Mode

39.6 Register

Offset	Name	Symbol
DAC(Base address: 0x40019800)		
0x00	DAC Control Register1	DAC_CR1
0x04	DAC Control Register2	DAC_CR2
0x08	DAC Config Register	DAC_CFGR
0x0C	DAC Software Trigger Register	DAC_SWTRGR
0x10	DAC Data Holding Register	DAC_DHR
0x14	DAC Interrupt Status Register	DAC_ISR
0x18	DAC Interrupt Enable Register	DAC_IER
0x1C	DAC Sample Hold Time Register	DAC_SHTR

39.6.1 DAC Control Register1 (DAC_CR1)

NAME	DAC_CR1							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							EN
access	U-0							R/W-0

bit	name	description
31:1	--	RFU: **Reserved, read as 0**
0	EN	DAC Enable 0: Disable DAC 1: Enable DAC

39.6.2 DAC Control Register2 (DAC_CR2)

NAME	DAC_CR2							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						DMAEN	TRGEN
access	U-0						R/W-0	R/W-0

bit	name	description
31:2	--	RFU: **Reserved, read as 0**
1	DMAEN	DMA Enable 0: Disable DMA 1: Enable DMA
0	TRGEN	Trigger Enable 0: Disable trigger 1: Enable trigger

39.6.3 DAC Config Register (DAC_CFGR)

NAME	DAC_CFGR							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							SHEN
access	U-0							R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	BUFEN	-	TRGSEL				-	SWIEN
access	R/W-0	U-0	R/W-0000				U-0	R/W-0

bit	name	description
31:9	--	RFU: **Reserved, read as 0**
8	SHEN	Sample Hold Enable 1: Use sample-and-hold output mode 0: Normal continuous output mode
7	BUFEN	Buffer Enable

bit	name	description
		0: Disable output buffer 1: Enable output buffer
6	--	RFU: **Reserved, read as 0**
5:2	TRGSEL	Trigger Select, the trigger source select must be modified when TRGEN is 0 See 39.5.3 DAC Trigger Source Select
1	--	RFU: **Reserved, read as 0**
0	SWIEN	Switch Enable 0: Disable switch 1: Enable switch

39.6.4 DAC Software Trigger Register (DAC_SWTRGR)

NAME	DAC_SWTRGR							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-							SWTRIG
access	U-0							W-0

bit	name	description
31:1	--	RFU: **Reserved, read as 0**
0	SWTRIG	Software Trigger, software set, when the data in DAC_DHR is loaded into the DAC_DOR register, the hardware is automatically cleared.

39.6.5 DAC Data Holding Register (DAC_DHR)

NAME	DAC_DHR							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-				DHR			
access	U-0				R/W-0000			
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DHR							
access	R/W-0000 0000							

bit	name	description
31:12	--	RFU: **Reserved, read as 0**
11:0	DHR	Data Holding Register

39.6.6 DAC Interrupt Status Register (DAC_ISR)

NAME	**DAC_ISR**							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				DMAERR	EOH	EOS	DOU
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0

bit	name	description
31:4	--	RFU: **Reserved, read as 0**
3	DMAERR	DMA Error Only valid when DMAEN=1, software writes 1 to clear
2	EOH	End of Holding Phase Only valid in sample-and-hold mode (SHEN=1), set when the hold phase ends, and cleared by software writing 1
1	EOS	End of Sampling Phase Only valid in sample-and-hold mode (SHEN=1), set after the end of the sampling phase, and cleared by software writing 1
0	DOU	Data Output Updated When the contents of the DHR register are written to DOR, this

bit	name	description
		flag is set, and the software writes 1 to clear it. It is valid only in trigger mode (TRGEN=1).

39.6.7 DAC Interrupt Enable Register (DAC_IER)

NAME	DAC_IER							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				DMAE_IE	EOH_IE	EOS_IE	DOU_IE
access	U-0				R/W-0	R/W-0	R/W-0	R/W-0

bit	name	description
31:4	--	RFU: **Reserved, read as 0**
3	DMAE_IE	DMA Error Interrupt Enable 1: Enable DMAERR interrupt 0: Disable DMAERR interrupt
2	EOH_IE	End of Holding Phase Interrupt Enable 1: Enable EOH interrupt 0: Disable EOH interrupt
1	EOS_IE	End of Sampling Phase Interrupt Enable 1: Enable EOS interrupt 0: Disable EOS interrupt
0	DOU_IE	Data Output Updated Interrupt Enable 1: Enable DOU interrupt 0: Disable DOU interrupt

39.6.8 DAC Sample Hold Time Register (DAC_SHTR)

NAME	DAC_SHTR							
Offset	0x1C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	THLD[15:8]							
access	R/W-0000 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	THLD[7:0]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TSMPL							
access	R/W-0000 0000							

bit	name	description
31:24	--	RFU: **Reserved, read as 0**
23:8	THLD	Time Holding in sample-and-hold mode, the unit is RCLP clock cycle 0x0: 1T 0x1: 2T …… 0xFFFF: 65536T
7:0	TSMPL	Time Sampling in sample-and-hold mode, the unit is RCLP clock cycle 0x0: 1T 0x1: 2T …… 0xFF: 256T

Note: *It is forbidden to modify the TSMPL and THLD registers when SHEN=1.*

40 Programmable Glue Logic (PGL)

40.1 Introduction

Programmable Glue Logic (PGL) is a simple programmable logic based on a lookup table (LUT). Its input and output can be connected to chip pins and internal signals to realize some simple glue logic. In some applications, it can help system design to reduce logic devices on the PCB.

Each LUT contains 4 inputs, 1 outputs, 1 truth table, and optional synchronization/filtering circuit. Users can obtain the desired combinational logic output expression by programming the truth table. Each input signal can be individually shielded.

The basic characteristics of PGL are as follows:

- Realize simple glue logic and simplify PCB
- Four 4-inputs lookup table
- Logical expressions such as AND, NAND, OR, NOR, XOR, NOT can be realized through truth table programming
- Timing synchronization or filtering
- Flexible LUT input selection: IO, internal signal, other LUT output
- Output can be triggered by connecting to IO or other peripherals

40.2 Block Diagram

The structure of the LUT is shown in the figure below.

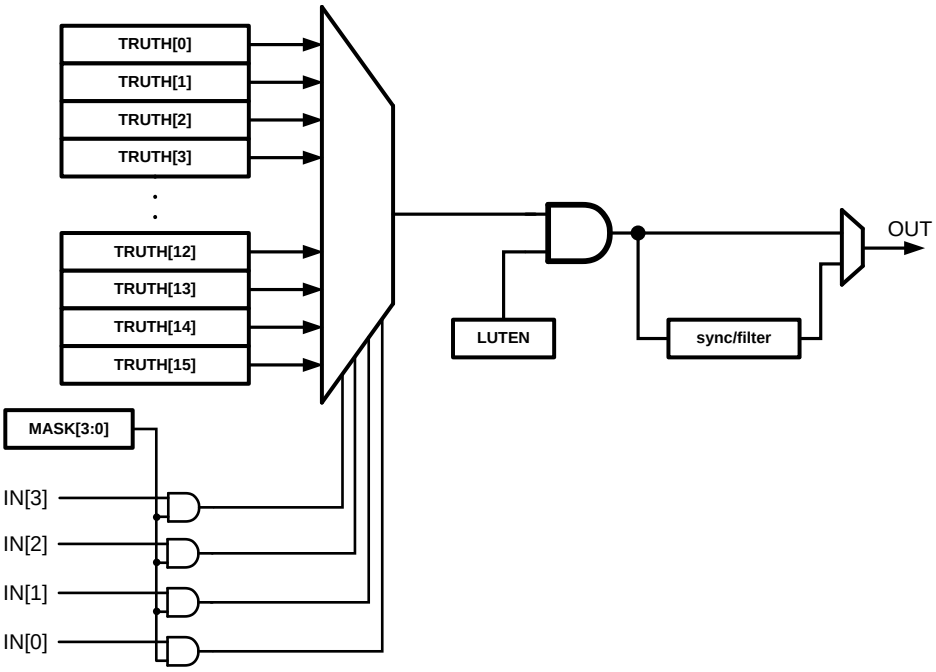


Figure 40–1 LUT Block Diagram

The PGL module contains a total of 4 LUTs.

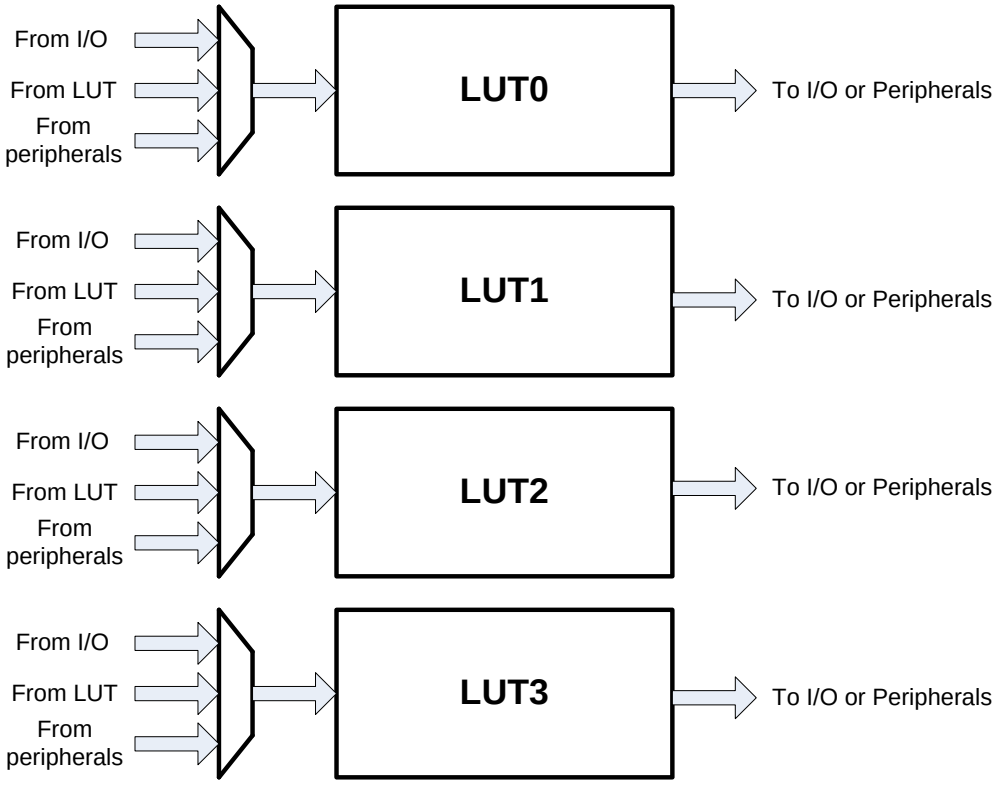


Figure 40–2 PGL Block Diagram

40.3 Pin Definition

The input and output pins of PGL are multiplexed with general GPIO, and the corresponding IO needs to be configured as GPIO input or input to use the PGL function.

Pin	Type	Description
PC2	GPIO Input	LUT0 inputs 0
PC3		LUT0 inputs 1
PC4		LUT0 inputs 2
PC5		LUT0 inputs 3
PC6		LUT1 inputs 0
PC7		LUT1 inputs 1
PC8		LUT1 inputs 2
PC9		LUT1 inputs 3
PB7		LUT2 inputs 0
PB8		LUT2 inputs 1
PB9		LUT2 inputs 2
PB10		LUT2 inputs 3
PB11		LUT3 inputs 0
PB12		LUT3 inputs 1
PB13		LUT3 inputs 2
PB14		LUT3 inputs 3
PC0	Extra Digital Function	LUT0 output
PC1		LUT1 output
PA1		LUT2 output
PA0		LUT3 output

Table 40-1 PGL Related Pins

When using LUT pin input, you need to configure the corresponding pin as a GPIO input, clear the input MASK register, and enable LUT.

40.4 Functional Description

40.4.1 LUT Truth Table

The desired combinational logic function can be obtained by querying the data in the LUT truth table through inputting state combination.

Each LUT supports 4 inputs, when the input is used as the address, 16 outputs can be queried. The LUT itself is a 4bit address addressed storage space. Register TRUTH[15:0] saves 16 output data, and the address correspondence is shown in the table below.

