BL702/704/706

Reference Manual

version：1.1

www.bouffalolab.com

Contents

List of Figures

List of Tables

1 System and memory overview

1.1 Introduction

The on-chip processor uses RISC-V 32-bit with floating point. With high-speed processing memory system (see the L1C chapter for details), to achieve high-quality computing efficiency. External to the processor is a multilayer 32-bit AHB architecture with low power consumption, low latency, and high flexibility. The memory section contains high-speed tightly coupled memory as well as cache and system shared memory. Off-chip memory supports Flash expansion.

1.2 Main features

- RISC-V 32-bit with floating point
- Multi-layer 32-bit AHB bus architecture
- 132KB RAM
- 192KB ROM
- Off-chip memory Flash

1.3 Function description

The BL702/704/706 bus connection and address access are summarized as follows:

The bus master includes CPU, Ethernet, DMA, encryption engine, debugging interface. The bus slave includes memory, peripherals, and Zigbee/BLE. Except for Ethernet and encryption engine which can only access memory, all other bus masters can access all bus slaves.

Table 1.1: Bus connection

Slave/Master	CPU	Ethernet	DMA	Encryption engine	Debug interface
memory	V	V	V	V	V
Peripheral	V		V	-	V
Zigbee/BLE	V	-	V	-	V

The address access mainly distinguishes ”memory” or ”peripheral” by [27:24], and the [31:28] bits can be ignored. The memory space is consecutive addresses 0x2010000~0x202FFFF (128KB SRAM)，the read-only memory address is 0x1000000，and the deep sleep memory address is 0x0010000。The off-chip space address is 0x3000000 (maximum support 8MB Flash). The peripheral space is 0x0000000 ~ 0x000F000.

Table 1.2: Memory Map

Module	Base Address	Size	Description
RETRAM	0x40010000	4KB	Deep sleep memory (Retention RAM)
HBN	0x4000F000	4KB	Deep sleep control (Hibernate)
PDS	0x4000E000	4KB	Sleep control (Power Down Sleep)
USB	0x4000D800	1KB	USB control
EMAC	0x4000D000	2KB	Ethernet MAC control
DMA	0x4000C000	4KB	DMA control
QSPI	0x4000B000	4KB	Flash/pSRAM QSPI control
I2S	0x4000AA00	256B	I2S control
KYS	0x4000A900	256B	Key-Scan control
QDEC2	0x4000A880	64B	Quadrature decoder control
QDEC1	0x4000A840	64B	Quadrature decoder control
QDEC0	0x4000A800	64B	Quadrature decoder control
IRR	0x4000A600	256B	IR Remote control
TIMER	0x4000A500	256B	Timer control
PWM	0x4000A400	256B	Pulse Width Modulation *5 control
I2C	0x4000A300	256B	I2C control
SPI	0x4000A200	256B	SPI master/slave control
UART1	0x4000A100	256B	UART control (support LIN-bus)
UART0	0x4000A000	256B	UART control (support LIN-bus)
L1C	0x40009000	4KB	Cache control

Table 1.2: Memory Map

Module	Base Address	Size	Description
eFuse	0x40007000	4KB	eFuse memory control
SEC	0x40004000	4KB	Security engine
GPIP	0x40002000	4KB	General purpose DAC/ADC/ACOMP interface control
MIX	0x40001000	4KB	Mixed signal register
GLB	0x40000000	4KB	Global control register
pSRAM	0x24000000	8MB	pSRAM memory
XIP	0x23000000	8MB	XIP Flash memory
OCRAM	0x22020000	64KB	On-chip memory
DTCM	0x22014000	48KB	Data cache memory
ITCM	0x22010000	16KB	Instruction cache memory
ROM	0x21000000	192KB	Read-only memory

There are 64 interrupt sources. The level or edge trigger is configured by the CPU and can be masked. Details as follows:

Table 1.3: Interrupt source

Num	Signal source
54~63	wireless
53	brown-out
51~52	hbn_irq
50	pds_int
47~49	wireless
44	gpio_irq
40~42	qdec_int
39	kys_int
35~38	timer_irq
34	pwm_int
32	i2c_int
30	uart1_irq

Table 1.3: Interrupt source

Num	Signal source
29	uart0_irq
27	spi_int
26	efuse_int
25	adc_int
23	flash_int
22	emac_int
21	usb_int
19~20	ir_remote_int
18	i2s_int
15	dma_int
9~14	sec_eng_int
8	err_int
5~6	rf_int
0~4	err_int

2 Reset and clock

2.1 Introduction

The reset sources included in the chip: hardware reset, watchdog reset, software reset. The chip contains multiple clock sources: XTAL, PLL, RC. It is allocated to each module through configuration such as frequency division.

2.2 Reset source

The reset sources are as follows:

- Hardware reset: reset via pin
  - Pin power reset (PU_CHIP = 0-> 1): similar to power reset
  - Power-on reset: the chip recovers from power failure, and HBN logic resets the chip system
- Watchdog reset
  - When the watchdog alarm triggers the reset signal, the reset management unit will reset the chip system after necessary preparations, and the internal logic of the watchdog will record the status of the watchdog reset
- Software reset: partial reset by software setting register
  - Software initial reset (reg_ctrl_pwron_rst): trigger the rising edge of this register by software to reset the chip system
  - Software CPU reset (reg_ctrl_cpu_reset): Trigger the rising edge of this register by software to reset the CPU part of the system
  - Software system reset (reg_ctrl_sys_reset): Trigger the rising edge of this register by software, retain necessary logic processing such as power management unit, and reset the chip part of the system
  - Software module reset: Set software reset according to the needs of specific modules

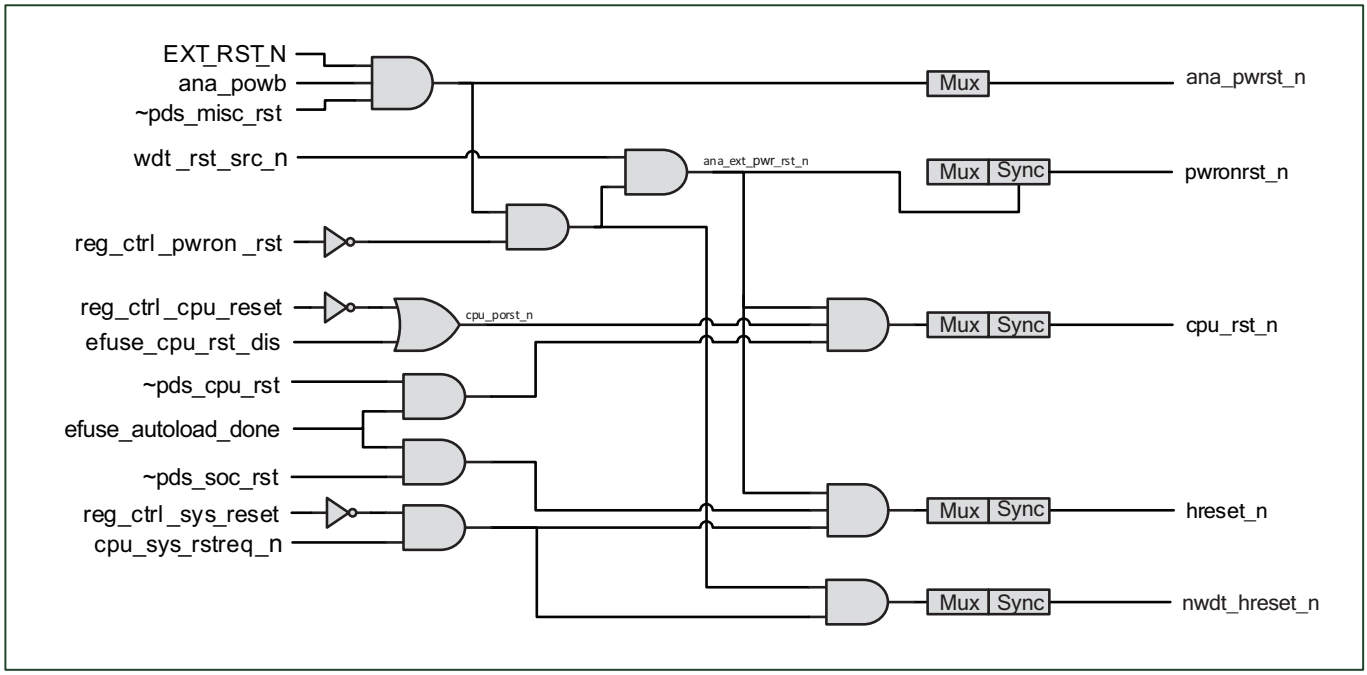

Figure 2.1: Reset source

2.3 Clock source

The clock source includes:

- XTAL: External crystal oscillator clock, mainly supporting frequency 32MHz
- XTAL32K: external crystal oscillator clock, frequency 32KHz
- RC32M: RC oscillator clock, frequency 32MHz, provides calibration
- PLL: PLL clock, internal system high-speed clock, the highest frequency supports 144MHz

The clock control unit distributes the clock from the oscillator to the core and peripheral devices.

The system clock source, dynamic divider, clock configuration, and sleep use 32KHz clock can be selected to achieve low-power clock management.

Peripheral clocks include: Flash, USB2.0, Ethernet, UART, I2C, I2S, SPI, PWM, IR-remote, QDEC, KeyScan, ADC, DAC.

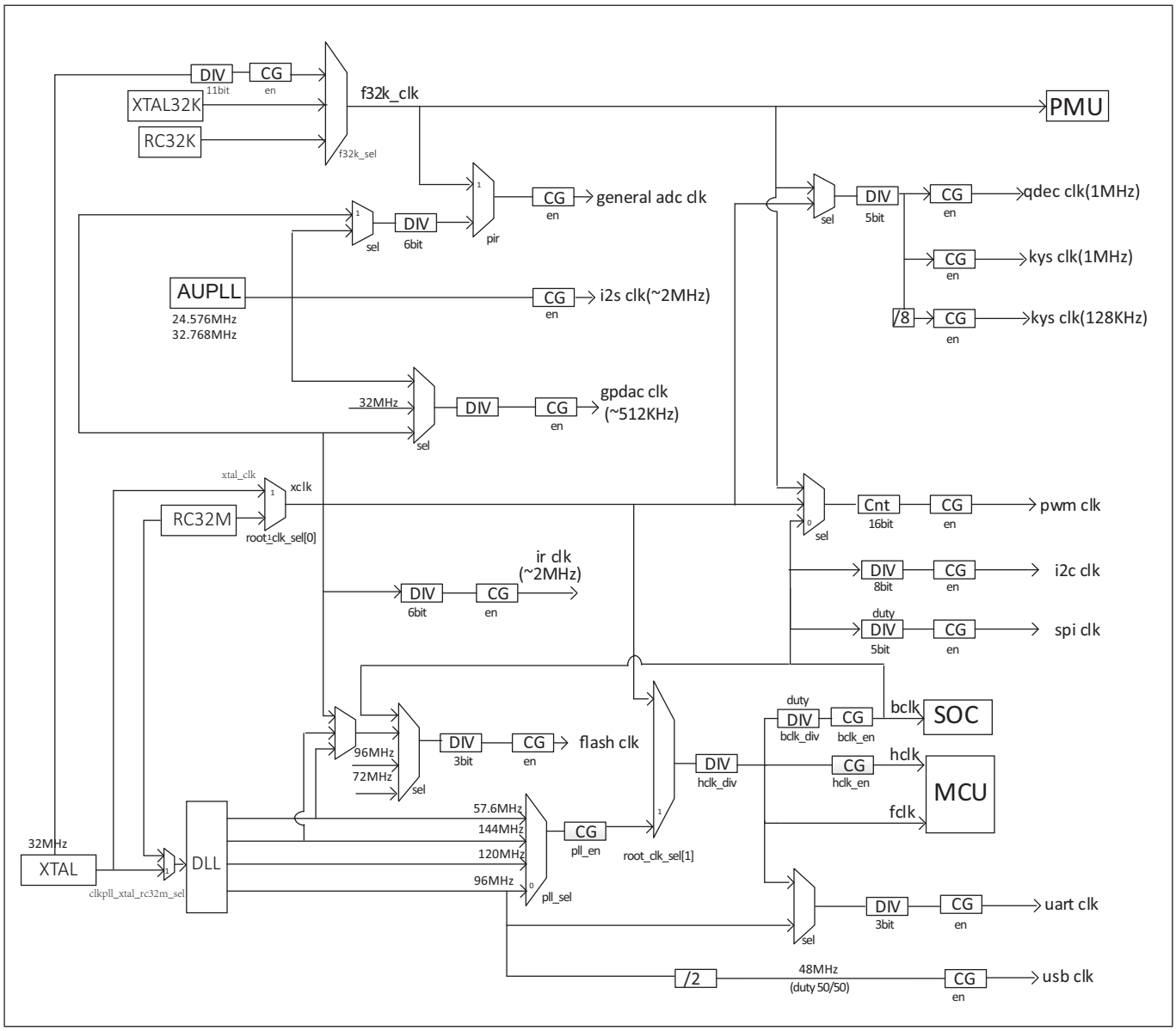

Figure 2.2: Clock architecture

3 GLB

3.1 Introduction

GLB (Global Register) is a chip general global setting module, which mainly includes functions such as clock management, reset management, bus management, memory management, and GPIO management.

3.2 Function description

3.2.1 Clock management

The clock management function is mainly used to set the clock of the processor, bus, and various peripherals. Through this module, you can set the clock source and clock frequency division of the above-mentioned module. At the same time, it can also realize the gating of the above-mentioned module clock. To achieve the purpose of low power consumption of the system.

For detailed settings, please refer to the chapters on reset and clock.

3.2.2 Reset management

Provide individual reset function and chip reset function of each peripheral module. Chip reset includes:

- CPU reset: just reset the CPU module, the program will run again, and the peripherals will not be reset
- System reset: All peripherals and CPU will be reset, but related registers in the AON domain will not be reset
- Power-on reset: the entire system including the related registers of the AON domain will be reset

The application can choose to use the corresponding reset method as needed.

Table 3.1: Software reset function table

BL702	EXT_RST (GPIO) ana_powb(HBN) wdt_rst_src_n(Watch Dog) reg_ctrl_pwron_rst	SW.Reset	cpuresetreq (CPU) pds_rst_soc reg_ctrl_sys_reset	pds_rst_np reg_ctrl_cpu_reset	pds_rst_usb
CPU	✓			✓	
bus	✓		✓		
glb	✓	swrst_s1[0]			
gpip	✓	swrst_s1[2]	✓		
I1c	✓	swrst_s1[9]	✓	✓	
dma	✓	swrst_s1[12]	✓		
emac	✓	swrst_s1[13]	✓		
usb	✓	swrst_s1a[12]	✓		✓
pds		swrst_s1[14]			
uart0	✓	swrst_s1a[0]	✓		
uart1	✓	swrst_s1a[1]	✓		
spi	✓	swrst_s1a[2]	✓		
I2C	✓	swrst_s1a[3]	✓		
pwm	✓	swrst_s1a[4]	✓		
timer	✓	swrst_s1a[5]	✓		
irr	✓	swrst_s1a[6]	✓		
qdec0	✓	swrst_s1a[8]	✓		
qdec1	✓	swrst_s1a[8]	✓		
qdec2	✓	swrst_s1a[8]	✓		
kys	✓	swrst_s1a[9]	✓		
i2s	✓	swrst_s1a[10]	✓		

3.2.3 Bus management

Provide bus arbitration settings and bus error settings, you can set whether to generate an interrupt when a bus error occurs, and provide error bus address information to facilitate user debugging.

3.2.4 Memory management

Provides power management of each memory module in the chip system low power mode, including two setting modes:

- Retention mode: In this mode, the data on the memory can be saved, but it cannot be read or written before exiting the low-power mode
- Sleep mode: In this mode, the data in the memory will be lost and it is only used to reduce system power consumption

3.2.5 GPIO overview

The GPIO management function provides GPIO control registers to implement software configuration of GPIO properties, enabling users to conveniently operate GPIO. Each GPIO can be configured as software GPIO or other multiplexing functions. Under each function, three port states are provided: pull-up, pull-down, and floating (must be set to floating when configured as an analog function). In addition, GPIO also provides interrupt function, which can be configured as rising edge trigger, falling edge trigger, high level trigger or low level trigger. Each GPIO can have two sets of interrupt configurations, and the two sets of interrupt configurations can take effect at the same time. For example, if the INT0 of GPIO0 is configured as a rising edge trigger and INT1 is configured as a falling edge trigger, the final effect is that both edges of GPIO will trigger interrupts.

3.2.6 GPIO main features

- Can be configured as software GPIO function
- It can be configured as other multiplexed functions and used with peripheral functions. When configured as analog functions, it must be set to floating
- You can set the input or output mode and set it as pull-up, pull-down or floating
- The drive capacity can be set to provide greater output current
- Schmitt trigger function can be set to provide simple hardware anti-shake function

3.2.7 GPIO function description

Each GPIO can be configured by software as:

- Multiple functions: I2S, SPI, I2C, UART, PWM, USB, SWGPIO... up to 24 functions
- InputEnable/OutputEnable: input, output, high impedance (ie=0, oe=0)
- PullUp/PullDown: pull up, pull down, float (pu=0, pd=0)
- drive strength: four gears of 0, 1, 2, and 3. The larger the value, the stronger the drive ability
- smt trigger: smt enable, smt disable, to prevent jitter near the trigger threshold

When the GPIO multiplexing function is configured as SWGPIO input, you can also configure the interrupt trigger mode for it, and each GPIO can have two interrupt modes, and both interrupt modes can be effective separately/simultaneously:

- Interrupt mode 1: rising edge trigger, falling edge trigger, high level trigger, low level trigger
- Interrupt mode 2: rising edge trigger, falling edge trigger, high level trigger, low level trigger

The basic block diagram of the GPIO module is shown in the figure.

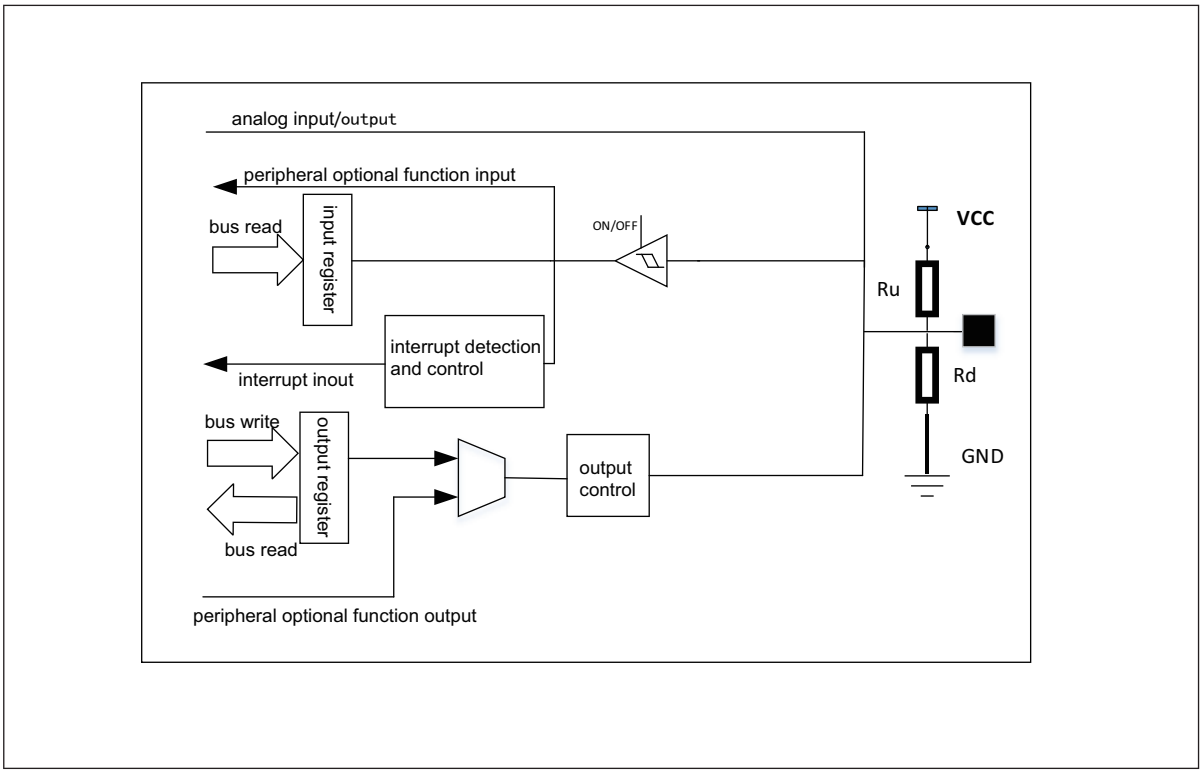

Figure 3.1: Basic block diagram of GPIO

3.2.8 GPIO function setting

The function of GPIO is set through the GPIO_CFGCTL register group. The main setting items include:

- func_sel: select GPIO function
- pu: Choose whether to pull up
- pd: Choose whether to drop down
- drv: set drive capability
- smt: Choose whether to enable Schmitt trigger
- ie: set input enable
- oe: set output enable

The functions that GPIO can be set include:

- Flash/QSPI: Set GPIO as QSPI function, which can be connected to Flash as a program storage/running medium
- SPI: Set GPIO as SPI function
- I2C: Set GPIO as I2C function
- UART: Set GPIO as UART function
- PWM: set GPIO as PWM function
- ANA: Set GPIO as Analog function
- SWGPIO: set GPIO as general IO function
- JTAG: Set GPIO as JTAG function
- Other multiplexing functions

In order to meet customer needs as much as possible, each GPIO can basically select the above optional functions. When an optional function is selected, GPIO and the corresponding function signal are shown in the following table:

Table 3.2: GPIO function table 1

GPIO	CLK_OUT	Flash_PSRAM	I2S	SPI0	I2C	UART	PWM
GPIO0	clk_out[0]		BCLK	MISO/MOSI	SCL	SIG0/SIG4	PWM[0]
GPIO1	clk_out[1]		FS	MOSI/MISO	SDA	SIG1/SIG5	PWM[1]
GPIO2	clk_out[0]		DIO/DO	SS	SCL	SIG2/SIG6	PWM[2]
GPIO3	clk_out[1]		RCLK_O/DI	SCLK	SDA	SIG3/SIG7	PWM[3]
GPIO4	clk_out[0]		BCLK	MISO/MOSI	SCL	SIG4/SIG0	PWM[4]
GPIO5	clk_out[1]		FS	MOSI/MISO	SDA	SIG5/SIG1	PWM[0]
GPIO6	clk_out[0]		DIO/DO	SS	SCL	SIG6/SIG2	PWM[1]
GPIO7	clk_out[1]		RCLK_O/DI	SCLK	SDA	SIG7/SIG3	PWM[2]
GPIO8	clk_out[0]		BCLK	MISO/MOSI	SCL	SIG0/SIG4	PWM[3]
GPIO9	clk_out[1]		FS	MOSI/MISO	SDA	SIG1/SIG5	PWM[4]
GPIO10	clk_out[0]		DIO/DO	SS	SCL	SIG2/SIG6	PWM[0]
GPIO11	clk_out[1]		RCLK_O/DI	SCLK	SDA	SIG3/SIG7	PWM[1]
GPIO12	clk_out[0]		BCLK	MISO/MOSI	SCL	SIG4/SIG0	PWM[2]
GPIO13	clk_out[1]		FS	MOSI/MISO	SDA	SIG5/SIG1	PWM[3]
GPIO14	clk_out[0]		DIO/DO	SS	SCL	SIG6/SIG2	PWM[4]
GPIO15	clk_out[1]		RCLK_O/DI	SCLK	SDA	SIG7/SIG3	PWM[0]
GPIO16	clk_out[0]		BCLK	MISO/MOSI	SCL	SIG0/SIG4	PWM[1]

Table 3.2: GPIO function table 1

GPIO	CLK_OUT	Flash_PSRAM	I2S	SPI0	I2C	UART	PWM
GPIO17	clk_out[1]	SF_IO_0/SF2_CS2	FS	MOSI/MISO	SDA	SIG1/SIG5	PWM[2]
GPIO18	clk_out[0]	SF_IO_1	DIO/DO	SS	SCL	SIG2/SIG6	PWM[3]
GPIO19	clk_out[1]	SF_CS	RCLK_O/DI	SCLK	SDA	SIG3/SIG7	PWM[4]
GPIO20	clk_out[0]	SF_IO_3	BCLK	MISO/MOSI	SCL	SIG4/SIG0	PWM[0]
GPIO21	clk_out[1]	SF_CLK	FS	MOSI/MISO	SDA	SIG5/SIG1	PWM[1]
GPIO22	clk_out[0]	SF_IO_2	DIO/DO	SS	SCL	SIG6/SIG2	PWM[2]
GPIO23	clk_out[1]	SF2_IO_2	RCLK_O/DI	SCLK	SDA	SIG7/SIG3	PWM[3]
GPIO24	clk_out[0]	SF2_IO_1	BCLK	MISO/MOSI	SCL	SIG0/SIG4	PWM[4]
GPIO25	clk_out[1]	SF2_CS	FS	MOSI/MISO	SDA	SIG1/SIG5	PWM[0]
GPIO26	clk_out[0]	SF2_IO_3	DIO/DO	SS	SCL	SIG2/SIG6	PWM[1]
GPIO27	clk_out[1]	SF2_CLK	RCLK_O/DI	SCLK	SDA	SIG3/SIG7	PWM[2]
GPIO28	clk_out[0]	SF2_IO_0	BCLK	MISO/MOSI	SCL	SIG4/SIG0	PWM[3]
GPIO29	clk_out[1]		FS	MOSI/MISO	SDA	SIG5/SIG1	PWM[4]
GPIO30	clk_out[0]		DIO/DO	SS	SCL	SIG6/SIG2	PWM[0]
GPIO31	clk_out[1]		RCLK_O/DI	SCLK	SDA	SIG7/SIG3	PWM[1]

Table 3.3: GPIO function table 2

GPIO	Analog	SWGPIO	JTAG	Ether_Mac	QDEC
GPIO0		GPIO[0]	TMS/TCK	MII_REF_CLK	a
GPIO1		GPIO[1]	TDI/TDO	MII_TXD[0]	b
GPIO2		GPIO[2]	TCK/TMS	MII_TXD[1]	led
GPIO3		GPIO[3]	TDO/TDI		a
GPIO4		GPIO[4]	TMS/TCK		b
GPIO5		GPIO[5]	TDI/TDO		led
GPIO6		GPIO[6]	TCK/TMS		a
GPIO7	USB_DP/gpip_ch[6]/gpdac_vref_ext	GPIO[7]	TDO/TDI	MII_RXD[0]	b
GPIO8	USB_DM/gpip_ch[0]	GPIO[8]	TMS/TCK	MII_RXD[1]	led
GPIO9	pmip_dc_tp/clkpll_dc_tp/gpip_ch[7]	GPIO[9]	TDI/TDO		a
GPIO10	MICBIAS	GPIO[10]	TCK/TMS		b

Table 3.3: GPIO function table 2

GPIO	Analog	SWGPIO	JTAG	Ether_Mac	QDEC
GPIO11	gpip_ch[3]	GPIO[11]	TDO/TDI		led
GPIO12	gpip_ch[4]	GPIO[12]	TMS/TCK		a
GPIO13		GPIO[13]	TDI/TDO		b
GPIO14	gpip_ch[5]/atest_out_0	GPIO[14]	TCK/TMS		led
GPIO15	gpip_ch[1]/atest_out_1	GPIO[15]	TDO/TDI		a
GPIO16		GPIO[16]	TMS/TCK		b
GPIO17	gpip_ch[2]/psw_irrcv	GPIO[17]	TDI/TDO		led
GPIO18	gpip_ch[8]	GPIO[18]	TCK/TMS	RMII_MDC	a
GPIO19	gpip_ch[9]	GPIO[19]	TDO/TDI	RMII_MDIO	b
GPIO20	gpip_ch[10]	GPIO[20]	TMS/TCK	RMII_RXERR	led
GPIO21	gpip_ch[11]	GPIO[21]	TDI/TDO	RMII_TX_EN	a
GPIO22	leddrv[0]	GPIO[22]	TCK/TMS	RMII_RX_DV	b
GPIO23	leddrv[1]/flash_pull_out[0]	GPIO[23]	TDO/TDI		led
GPIO24	flash_pull_out[1]	GPIO[24]	TMS/TCK	RMII_MDC	a
GPIO25	flash_pull_out[2]	GPIO[25]	TDI/TDO	RMII_MDIO	b
GPIO26	flash_pull_out[3]	GPIO[26]	TCK/TMS	RMII_RXERR	led
GPIO27	flash_pull_out[4]	GPIO[27]	TDO/TDI	RMII_TX_EN	a
GPIO28	flash_pull_out[5]	GPIO[28]	TMS/TCK	RMII_RX_DV	b
GPIO29		GPIO[29]	TDI/TDO		led
GPIO30		GPIO[30]	TCK/TMS		a
GPIO31		GPIO[31]	TDO/TDI		b

Table 3.4: GPIO function table 3

GPIO	SWGPIO
GPIO	SWGPIO
GPIO17	pad_irrx_i,irrxgpsl=1
GPIO18	pad_irrx_i,irrxgpsl=2
GPIO19	pad_irrx_i,irrxgpsl=3
GPIO20	pad_irrx_i,irrxgpsl=4

Table 3.4: GPIO function table 3

GPIO	SWGPIO
GPIO21	pad_irrx_i,irrxgpsl=5
GPIO22	pad_irrx_i,irrxgpsl=6
GPIO23	pad_irrx_i,irrxgpsl=7
GPIO24	pad_irrx_i,irrxgpsl=8
GPIO25	pad_irrx_i,irrxgpsl=9
GPIO26	pad_irrx_i,irrxgpsl=10
GPIO27	pad_irrx_i,irrxgpsl=11
GPIO28	pad_irrx_i,irrxgpsl=12
GPIO29	pad_irrx_i,irrxgpsl=13
GPIO30	pad_irrx_i,irrxgpsl=14
GPIO31	pad_irrx_i,irrxgpsl=15

When using the IR function, you need to set GPIO as SWGPIO and set the irrxgpsl register (GPIO17~GPIO31 can be used as IR pins)

In the above table, when the UART function is selected, only a signal of the UART is selected, and the specific function of the pin (such as UART TX or UART RX) is not specified. The user needs to further select specific UART signals and corresponding functions through UART_SIGX_SEL (X=0-7).

The signals that can be selected for each UART_SIGX_SEL include:

- 0 : UART0_RTS
- 1 : UART0_CTS
- 2 : UART0_TXD
- 3 : UART0_RXD
- 4 : UART1_RTS
- 5 : UART1_CTS
- 6 : UART1_TXD
- 7 : UART1_RXD

Take GPIO0 as an example, when fun_sel selects UART, GPIO0 selects UART_SIG0. By default, the value of UART_SIG0_SEL is 0, which is UART0_RTS, that is, GPIO is UART0_RTS function. If the application wants to use GPIO

as UART1_TXD, then just set UART_SIG0_SEL to 6, then the function of GPIO0 is UART1_TXD.

3.2.9 GPIO output settings

By setting func_sel to SWGPIO, GPIO can be used as the input/output of ordinary GPIO. Set IE to 0 and OE to 1, then GPIO can be configured as an output function, and the output value is set through the GPIO_O register group.

When the corresponding Bit of GPIO_O is set to 0, GPIO outputs low level. When the corresponding Bit of GPIO_O is set to 1, GPIO outputs high level. The output capability can be set through the DRV control bit.

3.2.10 GPIO input settings

By setting func_sel to SWGPIO, setting IE to 1, and OE to 0, users can configure GPIO as an input function. Set whether to enable the Schmitt trigger through the SMT control bit, and set the pull-up and pull-down attributes through the PD and PU control bits.

The value of external input can be obtained by reading the bit corresponding to the GPIO_I register.

3.2.11 GPIO optional function settings

By setting func_sel as the corresponding peripheral function, the connection between the GPIO and the peripheral can be realized, and the input and output of the peripheral can be realized. It can be seen from the basic functional block diagram of GPIO that when selecting optional functions, you need to set IE to 1, and OE to 0, which means that the output control function of ordinary GPIO is disconnected.

In this way, for a peripheral with a fixed input function, the OE signal of the peripheral is always 0, thereby realizing the input function. For a fixed output peripheral, its OE signal is always 1, so that the output is controlled by the peripheral, and the input signal at this time is the output signal, but it will not be collected by the peripheral being output. When the peripheral needs both input and output, the input and output can be realized by controlling the OE signal of the peripheral.

That is: for functions other than swgpio, as the output direction function, the configuration values of IE and OE do not affect the function. But as the input direction, IE must be set and the configuration of OE does not affect the function; when used as swgpio, both IE and OE need to be configured correctly.

3.2.12 GPIO interrupt settings

To use the interrupt function of GPIO, you need to set GPIO to input mode first, and the interrupt trigger mode is set through the GPIO_INT_MODE_SET register group. The interrupt modes that can be set include:

- Falling edge trigger interrupt
- Rising edge trigger interrupt
- Low level trigger interrupt
- High level trigger interrupt

Each GPIO can be set as an interrupt function. Whether to enable a GPIO interrupt can be set through the GPIO_INT_MASK register. When an interrupt is generated, the GPIO pin number that generated the interrupt can be obtained through the GPIO_INT_STAT register in the interrupt function. Clear the corresponding interrupt signal through GPIO_INT_CLR.

3.3 Register description

Name	Description
clk_cfg0	Clock configuration-processor, bus
clk_cfg1	Clock configuration1-ble,qdec,i2s
clk_cfg2	Clock configuration-UART,Flash
clk_cfg3	Clock configuration-I2C,SPI
swrst_cfg0	software reset configuration0
swrst_cfg1	software reset configuration1
swrst_cfg2	software reset configuration2
cgen_cfg0	clock ungate configuration0
cgen_cfg1	clock ungate configuration1
bmx_cfg1	Bus configuration1
bmx_cfg2	Bus configuration2
bmx_err_addr	Bus error address
seam_misc	em select
CPU_CLK_CFG	CPU clock configuration
GPADC_32M_SRC_CTRL	Clock configuration-GPADC
DIG32K_WAKEUP_CTRL	DIG32K wakeup control
UART_SIG_SEL_0	UART Signal Select
GPIO_CFGCTL0	GPIO0, GPIO1 configuration
GPIO_CFGCTL1	GPIO2, GPIO3 configuration
GPIO_CFGCTL2	GPIO4, GPIO5 configuration
GPIO_CFGCTL3	GPIO6, GPIO7 configuration
GPIO_CFGCTL4	GPIO8, GPIO9 configuration
GPIO_CFGCTL5	GPIO10, GPIO11 configuration
GPIO_CFGCTL6	GPIO12, GPIO13 configuration

Name	Description
GPIO_CFGCTL7	GPIO14, GPIO15 configuration
GPIO_CFGCTL8	GPIO16, GPIO17 configuration
GPIO_CFGCTL9	GPIO18, GPIO19 configuration
GPIO_CFGCTL10	GPIO20, GPIO21 configuration
GPIO_CFGCTL11	GPIO22, GPIO23 configuration
GPIO_CFGCTL12	GPIO24, GPIO25 configuration
GPIO_CFGCTL13	GPIO26, GPIO27 configuration
GPIO_CFGCTL14	GPIO28,GPIO29 configuration
GPIO_CFGCTL15	GPIO30,GPIO31 configuration
GPIO_CFGCTL16	GPIO32,GPIO33 configuration
GPIO_CFGCTL17	GPIO34,GPIO35 configuration
GPIO_CFGCTL18	GPIO36,GPIO37 configuration
GPIO_CFGCTL30	Register Controlled GPIO Input value
GPIO_CFGCTL32	Register controlled GPIO output value
GPIO_CFGCTL34	Register controlled GPIO output enable
GPIO_CFGCTL35	Register controlled GPIO output enable
GPIO_INT_MASK1	GPIO interrupt mask
GPIO_INT_STAT1	GPIO interrupt state
GPIO_INT_CLR1	GPIO interrupt clear
GPIO_INT_MODE_SET1	GPIO interrupt mode set1
GPIO_INT_MODE_SET2	GPIO interrupt mode set2
GPIO_INT_MODE_SET3	GPIO interrupt mode set3
GPIO_INT_MODE_SET4	GPIO interrupt mode set4
GPIO_INT2_MASK1	GPIO interrupt2 mask1
GPIO_INT2_STAT1	GPIO interrupt2 state1
GPIO_INT2_CLR1	GPIO interrupt2 clear1
GPIO_INT2_MODE_SET1	GPIO interrupt 2 mode set1
GPIO_INT2_MODE_SET2	GPIO interrupt 2 mode set2
GPIO_INT2_MODE_SET3	GPIO interrupt 2 mode set3

Name	Description
GPIO_INT2_MODE_SET4	GPIO interrupt 2 mode set4
led_driver	Led driver
usb_xcvr	USB configuration
usb_xcvr_config	USB xcvr configuration

3.3.1 clk_cfg0

Address：0x40000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD								BCLKDIV							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCLKDIV								RCSEL		PLLSEL		BLCK EN	HCLK EN	FCLK EN	PLL EN

Bits	Name	Type	Reset	Description
31:24	RSVD			
23:16	BCLKDIV	R/W	0	bclk divide from hclk
15:8	HCLKDIV	R/W	0	hclk divide from root clock (clock source selected by hbn_-root_clk_sel)
7:6	RCSEL	R	0	root clock selection from HBN (0: RC32M 1: XTAL 2/3: PLL others)
5:4	PLLSEL	R/W	0	pll clock selection 0: 57.6MHz 1: 96MHz 2: 144MHz 3: 120MHz (Clock Freq will be changes with Audio PLL, not sugges use this CLK)
3	BLCKEN	R/W	1	bclk force on
2	HCLKEN	R/W	1	hclk force on
1	FCLKEN	R/W	1	fclk force on
0	PLLEN	R/W	1	pll clock enable for fclk

3.3.2 clk_cfg1

Address：0x40000004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD						MACZBEN	BLEEN	RSVD		BLECLKSL					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	I2SCLK	I2SCLEN	I2SCSEL	RSVD		USBDIV	USBCLK	QDECCSEL	RSVD		QDECCDIV				

Bits	Name	Type	Reset	Description
31:26	RSVD			
25	MACZBEN	R/W	1	mac154 zigbee clock enable
24	BLEEN	R/W	1	ble clock enable
23:22	RSVD			
21:16	BLECLKSL	R/W	6'd16	HW reserved ; for ble to generate 1us/0.5us tick pulse
15	RSVD			
14	I2SCLK	R/W	0	0 : no output reference clock on I2S_0 ref_clock port ; 1: output reference clock on I2S_0 ref_clock port
13	I2SCLEN	R/W	0	I2S0 Clock Enable
12	I2SCSEL	R/W	0	I2S Clock Source Select. 0: Audio PLL, 1: reserved
11:10	RSVD			
9	USBDIV	R/W	1	USB Clock div2 from 96MHz
8	USBCLK	R/W	1	USB Clock Enable
7	QDECCSEL	R/W	0	QDEC Clock Source Select. 0:xclk 1:f32k (PDS mode)
6:5	RSVD			
4:0	QDECCDIV	R/W	5'd31	QDEC Clock Divider qdec_clk_sel/(N+1), default 1MHz

3.3.3 clk_cfg2

Address：0x40000008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAEN								IRCLEN	RSVD	IRCLDIV					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SFSEL2		SFSEL		SFEN	SFDIV			HUCSEL	RSVD		UARTEN	RSVD	UARTDIV		

Bits	Name	Type	Reset	Description
31:24	DMAEN	R/W	8'hff	DMA CH0 7 Clock Enable ; config all 0 if disable DMA
23	IRCLEN	R/W	1	IR remote Clock Enable
22	RSVD			
21:16	IRCLDIV	R/W	6'd15	IR remote Clock Divider xclk(N+1)
15:14	SFSEL2	R/W	0	For sf_clk_sel=0 0: 144MHz 1: xclk 3: 57.6MHz
13:12	SFSEL	R/W	2'd2	Flash Clock Select (0:sf_clk_sel2, 1:72M, 2:bclk, 3:96M)
11	SFEN	R/W	1	Flash Clock Enable
10:8	SFDIV	R/W	3'd3	Flash Clock Divider (Selected Flash Clock)/(N+1)
7	HUCSEL	R	0	uart clock selection from HBN (0:fclk 1:96MHz)
6:5	RSVD			
4	UARTEN	R/W	1	UART Clock Enable
3	RSVD			
2:0	UARTDIV	R/W	3'd7	UART Clock Divider hbn_uart_clk_sel/(N+1)

3.3.4 clk_cfg3

Address：0x4000000c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCO1S		CCO0S		RSVD			I2CEN	I2CDIV							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					ICTMRX	IRFTCLK	SPIEN	ICTMTX	ICTPHY	SREFCLK	SPIDIV				

Bits	Name	Type	Reset	Description
31:30	CCO1S	R/W	2'b0	clk_out_1 selection (0: no chip clock out, 1: i2s_ref_clk out, 2: audio_pll_clk out, 3: clk_xtal_soc_32M)
29:28	CCO0S	R/W	2'b0	clk_out_0 selection (0: no chip clock out, 1: i2s_ref_clk out, 2: audio_pll_clk out, 3: clk_xtal_soc_32M)
27:25	RSVD			
24	I2CEN	R/W	1	I2C Master Clock Out Enable
23:16	I2CDIV	R/W	8'd255	I2C Master Clock Out Divider (Freq_of_BCLK/(N+1))
15:11	RSVD			
10	ICTMRX	R/W	1'b1	Invert clock to our RMII MAC RX
9	IRFTCLK	R/W	1	1: invert rf_test_clk out, 0 : no change
8	SPIEN	R/W	1	SPI Clock Enable (Default : Enable)
7	ICTMTX	R/W	1'b1	Invert clock to our RMII MAC TX
6	ICTPHY	R/W	1'b1	if [5] = 1, set this bit to 1 will invert the clock to RMII PHY
5	SREFCLK	R/W	1'b0	0 : select outside 50MHz RMII ref clock, 1: select inside 50MHz RMII ref clock
4:0	SPIDIV	R/W	5'd3	SPI Clock Divider (BUS_CLK/(N+1)), default BUS_CLK/4

3.3.5 swrst_cfg0

Address：0x40000010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							SRS30	RSVD			SRS20	RSVD			

