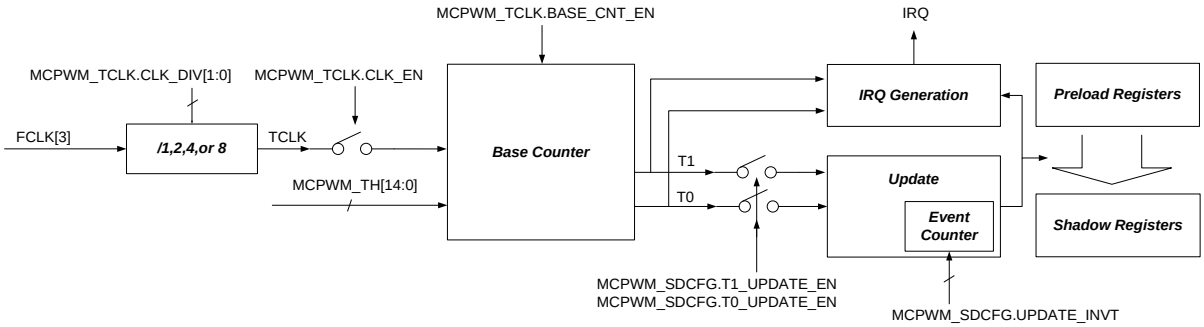
Linko Semiconductor Co., Ltd.

LKS32MC05x User Manual

Contents

List of Tables

List of Figures



1 Document Convention

1.1 Register Read/Write Permissions

RW Read/write, available for software read and write.

RO Read-only, software can only read.

WO Write-only, software can only write. The default value will be returned when reading this bit.

RW1C Read and Write 1 to Clear

1.2 Abbreviations

Word: 32-bit data/instruction.

Halfword: 16-bit data/instruction.

Byte: 8-bit data.

Double word: 64-bit data.

ADC: Analog-Digital Converter

DAC: Digital-Analog Converter

BGP: Bandgap, Bandgap voltage reference

WDT: Watch dog

LSI:Low Speed Internal Clock, the 64KHz RC oscillator

HSI: High Speed Internal Clock, the 4MHz RC oscillator

HSE: High Speed External Clock, the 4~8MHz external crystal clock

PLL: Phase Lock Loop Clock, the 96MHz PLL clock, which usually used as high-speed system clock

POR: Power-On Reset. Reset signal generated when the chip system is powered on

NVR: Non-Volatile Register. A storage area in the flash that is different from the main area.

IAP (In-Application Programming): IAP means that the Flash of the microcontroller can be reprogrammed while the user program is running. ICP (In-Circuit Programming): ICP means that you can use the JTAG protocol, SWD protocol or bootloader to program the Flash of the microcontroller when the device is installed on the Subscriber Circuit Board.

CW: Clock wise

CCW: Counter clock wise

Option bytes: Option byte, the MCU configuration byte saved in Flash

2 Address Space

The data bytes are stored in memory in little-endian format. The lowest address byte in a word is considered the least significant byte of the word, and the highest address byte is the most significant byte. All other unallocated on-chip memory and external memory are reserved address spaces.

Table 2-1 System Address Space Allocation

Peripheral	Clock/Soft Reset	Start Address	End Address	Size	Description
FLASH	Same as bus	0x0000_0000	0x0000_7FFF	32KB	FLASH memory space
RAM	Same as bus	0x2000_0000	0x2000_09FF	2560B	RAM
SYS	Same as bus	0x4000_0000	0x4000_03FF	1kB	SYSTEM control, Clock / Reset Management
FLSCR	Same as bus	0x4000_0400	0x4000_07FF	1kB	FLASH control registers
SPI	Peripheral frequency division clock [0] Soft reset [0]	0x4001_0000	0x4001_03FF	1kB	SPI interface
I2C	Peripheral frequency division clock [0] Soft reset [0]	0x4001_0400	0x4001_07FF	1kB	I2C interface
IO MASK	Allocation	0x4001_0800	0x4001_0BFF	1kB	IO MASK
CMP	Same as bus	0x4001_0C00	0x4001_0FFF	1kB	Comparator
HALL	Peripheral frequency division clock [1] Soft reset[1]	0x4001_1000	0x4001_13FF	1kB	HALL interface
ADC	ACLK	0x4001_1400	0x4001_17FF	1kB	ADC interface
TIMER	Peripheral frequency division clock [2] Soft reset [2]	0x4001_1800	0x4001_1BFF	1kB	General Purpose Timer
MCPWM	Peripheral frequency division clock [3] Soft reset[3]	0x4001_1C00	0x4001_1FFF	1kB	Motor Control Pulse Width Modulation
GPIO	Same as bus	0x4001_2000	0x4001_23FF	1kB	General Purpose Input / Output
CRC	Same as bus	0x4001_2400	0x4001_27FF	1kB	Cyclic redundancy check
UART0	Peripheral frequency division clock [4] Soft reset [4]	0x4001_2800	0x4001_2BFF	1kB	
UART1	Peripheral frequency division clock [5] Soft reset [5]	0x4001_2C00	0x4001_2FFF	1kB	
Co-processor	Same as bus	0x4001_5000	0x4001_5FFF	4kB	

3 Interrupt

The nested vectored interrupt controller is located inside Cortex-M0. .When an interrupt event occurs, it will notify the M0 to suspend the execution of the main program, and enter the interrupt service function according to priority setting.

It can support up to 32 independent interrupt sources and interrupt vectors, of which 18 interrupt sources are used in the LKS32MC05X series chips, and the last 14 are reserved.

It supports up to four interrupt priority levels for programming.

Table 3 Interrupt Number List

Interrupt No.	Description	Interrupt No.	Description
-14	NMI		
-13	HardFault		
-12	Reserved		
-11			
-10			
-9			
-8			
-7			
-6			
-5	SVCall		
-4	Reserved		
-3			
-2	PendSV		
-1	SysTick		
0	TIMER0	16	Reserved
1	TIMER1	17	Reserved
2	TIMER2	18	Reserved
3	TIMER3	19	Reserved
4	I2C	20	Reserved
5	SPI	21	Reserved
6	GPIO	22	Reserved
7	HALL	23	Reserved
8	UART0	24	Reserved
9	UART1	25	Reserved
10	ADC	26	Reserved
11	MCPWM	27	Reserved
12	CMP	28	Reserved
13	WAKEUP, wakeup interrupt	29	Reserved
14	Reserved	30	Reserved
15	Reserved	31	Reserved

4 Analog Circuit

4.1 Introduction

The analog circuit contains the following modules:

- Built-in one channel of 12bit SAR ADC, sampling rate 2MHz. 16 channels
- Built-in 2 operational amplifiers. PGA mode is available.
- Built-in 2 comparators. Hysteresis mode is available.
- Built-in 12bit digital-to-analog converter (DAC)
- Built-in ± 2 ℃ temperature sensor
- Built-in high-accuracy reference source

The interrelationship between the modules and the control register of each module (see the "Analog Register Table" below for register description) are shown in the figure below.

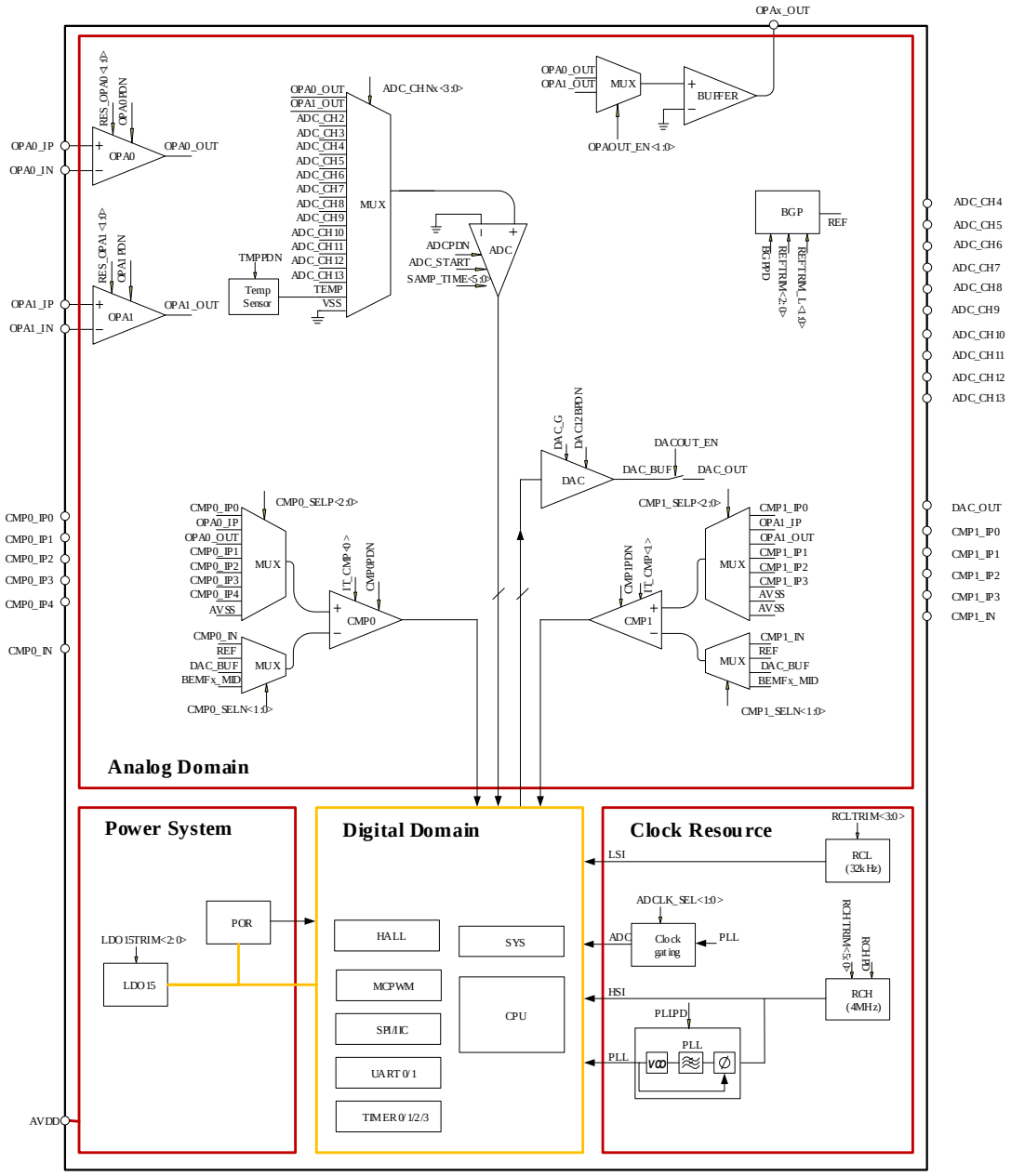


Fig. 4-1 Functional Block Diagram of Analog Circuit

4.2 Power Management System

The power management system is composed of LDO15 module and power-on/power-off reset module (POR).

The chip is powered by single 3.3 ~ 5V power supply to save off-chip power costs. All internal digital circuits and PLL modules are powered by an internal LDO15.

The LDO automatically turns on after power-on, without software configuration, and the LDO output voltage can be adjusted through software.

The LPOR module monitors the voltage of the LDO15 and provides a reset signal for the digital circuit to avoid faults occurred when the voltage of LDO15 is lower than 1.25V (e.g.: power-on or power-off).

The HPOR module monitors the voltage of the AVDD and provides a reset signal for the digital circuit to avoid faults occurred when the voltage of AVDD is lower than 2.3V (e.g.: power-on or power-off).

4.3 Clock System

The clock system consists of a 64KHz RC oscillator, an internal 4MHz RC oscillator and a PLL.

The 64 kHz RC clock LSI is mainly used for watchdog module and reset/wakeup source filters in the system. The 4MHz RC clock can be used as the main clock of the MCU, and can provide a reference clock to PLL. PLL clock is up to 96MHz.

Both 64kHz and 4MHz RC clocks have been through the correct calibration procedure at the factory, in the range of -40 ~ 105 ℃, the accuracy of the 64 KHz RC clock is ± 50%, and the accuracy of the 4MHz RC clock is ± 1%.

The 4MHz RC clock is turned on by setting RCHPD = ’0’ (ON by default, turn off when set to "1"). The RC clock needs a reference voltage and current provided by the Bandgap voltage reference module; thus, do remember to turn on the BGP module (BGPPD="0") before turning on the RC clock. When the chip is powered on, the 4MHz RC clock and BGP module are both turned on automatically. 64kHz RC clock is always turned on and cannot be turned off.

The PLL multiplies the 4MHz RC clock to provide a higher frequency clock for modules like MCU and ADC. The highest frequency of MCU and PWM module is 96MHz, and the highest frequency of ADC module is 48MHz. It can be set to different frequency by the register ADCLKSEL [1:0].

PLL is turned on by setting PLLPDN = ’1’ (OFF by default, turn on when set to '1'). Before turning on the PLL module, the BGP module should be also turned on first. After the PLL is turned on, it needs a settling time of 6us to achieve a stable frequency output. When the chip is powered on, the RCH clock and BGP module are both turned on. PLL is OFF by default and enabled by software.

For the description of ADCLKSEL<1:0>, see the analog register SYS_AFE_REG7

For the description of BGPPD/RCHPD/XTALPDN/PLLPDN, see the analog register SYS_AFE_REG5

4.4 Bandgap Voltage Reference

Bandgap Reference (BGP REF) provides reference voltage and current for ADC, DAC, RC clock, PLL, temperature sensor, operational amplifier, comparator and FLASH. Turn on the Bandgap before using any of the above modules.

When the chip is powered on, the BGP module is turned on automatically. The voltage reference

is turned on by setting BGPPD = ’0’. From OFF to ON, BGP needs about 6 us to stabilize. BGP output voltage is about 1.2V, and accuracy is ± 0.8%

The voltage of the bandgap reference can be set by the registers RETRIMM_L <1: 0>, REF_LTRIM and RETRIMM <2: 0>. The bandgap reference has been calibrated before it leaves the factory. Generally, users do not need to configure these registers again. If fine-tuning of the voltage is required, please read the original setting first, and then calculate the new settings accordingly.

For the description of BGPPD, see the analog register SYS_AFE_REG5

4.5 ADC module

Please refer to Chapter 9

4.6 Operational Amplifier

Two channels of rail-to-rail operational amplifiers are integrated, with a built-in feedback resistor, and an external resistor R_0 connected to the signal source on the pin. The resistance of feedback resistors R_2: R_1 can be adjusted by register RES_OPAx [1:0] to achieve different gains.

For the description of RES_OPAx<1:0>, see the analog register SYS_AFE_REG0

The schematic diagram of the amplifier is as follows:

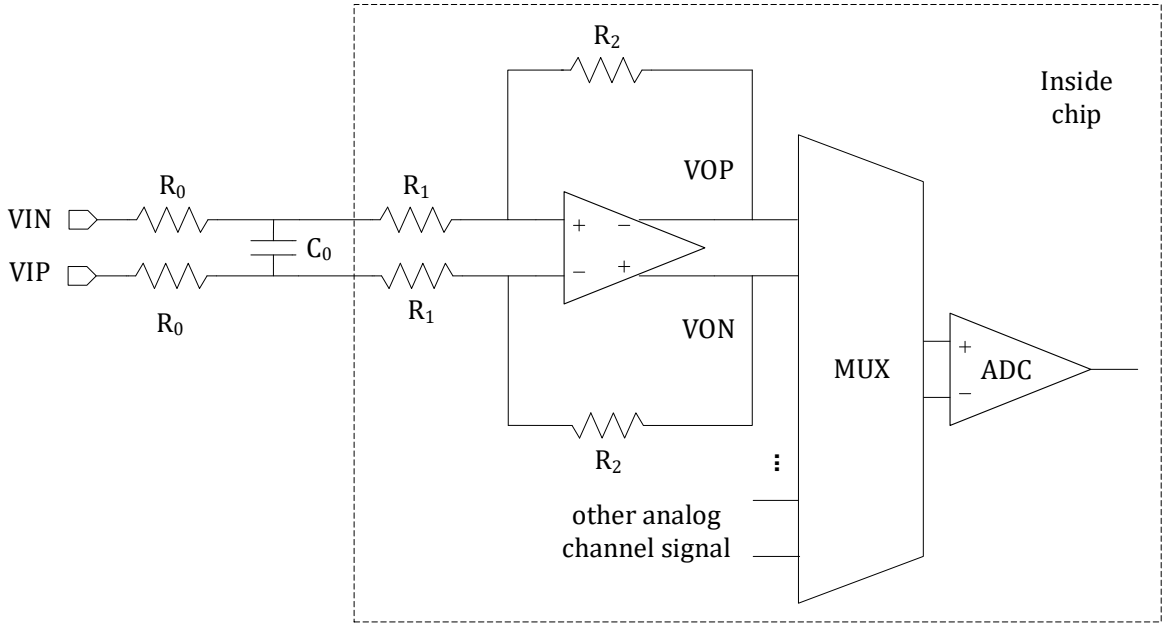


Fig. 4-2 Amplifier Block Diagram

The two R_0 in the figure are the resistance of the external resistor, so the resistance must be equal. The final amplification is $R_2/(R_1+R_0)$.

For the application of MOS resistance direct sampling, it is recommended to connect an external resistance of >20kΩ to reduce the current flowing into the chip pin to avoid the voltage signal rises to tens of volts when lower MOS tube is turned off, and the upper tube is turned on;

For the application of shunt resistance sampling, it is recommended to connect an external resistor of 100 ~ 2KΩ. C_0 is the signal filter capacitor; it forms a first-order RC filter circuit with R_0. The specific resistance of R_0 can be determined based on the filter constant of R_0* C_0. The filtering is not necessary if the Signal to Noise Ratio (SNR) is small, or C_0 can be omitted when the signal requires a large bandwidth (higher response speed).

The amplifier can select one of the output signals of the 2-channel amplifiers by setting OPAOUT_EN <1:0>, and send it to the P2.7 IO port through a buffer for measurement (see the corresponding relationship in the datasheet 'Pin Functional Description"). Because of the BUFFER, the operational amplifier is also able to send one output signal in the normal working mode.

For the description of OPAOUT_EN<1:0>, see the analog register SYS_AFE_REG0

When the chip is powered on, the amplifier module is OFF by default. It can be turned on by setting OPAxPDN = 1 (x=0, 1, 2, 3). Turn on the BGP module before turning on the amplifier.

For the description of OPAxPDN, see the analog register SYS_AFE_REG5

For built-in clamp diodes are integrated between the positive and negative operational amplifier inputs, the motor phase line could be directly connected to the input terminal through a matching resistor, thereby simplifying the external circuit for MOSFET current sampling.

The chip pins have 4 groups of OPA input differential signals, including the OPA0_IP/OPA0_IN and the OPA2_IP/OPA2_IN time-time-multiplexing OPAMP OPA_A, the OPA1_IP/OPA1_IN and the OPA3_IP/OPA3_IN time-time-multiplexing OPAMP OPA_B.

4.7 Comparator

Built-in 2-channel rail-to-rail comparators with programmable comparator speed, hysteresis voltage, and signal source.

The comparison delay can be set to 0.15 us, and can be set to be less than 30 ns. The hysteresis voltage can be set to 20mV/0mV by CMP_HYS.

The signal sources of the positive and negative input of the comparator can be set by the registers CMPx_SELP[2:0] and CMPx_SELN[1:0] (x=0/1, which represents the two comparators, CMP0 and CMP1).

It should be noted that The BEMFx_MID signals at the negative input terminals of the two comparators are the average of the CMPx_IP1/CMPx_IP2/CMPx_IP3 signals at the positive input terminals of the comparator. The specific connection method is shown in Fig. 4-3. Among them, the resistance R=8.2k ohms, the switch in the picture will be turned on only after the negative input signal of the comparator is selected as BEMFx_MID, otherwise the switches are in the off state.

BEMFx_MID is mainly used for BLDC square wave mode control, the virtual motor phase line

center point voltage, used for back-EMF zero-crossing detection. After the three phase lines are divided, connect to CMPx_IP1, CMPx_IP2, and CMPx_IP3 respectively. The MCU controls the negative end of the comparator to select BEMFx_MID, and the multiplexer at the positive end of the comparator selects CMPx_IP1, CMPx_IP2, and CMPx_IP3 in a time-division multiplexing manner. Compare the zero-crossing point of the back EMF.

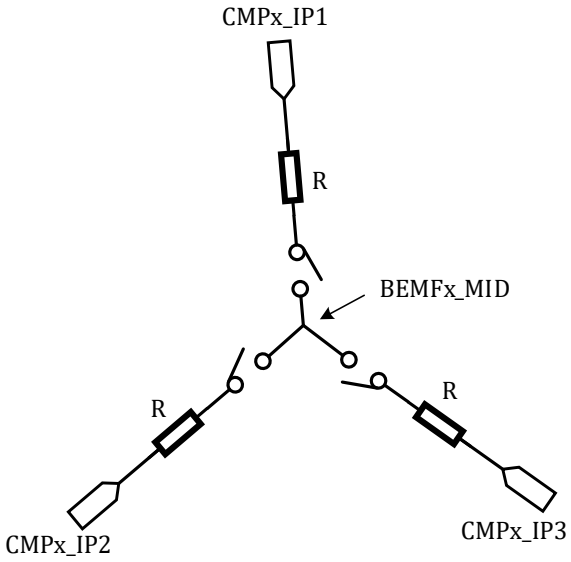


Fig. 4-4 BEMFx_MID Signal

The output of the comparator can be read through the register CMP_DATA.

For the description of CMPx_SELN<1:0>/ CMPx_SELP<2:0>/ CMP_HYS, see the analog register SYS_AFE_REG3

When the chip is powered on, the comparator module is OFF. The comparator is turned on by setting CMPxPDN=1 (x=0,1), and turn on the BGP module before turning on the comparator.

For the description of CMPxPDN, see the analog register SYS_AFE_REG5

4.8 Temperature Sensor

The chip has a built-in temperature sensor with an accuracy of 2°C in the range of -40°C to 85°C. The accuracy is 3℃ in the range of 85℃ to 105℃.

The operating temperature of chips will be corrected before leaving the factory, and the corrected value is saved in the flash info area.

When the chip is powered on, the temperature sensor module is OFF. Turn on the BGP module before turning on the temperature sensor.

The temperature sensor is turned on by setting TMPPDN = '1', and it takes about 2us to be stable after turning on. Thus, it should be turned on at least 2us ahead before the ADC measures the sensor output.

The temperature sensor signal is connected to channel 14 of the ADC.

For the ADC settings, please refer to Section Analog to Digital Converter(ADC)

For the description of TMPPDN,, see the analog register SYS_AFE_REG5

The typical curve of the temperature sensor is shown in the figure below:

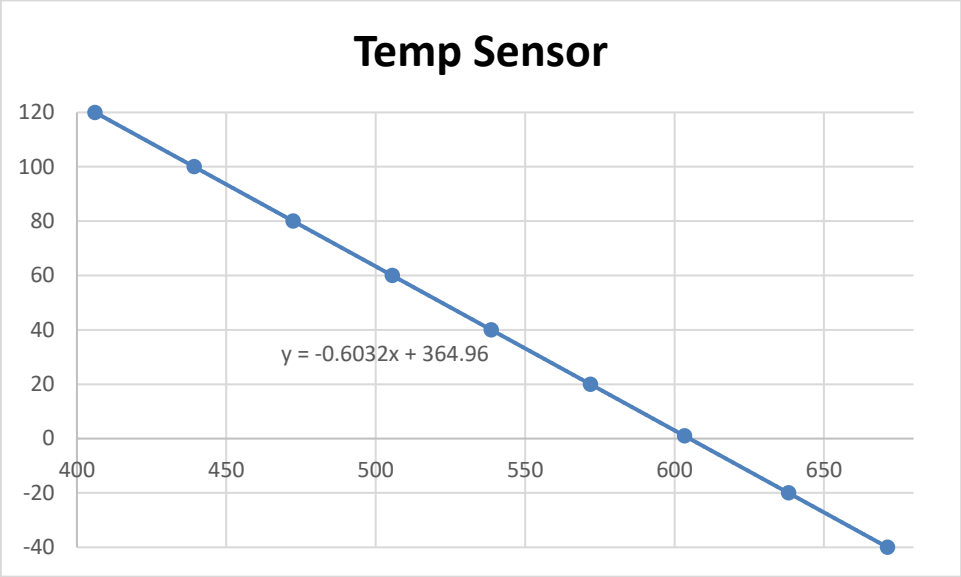


Fig. 4-5 Temperature Sensor Curve

The X axis in the figure is the ADC value corresponding to the temperature signal of the temperature sensor, and the Y axis is the temperature of the sensor. When measuring temperature, configure the sensor-related registers according to the above requirements, and put the ADC value as X into the formula after the value obtained:

y=-0.6032x+364.96

The calculated value Y is the current temperature.

There are two coefficients in the formula, a = -0.6032, b = 364.96, and the value of the coefficient b differs from different chips. The temperature sensor will be calibrated in the factory. Write the coefficient b into the Flash info area, and the address is 0x0000039C. The decimal point of the coefficient b will be shifted to the right by one digit (multiplied by 10) and stored in the info area. The second digit after the decimal point will not be saved.

For the convenience of customers, the coefficient a will also be stored in the Flash info area, and the address is 0x00000398. The decimal point of the coefficient a will be shifted to the right by four digits (multiplied by 10000) and stored in the info area.

In actual use, the coefficient a and b should be read first from the address in different Flash info area, and then put the measured ADC value into the formula to calculate the current temperature. The unit is degree Celsius. When calculating, pay attention to the digits of the decimal points of coefficient a and b, i.e., the coefficient a should be divided by 10000, and the coefficient b should be

divided by 10.

Please note that the above calculation formula is implemented based on ADC right-aligned mode. If the calculation adopts left-aligned mode, the ADC sampling value should be shifted right by four bits before putting into the above formula.

4.9 Digital-to-analog Converter (DAC) Module

The chip has a 1-channel 12 bit DAC, and the maximum range of the output signal can be set to 1.2V/4.8V through the register DAC_G.

The 12bit DAC can be output via IO port P0.0 by setting register DACOUT_EN = 1, which can drive a load resistance of over 5kΩ and a load capacitance of 50pF.

The maximum output code rate of the DAC is 1MHz.

When the chip is powered on, the DAC module is OFF. DAC can be turned on by setting DAC12BPDN = 1. Turn on the BGP module before turning on the DAC module.

The input digital signal register of DAC is SYS_AFE_DAC, low 12BIT is effective. The signal range is 0x000 ~ 0xFFF. The zero analog output corresponding to the signal range 0x000 is 0V, and the full-scale analog output corresponding to 0xFFF is DAC_{fs}. As mentioned above, the value of DAC_{fs}can be set by the DAC12B_FS register. The analog signal amplitude corresponding to each gear signal (LSB) is .$\frac{DAC_{fs}}{4096}$. If the digital value of SYS_AFE_DAC is Din, the analog signal of the DAC output corresponding to the digital signal is $\frac{DAC_{fs}}{4096} * Din$

There are manufacturing deviations in the DAC by using different chips. Thus, the DAC has a calibration hardware module to offset the deviations. The DAC output formula is: y=ax+b. x is the value filled in SYS_AFE_DAC (ideal digital quantity). a is the value filled in SYS_AFE_DAC_AMC register and b is the value filled in SYS_AFE_DAC_DC register. The hardware multiplies and adds SYS_AFE_DAC, SYS_AFE_DAC_AMC and SYS_AFE_DAC_DC to obtain the corrected digital quantity, and sends it to the DAC input terminal, so that the final analog value of the DAC output is the ideal digital quantity. After the system is powered on, the calibration value of 3V is loaded by default. If it is changed to another range, the software will read the flash info area and reload it into the corresponding register.

Address 0x00000330 is the parameter a of the 1.2V range, and address 0x00000340 is the parameter b of the 1.2 V range.

The address 0x00000334 is the parameter a of the 4.8V range, and the address 0x00000344 is the parameter b of the 4.8 V range.

Except for external module usage via IO port, the analog signal output by the DAC can also be used as a reference signal for the comparator by connecting the configuration register to the negative side of the two-channel comparator inside the chip. See the section Comparator for details.

For the description of DACOUT_EN and DAC_G, see the analog register SYS_AFE_REG3

For the description of DAC12BPDN, see the analog register SYS_AFE_REG5

For the description of SYS_AFE_DAC, see the register SYS_AFE_DAC

5 System Clock Reset

5.1 Clock

5.1.1 Clock source

As shown in the following table, the system includes 5 clock sources, of which the internal low-speed RC oscillator (LSI, Low Speed Internal Clock) and internal high-speed RC oscillator (HSI, High Speed Internal Clock) will not stop vibration.

Table 5-1 System Clock Source

Clock source	Frequency	Source	Error	Description
LSI	64kHz	Internal RC Oscillator	Full temperature range error <50%	Internal system clock, used for watchdog module, and filtering and widening of reset signal.
HSI	4MHz	Internal RC Oscillator	Full temperature range error <1%	Can be used as PLL source clock
PLL	96MHz	PLL clock	0	Taking the HSI as the input, PLL outputs the clock that is 24 times the frequency of the HSI clock, which used as the main system clock.
SWD	1MHz	Debugger		SWD's JTAG clock

*The typical value and actual size of the SWD clock rate are related to the hardware environment.

The system can use the internal high-speed RC clock HSI as the reference clock of the PLL. The PLL multiplies the 4MHz reference clock HSI by 24 times to 96MHz.

After the PLL is divided by n/8, the high-speed clock of 96MHz × n/8 can be obtained. SYS_CLK_CFG.CLK_SEL chooses one of the divided high-speed clock and the 4MHz CLK_HS as the main system clock (MCLK). When the system is reset, the PLL is turned off and the HSI is turned on by default. The system selects the HSI clock, i.e., 4MHz as the main system clock, so as to ensure that the power is low when the system is powered on.

MCLK is the main system clock. The n/8 frequency division can be controlled by the CLK_DIV bit field in the SYS_CLK_CFG register, which can generate 12, 24, 48, 96 MHz and other frequency values. SYS_CLK_CFG.CLK_SEL means PLL or CLK_HS is selected as the main system clock. When SYS_CLK_CFG.CLK_SEL is 1, SYS_CLK_CFG.CLK_DIV is used as the PLL frequency division factor. When SYS_CLK_CFG.CLK_SEL is 0, SYS_CLK_CFG.CLK_DIV has no effect.

Table 5-2 Frequency Division Configuration When PLL is Used as MCLK Clock

SYS_CLK_CFG	Frequency	Frequency/MHz	If Uniform

	division coefficient		
0x0101	1/8	12	Yes
0x0111	2/8	24	Yes
0x0155	4/8	48	Yes
0x01FF	8/8	96	Yes

The MCLK clock is supplied to the peripheral clock after the switch controlled by the SYS_CLK_FEN register. The I2C clock could be further divided by the SYS_CLK_DIV0 register, and the UART clock could be further divided by the SYS_CLK_DIV2 register.

The clock output by the PLL is used as the ADC clock (typical frequency is 48MHz) after being divided by 2/4/8 controlled by SYS_AFE_REG7.ADCLKSEL, which is ACLK.

The 64kHz RC oscillator generates an LSI clock (LCLK), which is mainly used for the WDT working clock, as well as part of the system control, reset filtering and so on.

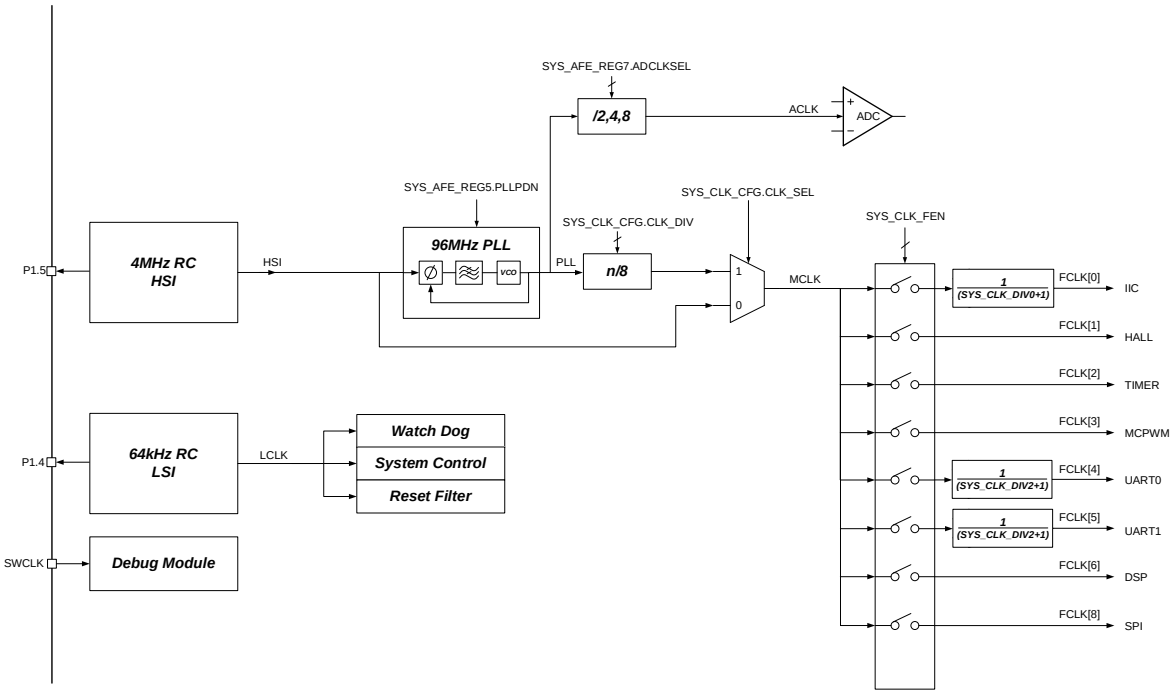


Fig. 5-1 Clock Architecture

To ensure the system reliability, the clock system has a mechanism to prevent the clock from being shut down by mistake. For example, when the PLL is used as the main clock, the PLL cannot be turned off, and the HSI, which is the reference clock, cannot be turned off by software; Once powered on, the 64kHz LSI clock start operating and cannot be turned off. SWCLK is provided by the debugger, and the frequency can be selected in the debug interface.

To facilitate debugging and factory calibration, the high-speed RC oscillator HSI and low-speed clock LSI can be output through chip pins by setting the second function of GPIO.

5.1.2 Power Management and Sleep Wakeup

5.1.2.1 Sleep

MCLK can be gated by configuration, so that most digital circuits including the CPU and all peripherals are in a sleep state. During gating, the PMU state machine turns off analog modules such as PLL, HSI/HSE, and BGP in order to reduce power consumption.

You can refer to the sleep routine to turn off the clock of each module in the digital part and turn off the analog ADC/OPA/CMP/DAC module before entering sleep.

Please note that only high-speed clocks such as PLL and HSI are turned off when the system enters a sleep mode, and the LSI clock is still working. If the LSI-driven watchdog is enabled, the watchdog reset can be considered as a global reset and return the system to the initial state and start working again.

Write 0xDEAD to the SYS_CLK_SLP register to make the chip enter the sleep state, and then execute the __WFI () macro instruction to stop the CPU from fetching instructions.

Please configure the wake-up conditions when programming the application.

5.1.2.2 Wakeup

After sleep, external IO events and internal wake-up Timer can be used as wake-up sources.

The internal wake-up Timer is an independent Timer independent of the UTimer module. It uses an LSI clock, which is different from the general Timer and serves for the main system clock. The wake-up Timer can be set to a total of 8 wake-up time intervals of 0.25s, 0.5s, 1s, 2s, 4s, 8s, 16s and 32s by the SYS_RST_CFG.WK_INTV.

Only the four IOs: P0 [1: 0] and P1 [1: 0], can be used as external wake-up IOs, and has independent enable and polarity. Please note that the external IO wake-up is a level trigger. If the external IO is at the wake-up level, it will wake up the chip immediately after it sleeps.

When programming the application, please try to avoid entering the sleep mode once the chip is powered on. If the internal wake-up Timer is the wake-up sources, the chip will sleep again after waking up, resulting in failure of ordinary downloaders to connect and debug, and an offline downloader provided by the chip vendor will be needed for application erasure and rewriting.

After the wake-up event occurs, the system will turn on Bandgap, 4MHz RC clock, PLL clock, etc., and generate a WAKEUP interrupt to notify the CPU. Usually this interrupt wakes the CPU from the __WFI() instruction.

5.1.2.3 Peripheral Clock Gating

The peripheral clock is divided by the system high-speed clock MCLK; it can be close turned off by setting the SYS_CLK_FEN register gating when the peripheral is not in use. There are 7 peripheral clocks available for different peripheral modules that could be turned off when not being in use, and each peripheral clock has a clock gating. The gated clock is turned off by default after power-on, and should be turned on by software before using the corresponding peripheral module.

I2C: FCLK[0]

Hall module: FCLK [1]

Timer module: FCLK [2]

MCPWM module: FCLK [3]

UART0/UART1: FCLK [4]/ FCLK [5]

Co-processor: FCLK [6]

SPI: FCLK [8]

5.1.2.4 Peripheral Clock Divider

Some peripherals have independent clock dividers, which enables it to work with an appropriate frequency.

Among them, SYS_CLK_DIV[0] is the division factor of I2C, while SYS_CLK_DIV[2] is the division factor of UART0/1. Besides, the UART baud rate has an additional clock divider inside the UART module.

5.2 Reset

5.2.1 Reset Source

The reset source of the chip includes hardware reset and software reset.

5.2.1.1 Hardware reset

As shown in Table 5-3 Hardware Reset Sources, the system has four hardware reset sources. The resets generated are all chip global resets. After the reset, the chip program counter returns to address 0, and all registers are restored to their default values. The four hardware resets are all active low.

Table 5-3 Hardware Reset Sources

Name	Source	Description
LPORn	Internal 1.5V Power Management	Monitor 1.5V digital power supply, reset when it's below 1.25V
HPORn	Internal 3.3V Power Management	Monitor 3.3V digital power supply, reset when it's below 2.5V
RESETn	External Keys	External RC reset circuit
WDTn	Hardware watchdog	If not feed the watchdog, then it will reset the CPU at a regular time, and the reset interval is configurable.

5.2.1.1.1 Hardware Reset Architecture

As shown below, LPORn/HPORn is an internal analog circuit, and RSTn is an external key.

WDTn is a LSI clock cycle width signal, which is an internal digital signal. WDTn signal will be shielded in Debug mode.

After pre-filtering and broadening the reset signal, a stable and reliable reset signal is output through AND algorithm.

A reset signal short than 16us to P0.2 will be filtered. A reliable reset signal should be as long as 200us.

The four reset signals are global reset, so the reset levels and scopes are the same.

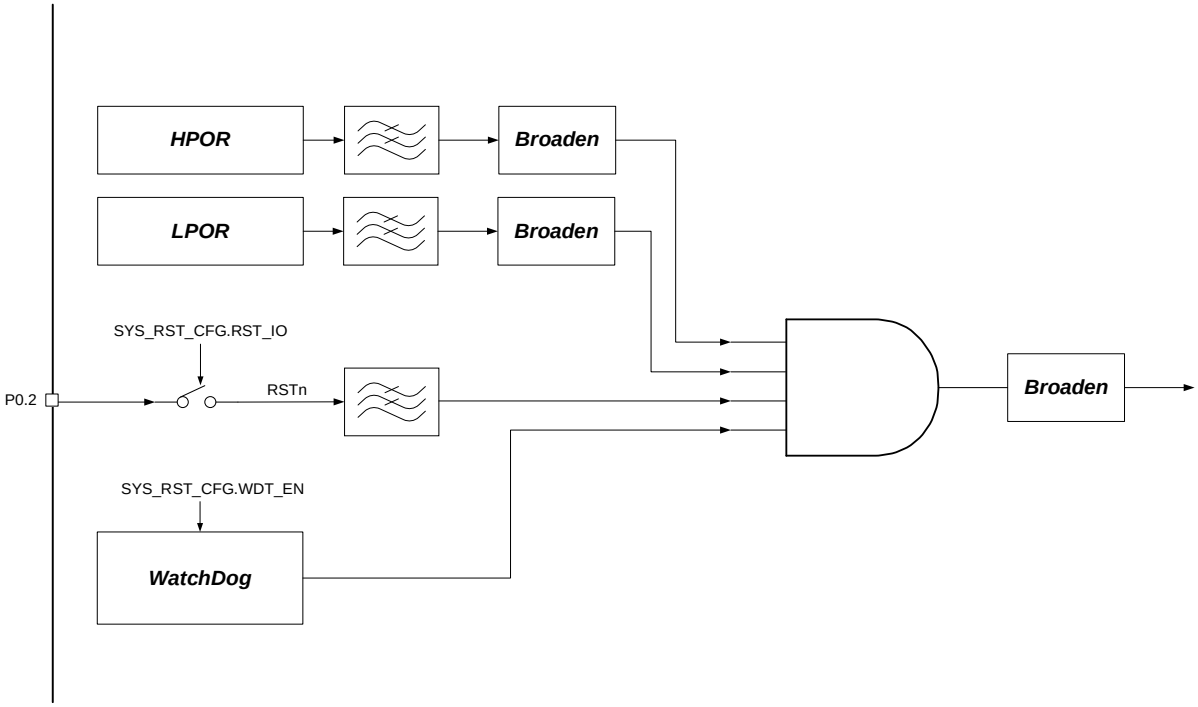


Fig. 5-2 Hardware Reset Architecture

5.2.1.1.2 Hardware Reset Records

The SYS_RST_SRC register is used to save hardware reset events. When a hardware reset occurs, the corresponding bit of SYS_RST_SRC is set. The SYS_RST_SRC register itself be reset by the reset signal; it can only clear the record by writing 0xCA40 to the SYS_CLR_RST register. With the reset record, we can easily understand whether and what kind of reset has occurred.

5.2.1.2 Software Reset

CPU soft reset can return the PC (PC: Program Counter) to address 0, but it has no effect on the registers in all peripherals.

In the IDE (IDE: Integrated Development Environment) debug mode, clicking "Reset" has the same effect as "CPU Soft Reset", which will only return the PC to address 0, and do not affect the registers in all peripherals. However, if a soft reset of the peripheral module is performed in the bootloader, the peripheral register will be reset to the default value. For further details, please

contact the chip vendor.

Some peripheral modules have a soft reset, which is performed by using the SYS_SFT_RST register. Write the corresponding bit to the register, restore the module state machine to its initial state, and restore the module register to the default value.

5.2.2 Reset Scope

Table 5-4 Reset Action Scope

Reset Source	Action scope
LPORn	Internal 1.5V power management, global reset
HPOR	Internal 3.3 V power management, global reset, except for very few registers
RSTn	External keys, global reset, except for very few registers
WDTn	Hardware watchdog, global reset, except for very few registers
SYS_SFT_RST.ADC_SFT_RST	ADC module
SYS_SFT_RST.SPI_SFT_RST	SPI module
SYS_SFT_RST.DSP_SFT_RST	Co-processor module
SYS_SFT_RST.UART1_SFT_RST	UART1 module
SYS_SFT_RST.UART0_SFT_RST	UART0 module
SYS_SFT_RST.MCPWM_SFT_RST	MCPWM module
SYS_SFT_RST.UTIMER_SFT_RST	UTIMER module
SYS_SFT_RST.HALL_SFT_RST	HALL module
SYS_SFT_RST.I2C_SFT_RST	I2C module
NVIC_SystemReset();	CPU soft reset, only resets the CPU core. When the PC is reset to 0, all peripheral register values are still maintained.

SYS_RST_CFG.RST_IO (used to control whether P0[2] is used as a GPIO or an external reset pin), SYS_RST_CFG.WDT_EN (used to control watchdog enable); reset record register SYS_RST_SRC is only affected by LPOR reset and is not reset by other reset signals.

Generally, a global reset will reset all-chip registers, including CPU core registers and all peripheral registers, except for the very few registers aforesaid.

Because the CPU soft reset only resets the CPU core and does not reset the peripheral registers, it is recommended to reset the peripheral registers by power-off and power-on or external reset after re-programming.

Flash storage content and SRAM storage content are not affected by reset.

5.3 System Register

5.3.1 Address Allocation

The base address of the system module register is 0x4000_0000.

Table 5-5 System Control Register

Name	Offset	Description
SYS_WDT_PSW	0x00	Watchdog password register

SYS_WDT_CLR	0x04	Watchdog reset register
SYS_WDT_TH	0x08	Watchdog counter threshold value
SYS_WDT_CNT	0x0C	Current count value of watchdog counter
SYS_OPA_SEL	0x14	OPA Channel Select register
SYS_AFE_REG0	0x20	Analog configuration register 0
SYS_AFE_REG1	0x24	Analog configuration register 1
--	--	--
SYS_AFE_REG3	0x2C	Analog configuration register 3
--	--	--
SYS_AFE_REG5	0x34	Analog configuration register 5
SYS_AFE_REG6	0x38	Analog configuration register 6
SYS_AFE_REG7	0x3C	Analog configuration register 7
--	--	--
SYS_TMP_A	0x54	Temperature sensor factor A
SYS_TMP_B	0x58	Temperature sensor factor B
SYS_AFE_DAC	0x60	DAC digital register
SYS_DAC_AMC	0x64	DAC gain correction register
SYS_DAC_DC	0x68	DAC DC offset register
SYS_CLK_CFG	0x80	Clock control register
SYS_RST_CFG	0x84	Reset control register
SYS_RST_SRC	0x88	Reset source record register
SYS_CLR_RST	0x8C	Reset source record clearing register
SYS_CLK_DIV0	0x90	Peripheral clock divider register 0
SYS_CLK_DIV2	0x98	Peripheral clock divider register 2
SYS_CLK_FEN	0x9C	Peripheral clock gating register
SYS_CLK_SLP	0xA0	Sleep register
SYS_SFT_RST	0xAC	Soft reset register
SYS_PROTECT	0xB0	Write protection register

5.3.2 SYS_WDT_PSW Watchdog Password Register

Address: 0x4000_0000

Reset value: 0x0

Table 5-6 SYS_WDT_PSW Watchdog Password Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	SYS_WDT_PSW	Write 0xA6B4 before writing to WDT_CLR/WDT_TH, etc. Writing to SYS_WDT_CLR or SYS_WDT_TH will clear the password, so the password needs to be written in every time that the watchdog is written

5.3.3 SYS_WDT_CLR Watchdog Clear Register

Address: 0x4000_0004

Reset value: 0x0

Table 5-7 SYS_WDT_CLR Watchdog Clear Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	SYS_WDT_CLR	Write byte 16'b0111_1001_1000_1B2B1B$_0$, the upper 15-bit is the password, and B [0] can be written only when the password is correct. B[0] is CLR. Write 1 to reset WDT counter as TH, and such bit will be cleared automatically after such bit is written.

5.3.4 SYS_WDT_TH Watchdog Threshold Register

Address: 0x4000_0008

Reset value: 0x001FF000

Table 5-8 Watchdog Threshold Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											SYS_WDT_TH				
											RW				
											0x1F				

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

Location	Bit name	Description
[31:21]		Unused
[20:12]	SYS_WDT_TH	The watchdog threshold; the watchdog starts counting from TH with the LRC clock, and resets when counting to 0. Writing 9'h000 to this register segment will be forced to rewrite to 9'h001 by hardware. Write the correct password to SYS_WDT_PSW before rewriting the SYS_WDT_TH register. Rewriting SYS_WDT_TH also has the function of resetting the watchdog counter, and the watchdog will start counting from the new TH.
[11:0]		Unused

In order to prevent SYS_WDT_TH from being written as 0, when the software write value is all 0, the hardware will forcibly rewrite it to 0x1000, and the lower 12 bits of the register are always 0. The minimum reset time interval is 4096/64kHz = 0.064s. The reset time range is 0.064~32s, with 0.064s as the minimum step, which can be continuously configured.

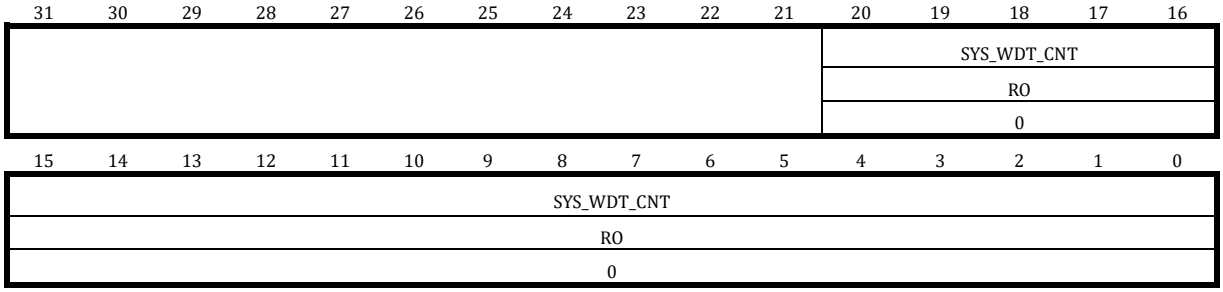
In the full temperature range, the 64k RC clock will still have a deviation of about ±16%, so to be on the safe side, it is recommended that the watchdog feeding time should be 20~30% earlier than the theoretical value calculated by the 64kHz.

5.3.5 SYS_WDT_CNT Watchdog Count Value Register

Address: 0x4000_000C

Reset value: 0x0

Table 5-9 SYS_WDT_CNT Watchdog Count Value Register



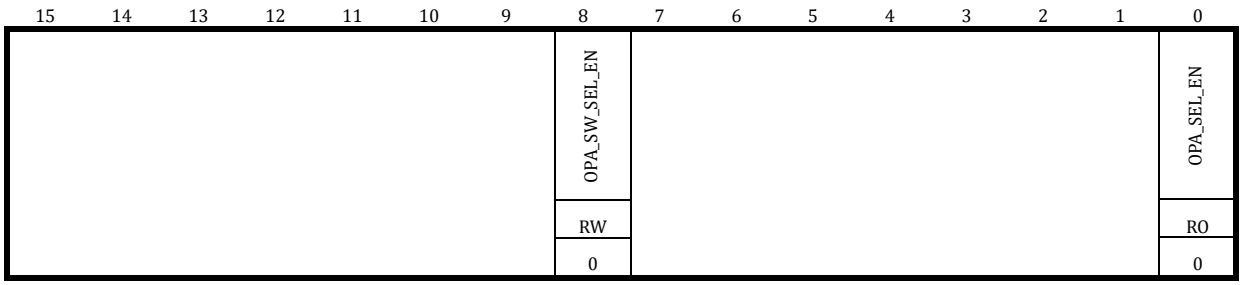
Location	Bit name	Description
[31:21]		Unused
[20:0]	SYS_WDT_CNT	Current count value of watchdog; such value ≤SYS_WDT_TH.

5.3.6 SYS_OPA_SEL OPA Channel Select Register

Address: 0x4000_0014

Reset value: 0x0

Table 5-10 SYS_OPA_SEL



Location	Bit name	Description

[31:9]		Unused
[8]	OPA_SW_SEL_EN	ADC_CH8/9 Signal source 0：OPA2/3 output 1：ADC_CH8/9 at IO
[7:1]		Unused
[0]	OPA_SEL_EN	OPA2/OPA3 could be used, active high. This reg is controlled by hardware and read-only

Table 5-11 ADC Sample signal source and register settings

ADC_CHNn<3:0>	0000: OPA0 Output; 0001: OPA1 Output; 0010: ADC_CH2; 0011: ADC_CH3; 0100: ADC_CH4; 0101: ADC_CH5; 0110: ADC_CH6; 0111: ADC_CH7; 1000: ADC_CH8/OPA2 Output; 1001: ADC_CH9/OPA3 Output; 1010: ADC_CH10; 1011: ADC_CH11; 1100: ADC_CH12; 1101: ADC_CH13; 1110: Temperature Sensor; 1111: GND;

Where CH8/CH9 could be shared by OPA2/3

Table 5-12 ADC channel settings and OPA multiplexing

OPA_SEL_EN	OPA_SW_SEL_EN	ADC Channel setting	ADC Channel actually sampled
0	X	8	ADC_CH8
		9	ADC_CH9
		8	ADC_CH8
		9	ADC_CH9
1	0	8	OPA2_OUT
	0	9	OPA3_OUT
	1	8	ADC_CH8
	1	9	ADC_CH9

If OPA_SEL_EN=0, or OPA_SEL_EN=1 and OPA_SW_SEL_EN=1, ADC actually samples ADC_CH8/9 signal from IO when setting to sample channel 8/9.

If OPA_SEL_EN=1 and OPA_SW_SEL_EN=0, ADC actually samples the output of OPA2/3 when setting to sample channel 8/9.

5.3.7 Introduction to Analog Register

The Analog register SYS_AFE_REG0 ~ SYS_AFE_REG7, the corresponding address space is 0x40000020 ~ 0x40000050. Among them, the address space 0x40000040 ~ 0x40000050 is the calibration register for simulating each module. The calibration values of these registers are saved in the Flash info area in the factory, and will be loaded into SYS_AFE_REG8 ~ SYS_AFE_REGC

automatically after power on. Generally, users are advised not to configure or change these values. If fine-tuning of certain analog characteristics is required, please read the original corrected value first, and then carry out the fine tuning accordingly.

Address space of 0x40000020 ~ 0x4000003C is a register open to users, among which the reserved registers (Res) must all be set as 0 (it will be reset to 0 after power on). Other registers will be configured according to the actual application scenarios.

The following is a detailed description of each analog register.