Input	Address	Output Data
0000	0	TRUTH[0]
0001	1	TRUTH[1]
0010	2	TRUTH[2]
0011	3	TRUTH[3]
0100	4	TRUTH[4]
0101	5	TRUTH[5]
0110	6	TRUTH[6]
0111	7	TRUTH[7]
1000	8	TRUTH[8]
1001	9	TRUTH[9]
1010	10	TRUTH[10]
1011	11	TRUTH[11]
1100	12	TRUTH[12]
1101	13	TRUTH[13]
1110	14	TRUTH[14]
1111	15	TRUTH[15]

Table 40-2 Lookup Table (LUT)

For example, if you want to realize a 2-input NAND logic function through LUT0, you can use the following configuration:

- Mask input 2 and input 3 of LUT0 to 0.
- Input 0 and input 1 of LUT0 are connected to GPIO
- The lower 4 bits of the truth table register of LUT0 are written to 0111

Input	Address	Output Data
00	0	TRUTH[0]=1
01	1	TRUTH[1]=1
10	2	TRUTH[2]=1
11	3	TRUTH[3]=0

The above truth table corresponds to a 2-input NAND.

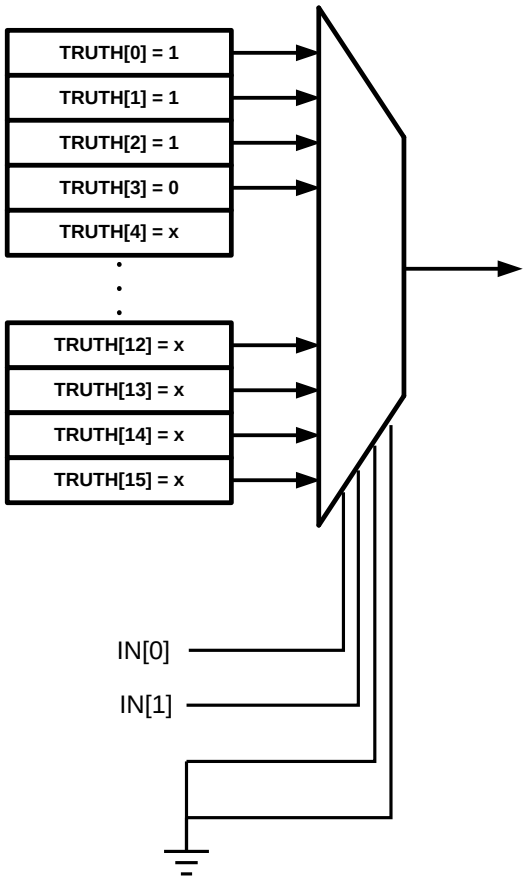


Figure 40–3 LUT Implementation of 2-Input NAND Diagram

40.4.2 LUT Truth Table

Each LUT has 4 inputs, and each input can come from pins, chip peripherals, or other LUT outputs.

Through the INSEL register, you can select the LUT input signal source, please refer to the register chapter for details.

40.4.3 LUT Output

The output of the LUT can be connected to pins, or other LUT inputs. As shown in the table below.

LUT Output	Pin	LUT Input
LUT0 output	PC0	LUT1,LUT2,LUT3
LUT1 output	PC1	LUT2,LUT3
LUT2 output	PA1	LUT3
LUT3 output	PA0	-

Table 40-3 LUT Output Connection

40.4.4 Filtering and Sampling

LUT supports filtering and synchronous sampling of the output, and its block diagram is as follows.

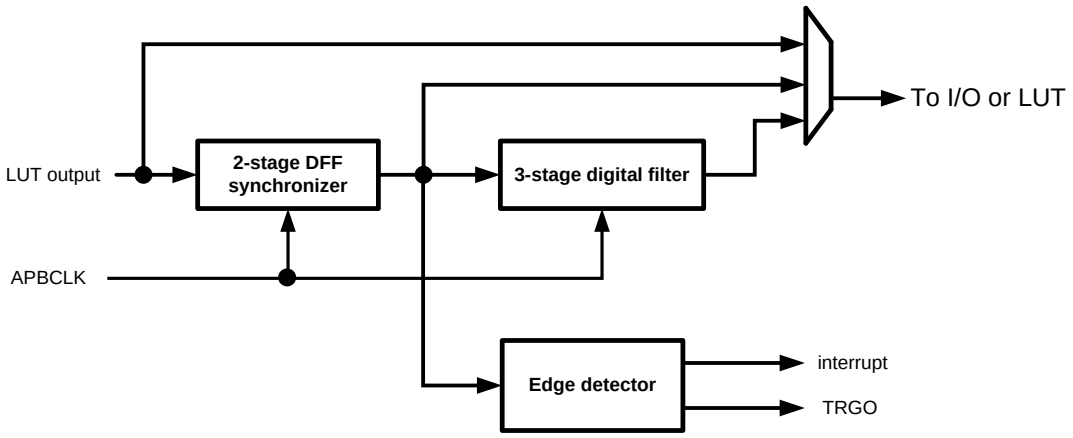


Figure 40–4 Output Filtering and Sampling

Digital filtering adopts the following method. When APBCLK is continuously sampled to 3 same levels, it is regarded as a legal level, otherwise the filtered output will not change.

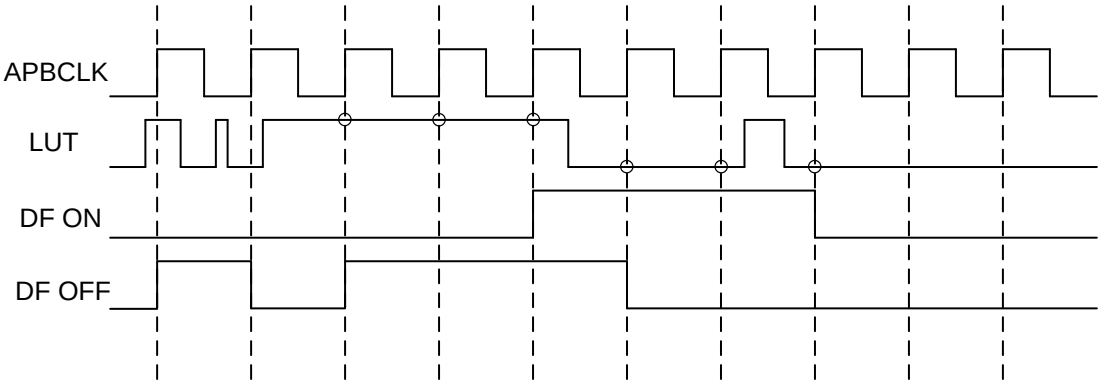


Figure 40–5 Digital Filtering

According to the register configuration, the application can choose the output of the LUT to be the combinational logic output, the output after the synchronous sampling of APBCLK, or the output after the digital filtering after sampling.

After synchronous sampling, through the edge generation circuit, a rising or falling edge pulse signal with a cycle width of APBCLK can be generated. This synchronization pulse signal can be used for trigger signals of other peripherals, or to generate interrupts.

40.4.5 Interrupt and Trigger

After the LUT output is sampled and edge selected, it can generate an interrupt flag or output a trigger signal.

The LUT peripheral clock must be enabled to generate an interrupt or trigger.

The 4 LUTs of PGL can generate 4 trigger signal outputs, and the fan-out relationship is as follows:

Trigger Output	Receiving Module
LUT0_TRGO	ADC, GPTIM2
LUT1_TRGO	ADC, GPTIM1
LUT2_TRGO	ADC, GPTIM2
LUT3_TRGO	ADC, GPTIM1

Table 40-4 LUT Output Connection

40.4.6 Low Power Mode

PGL can be used in low power mode, but it should be noted that filtering and sampling functions need APB clock, so PGL cannot be used in sleep mode.

If you only use the combinational logic output function of LUT, the application can disable the PGL peripheral module clock after configuring the registers to save power.

40.5 Register

Offset	Name	Symbol
PGL(Base address: 0x40016C00)		
0x00	PGL Control Register	PGL_CR
0x04	PGL Config Register0	PGL_CFGR0
0x08	PGL Config Register1	PGL_CFGR1
0x0C	PGL Config Register2	PGL_CFGR2
0x10	PGL Config Register3	PGL_CFGR3
0x14	PGL Interrupt Enable Register	PGL_IER
0x18	PGL Interrupt Status Register	PGL_ISR
0x1C	Look Up Table0	PGL_LUT0
0x20	Look Up Table1	PGL_LUT1
0x24	Look Up Table2	PGL_LUT2
0x28	Look Up Table3	PGL_LUT3

40.5.1 PGL Control Register (PGL_CR)

NAME	PGL_CR							
Offset	0x00							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				LUTEN[3:0]			
access	U-0				R/W-0000			

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3:0	LUTEN	LUT Enable Register, respectively control LUT0~3 1: Enable LUT[x] 0: Disable LUT[x]

40.5.2 PGL Config Register0 (PGL_CFGR0)

NAME	PGL_CFGR0							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				EDGESEL		OUTSEL	
access	U-0				R/W-00		R/W-00	
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
access	R/W-00		R/W-00		R/W-00		R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				MASK[3:0]			
access	U-0				R/W-0000			

bit	name	functional description
31:20	--	RFU: **Reserved, read as 0**
19:18	EDGESEL	LUT0 Valid Edge Select 00: Rising edge generates interrupt and TRGO 01: Falling edge generates interrupt and TRGO 10: Rising and falling edge generate interrupt and TRGO 11: No interrupt and TRGO
17:16	OUTSEL	LUT0 Output Select 00: Combinational logic output 01: Synchronous sampling output 10: Digital filter output 11: RFU
15:14	IN3SEL	LUT0 Input 3 Select Register 00: GPIO 01: LPTIM16_CH1 10: LPTIM32_CH1 11: ATIM_CH1
13:12	IN2SEL	LUT0 Input 2 Select Register 00: GPIO 01: RFU 10: COMP3_OUT 11: GPTIM2_CH1
11:10	IN1SEL	LUT0 Input 1 Select Register 00: GPIO 01: RFU 10: COMP2_OUT 11: GPTIM1_CH1

bit	name	functional description
9:8	IN0SEL	LUT0 Input 0 Select Register 00: GPIO 01: RFU 10: COMP1_OUT 11: GPTIM0_CH1
7:4	--	RFU: **Reserved, read as 0**
3:0	MASK	LUT0 Input Mask Register 1: Mask the corresponding input (i.e. fixed to 0) 0: Do not mask input

40.5.3 PGL Config Register1 (PGL_CFGR1)

NAME	PGL_CFGR1							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				EDGESEL		OUTSEL	
access	U-0				R/W-00		R/W-00	
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
access	R/W-00		R/W-00		R/W-00		R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				MASK[3:0]			
access	U-0				R/W-0000			

bit	name	functional description
31:20	--	RFU: **Reserved, read as 0**
19:18	EDGESEL	LUT1 Valid Edge Select 00: Rising edge generates interrupt and TRGO 01: Falling edge generates interrupt and TRGO 10: Rising and falling edge generate interrupt and TRGO 11: No interrupt and TRGO
17:16	OUTSEL	LUT1 Output Select 00: Combinational logic output 01: Synchronous sampling output 10: Digital filter output 11: RFU
15:14	IN3SEL	LUT1 Input 3 Select Register 00: GPIO 01: LPTIM16_CH1

bit	name	functional description
		10: LPTIM32_CH1 11: ATIM_OC1REF
13:12	IN2SEL	LUT1 Input 2 Select Register 00: GPIO 01: RFU 10: COMP3_OUT 11: GPTIM2_CH1
11:10	IN1SEL	LUT1 Input 1 Select Register 00: GPIO 01: RFU 10: COMP2_OUT 11: GPTIM1_CH1
9:8	IN0SEL	LUT1 Input 0 Select Register 00: GPIO 01: LUT0 output 10: COMP1_OUT 11: GPTIM0_CH1
7:4	--	RFU: **Reserved, read as 0**
3:0	MASK	LUT1 Input Mask Register 1: Mask the corresponding input (i.e. fixed to 0) 0: Do not mask input