Bits	Name	Type	Reset	Description
31:9	RSVD			
8	SRS30	R/W	0	software reset BLE
7:5	RSVD			
4	SRS20	R/W	0	software reset M154
3:0	RSVD			

3.3.6 swrst_cfg1

Address：0x40000014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			SRS1AC	RSVD	SRS1AA	SRS1A9	SRS1A8	SRS1A7	SRS1A6	SRS1A5	SRS1A4	SRS1A3	SRS1A2	SRS1A1	SRS1A0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	SRS1E	SRS1D	SRS1C	SRS1B	RSVD	SRS19	SRS18	SRS17	SRS16	SRS15	SRS14	SRS13	SRS12	SRS11	SRS10

Bits	Name	Type	Reset	Description
31:29	RSVD			
28	SRS1AC	R/W	0	software reset USB
27	RSVD			
26	SRS1AA	R/W	0	software reset I2S
25	SRS1A9	R/W	0	software reset KYS
24	SRS1A8	R/W	0	software reset QDEC
23	SRS1A7	R/W	0	software reset Checksum
22	SRS1A6	R/W	0	software reset IR Remote
21	SRS1A5	R/W	0	software reset Timer
20	SRS1A4	R/W	0	software reset PWM
19	SRS1A3	R/W	0	software reset I2C
18	SRS1A2	R/W	0	software reset SPI
17	SRS1A1	R/W	0	software reset UART1
16	SRS1A0	R/W	0	software reset UART0
15	RSVD			
14	SRS1E	R/W	0	software reset PDS
13	SRS1D	R/W	0	software reset EMAC
12	SRS1C	R/W	0	software reset DMA
11	SRS1B	R/W	0	software reset SF
10	RSVD			
9	SRS19	R/W	0	software reset L1C
8	SRS18	R/W	0	software reset CCI
7	SRS17	R/W	0	software reset eFuse
6	SRS16	R/W	0	software reset TZ2

Bits	Name	Type	Reset	Description
5	SRS15	R/W	0	software reset TZ1
4	SRS14	R/W	0	software reset SEC_ENG
3	SRS13	R/W	0	software reset SEC_DBG
2	SRS12	R/W	0	software reset GPIP
1	SRS11	R/W	0	software reset MIX
0	SRS10	R/W	0	software reset GLB

3.3.7 swrst_cfg2

Address：0x40000018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD							PKA CLK	RSVD							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								RSTDUM				RSVD	SYS RST	CPU RST	PWO RST

Bits	Name	Type	Reset	Description
31:25	RSVD			
24	PKACLK	R/W	0	pka clock select 0:HCLK 1:96M
23:8	RSVD			
7:4	RSTDUM	R/W	0	reset dummy
3	RSVD			
2	SYSRST	R/W	0	system reset, see the software reset table for details
1	CPURST	R/W	0	cpu reset, see the software reset table for details
0	PWORST	R/W	0	power on reset, see the software reset table for details

3.3.8 cgen_cfg0

Address：0x40000020

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	CGENM	R/W	8'hff	[0] CPU clock ungate enable [1] SDU clock ungate enable [2] SEC clock ungate enable [3] DMA clock ungate enable [4] CCI clock ungate enable

3.3.9 cgen_cfg1

Address：0x40000024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CGENS1A															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CGENS1															

Bits	Name	Type	Reset	Description
31:16	CGENS1A	R/W	16'h9B23	[0] uart0 clock ungate enable [1] uart1 clock ungate enable [2] spi clock ungate enable [3] i2c clock ungate enable [4] pwm clock ungate enable [5] timer clock ungate enable [6] ir_remote clock ungate enable [7] checksum clock ungate enable [8] qdec0 clock ungate enable [9] qdec1 clock ungate enable [10] i2s and qdec2 clock ungate enable [11] kys clock ungate enable [12] usb clock ungate enable

Bits	Name	Type	Reset	Description
15:0	CGENS1	R/W	16'hCFFF	[2] gpip (gpadc, gpdac) clock ungate enable [3] sec_dbg clock ungate enable [4] sec_eng clock ungate enable [5] tzc clock ungate enable [7] ef_ctrl clock ungate enable [11] sf_ctrl clock ungate enable [12] DMA clock ungate enable [13] emac clock ungate enable

3.3.10 bmx_cfg1

Address：0x40000050

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

Bits	Name	Type	Reset	Description
31:9	RSVD			
8	ERREN	R/W	0	Bus error response enable
7:6	RSVD			
5:4	ARBMODE	R/W	0	Bus arbiter: 0:fix, 2:round-robin, 3:random
3:0	TIMOUTEN	R/W	0	Bus timeout enable: detect slave no reaponse in 1024 cycles

3.3.11 bmx_cfg2

Address：0x40000054

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

Bits	Name	Type	Reset	Description
31:6	RSVD			
5	ERRTZ	R	0	Bus trustzone decode error
4	ERRDEC	R	0	Bus addr decode error
3:1	RSVD			
0	ADDRDIS	R/W	0	Bus addr error monitor disable

3.3.12 bmx_err_addr

Address：0x40000058

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

Bits	Name	Type	Reset	Description
31:0	ADDR	R	0	Bus error addr (the first one)

3.3.13 seam_misc

Address：0x4000007c

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3:0	EMSEL	R/W	4'h3	em/ocram share memory em select 0x0: 0KB ; 0x3:8KB ; 0xF:16KB

3.3.14 CPU_CLK_CFG

Address：0x40000090

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

Bits	Name	Type	Reset	Description
31:20	RSVD			
19	RTCSEL	R/W	1	0: Bus clock 1: 32KHz
18	RTCEN	R/W	0	CPU RTC clock enable
17	RSVD			
16:0	RTCDIV	R/W	17'h10	CPU RTC clock = (e21_rtc_sel) / (e21_rtc_div+1)

3.3.15 GPADC_32M_SRC_CTRL

Address：0x400000a4

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

Bits	Name	Type	Reset	Description
31:9	RSVD			
8	GADCDIV	R/W	1	GPADC 32M Clock Dvider Enable
7	GADCSEL	R/W	0	GPADC Clock Source Select. 0: Audio PLL, 1: xclk
6	RSVD			
5:0	GADCDIV	R/W	6'd2	GPADC 32M Clock Divider (gpadc clock)/(N+1)

3.3.16 DIG32K_WAKEUP_CTRL

Address：0x400000a8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		CLKSCSEL		RSVD		COMP	512K EN	RSVD	512KDIV						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		32K COMP	32K EN	RSVD	32KDIV										

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:28	CLKSCSEL	R/W	0	0:PLL 32MHz 1:xclk 2:Audio PLL
27:26	RSVD			
25	COMP	R/W	1	Compensation => Duty of dig_512k_out = N : N+1
24	512KEN	R/W	1	Enable dig_512k_out
23	RSVD			
22:16	512KDIV	R/W	7'd62	(PLL 32MHz or xclk) / dig_512k_div Ex: Set 46 for 24MHz ; Set 62 for 32MHz ; Set 75 for 38.4MHz ; Set 78 for 40MHz
15:14	RSVD			
13	32KCOMP	R/W	0	Compensation => Duty of dig_32k_out = dig_32k_div : (dig_32k_div+1)
12	32KEN	R/W	1	Enable dig_32k_out
11	RSVD			
10:0	32KDIV	R/W	11'd1000	(PLL 32MHz or xclk) / dig_32k_div Ex: Set 750 for 24MHz ; Set 1000 for 32MHz ; Set 1200 for 38.4MHz ; Set 1250 for 40MHz

3.3.17 UART_SIG_SEL_0

Address：0x400000c0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIG7SEL				SIG6SEL				SIG5SEL				SIG4SEL			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIG3SEL				SIG2SEL				SIG1SEL				SIG0SEL			

Bits	Name	Type	Reset	Description
31:28	SIG7SEL	R/W	4'd7	UART Signal7 Select : 0 : UART0_RTS 1 : UART0_CTS 2 : UART0_TXD 3 : UART0_RXD 4 : UART1_RTS 5 : UART1_CTS 6 : UART1_TXD 7 : UART1_RXD
27:24	SIG6SEL	R/W	4'd6	UART Signal6 Select : 0 : UART0_RTS 1 : UART0_CTS 2 : UART0_TXD 3 : UART0_RXD 4 : UART1_RTS 5 : UART1_CTS 6 : UART1_TXD 7 : UART1_RXD
23:20	SIG5SEL	R/W	4'd5	UART Signal5 Select : 0 : UART0_RTS 1 : UART0_CTS 2 : UART0_TXD 3 : UART0_RXD 4 : UART1_RTS 5 : UART1_CTS 6 : UART1_TXD 7 : UART1_RXD
19:16	SIG4SEL	R/W	4'd4	UART Signal4 Select : 0 : UART0_RTS 1 : UART0_CTS 2 : UART0_TXD 3 : UART0_RXD 4 : UART1_RTS 5 : UART1_CTS 6 : UART1_TXD 7 : UART1_RXD

Bits	Name	Type	Reset	Description
15:12	SIG3SEL	R/W	4'd3	UART Signal3 Select : 0 : UART0_RTS 1 : UART0_CTS 2 : UART0_TXD 3 : UART0_RXD 4 : UART1_RTS 5 : UART1_CTS 6 : UART1_TXD 7 : UART1_RXD
11:8	SIG2SEL	R/W	4'd2	UART Signal2 Select : 0 : UART0_RTS 1 : UART0_CTS 2 : UART0_TXD 3 : UART0_RXD 4 : UART1_RTS 5 : UART1_CTS 6 : UART1_TXD 7 : UART1_RXD
7:4	SIG1SEL	R/W	4'd1	UART Signal1 Select : 0 : UART0_RTS 1 : UART0_CTS 2 : UART0_TXD 3 : UART0_RXD 4 : UART1_RTS 5 : UART1_CTS 6 : UART1_TXD 7 : UART1_RXD
3:0	SIG0SEL	R/W	0	UART Signal0 Select : 0 : UART0_RTS 1 : UART0_CTS 2 : UART0_TXD 3 : UART0_RXD 4 : UART1_RTS 5 : UART1_CTS 6 : UART1_TXD 7 : UART1_RXD

3.3.18 GPIO_CFGCTL0

Address：0x40000100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP1FUNC					RSVD		GP1 PD	GP1 PU	GP1DRV		GP1 SMT	GP1 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP0FUNC					RSVD		GP0 PD	GP0 PU	GP0DRV		GP0 SMT	GP0 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP1FUNC	R/W	5'hE	GPIO Function Select (Default : JTAG)
23:22	RSVD			
21	GP1PD	R/W	0	GPIO Pull Down Control
20	GP1PU	R/W	0	GPIO Pull Up Control
19:18	GP1DRV	R/W	0	GPIO Driving Control
17	GP1SMT	R/W	1	GPIO SMT Control
16	GP1IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP0FUNC	R/W	5'hE	GPIO Function Select (Default : JTAG)
7:6	RSVD			
5	GP0PD	R/W	0	GPIO Pull Down Control
4	GP0PU	R/W	0	GPIO Pull Up Control
3:2	GP0DRV	R/W	0	GPIO Driving Control
1	GP0SMT	R/W	1	GPIO SMT Control
0	GP0IE	R/W	1	GPIO Input Enable

3.3.19 GPIO_CFGCTL1

Address：0x40000104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP3FUNC					RSVD		GP3 PD	GP3 PU	GP3DRV		GP3 SMT	GP3 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP2FUNC					RSVD		GP2 PD	GP2 PU	GP2DRV		GP2 SMT	GP2 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP3FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP3PD	R/W	0	GPIO Pull Down Control
20	GP3PU	R/W	0	GPIO Pull Up Control
19:18	GP3DRV	R/W	0	GPIO Driving Control
17	GP3SMT	R/W	1	GPIO SMT Control
16	GP3IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP2FUNC	R/W	5'hE	GPIO Function Select (Default : JTAG)
7:6	RSVD			
5	GP2PD	R/W	0	GPIO Pull Down Control
4	GP2PU	R/W	0	GPIO Pull Up Control
3:2	GP2DRV	R/W	0	GPIO Driving Control
1	GP2SMT	R/W	1	GPIO SMT Control
0	GP2IE	R/W	1	GPIO Input Enable

3.3.20 GPIO_CFGCTL2

Address：0x40000108

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP5FUNC					RSVD		GP5 PD	GP5 PU	GP5DRV		GP5 SMT	GP5 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP4FUNC					RSVD		GP4 PD	GP4 PU	GP4DRV		GP4 SMT	GP4 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP5FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP5PD	R/W	0	GPIO Pull Down Control
20	GP5PU	R/W	0	GPIO Pull Up Control

Bits	Name	Type	Reset	Description
19:18	GP5DRV	R/W	0	GPIO Driving Control
17	GP5SMT	R/W	1	GPIO SMT Control
16	GP5IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP4FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP4PD	R/W	0	GPIO Pull Down Control
4	GP4PU	R/W	0	GPIO Pull Up Control
3:2	GP4DRV	R/W	0	GPIO Driving Control
1	GP4SMT	R/W	1	GPIO SMT Control
0	GP4IE	R/W	1	GPIO Input Enable

3.3.21 GPIO_CFGCTL3

Address：0x4000010c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP7FUNC					RSVD		GP7 PD	GP7 PU	GP7DRV		GP7 SMT	GP7 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP6FUNC					RSVD		GP6 PD	GP6 PU	GP6DRV		GP6 SMT	GP6 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP7FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP7PD	R/W	0	GPIO Pull Down Control
20	GP7PU	R/W	0	GPIO Pull Up Control
19:18	GP7DRV	R/W	0	GPIO Driving Control
17	GP7SMT	R/W	1	GPIO SMT Control
16	GP7IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP6FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)

Bits	Name	Type	Reset	Description
7:6	RSVD			
5	GP6PD	R/W	0	GPIO Pull Down Control
4	GP6PU	R/W	0	GPIO Pull Up Control
3:2	GP6DRV	R/W	0	GPIO Driving Control
1	GP6SMT	R/W	1	GPIO SMT Control
0	GP6IE	R/W	1	GPIO Input Enable

3.3.22 GPIO_CFGCTL4

Address：0x40000110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP9FUNC					RSVD		GP9 PD	GP9 PU	GP9DRV		GP9 SMT	GP9 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP8FUNC					RSVD		GP8 PD	GP8 PU	GP8DRV		GP8 SMT	GP8 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP9FUNC	R/W	5'hE	GPIO Function Select (Default : JTAG)
23:22	RSVD			
21	GP9PD	R/W	0	GPIO Pull Down Control (Use this bit if reg_en_hw_pu_pd := 0 (0x4000F014[16])
20	GP9PU	R/W	0	GPIO Pull Up Control (Use this bit if reg_en_hw_pu_pd := 0 (0x4000F014[16])
19:18	GP9DRV	R/W	0	GPIO Driving Control
17	GP9SMT	R/W	1	GPIO SMT Control(Useless, IE depend on reg_aon_pad_-ie_smt[0] : 0x4000F014[8])
16	GP9IE	R/W	0	GPIO Input Enable (Useless, IE depend on reg_aon_pad_-ie_smt[0] : 0x4000F014[8])
15:13	RSVD			
12:8	GP8FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP8PD	R/W	0	GPIO Pull Down Control

Bits	Name	Type	Reset	Description
4	GP8PU	R/W	0	GPIO Pull Up Control
3:2	GP8DRV	R/W	0	GPIO Driving Control
1	GP8SMT	R/W	1	GPIO SMT Control
0	GP8IE	R/W	1	GPIO Input Enable

3.3.23 GPIO_CFGCTL5

Address：0x40000114

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP11FUNC					RSVD		GP11 PD	GP11 PU	GP11DRV		GP11 SMT	GP11 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP10FUNC					RSVD		GP10 PD	GP10 PU	GP10DRV		GP10 SMT	GP10 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP11FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP11PD	R/W	0	GPIO Pull Down Control
20	GP11PU	R/W	0	GPIO Pull Up Control
19:18	GP11DRV	R/W	0	GPIO Driving Control
17	GP11SMT	R/W	1	GPIO SMT Control(Useless, IE depend on reg_aon_pad_-ie_smt[2] : 0x4000F014[10])
16	GP11IE	R/W	1	GPIO Input Enable (Useless, IE depend on reg_aon_pad_-ie_smt[2] : 0x4000F014[10])
15:13	RSVD			
12:8	GP10FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP10PD	R/W	0	GPIO Pull Down Control
4	GP10PU	R/W	0	GPIO Pull Up Control
3:2	GP10DRV	R/W	0	GPIO Driving Control
1	GP10SMT	R/W	1	GPIO SMT Control(Useless, IE depend on reg_aon_pad_-ie_smt[1] : 0x4000F014[9])

Bits	Name	Type	Reset	Description
0	GP10IE	R/W	1	GPIO Input Enable (Useless, IE depend on reg_aon_pad_ie_smt[1] : 0x4000F014[9])

3.3.24 GPIO_CFGCTL6

Address：0x40000118

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP13FUNC					RSVD		GP13PD	GP13PU	GP13DRV		GP13SMT	GP13IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP12FUNC					RSVD		GP12PD	GP12PU	GP12DRV		GP12SMT	GP12IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP13FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP13PD	R/W	0	GPIO Pull Down Control
20	GP13PU	R/W	0	GPIO Pull Up Control
19:18	GP13DRV	R/W	0	GPIO Driving Control
17	GP13SMT	R/W	1	GPIO SMT Control(Useless, IE depend on reg_aon_pad_ie_smt[4] : 0x4000F014[12])
16	GP13IE	R/W	1	GPIO Input Enable (Useless, IE depend on reg_aon_pad_ie_smt[4] : 0x4000F014[12])
15:13	RSVD			
12:8	GP12FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP12PD	R/W	0	GPIO Pull Down Control
4	GP12PU	R/W	0	GPIO Pull Up Control
3:2	GP12DRV	R/W	0	GPIO Driving Control
1	GP12SMT	R/W	1	GPIO SMT Control(Useless, IE depend on reg_aon_pad_ie_smt[3] : 0x4000F014[11])
0	GP12IE	R/W	1	GPIO Input Enable (Useless, IE depend on reg_aon_pad_ie_smt[3] : 0x4000F014[11])

3.3.25 GPIO_CFGCTL7

Address：0x4000011c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP15FUNC					RSVD		GP15 PD	GP15 PU	GP15DRV		GP15 SMT	GP15 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP14FUNC					RSVD		GP14 PD	GP14 PU	GP14DRV		GP14 SMT	GP14 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP15FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP15PD	R/W	0	GPIO Pull Down Control
20	GP15PU	R/W	0	GPIO Pull Up Control
19:18	GP15DRV	R/W	0	GPIO Driving Control
17	GP15SMT	R/W	1	GPIO SMT Control
16	GP15IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP14FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP14PD	R/W	0	GPIO Pull Down Control
4	GP14PU	R/W	0	GPIO Pull Up Control
3:2	GP14DRV	R/W	0	GPIO Driving Control
1	GP14SMT	R/W	1	GPIO SMT Control
0	GP14IE	R/W	1	GPIO Input Enable

3.3.26 GPIO_CFGCTL8

Address：0x40000120

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP17FUNC					RSVD		GP17 PD	GP17 PU	GP17DRV		GP17 SMT	GP17 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP16FUNC					RSVD		GP16 PD	GP16 PU	GP16DRV		GP16 SMT	GP16 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP17FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP17PD	R/W	0	GPIO Pull Down Control (Use this bit if reg_en_hw_pu_pd := 0 (0x4000F014[16])
20	GP17PU	R/W	0	GPIO Pull Up Control (Use this bit if reg_en_hw_pu_pd := 0 (0x4000F014[16])
19:18	GP17DRV	R/W	0	GPIO Driving Control
17	GP17SMT	R/W	1	GPIO SMT Control
16	GP17IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP16FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP16PD	R/W	0	GPIO Pull Down Control
4	GP16PU	R/W	0	GPIO Pull Up Control
3:2	GP16DRV	R/W	0	GPIO Driving Control
1	GP16SMT	R/W	1	GPIO SMT Control
0	GP16IE	R/W	1	GPIO Input Enable

3.3.27 GPIO_CFGCTL9

Address：0x40000124

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP19FUNC					RSVD		GP19 PD	GP19 PU	GP19DRV		GP19 SMT	GP19 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP18FUNC					RSVD		GP18 PD	GP18 PU	GP18DRV		GP18 SMT	GP18 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP19FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP19PD	R/W	0	GPIO Pull Down Control
20	GP19PU	R/W	0	GPIO Pull Up Control
19:18	GP19DRV	R/W	0	GPIO Driving Control
17	GP19SMT	R/W	1	GPIO SMT Control
16	GP19IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP18FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP18PD	R/W	0	GPIO Pull Down Control
4	GP18PU	R/W	0	GPIO Pull Up Control
3:2	GP18DRV	R/W	0	GPIO Driving Control
1	GP18SMT	R/W	1	GPIO SMT Control
0	GP18IE	R/W	1	GPIO Input Enable

3.3.28 GPIO_CFGCTL10

Address：0x40000128

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP21FUNC					RSVD		GP21 PD	GP21 PU	GP21DRV		GP21 SMT	GP21 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP20FUNC					RSVD		GP20 PD	GP20 PU	GP20DRV		GP20 SMT	GP20 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP21FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP21PD	R/W	0	GPIO Pull Down Control
20	GP21PU	R/W	0	GPIO Pull Up Control

Bits	Name	Type	Reset	Description
19:18	GP21DRV	R/W	0	GPIO Driving Control
17	GP21SMT	R/W	1	GPIO SMT Control
16	GP21IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP20FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP20PD	R/W	0	GPIO Pull Down Control
4	GP20PU	R/W	0	GPIO Pull Up Control
3:2	GP20DRV	R/W	0	GPIO Driving Control
1	GP20SMT	R/W	1	GPIO SMT Control
0	GP20IE	R/W	1	GPIO Input Enable

3.3.29 GPIO_CFGCTL11

Address：0x4000012c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP23FUNC					RSVD		GP23 PD	GP23 PU	GP23DRV		GP23 SMT	GP23 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP22FUNC					RSVD		GP22 PD	GP22 PU	GP22DRV		GP22 SMT	GP22 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP23FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP23PD	R/W	0	GPIO Pull Down Control
20	GP23PU	R/W	0	GPIO Pull Up Control
19:18	GP23DRV	R/W	0	GPIO Driving Control
17	GP23SMT	R/W	1	GPIO SMT Control
16	GP23IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP22FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)

Bits	Name	Type	Reset	Description
7:6	RSVD			
5	GP22PD	R/W	0	GPIO Pull Down Control
4	GP22PU	R/W	0	GPIO Pull Up Control
3:2	GP22DRV	R/W	0	GPIO Driving Control
1	GP22SMT	R/W	1	GPIO SMT Control
0	GP22IE	R/W	1	GPIO Input Enable

3.3.30 GPIO_CFGCTL12

Address：0x40000130

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP25FUNC					RSVD		GP25 PD	GP25 PU	GP25DRV		GP25 SMT	GP25 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP24FUNC					RSVD		GP24 PD	GP24 PU	GP24DRV		GP24 SMT	GP24 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP25FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP25PD	R/W	0	GPIO Pull Down Control
20	GP25PU	R/W	0	GPIO Pull Up Control
19:18	GP25DRV	R/W	0	GPIO Driving Control
17	GP25SMT	R/W	1	GPIO SMT Control
16	GP25IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP24FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP24PD	R/W	1	GPIO Pull Down Control
4	GP24PU	R/W	0	GPIO Pull Up Control
3:2	GP24DRV	R/W	0	GPIO Driving Control
1	GP24SMT	R/W	1	GPIO SMT Control

Bits	Name	Type	Reset	Description
0	GP24IE	R/W	1	GPIO Input Enable

3.3.31 GPIO_CFGCTL13

Address：0x40000134

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP27FUNC					RSVD		GP27 PD	GP27 PU	GP27DRV		GP27 SMT	GP27 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP26FUNC					RSVD		GP26 PD	GP26 PU	GP26DRV		GP26 SMT	GP26 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP27FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP27PD	R/W	0	GPIO Pull Down Control
20	GP27PU	R/W	0	GPIO Pull Up Control
19:18	GP27DRV	R/W	0	GPIO Driving Control
17	GP27SMT	R/W	1	GPIO SMT Control
16	GP27IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP26FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP26PD	R/W	0	GPIO Pull Down Control
4	GP26PU	R/W	0	GPIO Pull Up Control
3:2	GP26DRV	R/W	0	GPIO Driving Control
1	GP26SMT	R/W	1	GPIO SMT Control
0	GP26IE	R/W	1	GPIO Input Enable

3.3.32 GPIO_CFGCTL14

Address：0x40000138

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP29FUNC					RSVD		GP29 PD	GP29 PU	GP29DRV		GP29 SMT	GP29 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP28FUNC					RSVD		GP28 PD	GP28 PU	GP28DRV		GP28 SMT	GP28 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP29FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP29PD	R/W	0	GPIO Pull Down Control
20	GP29PU	R/W	0	GPIO Pull Up Control
19:18	GP29DRV	R/W	0	GPIO Driving Control
17	GP29SMT	R/W	1	GPIO SMT Control
16	GP29IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP28FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP28PD	R/W	0	GPIO Pull Down Control
4	GP28PU	R/W	0	GPIO Pull Up Control
3:2	GP28DRV	R/W	0	GPIO Driving Control
1	GP28SMT	R/W	1	GPIO SMT Control
0	GP28IE	R/W	1	GPIO Input Enable

3.3.33 GPIO_CFGCTL15

Address：0x4000013c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			GP31FUNC					RSVD		GP31 PD	GP31 PU	GP31DRV		GP31 SMT	GP31 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			GP30FUNC					RSVD		GP30 PD	GP30 PU	GP30DRV		GP30 SMT	GP30 IE

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	GP31FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP31PD	R/W	0	GPIO Pull Down Control
20	GP31PU	R/W	0	GPIO Pull Up Control
19:18	GP31DRV	R/W	0	GPIO Driving Control
17	GP31SMT	R/W	1	GPIO SMT Control
16	GP31IE	R/W	1	GPIO Input Enable
15:13	RSVD			
12:8	GP30FUNC	R/W	5'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP30PD	R/W	0	GPIO Pull Down Control
4	GP30PU	R/W	0	GPIO Pull Up Control
3:2	GP30DRV	R/W	0	GPIO Driving Control
1	GP30SMT	R/W	1	GPIO SMT Control
0	GP30IE	R/W	1	GPIO Input Enable

3.3.34 GPIO_CFGCTL16

Address：0x40000140

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD										GP33 PD	GP33 PU	GP33DRV		GP33 SMT	GP33 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										GP32 PD	GP32 PU	GP32DRV		GP32 SMT	GP32 IE

Bits	Name	Type	Reset	Description
31:22	RSVD			
21	GP33PD	R/W	0	GPIO Pull Down Control
20	GP33PU	R/W	0	GPIO Pull Up Control
19:18	GP33DRV	R/W	0	GPIO Driving Control
17	GP33SMT	R/W	1	GPIO SMT Control

Bits	Name	Type	Reset	Description
16	GP33IE	R/W	1	GPIO Input Enable
15:6	RSVD			
5	GP32PD	R/W	0	GPIO Pull Down Control
4	GP32PU	R/W	0	GPIO Pull Up Control
3:2	GP32DRV	R/W	0	GPIO Driving Control
1	GP32SMT	R/W	1	GPIO SMT Control
0	GP32IE	R/W	1	GPIO Input Enable

3.3.35 GPIO_CFGCTL17

Address：0x40000144

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD										GP35 PD	GP35 PU	GP35DRV		GP35 SMT	GP35 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										GP34 PD	GP34 PU	GP34DRV		GP34 SMT	GP34 IE

Bits	Name	Type	Reset	Description
31:22	RSVD			
21	GP35PD	R/W	0	GPIO Pull Down Control
20	GP35PU	R/W	0	GPIO Pull Up Control
19:18	GP35DRV	R/W	0	GPIO Driving Control
17	GP35SMT	R/W	1	GPIO SMT Control
16	GP35IE	R/W	1	GPIO Input Enable
15:6	RSVD			
5	GP34PD	R/W	0	GPIO Pull Down Control
4	GP34PU	R/W	0	GPIO Pull Up Control
3:2	GP34DRV	R/W	0	GPIO Driving Control
1	GP34SMT	R/W	1	GPIO SMT Control
0	GP34IE	R/W	1	GPIO Input Enable

3.3.36 GPIO_CFGCTL18

Address：0x40000148

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD										GP37 PD	GP37 PU	GP37DRV		GP37 SMT	GP37 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										GP36 PD	GP36 PU	GP36DRV		GP36 SMT	GP36 IE

Bits	Name	Type	Reset	Description
31:22	RSVD			
21	GP37PD	R/W	0	GPIO Pull Down Control
20	GP37PU	R/W	0	GPIO Pull Up Control
19:18	GP37DRV	R/W	0	GPIO Driving Control
17	GP37SMT	R/W	1	GPIO SMT Control
16	GP37IE	R/W	1	GPIO Input Enable
15:6	RSVD			
5	GP36PD	R/W	0	GPIO Pull Down Control
4	GP36PU	R/W	0	GPIO Pull Up Control
3:2	GP36DRV	R/W	0	GPIO Driving Control
1	GP36SMT	R/W	1	GPIO SMT Control
0	GP36IE	R/W	1	GPIO Input Enable

3.3.37 GPIO_CFGCTL30

Address：0x40000180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP31 I	GP30 I	GP29 I	GP28 I	GP27 I	GP26 I	GP25 I	GP24 I	GP23 I	GP22 I	GP21 I	GP20 I	GP19 I	GP18 I	GP17 I	GP16 I
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP15 I	GP14 I	GP13 I	GP12 I	GP11 I	GP10 I	GP9 I	GP8 I	GP7 I	GP6 I	GP5 I	GP4 I	GP3 I	GP2 I	GP1 I	GP0 I

Bits	Name	Type	Reset	Description
31	GP31I	R	0	Register Controlled GPIO Input value

Bits	Name	Type	Reset	Description
30	GP30I	R	0	Register Controlled GPIO Input value
29	GP29I	R	0	Register Controlled GPIO Input value
28	GP28I	R	0	Register Controlled GPIO Input value
27	GP27I	R	0	Register Controlled GPIO Input value
26	GP26I	R	0	Register Controlled GPIO Input value
25	GP25I	R	0	Register Controlled GPIO Input value
24	GP24I	R	0	Register Controlled GPIO Input value
23	GP23I	R	0	Register Controlled GPIO Input value
22	GP22I	R	0	Register Controlled GPIO Input value
21	GP21I	R	0	Register Controlled GPIO Input value
20	GP20I	R	0	Register Controlled GPIO Input value
19	GP19I	R	0	Register Controlled GPIO Input value
18	GP18I	R	0	Register Controlled GPIO Input value
17	GP17I	R	0	Register Controlled GPIO Input value
16	GP16I	R	0	Register Controlled GPIO Input value
15	GP15I	R	0	Register Controlled GPIO Input value
14	GP14I	R	0	Register Controlled GPIO Input value
13	GP13I	R	0	Register Controlled GPIO Input value
12	GP12I	R	0	Register Controlled GPIO Input value
11	GP11I	R	0	Register Controlled GPIO Input value
10	GP10I	R	0	Register Controlled GPIO Input value
9	GP9I	R	0	Register Controlled GPIO Input value
8	GP8I	R	0	Register Controlled GPIO Input value
7	GP7I	R	0	Register Controlled GPIO Input value
6	GP6I	R	0	Register Controlled GPIO Input value
5	GP5I	R	0	Register Controlled GPIO Input value
4	GP4I	R	0	Register Controlled GPIO Input value
3	GP3I	R	0	Register Controlled GPIO Input value
2	GP2I	R	0	Register Controlled GPIO Input value
1	GP1I	R	0	Register Controlled GPIO Input value
0	GP0I	R	0	Register Controlled GPIO Input value

3.3.38 GPIO_CFGCTL32

Address：0x40000188

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP31O	GP30O	GP29O	GP28O	GP27O	GP26O	GP25O	GP24O	GP23O	GP22O	GP21O	GP20O	GP19O	GP18O	GP17O	GP16O
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP15O	GP14O	GP13O	GP12O	GP11O	GP10O	GP9O	GP8O	GP7O	GP6O	GP5O	GP4O	GP3O	GP2O	GP1O	GP0O

Bits	Name	Type	Reset	Description
31	GP31O	R/W	0	Register Controlled GPIO Output Value
30	GP30O	R/W	0	Register Controlled GPIO Output Value
29	GP29O	R/W	0	Register Controlled GPIO Output Value
28	GP28O	R/W	0	Register Controlled GPIO Output Value
27	GP27O	R/W	0	Register Controlled GPIO Output Value
26	GP26O	R/W	0	Register Controlled GPIO Output Value
25	GP25O	R/W	0	Register Controlled GPIO Output Value
24	GP24O	R/W	0	Register Controlled GPIO Output Value
23	GP23O	R/W	0	Register Controlled GPIO Output Value
22	GP22O	R/W	0	Register Controlled GPIO Output Value
21	GP21O	R/W	0	Register Controlled GPIO Output Value
20	GP20O	R/W	0	Register Controlled GPIO Output Value
19	GP19O	R/W	0	Register Controlled GPIO Output Value
18	GP18O	R/W	0	Register Controlled GPIO Output Value
17	GP17O	R/W	0	Register Controlled GPIO Output Value
16	GP16O	R/W	0	Register Controlled GPIO Output Value
15	GP15O	R/W	0	Register Controlled GPIO Output Value
14	GP14O	R/W	0	Register Controlled GPIO Output Value
13	GP13O	R/W	0	Register Controlled GPIO Output Value
12	GP12O	R/W	0	Register Controlled GPIO Output Value
11	GP11O	R/W	0	Register Controlled GPIO Output Value
10	GP10O	R/W	0	Register Controlled GPIO Output Value
9	GP9O	R/W	0	Register Controlled GPIO Output Value
8	GP8O	R/W	0	Register Controlled GPIO Output Value

Bits	Name	Type	Reset	Description
7	GP7O	R/W	0	Register Controlled GPIO Output Value
6	GP6O	R/W	0	Register Controlled GPIO Output Value
5	GP5O	R/W	0	Register Controlled GPIO Output Value
4	GP4O	R/W	0	Register Controlled GPIO Output Value
3	GP3O	R/W	0	Register Controlled GPIO Output Value
2	GP2O	R/W	0	Register Controlled GPIO Output Value
1	GP1O	R/W	0	Register Controlled GPIO Output Value
0	GP0O	R/W	0	Register Controlled GPIO Output Value

3.3.39 GPIO_CFGCTL34

Address：0x40000190

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP31 OE	GP30 OE	GP29 OE	GP28 OE	GP27 OE	GP26 OE	GP25 OE	GP24 OE	GP23 OE	GP22 OE	GP21 OE	GP20 OE	GP19 OE	GP18 OE	GP17 OE	GP16 OE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP15 OE	GP14 OE	GP13 OE	GP12 OE	GP11 OE	GP10 OE	GP9 OE	GP8 OE	GP7 OE	GP6 OE	GP5 OE	GP4 OE	GP3 OE	GP2 OE	GP1 OE	GP0 OE

Bits	Name	Type	Reset	Description
31	GP31OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
30	GP30OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
29	GP29OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
28	GP28OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
27	GP27OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
26	GP26OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
25	GP25OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)

Bits	Name	Type	Reset	Description
24	GP24OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
23	GP23OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
22	GP22OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
21	GP21OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
20	GP20OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
19	GP19OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
18	GP18OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
17	GP17OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
16	GP16OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
15	GP15OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
14	GP14OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
13	GP13OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
12	GP12OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
11	GP11OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
10	GP10OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
9	GP9OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
8	GP8OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
7	GP7OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)

Bits	Name	Type	Reset	Description
6	GP6OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
5	GP5OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
4	GP4OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
3	GP3OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
2	GP2OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
1	GP1OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
0	GP0OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)

3.3.40 GPIO_CFGCTL35

Address：0x40000194

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							GP40 OE	GP39 OE	GP38 OE	GP37 OE	GP36 OE	GP35 OE	GP34 OE	GP33 OE	GP32 OE

Bits	Name	Type	Reset	Description
31:9	RSVD			
8	GP40OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
7	GP39OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
6	GP38OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
5	GP37OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
4	GP36OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)

Bits	Name	Type	Reset	Description
3	GP35OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
2	GP34OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
1	GP33OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
0	GP32OE	R/W	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)

3.3.41 GPIO_INT_MASK1

Address：0x400001a0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPITMS1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPITMS1															

Bits	Name	Type	Reset	Description
31:0	GPITMS1	R/W	32'hFFFFFFFF	reg_gpio_int_mask[31:0]

3.3.42 GPIO_INT_STAT1

Address：0x400001a8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPITSTA1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPITSTA1															

Bits	Name	Type	Reset	Description
31:0	GPITSTA1	R	0	gpio_int_stat[31:0]

3.3.43 GPIO_INT_CLR1

Address：0x400001b0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPITCLR1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPITCLR1															

Bits	Name	Type	Reset	Description
31:0	GPITCLR1	R/W	0	reg_gpio_int_clr[31:0]

3.3.44 GPIO_INT_MODE_SET1

Address：0x400001c0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		GPITMS1													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPITMS1															

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:0	GPITMS1	R/W	0	[2:0] : GPIO 0 Interrupt Mode Control[2:0] [5:3] : GPIO 1 Interrupt Mode Control[2:0] [8:6] : GPIO 2 Interrupt Mode Control[2:0] [11:9] : GPIO 3 Interrupt Mode Control[2:0] [14:12] : GPIO 4 Interrupt Mode Control[2:0] [17:15] : GPIO 5 Interrupt Mode Control[2:0] [20:18] : GPIO 6 Interrupt Mode Control[2:0] [23:21] : GPIO 7 Interrupt Mode Control[2:0] [26:24] : GPIO 8 Interrupt Mode Control[2:0] [29:27] : GPIO 9 Interrupt Mode Control[2:0] GPIO X Interrupt Mode Control [2:0] [2] : 1 : async mode 0 : sync mode [1:0] : 00 : negedge pulse 01 : posedge pulse 10 : negedge level (32k 3T) 11 : posedge level (32k 3T)

3.3.45 GPIO_INT_MODE_SET2

Address：0x400001c4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		GPITMS2													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPITMS2															

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:0	GPITMS2	R/W	0	[2:0] : GPIO 10 Interrupt Mode Control[2:0] [5:3] : GPIO 11 Interrupt Mode Control[2:0] [8:6] : GPIO 12 Interrupt Mode Control[2:0] [11:9] : GPIO 13 Interrupt Mode Control[2:0] [14:12] : GPIO 14 Interrupt Mode Control[2:0] [17:15] : GPIO 15 Interrupt Mode Control[2:0] [20:18] : GPIO 16 Interrupt Mode Control[2:0] [23:21] : GPIO 17 Interrupt Mode Control[2:0] [26:24] : GPIO 18 Interrupt Mode Control[2:0] [29:27] : GPIO 19 Interrupt Mode Control[2:0] GPIO X Interrupt Mode Control [2:0] [2] : 1 : async mode 0 : sync mode [1:0] : 00 : negedge pulse 01 : posedge pulse 10 : negedge level (32k 3T) 11 : posedge level (32k 3T)

3.3.46 GPIO_INT_MODE_SET3

Address：0x400001c8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		GPITMS3													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPITMS3															

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:0	GPITMS3	R/W	0	[2:0] : GPIO 20 Interrupt Mode Control[2:0] [5:3] : GPIO 21 Interrupt Mode Control[2:0] [8:6] : GPIO 22 Interrupt Mode Control[2:0] [11:9] : GPIO 23 Interrupt Mode Control[2:0] [14:12] : GPIO 24 Interrupt Mode Control[2:0] [17:15] : GPIO 25 Interrupt Mode Control[2:0] [20:18] : GPIO 26 Interrupt Mode Control[2:0] [23:21] : GPIO 27 Interrupt Mode Control[2:0] [26:24] : GPIO 28 Interrupt Mode Control[2:0] [29:27] : GPIO 29 Interrupt Mode Control[2:0] GPIO X Interrupt Mode Control [2:0] [2] : 1 : async mode 0 : sync mode [1:0] : 00 : negedge pulse 01 : posedge pulse 10 : negedge level (32k 3T) 11 : posedge level (32k 3T)

3.3.47 GPIO_INT_MODE_SET4

Address：0x400001cc

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										GPITMS4					

Bits	Name	Type	Reset	Description
31:6	RSVD			

Bits	Name	Type	Reset	Description
5:0	GPITMS4	R/W	0	[2:0] : GPIO 30 Interrupt Mode Control[2:0] [5:3] : GPIO 31 Interrupt Mode Control[2:0] [8:6] : GPIO 32 Interrupt Mode Control[2:0] [11:9] : GPIO 33 Interrupt Mode Control[2:0] [14:12] : GPIO 34 Interrupt Mode Control[2:0] [17:15] : GPIO 35 Interrupt Mode Control[2:0] [20:18] : GPIO 36 Interrupt Mode Control[2:0] [23:21] : GPIO 37 Interrupt Mode Control[2:0] [26:24] : GPIO 38 Interrupt Mode Control[2:0] [29:27] : GPIO 39 Interrupt Mode Control[2:0] GPIO X Interrupt Mode Control [2:0] [2] : 1 : async mode 0 : sync mode [1:0] : 00 : negedge pulse 01 : posedge pulse 10 : negedge level (32k 3T) 11 : posedge level (32k 3T)

3.3.48 GPIO_INT2_MASK1

Address：0x400001d0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIT2M1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIT2M1															

Bits	Name	Type	Reset	Description
31:0	GPIT2M1	R/W	32'hFFFFFFFF	reg_gpio_int2_mask[31:0]

3.3.49 GPIO_INT2_STAT1

Address：0x400001d4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIT2ST1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIT2ST1															

Bits	Name	Type	Reset	Description
31:0	GPIT2ST1	R	0	gpio_int2_stat[31:0]

3.3.50 GPIO_INT2_CLR1

Address：0x400001d8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIT2CL1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIT2CL1															

Bits	Name	Type	Reset	Description
31:0	GPIT2CL1	R/W	0	reg_gpio_int2_clr[31:0]

3.3.51 GPIO_INT2_MODE_SET1

Address：0x400001dc

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		GPIT2MS1													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIT2MS1															