5.3.8 SYS_AFE_REG0 Analog Register 0

Address: 0x4000_0020

Reset value: 0x0

Table 5-13 SYS_AFE_REG0 Analog Register 0

15 14	13 12	11	10	9 8 7	6	5 4	3 2	1 0
Resv.	Resv.	Resv.	Resv.	Resv.	GA_AD	OPAOUT_EN	RES_OPAB	RES_OPAA
RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15:14]	RESERVED	Must be 0
[13:12]	RESERVED	Must be 0
[11]	RESERVED	Must be 0
[10]	RESERVED	Must be 0
[9:7]	RESERVED	Must be 0
[6]	GA_AD	ADC gain selection 0: 2/3 times 1: 1 time
[5:4]	OPAOUT_EN	Enable OPAx output signal to be sent to IO port P2.7 00: Not output; 01: Output OPA0 signal to IO port P2.7; 10: Output OPA1 signal to IO port P2.7; 11: This configuration is not allowed
[3:2]	RES_OPAB	Operational amplifier B feedback resistor 00: 200k:10.6k 01: 190k:20.6k 10: 180k:30.6k 11: 170k:40.6k Operational amplifier B can amplify the input of OPA1/OPA3 at different times according to the configuration

[1:0]	RES_OPAA	Operational amplifier A feedback resistor 00: 200k:10.6k 01: 190k:20.6k 10: 180k:30.6k 11: 170k:40.6k Operational amplifier A can amplify the input of OPA0/OPA2 at different times according to the configuration

5.3.9 SYS_AFE_REG1 Analog Register 1

Address: 0x4000_0024

Reset value: 0x0

Table 5-14 SYS_AFE_REG1 Analog Register 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resv.	Resv.		Resv.												CMP_FT
RW	RW		RW												RW
0	0		0												0

Location	Bit name	Description
[31:16]		Unused
[15]	RESERVED	Must be 0
[14]	RESERVED	Must be 0
[13]		Unused
[12]	RESERVED	Must be 0
[11:1]		Unused
[0]	CMP_FT	Comparator quick comparison enabling switch 1: Enabling the comparison speed of comparator to be less than 30ns 0: Disabling

5.3.10 SYS_AFE_REG3 Analog Register 3

Address: 0x4000_002C

Reset value: 0x0

Table 5-15 SYS_AFE_REG3 Analog Register 3

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

DAC_G	CMP1_SELP	DACOUT_EN	CMP0_SELP	CMP_HYS	Resv.	CMP1_SELN	CMP0_SELN		LDOOUT_EN
RW	RW	RW	RW	RW	RW	RW	RW		RW
0	0	0	0	0	0	0	0		0

Location	Bit name	Description
[31:16]		Unused
[15]	DAC_G	DAC output configuration 0: DAC output configuration, full scale is 1.2V 1: DAC output configuration, full scale is 4.8V
[14:12]	CMP1_SELP	Positive end selection of comparator 1 signal 000: Connect to CMP1_IP0 001: Connect to OPA1_IP 010: Connect to OPA1_OUT 011: Connect to CMP1_IP1 100: Connect to CMP1_IP2 101: Connect to CMP1_IP3 110: Connect to AVSS 111: Connect to AVSS Note: The above are pin names except AVSS/OPA1_OUT. For the definition of the pins, please refer to the specific section in the DATASHEET.
[11]	DACOUT_EN	DAC output to IO enabling 0: Disabling 1: Output DAC results to IO port P0.0
[10:8]	CMP0_SELP	Positive end selection of Comparator 0 signal 000: Connect to CMP0_IP0 001: Connect to OPA0_IP 010: Connect to OPA0_OUT 011: Connect to CMP0_IP1 100: Connect to CMP0_IP2 101: Connect to CMP0_IP3 110: Connect to CMP0_IP4 111: Connect to AVSS Note: The above are pin names except OPA0_OUT/OPA1_OUT. For the definition of the pins, please refer to the specific section in the DATASHEET.
[7]	CMP_HYS	Comparator hysteresis selection, the default configuration is 0 0: 20mv; 1: 0mv
[6]	RESERVED	Must be 0
[5:4]	CMP1_SELN	Negative end selection of Comparator 1 signal 00: Connect to CMP1_IN 01: Connect to REF 10: Connect to DAC output 11: Connect to HALL1_MID The above CMP1_IN is the pin name, For the pin definition, please refer to the specific section in the DATASHEET; REF is the 1.2V BANDGAP reference inside the chip; the DAC output is the analog signal output by the DAC module inside the chip; HALL1_MID is

		the average value obtained by connecting the CMP1_IP1, CMP1_IP2, and CMP1_IP3 signal through a star connection.
[3:2]	CMP0_SELN	Negative end selection of comparator 0 signal 00: Connect to CMP0_IN 01: Connect to REF 10: Connect to DAC output 11: Connect to HALL0_MID Note: The above CMP0_IN is the pin name, For the pin definition, please refer to the specific section in the DATASHEET; REF is the 1.2V BANDGAP reference inside the chip; the DAC output is the analog signal output by the DAC module inside the chip; HALL0_MID is the average value obtained by connecting the CMP0_IP1, CMP0_IP2, and CMP0_IP3 signal through a star connection.
[1]		Unused
[0]	LDOOUT_EN	LDO Output and IO Enable; the default configuration is no output 0: No output 1: Output LDO to IO port 2.7

5.3.11 SYS_AFE_REG5 Analog Register 5

Address: 0x4000_0034

Reset value: 0x0

Table 5-16 SYS_AFE_REG5 Analog Register 5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLPDN		TMPPDN	DAC12BPDN		RCHPD		BGPPD	CMP1PDN	CMP0PDN			OPA1PDN	OPA0PDN		ADCPDN
RW		RW	RW		RW		RW	RW	RW			RW	RW		RW
0		0	0		0		0	0	0			0	0		0

Location	Bit name	Description
[31:16]		Unused
[15]	PLLPDN	PLL ON/OFF control; the default configuration is PLL OFF 0: PLL OFF 1: PLL ON
[14]		Unused
[13]	TMPPDN	Temperature sensor enabling; the default configuration is temperature sensor OFF 0: Temperature sensor OFF 1: Temperature sensor ON
[12]	DAC12BPDN	12BIT DAC enabling; the default configuration is DAC OFF 0: DAC OFF 1: DAC ON
[11]		Unused
[10]	RCHPD	RCH Clock ON/OFF control; the default configuration is RCH ON 0: RCH ON 1: RCH OFF

[9]		Unused
[8]	BGPPD	BGP ON/OFF control; the default configuration is BGP ON 0: BGP ON 1: BGP OFF
[7]	CMP1PDN	CMP1 enabling; the default configuration is CMP1 OFF 0: CMP1 OFF 1: CMP1 ON
[6]	CMP0PDN	CMP0 enabling; the default configuration is CMP0 OFF 0: CMP0 OFF 1: CMP1 ON
[5:4]		Unused
[3]	OPA1PDN	OPA1 enabling; the default configuration is OPA1 OFF 0: OPA1 OFF 1: OPA1 ON
[2]	OPA0PDN	OPA0 enabling; the default configuration is OPA0 OFF 0: OPA0 OFF 1: OPA0 ON
[1]		Unused
[0]	ADCPDN	ADC enabling; the default configuration is ADC OFF 0: ADC OFF 1: ADC ON

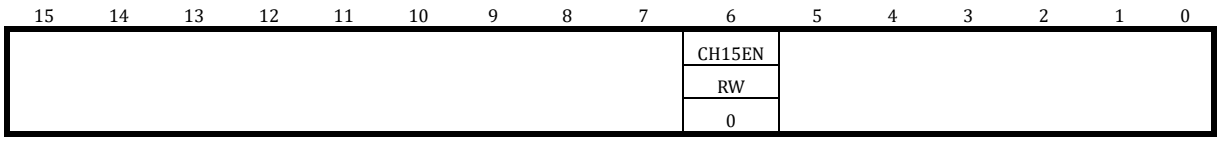
If SYS_CLK_CFG selects PLL clock, PLLPDN is hardware-controlled, and software configuration of PLLPDN to disable PLL is invalid. Disabling PLL requires PLLPDN=0, and SYS_CLK_CFG does not select PLL as the chip master clock. Both conditions must be satisfied. Similarly, if SYS_CLK_CFG selects HRC clock, then RCHPD is controlled by hardware. It is invalid to configure RCHPD to disable RCH directly by software. Turning off PLL requires RCHPD=1 and the chip goes to sleep. If the chip master clock is a PLL clock and the HRC is a PLL reference clock, then the RCH is also controlled by the hardware. Since RCH and PLL depend on BGP, BGPPD is also hardware-controlled. When RCH or PLL is used on a chip, it is invalid to configure BGPPD in software to disable BGP. To disable BGP, disable the PLL and RCH in sequence and the chip goes to sleep.

5.3.12 SYS_AFE_REG6 Analog Register 6

Address: 0x4000_0038

Reset value: 0x0

Table 5-17 SYS_AFE_REG6 Analog Register 6



Location	Bit name	Description
[31:7]		Unused
[6]	CH15EN	ADC CH15 channel signal enabling 0: ADC CH15 connects to internal GND 1: ADC CH15 connects to SWCK IO
[5:0]		Unused

5.3.13 SYS_AFE_REG7 Analog Register 7

Address: 0x4000_003C

Reset value: 0x0

Table 5-18 SYS_AFE_REG7 Analog Register 7

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

Location	Bit name	Description
[31:14]		Unused
[13:8]	SAMP_TIME	000000->111111: The corresponding sampling time increases successively from 4->67 ADC clock cycles
[7:6]		Unused
[5:4]	ADCLKSEL	ADC clock frequency selection 00: 48MHz 01: Prohibition 10:12MHz 11:24MHz
[3:0]		Unused

5.3.14 SYS_TMP_A Temperature Sensor Factor A Register

Address: 0x4000_0054

Reset value: 0x0

Table 5-19 SYS_TMP_A Temperature Sensor Factor A Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	TMP_GAIN_A	Temperature sensor gain correction factor A

5.3.15 SYS_TMP_B Temperature Sensor Factor B Register

Address: 0x4000_0058

Reset value: 0x0

Table 5-20 SYS_TMP_B Temperature Sensor Factor B Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMP_SIGN				TMP_OFFSET_B											
R				RW											
0				0											

Location	Bit name	Description
[31:16]		Unused
[15:12]	TMP_SIGN	Temperature sensor offset correction factor B symbol extension bit
[11:0]	TMP_OFFSET_B	Temperature sensor offset correction factor B

5.3.16 SYS_AFE_DAC DAC Digital Register

Address: 0x4000_0060

Reset value: 0x0

Table 5-21 SYS_AFE_DAC DAC Digital Register

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

Location	Bit name	Description
[31:12]		Unused
[11:0]	DAC_IN	Input of DAC digital quantity to be converted

5.3.17 SYS_AFE_DAC_AMC DAC Gain Correction Register

Address: 0x4000_0064

Reset value: 0x0200

Table 5-22 SYS_AFE_DAC_AMC DAC Gain Correction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						DAC_AMC									
						RW									
						0x200									

Location	Bit name	Description
[31:10]		Unused
[9:0]	DAC_AMC	DAC gain calibration value. It's a 10-bit unsigned fixed-point number, of which B [9] is an integer, and B [8: 0] is a decimal.

5.3.18 SYS_AFE_DAC_DC DAC DC Offset Register

Address: 0x4000_0068

Reset value: 0x0

Table 5-23 SYS_AFE_DAC_DC DAC DC Offset Register

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

Location	Bit name	Description
[31:8]		Unused
[7:0]	DAC_DC	DAC DC offset, 8bit signed number, B[7] is the sign bit

DAC gain calibration. The 12-bit DAC value of the analog output is DAC_raw, and the 12-bit DAC value after calibration is DAC_cali.

DAC_cali = DAC_raw * DAC_AMC – DAC_DC;

DAC_AMC is the DAC gain correction coefficient, which is a 10-bit unsigned fixed-point number, and B [9] is an integer, while B [8: 0] is a decimal about 1. For example, DAC_AMC = 10'b10_0001_0000 = 1+1/32, or

DAC_AMC = 10'b01_1110_1100 = 1-5/128

DAC_DC Bit DAC DC bias, which is a 8-bit signed integer.

After gain calibration , the calculation result is truncated and reserved, and the final DAC_cali is still a 12-bit integer.

Besides, the value after gain calibration and DC bias calibration will be saturated. The maximum value is 0xfff and the minimum value is 0x000.

Please note that the DAC has two output gears. After power-on, the DAC calibration value will be loaded automatically by default. If switch to another gear, please use the library function provided by the manufacturer.

5.3.19 SYS_CLK_CFG Clock Control Register

Address: 0x4000_0080

Reset value: 0x0

Table 5-24 SYS_CLK_CFG Clock Control Register

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

Location	Bit name	Description
[31:9]		Unused
[8]	CLK_SEL	The source selection signal of the system clock MCLK; CLK_HS is selected by default 0: CLK_HS 1: PLL Note that the PLL is turned off by default after power-on and requires being turned on by software.
[7:0]	CLK_DIV	PLL output frequency division control; the default is 0x00: 1/8 frequency division 0x01: 1/8 frequency division 0x11: 1/4 frequency division 0x55: 1/2 frequency division 0xFF: 1/1 frequency division Other configuration values are not recommended.

When CLK_SEL = 0, MCLK selects CLK_HS clock (4MHz). At this time, the frequency division factor of SYS_CLK_CFG[7:0] is invalid. The final output system clock frequency is 4MHz.

5.3.20 SYS_RST_CFG Reset Control Register

Address: 0x4000_0084

Reset value: 0x40

Table 5-25 SYS_RST_CFG Reset Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SWDMUX	RST_IO	WK_INTV				WDT_EN
									RW	RW	RW				RW
									1	0	0				0

Location	Bit name	Description
[31:7]		Unused
[6]	SWDMUX	SWD reuse control signal; the default configuration is SWD 0: P2.13 and P2.0 are used as normal GPIOs 1: P2.13 is reused as SWCLK, P2.0 is reused as SWDIO When used as SWD signal, the pull-up is turned on by default and is not controlled by software When used as GPIO, the pull-up is controlled by GPIO2_PUE[0] and GPIO2_PUE[13]
[5]	RST_IO	RSTn/P0.2 reuse control signal; the default configuration is RSTn 0: RSTn 1: P0.2 Note that the default is RSTn after power-on, the subsequent software can enable this bit, and the RSTn function is invalid. When used as RSTn, the pull-up is turned on by default and is not controlled by software When used as GPIO, the pull-up is controlled by GPIO0_PUE[2].
[4:2]	WK_INTV	Sleep Wake-up

		Interval Setting 100:4S 000: 0.25S 101:8S 001:0.5S 110:16S 010:1S 111:32S 011:2S
[1]		Unused
[0]	WDT_EN	Watchdog enabling control signal; the default is watchdog OFF 0: Watchdog OFF 1: Watchdog ON

For safety reasons, the IO of the SWD cannot be switched to the GPIO function within 16 ms after power-up, and can only be used as the SWD. Even if the software overwrites the SYS_IO_CFG[6], it will not work and the SYS_IO_CFG[6] needs to be overwritten again after 16 ms. The behavior is that software can write 1'b1 to the SYS_IO_CFG[6] within 16 ms, but the bit is still read back as 1'b0. If 1'b1 is written again after 16 ms, 1'b1 can be read back. That is, writing 1'b1 to the software SYS_IO_CFG[6] is invalid within 16 ms after power-on, and writing 1'b1 to the SYS_IO_CFG[6] by the software is valid 16 ms after power-on. This is to prevent the SWD function from being switched to GPIO when the application is powered on, so that the chip cannot erase and burn the flash through the SWD subsequently.

Note that, after P2.13 and P2.0 are switched to GPIO function, unless the application software is specially designed, the original downloader can only be used to erase the downloaded program.

5.3.21 SYS_RST_SRC Reset Source Record Register

Address: 0x4000_0088

Reset value: 0x0

Table 5-26 SYS_RST_SRC Reset Source Record Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												WDT_RST_RCD	KEY_RST_RCD	HPOR_RST_RCD	LPOR_RST_RCD
												RO	RO	RO	RO
												0	0	0	0

Location	Bit name	Description
[31:4]		Unused
[3]	WDT_RST_RCD	Watchdog reset occurred flag. Highly effective
[2]	KEY_RST_RCD	Key reset occurred flag. Highly effective
[1]	HPOR_RST_RCD	HPOR reset occurred flag. Highly effective
[0]	LPOR_RST_RCD	LPOR reset occurred flag. Highly effective

5.3.22 SYS_CLR_RST Reset Source Record Clear Register

Address: 0x4000_008C

Reset value: 0x0

Table 5-27 SYS_CLR_RST Reset Source Record Clear Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	PSW	Write 0xCA40 to clear the reset flag record Please note that since the reset recording works in the low-speed clock domain, it may take a certain time to finish clearing, and the state of the record should not be read immediately after clearing.

5.3.23 SYS_CLK_DIV0 Peripheral Clock Divider Register 0

Address: 0x4000_0090

Reset value: 0x0

Table 5-28 SYS_CLK_DIV0 Peripheral Clock Divider Register 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV0															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DIV0	I2C working clock=MCLK/(CLK_DIV0+1) MCLK is determined by the SYS_CLK_CFG division factor.

5.3.24 SYS_CLK_DIV2 Peripheral Clock Divider Register 2

Address: 0x4000_0098

Reset value: 0x0

Table 5-29 SYS_CLK_DIV2 Peripheral Clock Divider Register 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV2															
RW															
0															

Locatio	Bit	Description

n	name	
[31:16]		Unused
[15:0]	DIV2	UART working clock=MCLK/(CLK_DIV2+1) After sharing this frequency division configuration, the baud rate is further divided by the UART baud rate register, where MCLK is determined by the SYS_CLK_CFG division factor.

5.3.25 SYS_CLK_FEN Peripheral Clock-Gating Register

Address: 0x4000_009C

Reset value: 0x0

Table 5-30 SYS_CLK_FEN Peripheral Clock-Gating Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SPI_CLK_EN		DSP_CLK_EN	UART1_CLK_EN	UART0_CLK_EN	MCPWM_CLK_EN	UTIMER_CLK_EN	HALL_CLK_EN	I2C_CLK_EN
							RW		RW	RW	RW	RW	RW	RW	RW
							0		0	0	0	0	0	0	0

Location	Bit name	Description
[31:9]		Unused
[8]	SPI_CKL_EN	SPI module clock control signal; the default is SPI module clock OFF 1: SPI module clock ON 0: SPI module clock OFF
[7]	--	Unused
[6]	DSP_CLK_EN	Co-processor module clock control signal; the default is co-processor module clock OFF 1: Co-processor module clock control ON 0: Co-processor module clock control OFF
[5]	UART1_CLK_EN	UART1 module clock control signal; the default is UART1 module clock OFF 1: UART1 module clock ON 0: UART1 module clock OFF
[4]	UART0_CLK_EN	UART0 module clock control signal; the default is UART0 module clock OFF 1: UART0 module clock ON 0: UART0 module clock OFF
[3]	MCPWM_CLK_EN	MCPWM module clock control signal; the default is MCPWM module clock OFF 1: MCPWM module clock ON 0: MCPWM module clock control OFF
[2]	UTIMER_CLK_EN	UTIMER module clock control signal; the default is UTIMER module clock OFF 1: UTIMER module clock ON 0: UTIMER module clock control OFF
[1]	HALL_CLK_EN	HALL module clock control signal; the default is HALL module

		clock OFF 1: HALL module clock ON 0: HALL module clock OFF
[0]	I2C_CLK_EN	I2C module clock control signal; the default is I2C module clock OFF 1: I2C module clock control ON 0: I2C module clock control OFF

Note that the clock of each of the above modules is the working clock of the internal circuit of the respective module. Even if the clock of the respective module is not turned on, it will not affect the software access to the register of the respective module.

5.3.26 SYS_CLK_SLP Sleep Register

Address: 0x4000_00A0

Reset value: 0x0

Table 5-31 SYS_CLK_SLP Sleep Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	PSW	Write the password 0xDEAD, the system turns off the high-speed clock and enters the sleep state.

5.3.27 SYS_SFT_RST Soft Reset Register

Address: 0x4000_00AC

Reset value: 0x0

Table 5-32 SYS_SFT_RST Soft Reset Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						ADC_SFT_RST	SPI_SFT_RST		DSP_SFT_RST	UART1_SFT_RST	UART0_SFT_RST	MCPWM_SFT_RST	UIMTER_SFT_RST	HALL_SFT_RST	I2C_SFT_RST
						WO	WO		WO	WO	WO	WO	WO	WO	WO
						0	0		0	0	0	0	0	0	0

Location	Bit name	Description

[31:10]		Unused
[9]	ADC_SFT_RST	ADC module soft reset control signal; the default configuration is not resetting ADC module 1: Reset ADC module 0: Release ADC module
[8]	SPI_SFT_RST	SPI module soft reset control signal; the default configuration is not resetting SPI module 1: Reset SPI module 0: Release SPI module
[7]		Unused
[6]	DSP_SFT_RST	Co-processor module soft reset control signal; the default configuration is not resetting co-processor module 1: Reset the co-processor module 0: Release the co-processor module
[5]	UART1_SFT_RST	UART1 module soft reset control signal; the default configuration is not resetting UART1 module 1: Reset UART1 module 0: Release UART1 module
[4]	UART0_SFT_RST	UART0 module soft reset control signal; the default configuration is not resetting UART0 module 1: Reset UART0 module 0: Release UART0 module
[3]	MCPWM_SFT_RST	MCPWM module soft reset control signal; the default configuration is not resetting MCPWM module 1: Reset MCPWM module 0: Release MCPWM module
[2]	UTIMER_SFT_RST	UTIMER module soft reset control signal; the default configuration is not resetting UTIMER module 1: Reset UTIMER module 0: Release UTIMER module
[1]	HALL_SFT_RST	HALL module soft reset control signal; the default configuration is not resetting HALL module 1: Reset HALL module 0: Release HALL module
[0]	I2C_SFT_RST	I2C module soft reset control signal; the default configuration is not resetting I2C module 1: Reset I2C module 0: Release I2C module

Please note that the module soft reset state will remain in the reset state after writing 1 to the corresponding bit of SYS_SFT_RST, and write 0 again to release the reset state.

5.3.28 SYS_PROTECT Write Protection Register

Address: 0x4000_00B0

Reset value: 0x0

Table 5-33 SYS_PROTECT Write Protection Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	PSW	Except for watchdog registers such as SYS_AFE_DAC, SYS_AFE_DAC_AMC, SYS_AFE_DAC_DC, SYS_WDT_PSW SYS_WDT_CLR/SYS_WDT_TH/SYS_WDT_CNT, other system registers are protected by SYS_PROTECT. Write 0x7A83 to enable the register write operation. Write other values to prohibit write operations of registers.

6 FLASH

6.1 Introduction

The FLASH memory contains two parts: NVR and MAIN. NVR size is 1KB, and MAIN is 32KB.

The main flash memory area (MAIN) includes application programs and user data area.

The information storage area (info area/NVR) includes three parts:

- Option bytes: includes user options for hardware and storage protection (not occupying NVR space in the figure below)
- System memory: includes the boot loader code (not occupying the NVR space in the figure below)
- User data area: 1KB reserved for users

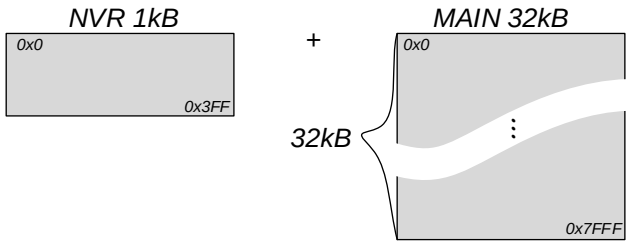


Fig. 6-1 Block Diagram of FLASH Memory Space Division

- Available for repeated erasure and writing for at least 20,000 times
- Data retention at room temperature for up to 100 years
- Single byte program: 7.5us(max), Sector erase: 5ms(max).
- Sector size is 512 bytes. The data can be erased and written according to Sector. It supports runtime programming.
- Flash data anti-theft (any value other than 0xFFFFFFFF must be written to the last word)

6.2 Features

The FLASH controller module mainly implements the related operations on the FLASH memory. These include:

- FLASH reading, including reading operations of the NVR and MAIN.

- FLASH writing, including writing operations to the NVR and MAIN.
- FLASH erase, including CHIP erase and SECTOR erase. SECTOR erase only available in NVR, and the MAIN part supports both CHIP erase and SECTOR erase.
- FLASH deep sleep, reducing the sleep power consumption of the chip.
- FLASH memory content encryption.
- FLASH reading acceleration, which improves the overall operation efficiency of the chip.
- FLASH control register access.

6.2.1 Functional Description

The control module implements operations such as reset/read/write/erase/sleep of the FLASH memory bank. The following is the state transition diagram of the control module:

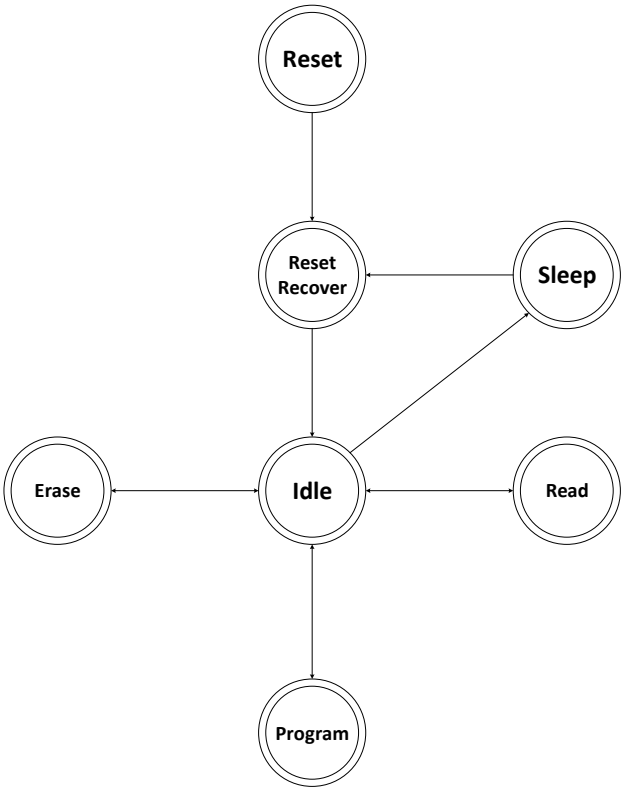


Fig. 6-2 FLASH Control State Transform Diagram

6.2.1.1 Reset

After the system is reset, it takes some times for the FLASH recovery, thus to ensure the internal circuit stability of the FLASH memory. After then, other operations could be performed on FLASH. This reset operation is automatically realized by hardware without software intervention.

6.2.1.2 Sleep

The sleep operation of FLASH is divided into two parts: Standby and Deep Sleep. When no operations will be performed on FLASH, FLASH enters the StandBy state automatically (if prefetching is turned on, this function is disabled). When the system executes Deep Sleep operation, it will trigger FLASH to enter Deep Sleep, to further reduce power consumption. The operation of FLASH entering Deep Sleep is completed by hardware automatically without software intervention.

When system is waked up by the outside world, the FLASH will be waked up at the same time. After a period of recovery, FLASH operations can be performed normally. This wake-up recovery operation is automatically completed by hardware without software intervention.

6.2.1.3 FLASH Read

The read operation is the basic operation of FLASH. The system can access the data in FLASH through two paths.

- The MCU fetches instructions and accesses data directly on the FLASH through the AHB bus. The fetch width is 32bit, and can only access data in the MAIN space. The hardware also provides an acceleration function to speed up the MCU to fetch instructions and access data.
- The MCU accesses the register of the controller through the AHB bus to read FLASH internal data indirectly. Both the data in the MAIN and NVR spaces can be accessed; if a continuous reading is required, the hardware will accumulate the addresses automatically without updating the address register value each time.

The FLASH_CFG.REGION bit indicates which space is currently being accessed. See as follows:

Table 6-1 FLASH Access Space Allocation

NVR (FLASH_CFG.REGION)	Zone of Access
0	Main zone
1	NVR zone

The process for reading the internal data indirectly on FLASH by accessing the register of the controller is as follows:

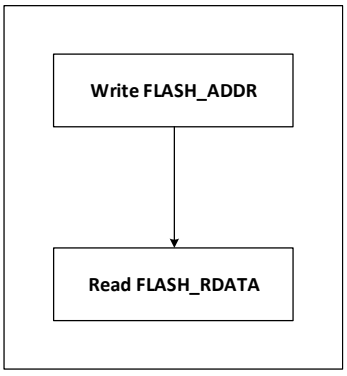


Fig. 6-3 Flow Chart of FLASH Indirect Read

6.2.1.4 FLASH Programming

FLASH Programming refers to programming operations on the FLASH memory bank. Generally, an erase operation should be performed before data programming. And, the programming can only be performed by accessing the registers of the FLASH controller. The specific process is:

- Control configuration register (CFG), enable programming operation
- Address register (ADDR), write programming address
- Write data register (WDATA), write programming data

The process for FLASH programming by accessing the register of this controller is as follows:

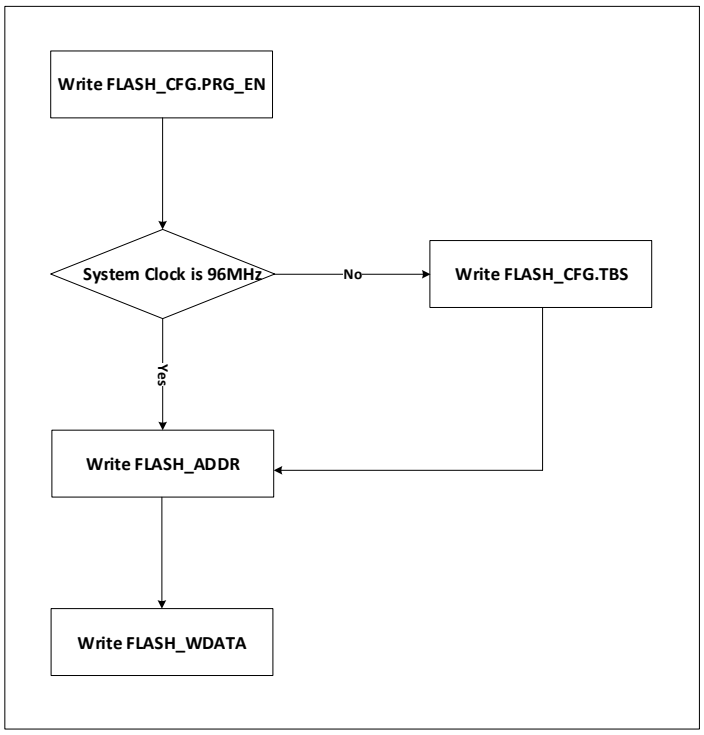


Fig. 6-4 Flow Chart of FLASH Programming

For the selection of operating frequency, please refer to the configuration of SYS_CLK_CFG. The absolute time of the FLASH write/erase operation is fixed, and the count value corresponding to these absolute times should be saved in the FLASH controller. The default value of FLASH_CFG.TBS is the count value at 96MHz clock frequency; When the chip works at other frequencies, the value of FLASH_CFG.TBS should be set to achieve the count values of 48MHz/24MHz and 12MHz (other frequencies are not supported). In this way, the value obtained by multiplying the count value by the clock frequency is equal to a constant time. For the corresponding FLASH_CFG.TBS values at different frequencies. Please note that only the values provided in the register description could be set to the FLASH_CFG.TBS, and other values are not available for writing into; otherwise, it may cause FLASH programming/erasing failure. It is recommended to read first on the FLASH_CFG, and then perform other operations by follow the OR/AND method. Besides, the MCU will stop temporarily when the

FLASH program/erase operation is performing until the operation is finished.

Figure 5-4 only shows the flow of one programming. If continuous programming is required, set the FLASH_CFG.ADR_INC before writing to the FLASH_ADDR register to enable the address auto-increment mode. After then, repeatedly write into the FLASH_WDATA register, and the address will be added by 0x4 automatically each time when writing data into the FLASH_ADDR. The operation of continuous reading is similar to this. The continuous programming process is as follows:

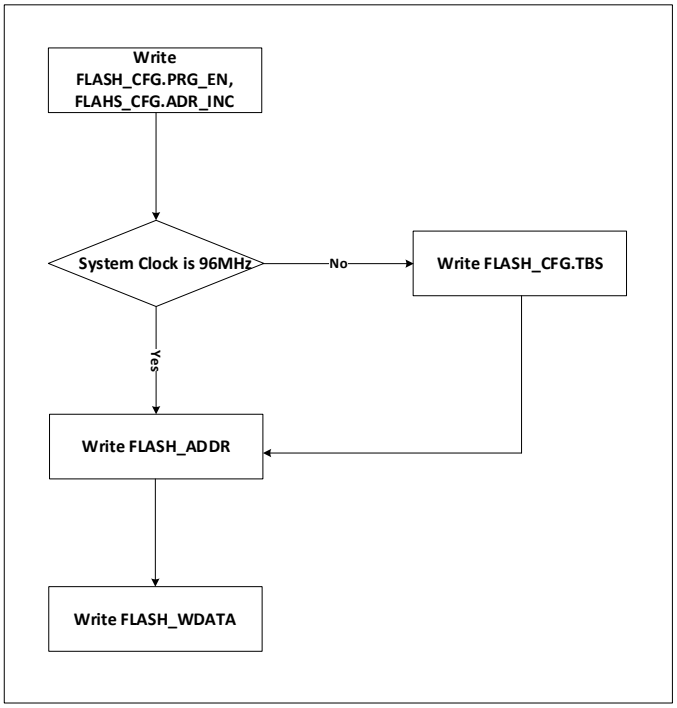


Fig. 6-5 Flow Chart of FLASH Programming

6.2.1.5 FLASH Erase

The erase operation is the basic operation of FLASH， which can only be achieved by accessing the registers of the FLASH controller. The specific process is:

- FLASH erase enable
- Address register (ADDR), write erase address
- Write erase register (ERASE), trigger erase

Perform flash erase on the FLASH memory bank. The erase operation is divided into Sector erasure and FullChip erasure, corresponding to 512Byte erasure and 32KB erasure respectively. Which type of erase operation is performed can be determined by setting the FLASH control register.

The following table is the address allocation space of Block and Sector.

Table 6-2 FLASH Sector Address Allocation

Name	Addresses	Size(Bytes)
Sector 0	0x0000 0000 - 0x0000 01FF	512
Sector1	0x0000 0200 - 0x0000 03FF	512
Sector2	0x0000 0400 - 0x0000 05FF	512
…	…	…
Sector63	0x0000 7E00 - 0x0000 7FFF	512

The NVR area can only realize Sector erasure, and the MAIN area can realize both Sector erasure and FULL erasure. See the following table:

NVR（CFG B[11]）	Sector Erase	FULL Erase
0	Main zone	Main zone
1	NVR zone	Main zone

The flash erase operation flow is shown below.

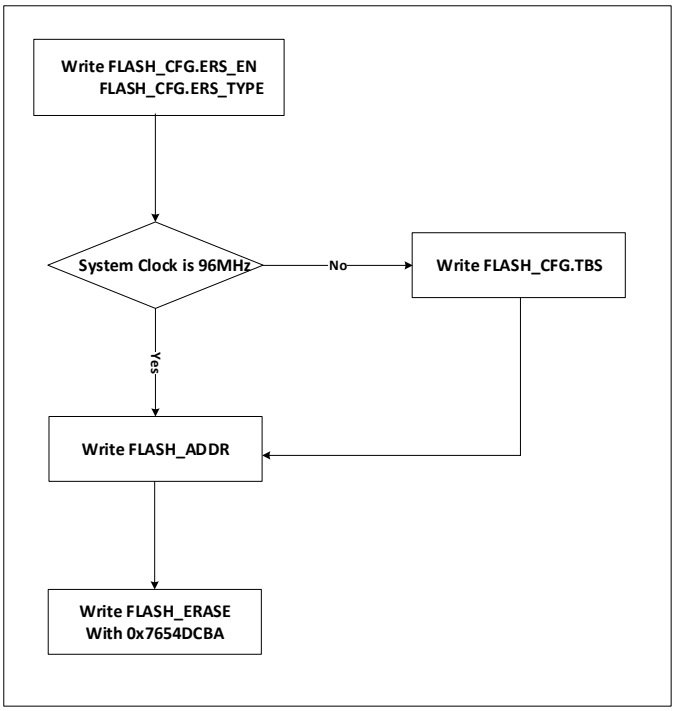


Fig. 6-6 Flow Chart of FLASH Erase

For Secotor erasure, determine which Secotor to be erased by FLASH_ADDR; For FullChip mode, the value of FLASH_ADDR will be invalid. Writing 0x7654DCBA to FLASH_ERASE, and triggers the erase operation.

6.2.1.6 FLASH Prefetch

Due to the speed limitation of FLASH memory, this operation cannot reach the speed of 96MHz. When reading the FLASH, it takes more than 1 clock cycle to finish reading the data. In order to speed up the reading of data, the FLASH controller adds a prefetch function. After the FLASH controller

finishes the current read operation, it will prefetch the data of the next WORD in turn without affecting the normal program execution. The prefetch operation can be turned on and off only by setting FLASH_CFG.PREF.

6.2.1.7 FLASH Encryption

If the data in the FLASH memory is encrypted, users can decrypt the data in the FLASH memory. On the contrary, if the data in the FLASH memory is decrypted, users can encrypt the data in the FLASH memory. The data in the FLASH memory is encrypted by default. After the chip is powered on and reset, the hardware will perform an encryption status update automatically. Whether the data is encrypted or decrypted, it will remain unchanged after being updated.

The total Flash memory is 32 KB, and the last WORD in the corresponding specification is designed as an encrypted word. When the content of this WORD is written as all "1", it indicates that FLASH is in decryption state; When the content of this WORD is written as not-all "1", it indicates that the FLASH is in an encrypted state. If encryption is required, perform the programming of the last WORD, write a non-all "1" value, and read the FLASH_PROTECT register to trigger an encryption status update to finish encryption (reading the return value of FLASH_PROTECT has no reference significance).

There are two cases for the corresponding decryption process. If the last WORD has not been programmed to write a non-all "1" value, reading the FLASH_PROTECT register will finish the decryption update (regardless of the current return value). If it has been programmed and written a non-all "1" value, decrypt by erasure operation. Firstly, perform an erase operation on FLASH, restore the last WORD to all "1" value, and then read the FLASH_PROTECT register to trigger an encryption status update and finish decryption (reading the return value of FLASH_PROTECT has no reference significance).

6.2.1.8 FLASH_IAP FLASH Online Upgrade (IAP) Register

The IAP mode is used to implement remapping of the interrupt vector table. The LKS32MC05X series chip contains the register VTOR, whose address is 0xE000_ED08, to remap the entry address of the interrupt vector table.

Table 6-3 Register Description of IAP VTOR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VTOR															
RW															
0															

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

Location	Bit name	Description
[31:7]	VTOR	Perform write operation to write entry address of interrupt vector table

The default value is 0x0, and the entry address of the interrupt vector table is 0x0. When a non-zero value is written, the entry address of the interrupt vector table will be mapped to the address corresponding to the written value and take effect immediately. When read back, it shifts right by 7 bits.

Since the LKS32MC05X series chip has a VTOR register, users can update the entire FLASH content in situations. Besides, the interrupt operation can be turned on or off during online upgrades.

6.2.1.8.1 Start Interrupted Online Upgrade

Recommended software configuration process:

Turn off the interrupt controller of the MCU to stop receiving the new interrupt responses temporarily;

Place the interrupt processing function code at the new interrupt entry address;

Write the new interrupt entry address to the VTOR register;

Turn on the interrupt controller of the MCU to enable interrupts;

Jump to the online upgrade function to start the online upgrade;

Turn off the interrupt controller of the MCU after upgraded and set VTOR as the default value of 0.

Perform a MCU soft reset, and the PC restarts the upgraded program from address 0.

6.2.1.8.2 End Interrupted Online Upgrade

Turn off the interrupt controller of the MCU to stop receiving the new interrupt responses temporarily;

Jump to the online upgrade function to start the online upgrade; If the online upgrade uses UART-like peripheral communication, the MCU should poll the UART interrupt flag bit.

Perform a MCU soft reset, and the PC restarts the upgraded program from address 0.

6.2.1.8.3 Location of Online Upgrade Function

If the flash should be erased, place the online upgrade function in RAM; if an interrupt is required, place the new interrupt vector entry address in the RAM address space.

If only part of the flash area occupied by the application should be erased, place the online upgrade function in the free area of the high flash address, and then use the block to erase the old flash application and write the new application.

6.3 Register

6.3.1 Address Allocation

The base address of the FLASH controller module register is 0x4000_0400.

Table 6-4 List of FLASH Controller Register

Name	Offset	Description
FLASH_CFG	0x00	FLASH configuration register
FLASH_ADDR	0x04	Address register
FLASH_WDATA	0x08	Write data register
FLASH_RDATA	0x0C	Read data register
FLASH_ERASE	0x10	Erase enable register
FLASH_PROTECT	0x14	FLASH protection status register
FLASH_READY	0x18	FLASH free and busy status register

6.3.2 FLASH_CFG Configuration Register (Read back first, and then modify by the OR/AND form)

Address: 0x4000_0400

Reset value: 0x60

Table 6-5 FLASH_CFG Configuration Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERS_EN				PRG_EN				ADR_INC				PREF			
RW				RW				RW				RW			
0				0				0				0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERS_TYPE				REGION					TBS						
RW				RW					RW						
0				0					60						

Location	Bit name	Description
[31]	ERS_EN	FLASH erasure enable. The default value is 0. 0: Erasure off 1: Erasure on
[27]	PRG_EN	FLASH programming enable. The default value is 0. 0: programming off 1: Programming on
[23]	ADR_INC	FLASH address increment enable. The default value is 0. 0: Address increment off 1: Address increment on When performing FLASH continuous read and write access, you can enable this function to reduce the operation of the address.
[19]	PREF	FLASH prefetch acceleration enable. The default value is 0. 0: Acceleration off

		1: Acceleration on
[15]	ERS_TYPE	FLASH erasure type selection. The default value is 0. 0:Sector 1:FULL
[11]	REGION	Access FLASH area selection. The default value is 0. 0:MAIN 1:NVR
[6:0]	TBS	Program/erase time base register. The default value is 0x60. Only the following values can be set: 0x60: FLASH programming/erasing time base value at 96Mhz operating frequency. 0x2F: FLASH programming/erasing time base value at 48Mhz operating frequency. 0x17: FLASH programming/erasing time base value at 24Mhz operating frequency. 0x0B: FLASH programming/erasing time base value at 12Mhz operating frequency.

6.3.3 FLASH_ADDR Address Register

Address: 0x4000_0404

Reset value: 0x0

Table 6-6 FLASH_ADDR Address Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	ADDR	Address register. Address register corresponding to read/write/erase operation. The lowest two digits will be ignored by the FLASH controller because of the WORD operation. When performing the erase operation, the addresses should be aligned according to the erase type. One Sector is 512-Byte. If performing the Sector erase, the address should be an integer multiple of 512 (if offset, the offset will be ignored). If performing a full chip erase, the value of this register is Unused for reference.

6.3.4 FLASH_WDATA Write Register

Address: 0x4000_0408

Reset value: 0x0

Table 6-7 FLASH_WDATA Write Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

WDATA
WO
0

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

Location	Bit name	Description
[31:0]	WDATA	Perform write operation to write the FLASH value

6.3.5 FLASH_RDATA Read Register

Address: 0x4000_040C

Reset value: 0x0

Table 6-8 FLASH_RDATA Read Register

RDATA
RO
0

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

Location	Bit name	Description
[31:0]	RDATA	Perform read operation to read the FLASH value

6.3.6 FLASH_ERASE Erase Control Register

Address: 0x4000_0410

Reset value: 0x0

Table 6-9 FLASH_ERASE Erase Control Register

ERASE
WO
0

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

0

Location	Bit name	Description
[31:0]	ERASE	Write 0x7654DCBA to trigger the erase operation

6.3.7 FLASH_PROTECT Encryption Status Register

Address: 0x4000_0414

Reset value: 0x0

Table 6-10 FLASH_PROTECT Encryption Status Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PROTECT															
RO															
0															

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

Location	Bit name	Description
[31:0]	PROTECT	Read this register to update the encryption/decryption status. Reading the return value has no reference significance.

6.3.8 FLASH_READY Working Status Register

Address: 0x4000_0418

Reset value: 0x0

Table 6-11 FLASH_READY Working Status Register

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

Location	Bit name	Description
[31:1]		Unused
[0]	READY	1: FLASH is idle 0: FLASH is busy

6.3.9 NVR calibration parameter address information

LKS32MC5x NVR area Instructions:

LKS32MC05x NVR area features:

NVR area, the size is 1KB; The size of one sector is 512 bytes. Therefore, there are two sectors in the NVR area.

The NVR area, which is only used for data storage, can only be accessed through FLASH registers;

The operations supported by the NVR area include reading, programming, and erasing;

Only sector data in the NVR region can be erased. If the entire chip is erased, data in the NVR region will not be erased.

In the NVR area, you are advised to disable the system interruption and perform the access operation. In particular, performing programming and erasing operations;

LKS provides library functions that contain the above operations:

The lks32MC05x Flash.c file contains programming and erase functions

Erase functions: void EraseSector(u32 adr, u16 nvr);

Programming function: int ProgramPage(u32 adr, u32 sz, u8 *buf, u16 nvr);

The lks32mc05x VR. o file contains the Read function:

Read function: uint32 t Read NVR(uint32 t adr);

LKS32MC05x product memory chip calibration parameters:

Calibration parameters, each product is independently calibrated, each product does not support the mixing of calibration parameters;

Calibration parameters are written before the factory, after the factory does not support programming and erasing, only support reading;

Calibration parameters are accessed via library functions provided by LKS;

Calibrate parameters. It is recommended to turn off the system interrupt before performing the access operation.

The lks32mc05x VR. o file contains the Read function of calibration parameters:

Read function: uint32 t Read Trim(uint32 t adr);

Table 6-12 Calibration Parameter List

地址	内容
0x00000310	ADC0 DC0 Calibration
0x00000314	ADC0 DC0 Calibration
0x00000318	ADC0_AMC0 Calibration
0x0000031C	ADC0_AMC1 Calibration
0x00000330	DAC select 1.20V gear, SYS AFE DAC AMC calibration value (512x expanded results)
0x00000334	DAC select 4.80V gear, SYS AFE DAC AMC calibration value (512x expanded results)
0x00000340	DAC set 1.20V, SYS AFE DAC DC calibration value
0x00000344	DAC set 4.80V, SYS AFE DAC DC calibration value
0x00000348	DAC set 1.20V, DAC secondary calibration value
0x0000034C	DAC set 4.80V, DAC secondary calibration value
0x00000350	OPA0, 200K ohms VS 10.6K ohms, GAIN calibration (1000-fold amplification results), R0 is 0 ohms
0x00000354	OPA0, 190K ohms VS 20.6K ohms, GAIN calibration (1000-fold amplification results), R0 is 0 ohms
0x00000358	OPA0, 180K ohms VS 30.6K ohms, GAIN calibration (1000-fold amplification results), R0 is 0 ohms
0x0000035C	OPA0, 170K ohms VS 40.6K ohms, GAIN calibration (1000-fold amplification results), R0 is 0 ohms
0x00000360	OPA1, 200K ohms VS 10.6K ohms, GAIN calibration (1000-fold amplification results), R0 is 0 ohms
0x00000364	OPA1, 190K ohms VS 20.6K ohms, GAIN calibration (1000-fold amplification results), R0 is 0 ohms
0x00000368	OPA1, 180K ohms VS 30.6K ohms, GAIN calibration (1000-fold amplification results), R0 is 0 ohms
0x0000036C	OPA1, 170K ohms VS 40.6K ohms, GAIN calibration (1000-fold amplification results), R0 is 0 ohms
0x000002D0	OPA0, 200K ohm VS 10.6K Ohm, high 16-bit store R2 actual resistance value (magnify 100x store) Low 16-bit store R1 resistance value (magnify 100x store)
0x000002D4	OPA0, 190K ohm VS 20.6K Ohm, high 16-bit store R2 actual resistance value (magnify 100x store) Low 16-bit store R1 resistance value (magnify 100x store)
0x000002D8	OPA0, 180K ohm VS 30.6K Ohm, high 16-bit store R2 actual resistance value (magnify 100x store) Low 16-bit store R1 resistance value (magnify 100x store)
0x000002DC	OPA0, 170K ohm VS 40.6K Ohm, high 16-bit store R2 actual resistance value (magnify 100x store) Low 16-bit store R1 resistance value (magnify 100x store)
0x000002E0	OPA1, 200K ohm VS 10.6K Ohm, high 16-bit store R2 actual resistance value (magnify 100x store) Low 16-bit store R1 resistance value (magnify 100x store)
0x000002E4	OPA1, 190K ohm VS 20.6K Ohm, high 16-bit store R2 actual resistance value (magnify 100x store) Low 16-bit store R1 resistance value (magnify 100x store)
0x000002E8	OPA1, 180K ohm VS 30.6K Ohm, high 16-bit store R2 actual resistance value (magnify 100x store) Low 16-bit store R1 resistance value (magnify 100x store)
0x000002EC	OPA1, 170K ohm VS 40.6K Ohm, high 16-bit store R2 actual resistance value (magnify 100x store) Low 16-bit store R1 resistance value (magnify 100x store)
0x000002F0	Offset values under OPA0 configuration for four gain levels (unit: V, scaled by 10,000) Bit[31:16]: Offset value for 4x gain, Level 3 Bit[15:0]: Offset value for 8x gain, Level 2
0x000002F4	Offset values under OPA0 configuration for four gain levels (unit: V, scaled by 10,000) Bit[31:16]: Offset value for 16x gain, Level1

	Bit[15:0]: Offset value for 32x gain, Level 0
0x000002F8	Offset values under OPA1 configuration for four gain levels (unit: V, scaled by 10,000) Bit[31:16]: Offset value for 4x gain, Level 3 Bit[15:0]: Offset value for 8x gain, Level 2
0x000002FC	Offset values under OPA1 configuration for four gain levels (unit: V, scaled by 10,000) Bit[31:16]: Offset value for 16x gain, Level1 Bit[15:0]: Offset value for 32x gain, Level 0
0x00000398	Temperature sensor, slope calibration value
0x0000039C	Temperature sensor, bias calibration value
0x000003B0	OPA1 common-mode voltage for high 16-bit storage, OPA0 common-mode voltage for low 16-bit storage (10000 times enlarged results)
0x000003B4	OPA3 common-mode voltage for high 16-bit storage, OPA2 common-mode voltage for low 16-bit storage (10000 times enlarged results)
0x000002F0	OPA0 Offset Value (Signed Type, Unit: V, Amplified 10000x) Bit[31:16] 4x Amplification Gain, 3rd Gear Bit[15:0] 8x Amplification Gain, 2nd Gear
0x000002F4	OPA0 Offset Value (Signed Type, Unit: V, Amplified 10000x) Bit[31:16] 16x gain 1st Gear Bit[15:0] 32x gain 0th Gear
0x000002F8	Offset value at OPA1 stage (signed type, unit: V, amplified 10,000 times) Bit[31:16] 4x gain Bit[15:0] 8x gain
0x000002FC	Offset value at OPA1 position (signed type, unit: V, amplified 10000x) Bit[31:16] 16x gain Bit[15:0] 32x gain

7 GPIO

7.1 Introduction

LSK32MC05X series chips integrate a total of 3 groups of 16-bit GPIO. Four GPIOs: P0.0, P0.1, P1.0, and P1.1 can be used as system wake-up sources. Sixteen GPIOs from P0.15 to P0.0 can be used as external interrupt source input.

7.1.1 Functional Block Diagram

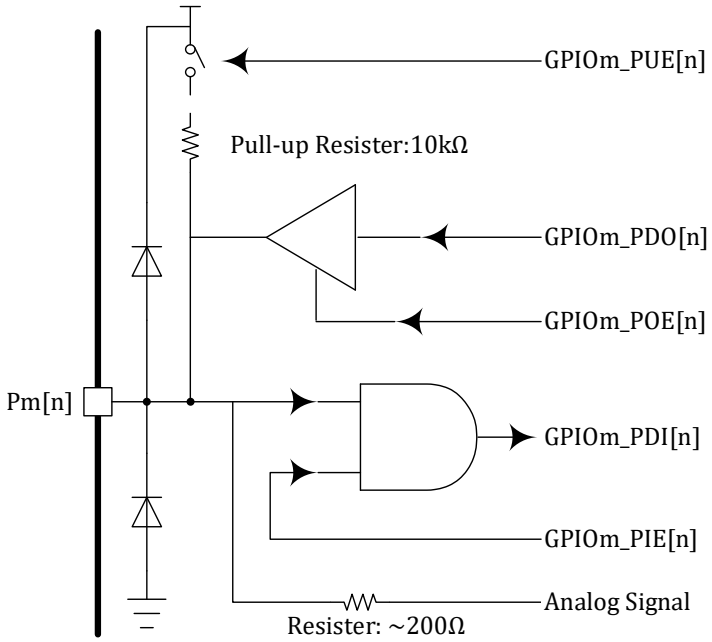


Fig. 7-1 GPIO Functional Block Diagram

As shown in the figure above, Pm [n] is the chip PAD, m can be 0 ~ 2, which means any group of three groups of GPIO, n can be 0 ~ 15, which means one IO in a group of 16-bit GPIO. The analog signal is directly connected to the PAD through a resistor in series,the resistance value is 100 to 200Ω. The digital signal is output through a three-state gate. When the output enables GPIOm_POE [n]=0, the buffer outputs a high-impedance state. Otherwise, the buffer output is at the same level as GPIOm_PDO [n]. Digital signal input enters the chip through an AND gate. When GPIOm_PIE [n]=0, GPIOm_PDI [n] is always 0; When GPIOm_PIE [n]=1, that is, the input enable is turned on, the level of GPIOm_PDI [n] is at the same level as Pm [n]. The chip PAD can be configured with pull-ups. The P0 [2] pin is reused as the external reset pin RSTN. The pull-up resistor is 100kΩ and the remaining pull-up resistors are 10kΩ. Please note that not all PADs are equipped with pull-up resistors. The PAD without a pull-up resistor can also be set by the GPIOm_PUE [n] register, but it has no practical effect.