40.5.4 PGL Config Register2 (PGL_CFGR2)

NAME	PGL_CFGR2							
Offset	0x0C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-				EDGESEL		OUTSEL	
access	U-0				R/W-00		R/W-00	
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
access	R/W-00		R/W-00		R/W-00		R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				MASK[3:0]			
access	U-0				R/W-0000			

bit	name	functional description
31:20	--	RFU: **Reserved, read as 0**
19:18	EDGESEL	LUT2 Valid Edge Select

bit	name	functional description
		00: Rising edge generates interrupt and TRGO 01: Falling edge generates interrupt and TRGO 10: Rising and falling edge generate interrupt and TRGO 11: No interrupt and TRGO
17:16	OUTSEL	LUT2 Output Select 00: Combinational logic output 01: Synchronous sampling output 10: Digital filter output 11: RFU
15:14	IN3SEL	LUT2 Input 3 Select Register 00: GPIO 01: LPTIM16_CH1 10: LPTIM32_CH1 11: ATIM_CH1
13:12	IN2SEL	LUT2 Input 2 Select Register 00: GPIO 01: RFU 10: COMP3_OUT 11: GPTIM2_CH1
11:10	IN1SEL	LUT2 Input 1 Select Register 00: GPIO 01: RFU 10: COMP2_OUT 11: GPTIM1.OC1REF
9:8	IN0SEL	LUT2 Input 0 Select Register 00: GPIO 01: LUT0 output 10: COMP1_OUT 11: GPTIM0_CH1
7:4	--	RFU: **Reserved, read as 0**
3:0	MASK	LUT2 Input Mask Register 1: Mask the corresponding input (i.e. fixed to 0) 0: Do not mask input

40.5.5 PGL Config Register3 (PGL_CFGR3)

NAME	PGL_CFGR3							
Offset	0x10							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16

name	-				EDGESEL		OUTSEL	
access	U-0				R/W-00		R/W-00	
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	IN3SEL		IN2SEL		IN1SEL		IN0SEL	
access	R/W-00		R/W-00		R/W-00		R/W-00	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				MASK[3:0]			
access	U-0				R/W-0000			

bit	name	functional description
31:20	--	RFU: **Reserved, read as 0**
19:18	EDGESEL	LUT3 Valid Edge Select 00: Rising edge generates interrupt and TRGO 01: Falling edge generates interrupt and TRGO 10: Rising and falling edge generate interrupt and TRGO 11: No interrupt and TRGO
17:16	OUTSEL	LUT3 Output Select 00: Combinational logic output 01: Synchronous sampling output 10: Digital filter output 11: RFU
15:14	IN3SEL	LUT3 Input 3 Select Register 00: GPIO 01: LPTIM16_CH1 10: LPTIM32_CH1 11: ATIM_CH1
13:12	IN2SEL	LUT3 Input 2 Select Register 00: GPIO 01: LUT2 output 10: COMP3_OUT 11: GPTIM2_CH1
11:10	IN1SEL	LUT3 Input 1 Select Register 00: GPIO 01: LUT1 output 10: COMP2_OUT 11: GPTIM1_CH1
9:8	IN0SEL	LUT3 Input 0 Select Register 00: GPIO 01: LUT0 output 10: COMP1_OUT 11: GPTIM0_CH1
7:4	--	RFU: **Reserved, read as 0**
3:0	MASK	LUT3 Input Mask Register 1: Mask the corresponding input (i.e. fixed to 0)

bit	name	functional description
		0: Do not mask input

40.5.6 PGL Interrupt Enable Register (PGL_IER)

NAME	PGL_IER							
Offset	0x14							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				LUTIE[3:0]			
access	U-0				R/W-0000			

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3:0	LUTIE	LUT Interrupt Enable Register, respectively controlLUT0~3 1: Enable LUT[x] interrupt 0: Disable LUT[x] interrupt

40.5.7 PGL Interrupt Status Register (PGL_ISR)

NAME	PGL_ISR							
Offset	0x18							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-				LUTIF[3:0]			
access	U-0				R/W-0000			

bit	name	functional description
31:4	--	RFU: **Reserved, read as 0**
3:0	LUTIF	LUT Interrupt Status Register, hardware set, write 1 to clear by software

40.5.8 Look Up Table0 (PGL_LUT0)

NAME	PGL_LUT0							
Offset	0x1C							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	TRUTH[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TRUTH[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	TRUTH	LUT0 truth table, software can write any value to obtain the required logical combination

40.5.9 Look Up Table1 (PGL_LUT1)

NAME	PGL_LUT1							
Offset	0x20							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	TRUTH[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TRUTH[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	TRUTH	LUT1 truth table, software can write any value to obtain the required logical combination

40.5.10 Look Up Table2 (PGL_LUT2)

NAME	PGL_LUT2							
Offset	0x24							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	TRUTH[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TRUTH[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	TRUTH	LUT2 truth table, software can write any value to obtain the required logical combination

40.5.11 Look Up Table3 (PGL_LUT3)

NAME	PGL_LUT3							
Offset	0x28							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	TRUTH[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	TRUTH[7:0]							

access	R/W-0000 0000

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	TRUTH	LUT3 truth table, software can write any value to obtain the required logical combination

41 General-purpose I/Os (GPIO)

41.1 Introduction

Main features of the I/O ports:

- The GPIO pin input voltage cannot be higher than the power supply voltage
- GPIO digital inputs with Schmitt characteristics
- Some GPIO inputs support analog filtering
- Some GPIO inputs support digital filtering
- GPIOs can be configured as pull-up outputs and open-drain outputs
- Keep configuration mode in Sleep/DeepSleep mode
- PH15 is powered by VAO power supply

41.2 Pin Type

Most pins of FM33LG0 support input and output, digital peripheral functions, analog peripheral channels, controlled pull-up resistors, and controlled open-drain output functions. In addition to the above functions, the strong drive pin has an enhanced push-pull output drive capability.

41.2.1 GPIO, input and output enable, controlled pull-up resistor, controlled open-drain output

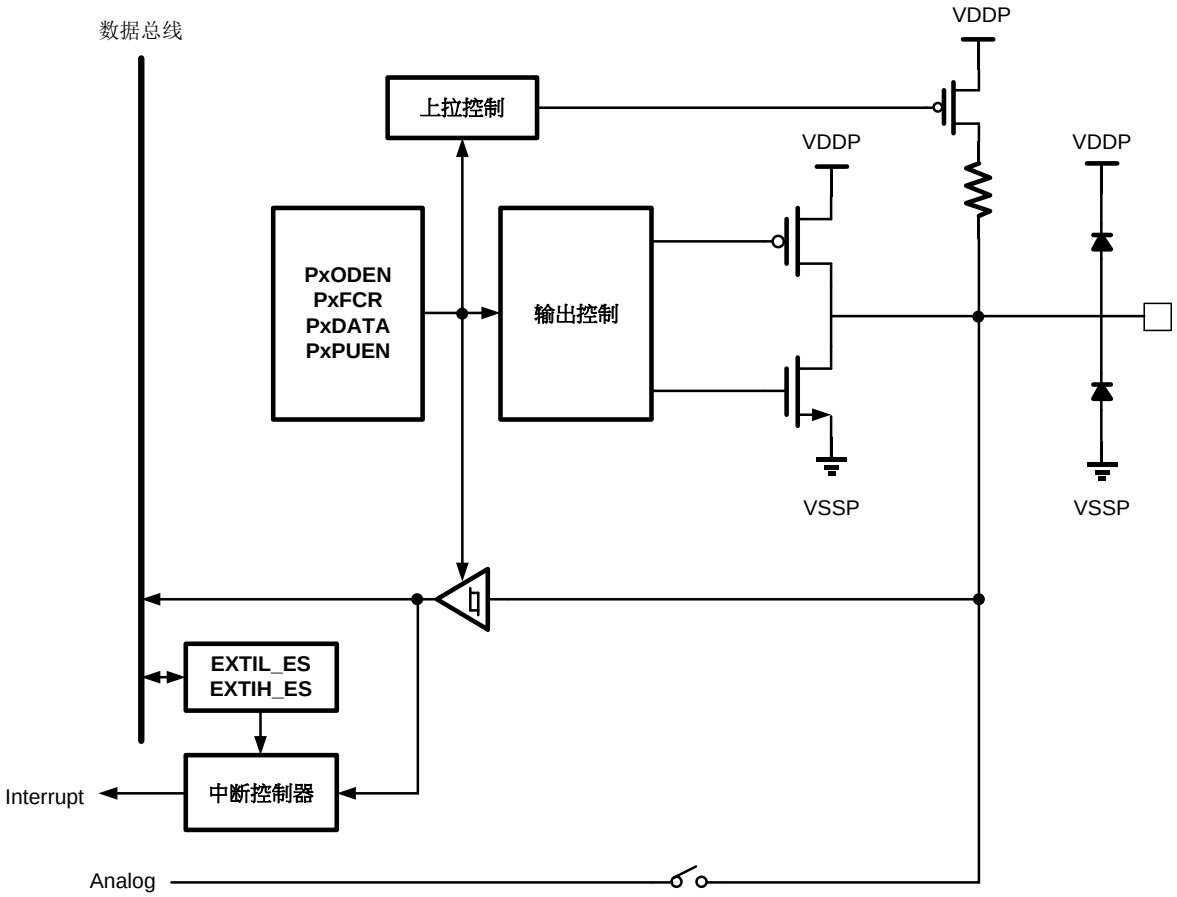


Figure 41–1 GPIO Block Diagram

The control logic is defined as follows:

Registers					PAD Interface		
FCR	INEN	ODEN	PUEN	DATA	INPUT_EN	OUTPUT_EN	PUEN
00	0	x	0/1	x	0	0	0/1
	1				1		
01	x	0	0/1	x	0	1	0/1
	x	1		0	0	1	
				1	0	0	
10	x	x	0/1	Peripheral input function	1	0	0/1
	0	0		Peripheral push-pull output	0	1	
	1				1	1	

				function			
	0	1		Peripheral open-drain output 0	0	1	
	1				1	1	
	0			Peripheral open-drain output1	0	0	
	1				1	0	
11	x	x	x	x	0	0	0

Table 41-1 GPIO Function Logic Definition

41.2.2 GPIO, input and output enable, 2 controlled pull-up resistor, controlled open-drain output (PC12-7816 data port only)

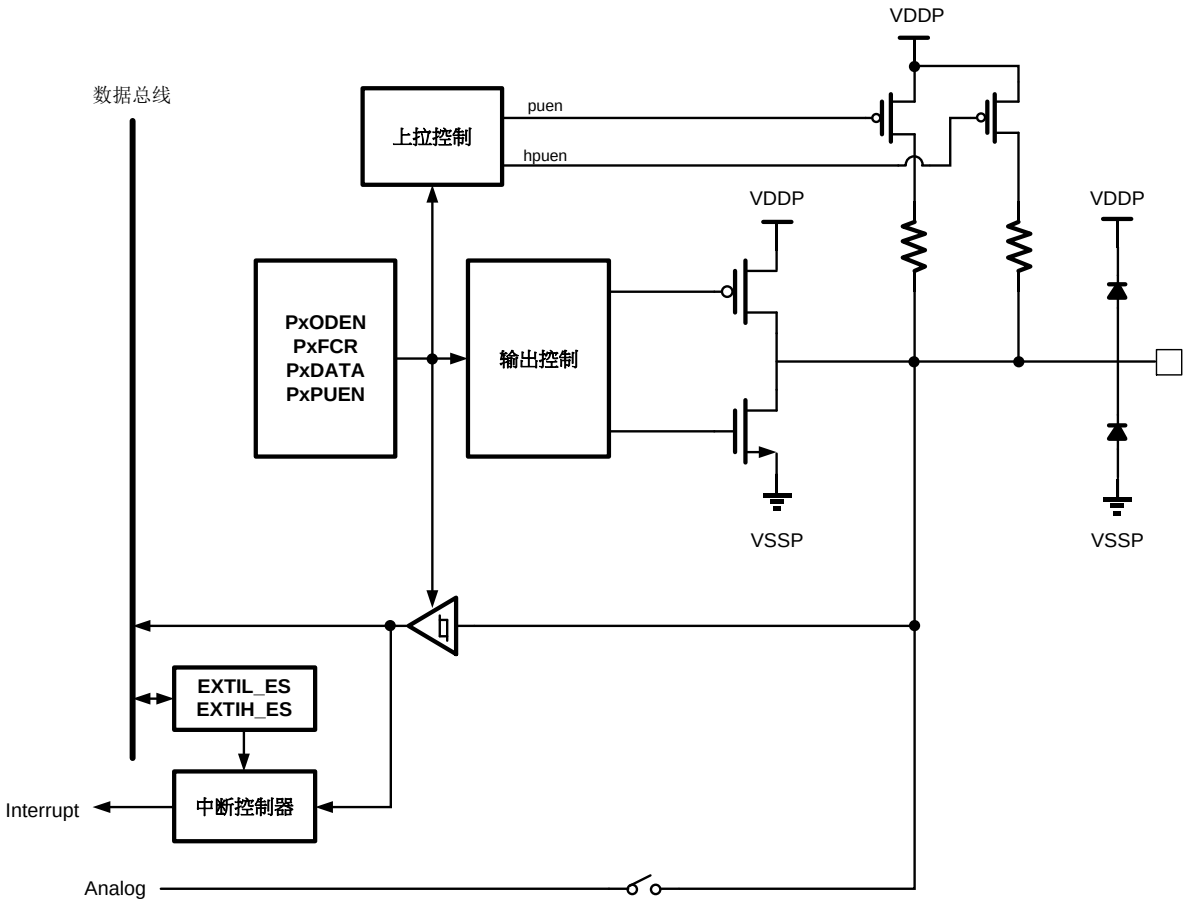


Figure 41–2 GPIO (Two Pull-ups) Block Diagram

The register control logic of the above IO is the same as other GPIOs, and the parallel pull-up control (strong pull-up) is only controlled automatically by the 7816 module.