Bits	Name	Type	Reset	Description
31:30	RSVD			

Bits	Name	Type	Reset	Description
29:0	GPIT2MS1	R/W	0	[2:0] : GPIO 0 Interrupt Mode Control[2:0] [5:3] : GPIO 1 Interrupt Mode Control[2:0] [8:6] : GPIO 2 Interrupt Mode Control[2:0] [11:9] : GPIO 3 Interrupt Mode Control[2:0] [14:12] : GPIO 4 Interrupt Mode Control[2:0] [17:15] : GPIO 5 Interrupt Mode Control[2:0] [20:18] : GPIO 6 Interrupt Mode Control[2:0] [23:21] : GPIO 7 Interrupt Mode Control[2:0] [26:24] : GPIO 8 Interrupt Mode Control[2:0] [29:27] : GPIO 9 Interrupt Mode Control[2:0] GPIO X Interrupt Mode Control [2:0] [2] : 1 : async mode 0 : sync mode [1:0] : 00 : negedge pulse 01 : posedge pulse 10 : negedge level (32k 3T) 11 : posedge level (32k 3T)

3.3.52 GPIO_INT2_MODE_SET2

Address：0x400001e0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		GPIT2MS2													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIT2MS2															

Bits	Name	Type	Reset	Description
31:30	RSVD			

Bits	Name	Type	Reset	Description
29:0	GPIT2MS2	R/W	0	[2:0] : GPIO 10 Interrupt Mode Control[2:0] [5:3] : GPIO 11 Interrupt Mode Control[2:0] [8:6] : GPIO 12 Interrupt Mode Control[2:0] [11:9] : GPIO 13 Interrupt Mode Control[2:0] [14:12] : GPIO 14 Interrupt Mode Control[2:0] [17:15] : GPIO 15 Interrupt Mode Control[2:0] [20:18] : GPIO 16 Interrupt Mode Control[2:0] [23:21] : GPIO 17 Interrupt Mode Control[2:0] [26:24] : GPIO 18 Interrupt Mode Control[2:0] [29:27] : GPIO 19 Interrupt Mode Control[2:0] GPIO X Interrupt Mode Control [2:0] [2] : 1 : async mode 0 : sync mode [1:0] : 00 : negedge pulse 01 : posedge pulse 10 : negedge level (32k 3T) 11 : posedge level (32k 3T)

3.3.53 GPIO_INT2_MODE_SET3

Address：0x400001e4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		GPIT2MS3													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIT2MS3															

Bits	Name	Type	Reset	Description
31:30	RSVD			

Bits	Name	Type	Reset	Description
29:0	GPIT2MS3	R/W	0	[2:0] : GPIO 20 Interrupt Mode Control[2:0] [5:3] : GPIO 21 Interrupt Mode Control[2:0] [8:6] : GPIO 22 Interrupt Mode Control[2:0] [11:9] : GPIO 23 Interrupt Mode Control[2:0] [14:12] : GPIO 24 Interrupt Mode Control[2:0] [17:15] : GPIO 25 Interrupt Mode Control[2:0] [20:18] : GPIO 26 Interrupt Mode Control[2:0] [23:21] : GPIO 27 Interrupt Mode Control[2:0] [26:24] : GPIO 28 Interrupt Mode Control[2:0] [29:27] : GPIO 29 Interrupt Mode Control[2:0] GPIO X Interrupt Mode Control [2:0] [2] : 1 : async mode 0 : sync mode [1:0] : 00 : negedge pulse 01 : posedge pulse 10 : negedge level (32k 3T) 11 : posedge level (32k 3T)

3.3.54 GPIO_INT2_MODE_SET4

Address：0x400001e8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										GPIT2MS4					

Bits	Name	Type	Reset	Description
31:6	RSVD			

Bits	Name	Type	Reset	Description
5:0	GPIT2MS4	R/W	0	[2:0] : GPIO 30 Interrupt Mode Control[2:0] [5:3] : GPIO 31 Interrupt Mode Control[2:0] [8:6] : GPIO 32 Interrupt Mode Control[2:0] [11:9] : GPIO 33 Interrupt Mode Control[2:0] [14:12] : GPIO 34 Interrupt Mode Control[2:0] [17:15] : GPIO 35 Interrupt Mode Control[2:0] [20:18] : GPIO 36 Interrupt Mode Control[2:0] [23:21] : GPIO 37 Interrupt Mode Control[2:0] [26:24] : GPIO 38 Interrupt Mode Control[2:0] [29:27] : GPIO 39 Interrupt Mode Control[2:0] GPIO X Interrupt Mode Control [2:0] [2] : 1 : async mode 0 : sync mode [1:0] : 00 : negedge pulse 01 : posedge pulse 10 : negedge level (32k 3T) 11 : posedge level (32k 3T)

3.3.55 led_driver

Address：0x40000224

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

Bits	Name	Type	Reset	Description
31	PULDRV	R/W	1'b0	power up ir led drive
30	RSVD			
29:28	LDRVOEN	R/W	2'd3	ir led output enable [1]GPIO23 [0]GPIO22
27:12	RSVD			

Bits	Name	Type	Reset	Description
11:8	IRRXGPSL	R/W	4'h0	0 : disable ir_rx select gpio 1 15 :select gpio 19 33 as ir_rx (GPIO need to set as SWG-PIO mode)
7:4	LDRVIBIA	R/W	4'h8	ir led drive strength, [7:4]*8mA
3:0	RSVD			

3.3.56 usb_xcvr

Address：0x40000228

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD				RCV	VIP	VIM	BD	PU	SUS	SPD	ENUM	RSVD			DACV
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	OEB	OEB REG	OEB SEL	RSVD	ROUTPMOS			RSVD	ROUTNMOS			PU IDO	IDOVFB		

Bits	Name	Type	Reset	Description
31:28	RSVD			
27	RCV	R	1'b0	rx_d
26	VIP	R	1'b0	rx_dp
25	VIM	R	1'b0	rx_dm
24	BD	R	1'b0	bd output
23	PU	R/W	1'b0	set 1 for power up usb transceiver
22	SUS	R/W	1'b0	suspend mode enable set 1 for suspend
21	SPD	R/W	1'b1	full-speed/low-speed mode select must be 1
20	ENUM	R/W	1'b0	0: 1.5k pull up resistor floating; 1: 1.5k pull up resistor connect to D+/D- depends on spd set 1 for enumerate
19:17	RSVD			
16	DACV	RW	1'b0	0: original vip/vim; 1: converted vip/vim Leave it as the default
15	RSVD			
14	OEB	R	1'b1	oeb

Bits	Name	Type	Reset	Description
13	OEBREG	RW	1'b1	Leave it as the default
12	OEBSEL	RW	1'b0	0: usb_oeb; 1: usb_oeb_reg Leave it as the default
11	RSVD			
10:8	ROUTPMOS	R/W	3'h3	Leave it as the default
7	RSVD			
6:4	ROUTNMOS	R/W	3'h3	Leave it as the default
3	PUIDO	R/W	1'b0	power up usb ldo Leave it as the default
2:0	IDOVFB	R/W	3'h3	usb ldo ref voltage Leave it as the default

3.3.57 usb_xcvr_config

Address：0x4000022c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD	SRPR			RSVD	SRPF			RSVD	SRMR			RSVD	SRMF		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	PUTUNE			USE CTRL	STRDRV			USE XCVR	BDVTH			VHYSP		VHYSM	

Bits	Name	Type	Reset	Description
31	RSVD			
30:28	SRPR	R/W	3'h4	D+ rise slew rate control Leave it as the default
27	RSVD			
26:24	SRPF	R/W	3'h3	D+ fall slew rate control Leave it as the default
23	RSVD			
22:20	SRMR	R/W	3'h4	D- rise slew rate control Leave it as the default
19	RSVD			
18:16	SRMF	R/W	3'h3	D- fall slew rate control Leave it as the default

Bits	Name	Type	Reset	Description
15	RSVD			
14:12	PUTUNE	R/W	3'h2	pull up resistor tuning Leave it as the default
11	USECTRL	R/W	1'b1	1: USB XCVR use on-chip usb controller ; 0: USB XCVR usb off-chip usb controller Leave it as the default
10:8	STRDRV	R/W	3'h0	driver strength Leave it as the default
7	USEXCVR	R/W	1'b1	1: USB core use on-chip usb_xcvr; 0: USB core usb off-chip usb_xcvr Leave it as the default
6:4	BDVTH	R/W	3'h1	bd threashold voltage set 7
3:2	VHYSP	R/W	2'b1	D+ input hysteresis select Leave it as the default
1:0	VHYSM	R/W	2'b1	D- input hysteresis select Leave it as the default

4 ADC

4.1 ADC introduction

The chip contains a 12-bit successive approximation analog-to-digital converter (ADC), which supports 12 external analog inputs and several internal analog signal selections.

The ADC works in two modes: single conversion and multi-channel scanning. The conversion result is 12/14/16Bits left-justified mode. The ADC has a depth of 32 FIFOs and supports multiple interrupts and DMA operations. In addition to ordinary analog signal measurement, the ADC can also be used to measure the supply voltage. In addition, the ADC can also be used for temperature detection by measuring the internal/external diode voltage.

4.2 ADC main features

- High performance
  - 12-bit, 14-bit or 16-bit conversion result output
  - ADC maximum working clock is 2MHZ
  - 2.0V,3.2V optional reference voltage
  - DMA support
  - Two working modes: single-channel conversion and multi-channel scanning
  - Two input modes: single-ended and differential
  - Support jitter compensation
  - User can set conversion result offset value
- Analog channels
  - 12 external analog channels
  - 2 DAC internal channels

- 1 VBAT / 2 channel
- 1 TSEN channel

4.3 ADC functional description

The basic block diagram of the ADC is shown below.

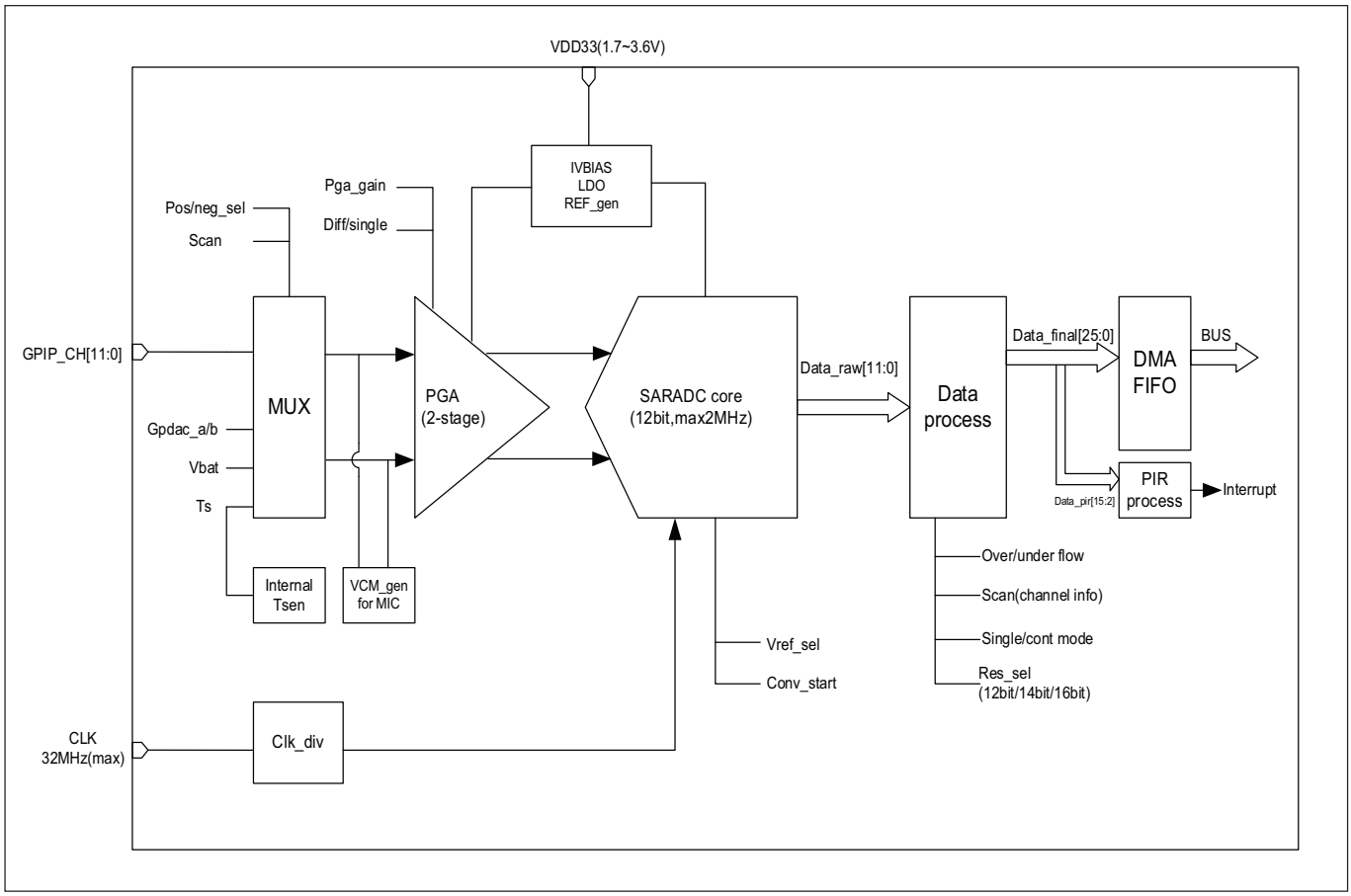

Figure 4.1: ADC block diagram

The ADC consists of five parts: front-end input channel selector, program-controlled amplifier, ADC sampling module, data processing module, and FIFO.

The input channel selector is used to select the channel to be sampled. It contains both external analog signals and internal analog signals. The program-controlled amplifier is used to further process the input signal. It can be set according to the characteristics of the input signal, such as DC and AC. In order to get more accurate conversion values.

The ADC sampling module is the most important function module. It obtains the conversion from analog signals to digital signals through successive comparisons. The conversion result is 12Bit. The data processing module is responsible for further processing the conversion results, including adding channel information. The resulting data is pushed into the FIFO.

4.3.1 ADC pins and internal signals

Table 4.1: ADC internal signals

Internal signals	Signal type	Description
VBAT/2	Input	Voltage signal divided from the power pin
TSEN	Input	Internal temperature sensor output voltage
VREF	Input	Internal analog module reference voltage
DACOUTA	Input	DAC module output
DACOUTB	Input	DAC module output

Table 4.2: ADC external pins

External pins	Signal type	Description
VDDA	Input	Analog power supply and positive reference voltage for the ADC
VSSA	Input	Ground for analog power supply
ADC_CHX	Input	12 analog input channels

4.3.2 ADC channel

The channels that can be selected by the ADC include the input signals of external analog pins and the optional signals inside the chip:

- ADC CH0
- ADC CH1
- ADC CH2
- ADC CH3
- ADC CH4
- ADC CH5
- ADC CH6
- ADC CH7
- ADC CH8

- ADC CH9
- ADC CH10
- ADC CH11
- DAC OUTA
- DAC OUTB
- VBAT/2
- TSEN
- VREF
- GND

It should be noted that if VBAT/2 or TSEN is selected as the input signal to be acquired, gpadc_vbat_en or gpadc_ts_en needs to be set.

The ADC module can support single-ended input or differential input. If it is single-ended input mode, the negative input channel needs to select GND.

4.3.3 ADC clock

The working clock source of the ADC module is shown in the following figure:

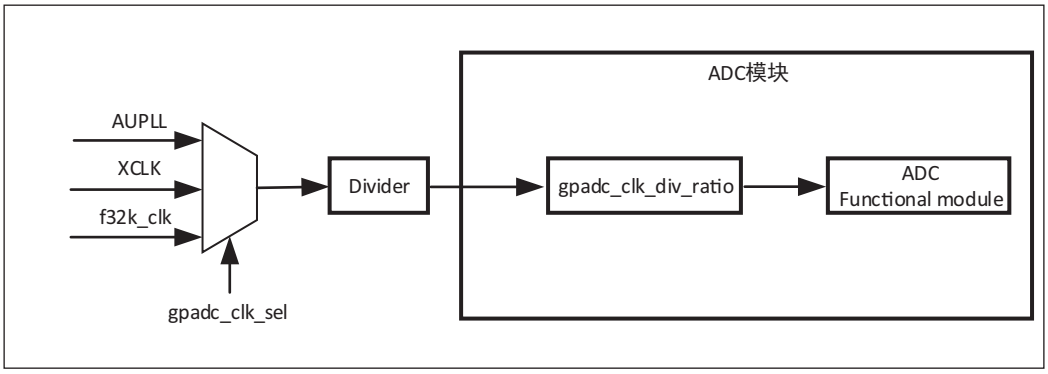

Figure 4.2: ADC Clock

The ADC clock source can select Audio PLL，XTAL or f32k.The clock source selection is set in the GLB module.While selecting, the clock source can be divided by the frequency divider.

In general voltage measurement applications, users can choose XTAL as the clock source. In audio applications, users can use Audio PLL to generate common sampling clocks such as 8KHZ and 44.1KHZ. f32k is a low-frequency clock, which provides a wake-up clock when the MCU sleeps.

Inside the ADC module, a clock divider is provided, which can divide the input clock by 1/4/8/12/16/20/24/32. Users can adjust the ADC clock source and various frequency division coefficients according to actual sampling require-

ments. Note that the maximum input clock of ADC is 2.048MHZ.

The width of the gpadc_32m_clk_div divider register is 6Bits, the maximum divider is 64, and the divider formula is fout=fsource/(gpadc_32m_clk_div+1).

The gpadc_clk_div_ratio divider register is located inside the ADC module, with a width of 3Bits, and its divider value is defined as follows:

```
ADC_CLK_DIV_1,                  /*!< ADC clock:on 32M clock is 32M */
ADC_CLK_DIV_4,                  /*!< ADC clock:on 32M clock is 8M */
ADC_CLK_DIV_8,                  /*!< ADC clock:on 32M clock is 4M */
ADC_CLK_DIV_12,                 /*!< ADC clock:on 32M clock is 2.666M */
ADC_CLK_DIV_16,                 /*!< ADC clock:on 32M clock is 2M */
ADC_CLK_DIV_20,                 /*!< ADC clock:on 32M clock is 1.6M */
ADC_CLK_DIV_24,                 /*!< ADC clock:on 32M clock is 1.333M */
ADC_CLK_DIV_32,                 /*!< ADC clock:on 32M clock is 1M */
```

If the user wants to adjust the ADC input clock, there are four ways.

1. Switch the clock source, XTAL defaults to 32MHZ, Audio PLL (can be configured to 11.288MHZ or 11.2896MHZ).
2. Use a frequency divider with a length of 6BITS in the clock module.
3. Using the frequency divider in the ADC module, the optional frequency division is 1/4/8/12/16/20/24/32.
4. By configuring the gpadc_res_sel register, change the value of OSR to achieve the frequency effect. If OSR=256, the actual equivalent ADC input clock is divided by 256.

Assuming that the clock source selection is Audio PLL=11.2896MHZ, the GLB frequency division selection configuration is 1, the ADC internal frequency divider is selected ADC_CLK_DIV_4, OSR=128, then the final clock output is fout = 11289600 / (1 + 1) / 4/128 = 11025HZ

4.3.4 ADC conversion mode

The ADC supports two conversion modes: single-channel conversion mode and scan mode.

In single-channel conversion mode, the user needs to select the positive input channel through gpadc_pos_sel, select the negative input channel through gpadc_neg_sel, and set the gpadc_cont_conv_en control bit to 0, which means single-channel conversion, and then set the gpadc_conv_start control bit to start the conversion.

In scan conversion mode, the gpadc_cont_conv_en control bit needs to be set to 1, and the number of conversion channels set by the ADC according to the gpadc_scan_length control bit. According to the channel order set by the gpadc_reg_scn_posX (X = 1, 2) and gpadc_reg_scn_negX (X = 1, 2) registers, the conversion is performed one by one, and the result of the conversion is automatically pushed into the ADC FIFO. The channels set by the gpadc_reg_scn_posX (X = 1, 2) and gpadc_reg_scn_negX (X = 1, 2) registers can be the same, which means that users can implement multiple sampling conversions on a channel.

ADC conversion results are generally placed in the FIFO. The ADC module does not provide conversion completion interrupts. Users need to set the FIFO receive data threshold interrupt based on the actual number of conversion channels. The FIFO threshold interrupt is used as the ADC conversion completion interrupt.

4.3.5 ADC consequence

The gpadc_raw_data register stores the raw result of the ADC. In single-ended mode, the data valid bit is 12Bits, unsigned bit. In differential mode, the highest bit is the sign bit. The remaining 11Bits represent the result of the conversion.

The gpadc_data_out register stores the ADC result. This result contains the ADC result, sign bit and channel information. The data format is as follows:

Table 4.3: ADC conversion result format

BitS	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mean	Positive channel number					Negative channel number					Conversion result															

Bit21-Bit25 of the conversion result is the positive channel number, Bit16-Bit20 is the negative channel number, and Bit0-Bit15 is the converted value.

The gpadc_res_sel control bit can set the number of bits of the conversion result, which are 12 bits, 14 bits, and 16 bits, respectively. Among them, 14 bits and 16 bits are the results obtained by multiple sampling to improve the accuracy.

The values that can be set are as follows:

- 3'b000 12bit 2MS/s, OSR=1
- 3'b001 14bit 125kS/s, OSR=16
- 3'b010 14bit 31.25kS/s, OSR=64
- 3'b011 16bit 15.625KS/s, OSR=128
- 3'b100 16bit 7.8125KS/s, OSR=256

The ADC conversion result is left-justified. When 12 bits are selected, Bit15-Bit4 of the conversion result is valid. When 14 bits are selected, Bit15-Bit2 of the conversion result is valid. When 16 bits are selected, Bit15-Bit0 of the conversion result is valid.

Similarly, in the differential mode, the highest is the sign, that is, when 14 bits are selected, Bit15 is the sign bit, Bit14-Bit2 is the conversion result, and Bit14 is the MSB.

In single-ended mode, there is no sign bit, that is, when 12 bits are selected, Bit15-Bit4 is the conversion result and Bit15 is the MSB.

In actual use, the results of the ADC are generally placed in the FIFO, which is particularly important in the multi-channel scan mode. Therefore, users generally obtain conversion results from the ADC FIFO. The data format of the ADC FIFO is the same in the gpadc_data_out register.

4.3.6 ADC interrupt

The ADC module can generate interrupts when the positive sampling is saturated and the negative sampling is saturated. The respective interrupts can be masked by gpadc_pos_satur_mask, gpadc_neg_satur_mask.

When the interrupt is generated, the interrupt status can be queried by the gpadc_pos_satur, and gpadc_neg_satur registers, and the interrupt can be cleared by gpadc_pos_satur_clr and gpadc_neg_satur_clr. This function can be used to determine whether the input voltage is abnormal.

4.3.7 ADC FIFO

The ADC module has a FIFO with a depth of 32 and a data width of 26Bits. After the ADC completes the conversion, it will automatically push the result into the FIFO. The ADC's FIFO has the following status and interrupt management functions:

- FIFO Overrun interrupt
- FIFO Underrun interrupt
- FIFO threshold interrupt

When the FIFO is full, but the user does not read the value through DMA or direct access to the register, and data enters the FIFO again, the module will generate a FIFO Overrun interrupt at this time.

When the FIFO is empty, but the user still requests data from the FIFO, the module will generate a FIFO Underrun interrupt at this time.

The user can configure the FIFO threshold register gpadc_fifo_thl, select the threshold for FIFO to generate interrupts, and choose 1, 4, 8, and 16. If the number of ADC FIFOs reaches the set threshold, a threshold interrupt will be generated.

When an interrupt occurs, the interrupt flag can be cleared by the corresponding clear bit.

Using the ADC's FIFO, users can implement three modes of data acquisition: query mode, interrupt mode, and DMA mode.

Query mode

The CPU polls the length of the ADC FIFO. When the length of the FIFO is not empty, it indicates that there are valid data in the FIFO, and the CPU can read these data from the FIFO.

Interrupt mode

Using the threshold interrupt of the FIFO, when the interrupt is generated and the number of ADC data reaches the threshold, the CPU can read the length of the ADC FIFO in the interrupt service function and read it all.

DMA mode

The user sets the dmaen control bit, which can cooperate with the DMA to complete the transfer of the converted data to the memory. When using the DMA mode, set the threshold of the number of data sent by the ADC FIFO through the fifothl. When the DMA receives the request, it will automatically follow the user settings. The specified parameters transfer the specified number of results from the FIFO to the corresponding memory.

4.3.8 ADC configuration process

Setting the ADC clock

According to the ADC conversion speed requirements, determine the working clock of the ADC, set the ADC clock source and frequency division of the GLB module, and combine with clkdvrt to determine the final working module's clock frequency.

Set GPIO according to the channel used

According to the analog pin used, determine the channel number used, initialize the corresponding GPIO as an analog function. It should be noted that when setting the GPIO as an analog input, do not set the GPIO pull-up or pull-down, you need to set it to float.

Set the channel to be converted

Set the corresponding channel register according to the analog channel and conversion mode used.

For single-channel conversion, set the converted channel information in the possel and negsel registers.

For multi-channel scanning mode, set sclen, scpX and scnX according to the number of scanning channels and scanning order.

Set the data reading method

According to the way of reading data introduced by ADC FIFO, select the mode to use and set the corresponding register. If you use DMA, you also need to configure a channel of DMA to cooperate with the ADC FIFO to complete the data transfer.

Start conversion

Finally set ressel to select the precision of the data conversion result. Finally set gben = 1 and cvst = 1 to start the ADC to start conversion.

When the conversion is complete and needs to be converted again, cvst needs to be set to 0 and then set to 1 in order to trigger the conversion again.

4.3.9 VBAT measurement

The VBAT/2 measurement is the voltage of the chip VDD33, not the voltage of an external battery such as a lithium battery. If you need to measure the voltage of a power supply head such as a lithium battery, you can divide the voltage and then input it to the ADC's GPIO analog channel. Measuring the voltage of VDD33 can reduce the use of

GPIO.

The VBAT/2 voltage measured by the ADC module is after a partial pressure.The actual input voltage to the ADC module is half of VDD33, that is, VBAT/2 = VDD33/2. Because the voltage is divided, in order to obtain higher accuracy, it is recommended that the reference voltage of the ADC is 2V, single-ended mode is used, the positive input voltage is VBAT/2, the negative input voltage is GND, and vbaten is set to 1 to start.

After conversion, multiply the corresponding conversion result by 2 to get the VDD33 voltage.

4.3.10 TSEN measurement

The ADC can measure the internal diode or external diode voltage value, and the voltage difference between the diode and temperature is related, so by measuring the voltage of the diode, the ambient temperature can be calculated. We call it Temperature Sensor, referred to as TSEN.

The test principle of TSEN is to generate a fitted curve by measuring the voltage difference ΔV generated by two different currents on a diode with temperature.

Regardless of the measurement of the external or internal diode, the final output value is related to temperature, which can be expressed as Δ(ADC_out) = 7.753T + X. When we know the voltage value, we also know the temperature T. Here X is an offset value that can be used as a standard value. Before actual use, we need to determine X. The chip manufacturer will measure Δ(ADC_out) at a standard temperature, such as 25 degrees at room temperature, before the chip leaves the factory to get X.

When the user uses it, as long as the formula T = [Δ(ADC_out) -X]/7.753, the temperature T can be obtained.

When using TSEN, it is recommended to set the ADC to 16Bits mode, reduce the error by multiple sampling, and select 2V as the reference voltage to improve accuracy. Set tsen to 1 to enable the TSEN function. If the internal diode is selected, tsxten = 0. External diode, tsxten = 1, select the forward input channel according to the actual situation.

If it is an internal diode, select the TSEN channel. If it is external, select the corresponding analog GPIO channel. Select the negative input terminal as GND. After the above settings are completed, set tsdc = 0 to start the measurement and get the measurement result V0, then set tsdc = 1 to start the measurement and get the measurement result V1, Δ(ADC_out) = V1-V0, according to the formula T = [Δ(ADC_out) -X] /7.753 to obtain the temperature T.

4.4 Register description

Name	Description
gpadc_config	GPADC configuration
gpadc_dma_rdata	GPADC DMA read data
gpadc_reg_cmd	GPADC register configuration 0
gpadc_reg_config1	GPADC register configuration 1

Name	Description
gpadc_reg_config2	GPADC register configuration 2
gpadc_reg_scn_pos1	adc converation sequence 1
gpadc_reg_scn_pos2	adc converation sequence 2
gpadc_reg_scn_neg1	adc converation sequence 3
gpadc_reg_scn_neg2	adc converation sequence 4
gpadc_reg_status	GPADC register status
gpadc_reg_isr	GPADC register operation
gpadc_reg_raw_result	GPADC register raw result
gpadc_reg_define	GPADC register define

4.4.1 gpadc_config

Address：0x40002000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD								FIFOTHL		FIFODACN					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFO RDYM	FURM	FORM	RDYM	RSVD	URCL	ORCL	RDY CLR	FIFO RDY	FIFO UR	FIFO OR	RDY	FIFO FULL	FIFO NE	FIFO CLR	DMA EN

Bits	Name	Type	Reset	Description
31:24	RSVD			
23:22	FIFOTHL	R/W	2'd0	fifo threshold 2'b00: 1 data 2'b01: 4 data 2'b10: 8 data 2'b11: 16 data
21:16	FIFODACN	R	6'd0	fifo data number
15	FIFORDYM	R/W	1'b1	write 1 mask
14	FURM	R/W	1'b0	write 1 mask
13	FORM	R/W	1'b0	write 1 mask
12	RDYM	R/W	1'b0	write 1 mask
11	RSVD			

Bits	Name	Type	Reset	Description
10	URCL	W1C	1'b0	Write 1 to clear flag
9	ORCL	W1C	1'b0	Write 1 to clear flag
8	RDYCLR	W1C	1'b0	Write 1 to clear flag
7	FIFORDY	R	1'b0	FIFO ready interrupt flag
6	FIFOUR	R	1'b0	FIFO underrun interrupt flag
5	FIFOOR	R	1'b0	FIFO overrun interrupt flag
4	RDY	R	1'b0	Conversion data ready interrupt flag
3	FIFOFULL	R	1'b0	FIFO full flag
2	FIFONE	R	1'b0	FIFO not empty flag
1	FIFOCLR	W1C	1'b0	FIFO clear signal
0	DMAEN	R/W	1'b0	GPADC DMA enbale

4.4.2 gpadc_dma_rdata

Address：0x40002004

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

Bits	Name	Type	Reset	Description
31:26	RSVD			
25:0	DMARDA	R	26'd0	GPADC finial conversion result stored in the FIFO

4.4.3 gpadc_reg_cmd

Address：0x4000f90c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD	SENTEN	SENSEL		CSENPU	RSVD			MB32EN	MP2G		M1DF	M2DF	DWAEN	RSVD	MBBP
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPEN	MBEN	NGGR	POSSEL					NEGSEL					SFRS	CVST	GBEN

Bits	Name	Type	Reset	Description
31	RSVD			
30	SENTEN	R/W	1'b0	enable sensor dc test mux
29:28	SENSEL	R/W	2'h0	selected output current channel and measurement channel 2'h0: 1st channel 2'h1: 2nd channel 2'h2: 3rd channel 2'h3: 4th channel
27	CSENPU	R/W	1'b0	enable chip sensor test 1'b0: disable 1'b1: enable
26:24	RSVD			
23	MB32EN	R/W	1'b0	micboost 32db enable 1'b0: 16dB 1'b1: 32dB
22:21	MP2G	R/W	2'h0	mic_pga2_gain 2'h0: 0dB 2'h1: 6dB 2'h2: -6dB 2'h3: 12dB
20	M1DF	R/W	1'b0	mic1 diff enable 1'b0: single 1'b1: diff
19	M2DF	R/W	1'b0	mic2 diff enable 1'b0: single 1'b1: diff
18	DWAEN	R/W	1'b0	dwa enable 1'b0: dwa disable 1'b1: dwa enable
17	RSVD			
16	MBBP	R/W	1'b0	micboost amp bypass 1'b0: not bypass 1'b1: bypass
15	MPEN	R/W	1'b0	micpga enable 1'b0: micpga disable 1'b1: miapga enable
14	MBEN	R/W	1'b0	enable micbias 1'b0: micbias power down 1'b1: miabias power on

Bits	Name	Type	Reset	Description
13	NGGR	R/W	1'b0	set negative input of adc to ground 1'b0: disable 1'b1: enable
12:8	POSSEL	R/W	5'hf	select adc positive input in none-scan mode 5'h0 gpip_ch[0] 5'h1 gpip_ch[1] 5'h2 gpip_ch[2] 5'h3 gpip_ch[3] 5'h4 gpip_ch[4] 5'h5 gpip_ch[5] 5'h6 gpip_ch[6] 5'h7 gpip_ch[7] 5'h8 gpip_ch[8] 5'h9 gpip_ch[9] 5'h10 gpip_ch[10] 5'h11 gpip_ch[11] 5'h12 daca 5'h13 dacb 5'h14 temp_p 5'h16 vref 5'h18 vbat/2 5'h23-31 avss
7:3	NEGSEL	R/W	5'hf	select adc negative input in none-scan mode 5'h0 gpip_ch[0] 5'h1 gpip_ch[1] 5'h2 gpip_ch[2] 5 'h3 gpip_ch[3] 5'h4 gpip_ch[4] 5'h5 gpip_ch[5] 5'h6 gpip_ch[6] 5'h7 gpip_ch[7] 5'h8 gpip_ch[8] 5'h9 gpip_ch[9] 5'h10 gpip_ch[10] 5'h11 gpip_ch[11] 5'h12 daca 5'h13 dacb 5'h14 temp_p 5'h16 vref 5'h18 vbat/2 5'h23-31 avss

Bits	Name	Type	Reset	Description
2	SFRS	R/W	1'b0	user reset the whole block 1'h0: not reset 1'h1: reset
1	CVST	R/W	1'b0	1'h0: stop converation 1'h1: start converation
0	GBEN	R/W	1'b0	1'h0: disable ADC 1'h1: enable ADC

4.4.4 gpadc_reg_config1

Address：0x4000f910

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD	V18SEL		V11SEL		DTEN	SCEN	SCLEN				CLKDVRT			ALCLK INV	RSVD
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					LDEN	HYST SEL	SEL EN	RSVD			RESSEL			CTCV EN	CAL OSEN

Bits	Name	Type	Reset	Description
31	RSVD			
30:29	V18SEL	R/W	2'h0	internal vdd18 select
28:27	V11SEL	R/W	2'h0	internal vdd11 select
26	DTEN	R/W	1'h0	Dither compensation enable
25	SCEN	R/W	1'h0	select scan mode enable: 0: select gpadc_pos/neg_sel;1: select : select gpadc_scan_pos_x and gpadc_scan_neg_x

Bits	Name	Type	Reset	Description
24:21	SCLEN	R/W	4'h0	select scan mode length 4'b0000 : select gpadc_scan_pos_0 and gpadc_scan_-neg_0 4'b0001 : select gpadc_scan_pos_1 and gpadc_scan_-neg_1 4'b0010 : select gpadc_scan_pos_2 and gpadc_scan_-neg_2 4'b0011 : select gpadc_scan_pos_3 and gpadc_scan_-neg_3 4'b0100 : select gpadc_scan_pos_4 and gpadc_scan_-neg_4 4'b0101 : select gpadc_scan_pos_5 and gpadc_scan_-neg_5 4'b0110 : select gpadc_scan_pos_6 and gpadc_scan_-neg_6 4'b0111 : select gpadc_scan_pos_7 and gpadc_scan_-neg_7 4'b1000 : select gpadc_scan_pos_8 and gpadc_scan_-neg_8 4'b1001 : select gpadc_scan_pos_9 and gpadc_scan_-neg_9 4'b1010 : select gpadc_scan_pos_10 and gpadc_scan_-neg_10 4'b1011 : select gpadc_scan_pos_11 and gpadc_scan_-neg_11
20:18	CLKDVRT	R/W	3'h3	analog 32M clock division ratio 3'b000: div=1 3'b001: div=4 3'b010: div=8 3'b011: div=12 3'b100: div=16 3'b101: div=20 3'b110: div=24 3'b111: div=32
17	ALCLKINV	R/W	1'b0	analog clock 2M inverted
16:11	RSVD			
10	LDEN	R/W	1'b0	Low power supply detected enable
9	HYSTSEL	R/W	1'b0	pga vcm hystersis select when vcm_sel_en is enabled
8	SELEN	R/W	1'b0	pga vcm selected when lowv_det_en is enable
7:5	RSVD			

Bits	Name	Type	Reset	Description
4:2	RESSEL	R/W	3'h0	adc resolution/over-sample rate select 3'b000 12bit 2MS/s, OSR=1 3'b001 14bit 125kS/s, OSR=16 3'b010 14bit 31.25kS/s, OSR=64 3'b011 16bit 15.625KS/s, OSR=128 (voice mode16KS/s) 3'b100 16bit 7.8125KS/s, OSR=256 (voice mode 8KS/s)
1	CTCVEN	R/W	1'b1	To enable continuous conversion 1'h0: one shot conversion 1'h1: continuous conversion
0	CALOSEN	R/W	1'b0	offset calibration enable

4.4.5 gpadc_reg_config2

Address：0x4000f914

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSDC	CVSP			PGA1GAIN			PGA2GAIN			TESTSEL			TEST EN	BSEL	CHOM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHOM	PVB EN	PGA EN	POSCAL				PGAVCM		TSEN	TSXT EN	VBAT EN	VREF SEL	DFMD	RSVD	

Bits	Name	Type	Reset	Description
31	TSDC	R/W	1'b0	tsen diode current
30:28	CVSP	R/W	3'h0	adc conversion speed
27:25	PGA1GAIN	R/W	3'h0	3'h0: disable 3'h1: gain=1 3'h2: gain=2 3'h3: gain=4 3'h4: gain=8 3'h5: gain=16 3'h6: gain=32 3'h7: gain=32

Bits	Name	Type	Reset	Description
24:22	PGA2GAIN	R/W	3'h0	3'h0: disable 3'h1: gain=1 3'h2: gain=2 3'h3: gain=4 3'h4: gain=8 3'h5: gain=16 3'h6: gain=32 3'h7: gain=32
21:19	TESTSEL	R/W	3'h0	select test point 0 7
18	TESTEN	R/W	1'b0	Analog test enable.
17	BSEL	R/W	1'b0	adc analog portion low power mode select 1'h0: bandgap system 1'h1:aon bandgap
16:15	CHOM	R/W	2'h3	2'b11 all off 2'b11 Vref AZ on 2'b11 Vref AZ and PGA chop on 2'b11 Vref AZ and PGA chop+RPC on
14	PVBEN	R/W	1'b0	enable pga input vcm bias
13	PGAEN	R/W	1'b0	1'h0: disable PGA 1'h1 enable PGA
12:9	POSCAL	R/W	4'h8	pga offset calibration
8:7	PGAVCM	R/W	2'h2	Audio PGA output common mode control 2'b00: cm=1.3V 2'b11: cm=1.4V 2'b11: cm=1.5V 2'b11: cm=1.6V
6	TSEN	R/W	1'b0	1'h0: disable temperature sensor 1'h1: enable temperature sensor
5	TSXTEN	R/W	1'b0	1'h0: internal diode mode 1'h1: external diode mode
4	VBATEN	R/W	1'b0	1'h0: disable VBAT sensor 1'h1 enable VBAT sensor
3	VREFSEL	R/W	1'b0	ADC reference select 1'h0 3.2V 1'h1 2.0V
2	DFMD	R/W	1'b0	1'h0 single-ended 1'h1 differential
1:0	RSVD			

4.4.6 gpadc_reg_scn_pos1

Address：0x4000f918

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SCP5						SCP4				SCP3			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCP3		SCP2				SCP1					SCP0				

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:25	SCP5	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
24:20	SCP4	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
19:15	SCP3	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
14:10	SCP2	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
9:5	SCP1	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
4:0	SCP0	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel

4.4.7 gpadc_reg_scn_pos2

Address：0x4000f91c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SCP11						SCP10				SCP9			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCP9		SCP8				SCP7					SCP6				

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:25	SCP11	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
24:20	SCP10	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
19:15	SCP9	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
14:10	SCP8	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
9:5	SCP7	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
4:0	SCP6	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel

4.4.8 gpadc_reg_scn_neg1

Address：0x4000f920

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SCN5					SCN4					SCN3			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCN3		SCN2					SCN1					SCN0			

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:25	SCN5	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
24:20	SCN4	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
19:15	SCN3	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
14:10	SCN2	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
9:5	SCN1	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
4:0	SCN0	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel

4.4.9 gpadc_reg_scn_neg2

Address：0x4000f924

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SCN11					SCN10					SCN9			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCN9		SCN8					SCN7					SCN6			

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:25	SCN11	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
24:20	SCN10	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
19:15	SCN9	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
14:10	SCN8	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
9:5	SCN7	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
4:0	SCN6	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel

4.4.10 gpadc_reg_status

Address：0x4000f928

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

Bits	Name	Type	Reset	Description
31:1	RSVD			
0	DARD	R	1'b0	ADC final conversion data ready

4.4.11 gpadc_reg_isr

Address：0x4000f92c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						PSM	NSM	RSVD		PSC	NSC	RSVD		PS	NS

Bits	Name	Type	Reset	Description
31:10	RSVD			
9	PSM	R/W	1'h0	write 1 mask
8	NSM	R/W	1'h0	write 1 mask
7:6	RSVD			
5	PSC	R/W	1'b0	Write 1 to clear flag
4	NSC	R/W	1'b0	Write 1 to clear flag
3:2	RSVD			
1	PS	R	1'b0	ADC data positive side saturation interrupt flag
0	NS	R	1'b0	ADC data negative side saturation interrupt flag

4.4.12 gpadc_reg_raw_result

Address：0x4000f934

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

Bits	Name	Type	Reset	Description
31:12	RSVD			
11:0	RAWDA	R	12'h0	ADC Raw data

4.4.13 gpadc_reg_define

Address：0x4000f938

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	OSCALDA	R/W	16'h0	User defined or self calculated offset data 16-bit signed

5 DAC

5.1 DAC introduction

The chip has a built-in 10Bits digital-to-analog converter (DAC) with a FIFO depth of 1, and supports 2 DAC modulation outputs. Can be used for audio playback, conventional analog signal modulation.