7.1.2 Features

- Three groups of 16 bit GPIO
- Support Open drain
- Some IOs has internal pull-up resistor
- Support configuration lock protection
- Support external interrupt
- Support GPIO wake-up

7.2 Register

7.2.1 Address Allocation

The base address of the GPIO 0 module in the chip is 0x40012000.

The base address of the GPIO 1 module in the chip is 0x40012040.

The base address of the GPIO 2 module in the chip is 0x40012080.

Except for the base address, the register definitions of GPIO 0, 1 and 2 are the same.

Table 7-1 GPIOx Register List

Name	Offset address	Description
GPIOx_PIE	0x00	GPIO x input enable
GPIOx_POE	0x04	GPIO x output enable
GPIOx_PDI	0x08	GPIO x input data
GPIOx_PDO	0x0C	GPIO x output data
GPIOx_PUE	0x10	GPIO x pull-up enable
GPIOx_PODE	0x18	GPIO x open-drain enable
GPIOx_F3210	0x20	GPIO x [3: 0] function selection
GPIOx_F7654	0x24	GPIO x [7: 4] function selection
GPIOx_FBA98	0x28	GPIO x [11: 8] function selection
GPIOx_FFEDC	0x2C	GPIO x [15:12] function selection
GPIOx_BSRR	0x30	GPIO x bit operation register
GPIOx_BRR	0x34	GPIO x bit clear register

The base address of the GPIO interrupt/wake-up module is 0x40012100.

Table 7-2 Register List of GPIO Interrupt/Wake-up Module

Name	Offset address	Description
EXTI_CR0	0x00	GPIO 0 [7: 0] interrupt trigger type
EXTI_CR1	0x04	GPIO 0 [15:8] interrupt trigger type
EXTI_IF	0x08	GPIO interrupt flag
WAKE_POL	0x10	GPIO wake signal polarity
WAKE_EN	0x14	GPIO wake-up enable

7.2.2 GPIOx_PIE (where x = 0,1,2)

The addresses are: 0x4001_2000, 0x4001_2040 and 0x4001_2080.

Reset value: 0x0

Table 7-3 GPIOx_PIE GPIOx Input Enable Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIE15	PIE14	PIE13	PIE12	PIE11	PIE10	PIE9	PIE8	PIE7	PIE6	PIE5	PIE4	PIE3	PIE2	PIE1	PIE0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15]	PIE15	GPIO x[15] / Px[15] input enable
[14]	PIE14	GPIO x[14] / Px[14] input enable
[13]	PIE13	GPIO x[13] / Px[13] input enable
[12]	PIE12	GPIO x[12] / Px[12] input enable
[11]	PIE11	GPIO x[11] / Px[11] input enable
[10]	PIE10	GPIO x[10] / Px[10] input enable
[9]	PIE9	GPIO x[9] / Px[9] input enable
[8]	PIE8	GPIO x[8] / Px[8] input enable
[7]	PIE7	GPIO x[7] / Px[7] input enable
[6]	PIE6	GPIO x[6] / Px[6] input enable
[5]	PIE5	GPIO x[5] / Px[5] input enable
[4]	PIE4	GPIO x[4] / Px[4] input enable
[3]	PIE3	GPIO x[3] / Px[3] input enable
[2]	PIE2	GPIO x[2] / Px[2] input enable
[1]	PIE1	GPIO x[1] / Px[1] input enable
[0]	PIE0	GPIO x[0] / Px[0] input enable

7.2.3 GPIOx_POE (where x = 0,1,2)

The addresses are: 0x4001_2004, 0x4001_2044 and 0x4001_2084.

Reset value: 0x0

Table 7-4 GPIOx_POE GPIOx Output Enable Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POE15	POE14	POE13	POE12	POE11	POE10	POE9	POE8	POE7	POE6	POE5	POE4	POE3	POE2	POE1	POE0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15]	POE15	GPIO x[15] / Px[15] output enable

[14]	POE14	GPIO x[14] / Px[14] output enable
[13]	POE13	GPIO x[13] / Px[13] output enable
[12]	POE12	GPIO x[12] / Px[12] output enable
[11]	POE11	GPIO x[11] / Px[11] output enable
[10]	POE10	GPIO x[10] / Px[10] output enable
[9]	POE9	GPIO x[9] / Px[9] output enable
[8]	POE8	GPIO x[8] / Px[8] output enable
[7]	POE7	GPIO x[7] / Px[7] output enable
[6]	POE6	GPIO x[6] / Px[6] output enable
[5]	POE5	GPIO x[5] / Px[5] output enable
[4]	POE4	GPIO x[4] / Px[4] output enable
[3]	POE3	GPIO x[3] / Px[3] output enable
[2]	POE2	GPIO x[2] / Px[2] output enable
[1]	POE1	GPIO x[1] / Px[1] output enable
[0]	POE0	GPIO x[0] / Px[0] output enable

7.2.4 GPIOx_PDI (where x = 0,1,2)

The addresses are: 0x4001_2008, 0x4001_2048 and 0x4001_2088

Reset value: 0x0

Table 7-5 GPIOx_PDI GPIOx Data Input Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	PDI	GPIO x input data

When GPIOx PIE=0, GPIOx PDI read-back is 0.

7.2.5 GPIOx_PDO (where x = 0,1,2)

The addresses are: 0x4001_200C, 0x4001_204C and 0x4001_208C

Reset value: 0x0

Table 7-6 GPIOx_PDO GPIOx Data Output Register

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

Location	Bit name	Description
[31:16]		Unused

[15:0]	PDO	GPIO x output data

7.2.6 GPIOx_PUE (where x = 0,1,2)

The addresses are: 0x4001_2010, 0x4001_2050 and 0x4001_2090

Reset value: 0x0

Table 7-7 GPIOx_PUE GPIOx Pull-up Enable Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUE15	PUE14	PUE13	PUE12	PUE11	PUE10	PUE9	PUE8	PUE7	PUE6	PUE5	PUE4	PUE3	PUE2	PUE1	PUE0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15]	PUE15	GPIO x[15]/Px[15] pull-up enable
[14]	PUE14	GPIO x[14]/Px[14] pull-up enable
[13]	PUE13	GPIO x[13]/Px[13] pull-up enable
[12]	PUE12	GPIO x[12]/Px[12] pull-up enable
[11]	PUE11	GPIO x[11]/Px[11] pull-up enable
[10]	PUE10	GPIO x[10]/Px[10] pull-up enable
[9]	PUE9	GPIO x[9]/Px[9] pull-up enable
[8]	PUE8	GPIO x[8]/Px[8] pull-up enable
[7]	PUE7	GPIO x[7]/Px[7] pull-up enable
[6]	PUE6	GPIO x[6]/Px[6] pull-up enable
[5]	PUE5	GPIO x[5]/Px[5] pull-up enable
[4]	PUE4	GPIO x[4]/Px[4] pull-up enable
[3]	PUE3	GPIO x[3]/Px[3] pull-up enable
[2]	PUE2	GPIO x[2]/Px[2] pull-up enable
[1]	PUE1	GPIO x[1]/Px[1] pull-up enable
[0]	PUE0	GPIO x[0]/Px[0] pull-up enable

Note: Not all IOs have pull-up function. The PUE register corresponding to the IO without the pull-up function is also not implemented, so writing 1 to these positions is invalid, and the read-back is always 0.

7.2.7 GPIOx_PODE (where x = 0,1,2)

The addresses are: 0x4001_2018, 0x4001_2058, 0x4001_2098

Reset value: 0x0

Table 7-8 GPIOx_PODE GPIOx Open-drain Enable Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PODE15	PODE14	PODE13	PODE12	PODE11	PODE10	PODE9	PODE8	PODE7	PODE6	PODE5	PODE4	PODE3	PODE2	PODE1	PODE0

RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15]	PODE15	GPIO x[15]/Px[15] open-drain enable
[14]	PODE14	GPIO x[14]/Px[14] open-drain enable
[13]	PODE13	GPIO x[13]/Px[13] open-drain enable
[12]	PODE12	GPIO x[12]/Px[12] open-drain enable
[11]	PODE11	GPIO x[11]/Px[11] open-drain enable
[10]	PODE10	GPIO x[10]/Px[10] open-drain enable
[9]	PODE9	GPIO x[9]/Px[9] open-drain enable
[8]	PODE8	GPIO x[8]/Px[8] open-drain enable
[7]	PODE7	GPIO x[7]/Px[7] open-drain enable
[6]	PODE6	GPIO x[6]/Px[6] open-drain enable
[5]	PODE5	GPIO x[5]/Px[5] open-drain enable
[4]	PODE4	GPIO x[4]/Px[4] open-drain enable
[3]	PODE3	GPIO x[3]/Px[3] open-drain enable
[2]	PODE2	GPIO x[2]/Px[2] open-drain enable
[1]	PODE1	GPIO x[1]/Px[1] open-drain enable
[0]	PODE0	GPIO x[0]/Px[0] open-drain enable

7.2.8 GPIOx_PFLT (where x = 0,1,2)

The addresses are: 0x4001_201C, 0x4001_205C, 0x4001_209C

Reset value: 0x0

Table 7-9 GPIOx_PFLT GPIOx Filter Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PFLT 15	PFLT 14	PFLT 13	PFLT 12	PFLT 11	PFLT 10	PFLT 9	PFLT 8	PFLT 7	PFLT 6	PFLT 5	PFLT 4	PFLT 3	PFLT 2	PFLT1	PFLT0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15]	PFLT15	GPIO x[15]/Px[15] configuration filtering
[14]	PFLT14	GPIO x[14]/Px[14] configuration filtering
[13]	PFLT13	GPIO x[13]/Px[13] configuration filtering
[12]	PFLT12	GPIO x[12]/Px[12] configuration filtering
[11]	PFLT11	GPIO x[11]/Px[11] configuration filtering
[10]	PFLT10	GPIO x[10]/Px[10] configuration filtering
[9]	PFLT9	GPIO x[9]/Px[9] configuration filtering

[8]	PFLT8	GPIO x[8]/Px[8] configuration filtering
[7]	PFLT7	GPIO x[7]/Px[7] configuration filtering
[6]	PFLT6	GPIO x[6]/Px[6] configuration filtering
[5]	PFLT 5	GPIO x[5]/Px[5] configuration filtering
[4]	PFLT 4	GPIO x[4]/Px[4] configuration filtering
[3]	PFLT 3	GPIO x[3]/Px[3] configuration filtering
[2]	PFLT 2	GPIO x[2]/Px[2] configuration filtering
[1]	PFLT1	GPIO x[1]/Px[1] configuration filtering
[0]	PFLT0	GPIO x[0]/Px[0] configuration filtering

7.2.9 GPIOx_F3210 (where x = 0,1,2)

The addresses are: 0x4001_2020, 0x4001_2060, 0x4001_20A0

Reset value: 0x0

Table 7-10 GPIOx_F3210 GPIOx Function Selection Register

15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
F3	F2	F1	F0
RW	RW	RW	RW
0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15:12]	F3	GPIO x[3]/Px[3] function selection
[11:8]	F2	GPIO x[2]/Px[2] function selection
[7:4]	F1	GPIO x[1]/Px[1] function selection
[3:0]	F0	GPIO x[0]/Px[0] function selection

Table 7-11 GPIO Reuse Function

GPIOx_Fxxxx Set value	Second Function Code	Description
0x0	AF0	Analog function
0x1	AF1	SYS_AF, digital signal output function such as comparator and clock
0x2	AF2	HALL
0x3	AF3	MCPWM
0x4	AF4	UART
0x5	AF5	SPI
0x6	AF6	IIC
0x7	AF7	Timer0/Time1
0x8	AF8	Timer2/Timer3
0x9	AF9	ADC trigger debug

7.2.10 GPIOx_F7654 (where x = 0,1,2)

The addresses are: 0x4001_2024, 0x4001_2064, 0x4001_20A4

Reset value: 0x0

Table 7-12 GPIOx_F7654 GPIOx Function Selection Register

15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
F7	F6	F5	F4
RW	RW	RW	RW
0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15:12]	F7	GPIO x[7] / Px[7] function selection
[11:8]	F6	GPIO x[6] / Px[6] function selection
[7:4]	F5	GPIO x[5] / Px[5] function selection
[3:0]	F4	GPIO x[4] / Px[4] function selection

7.2.11 GPIOx_FBA98 (where x = 0,1,2)

The addresses are: 0x4001_2028, 0x4001_2068, 0x4001_20A8

Reset value: 0x0

Table 7-13 GPIOx_FBA98 GPIOx Function Selection Register

15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
F11	F10	F9	F8
RW	RW	RW	RW
0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15:12]	F11	GPIO x[11] / Px[11] function selection
[11:8]	F10	GPIO x[10] / Px[10] function selection
[7:4]	F9	GPIO x[9] / Px[9] function selection
[3:0]	F8	GPIO x[8] / Px[8] function selection

7.2.12 GPIOx_FFEDC (where x = 0,1,2)

The addresses are: 0x4001_202C, 0x4001_206C, 0x4001_20AC

Reset value: 0x0

Table 7-14 GPIOx_FFEDC GPIOx Function Selection Register

15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
F15	F14	F13	F12
RW	RW	RW	RW

0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15:12]	F15	GPIO x[15] / Px[15] function selection
[11:8]	F14	GPIO x[14] / Px[14] function selection
[7:4]	F13	GPIO x[13] / Px[13] function selection
[3:0]	F12	GPIO x[12] / Px[12] function selection

For a detailed list of GPIO function reuse, please refer to the corresponding `device` pin location in DATASHEET.

7.2.13 GPIOx_BSRR (where x = 0,1,2)

These addresses are: 0x4001_2030, 0x4001_2070, 0x4001_20B0

Reset value: 0x0

Table 7-15 GPIOx_BSRR GPIOx Bit Operation Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLR15	CLR14	CLR13	CLR12	CLR11	CLR10	CLR9	CLR8	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SET15	SET14	SET13	SET12	SET11	SET10	SET9	SET8	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31]	CLR15	Write 1 to clear GPIO x[15], and writing 0 is not effective
[30]	CLR14	Write 1 to clear GPIO x[14], and writing 0 is not effective
[29]	CLR13	Write 1 to clear GPIO x[13], and writing 0 is not effective
[28]	CLR12	Write 1 to clear GPIO x[12], and writing 0 is not effective
[27]	CLR11	Write 1 to clear GPIO x[11], and writing 0 is not effective
[26]	CLR10	Write 1 to clear GPIO x[10], and writing 0 is not effective
[25]	CLR9	Write 1 to clear GPIO x[9], and writing 0 is not effective
[24]	CLR8	Write 1 to clear GPIO x[8], and writing 0 is not effective
[23]	CLR7	Write 1 to clear GPIO x[7], and writing 0 is not effective
[22]	CLR6	Write 1 to clear GPIO x[6], and writing 0 is not effective
[21]	CLR5	Write 1 to clear GPIO x[5], and writing 0 is not effective
[20]	CLR4	Write 1 to clear GPIO x[4], and writing 0 is not effective

[19]	CLR3	Write 1 to clear GPIO x[3], and writing 0 is not effective
[18]	CLR2	Write 1 to clear GPIO x[2], and writing 0 is not effective
[17]	CLR1	Write 1 to clear GPIO x[1], and writing 0 is not effective
[16]	CLR0	Write 1 to clear GPIO x[0], and writing 0 is not effective
[15]	SET15	Write 1 to set GPIO x[15] to 1, and writing 0 is not effective
[14]	SET14	Write 1 to set GPIO x[14] to 1, and writing 0 is not effective
[13]	SET13	Write 1 to set GPIO x[13] to 1, and writing 0 is not effective
[12]	SET12	Write 1 to set GPIO x[12] to 1, and writing 0 is not effective
[11]	SET11	Write 1 to set GPIO x[11] to 1, and writing 0 is not effective
[10]	SET10	Write 1 to set GPIO x[10] to 1, and writing 0 is not effective
[9]	SET9	Write 1 to set GPIO x[9] to 1, and writing 0 is not effective
[8]	SET8	Write 1 to set GPIO x[8] to 1, and writing 0 is not effective
[7]	SET7	Write 1 to set GPIO x[7] to 1, and writing 0 is not effective
[6]	SET6	Write 1 to set GPIO x[6] to 1, and writing 0 is not effective
[5]	SET5	Write 1 to set GPIO x[5] to 1, and writing 0 is not effective
[4]	SET4	Write 1 to set GPIO x[4] to 1, and writing 0 is not effective
[3]	SET3	Write 1 to set GPIO x[3] to 1, and writing 0 is not effective
[2]	SET2	Write 1 to set GPIO x[2] to 1, and writing 0 is not effective
[1]	SET1	Write 1 to set GPIO x[1] to 1, and writing 0 is not effective
[0]	SET0	Write 1 to set GPIO x[0] to 1, and writing 0 is not effective

If the upper 16 bits and lower 16 bits of BSRR are used to set and clear the same position of GPIO at the same time, the bit will be cleared.

7.2.14 GPIOx_BRR (where x = 0,1,2)

The addresses are: 0x4001_2034, 0x4001_2074, 0x4001_20B4

Reset value: 0x0

Table 7-16 GPIOx_BRR GPIOx Bit Clear Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLR15	CLR14	CLR13	CLR12	CLR11	CLR10	CLR9	CLR8	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15]	CLR15	Write 1 to clear GPIO x[15]; writing 0 is not effective
[14]	CLR14	Write 1 to clear GPIO x[14]; writing 0 is not effective
[13]	CLR13	Write 1 to clear GPIO x[13]; writing 0 is not effective
[12]	CLR12	Write 1 to clear GPIO x[12]; writing 0 is not effective

[11]	CLR11	Write 1 to clear GPIO x[11]; writing 0 is not effective
[10]	CLR10	Write 1 to clear GPIO x[10]; writing 0 is not effective
[9]	CLR9	Write 1 to clear GPIO x[9]; writing 0 is not effective
[8]	CLR8	Write 1 to clear GPIO x[8]; writing 0 is not effective
[7]	CLR7	Write 1 to clear GPIO x[7]; writing 0 is not effective
[6]	CLR6	Write 1 to clear GPIO x[6]; writing 0 is not effective
[5]	CLR5	Write 1 to clear GPIO x[5]; writing 0 is not effective
[4]	CLR4	Write 1 to clear GPIO x[4]; writing 0 is not effective
[3]	CLR3	Write 1 to clear GPIO x[3]; writing 0 is not effective
[2]	CLR2	Write 1 to clear GPIO x[2]; writing 0 is not effective
[1]	CLR1	Write 1 to clear GPIO x[1]; writing 0 is not effective
[0]	CLR0	Write 1 to clear GPIO x[0]; writing 0 is not effective

7.2.15 External Interrupt, Wake-up

7.2.15.1 EXTI_CR0

Address: 0x4001_2100

Reset value: 0x0

Table 7-17 EXTI_CR0 External Interrupt Configuration Register

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
T7	T6	T5	T4	T3	T2	T1	T0
RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15:14]	T7	GPIO 0[7]/ P0[7] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[13:12]	T6	GPIO 0 [6]/P0 [6] External interrupt trigger type selection 000: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[11:10]	T5	GPIO 0[5]/ P0[5] External interrupt trigger type selection 000: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[9:8]	T4	GPIO 0 [4]/P0 [4] External interrupt trigger type selection 00: not triggered 01: falling edge trigger

		10: rising edge trigger 11: trigger on both rising and falling edges
[7:6]	T3	GPIO 0[3]/ P0[3] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[5:4]	T2	GPIO 0 [2]/P0 [2] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[3:2]	T1	GPIO 0 [1]/P0 [1] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[1:0]	T0	GPIO 0 [0]/P0 [0] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges

7.2.15.2 EXTI_CR1

Address: 0x4001_2104

Reset value: 0x0

Table 7-18 EXTI_CR1 External Interrupt Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T15		T14		T13		T12		T11		T10		T9		T8	
RW		RW		RW		RW		RW		RW		RW		RW	
0		0		0		0		0		0		0		0	

Location	Bit name	Description
[31:16]		Unused
[15:14]	T15	GPIO 0 [15]/P0 [15] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[13:12]	T14	GPIO 0 [14]/P0 [14] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[11:10]	T13	GPIO 0 [13]/P0 [13] External interrupt trigger type selection 000: not triggered 01: falling edge trigger

		10: rising edge trigger 11: trigger on both rising and falling edges
[9:8]	T12	GPIO 0 [12]/P0 [12] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[7:6]	T11	GPIO 0 [11]/P0 [11] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[5:4]	T10	GPIO 0 [10]/P0 [10] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[3:2]	T9	GPIO 0 [9]/P0 [9] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges
[1:0]	T8	GPIO 0 [8]/P0 [8] External interrupt trigger type selection 00: not triggered 01: falling edge trigger 10: rising edge trigger 11: trigger on both rising and falling edges

7.2.15.3 EXTI_IF

Address: 0x4001_2108

Reset value: 0x0

Table 7-19 EXTI_IF Ex External Interrupt Flag Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF15	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7	IF6	IF5	IF4	IF3	IF2	IF1	IF0
RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15]	IF15	GPIO 0[15]/P0[15] External interrupt flag. Interrupt flag is active high, write 1 to clear
[14]	IF14	GPIO 0[14]/P0[14] External interrupt flag. Interrupt flag is active high, write 1 to clear
[13]	IF13	GPIO 0[13]/P0[13] External interrupt flag. Interrupt flag is active high, write 1 to clear
[12]	IF12	GPIO 0[12]/P0[12] External interrupt flag. Interrupt flag is active high,

		write 1 to clear
[11]	IF11	GPIO 0[11]/P0[11] External interrupt flag. Interrupt flag is active high, write 1 to clear
[10]	IF10	GPIO 0[10]/P0[10] External interrupt flag. Interrupt flag is active high, write 1 to clear
[9]	IF9	GPIO 0[9]/P0[9] External interrupt flag. Interrupt flag is active high, write 1 to clear
[8]	IF8	GPIO 0[8]/P0[8] External interrupt flag. Interrupt flag is active high, write 1 to clear
[7]	IF7	GPIO 0[7] / P0[7] External interrupt flag. Interrupt flag is active high, write 1 to clear
[6]	IF6	GPIO 0[6] / P0[6] External interrupt flag. Interrupt flag is active high, write 1 to clear
[5]	IF5	GPIO 0[5] / P0[5] External interrupt flag. Interrupt flag is active high, write 1 to clear
[4]	IF4	GPIO 0[4] / P0[4] External interrupt flag. Interrupt flag is active high, write 1 to clear
[3]	IF3	GPIO 0[3] / P0[3] External interrupt flag. Interrupt flag is active high, write 1 to clear
[2]	IF2	GPIO 0[2] / P0[2] External interrupt flag. Interrupt flag is active high, write 1 to clear
[1]	IF1	GPIO 0[1] / P0[1] External interrupt flag. Interrupt flag is active high, write 1 to clear
[0]	IF0	GPIO 0[0] / P0[0] External interrupt flag. Interrupt flag is active high, write 1 to clear

7.2.15.4 WAKE_POL

Address: 0x4001_2110

Reset value: 0x0

Table 7-20 WAKE_POL External Wake-up Source Polarity Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												GPIO1_1_POL	GPIO1_0_POL	GPIO0_1_POL	GPIO0_0_POL
												RW	RW	RW	RW
												0	0	0	0

Location	Bit name	Description
[31:4]		Unused
[3]	GPIO1_1_POL	GPIO 1 [1]/P1 [1] External wake-up enable. 1: enable; 0: disable
[2]	GPIO1_0_POL	GPIO 1 [0]/P1 [0] External wake-up enable. 1:enable; 0: disable
[1]	GPIO0_1_POL	GPIO 0 [1]/P0 [1] External wake-up enable. 1:enable; 0: disable
[0]	GPIO0_0_POL	GPIO 0 [0]/P0 [0] External wake-up enable. 1:enable; 0: disable

7.2.15.5 WAKE_EN

Address: 0x4001_2114

Reset value: 0x0

Table 7-21 WAKE_EN External Wake-up Source Enable Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												GPIO1_1_WKEN	GPIO1_0_WKEN	GPIO0_1_WKEN	GPIO0_0_WKEN
												RW	RW	RW	RW
												0	0	0	0

Location	Bit name	Description
[31:4]		Unused
[3]	GPIO1_1_WKEN	GPIO 1[1] / P1[1] External wake-up enable. 1: Enable; 0: Disable.
[2]	GPIO1_0_WKEN	GPIO 1[0] / P1[0] External wake-up enable. 1: Enable; 0: Disable.
[1]	GPIO0_1_WKEN	GPIO 0[1] / P0[1] External wake-up enable. 1: Enable; 0: Disable.
[0]	GPIO0_0_WKEN	GPIO 0[0] / P0[0] External wake-up enable. 1: Enable; 0: Disable.

7.3 Function Implementation

7.3.1 Pull-up

LKS32MC05X series chips are implemented by internal analog circuits with pull-up function. All GPIOs have a pull-up control register PUE, but not all GPIOs have a pull-up circuit. The GPIOs equipped with the pull-up function are as follows:

Table 7-22 GPIO Pull-up Resource Distribution

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P0	√	√		√	√				√	√	√	√	√	√		√
P1					√								√			√
P2	√		√			√	√		√	√	√	√			√	√

7.4 Application Guide

7.4.1 External Interrupt

Examples:

```
  GPIO0_PIE = 0x0080;       //enable P0 [7] input
NVIC_EnableIRQ(GPIO_IRQn); // Enable GPIO interrupt
__enable_irq (); // Enable interrupt
  i = 1000;
  while(i--);
```

```
        // P0 [7] External square wave signal on IO
    EXTI_CR0 = 0x8000; // enable P0 [7] rising edge trigger and generate external interrupt
    while(irq_flag != 2);      // External signal is flipped twice, two interrupts are generated, irq_flag is handled in GPIO interrupt
                                        Increment twice in the program
    EXTI_CR0 = 0x4000; // enable p0 [7] falling edge trigger and generate external interrupt
    while(irq_flag != 4);
    EXTI_CR0 = 0xC000; // Enable P0 [7] rising edge and falling edge trigger at the same time, and generate external interrupt
    while(irq_flag != 8);
    EXTI_CR0 = 0x0000; // disable P [7] rising and falling edge trigger at the same time, external interrupt cannot be generated
      i = 1000;
      while(i--);
    if(irq_flag != 8)FAIL;
      i = 1000;
      while(i--);
      PASS;
    }
```

7.4.2 GPIO Analog Mode

Turn off the GPIO IE and OE to use the analog function. And then, the PAD is directly connected to the analog module through the internal resistance.

8 CRC

8.1 Introduction

CRC, or Cyclic Redundancy Check Code: It is the most commonly used error check code in the field of data communication. Its characteristic is that the length of the information field and the check field can be selected at will. Cyclic Redundancy Check (CRC) is a data transmission error detection function that performs polynomial calculation on the data and appends the obtained result to the back of the frame; The receiving device also implements a similar algorithm to ensure the correctness and integrity of the data transmission.

The process of error detection using CRC can be described as to generate an R-bit CRC code for verification with certain rules at the transmit end according to the K-bit binary code sequence, and then append the CRC code to the original information to form a new binary code sequence number of K+R bits, and then transmit. After then, check at the receiving end according to the rules followed between the information code and the CRC code, to determine whether there is an error in transmission. This rule is called "generator polynomial" in error control theory.

8.2 Basic Principles

Basic principles of CRC: Append the R-bit check code to the K-bit information code, and the entire code length is N bits. Thus, such code is also called (N, K) code. For a given (N, K) code, it can be proved that there is a polynomial G (x) with the highest powerof N-K=R. A K-bit information check code can be generated with a G (x), and G (x) is called the generator polynomial. The generation process of the check code is: Assuming that the information to be transmitted is polynomial C (X), and then shift C (x) to the left by R bits (C (x)*2^R). Then, the R bit will be vacated to the right of C (x), which is the position of the check code. The remainder obtained by dividing C (x)*2^R by the generator polynomial G (x) is the check code.

Any code composed of binary bit strings can correspond to a polynomial whose coefficients are only '0' or '1'. For example, the polynomial corresponding to the code 1010111 is $x^6+x^4+x^2+x+1$, and the polynomial $x^5+x^3+x^2+x+1$ corresponds to the code 101111.

8.3 Basic Concepts

8.3.1 Correspondence

Direct correspondence between polynomials and binary numbers: The highest power of X corresponds to the highest bit of the binary number, and the following bits correspond to the powers of the polynomial. A term with the same power corresponds to 1, and a term without this corresponds to 0. It can be seen that the highest power of X is R, and the converted binary number has R+1 bits.

Polynomials include generator polynomial G (X) and information polynomial C (X).

If the generator polynomial is $G(X)=X^4+X^3+X+1$, it can be converted into binary number 11011.

If the information transmitted is 101111, it can be converted into a data polynomial of $C(X)=X^5+X^3+X^2+X+1$.

8.3.2 Generator Polynomial

The generator polynomial is an agreement between the receiver and the sender, that is, a binary number. This number remains unchanged throughout the transmission process.

On the sender side, the generator polynomial is used to divide the information polynomial by 2 to generate a check code. On the receiving side, the generator polynomial is used to perform modulo-2 division detection on the received coding polynomial and determine the error location

The following conditions should be met:

A. The highest and lowest bits of the generator polynomial must be 1.

B. When an error occurs in any bit of the transmitted information (CRC code), the remainder should not be "0" after being divided by the generator polynomial.

C. When errors occur in different bits, the remainder should be different.

D. If continue to divide the remainder, the remainder should be circulated.

8.3.3 CRC Digits

CRC check digits = generator polynomial digits - 1. Please note that some shorthands for generator polynomials have omitted the highest bit 1 of the generator polynomial.

8.3.4 Generation Steps

1. Convert the generator polynomial G (X) with the highest power of X to R into the corresponding R+1 binary number.

2. Shift the information code to the left by R bits, which is equivalent to the corresponding information polynomial $C(X)*2^R$.

3. Divide the information code with a generator polynomial (binary number) to obtain the remainder of the R bits. (Note: The remainder obtained by binary division is actually the remainder obtained by modulo 2 division, and it is not equal to the remainder obtained by dividing the corresponding decimal number.) 。

4. Append the remainder to the information code, shift left and get the complete CRC code by vacating the position.

[Example] Suppose the generator polynomial used is G(X)=X3+X+1. The original 4-bit message is 1010. Find the encoded message.

Solution:

1. Convert the generator polynomial G(X)=X3+X+1 into the corresponding binary divisor 1011.

2. The generator polynomial in this question has 4 bits (R+1)(Note: The check code calculated by the 4-bit generator polynomial is 3 bits, R is the number of check codes), and the original message C (X) is shifted to the left by 3 (R) bits to 1010 000

3. Use the binary number corresponding to the generator polynomial to divide the original message shifted left by 3 bits by modulo 2 (high-bit alignment), which is equivalent to bitwise XOR:

1010000

1011

0001000

0001011

0000011

Obtained the remaining bits 011, so the final code is: 1010 011

POL=0x13，data=0x77

011101110000000

10010011

01111101000000

10010011

0110100100000

10010011

010000010000

10010011

00010001000

10010011

00011011

8.4 Register

8.4.1 Address Allocation

The base address of CRC is 0x4001_2400.

Table 8-1 CRC Register List

Name	Offset address	Description
CRC_DR	0x00	CRC data register (input information code/output code)
CRC_CR	0x04	CRC control register
CRC_INIT	0x08	CRC initial code register
CRC_POL	0x0C	Binary code register for CRC generator polynomial

8.4.2 Register Description

8.4.2.1 CRC_DR CRC Data Register

Address: 0x4001_2400

Reset value: 0x0

Table 8-2 CRC_DR CRC Data Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR															
RW															
0															

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

Location	Bit name	Description
[31:0]	DR	Store the information code to be encoded and the code after CRC check

The CRC_DR register is used not only to put the data to be checked, but also to return the check result. Writing to the CRC_DR register triggers once CRC calculation. The data to be encoded should be written last after the configuration of CR and other registers is completed, thus to trigger the CRC calculation.

8.4.2.2 CRC_CR CRC Control Register

Address: 0x4001_2404

Reset value: 0x0

Table 8-3 CRC_CR CRC Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			REV_OUT_TYPE			REV_IN_TYPE				POLY_SIZE					RESET
			RW			RW				RW					WO
			0			0				0					0

Location	Bit name	Description
[31:13]		Unused
[12]	REV_OUT_TYPE	Whether to invert the code after CRC check and output, i.e., b [31] = b [0], b [30] = b [1],... [b0] = b [31]
[11:10]		Unused
[9:8]	REV_IN_TYPE	Type of data inversion to be encoded 00: not reverse 01: Reverse by byte, i.e., b [31] = b [24], b [30] = b [25],..., b [24] = b [31],..., b [7] = b [0], b [6] = b [1],..., b [0] = b [7] 10: Reverse by half word (16 bit), i.e., b [31] = b [16], b [30] = b [17],..., b [16] = b [31],..., b [15] = b [0], b [14] = b [1],..., b [0] = b [15] 11: Reverse by words, i.e., b [31] = b [0], b [30] = b [1],... [b0] = b [31]
[7:6]		Unused
[5:4]	POLY_SIZE	Output coding (polynomial) bit width 00: 32bits 01: 16bits 10: 8bits 11: 7bits
[3:1]		Unused
[0]	RESET	Data source for CRC calculation with input information code 0: from the last calculation result 1: from CRC_INIT Write 1 to reset the CRC data and automatically clear it. The value is always 0 after readback.

It should also be noted that writing 1 to CRC_CR.RESET will reset the CRC_INIT register to 0xFFFFFFFF.

If clearing the CRC calculation result is required, write 1 to CRC_CR.RESET; otherwise, the subsequent CRC calculation will take the previous calculation result as the initial value.

8.4.2.3 CRC_INIT CRC Initial Code Register

Address: 0x4001_2408

Reset value: 0x0

Table 8-4 CRC_INIT CRC Initial Code Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INIT															
RW															
0															

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

Location	Bit name	Description
[31:0]	INIT	Store initial code

CRC_DR and CRC_INIT start to perform CRC check calculation after XOR.

8.4.2.4 CRC_POL CRC Generation Code Register

Address: 0x4001_240C

Reset value: 0x0

Table 8-5 CRC_POL CRC Generation Code Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POL															
RW															
0															

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

Location	Bit name	Description
[31:0]	POL	Store the generator code for the generator polynomial

9 ADC

9.1 Introduction

The LKS32MC05X series chip integrates a 12-bit SAR ADC and has 16 input channels. The main features are as follows:

- Max 3Msps sampling rate, system frequency is 48MHz.
- Support up to 16 channels selection.
- Support software and hardware trigger.
- Link with MCPWM and Timer units to trigger indication signal, which can be transmit through GPIO for debugging.
- Support sampling with custom sampling sequences, such as single-stage, double-stage, and four-stage, and the sequence number and channel number can be set flexibly.
- Support left and right alignment mode.

The quantifier meaning conventions for ADC sampling are:

Sampling for a time: complete the sampling conversion of the corresponding analog signal quantity to the data signal quantity of one channel, and store the digital quantity to the ADC_DATx register;

One episode of sampling: may contain one or several samples. The analog channels for several samples can be the same or different. Sampling is usually triggered by MCPWM, UTimer, or software. One trigger signal is for one sampling. After sampling is completed, a corresponding stage sampling completion interrupt is generated; Taking the four-stage sampling triggered by MCPWM as an example, each stage is sampled three times (complete three analog samples). TADC [0] triggers the ADC to start the first stage of sampling. After the first stage of sampling is completed, the ADC enters the waiting state and waits for the trigger event of TADC [1]; After TADC [1] occurs, trigger the ADC to start the second stage of sampling; In the same way, TADC [2]/TADC [3] triggers the sampling of the third and fourth stages respectively.

One round of sampling: may contain one, two or four stage of sampling. Each stage is triggered by a specific trigger signal; After completing one round of sampling, the ADC returns to the idle state and waits for the next trigger.

The maximum ADC clock is 48MHz, the sampling time is configurable with 4-68 ADC clock cycles, and the data conversion time is fixed at 12 cycles. Taking the sampling time of 4Cycle as an example, sampling + conversion is at least 16Cycle, that is, the maximum conversion rate of ADC is 48/16=3Msps. However, considering the sampling establishment time, it is recommended to set the sampling time to 12Cycle or above, which corresponds to the conversion speed rate of 48/(12+12)=2Msps.

9.1.1 Functional Block Diagram

The ADC interface includes 16 data registers ((the digital quantity corresponding to the analog quantity of each channel for 16 ADC samplings)) and several control registers.

The data register ADC_DATx is used to store the digital value obtained by the analog-to-digital converter (ADC) at the xth sampling. The source of the converted analog signal is selected by a 4-bit result within the register ADC_CHNx (see 9.2.3 Signal Source and Register for details) Taking ADC_CHN0 as an example, select the analog channel number of the 0th sampling for bits [3:0], and the channel numbers CH0~CH15 are optional. If ADC_CHN0 [3:0] = 0 and ADC_CHN0 [7:4] = 3, the analog value of the 0th sampling corresponds to channel CH0, the analog value of the 1st sampling corresponds to channel CH3, and so on.

The sampling times register ADC_CHNT controls the number of samples per stage. 1-15 correspond to 1- 15 samplings.

The control logic selects the trigger signal from MCPWM or the universal timer (UTimer) according to trigger configuration register ADC_TRIG to start sampling or initiates the sampling through the software trigger. MCPWM/UTimer will send timed trigger signal TADC [0]/TADC[1]/TADC[2]/TADC[3], which can be selected as the trigger signal. The selection of trigger signals is stored in the control register.

After a stage conversion is completed (sampling and conversion of all channels within a stage is completed), the ADC conversion completion interruption will be triggered. In the multi-stage trigger mode, the conversion completion of each stage can trigger one conversion done interruption.

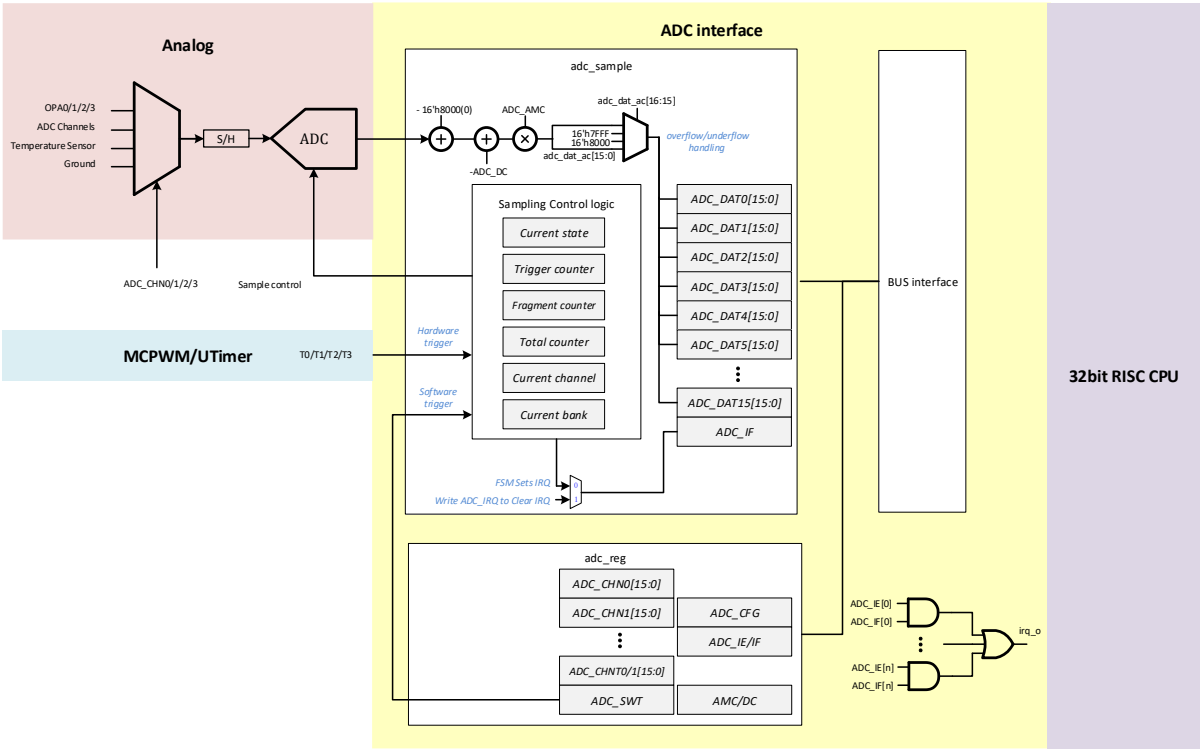


Fig. 9-1 Functional Block Diagram of ADC Sampling

The user can set the sampling sequence and the source of the sampled signal flexibly, and even realize multiple sampling of a single signal. The control register allows the user to set the number of sampling, thus to improve the sampling frequency or reduce the sampling power consumption.

9.1.2 ADC Trigger Mode

- Support single-stage trigger, double-stage trigger, four-stage trigger to complete sampling
- Single-stage trigger can set the number of trigger events
- The trigger source of the double-stage trigger can only be the timing signal TADC[0]+TADC[1] of MCPWM/UTimer, or the twice software triggers.
- The trigger source of the four-stage trigger can only be the timing signal TADC[0]+TADC[1]+TADC[2]+TADC[3] of MCPWM/UTimer, or quartic software triggers.
- Set to generate an interrupt after each trigger is completed.
- Trigger indication signal can be transmitted through GPIO for debugging

9.1.3 ADC Output Digital System

The ADC output data is 12-bit complement. The input signal 0 corresponds to 12h'0000_0000_0000. Taking the double gain configuration as an example, the input signal -2.4V corresponds to 12h'1000_0000_0000, and the input signal+2.4V corresponds to 12h'0111_1111_1111. The 12-bit complementary code after ADC conversion should be expanded to 16-bit and stored in the sampling data register of 16-bit width. Left or right alignment can be set by the configuration register. Taking 12'h1000_0000_1101 as an example, if the configuration is left-aligned, the right side is filled with four "0", and the value stored in ADCx_DAT is 16'h1000_0000_1101_0000; If the configuration is right-aligned, sign expansion is performed on the left, and the value stored in ADCx_DAT is 16'h1111_1000_0000_1101. Left alignment is recommended.

Please note that the final ADC data may exceed the 12-bit signed number range for the sake of gain calibration and DC offset calibration. For example, in the right-aligned mode, the number of ADC conversions may be 0xF745, intercepting the lower 12 bits, and taking out 0x745 at this time may cause negative numbers to be treated as positive numbers, i.e., an overflow error may occur. Or the digital quantity of a certain conversion of ADC may be 0x0810. Then, intercept the lower 12bit and take out 0x810, which may cause positive numbers to be treated as negative numbers by mistake. Therefore, the ADC data should be processed as a 16-bit signed number.

Table 9-1 Conversion of ADC Output Digital System

ADC Double Gain Input Analog Value/V	ADC 2/3 Times Gain Input Analog Value/V	Converted Signed Number
2.4	3.6	12'h0111_1111_1111
0	0	12'h0000_0000_0000
-2.4	-3.6	12'h1000_0000_0000

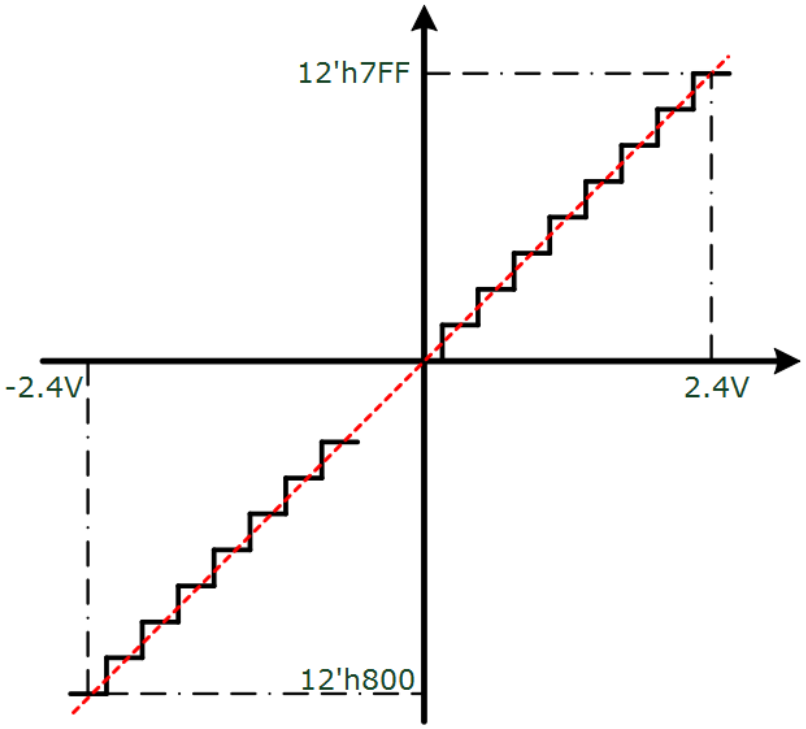


Fig. 9-2 ADC Digital Range at Double Gain Setting

9.1.4 ADC Range

The ADC has two gain modes: high gain (1 time) and low gain (2/3 times). The ADC ranges of these two gains are also different. In 1-time gain mode, the maximum input signal amplitude is ± 2.4V; in 2/3-time gain mode, the maximum input signal amplitude is ± 3.6V.

When the ADC sampling channel is set as the output signal of the OPA (i.e. OPA0 ~ OPA3), the appropriate OPA gain should be selected. This will allow the maximum signal in a specific application to be amplified to a level close to +/- 3.3V, while setting the ADC to a 2/3-time gain. . For example, the maximum phase current is 100A (effective value of sine wave), and the MOS internal resistance (assuming as MOS internal resistance sampling) is 5mR, then the maximum input signal amplitude of the OPA is +/- 707mV. Then, the OPA gain should be selected to be 4.5 times, and the amplified signal is about +/- 3.18V.

If the output signal of the OPA is amplified and the maximum signal is still less than +/- 2.4V due to objective reasons, the gain of the ADC should be set to 1 time.

When the ADC sampling channel is set as the input signal of the GPIO multiplex port, the ADC gain is also selected according to the maximum amplitude of the signal. Due to the limitation of the IO port, the signal range of the GPIO multiplex port can only be between -0.3V ~ AVDD+0.3V.

9.1.5 ADC Correction

The ADC hardware interface module can perform DC offset calibration and gain calibration.

The $AMP_{correction}$, which is a gain correction factor, stored by ADC_AMC is a 10-bit unsigned fixed-

point number. ADC_AMC [9] is the integer, and ADC_AMC [8: 0] is the decimal, which can represent a fixed-point number whose value is around "1".

ADC_DC stores the DC offset of the ADC. Usually, the ADC DC offset value is obtained by measuring the AVSS (internal ground) of channel 15 (counted from 0) during the correction phase and stored in the flash, and is stored in the flash. The DC offset is written into the ADC_DC register.by the software at the system loading phase.

Record that the digital quantity output by the ADC is D_{ADC}, the true value corresponding to DADC is D, and D_0 is 0 in the coding system, then

$D = Saturation((D_{ADC}-D_0)*AMP_{correction}-DC_{offset})$

Finally, the hardware will store the corrected D into the corresponding sampling data register.

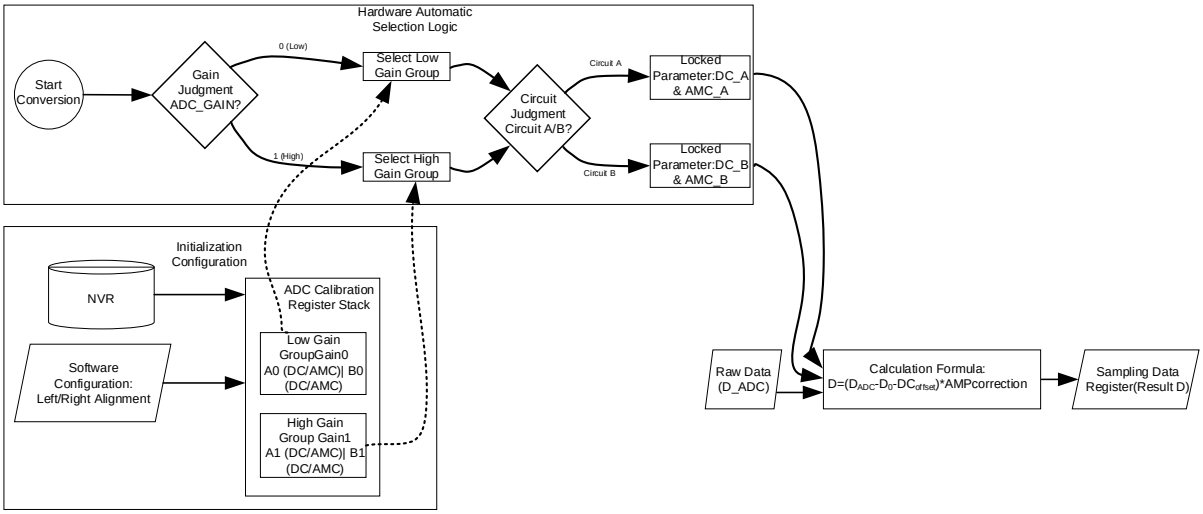


Fig. 9-3 ADC Calibration Processing Flowchart

9.1.6 ADC Signal Source

Table 9-2 ADC Sampling Signal Source

ADC_CHNx	Sampling signal
0	OPA0 output
1	OPA1 output
2	ADC_CH2
3	ADC_CH3
4	ADC_CH4
5	ADC_CH5
6	ADC_CH6
7	ADC_CH7
8	ADC_CH8
9	ADC_CH9
A	ADC_CH10
B	ADC_CH11
C	ADC_CH12
D	ADC_CH13
E	Temperature sensor

F	Internal ground/SWCLK

When SYS_AFE_REG6.CH15EN=1, ADC Channel 15 is connected to SWCLK IO instead of Internal ground.

9.1.7 ADC configuration process

Recommended configuration process:

1. Turn on the ADC analog switch and select the ADC operating frequency

   Turn on the ADC by setting SYS_AFE_REG5.ADCPDN to 1. Before turning on ADC, the BGP module, 4MHz RC oscillator and PLL should be turned on first, and then select the ADC frequency by setting the SYS_AFE_REG7.ADCCLKSEL[5:4]. 00 means 48MHz, 01 means 12MHz, 11 means 24MHz.

2. Configure ADC data output format

   The output format of the ADC can be left or right alignment by setting the ADC_CFG.DATA_ALIGN. 0 means left alignment, 1 means right alignment.

3. Configure ADC sampling mode

   The sampling mode of the ADC can be selected by setting the ADC_TRIG.TRG_MODE[9:8]. 00 means single-stage trigger, 01 means double-stage trigger, 11 means four-stage trigger.

4. Configure ADC trigger events

   The trigger event of ADC sampling can be selected by setting the ADC_CFG.SEL, and there are a total of 8 sampling events to choose from. In single-segment sampling mode, the ADC_TRIG.SINGLE_TCNT[7:4] can be set to select the number of events required to trigger one sampling. The setting range is 0~15. 0 means that one event triggers one sampling, and 15 means that 16 events trigger one sampling.

5. Configure ADC Range

   The gain mode can be selected by configuring the SYS_AFE_REG0.GA_AD. 0 means low gain (2/3 times), 1 means high gain (1 times).

6. Configure the number of ADC channels and select the sampling signal source

   Setting ADC_CHNT can select the number of channels to be sampled in each sampling mode. The setting range is 1~20, and 1 represents one channel. The ADC_CHN0, ADC_CHN1 and other registers are configured to select the sampling signal source of the ADC, and the setting range is 0~15.

7. Configuring ADC Interrupts

   The ADC has a total of six interrupts: the first to fourth segment sampling completion interrupts, software triggered interrupts that occur in non-idle states and hardware triggered interrupts that occur in non-idle states. The above interrupts can be enabled by configuring the ADC_IE register. Even if interrupts are

not enabled, interrupt events can still set ADC_IF, but no interrupt request will be raised.

9.2 Register

9.2.1 Address Allocation

Base address of ADC is 0x40011400.