41.3 IO Function Definition

Most of the chip pins are digital-analog mixed IO, and each GPIO has 4bit control registers: FCR[1:0], PUEN, ODEN, where FCR is used to select IO pin function, defined as follows:

FCR: Function Control Register	PAD function
00	GPIO input
01	GPIO output
10	Digital Function
11	Analog

Table 41-2 FCR Definition

41.3.1 GPIO Input

When a GPIO is configured for input function and the corresponding input enable register is set:

- The output drive buffer is turned off
- Schmitt trigger is enabled
- Pull-up resistors are enabled or disabled by the GPIOx_PUEN register
- GPIOx_DIN register directly responds to the level status on the IO

41.3.2 GPIO Output

When a GPIO is configured as an output function and the corresponding output enable register is set:

- Output drive buffer enable
  - Open-drain output mode (GPIOx_ODEN=1): IO drive low at output 0, IO drive buffer off at output 1
  - Push-pull output mode (GPIOx_ODEN=0): IO drive low when output 0, IO drive high when output 1
- The pull-up resistor is controlled by the GPIOx_PUEN register to enable or disable
- Software reads the GPIOx_DO register to get the last written value

41.3.3 Digital Peripheral Functions

When a GPIO is configured for a digital peripheral function:

- The input or output direction of the IO is determined by the function of the connected peripheral
- GPIOx_ODEN controls whether the output is open-drain or push-pull output

- The pull-up resistor is enabled or disabled by the GPIOx_PUEN register
- When the GPIOx_INEN register is set, software can read the GPIOx_DIN register to obtain the level status on the IO

Some of the pins support multiple digital peripheral functions, then additional control registers (GPIOx_DFS) are required to distinguish them.

The pins that support multiple digital peripheral functions are:

GPIO	Digital Feature1 GPIOx_DFS[x]=0	Digital Feature2 GPIOx_DFS[x]=1	Additional AFSEL
PA0	UART4_RX	LUT3_OUT	GPIOA_DFS[0]
PA1	UART4_TX	LUT2_OUT	GPIOA_DFS[1]
PA2	UART0_RX	LPUART0_RX	GPIOA_DFS[2]
PA3	UART0_TX	LPUART0_TX	GPIOA_DFS[3]
PA4	GPT1_CH3	-	-
PA5	GPT1_CH4	-	-
PA6	CAN_RX	-	-
PA7	CAN_TX	-	-
PA8	SPI1_SSN	LPT32_CH1	GPIOA_DFS[8]
PA9	SPI1_SCK	LPT32_CH2	GPIOA_DFS[9]
PA10	LPT32_ETR	-	-
PA11	SCL	LPUART2_RX	GPIOA_DFS[11]
PA12	SDA	LPUART2_TX	GPIOA_DFS[12]
PA13	UART0_RX	LPUART0_RX	GPIOA_DFS[13]
PA14	UART0_TX	LPUART0_TX	GPIOA_DFS[14]
PA15	COMP3_OUT	-	-
PB0	SPI1_MISO	UART3_RX	GPIOB_DFS[0]
PB1	SPI1_MOSI	UART3_TX	GPIOB_DFS[1]
PB2	UART4_RX	ATIM_CH1N	GPIOB_DFS[2]
PB3	UART4_TX	ATIM_CH2N	GPIOB_DFS[3]
PB4	LPUART2_RX	ATIM_CH1	GPIOB_DFS[4]
PB5	LPUART2_TX	ATIM_CH2	GPIOB_DFS[5]
PB6	SPI2_SSN	ATIM_CH3	GPIOB_DFS[6]
PB7	SPI2_SCK	ATIM_CH4	GPIOB_DFS[7]
PB8	SPI0_SSN	ATIM_CH3N	GPIOB_DFS[8]
PB9	SPI0_SCK	GPT0_ETR	GPIOB_DFS[9]
PB10	SPI0_MISO	GPT0_CH1	GPIOB_DFS[10]
PB11	SPI0_MOSI	GPT0_CH2	GPIOB_DFS[11]
PB12	FOUT1	ATIM_ETR	GPIOB_DFS[12]
PB13	UART1_RX	LPUART1_RX	GPIOB_DFS[13]

GPIO	Digital Feature1 GPIOx_DFS[x]=0	Digital Feature2 GPIOx_DFS[x]=1	Additional AFSEL
PB14	UART1_TX	LPUART1_TX	GPIOB_DFS[14]
PB15	SCL	SPI2_MISO	GPIOB_DFS[15]
PC0	GPT1_CH1	LUT0_OUT	GPIOC_DFS[0]
PC1	GPT1_CH2	LUT1_OUT	GPIOC_DFS[1]
PC2	UART1_RX	LPUART1_RX	GPIOC_DFS[2]
PC3	UART1_TX	LPUART1_TX	GPIOC_DFS[3]
PC4	UART5_RX	COMP1_OUT	GPIOC_DFS[4]
PC5	UART5_TX	COMP2_OUT	GPIOC_DFS[5]
PC6	GPT1_ETR	-	-
PC7	SPI2_SSN	-	-
PC8	SPI2_SCK	-	-
PC9	SPI2_MISO	-	-
PC10	SPI2_MOSI	-	-
PC11	U7816_CLK	GPT0_CH3	GPIOC_DFS[11]
PC12	U7816_IO	GPT0_CH4	GPIOC_DFS[12]
PC13	U7816_CLK	LPT16_CH1	GPIOC_DFS[13]
PC14	U7816_IO	LPT16_CH2	GPIOC_DFS[14]
PC15	LPT32_CH3	CAN_RX	GPIOC_DFS[15]
PD0	UART5_RX	GPT2_CH1	GPIOD_DFS[0]
PD1	UART5_TX	GPT2_CH2	GPIOD_DFS[1]
PD2	SPI1_SSN	GPT2_CH3	GPIOD_DFS[2]
PD3	SPI1_SCK	GPT2_CH4	GPIOD_DFS[3]
PD4	SPI1_MISO	GPT2_ETR	GPIOD_DFS[4]
PD5	SPI1_MOSI	LPT16_ETR	GPIOD_DFS[5]
PD6	ATIM_BRK2	-	-
PD7	SWCLK	UART3_RX	GPIOD_DFS[7]
PD8	SWIO	UART3_TX	GPIOD_DFS[8]
PD9	-	-	-
PD10	CAN_RX	-	-
PD11	FOUT0	ATIM_BKR	GPIOD_DFS[11]
PD12	SDA	SPI2_MOSI	GPIOD_DFS[12]
PD13	-	-	-
PD14	-	-	-
PE0	SPI0_SSN	-	-
PE1	SPI0_SCK	-	-
PE2	SPI0_MISO	-	-
PE3	SPI0_MOSI	-	-
PE4	-	-	-

GPIO	Digital Feature1 GPIOx_DFS[x]=0	Digital Feature2 GPIOx_DFS[x]=1	Additional AFSEL
PE5	LPT32_CH4	CAN_TX	GPIOE_DFS[5]
PE6	-	-	-
PE7	COMP3_OUT	-	-
PE8	-	-	-
PE9	CAN_TX	-	-

Table 41-3 Digital Peripheral Function Selection Table

41.3.4 Analog Function

When a GPIO is configured to analog function:

- Output buffer off
- Digital input function off
- Pull-up resistor off
- Software reads GPIOx_DIN and returns 0
- IO analog channel is connected to a specific analog peripheral
- If an IO is connected to multiple analog peripherals at the same time, only one can be enabled at the same time

Each GPIO has two analog channels, namely the resistance channel and the switch channel.The resistance channel cannot be disabled, any signal on the pin will be transmitted to the inside of the chip, and the switch channel can be enabled or disabled by the ANEN signal.

Most analog signals of the chip, such as COM/SEG, are connected to the resistance channel of GPIO, and the output level is disabled or enabled internally by the DISP module.Other analog signals partly go through the resistance channel and partly through the switch channel. See the table below:

Analog function	PAD channel	Pin	Description
COM	RESISTANCE		Switch inside the module
SEG	RESISTANCE		
XTHF	RESISTANCE		Switch inside the module
XTLF	RESISTANCE		
ADC_Inx	RESISTANCE		The module adopts T-switch to separate crosstalk
OPAx_INN/INP	RESISTANCE		
OPAx_OUT	SWITCH	PB0, PB1, PC4	Cannot be used with LCD display at the same time
ANATST	SWITCH	PB7,PD6	Enable switch when FCR=11 and ANEN=1
VCINx	SWITCH	PB2,PB3	Enable switch whenFCR=11 and

			ANEN=1
VDISPx	SWITCH	PB15,PC0,PC1,PD12	Enable switch whenFCR=11 and ANEN=1

Table 41-4 Analog Function Channel Selection Table

41.3.5 Using External Crystal Pins

FM33LG0 supports external 32768Hz crystal and 4~24MHz high frequency crystal.

PC2 and PC3 are GPIO by default, and can be used as XTHFIN and XTHFOUT external high frequency crystals after configured as analog function.

XT32KI and XT32KO are analog functions, which can be externally connected with 32768Hz crystal and cannot be configured for GPIO.

If you want to use an external 32K clock input, you can inject 32K clock from XT32KI.

41.4 VBAT Power Supply Pin

PH15 is the VBAT power supply pin, and its power supply is switched between VDD and VBAT. Limited by the conduction capability of the power switch, this IO cannot source a large current, and the output current should be limited to less than 1mA.

When the chip VDD is powered off, all other IOs are not powered except for PH15.

41.5 SWD Pin

The ARM SWD pins are multiplexed with PD7 (SWCLK) and PD8 (SWIO). The ARM SWD pins are multiplexed with PD7 (SWCLK) and PD8 (SWIO). These two GPIOs default to the SWD function after power-on reset, and the internal pull-up resistor is enabled by default to save external pull-ups.

41.6 WKUPx Pin

FM33LG0 has eight WKUP pins that can wake up the chip from Sleep/DeepSleep mode, even if the on-chip oscillators are all stopped.

WKUPx pin input rising edge, falling edge or rising falling edge (software configuration) can wake up the chip from Sleep mode. In order to enable this function, the corresponding pin needs to be configured for GPIO input function and the corresponding PINWKEN set.Note that the PAD has an internal pull-up resistor, which must be disabled if configured for wake-up on rising edge.

Each WKUP-enabled IO comes with an on-chip analog filter of approximately 100ns, which filters out

burrs on the input signal to avoid false triggering.

In Sleep/DeepSleep mode, any pulse greater than 100ns on the enabled WKUPx pin will trigger the chip to wake up.

The 8-channel WKUPx circuit structure is completely independent. The following figure shows the structure diagram of one channel of the WKUP functions.

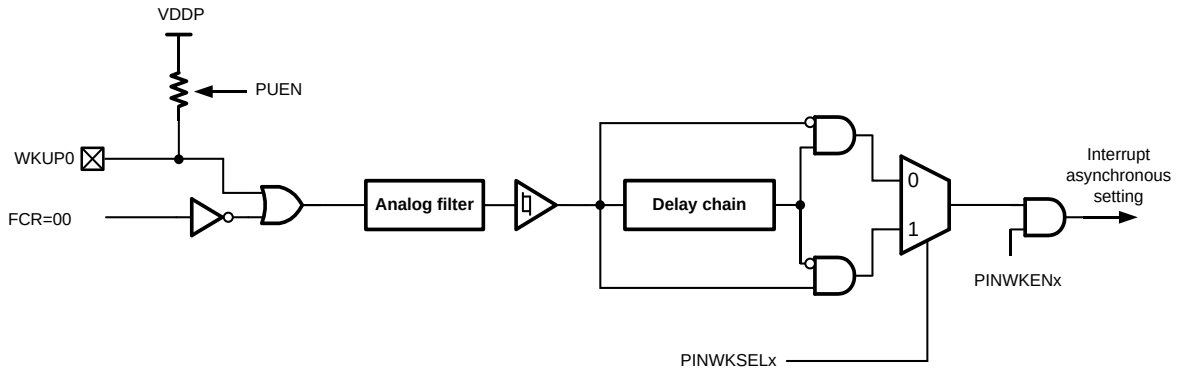


Figure 41–3 WKUPx Function Structure Diagram

The WKUPx function requires attention to the initial state of the external pin inputs when used. When enabling WKUP, a false wake-up event may result due to the initial state, which the software should take care to identify and handle.