5.2 DAC main feature

- DAC modulation accuracy is 10-Bits
- DAC input clock can be selected as 32M or Audio PLL
- Support DMA to transfer memory to DAC modulation register
- Support dual channel playback DMA transport mode
- The output pin of DAC is fixed to ChannelA as GPIO11, ChannelB as GPIO17
- DAC reference voltage can be selected internally

5.3 DAC function description

The basic block diagram of the DAC module is shown in the figure.

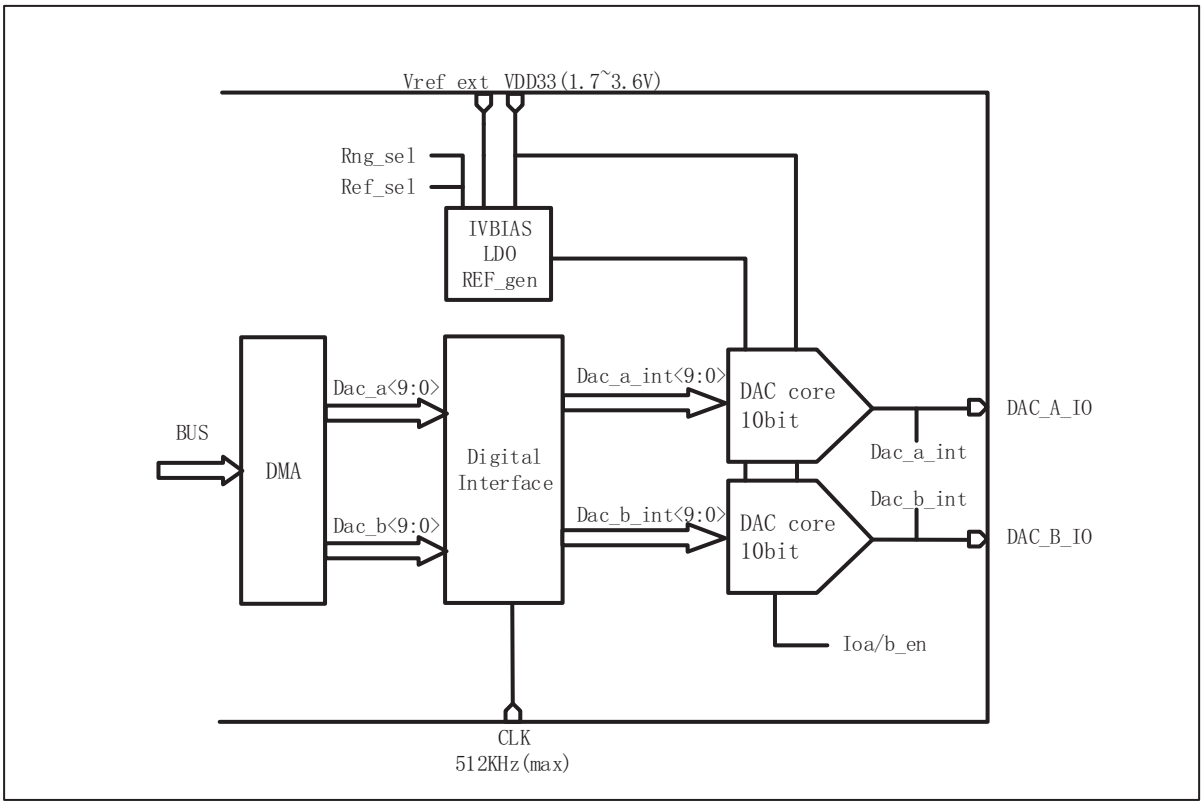

Figure 5.1: DAC basic block diagram

The DAC module contains two DAC modulation circuits and a power supply circuit related to modulating analog signals. The user can use Ref_Sel to select whether the DAC reference voltage is external/internal, and Ref_Rng to select the internal reference voltage source.

The modulation data of the DAC can be directly written into the DAC modulation register (GLB_GPDAC_A_DATA, GLB_GPDAC_B_DATA in 0x40000314) by the CPU, or it can be transferred to the gpdac_dma_wdata (0x40002048) register by the DMA.

DAC data writing method

The CPU directly writes the GLB_GPDAC_A_DATA, GLB_GPDAC_B_DATA registers to complete the modulation, or uses DMA to transfer the data that needs to be modulated to gpdac_dma_wdata.

DMA handling mode

gpdac_dma_wdata (0x40002048) is a 32BITS register. The default meaning is that the 32BITS values are all modulated on the ChannelA pin in order. It can also be configured as the high 16 bits which correspond to the analog voltage output of Channel B by default, and the low 16 bits correspond to the analog voltage output of Channel A.

Note that whether it is 32/16-bit modulation, only the lower 10 bits are valid, because the maximum modulation accuracy of the DAC is 10BITS. The user can modify the meaning of the high and low bytes transported by the DMA by configuring the gpdac_dma_format register.

If gpdac_dma_format is 0, the data transferred by DMA into gpdac_dma_wdata are all modulated in Channel A in

sequence, and the modulation order is {A0},{A1},{A2},...

If gpdac_dma_format is 1, the high 16 bits of the data transferred into gpdac_dma_wdata by DMA are modulated in Channel B, and the low 16 bits are modulated in Channel A. The modulation sequence is {B0,A0},{B1,A1},{B2,A2},.... This feature is very useful in stereo playback.

If gpdac_dma_format is 2, the data transferred by DMA into gpdac_dma_wdata is all modulated in Channel A, but the order of modulation is {A1,A0},{A3,A2},{A5,A4},...

DAC external reference voltage selection

The user can select external reference voltage or internal reference voltage by configuring gpdac_ref_sel (0x40000308[8]).

If the internal reference voltage is selected, the configuration is shown in the following table.

Table 5.1: Internal reference voltage

gpdac_a_rng	gpdac_ref_sel	Output range
00	0	0.2-1
01/10	0	0.225-1.425
11	0	0.2-1.8

If you choose an external reference voltage, please connect the external voltage to the fixed GPIO7.

5.4 Register description

Name	Description
gpdac_config	GPDAC configuration
gpdac_dma_config	GPDAC dma configuration
gpdac_dma_wdata	GPDAC dma write data
gpdac_ctrl	GPDAC control
gpdac_actrl	GPDAC channelA control
gpdac_bctrl	GPDAC channelB control
gpdac_data	GPDAC data

5.4.1 gpdac_config

Address：0x40002040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD								CHBSEL				CHASEL			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					MODE			RSVD		DSMMODE		RSVD		EN2	EN

Bits	Name	Type	Reset	Description
31:24	RSVD			
23:20	CHBSEL	R/W	0	Channel B Source Select 0: Reg 1: DMA 2: DMA + Filter 3: Sin Gen 4: A (The same as channel A) 5: A (Inverse of channel A)
19:16	CHASEL	R/W	0	Channel A Source Select 0: Reg 1: DMA 2: DMA + Filter 3: Sin Gen
15:11	RSVD			
10:8	MODE	R/W	0	0:32k, 1:16k, 3:8k, 4:512k(for DMA only)
7:6	RSVD			
5:4	DSMMODE	R/W	0	0:bypass, 1:dsm order=1, 2: dsm order=2
3:2	RSVD			
1	EN2	R/W	0	GPDAC enable 2 (for B channel)
0	EN	R/W	0	GPDAC enable

5.4.2 gpdac_dma_config

Address：0x40002044

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

Bits	Name	Type	Reset	Description
31:6	RSVD			
5:4	DMAFM	R/W	0	DMA TX format (Data 12-bit) 0: A0, A1, A2… 1: B0,A0, B1,A1, B2,A2… 2: A1,A0, A3,A2, A5,A4… (Note: 20'h0,[11:0] or 4'h0,[27:16],4'h0,[11:0])
3:1	RSVD			
0	DMATXEN	R/W	0	GPDAC DMA TX enable

5.4.3 gpdac_dma_wdata

Address：0x40002048

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

Bits	Name	Type	Reset	Description
31:0	DMAWDA	W	x	GPDAC DMA TX data

5.4.4 gpdac_ctrl

Address：0x40000308

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							REF SEL	TSEN	RSVD					BRS ANA	ARS ANA

Bits	Name	Type	Reset	Description
31:9	RSVD			
8	REFSEL	R/W	1'h0	Reference select 1'h0 Internal reference 1'h1 External reference

Bits	Name	Type	Reset	Description
7	TSEN	R/W	1'h0	Test enable 1'h0 analog test disabled (ATEST is set in Hi-Z state) 1'h1 analog test point enabled to ATEST
6:2	RSVD			
1	BRSANA	R/W	1'h1	Soft reset for DAC channel B, active low
0	ARSANA	R/W	1'h1	Soft reset for DAC channel A, active low

5.4.5 gpdac_actrl

Address：0x4000030c

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

Bits	Name	Type	Reset	Description
31:23	RSVD			
22:20	AOMUX	R/W	3'h0	
19:18	ARNG	R/W	2'h3	Output voltage range control with internal/external reference
17:2	RSVD			
1	IOAEN	R/W	1'h0	Channel A conversion output to pad enable 1'h0 Disable channel A conversion result to GPIO 1'h1 Enable channel A conversion result to GPIO
0	AEN	R/W	1'h0	Channel A enable/disable signal 1'h0 Disable channel A conversion. 1'h1 Enable channel A conversion

5.4.6 gpdac_bctrl

Address：0x40000310

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

Bits	Name	Type	Reset	Description
31:23	RSVD			
22:20	BOM	R/W	3'h0	
19:18	BRNG	R/W	2'h3	Output voltage range control with internal/external reference
17:2	RSVD			
1	IOBEN	R/W	1'h0	channel B conversion output to pad enable 1'h0 Disable channel B conversion result to GPIO 1'h1 Enable channel B conversion result to GPIO
0	BEN	R/W	1'h0	channel B enable/disable signal 1'h0 Disable channel B conversion. 1'h1 Enable channel B conversion

5.4.7 gpdac_data

Address：0x40000314

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

Bits	Name	Type	Reset	Description
31:26	RSVD			
25:16	ADATA	R/W	10'h0	Channel A Data input
15:10	RSVD			
9:0	BDATA	R/W	10'h0	Channel B Data input

6 DMA

6.1 DMA Introduction

DMA (Direct Memory Access) is a memory access technology that can independently read and write system memory directly without processor intervention. Under the same degree of processor load, DMA is a fast data transfer method. The DMA controller has 8 channels, which manage the data transfer between peripheral devices and memory to improve bus efficiency.

There are three main types of transfers: memory to memory, memory to peripheral, and peripheral to memory. And support LLI link list function. Use the software to configure the transmission data size, data source address, and destination address.

6.2 DMA main features

- 8 independently configurable channels (requests) on DMA
- Independent control of source and destination access width (single-byte, double-byte, four-byte)
- Each channel acts as a read-write cache independently
- Each channel can be triggered by independent peripheral hardware or software
- Support peripherals including UART、I2C、SPI、ADC、I2S。
- 8 kinds of process control
  - DMA flow control, source memory, target memory
  - DMA flow control, source memory, target peripheral
  - DMA flow control, source peripheral, target memory
  - DMA flow control, source peripheral, target peripheral
  - Target peripheral process control, source peripheral, target peripheral

  - Target peripheral process control, source memory, target peripheral
  - Source peripheral process control, source peripheral, target memory
  - Source peripheral process control, source peripheral, target peripheral
- Support LLI linked list function to improve DMA efficiency

6.3 DMA functional description

6.3.1 Working principle

When a device attempts to transfer data (usually a large amount of data) directly to another device via the bus, it will first send a DMA request signal to the CPU. The peripheral device makes a bus request to the CPU to take over the bus control right through the DMA. After the CPU receives the signal, after the current bus cycle ends, it will respond to the DMA signal according to the priority of the DMA signal and the order of the DMA request.

When the CPU responds to a DMA request to a device interface, it will give up bus control.

Therefore, under the management of the DMA controller, the peripherals and the memory directly exchange data without CPU intervention. After the data transfer is complete, the device sends a DMA end signal to the CPU, returning the bus control.

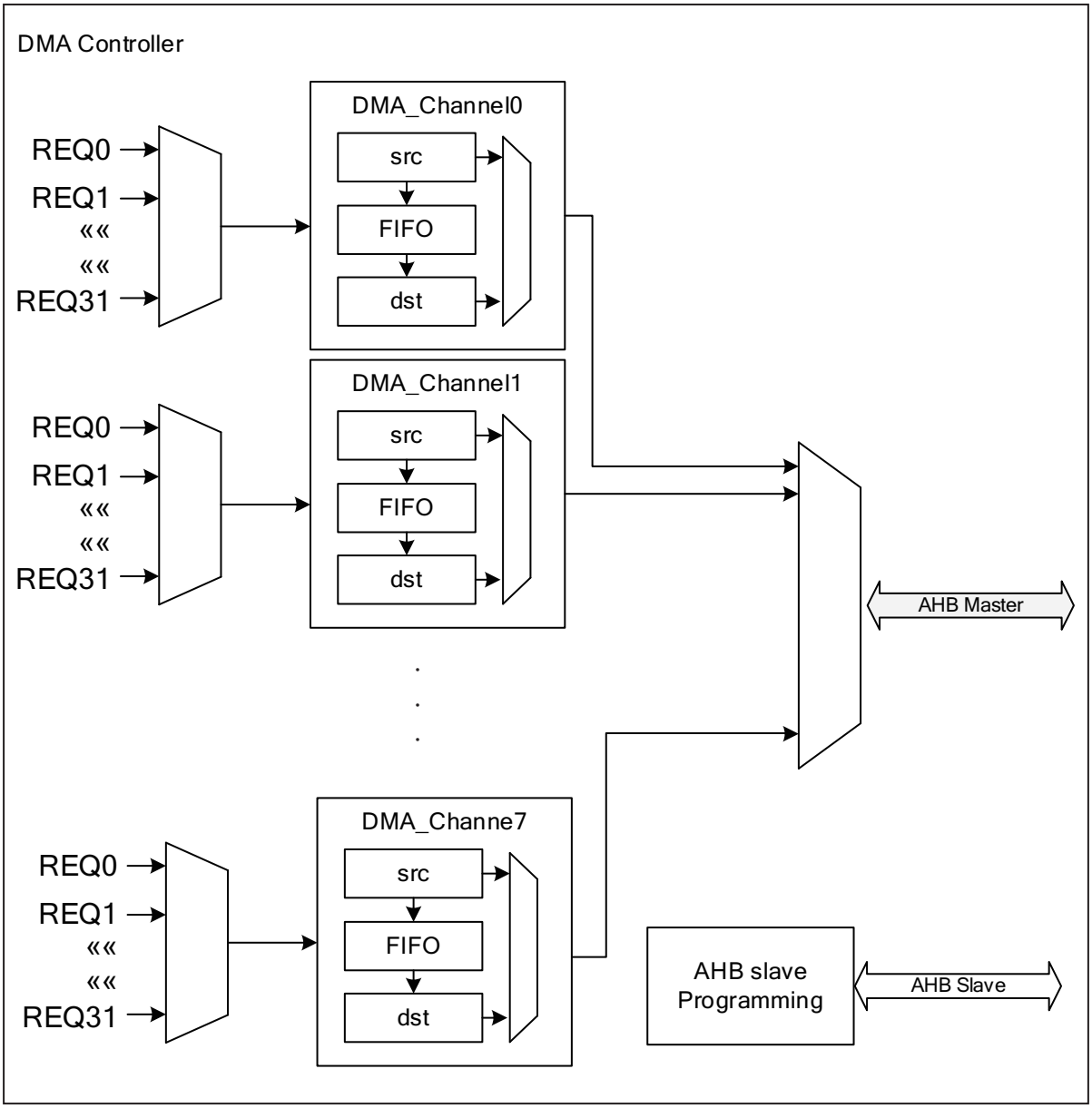

Figure 6.1: DMA architecture

The DMA includes a set of AHB Master interfaces and a set of AHB Slave interfaces. The AHB Master interface actively accesses memory or peripherals through the system bus according to the current configuration requirements, as a port for data movement. The AHB Slave interface is used to configure the DMA interface and only supports 32-bit access.

6.3.2 DMA channel configuration

DMA supports 8 channels in total, each channel does not interfere with each other and can run at the same time. The following is the configuration process of DMA channel x:

1. Set 32-bit source address in DMA_C0SrcAddr register

2. Set the 32-bit target address in the DMA_C0DstAddr register
3. Configure SI (source) and DI (destination) in the DMA_C0Control register to set whether to enable the automatic address accumulation mode. When set to 1, enable the automatic address accumulation mode
4. Set the transmission data width by configuring the STW (source) and DTW (destination) bits in the DMA_C0Control register. The width options are single-byte, double-self-knot, and four-byte
5. Burst type, which can be set by configuring the SBS (source) and DBS (destination) bits in the DMA_C0Control register. The configuration options are Single, INCR4, INCR8, INCR16
6. Special attention should be paid to the configured combination. A single burst cannot exceed 16 bytes
7. Set the data transmission length range: 0-4095

6.3.3 Peripheral support

The SrcPeripheral (source) and DstPeripheral (destination) are configured to determine the peripherals that the current DMA cooperates with. The relationship is 0-5 : UART / 6-9 : I2C / 10-13 : SPI / 18-21 : I2S / 22-23 : ADC

UART uses DMA to transfer data

UART sends data packets, using DMA method can greatly reduce CPU processing time, so that its CPU resources are not wasted a lot. Especially when the UART sends and receives a large number of data packets (such as high-frequency sending and receiving instructions) has obvious advantages.

Taking UART0 transmission as an example, the configuration process is as follows:

1. Set the value of the register DMA_C0Config [SRCPH] bit to 1, that is, set the Source peripheral to UART_TX
2. Set the value of the DMA_C0Config [DSTPH] bit to 0, that is, set the Destination peripheral to UART_RX

I2C uses DMA to transfer data

The configuration is as follows:

1. Set the value of the register DMA_C0Config [SRCPH] bit to 7, that is, set the Source peripheral to I2C_TX
2. Set the value of the DMA_C0Config [DSTPH] bit to 6, that is, set the Destination peripheral to I2C_RX

SPI uses DMA to transfer data

The configuration is as follows:

1. Set the value of the DMA_C0Config [SRCPH] bit to 11, that is, set the Source peripheral to SPI_TX
2. Set the value of the DMA_C0Config [DSTPH] bit to 10, that is, set the Destination peripheral to SPI_RX

ADC0/1 uses DMA to transfer data

The configuration is as follows:

1. Set the value of the DMA_C0Config [SRCPH] bit to 22/23, that is, set the Source peripheral to GPADC0 / GPADC1

6.3.4 Linked List Mode

DMA supports linked list operation mode. When performing a DMA read or write operation, you can fill the data in the next linked list. After completing the data transfer of the current linked list, read the DMA_C0LLI register to obtain the start address of the next linked list, and directly transfer the data in the next linked list.

Ensure continuous and uninterrupted work during DMA transfer, and improve the efficiency of CPU and DMA.

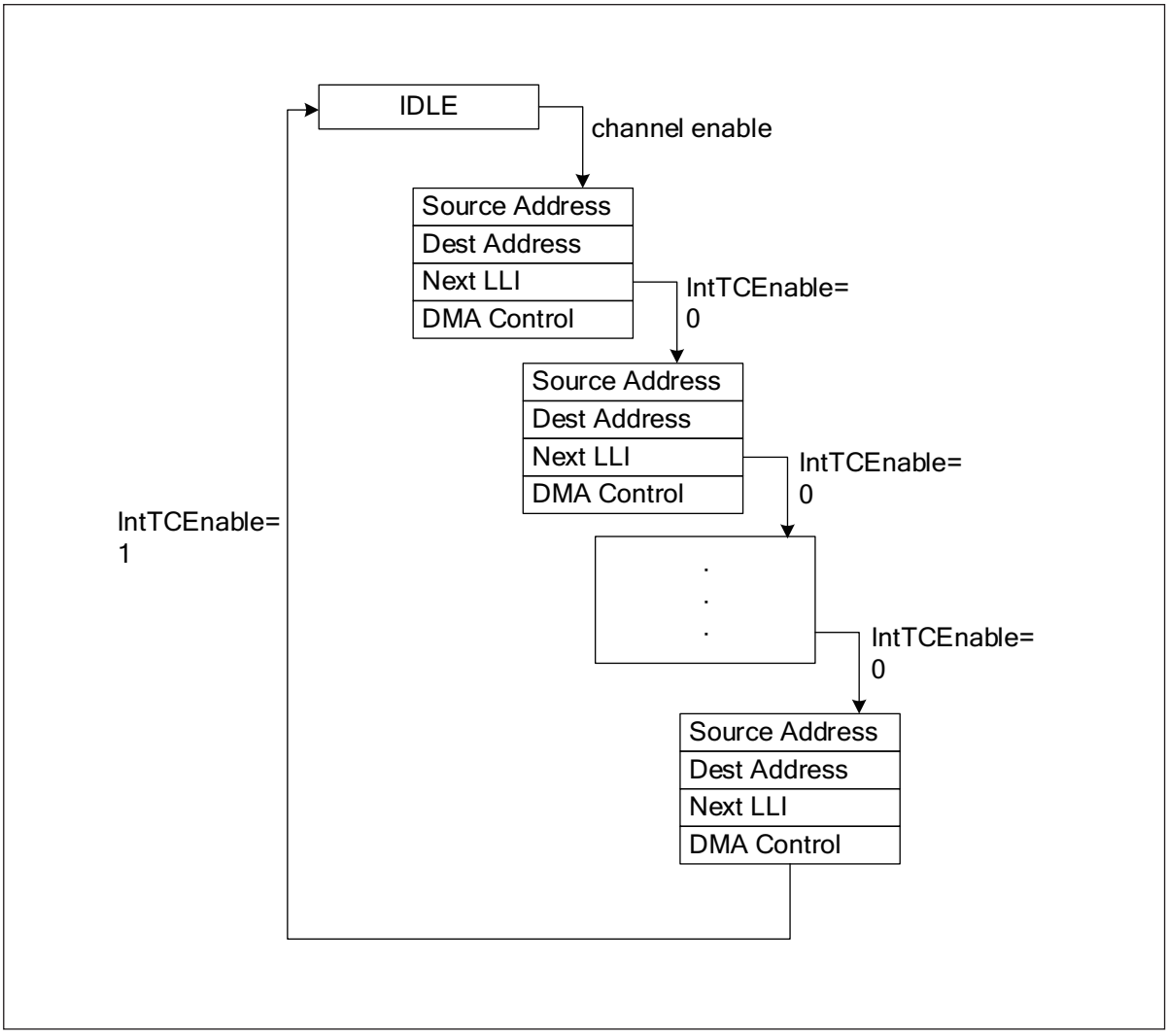

Figure 6.2: LLI architecture

6.3.5 DMA interrupt

- DMA_INT_TCOMPLETED
  - Data transmission completed interrupt. When a data transmission is completed, this interrupt will be entered.
- DMA_INT_ERR

- Data transmission error interrupt, when an error occurs during data transmission, this interrupt will be entered

6.4 Transmission mode

6.4.1 Memory to memory

After this mode is started, the DMA will move the data from the source address to the destination address according to the set transfer size. After the transfer, the DMA controller will automatically return to the idle state and wait for the next transfer.

The specific configuration process is as follows:

1. Set the value of the register DMA_C0SrcAddr to the memory address of the source
2. Set the value of the register DMA_C0DstAddr to the target memory address
3. Select the transmission mode and set the value of the DMA_C0Config [FLOWCTRL] bit to 0, that is, select the memory-to-memory mode
4. Set the value of the corresponding bit in the DMA_C0Control register: set the DI and SI bits to 1 to enable the automatic address accumulation mode, the DTW and STW bits set the transmission width of the source and destination, and the DBS and SBS bits set the burst type of the source and destination
5. Select the appropriate channel, enable DMA, and complete the data transfer

6.4.2 Memory to peripheral

In this working mode, the DMA will move data from the source to the internal cache according to the set transfer size (TransferSize). When the cache space is insufficient, the DMA will automatically suspend it. When there is sufficient cache space, continue to transfer until it reaches Set the moving quantity.

On the other hand, when the target peripheral request triggers, it will burst the target configuration to the target address until it reaches the set number of moves and automatically returns to the idle state, waiting for the next startup.

The specific configuration process is as follows:

1. Set the value of the register DMA_C0SrcAddr to the memory address of the source
2. Set the value of the register DMA_C0DstAddr to the target peripheral address
3. Select the transfer mode and set the value of the DMA_C0Config [FLOWCTRL] bit to 1 to select the memory-to-peripheral mode
4. Set the value of the corresponding bit in the DMA_C0Control register: set the DI and SI bits to 1 to enable the automatic address accumulation mode, the DTW and STW bits set the transmission width of the source and destination, and the DBS and SBS bits set the burst type of the source and destination
5. Select the appropriate channel, enable DMA, and complete the data transfer

6.4.3 Peripheral to memory

In this working mode, when the source peripheral request is triggered, the source configuration is burst to the buffer until the set number of moves reaches the stop. On the other hand, when the internal cache is enough for the target burst number once, the DMA will automatically move the cached content to the target address until it reaches the set number of moves and automatically returns to the idle state, waiting for the next startup

The specific configuration process is as follows:

1. Set the value of the register DMA_C0SrcAddr to the source peripheral address
2. Set the value of the register DMA_C0DstAddr to the target memory address
3. Select the transfer mode and set the value of the DMA_C0Config [FLOWCTRL] bit to 2 to select the Peripheral-to-memory mode
4. Set the value of the corresponding bit in the DMA_C0Control register: set the DI and SI bits to 1 to enable the automatic address accumulation mode, the DTW and STW bits set the transmission width of the source and destination respectively, and the DBS and SBS bits set the burst type of the source and destination respectively
5. Select the appropriate channel, enable DMA, and complete the data transfer

6.5 Register description

Name	Description
DMA_IntStatus	Interrupt status
DMA_IntTCStatus	Interrupt terminal count request status
DMA_IntTCClear	Terminal count request clear
DMA_IntErrorStatus	Interrupt error status
DMA_IntErrClr	Interrupt error clear
DMA_RawIntTCStatus	Status of the terminal count interrupt prior to masking
DMA_RawIntErrorStatus	Status of the error interrupt prior to masking
DMA_EnbldChns	Channel enable status
DMA_SoftBReq	Software burst request
DMA_SoftSReq	Software single request
DMA_SoftLBReq	Software last burst request
DMA_SoftLSReq	Software last single request
DMA_Config	DMA general configuration
DMA_Sync	DMA request asynchronous setting

Name	Description
DMA_C0SrcAddr	Channel DMA source address
DMA_C0DstAddr	Channel DMA Destination address
DMA_C0LLI	Channel DMA link list
DMA_C0Control	Channel DMA bus control
DMA_C0Config	Channel DMA configuration
DMA_C1SrcAddr	Channel DMA source address
DMA_C1DstAddr	Channel DMA Destination address
DMA_C1LLI	Channel DMA link list
DMA_C1Control	Channel DMA bus control
DMA_C1Config	Channel DMA configuration
DMA_C2SrcAddr	Channel DMA source address
DMA_C2DstAddr	Channel DMA Destination address
DMA_C2LLI	Channel DMA link list
DMA_C2Control	Channel DMA bus control
DMA_C2Config	Channel DMA configuration
DMA_C3SrcAddr	Channel DMA source address
DMA_C3DstAddr	Channel DMA Destination address
DMA_C3LLI	Channel DMA link list
DMA_C3Control	Channel DMA bus control
DMA_C3Config	Channel DMA configuration
DMA_C4SrcAddr	Channel DMA source address
DMA_C4DstAddr	Channel DMA Destination address
DMA_C4LLI	Channel DMA link list
DMA_C4Control	Channel DMA bus control
DMA_C4Config	Channel DMA configuration
DMA_C5SrcAddr	Channel DMA source address
DMA_C5DstAddr	Channel DMA Destination address
DMA_C5LLI	Channel DMA link list
DMA_C5Control	Channel DMA bus control

Name	Description
DMA_C5Config	Channel DMA configuration
DMA_C6SrcAddr	Channel DMA source address
DMA_C6DstAddr	Channel DMA Destination address
DMA_C6LLI	Channel DMA link list
DMA_C6Control	Channel DMA bus control
DMA_C6Config	Channel DMA configuration
DMA_C7SrcAddr	Channel DMA source address
DMA_C7DstAddr	Channel DMA Destination address
DMA_C7LLI	Channel DMA link list
DMA_C7Control	Channel DMA bus control
DMA_C7Config	Channel DMA configuration

6.5.1 DMA_IntStatus

Address：0x40007000

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	INTSTA	R	0	Status of the DMA interrupts after masking

6.5.2 DMA_IntTCStatus

Address：0x40007004

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	INTTCSTA	R	0	Interrupt terminal count request status

6.5.3 DMA_IntTCClear

Address：0x40007008

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	TCRC	W	0	Terminal count request clear

6.5.4 DMA_IntErrorStatus

Address：0x4000700c

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	IES	R	0	Interrupt error status

6.5.5 DMA_IntErrClr

Address：0x40007010

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	IEC	W	0	Interrupt error clear

6.5.6 DMA_RawIntTCStatus

Address：0x40007014

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	SOTCIPTM	R	0	Status of the terminal count interrupt prior to masking

6.5.7 DMA_RawIntErrorStatus

Address：0x40007018

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	SOTEIPTM	R	0	Status of the error interrupt prior to masking

6.5.8 DMA_EnbldChns

Address：0x4000701c

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	CES	R	0	Channel enable status

6.5.9 DMA_SoftBReq

Address：0x40007020

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

Bits	Name	Type	Reset	Description
31:0	SBR	R/W	0	Software burst request

6.5.10 DMA_SoftSReq

Address：0x40007024

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

Bits	Name	Type	Reset	Description
31:0	SSR	R/W	0	Software single request

6.5.11 DMA_SoftLBReq

Address：0x40007028

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

Bits	Name	Type	Reset	Description
31:0	SLBR	R/W	0	Software last burst request

6.5.12 DMA_SoftLSReq

Address：0x4000702c

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

Bits	Name	Type	Reset	Description
31:0	SLSR	R/W	0	Software last single request

6.5.13 DMA_Config

Address：0x40007030

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

Bits	Name	Type	Reset	Description
31:2	RSVD			
1	AHBMEC	R/W	0	AHB Master endianness configuration: 0 = little-endian, 1 = big-endian
0	SDMAEN	R/W	0	SMDMA Enable.

6.5.14 DMA_Sync

Address：0x40007034

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

Bits	Name	Type	Reset	Description
31:0	DSLFDRS	R/W	0	DMA synchronization logic for DMA request signals: 0 = enable, 1 = disable

6.5.15 DMA_C0SrcAddr

Address：0x40007100

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

Bits	Name	Type	Reset	Description
31:0	DMASA	R/W	0	DMA source address

6.5.16 DMA_C0DstAddr

Address：0x40007104

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

Bits	Name	Type	Reset	Description
31:0	DMADA	R/W	0	DMA Destination address

6.5.17 DMA_C0LLI

Address：0x40007108

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

Bits	Name	Type	Reset	Description
31:0	FLLI	R/W	0	First linked list item. Bits [1:0] must be 0.

6.5.18 DMA_C0Control

Address：0x4000710c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCI EN	PROTECT			DI	SI	RSVD	FIXCNT		DTW		RSVD	STW		ADD MODE	DBS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBS	MIN MODE	SBS		TS											

Bits	Name	Type	Reset	Description
31	TCIEN	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROTECT	R/W	0	Protection.
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25	RSVD			
24:23	FIXCNT	R/W	2'd0	Only effect when dst_min_mode = 1 Destination transfer cnt = (total src byte cnt - (fix_cnt«DWidth))«DWidth
22:21	DTW	R/W	2'b10	Destination transfer width: 8/16/32
20	RSVD			

Bits	Name	Type	Reset	Description
19:18	STW	R/W	2'b10	Source transfer width: 8/16/32
17	ADDMODE	R/W	1'b0	Add mode : issue remain destination traffic
16:15	DBS	R/W	2'b01	Destination burst size: 1/4/8/16
14	MINMODE	R/W	1'b0	Minus mode : Not issue all destination traffic
13:12	SBS	R/W	2'b01	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TS	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.19 DMA_C0Config

Address：0x40007110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		LLICOUNT										RSVD	HALT	ACTIVE	LOCK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCIM	IEM	FLOWCTRL			DSTPH					SRCPH					CHEN

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:20	LLICOUNT	R	0	LLI counter. Increased 1 each LLI run. Cleared 0 when config Control.
19	RSVD			
18	HALT	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	ACTIVE	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	LOCK	R/W	0	Lock.
15	TCIM	R/W	0	Terminal count interrupt mask.
14	IEM	R/W	0	Interrupt error mask.

Bits	Name	Type	Reset	Description
13:11	FLOWCTRL	R/W	0	000: Memory-to-memory (DMA) 001: Memory-to-peripheral (DMA) 010: Peripheral-to-memory (DMA) 011: Source peripheral-to-Destination peripheral (DMA) 100: Source peripheral-to-Destination peripheral (Destination peripheral) 101: Memory-to-peripheral (peripheral) 110: Peripheral-to-memory (peripheral) 111: Source peripheral-to-Destination peripheral (Source peripheral)
10:6	DSTPH	R/W	0	Destination peripheral. [23:22] GPADC [21:18] I2S [17:14] PDM [13:10] SSP [9: 6] I2C [5: 0] UART
5:1	SRCPH	R/W	0	Source peripheral.
0	CHEN	R/W	0	Channel enable.

6.5.20 DMA_C1SrcAddr

Address：0x40007200

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

Bits	Name	Type	Reset	Description
31:0	DMASA	R/W	0	DMA source address

6.5.21 DMA_C1DstAddr

Address：0x40007204

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

Bits	Name	Type	Reset	Description
31:0	DMADA	R/W	0	DMA Destination address

6.5.22 DMA_C1LLI

Address：0x40007208

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

Bits	Name	Type	Reset	Description
31:2	FLLI	R/W	0	First linked list item. Bits [1:0] must be 0.
1:0	RSVD			

6.5.23 DMA_C1Control

Address：0x4000720c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCI EN	PROTECT			DI	SI	RSVD	FIXCNT		DTW		STW			ADD MODE	DBS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBS	MIN MODE	SBS		TS											

Bits	Name	Type	Reset	Description
31	TCIEN	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROTECT	R/W	0	Protection.

Bits	Name	Type	Reset	Description
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25	RSVD			
24:23	FIXCNT	R/W	2'd0	Only effect when dst_min_mode = 1 Destination transfer cnt = (total src byte cnt - (fix_cnt«DWidth))«DWidth
22:21	DTW	R/W	3'b010	Destination transfer width: 8/16/32
20:18	STW	R/W	3'b010	Source transfer width: 8/16/32
17	ADDMODE	R/W	1'b0	Add mode : issue remain destination traffic
16:15	DBS	R/W	3'b001	Destination burst size: 1/4/8/16
14	MINMODE	R/W	1'b0	Minus mode : Not issue all destination traffic
13:12	SBS	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TS	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.24 DMA_C1Config

Address：0x40007210

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD													HALT	ACTIVE	LOCK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCIM	IEM	FLOWCTRL			DSTPH					SRCPH					CHEN

Bits	Name	Type	Reset	Description
31:19	RSVD			
18	HALT	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	ACTIVE	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	LOCK	R/W	0	Lock.
15	TCIM	R/W	0	Terminal count interrupt mask.

Bits	Name	Type	Reset	Description
14	IEM	R/W	0	Interrupt error mask.
13:11	FLOWCTRL	R/W	0	000: Memory-to-memory (DMA) 001: Memory-to-peripheral (DMA) 010: Peripheral-to-memory (DMA) 011: Source peripheral-to-Destination peripheral (DMA) 100: Source peripheral-to-Destination peripheral (Destination peripheral) 101: Memory-to-peripheral (peripheral) 110: Peripheral-to-memory (peripheral) 111: Source peripheral-to-Destination peripheral (Source peripheral)
10:6	DSTPH	R/W	0	Destination peripheral. [23:22] GPADC [21:18] I2S [17:14] PDM [13:10] SSP [9: 6] I2C [5: 0] UART
5:1	SRCPH	R/W	0	Source peripheral.
0	CHEN	R/W	0	Channel enable.

6.5.25 DMA_C2SrcAddr

Address：0x40007300

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

Bits	Name	Type	Reset	Description
31:0	DMASA	R/W	0	DMA source address

6.5.26 DMA_C2DstAddr

Address：0x40007304

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

Bits	Name	Type	Reset	Description
31:0	DMADA	R/W	0	DMA Destination address

6.5.27 DMA_C2LLI

Address：0x40007308

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

Bits	Name	Type	Reset	Description
31:2	FLLI	R/W	0	First linked list item. Bits [1:0] must be 0.
1:0	RSVD			

6.5.28 DMA_C2Control

Address：0x4000730c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCI EN	PROTECT			DI	SI	RSVD	FIXCNT		DTW		STW			ADD MODE	DBS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBS	MIN MODE	SBS		TS											

Bits	Name	Type	Reset	Description
31	TCIEN	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROTECT	R/W	0	Protection.

Bits	Name	Type	Reset	Description
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25	RSVD			
24:23	FIXCNT	R/W	2'd0	Only effect when dst_min_mode = 1 Destination transfer cnt = (total src byte cnt - (fix_cnt«DWidth))«DWidth
22:21	DTW	R/W	3'b010	Destination transfer width: 8/16/32
20:18	STW	R/W	3'b010	Source transfer width: 8/16/32
17	ADDMODE	R/W	1'b0	Add mode : issue remain destination traffic
16:15	DBS	R/W	3'b001	Destination burst size: 1/4/8/16
14	MINMODE	R/W	1'b0	Minus mode : Not issue all destination traffic
13:12	SBS	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TS	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.29 DMA_C2Config

Address：0x40007310

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD													HALT	ACTIVE	LOCK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCIM	IEM	FLOWCTRL			DSTPH					SRCPH					CHEN

Bits	Name	Type	Reset	Description
31:19	RSVD			
18	HALT	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	ACTIVE	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	LOCK	R/W	0	Lock.
15	TCIM	R/W	0	Terminal count interrupt mask.

Bits	Name	Type	Reset	Description
14	IEM	R/W	0	Interrupt error mask.
13:11	FLOWCTRL	R/W	0	000: Memory-to-memory (DMA) 001: Memory-to-peripheral (DMA) 010: Peripheral-to-memory (DMA) 011: Source peripheral-to-Destination peripheral (DMA) 100: Source peripheral-to-Destination peripheral (Destination peripheral) 101: Memory-to-peripheral (peripheral) 110: Peripheral-to-memory (peripheral) 111: Source peripheral-to-Destination peripheral (Source peripheral)
10:6	DSTPH	R/W	0	Destination peripheral. [23:22] GPADC [21:18] I2S [17:14] PDM [13:10] SSP [9: 6] I2C [5: 0] UART
5:1	SRCPH	R/W	0	Source peripheral.
0	CHEN	R/W	0	Channel enable.

6.5.30 DMA_C3SrcAddr

Address：0x40007400

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

Bits	Name	Type	Reset	Description
31:0	DMASA	R/W	0	DMA source address

6.5.31 DMA_C3DstAddr

Address：0x40007404

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

Bits	Name	Type	Reset	Description
31:0	DMADA	R/W	0	DMA Destination address

6.5.32 DMA_C3LLI

Address：0x40007408

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

Bits	Name	Type	Reset	Description
31:2	FLLI	R/W	0	First linked list item. Bits [1:0] must be 0.
1:0	RSVD			

6.5.33 DMA_C3Control

Address：0x4000740c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCI EN	PROTECT			DI	SI	RSVD	FIXCNT		DTW		STW			ADD MODE	DBS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBS	MIN MODE	SBS		TS											

Bits	Name	Type	Reset	Description
31	TCIEN	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROTECT	R/W	0	Protection.

Bits	Name	Type	Reset	Description
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25	RSVD			
24:23	FIXCNT	R/W	2'd0	Only effect when dst_min_mode = 1 Destination transfer cnt = (total src byte cnt - (fix_cnt«DWidth))«DWidth
22:21	DTW	R/W	3'b010	Destination transfer width: 8/16/32
20:18	STW	R/W	3'b010	Source transfer width: 8/16/32
17	ADDMODE	R/W	1'b0	Add mode : issue remain destination traffic
16:15	DBS	R/W	3'b001	Destination burst size: 1/4/8/16
14	MINMODE	R/W	1'b0	Minus mode : Not issue all destination traffic
13:12	SBS	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TS	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.34 DMA_C3Config

Address：0x40007410

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD													HALT	ACTIVE	LOCK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCIM	IEM	FLOWCTRL			DSTPH					SRCPH					CHEN

Bits	Name	Type	Reset	Description
31:19	RSVD			
18	HALT	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	ACTIVE	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	LOCK	R/W	0	Lock.
15	TCIM	R/W	0	Terminal count interrupt mask.

Bits	Name	Type	Reset	Description
14	IEM	R/W	0	Interrupt error mask.
13:11	FLOWCTRL	R/W	0	000: Memory-to-memory (DMA) 001: Memory-to-peripheral (DMA) 010: Peripheral-to-memory (DMA) 011: Source peripheral-to-Destination peripheral (DMA) 100: Source peripheral-to-Destination peripheral (Destination peripheral) 101: Memory-to-peripheral (peripheral) 110: Peripheral-to-memory (peripheral) 111: Source peripheral-to-Destination peripheral (Source peripheral)
10:6	DSTPH	R/W	0	Destination peripheral. [23:22] GPADC [21:18] I2S [17:14] PDM [13:10] SSP [9: 6] I2C [5: 0] UART
5:1	SRCPH	R/W	0	Source peripheral.
0	CHEN	R/W	0	Channel enable.

6.5.35 DMA_C4SrcAddr

Address：0x40007500

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

Bits	Name	Type	Reset	Description
31:0	DMASA	R/W	0	DMA source address

6.5.36 DMA_C4DstAddr

Address：0x40007504

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

Bits	Name	Type	Reset	Description
31:0	DMADA	R/W	0	DMA Destination address

6.5.37 DMA_C4LLI

Address：0x40007508

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

Bits	Name	Type	Reset	Description
31:2	FLLI	R/W	0	First linked list item. Bits [1:0] must be 0.
1:0	RSVD			

6.5.38 DMA_C4Control

Address：0x4000750c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCI EN	PROTECT			DI	SI	RSVD	FIXCNT		DTW		STW			ADD MODE	DBS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBS	MIN MODE	SBS		TS											

Bits	Name	Type	Reset	Description
31	TCIEN	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROTECT	R/W	0	Protection.