Table 9-3 ADC0 Register List

Name	Offset address	Description
ADC_DAT0	0x00	ADC 0th sampling data
ADC_DAT1	0x04	ADC 1st sampling data
ADC_DAT2	0x08	ADC 2nd sampling data
ADC_DAT3	0x0C	ADC 3rd sampling data
ADC_DAT4	0x10	ADC 4th sampling data
ADC_DAT5	0x14	ADC 5th sampling data
ADC_DAT6	0x18	ADC 6th sampling data
ADC_DAT7	0x1C	ADC 7th sampling data
ADC_DAT8	0x20	ADC 8th sampling data
ADC_DAT9	0x24	ADC 9th sampling data
ADC_DAT10	0x28	ADC 10th sampling data
ADC_DAT11	0x2C	ADC 11th sampling data
ADC_DAT12	0x30	ADC 12th sampling data
ADC_DAT13	0x34	ADC 13th sampling data
ADC_DAT14	0x38	ADC 14th sampling data
ADC_DAT15	0x3C	ADC 15th sampling data
ADC_CHN0	0x40	0th ~ 3rd Sampling signal selection
ADC_CHN1	0x44	4th ~ 7th Sampling signal selection
ADC_CHN2	0x48	8th ~ 11th Sampling signal selection
ADC_CHN3	0x4C	12th ~ 15th Sampling signal selection
ADC_CHNT	0x50	Number of samples in various trigger modes
ADC_CFG	0x54	ADC alignment mode configuration
ADC_SWT	0x58	ADC software trigger
ADC_DC	0x60	ADC DC offset
ADC_AMC	0x64	ADC gain correction
ADC_IE	0x68	ADC interrupt enabling
ADC_IF	0x6C	ADC interrupt flag

9.2.2 Sampling Data Register

9.2.2.1 ADC_DAT0

Address: 0x4001_1400

Reset value: 0x0

Table 9-4 ADC_DAT0 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT0															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT0	ADC 0th sampling data

9.2.2.2 ADC_DAT1

Address: 0x4001_1404

Reset value: 0x0

Table 9-5 ADC_DAT1 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT1															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT1	ADC 1st sampling data

9.2.2.3 ADC_DAT2

Address: 0x4001_1408

Reset value: 0x0

Table 9-6 ADC_DAT2 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT2															
RW															
0															

Location	Bit name	Description
[31:16]		Unused

[15:0]	DAT2	ADC 2nd sampling data

9.2.2.4 ADC_DAT3

Address: 0x4001_140C

Reset value: 0x0

Table 9-7 ADC_DAT3 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT3															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT3	ADC 3rd sampling data

9.2.2.5 ADC_DAT4

Address: 0x4001_1410

Reset value: 0x0

Table 9-8 ADC_DAT4 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT4															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT4	ADC 4th sampling data

9.2.2.6 ADC_DAT5

Address: 0x4001_1414

Reset value: 0x0

Table 9-9 ADC_DAT5 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT5															

RW
0

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT5	ADC 5th sampling data

9.2.2.7 ADC_DAT6

Address: 0x4001_1418

Reset value: 0x0

Table 9-10 ADC_DAT6 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT6															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT6	ADC 6th sampling data

9.2.2.8 ADC_DAT7

Address: 0x4001_141C

Reset value: 0x0

Table 9-11 ADC_DAT7 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT7															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT7	ADC 7th sampling data

9.2.2.9 ADC_DAT8

Address: 0x4001_1420

Reset value: 0x0

Table 9-12 ADC_DAT5 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT8															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT8	ADC 8th sampling data

9.2.2.10 ADC_DAT9

Address: 0x4001_1424

Reset value: 0x0

Table 9-13 ADC_DAT9 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT9															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT9	ADC 9th sampling data

9.2.2.11 ADC_DAT10

Address: 0x4001_1428

Reset value: 0x0

Table 9-14 ADC_DAT10 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT10															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT10	ADC 10th sampling data

9.2.2.12 ADC_DAT11

Address: 0x4001_142C

Reset value: 0x0

Table 9-15 ADC_DAT11 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT11															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT11	ADC 11th sampling data

9.2.2.13 ADC_DAT12

Address: 0x4001_1430

Reset value: 0x0

Table 9-16 ADC_DAT12 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT12															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT12	ADC 12th sampling data

9.2.2.14 ADC_DAT13

Address: 0x4001_1434

Reset value: 0x0

Table 9-17 ADC_DAT 13 Sampling Data Register

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

DAT13
RW
0

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT13	ADC 13th sampling data

9.2.2.15 ADC_DAT14

Address: 0x4001_1438

Reset value: 0x0

Table 9-18 ADC_DAT 14 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT14															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT14	ADC 14th sampling data

9.2.2.16 ADC_DAT15

Address: 0x4001_143C

Reset value: 0x0

Table 9-19 ADC_DAT 15 Sampling Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT15															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	DAT15	ADC 15th sampling data

9.2.3 Signal Source Register

9.2.3.1 ADC_CHN0

Address: 0x4001_1440

Reset value: 0x0

Table 9-20 ADC_CHN0 Signal Source Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS3				DS2				DS1				DS0			
RW				RW				RW				RW			
0				0				0				0			

Location	Bit name	Description
[31:16]		Unused
[15:12]	DS3	ADC sampling signal selection 3
[11:8]	DS2	ADC sampling signal selection 2
[7:4]	DS1	ADC sampling signal selection 1
[3:0]	DS0	ADC sampling signal selection 0

9.2.3.2 ADC_CHN1

Address: 0x4001_1444

Reset value: 0x0

Table 9-21 ADC_CHN1 Signal Source Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS7				DS6				DS5				DS4			
RW				RW				RW				RW			
0				0				0				0			

Location	Bit name	Description
[31:16]		Unused
[15:12]	DS7	ADC sampling signal selection 7
[11:8]	DS6	ADC sampling signal selection 6
[7:4]	DS5	ADC sampling signal selection 5
[3:0]	DS4	ADC sampling signal selection 4

9.2.3.3 ADC_CHN2

Address: 0x4001_1448

Reset value: 0x0

Table 9-22 ADC_CHN2 Signal Source Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS11				DS10				DS9				DS8			
RW				RW				RW				RW			
0				0				0				0			

Location	Bit name	Description
[31:16]		Unused
[15:12]	DS11	ADC sampling signal selection 11
[11:8]	DS10	ADC sampling signal selection 10
[7:4]	DS9	ADC sampling signal selection 9
[3:0]	DS8	ADC sampling signal selection 8

9.2.3.4 ADC_CHN3

Address: 0x4001_144C

Reset value: 0x0

Table 9-23 ADC_CHN3 Signal Source Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DS15				DS14				DS13				DS12			
RW				RW				RW				RW			
0				0				0				0			

Location	Bit name	Description
[31:16]		Unused
[15:12]	DS15	ADC sampling signal selection 15
[11:8]	DS14	ADC sampling signal selection 14
[7:4]	DS13	ADC sampling signal selection 13
[3:0]	DS12	ADC sampling signal selection 12

9.2.4 Sampling Times Register

9.2.4.1 ADC_CHNT

Address: 0x4001_1450

Reset value: 0x0

Table 9-24 ADC_CHNT Sampling Times Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S4				S3				S2				S1			

RW	RW	RW	RW
0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15:12]	S4	Number of samples for the fourth stage in four-stage sampling mode
[11:8]	S3	Number of samples for the third stage in four-stage sampling mode
[7:4]	S2	Number of samples for the second stage in two-stage or four-stage sampling mode
[3:0]	S1	Number of samples for the first stage in single-stage, two-stage or four-stage sampling mode

Note: The number of samplings is not allowed to be set to 0 for each stage. 1 means one sample, 2 means two samples, ..., 12 means twelve samples, ..., 16 means sixteen samples.

9.2.5 Configuration Register

9.2.5.1 ADC_CFG

Address: 0x4001_1454

Reset value: 0x0

Table 9-25 ADC_CFG Mode Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			TSEL	FSM_RESET	DATA_ALIGN	TRG_MODE		SINGLE_TCNT				TRG_EN			
			RW	WO	RW	RW		RW				RW			
			0	0	0	0		0				0			

Location	Bit name	Description
[31:13]		Unused
[12]	SEL	TADC Trigger Control Register 0:MCPWM，1:UTimer
[11]	FSM_RESET	State machine reset control signal. Write 1 to the software to generate reset, and the internal state machine of ADC will return to the initial state. Such reset control does not affect the configuration value of other ADC registers.
[10]	DATA_ALIGN	ADC_DAT alignment 0: Left aligned, with 4'h0 at the right, 1: right-aligned, with 4bit sign bit on the left
[9:8]	TRG_MODE	Trigger Mode Selection 0: single-stage trigger; 1: double-stage trigger;2: reserved; 3: four-stage trigger
[7:4]	SINGLE_TCNT	Number of events required to trigger a sample in single-stage trigger mode 0: means one sampling can be triggered by one event occurred 1: means one sampling can be triggered by two events occurred …… 15: means one sampling can be triggered by sixteen events

		occurred
[3:0]	TRG_EN	TADC triggers ADC sampling enabling signal. Active high, low by default. TRG_EN[0]: TADC[0] enable switch TRG_EN[1]: TADC[1] enable switch TRG_EN[2]: TADC[2] enable switch TRG_EN[3]: TADC[3] enable switch High corresponding position means enabling. The pull-down means shutdown.

The trigger source of ADC is MCPWM or UTimer. It is not possible to use MCPWM and UTimer to trigger at the same time. You can only choose one of the two. There are four trigger events in total-TADC[3:0].

If the ADC trigger source is UTimer, the ADC sampling trigger event is UTimer_T0/ UTimer_T1/ UTimer_T2/ UTimer_T3; it corresponds to the comparison events of Timer2 channel 0/1 and Timer3 channel 0/1 in turn.

If the ADC trigger source is MCPWM. ADC sampling trigger events are MCPWM_T0/ MCPWM_T1/ MCPWM_T2/ MCPWM_T3 generated by MCPWM.

The trigger signal of UTimer to ADC can be sent by setting GPIO as the eighth function, i.e., Timer2/3 function, which is used to capture and debug.

The trigger signal of MCPWM to ADC can be sent by setting GPIO as the ninth function, i.e., ADC_TRIGGER function, which is used to capture and debug. Every time an ADC trigger occurs, the ADC_TRIGGER signal will flip once.

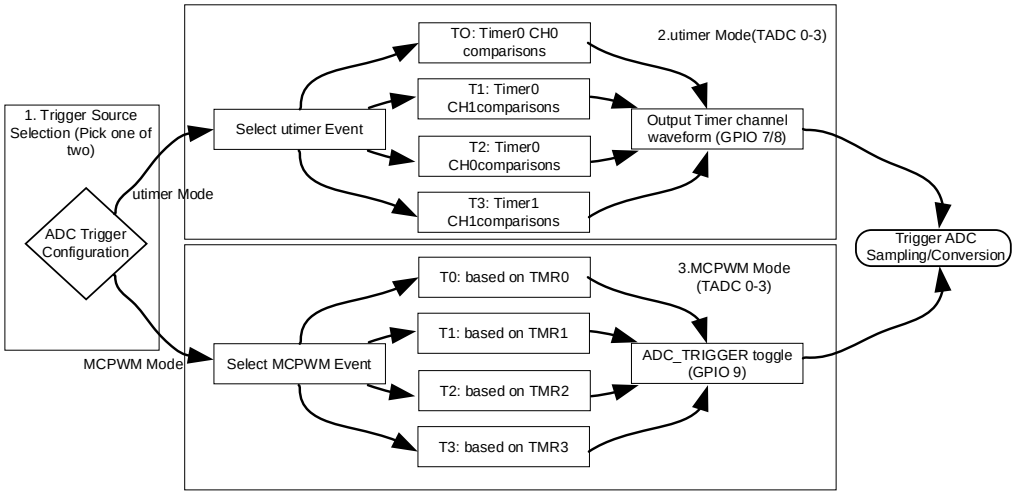


Fig. 9-4 ADC Trigger Source Selection & Debug Logic Flowchart

9.2.6 Software Trigger Register

9.2.6.1 ADC_SWT

Address: 0x4001_1458

Reset value: 0x0

Table 9-26 ADC_SWT Software Trigger Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	SWT	When the written data is 0x5AA5, a software trigger will be generated

Please note that the software trigger acquisition register is write-only, and the software trigger event is generated only when the write data is 0x5AA5. One write to the bus generates one software trigger. When a software trigger is generated after data is written, the register will be automatically cleared, waiting for the next software trigger.

9.2.7 DC Offset Register

A signal source of the ADC is GND, and the DC offset of the system can be obtained by measuring this channel.

Generally, after initialized, the system will measure the GND signal once and the measured value is stored in the DC offset register. Each subsequent sampling value of other channel signals will automatically subtract the DC offset, and then store it in the converted digital register.

Taking into account the signal error, the signal that removes the DC offset may overflow, and the overflow data will be saturated to make the signal in the range of -2.4V~2.4V or -3.6V~3.6.The chip has been calibrated by the factory before delivery, and the calibration data is stored in Flash info. When the chip is powered on, the calibration parameters are automatically loaded. When initializing the ADC module, you need to configure the DC offset according to the data alignment mode. See the library function provided by the chip vendor. After the ADC module is soft reset using SYS_SFT_RST, the registers inside the ADC are reset. The ADC initialization function needs to be reused to read the calibration parameters such as DC/AMC from the NVR.

9.2.7.1 ADC_DC

Address: 0x4001_1460

Reset value: 0x0

Table 9-27 ADC_DC Offset Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	DC_OFFSET	Sampling Circuit Gain ADC DC Offset

Considering that the ADC DC offset is close to 0, the register here only has the low 10bit which is actually realized, the high 6bit is only the sign extension of ADC_DC[9], and the ADC_DC register will also extend the sign when participating in ADC data correction. I.e., if you write 0x12B0 to ADC_DC, 0x2B0 is actually written, and 0xFFB0 will be read during reading.

The ADC_DC value stored here should correspond to the right-aligned offset value. When the ADC_CFG register is configured to be left-aligned, the hardware will automatically adjust ADC_DC to participate in ADC correction according to the alignment setting. Specifically, during the right alignment, ADC_DC[15:0] directly participates in the correction operation. During left alignment, ADC_DC[15:0] firstly shifts to the left by 4 bits and then participates in the ADC correction operation.

9.2.8 Gain Correction Register

9.2.8.1 ADC_AMC

Address: 0x4001_1464

Reset value: 0x0

Table 9-28 ADC_AMC Gain Correction Register

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

Location	Bit name	Description
[31:10]		Unused
[9:0]	AM_CALI	Sampling Circuit ADC Gain Correction Register

9.2.9 Interrupt Register

9.2.9.1 ADC_IE

Address: 0x4001_1468

Reset value: 0x0

Table 9-29 ADC_IE Interrupt Enable Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										HERR_IE	SERR_IE	S4_IE	S3_IE	S2_IE	S1_IE
										RW	RW	RW	RW	RW	RW
										0	0	0	0	0	0

Location	Bit name	Description
[31:6]		Unused
[5]	HERR_IE	Hardware trigger interrupt enable that occurs in non-idle state
[4]	SERR_IE	Software trigger interrupt enable that occurs in non-idle state
[3]	S4_IE	Interrupt enable triggered by the completion of the fourth-stage sampling
[2]	S3_IE	Interrupt enable triggered by the completion of the third-stage sampling
[1]	S2_IE	Interrupt enable triggered by the completion of the second-stage sampling
[0]	S1_IE	Interrupt enable triggered by the completion of the first-stage sampling

9.2.9.2 ADC_IF

Address: 0x4001_146C

Reset value: 0x0

Table 9-30 ADC_IF Interrupt Flag Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										HERR_IF	SERR_IF	S4_IF	S3_IF	S2_IF	S1_IF
										RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
										0	0	0	0	0	0

Location	Bit name	Description
[31:6]		Unused

[5]	HERR_IF	Interrupt flag for hardware trigger occurs in non-idle state
[4]	SERR_IF	Interrupt flag for software trigger occurs in non-idle state
[3]	S4_IF	Interrupt flag for the fourth-stage sampling completion
[2]	S3_IF	Interrupt flag for the third-stage sampling completion
[1]	S2_IF	Interrupt flag for the second-stage sampling completion
[0]	S1_IF	Interrupt flag for the first-stage sampling completion

In the above ADC_IF flag, 0: indicates that no interrupt has occurred, 1: indicates that an interrupt has occurred, write 1 to clear.

9.3 Application Guide

9.3.1 ADC Sampling Trigger Mode

The ADC supports one-stage, two-stage, and four-stage sampling modes. The sampling start of each stage requires a specific external event to trigger, and each stage of sampling supports setting different sampling times and sampling signal channels. The state transition inside the ADC is described as follows, including eight states: sampling state 0 ~ 3 and idle state 0 ~ 3.

First Trigger

ADC sampling can be triggered by the timing event TADC[0]/TADC[1]/TADC[2]/TADC[3] from MCPWM/UTimer. Any one or several trigger samples of four trigger sources are available for selection, and it can also be triggered by writing command words to ADC_SWT using 16'h5AA5 software.

First Stage of Sampling

Determine whether it is one-stage sampling.

Yes: When the sampling times reaches the preset value ADC_CHNT[3:0], the ADC will return to the idle state 0; if sampling times has not reached the preset value, the sampling continues.

No: When the sampling times reaches the preset value ADC_CHNT[3:0], the ADC will enter the idle state 1 (the first stage of two-stage or four-stage sampling is completed, waiting for the second stage to be triggered); if the sampling times has not reached the preset value, the first stage of sampling continues.

Second Trigger

Second Stage of Sampling

When the sampling times of the second stage reaches the preset value ADC_CHNT[7:4], it will determine whether it is a two-stage sampling.

Yes: End this sampling and return to idle state 0.

No: Enter idle state 2 and wait for the third and fourth triggers to complete sampling.

Third Trigger

Third Stage of Sampling

When the sampling times of the third stage reaches the preset value ADC_CHNT[11:8], the ADC will enter the idle state 3.

Fourth Trigger

Fourth Stage of Sampling

When the sampling times of the fourth stage reaches the preset value ADC_CHNT [15:12], the ADC will return to the idle state 0.

The trigger conditions of various hardware trigger modes are summarized in Table 9-30 ADC Sampling Trigger Mode. The single-stage sampling mode is different. It can determine by the ADC0_CFG register that whether one TADC event triggers sampling, or the sampling is triggered only when a certain number of TADC events had occurred, while the two-stage and four-stage sampling modes only support one round of sampling to be triggered by one TADC event.

Besides, the ADC module also supports the sampling to be triggered by writing special values through software. The software trigger also only supports trigger sampling by writing once.

Table 9-31 ADC Sampling Trigger Mode

<table>
<tr><th></th><th>Single-stage Trigger</th><th>Two-stage Trigger</th><th>Four-stage Trigger</th></tr>
<tr><td rowspan="6">TADC trigger</td><td>None (TADC trigger is not enabled)</td><td rowspan="6">The first stage TADC[0]
The second stage TADC[1]</td><td rowspan="6">The first stage TADC[0]
The second stage TADC[1]
The third stage TADC[2]
The fourth stage TADC[3]</td></tr>
<tr><td>C times TADC[0]</td></tr>
<tr><td>C times TADC[1]</td></tr>
<tr><td>C times TADC[2]</td></tr>
<tr><td>C times TADC[3]</td></tr>
<tr><td>C times* TADC[0]/TADC[1]/ TADC[2]/TADC[3]</td></tr>
<tr><td>Software trigger</td><td>Write 16‘h5aa5 to ADC_SWT</td><td>The first stage: Write 16‘h5aa5 to ADC_SWT
The second stage: Write 16‘h5aa5 to ADC_SWT</td><td>The first stage: Write 16‘h5aa5 to ADC_SWT
The second stage: Write 16‘h5aa5 to ADC_SWT
The third stage: Write 16‘h5aa5 to ADC_SWT
The fourth stage: Write 16‘h5aa5 to ADC_SWT</td></tr>
</table>

*C times is set by ADC_CFG.SINGLE_TCNT. ADC_CFG.SINGLE_TCNT is only used under single-stage trigger mode. If TADC[3:0] is all enabled at the same time, all the 4 trigger sources will be counted, and an ADC sampling conversion will be triggered every time SINGLE_TCNT is reached.

9.3.1.1 Single-stage Trigger Mode

Single-stage triggering completes a sampling action when a trigger is received. One stage of sampling may include multiple samplings of the analog signal, and the sampling times are set by the

stage register ADC_CHNT; When the register value is 1 ~ 16, the corresponding sampling times are 1 ~ 16.

Assuming that the number of single-stage sampling configuration channels is 4, the sampled and converted data will be filled in ADC_DAT0, ADC_DAT1, ADC_DAT2, and ADC_DAT3 in turn.

The trigger event can be triggered by the external MCPWM/UTimer timing signals TADC [0], TADC [1], TADC [2], TADC [3] to a preset number of times, or triggered by software.

Each sampled signal source is set and selected by the signal source register ADC_CHN0/1/2/3. The signal source should be selected before the trigger and shall remain unchanged before the sampling completion.

It will enter the idle state and generate a sampling completion interrupt upon the completion of one-stage sampling.

Take the single-stage sampling triggered by MCPWM as an example, set that the event is triggered when the TADC [2] occurs four times. The state transition is shown in Fig. 9-3 State Transition Diagram of ADC Single-stage Sampling.

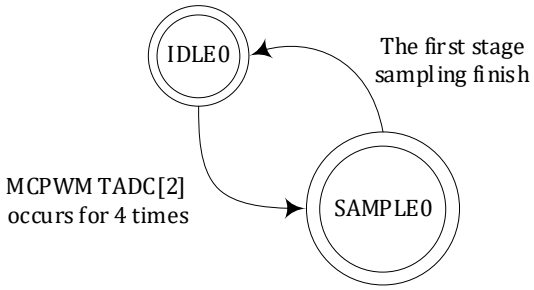


Fig. 9-5 State Transition Diagram of ADC Single-stage Sampling

9.3.1.2 Two-stage Trigger Mode

Two-stage trigger requires two triggers to complete one stage of sampling. The first stage is sampled when the first trigger arrives and the second stage is sampled when the second trigger arrives.

Assuming that the number of two sampling configuration channels is 2 and 3 respectively, the data after sampling and conversion of the first stage will be filled into ADC_DAT0 and ADC_DAT1 in turn, and the data after sampling and conversion of the second stage will be filled into ADC_DAT2, ADC_DAT3, and ADC_DAT4 in turn.

The trigger event can be triggered by external MCPWM timing signals TADC [0] and TADC [1] or triggered twice by software.

When TADC [0] or software trigger occurs, the 0th to 3rd ADC_CHNT sampling starts. And then, it will enter the idle state upon sampling completion and wait for the next trigger signal; When TADC [1] or software trigger occurs as the second trigger signal, the 4th to 7th ADC_CHNT sampling starts. The sampling times are set by the ADC_CHNT register.

Each sampled signal source is set and selected by the register. The signal source should be

selected before the trigger and shall remain unchanged before the sampling completed.

Software trigger has a lower priority than hardware trigger. If a software trigger occurs during the hardware-triggered sampling, the state machine will not process it, but generates an error interrupt. I.e., the sampling request triggered by the software will be processed only when the state machine is in the idle state. If software trigger is required, the hardware trigger should be turned off in advance. Then write a 0x5AA5 to the ADC_SWT register to generate a software trigger.

Take the two-stage sampling triggered by two software triggers as an example, the state transition is as shown in Fig. 9-4.

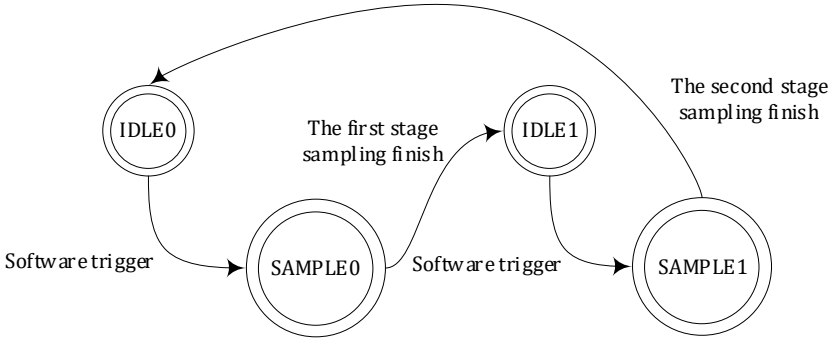


Fig. 9-6 State Transition Diagram of ADC Two-stage Sampling

9.3.1.3 Four-stage Trigger Mode

Similar to the two-stage sampling trigger, the four trigger sources are TADC [0], TADC [1], TADC [2] and TADC [3], and they should be triggered sequentially by the four-stage sampling of MCPWM TADC [0]/TADC [1]/TADC [2]/TADC [3]; or it can be triggered by four software triggers. The sampling times of the first, second, third and fourth sampling in the four-stage sampling are ADC_CHNT1[12:8], ADC_CHNT1[4:0], ADC_CHNT0[12:8], ADC_CHNT0[4:0], respectively. Taking the four-stage sampling triggered by the MCPWM TADC [0]/TADC [1]/TADC [2]/TADC [3] as an example, the state transition is as shown in Fig. 9-5 State Transition Diagram of ADC Four-stage Sampling.

Assuming that the number of four-stage sampling configuration channels are 2, 3, 1, and 5, respectively, the data after sampling and conversion of the first stage will be filled in ADC0_DAT0, ADC0_DAT1 in turn, and the data after sampling and conversion of the second stage will be filled in ADC0_DAT2, ADC0_DAT3, ADC0_DAT4, the data after sampling and conversion of the third stage will be filled into ADC0_DAT5, and the data after sampling and conversion of the fourth stage will be filled into ADC0_DAT6, ADC0_DAT7, ADC0_DAT8, ADC0_DAT9, ADC0_DAT10 in turn.

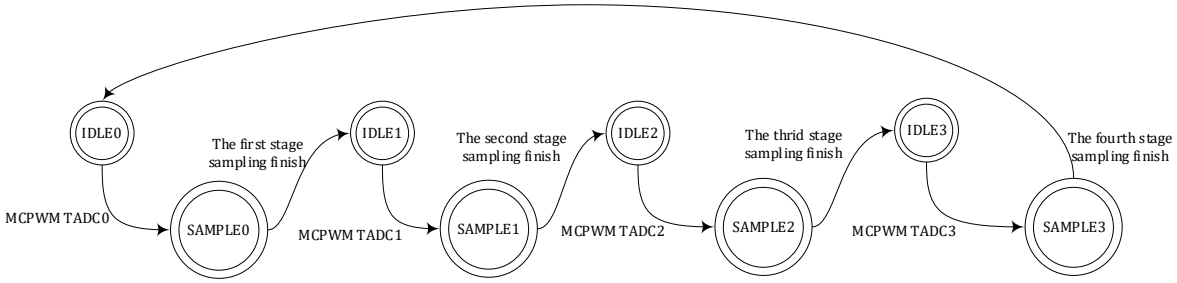


Fig. 9-7 State Transition Diagram of ADC Four-stage Sampling.

Before using the MCPWM timer to generate the ADC sampling trigger signals, MCPWM_TMR0/ MCPWM_TMR1/MCPWM_TMR2/ MCPWM_TMR3 registers should be set in advance, which correspond to the MCPWM counter value when TADC0/1/2/3 occurs. The MCPWM_TH should also be set at the same time. Set the counter count range and MCPWM_TCLK, and then set the count clock frequency and enable the clock.

9.3.1.4 ADC_IF trigger Time

The CPU software runs in the PLL clock domain and the ADC module runs in the ADC clock domain. Typically, the ADC clock cycle is twice the PLL clock cycle (bus clock cycle). Clearing ADC IF requires operating registers in the ADC clock domain and takes up to 20 bus cycles. It is recommended that the ADC IF flag be read clear at the interrupt function entrance. If the clear statement is at the end of the interrupt function, the flag may not be cleared after the interrupt function exits, causing the interrupt to enter again.

Similarly, IF the software queries the ADC IF immediately after clearing the ADC IF, the ADC IF may remain the original value.

9.3.2 Interrupt

9.3.2.1 Interrupt of Single-stage Trigger Sampling

An interrupt is triggered when sampling is completed.

9.3.2.2 Interrupt of Two-stage Trigger Sampling

An interrupt is triggered when the first-stage sampling is completed, and another interruption will be triggered when the second-stage sampling is completed.

9.3.2.3 Interrupt of Four-stage Trigger Sampling

One interrupt is triggered when the first-stage, second-stage, third-stage and fourth-stage sampling are completed accordingly.

9.3.3 Configuration Modification

It is recommended to configure and modify ADC_CHNx in the ADC interrupt. After entering the ADC interrupt, it means that the ADC has completed one stage of sampling and is now in an idle state. Since the ADC operating status cannot be confirmed in the main program, the ADC trigger should be

turned off in advance if the ADC_CHNx and ADC_CHNT registers should be modified in the main program, and then write "1" to ADC_CFG[11] to reset the ADC interface circuit state machine, thus ensuring that the ADC is not in a working state. If the ADC settings change during operation, the subsequent behavior will be unpredictable.

The sample program is as follows:

```
ADCx_CFG_temp = ADCx_CFG;

ADCx_CFG = 0x0000;

ADCx_CFG = 0x0800;

/*

    Add your code below, like:

ADCx_CHNT = 0x0005

ADCx_CHN0 = 0x3210;

ADCx_CHN1 = 0x7654;

*/

ADCx_CFG = ADCx_CFG_temp;
```

9.3.4 Select the Corresponding Analog Channel

For the channel corresponding to the signal sampled by the ADC, please refer to Table 2.2 Pin Function Selection in the DATASHEET. Turn off the corresponding IO IE and OE to use the function.

10 General Timer

10.1 Introduction

10.1.1 Functional Block Diagram

The universal timer UTIMER mainly includes four independent Timers, which can be independently configured to run the count clock and filter constant. Each Timer can be used to output a waveform with a specific duty cycle, or it can capture an external waveform to detect the duty cycle.

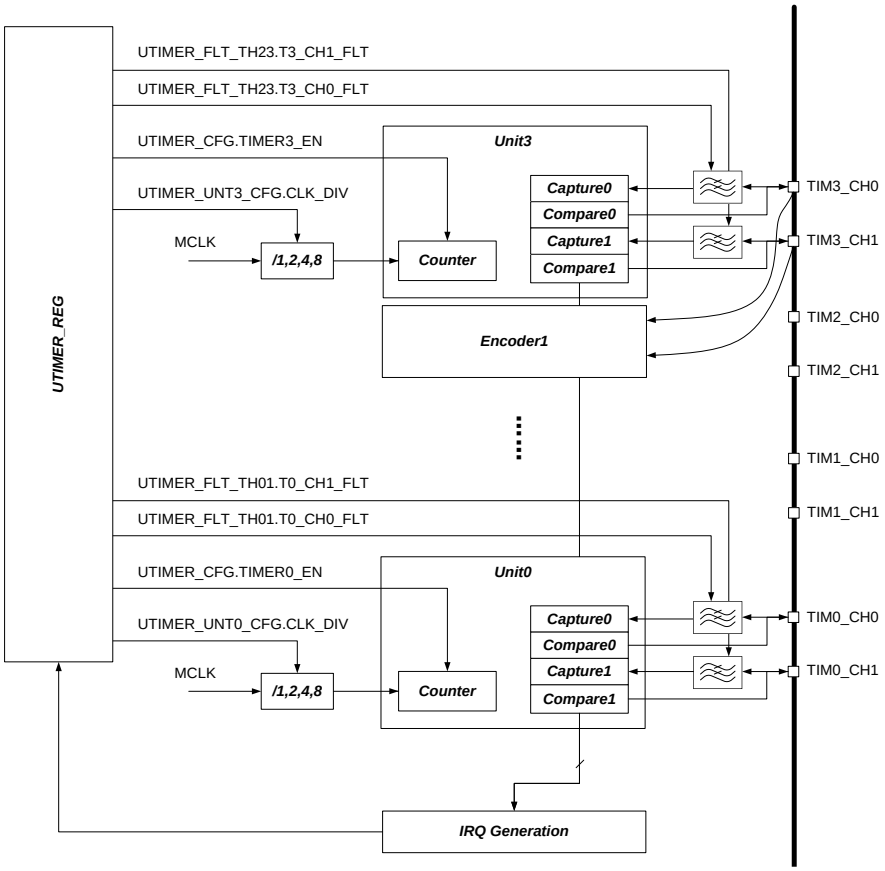


Fig. 10-1 Block Top Functional Block Diagram

10.1.1.1 Bus Interface Module

The bus interface module includes:

Translate the access signals from the AHB bus into register read and write signals, control the clock of the register module, and initiate read and write to the register module.

CG clock gating module. When the AHB bus has no access, turn off the register module clock to reduce power consumption.

10.1.1.2 Register Module

Register module (utimer_reg) for realizing:

Read and write control registers of each submodule.

Access to the status and result registers of each submodule.

Interrupt signal processing and interrupt generation for each submodule.

10.1.1.3 IO Filter Module

The IO filter module filters the input signal from outside the chip to reduce the effect of glitches on the timer.

10.1.1.4 General Timer Module

The utimer_unt module implements general timer functions, including comparison and capture modes, which can process two external input signals or generate two pulse signals to be transmitted outside the chip. The timer module includes a total of four independent general timers. The bit width of Timer0/Unit0 and Timer1/Unit1 is 16-bit, and the bit width of Timer2/Unit2 and Timer3/Unit3 is 32-bit. Each timer has two channels.

utimer_unt module, support external events to start counting, and the source of external events can be configured. When an external event is triggered, the utimer_unt timer starts to increment.

10.1.1.5 Clock Divider Module

The clock frequency dividing module is used to generate various signals of clock frequency dividing.

10.1.2 Features

The timer module has the following features:

- Available for working in at different frequencies independently
- Timer0 and Timer1 are 16-bit general timers
- Timer2 and Timer3 are 32-bit general timers
- Each general timer processes two external input signals (capture mode) or generates two output signals (comparison mode)
- Available for filtering each input signal of up to 120 main clocks, i.e., when the chip works at a clock frequency of 96MHz, it can filter out glitches below 1.25uS width.

10.2 Implementation Description

10.2.1 Clock Divider

Each timer works at the main system frequency, and the frequency division counter is adopted to reduce the counting frequency, thus to realize the independent frequency division of each timer for better write interrupt/count value.

10.2.2 Interrupt Flag Clear

The design of writing 1 into each interrupt flag bit to clear the flag bit is adopted.

10.2.3 Filtering

The timer module has a total of 8/4 pairs of channel inputs, and the timer can filter each input to varying degrees.

The filter width can be adjusted by setting the filter register, 0 ~ 120 system clock widths are available.

The original input signal was reversed at several moments from t1 to t6, and the filter width was set as T. It can be seen that only the inversions that occur at times t3 and t6 are maintained for a time greater than T, and we can see from the filter output that the signal has only inverted twice.

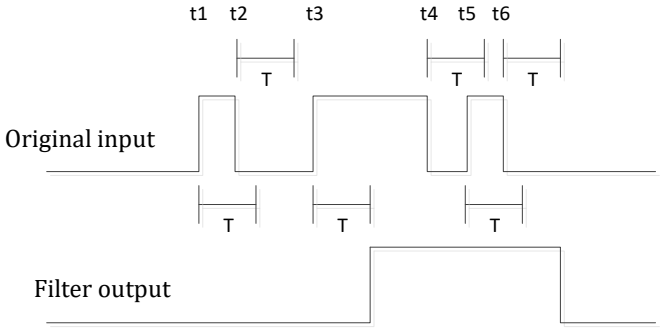


Fig. 10-2 Filter Diagram

10.2.4 Mode

10.2.4.1 Counter

The counter in Timer counts in increasing direction.

The counter counts from 0 to the TH value, and then returns to 0 to restart counting. When the counter returns to 0, a zero return interrupt is generated. The actual counting period is (TH+1) / clk_freq, where clk_freq specifies the clock frequency of the timer.

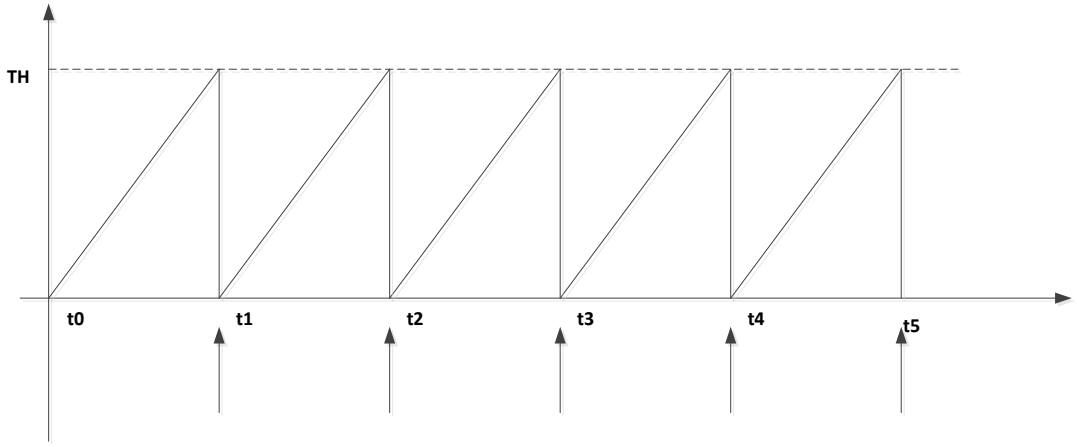
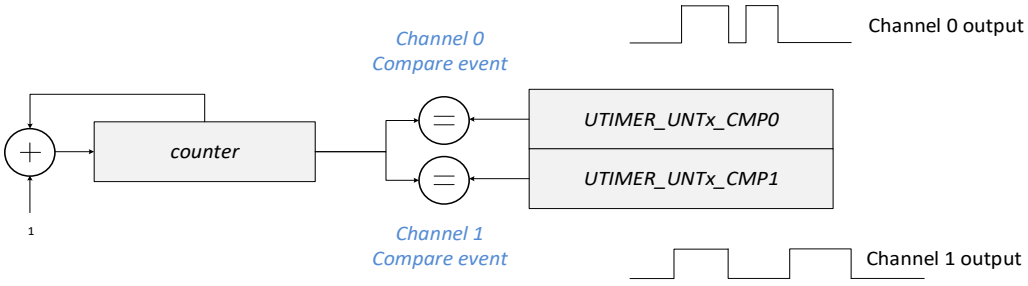


Fig. 10-3 General Counter

10.2.4.2 Comparison Mode

In comparison mode, a compare interrupt is generated when the counter counts to the UTIMER_UNTx_CMP value. The comparison mode can drive a compare pulse. When returning to zero, it will output a level to the IO port (polarity can be configured); When the comparison event occurs, the level is inverted, and another level is output to the IO port. When the counter returns to zero, the zero return interrupt will still be generated. Set UTIMER_UNTx_CMP0=0 to make the channel 0 of Timer X always 1; set UTIMER_UNTx_CMP0=UTIMER_UNTx_TH+1 to make the channel 0 of Timer X always 0.



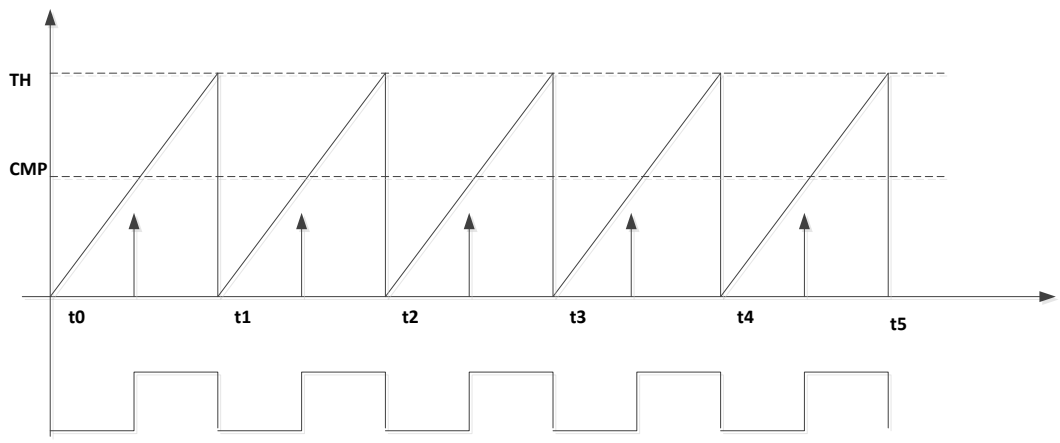


Fig. 10-4 Comparison Mode

10.2.4.3 Capture Mode

Timer can detect the rising/falling or double edge of the input signal in the capture mode. When a capture event (i.e., the input signal level changes) occurs, the timer count value is stored in the UTIMER_UNTx_CMP register and a capture interrupt is generated. When the counter returns to zero, the zero return interrupt will still be generated.

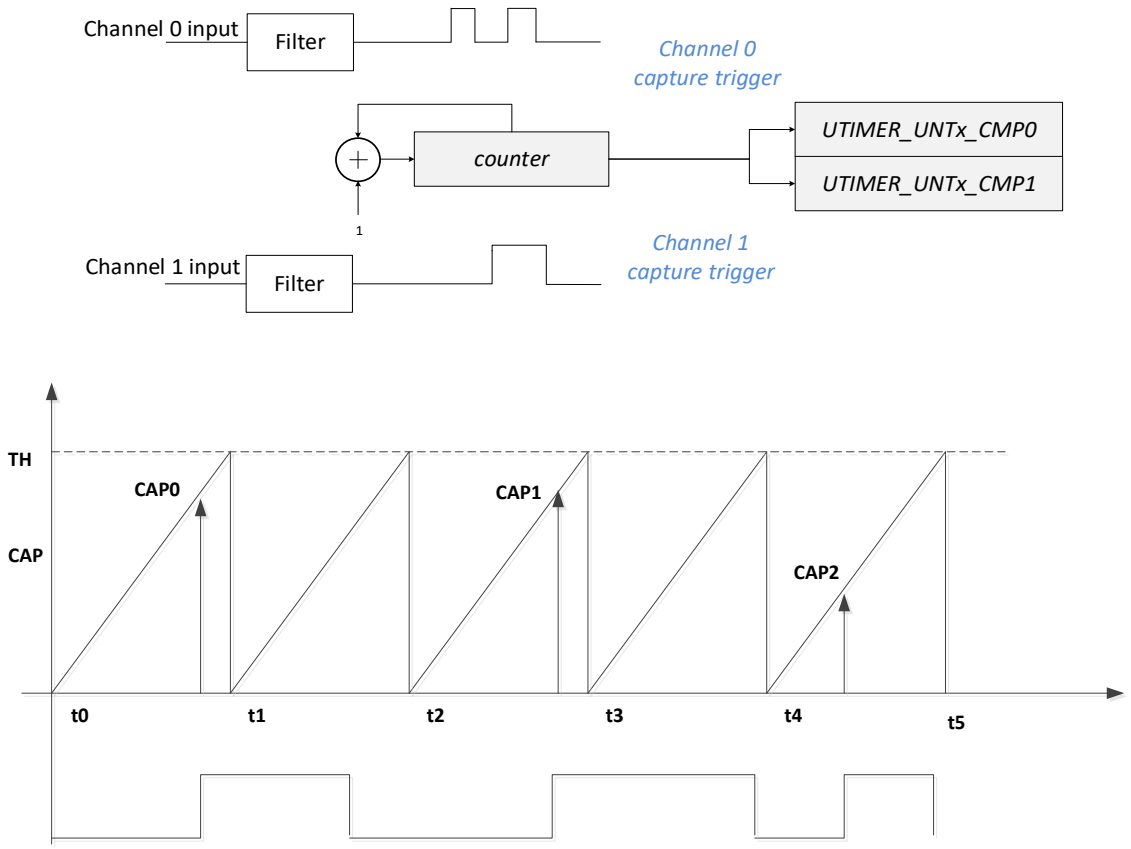


Fig. 10-5 Capture Mode

As shown in Figure 10-5, the timer is set to capture on the rising edge. When changes in the rising edge of the input signal is captured at timing CAP0, CAP1 and CAP2, the timer count value at the corresponding time will be stored in the UTIMER_UNTx_CMP register.

10.3 Register

10.3.1 Address Allocation

The base address of the general timer module is 0x40011800.

Table 10-1 Address Allocation of General Timer Configuration Register

Name	Offset	Description
UTIMER_UNT0_CFG	0x00	Timer0 Configuration Register
UTIMER_UNT0_TH	0x04	Timer0 count threshold register
UTIMER_UNT0_CNT	0x08	Timer0 count value register
UTIMER_UNT0_CMP0	0x0C	Timer0 compare/capture register 0
UTIMER_UNT0_CMP1	0x10	Timer0 compare/capture register 1
UTIMER_UNT0_EVT	0x14	Timer0 external event selection register
UTIMER_UNT1_CFG	0x20	Timer1 configuration register
UTIMER_UNT1_TH	0x24	Timer1 count threshold register
UTIMER_UNT1_CNT	0x28	Timer1 count value register
UTIMER_UNT1_CMP0	0x2C	Timer1 compare/capture register 0
UTIMER_UNT1_CMP1	0x30	Timer1 compare/capture register 1
UTIMER_UNT1_EVT	0x34	Timer1 external event selection register
UTIMER_UNT2_CFG	0x40	Timer2 configuration register
UTIMER_UNT2_TH	0x44	Timer2 count threshold register
UTIMER_UNT2_CNT	0x48	Timer2 count value register
UTIMER_UNT2_CMP0	0x4C	Timer2 compare/capture register 0
UTIMER_UNT2_CMP1	0x50	Timer2 compare/capture register 1
UTIMER_UNT2_EVT	0x54	Timer2 external event selection register
UTIMER_UNT3_CFG	0x60	Timer3 configuration register
UTIMER_UNT3_TH	0x64	Timer3 count threshold register
UTIMER_UNT3_CNT	0x68	Timer3 count value register
UTIMER_UNT3_CMP0	0x6C	Timer3 compare/capture register 0
UTIMER_UNT3_CMP1	0x70	Timer3 compare/capture register 1
UTIMER_UNT3_EVT	0x74	Timer3 external event selection register
UTIMER_FLT_TH01	0xA0	Filter Threshold Register 01
UTIMER_FLT_TH23	0xA4	Filter Threshold Register 23
UTIMER_CFG	0xF0	General timer configuration register
UTIMER_IE	0xF4	Interrupt enabling register
UTIMER_IF	0xF8	Interrupt flag register

10.3.2 System Control Register

10.3.2.1 UTIMER_CFG

Address: 0x4001_18F0

Reset value: 0x0

Table 10-2 UTIMER_CFG UTIMER Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TIMER3_EN	TIMER2_EN	TIMER1_EN	TIMER0_EN				
								RW	RW	RW	RW				
								0	0	0	0				

Location	Bit name	Description
[31:8]		Unused
[7]	TIMER3_EN	Timer3 enable. When TIMER3_EN is 0, timer3 stops counting, and all interrupt flags output 0 at the same time.
[6]	TIMER2_EN	Timer2 enable. When TIMER2_EN is 0, timer2 stops counting, and all interrupt flags output 0 at the same time.
[5]	TIMER1_EN	Timer1 enable. When TIMER1_EN is 0, timer1 stops counting, and all interrupt flags output 0 at the same time.
[4]	TIMER0_EN	Timer0 enable. When TIMER0_EN is 0, timer0 stops counting, and all interrupt flags output 0 at the same time.
[3:0]	RESERVED	Reserved bit, better write 0

10.3.3 Filter Control Register

10.3.3.1 UTIMER_FLT_TH01

Address: 0x4001_18A0

Reset value: 0x0

Table 10-3 UTIMER_FLT_TH01 Filter Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T1_CH1_FLT				T1_CH0_FLT				T0_CH1_FLT				T0_CH0_FLT			
RW				RW				RW				RW			
0				0				0				0			

Location	Bit name	Description
[31:16]		Unused
[15:12]	T1_CH1_FLT	TIM1_CH1 signal filter width selection, value range 0 ~ 15. [15:12] When it's 0, TIM1_CH1 is not filtered. [15:12] When it is not 0, the TIM1_CH1 signal is filtered: the filter width is T1_CH1_FLT×8. When the TIM1_CH1 level is stable beyond the width of T1_CH1_FLT×8 system clock cycles, the filter output is updated to the TIM1_CH1 signal value; otherwise, the filter keeps the current output unchanged.
[11:8]	T1_CH0_FLT	TIM1_CH0 signal filter width selection. Value range:0 ~ 15. [11:08] When it is 0, TIM1_CH0 is not filtered. [11:08] When it is not 0, the TIM1_CH0 signal is filtered: the filter width is T1_CH0_FLT×8. When the TIM1_CH0 level is stable beyond the width of T1_CH0_FLT×8 system clock cycles, the filter output is updated to the TIM1_CH0 signal value; otherwise, the filter keeps the current output unchanged.
[7:4]	T0_CH1_FLT	TIM0_CH1 signal filter width selection. Value range:0 ~ 15.

		[07:04] When it's 0, TIM0_CH1 is not filtered. [07:04] When it is not 0, the TIM0_CH1 signal is filtered: the filter width is T0_CH1_FLT×8. When the TIM0_CH1 level is stable beyond the width of T0_CH1_FLT×8 system clock cycles, the filter output is updated to the TIM0_CH1 signal value; otherwise, the filter keeps the current output unchanged.
[3:0]	T0_CH0_FLT	TIM0_CH0 signal filter width selection. Value range:0 ~ 15. [03:00] When it's 0, TIM0_CH0 is not filtered. [03:00] When it is not 0, the TIM0_CH0 signal is filtered: the filter width is T0_CH0_FLT×8. When the TIM0_CH0 level is stable beyond the width of T0_CH0_FLT×8 system clock cycles, the filter output is updated to the TIM0_CH0 signal value; otherwise, the filter keeps the current output unchanged.

10.3.3.2 UTIMER_FLT_TH23

Address: 0x4001_18A4

Reset value: 0x0

Table 10-4 UTIMER_FLT_TH23 Filter Control Register

15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
T3_CH1_FLT	T3_CH0_FLT	T2_CH1_FLT	T2_CH0_FLT
RW	RW	RW	RW
0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15:12]	T3_CH1_FLT	TIM3_CH1 signal filter width selection, value range 0 ~ 15. [15:12] When it is 0, TIM3_CH1 is not filtered. [15:12] When it is not 0, the TIM3_CH1 signal is filtered: the filter width is T3_CH1_FLT×8. When the TIM3_CH1 level is stable beyond the width of T3_CH1_FLT×8 system clock cycles, the filter output is updated to the TIM3_CH1 signal value; otherwise, the filter keeps the current output unchanged.
[11:8]	T3_CH0_FLT	TIM3_CH0 signal filter width selection. Value range:0 ~ 15. [11:08] When it is 0, TIM3_CH0 is not filtered. [11:08] When it is not 0, the TIM3_CH0 signal is filtered: the filter width is T3_CH0_FLT×8. When the TIM3_CH0 level is stable beyond the width of T3_CH0_FLT×8 system clock cycles, the filter output is updated to the TIM3_CH0 signal value; otherwise, the filter keeps the current output unchanged.
[7:4]	T2_CH1_FLT	TIM2_CH1 signal filter width selection. Value range:0 ~ 15. [07:04] When it's 0, TIM2_CH1 is not filtered. [07:04] When it is not 0, the TIM2_CH1 signal is filtered: the filter width is T2_CH1_FLT×8. When the TIM2_CH1 level is stable beyond the width of T2_CH1_FLT×8 system clock cycles, the filter output is updated to the TIM2_CH1 signal value; otherwise, the filter keeps the current output unchanged.
[3:0]	T2_CH0_FLT	TIM2_CH0 signal filter width selection. Value range:0 ~ 15. [03:00] When it's 0, TIM2_CH0 is not filtered. [03:00] When it is not 0, the TIM2_CH0 signal is filtered: the filter

		width is T2_CH0_FLT×8. When the TIM2_CH0 level is stable beyond the width of T2_CH0_FLT×8 system clock cycles, the filter output is updated to the TIM2_CH0 signal value; otherwise, the filter keeps the current output unchanged.

Note that the working clock of the above filter is the same clock as the divided working clock of the corresponding Timer, which is controlled by the frequency division factor of UTIMER_UNTx_CFG[9:8].

Suppose UTIMER_FLT_TH23[11:8] = 0x6; UTIMER_UNT3_CFG[9:8] = 0x2; then the running clock of Timer3 is divided by 4 in frequency relative to the system clock. The channel 0 input signal of Timer3 needs to be filtered by 8×6 times the running clock of Timer3, or 8×6×4 times the system clock.

10.3.4 Timer Register

Timer0 and Time1 are the same. Here the example given is Timer0 register.

Timer2 and Timer3 are the same. The difference with Timer0 and Timer1 is that the related registers of Timer2 and Timer3 counter are 32 bits wide, while the related registers of Timer0 and Timer1 counter are 16 bits wide.

10.3.4.1 UTIMER_UNT0_CFG Timer0 Configuration Register

Address: 0x4001_1800

Reset value: 0x0

Table 10-5 UTIMER_UNT0_CFG Timer0 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONE_TRIG	SRC1	SRC0	ETON		CLK_DIV		CH1_POL	CH1_MODE	CH1_FE_CAP_EN	CH1_RE_CAP_EN	CH0_POL	CH0_MODE	CH0_FE_CAP_EN	CH0_RE_CAP_EN
	RW	RW	RW	RW		RW		RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0		0		0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:15]		Unused
[14]	ONE_TRIG	In comparison mode, when UTIMER_CFG[4] is 0, writing 1 triggers Timer0 to send a pulse with a specific duty cycle. This bit is 1 during the pulse sending period. After a Timer period, it is automatically cleared.
[13]	SRC1	Signal source of Timer0 capture mode channel 1. The default value is 0.