When using the WKUP function, user must configure the FCR register of the corresponding pin to 00 (GPIO input), set the wake-up edge (PINWKSELx) and enable the PINWKENx register as required. When a wake-up event is generated on one of the WKUPx pins, the corresponding bit in the wake-up source flag register inside the PMU module will be set automatically.

41.7 External Pin Interrupts (EXIT)

41.7.1 Function Description

The 5 groups of GPIOs (A~E) of FM33LG0 can generate up to 19 EXTI interrupts, each group of GPIOs can generate 4 EXTI interrupt flags respectively, and finally all EXTI interrupts are aggregated to the #46 entry of NVIC.

The interrupt flags and pins correspond to the following table.

GPIO	EXTI Input Select	EXTI
PA0~PA3	EXTI_ASEL[1:0]	EXTI[0]
PA4~PA7	EXTI_ASEL[3:2]	EXTI[1]
PA8~PA11	EXTI_ASEL[5:4]	EXTI[2]
PA12~PA15	EXTI_ASEL[7:6]	EXTI[3]
PB0~PB3	EXTI_BSEL[1:0]	EXTI[4]

PB4~PB7	EXTI_BSEL[3:2]	EXTI[5]
PB8~PB11	EXTI_BSEL[5:4]	EXTI[6]
PB12~PB15	EXTI_BSEL[7:6]	EXTI[7]
PC0~PC3	EXTI_CSEL[1:0]	EXTI[8]
PC4~PC7	EXTI_CSEL[3:2]	EXTI[9]
PC8~PC11	EXTI_CSEL[5:4]	EXTI[10]
PC12~PC15	EXTI_CSEL[7:6]	EXTI[11]
PD0~PD3	EXTI_DSEL[1:0]	EXTI[12]
PD4~PD7	EXTI_DSEL[3:2]	EXTI[13]
PD8~PD11	EXTI_DSEL[5:4]	EXTI[14]
PD12	EXTI_DSEL[7:6]	EXTI[15]
PE0~PE3	EXTI_ESEL[1:0]	EXTI[16]
PE5~PE7	EXTI_ESEL[3:2]	EXTI[17]
PE8~PE9	EXTI_ESEL[5:4]	EXTI[18]
-	EXTI_ESEL[7:6]	EXTI[19]

The EXTISELx register is used to select an IO to access the EXTI channel, and the EXTI module can configure whether to digitally filter the input signal.

The digital filtering is implemented by the IO sampling clock to continuously sample input and get same level three times before it is considered a valid level input, as shown in the figure below.

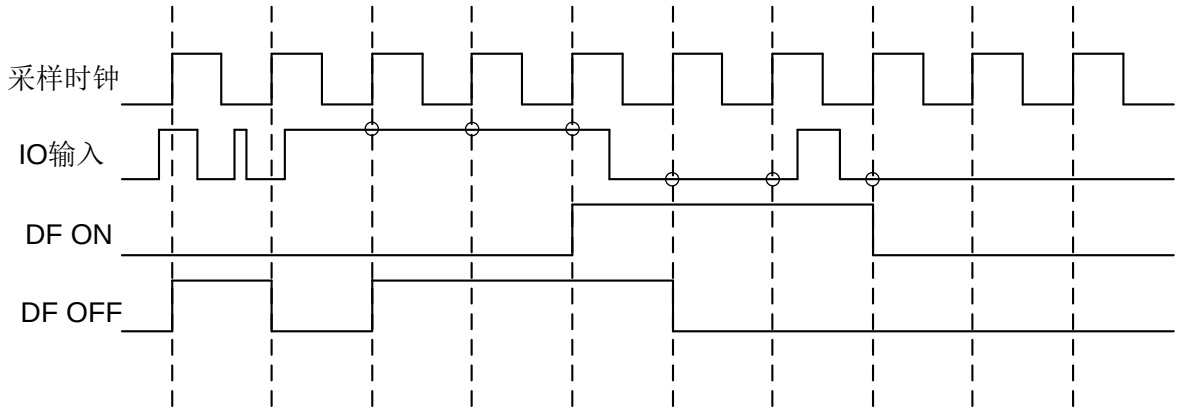


Figure 41–4 Pin Input Digital Filtering

The software can select the sampling clock for digital filtering as APBCLK or LSCLK.

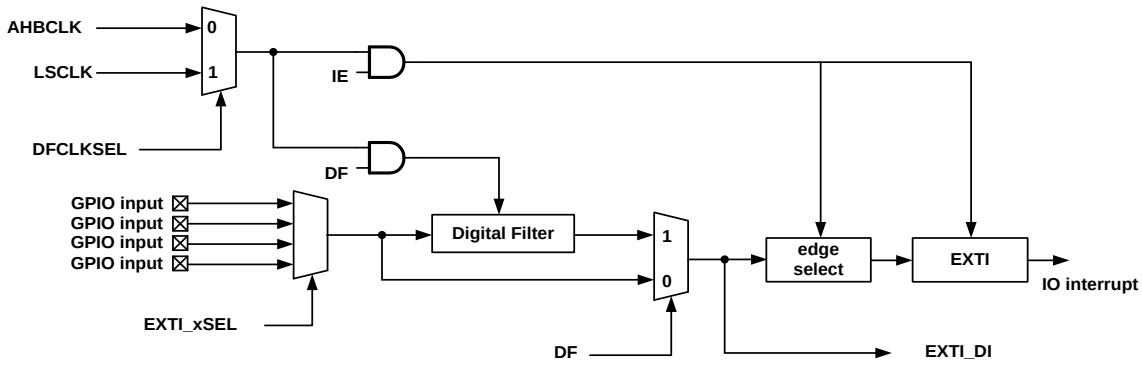


Figure 41–5 EXTI Signal Input Schematic

Users should enable or disable the digital filtering function according to the pin function needs. After enabling the digital filtering, different sampling delays will be introduced to the IO input signal depending on the AHBCLK frequency. The output signal after digital filtering can also be read by software in EXTI_DI register.

EXTI can also configure the effective edge of the input signal to support rising edge, falling edge, rising falling edge triggered interrupt, or disable EXTI interrupt triggering, as configured by the EXTI_EDS register.

41.7.2 Application Guidelines

To activate the EXTI interrupt wake-up function in Sleep/DeepSleep mode, the following steps are recommended:

- Turn off all EXTI enable
- Configure the SLP_ENEXTI bit in the SYSCLKSEL register to 1 and select LSCLK for EXTI sampling
- Turn on or off the EXTI digital filtering enable as needed
- Configure the corresponding GPIO as input
- Configure the EXTI_SEL register to select the corresponding IO
- Set OPCCR3.EXTICKE to turn on EXTI operating clock enable
- Wait at least 4 LSCLK cycles
- Configure EXTI_EDS trigger edge selection to enable the required EXTI interrupt
- Enter Sleep mode normally

All EXTIs are disabled by default after the chip is powered on, while the default pin interrupt sampling clock is the system clock APBCLK. If the user uses the system clock to generate EXTIs, the

recommended flow is as follows：

- Turn on digital filter enable (if required)
- Configure GPIO as input
- Set OPCCR3.EXTICKE to turn on EXTI operating clock enable
- Wait at least 4 APBCLK cycles
- Configure EXTI_EDS trigger edge selection to enable the required EXTI interrupt

If user wish to use a low-speed LSCLK to generate EXTI, the recommended flow is as follows:

- Configure the EXTI sample clock as LSCLK
- Turn on digital filtering enable (if required)
- Configure GPIO as input
- Set OPCCR3.EXTICKE to turn on EXTI sample clock enable
- Wait at least 4 LSCLK cycles
- Configure EXTI_EDS trigger edge to enable the required EXTI interrupt

41.8 Fast GPIO Output

The FM33LG0 can quickly change the output data of each GPIO through the set-reset function to improve the IO output efficiency, especially the efficiency and reliability of read-modify-write operation.The method is that each GPIO group output data register has two sets of set-reset mapped virtual addresses. Writing 1 to a specific bit of the set register can set the bit of the corresponding data register, and writing 1 to a specific address of the reset register can clear the bit of the corresponding data register.

41.9 Register

Offset	Name	Symbol
GPIO(Base address: 0x40000C00)		
0x00	GPIOA Input Enable Register	GPIOA_INEN
0x04	GPIOA Pull-Up Enable Register	GPIOA_PUEN
0x08	GPIOA Open-Drain Enable Register	GPIOA_ODEN
0x0C	GPIOA Function Control Register	GPIOA_FCR
0x10	GPIOA Data Output Register	GPIOA_DO
0x14	GPIOA Data Set Register	GPIOA_DSET
0x18	GPIOA Data Reset Register	GPIOA_DRST
0x1C	GPIOA Data Input Register	GPIOA_DI
0x20	GPIOA Digital Function Select	GPIOA_DFS
0x24	-	-
0x28	GPIOA Analog Channel Enable Register	GPIOA_ANEN
0x2C	GPIOA Voltage Input Low Register	GPIOA_VILR
0x40	GPIOB Input Enable Register	GPIOB_INEN
0x44	GPIOB Pull-Up Enable Register	GPIOB_PUEN
0x48	GPIOB Open-Drain Enable Register	GPIOB_ODEN
0x4C	GPIOB Function Control Register	GPIOB_FCR
0x50	GPIOB Data Output Register	GPIOB_DO
0x54	GPIOB Data Set Register	GPIOB_DSET
0x58	GPIOB Data Reset Register	GPIOB_DRST
0x5C	GPIOBData Input Register	GPIOB_DIN
0x60	GPIOB Digital Function Select	GPIOB_DFS
0x64	-	-
0x68	GPIOB Analog Channel Enable Register	GPIOB_ANEN
0x6C	GPIOB Voltage Input Low Register	GPIOB_VILR
0x80	GPIOC Input Enable Register	GPIOC_INEN
0x84	GPIOC Pull-Up Enable Register	GPIOC_PUEN
0x88	GPIOC Open-Drain Enable Register	GPIOC_ODEN
0x8C	GPIOC Function Control Register	GPIOC_FCR
0x90	GPIOC Data Output Register	GPIOC_DO
0x94	GPIOC Data Set Register	GPIOC_DSET
0x98	GPIOC Data Reset Register	GPIOC_DRST
0x9C	GPIOC Data Input Register	GPIOC_DIN
0xA0	GPIOC Digital Function Select	GPIOC_DFS
0xA4	-	-
0xA8	GPIOC Analog Channel Enable Register	GPIOC_ANEN
0xAC	GPIOC Voltage Input Low Register	GPIOC_VILR

Offset	Name	Symbol
0xC0	GPIOD Input Enable Register	GPIOD_INEN
0xC4	GPIOD Pull-Up Enable Register	GPIOD_PUEN
0xC8	GPIOD Open-Drain Enable Register	GPIOD_ODEN
0xCC	GPIOD Function Control Register	GPIOD_FCR
0xD0	GPIOD Data Output Register	GPIOD_DO
0xD4	GPIOD Data Set Register	GPIOD_DSET
0xD8	GPIOD Data Reset Register	GPIOD_DRST
0xDC	GPIOD Data Input Register	GPIOD_DIN
0xE0	GPIOD Digital Function Select	GPIOD_DFS
0xE4	-	-
0xE8	GPIOD Analog Channel Enable Register	GPIOD_ANEN
0xEC	GPIOD Voltage Input Low Register	GPIOD_VILR
0x100	GPIOE Input Enable Register	GPIOE_INEN
0x104	GPIOE Pull-Up Enable Register	GPIOE_PUEN
0x108	GPIOE Open-Drain Enable Register	GPIOE_ODEN
0x10C	GPIOE Function Control Register	GPIOE_FCR
0x110	GPIOE Data Output Register	GPIOE_DO
0x114	GPIOE Data Set Register	GPIOE_DSET
0x118	GPIOE Data Reset Register	GPIOE_DRST
0x11C	GPIOE Data Input Register	GPIOE_DIN
0x100	GPIOE Digital Function Select	GPIOE_DFS
0x104	-	-
0x108	GPIOD Analog channel Enable Register	GPIOE_ANEN
0x10C	GPIOD Voltage Input Low Register	GPIOE_VILR
0x1C0	External Interrupt Input Select Register0	GPIO_EXTISEL0
0x1C4	External Interrupt Input Select Register1	GPIO_EXTISEL1
0x1C8	External Interrupt Edge Select and Enable Register0	GPIO_EXTIEDS0
0x1CC	External Interrupt Edge Select and Enable Register1	GPIO_EXTIEDS1
0x1D0	External Interrupt Digital Filter Register	GPIO_EXTIDF
0x1D4	External Interrupt and Status Register	GPIO_EXTIISR
0x1D8	External Interrupt Data Input Register	GPIO_EXTIDI
0x200	Frequency Output Select Register	GPIO_FOUTSEL
0x300	Wakeup Enable Register	GPIO_PINWKEN