Bits	Name	Type	Reset	Description
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25	RSVD			
24:23	FIXCNT	R/W	2'd0	Only effect when dst_min_mode = 1 Destination transfer cnt = (total src byte cnt - (fix_cnt«DWidth))«DWidth
22:21	DTW	R/W	3'b010	Destination transfer width: 8/16/32
20:18	STW	R/W	3'b010	Source transfer width: 8/16/32
17	ADDMODE	R/W	1'b0	Add mode : issue remain destination traffic
16:15	DBS	R/W	3'b001	Destination burst size: 1/4/8/16
14	MINMODE	R/W	1'b0	Minus mode : Not issue all destination traffic
13:12	SBS	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TS	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.39 DMA_C4Config

Address：0x40007510

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD													HALT	ACTIVE	LOCK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCIM	IEM	FLOWCTRL			DSTPH					SRCPH					CHEN

Bits	Name	Type	Reset	Description
31:19	RSVD			
18	HALT	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	ACTIVE	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	LOCK	R/W	0	Lock.
15	TCIM	R/W	0	Terminal count interrupt mask.

Bits	Name	Type	Reset	Description
14	IEM	R/W	0	Interrupt error mask.
13:11	FLOWCTRL	R/W	0	000: Memory-to-memory (DMA) 001: Memory-to-peripheral (DMA) 010: Peripheral-to-memory (DMA) 011: Source peripheral-to-Destination peripheral (DMA) 100: Source peripheral-to-Destination peripheral (Destination peripheral) 101: Memory-to-peripheral (peripheral) 110: Peripheral-to-memory (peripheral) 111: Source peripheral-to-Destination peripheral (Source peripheral)
10:6	DSTPH	R/W	0	Destination peripheral. [23:22] GPADC [21:18] I2S [17:14] PDM [13:10] SSP [9: 6] I2C [5: 0] UART
5:1	SRCPH	R/W	0	Source peripheral.
0	CHEN	R/W	0	Channel enable.

6.5.40 DMA_C5SrcAddr

Address：0x40007600

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

Bits	Name	Type	Reset	Description
31:0	DMASA	R/W	0	DMA source address

6.5.41 DMA_C5DstAddr

Address：0x40007604

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

Bits	Name	Type	Reset	Description
31:0	DMADA	R/W	0	DMA Destination address

6.5.42 DMA_C5LLI

Address：0x40007608

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

Bits	Name	Type	Reset	Description
31:2	FLLI	R/W	0	First linked list item. Bits [1:0] must be 0.
1:0	RSVD			

6.5.43 DMA_C5Control

Address：0x4000760c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCI EN	PROTECT			DI	SI	RSVD	FIXCNT		DTW		STW			ADD MODE	DBS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBS	MIN MODE	SBS		TS											

Bits	Name	Type	Reset	Description
31	TCIEN	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROTECT	R/W	0	Protection.

Bits	Name	Type	Reset	Description
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25	RSVD			
24:23	FIXCNT	R/W	2'd0	Only effect when dst_min_mode = 1 Destination transfer cnt = (total src byte cnt - (fix_cnt«DWidth))«DWidth
22:21	DTW	R/W	3'b010	Destination transfer width: 8/16/32
20:18	STW	R/W	3'b010	Source transfer width: 8/16/32
17	ADDMODE	R/W	1'b0	Add mode : issue remain destination traffic
16:15	DBS	R/W	3'b001	Destination burst size: 1/4/8/16
14	MINMODE	R/W	1'b0	Minus mode : Not issue all destination traffic
13:12	SBS	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TS	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.44 DMA_C5Config

Address：0x40007610

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD													HALT	ACTIVE	LOCK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCIM	IEM	FLOWCTRL			DSTPH					SRCPH					CHEN

Bits	Name	Type	Reset	Description
31:19	RSVD			
18	HALT	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	ACTIVE	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	LOCK	R/W	0	Lock.
15	TCIM	R/W	0	Terminal count interrupt mask.

Bits	Name	Type	Reset	Description
14	IEM	R/W	0	Interrupt error mask.
13:11	FLOWCTRL	R/W	0	000: Memory-to-memory (DMA) 001: Memory-to-peripheral (DMA) 010: Peripheral-to-memory (DMA) 011: Source peripheral-to-Destination peripheral (DMA) 100: Source peripheral-to-Destination peripheral (Destination peripheral) 101: Memory-to-peripheral (peripheral) 110: Peripheral-to-memory (peripheral) 111: Source peripheral-to-Destination peripheral (Source peripheral)
10:6	DSTPH	R/W	0	Destination peripheral. [23:22] GPADC [21:18] I2S [17:14] PDM [13:10] SSP [9: 6] I2C [5: 0] UART
5:1	SRCPH	R/W	0	Source peripheral.
0	CHEN	R/W	0	Channel enable.

6.5.45 DMA_C6SrcAddr

Address：0x40007700

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

Bits	Name	Type	Reset	Description
31:0	DMASA	R/W	0	DMA source address

6.5.46 DMA_C6DstAddr

Address：0x40007704

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

Bits	Name	Type	Reset	Description
31:0	DMADA	R/W	0	DMA Destination address

6.5.47 DMA_C6LLI

Address：0x40007708

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

Bits	Name	Type	Reset	Description
31:2	FLLI	R/W	0	First linked list item. Bits [1:0] must be 0.
1:0	RSVD			

6.5.48 DMA_C6Control

Address：0x4000770c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCI EN	PROTECT			DI	SI	RSVD	FIXCNT		DTW		STW			ADD MODE	DBS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBS	MIN MODE	SBS		TS											

Bits	Name	Type	Reset	Description
31	TCIEN	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROTECT	R/W	0	Protection.

Bits	Name	Type	Reset	Description
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25	RSVD			
24:23	FIXCNT	R/W	2'd0	Only effect when dst_min_mode = 1 Destination transfer cnt = (total src byte cnt - (fix_cnt«DWidth))«DWidth
22:21	DTW	R/W	3'b010	Destination transfer width: 8/16/32
20:18	STW	R/W	3'b010	Source transfer width: 8/16/32
17	ADDMODE	R/W	1'b0	Add mode : issue remain destination traffic
16:15	DBS	R/W	3'b001	Destination burst size: 1/4/8/16
14	MINMODE	R/W	1'b0	Minus mode : Not issue all destination traffic
13:12	SBS	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TS	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.49 DMA_C6Config

Address：0x40007710

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD													HALT	ACTIVE	LOCK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCIM	IEM	FLOWCTRL			DSTPH					SRCPH					CHEN

Bits	Name	Type	Reset	Description
31:19	RSVD			
18	HALT	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	ACTIVE	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	LOCK	R/W	0	Lock.
15	TCIM	R/W	0	Terminal count interrupt mask.

Bits	Name	Type	Reset	Description
14	IEM	R/W	0	Interrupt error mask.
13:11	FLOWCTRL	R/W	0	000: Memory-to-memory (DMA) 001: Memory-to-peripheral (DMA) 010: Peripheral-to-memory (DMA) 011: Source peripheral-to-Destination peripheral (DMA) 100: Source peripheral-to-Destination peripheral (Destination peripheral) 101: Memory-to-peripheral (peripheral) 110: Peripheral-to-memory (peripheral) 111: Source peripheral-to-Destination peripheral (Source peripheral)
10:6	DSTPH	R/W	0	Destination peripheral. [23:22] GPADC [21:18] I2S [17:14] PDM [13:10] SSP [9: 6] I2C [5: 0] UART
5:1	SRCPH	R/W	0	Source peripheral.
0	CHEN	R/W	0	Channel enable.

6.5.50 DMA_C7SrcAddr

Address：0x40007800

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

Bits	Name	Type	Reset	Description
31:0	DMASA	R/W	0	DMA source address

6.5.51 DMA_C7DstAddr

Address：0x40007804

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

Bits	Name	Type	Reset	Description
31:0	DMADA	R/W	0	DMA Destination address

6.5.52 DMA_C7LLI

Address：0x40007808

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

Bits	Name	Type	Reset	Description
31:2	FLLI	R/W	0	First linked list item. Bits [1:0] must be 0.
1:0	RSVD			

6.5.53 DMA_C7Control

Address：0x4000780c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCI EN	PROTECT			DI	SI	RSVD	FIXCNT		DTW		STW			ADD MODE	DBS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBS	MIN MODE	SBS		TS											

Bits	Name	Type	Reset	Description
31	TCIEN	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROTECT	R/W	0	Protection.

Bits	Name	Type	Reset	Description
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25	RSVD			
24:23	FIXCNT	R/W	2'd0	Only effect when dst_min_mode = 1 Destination transfer cnt = (total src byte cnt - (fix_cnt«DWidth))«DWidth
22:21	DTW	R/W	3'b010	Destination transfer width: 8/16/32
20:18	STW	R/W	3'b010	Source transfer width: 8/16/32
17	ADDMODE	R/W	1'b0	Add mode : issue remain destination traffic
16:15	DBS	R/W	3'b001	Destination burst size: 1/4/8/16
14	MINMODE	R/W	1'b0	Minus mode : Not issue all destination traffic
13:12	SBS	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TS	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.54 DMA_C7Config

Address：0x40007810

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD													HALT	ACTIVE	LOCK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCIM	IEM	FLOWCTRL			DSTPH					SRCPH					CHEN

Bits	Name	Type	Reset	Description
31:19	RSVD			
18	HALT	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	ACTIVE	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	LOCK	R/W	0	Lock.
15	TCIM	R/W	0	Terminal count interrupt mask.

Bits	Name	Type	Reset	Description
14	IEM	R/W	0	Interrupt error mask.
13:11	FLOWCTRL	R/W	0	000: Memory-to-memory (DMA) 001: Memory-to-peripheral (DMA) 010: Peripheral-to-memory (DMA) 011: Source peripheral-to-Destination peripheral (DMA) 100: Source peripheral-to-Destination peripheral (Destination peripheral) 101: Memory-to-peripheral (peripheral) 110: Peripheral-to-memory (peripheral) 111: Source peripheral-to-Destination peripheral (Source peripheral)
10:6	DSTPH	R/W	0	Destination peripheral. [23:22] GPADC [21:18] I2S [17:14] PDM [13:10] SSP [9: 6] I2C [5: 0] UART
5:1	SRCPH	R/W	0	Source peripheral.
0	CHEN	R/W	0	Channel enable.

7 L1C

7.1 L1C introduction

L1 Cache Controller is a unit module located outside the processor, used to manage the code or data buffer on Flash/pSRAM and improve the speed of CPU access to Flash/pSRAM. The architecture is as follows:

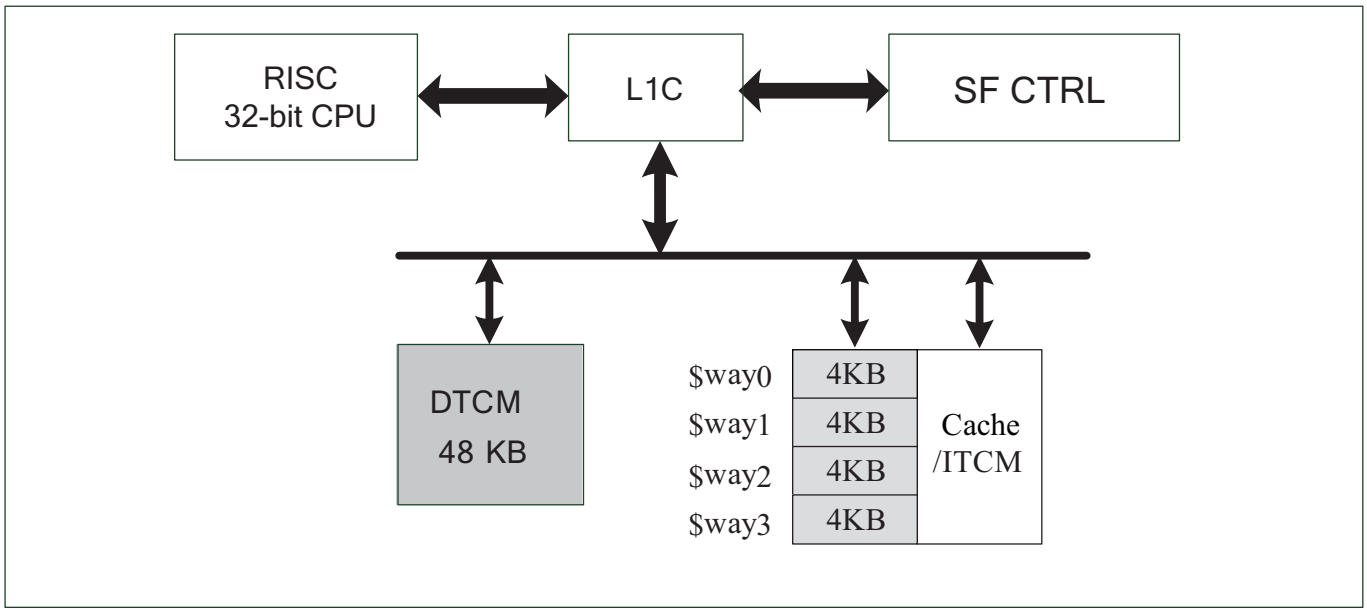

Figure 7.1: L1c architecture

L1C is a high-speed unit integrated between the processor and Flash. Because the speed of the processor is very fast, when the processor needs to wait for a long time to access the Flash, the waiting time represents wasteful time. The L1C cache can be used as a lubricating role between the processor and the Flash to improve the efficiency of the processor.

7.2 L1C main features

- 4-way Set-Associative mapping

- Variable cache size
- Connect to TCM address space, can easily configure L1C space as TCM space
- Support cache performance statistics

7.3 L1C function description

7.3.1 Mutual conversion between TCM and Cache RAM resources

In order to increase memory usage efficiency, it is supported to adjust all or part of the Cache's 16K RAM to TCM space, so that users can adjust the memory usage and efficiency according to the actual situation. The default size of Cache is 16K, divided into 4 ways, each way is 4K, and the unit of adjustment is 1 way, which is 4K. The default size of ITCM is 16K. Set by WayDisable. The actual space size of Cache and ITCM can be flexibly adjusted.

Table 7.1: WayDisable settings

WayDisable	Cache	ITCM
none	16K	0K
one way	12K	4K
two way	8K	8K
three way	4K	12K
four way	0K	16K

7.3.2 Cache

The unit of each line buffer is 32 bytes, and the 4-way associative mapping cache is used. The application architecture is as follows:

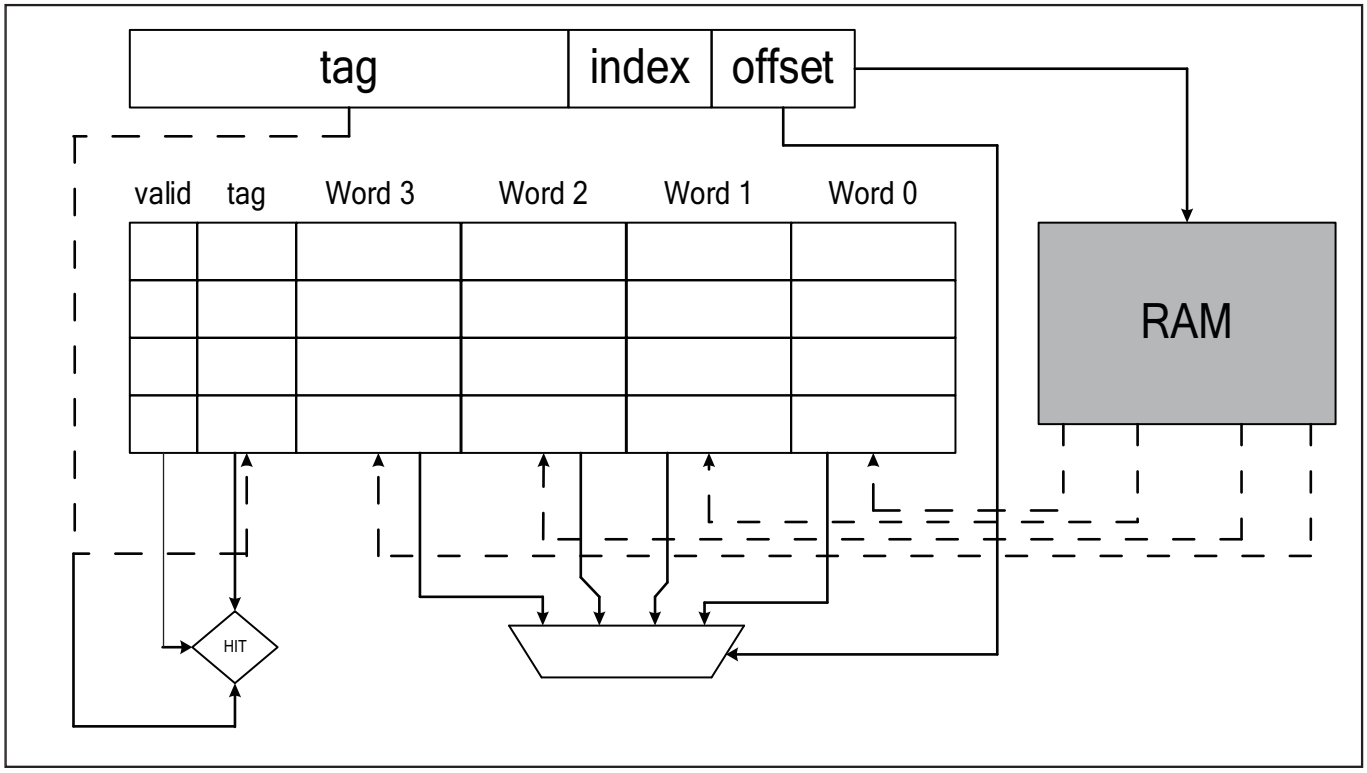

Figure 7.2: Cache architecture

Each set of associative mapping caches contains two parts, the first is a tag, which contains the valid value and the address mapping relationship. The second part is data storage. When the processor accesses the cache, the cache processor compares the relationship between the address and the tag. When the address comparison is successful, the representative can directly get data from the cache. Conversely, the cache processor will capture related data through the AHB Master and put the data into the cache and respond to the processor's data.

When most of the data can be successfully compared in the tag, the waiting time of the processor can be greatly reduced, and the use efficiency can be increased.

7.4 Register description

Name	Description
l1c_config	L1C configuration
hit_cnt_lsb	Low 32-bit hit couter
hit_cnt_msb	High 32-bit hit counter
miss_cnt	Miss counter

7.4.1 l1c_config

Address：0x40009000

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

Bits	Name	Type	Reset	Description
31:12	RSVD			
11:8	WAYDIS	R/W	4'b1111	Disable part of cache ways & used as ITCM
7:2	RSVD			
1	CNTEN	R/W	0	
0	CACABLE	R/W	0	

7.4.2 hit_cnt_lsb

Address：0x40009004

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

Bits	Name	Type	Reset	Description
31:0	CNTLSB	R	0	

7.4.3 hit_cnt_msb

Address：0x40009008

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

Bits	Name	Type	Reset	Description
31:0	CNTMSB	R	0	total hit count = hit_cnt_msb*2^32 + hit_cnt_lsb

7.4.4 miss_cnt

Address：0x4000900c

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

Bits	Name	Type	Reset	Description
31:0	MISSCNT	R	0	

8 IR

8.1 IR introduction

Infrared remote (IR for short) is a wireless, non-contact control technology, which has the advantages of strong anti-interference ability, reliable information transmission, low power consumption and low cost. The infrared remote control transmitting circuit uses infrared light emitting diodes to emit modulated infrared light waves. The receiving circuit consists of infrared receiving diodes, triodes or silicon photocells. They convert the infrared light emitted by the infrared transmitter into the corresponding electrical signal and send it to the rear amplifier.

8.2 IR main features

- Receiving data with NEC, RC-5 protocol
- Receiving arbitrary format data in pulse width counting mode
- Powerful infrared waveform editing capabilities, which can emit waveforms conforming to various protocols
- Power settings of up to 15 gears to suit different power requirements
- Supports up to 64-bit data bits
- 64-byte receive FIFO
- Programmable carrier frequency and duty cycle

8.3 IR function description

8.3.1 Fixed receiving protocol

IR receiver supports two fixed protocols, NEC protocol and RC-5 protocol.

- NEC protocol

The logic 1 and logic 0 waveforms of the NEC protocol are shown in the following figure:

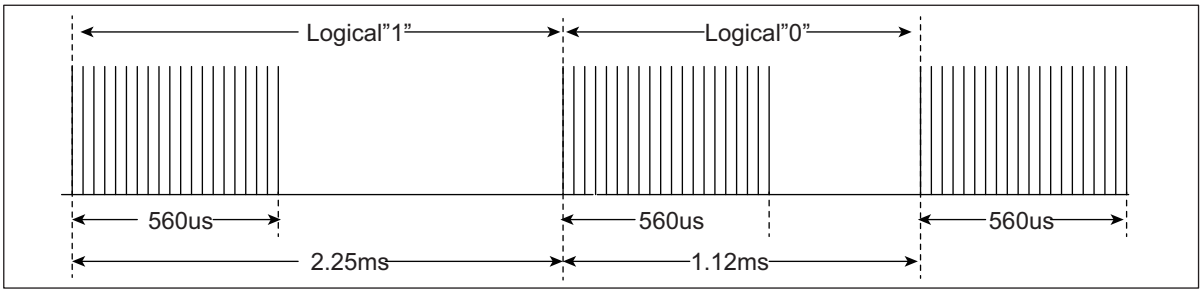

Figure 8.1: nec logical

Logic 1 is 2.25ms, pulse time is 560us; logic 0 bit is 1.12ms, pulse time is 560us.

The specific format of the NEC protocol is shown in the following figure:

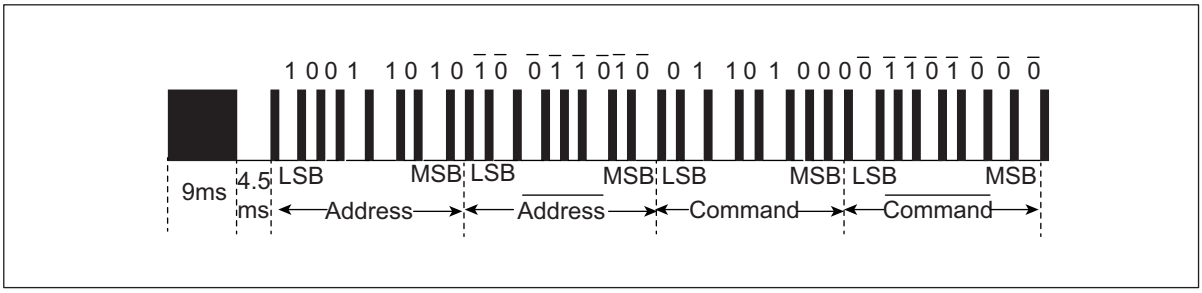

Figure 8.2: nec

The first pulse is a high-level pulse of 9ms and a low-level of 4.5ms, followed by an 8-bit address code and its inverse code, and then an 8-bit command code and its inverse code. The tail pulse is 560us high and 560us low.

- RC-5 protocol

The logic 1 and logic 0 waveforms of the RC-5 protocol are shown in the following figure:

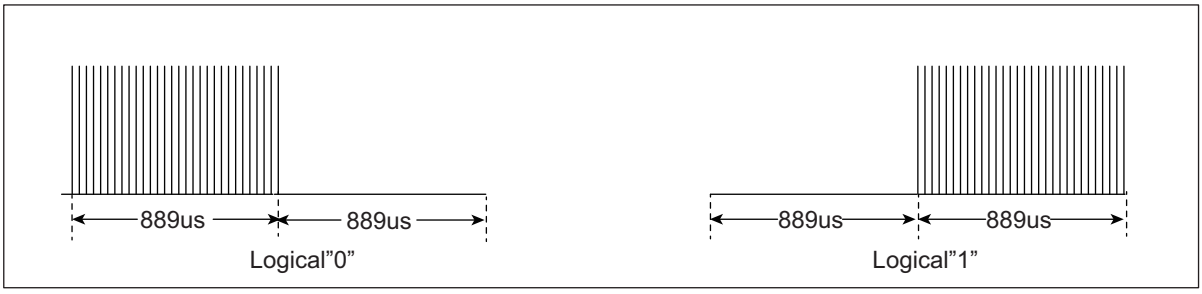

Figure 8.3: rc5 logical

Logic 1 is 1.778ms, which is 889us low and then 889us high; logic 0 and logic 1 have opposite waveforms.

The specific format of the RC-5 protocol is shown in the following figure:

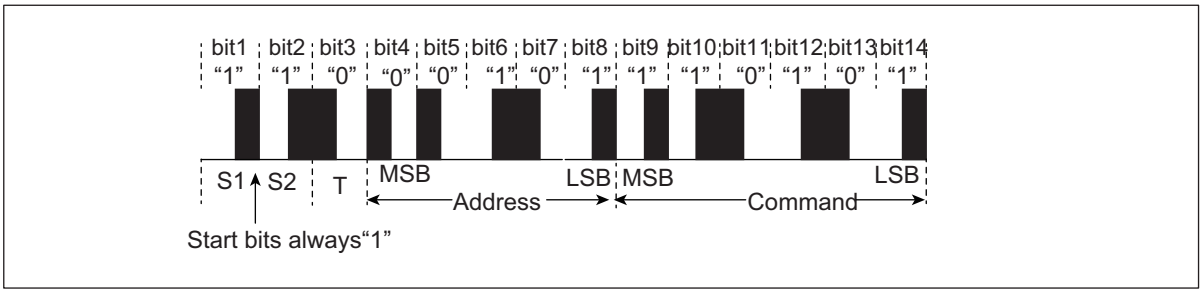

Figure 8.4: rc5

The first two bits are the start bit, fixed to logic 1, and the third bit is the flip bit. When a key value is issued and then pressed, the bit will be inverted. The next 5 bits are the address code and the 6 bits command code. The first two bits are the start bit, fixed to logic 1, and the third bit is the flip bit. When a key value is issued and then pressed, the bit will be inverted. The next 5 digits are the address code and the 6-digit command code.

It should be noted that in order to improve the receiving sensitivity, the common infrared integrated receiver head outputs a low level after receiving a high level, so when the IR receiving function is used, the receiving flip function must be turned on.

8.3.2 Pulse width reception

For data in any format other than the NEC and RC-5 protocols, the IR will count the duration of each high and low level in turn using its clock, and then store the data in a 64-byte depth receiving FIFO.

8.3.3 Normal sending mode

Users can configure the corresponding configurations of the head pulse, tail pulse, logic 0 and logic 1 pulses according to specific protocols. When setting, it is necessary to calculate the common pulse width unit of various pulses with different widths in the protocol used, that is, the greatest common divisor, fill in the lower 12 bits of the register IRTX_PULSE_WIDTH, and each pulse fills its corresponding multiple in the register IRTX_PW.

IR supports a maximum of 64-bit data bits and is divided into two 32-bit registers IRTX_DATA_WORD0 and IRTX_DATA_WORD1.

8.3.4 Pulse width transmission

For protocols that are not suitable for normal transmission mode, IR provides a pulse width transmission method. First calculate the common pulse width unit of the pulses of different widths in the protocol used, that is, the greatest common divisor, and fill in the lower 12 bits of the register IRTX_PULSE_WIDTH. Then fill the register IRTX_SWM_PW_n(0 <= n <= 7)with multiples corresponding to the respective level widths from the first high level to the last level, each level width multiple occupies 4-bit .

8.3.5 Carrier modulation

Setting the upper 16 bits of the IRTX_PULSE_WIDTH register can generate carriers with different frequencies and duty cycles. The <TXMPH1W> bit in this register sets the width of carrier phase 1, and the <TXMPH0W> bit sets the width of carrier phase 0.

8.3.6 IR interrupt

IR has separate transmit and receive interrupts, and a transmit interrupt is generated when a transmit operation ends. When a piece of data is received, it will wait for the continuous level to reach the set end threshold to generate a receive interrupt.

The user can query the send interrupt status and clear the interrupt by register IRTX_INT_STS, and query the receive interrupt status and clear the interrupt by register IRRX_INT_STS.

8.4 Register description

Name	Description
irtx_config	IR TX configuration register
irtx_int_sts	IR TX interrupt status
irtx_data_word0	IR TX data word0
irtx_data_word1	IR TX data word1
irtx_pulse_width	IR TX pulse width
irtx_pw	IR TX pulse width of phase
irtx_swm_pw_0	IR TX Software Mode pulse width data0
irtx_swm_pw_1	IR TX Software Mode pulse width data1
irtx_swm_pw_2	IR TX Software Mode pulse width data2
irtx_swm_pw_3	IR TX Software Mode pulse width data3
irtx_swm_pw_4	IR TX Software Mode pulse width data4
irtx_swm_pw_5	IR TX Software Mode pulse width data5
irtx_swm_pw_6	IR TX Software Mode pulse width data6
irtx_swm_pw_7	IR TX Software Mode pulse width data7
irrx_config	IR RX configuration register
irrx_int_sts	IR RX interrupt status
irrx_pw_config	IR RX pulse width configuration
irrx_data_count	IR RX data bit count

Name	Description
irrx_data_word0	IR RX data word0
irrx_data_word1	IR RX data word1
irrx_swm_fifo_config_0	IR RX FIFO configuration
irrx_swm_fifo_rdata	IR RX software mode pulse width data

8.4.1 irtx_config

Address：0x4000a600

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD														TXDATANU	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXDATANU				TPHLIS	TXTPEN	TXHHLI	TXHEN	RSVD	TXL1HLI	TXL0HLI	TXDAEN	TXSWEN	TXMDEN	TXOEN	TXEN

Bits	Name	Type	Reset	Description
31:18	RSVD			
17:12	TXDATANU	R/W	6'd31	Bit count of Data phase (unit: bit / PW for normal / SWM)
11	TPHLIS	R/W	1'b0	Tail pulse H/L inverse signal (Don't care if SWM is enabled) 0: Phase 0 is High (Active), phase 1 is Low (Idle) (H -> L) 1: Phase 0 is Low (Idle), phase 1 is High (Active) (L -> H)
10	TXTPEN	R/W	1'b1	Enable signal of tail pulse (Don't care if SWM is enabled)
9	TXHHLI	R/W	1'b0	Tail pulse H/L inverse signal (Don't care if SWM is enabled) 0: Phase 0 is High (Active), phase 1 is Low (Idle) (H -> L) 1: Phase 0 is Low (Idle), phase 1 is High (Active) (L -> H)
8	TXHEN	R/W	1'b1	Enable signal of head pulse (Don't care if SWM is enabled)
7	RSVD			
6	TXL1HLI	R/W	1'b0	Logic 1 H/L inverse signal (Don't care if SWM is enabled) 0: Phase 0 is High (Active), phase 1 is Low (Idle) (H -> L) 1: Phase 0 is Low (Idle), phase 1 is High (Active) (L -> H)
5	TXL0HLI	R/W	1'b0	Logic 0 H/L inverse signal (Don't care if SWM is enabled) 0: Phase 0 is High (Active), phase 1 is Low (Idle) (H -> L) 1: Phase 0 is Low (Idle), phase 1 is High (Active) (L -> H)
4	TXDAEN	R/W	1'b1	Enable signal of data phase (Don't care if SWM is enabled)
3	TXSWEN	R/W	1'b0	Enable signal of IRTX Software Mode (SWM)

Bits	Name	Type	Reset	Description
2	TXMDEN	R/W	1'b0	Enable signal of output modulation
1	TXOEN	R/W	1'b0	Output inverse signal 1'b0: Output stays at Low during idle state 1'b1: Output stays at High during idle state
0	TXEN	R/W	1'b0	Enable signal of IRTX function Asserting this bit will trigger the transaction, and should be de-asserted after finish

8.4.2 irtx_int_sts

Address：0x4000a604

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

Bits	Name	Type	Reset	Description
31:25	RSVD			
24	TXEEN	R/W	1'b1	Interrupt enable of irtx_end_int
23:17	RSVD			
16	TXECLR	W1C	1'b0	Interrupt clear of irtx_end_int
15:9	RSVD			
8	TXEMASK	R/W	1'b1	Interrupt mask of irtx_end_int
7:1	RSVD			
0	TXEINT	R	1'b0	IRTX transfer end interrupt

8.4.3 irtx_data_word0

Address：0x4000a608

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXDW0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXDW0															

Bits	Name	Type	Reset	Description
31:0	TXDW0	R/W	32'h0	TX data word 0 (Don't care if SWM is enabled)

8.4.4 irtx_data_word1

Address：0x4000a60c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXDW1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXDW1															

Bits	Name	Type	Reset	Description
31:0	TXDW1	R/W	32'h0	TX data word 1 (Don't care if SWM is enabled)

8.4.5 irtx_pulse_width

Address：0x4000a610

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXMPH1W								TXMPH0W							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				TXPWU											

Bits	Name	Type	Reset	Description
31:24	TXMPH1W	R/W	8'd34	Modulation phase 1 width
23:16	TXMPH0W	R/W	8'd17	Modulation phase 0 width
15:12	RSVD			
11:0	TXPWU	R/W	12'd1124	Pulse width unit

8.4.6 irtx_pw

Address：0x4000a614

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXTPH1W				TXTPH0W				TXHPH1W				TXHPH0W			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXL1PH1W				TXL1PH0W				TXL0PH1W				TXL0PH0WS			

Bits	Name	Type	Reset	Description
31:28	TXTPH1W	R/W	4'd0	Pulse width of tail pulse phase 1 (Don't care if SWM is enabled)
27:24	TXTPH0W	R/W	4'd0	Pulse width of tail pulse phase 0 (Don't care if SWM is enabled)
23:20	TXHPH1W	R/W	4'd7	Pulse width of head pulse phase 1 (Don't care if SWM is enabled)
19:16	TXHPH0W	R/W	4'd15	Pulse width of head pulse phase 0 (Don't care if SWM is enabled)
15:12	TXL1PH1W	R/W	4'd2	Pulse width of logic1 phase 1 (Don't care if SWM is enabled)
11:8	TXL1PH0W	R/W	4'd0	Pulse width of logic1 phase 0 (Don't care if SWM is enabled)
7:4	TXL0PH1W	R/W	4'd0	Pulse width of logic0 phase 1 (Don't care if SWM is enabled)
3:0	TXL0PH0WS	R/W	4'd0	Pulse width of logic0 phase 0 (Don't care if SWM is enabled)

8.4.7 irtx_swm_pw_0

Address：0x4000a640

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSWPW0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSWPW0															

Bits	Name	Type	Reset	Description
31:0	TXSWPW0	R/W	32'h0	IRTX Software Mode pulse width data #0 #7, each pulse is represented by 4-bit ([3:0] is the 1st pulse, [7:4] is the 2nd pulse, [11:8] is the 3rd pulse, etc)

8.4.8 irtx_swm_pw_1

Address：0x4000a644

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSWPW1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSWPW1															

Bits	Name	Type	Reset	Description
31:0	TXSWPW1	R/W	32'h0	IRTX Software Mode pulse width data #8 #15, each pulse is represented by 4-bit ([3:0] is the 1st pulse, [7:4] is the 2nd pulse, [11:8] is the 3rd pulse, etc)

8.4.9 irtx_swm_pw_2

Address：0x4000a648

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSWPW2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSWPW2															

Bits	Name	Type	Reset	Description
31:0	TXSWPW2	R/W	32'h0	IRTX Software Mode pulse width data #16 #23, each pulse is represented by 4-bit ([3:0] is the 1st pulse, [7:4] is the 2nd pulse, [11:8] is the 3rd pulse, etc)

8.4.10 irtx_swm_pw_3

Address：0x4000a64c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSWPW3															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSWPW3															

Bits	Name	Type	Reset	Description
31:0	TXSWPW3	R/W	32'h0	IRTX Software Mode pulse width data #24 #31, each pulse is represented by 4-bit ([3:0] is the 1st pulse, [7:4] is the 2nd pulse, [11:8] is the 3rd pulse, etc)

8.4.11 irtx_swm_pw_4

Address：0x4000a650

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSWPW4															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSWPW4															

Bits	Name	Type	Reset	Description
31:0	TXSWPW4	R/W	32'h0	IRTX Software Mode pulse width data #32 #39, each pulse is represented by 4-bit ([3:0] is the 1st pulse, [7:4] is the 2nd pulse, [11:8] is the 3rd pulse, etc)

8.4.12 irtx_swm_pw_5

Address：0x4000a654

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSWPW5															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSWPW5															

Bits	Name	Type	Reset	Description
31:0	TXSWPW5	R/W	32'h0	IRTX Software Mode pulse width data #40 #47, each pulse is represented by 4-bit ([3:0] is the 1st pulse, [7:4] is the 2nd pulse, [11:8] is the 3rd pulse, etc)

8.4.13 irtx_swm_pw_6

Address：0x4000a658

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSWPW6															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSWPW6															

Bits	Name	Type	Reset	Description
31:0	TXSWPW6	R/W	32'h0	IRTX Software Mode pulse width data #48 #55, each pulse is represented by 4-bit ([3:0] is the 1st pulse, [7:4] is the 2nd pulse, [11:8] is the 3rd pulse, etc)

8.4.14 irtx_swm_pw_7

Address：0x4000a65c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSWPW7															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSWPW7															

Bits	Name	Type	Reset	Description
31:0	TXSWPW7	R/W	32'h0	IRTX Software Mode pulse width data #56 #63, each pulse is represented by 4-bit ([3:0] is the 1st pulse, [7:4] is the 2nd pulse, [11:8] is the 3rd pulse, etc)

8.4.15 irrx_config

Address：0x4000a680

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

Bits	Name	Type	Reset	Description
31:12	RSVD			
11:8	RXDEGCNT	R/W	4'd0	De-glitch function cycle count
7:5	RSVD			
4	RXDGEN	R/W	1'b0	Enable signal of IRRX input de-glitch function
3:2	RXMODE	R/W	2'd0	IRRX mode 0: NEC 1: RC5 2: SW pulse-width detection mode (SWM) 3: Reserved
1	RXININV	R/W	1'b1	Input inverse signal
0	RXEN	R/W	1'b0	Enable signal of IRRX function Asserting this bit will trigger the transaction, and should be de-asserted after finish

8.4.16 irrx_int_sts

Address：0x4000a684

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

Bits	Name	Type	Reset	Description
31:25	RSVD			
24	RXEEN	R/W	1'b1	Interrupt enable of irrx_end_int
23:17	RSVD			
16	RXECLR	W1C	1'b0	Interrupt clear of irrx_end_int
15:9	RSVD			
8	RXEMASK	R/W	1'b1	Interrupt mask of irrx_end_int
7:1	RSVD			
0	RXEINT	R	1'b0	IRRX transfer end interrupt

8.4.17 irrx_pw_config

Address：0x4000a688

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

Bits	Name	Type	Reset	Description
31:16	RXETH	R/W	16'd8999	Pulse width threshold to trigger END condition
15:0	RXDATH	R/W	16'd3399	Pulse width threshold for Logic0/1 detection (Don't care if SWM is enabled)

8.4.18 irrx_data_count

Address：0x4000a690

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

Bits	Name	Type	Reset	Description
31:7	RSVD			
6:0	RXDACNT	R	7'd0	RX data bit count (pulse-width count for SWM)

8.4.19 irrx_data_word0

Address：0x4000a694

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXDAW0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDAW0															

Bits	Name	Type	Reset	Description
31:0	RXDAW0	R	32'h0	RX data word 0

8.4.20 irrx_data_word1

Address：0x4000a698

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXDAW1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDAW1															

Bits	Name	Type	Reset	Description
31:0	RXDAW1	R	32'h0	RX data word 1

8.4.21 irrx_swm_fifo_config_0

Address：0x4000a6c0

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

Bits	Name	Type	Reset	Description
31:11	RSVD			
10:4	RXFIFOCN	R	7'd0	RX FIFO available count
3	RXFUF	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
2	RXFOF	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
1	RSVD			
0	RXFCLR	W1C	1'b0	Clear signal of RX FIFO

8.4.22 irrx_swm_fifo_rdata

Address：0x4000a6c4

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	RXFRDA	R	16'h0	IRRX Software Mode pulse width data

9 SPI

9.1 SPI introduction

Serial Peripheral Interface Bus(SPI) is a synchronous serial communication interface specification for short-range communication. Devices use full-duplex mode for communication. There is a master and one or more slaves. Requires at least 4 wires, in fact 3 wires are also available (the one-way transmission), including SDI (data input), SDO (data output), SCLK (clock), CS (chip select).

9.2 SPI main features

- Can be used as SPI master or SPI slave
- The transmit and receive channels each have a FIFO with a depth of 4 words
- Both master and slave devices support 4 clock formats(CPOL,CPHA)
- Both master and slave devices support 1/2/3/4 byte transmission mode
- Flexible clock configuration, support up to 40M clock
- Configurable MSB/LSB priority transmission
- Acceptance filtering function
- Timeout mechanism under the slave
- Support DMA transfer mode

9.3 SPI function description

9.3.1 Clock control

According to different clock phases and polarity settings, the SPI clock has four modes, which can be set by bit4 (CPOL) and bit5 (CPHA) of the SPI_CONFIG register. CPOL is used to determine the level of the SCK clock signal

when idle, CPOL = 0 means the idle level is low, and CPOL = 1 means the idle level is high. CPHA is used to determine the sampling time. CPHA = 0 samples on the first clock edge of each cycle, and CPHA = 1 samples on the second clock edge of each cycle.