		0: Chip GPIO (see Datasheet and application configuration) 1: Output of comparator 1
[12]	SRC0	Signal source of Timer0 capture mode channel 0. The default value is 0. 0: Chip GPIO (see Datasheet and application configuration) 1: Output of comparator 0
[11]	ETON	Timer0 counter count enable configuration. The default value is 0. 0, Auto run, counting continuously; 1: Wait for external event to trigger counting, which stops after a Timer cycle. External event could be selected by setting UTIMER_UNT0_EVT
[10]		Unused
[9:8]	CLK_DIV	Timer0 counter frequency configuration. The counting frequency of the counter is divided by 1/2/4/8 of the main clock frequency: The default value is 0, no frequency division. 00: 1 frequency division, 01: 2 frequency division, 10: 4 frequency division, 11: 8 frequency division
[7]	CH1_POL	Timer0 Channel 1 output polarity control in the comparison mode: the output value when the counter count value returns to zero.
[6]	CH1_MODE	Timer0 channel 1 working mode selection, the default value is 0. 0: Comparison mode. Output square wave; IO is reversed when the channel 1 counter count value reaches 0 or the compare capture register value of Timer 0. 1: Capture mode. When a capture event occurs on the channel 1 input signal of Timer 0, the counter count value is stored in the channel 1 compare capture register of Timer 0.
[5]	CH1_FE_CAP_EN	Timer0 channel 1 falling edge capture event enable. 1: Enable; 0: Disable. A 1→0 transition on the Timer 0 channel 1 input signal is considered a capture event. Falling edge event enable can coexist with rising edge event enable.
[4]	CH1_RE_CAP_EN	Timer 0 Channel 1 rising edge capture event enable. 1: Enable; 0: Disable. A 0→1 transition on the Timer 0 channel 1 input signal is considered a capture event. Rising edge event enable can coexist with falling edge event enable.
[3]	CH0_POL	Timer 0 channel 0 output polarity control in compare mode: the output value when the counter count value returns to zero.
[2]	CH0_MODE	Timer0 channel 0 working mode selection, the default value is 0. 0: Comparison mode. Output square wave; IO is reversed when the Timer 0 channel 0 counter count value reaches 0 or the compare capture register value of Timer 0. 1: Capture mode. When a capture event occurs on the channel 0 input signal of Timer 0, the counter count value is stored in the channel 0 compare capture register of Timer 0.
[1]	CH0_FE_CAP_EN	Timer0 channel 0 falling edge capture event enable. 1: Enable; 0: Disable. A 1→0 transition on the Timer 0 channel 0 input signal is considered a capture event. Falling edge event enable can coexist with rising edge event enable.
[0]	CH0_RE_CAP_EN	Timer 0 Channel 0 rising edge capture event enable. 1: Enable; 0: Disable. A 0 →1 transition on the Timer 0 channel 0 input signal is considered a capture event. Rising edge event enable can coexist with falling edge event enable.

10.3.4.2 UTIMER_UNT1_CFG Timer1 Configuration Register

Address: 0x4001_1820

Reset value: 0x0

Table 10-6 UTIMER_UNT1_CFG Timer 1 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONE_TRIG	SRC1	SRC0	ETON		CLK_DIV		CH1_POL	CH1_MODE	CH1_FE_CAP_EN	CH1_RE_CAP_EN	CH0_POL	CH0_MODE	CH0_FE_CAP_EN	CH0_RE_CAP_EN
	RW	RW	RW	RW		RW		RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0		0		0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:15]		Unused
[14]	ONE_TRIG	In comparison mode, when UTIMER_CFG[5] is 0, writing 1 triggers Timer1 to send a pulse with a specific duty cycle. This bit is 1 during the pulse sending period. After a Timer period, it is automatically cleared.
[13]	SRC1	Signal source of Timer1 capture mode channel 1. The default value is 0. 0: Chip GPIO (see Datasheet and application configuration) 1: Output of comparator 1
[12]	SRC0	Signal source of Timer1 capture mode channel 0. The default value is 0. 0: Chip GPIO (see Datasheet and application configuration) 1: Output of comparator 0
[11]	ETON	Timer1 counter count enable configuration. The default value is 0. 0, Auto run, counting continuously; 1: Wait for external event to trigger counting, which stops in a Timer cycle. External event could be selected by setting UTIMER_UNT1_EVT.
[10]		Unused
[9:8]	CLK_DIV	Timer1 counter frequency configuration. The counting frequency of the counter is divided by 1/2/4/8 of the main clock frequency: The default value is 0, no frequency division. 00: 1 frequency division, 01: 2 frequency division, 10: 4 frequency division, 11: 8 frequency division
[7]	CH1_POL	Timer1 Channel 1 output polarity control in the comparison mode: the output value when the counter count value returns to zero.
[6]	CH1_MODE	Timer1 channel 1 working mode selection, the default value is 0. 0: Comparison mode. Output square wave; IO is reversed when the timer 1 channel 1 counter count value reaches 0 or the compare capture register value of Timer 1. 1: Capture mode. When a capture event occurs on Timer 1 channel 1 input signal, the counter count value is stored in the Timer 1 channel

		1 compare capture register.
[5]	CH1_FE_CAP_EN	Timer1 channel 1 falling edge capture event enable. 1: Enable; 0: Disable. A 1→ 0 transition on the Timer 1 channel 1 input signal is considered a capture event. Falling edge event enable can coexist with rising edge event enable.
[4]	CH1_RE_CAP_EN	Timer1 channel 1 rising edge capture event enable. 1: Enable; 0: Disable. A 0 →1 transition on the Timer 1 channel 1 input signal is considered a capture event. Rising edge event enable can coexist with falling edge event enable.
[3]	CH0_POL	Timer 1 channel 0 output polarity control in compare mode: the output value when the counter count value returns to zero.
[2]	CH0_MODE	Timer1 channel 0 working mode selection, the default value is 0. 0: Comparison mode. Output square wave; IO is reversed when the timer 1 channel 0 counter count value reaches 0 or the compare capture register value of Timer 1. 1: Capture mode. When a capture event occurs on the channel 0 input signal of Timer 1, the counter count value is stored in the channel 0 compare capture register of Timer 1.
[1]	CH0_FE_CAP_EN	Timer1 channel 0 falling edge capture event enable. 1: Enable; 0: Disable. A 1→ 0 transition on the Timer 1 channel 0 input signal is considered a capture event. Falling edge event enable can coexist with rising edge event enable.
[0]	CH0_RE_CAP_EN	Timer1 channel 0 rising edge capture event enable. 1: Enable; 0: Disable. A 0 →1 transition on the Timer 1 channel 0 input signal is considered a capture event. Rising edge event enable can coexist with falling edge event enable.

10.3.4.3 UTIMER_UNT2_CFG Timer2 Configuration Register

Address: 0x4001_1840

Reset value: 0x0

Table 10-7 UTIMER_UNT2_CFG Timer 2 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONE_TRIG	SRC1	SRC0	ETON		CLK_DIV		CH1_POL	CH1_MODE	CH1_FE_CAP_EN	CH1_RE_CAP_EN	CH0_POL	CH0_MODE	CH0_FE_CAP_EN	CH0_RE_CAP_EN
	RW	RW	RW	RW		RW		RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0		0		0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:15]		Unused
[14]	ONE_TRIG	In comparison mode, when UTIMER_CFG[6] is 0, writing 1 triggers Timer2 to send a pulse with a specific duty cycle. This bit is 1 during

		the pulse sending period. After a Timer period, it is automatically cleared.
[13]	SRC1	Signal source of Timer2 capture mode channel 1. The default value is 0. 0: Chip GPIO (see Datasheet and application configuration) 1: Output of comparator 1
[12]	SRC0	Signal source of Timer2 capture mode channel 0. The default value is 0. 0: Chip GPIO (see Datasheet and application configuration) 1: Output of comparator 0
[11]	ETON	Timer2 counter count enable configuration. The default value is 0. 0, Auto run, counting continuously; 1: Wait for external event to trigger counting, which stops in a Timer cycle. External event could be selected by setting UTIMER_UNT2_EVT.
[10]		Unused
[9:8]	CLK_DIV	Timer2 counter frequency configuration. The counting frequency of the counter is divided by 1/2/4/8 of the main clock frequency: The default value is 0, no frequency division. 00: 1 frequency division, 01: 2 frequency division, 10: 4 frequency division, 11: 8 frequency division
[7]	CH1_POL	Timer2 Channel 1 output polarity control in the comparison mode: the output value when the counter count value returns to zero.
[6]	CH1_MODE	Timer2 channel 1 working mode selection, the default value is 0. 0: Comparison mode. Output square wave; IO is reversed when the timer 2 channel 1 counter count value reaches 0 or the compare capture register value of Timer 2. 1: Capture mode. When a capture event occurs on Timer 2 channel 1 input signal, the counter count value is stored in the Timer 2 channel 1 compare capture register.
[5]	CH1_FE_CAP_EN	Timer2 channel 1 falling edge capture event enable. 1: Enable; 0: Disable. A 1→ 0 transition on the Timer 2 channel 1 input signal is considered a capture event. Falling edge event enable can coexist with rising edge event enable.
[4]	CH1_RE_CAP_EN	Timer2 channel 1 rising edge capture event enable. 1: Enable; 0: Disable. A 0→ 1 transition on the Timer 2 channel 1 input signal is considered a capture event. Rising edge event enable can coexist with falling edge event enable.
[3]	CH0_POL	Timer 2 channel 0 output polarity control in compare mode: the output value when the counter count value returns to zero.
[2]	CH0_MODE	Timer2 channel 0 working mode selection, the default value is 0. 0: Comparison mode. Output square wave; IO is reversed when the timer 2 channel 0 counter count value reaches 0 or the compare capture register value of Timer 2. 1: Capture mode. When a capture event occurs on the channel 0 input signal of Timer 2, the counter count value is stored in the channel 0 compare capture register of Timer 2.
[1]	CH0_FE_CAP_EN	Timer2 channel 0 falling edge capture event enable. 1: Enable; 0: Disable. A 1→ 0 transition on the Timer 2 channel 0 input signal is considered a capture event. Falling edge event enable can coexist with rising edge event enable.
[0]	CH0_RE_CAP_EN	Timer2 channel 0 rising edge capture event enable. 1: Enable; 0: Disable.

		A 0→ 1 transition on the Timer 2 channel 0 input signal is considered a capture event. Rising edge event enable can coexist with falling edge event enable.

10.3.4.4 UTIMER_UNT3_CFG Timer3 Configuration Register

Address: 0x4001_1860

Reset value: 0x0

Table 10-8 UTIMER_UNT3_CFG Timer 3 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONE_TRIG	SRC1	SRC0	ETON		CLK_DIV		CH1_POL	CH1_MODE	CH1_FE_CAP_EN	CH1_RE_CAP_EN	CH0_POL	CH0_MODE	CH0_FE_CAP_EN	CH0_RE_CAP_EN
	RW	RW	RW	RW		RW		RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0		0		0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:15]		Unused
[14]	ONE_TRIG	In comparison mode, when UTIMER_CFG[7] is 0, writing 1 triggers Timer3 to send a pulse with a specific duty cycle. This bit is 1 during the pulse sending period. After a Timer period, it is automatically cleared.
[13]	SRC1	Signal source of Timer3 capture mode channel 1. The default value is 0. 0: Chip GPIO (see Datasheet and application configuration) 1: Output of comparator 1
[12]	SRC0	Signal source of Timer3 capture mode channel 0. The default value is 0. 0: Chip GPIO (see Datasheet and application configuration) 1: Output of comparator 0
[11]	ETON	Timer3 counter count enable configuration. The default value is 0. 0, Auto run, counting continuously; 1: Wait for external event to trigger counting, which stops in a Timer cycle. External event could be selected by setting UTIMER_UNT3_EVT
[10]		Unused
[9:8]	CLK_DIV	Timer3 counter frequency configuration. The counting frequency of the counter is divided by 1/2/4/8 of the main clock frequency: The default value is 0, no frequency division. 00: 1 frequency division, 01: 2 frequency division, 10: 4 frequency division, 11: 8 frequency division
[7]	CH1_POL	Timer3 Channel 1 output polarity control in the comparison mode: the output value when the counter count value returns to zero.
[6]	CH1_MODE	Timer3 channel 1 working mode selection, the default value is 0. 0: Comparison mode. Output square wave; IO is reversed when the timer 3 channel 1 counter count value reaches 0 or the compare capture register value of Timer 3. 1: Capture mode. When a capture event occurs on Timer 3 channel 1

		input signal, the counter count value is stored in the Timer 3 channel 1 compare capture register.
[5]	CH1_FE_CAP_EN	Timer3 channel 1 falling edge capture event enable. 1: Enable; 0: Disable. A 1→ 0 transition on the Timer 3 channel 1 input signal is considered a capture event. Falling edge event enable can coexist with rising edge event enable.
[4]	CH1_RE_CAP_EN	Timer3 channel 1 rising edge capture event enable. 1: Enable; 0: Disable. A 0 →1 transition on the Timer 3 channel 1 input signal is considered a capture event. Rising edge event enable can coexist with falling edge event enable.
[3]	CH0_POL	Timer 3 channel 0 output polarity control in compare mode: the output value when the counter count value returns to zero.
[2]	CH0_MODE	Timer3 channel 0 working mode selection, the default value is 0. 0: Comparison mode. Output square wave; IO is reversed when the timer 3 channel 0 counter count value reaches 0 or the compare capture register value of Timer 3. 1: Capture mode. When a capture event occurs on the channel 0 input signal of Timer 3, the counter count value is stored in the channel 0 compare capture register of Timer 3.
[1]	CH0_FE_CAP_EN	Timer3 channel 0 falling edge capture event enable. 1: Enable; 0: Disable. A 1→ 0 transition on the Timer 3 channel 0 input signal is considered a capture event. Falling edge event enable can coexist with rising edge event enable.
[0]	CH0_RE_CAP_EN	Timer3 channel 0 rising edge capture event enable. 1: Enable; 0: Disable. A 0 →1 transition on the Timer 3 channel 0 input signal is considered a capture event. Rising edge event enable can coexist with falling edge event enable.

Among the LKS32MC05X series, Timer0/1 is 16bit; Timer2/3 is 32bit, so the following related registers of Timer2/3 are 32bit.

10.3.4.5 UTIMER_UNT0_TH Timer 0 Threshold Register

Address: 0x4001_1804

Reset value: 0x0

Table 10-9 UTIMER_UNT0_TH Timer 0 Threshold Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT0_TH															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	UNT0_TH	Timer 0 counter count threshold. The counter counts from 0 to UTIMER_UNT0_TH, and then returns to 0 to start counting.

10.3.4.6 UTIMER_UNT1_TH Timer 1 Threshold Register

Address: 0x4001_1824

Reset value: 0x0

Table 10-10 UTIMER_UNT1_TH Timer 1 Threshold Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT1_TH															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	UNT1_TH	Timer 1 counter count threshold. The counter counts from 0 to UTIMER_UNT1_TH, and then returns to 0 to start counting.

10.3.4.7 UTIMER_UNT2_TH Timer 2 Threshold Register

Address: 0x4001_1844

Reset value: 0x0

Table 10-11 UTIMER_UNT2_TH Timer 2 Threshold Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT2_TH															
RW															
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT2_TH															
RW															
0															

Location	Bit name	Description
[31:0]	UNT2_TH	Timer 2 counter count threshold. The counter counts from 0 to UTIMER_UNT2_TH, and then returns to 0 to start counting.

10.3.4.8 UTIMER_UNT3_TH Timer 3 Threshold Register

Address: 0x4001_1864

Reset value: 0x0

Table 10-12 UTIMER_UNT3_TH Timer 3 Threshold Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT3_TH															
RW															

0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT3_TH															
RW															
0															

Location	Bit name	Description
[31:0]	UNT3_TH	Timer 3 counter count threshold. The counter counts from 0 to UTIMER_UNT3_TH, and then returns to 0 to start counting.

10.3.4.9 UTIMER_UNT0_CNT Timer 0 Threshold Register

Address: 0x4001_1808

Reset value: 0x0

Table 10-13 UTIMER_UNT0_CNT Timer 0 Count Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT0_CNT															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	UNT0_CNT	The current count value of the Timer 0 counter. New count value can be written.

Note: Before writing UTIMER_UNT0_CNT, the timer clock needs to be turned on through SYS_CLK_FEN.

10.3.4.10 UTIMER_UNT1_CNT Timer 1 Count Register

Address: 0x4001_1828

Reset value: 0x0

Table 10-14 UTIMER_UNT1_CNT Timer 1 Count Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT1_CNT															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	UNT1_CNT	The current count value of the Timer 1 counter. New count value can be written.

Note: Before writing UTIMER_UNT1_CNT, the timer clock needs to be turned on through SYS_CLK_FEN.

10.3.4.11 UTIMER_UNT2_CNT Timer 2 Count Register

Address: 0x4001_1848

Reset value: 0x0

Table 10-15 UTIMER_UNT2_CNT Timer 2 Count Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT2_CNT															
RW															
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT2_CNT															
RW															
0															

Location	Bit name	Description
[31:0]	UNT2_CNT	The current count value of the Timer 2 counter. New count value can be written.

Note: Before writing UTIMER_UNT2_CNT, the timer clock needs to be turned on through SYS_CLK_FEN.

10.3.4.12 UTIMER_UNT3_CNT Timer 3 Count Register

Address: 0x4001_1868

Reset value: 0x0

Table 10-16 UTIMER_UNT3_CNT Timer 3 Count Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT3_CNT															
RW															
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT3_CNT															
RW															
0															

Location	Bit name	Description
[31:0]	UNT3_CNT	The current count value of the Timer 3 counter. New count value can be written.

Note: Before writing UTIMER_UNT3_CNT, the timer clock needs to be turned on through SYS_CLK_FEN.

10.3.4.13 UTIMER_UNT0_CMP0 Timer 0 Channel 0 Compare Capture Register

Address: 0x4001_180C

Reset value: 0x0

Table 10-17 UTIMER_UNT0_CMP0 Timer 0 Channel 0 Compare Capture Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT0_ CMP0															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	UNT0_ CMP0	When Timer 0 channel 0 works in compare mode and the counter count value is equal to UTIMER_UNT0_CMP0, a comparison event occurs. When Timer 0 channel 0 works in the capture mode, the counter count value when the capture event occurs is stored in the UTIMER_UNT0_CMP0 register.

10.3.4.14 UTIMER_UNT0_CMP1 Timer 0 Channel 1 Compare Capture Register

Address: 0x4001_1810

Reset value: 0x0

Table 10-18 UTIMER_UNT0_CMP1 Timer 0 Channel 1 Compare Capture Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT0_CMP1															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	UNT0_CMP1	When Timer 0 channel 1 works in compare mode and the counter count value is equal to UTIMER_UNT0_CMP1, a comparison event occurs. When Timer 0 channel 1 works in the capture mode, the counter count value when the capture event occurs is stored in the UTIMER_UNT0_CMP1 register.

10.3.4.15 UTIMER_UNT1_CMP0 Timer 1 Channel 0 Compare Capture Register

Address: 0x4001_182C

Reset value: 0x0

Table 10-19 UTIMER_UNT1_CMP0 Timer 1 Channel 0 Compare Capture Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT1_ CMP0															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	UNT1_ CMP0	When Timer 1 channel 0 works in compare mode and the counter count value is equal to UTIMER_UNT1_CMP0, a comparison event occurs. When Timer 1 channel 0 works in the capture mode, the counter count value when the capture event occurs is stored in the UTIMER_UNT1_CMP0 register.

10.3.4.16 UTIMER_UNT1_CMP1 Timer 1 Channel 1 Compare Capture Register

Address: 0x4001_1830

Reset value: 0x0

Table 10-20 UTIMER_UNT1_CMP1 Timer 1 Channel 1 Compare Capture Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT1_CMP1															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	UNT1_CMP1	When Timer 1 channel 1 works in compare mode and the counter count value is equal to UTIMER_UNT1_CMP1, a comparison event occurs. When Timer 1 channel 1 works in the capture mode, the counter count value when the capture event occurs is stored in the UTIMER_UNT1_CMP1 register.

10.3.4.17 UTIMER_UNT2_CMP0 Timer 2 Channel 0 Compare Capture Register

Address: 0x4001_184C

Reset value: 0x0

Table 10-21 UTIMER_UNT2_CMP0 Timer 2 Channel 0 Compare Capture Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT2_ CMP0															
RW															
0															

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

UNT2_ CMP0
RW
0

Location	Bit name	Description
[31:0]	UNT2_ CMP0	When Timer 2 channel 0 works in compare mode and the counter count value is equal to UTIMER_UNT2_CMP0, a comparison event occurs. When Timer 2 channel 0 works in the capture mode, the counter count value when the capture event occurs is stored in the UTIMER_UNT2_CMP0 register.

10.3.4.18 UTIMER_UNT2_CMP1 Timer 2 Channel 1 Compare Capture Register

Address: 0x4001_1850

Reset value: 0x0

Table 10-22 UTIMER_UNT2_CMP1 Timer 2 Channel 1 Compare Capture Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT2_CMP1															
RW															
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT2_CMP1															
RW															
0															

Location	Bit name	Description
[31:0]	UNT2_CMP1	When Timer 2 channel 1 works in compare mode and the counter count value is equal to UTIMER_UNT2_CMP1, a comparison event occurs. When Timer 2 channel 1 works in the capture mode, the counter count value when the capture event occurs is stored in the UTIMER_UNT2_CMP1 register.

10.3.4.19 UTIMER_UNT3_CMP0 Timer 3 Channel 0 Compare Capture Register

Address: 0x4001_186C

Reset value: 0x0

Table 10-23 UTIMER_UNT3_CMP0 Timer 3 Channel 0 Compare Capture Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT3_ CMP0															
RW															
0															

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

UNT3_ CMP0
RW
0

Location	Bit name	Description
[31:0]	UNT3_ CMP0	When Timer 3 channel 0 works in compare mode and the counter count value is equal to UTIMER_UNT3_CMP0, a comparison event occurs. When Timer 3 channel 0 works in the capture mode, the counter count value when the capture event occurs is stored in the UTIMER_UNT3_CMP0 register.

10.3.4.20 UTIMER_UNT3_CMP1 Timer 3 Channel 1 Compare Capture Register

Address: 0x4001_1870

Reset value: 0x0

Table 10-24 UTIMER_UNT3_CMP1 Timer 3 Channel 1 Compare Capture Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNT3_CMP1															
RW															
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNT3_CMP1															
RW															
0															

Location	Bit name	Description
[31:0]	UNT3_CMP1	When Timer 3 channel 1 works in compare mode and the counter count value is equal to UTIMER_UNT3_CMP1, a comparison event occurs. When Timer 3 channel 1 works in the capture mode, the counter count value when the capture event occurs is stored in the UTIMER_UNT3_CMP1 register.

10.3.4.21 UTIMER_UNT0_EVT Timer0 External Event Select Register

Address: 0x4001_1814

Reset value: 0x0

Table 10-25 UTIMER_UNT0_EVT Timer0 External Event Select Register

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

Location	Bit name	Description
[31:3]		Unused
[2:0]	EVT_SRC	Timer0 external event selection register. This register should be used with UTIMER_UNT0_CFG [11]. When U0CFG [11] is high, select the event that triggers Timer0 count by this register. 0: MCPWM TADC [2] comparison event 1: MCPWM TADC [3] comparison event 2: TIMER1 channel 0 comparison event 3: TIMER1 channel 1 comparison event 4: TIMER2 channel 0 comparison event 5: TIMER2 channel 1 comparison event 6: TIMER3 channel 0 comparison event 7: TIMER3 channel 1 comparison event

10.3.4.22 UTIMER_UNT1_EVT Timer1 External Event Select Register

Address: 0x4001_1834

Reset value: 0x0

Table 10-26 UTIMER_UNT1_EVT Timer1 External Event Select Register

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

Location	Bit name	Description
[31:3]		Unused
[2:0]	EVT_SRC	Timer1 external event selection register. This register should be used with UTIMER_UNT1_CFG[11]. UTIMER_UNT1_CFG[11] is high, select the event that triggers Timer1 count by this register. 0: TIMER0 channel 0 comparison event 1: TIMER0 channel 1 comparison event 2: MCPWM TADC [2] comparison event 3: MCPWM TADC [3] comparison event 4: TIMER2 channel 0 comparison event 5: TIMER2 channel 1 comparison event 6: TIMER3 channel 0 comparison event 7: TIMER3 channel 1 comparison event

10.3.4.23 UTIMER_UNT2_EVT Timer2 External Event Select Register

Address: 0x4001_1854

Reset value: 0x0

Table 10-27 UTIMER_UNT2_EVT Timer 2 External Event Select Register

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

	0

Location	Bit name	Description
[31:3]		Unused
[2:0]	EVT_SRC	Timer2 external event selection register. This register should be used with UTIMER_UNT2_CFG [11]. UTIMER_UNT2_CFG[11] is high, select the event that triggers Timer2 count by this register. 0: TIMER0 channel 0 comparison event 1: TIMER0 channel 1 comparison event 2: TIMER1 channel 0 comparison event 3: TIMER1 channel 1 comparison event 4: MCPWM TADC [2] comparison event 5: MCPWM TADC [3] comparison event 6: TIMER3 channel 0 comparison event 7: TIMER3 channel 1 comparison event

10.3.4.24 UTIMER_UNT3_EVT Timer3 External Event Selection Register

Address: 0x4001_1874

Reset value: 0x0

Table 10-28 UTIMER_UNT3_EVT Timer3 External Event Select Register

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

Location	Bit name	Description
[31:3]		Unused
[2:0]	EVT_SRC	Timer3 external event selection register. This register should be used with UTIMER_UNT3_CFG [11]. UTIMER_UNT3_CFG[11] is high, select the event that triggers Timer3 count by this register. 0: TIMER0 channel 0 comparison event 1: TIMER0 channel 1 comparison event 2: TIMER1 channel 0 comparison event 3: TIMER1 channel 1 comparison event 4: TIMER2 channel 0 comparison event 5: TIMER2 channel 1 comparison event 6: MCPWM TADC [2] comparison event 7: MCPWM TADC [3] comparison event

10.3.5 Interrupt Management Register

The interrupt management register includes the interrupt flag register UTIMER_IF and the interrupt enable register UTIMER_IE. Each bit of the two registers corresponds to the same interrupt.

10.3.5.1 UTIMER_IE Interrupt Enable Register

Address: 0x4001_18F4

Reset value: 0x0

Table 10-29 UTIMER_IE Interrupt Enable Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				T3_CH1_IE	T3_CH0_IE	T3_ZC_IE	T2_CH1_IE	T2_CH0_IE	T2_ZC_IE	T1_CH1_IE	T1_CH0_IE	T1_ZC_IE	T0_CH1_IE	T0_CH0_IE	T0_ZC_IE
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
				0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:12]		Unused
[11]	T3_CH1_IE	Timer3 channel 1 compare/capture interrupt enable. Active high.
[10]	T3_CH0_IE	Timer3 channel 0 compare/capture interrupt enable. Active high.
[9]	T3_ZC_IE	Timer3 counter over-zero interrupt enable. Active high.
[8]	T2_CH1_IE	Timer2 channel 1 compare/capture interrupt enable. Active high.
[7]	T2_CH0_IE	Timer2 channel 0 compare/capture interrupt enable. Active high.
[6]	T2_ZC_IE	Timer2 counter over-zero interrupt enable. Active high.
[5]	T1_CH1_IE	Timer1 channel 1 compare/capture interrupt enable. Active high.
[4]	T1_CH0_IE	Timer1 channel 0 compare/capture interrupt enable. Active high.
[3]	T1_ZC_IE	Timer1 counter over-zero interrupt enable. Active high.
[2]	T0_CH1_IE	Timer0 channel 1 compare/capture interrupt enable. Active high.
[1]	T0_CH0_IE	Timer0 channel 0 compare/capture interrupt enable. Active high.
[0]	T0_ZC_IE	Timer0 counter over-zero interrupt enable. Active high.

10.3.5.2 UTIMER_IF Interrupt Flag Register

Address: 0x4001_18F8

Reset value: 0x0

Table 10-30 UTIMER_IF Interrupt Flag Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				T3_CH1_IF	T3_CH0_IF	T3_ZC_IF	T2_CH1_IF	T2_CH0_IF	T2_ZC_IF	T1_CH1_IF	T1_CH0_IF	T1_ZC_IF	T0_CH1_IF	T0_CH0_IF	T0_ZC_IF

	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:12]		Unused
[11]	T3_CH1_IF	Timer3 channel 1 compare/capture interrupt flag. Active high. Write 1 to clear this bit.
[10]	T3_CH0_IF	Timer3 channel 0 compare/capture interrupt flag. Active high. Write 1 to clear this bit.
[9]	T3_ZC_IF	Timer3 counter over-zero interrupt flag. Active high. Write 1 to clear this bit.
[8]	T2_CH1_IF	Timer2 channel 1 compare/capture interrupt flag. Active high. Write 1 to clear this bit.
[7]	T2_CH0_IF	Timer2 channel 0 compare/capture interrupt flag. Active high. Write 1 to clear this bit.
[6]	T2_ZC_IF	Timer2 counter over-zero interrupt flag. Active high. Write 1 to clear this bit.
[5]	T1_CH1_IF	Timer1 channel 1 compare/capture interrupt flag. Active high. Write 1 to clear this bit.
[4]	T1_CH0_IF	Timer1 channel 0 compare/capture interrupt flag. Active high. Write 1 to clear this bit.
[3]	T1_ZC_IF	Timer1 counter over-zero interrupt flag. Active high. Write 1 to clear this bit.
[2]	T0_CH1_IF	Timer0 channel 1 compare/capture interrupt flag. Active high. Write 1 to clear this bit.
[1]	T0_CH0_IF	Timer0 channel 0 compare/capture interrupt flag. Active high. Write 1 to clear this bit.
[0]	T0_ZC_IF	Timer0 counter over-zero interrupt flag. Active high. Write 1 to clear this bit.

The comparison event of Timer2 channel 0/1 and Timer3 channel 0/1 can be used as ADC sampling trigger events UTimer_T0/ UTimer_T1 /UTimer_T2/UTimer_T3;

MCPWM_T0, MCPWM_T1, MCPWM_T2, and MCPWM_T3 generated by MCPWM are enabled controlled within the ADC, along with these four events to obtain four ADC sampling trigger events TADC[0]/TADC[1]/TADC[2]/TADC[3].

Write 1 to clear the interrupt flag. Generally, it is not recommended to clear with the following |= method, because |= is the first reading interrupt flag. Modify the corresponding bit to 1 and then re-write it to clear. If other interrupt flags are set at the same time, they will be cleared together, which is not expected by the software. For example, the following writing method is intended to clear T0_ZC_IF, but if T0_CH0_IF is set to 1 before the write execution at the same time, the software will firstly read back the UTIMER_IF value 0x2, and then execute the OR operation 0x2|0x1=0x3, write and clear T0_CH0_IF and T0_ZC_IF at the same time, which may cause the Timer to enter one less interrupt due to capture.

UTIMER_IF|=0x1;

If you hope to clear the T0_ZC_IF flag bit, directly write 1 to BIT0 as follows.

UTIMER_IF=0x1;

11 HALL Signal Processing Module

11.1 Introduction

The chip supports three HALL signal inputs.

The processing of the input HALL sensor signal includes:

Filtering. Eliminate the effect of HALL signal glitches.

Capture. When the HALL input changes, record the current timer value and output an interrupt.

Overflow. When the HALL signal remains changed for a long time, resulting in the counter overflows, an interrupt is output.

11.2 Implementation Description

11.2.1 Signal Source

The HALL signal comes from GPIO. For each circuit of Hall signal, the chip has two IPs as the source of such signal. Users can choose to use one of the GPIO input signals as the HALL signal by setting the GPIO register.

Please see DATASHEET for detailed pin location.

11.2.2 Working Clock

The working frequency of HALL module is adjustable. Users can select the 1/2/4/8 frequency division of the main clock as the operating frequency of the HALL module by setting the HALL_CFG.CLK_DIV register. Both filtering and counting work at this frequency.

11.2.3 Signal Filtering

The filter module is mainly used to remove glitches on the HALL signal.

The filtering includes two stages of filters. The two-stage filtering circuit can be turned on independently or fully:

In the first stage, the "5 in 7" rule is used for filtering, i.e., if five "1" is reached while filtering the seven consecutive sampling points, output as "1"; if reached or over five "0", output as "0"; otherwise, the output result would remain unchanged as the last filtering. You can select whether to enable the first-stage filter by setting HALL_CFG.FIL_75. As shown in the figure below:

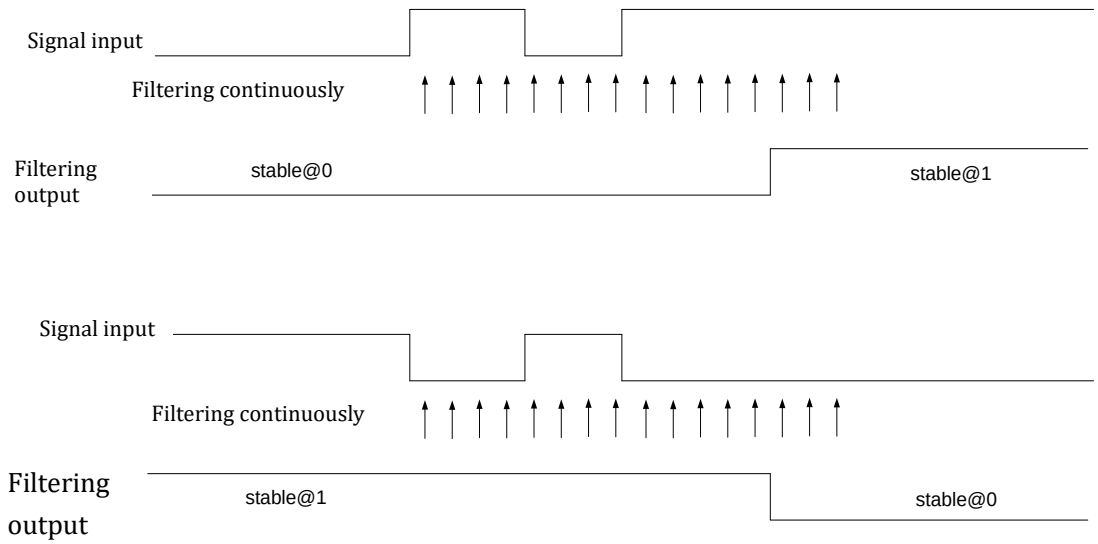


Fig. 11-1 7/5 Filter Module Block Diagram

In the second stage, continuous filtering is adopted. If all results filtered are zero while filtering N consecutive sampling points, output as "0"; if all results filtered are "1", output as "1"; otherwise, the output result would remain unchanged as the previous filtering.

Select the filtering depth of the second-stage filter by setting HALL_CFG.FIL_LEN, i.e., the number of consecutive samples. The maximum number of consecutive samples is 2^{15}, and the calculation formula of the filter time constant is as follows:

$$T_{fit} = T_{clk} * (HALL\_CFG.FIL\_LEN[14:0] + 1)$$

For example, at a 96MHz operating frequency, the period T_{clk} is 10.4ns, the maximum register configuration is 32767, and the longest filter width is about 10.4ns × 32768≈340us.

Capture the filtered HALL signal by accessing HALL_INFO.FIL_DATA [2: 0]; HALL_INFO.RAW_DATA [2: 0] is the original HALL input signal before filtering.

11.2.4 Capture

The capture module is used to measure the time between two HALL signal changes, with a 24-bit counter as its core, which can can record a maximum time width of about 1.39 seconds at a 96MHz operating frequency, and achieve a time resolution of 10ns.

HALL_CNT starts counting from 0. When the HALL signal changes, the current HALL_CNT value will be saved to the HALL_WIDTH register, and the current HALL signal is saved to HALL_INFO.FIL_DATA. Then, a HALL signal change interrupt is output, and HALL_CNT starts counting from 0 again.

When the counter count value reaches HALL_TH, the HALL counter overflow interrupt is output, and the counter starts counting from 0 again.

11.2.5 Interrupt

Capture and overflow events trigger interrupts. The interrupt enable control bits are in HALL_CFG.CHG_IE and HALL_CFG.OV_IE, and the interrupt flag bits are in HALL_INFO.CHG_IF and HALL_INFO.OV_IF. The terminal flag can be cleared by writing 1 to HALL_INFO.CHG_IF and HALL_INFO.OV_IF.

11.2.6 Data Flow

The data flow of the HALL module is shown in the figure below. FCLK [1] is the main clock controlled by the SYS_CLK_FEN gate control, which is usually a 96MHz PLL clock.

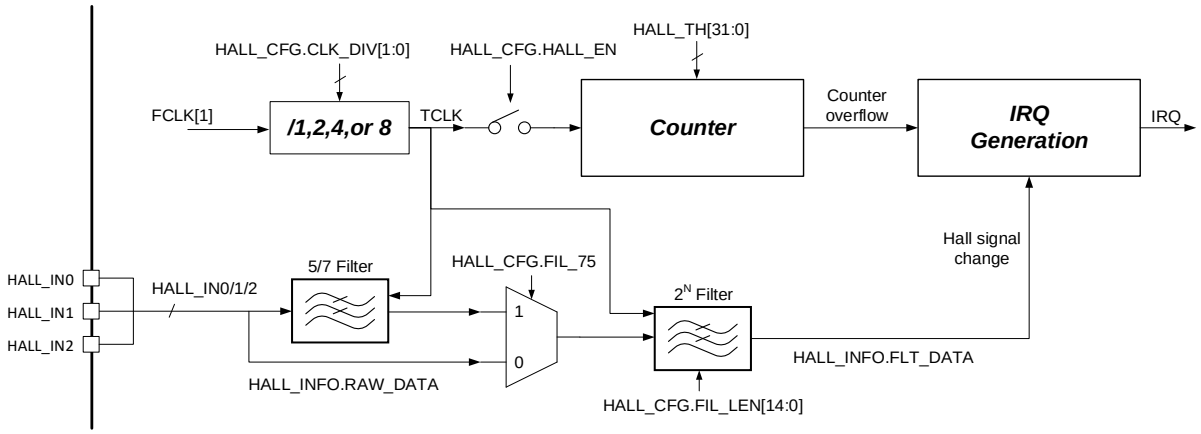


Fig. 11-2 Data Flow Diagram

11.3 Register

11.3.1 Address Allocation

The base address of the HALL module register is 0x4001_1000.

Table 11-1 HALL Module Register Address Allocation

Name	Offset	Description
HALL_CFG	0x00	HALL module configuration register
HALL_INFO	0x04	HALL module information register
HALL_WIDTH	0x08	HALL width count value register
HALL_TH	0x0C	HALL module counter threshold register
HALL_CNT	0x10	HALL count register

11.3.2 HALL_CFG HALL Module Configuration Register

Address: 0x4001_1000

Reset value: 0x0

Table 11-2 HALL_CFG HALL Module Configuration Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SW_IE	OV_IE	CHG_IE				HALL_EN				FIL_75			CLK_DIV	
	RW	RW	RW				RW				RW			RW	
	0	0	0				0				0			0	

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

Location	Bit name	Description
[31]		Unused
[30]	SW_IE	Software-triggered HALL signal change interrupt enable. Active high. After this bit is valid, writing 1 to INFO [18], a HALL signal change interrupt will be generated manually.
[29]	OV_IE	HALL counter overflow interrupt enable switch. Off by default. 1: Enable 0: Disable
[28]	CHG_IE	HALL signal change interrupt enable switch. Off by default. 1: Enable 0: Disable
[27:25]		Unused
[24]	HALL_EN	HALL module enabling switch. Off by default. 1: Enable 0: Disable
[23:21]		Unused
[20]	FIL_75	7/5 filter switch (sequential sampling for seven times, and five results should be the same). Off by default. 1: Enable 0: Disable
[19:18]		Unused
[17:16]	CLK_DIV	HALL clock division factor. No frequency division by default. 00: No frequency division 01: Two-divided frequency 10: Four-divided frequency 11: Eight-divided frequency
[15]		Unused
[14:0]	FIL_LEN	Filter width. Signals below the corresponding pulse width will be automatically filtered by the hardware. The calculation formula of the filter width is [14: 0]+1.

11.3.3 HALL_INFO HALL Module Information Register

Address: 0x4001_1004

Reset value: 0x0

Table 11-3 HALL_INFO HALL Module Information Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													SW_IF	OV_IF	CHG_IF

																RW	RW	RW	
																0	0	0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
					RAW_DATA								FLT_DATA						
					RO								RO						
					0								0						

Location	Bit name	Description
[31:19]		Unused
[18]	SW_IF	Software-triggered HALL signal change interrupt. Trigger by writing 1, and clear automatically.
[17]	OV_IF	HALL counter overflow event flag. Write 1 to clear
[16]	CHG_IF	HALL signal change event flag. Write 1 to clear
[15:11]	RESERVED	Reserved bit. Write 0, and read 0
[10:8]	RAW_DATA	HALL value. Unfiltered result
[7:3]	RESERVED	Reserved bit. Write 0, and read 0
[2:0]	FLT_DATA	HALL value. Filtered result

11.3.4 HALL Width Count Value Register

Address: 0x4001_1008

Reset value: 0x0

Table 11-4 HALL Width Count Value Register

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

Location	Bit name	Description
[31:24]		Unused
[23:0]	CAP_CNT	HALL Width Count Value Register

11.3.5 HALL_TH HALL Module Counter Threshold Register

Address: 0x4001_100C

Reset value: 0x0

Table 11-5 HALL_TH HALL Module Counter Threshold Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								TH							

	RW
	0

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

Location	Bit name	Description
[31:24]		Unused
[23:0]	TH	HALL Module Counter Threshold Register Value

11.3.6 HALL_CNT HALL Count Register

Address: 0x4001_1010

Reset value: 0x0

Table 11-6 HALL_CNT HALL Count Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CNT							
								RW							
								0							

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

Location	Bit name	Description
[31:24]		Unused
[23:0]	CNT	HALL count value. Write any value to clear

12 MCPWM

12.1 Introduction

The MCPWM module is a module that precisely controls the output of the motor drive waveform,

which contains a 16-bit counter-up counter to provide a basic period. The counter has four clock frequency divisions, 1-, 2-, 4-, and 8-divided frequency division, and the divided clock frequencies generated are 96MHz, 48MHz, 24MHz and 12MHz, respectively.

It contains four groups of PWM generation modules.

-Able to produce four pairs (complementary signals) or eight independent (edge-aligned mode) non-overlapping PWM signals;

-Support edge-aligned PWM

-Center-aligned PWM

-Phase shift PWM

Besides, it can generate four channels of timing information at the same time as MCPWM, which is used to trigger the synchronous sampling of the ADC module for linkage with MCPWM.

It also contains a set of emergencies stop protection modules for quickly shutting down the output of the MCPWM module without relying on CPU software processing. The MCPWM module can input four emergency stop signals, two of which come from IO and two from the output of the on-chip comparator. When an emergency stop event occurs (supports effective level polarity selection), reset all MCPWM output signals to the specified state to avoid short circuit.

Moreover, there is an independent filter module for the emergency stop signal.

Each output IO of MCPWM supports two control modes: PWM hardware control or software direct control (for EABS soft brake, or BLDC square-wave commutation control).

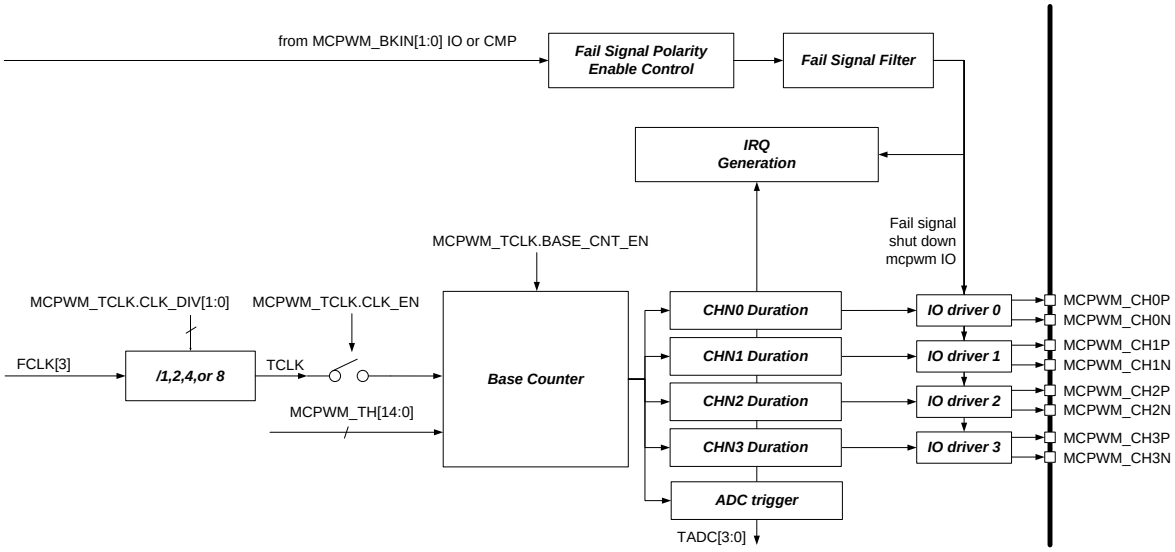


Fig. 12-1 MCPWM Module Block Diagram

Usually, a 96MHz clock is used as the operating frequency of the MCPWM module to ensure the timing accuracy.

12.1.1 Base Counter Module

The module is mainly composed of a count-up counter, and its count threshold is MCPWM_TH. The counter starts at time t0, and counts up from -TH. It passes the time zero at time t1, counts at time t2 to TH to complete a counting cycle, and then returns to -TH to restart counting. The count period is (TH×2+1) times the count clock period.

Timed event interrupt can be generated at t0/t1 (current time t0 is the previous t2), MCPWM_IF.T0_IF and MCPWM_IF.T1_IF will be set.

The start and stop of the Base Counter can be controlled by register configuration MCPWM_TCLK.BASE_CNT_EN.

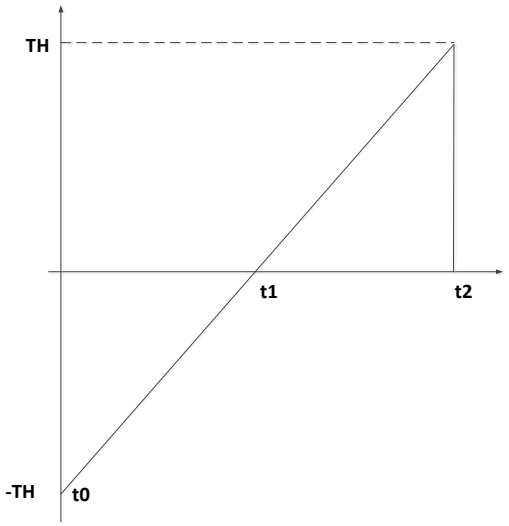


Fig. 12-2 Base Counter t0/t1 Timing

Before running the MCPWM module, users should set the corresponding comparison thresholds (MCPWM_TH00 ~ MCWM_TH31), dead-zone registers (MCPWM_DTH00 ~ MCWM_DTH31) and the PWM period (MCPWM_TH) in advance. In the actual operation process, the comparison threshold value and the PWM period register can also be changed. Update manually by writing to the MCPWM_UPDATE register, or complete hardware auto-update by setting MCPWM_SDCFG.T1_UPDATE_EN and MCPWM_SDCFG.T0_UPDATE_EN. The hardware update can only generate update events at time t0 and t1 (update t0 or t1 and update both t0 and t1 at all times), and the hardware loads the value of the load register into the running register. The occurrence frequency of the update event can be set, i.e., the update occurs every N time t0 and t1. Regardless of whether an update occurs, a corresponding interrupt can be generated at t0 and t1. If the hardware loads the value of the load register into the running register, a load interrupt is generated.

Select whether the update occurs at t0 or t1 or both by setting the MCPWM_SDCFG register, and set the update interval number as 1 ~ 16. The most frequent update configuration is that updates occur at t0 and t1, which occur continuously. The lowest speed update configuration is that the update occurs at t1, and updates every sixteen t1.

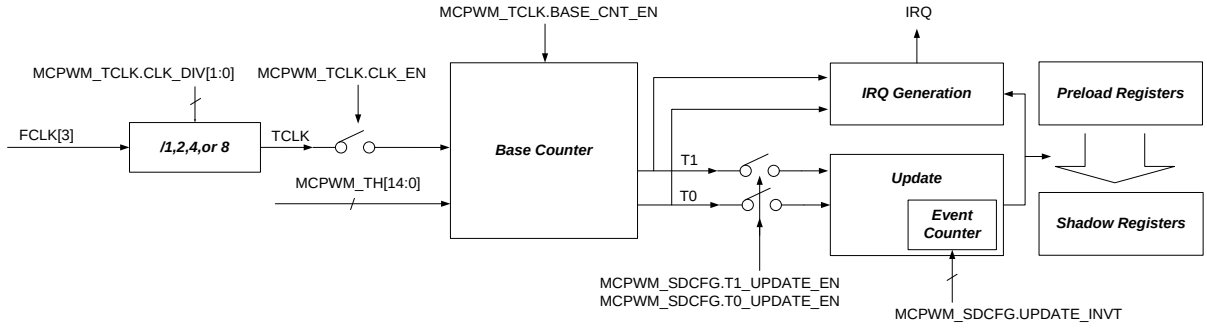


Fig. 12-3 MCPWM Update Mechanism

12.1.2 Fail Signal Processing

The Fail signal is an emergency stop signal, which is mainly used to quickly turn off the power MOS when abnormality occurs, so as to avoid irreversible hardware damage. The signal processing module mainly realizes the rapid shutdown of the PWM output by setting emergency stop event in situations. There are two fail signal inputs to MCPWM, namely FAIL0 and FAIL1, which come from the chip IO MCPWM_BKIN [1: 0] or the output CMP [1: 0] of the on-chip comparator.

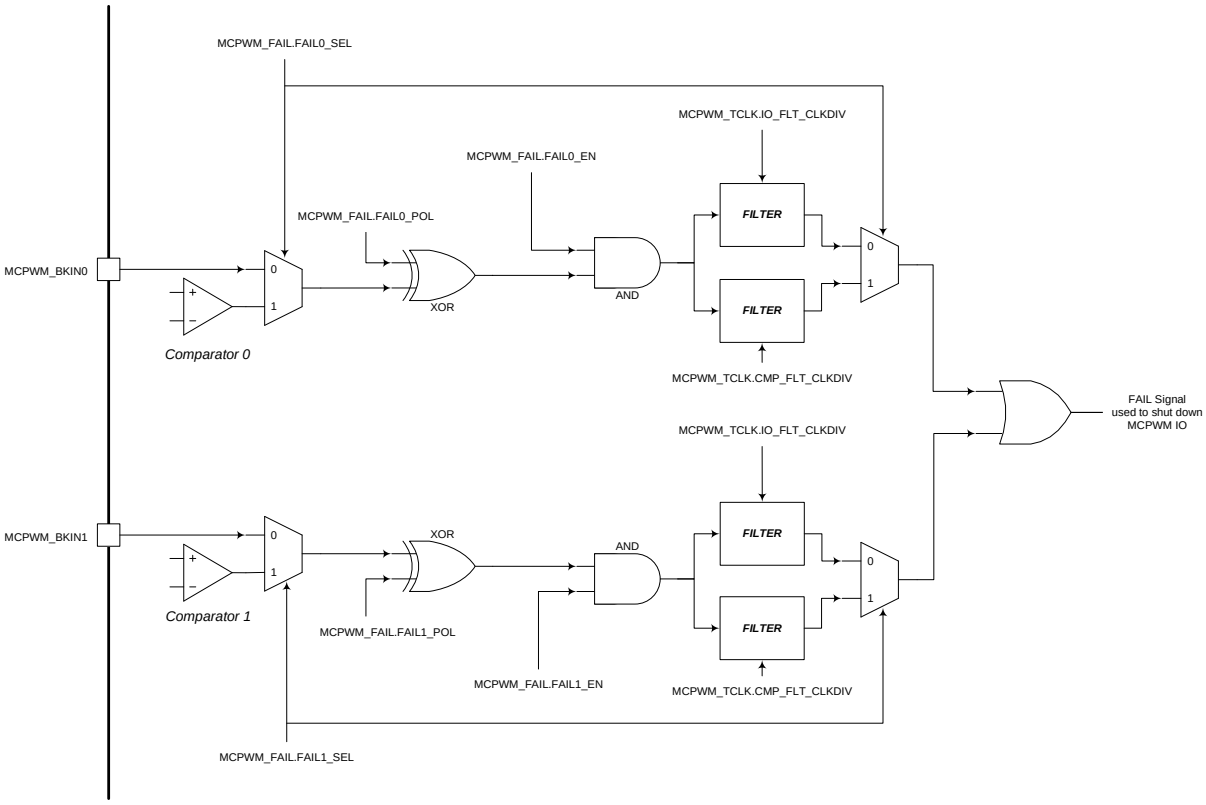


Fig. 12-4 MCPWM FAIL Logic Diagram

The clock of the Filter module comes from the gated clock FCLK [3] of the system main clock MCLK, and is divided by two stages. The first-stage frequency division is controlled by MCPWM_TCLK.CLK_DIV, which divides by 1/2/4/8 times. The second-stage frequency division can achieve 1 ~ 16 times the frequency division. If the Fail signal comes from MCPWM_BKIN [1: 0], then use MCPWM_TCLK.IO_FLT_CLKDIV [3: 0] as the second-stage frequency division coefficient; If the Fail signal comes from the internal comparator output, then use MCPWM_TCLK.CMP_FLT_CLKDIV [3: 0] as the frequency division factor of the second stage.

The MCPWM module uses the frequency-divided clock to filter the Fail signal. The filter width is fixed at 16 cycles, that is, the input signal must be stable for at least 16 clock cycles (clock frequency divided by two) before the hardware determines it as a valid input signal. The formula for the filter time constant is as follows, where T_{MCLK} is the clock period of MCLK/FCLK [3], 96MHz corresponds to 10.4ns. MCPWM_TCLK.FLT_CLKDIV may be MCPWM_TCLK.IO_FLT_CLKDIV or MCPWM_TCLK.CMP_FLT_CLKDIV depending on the configuration.

$$T = T_{MCLK} \times (MCPWM\_TCLK.CLK\_DIV) \times (MCPWM\_TCLK.FLT\_CLKDIV+1) \times 16$$

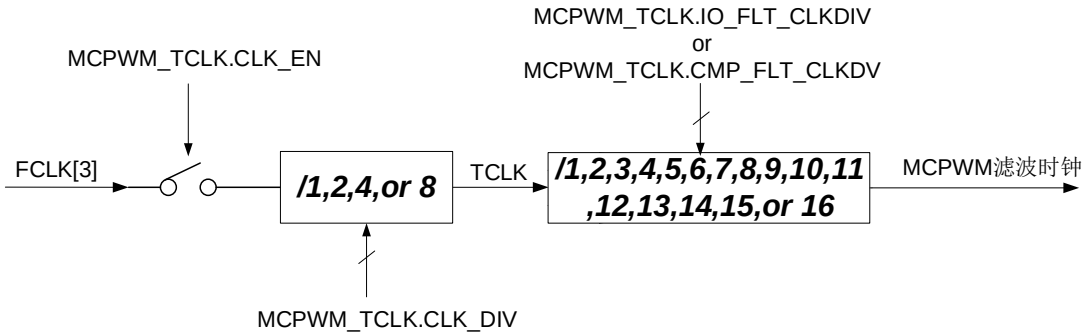


Fig. 12-5 MCPWM Fail Signal Filtering Clock Generation Logic

Once a Fail event occurs, the hardware forces the IO output to the default value of the fault specified in the MCPWM_FAIL.CHxN_DEFAULT and MCPWM_FAIL.CHxP_DEFAULT registers. After then, the values of MCPWM_FAIL.CHxN_DEFAULT and MCPWM_FAIL.CHxP_DEFAULT are directly output to the IO port, and are no longer affected by the polarity control such as MCPWM_FAIL.FAIL_POL.