41.9.1 GPIOx Input Enable Register (GPIOx_INEN)

NAME	GPIOx_INEN(x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x00 + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	INEN15	INEN14	INEN13	INEN12	INEN11	INEN10	INEN9	INEN8
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	INEN7	INEN6	INEN5	INEN4	INEN3	INEN2	INEN1	INEN0
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	INEN	GPIO Input Enable Control 0: Input disable 1: Input enable

41.9.2 GPIOxPull-up Enable Register (GPIOx_PUEN)

NAME	GPIOx_PUEN(x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x04 + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

name	PUEN15	PUEN14	PUEN13	PUEN12	PUEN11	PUEN10	PUEN9	PUEN8
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PUEN7	PUEN6	PUEN5	PUEN4	PUEN3	PUEN2	PUEN1	PUEN0
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	PUEN	GPIO Pull-up Control 0: Pull-up disable 1: Pull-up enable

41.9.3 GPIOxOpen-drain Enable Register (GPIOx_ODEN)

NAME	GPIOx_ODEN(x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x08 + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ODEN15	ODEN14	ODEN13	ODEN12	ODEN11	ODEN10	ODEN9	ODEN8
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ODEN7	ODEN6	ODEN5	ODEN4	ODEN3	ODEN2	ODEN1	ODEN0
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	ODEN	GPIO Open-drain Output Enable 0: Open-drain output disable 1: Open-drain output enable

41.9.4 GPIOxFunction Control Register (GPIOx_FCR)

NAME	GPIOx_FCR(x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x0C + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	Px15FCR		Px14FCR		Px13FCR		Px12FCR	
access	R/W-0		R/W-0		R/W-0		R/W-0	
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	Px11FCR		Px10FCR		Px9FCR		Px8FCR	
access	R/W-0		R/W-0		R/W-0		R/W-0	
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	Px7FCR		Px6FCR		Px5FCR		Px4FCR	
access	R/W-0		R/W-0		R/W-0		R/W-0	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	Px3FCR		Px2FCR		Px1FCR		Px0FCR	
access	R/W-0		R/W-0		R/W-0		R/W-0	

bit	name	functional description
31:30	Px15FCR	Px[15] Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
29:28	Px14FCR	Px[14]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
27:26	Px13FCR	Px[13]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
25:24	Px12FCR	Px[12]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function

bit	name	functional description
		11: Analog function
23:22	Px11FCR	Px[11]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
21:20	Px10FCR	Px[10]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
19:18	Px9FCR	Px[9]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
17:16	Px8FCR	Px[8]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
15:14	Px7FCR	Px[7]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
13:12	Px6FCR	Px[6]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
11:10	Px5FCR	Px[5]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
9:8	Px4FCR	Px[4]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
7:6	Px3FCR	Px[3]Pin Function Select 00: GPIO input

bit	name	functional description
		01: GPIO output 10: Digital function 11: Analog function
5:4	Px2FCR	Px[2]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
3:2	Px1FCR	Px[1]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function
1:0	Px0FCR	Px[0]Pin Function Select 00: GPIO input 01: GPIO output 10: Digital function 11: Analog function

41.9.5 GPIOxData Output Register (GPIOx_DO)

NAME	GPIOx_DO(x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x10 + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	DO	GPIO Output Data Register

41.9.6 GPIOxData Set Register (GPIOx_DSET)

NAME	GPIOx_DSET(x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x14 + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	SET15	SET14	SET13	SET12	SET11	SET10	SET9	SET8
access	W	W	W	W	W	W	W	W
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
access	W	W	W	W	W	W	W	W

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	SET	GPIO Output Data Set Register Example: Write 0x0000_8000 to PADSET, then PADO[15] is set and the rest of the bits remain unchanged.

41.9.7 GPIOxData Reset Register (GPIOx_DRST)

NAME	GPIOx_DRST(x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x18 + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	RESET15	RESET14	RESET13	RESET12	RESET11	RESET10	RESET9	RESET8
access	W	W	W	W	W	W	W	W
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	RESET7	RESET6	RESET5	RESET4	RESET3	RESET2	RESET1	RESET0
access	W	W	W	W	W	W	W	W

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	RESET	GPIO Output Data Reset Register Example: Write 0x0000_8000 to PADRST, then PADO[15] is cleared to zero and the rest of the bits remain unchanged.

41.9.8 GPIOxData Input Register (GPIOx_DIN)

NAME	GPIOx_DIN(x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x1C + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DIN15	DIN14	DIN13	DIN12	DIN11	DIN10	DIN9	DIN8
access	R	R	R	R	R	R	R	R
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
access	R	R	R	R	R	R	R	R

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	DIN	Portx Input Data Register

bit	name	functional description
		This register only occupies address space and has no physical implementation. Software reads this register to return the pin input signal directly, the chip does not latch the pin input.

41.9.9 GPIOxDigital Function Select (GPIOx_DFS)

NAME	GPIOx_DFS(x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x20 + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DFS[15:8]							
access	R/W-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	DFS[7:0]							
access	R/W-0							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	DFS	Portx Digital Function Select For pins with multiple digital peripheral functions, the PxDFS register allows you to select which peripheral function to use. Note that the valid register locations are different for different IO groupings, please refer to Table 41-3 for detailed definitions.

41.9.10 GPIOxAnalog Channel Enable Register (GPIOx_ANEN)

NAME	GPIOx_ANEN(x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x28 + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	ANEN[15:8]							
access	R/W-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	ANEN[7:0]							
access	R/W-0							

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	ANEN	PortX Analog Channel Enable 1: IO analog channel enable 0: IO analog channel disable ***Note:*** *IO supporting analog channels include PB0, PB1,PB2, PB3, PB15, PC0, PC1, PC4,PD6,PD12* *The ANEN registers corresponding to the above IOs is valid; the other registers are invalid.*

41.9.11 GPIOx Voltage Input Low Register (GPIOx_VILR)

NAME	GPIOx_VILR (x=A,B,C,D,E)							
Offset	PA, y=0 PB, y=1 PC, y=2 PD, y=3 PE, y=4 0x2C + y*0x40							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							

bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	VIL15	VIL14	VIL13	VIL12	VIL11	VIL10	VIL9	VIL8
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	VIL7	VIL6	VIL5	VIL4	VIL3	VIL2	VIL1	VIL0
access	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	name	functional description
31:16	--	RFU: **Reserved, read as 0**
15:0	VIL	GPIO Voltage Input Low Control 0: Voltage input low is normal 1: Voltage input low is reduced

41.9.12 External Interrupt Input Select Register0 (GPIO_EXTISEL0)

NAME	GPIO_EXTISEL0							
Offset	0x1C0							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	EXTI_DSEL							
access	R/W-00 0000							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	EXTI_CSEL							
access	R/W-00 0000							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	EXTI_BSEL							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	EXTI_ASEL							
access	R/W-0000 0000							

bit	name	functional description
31:24	EXTI_DSEL[5:0]	PortD External Interrupt Input Select EXTI_DSEL[7:6] – 00: PD12 01: RFU 10: RFU 11: RFU EXTI_DSEL[5:4] – 00: PD8

bit	name	functional description
		01: PD9 10: PD10 11: PD11 EXTI_DSEL[3:2] – 00: PD4 01: PD5 10: PD6 11: PD7 EXTI_DSEL[1:0] – 00: PD0 01: PD1 10: PD2 11: PD3
23:16	EXTI_CSEL[5:0]	PortC External Interrupt Input Select EXTI_CSEL[5:4] – 00: PC8 01: PC9 10: PC10 11: PC11 EXTI_CSEL[3:2] – 00: PC4 01: PC5 10: PC6 11: PC7 EXTI_CSEL[1:0] – 00: PC0 01: PC1 10: PC2 11: PC3
15:8	EXTI_BSEL[7:0]	PortB External Interrupt Input Select EXTI_BSEL[7:6] – 00: PB12 01: PB13 10: PB14 11: PB15 EXTI_BSEL[5:4] – 00: PB8 01: PB9 10: PB10 11: PB11 EXTI_BSEL[3:2] –

bit	name	functional description
		00: PB4 01: PB5 10: PB6 11: PB7 EXTI_BSEL[1:0] – 00: PB0 01: PB1 10: PB2 11: PB3
7:0	EXTI_ASEL	PortA External Interrupt Input Select EXTI_ASEL[7:6] – 00: PA12 01: PA13 10: PA14 11: PA15 EXTI_ASEL[5:4] – 00: PA8 01: PA9 10: PA10 11: PA11 EXTI_ASEL[3:2] – 00: PA4 01: PA5 10: PA6 11: PA7 EXTI_ASEL[1:0] – 00: PA0 01: PA1 10: PA2 11: PA3

41.9.13 External Interrupt Input Select Register1 (GPIO_EXTISEL1)

NAME	GPIO_EXTISEL1							
Offset	0x1C4							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8

name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	EXTI_ESEL							
access	R/W-0000 0000							

bit	name	functional description
31:24	--	RFU: **Reserved, read as 0**
7:0	EXTI_ESEL	PortE External Interrupt Input Select EXTI_ESEL[7:6] – 00: RFU 01: RFU 10: RFU 11: RFU EXTI_ESEL[5:4] – 00: PE8 01: PE9 10: RFU 11: RFU EXTI_ESEL[3:2] – 00: RFU 01: PE5 10: PE6 11: PE7 EXTI_ESEL[1:0] – 00: PE0 01: PE1 10: PE2 11: PE3

41.9.14 External Interrupt Edge Select and Enable Register0 (GPIO_EXTIEDS0)

NAME	GPIO_EXTIEDS0							
Offset	0x1C8							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	EXTI15_EDS		EXTI14_EDS		EXTI13_EDS		EXTI12_EDS	
access	R/W-11		R/W-11		R/W-11		R/W-11	
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	EXTI11_EDS		EXTI10_EDS		EXTI9_EDS		EXTI8_EDS	
access	R/W-11		R/W-11		R/W-11		R/W-11	

bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	EXTI7_EDS		EXTI6_EDS		EXTI5_EDS		EXTI4_EDS	
access	R/W-11		R/W-11		R/W-11		R/W-11	
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	EXTI3_EDS		EXTI2_EDS		EXTI1_EDS		EXTI0_EDS	
access	R/W-11		R/W-11		R/W-11		R/W-11	

bit	name	functional description
31:30	EXTI15_EDS	EXTI[15] Edge Select 00: Rising 01: Falling 10: Both 11: Disable
…	…	
1:0	EXTI0_EDS	EXTI1[0] Edge Select 00: Rising 01: Falling 10: Both 11: Disable

41.9.15 External Interrupt Edge Select and Enable Register1 (GPIO_EXTIEDS1)

NAME	GPIO_EXTIEDS1							
Offset	0x1CC							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-		EXTI18_EDS		EXTI17_EDS		EXTI16_EDS	
access	U-0		R/W-11		R/W-11		R/W-11	

bit	name	functional description
31:6	--	RFU: **Reserved, read as 0**
5:4	EXTI18_EDS	EXTI[18] Edge Select 00: Rising 01: Falling

bit	name	functional description
		10: Both 11: Disable
3:2	EXTI17_EDS	EXTI[17] Edge Select 00: Rising 01: Falling 10: Both 11: Disable
1:0	EXTI16_EDS	EXTI1[16] Edge Select 00: Rising 01: Falling 10: Both 11: Disable

41.9.16 External Interrupt Digital Filter Register (GPIO_EXTIDF)

NAME	GPIO_EXTIDF							
Offset	0x1D0							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-					EXTI_DF[18:16]		
access	U-0					R/W -000		
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	EXTI_DF[15:8]							
access	R/W -0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	EXTI_DF[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:19	--	RFU: **Reserved, read as 0**
18:0	EXTI_DF	EXTI Input Digital Filter Function Enable 0: Disable EXTI digital filtering 1: Enable EXTI digital filtering

41.9.17 External Interrupt and Status Register (GPIO_EXTIISR)

NAME	GPIO_EXTIISR							
Offset	0x1D4							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							

access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-					EXTI[18:16]		
access	U-0					R/W-000		
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	EXTI[15:8]							
access	R/W-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	EXTI[7:0]							
access	R/W-0000 0000							

bit	name	functional description
31:19	--	RFU: **Reserved, read as 0**
18:0	EXTI	External Interrupt and Status Register, a total of 19 pin interrupts can be generated. Hardware set, write 1 to clear by software.