By setting registers SPI_PRD_0 and SPI_PRD_1, you can also adjust the start and end level duration of the clock, the time of phase 0/1, and the interval between each frame of data. The specific settings in the four modes are shown below:

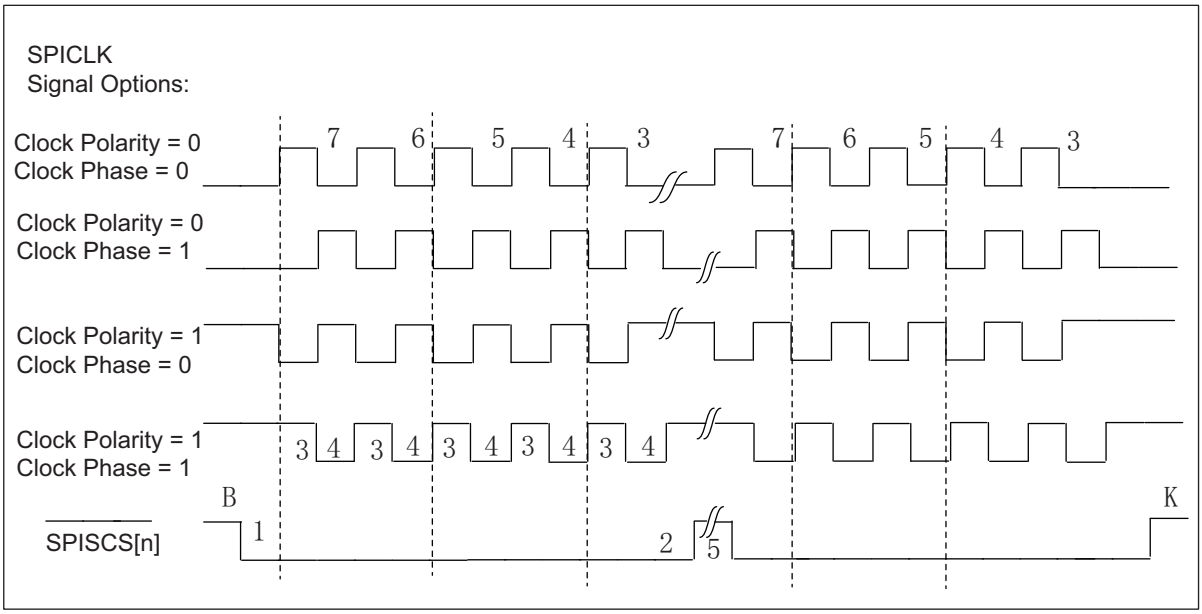

Figure 9.1: SPI clock

The meaning of each number is as follows: 1 is the length of the start condition, 2 is the length of the stop condition, 3 is the length of phase 0, 4 is the length of phase 1, and 5 is the interval between each frame of data.

9.3.2 Master continuous transmission mode

When this mode is enabled, the CS signal will not be released when the current data is transmitted and there is still data available in the FIFO.

9.3.3 Acceptance filtering function

By setting the start and end bits that need to be filtered out, the SPI discards the corresponding data segment in the received data. As shown below:

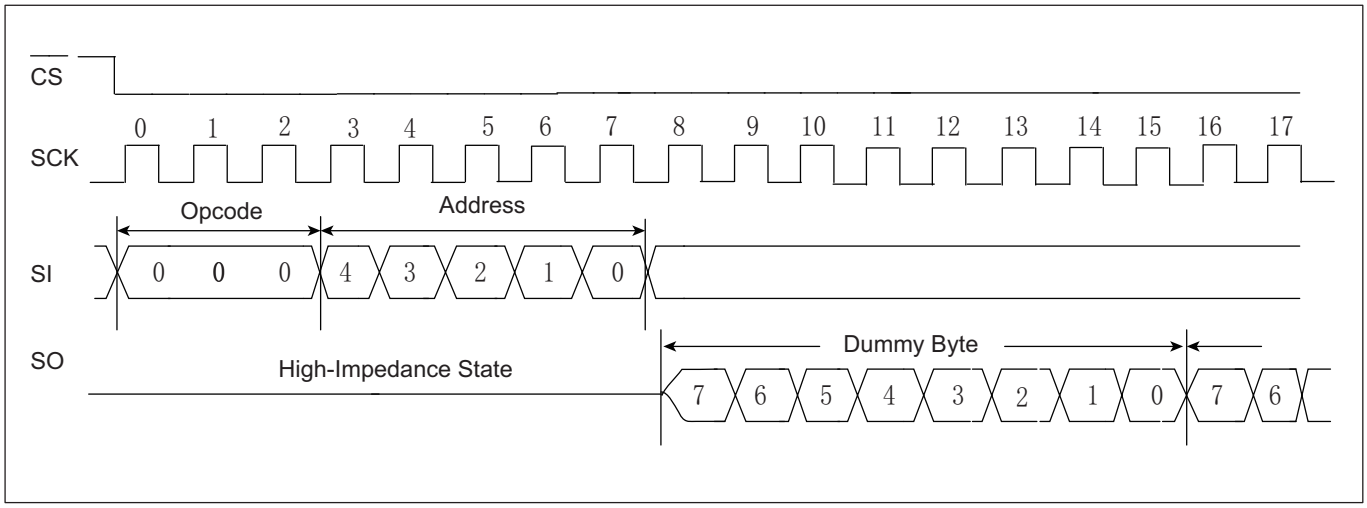

Figure 9.2: SPI Ignore waveform

In the figure above, the start bit of the filter is set to 0, the end bit is set to 7, the dummy byte is received, and the end bit is set to 15, the dummy byte is discarded.

9.3.4 Receive error correction

By enabling this function and setting the threshold, the SPI will discard data that does not reach the threshold width.

9.3.5 Slave mode timeout mechanism

By setting a timeout threshold, an interrupt will be triggered when the SPI does not receive a clock signal after exceeding this time value in slave mode.

9.3.6 I/O transfer mode

The chip communications processor can perform FIFO fill and empty operations in response to interrupts from the FIFO. Each FIFO has a programmable FIFO trigger threshold to trigger interrupts. When the value in the RX FIFO exceeds the RX FIFO trigger threshold in the SPI controller 1, an interrupt will be generated and a signal will be sent to the chip communication processor to clear the RX FIFO. When the value in the TX FIFO is less than or equal to the TX FIFO trigger threshold in the SPI control register 1 plus 1, an interrupt will be generated and a signal will be sent to the chip communication processor to refill the TX FIFO.

Query the SPI status register to determine the sampled value in the FIFO and the status of the FIFO. Software is responsible for ensuring the correct RX FIFO trigger threshold and TX FIFO trigger threshold to prevent receive FIFO overrun and transmit FIFO underrun.

9.3.7 DMA transfer mode

SPI supports DMA transfer mode. The use of this mode requires the TX and RX FIFO thresholds to be set separately. When this mode is enabled, the UART will check the TX / RX FIFO. Once the TX / RX FIFO available count value

is greater than its set threshold, a DMA request will be initiated , DMA will move data to TX FIFO or out of RX FIFO according to the setting.

9.3.8 SPI interrupt

SPI has a variety of interrupt control, including the following interrupt modes:

- SPI transfer end interrupt
- TX FIFO request interrupt
- RX FIFO request interrupt
- Slave mode transfer timeout interrupt
- Slave mode TX overload interrupt
- TX / RX FIFO overflow interrupt

In master mode, the SPI transfer end interrupt is triggered at the end of each frame of data transfer; in slave mode, the SPI transfer end interrupt is triggered when the CS signal is released. The TX / RX FIFO request interrupt will be triggered when its available FIFO count is greater than its set threshold. When the condition is not met, the interrupt flag will be automatically cleared. Slave mode transmission timeout interrupt is triggered when the threshold is exceeded in slave mode and no clock signal is received. If the TX / RX FIFO overflows or underflows, the TX / RX FIFO overflow interrupt will be triggered. When the FIFO clear bit TFC / RFC is set to 1, the corresponding FIFO will be cleared and the overflow interrupt flag will be automatically cleared.

Query the interrupt status through register SPI_INT_STS and write 1 to the corresponding bit to clear the interrupt.

9.4 Register description

Name	Description
spi_config	SPI configuration register
spi_int_sts	SPI interrupt status
spi_bus_busy	SPI bus busy
spi_prd_0	SPI length control register
spi_prd_1	SPI length of interval
spi_rxd_ignr	SPI ingnore function
spi_sto_value	SPI time-out value
spi_fifo_config_0	SPI FIFO configuration register0
spi_fifo_config_1	SPI FIFO configuration register1
spi_fifo_wdata	SPI FIFO write data

Name	Description
spi_fifo_rdata	SPI FIFO read data

9.4.1 spi_config

Address：0x4000a200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEGCNT				DEG EN	RSVD	MCEN	IGNR EN	BYTE INV	BIT INV	SCLK PH	SCLK POL	FSIZE		SEN	MEN

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:12	DEGCNT	R/W	4’d0	De-glitch function cycle count
11	DEGEN	R/W	1’b0	Enable signal of all input de-glitch function
10	RSVD			
9	MCEN	R/W	1’b0	Enable signal of master continuous transfer mode 1’b0: Disabled, SS_n will de-assert between each data frame 1’b1: Enabled, SS_n will stay asserted between each consecutive data frame if the next data is valid in the FIFO
8	IGNREN	R/W	1’b0	Enable signal of RX data ignore function
7	BYTEINV	R/W	1’b0	Byte-inverse signal for each FIFO entry data 0: Byte[0] is sent out first 1: Byte[3] is sent out first
6	BITINV	R/W	1’b0	Bit-inverse signal for each data byte 0: Each byte is sent out MSB-first 1: Each byte is sent out LSB-first
5	SCLKPH	R/W	1’b0	SCLK clock phase inverse signal
4	SCLKPOL	R/W	1’b0	SCLK polarity 0: SCLK output LOW at IDLE state 1: SCLK output HIGH at IDLE state

Bits	Name	Type	Reset	Description
3:2	FSIZE	R/W	2'd0	SPI frame size (also the valid width for each FIFO entry) 2'd0: 8-bit 2'd1: 16-bit 2'd2: 24-bit 2'd3: 32-bit
1	SEN	R/W	1'b0	Enable signal of SPI Slave function, Master and Slave should not be both enabled at the same time (This bit becomes don't-care if cr_spi_m_en is enabled)
0	MEN	R/W	1'b0	Enable signal of SPI Master function Asserting this bit will trigger the transaction, and should be de-asserted after finish

9.4.2 spi_int_sts

Address：0x4000a204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		FER EN	TXU EN	STO EN	RXF EN	TXF EN	END EN	RSVD			TXU CLR	STO CLR	RSVD		END CLR
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		FER MASK	TXU MASK	STO MASK	RXF MASK	TXF MASK	END MASK	RSVD		FER INT	TXU INT	STO INT	RXF INT	TXF INT	END INT

Bits	Name	Type	Reset	Description
31:30	RSVD			
29	FEREN	R/W	1'b1	Interrupt enable of spi_fer_int
28	TXUEN	R/W	1'b1	Interrupt enable of spi_txu_int
27	STOEN	R/W	1'b1	Interrupt enable of spi_sto_int
26	RXFEN	R/W	1'b1	Interrupt enable of spi_rxv_int
25	TXFEN	R/W	1'b1	Interrupt enable of spi_txe_int
24	ENDEN	R/W	1'b1	Interrupt enable of spi_end_int
23:21	RSVD			
20	TXUCLR	W1C	1'b0	Interrupt clear of spi_txu_int
19	STOCLR	W1C	1'b0	Interrupt clear of spi_sto_int
18:17	RSVD			
16	ENDCLR	W1C	1'b0	Interrupt clear of spi_end_int

Bits	Name	Type	Reset	Description
15:14	RSVD			
13	FERMASK	R/W	1'b1	Interrupt mask of spi_fer_int
12	TXUMASK	R/W	1'b1	Interrupt mask of spi_txu_int
11	STOMASK	R/W	1'b1	Interrupt mask of spi_sto_int
10	RXFMASK	R/W	1'b1	Interrupt mask of spi_rxv_int
9	TXFMASK	R/W	1'b1	Interrupt mask of spi_txe_int
8	ENDMASK	R/W	1'b1	Interrupt mask of spi_end_int
7:6	RSVD			
5	FERINT	R	1'b0	SPI TX/RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
4	TXUINT	R	1'b0	SPI slave mode TX underrun error flag, triggered when TXD is not ready during transfer in slave mode
3	STOINT	R	1'b0	SPI slave mode transfer time-out interrupt, triggered when SPI bus is idle for a given value
2	RXFINT	R	1'b0	SPI RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto-cleared when data is popped
1	TXFINT	R	1'b0	SPI TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed
0	ENDINT	R	1'b0	SPI transfer end interrupt, shared by both master and slave mode Master mode: Triggered when the final frame is transferred Slave mode: Triggered when CS_n is de-asserted

9.4.3 spi_bus_busy

Address：0x4000a208

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

Bits	Name	Type	Reset	Description
31:1	RSVD			
0	BUSBUSY	R	1'b0	Indicator of SPI bus busy

9.4.4 spi_prd_0

Address：0x4000a210

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRDPH1								PRDPH0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRDP								PRDS							

Bits	Name	Type	Reset	Description
31:24	PRDPH1	R/W	8'd15	Length of DATA phase 1 (please refer to "Timing" tab)
23:16	PRDPH0	R/W	8'd15	Length of DATA phase 0 (please refer to "Timing" tab)
15:8	PRDP	R/W	8'd15	Length of STOP condition (please refer to "Timing" tab)
7:0	PRDS	R/W	8'd15	Length of START condition (please refer to "Timing" tab)

9.4.5 spi_prd_1

Address：0x4000a214

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	PRDI	R/W	8'd15	Length of INTERVAL between frame (please refer to "Timing" tab)

9.4.6 spi_rxd_ignr

Address：0x4000a218

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

Bits	Name	Type	Reset	Description
31:21	RSVD			
20:16	RXDIGS	R/W	5'd0	Starting point of RX data ignore function
15:5	RSVD			
4:0	RXDIGP	R/W	5'd0	Stopping point of RX data ignore function

9.4.7 spi_sto_value

Address：0x4000a21c

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

Bits	Name	Type	Reset	Description
31:12	RSVD			
11:0	STOV	R/W	12'hFFF	Time-out value for spi_sto_int triggering

9.4.8 spi_fifo_config_0

Address：0x4000a280

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								RFUF	RFOF	TFUF	TFOF	RFC	TFC	DMAREN	DMATEN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	RFUF	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	RFOF	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	TFUF	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	TFOF	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	RFC	W1C	1'b0	Clear signal of RX FIFO

Bits	Name	Type	Reset	Description
2	TFC	W1C	1'b0	Clear signal of TX FIFO
1	DMAREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface
0	DMATEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface

9.4.9 spi_fifo_config_1

Address：0x4000a284

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

Bits	Name	Type	Reset	Description
31:26	RSVD			
25:24	RFTH	R/W	2'd0	RX FIFO threshold, dma_rx_req will not be asserted if tx_-fifo_cnt is less than this value
23:18	RSVD			
17:16	TFTH	R/W	2'd0	TX FIFO threshold, dma_tx_req will not be asserted if tx_-fifo_cnt is less than this value
15:11	RSVD			
10:8	RFCNT	R	3'd0	RX FIFO available count
7:3	RSVD			
2:0	TFCNT	R	3'd4	TX FIFO available count

9.4.10 spi_fifo_wdata

Address：0x4000a288

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

Bits	Name	Type	Reset	Description
31:0	FWDATA	W	x	SPI FIFO write data

9.4.11 spi_fifo_rdata

Address：0x4000a28c

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

Bits	Name	Type	Reset	Description
31:0	FRDATA	R	32'h0	SPI FIFO read data

10 UART

10.1 UART introduction

Universal Asynchronous Receiver / Transmitter (commonly known as UART) is an asynchronous transceiver that provides a flexible way to exchange full-duplex data with external devices.

BL702 has two sets of UART ports (UART0 and UART1). By using with DMA, you can achieve efficient data communication.

10.2 UART main features

- Full-duplex asynchronous communication
- Data bit length can be selected from 5/6/7/8 bits
- Stop bit length can be selected from 0.5/1/1.5/2 bits
- Supports odd/even/no parity bits
- Detects wrong start bit
- Support LIN protocol (receive and send REAK/SYNC)
- Multiple interrupt control
- Support hardware flow control (RTS / CTS)
- Convenient baud rate programming
- Configurable MSB / LSB priority transmission
- Normal / fixed character automatic baud rate detection
- 32-byte transmit / receive FIFO
- Support DMA transfer mode

10.3 UART function description

10.3.1 Data format description

Normal UART communication data is composed of a start bit, a data bit, a parity bit, and a stop bit. The BL702's UART supports configurable data bits, parity bits, and stop bits, all of which are set in the UTX_CONFIG and URX_CONFIG registers. The waveform of one frame of data is shown below:

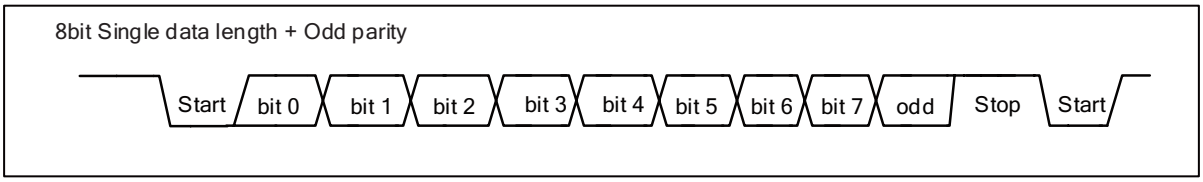

Figure 10.1: UART data

The start bit of a data frame occupies 1-bit, and the stop bit can be configured to be 0.5 / 1 / 1.5 / 2 bits wide by configuring <TXBCNTP> and <CR_URX_BIT_CNT_P>. The start bit is low and the stop bit is high.

The data bit width can be configured to 5/6/7/8 bit width by <TXBCNTD> and <RXBCNTD>.

When <TXPREN> and <RXPREN> are set, the data frame adds a parity bit after the data. <TXPRSEL> and <RXPRSEL> are used to select odd or even parity. When the receiver detects a parity error in the input data, a parity error interrupt is generated.

Odd parity calculation method: If the current data bit 1 is an odd number, the odd parity bit is 0; otherwise, it is 1.

Calculation method of even parity: If the number of current data bit 1 is odd, even parity is 1; otherwise it is 0.

10.3.2 Clock source

The UART has two clock sources: 96MHz PLL_CLK and FCLK. The frequency divider in the clock is used to divide the clock source and then generate a clock signal to drive the UART module. As shown below:

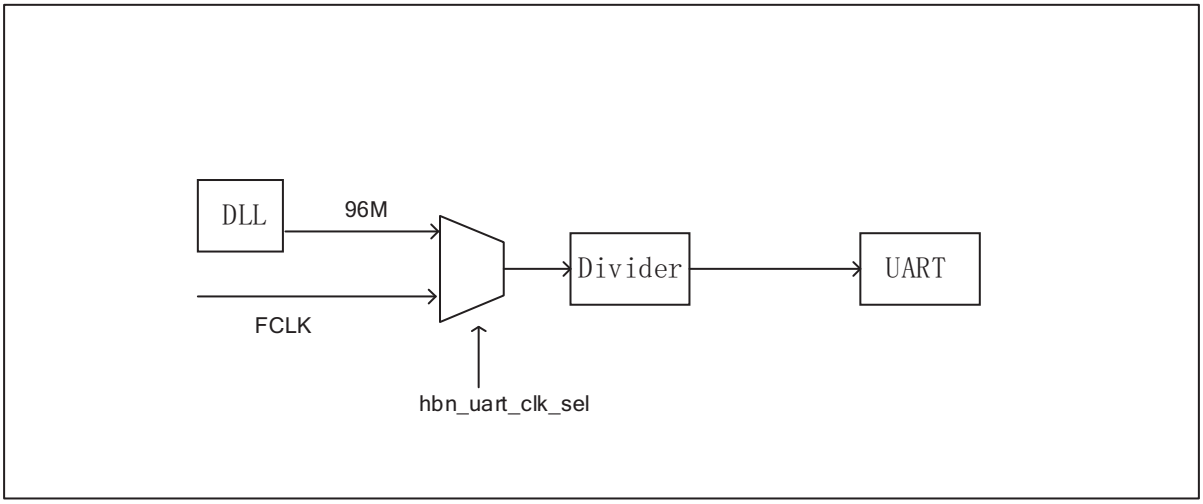

Figure 10.2: UART clock

10.3.3 Baud rate setting

he user can generate the required baud rate by setting the register UART_BIT_PRD. The upper 16 bits and lower 16 bits of this register correspond to RX and TX respectively, that is, the baud rates of RX and TX can be set independently. The 16-bit value needs It is calculated that the formula is as follows:

Baud rate = UART clock / (16-bit coefficient + 1)

That is: 16-bit coefficient = UART clock / baud rate -1

The meaning of the 16-bit coefficient is the count value obtained by counting the current baud rate bit width with the UART clock. Because the maximum 16-bit coefficient is 65535, the minimum baud rate supported by the UART is: UART clock / 65536. The maximum baud rate supported by the UART is 10Mbps.

Before the UART samples the data, it will first filter the data to filter out the glitches in the waveform. Sampling is then performed at the intermediate value of the above 16-bit coefficients, so that different sampling times are adjusted according to different baud rates to keep the median value always being taken, greatly improving flexibility and accuracy. The sampling process is shown in the following figure:

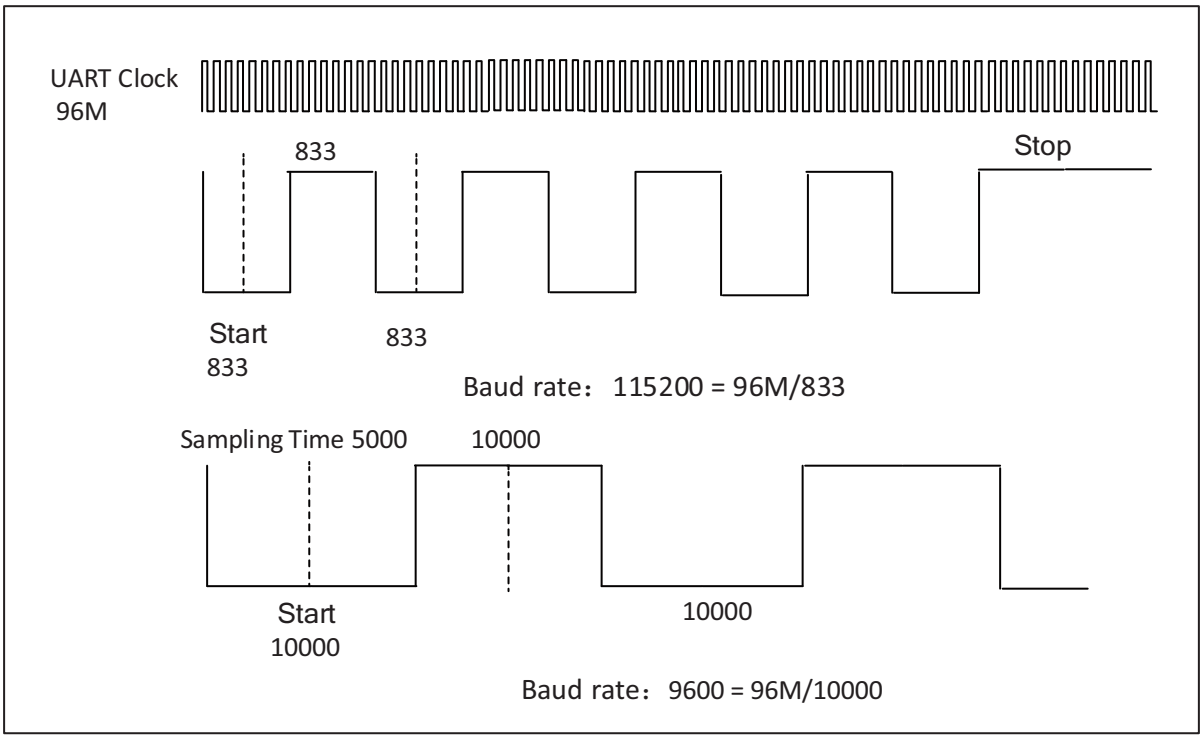

Figure 10.3: UART sampling waveform

10.3.4 Transmitter

The transmitter contains a 128-byte transmit FIFO to store the data to be transmitted. Software can write TX FIFO through APB bus, and can also move data into TX FIFO through DMA. When the transmit enable bit is set, the data stored in the FIFO will be output from the TX pin. Software can choose to transfer data to TX FIFO through DMA or APB bus. The software can check the status of the transmitter by querying the count value of the remaining available space in the TX FIFO through the bit <TFICNT> of the register UART_FIFO_CONFIG_1. The FreeRun mode of the transmitter is as follows: - If the FreeRun mode is not turned on, the sending behavior will be terminated and an interrupt will be generated when the sent byte reaches the specified length. If you want to continue sending, you need to turn off and then enable the send enable bit. - If the FreeRun mode is turned on, the transmitter will transmit when there is data in the TX FIFO, and will not terminate because the transmitted byte reaches the specified length.

10.3.5 Receiver

The receiver contains a 128-byte receive FIFO to store the received data. The software can check the status of the receiver by querying the count value of the available data in the RX FIFO through the bit <RFICNT> of the register UART_FIFO_CONFIG_1. The lower 8 bits of the URX_RTO_TIMER register are used to set a receiving timeout threshold. When the receiver does not receive data for more than this time value, it will trigger an interrupt. The bits <DEGEN> and <DEGCNT> of the URX_CONFIG register are used to enable the deburring function and set the threshold. It controls the filtering part before the UART sampling, and the UART will filter out the burrs whose width is lower than the threshold in the waveform . Then send it to sampling.

10.3.6 Automatic baud rate detection

The UART module supports automatic baud rate detection. The detection is divided into two types, one is general mode and the other is fixed character mode. The two detection modes will be enabled every time the bit <ABREN> of the register URX_CONFIG is turned on.

General mode

For any character data received, the UART module will count the number of clocks in the bit width. This number will then be written into the lower 16 bits of the register STS_URX_ABR_PRD and used to calculate the baud rate. Therefore, when the first received data bit is 1, the correct baud rate can be obtained. Such as '0x01' under LSB-FIRST.

Fixed character mode

In this mode, after counting the number of clocks in the starting bit width, the UART module will continue to count the number of clocks of subsequent data bits and compare with the start bit. If it fluctuates within the allowable error range, it will pass the test, otherwise the count value will be discarded. Therefore, only when the fixed characters '0x55'/'0xD5' under LSB-FIRST or '0xAA'/'0xAB' under MSB-FIRST are received. The UART module will write the count value of the number of clocks in the starting bit width into the upper 16 bits of the register STS_URX_ABR_PRD. As shown below:

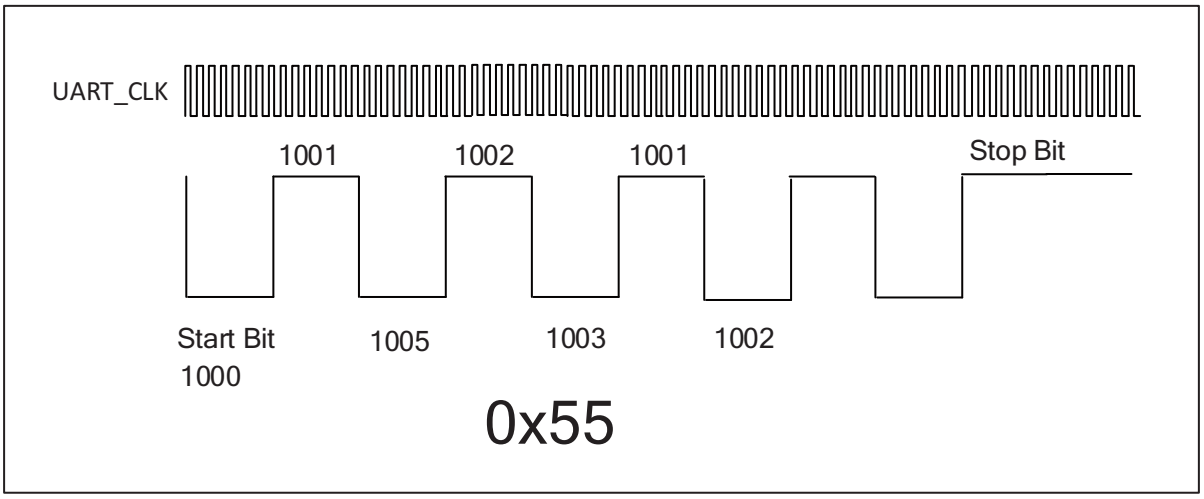

Figure 10.4: UART fixed character mode waveform

For an unknown baud rate, the UART uses UART_CLK to count the start bit width of 1000, and the second bit width of 1001. If there is no more than 4 UART_CLK floating up and down from the previous bit width, the UART will continue to count the third bit, which is 1005. If the difference with the start bit exceeds 4, the detection fails and the data is discarded. The UART will sequentially compare the first 6 bits of the data bit with the start bit.

The formula for calculating the detected baud rate is as follows: Baud rate=source clock/(16-bit detection value+1)

10.3.7 Hardware flow control

The UART supports hardware flow control in CTS / RTS mode to prevent data in the FIFO from being lost because it is too late to process. The hardware flow control connection is shown in the following figure:

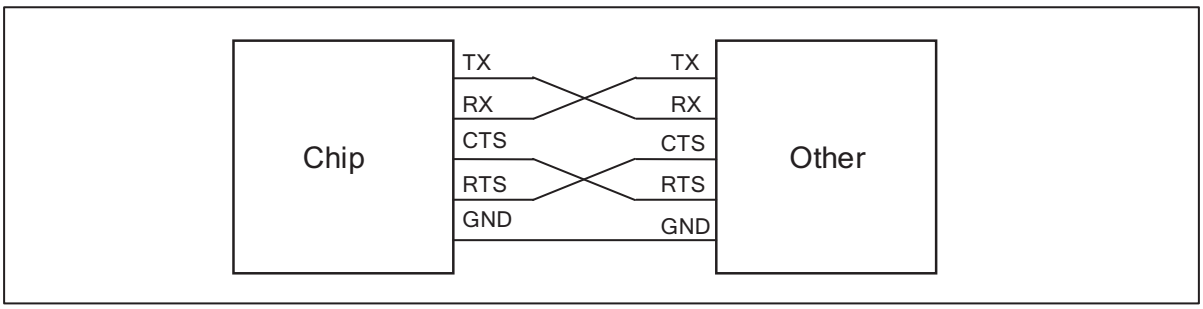

Figure 10.5: UART flow control

When using the hardware flow control function, the output signal RTS is low to indicate that the other party is requested to send data, and RTS is high to indicate that the other party is notified to suspend data transmission until RTS is restored to low level. There are two ways to control the hardware flow of the transmitter.

- The bit <RTSSWM> of the URX_CONFIG register is equal to 0: when the remaining space in the RX FIFO is less than or equal to 1 byte, the RTS level is raised.
- The bit <RTSSWM> of the URX_CONFIG register is equal to 1: The bit <RTSSWV> of the configuration register URX_CONFIG can be used to change the RTS level.

The TX CTS can be enabled by configuring the bit <CTSEN> of the UTX_CONFIG register. When the device detects that the input signal CTS is pulled high, TX will stop sending data until it detects that CTS is pulled low before continuing to send.

10.3.8 LIN transmission mode

When the transmitter needs to use the LIN transmission mode, it can send the BREAK field and the SYNC field by configuring <TXLINEN>. The width of the interval field is determined by <TXBCNTB>.

When the receiver needs to use the LIN transmission mode, you can configure <RXLINEN> to detect the interval field and the synchronization field, and trigger the corresponding interrupt <RLSEINT> when the format of the synchronization field is wrong.

10.3.9 DMA transfer mode

UART supports DMA transfer mode. To use this mode, you need to set the thresholds of TX and RX FIFO through the bits <TFITH> and <RFITH> of the register UART_FIFO_CONFIG_1. When this mode is enabled, the UART will check the TX/RX FIFO. Once the TX/RX FIFO available count value is greater than its set threshold, a DMA request will be initiated, and the DMA will move the data to the TX FIFO or out of the RX FIFO according to the setting.

10.3.10 UART interrupt

UART has a wealth of interrupt control, including the following interrupt modes:

- TX transmission end interrupt
- RX transmission end interrupt
- TX FIFO request interrupt
- RX FIFO request interrupt
- RX timeout interrupt
- RX parity error interrupt
- TX FIFO overflow interrupt
- RX FIFO overflow interrupt
- RX LIN mode synchronization field (SYNC Field) error interrupt

TX and RX can respectively set a transmission length value through the upper 16 bits of the UTX_CONFIG and URX_CONFIG registers. When the number of bytes transmitted reaches this value, the corresponding TX/RX transmission end interrupt will be triggered.

The TX/RX FIFO request interrupt will be triggered when the available FIFO count value is greater than the threshold set in the register UART_FIFO_CONFIG_1, and the interrupt flag will be automatically cleared when the condition is not met.

The RX timeout interrupt will be triggered when the receiver exceeds the timeout threshold without receiving data, and the RX parity error interrupt will occur when a parity error occurs. If the TX/RX FIFO overflows or underflows, the corresponding overflow interrupt will be triggered.

When the FIFO clear bit TFICLR/RFICLR is set to 1, the corresponding FIFO will be cleared and the overflow interrupt flag will be automatically cleared. When the LIN mode is enabled, the synchronization field (SYNC Field) should be 0x55 according to the protocol. Therefore, when the received data is not 0x55, the synchronization field error interrupt is triggered.

The interrupt status can be inquired through the register UART_INT_STS, and the interrupt can be cleared by writing 1 to the corresponding bit of the register UART_INT_CLEAR.

10.4 Register description

Name	Description
utx_config	UART TX configuration register
urx_config	UART RX configuration register

Name	Description
uart_bit_prd	UART period control register
data_config	UART data configuration register
utx_ir_position	UART TX ir position control register
urx_ir_position	UART RX ir position control register
urx_rto_timer	RTO interrupt control register
uart_sw_mode	UART SW mode configuration register
uart_int_sts	UART interrupt status
uart_int_mask	UART interrupt mask
uart_int_clear	UART interrupt clear
uart_int_en	UART interrupt enable
uart_status	UART status control register
sts_urx_abr_prd	Auto baud detection control register
uart_fifo_config_0	UART FIFO configuration register0
uart_fifo_config_1	UART FIFO configuration register1
uart_fifo_wdata	UART FIFO write data
uart_fifo_rdata	UART FIFO read data

10.4.1 utx_config

Address：0x4000a000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXLEN															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXBCNTP			TXBCNTP		TXBCNTD			IRTX INV	IRTX EN	TXPR SEL	TXPR EN	TLIN EN	FRM EN	CTS EN	EN

Bits	Name	Type	Reset	Description
31:16	TXLEN	R/W	16'd0	Length of UART TX data transfer (Unit: character/byte) (Don't-care if cr_utx_frm_en is enabled)

Bits	Name	Type	Reset	Description
15:13	TXBCNTP	R/W	3'd4	UART TX BREAK bit count (for LIN protocol) Note: Additional 8 bit times will be added since LIN Break field requires at least 13 bit times
12:11	TXBCNTP	R/W	2'd1	UART TX STOP bit count (unit: 0.5 bit)
10:8	TXBCNTD	R/W	3'd7	UART TX DATA bit count for each character
7	IRTXINV	R/W	1'b0	Inverse signal of UART TX output in IR mode
6	IRTXEN	R/W	1'b0	Enable signal of UART TX IR mode
5	TXPRSEL	R/W	1'b0	Select signal of UART TX parity bit 1: Odd parity 0: Even parity
4	TXPREN	R/W	1'b0	Enable signal of UART TX parity bit
3	TLINEN	R/W	1'b0	Enable signal of UART TX LIN mode (LIN header will be sent before sending data)
2	FRMEN	R/W	1'b0	Enable signal of UART TX freerun mode (utx_end_int will be disabled)
1	CTSEN	R/W	1'b0	Enable signal of UART TX CTS flow control function
0	EN	R/W	1'b0	Enable signal of UART TX function Asserting this bit will trigger the transaction, and should be de-asserted after finish

10.4.2 urx_config

Address：0x4000a004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXLEN															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEGCNT				DEG EN	RXBCNTD			IRRX INV	IRRX EN	RXPR SEL	RXPR EN	RLIN EN	RSVD	ABR EN	EN

Bits	Name	Type	Reset	Description
31:16	RXLEN	R/W	16'd0	Length of UART RX data transfer (Unit: character/byte) urx_end_int will assert when this length is reached
15:12	DEGCNT	R/W	4'd0	De-glitch function cycle count
11	DEGEN	R/W	1'b0	Enable signal of RXD input de-glitch function
10:8	RXBCNTD	R/W	3'd7	UART RX DATA bit count for each character

Bits	Name	Type	Reset	Description
7	IRRXINV	R/W	1'b0	Inverse signal of UART RX input in IR mode
6	IRRXEN	R/W	1'b0	Enable signal of UART RX IR mode
5	RXPRSEL	R/W	1'b0	Select signal of UART RX parity bit 1: Odd parity 0: Even parity
4	RXPREN	R/W	1'b0	Enable signal of UART RX parity bit
3	RLINEN	R/W	1'b0	Enable signal of UART RX LIN mode (LIN header will be required and checked before receiving data)
2	RSVD			
1	ABREN	R/W	1'b0	Enable signal of UART RX Auto Baud Rate detection function
0	EN	R/W	1'b0	Enable signal of UART RX function

10.4.3 uart_bit_prd

Address：0x4000a008

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

Bits	Name	Type	Reset	Description
31:16	RBITPRD	R/W	16'd255	Period of each UART RX bit, related to baud rate
15:0	TBITPRD	R/W	16'd255	Period of each UART TX bit, related to baud rate

10.4.4 data_config

Address：0x4000a00c

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

Bits	Name	Type	Reset	Description
31:1	RSVD			
0	BITINV	R/W	1'b0	Bit-inverse signal for each data byte 0: Each byte is sent out LSB-first 1: Each byte is sent out MSB-first

10.4.5 utx_ir_position

Address：0x4000a010

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

Bits	Name	Type	Reset	Description
31:16	TXIRPP	R/W	16'd159	STOP position of UART TX IR pulse
15:0	TXIRPS	R/W	16'd112	START position of UART TX IR pulse

10.4.6 urx_ir_position

Address：0x4000a014

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	RXIRPS	R/W	16'd111	START position of UART RXD pulse recovered from IR signal

10.4.7 urx_rto_timer

Address：0x4000a018

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	RXRTOVA	R/W	8'd15	Time-out value for triggering RTO interrupt (unit: bit time)

10.4.8 uart_sw_mode

Address：0x4000a01c

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3	RRV	R/W	1'b0	UART RX RTS output SW control value
2	RRM	R/W	1'b0	UART RX RTS output SW control mode
1	TTV	R/W	1'b0	UART TX TXD output SW control value
0	TTM	R/W	1'b0	UART TX TXD output SW control mode

10.4.9 uart_int_sts

Address：0x4000a020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							RLSE INT	RFER INT	TFIN	RPCE INT	RRTO INT	RFIN	TFIN	REIN	TEIN

Bits	Name	Type	Reset	Description
31:9	RSVD			

Bits	Name	Type	Reset	Description
8	RLSEINT	R	1'b0	UART RX LIN mode sync field error interrupt
7	RFERINT	R	1'b0	UART RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
6	TFIN	R	1'b0	UART TX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
5	RPCEINT	R	1'b0	UART RX parity check error interrupt
4	RRTOINT	R	1'b0	UART RX Time-out interrupt
3	RFIN	R	1'b0	UART RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto-cleared when data is popped
2	TFIN	R	1'b0	UART TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed
1	REIN	R	1'b0	UART RX transfer end interrupt (set according to cr_urx_-len)
0	TEIN	R	1'b0	UART TX transfer end interrupt (set according to cr_utx_-len)

10.4.10 uart_int_mask

Address：0x4000a024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							RLSE MASK	RFER MASK	TFER MASK	RPCE MASK	RRTO MASK	RFMS	TFMS	REMS	TEMS

Bits	Name	Type	Reset	Description
31:9	RSVD			
8	RLSEMASK	R/W	1'b1	Interrupt mask of urx_lse_int
7	RFERMASK	R/W	1'b1	Interrupt mask of urx_fer_int
6	TFERMASK	R/W	1'b1	Interrupt mask of utx_fer_int
5	RPCEMASK	R/W	1'b1	Interrupt mask of urx_pce_int
4	RRTOMASK	R/W	1'b1	Interrupt mask of urx_rto_int
3	RFMS	R/W	1'b1	Interrupt mask of urx_fifo_int
2	TFMS	R/W	1'b1	Interrupt mask of utx_fifo_int

Bits	Name	Type	Reset	Description
1	REMS	R/W	1'b1	Interrupt mask of urx_end_int
0	TEMS	R/W	1'b1	Interrupt mask of utx_end_int

10.4.11 uart_int_clear

Address：0x4000a028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							RLSE CLR	RSVD		RPCE CLR	RRTO CLR	RSVD		RECL	TECL

Bits	Name	Type	Reset	Description
31:9	RSVD			
8	RLSECLR	W1C	1'b0	Interrupt clear of urx_lse_int
7:6	RSVD			
5	RPCECLR	W1C	1'b0	Interrupt clear of urx_pce_int
4	RRTOCLR	W1C	1'b0	Interrupt clear of urx_rto_int
3:2	RSVD			
1	RECL	W1C	1'b0	Interrupt clear of urx_end_int
0	TECL	W1C	1'b0	Interrupt clear of utx_end_int

10.4.12 uart_int_en

Address：0x4000a02c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							RLSE	RFER	TFER	RPCE	RRTO	RFIF	TFIF	REND	TEND

Bits	Name	Type	Reset	Description
31:9	RSVD			
8	RLSE	R/W	1'b1	Interrupt enable of urx_lse_int

Bits	Name	Type	Reset	Description
7	RFER	R/W	1'b1	Interrupt enable of urx_fer_int
6	TFER	R/W	1'b1	Interrupt enable of utx_fer_int
5	RPCE	R/W	1'b1	Interrupt enable of urx_pce_int
4	RRTO	R/W	1'b1	Interrupt enable of urx_rto_int
3	RFIF	R/W	1'b1	Interrupt enable of urx_fifo_int
2	TFIF	R/W	1'b1	Interrupt enable of utx_fifo_int
1	REND	R/W	1'b1	Interrupt enable of urx_end_int
0	TEND	R/W	1'b1	Interrupt enable of utx_end_int

10.4.13 uart_status

Address：0x4000a030

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

Bits	Name	Type	Reset	Description
31:2	RSVD			
1	RBB	R	1'b0	Indicator of UART RX bus busy
0	TBB	R	1'b0	Indicator of UART TX bus busy

10.4.14 sts_urx_abr_prd

Address：0x4000a034

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

Bits	Name	Type	Reset	Description
31:16	ABRPRD	R	16'd0	Bit period of Auto Baud Rate detection using codeword 0x55

Bits	Name	Type	Reset	Description
15:0	ABRPRDS	R	16'd0	Bit period of Auto Baud Rate detection using START bit

10.4.15 uart_fifo_config_0

Address：0x4000a080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								RFIU	RFIO	TFIU	TFIO	RFI CLR	TFI CLR	UDR EN	UDT EN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	RFIU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	RFIO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	TFIU	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	TFIO	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	RFICLR	W1C	1'b0	Clear signal of RX FIFO
2	TFICLR	W1C	1'b0	Clear signal of TX FIFO
1	UDREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface
0	UDTEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface

10.4.16 uart_fifo_config_1

Address：0x4000a084

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

Bits	Name	Type	Reset	Description
31	RSVD			

Bits	Name	Type	Reset	Description
30:24	RFITH	R/W	7'd0	RX FIFO threshold, dma_rx_req will not be asserted if tx_-fifo_cnt is less than this value
23	RSVD			
22:16	TFITH	R/W	7'd0	TX FIFO threshold, dma_tx_req will not be asserted if tx_-fifo_cnt is less than this value
15:8	RFICNT	R	8'd0	RX FIFO available count
7:0	TFICNT	R	8'd128	TX FIFO available count

10.4.17 uart_fifo_wdata

Address：0x4000a088

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	UFIWD	W	x	UART FIFO write data

10.4.18 uart_fifo_rdata

Address：0x4000a08c

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	UFIRD	R	8'h0	UART FIFO read data

11 I2C

11.1 I2C introduction

I2C (Inter-Intergrated Circuit) is a serial communication bus that uses a multi-master-slave architecture to connect low-speed peripheral devices.