The two Fail signals from the comparator are controlled by the comparator windowing, but are not controlled by the comparator filter. After the Fail signal enters MCPWM, it can be filtered in the MCPWM module.

12.1.3 MCPWM Special Output Status

All zero and all 1 output states are often used in motor control. The following complementary mode settings can get the desired output.

1. If THn0≥THn1, the chip is in a constant 0 state (CH <n> P off, CH <n> N on), no dead-zone
2. If THn0 = -TH, THn1 = TH, the chip is in a constant 1 state (CH <n> P is on, CH <n> P is off), no dead-zone

12.1.4 IO DRIVER Module

This module sets IO to the corresponding level according to the actual MCPWM register configuration. The overall data flow chart of the IO Driver module is as follows:

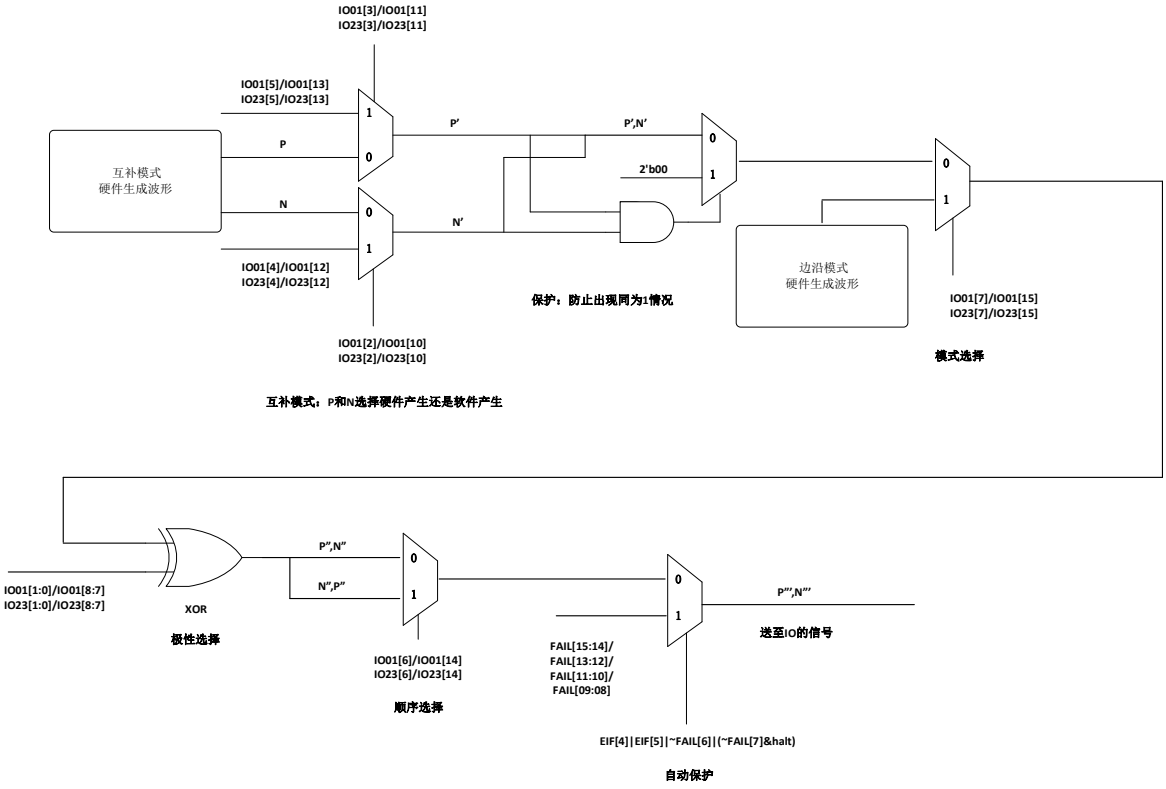


Fig. 12-6 IO Driver Module Data Flow Chart

12.1.4.1 MCPWM Wave-form Output: Center-aligned Mode

The four MCPWM IO Drivers use independent control thresholds and independent dead-zone widths (each pair of complementary IO dead-zones need to be configured independently, that is, four dead-zone configuration registers) and share data update events.

TH <n> 0 and TH <n> 1 are used to control the start and shutdown of the <n> MCPWM IO, n is 1/2/3/4.

When the counter CNT counts up to TH <n> 0, CH <n> N is turned off at time t3, and Tdead is delayed after dead-zone delay, and CH <n> P is turned on.

When the counter CNT value counts up to reach TH <n> 1, CH <n> P is turned off at time t4, after dead-zone delay Tdead, CH <n> N is turned on.

The independent startup and shutdown time control is adopted to provide phase control.

The dead-zone delay guarantees that CH <n> P/CH <n> N will not be high at the same time to avoid short circuit.

Both t3 and t4 will generate corresponding interrupts.

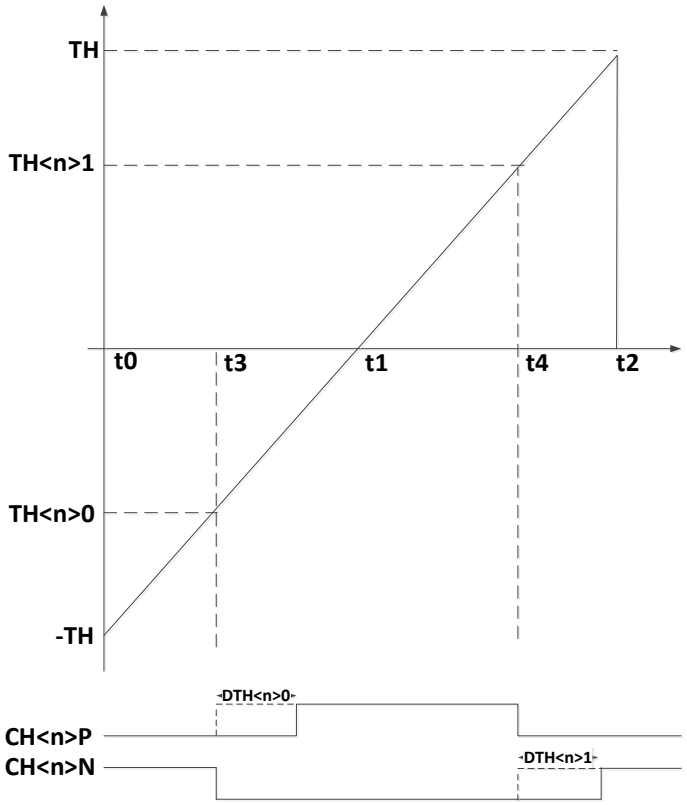


Fig. 12-7 MCPWM Timing TH <n> 0 and TH <n> 1-Center Aligned Mode

12.1.4.2 MCPWM Wave-form Control: Center-aligned Push-pull Mode

Center-aligned push-pull mode. In the first cycle, CH <n> P is turned on at time t3, and CH <n> P is turned off at time t4. In the second cycle, CH <n> N is turned on at time t3, and CH <n> N is turned off at time t4.

Both t3 and t4 will generate corresponding interrupts.

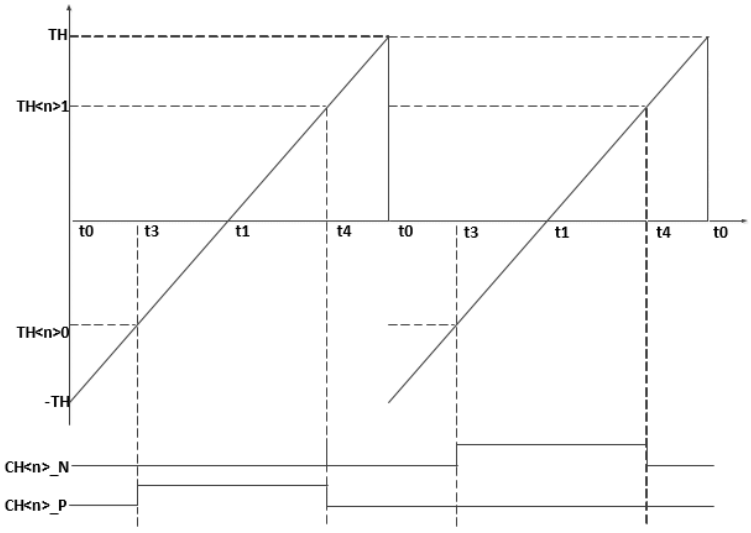


Fig. 12-8 MCPWM Timing TH <n> 0 and TH <n> 1 - Center-aligned Push-pull Mode

12.1.4.3 MCPWM Wave-form Output-Edge-aligned Mode

In edge-aligned mode, CH <n> P and CH <n> N are turned on at time t0 at the same time, then CH <n> P is turned off at time t3, and CH <n> N is turned off at time t4.

Both t3 and t4 will generate corresponding interrupts.

In edge-aligned mode, CH <n> P and CH <n> N don't need dead-zone protection.

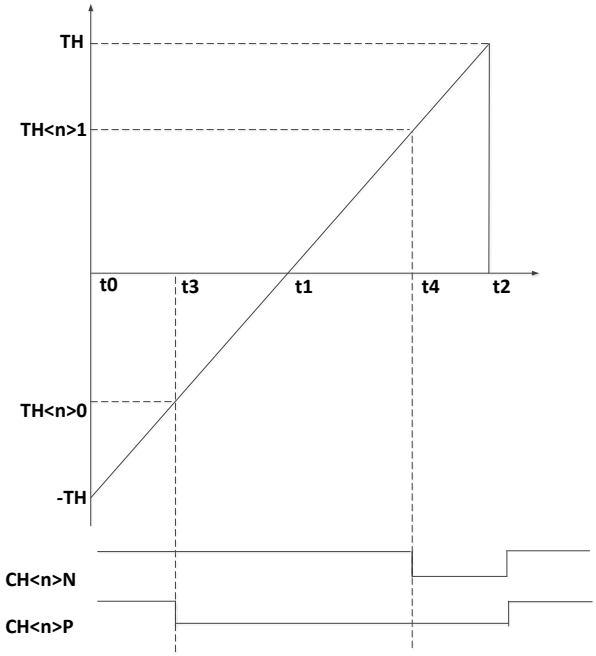


Fig. 12-9 MCPWM Timing Edge-aligned Mode

12.1.4.4 MCPWM Wave-form Control: Edge-aligned Push-pull Mode

Edge-aligned push-pull mode. In the first cycle, CH <n> P is turned on at time t0, and CH <n> P is turned off at time t3. In the second cycle, CH <n> N is turned on at time t0, and CH <n> N is turned off at time t3.

Both t0 and t3 will generate corresponding interrupts.

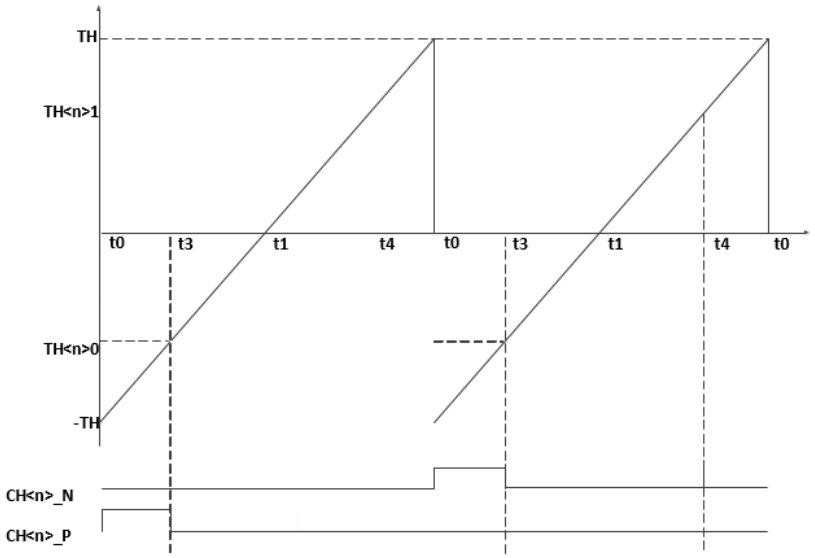


Fig. 12-10 MCPWM Timing TH <n> 0 and TH <n> 1 - Edge-aligned Push-pull Mode

12.1.4.5 MCPWM IO: Dead-zone Control

MCPWM IO is a pair of mutually exclusive control signals CH <n> P/CH <n> N, which controls the circuit shown in the figure below

When CH <n> P is high and CH <n> N is low, Vout output is high (VDD);

When CH <n> P is low and CH <n> N is high, Vout output is low (VSS);

When CH <n> P is high and CH <n> N is high, Vout output is undefined, but a short circuit from VDD to VSS will occur accordingly;

When CH <n> P is low and CH<n>N is low, the Vout output is undefined.

It is necessary to avoid the situation where CH <n> P and CH <n> N are both turned on. The introduction of dead-zone can avoid the short circuit from VDD to VSS effectively.

The dead-zone width of the four groups of MCPWM IO can be adjusted independently.

For complementary mode, MCPWM IO is automatically inserted into the dead zone.

For edge-aligned mode, MCPWM IO has no dead zone.

Added conflict detection for CH <n> P and CH <n> N in the IO Driver module. When a conflict occurs, it will pull IO low automatically and output an error interrupt (the error interrupt flag can be cleared by the software or automatically cleared by the hardware).

MCPWM IO can also be output by software configuration. At this moment, the dead-zone control is realized by software. If the MCPWM module is configured as the center-aligned mode, the hardware short-circuit protection mechanism is still effective, ensuring that P and N are not turned on at the same time.

The position of CH <n> P and CH <n> N can be interchanged on IO.

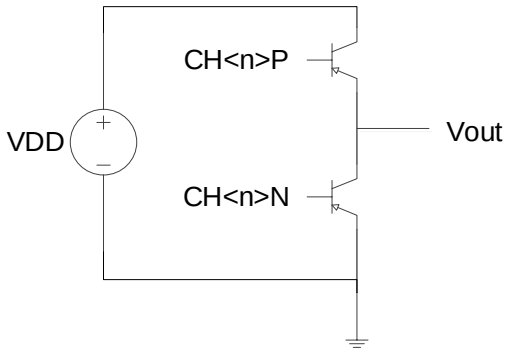


Fig. 12-11 MCPWM IO Control Diagram

12.1.4.6 MCPWM IO Polarity Setting

The effective levels of CH <n> P and CH <n> N can be set as high and low, and the effective level of each IO can be set individually. The position of CH <n> P and CH <n> N output to IO can be interchanged by software configuration.

12.1.4.7 MCPWM IO Auto-protection

When an emergency stop event (Fail event) occurs, CH <n> P and CH <n> N should be switched to the off state automatically. Remember to turn off the level configuration (MCPWM_FAIL.CHxN_DEFAULT and MCPWM_FAIL.CHxP_DEFAULT control the default level).

- After the chip works normally, the default output level of IO is the specified value of register MCPWM_FAIL.CHxN_DEFAULT and MCPWM_FAIL.CHxP_DEFAULT. When the user configuration is completed and MCPWM works normally, set MCPWM_FAIL.MCPWM_OE (i.e. MOE) to 1, and the IO output level is controlled by MCPWM IO module.
- When a Fail short circuit condition occurs, the hardware switches to the IO default output level immediately.
- When MCU Halt occurs during chip debugging, configuration can be made to stop the normal output of MCPWM and output the value of FAIL[15:8] register.

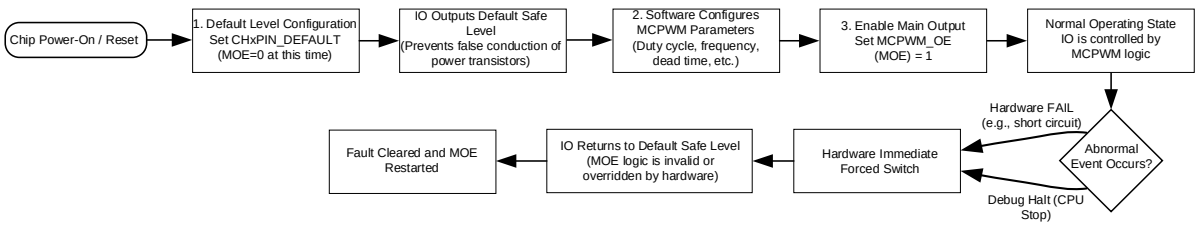


Fig. 12-12 Flow Chart of Fail-Safe Protection and Output Control

12.1.5 ADC Trigger Timer Module

MCPWM can provide ADC sampling control. When the counter counts to MCPWM_TMR0/MCPWM_TMR1/MCPWM_TMR2/MCPWM_TMR3, a timing event can be generated

to trigger ADC sampling. Besides, the trigger signal can be output to GPIO for debugging. For the specific GPIO output, please refer to the datasheet of the corresponding device.

Table 12-1 MCPWM Counter Threshold and Events

t0	-th
t1	0
tio0[0]	th00
tio0[1]	th01
TADC[0]	tmr0
TADC [1]	tmr1
TADC [2]	tmr2
TADC [3]	tmr3

12.2 Register

12.2.1 Address Allocation

The base address of the MCPWM module register is 0x4001_1C00.

Table 12-2 MCPWM Module Register List

Name	Offset address	Description
MCPWM_TH00	0x00	MCPWM CH0_P compare threshold register
MCPWM_TH01	0x04	MCPWM CH0_N compare threshold register
MCPWM_TH10	0x08	MCPWM CH1_P compare threshold register
MCPWM_TH11	0x0C	MCPWM CH1_N compare threshold register
MCPWM_TH20	0x10	MCPWM CH2_P compare threshold register
MCPWM_TH21	0x14	MCPWM CH2_N compare threshold register
MCPWM_TH30	0x18	MCPWM CH3_P compare threshold register
MCPWM_TH31	0x1C	MCPWM CH3_N compare threshold register
MCPWM_TMR0	0x20	Compare threshold 0 register for ADC sampling timer
MCPWM_TMR1	0x24	Compare threshold 1 register for ADC sampling timer
MCPWM_TMR2	0x28	Compare threshold 2 register for ADC sampling timer
MCPWM_TMR3	0x2C	Compare threshold 3 register for ADC sampling timer
MCPWM_TH	0x30	MCPWM threshold register
MCPWM_UPDATE	0x34	MCPWM load control register
MCPWM_IE	0x38	MCPWM interrupt control register
MCPWM_IF	0x3C	MCPWM interrupt flag register
MCPWM_EIE	0x40	MCPWM abnormal interrupt control register
MCPWM_EIF	0x44	MCPWM abnormal interrupt flag register
	0x48	Reserved
MCPWM_PP	0x4C	MCPWM push-pull mode enable register
MCPWM_IO01	0x50	MCPWM IO01 control register
MCPWM_IO23	0x54	MCPWM IO23 control register
MCPWM_SDCFG	0x58	MCPWM load configuration register
MCPWM_AUEN	0x5C	MCPWM auto update enabling register
MCPWM_TCLK	0x60	MCPWM clock divider control register
MCPWM_FAIL	0x64	MCPWM short circuit control register
	0x68	Reserved
	0x6C	Reserved
MCPWM_PRT	0x74	MCPWM protection register
MCPWM_CNT	0x78	MCPWM counter register

MCPWM_SWAP	0x7C	MCPWM channel mapping register
MCPWM_DTH00	0x80	MCPWM CH0 P channel dead-zone width control register
MCPWM_DTH01	0x84	MCPWM CH0 N channel dead-zone width control register
MCPWM_DTH10	0x88	MCPWM CH1 P channel dead-zone width control register
MCPWM_DTH11	0x8C	MCPWM CH1 N channel dead-zone width control register
MCPWM_DTH20	0x90	MCPWM CH2 P channel dead-zone width control register
MCPWM_DTH21	0x94	MCPWM CH2 N channel dead-zone width control register
MCPWM_DTH30	0x98	MCPWM CH3 P channel dead-zone width control register
MCPWM_DTH31	0x9C	MCPWM CH3 N channel dead-zone width control register

Table 12-3 Registers Protected by MCPWM_PRT

Name	Offset address	Description
MCPWM_TH	0x30	MCPWM threshold register
MCPWM_IE	0x38	MCPWM interrupt control register
MCPWM_EIE	0x40	MCPWM abnormal interrupt control register
MCPWM_PP	0x4C	MCPWM push-pull mode enable register
MCPWM_IO01	0x50	MCPWM IO01 control register
MCPWM_IO23	0x54	MCPWM IO23 control register
MCPWM_SDCFG	0x58	MCPWM load configuration register
MCPWM_AUEN	0x5C	MCPWM auto load enabling register
MCPWM_TCLK	0x60	MCPWM clock divider control register
MCPWM_FAIL	0x64	MCPWM short circuit control register
MCPWM_DTH00	0x80	MCPWM CH0 P channel dead-zone width control register
MCPWM_DTH01	0x84	MCPWM CH0 N channel dead-zone width control register
MCPWM_DTH10	0x88	MCPWM CH1 P channel dead-zone width control register
MCPWM_DTH11	0x8C	MCPWM CH1 N channel dead-zone width control register
MCPWM_DTH20	0x90	MCPWM CH2 P channel dead-zone width control register
MCPWM_DTH21	0x94	MCPWM CH2 N channel dead-zone width control register
MCPWM_DTH30	0x98	MCPWM CH3 P channel dead-zone width control register
MCPWM_DTH31	0x9C	MCPWM CH3 N channel dead-zone width control register

Table 12-4 Registers with Shadow Registers

Name	Offset address	Description
MCPWM_TH00	0x00	MCPWM CH0_P compare threshold register
MCPWM_TH01	0x04	MCPWM CH0_N compare threshold register
MCPWM_TH10	0x08	MCPWM CH1_P compare threshold register

MCPWM_TH11	0x0C	MCPWM CH1_N compare threshold register
MCPWM_TH20	0x10	MCPWM CH2_P compare threshold register
MCPWM_TH21	0x14	MCPWM CH2_N compare threshold register
MCPWM_TH30	0x18	MCPWM CH3_P compare threshold register
MCPWM_TH31	0x1C	MCPWM CH3_N compare threshold register
MCPWM_TMR0	0x20	Compare threshold 0 register for ADC sampling timer
MCPWM_TMR1	0x24	Compare threshold 1 register for ADC sampling timer
MCPWM_TMR2	0x28	Compare threshold 2 register for ADC sampling timer
MCPWM_TMR3	0x2C	Compare threshold 3 register for ADC sampling timer
MCPWM_TH	0x30	MCPWM threshold register
MCPWM_CNT	0x78	MCPWM counter register

12.2.2 MCPWM_TH00

Unprotected register

Address: 0x4001_1C00

Reset value: 0x0

Table 12-5 MCPWM_TH00 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH00															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	TH00	MCPWM CH0_P comparison threshold, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.3 MCPWM_TH01

Unprotected register

Address: 0x4001_1C04

Reset value: 0x0

Table 12-6 MCPWM_TH01 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH01															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	TH01	MCPWM CH0_N comparison threshold, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.4 MCPWM_TH10

Unprotected register

Address: 0x4001_1C08

Reset value: 0x0

Table 12-7 MCPWM_TH10 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH10															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
15:0]	TH10	MCPWM CH1_P comparison threshold, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.5 MCPWM_TH11

Unprotected register

Address: 0x4001_1C0C

Reset value: 0x0

Table 12-8 MCPWM_TH11 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH11															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	TH11	MCPWM CH1_N comparison threshold, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.6 MCPWM_TH20

Unprotected register

Address: 0x4001_1C10

Reset value: 0x0

Table 12-9 MCPWM_TH20 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH20															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	TH20	MCPWM CH2_P comparison threshold, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.7 MCPWM_TH21

Unprotected register

Address: 0x4001_1C14

Reset value: 0x0

Table 12-10 MCPWM_TH21 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH21															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	TH21	MCPWM CH2_N comparison threshold, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.8 MCPWM_TH30

Unprotected register

Address: 0x4001_1C18

Reset value: 0x0

Table 12-11 MCPWM_TH30 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH30															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	TH30	MCPWM CH3_P comparison threshold, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.9 MCPWM_TH31

Unprotected register

Address: 0x4001_1C1C

Reset value: 0x0

Table 12-12 MCPWM_TH31 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH31															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	TH31	MCPWM CH3_N comparison threshold, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.10 MCPWM_TMR0

Unprotected register

Address: 0x4001_1C20

Reset value: 0x0

Table 12-13 MCPWM_TMR0 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0															
RW															
0															

Location	Bit name	Description

[31:16]		Unused
[15:0]	TMR0	Compare threshold 0 register for ADC sampling timer, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM operating system.

12.2.11 MCPWM_TMR1

Unprotected register

Address: 0x4001_1C24

Reset value: 0x0

Table 12-14 MCPWM_TMR1 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR1															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	TMR1	Compare threshold 1 register for ADC sampling timer, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.12 MCPWM_TMR2

Unprotected register

Address: 0x4001_1C28

Reset value: 0x0

Table 12-15 MCPWM_TMR2 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR2															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	TMR2	Compare threshold 2 register for ADC sampling timer, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.13 MCPWM_TMR3

Unprotected register

Address: 0x4001_1C2C

Reset value: 0x0

Table 12-16 MCPWM_TMR3 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR3															
RW															
0															

Location	Bit name	Description
[31:16]		Unused
[15:0]	TMR3	Compare threshold 3 register for ADC sampling timer, 16-bit signed number; after an update event occurs, this register is loaded into the MCPWM operating system.

12.2.14 MCPWM_TH

Write-protected register

Address: 0x4001_1C30

Reset value: 0x0

Table 12-17 MCPWM_TH Configuration Register

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

Location	Bit name	Description
[31:15]		Unused
[14:0]	TH	MCPWM counter threshold value, 15-bit unsigned number, the counter in the operating system of MCPWM counts from -TH to TH; after an update event occurs, this register is loaded into the MCPWM actual operating system.

12.2.15 MCPWM_UPDATE

Unprotected register

Address: 0x4001_1C34

Reset value: 0x0

Table 12-18 MCPWM_UPDATE Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		CNT_UPDATE	TH_UPDATE	TMR3_UPDATE	TMR2_UPDATE	TMR1_UPDATE	TMR0_UPDATE	TH31_UPDATE	TH30_UPDATE	TH21_UPDATE	TH20_UPDATE	TH11_UPDATE	TH10_UPDATE	TH01_UPDATE	TH00_UPDATE
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
		0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:14]		Unused
[13]	CNT_UPDATE	Manually load the contents of the MCPWM_CNT register into the MCPWM operating system. 1: Load; 0: Do not load.
[12]	TH_UPDATE	Manually load the contents of the MCPWM_TH register into the MCPWM operating system. 1: Load; 0: Do not load.
[11]	TMR3_UPDATE	Manually load the contents of the MCPWM_TMR3 register into the MCPWM operating system. 1: Load; 0: Do not load.
[10]	TMR2_UPDATE	Manually load the contents of the MCPWM_TMR2 register into the MCPWM operating system. 1: Load; 0: Do not load.
[9]	TMR1_UPDATE	Manually load the contents of the MCPWM_TMR1 register into the MCPWM operating system. 1: Load; 0: Do not load.
[8]	TMR0_UPDATE	Manually load the contents of the MCPWM_TMR0 register into the MCPWM operating system. 1: Load; 0: Do not load.
[7]	TH31_UPDATE	Manually load the contents of the MCPWM_TH31 register into the MCPWM operating system. 1: Load; 0: Do not load.
[6]	TH30_UPDATE	Manually load the contents of the MCPWM_TH30 register into the MCPWM operating system. 1: Load; 0: Do not load.
[5]	TH21_UPDATE	Manually load the contents of the MCPWM_TH21 register into the MCPWM operating system. 1: Load; 0: Do not load.
[4]	TH20_UPDATE	Manually load the contents of the MCPWM_TH20 register into the MCPWM operating system. 1: Load; 0: Do not load.
[3]	TH11_UPDATE	Manually load the contents of the MCPWM_TH11 register into the MCPWM operating system. 1: Load; 0: Do not load.
[2]	TH10_UPDATE	Manually load the contents of the MCPWM_TH10 register into the MCPWM operating system. 1: Load; 0: Do not load.
[1]	TH01_UPDATE	Manually load the contents of the MCPWM_TH01 register into the MCPWM operating system. 1: Load; 0: Do not load.
[0]	TH00_UPDATE	Manually load the contents of the MCPWM_TH00 register into the MCPWM operating system.

		1: Load; 0: Do not load.

12.2.16 MCPWM_IE

Write-protected register

Address: 0x4001_1C38

Reset value: 0x0

Table 12-19 MCPWM_IE Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SHADE_IE	TMR3_IE	TMR2_IE	TMR1_IE	TMR0_IE	TH31_IE	TH30_IE	TH21_IE	TH20_IE	TH11_IE	TH10_IE	TH01_IE	TH00_IE	T1_IE	T0_IE
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:15]		Unused
[14]	SHADE_IE	MCPWM_TH/MCPWM_TH00 ~ MCPWM_TH31/MCPWM_TMR0 ~ MCPWM_TMR3 and other registers are updated to enable interrupt source of MCPWM actual operating system. 1: enable; 0: disable.
[13]	TMR3_IE	The count value of the counter in the MCPWM actual operating system is equal to MCPWM_TMR3 interrupt source enable. 1: enable; 0: disable.
[12]	TMR2_IE	The count value of the counter in the MCPWM operating system is equal to MCPWM_TMR2 interrupt source enable. 1: enable; 0: disable.
[11]	TMR1_IE	The count value of the counter in the MCPWM operating system is equal to MCPWM_TMR1 interrupt source enable. 1: enable; 0: disable.
[10]	TMR0_IE	The count value of the counter in the MCPWM actual operating system is equal to MCPWM_TMR0 interrupt source enable. 1: enable; 0: disable.
[9]	TH31_IE	The count value of the counter in the MCPWM actual operating system is equal to MCPWM_TH31 interrupt source enable. 1: enable; 0: disable.
[8]	TH30_IE	The count value of the counter in the MCPWM actual operating system is equal to MCPWM_TH30 interrupt source enable. 1: enable; 0: disable.
[7]	TH21_IE	The count value of the counter in the MCPWM actual operating system is equal to MCPWM_TH21 interrupt source enable. 1: enable; 0: disable.
[6]	TH20_IE	The count value of the counter in the MCPWM actual operating system is equal to MCPWM_TH20 interrupt source enable. 1: enable; 0: disable.
[5]	TH11_IE	The count value of the counter in the MCPWM actual operating system is equal to MCPWM_TH11 interrupt source enable. 1: enable; 0: disable.
[4]	TH10_IE	The count value of the counter in the MCPWM actual operating system is equal to MCPWM_TH10 interrupt source enable. 1: enable; 0: disable.
[3]	TH01_IE	The count value of the counter in the MCPWM actual operating system is equal to MCPWM_TH01 interrupt source enable. 1: enable; 0: disable.
[2]	TH00_IE	The count value of the counter in the MCPWM actual operating system is equal to MCPWM_TH00 interrupt source enable. 1: enable; 0: disable.
[1]	T1_IE	t1 event. The count value of the counter reaches 0 and the interrupt source

		is enabled. 1: enable; 0: disable.
[0]	T0_IE	t0 event. The count value of the counter returns to MCPWM_TH, and the interrupt source is enabled. 1: enable; 0: disable.

12.2.17 MCPWM_IF

Unprotected register

Address: 0x4001_1C3C

Reset value: 0x0

Table 12-20 MCPWM_IF Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SHADE_IF	TMR3_IF	TMR2_IF	TMR1_IF	TMR0_IF	TH31_IF	TH30_IF	TH21_IF	TH20_IF	TH11_IF	TH10_IF	TH01_IF	TH00_IF	T1_IF	T0_IF
	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:15]		Unused
[14]	SHADE_IF	MCPWM_TH/MCPWM_TH00 ~ MCPWM_TH31/MCPWM_TMR0 ~ MCPWM_TMR3 and other registers are updated to the interrupt source event of MCPWM actual operating system. 1: occurred; 0: did not occurred. Clear by writing 1.
[13]	TMR3_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TMR3 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[12]	TMR2_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TMR2 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[11]	TMR1_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TMR1 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[10]	TMR0_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TMR0 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[9]	TH31_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TH31 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[8]	TH30_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TH30 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[7]	TH21_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TH21 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.

[6]	TH20_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TH20 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[5]	TH11_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TH11 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[4]	TH10_IF	The count value of the counter in the MCPWM operating system is equal to the MCPWM_TH10 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[3]	TH01_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TH01 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[2]	TH00_IF	The count value of the counter in the MCPWM actual operating system is equal to the MCPWM_TH00 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[1]	T1_IF	t1 event. Interrupt source event where the count value of the counter reaches 0. 1: occurred; 0: did not occurred. Clear by writing 1.
[0]	T0_IF	t0 event. Interrupt source event where the count value of the counter returns to MCPWM_TH. 1: occurred; 0: did not occurred. Clear by writing 1.

12.2.18 MCPWM_EIE

Write-protected register

Address: 0x4001_1C40

Reset value: 0x0

Table 12-21 MCPWM_EIE Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										FAIL1_IE	FAIL0_IE	CH3_SHORT_IE	CH2_SHORT_IE	CH1_SHORT_IE	CH0_SHORT_IE
										RW	RW	RW	RW	RW	RW
										0	0	0	0	0	0

Location	Bit name	Description
[31:6]		Unused
[5]	FAIL1_IE	FAIL1 interrupt source enable. 1: enable; 0: disable.
[4]	FAIL0_IE	FAIL0 interrupt source enable. 1: enable; 0: disable.
[3]	CH3_SHORT_IE	MCPWM CH3_P and CH3_N are valid at the same time, the interrupt source is enabled. 1: enable; 0: disable.
[2]	CH2_SHORT_IE	MCPWM CH2_P and CH2_N are valid at the same time, the interrupt source is enabled. 1: enable; 0: disable.
[1]	CH1_SHORT_IE	MCPWM CH1_P and CH1_N are valid at the same time, the interrupt source is enabled. 1: enable; 0: disable.
[0]	CH0_SHORT_IE	MCPWM CH0_P and CH0_N are valid at the same time, and the interrupt source is enabled. 1: enable; 0: disable.

12.2.19 MCPWM_EIF

Unprotected register

Address: 0x4001_1C44

Reset value: 0x0

Table 12-22 MCPWM_EIF Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										FAIL1_IF	FAIL0_IF	CH3_SHORT_IF	CH2_SHORT_IF	CH1_SHORT_IF	CH0_SHORT_IF
										RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
										0	0	0	0	0	0

Location	Bit name	Description
[31:6]		Unused
[5]	FAIL1_IF	FAIL1 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[4]	FAIL0_IF	FAIL0 interrupt source event. 1: occurred; 0: did not occurred. Clear by writing 1.
[3]	CH3_SHORT_IF	MCPWM CH3_P and CH3_N are active at the same time, interrupt source event occurred. 1: occurred; 0: did not occurred. Clear by writing 1.
[2]	CH2_SHORT_IF	MCPWM CH2_P and CH2_N are valid at the same time, interrupt source event occurred. 1: occurred; 0: did not occurred. Clear by writing 1.
[1]	CH1_SHORT_IF	MCPWM CH1_P and CH1_N are valid at the same time, interrupt source event occurred. 1: occurred; 0: did not occurred. Clear by writing 1.
[0]	CH0_SHORT_IF	MCPWM CH0_P and CH0_N are valid at the same time, interrupt source event occurred. 1: occurred; 0: did not occurred. Clear by writing 1.

12.2.20 MCPWM_PP

Write-protected register

Address: 0x4001_1C4C

Reset value: 0x0

Table 12-23 MCPWM_PP Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												IO3_PPE	IO2_PPE	IO1_PPE	IO0_PPE
												RW	RW	RW	RW
												0	0	0	0

Location	Bit name	Description
[31:4]		Unused
[3]	IO3_PPE	IO3 push-pull mode enable signal. Write 1 to enable; write 0 to disable.
[2]	IO2_PPE	IO2 push-pull mode enable signal. Write 1 to enable; write 0 to disable.
[1]	IO1_PPE	IO1 push-pull mode enable signal. Write 1 to enable; write 0 to disable.
[0]	IO0_PPE	IO 0 push-pull mode enable signal. Write 1 to enable; write 0 to disable.

Push-pull mode enable signal varies according to different operating modes Edge mode: turn on the edge-aligned push-pull mode; center alignment: turn on the central-aligned push-pull mode.

12.2.21 MCPWM_IO01

Write-protected register

Address: 0x4001_1C50

Reset value: 0x0

Table 12-24 MCPWM_IO01 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1_WM	CH1_PN_SW	CH1_SCTRLP	CH1_SCTRLN	CH1_PS	CH1_NS	CH1_PP	CH1_NP	CH0_WM	CH0_PN_SW	CH0_SCTRLP	CH0_SCTRLN	CH0_PS	CH0_NS	CH0_PP	CH0_NP
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15]	CH1_WM	CH1 working mode selection. 1: Edge mode; 0: complementary mode.
[14]	CH1_PN_SW	CH1 P and N channel output interchange selection. I.e., the P channel signal is finally output from the N channel, and the N channel signal is finally output from the P channel. 1: interchangeable; 0: not interchangeable.
[13]	CH1_SCTRLP	When CH1_PS = 1, the value output to CH1 P channel.
[12]	CH1_SCTRLN	When CH1_NS = 1, the value output to CH1 N channel.
[11]	CH1_PS	CH1 P source. 1: From CH1_SCTRLP; 0: MCPWM internal counter is generated.
[10]	CH1_NS	CH1 N source. 1: From CH1_SCTRLN; 0: MCPWM internal counter is generated.
[9]	CH1_PP	CH1 P polarity selection. 1: CH1 P signal is inverted and output; 0: CH1 P signal is output normally.
[8]	CH1_NP	CH1 N polarity selection. 1: CH1 N signal is inverted and output; 0: CH1 N signal is output normally.
[7]	CH0_WM	CH0 working mode selection. 1: Edge mode; 0: complementary mode.
[6]	CH0_PN_SW	CH0 P and N channel output interchange selection. I.e., the P channel signal is finally output from the N channel, and the N channel signal is finally output from the P channel. 1: interchangeable; 0: not

		interchangeable.
[5]	CH0_SCTRLP	When CH0_PS = 1, the value output to CH0 P channel.
[4]	CH0_SCTRLN	When CH0_NS = 1, the value output to CH0 N channel.
[3]	CH0_PS	CH0 P source. 1: From CH0_SCTRLP; 0: The counter is generated in the MCPWM actual operating system.
[2]	CH0_NS	CH0 N source. 1: From CH0_SCTRLN; 0: The counter is generated in the MCPWM actual operating system.
[1]	CH0_PP	CH0 P polarity selection. 1: CH0 P signal is inverted and output; 0: CH0 P signal is output normally.
[0]	CH0_NP	CH0 N polarity selection. 1: CH0 N signal is inverted and output; 0: CH0 N signal is output normally. **Polarity selection follows channel switching. For example, CH0 N selects the inverted output, and at the same time selects channel switching, the CH0 N after the exchange is still the inverted output.**

12.2.22 MCPWM_IO23

Write-protected register

Address: 0x4001_1C54

Reset value: 0x0

Table 12-25 MCPWM_IO23 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3_WM	CH3_PN_SW	CH3_SCTRLP	CH3_SCTRLN	CH3_PS	CH3_NS	CH3_PP	CH3_NP	CH2_WM	CH2_PN_SW	CH2_SCTRLP	CH2_SCTRLN	CH2_PS	CH2_NS	CH2_PP	CH2_NP
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Bit name	Description
[31:16]		Unused
[15]	CH3_WM	CH3 working mode selection. 1: Edge mode; 0: complementary mode.
[14]	CH3_PN_SW	CH3 P and N channel output interchange selection. I.e., the P channel signal is finally output from the N channel, and the N channel signal is finally output from the P channel. 1: interchangeable; 0: not interchangeable.
[13]	CH3_SCTRLP	When CH3_PS = 1, the value output to CH3 P channel.
[12]	CH3_SCTRLN	When CH3_NS = 1, the value output to CH3 N channel.
[11]	CH3_PS	CH3 P source. 1: From CH3_SCTRLP; 0: The counter is generated in the MCPWM actual operating system.
[10]	CH3_NS	CH3 N source. 1: From CH3_SCTRLN; 0: The counter is generated in the MCPWM actual operating system.
[9]	CH3_PP	CH3 P polarity selection. 1: CH3 P signal is inverted and output; 0: CH3 P signal is output normally.
[8]	CH3_NP	CH3 N polarity selection. 1: CH3 N signal is inverted and output; 0: CH3 N signal is output normally.

[7]	CH2_WM	CH2 working mode selection. 1: Edge mode; 0: complementary mode.
[6]	CH2_PN_SW	CH2 P and N channel output interchange selection. I.e., the P channel signal is finally output from the N channel, and the N channel signal is finally output from the P channel. 1: interchangeable; 0: not interchangeable.
[5]	CH2_SCTRLP	When CH2_PS = 1, the value output to CH2 P channel.
[4]	CH2_SCTRLN	When CH2_NS = 1, the value output to CH2 N channel.
[3]	CH2_PS	CH2 P source. 1: From CH2_SCTRLP; 0: The counter is generated in the MCPWM actual operating system.
[2]	CH2_NS	CH2 N source. 1: From CH2_SCTRLN; 0: The counter is generated in the MCPWM operating system.
[1]	CH20_PP	CH2 P polarity selection. 1: CH2 P signal is inverted and output; 0: CH2 P signal is output normally.
[0]	CH2_NP	CH2 N polarity selection. 1: CH2 N signal is inverted and output; 0: CH2 N signal is output normally. **Polarity selection follows channel switching. For example, CH0 N selects the inverted output, and at the same time selects channel switching, the CH0 N after the exchange is still the inverted output.**

12.2.23 MCPWM_SDCFG

Write-protected register

Address: 0x4001_1C58

Reset value: 0x0

Table 12-26 MCPWM_SDCFG Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									AUTO_ERR_CLR	T1_UPDATE_EN	T0_UPDATE_EN	UPDATE_INTV			
									RW	RW	RW	RW			
									0	0	0	0			

Location	Bit name	Description
[31:8]		Unused
[7]	RESERVED	Reserved bit
[6]	AUTO_ERR_CLR	Whether the AUTO_ERR_CLR update event automatically clears MCPWM_EIF [5: 4] and sets MOE to restore MCPWM signal output. 1: Enable automatic fault clearing function; 0: Disable automatic fault clearing function.
[5]	T1_UPDATE_EN	The t1 (over-zero) event update is enabled. 1: enable; 0, disable.
[4]	T0_UPDATE_EN	t0 (starting point) event update enable. 1: enable; 0, disable.
[3:0]	UPDATE_INTV	Update interval. Once the number of t0 and t1 events is equal to UPDATE_INTV + 1, the MCPWM system triggers the operation of the MCPWM_TH (including THxx) and MCPWM_TMR registers automatically, and loaded into the MCPWM operating system. If both B [5] and B [4] are closed, this type of loading will not be triggered, and the loading can only be triggered manually.

12.2.24 MCPWM_AUEN

Write-protected register

Address: 0x4001_1C5C

Reset value: 0x0

Table 12-27 MCPWM_AUEN MCPWM Auto Update Enabling Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		CNT_AUEN	TH_AUEN	TMR3_AUEN	TMR2_AUEN	TMR1_AUEN	TMR0_AUEN	TH31_AUEN	TH30_AUEN	TH21_AUEN	TH20_AUEN	TH11_AUEN	TH10_AUEN	TH01_AUEN	TH00_AUEN
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
		0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:14]		Unused
[13]	CNT_AUEN	MCPWM_CNT auto load enabling. 1: Load; 0: Do not load.
[12]	TH_ AUEN	MCPWM_TH auto load enabling. 1: Load; 0: Do not load.
[11]	TMR3_ AUEN	MCPWM_TMR3 auto load enabling. 1: Load; 0: Do not load.
[10]	TMR2_ AUEN	MCPWM_TMR2 auto load enabling. 1: Load; 0: Do not load.
[9]	TMR1_ AUEN	MCPWM_TMR1 auto load enabling. 1: Load; 0: Do not load.
[8]	TMR0_ AUEN	MCPWM_TMR0 auto load enabling. 1: Load; 0: Do not load.
[7]	TH31_ AUEN	MCPWM_TH31 auto load enabling. 1: Load; 0: Do not load.
[6]	TH30_ AUEN	MCPWM_TH30 auto load enabling. 1: Load; 0: Do not load.
[5]	TH21_ AUEN	MCPWM_TH21 auto load enabling. 1: Load; 0: Do not load.
[4]	TH20_ AUEN	MCPWM_TH20 auto load enabling. 1: Load; 0: Do not load.
[3]	TH11_ AUEN	MCPWM_TH11 auto load enabling. 1: Load; 0: Do not load.
[2]	TH10_ AUEN	MCPWM_TH10 auto load enabling. 1: Load; 0: Do not load.
[1]	TH01_ AUEN	MCPWM_TH01 auto load enabling. 1: Load; 0: Do not load.
[0]	TH00_ AUEN	MCPWM_TH00 auto load enabling. 1: Load; 0: Do not load.

12.2.25 MCPWM_TCLK

Write-protected register

Address: 0x4001_1C60

Reset value: 0x0

Table 12-28 MCPWM_TCLK Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP_FLT_CLKDIV				IO_FLT_CLKDIV								BASE_CNT_EN	CLK_EN	CLK_DIV	
RW				RW								RW	RW	RW	
0				0								0	0	0	

Location	Bit name	Description
[31:16]		Unused
[15:12]	CMP_FLT_CLKDIV	The filter clock divider register output by the comparator is divided based on the system clock and affects MCPWM_FAIL [1: 0]. The formula is as follows: System clock/(B[15:12]+1). The frequency division range is 1 to 16
[11:8]	IO_FLT_CLKDIV	The filter clock divider register of the GPIO input is divided based on the system clock, and affects MCPWM_FAIL [1: 0]. The formula is as follows: System clock/ (B[11:8] + 1) The frequency division range is 1 to 16
[7:4]		Unused
[3]	BASE_CNT_EN	MCPWM operation counter enable switch. 1: Enable; 0: Disable.
[2]	CLK_EN	MCPWM working clock enabling. 1: Enable; 0: Disable.
[1:0]	CLK_DIV	MCPWM working clock divider register. 0: System clock 1: System clock/2 2: System clock/4 3: System clock/8

12.2.26 MCPWM_FAIL

Write-protected register

Address: 0x4001_1C64

Reset value: 0x0

Table 12-29 MCPWM_FAIL Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3P_DEFAULT	CH3N_DEFAULT	CH2P_DEFAULT	CH2N_DEFAULT	CH1P_DEFAULT	CH1N_DEFAULT	CH0P_DEFAULT	CH0N_DEFAULT	HALT_PRT	MCPWM_OE	FAIL1_EN	FAIL0_EN	FAIL1_POL	FAIL0_POL	FAIL1_SEL	FAIL0_SEL
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:16]		Unused
[15]	CH3N_DEFAULT	CH3 N channel default value
[14]	CH3P_DEFAULT	CH3 P channel default value
[13]	CH2N_DEFAULT	CH2 N channel default value
[12]	CH2P_DEFAULT	CH2 P channel default value
[11]	CH1N_DEFAULT	CH1 N channel default value
[10]	CH1P_DEFAULT	CH1 P channel default value
[9]	CH0N_DEFAULT	CH0 N channel default value

[8]	CH0P_DEFAULT	CH0 P channel default value When a FAIL event occurs or MOE is 0, the corresponding channel outputs the default level. **The default level output controls the channel output, and is not affected by the exchange and polarity control of BIT0, BIT1, BIT8, BIT9, BIT6, BIT14 of MCPWM_IO01 and MCPWM_IO23.**
[7]	HALT_PRT	The MCU enters the HALT state, and the MCPWM output value is selected. 1: Normal output; 0: Force MCPWM to output protection value.
[6]	MCPWM_OE	MOE controls MCPWM CH P and N output values. 1: Output the normal signal generated by MCPWM 0: Output CHxN_DEFAULT and CHxP_DEFAULT default values. This default value is not controlled by polarity, channel selection, etc. Any change of MCPWM_EIF.FAIL1_IF and MCPWM_EIF.FAIL0_IF to "1" will trigger MCPWM_OE to become "0", and output the default value.
[5]	FAIL1_EN	FAIL1 input enable. 1: Enable; 0: Disable.
[4]	FAIL0_EN	FAIL0 input enable. 1: Enable; 0: Disable.
[3]	FAIL1_POL	FAIL1 polarity selection. 1: Invert signal polarity input. The input signal is active low; 0: Normal signal polarity input. The input signal is active high.
[2]	FAIL0_POL	FAIL0 polarity selection. 1: Invert signal polarity input. The input signal is active low; 0: Normal signal polarity input. The input signal is active high.
[1]	FAIL1_SEL	FAIL1 source selection. 1: Comparator 1 result; 0: From GPIO No. 1.
[0]	FAIL0_SEL	FAIL0 source selection. 1: Comparator 0 result; 0: From GPIO No. 0.

MCPWM_FAIL can be used to set emergency stop events and block MCPWM signal output. There are two main emergency stop events, FAIL0 and FAIL1. There are four signal sources, Comparator 0 output and Comparator 1 output, and MCPWM_BKIN0 and MCPWM_BKIN0. FAIL0 can come from comparator 0 output or chip IO MCPWM_BKIN0, FAIL1 can come from comparator 1 output or chip IO MCPWM_BKIN1.

The input signal of FAIL can be processed by digital filtering, and the first frequency division of the filtered clock is set by the MCPWM_TCLK.CLK_DIV register. The signal source Comparator 0 output and the filtered clock output of Comparator 1 are set by MCPWM_TCLK.CMP_FLT_CLKDIV; The filter clock frequency division of the signal sources MCPWM_BKIN0 and MCPWM_BKIN0 is set by MCPWM_TCLK.IO_FLT_CLKDIV.

Finally, FAIL0 and FAIL1 filter the signal with 16 filter clocks, that is, the signal can only pass through the filter if the signal stabilization time exceeds 16 filter cycles. **Filter width = filter clock period*16.**

See more in Fail Signal Processing.

12.2.27 MCPWM_PRT

Unprotected register

Address: 0x4001_1C74

Reset value: 0x0

Table 12-30 MCPWM_PRT Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	PRT	Write 0xDEAD to release the write protection of the MCPWM register; write other values, the MCPWM register enters write protection state. The register always reads out 0.

12.2.28 MCPWM_CNT

Unprotected register

Address: 0x4001_1C78

Reset value: 0x0

Table 12-31 MCPWM_CNT Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	CNT	Write to this register to change the set value of MCPWM_CNT. After an update event occurs, this register is loaded into the CNT of the MCPWM actual operating system. The read data is the counter value in MCPWM actual operating system. The actual read count range is 0x8000-TH ~ 0x8000+TH

12.2.29 MCPWM_SWAP

Write-protected register

Address: 0x4001_1C7C

Reset value: 0x0

Table 12-32 MCPWM_SWAP Register

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

	MCPWM_SWAP
	RW
	0

Location	Bit name	Description
[31:1]		Unused
[0]	MCPWM_SWAP	Write 0x67 to such register to write BIT[0] to 1, and write other values to write BIT[0] to 0

When the value of MCPWM_SWAP is 0, the relationship between MCPWM channel output and GPIO is as follows:

Table 12-33 MCPWM Default Output Table

MCPWM Output Sequence	GPIO Corresponding Sequence
MCPWM_CH0P	P1.4
MCPWM_CH0N	P1.5
MCPWM_CH1P	P1.6
MCPWM_CH1N	P1.7
MCPWM_CH2P	P1.8
MCPWM_CH2N	P1.9
MCPWM_CH3P	P1.10
MCPWM_CH3N	P1.11

When the value of MCPWM_SWAP is 1, it is used to include the pre-drive chip application environment. The sequence is converted within the logic to facilitate the interconnection of the chip and the drive chip. In general applications, only three sets of MCPWM channels are required, so only three sets of sequences are converted. The relationship is as follows:

Table 12-34 MCPWM Output Table after Modification

MCPWM Output Sequence	GPIO Corresponding Sequence
MCPWM_CH0N	P1.4
MCPWM_CH1N	P1.5
MCPWM_CH2N	P1.6
MCPWM_CH0P	P1.7
MCPWM_CH1P	P1.8
MCPWM_CH2P	P1.9
MCPWM_CH3P	P1.10
MCPWM_CH3N	P1.11

12.2.30 MCPWM_DTH00

Write-protected register

Address: 0x4001_1C80

Reset value: 0x0

Table 12-35 MCPWM_DTH00 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						DTH00									
						RW									
						0									

Location	Bit name	Description
[31:10]		Unused
[9:0]	DTH00	MCPWM CH0 P channel dead-zone width control register, 10-bit no-signed value

12.2.31 MCPWM_DTH01

Write-protected register

Address: 0x4001_1C84

Reset value: 0x0

Table 12-36 MCPWM_DTH01 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						DTH01									
						RW									
						0									

Location	Bit name	Description
[31:10]		Unused
[9:0]	DTH01	MCPWM CH0 N channel dead-zone width control register, 10-bit no-signed value

12.2.32 MCPWM_DTH10

Write-protected register

Address: 0x4001_1C88

Reset value: 0x0

Table 12-37 MCPWM_DTH10 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						DTH10									
						RW									
						0									

Location	Bit name	Description

[31:10]		Unused
[9:0]	DTH10	MCPWM CH1 P channel dead-zone width control register, 10-bit no-signed value

12.2.33 MCPWM_DTH11

Write-protected register

Address: 0x4001_1C8C

Reset value: 0x0

Table 12-38 MCPWM_DTH11 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						DTH11									
						RW									
						0									

Location	Bit name	Description
[31:10]		Unused
[9:0]	DTH11	MCPWM CH1 N channel dead-zone width control register, 10-bit no-signed value

12.2.34 MCPWM_DTH20

Write-protected register

Address: 0x4001_1C90

Reset value: 0x0

Table 12-39 MCPWM_DTH20 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						DTH20									
						RW									
						0									

Location	Bit name	Description
[31:10]		Unused
[9:0]	DTH20	MCPWM CH2 P channel dead-zone width control register, 10-bit no-signed value

12.2.35 MCPWM_DTH21

Write-protected register

Address: 0x4001_1C94

Reset value: 0x0

Table 12-40 MCPWM_DTH21 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						DTH21									
						RW									
						0									

Location	Bit name	Description
[31:10]		Unused
[9:0]	DTH21	MCPWM CH2 N channel dead-zone width control register, 10-bit no-signed value

12.2.36 MCPWM_DTH30

Write-protected register

Address: 0x4001_1C98

Reset value: 0x0

Table 12-41 MCPWM_DTH30 Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						DTH30									
						RW									
						0									

Location	Bit name	Description
[31:10]		Unused
[9:0]	DTH30	MCPWM CH3 P channel dead-zone width control register, 10-bit no-signed value

12.2.37 MCPWM_DTH31

Write-protected register

Address: 0x4001_1C9C

Reset value: 0x0

Table 12-42 MCPWM_TH31 Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						DTH31									

	RW
	0

Location	Bit name	Description
[31:10]		Unused
[9:0]	DTH31	MCPWM CH3 N channel dead-zone width control register, 10-bit no-signed value

13 UART

13.1 Introduction

The Universal Asynchrounous Receiver/Transmitter, usually called UART, is an asynchronous receiver/transmitter.