41.9.18 External Interrupt Data Input Register (GPIO_EXTIDI)

NAME	GPIO_EXTIDI							
Offset	0x1D8							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-					EXTI_DI[18:16]		
access	U-0					R-000		
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	EXTI_DI[15:8]							
access	R-0000 0000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	EXTI_DI[7:0]							
access	R-0000 0000							

bit	name	functional description
31:19	--	RFU: **Reserved, read as 0**
18:0	EXTI_DI	EXTI Input Signal Read-only Register, software can read this register to observe the current status of EXTI's 19 input signals. ***Note:*** *When digital filtering is enabled, software can read the filtered status of an IO input signal from this register.*

41.9.19 Frequency Output Select Register (GPIO_FOUTSEL)

NAME	GPIO_FOUTSEL							
Offset	0x200							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	FOUT1SEL				FOUT0SEL			
access	R/W-0000				R/W-0000			

bit	name	functional description
31:8	--	RFU: **Reserved, read as 0**
7:4	FOUT1SEL	PB12 Output Select 0000: XTLF 0001: RCLP 0010: ADCCLK 0011: LSCLK 0100: EOC 0101: RTCTM 0110: PLLO/64 0111: EOCAL 1000: APBCLK/64 1001: ROSC_TDLV 1010: RCLF 1011: RCHF 1100: XTHF/64 1101: ADCCLK/64 1110: CLK8K 1111: ROSC_TDHV
3:0	FOUT0SEL	PD11 Output Select 0000: XTLF 0001: RCLP 0010: RCHF/64 0011: LSCLK 0100: AHBCLK/64 0101: RTCTM 0110: PLLO/64

bit	name	functional description
		0111: RTCCLK64Hz 1000: APBCLK/64 1001: PLLO 1010: RCLF 1011: RCHF 1100: XTHF/64 1101: COMP1O 1110: CLK8K 1111: ADC_CLK

41.9.20 Wakeup Enable Register (GPIO_PINWKEN)

NAME	GPIO_PINWKEN							
Offset	0x300							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	WKSEL	-						
access	R/W-0	U-0						
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	PINWKSEL[7:0]							
access	R/W-00000000							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	PINWKEN[7:0]							
access	R/W-00000000							

bit	name	functional description
31	WKSEL	WKUPx Interrupt Entry Select 0: NMI interrupt 1: #38 interrupt ***Note:*** *If you need to run the main program directly without entering the ISR after waking up, you can set WKSEL to 1, and set the PRIMASK register to enter sleep. When the WKUPx event arrives, the chip will exit the sleep mode, but will not enter the interrupt service program because PRIMASK=1.*
30	--	RFU: **Reserved, read as 0**
29:10	PINWKSEL	WKUP Edge Select 00: Corresponding WKUP pin for falling edge wake-up 01: Corresponding WKUP pin for rising edge wake-up 10/11: Corresponding WKUP pin for rising and falling edge

bit	name	functional description
		wake-up Corresponding bit of registers Bit[11:10] corresponds to WKUP0 Bit[13:12] corresponds toWKUP1 …… Bit[27:26] corresponds toWKUP8 Bit[29:28] corresponds toWKUP9
9:0	PINWKEN	WKUP Pin Enable Signal 1: The corresponding WKUP pin function is valid 0: The corresponding WKUP pin function is invalid PINWKEN[x] controls enable of WKUPx pin

42 Serial Wire Debug (SWD)

42.1 Introduction

FM33LG0 chip can use the dedicated programmer provided by Fudan Microelectronics or download the user program through Bootloader. The programmer communicates with the chip through the serial wire debug (SWD) to complete the program download and perform Checksum verification of the full space contents of the Flash.

42.2 Programmer Instruction

For the instruction of the programmer, please refer to the application manual, or contact Fudan Microelectronics.

43 Debug Support

43.1 Introduction

FM33LG0 chip is based on the ARM Cortex-M0 processor and supports the corresponding debug features. Through hardware breakpoints and data watchpoints, the debugger can stop the CPU core operation during specific instruction fetches and data accesses, inspect the core registers and system peripheral status, and resume the core operation as needed.

The simulation debugging master is interconnected with the FM33LG0 chip through the SWD interface, and realizes simulation debugging.

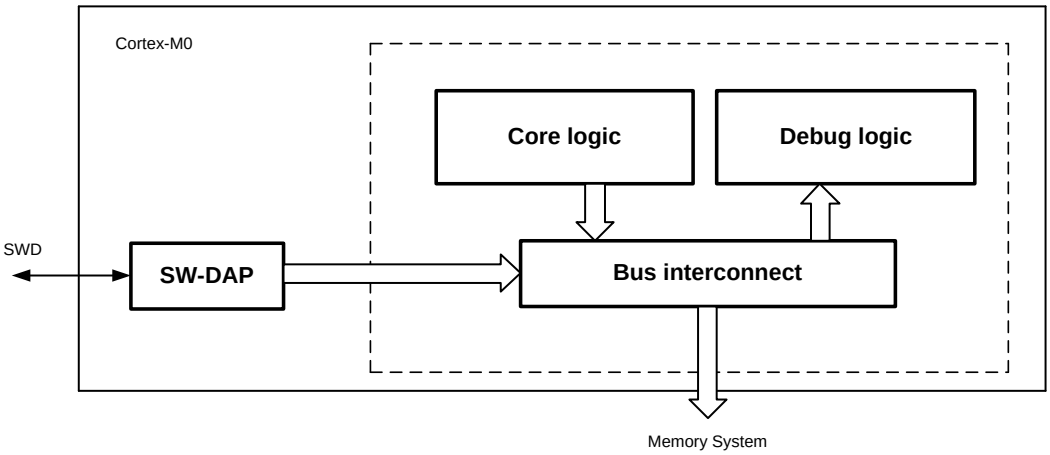


Figure 43–1 Cortex-M0 Debug System Diagram

For debug features of the Cortex-M0 core, please refer to the Cortex-M0 Technical Reference Manual from ARM.

43.2 Debug Pin

43.2.1 SWD Pin

The SWD pins of FM33LG0 series MCU are as follows:

SWD pins	Debug function	Pin definition
SWDIO	SWD data input/output	PD8
SWCLK	SWD clock input	PD7

Note: *Both PD7 and PD8 pins aredefault to be input status after chip reset, unlike most GPIOs.*

43.2.2 Pull-up Resistance

After chip reset, SWDIO, SWCLK pins enable the internal pull-up (~ 100K ohm) by default, note the pull-up resistor state to prevent floating of the input pin and increase the leakage.

43.3 SWD Port Protocol

43.3.1 Protocol Introduction

SWD protocol uses LSB-first for data sending and receiving. Through the SWD interface, the debug master can read and write DPACC and APACC register groups.

SWIO needs to insert turn-around time on the bus each time the data direction is switched, and neither the master nor the slave will drive SWIO during this time. Between two transmissions, the master must drive the line low to enter the idle status, or continue to send the start bit of a new transmission to continue transmission. After a packet transmission, the master can also be idle to keep the line high or be pulled up by a pull-up resistor. The SWD protocol does not have an explicit reset signal, and the master or target will detect a reset when it does not see the expected signal. By holding the line high for 50 clock cycles followed by a request to read the ID, a successful resynchronization can be ensured after an error or reset is detected.

43.3.2 Transfer Sequence

Each SWD communication transmission sequence consists of three parts.

1. Packet request (8bits), sent by the master

2. ACK response (3bits), sent back by the slave

3. Data transfer phase (33bits), sent by the master or slave

where the packet request byte is defined as follows:

Bit	Name	Description
0	Start	Start bit, must be 1
1	ApnDP	AP/DP select 0: DP access 1: AP access
2	RnW	Read/Write select 0: Write request 1: Read request
4:3	A[3:2]	Address field of DP/AP register
5	Parity	Check bits for Bit0~Bit4 data
6	Stop	0
7	Park	Master is not driven, it is pulled up through the bus, and the slave is read as 1

After the packet request is sent, there is always a 1bit turn-around time on the bus.

The ACK response is defined as follows:

Bit	Name	Description
0:2	ACK	001: FAULT 010: WAIT 100: OK

If the master initiates a read operation, or if the ACK is WAIT or FAULT, a turn-around time must be inserted after the ACK.

The data transfer format is as follows:

Bit	Name	Description
0:31	Data	Data read or written
32	Parity	Parity for 32bit data

43.3.3 SW-DP ID Code

The SW-DP of Cortex-M0 has a fixed ID code: 0x0BB11477

The SW-DP is inactive until the host reads the ID code.

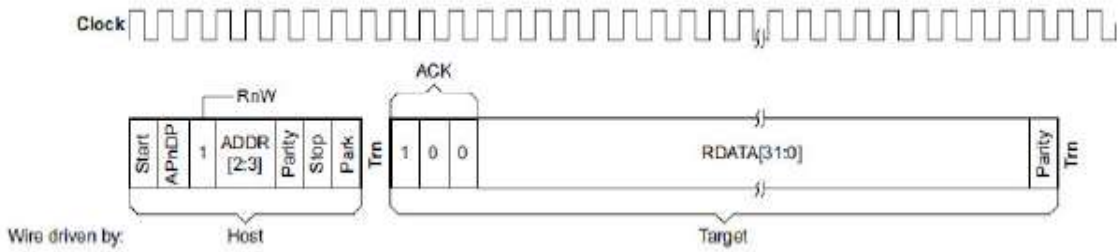
- SW-DP is in RESET status after chip reset, or after SWIO is pulled high for 50 SWCLK cycles
- After pulling SWIO low for at least 2 SWCLK cycles, SW-DP enters IDLE status
- When the SW-DP is in RESET, the master must first bring it into IDLE and then perform a read operation on the ID code register to activate the SW-DP. Otherwise the slave will respond with a FAULT response to the master's communication.

43.3.4 Master Read

A successful read operation consists of the following three phases:

- An 8-bit read packet request from the master to the target.
- A 3-bit answer (ack) from the target to the host. A successful OK response is 100, a WAIT response is 010, and a FAULT response is 001.
- A 33-bit data read phase (payload) from the master to the target.

By default, there is a clocked turnaround period between the first and second phases and after the third phase, and a successful read operation is shown below.

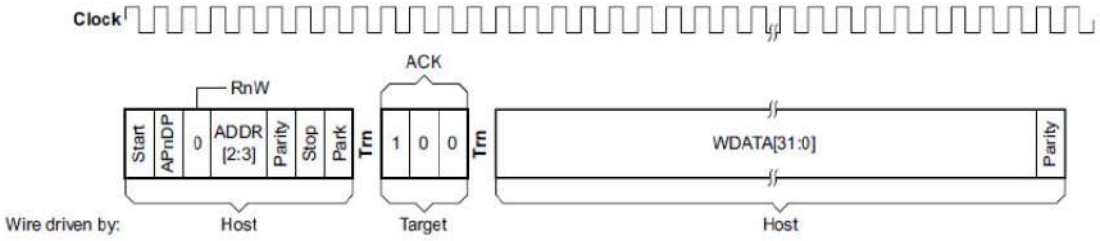


43.3.5 Master Write

A write operation consists of the following three phases

- An 8-bit write packet request (header) from the master to the target.
- A 3-bit answer (ack) from the target to the master. OK ack is 100 and FAULT ack is 001.
- A 33-bit data write phase (payload) from the master to the target.

By default, there is a clocked turnaround period between each two phases, and a successful write operation is shown below.