Each device has a unique address identification and can be used as a transmitter or receiver. Each device connected to the bus can set the address by software with a unique address and the always-receiving master-slave relationship. The host can be used as a host transmitter or a host receiver.

If two or more hosts are initialized at the same time, data transmission can prevent data from being destroyed through collision detection and arbitration.

BL702 includes an I2C controller host, which can be flexibly configured with slaveAddr, subAddr, and data transmission to facilitate communication with slave devices. It provides 2 word depth fifo and provides interrupt functions. It can be used with DMA to improve efficiency and flexibly adjust clock frequency.

11.2 I2C main features

- Support host mode
- Support multi-master mode and arbitration function
- Flexible clock frequency adjustment

11.3 I2C function description

Table 11.1: Pin lists

Name	Type	Description
I2Cx_SCL	input/output	I2C serial clock signal
I2Cx_SDA	input/output	I2C serial data signal

11.3.1 Start and stop conditions

All transfers begin with a START condition and end with a STOP condition.

The start and stop conditions are generally generated by the master. The bus is considered to be in a busy state after the start condition, and is considered to be in an idle state for a period of time after the stop condition.

Start condition: SDA generates a high-to-low level transition when SCL is high;

Stop condition: SDA generates a low-to-high level transition when SCL is high.

The waveform diagram is as follows:

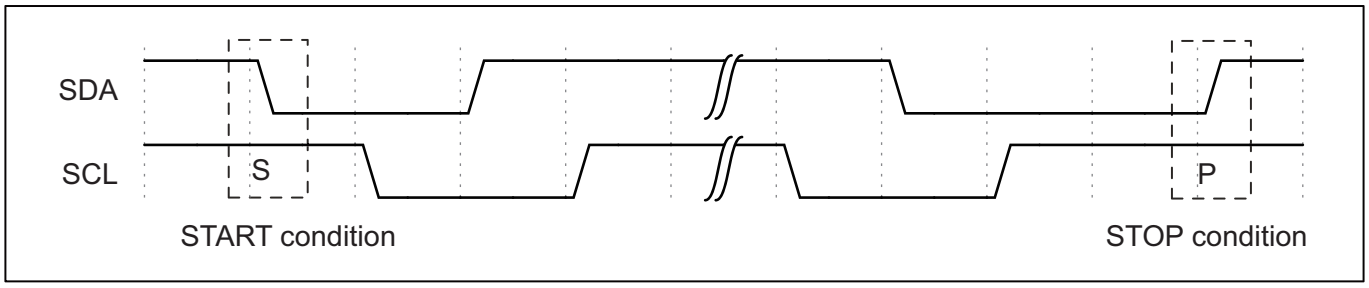

Figure 11.1: I2C stop/start condition

11.3.2 Data transmission format

The first 8 bits transmitted are the address byte, including the 7-bit slave address and the 1-bit direction bit. Data sent or received by the host is controlled by the eighth bit of the first byte sent by the host.

If it is 0, it means that the data is sent by the master; if it is 1, it means that the data is received by the master, and then the slave sends an acknowledge bit (ACK). After the data transmission is completed, the master sends a stop signal. The waveform is as follows:

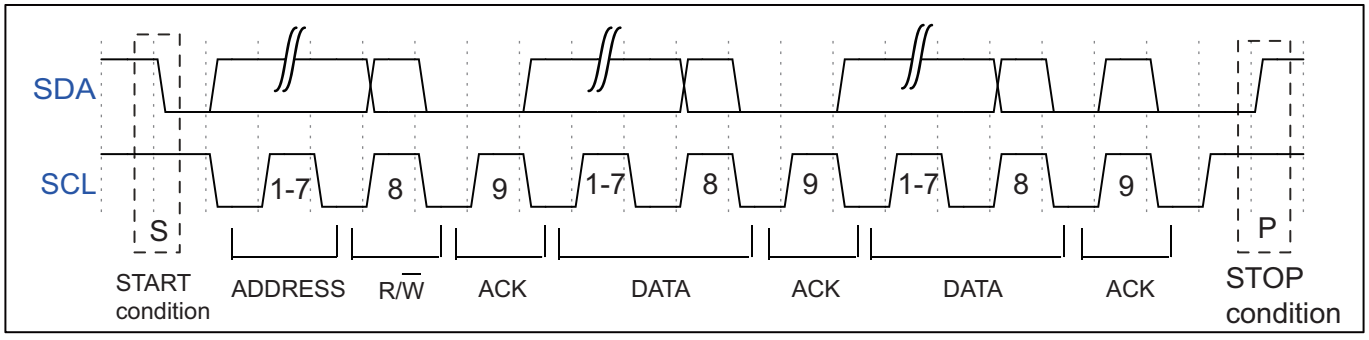

Figure 11.2: I2C data transmission format

Timing of master transmission and slave reception

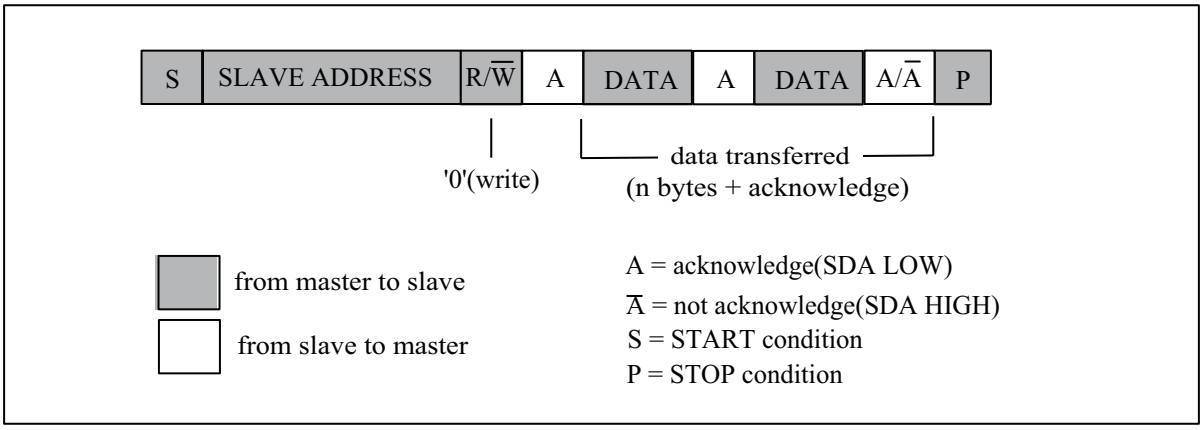

Figure 11.3: Master tx and slave rx

Timing of master receive and slave send

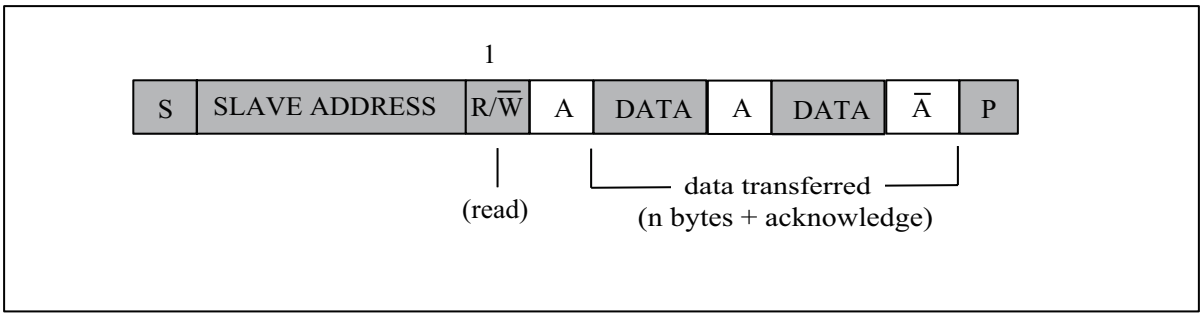

Figure 11.4: Master rx and slave tx

11.3.3 Arbitration

When there are multiple masters on the I2C bus, multiple masters may start transmitting at the same time. At this time, it is necessary to rely on the arbitration mechanism to determine which master has the right to complete the next data transfer. The remaining masters must give up control of the bus. The transmission cannot be started again until the bus is free.

During the transmission process, all hosts need to check whether SDA is consistent with the data they want to send when SCL is high. When the SDA level is different from expected, it means that other hosts are also transmitting at the same time. Hosts with different SDA levels will lose the arbitration and other hosts will complete the data transmission.

The waveform diagram of two hosts transmitting data and starting the arbitration mechanism at the same time is as follows:

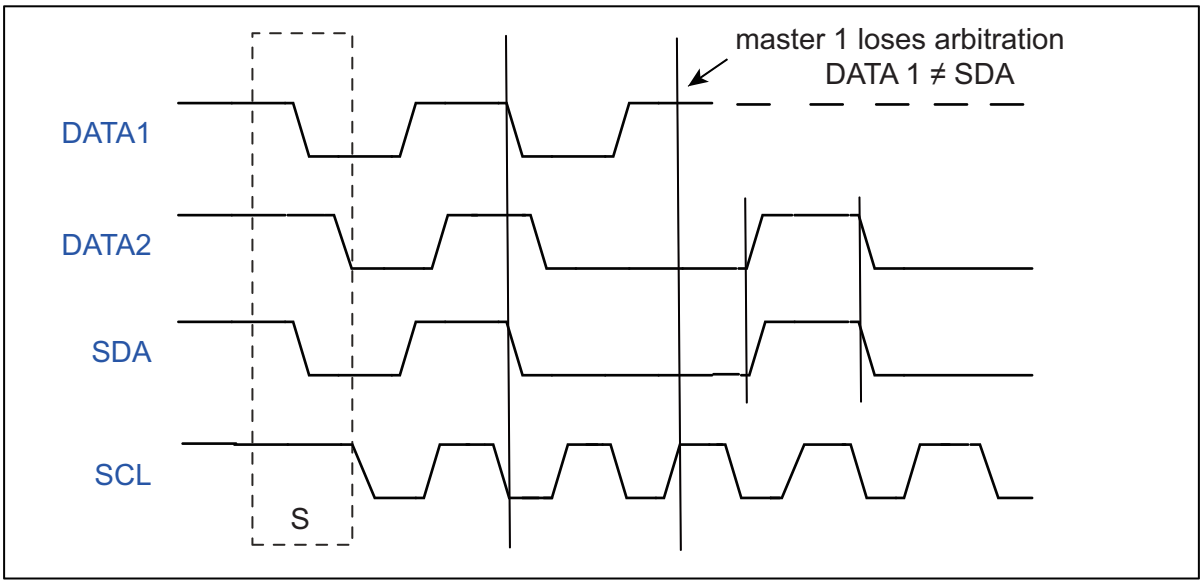

Figure 11.5: Tx and Rx together

11.4 I2C clock setting

The I2C clock is derived from bclk (bus clock), which can be divided based on the bclk clock.

Register I2C_PRD_DATA can divide the clock of the data segment. The i2c module divides the data transmission into 4 phases. Each phase is controlled by a single byte in the register. The number of samples in each phase can be set. The 4 samples together determine the frequency division coefficient of the i2c clock. .

For example, bclk is 32M and the value of register I2C_PRD_DATA is 0x15151515 by default without configuration. Then the clock frequency of I2C is 32M / ((15 + 1) * 4) = 500K.

Similarly, the registers I2C_PRD_START and I2C_PRD_STOP also divide the clock of the start bit and stop bit respectively.

11.5 I2C configuration process

11.5.1 Configuration item

- Read and write flags
- Slave address
- Slave device address
- Slave device address length
- Data (when sending, configure the data to be sent; when receiving, store the received data)
- Data length

- Enable signal

11.5.2 Read and write flags

I2C supports two working states: sending and receiving. Register PKTDIR indicates the sending or receiving status. When it is set to 0, it indicates the sending state, and when it is set to 1, it indicates the receiving state.

11.5.3 Slave address

Each slave device connected to I2C will have a unique address. Usually the address length is 7 bits. The slave device address will be written into the register SLVADDR. I2C will automatically shift left by 1 bit before sending it from the device address. Transmit/receive direction bit on the low-order complement.

11.5.4 Slave register address

Slave device register address indicates the register address that I2C needs to read and write to a certain register of the slave device. The slave device address will be written to the register I2C_SUB_ADDR, and the register I2C_CR_-I2C_SUB_ADDR_EN needs to be set.

If the register SAEN is set to 0, the I2C master will skip the slave register address segment when transmitting.

11.5.5 Slave device address length

The slave device address length is decremented by one and written to the register SABC.

11.5.6 Data

The data part represents the data that needs to be sent to the slave device, or the data that needs to be received from the slave device.

When I2C sends data, the data needs to be written into the I2C FIFO in word units in turn, and the data is written to the register address I2C_FIFO_WDATA of the FIFO.

When the I2C receives data, it needs to read the data from the I2C FIFO in units of words in order, and the received data reads the register address I2C_FIFO_RDATA of the FIFO.

11.5.7 Data length

Decrement the data length by one and write to the register PKTLEN.

11.5.8 Enable signal

After the above configurations are completed, write the enable signal register MEN to 1 to automatically start the I2C transmission process.

When the read-write flag is set to 0, I2C sends data, and the host sends the process:

1. Start bit
2. (1 bit left from device address + 0) + ACK
3. Slave device address + ACK
4. 1 byte data + ACK
5. 1 byte data + ACK
6. Stop bit

When the read / write flag is set to 1, I2C receives data and the host sends the process:

1. Start bit
2. (1 bit left from device address + 0) + ACK
3. Slave device address + ACK
4. Start bit
5. (1 bit left from device address + 1) + ACK
6. 1 byte data + ACK
7. 1 byte data + ACK
8. Stop bit

11.6 FIFO management

The I2C FIFO depth is 2 words. I2C transmission and reception can be divided into RX FIFO and TX FIFO.

The register RFICNT indicates how much data (unit word) needs to be read in the RX FIFO.

The register TFICNT indicates how much space (in Word) is available for writing in the TX FIFO.

I2C FIFO status:

- RX FIFO underflow: When the data in the RX FIFO has been read or is empty, continue to read data from the RX FIFO, the register RFIU will be set;
- RX FIFO overflow: When I2C receives data until the 2 words of RX FIFO are filled. Without reading the RX FIFO, I2C receives the data again and the register RFIO will be set;
- TX FIFO underflow: When the size of the data filled in the TX FIFO does not meet the configured I2C data length PKTLEN, and there is no new data to be filled into the TX FIFO, the register TFIU will be set;
- TX FIFO overflow: After the two words of the TX FIFO are filled, before the data in the TX FIFO is sent out, fill the TX FIFO with data again. The register TFIO will be set.

11.7 Using DMA

I2C can use DMA to send and receive data. Set DTEN to 1 to enable the DMA transmission mode. After a channel is allocated for I2C, the DMA will transfer data from the memory area to the I2C_FIFO_WDATA register.

Set DREN to 1 to enable the DMA receive mode. After a channel is allocated for I2C, the DMA will transfer the data in the I2C_FIFO_RDATA register to the memory area.

When the I2C module is used with DMA, the data part will be automatically carried by the DMA. There is no need for the CPU to write data to the I2C TX FIFO or read data from the I2C RX FIFO.

11.7.1 DMA transmission process

1. Configure the read and write flags to 0
2. Configure the slave device address
3. Configure Slave Device Address
4. Configure slave device address length
5. Data length
6. Set the enable signal register
7. Configure DMA transfer size
8. Configure DMA source address transfer width
9. Configure the DMA destination address transfer width (Note that when I2C is used with DMA, the destination address transfer width needs to be set to 32bits and used in word alignment)
10. Configure the DMA source address as the memory address to store the transmitted data
11. Configure the DMA destination address as I2C TX FIFO address, I2C_FIFO_WDATA
12. Enable DMA

11.7.2 DMA receiving process

1. Configure the read and write flags to 1
2. Configure the slave device address
3. Configure Slave Device Address
4. Configure slave device address length
5. Data length
6. Set the enable signal register

7. Configure DMA transfer size
8. Configure the DMA source address transfer width (Note that when I2C is used with DMA, the source address transfer width needs to be set to 32bits and used in word alignment)
9. Configure DMA destination address transfer width
10. Configure the DMA source address as I2C RX FIFO address, I2C_FIFO_RDATA
11. Configure the DMA destination address as the memory address to store the received data
12. Enable DMA

11.8 I2C interrupt

I2C includes the following interrupts:

- I2C_TRANS_END_INT: I2C transfer end interrupt
- I2C_TX_FIFO_READY_INT: Interrupt is triggered when I2C TX FIFO has free space available for filling
- I2C_RX_FIFO_READY_INT: When I2C RX FIFO receives data, trigger interrupt
- I2C_NACK_RECV_INT: When the I2C module detects a NACK state, an interrupt is triggered
- I2C_ARB_LOST_INT: I2C arbitration lost interrupt
- I2C_FIFO_ERR_INT: I2C FIFO ERROR interrupt

11.9 Register description

Name	Description
i2c_config	I2C configuration register
i2c_int_sts	I2C interrupt status
i2c_sub_addr	I2C sub-address configuration
i2c_bus_busy	I2C bus busy control register
i2c_prd_start	I2C length of start phase
i2c_prd_stop	I2C length of stop phase
i2c_prd_data	I2C length of data phase
i2c_fifo_config_0	I2C FIFO configuration register0
i2c_fifo_config_1	I2C FIFO configuration register1
i2c_fifo_wdata	I2C FIFO write data

Name	Description
i2c_fifo_rdata	I2C FIFO read data

11.9.1 i2c_config

Address：0x4000a300

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEGCNT				RSVD				PKTLEN							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	SLVADDR							RSVD	SABC		SAEN	SCLS EN	DEG EN	PKT DIR	MEN

Bits	Name	Type	Reset	Description
31:28	DEGCNT	R/W	4'd0	De-glitch function cycle count
27:24	RSVD			
23:16	PKTLEN	R/W	8'd0	Packet length (unit: byte)
15	RSVD			
14:8	SLVADDR	R/W	7'd0	Slave address for I2C transaction (target address)
7	RSVD			
6:5	SABC	R/W	2'd0	Sub-address field byte count 2'd0: 1-byte, 2'd1: 2-byte, 2'd2: 3-byte, 2'd3: 4-byte
4	SAEN	R/W	1'b0	Enable signal of I2C sub-address field
3	SCLSEN	R/W	1'b1	Enable signal of I2C SCL synchronization, should be enabled to support Multi-Master and Clock-Stretching (Normally should not be turned-off)
2	DEGEN	R/W	1'b0	Enable signal of I2C input de-glitch function (for all input pins)
1	PKTDIR	R/W	1'b1	Transfer direction of the packet 1'b0: Write; 1'b1: Read
0	MEN	R/W	1'b0	Enable signal of I2C Master function Asserting this bit will trigger the transaction, and should be de-asserted after finish

11.9.2 i2c_int_sts

Address：0x4000a304

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		FER EN	ARB EN	NAK EN	RXF EN	TXF EN	END EN	RSVD			ARB CLR	NAK CLR	RSVD		END CLR
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		FER MASK	ARB MASK	NAK MASK	RXF MASK	TXF MASK	END MASK	RSVD		FER INT	ARB INT	NAK INT	RXF INT	TXF INT	END INT

Bits	Name	Type	Reset	Description
31:30	RSVD			
29	FEREN	R/W	1'b1	Interrupt enable of i2c_fer_int
28	ARBEN	R/W	1'b1	Interrupt enable of i2c_arb_int
27	NAKEN	R/W	1'b1	Interrupt enable of i2c_nak_int
26	RXFEN	R/W	1'b1	Interrupt enable of i2c_rxf_int
25	TXFEN	R/W	1'b1	Interrupt enable of i2c_txf_int
24	ENDEN	R/W	1'b1	Interrupt enable of i2c_end_int
23:21	RSVD			
20	ARBCLR	W1C	1'b0	Interrupt clear of i2c_arb_int
19	NAKCLR	W1C	1'b0	Interrupt clear of i2c_nak_int
18:17	RSVD			
16	ENDCLR	W1C	1'b0	Interrupt clear of i2c_end_int
15:14	RSVD			
13	FERMASK	R/W	1'b1	Interrupt mask of i2c_fer_int
12	ARBMASK	R/W	1'b1	Interrupt mask of i2c_arb_int
11	NAKMASK	R/W	1'b1	Interrupt mask of i2c_nak_int
10	RXFMASK	R/W	1'b1	Interrupt mask of i2c_rxf_int
9	TXFMASK	R/W	1'b1	Interrupt mask of i2c_txf_int
8	ENDMASK	R/W	1'b1	Interrupt mask of i2c_end_int
7:6	RSVD			
5	FERINT	R	1'b0	I2C TX/RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
4	ARBINT	R	1'b0	I2C arbitration lost interrupt
3	NAKINT	R	1'b0	I2C NACK-received interrupt

Bits	Name	Type	Reset	Description
2	RXFINT	R	1'b0	I2C RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto-cleared when data is popped
1	TXFINT	R	1'b0	I2C TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed
0	ENDINT	R	1'b0	I2C transfer end interrupt

11.9.3 i2c_sub_addr

Address：0x4000a308

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUBAB3								SUBAB2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUBAB1								SUBAB0							

Bits	Name	Type	Reset	Description
31:24	SUBAB3	R/W	8'd0	I2C sub-address field - byte[3]
23:16	SUBAB2	R/W	8'd0	I2C sub-address field - byte[2]
15:8	SUBAB1	R/W	8'd0	I2C sub-address field - byte[1]
7:0	SUBAB0	R/W	8'd0	I2C sub-address field - byte[0] (sub-address starts from this byte)

11.9.4 i2c_bus_busy

Address：0x4000a30c

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

Bits	Name	Type	Reset	Description
31:2	RSVD			
1	BUSYCLR	W1C	1'b0	Clear signal of bus_busy status, not for normal usage (in case I2C bus hangs)

Bits	Name	Type	Reset	Description
0	BUSY	R	1'b0	Indicator of I2C bus busy

11.9.5 i2c_prd_start

Address：0x4000a310

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRDSPH3								PRDSPH2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRDSPH1								PRDSPH0							

Bits	Name	Type	Reset	Description
31:24	PRDSPH3	R/W	8'd15	Length of START condition phase 3
23:16	PRDSPH2	R/W	8'd15	Length of START condition phase 2
15:8	PRDSPH1	R/W	8'd15	Length of START condition phase 1
7:0	PRDSPH0	R/W	8'd15	Length of START condition phase 0

11.9.6 i2c_prd_stop

Address：0x4000a314

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRDPPH3								PRDPPH2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRDPPH1								PRDPPH0							

Bits	Name	Type	Reset	Description
31:24	PRDPPH3	R/W	8'd15	Length of STOP condition phase 3
23:16	PRDPPH2	R/W	8'd15	Length of STOP condition phase 2
15:8	PRDPPH1	R/W	8'd15	Length of STOP condition phase 1
7:0	PRDPPH0	R/W	8'd15	Length of STOP condition phase 0

11.9.7 i2c_prd_data

Address：0x4000a318

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRDDPH3								PRDDPH2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRDDPH1								PRDDPH0							

Bits	Name	Type	Reset	Description
31:24	PRDDPH3	R/W	8'd15	Length of DATA phase 3
23:16	PRDDPH2	R/W	8'd15	Length of DATA phase 2
15:8	PRDDPH1	R/W	8'd15	Length of DATA phase 1 Note: This value should not be set to 8'd0, adjust source clock rate instead if higher I2C clock rate is required
7:0	PRDDPH0	R/W	8'd15	Length of DATA phase 0

11.9.8 i2c_fifo_config_0

Address：0x4000a380

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								RFIU	RFIO	TFIU	TFIO	RFI CLR	TFI CLR	DREN	DTEN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	RFIU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	RFIO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	TFIU	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	TFIO	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	RFICLR	W1C	1'b0	Clear signal of RX FIFO
2	TFICLR	W1C	1'b0	Clear signal of TX FIFO
1	DREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface
0	DTEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface

11.9.9 i2c_fifo_config_1

Address：0x4000a384

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

Bits	Name	Type	Reset	Description
31:25	RSVD			
24	RFITH	R/W	1'd0	RX FIFO threshold, dma_rx_req will not be asserted if tx_-fifo_cnt is less than this value
23:17	RSVD			
16	TFITH	R/W	1'd0	TX FIFO threshold, dma_tx_req will not be asserted if tx_-fifo_cnt is less than this value
15:10	RSVD			
9:8	RFICNT	R	2'd0	RX FIFO available count
7:2	RSVD			
1:0	TFICNT	R	2'd2	TX FIFO available count

11.9.10 i2c_fifo_wdata

Address：0x4000a388

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

Bits	Name	Type	Reset	Description
31:0	FIWD	W	x	I2C FIFO write data

11.9.11 i2c_fifo_rdata

Address：0x4000a38c

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

Bits	Name	Type	Reset	Description
31:0	FIRD	R	32'h0	I2C FIFO read data

12 PWM

12.1 PWM introduction

Pulse width modulation (PWM) is an analog control method. The bias of the base of the transistor or the gate of the MOS tube is modulated according to the change of the corresponding load to realize the change of the conduction time of the transistor or the MOS tube. So as to realize the change of the output of the switch stable power supply. This method can keep the output voltage of the power supply constant when the working conditions change, and it is a very effective technology to control the analog circuit with the digital signal of the microprocessor. It is widely used in many fields from measurement and communication to power control and conversion.

12.2 PWM main features

- Support 5-channel PWM signal generation
- Three clock sources can be selected (bus clock <bclk>, crystal oscillator clock <xtal_ck>, slow clock <32k>), with 16-bit clock divider
- Double threshold setting, increase pulse flexibility

12.3 PWM function description

12.3.1 Clock and divider

There are three options for each PWM counter clock source, the sources are as follows:

A. bclk - Chip bus clock

B. XTAL - External crystal clock

C. f32k - System RTC clock

Each counter has its own 16-bit frequency divider. The selected clock can be divided by APB. The PWM counter will use the divided clock as the counting cycle unit, and perform one action every time a counting cycle passes .

12.3.2 Pulse generation principle

There is a counter in the PWM. When the counter is in the middle of two settable thresholds, the PWM output is 1, otherwise when the counter is outside the two set thresholds, the PWM output is 0. As shown below:

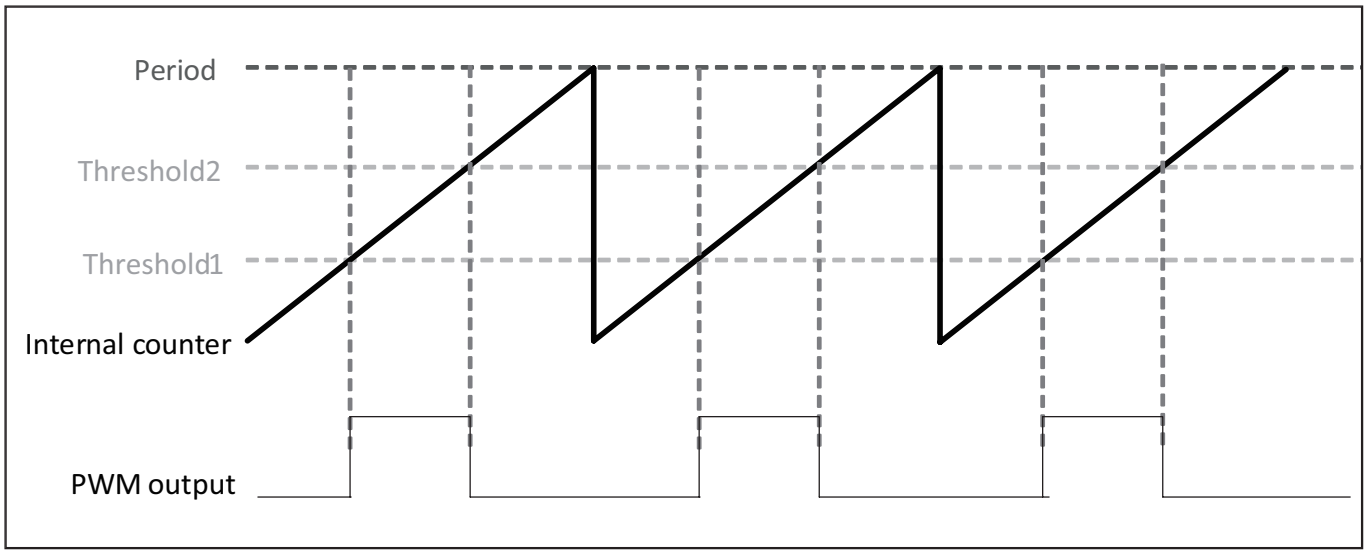

Figure 12.1: PWM waveform

The PWM cycle is determined by two parts, one is the clock frequency division coefficient, and the other is the clock duration.

The clock division coefficient is set by the register PWMn_CLK_DIV[15:0] (n is 0~5), which is used to divide the PWM source clock.

The clock duration is set by the register PWMn_PERIOD[15:0] (n is 0~5), which is used to set the number of divided clock cycles for a PWM cycle. That is, the period of PWM=PWM source clock/PWMn_CLK_DIV[15:0]/PWMn_PERIOD[15:0].

The duty cycle of PWM is determined by the clock duration and two thresholds. The first threshold is set by the register PWMn_THRE1[15:0] (n is 0~5), the second threshold is set by the register PWMn_THRE2[15:0] (n is 0~5), the PWM waveform will be in the first Pull up at one threshold, and pull down at the second threshold. That is, the duty cycle of PWM=(PWMn_THRE2[15:0]-PWMn_THRE1[15:0])/PWMn_PERIOD[15:0].

Example: If the PWM clock source is selected as bclk, which is 72MHz, to generate a 1kHz, 20% duty cycle PWM wave, set as follows:

PWMn_CLK_DIV[15:0]=2

PWMn_PERIOD[15:0]=72000000/2/1000=36000

PWMn_THRE1[15:0]=0

PWMn_THRE2[15:0]=0+36000*20%=7200

12.3.3 PWM interrupt

For each PWM channel, you can set the cycle count value. When the cycle number of the PWM output reaches this count value, a PWM interrupt will be generated.

Table 12.1: Duty Cycle Parameters

Freq/MHz	Supported duty cycle(n is an integer,and 2 <= n <= 65535^2)											
36	0%	50%	100%									
24	0%	33.33%	66.67%	100%								
18	0%	25%	50%	75%	100%							
14.4	0%	20%	40%	60%	80%	100%						
12	0%	16.67%	33.33%	50%	66.67%	83.33%	100%					
10.29	0%	14.29%	28.57%	42.86%	57.14%	71.43%	85.71%	100%				
9	0%	12.50%	25%	37.50%	50%	62.50%	75%	87.50%	100%			
8	0%	11.11%	22.22%	33.33%	44.44%	55.56%	66.67%	77.78%	88.89%	100%		
7.2	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	
•												
•												
•												
72/n	0/n	1/n	2/n	3/n	4/n	5/n	6/n	7/n	8/n	9/n	•••	n/n

12.4 Register description

Name	Description
pwm_int_config	PWM interrupt configuration register
pwm0_clkdiv	PWM0 clock division configuration register
pwm0_thre1	PWM0 first counter threshold configuration register
pwm0_thre2	PWM0 sencond counter threshold configuration register
pwm0_period	PWM0 period setting register
pwm0_config	PWM0 configuration register
pwm0_interrupt	PWM0 interrupt register
pwm1_clkdiv	PWM1 clock division configuration register
pwm1_thre1	PWM1 first counter threshold configuration register

Name	Description
pwm1_thre2	PWM1 sencond counter threshold configuration register
pwm1_period	PWM1 period setting register
pwm1_config	PWM1 configuration register
pwm1_interrupt	PWM1 interrupt register
pwm2_clkdiv	PWM2 clock division configuration register
pwm2_thre1	PWM2 first counter threshold configuration register
pwm2_thre2	PWM2 sencond counter threshold configuration register
pwm2_period	PWM2 period setting register
pwm2_config	PWM2 configuration register
pwm2_interrupt	PWM2 interrupt register
pwm3_clkdiv	PWM3 clock division configuration register
pwm3_thre1	PWM3 first counter threshold configuration register
pwm3_thre2	PWM3 sencond counter threshold configuration register
pwm3_period	PWM3 period setting register
pwm3_config	PWM3 configuration register
pwm3_interrupt	PWM3 interrupt register
pwm4_clkdiv	PWM4 clock division configuration register
pwm4_thre1	PWM4 first counter threshold configuration register
pwm4_thre2	PWM4 sencond counter threshold configuration register
pwm4_period	PWM4 period setting register
pwm4_config	PWM4 configuration register
pwm4_interrupt	PWM4 interrupt register

12.4.1 pwm_int_config

Address：0x4000a400

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

Bits	Name	Type	Reset	Description
31:14	RSVD			
13:8	INTCLR	W	6'd0	PWM channel interrupt clear
7:6	RSVD			
5:0	INTSTS	R	6'd0	PWM channel interrupt status

12.4.2 pwm0_clkdiv

Address：0x4000a420

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	CLKDIV	R/W	16'b0	PWM clock division

12.4.3 pwm0_thre1

Address：0x4000a424

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE1															

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE1	R/W	16'b0	PWM first counter threshold, can't be larger that pwm_thre2

12.4.4 pwm0_thre2

Address：0x4000a428

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE2															

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE2	R/W	16'd0	PWM sencond counter threshold, can't be smaller that pwm_thre1

12.4.5 pwm0_period

Address：0x4000a42c

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	PERIOD	R/W	16'd0	PWM period setting

12.4.6 pwm0_config

Address：0x4000a430

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								STOP STA	STOP EN	SW MODE	SW FVAL	STOP MODE	OUT INV	CLKSEL	

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	STOPSTA	R	1'b0	PWM stop status
6	STOPEN	R/W	1'b0	PWM stop enable

Bits	Name	Type	Reset	Description
5	SWMODE	R/W	1'b0	PWM SW Mode setting
4	SWFVAL	R/W	1'b0	PWM SW Mode force value
3	STOPMODE	R/W	1'b1	PWM stop mode, 1'b1 - graceful ; 1'b0 - abrupt
2	OUTINV	R/W	1'b0	PWM invert output mode
1:0	CLKSEL	R/W	2'd0	PWM clock source select, 2'b00-xclk ; 2'b01-bclk ; others-f32k_clk

12.4.7 pwm0_interrupt

Address：0x4000a434

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

Bits	Name	Type	Reset	Description
31:17	RSVD			
16	INTEN	R/W	1'b0	PWM interrupt enable
15:0	INTPECN	R/W	16'd0	PWM interrupt period counter threshold

12.4.8 pwm1_clkdiv

Address：0x4000a440

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	CLKDIV	R/W	16'b0	PWM clock division

12.4.9 pwm1_thre1

Address：0x4000a444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE1															

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE1	R/W	16'b0	PWM first counter threshold, can't be larger that pwm_thre2

12.4.10 pwm1_thre2

Address：0x4000a448

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE2															

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE2	R/W	16'd0	PWM sencond counter threshold, can't be smaller that pwm_thre1

12.4.11 pwm1_period

Address：0x4000a44c

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	PERIOD	R/W	16'd0	PWM period setting

12.4.12 pwm1_config

Address：0x4000a450

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								STOP STA	STOP EN	SW MODE	SW FVAL	STOP MODE	OUT INV	CLKSEL	

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	STOPSTA	R	1'b0	PWM stop status
6	STOPEN	R/W	1'b0	PWM stop enable
5	SWMODE	R/W	1'b0	PWM SW Mode setting
4	SWFVAL	R/W	1'b0	PWM SW Mode force value
3	STOPMODE	R/W	1'b1	PWM stop mode, 1'b1 - graceful ; 1'b0 - abrupt
2	OUTINV	R/W	1'b0	PWM invert output mode
1:0	CLKSEL	R/W	2'd0	PWM clock source select, 2'b00-xclk ; 2'b01-bclk ; others-f32k_clk

12.4.13 pwm1_interrupt

Address：0x4000a454

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

Bits	Name	Type	Reset	Description
31:17	RSVD			
16	INTEN	R/W	1'b0	PWM interrupt enable
15:0	INTPECN	R/W	16'd0	PWM interrupt period counter threshold

12.4.14 pwm2_clkdiv

Address：0x4000a460

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	CLKDIV	R/W	16'b0	PWM clock division

12.4.15 pwm2_thre1

Address：0x4000a464

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE1															

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE1	R/W	16'b0	PWM first counter threshold, can't be larger that pwm_thre2

12.4.16 pwm2_thre2

Address：0x4000a468

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE2															

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE2	R/W	16'd0	PWM sencond counter threshold, can't be smaller that pwm_thre1

12.4.17 pwm2_period

Address：0x4000a46c

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	PERIOD	R/W	16'd0	PWM period setting

12.4.18 pwm2_config

Address：0x4000a470

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								STOP STA	STOP EN	SW MODE	SW FVAL	STOP MODE	OUT INV	CLKSEL	

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	STOPSTA	R	1'b0	PWM stop status
6	STOPEN	R/W	1'b0	PWM stop enable

Bits	Name	Type	Reset	Description
5	SWMODE	R/W	1'b0	PWM SW Mode setting
4	SWFVAL	R/W	1'b0	PWM SW Mode force value
3	STOPMODE	R/W	1'b1	PWM stop mode, 1'b1 - graceful ; 1'b0 - abrupt
2	OUTINV	R/W	1'b0	PWM invert output mode
1:0	CLKSEL	R/W	2'd0	PWM clock source select, 2'b00-xclk ; 2'b01-bclk ; others-f32k_clk

12.4.19 pwm2_interrupt

Address：0x4000a474

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

Bits	Name	Type	Reset	Description
31:17	RSVD			
16	INTEN	R/W	1'b0	PWM interrupt enable
15:0	INTPECN	R/W	16'd0	PWM interrupt period counter threshold

12.4.20 pwm3_clkdiv

Address：0x4000a480

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	CLKDIV	R/W	16'b0	PWM clock division

12.4.21 pwm3_thre1

Address：0x4000a484

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE1															

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE1	R/W	16'b0	PWM first counter threshold, can't be larger that pwm_thre2

12.4.22 pwm3_thre2

Address：0x4000a488

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE2															

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE2	R/W	16'd0	PWM sencond counter threshold, can't be smaller that pwm_thre1

12.4.23 pwm3_period

Address：0x4000a48c

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	PERIOD	R/W	16'd0	PWM period setting

12.4.24 pwm3_config

Address：0x4000a490

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								STOP STA	STOP EN	SW MODE	SW FVAL	STOP MODE	OUT INV	CLKSEL	

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	STOPSTA	R	1'b0	PWM stop status
6	STOPEN	R/W	1'b0	PWM stop enable
5	SWMODE	R/W	1'b0	PWM SW Mode setting
4	SWFVAL	R/W	1'b0	PWM SW Mode force value
3	STOPMODE	R/W	1'b1	PWM stop mode, 1'b1 - graceful ; 1'b0 - abrupt
2	OUTINV	R/W	1'b0	PWM invert output mode
1:0	CLKSEL	R/W	2'd0	PWM clock source select, 2'b00-xclk ; 2'b01-bclk ; others-f32k_clk

12.4.25 pwm3_interrupt

Address：0x4000a494

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

Bits	Name	Type	Reset	Description
31:17	RSVD			
16	INTEN	R/W	1'b0	PWM interrupt enable
15:0	INTPECN	R/W	16'd0	PWM interrupt period counter threshold

12.4.26 pwm4_clkdiv

Address：0x4000a4a0

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	CLKDIV	R/W	16'b0	PWM clock division

12.4.27 pwm4_thre1

Address：0x4000a4a4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE1															

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE1	R/W	16'b0	PWM first counter threshold, can't be larger that pwm_thre2

12.4.28 pwm4_thre2

Address：0x4000a4a8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRE2															

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE2	R/W	16'd0	PWM sencond counter threshold, can't be smaller that pwm_thre1

12.4.29 pwm4_period

Address：0x4000a4ac

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	PERIOD	R/W	16'd0	PWM period setting

12.4.30 pwm4_config

Address：0x4000a4b0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								STOP STA	STOP EN	SW MODE	SW FVAL	STOP MODE	OUT INV	CLKSEL	

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	STOPSTA	R	1'b0	PWM stop status
6	STOPEN	R/W	1'b0	PWM stop enable

Bits	Name	Type	Reset	Description
5	SWMODE	R/W	1'b0	PWM SW Mode setting
4	SWFVAL	R/W	1'b0	PWM SW Mode force value
3	STOPMODE	R/W	1'b1	PWM stop mode, 1'b1 - graceful ; 1'b0 - abrupt
2	OUTINV	R/W	1'b0	PWM invert output mode
1:0	CLKSEL	R/W	2'd0	PWM clock source select, 2'b00-xclk ; 2'b01-bclk ; others-f32k_clk

12.4.31 pwm4_interrupt

Address：0x4000a4b4

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

Bits	Name	Type	Reset	Description
31:17	RSVD			
16	INTEN	R/W	1'b0	PWM interrupt enable
15:0	INTPECN	R/W	16'd0	PWM interrupt period counter threshold

13 TIMER

13.1 TIMER introduction

The chip has two 32-bit counters, each of which can independently control and configure its parameters and clock frequency.