The main features of UART are as follows:

Full-duplex operation

Support 7/8 data bit

Support 1/2 stop bit

Support odd/even/no parity mode

1 byte tx buffer

1 byte rx buffer

Support Multi-drop Slave/Master mode

13.2 Functions

13.2.1 Transmission

The UART includes a byte tx buffer. When the tx buffer has data, the UART loads the data of the tx buffer and sends it out via TX.

After the loading is completed, the tx buffer empty interrupt is generated. Then, user can fill the tx buffer with the next byte to be sent. After the transmission is completed, the UART will load this byte for transmission.

After the transmission is completed, a transmission completion interrupt will be generated.

13.2.2 Receiving

The UART includes a byte of receiving buffer area. When a byte is received, a receiving interrupt will be generated and the received byte will be stored in the receiving buffer area; the user should finish reading this byte before receiving the next byte in the UART; otherwise, the buffer area will be written to the newly received byte.

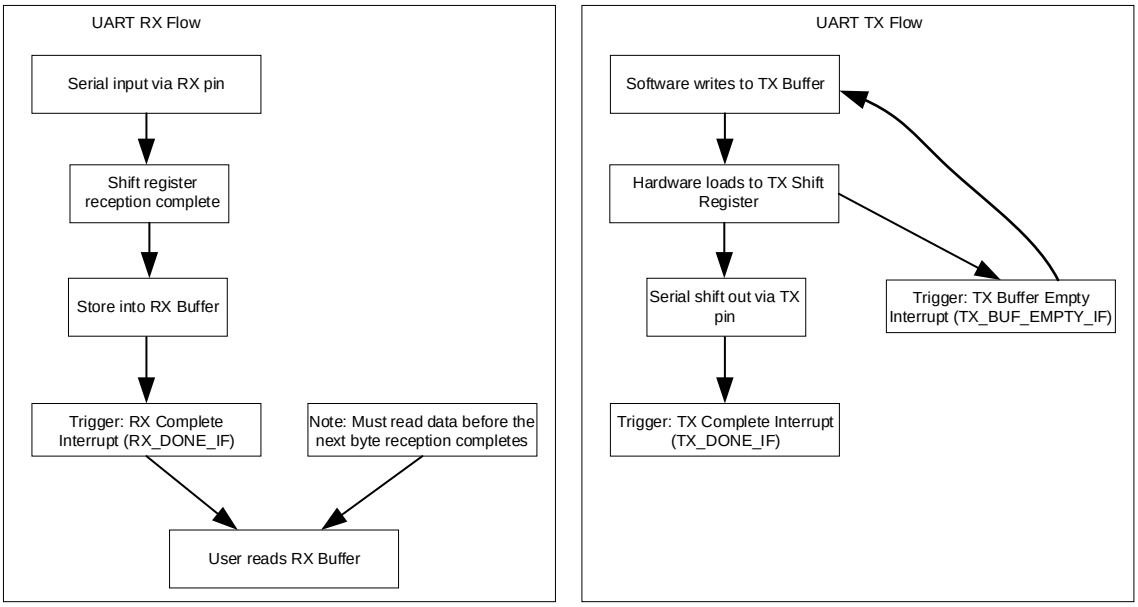


Fig. 13-1 UART Transmit and Receive Interrupts

13.2.3 Baud Rate Configuration

The UART input clock is the main clock, and the baud rate is realized by two-stage frequency division.

Baud rate = UART module clock/（256*DIVH+DIVL+1)

The UART module clock can be divided by SYS_CLK_DIV2

UART module clock = system main clock (1+SYS_CLK_DIV2)

Table 13-1 Example of UART Baud Rate Configuration

UART baud rate	SYS_CLK_DIV2	UART_DIVH	UART_DIVL
300	0x0007	0x9C	0x3F
600	0x0003	0x9C	0x3F
1200	0x0001	0x9C	0x3F
2400	0x0000	0x9C	0x3F
4800	0x0000	0x4E	0x1F
9600	0x0000	0x27	0x0F
19200	0x0000	0x13	0x87
38400	0x0000	0x09	0xC3
43000	0x0000	0x08	0xB8
56000	0x0000	0x06	0xB1
57600	0x0000	0x06	0x82
115200	0x0000	0x03	0x40

Note: The baud rate configuration factor is only an example and may not be unique.

13.2.4 Transmission and Receiving Port Interchange (TX/RX Exchange)

UART module supports Tx and Rx port interchange. By configuring the GPIO corresponding to Tx as input enable and the GPIO corresponding to Rx as output enable, the Tx and Rx ports can be interchanged. At this moment, the second function of GPIO is still selected as UART, and the configuration of UART itself does not need to be modified.

In addition, if you want to use a GPIO as Tx and Rx at the same time, you need to multiplex the IO as input or output by time division, corresponding to Rx or Tx, to achieve single-port half-duplex logic.

13.3 Register

13.3.1 Address Allocation

UART0 and UART1 are identical.

UART0 base address is 0x40012800.

UART1 base address is 0x40012C00.

Table 13-2 UARTx Address Allocation List

Name	Offset address	Description
UARTx_CTRL	0x00	UART Control Register
UARTx_DIVH	0x04	UART high byte register with UART baud rate setting
UARTx_DIVL	0x08	UART Low byte register with UART baud rate setting
UARTx_BUFF	0x0C	UART receiving and sending buffer register
UARTx_ADR	0x10	485 communication address matching register
UARTx_STT	0x14	UART status register
UARTx_IE	0x18	UART interrupt enable register
UARTx_IF	0x1C	UART interrupt flag register
UARTx_IOC	0x20	UART IO control

13.3.2 UARTx_CTRL UARTx Control Register (x = 0,1)

UART0_CTRL address: 0x4001_2800

UART1_CTRL address: 0x4001_2C00

Reset value: 0x0

Table 13-3 UARTx_CTRL UARTx Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									MDMASTER_BIT9	MD_EN	CK_EN	CK_TYPE	BIT_ORDER	STOP_LEN	DAT_LEN
									RW	RW	RW	RW	RW	RW	RW

	0	0	0	0	0	0	0

Location	Bit name	Description
[31:8]		Unused
[6]	MDMASTER_BIT9	The 9th data bit value in Multi-drop Master mode
[5]	MD_EN	Enable Multi-drop. Default value: 0. 0: disable, 1: enable
[4]	CK_EN	Enable data verification, default value: 0. 0: disable, 1: enable
[3]	CK_TYPE	Parity check. Default value: 0. 0: EVEN 1: ODD
[2]	BIT_ORDER	Data sending sequence configuration, default value: 0. 0: LSB； 1:MSB
[1]	STOP_LEN	Stop bit length configuration, default value: 0. 0:1-Bit； 1:2-Bit
[0]	DAT_LEN	Data length configuration, default value: 0 0:8-Bit； 1:7-Bit

13.3.3 UARTx_DIVH UARTx Baud Rate High-byte Register (x = 0,1)

UART0_CTRL address: 0x4001_2804

UART1_CTRL address: 0x4001_2C04

Reset value: 0x0

Table 13-4 UARTx_DIVH UARTx Baud Rate High-byte Register

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

Location	Bit name	Description
[31:8]		Unused
[7:0]	DIVH	Baud rate setting high byte BAUDRATE =main clock/(1+256* UARTx_DIVH+UARTx_DIVL)

13.3.4 UARTx_DIVL UARTx Baud Rate Low-byte Registe (x = 0,1)

UART0_CTRL address: 0x4001_2808

UART1_CTRL address: 0x4001_2C08

Reset value: 0x0

Table 13-5 UARTx_DIVL UARTx Baud Rate Low-byte Register

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

	RW
	0

Location	Bit name	Description
[31:8]		Unused
[7:0]	DIVL	Baud rate setting low byte BAUDRATE =main clock/(1+256* UARTx_DIVH+UARTx_DIVL)

13.3.5 UARTx_BUFF UARTx Transceiver Buffer Register (x = 0,1)

Table 13-6 UARTx_BUFF UARTx Transceiver Buffer Register

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

Location	Bit name	Description
[31:8]		Unused
[7:0]	BUFF	Write: data transmit buffer; read: data receive register

The Tx_buffer and Rx_buffer of the UART share UARTx_BUFF. Among them, Tx_buffer is write-only, Rx_buffer is read-only. Therefore, read access to UARTx_BUFF is to access UARTx_RX_BUFF, and write access to UARTx_BUFF is to access UARTx_TX_BUFF.

13.3.6 UARTx_ADR UARTx Address Match Register (x = 0,1)

UART0_CTRL address: 0x4001_2810

UART1_CTRL address: 0x4001_2C10

Reset value: 0x0

Table 13-7 UARTx_ADR UARTx Address Match Register

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

Location	Bit name	Description
[31:8]		Unused
[7:0]	ADR	Used as address for 485 communication

13.3.7 UARTx_STT UARTx Status Register (x = 0,1)

UART0_CTRL address: 0x4001_2814

UART1_CTRL address: 0x4001_2C14

Reset value: 0x0003

Table 13-8 UARTx_STT UARTx Status Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ADR_MATCH	TX_DONE	TX_BUF_EMPTY
													R	R	R
													0	1	1

Location	Bit name	Description
[31:3]		Unused
[2]	ADR_MATCH	Address match flag in Multi-drop mode. 1: match; 0: not match.
[1]	TX_DONE	Sending completion flag bit. 1: completed; 0: not completed.
[0]	TX_BUF_EMPTY	Sending buffer status bit. 1: empty; 0: not empty.

13.3.8 UARTx_IE UARTx Interrupt Enable Register (x = 0,1)

UART0_CTRL address: 0x4001_2818

UART1_CTRL address: 0x4001_2C18

Reset value: 0x0

Table 13-9 UARTx_IE UARTx Interrupt Enable Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											CK_ERR_IE	STOP_ERR_IE	TX_BUF_EMPTY_IE	RX_DONE_IE	TX_DONE_IE
											RW	RW	RW	RW	RW
											0	0	0	0	0

Location	Bit name	Description
[31:5]		Unused
[4]	CK_ERR_IE	Check error interrupt switch. The default value is 0.

		0: Off; 1: On.
[3]	STOP_ERR_IE	Stop bit error interrupt switch. The default value is 0. 0: Off; 1: On.
[2]	TX_BUF_EMPTY_IE	Sending buffer empty interrupt switch. The default value is 0. 0: Off; 1: On.
[1]	RX_DONE_IE	Receiving completion interrupt switch. The default value is 0. 0: Off; 1: On.
[0]	TX_DONE_IE	Sending completion interrupt switch. The default value is 0. 0: Off; 1: On.

13.3.9 UARTx_IF UARTx Interrupt Flag Register (x = 0,1)

UART0_CTRL address: 0x4001_281C

UART1_CTRL address: 0x4001_2C1C

Reset value: 0x0005

Table 13-10 UARTx_IF UARTx Interrupt Enable Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											CK_ERR_IF	STOP_ERR_IF	TX_BUF_EMPTY_IF	RX_DONE_IF	TX_DONE_IF
											RW1C	RW1C	RW1C	RW1C	RW1C
											0	0	1	0	1

Location	Bit name	Description
[31:5]		Unused
[4]	CK_ERR_IF	Check error interrupt flag, active high, write 1 to clear.
[3]	STOP_ERR_IF	Stop bit error interrupt flag, active high, write 1 to clear.
[2]	TX_BUF_EMPTY_IF	Sending buffer area empty interrupt flag, active high, write 1 to clear.
[1]	RX_DONE_IF	Receiving completion interrupt flag, active high, write 1 to clear.
[0]	TX_DONE_IF	Sending completion interrupt flag, active high, write 1 to clear.

After power-on, the sending buffer area empty flag TX_BUF_EMPTY_IF and the sending completion interrupt flag TX_DONE_IF are 1 by default.

Write 1 to clear the interrupt flag. Generally, it is not recommended to clear with the following |= method, because |= is the first reading interrupt flag. Modify the corresponding bit to 1 and then re-write it to clear. If other interrupt flags are set at the same time, they will be cleared together, which is not expected by the software. For example, the following writing method is intended to clear TX_DONE_IF, but if RX_DONE_IF is set to 1 before the write execution, the software will firstly read back the UARTx_IF value 0x2, then execute the OR operation 0x2|0x1=0x3, write, and simultaneously RX_DONE_IF and TX_DONE_IF are cleared, which may cause the UART to enter one less interrupt due to the data receiving, and thus one less byte of data is received.

UARTx_IF|=0x1;

If you want to clear the TX_DONE_IF flag bit, you should write 1 directly to BIT0 as follows.

UARTx_IF=0x1;

13.3.10 UARTx_INV UARTx INV Control Register (x = 0,1)

UART0_INV address: 0x4001_2820

UART1_INV address: 0x4001_2820

Reset value: 0x0

Table 13-11 UARTx_INV UARTxINV Control Register

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

Location	Bit name	Description
[31:2]		Unused
[1]	TXD_INV	TXD output polarity enable switch. The default value is 0. 0: Normal output; 1: Inverted output. Normal output polarity means that when the application sends "1", the hardware sends "1"; Invert output polarity means that the when the application sends "1", the hardware sends "0".
[0]	RXD_INV	RXD input polarity enable switch. The default value is 0. 0: Normal input; 1: Inverted input. Normal input polarity means that when the hardware receives "1", the application receives "1"; Invert input polarity means that when the hardware receives "1", the application receives "0".

14 Digital Signal Coprocessor

14.1 Introduction

The radicand is a 32-bit unsigned number, and the square root is a 16-bit unsigned number.

The trigonometric CORDIC module has a bit width of 16 bits and a Q15 fixed-point format.

Square root calculation will be completed in 8 bus cycles (96 MHz).

CORDIC calculation will be completed in 16 bus cycles (96 MHz).

14.2 Register

14.2.1 Address Allocation

The base address of the digital signal coprocessor register in the chip is 0x4001_5000.

Table 14-1 Coprocessor Register List

Name	Offset	Description
DSP_SC	0x00	Coprocessor status control register
DSP_THETA	0x04	Coprocessor sin/cos input angle register
DSP_XY	0x08	Coprocessor arctan/module computer input coordinate XY register
	0x0C	Reserved
DSP_SIN	0x10	Coprocessor sin/cos calculation result sin register
DSP_COS	0x14	Coprocessor sin/cos calculation result cos register
DSP_MOD	0x18	Coprocessor arctan calculation result sqrt(X^2+Y^2) register
DSP_ARCTAN	0x1C	Coprocessor arctan calculation result arctan(Y/X) angle register
	0x20	Reserved
	0x24	Reserved
	0x28	Reserved
	0x2C	Reserved
DSP_RAD	0x30	Radicand of coprocessor
DSP_SQRT	0x34	Square root

14.2.2 DSP_SC Coprocessor Status Control Register

Address: 0x4001_5000

Reset value: 0x0

Table 14-2 DSP_SC Processor Status Control Register

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

	RW	
	0	

Location	Bit name	Description
[31:3]	RESERVED	Reserved
[2]	CORDIC_MODE	CORDIC mode, 0: arctan, 1: sin/cos
[1:0]	RESERVED	Reserved

CORDIC_MODE bit, access to CORDIC module by MCU through the bus. sin/cos mode and arctan mode selection, sin/cos or arctan calculation by CORDIC module share the same hardware circuit. Therefore, before performing a certain calculation, you should configure the DSP_SC calculator to select an appropriate mode. When the CORDIC module calculates sin/cos, the angle theta is used as the input, and the sin/cos result is calculated and output; When calculating arctan, the coordinates X/Y are used as input, and the angles theta = arctan (y/x) and module = sqrt (x^2+y^2) are calculated and output.

14.2.3 Coprocessor sin/cos Register

Since the calculation of sin/cos and arctan in the Cordic module calculation uses the same data path, it's necessary to write DSP_SC[2] to 1 before performing the calculation of sin/cos with the Cordic module through the CPU, to make cordic enter sin/cos mode.

14.2.3.1 DSP_THETA

Address: 0x4001_5004

Reset value: 0x0

Table 14-3 Coprocessor sin/cos Angle Input Register

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

Location	Bit name	Description
[31:16]		Reserved bit. Sign extension when reading, i.e. {16 {DSP_THETA [15]}}
[15:0]	THETA	Coprocessor sin/cos input angle register

DSP_THETA is a 16-bit signed fixed-point number, indicating that the range (-32768 ~ 32767) corresponds to (-π ~ π).

14.2.3.2 DSP_SIN

Address: 0x4001_5010

Reset value: 0x0

Table 14-4 Coprocessor sin/cos Sine Result Register

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

Location	Bit name	Description
[31:16]		Reserved bit. Sign extension when reading, i.e. {16{DSP_SIN[15]}}
[15:0]	SIN	Coprocessor sin/cos calculation result sin register

DSP_SIN is a 16-bit signed fixed-point number, including 1-bit sign bit, 1-bit integer bit, and 14-bit decimal bit; representing the range (–1 ~ 1).

14.2.3.3 DSP_COS

Address: 0x4001_5014

Reset value: 0x0

Table 14-5 Coprocessor sin/cos Cosine Result Register

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

Location	Bit name	Description
[31:16]		Reserved bit. Sign extension when reading, i.e. {16 {DSP_COS[15]}}
[15:0]	COS	Coprocessor sin/cos calculation result cos register

DSP_COS is a 16-bit signed fixed-point number, including 1-bit sign bit, 1-bit integer bit, and 14-bit decimal bit; representing the range (–1 ~ 1).

14.2.4 Co-processor Arctan Register

14.2.4.1 DSP_XY

Address: 0x4001_5008

Reset value: 0x0

Table 14-6 Coprocessor Arctan/Module Coordinate Input Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
X															

WO
0

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

Location	Bit name	Description
[31:16]	X	Coprocessor arctan/module computer input coordinate X register
[15:0]	Y	Coprocessor arctan/module computer input coordinate Y register

DSP_MOD is a 16-bit signed fixed-point number, including 1-bit sign bit, 15-bit integer bit; representing the range (–32768 ~ 32767).

14.2.4.2 DSP_MOD

Address: 0x4001_5018

Reset value: 0x0

Table 14-7 Coprocessor arctan Angle Result arctan(Y/X) Angle Register

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

Location	Bit name	Description
[31:16]		Reserved bit. Always read as 0
[15:0]	MOD	Coprocessor arctan calculation result sqrt(X^2 +Y^2) register

DSP_MOD is a 16-bit signed fixed-point number, including 1-bit sign bit, 15-bit integer bit; representing the range (–32768 ~ 32767).

14.2.4.3 DSP_ARCTAN

Address: 0x4001_501C

Reset value: 0x0

Table 14-8 Coprocessor arctan Angle Result arctan(Y/X) Angle Register

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

Location	Bit name	Description
[31:16]		Reserved bit. Sign extension when reading, i.e. {16{DSP_ARCTAN[15]}}
[15:0]	ARCTAN	Coprocessor arctan calculation result arctan(Y/X) angle register

DSP_ARCTAN is a 16-bit signed fixed-point number, indicating that the range (-32768 ~ 32767) corresponds to (-π ~ π).

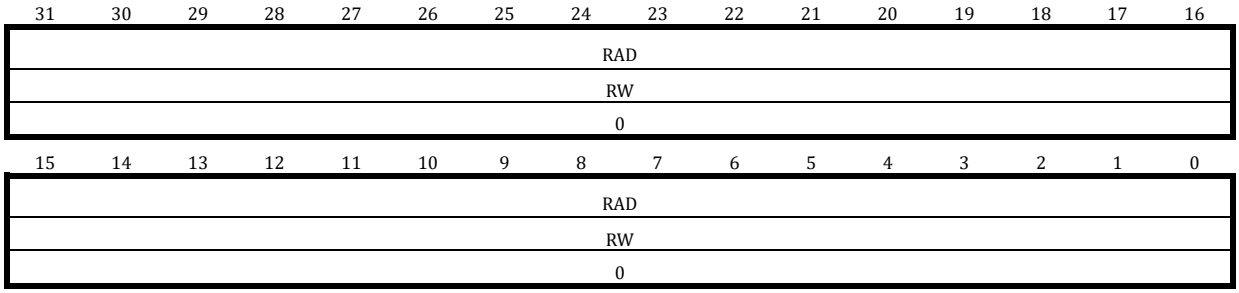
14.2.5 Co-processor Root Extraction Register

14.2.5.1 DSP_RAD

Address: 0x4001_5030

Reset value: 0x0

Table 14-9 Coprocessor Radicand Register



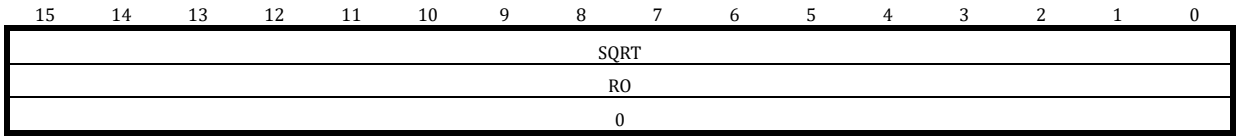
Location	Bit name	Description
[31:0]	RAD	Radicand of coprocessor

14.2.5.2 DSP_SQRT

Address: 0x4001_5024

Reset value: 0x0

Table 14-10 Co-processor Square Root Register



Location	Bit name	Description
[31:16]		Reserved bit. Always read as 0
[15:0]	SQRT	Co-processor Square Root Register

Write the radicand to the coprocessor; write the adicand to trigger a square root operation; 32-bit root extraction needs to be completed in 8 cycles. During the period, reading the square root

result DSP_SQRT to make the MCU enter the waiting status, waiting that the square root calculation is completed and the calculation result is returned through the bus.

15 I2C

15.1 Introduction

The I2C bus interface connects the microcontroller and the serial I2C bus. It provides multi-master functions to control the specific timing, protocol, arbitration and timing of all I2C buses. Besides, it supports standard and fast modes.

15.2 Main Features

- Multi-master function: this module can be used as both master and slave.

  I2C master device function: generate clock, START and STOP events.

  I2C slave device functions: programmable I2C hardware address comparison (only supports 7-bit hardware address), stop bit detection.

- Provide different communication speeds depending on the frequency division.
- Status flags: transmitter/receiver mode flag, byte transmission end flag, I2C bus busy flag.
- Error flags: Loss of arbitration in master mode, acknowledgment (ACK) error after address/data transmission, start or stop condition where misalignment was detected.
- An interrupt vector contains five interrupt sources: bus error interrupt source, completion interrupt source, NACK interrupt source, hardware address matching interrupt source, and transfer completion interrupt source.

15.3 Functional Description

15.3.1 Functional Block Diagram

This interface adopts a synchronous serial design to achieve I2C transmission between the MCU and external devices, and supports polling and interrupt mode to obtain transmission status information. The main functional modules of this interface are shown in the figure below.

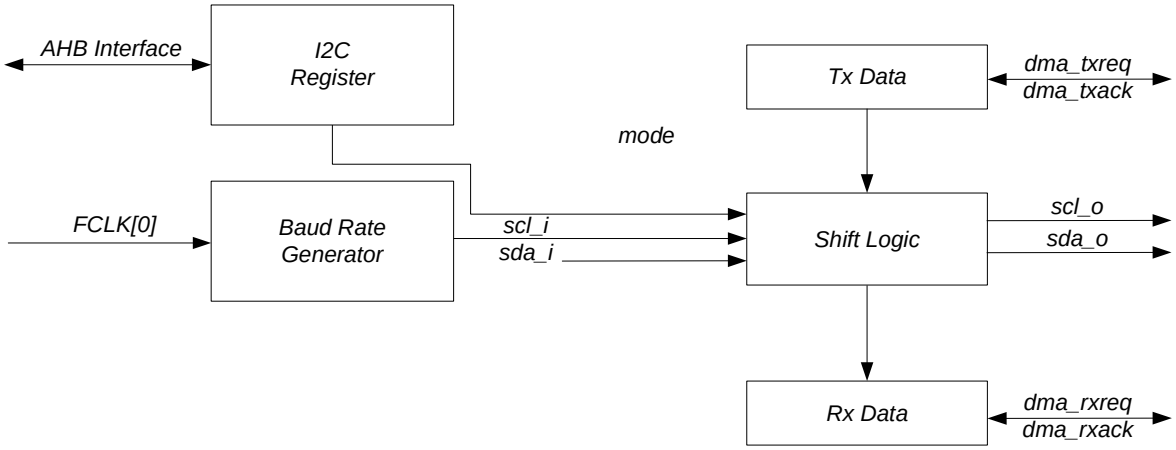


Fig. 15-1 I2C Module Top Functional Block Diagram

The I2C interface communicates with the outside world only with two signal lines, SCL and SDA. SDA is a bidirectional multiplexed signal line, controlled by sda_oe. Module-level I2C interface signals include scl_i, sda_i, scl_o, sda_o, and sda_oe.

scl_i: clock signal. When the I2C interface is set as slave mode, this is the clock input signal for the I2C bus.

sda_i: data signal. When the I2C interface receives data (regardless of master mode or slave mode), this is the data input signal of the I2C bus.

scl_o: clock signal. When the I2C interface is set as the main mode, this is the clock output signal of the I2C bus.

sda_o: data signal. When the I2C interface sends data (regardless of master mode or slave mode), this is the data output signal of the I2C bus.

sda_oe: data enable signal. When sda_o is output, sda_oe is valid; when sda_i is input, sda_oe is invalid.

15.3.2 Functions

The I2C module receives and sends data, and converts the data from serial to parallel, or parallel to serial , and can enable or disable interrupts. The interface is connected to the I2C bus via data pins (SDA) and clock pins (SCL).

15.3.2.1 Mode selection

The interface can operate in one of the following four modes:

- Slave sending mode
- Slave receiving mode
- Master sending mode
- Master receiving mode

The I2C interface is not enabled by default. The interface enters master mode or slave mode

according to the configuration. When arbitration is lost or a stop signal is generated, the master mode releases the bus automatically and generates an abnormal interrupt. Multiple host functions is available.

In master mode, the I2C interface starts data transmission and generates a clock signal. Serial data transmission always starts with a start condition and ends with a stop condition. The start condition and stop condition are generated by software control in the master mode.

In slave mode, the I2C interface can recognize its own address (7 bits). The software can control to enable or disable the hardware address comparison function, which can reduce the burden on the MCU. Only when the addresses match, the MCU is notified to perform relevant processing.

The data and address are transmitted in 8 bits/byte, with the high order first. The one byte following the start condition is the address, and the address is only sent in master mode.

During the ninth clock after the eight clocks for one byte transmission, the receiver must send back an acknowledge bit (ACK) to the transmitter.

The software can enable or disable acknowledgement (ACK), and can set the address of the I2C interface.

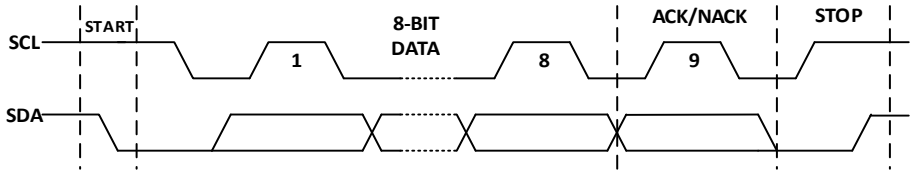


Fig. 15-2 Basic I2C Transmission Timing Diagram

In general, one byte at a time (single transmission can be repeated, and the data should be provided by software). All the above modes follow the basic principles below:

- Single byte transmission. An interrupt will be generated after the 8-bit data is sent and the response is received (either ACK/NACK).
- Single byte reception. An interrupt will be generated after the 8-bit data is received.
- When the I2C interface is set as the mastermode, the I2C interface will release the bus after detecting an error, restore to the initial state and generate an interrupt signal.

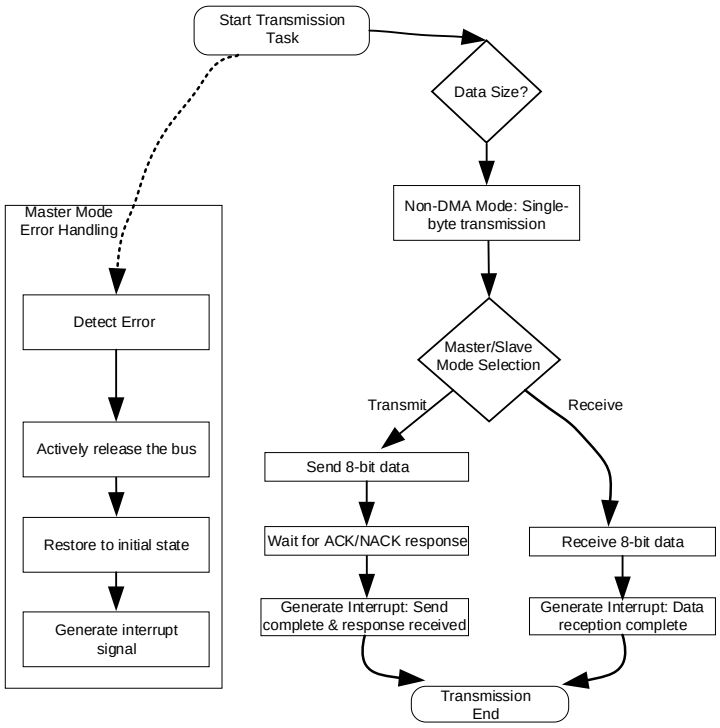


Fig. 15-3 I2C Transmission Mode Decision and Execution Flowchart

15.3.2.2 Slave Mode

Both the master mode and slave mode of the I2C interface are turned off by default. If operating in slave mode, the slave mode should be enabled. In order to generate the correct timing, the operating clock frequency of the I2C interface must be set by the register SYS_CLK_DIV0.

- In slave mode, the I2C interface monitors the signals on the bus at all times. Once the start condition is detected, it will save the address bit data and read-write bit data.
- In slave mode, if the hardware address matching function is enabled, an interrupt will be generated and the MCU will be notified for subsequent processing only when the addresses match. If the function is not enabled, an interrupt will be generated each time the address and read/write bit data is received.
- Slave Mode Receiving. Every time a byte of data is received, an interrupt is generated. At this time, the I2C interface can pull down SCL and continue subsequent operations until the interrupt is completed.
- Slave Mode Sending. After a byte is sent and a response (ACK/NACK) is received each time, an interrupt is generated. At this time, the I2C interface can pull down SCL until the interrupt is completed and continue subsequent operations.

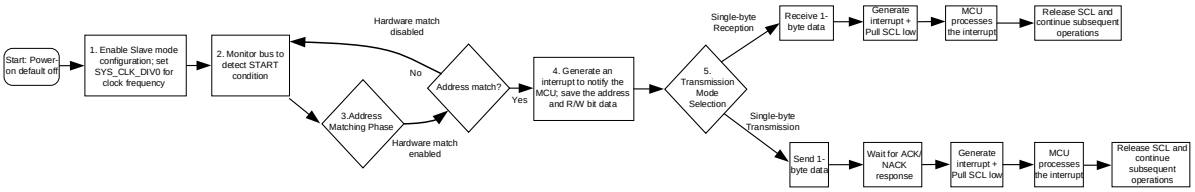


Fig. 15-4 I2C Slave Mode Workflow Diagram

15.3.2.2.1 Slave Mode Transmission

After a byte of data is transmitted each time, the hardware will generate an interrupt, and the software will determine whether to continue the transmission. Fig. 15-3 Schematic Diagram of Transmission in Slave Mode The figure shows the following process flow:

- Address match, generate address match interrupt, ready to start transmission.
- In the receiving mode, an interrupt is generated after a byte is received, the software determines whether to continue receiving, and returns an ACK/NACK response.
- In the sending mode, an interrupt is generated when receving the reponse (ACK/NACK) after a byte is sent, and subsequent operations are judged based on the response.
- Obtain the bus STOP event, this transmission is completed.

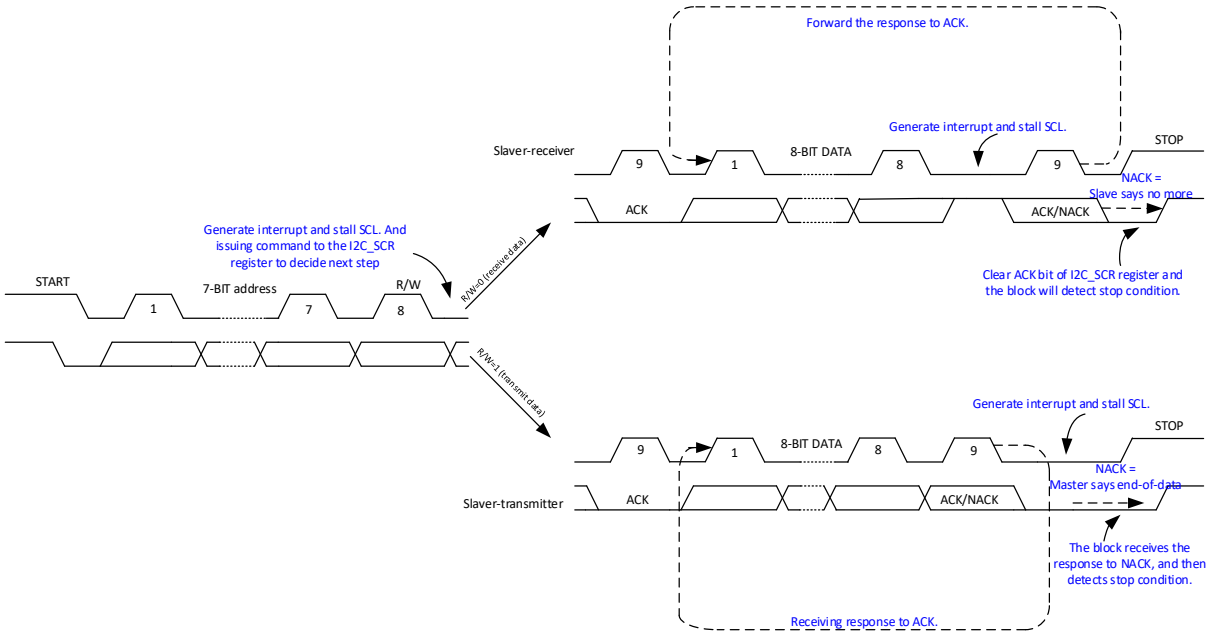


Fig. 15-5 Schematic Diagram of Transmission in Slave Mode

15.3.2.2.2 Slave Mode Sending

After the address matches, the slave sends the byte from the I2C_DATA register to the SDA line via the internal shift register. Before the I2C_DATA data is ready, the slave device can lower the SCL until the data to be sent has been written to the I2C_DATA register. The I2C interface performs the following operations after sending each byte:

- If the ACK bit is received, the next byte of data is loaded and the transmission continues. The SCL can be lowered during the loading process. The SCL can be lowered during the loading process.
- If the NACK bit is received, stop loading the next byte.
- A STOP event is generated to end this transmission.

15.3.2.2.3 Master Mode Single-byte Receiving

After the address matches, the data received from the SDA line through the internal shift register is stored in the I2C_DATA register. The I2C interface performs the following operations after receiving each byte:

- If the ACK bit is set, an ACK response pulse is generated after a byte is received.
- If the ACK bit is cleared, a NACK response pulse is generated after a byte is received.
- A STOP event is generated to end this transmission.

15.3.2.3 Master Mode

Both the master mode and slave mode of the I2C interface are turned off by default. If operating in master mode, the master mode should be enabled. In order to generate the correct timing, the operating clock frequency of the I2C interface must be set by the register SYS_CLK_DIV0.

Judge whether the bus is idle before performing the transmission via I2C interface in master mode Read BIT3 of the I2C_MSCR register to query the current bus status. If the bus is busy, turn on the I2C interrupt and determine whether the bus is idle by receiving the STOP interrupt event. Only in the idle state can the START state and subsequent data be sent normally.

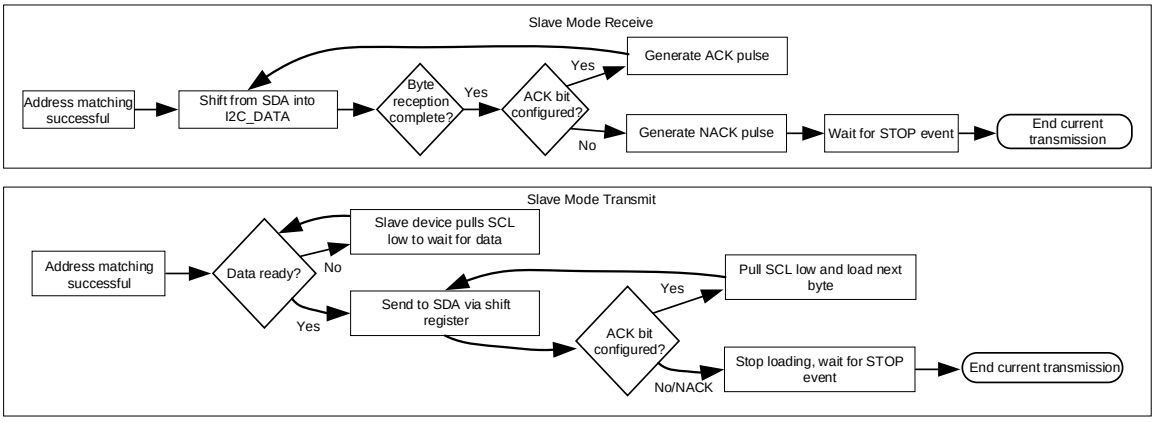


Fig. 15-6 I2C Slave Mode (Transmit/Receive) Logic Flowchart

15.3.2.3.1 Master Mode Sending

After each byte of data is transmitted, an interrupt will be generated to determine whether to continue the transmission. The following figure is a schematic diagram of a bus for master mode transmission. The figure shows the following process flow:

- Determine whether the bus is idle, if it is idle, prepare to start transmission.
- Firstly, send the slave address. If the address matches, continue the subsequent transmission. Otherwise, stop it.
- In the receiving mode, an interrupt is generated after a byte is received, the software determines whether to continue receiving, and returns an ACK/NACK response.
- In the sending mode, an interrupt is generated when receving the reponse (ACK/NACK) after a byte is sent, and subsequent operations are judged based on the response.
- Send the bus STOP event, and this transmission is completed

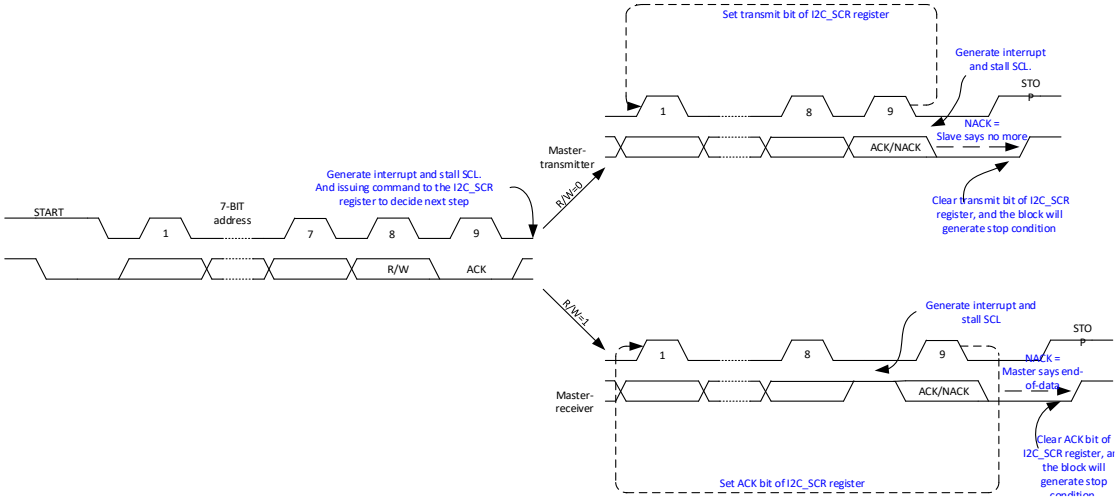


Fig. 15-7 Schematic Diagram of Single-byte Transmission in Master Mode

15.3.2.3.2 Master Mode Sending

After the transmission starts, the I2C interface sends the byte from the I2C_DATA register to the SDA line via the internal shift register, and sends the slave address through I2C_DATA. Before the I2C_DATA data is ready, the master device may not generate the SCL clock signal until the data to be sent has been written to the I2C_DATA register. The I2C interface performs the following operations after sending each byte:

- If the ACK bit is received, the next byte of data is loaded and the transmission continues. The SCL can be lowered during the loading process. The SCL can be lowered during the loading process.

- If the NACK bit is received, stop loading the next byte.
- For the master equipment, complete transmission. No matter whether ACK/NACK is received, subsequent transmissions are stopped.
- A STOP event is generated to end this transmission.

15.3.2.3.3 Master Mode Receiving

After the transmission is started, the I2C interface stores the data received from the SDA line through the internal shift register into the I2C_DATA register, and the slave address is also sent out through I2C_DATA. The I2C interface performs the following operations after receiving each byte:

- If the ACK bit is set, an ACK response pulse is generated after a byte is received.
- If the ACK bit is cleared, a NACK response pulse is generated after a byte is received.
- For the master equipment, complete transmission. No matter whether ACK/NACK is sent, subsequent transmissions are stopped.
- A STOP event is generated to end this transmission.

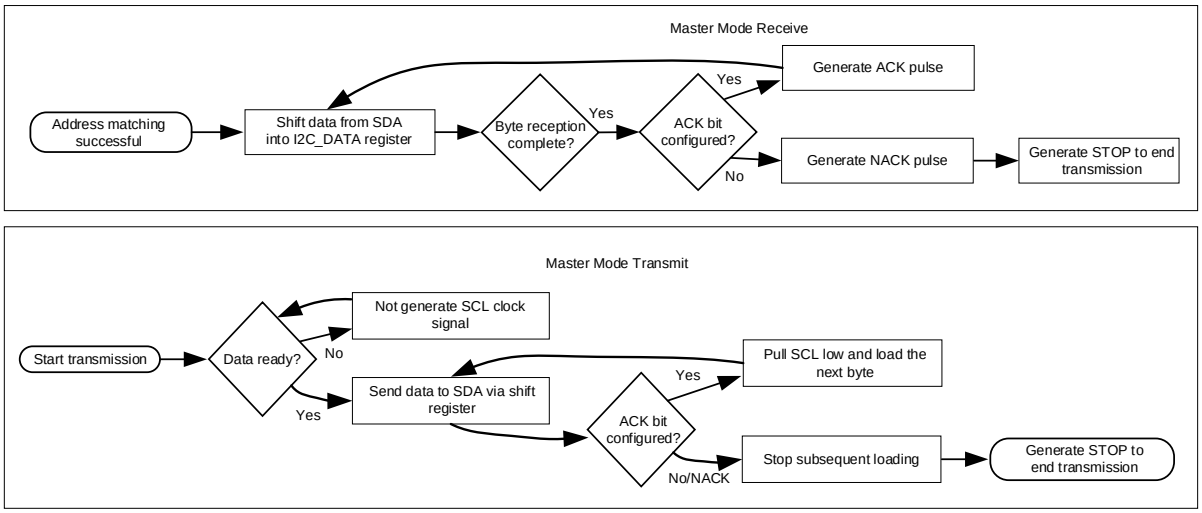


Fig. 15-8 Logic Flow Chart for I2C Master Mode Transmission and Reception

15.3.2.4 I2C Bus Exception Handling

During an address or data byte transmission, a bus error is generated when the I2C interface detects an external stop or start condition. Generally speaking, the bus error is caused by interference on the bus, and some I2C devices are not synchronized with the I2C network, resulting in a START event/STOP event sent automatically. According to the I2C protocol, when a bus error occurs, the interface logic of this I2C device must be reset after receiving a START event/STOP event. For the slave device, this operation is OK; for the master device, a bus error will force it to release the bus and reset its I2C interface logic. Since the master device does not respond to external START and STOP events, an interrupt handler function is required to handle this exception after a bus error

occurred, and instruct the master device to continue to monitor the bus situation, so as to perform subsequent I2C bus transmission. Because the master device does not respond to external START and STOP events, after a bus error occurs, an interrupt handling function is required to handle the exception and guide the master device to continuously monitor the bus condition for subsequent I2C bus transmission.

I2C Interface. In master mode, a bus error can be detected and a bus error interrupt will also be generated; in slave mode, a bus error will trigger address data to be received, and the I2C interface will be made to return to the idle status and generate an interrupt.

15.3.2.5 Interrupt Handling

The I2C interface includes three types of interrupt events: data transmission completion event, bus error event, STOP event, NACK event and hardware address matching event.

- Data transmission completion event. During the transmission process, the bus generates an erroneous START event/STOP event. Active high, write 0 to clear I2C_SCR.Done.
- Bus error event. During the transmission, the bus generates an erroneous START event/STOP event, which is active at high level. Write 0 to clear I2C_SCR.STT_ERR.
- STOP event. When the current data transmission is completed, the master device sends a STOP event. The slave device receives a STOP event and generates a corresponding interrupt. Active high, write 0 to clear I2C_SCR.STOP_EVT.
- NACK event. The sender receives a NACK response, indicating that the receiver cannot continue subsequent transmissions. Active high, write 0 to clear I2C_SCR.RX_ACK.
- Hardware address matching event. The address received in slave mode matches the address of this device, and a corresponding interrupt is generated. Active high, write 0 to clear I2C_SCR.ADDR_DATA.

15.3.2.6 Communication Speed Setting

The working clock of the I2C interface comes from the frequency division of the system clock. The frequency division register is CLK_DIV0 of the SYS module.

The I2C interface adopts a synchronous design, and the signals of external devices should be synchronously sampled. The synchronous clock is the working clock of the I2C interface.

- I2C module operating clock frequency = system frequency/ (CLK_DIV0 + 1)
- The Serial data(SDA) and Serial clock line(SCL) clock frequency = I2C module operating clock frequency/17
- I2C baud rate = I2C module operating clock frequency/17

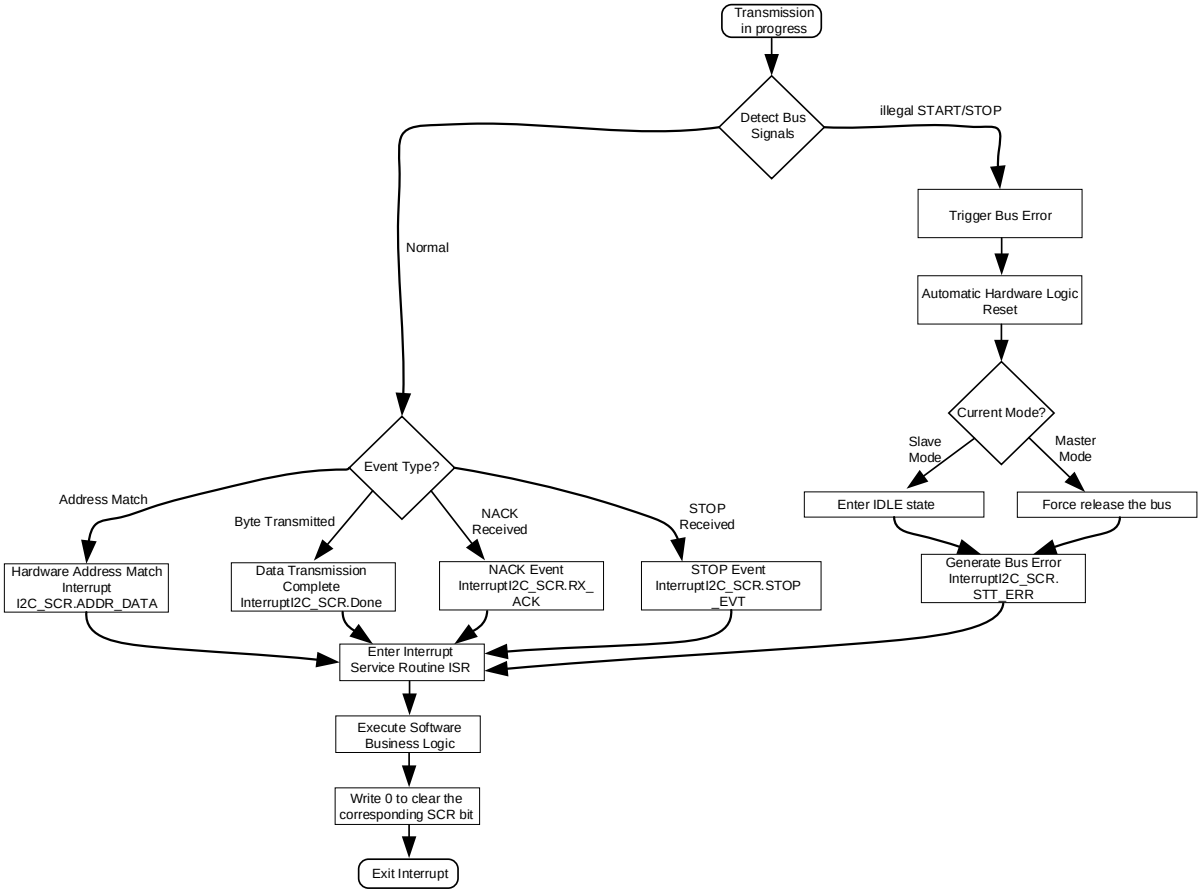


Fig. 15-9 Logic Flow Chart for I2C Bus Error Handling and Interrupt Management

15.4 Register

15.4.1 Address Allocation

The base address of the I2C module register is 0x4001_0400.

Table 15-1 I2C Register Address Allocation List

Name	Offset	Description
I2C_ADDR	0x00	I2C Address Register
I2C_CFG	0x04	I2C Configuration Register
I2C_SCR	0x08	I2C status register
I2C_DATA	0x0C	I2C data register
I2C_MSCR	0x10	I2C master mode register
I2C_BCR	0x14	I2C transmission control register

15.4.2 I2C_ADDR Address Register

Address: 0x4001_0400

Reset value: 0x0

Table 15-2 I2C_ADDR Address Register

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

Location	Bit name	Description
[31:8]		Unused
[7]	ADDR_CMP	I2C hardware address comparison enable switch. The default value is 0. 0: OFF 1: ON
[6:0]	ADDR	Only used for I2C device hardware address in slave mode. The salve device address needs to be written into the I2C_DATA register in the master mode.

15.4.3 I2C_CFG System Control Register

Address:0x4001_0404

Reset value: 0x0

Table 15-3 I2C_CFG System Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								IE	TC_IE	BUS_ERR_IE	STOP_IE			MST_MODE	SLV_MODE
								RW	RW	RW	RW			RW	RW
								0	0	0	0			0	0

Location	Bit name	Description
[31:8]		Unused
[7]	IE	I2C interrupt enable signal. The default value is 0. 1: enable I2C interrupt 0: disable I2C interrupt
[6]	TC_IE	I2C data transfer completion interrupt enable signal. The default value is 0. 1: enable this interrupt source 0: disable this interrupt source
[5]	BUS_ERR_IE	I2C bus error event interrupt enable signal. The default value is 0. 1: enable this interrupt source 0: disable this interrupt source
[4]	STOP_IE	I2C STOP event interrupt enable signal. The default value is 0. 1: enable this interrupt source 0: disable this interrupt source
[3:2]		Unused
[1]	MST_MODE	I2C master mode enable signal. The default value is 0. 1: enable master mode 0: disable master mode
[0]	SLV_MODE	I2C slave mode enable signal. The default value is 0.