43.4 SWD-DP Register

43.4.1 Register List

Address (A[3:2])	DPBANKSEL	Name	Access
00	x	DHCSR	RO
		ABORT	WO
01	0x0	CTRL/STAT	RW
	0x1	DLCR	RW
10	x	RESEND	RO
		SELECT	WO
11	x	RDBUFF	RO

For detailed description of the registers, please refer to the Cortex-M0 Technical Reference Manual.

43.5 Core Debug Register

Core debug can be implemented by operating the core debug registers. The master accesses the following core debug registers via SW-DP.

Address	Name	Type	Function
0xE000EDF0	DHCSR	RW	Debug Halting Control and Status Register
0xE000EDF4	DCRSR	WO	Debug Core Register Selector Register
0xE000EDF8	DCRDR	RW	Debug Core Register Data Register
0xE000EDFC	DEMCR	RW	Debug Exception and Monitor Control Register
0xE000EE00 to 0xE000EEFF	-	-	Reserved for Debug Extension

The above debug registers are not affected by system reset and are only affected by power-on reset. Immediate halt after CPU reset can be achieved by:

- Setting bit0 of DEMCR register (VC_CORRESET)
- Displacing bit0 of DHCSR register (C_DEBUGEN)
- Performing a system reset

43.6 Low-Power Debug Support

Generally, when the chip enters Sleep/DeepSleep mode, the FCLK and HCLK of the CPU will be turned off, which will cause the host to be unable to maintain the debugger connection with the chip.In order to support debugging in low power mode, when CPU is connected to the debugger, FCLK and HCLK must keep running, that is, the chip will not really go to sleep at this time. This should be paid attention to during debugging.

43.7 Debug-Related Configuration

By configuring the DBG_CR register, you can set whether the chip's internal timer and watchdog circuit continue to work in the debug status. For details, please refer to MCU DEBUG Config Register (DBG_CR).

43.8 Register

Address	Name	Symbol
DBG(Base address: 0x40000000)		
0x00	System Config Register	SYSCFG
0x04	MCU Debug Config Register	DBG_CR
0x08	HardFault Query Register	HDFR

43.8.1 System Config Register (SYSCFG)

*SYSCFG is located in the AO domain and is read by NVMIF and then written through the local bus; the AHB/APB bus cannot be written but can only be read.

NAME	SYSCFG							
Offset	0x40000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	LDT0FAIL	LDT1FAIL	DCTFAIL	-				
access	R-0	R-0	R-0	U-0				
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	RED_INFO							
access	R-xx							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

name	-	RAMCFG	FLSCFG	MODE
access	U-0	R-x	R-xx	R-00

bit	name	functional description
31	LDT0FAIL	Flash config information, namely LDT0 area, check error flag 1: A certain configuration information verification error occurs, the error item will remain the default value 0: Checked correctly
30	LDT1FAIL	OPTBYTE/ACLOCK, namely LDT1 area, check error flag 1: Verification fail, debug interface and ACLOCK protection related registers are set to 1 0: Verification passed
29	DCTFAIL	DCT page trim-tag check error flag 1: Verification failed, not 5A5A5A5A 0: Verification passed
28:16	--	RFU: **Reserved, read as 0**
15:8	RED_INFO	Flash redundancy information, redundancy page minimum 8bit
7:5	--	RFU: **Reserved, read as 0**
4	RAMCFG	RAM Version Config 0: 32KB 1: 16KB
3:2	FLSCFG	Flash Size Config 00/11: 256KB 01: 128KB 10: 64KB
1:0	MODE	Chip Permission Mode 00/10: User 01: Manufacture 11: Original

43.8.2 MCU Debug Config Register (DBGCR)

FM33LG0 expands the DBG_CR register to config the watchdog and timer in the Debug status. The DBG_CR register can be rewritten by the SWD interface or software.

NAME	DBG_CR							
Offset	0x04							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							DBG_SL

								EEP
access	U-0							R/W-0
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	DBG_LPT16_STOP	DBG_BT16_STOP	DBG_AT_STOP	DBG_LPT32_STOP	DBG_GT2_STOP	DBG_GT1_STOP	DBG_GT0_STOP	DBG_BT32_STOP
access	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-						DBG_WWDT_STOP	DBG_IWDT_STOP
access	U-0						R/W-1	R/W-1

bit	name	functional description
31:17	--	RFU: **Reserved, read as 0**
16	DBG_SLEEP	Debug Config Under Sleep Mode 0: Enter sleep normally, turn off HCLK and FCLK, the debugger cannot keep the connection 1: Without closing HCLK and FCLK, the debugger can keep the connection in sleep mode
15	DBG_LPT16_STOP	LPTIM16 enable control bit in Debug status 1: Turn off LPTIM16 during Debug 0: Keep the original state of LPTIM16 during Debug
14	DBG_BT16_STOP	BSTIM16 enable control bit in Debug status 1: Turn off BSTIM16 during Debug 0: Keep the original status of BSTIM16 during Debug
13	DBG_AT_STOP	ATIM enable control bit in Debug status 1: Turn off ATIM during Debug 0: Keep the original status of ATIM during Debug
12	DBG_LPT32_STOP	LPTIM32 enable control bit in Debug status 1: Turn off LPTIM32 during Debug 0: Keep the original status of LPTIM32 during Debug
11	DBG_GT2_STOP	GPTIM2 enable control bit in Debug status 1: Turn off GPTIM2 during Debug 0: Keep the original status of GPTIM2 during Debug
10	DBG_GT1_STOP	GPTIM1 enable control bit in Debug status 1: Turn off GPTIM1 during Debug 0: Keep the original status of GPTIM1 during Debug
9	DBG_GT0_STOP	GPTIM0 enable control bit in Debug status 1: Turn off GPTIM0 during Debug 0: Keep the original status of GPTIM0 during Debug
8	DBG_BT32_STOP	BSTIM32 enable control bit in Debug status 1: Turn off BSTIM32 during Debug 0: Keep the original status of BSTIM32 during Debug

7:2	--	RFU: **Reserved, read as 0**
1	DBG_WWDT_STOP	WWDT enable control bit in Debug status 1: Turn off WWDT during Debug 0: Keep the original status of WWDT during Debug
0	DBG_IWDT_STOP	IWDT enable control bit in Debug status 1: Turn off IWDT during Debug 0: Keep the original status of IWDT during Debug

43.8.3 HardFault Query Register (HDFR)

NAME	**HDFR**							
Offset	0x08							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-	DABORT_ADDR_FLAG	DABORT_RESP_FLAG	SVCUNDEF_FLAG	BKPT_FLAG	TBIT_FLAG	SPECIAL_OP_FLAG	HDF_REQUEST_FLAG
access	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit	**name**	**functional description**
31:7	--	RFU: **Reserved, read as 0**
6	DABORT_ADDR_FLAG	Address unaligned access error flag, write 1 to clear 1: Address misaligned access error 0: Address unaligned access is not performed
5	DABORT_RESP_FLAG	Illegal address access error flag, write 1 to clear 1: An illegal address was accessed during bus transmission, causing HRESP to be high and an error occurred 0: Illegal address was not accessed
4	SVCUNDEF_FLAG	SVC instructions undefined flag, write 1 to clear If the SVCall priority is lower than the currently activelevel, or if HardFault or NMI is active, or PRIMASK is set, the core should treat SVC instructions as though theywere UNDEFINED.

bit	name	functional description
3	BKPT_FLAG	Execute BKPT instruction flag, write 1 to clear 1: BKPT instruction executed 0: BKPT instruction not executed
2	TBIT_FLAG	Thumb-State flag, write 1 to clear 1: Switch to ARM status 0: In Thumb-State
1	SPECIAL_OP_FLAG	Special instruction flag, write 1 to clear 1: A special instruction code is executed, such as an attempt to fetch an instruction in the XN area 0: No special instruction code is executed
0	HDF_REQUEST_FLAG	HardFault flag bit, any type of HardFault will cause this bit to be set, write 1 to clear it 1: HardFault request 0: No HardFault request

44 Device Signature

Each FM33LG0 series MCU has its own device signature, including memory capacity information and unique device ID number.

44.1 Memory Capacity Query

The device Flash and RAM capacity information can be obtained by querying the SYSCON register.

NAME	SYSCFG							
Offset	0x40000000							
bit	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
name	-							
access	U-0							
bit	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
name	-							
access	U-0							
bit	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
name	-							
access	U-0							
bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
name	-			RAMCFG	FLSCFG		MODE	
access	U-0			R-x	R-xx		R-00	

bit	name	functional description
31:5	--	RFU: **Reserved, read as 0**
4	RAMCFG	RAM Version Config 0: 32KB 1: 16KB
3:2	FLSCFG	Flash Size Config 00/11: 256KB 01: 128KB 10: 64KB
1:0	--	RFU: **Reserved, read as 0**

44.2 Device UID

The device UID of each MCU of the FM33LG0 series is unique in the world, written by the original factory, and cannot be rewritten after leaving the factory.

The UID has a total of 128 bits and is stored in a special flash sector. This UID can be read when the software is running and used to implement code protection or secure boot applications.

The UID access address is 12 bytes starting from 0x1FFF_FA10. For details, please contact Shanghai Fudan Microelectronics.

Revision History

NUMBER	DATE	PAGE	CHAPTER & DIAGRAM	DETAILS
1.0	2020-6	825		Initial release
1.1	2020-9	833		Updated electrical parameters
1.2	2020-9	831		Updated Pin Number
1.3	2020-12	836		Modified description error
1.4	2021-3	838		Added the parameters of XTHF and OPA
1.5	2021-3	838	2.4	Added package thermal resistance characteristics
1.6	2021-5	841	26	Improved CAN Controller chapter

Contact Us

Shanghai Fudan Microelectronics Group Co., Ltd.
Address: Building 4, 127 Guotai Road, Shanghai, China
Postcode: 200433
Tel: (86-021) 6565 5050
Fax: (86-021) 6565 9115

Shanghai Fudan Microelectronics(HK)Co.,Ltd.
Address: Unit 506,5/f.,East Ocean Centre, 98 Granville Road,Tsim Sha Tsui East, Kowloon, Hong Kong
Tel: (852) 2116 3288/2116 3338
Fax: (852) 2116 0882

Beijing Fudan Microelectronics Technology Co.,Ltd.
Address:Gehua Buiding.B.423, North Street, Dongcheng District,Beijing City,China
Postcode: 100007
Tel: (86-10) 8418 6608 / 8418 7486
Fax: (86-10) 8418 6211

Shenzhen Fudan Microelectronics Co., Ltd.
Address:Room.1303,Century Bldg,Shengtingyuan Hotel, Huaqiang Rd.(North),Shenzhen,China
Postcode: 518028
Tel: (86-0755) 8335 0911 / 8335 1011 / 8335 2011 / 8335 0611
Fax: (86-0755) 8335 9011

Shanghai Fudan Microelectronics (HK) Ltd Taiwan Representative Office
Address: Unit 1225,12F,No.252,Sec.1 Neihu Rd.,Neihu Dist.,Taipei City 114,Taiwan
Tel: (886-2) 7721 1889 / 7721 1890
Fax: (886-2) 7722 3888

Shanghai Fudan Microelectronics (HK) LtdSingapore Branch Office
Address: 47 Kallang Pudding Road, #08-06, The Crescent @ Kallang, Singapore 349318
Tel: (65) 6443 0860
Fax: (65) 64431215

Fudan Microelectronics (USA) Inc.
Address: 97 EBrokaw Road, Suite 320, San Jose, CA 95112 USA
Tel: (+1) 408 335 6936

Website: http://www.fmsh.com/

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 32-bit Microcontrollers - MCU ***category:***

Click to view products by Fudan ***manufacturer:***

Other Similar products are found below :

MCF51AC256AVFUE MCF51AC256BCFUE MCF51AC256BVFUE MB91F464AAPMC-GSE2 R5S726B0D216FP#V0 MB91F248PFV-GE1 MB91243PFV-GS-136E1 SAK-TC1782F-320F180HR BA TC364DP64F300WAAKXUMA1 R5F566NNDDFP#30 R5F566NNDDFC#30 R5F566NNDDBD#20 MC96F8216ADBN A96G181HDN A96G140KNN A96G174FDN A31G213CL2N A96G148KNN A96G174AEN AC33M3064TLBN-01 V3s T3 A40i-H V526 A83T R11 V851s A133 V833 F1C100S T3L T507 A33 A63 T113-i H616 V853 V533 R16-J V536-H A64-H V831 V3LP T113-S3 F1C200S F133-A R128-S2 D1-H ADUCM360BCPZ128-TR APT32S003F8PT