		1: enable slave mode 0: disable slave mode

15.4.4 I2C_SCR Status Control Register

Address:0x4001_0408

Reset value: 0x0

Table 15-4 I2C_SCR Status Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								STT_ERR	LOST_ARB	STOP_EVT	BYTE_CMPLT	ADDR_DATA	DATA_DIR	RX_ACK	DONE
								RW	RW	RW	RW	RW	RW	RW	RW
								0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:8]		Unused
[7]	STT_ERR	Bus error status flag, only used in master mode, write 0 to clear. 0: no START/STOP bus error 1: START/STOP bus error
[6]	LOST_ARB	Bus arbitration lost status flag bit. Only used in master mode, this bit is set when a bus arbitration lost event occurs, no interrupt event is generated, and this bit should be checked in the byte completion interrupt. Any START event on the bus will cause the hardware to clear this bit. 0: No bus arbitration lost error occurred 1: A bus arbitration lost error occurred
[5]	STOP_EVT	STOP event status flag, used for master mode sending/slave mode sending/ master mode receiving/ slave mode receiving. write 0 to clear. 0: no STOP event 1: STOP event occurred
[4]	BYTE_CMPLT	ACK generation control bit, used for master mode receiving/ slave mode receiving. The receiver's response to it after the sender sends the current byte. For the sender, the value of the bit is reserved to 0. The receiver configures according to the actual situation. 0: byte sending is complete, return NACK response, indicating that the receiver cannot receive more data 1: byte sending is completed, return ACK response, indicating that the receiver can continuously receive data
[3]	ADDR_DATA	Address data flag, for master mode sending/ slave mode sending/ master mode receiving/ slave mode receiving. After START, the first byte is address data, and this bit is a reminder bit. write 0 to clear. 0: The data sent or received is not Address data 1: The data sent or received is Address data
[2]	DATA_DIR	Send or receive control bit. When the master mode sending / slave mode sending is used, set this bit to 1 to trigger sending, and the

		hardware will be automatically cleared; when the master mode receiving/ master mode receiving is used, set this bit to 0 to wait receiving. 0: Receive 1: Trigger sending
[1]	RX_ACK	Receive response flag bit, used for master mode sending/ slave mode sending to notify the sender of the receiver's feedback. After the sender receives the feedback, clear the bit. 0: This I2C interface sends data and receives an ACK response. 1: This I2C interface sends data and receives an NACK response.
[0]	Done	Transmission status flag, used for master mode sending/slave mode sending/ master mode receiving/ slave mode receiving. write 0 to clear. 0: Transmission not completed 1: Transmission completed

Generally, after entering the interrupt, you need to read the I2C_SCR register to obtain the current I2C bus status and the current transmission stage; then, write the I2C_SCR, write different values, and the software informs the hardware how to deal with the next step.

15.4.5 I2C_DATA Data Register

Address:0x4001_040C

Reset value: 0x0

Table 15-5 I2C_ADDR Address Register

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

Location	Bit name	Description
[31:8]		Unused
[7:0]	DATA	Data register, for master mode sending/ slave mode sending/ master mode receiving/ slave mode receiving. The sender writes the sent data; the receiver reads the received data. Note that the address data is also data, and the master mode can only write the address data to be sent into this register.

15.4.6 I2C_MSCR Master Mode Register

Address: 0x4001_0410

Reset value: 0x0

Table 15-6 I2C_MSCR Master Mode Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												BUSY	MST_CHECK	RESTART	START

	RW	RW	RW	RW
	0	0	0	0

Location	Bit name	Description
[31:4]		Unused
[3]	BUSY	I2C bus, idle and busy state. 0: Detected STOP event, busy. 1: Detected START event, idle.
[2]	MST_CHECK	Master mode scrambles for the bus flag. If the bus is scrambled, set to 1; if the STOP event or bus collision occurs, the module releases the bus and sets to 0.
[1]	RESTART	Trigger the START event again and writing 1 is valid. After sending START, the hardware is cleared to 0. Set I2C_CFG [1] to 1 to achieve write "1" operation.
[0]	START	Trigger START event and send address data to the bus. Writing 1 is valid. Set I2C_CFG [1] to 1 to achieve write "1" operation.

15.4.7 I2C_BCR I2C Transmission Control Register

Address:0x4001_0414

Reset value: 0x0

Table 15-7 I2C_BCR Transmission Control Register

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

Location	Bit name	Description
[31:8]		Unused
[7]	NACK_IE	I2C transmission. NACK event interrupt enable signal. 0: disable this interrupt source 1: enable this interrupt source
[6]	ADDR_CMP_IE	I2C transmission, hardware address matching interrupt enable signal. 0: disable this interrupt source 1: enable this interrupt source
[5:0]		Reserved

16 SPI

16.1 Introduction

The SPI interface is mainly used in application scenarios where the external design uses the SPI protocol. SPI working mode software is optional, the default is SPI Motorola mode. The SPI interface supports full-duplex transmission and half-duplex transmission. When the interface is set in Master mode, it can send clock signals for use by external Slave devices.

16.2 Main Features

- Support Master and Slave modes
- Support full-duplex transmission. Three or four signal lines can be used according to the application.
- Support half-duplex transmission. Two signal lines can be used according to the application.
- Programmable clock polarity and phase
- Programmable data sequence: MSB or LSB
- The fastest transmission speed is 1/8 of the system's highest clock frequency
- Chip select signals are optional. In the Master mode, the chip select signal can be controlled by software or generated by hardware; in the Slave mode, the chip select signal can be constant and effective, or it can come from an external device
- No local FIFO, including overflow detection and chip select signal anomaly detection

16.3 Functional Description

16.3.1 Functional Block Diagram

This interface uses a synchronous serial design to achieve SPI transmission between the MCU and external devices. and supports polling and interrupt mode to obtain transmission status information. The main functional modules of this interface are shown in the figure below.

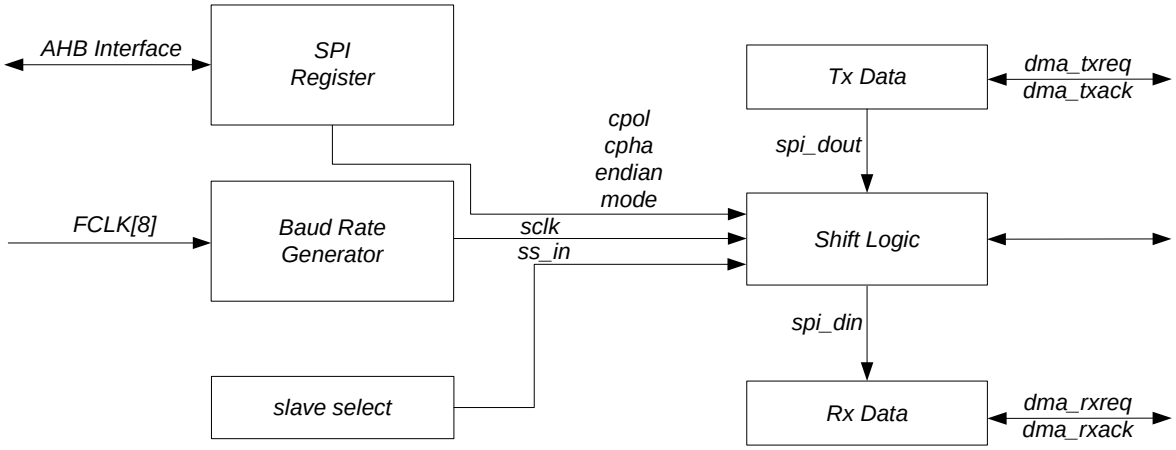


Fig. 16-1 SPI Module Structure Block Diagram

Interface signals include spi_din, spi_dout, sclk_in, sclk_out, ss_in and ss_out.

Spi_din: data signal received by the interface. Compared with the SPI protocol, when the interface is configured in Master mode, it is equivalent to MISO; when the interface is configured in Slave mode, it is equivalent to MOSI.

spi_dout: data signal sent by the interface. Compared with the SPI protocol, when the interface is configured in Master mode, it is equivalent to MOSI; when the interface is configured in Slave mode, it is equivalent to MISO

sclk_in: clock signal received by the interface. The working mode of the interface is Slave at this moment. This signal input is invalid in non-Slave mode.

sclk_out: clock signal sent by the interface. The working mode of the interface is Master, and in non-Master mode, the signal output is always 0.

ss_in: Chip select signal received by the interface. The working mode of the interface is Slave at this moment. This signal input is invalid in non-Slave mode.

ss_out: chip select signal sent by the interface. The working mode of the interface is Master, and this signal output is always 1 in non-Master mode.

16.3.2 Functions

16.3.2.1 Full duplex mode

The SPI interface is configured in full-duplex mode by default. Thus, two data lines are required for data transmission. The change of the data signal occurs on the edge of the clock signal, which is synchronized with the clock signal.

When the interface is in Master mode:

- spi_din is the data input, connected to the MISO of the external Slave device
- spi_dout is the data output, connected to the MOSI of the external Slave device
- spi_ss_out is a chip select signal, choose whether to use this signal or software to control other GPIO implementation according to the application

When the interface is in Slave mode:

- spi_din is the data input, connected to the MOSI of the external master device
- spi_dout is the data output, connected to the MISO of the external master device
- spi_ss_in is the chip select signal, depending on whether the signal is used or the chip select is always valid

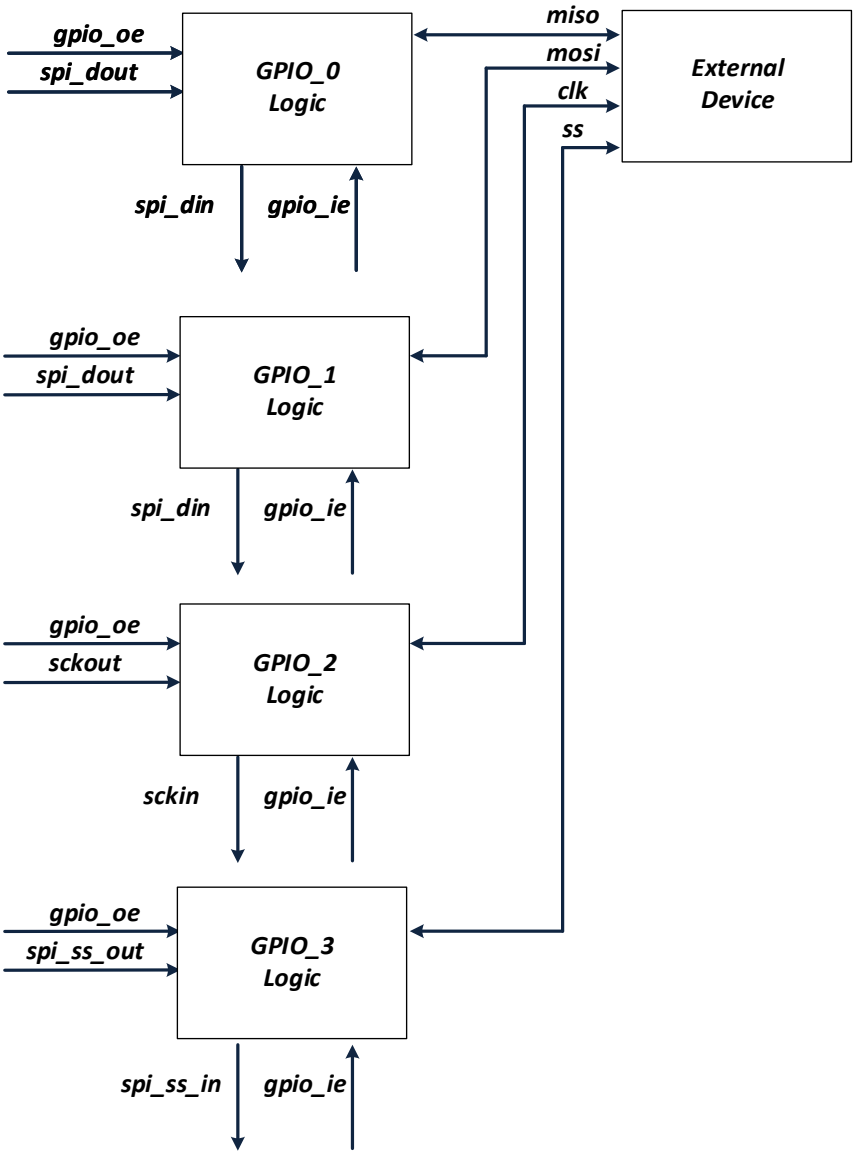


Fig. 16-2 SPI Interface Full Duplex Mode Interconnection Block Diagram

As can be seen from the above figure, if GPIO is configured as an output, the SPI interface can send data; if GPIO is configured as an input, the SPI interface can receive data. SPI data input and data output could be interchanged.

16.3.2.2 Half-duplex Mode

The SPI interface can be set in half-duplex mode. Thus, only one data line is needed for data transmission. The change of the data signal occurs on the edge of the clock signal, which is synchronized with the clock signal. A transmission can only be in one direction, either sending or receiving.

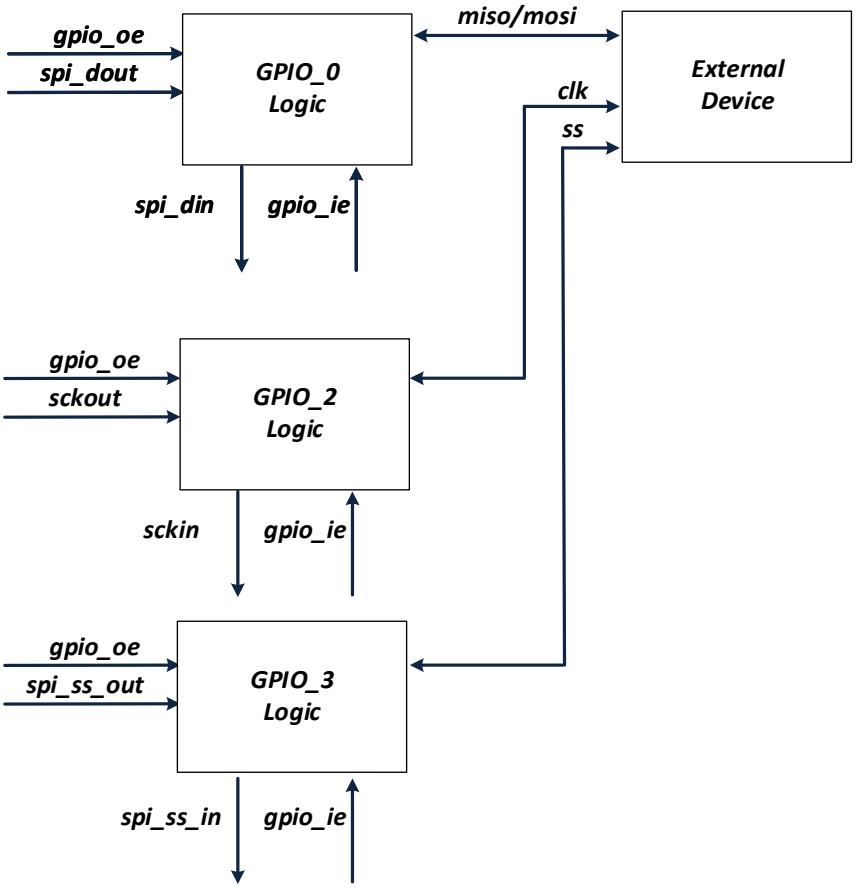


Fig. 16-3 SPI Interface Half-duplex Mode Interconnection Block Diagram

Note that if the interface is a Master in the above figure, CLK is the output signal of the interface; if the interface is the Slave, CLK is the input signal of the interface.

Send only

SPI_CFG. DUPLEX is set to 2, and half-duplex sending mode is valid. This interface can only send data at this moment. GPIO_0's oe is enabled, sending spi_dout data to the outside world; GPIO_0's ie is off, and the spi_din constant input is 0. It supports DMA transmission and supports sending in Master/Slave mode.

Receive only

SPI_CFG. DUPLEX is set to 3, and half-duplex receiving mode is valid. At this moment, the interface can only receive data. GPIO_0's oe is off, spi_dout cannot send data to the outside world;

GPIO_0's ie is on, and spi_din receives data from the outside. In this mode, it supports reception in Master/Slave mode.

Note that in full-duplex mode, two GPIOs are used for data transmission. In half-duplex mode, one GPIO can be selected for data transmission.

16.3.2.3 Chip Select Signal

When this interface is in Slave mode, the chip selection signal is optional, and CFG [5] determines the chip select source. ss is the strobe enable signal sent by the master device. Active low.

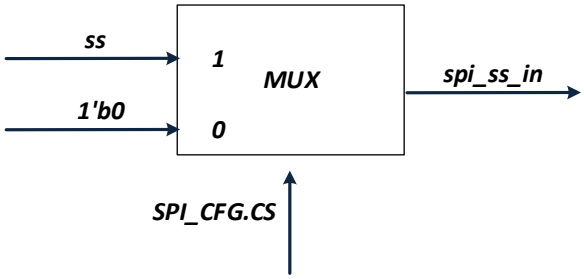


Fig. 16-4 SPI Module Chip Select Signal Selection in Slave Mode

When this interface is in Master mode, the chip selection signal is also selectable. The module hardware generates a standard chip-select signal, which can be shielded by actual application by software operating additional GPIO.

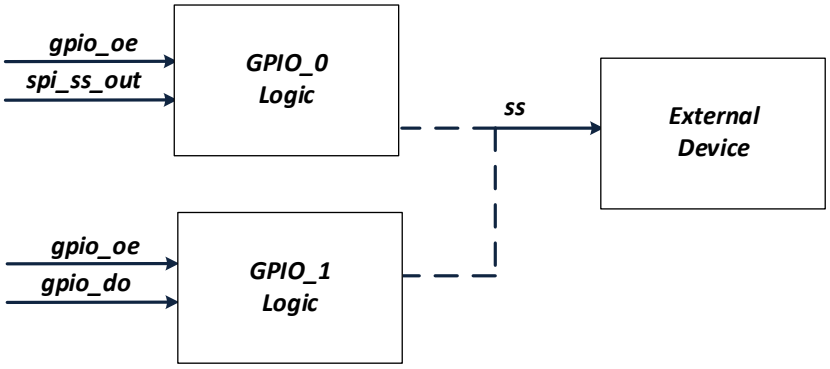


Fig. 16-5 SPI Module Chip Select Signal Selection in Master Mode

Note that the dotted line in Figure 16-5 only indicates uncertainty. If spi_ss_out is used as the source of ss, then GPIO_0 is interconnected with external devices; if software is used to operate GPIO, GPIO_1 can be interconnected with external devices.

16.3.2.4 Communication Format

In the SPI communication process, the sending or receiving operation is based on the SPI clock. The communication format is controlled by SPI_CFG. SAMPLE and SPI_ CLK_POL. SPI_CFG. SAMPLE is Phase control bit and SPI_ CLK_POL is Polarity control bit.

Polarity controls the level status of the SPI clock signal by default. When Polarity is 0, the default clock level is low; when Polarity is 1, the default level is high.

Phase controls the sending/receiving moment of SPI data. When Phase is 0, the clock transitions from the default level to the first transition edge is the time to sample data, and when Phase is 1, the clock transitions from the default level to the first transition edge is the time to transmit data.

16.3.2.5 Data Format and Length

SPI data transmission format is divided into two types: MSB and LSB. The data transmission format is controlled by SPI_CFG.ENDIAN. Note that the hardware automatically converts the transmission format during data transmission without software conversion.

The SPI data length is configurable, ranging from 8-Bit to 16-Bit. SPI_SIZE.BITSIZE Control Length.

16.3.2.6 Transmission

Only one SPI_SIZE.BITSIZE length of data can be sent/received at a time. After each completion, it is necessary to determine whether the transmission is completed by interrupt or polling. For the master mode or slave mode, the transmission can be triggered only by writing to the SPI_TX_DATA register. The master mode is active sending, and the slave mode is loading data to the sending queue and wait for the clock signal from the master mode to start transmission. The recommended software configuration process is as follows:

- Initialize the GPIO module and complete configuring the GPIO multiplexed by SPI.
- Initialize the SPI interface, and complete configuring the SPI_IE/SPI_CFG/SPI_BAUD/SPI_SIZE registers, etc.
- The MCU writes to the SPI_TX_DATA register and triggers the SPI interface to enter the sending process. In the slave mode, data is loaded into the internal state machine and waits for the master to initiate a read operation; in the master mode, the sending is triggered.

It supports continuous sending, controlled by SPI_BAUD.TRANS_MODE. It is mainly targeted to the master mode.

In the discontinuous mode, a complete output transmission is: The chip selection signal is valid, SPI_TX_DATA writes the sending value to trigger a data transmission with the length of SPI_SIZE.BITSIZE. After the completion, the chip selection signal becomes invalid.

In continuous mode, a complete output transmission is: the chip selection signal is valid, SPI_TX_DATA writes the sending value, a data transmission of SPI_SIZE.BITSIZE length is completed, SPI_TX_DATA writes a new value, and triggers the next data transmission of SPI_SIZE.BITSIZE length. The selection signal remains valid until the application sends this batch of data, and the chip selection signal becomes invalid.

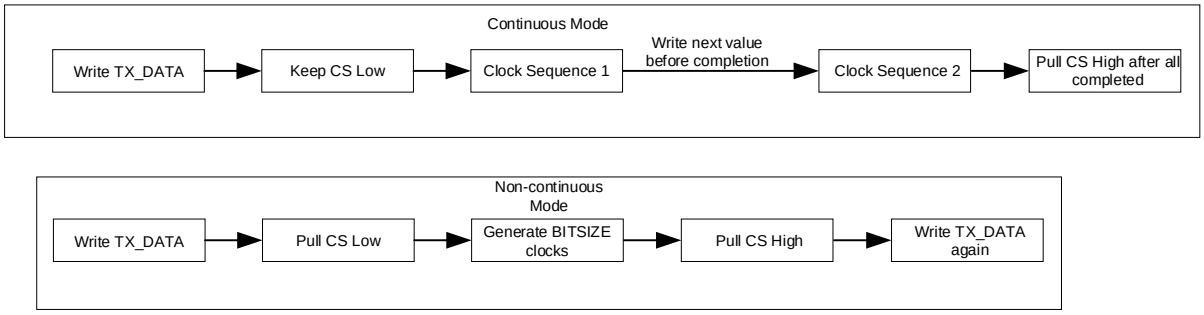


Fig. 16-6 Comparison diagram of Continuous Mode vs. Discontinuous Mode

16.3.2.7 Interrupt Handling

The SPI interface includes three types of interrupt events: data transmission completion event, exception event and overflow event.

- Data transmission completion event. The current data transmission is completed. Active high, write 0 to clear SPI_IE. CMPLT_IF.
- Exception event. The SPI interface is Slave mode. If the chip selection signal is disturbed during transmission and is pulled high, a chip selection exception event will occur. Active high, write 1 to clear SPI_IE. AB_IF.
- Overflow event. If the SPI_RX_DATA register data is not read in time, an overflow event will occur. Active high, write 1 to clear SPI_IE. OV_IF.

For the above events, SPI interrupt is not triggered by default. You can configure SPI_IE[7:4] to enable the event to generate an interrupt.

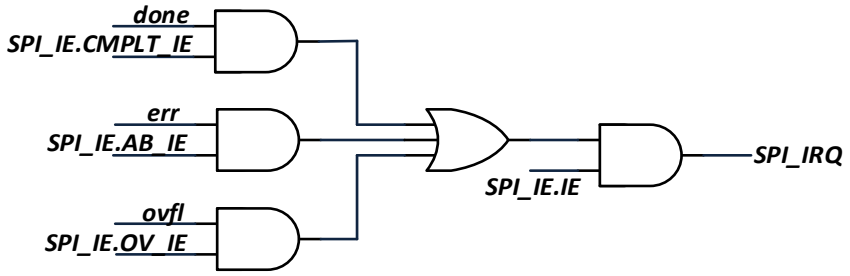


Fig. 16-7 SPI Module Interrupt Select Signal Generation Diagram

16.3.2.8 Baud Rate Setting

The SPI interface clock is obtained by dividing the system clock by the frequency division factor from SPI_BAUD. BAUD. The SPI transmission baud rate configuration calculation formula is:

SPI transmission baud rate = system clock / (2*(BAUD + 1))

The SPI protocol is a half-shot protocol. The rising edge sends data and the falling edge collects data; or the falling edge sends data and the rising edge uses data.

The SPI interface adopts a synchronous design, and the signals of external devices need to be

synchronously sampled. The synchronous clock is the system clock. Synchronization of data and clock signals (Slave mode) requires two beats of the system clock. Considering the clock phase shift, the redundancy of the system clock is required at this time. It is deduced from this that the fastest BAUD rate is 1/8 of the system clock, the high-level period is four-beat system clock, and the low-level period is four-beat system clock. Therefore, the configuration value of SPI_BAUD. BAUD cannot be less than 3.

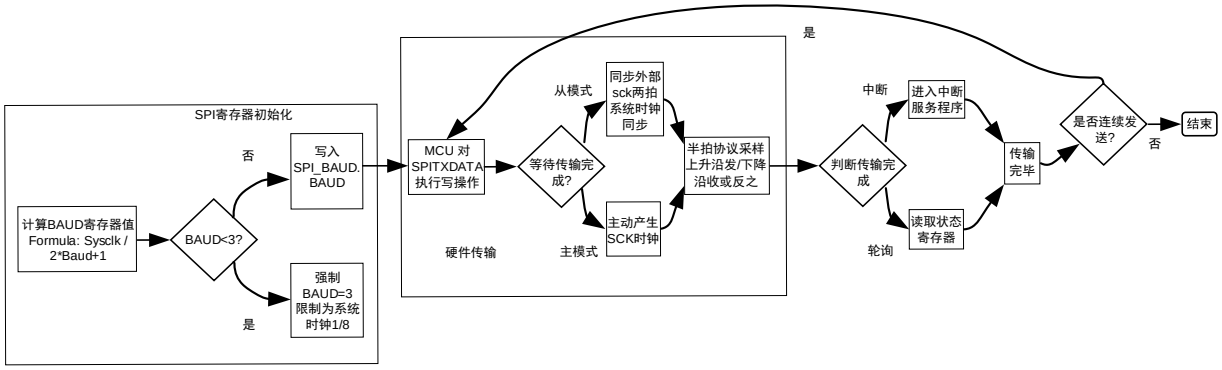


Fig. 16-8 Working Principle of SPI

16.4 Register

16.4.1 Address Allocation

The base address of the HALL module register is 0x40010000.

Table 16-1 List of SPI Module Control Register

Name	Offset	Description
SPI_CFG	0x00	SPI Configuration Register
SPI_IE	0x04	SPI Interrupt Register
SPI_BAUD	0x08	SPI Baud Rate Setting Register
SPI_TXDATA	0x0C	SPI Sending Data Register
SPI_RXDATA	0x10	SPI Receiving Data Register
SPI_SIZE	0x14	SPI Transfer Data Length Register

16.4.2 SPI_CFG SPI System Control Register

Address: 0x4001_0000

Reset value: 0x0024

Table 16-2 SPI_CFG System Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DUPLEX		CS	MS	CPHA	CPOL	ENDIAN	EN
								RW		RW	RW	RW	RW	RW	RW
								0		1	0	0	1	0	0

Location	Bit name	Description
[31:8]		Unused
[7:6]	DUPLEX	Half-duplex mode setting 0X: Turn off half-duplex mode 10: Turn on half-duplex mode, sending only 11: Turn on half-duplex mode, receiving only
[5]	CS	Source of chip selection signal under SPI slave device. The default value is 1. 0: The chip selection signal in Slave mode is always a valid value --0 1: The chip selection signal in Slave mode comes from the Master device
[4]	MS	SPI master-slave mode selection. The default value is 0. 0: Slave mode 1: Master mode
[3]	CPHA	SPI phase selection. The default value is 0. 0: Phase is 0 1: Phase is 1
[2]	CPOL	SPI polarity selection. The default value is 0. 0: Polarity is 0 1: Polarity is 1
[1]	ENDIAN	SPI module transmission sequence. The default value is 0. 0: MSB, high bit is transmitted first 1: LSB, low bit is transmitted first
[0]	EN	SPI module enable signal. The default value is 0. 0: turn off the SPI module 1: turn on the SPI module

16.4.3 SPI_IE SPI Interrupt Register

Address: 0x4001_0004

Reset value: 0x0

Table 16-3 SPI_IE Interrupt Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								IE	CMPLT_IE	AB_IE	OV_IE		CMPLT_IF	AB_IF	OV_IF
								RW	RW	RW	RW		RW	RW	RW
								0	0	0	0		0	0	0

Location	Bit name	Description
[31:8]		Unused
[7]	IE	SPI interrupt enable switch. The default value is 0. 0: disable SPI interrupt 1: enable SPI interrupt
[6]	CMPLT_IE	SPI transmission, complete event interrupt enable signal. 0: disable this interrupt source 1: enable this interrupt source
[5]	AB_IE	SPI transmission, abnormal event interrupt enable signal. 0: disable this interrupt source 1: enable this interrupt source
[4]	OV_IE	SPI transmission, interrupt enable signal for overflow event. The

		default value is 0. 0: disable this interrupt source 1: enable this interrupt source
[3]		
[2]	CMPLT_IF	SPI transmission, complete event. Active high, write 1 to clear.
[1]	AB_IF	SPI transmission, abnormal events. In Slave mode, the transmission is not completed, and the chip selection signal invalid event occurs. Active high, write 1 to clear.
[0]	OV_IF	SPI transmission, overflow event. The old data received last time has not been taken away, the new data received this time has arrived. Active high, write 1 to clear.

16.4.4 SPI_BAUD SP Baud Rate Setting Register

Address: 0x4001_0008

Reset value: 0x0

Table 16-4 SPI_BAUD Control Register

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

Location	Bit name	Description
[31:8]		Unused
[7]	TRANS_MODE	SPI continuous sending, only used for master mode. The default value is 0. 0: non-continuous sending. 1: Continuous sending.
[6:0]	BAUD	SPI transmission baud rate configuration. SPI actual transmission speed calculation formula is: SPI transmission speed = system clock/(2*(BAUD+1)) Remember, the set value of BAUD cannot be less than 3.

16.4.5 SPI_TXDATA SPI Sending Data Register

Address: 0x4001_000C

Reset value: 0x0

Table 16-5 SPI_TXDATA Sending Data Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	TX_DATA	SPI Sending Data Register

16.4.6 SPI_RXDATA SPI Receiving Data Register

Address: 0x4001_0010

Reset value: 0x0

Table 16-6 SPI_RXDATA Receiving Data Register

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

Location	Bit name	Description
[31:16]		Unused
[15:0]	RX_DATA	SPI Receiving Data Register

16.4.7 SPI_SIZE SPI Transfer Data Length Register

Address: 0x4001_0014

Reset value: 0x0

Table 16-7 SPI_SIZE Transfer Data Length Register

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

Location	Bit name	Description
[31:8]		Unused
[4:0]	BITSIZE	Byte length register. 0x00: Invalid value 0x07: Invalid value 0x08:8-Bit 0x09:9-Bit … 0x0E:14-Bit 0x0F:15-Bit 0x10:16-Bit

17 CMP

17.1 Introduction

The comparator signal processing module (Hereinafter referred to as CMP module. For better distinguishing, the analog comparator in the following figure is represented by Comparator, and the digital CMP module is represented by CMP) is used to process the output signals generated by the two analog rail-to-rail comparators and consists of a series of digital circuits such as enable, polarity control, and filtering. The signal processing clock is obtained by dividing the main clock. This module is also used to generate a comparator interrupt to the CPU.

CMP can be used for the following functions:

1. Compare the zero-crossing point of the back EMF
2. Hardware overcurrent detection
3. The source of the fail signal of MCPWM

CMP main features:

1. Each comparator has configurable plus and minus inputs used for flexible voltage selection:
    - Multi-channel GPIO pins
    - OPA output signal
    - OPA positive terminal output signal
    - 1.2V BANDGAP reference source
    - DAC output signal
2. Programmable comparison speed, programmable hysteresis voltage
3. The output signal can be filtered, and the filtering depth can be selected
4. Per-channel can generate CMP interrupt

The base address of the CMP module register is 0x40010C00.

The unfiltered original output value of the analog comparator can be obtained by reading the CMP_DATA value, The unfiltered original output value of the analog comparator can also be sent through P0.14 and P2.3 by configuring the second function of GPIO. For specific GPIO second function configuration and introduction location, please refer to the device datasheet.

For more information about the analog comparator, including the selection of its input signal and the configuration of hysteresis, please refer to ANALOG chapter

Recommended configuration process:

1. Open the CMP analog switch

   Under the default state, the comparator module is turned off. By configuring the SYS_AFE_REG5.CMPxPDN(x=0/1, which represents two comparators of CMP0 / CMP1), the comparator can be opened. Turn on the Bandgap before using the comparator module.

2. Open the digital clock switch and signal input switch, select the clock filter coefficient

   Turn on the digital clock switch of the CMP by configuring CMP_TCLK.CLK_EN, 1 means opening, 0 means closed. Select the filter clock frequency division by configuring CMP_TCLK.FIL_CLK_DIV16[7:4]. The range of numerical settings is 0 ~ 15, based on MCLK to divide by 1 to 16. By configuring CMP_CFG.CMPx_IN_EN to enable the signal input, 1 means enable, 0 means disable.

3. Configure the comparison speed and hysteresis of CMP

   The comparison speed of the comparator is 0.15uS/<30ns by configuring SYS_AFE_REG1.CMP_FT. The comparator hysteresis can be set to 20mV/0mV by configuring SYS_AFE_REG3.CMP_HYS.

4. Select the positive and negative signal source of CMP

   The positive signal of CMP has 8 signal sources to choose from. It can be set by configuring SYS_AFE_REG3.CMPx_SELP. There are 4 signal sources of the negative terminal to choose from, which can be set by configuring SYS_AFE_REG3.CMPx_SELN.

5. Configure the interrupt of CMP

   By configuring CMP_IE.CMPx_IE to enable the CMP interrupt, 1 means enable, 0 means disable.

17.2 Register

17.2.1 Address Allocation

The base address of the CMP module register is 0x4001_0000.

Table 17-1 Comparator Register List

Name	Offset address	Description
CMP_IE	0x00	Comparator interrupt enable register
CMP_IF	0x04	Comparator interrupt flag register
CMP_TCLK	0x08	Comparator divider clock control register
CMP_CFG	0x0C	Comparator control register
CMP_BLCWIN	0x10	Comparator window control register
CMP_DATA	0x14	Comparator output value register

17.2.2 CMP_IE Interrupt Enable Register

Address: 0x4001_0C00

Reset value: 0x0

Table 17-2 CMP_IE Comparator Interrupt Enable Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														CMP1_IE	CMP0_IE
														RW	RW
														0	0

Location	Bit name	Description
[31:2]		Unused
[1]	CMP1_IE	Comparator 1 interrupt enable, active high
[0]	CMP0_IE	Comparator 0 interrupt enable, active high

17.2.3 CMP_IF Interrupt Flag Register

Address: 0x4001_0C04

Reset value: 0x0

Table 17-3 CMP_IF Comparator Interrupt Flag Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														CMP1_IF	CMP0_IF
														RW1C	RW1C
														0	0

Location	Bit name	Description
[31:2]		Unused
[1]	CMP1_IF	Comparator 1 interrupt flag, active high, write 1 to clear
[0]	CMP0_IF	Comparator 0 interrupt flag, active high, write 1 to clear

17.2.4 CMP_TCLK Divider Clock Control Register

Address: 0x4001_0C08

Reset value: 0x0

Table 17-4 CMP_TCLK Comparator Divider Clock Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FIL_CLK_DIV16				CLK_EN		FIL_CLK_DIV1248	
								RW				RW		RW	
								0				0		0	

Location	Bit name	Description
[31:8]		Unused
[7:4]	FIL_CLK_DIV16	Comparator filter clock frequency division, based on MCLK to divide by 1 to 16, affecting the time to enter the comparator interrupt
[3]	CLK_EN	Clock enable, active high
[2]		Unused
[1:0]	FIL_CLK_DIV1248	Comparator filter clock divided by 2'b00: 1, 2'b01: 2, 2'b10: 4, 2'b11: 8

CMP Filter Operating Frequency

Freq(CMP_Filter) = Freq(MCLK)/$2^{CMP\_TCLK.FIL\_CLK\_DIV1248}$/(CMP_TCLK.FIL_CLK_DIV16+1), of which MCLK is the main clock, usually is a 96MHz full-speed clock. Note that the CMP_TCLK.CLK_EN bit should be enabled to generate the CMP filter clock.

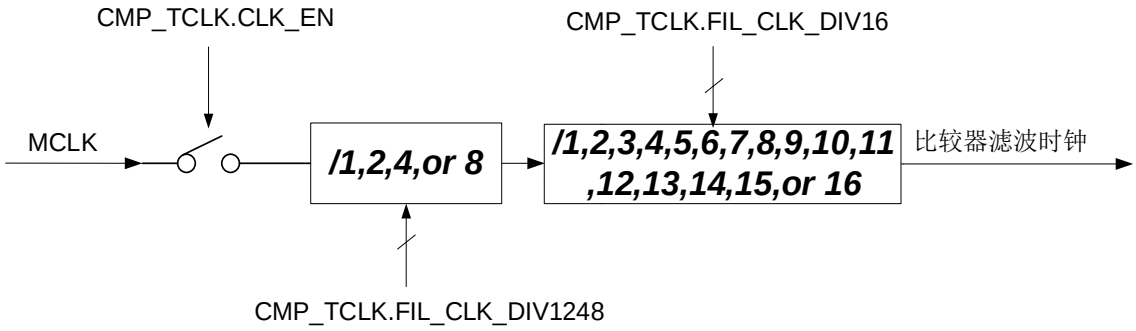


Fig. 17-1 Comparator Filter Clock Generation

The CMP module uses this filter clock to filter the output signal of the analog comparator for sixteen clock cycles, i.e., only the signal stabilization time exceeds sixteen filter clock cycles to pass the filter. The filtered signal output by the CMP module will change. If the input signal is stable for less than sixteen filter clock cycles, the filtered signal output by the CMP module will remain unchanged. **Filter width = filter clock period*16.**

17.2.5 CMP_CFG Control Register

Address: 0x4001_0C0C

Reset value: 0x0

Table 17-5 CMP_CFG Comparator Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CMP1_OW_PWM_POL	CMP1_IRQ_TRIG	CMP1_IN_EN	CMP1_POL	CMP0_OW_PWM_POL	CMP0_IRQ_TRIG	CMP0_IN_EN	CMP0_POL
								RW	RW	RW	RW	RW	RW	RW	RW

	0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:8]		Unused
[7]	CMP1_OW_PWM_POL	Comparator 1 window PWM signal polarity selection, used when CMP_BLCWIN is enabled
[6]	CMP1_IRQ_TRIG	Comparator 1 interrupt trigger type, 0: level trigger, 1: edge trigger
[5]	CMP1_IN_EN	Comparator 1 signal input enable
[4]	CMP1_POL	Comparator 1 polarity selection, 0: active high; 1: active low
[3]	CMP0_OW_PWM_POL	Comparator 0 window PWM signal polarity selection, used when CMP_BLCWIN is enabled
[2]	CMP0_IRQ_TRIG	Comparator 0 interrupt trigger type, 0: level trigger, 1: edge trigger
[1]	CMP0_IN_EN	Comparator 0 signal input enable
[0]	CMP0_POL	Comparator 0 polarity selection, 0: active high; 1: active low

The polarity and enable control of the comparator are as shown in.

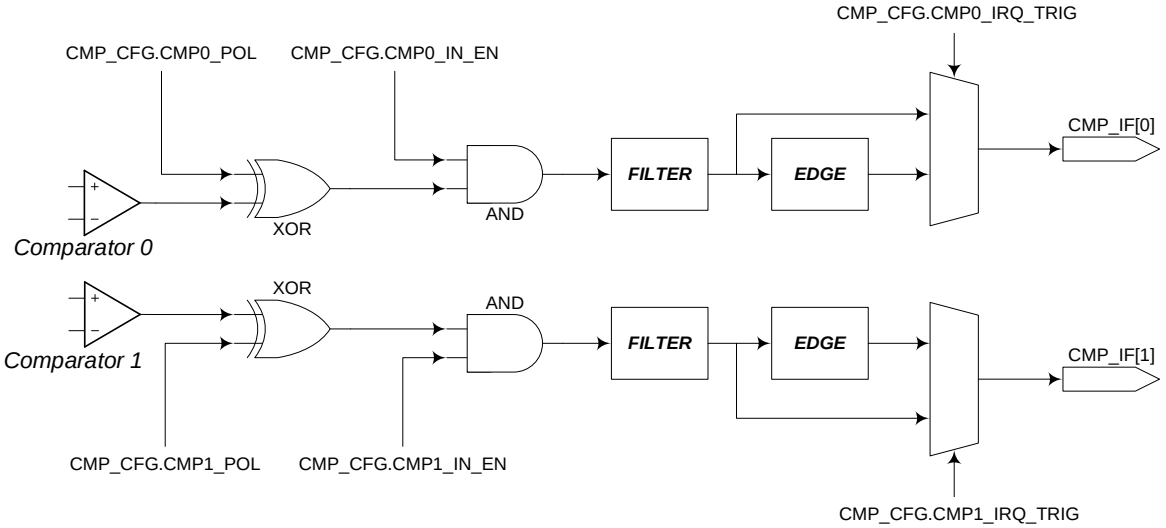


Fig. 17-2 Comparator Control and Interrupt Generation Logic

The comparator module and the MCPWM module can work together, and the P-tube control signal of the MCPWM module can be used as the control signal for comparator windowing. However, the interrupt signal of the comparator itself is generated regardless of the window control and is only affected by the CMP_CFG register.

The fail signal of MCPWM can come from GPIO or from the comparator module, and is controlled by the MCPWM_FAIL register. If the fail signal of MCPWM comes from the comparator, it is controlled by the window inside the comparator module. After the fail signal enters MCPWM, it will also be processed with polarity enable and filtering. It is similar to the comparator module but completely independent, and is controlled by the register inside MCPWM. The error interrupt signal related to fail in MCPWM is affected by the polarity-enable filter control register in MCPWM. For details, please refer to the MCPWM chapter.

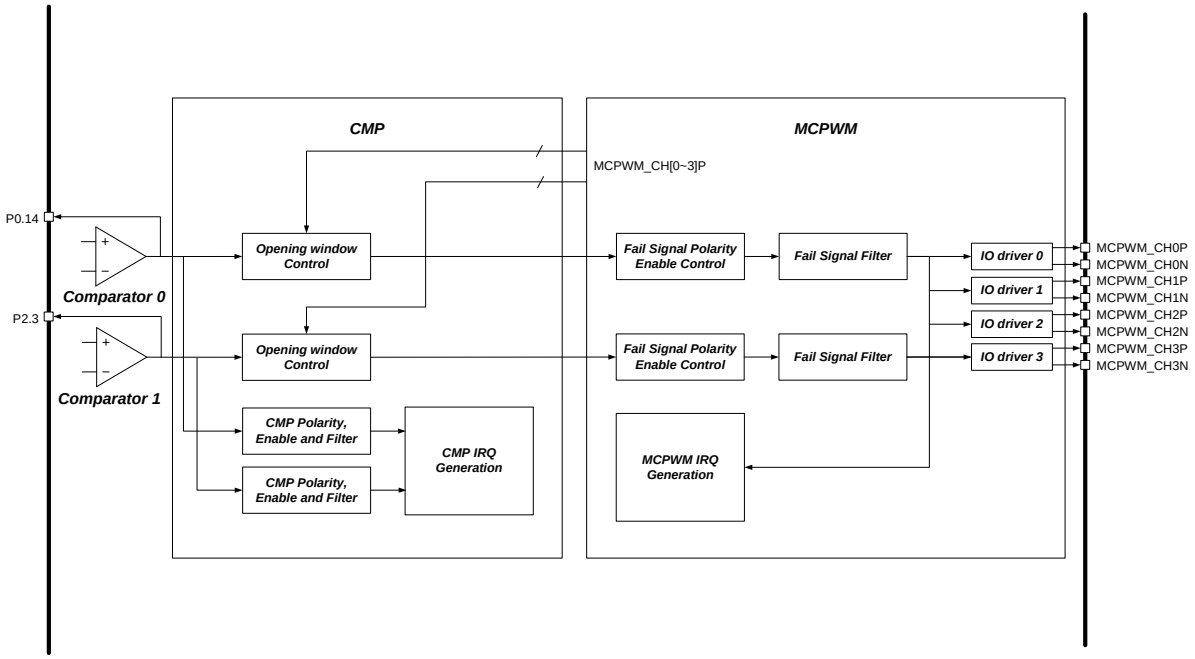


Fig. 17-3 CMP and MCPWM Linkage

For the windowing function of the comparator, if CMP_CFG.CMP0_PWM_POL = 1, then when the corresponding MCPWM CHNx_P signal is 1, the comparator 0 can generate a comparison signal output, and the comparison signal is 0 at other times; Conversely, if CMP_CFG.CMP0_PWM_POL = 0, then when the corresponding MCPWM CHNx_P signal is 0, the comparator 0 can generate a comparison signal output, and the comparison signal is 0 at other times. The window control signal polarity of the comparator 1 is controlled by the CMP_CFG.CMP1_PWM_POL bit, and the logic is the same.

Note: CMP_CFG.CMP0_PWM_POL and CMP_CFG.CMP1_PWM_POL will also affect the comparator signal sent to the MCPWM module as a fail signal.

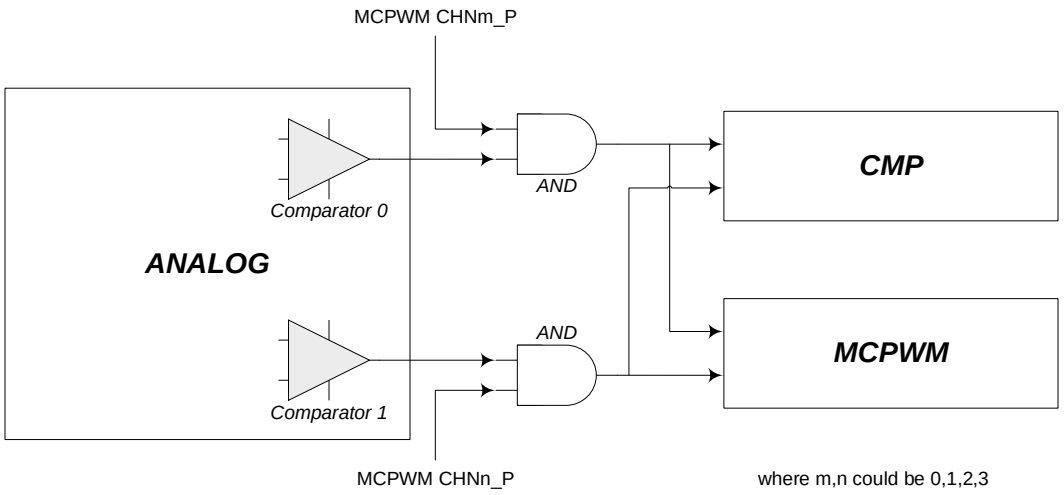


Fig. 17-4 Comparator Window Function Diagram

17.2.6 CMP_BLCWIN Window Control Register

Address: 0x4001_0C10

Reset value: 0x0

Table 17-6 CMP_ BLCWIN Comparator Window Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CMP1_CHN3P_WIN_EN	CMP1_CHN2P_WIN_EN	CMP1_CHN1P_WIN_EN	CMP1_CHN0P_WIN_EN	CMP0_CHN3P_WIN_EN	CMP0_CHN2P_WIN_EN	CMP0_CHN1P_WIN_EN	CMP0_CHN0P_WIN_EN
								RW	RW	RW	RW	RW	RW	RW	RW
								0	0	0	0	0	0	0	0

Location	Bit name	Description
[31:8]		Reserved
[7]	CMP1_CHN3P_WIN_EN	Use the P-tube switch control signal output from the CHN3_P channel of the MCPWM module as the comparator 1 window enable
[6]	CMP1_CHN2P_WIN_EN	Use the P tube switch control signal output from the CHN2_P channel of the CHPWM module as the comparator 1 window enable
[5]	CMP1_CHN1P_WIN_EN	Use the P-tube switch control signal output from the CHN1_P channel of the MCPWM module as the comparator 1 window enable
[4]	CMP1_CHN0P_WIN_EN	Use the P-tube switch control signal output from the CHN0_P channel of the MCPWM module as the comparator 1 window enable
[3]	CMP0_CHN3P_WIN_EN	Use the P-tube switch control signal output from the CHN3_P channel of the MCPWM module as the comparator 0 window enable
[2]	CMP0_CHN2P_WIN_EN	Use the P tube switch control signal output from the CHN2_P channel of the CHPWM module as the comparator 0 window enable
[1]	CMP0_CHN1P_WIN_EN	Use the P-tube switch control signal output from the CHN1_P channel of the MCPWM module as the comparator 0 window enable
[0]	CMP0_CHN0P_WIN_EN	Use the P-tube switch control signal output by the CHN0_P channel of the MCPWM module as the comparator 0 window enable

Usually 1-bit is 1 in CMP_ BLCWIN [3: 0] or CMP_ BLCWIN [7: 4], indicating that the corresponding CHNx_P is used to control the signal generation of the comparator 0/1. If CMP_ BLCWIN [3: 0] or CMP_ BLCWIN [7: 4] is 4'b0000, it means that the comparator 0/1 comparison signal is generated regardless of the PWM signal.

17.2.7 CMP_DATA Output Data Register

Address: 0x4001_0C14

Reset value: 0x0

Table 17-7 CMP_ DATA Comparator Output Data Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												CMP1_FLT_DATA	CMP0_FLT_DATA	CMP1_RAW_DATA	CMP0_RAW_DATA
												R	R	R	R
												0	0	0	0

Location	Bit name	Description
[31:4]		Reserved
[3]	CMP1_FLT_DATA	Signal of comparator 1 after filtering
[2]	CMP0_FLT_DATA	Signal of comparator 0 after filtering
[1]	CMP1_RAW_DATA	The original output signal of comparator 1, directly from analog comparator 1
[0]	CMP0_RAW_DATA	The original output signal of comparator 0, directly from analog comparator 0

18 Version History

Table 18-1 Document Version History

Date	Version No.	Description
Mar.05,2026	1.35	In the UART chapter, change UART_IO to UART_INV
Mar.03,2026	1.34	Revise the subheadings for each register in the SYS, SPI, ADC, UTIMER, MCPWM, UART, and IWDG modules
Feb.26,2026	1.34	Revise the reset values of all registers in the SYS, ADC, UART, and SPI modules.
Feb.10,2026	1.33	Supplemental Description for GPIO Module: x = 0, 1, 2
Feb.04,2026	1.32	Revise the I2C Slave Mode Workflow Diagram
Feb.02,2026	1.31	PWM module supplementary flowchart descriptions
Jan.29,2026	1.30	Revise the block diagrams of each module
Jan.28,2026	1.29	I2C module supplementary flowchart descriptions
Jan.26,2026	1.28	ADC, DMA, SPI, and UART modules supplementary flowchart descriptions
Jan.13, 2026	1.27	Timer counting period changed to (TH+1) / clk_freq
Oct.23, 2025	1.26	VTOR register description: When read back, it shifts right by 7 bits.
Oct.11, 2025	1.25	Add NVR storage address
Sep.26, 2025	1.24	5.3.13 Cell Merging
Jan.02, 2025	1.23	Add the FLASH NVR address for OPA Offset
Dec.02, 2024	1.22	Add description of the time required for ADC IF to trigger
Jun.17, 2024	1.21	Add NVR calibration parameter address information
Sep.25, 2023	1.20	Added notes on GPIO PDI
Aug.10, 2023	1.19	Modify errors in the GPIO sections
Apr.07, 2023	1.18	Added description of ADC conversion elapsed time
Mar. 23, 2023	1.17	Modify the LSI accuracy range
Feb. 18, 2023	1.16	Modified the description of MCPWM_SDCFG
Feb.10,2023	1.15	Add the description of PLLPDN, BGPPD, BGPPD
Dec.30,2022	1.14	Revised DSP calculation cycles
Nov.23,2022	1.13	Added soft reset ADC module description
Nov.17,2022	1.12	Revised OPA0 and OPA2 time-sharing multiplexing, OPA1 and OPA3 time-sharing multiplexing
Nov.10,2022	1.11	Add connection resistance between IO and internal analog circuit
Sep.15,2022	1.10	Add configuration flow for ADC/OPA/CMP
Aug.2,2022	1.09	Simplify SYS_AFE_REG description
Oct.18,2021	1.08	Revised Flash Erase&Program times
Aug.4, 2021	1.07	Revised OPA feedback resistors
May.20, 2021	1.06	Add SYS_OPA_SEL description
Nov.15, 2020	1.05	Revised Timer filter clock description
Sep.17, 2020	1.04	The LRC clock is changed from 32kHz to 64kHz, affecting the watchdog reset time
Jun.11, 2020	1.03	Revised ADC_DC description
Jun. 06, 2020	1.02	Revised MCPWM_DTHxx register description
May 01, 2020	1.01	Added reset source description, revised Timer ETON description
Jan. 18, 2020	1.0	Revised format, revised problematic sections description
May 08, 2019	0.1	Initial version, including the internal version on related description of test

Disclaimer

LKS and LKO are registered trademarks of Linko.

Linko tries its best to ensure the accuracy and reliability of this document, but reserves the right to change, correct, enhance, modify the product and/or document at any time without prior notice. Users can obtain the latest information before placing an order.

Customers should select the appropriate Linko product for their application needs and design, validate and test your application in detail to ensure that it meets the appropriate standards and any safety, security or other requirements. The customer is solely responsible for this.

Linko hereby acknowledges that no intellectual property licenses, express or implied, are granted to Linko or to third parties.

Resale of Linko products on terms other than those set forth herein shall void any warranty warranties made by Linko for such products.

Prohibited for military use or life care and maintenance systems.

For earlier versions, please refer to this document.