There is a watchdog counter in the chip. Unpredictable software or hardware behavior may cause the application to malfunction. A watchdog timer can help the system recover from it. If the current time exceeds the predetermined time, but the dog is not fed or closed Timer, which can trigger interrupt or system reset according to the setting.

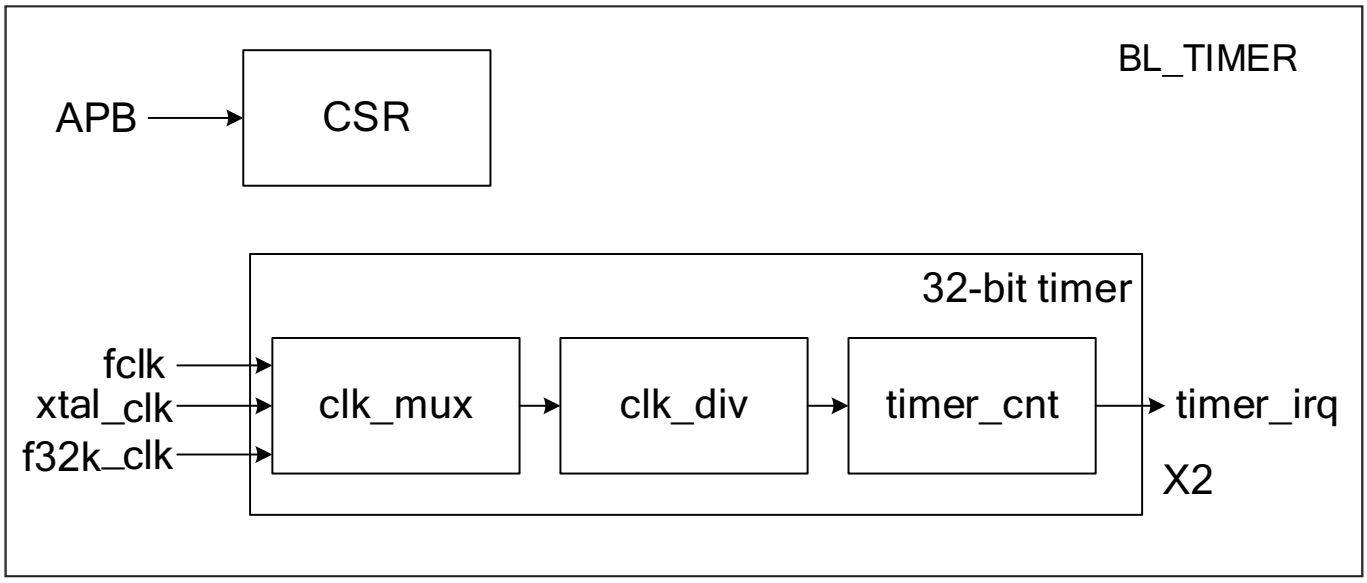

Figure 13.1: Timer block diagram

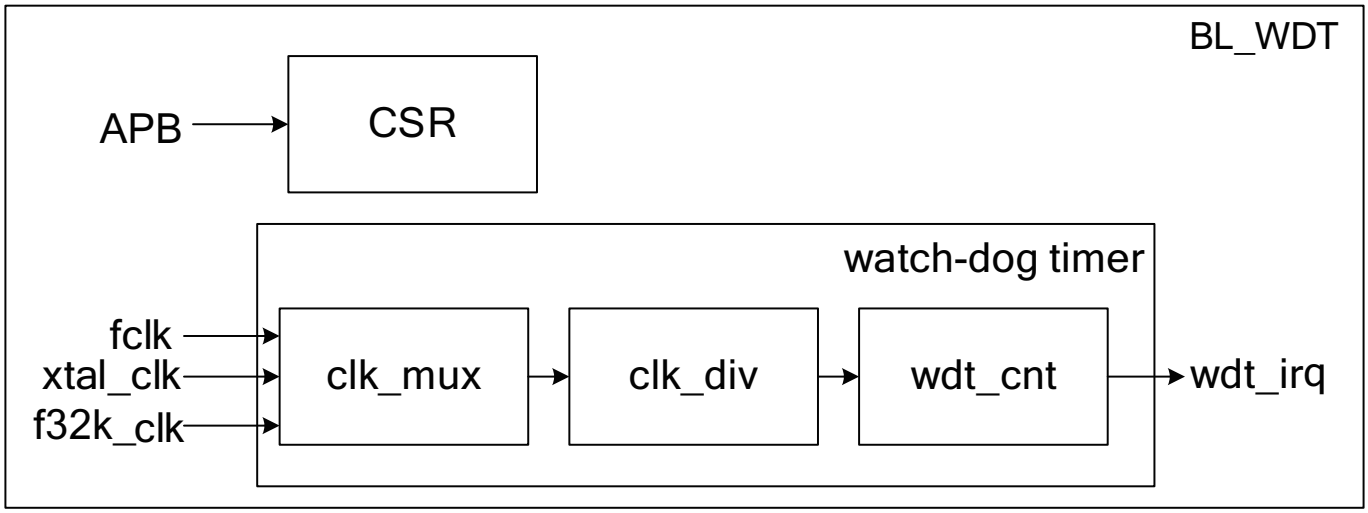

Figure 13.2: Watchdog timer block diagram

13.2 TIMER main features

- Multiple clock source options
- 8-bit clock divider with a division factor of 1-256.
- Two 32-bit timers
- Each timer contains three alarm value settings, which can be set independently to alarm when each alarm value overflows
- Support Free Run mode and Pre_load mode
- 16-bit watchdog timer
- Supports write password protection to prevent system abnormalities caused by incorrect settings
- Support two watchdog overflow methods: interrupt or reset

13.3 TIMER function description

13.3.1 8-bit divider

There are three types of Watchdog timer clocks:

- Fclk–System master clock
- 32K–32K clock
- Xtal–External crystal

There are four timer clock sources:

- Fclk–System master clock
- 32K–32K clock
- 1K–1K clock（32K frequency division）
- Xtal–External crystal

Each counter has its own 8-bit frequency divider. The selected clock can be divided by 1-256 through APB. Specifically, when it is set to 0, it means no frequency division, and when it is set to 1, it divides it by 2. The maximum frequency division coefficient is 256, the counter will use the divided clock as the unit of the counting cycle, each time a counting cycle is increased by one.

13.3.2 General timer operating mode

Each general-purpose timer includes three comparators, a counter and a preload register. When the clock source is set and the timer is started, the counter starts to count up. When the counter value is equal to the comparator, the comparison is performed. When the flag is set, a compare interrupt is generated.

The initial value of the counter depends on the timing mode. In FreeRun mode, the initial value of the counter is 0, and then counts up. When it reaches the maximum value, it starts counting from 0 again.

In PreLoad mode, the initial value of the counter is the value of the PreLoad register and then counts up. When the PreLoad condition is met, the value of the counter is set to the value of the PreLoad register, and then the counter starts to count up again. During the counting process, once the value of the counter matches one of the three comparators, the comparator's comparison flag will be set and a corresponding comparison interrupt can be generated.

If the value of the preload register is 10, the value of Comparator 0 is 13, the value of Comparator 1 is 16, and the value of Comparator 2 is 19, the working sequence of the timer in PreLoad mode is as follows:

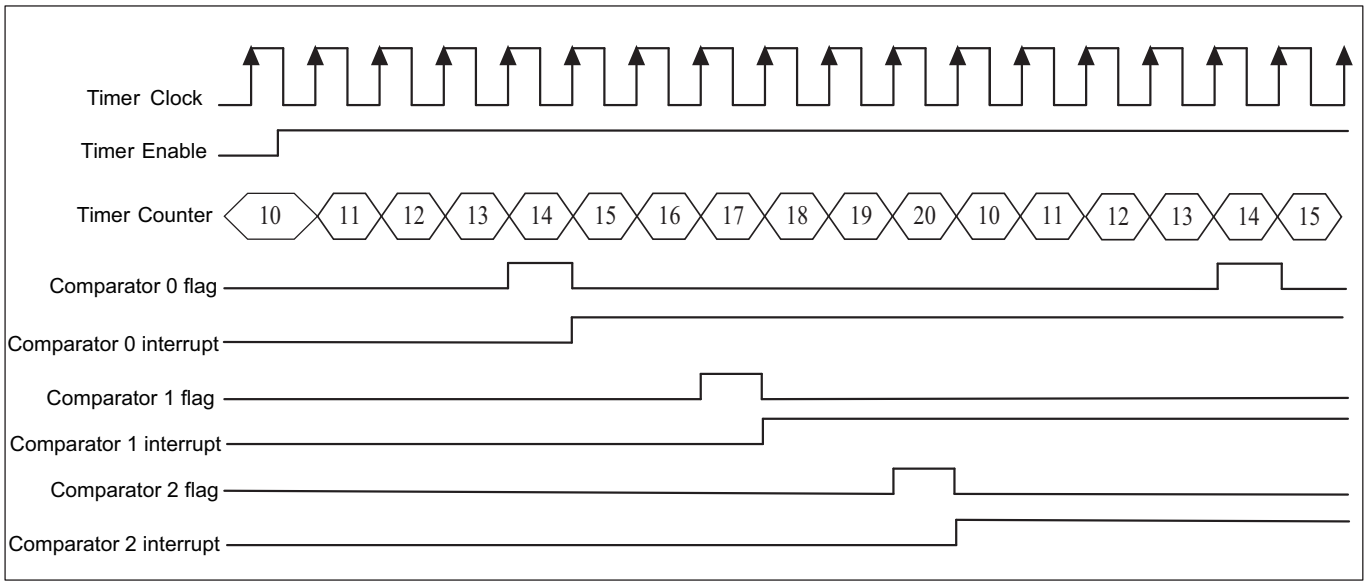

Figure 13.3: Timer Preload

In FreeRun mode, the timer working sequence is basically the same as PreLoad, the difference is that the counter will start to accumulate from 0 to the maximum value. The mechanism of the generated compare flags and compare interrupts is the same as in FreeRun mode.

13.3.3 Watchdog timer operating mode

The watchdog timer includes a counter and a comparator. The counter counts up from 0. If the counter is reset (feed the dog), it starts counting up from 0 again. When the counter value is equal to the comparator, a comparison interrupt signal or a system reset signal will be generated, and the user can choose to use one of them as required.

The watchdog counter is incremented by one in each counting cycle unit. Software can reset the watchdog counter to zero at any point in time through the APB.

If the value of the comparator is 6, the working sequence of Watchdog is shown in the figure below:

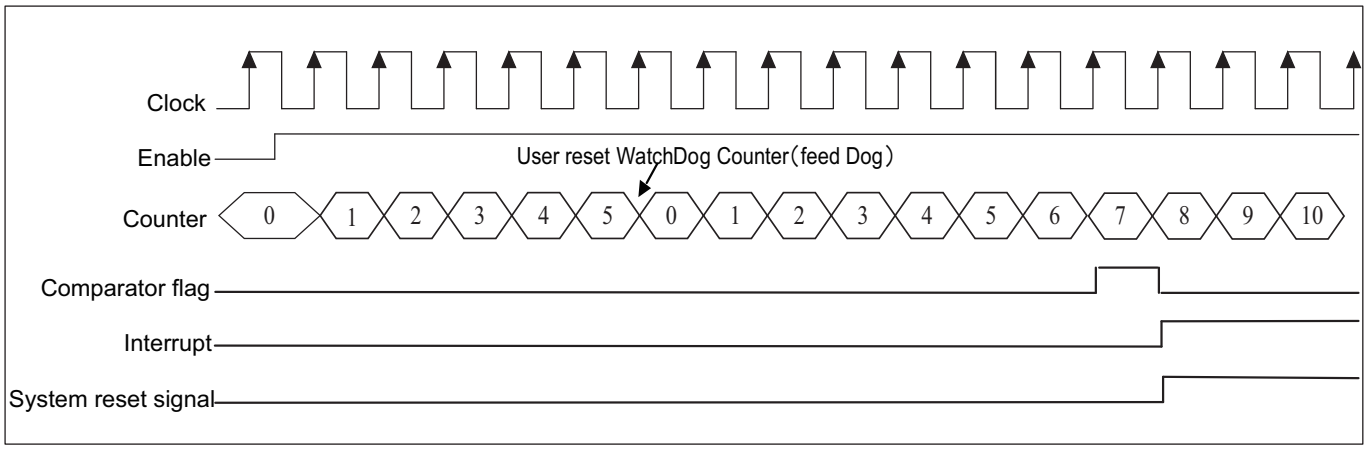

Figure 13.4: Watchdog timing

13.3.4 Alarm setting

Each counter has three comparison values, and can set whether each comparison value triggers an alarm interrupt. When the counter matches the comparison value and the setting will alarm, the counter will notify the processor through the interrupt.

The software can read through the APB whether an alarm has occurred and which comparison value triggered the alarm interrupt. When the alarm interrupt is cleared, the alarm status is also cleared simultaneously.

13.3.5 Watchdog alarm

A comparison value can be set for each counter. When the software fails to reset the watchdog counter to zero due to a system error, which causes the watchdog counter to exceed the comparison value, a watchdog alarm is triggered. There are two types of alarms. The first is to perform necessary actions through interrupt notification software. The second is to enter the system watchdog reset. When the watchdog reset is triggered, it will notify the system reset controller and prepare for system reset. When everything is ready, enter the system watchdog reset. It is worth noting

that software can read the WSR register through APB to know if a watchdog system reset has occurred.

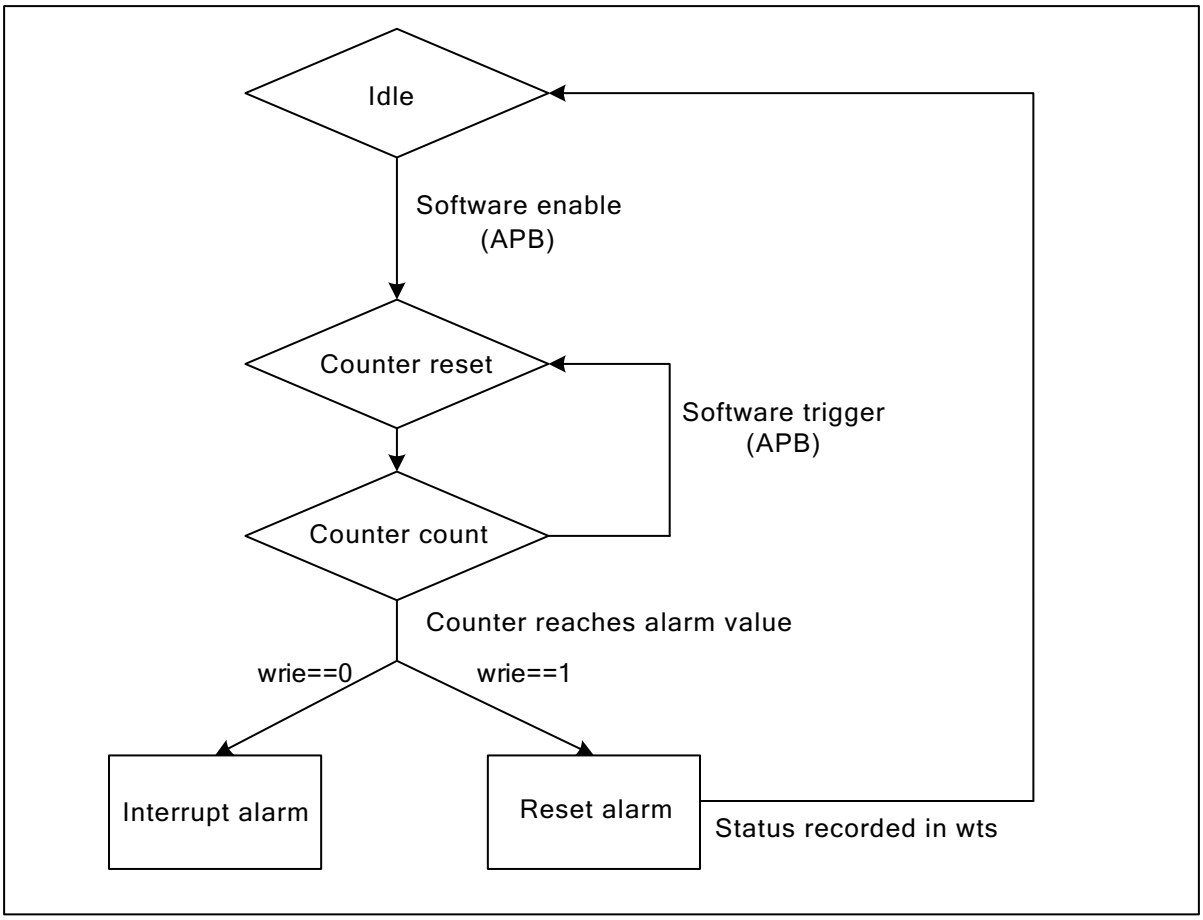

Figure 13.5: Watchdog alarm mechanism

13.4 Register description

Name	Description
TCCR	Timer clock source configuration register
TMR2_0	Timer2 match register 0
TMR2_1	Timer2 match register 1
TMR2_2	Timer2 match register 2
TMR3_0	Timer3 match register 0
TMR3_1	Timer3 match register 1
TMR3_2	Timer3 match register 2
TCR2	Timer2 counter register
TCR3	Timer3 counter register

Name	Description
TMSR2	Timer2 match register status
TMSR3	Timer3 match register status
TIER2	Timer2 match interrupt enable register
TIER3	Timer3 match interrupt enable register
TPLVR2	Timer2 pre-load value register
TPLVR3	Timer3 pre-load value register
TPLCR2	Timer2 pre-load control register
TPLCR3	Timer3 pre-load control register
WMER	WDT reset/interrupt mode register
WMR	WDT counter match value register
WVR	WDT counter value register
WSR	WDT timer reset indication register
TICR2	Timer2 Interrupt clear control register
TICR3	Timer3 Interrupt clear control register
WICR	WDT Interrupt clear register
TCER	Timer count enable register
TCMR	Timer count mode register
TILR2	Timer2 match interrupt mode register
TILR3	Timer3 match interrupt mode register
WCR	WDT timer count reset register
WFAR	WDT access key1 register
WSAR	WDT access key2 register
TCVWR2	Timer2 capture value of counter register
TCVWR3	Timer3 capture value of counter register
TCVSYN2	Timer2 synchronous value of counter register
TCVSYN3	Timer3 synchronous value of counter register
TCDR	WDT/Timer clock division register

13.4.1 TCCR

Address：0x4000a500

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						CSWDT		RSVD	CS2		RSVD	CS1		RSVD	

Bits	Name	Type	Reset	Description
31:10	RSVD			
9:8	CSWDT	R/W	2'd0	Clock Source for Timer #1/#2/#3/WDT 2'd0 - fclk 2'd1 - f32k_clk 2'd2 - 1 kHz 2'd3 - PLL 32MHz
7	RSVD			
6:5	CS2	R/W	2'd0	Clock Source for Timer #1/#2/#3/WDT 2'd0 - fclk 2'd1 - f32k_clk 2'd2 - 1 kHz 2'd3 - PLL 32MHz
4	RSVD			
3:2	CS1	R/W	2'd0	Clock Source for Timer #1/#2/#3/WDT 2'd0 - fclk 2'd1 - f32k_clk 2'd2 - 1 kHz 2'd3 - PLL 32MHz
1:0	RSVD			

13.4.2 TMR2_0

Address：0x4000a510

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR20															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR20															

Bits	Name	Type	Reset	Description
31:0	TMR20	R/W	32'hffffffff	Timer2 match register 0

13.4.3 TMR2_1

Address：0x4000a514

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR21															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR21															

Bits	Name	Type	Reset	Description
31:0	TMR21	R/W	32'hffffffff	Timer2 match register 1

13.4.4 TMR2_2

Address：0x4000a518

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR22															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR22															

Bits	Name	Type	Reset	Description
31:0	TMR22	R/W	32'hffffffff	Timer2 match register 2

13.4.5 TMR3_0

Address：0x4000a51c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR30															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR30															

Bits	Name	Type	Reset	Description
31:0	TMR30	R/W	32'hffffffff	Timer3 match register 0

13.4.6 TMR3_1

Address：0x4000a520

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR31															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR31															

Bits	Name	Type	Reset	Description
31:0	TMR31	R/W	32'hffffffff	Timer3 match register 1

13.4.7 TMR3_2

Address：0x4000a524

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR32															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR32															

Bits	Name	Type	Reset	Description
31:0	TMR32	R/W	32'hffffffff	Timer3 match register 2

13.4.8 TCR2

Address：0x4000a52c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCR2COUT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCR2COUT															

Bits	Name	Type	Reset	Description
31:0	TCR2COUT	R	32'h0	Timer2 counter register

13.4.9 TCR3

Address：0x4000a530

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCR3COUT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCR3COUT															

Bits	Name	Type	Reset	Description
31:0	TCR3COUT	R	32'h0	Timer3 counter register

13.4.10 TMSR2

Address：0x4000a538

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													T2MR2S	T2MR1S	T2MR0S

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	T2MR2S	R	1'b0	Timer2 match register 2 status/Clear interrupt would also clear this bit
1	T2MR1S	R	1'b0	Timer2 match register 1 status/Clear interrupt would also clear this bit
0	T2MR0S	R	1'b0	Timer2 match register 0 status/Clear interrupt would also clear this bit

13.4.11 TMSR3

Address：0x4000a53c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													T3MR2S	T3MR1S	T3MR0S

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	T3MR2S	R	1'b0	Timer3 match register 2 status/Clear interrupt would also clear this bit
1	T3MR1S	R	1'b0	Timer3 match register 1 status/Clear interrupt would also clear this bit
0	T3MR0S	R	1'b0	Timer3 match register 0 status/Clear interrupt would also clear this bit

13.4.12 TIER2

Address：0x4000a544

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TIER22	TIER21	TIER20

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TIER22	R/W	1'b0	Timer2 match register 2 interrupt enable register
1	TIER21	R/W	1'b0	Timer2 match register 1 interrupt enable register
0	TIER20	R/W	1'b0	Timer2 match register 0 interrupt enable register

13.4.13 TIER3

Address：0x4000a548

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TIER32	TIER31	TIER30

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TIER32	R/W	1'b0	Timer3 match register 2 interrupt enable register
1	TIER31	R/W	1'b0	Timer3 match register 1 interrupt enable register
0	TIER30	R/W	1'b0	Timer3 match register 0 interrupt enable register

13.4.14 TPLVR2

Address：0x4000a550

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPLVR2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPLVR2															

Bits	Name	Type	Reset	Description
31:0	TPLVR2	R/W	32'h0	Timer2 pre-load value register

13.4.15 TPLVR3

Address：0x4000a554

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPLVR3															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPLVR3															

Bits	Name	Type	Reset	Description
31:0	TPLVR3	R/W	32'h0	Timer3 pre-load value register

13.4.16 TPLCR2

Address：0x4000a55c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														TPLCR2	

Bits	Name	Type	Reset	Description
31:2	RSVD			
1:0	TPLCR2	R/W	2'h0	Timer2 pre-load control register 2'd0 - No pre-load 2'd1 - Pre-load with match comparator 0 2'd2 - Pre-load with match comparator 1 2'd3 - Pre-load with match comparator 2

13.4.17 TPLCR3

Address：0x4000a560

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														TPLCR3	

Bits	Name	Type	Reset	Description
31:2	RSVD			
1:0	TPLCR3	R/W	2'h0	Timer3 pre-load control register 2'd0 - No pre-load 2'd1 - Pre-load with match comparator 0 2'd2 - Pre-load with match comparator 1 2'd3 - Pre-load with match comparator 2

13.4.18 WMER

Address：0x4000a564

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

Bits	Name	Type	Reset	Description
31:2	RSVD			
1	WRIE	R/W	1'b0	WDT reset/interrupt mode register 1'b0 - WDT expiration to generate interrupt 1'b1 - WDT expiration to generate reset source
0	WE	R/W	1'b0	WDT enable register

13.4.19 WMR

Address：0x4000a568

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	WMR	R/W	16'hffff	WDT counter match value register

13.4.20 WVR

Address：0x4000a56c

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	WVR	R	16'h0	WDT counter value register

13.4.21 WSR

Address：0x4000a570

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

Bits	Name	Type	Reset	Description
31:1	RSVD			
0	WTS	R/W	1'b0	WDT timer reset indication, Indicates that reset was caused by the WDT. (Write)1'b0 - clear the WDT reset status (Write)1'b1 - no affect (Read)1'b0 - Watchdog timer did not cause reset because this bit was cleare (Read)1'b1 - Watchdog timer caused reset

13.4.22 TICR2

Address：0x4000a578

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TCLR22	TCLR21	TCLR20

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TCLR22	W	1'b0	Timer2 Interrupt clear for match comparator 2
1	TCLR21	W	1'b0	Timer2 Interrupt clear for match comparator 1
0	TCLR20	W	1'b0	Timer2 Interrupt clear for match comparator 0

13.4.23 TICR3

Address：0x4000a57c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TCLR32	TCLR31	TCLR30

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TCLR32	W	1'b0	Timer3 Interrupt clear for match comparator 2
1	TCLR31	W	1'b0	Timer3 Interrupt clear for match comparator 1
0	TCLR30	W	1'b0	Timer3 Interrupt clear for match comparator 0

13.4.24 WICR

Address：0x4000a580

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

Bits	Name	Type	Reset	Description
31:1	RSVD			
0	WICLR	W	1'b0	WDT Interrupt clear register

13.4.25 TCER

Address：0x4000a584

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TIM3EN	TIM2EN	RSVD

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TIM3EN	R/W	1'b0	Timer3 count enable
1	TIM2EN	R/W	1'b0	Timer2 count enable
0	RSVD			

13.4.26 TCMR

Address：0x4000a588

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TIM3 MODE	TIM2 MODE	RSVD

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TIM3MODE	R/W	1'b0	Timer1/2/3 count mode register 1'b0 - pre-load mode 1'b1 - free run mode
1	TIM2MODE	R/W	1'b0	Timer1/2/3 count mode register 1'b0 - pre-load mode 1'b1 - free run mode
0	RSVD			

13.4.27 TILR2

Address：0x4000a590

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

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TILR22	R/W	1'b0	Timer2 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt
1	TILR21	R/W	1'b0	Timer2 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt
0	TILR20	R/W	1'b0	Timer2 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt

13.4.28 TILR3

Address：0x4000a594

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TILR32	TILR31	TILR30

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TILR32	R/W	1'b0	Timer3 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt
1	TILR31	R/W	1'b0	Timer3 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt
0	TILR30	R/W	1'b0	Timer3 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt

13.4.29 WCR

Address：0x4000a598

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

Bits	Name	Type	Reset	Description
31:1	RSVD			
0	WCR	W	1'b0	WDT timer count reset register

13.4.30 WFAR

Address：0x4000a59c

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	WFAR	W	16'b0	WDT access key1 - 16'hBABA

13.4.31 WSAR

Address：0x4000a5a0

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	WSAR	W	16'b0	WDT access key2 - 16'hEB10

13.4.32 TCVWR2

Address：0x4000a5a8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCVWR2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCVWR2															

Bits	Name	Type	Reset	Description
31:0	TCVWR2	R	32'h0	Timer2 capture value of counter

13.4.33 TCVWR3

Address：0x4000a5ac

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCVWR3															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCVWR3															

Bits	Name	Type	Reset	Description
31:0	TCVWR3	R	32'h0	Timer3 capture value of counter

13.4.34 TCVSYN2

Address：0x4000a5b4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCVSYN2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCVSYN2															

Bits	Name	Type	Reset	Description
31:0	TCVSYN2	R	32'h0	Timer2 synchronous value of counter

13.4.35 TCVSYN3

Address：0x4000a5b8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCVSYN3															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCVSYN3															

Bits	Name	Type	Reset	Description
31:0	TCVSYN3	R	32'h0	Timer3 synchronous value of counter

13.4.36 TCDR

Address：0x4000a5bc

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WCDR								TCDR3							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCDR2								RSVD							

Bits	Name	Type	Reset	Description
31:24	WCDR	R/W	8'h0	WDT clock division value register
23:16	TCDR3	R/W	8'h0	Timer3 clock division value register
15:8	TCDR2	R/W	8'h0	Timer2 clock division value register
7:0	RSVD			

14 QDEC

14.1 QDEC introduction

The quadrature decoder is used to decode the two sets of pulses with a phase difference of 90 degrees generated by the dual-path rotary encoder into the corresponding speed and direction of rotation.

14.2 QDEC main features

- Three sets of QDEC are available
- The clock source of QDEC can be 32K (f32k_clk) or 32M (xclk). It is recommended to select 32M as the clock source during normal operation, and 32K is recommended when entering sleep mode and wish to be awakened by QDEC.
- Supports 5-digit frequency division value, which can be divided from 1 to 32.
- 16-bit pulse count range (-32768~32767 pulse/sample)
- 12 configurable sample periods (32us~131ms per sample at 1MHz)
- 16-bit configurable report period (0~65535 sample/report)
- Built-in a LED function that can flash with sampling (LED on/off 0~511 us/sample)
- Interrupt can be configured (sample interrupt, report interrupt, error interrupt, overflow interrupt)
- Can be configured as a wake-up source for PDS (clock source needs to be configured as 32k)

14.3 QDEC function description

The expected operating frequency of QDEC is 1MHz, and the faster the detection speed, the higher the operating frequency required.

Each sampling will decode the A/B two-phase pulse output by the encoder into high and low levels. Compare the previous sampling results to get the current encoder rotation direction and pulse count change (clockwise rotation +1,

counterclockwise rotation -1 , No change, no change, error report and count). After the sampling times set by the report, the rotation direction and pulse count of the encoder during this period can be obtained, and the average value of the rotational speed direction during the report period can be solved accordingly.

The period of each sampling can be configured. When the working frequency is 1MHz, the minimum is 32us for one sampling, and the maximum is 131ms for one sampling.

The interrupt can be configured as a single sampling end trigger (sample interrupt) and multiple sampling end trigger (report interrupt) to flexibly measure the speed.

Configurable LED blinking function, blinking frequency=LED cycle/sampling cycle, each blinking on/off is determined by the LED polarity.

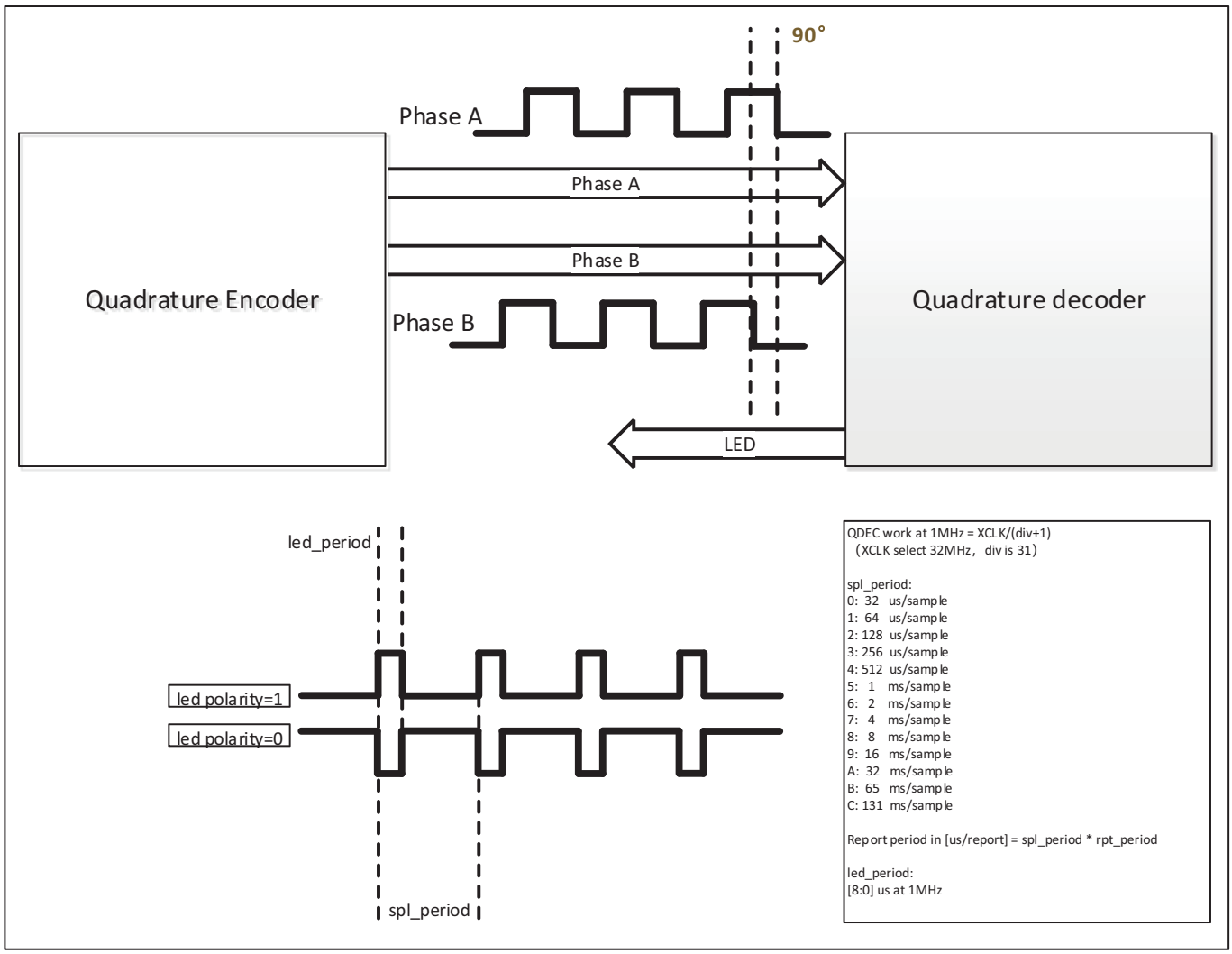

Figure 14.1: QDEC functional block diagram

14.4 Register description

Name	Description
qdec0_ctrl0	QDEC0 control0
qdec0_ctrl1	QDEC0 control1
qdec0_value	QDEC0 value
qdec0_int_en	QDEC0 interrupt enable
qdec0_int_sts	QDEC0 interrupt status
qdec0_int_clr	QDEC0 interrupt clear
qdec1_ctrl0	QDEC1 control0
qdec1_ctrl1	QDEC1 control1
qdec1_value	QDEC1 value
qdec1_int_en	QDEC1 interrupt enable
qdec1_int_sts	QDEC1 interrupt status
qdec1_int_clr	QDEC1 interrupt clear
qdec2_ctrl0	QDEC2 control0
qdec2_ctrl1	QDEC2 control1
qdec2_value	QDEC2 value
qdec2_int_en	QDEC2 interrupt enable
qdec2_int_sts	QDEC2 interrupt status
qdec2_int_clr	QDEC2 interrupt clear

14.4.1 qdec0_ctrl0

Address：0x4000a800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD				RPT											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPT				SPL				DEGCNT				DEG EN	LED POL	LED EN	QDEC EN

Bits	Name	Type	Reset	Description
31:28	RSVD			

Bits	Name	Type	Reset	Description
27:12	RPT	R/W	16'd10	"RPT" report period in [samples/report]. Specifies the number of samples to be accumulated in the ACC1 register before the RPT_RDY and DBL_RDY events can be generated "RPT_US" report period in [us/report] = SP * RP
11:8	SPL	R/W	4'h2	"SPL" sample period in [us/sample]. The SAMPLE register will be updated for every new sample (at 1MHz) 0: 32 us 1: 64 2: 128 3: 256 4: 512 5: 1 ms 6: 2 7: 4 8: 8 9: 16 A: 32 B: 65 C: 131
7:4	DEGCNT	R/W	0	deglitch strength
3	DEGEN	R/W	0	deglitch enable
2	LEDPOL	R/W	1	qdec led polarity
1	LEDEN	R/W	0	qdec led enable
0	QDECEN	R/W	0	qdec enable

14.4.2 qdec0_ctrl1

Address：0x4000a804

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

Bits	Name	Type	Reset	Description
31:25	RSVD			
24:16	LEDPED	R/W	0	Period in us the LED is switched on prior to sampling
15:4	RSVD			

Bits	Name	Type	Reset	Description
3	IS	R/W	0	input a/b swap
2	RPT	R/W	0	rpt option 0: Count time only if sample change 1: Continue time
1	SPL	R/W	0	spl option 0: Stop sample if rpt_rdy 1: Continue sample
0	ACC	R/W	1	acc option 0: Stop accumulate if overflow 1: Continue accumulate

14.4.3 qdec0_value

Address：0x4000a808

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SPLVAL		RSVD								ACC2VAL			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC1VAL															

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:28	SPLVAL	R	0	Sample value. Direction of last change 00: no change 01: clockwise 11: counter-clockwise 10: Error
27:20	RSVD			
19:16	ACC2VAL	R	0	Double error accumulation (0 15)
15:0	ACC1VAL	R	0	Sample accumulation (-32768 32767) clockwise +1, counter-clockwise -1

14.4.4 qdec0_int_en

Address：0x4000a810

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3	OFEN	R/W	0	overflow interrupt enable
2	DREN	R/W	0	double error interrupt enable
1	SREN	R/W	0	sample interrupt enable
0	RREN	R/W	1	report interrupt enable

14.4.5 qdec0_int_sts

Address：0x4000a814

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3	OFSTS	R	0	ACC1 or ACC2 overflow
2	DRSTS	R	0	ACC2 double error
1	SRSTS	R	0	Event being generated for every new sample value written to the SAMPLE register
0	RRSTS	R	0	Non-null report ready

14.4.6 qdec0_int_clr

Address：0x4000a818

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

Bits	Name	Type	Reset	Description
31:4	RSVD			

Bits	Name	Type	Reset	Description
3	OFCL	W1C	0	overflow interrupt clear
2	DRCL	W1C	0	double error interrupt clear
1	SRCL	W1C	0	sample interrupt clear
0	RRCL	W1C	0	report interrupt clear

14.4.7 qdec1_ctrl0

Address：0x4000a840

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD				RPT											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPT				SPL				DEGCNT				DEG EN	LED POL	LED EN	QDEC EN

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:12	RPT	R/W	16'd10	"RPT" report period in [samples/report]. Specifies the number of samples to be accumulated in the ACC1 register before the RPT_RDY and DBL_RDY events can be generated "RPT_US" report period in [us/report] = SP * RP
11:8	SPL	R/W	4'h2	"SPL" sample period in [us/sample]. The SAMPLE register will be updated for every new sample (at 1MHz) 0: 32 us 1: 64 2: 128 3: 256 4: 512 5: 1 ms 6: 2 7: 4 8: 8 9: 16 A: 32 B: 65 C: 131
7:4	DEGCNT	R/W	0	deglitch strength
3	DEGEN	R/W	0	deglitch enable

Bits	Name	Type	Reset	Description
2	LEDPOL	R/W	1	qdec led polarity
1	LEDEN	R/W	0	qdec led enable
0	QDECEN	R/W	0	qdec enable

14.4.8 qdec1_ctrl1

Address：0x4000a844

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

Bits	Name	Type	Reset	Description
31:25	RSVD			
24:16	LEDPED	R/W	0	Period in us the LED is switched on prior to sampling
15:4	RSVD			
3	IS	R/W	0	input a/b swap
2	RPT	R/W	0	rpt option 0: Count time only if sample change 1: Continue time
1	SPL	R/W	0	spl option 0: Stop sample if rpt_rdy 1: Continue sample
0	ACC	R/W	1	acc option 0: Stop accumulate if overflow 1: Continue accumulate

14.4.9 qdec1_value

Address：0x4000a848

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SPLVAL		RSVD								ACC2VAL			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC1VAL															

Bits	Name	Type	Reset	Description
31:30	RSVD			

Bits	Name	Type	Reset	Description
29:28	SPLVAL	R	0	Sample value. Direction of last change 00: no change 01: clockwise 11: counter-clockwise 10: Error
27:20	RSVD			
19:16	ACC2VAL	R	0	Double error accumulation (0 15)
15:0	ACC1VAL	R	0	Sample accumulation (-32768 32767) clockwise +1, counter-clockwise -1

14.4.10 qdec1_int_en

Address：0x4000a850

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3	OFEN	R/W	0	overflow interrupt enable
2	DREN	R/W	0	double error interrupt enable
1	SREN	R/W	0	sample interrupt enable
0	RREN	R/W	1	report interrupt enable

14.4.11 qdec1_int_sts

Address：0x4000a854

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3	OFSTS	R	0	ACC1 or ACC2 overflow
2	DRSTS	R	0	ACC2 double error
1	SRSTS	R	0	Event being generated for every new sample value written to the SAMPLE register
0	RRSTS	R	0	Non-null report ready

14.4.12 qdec1_int_clr

Address：0x4000a858

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3	OFCL	W1C	0	overflow interrupt clear
2	DRCL	W1C	0	double error interrupt clear
1	SRCL	W1C	0	sample interrupt clear
0	RRCL	W1C	0	report interrupt clear

14.4.13 qdec2_ctrl0

Address：0x4000a880

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD				RPT											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPT				SPL				DEGCNT				DEG EN	LED POL	LED EN	QDEC EN

Bits	Name	Type	Reset	Description
31:28	RSVD			

Bits	Name	Type	Reset	Description
27:12	RPT	R/W	16'd10	"RPT" report period in [samples/report]. Specifies the number of samples to be accumulated in the ACC1 register before the RPT_RDY and DBL_RDY events can be generated "RPT_US" report period in [us/report] = SP * RP
11:8	SPL	R/W	4'h2	"SPL" sample period in [us/sample]. The SAMPLE register will be updated for every new sample (at 1MHz) 0: 32 us 1: 64 2: 128 3: 256 4: 512 5: 1 ms 6: 2 7: 4 8: 8 9: 16 A: 32 B: 65 C: 131
7:4	DEGCNT	R/W	0	deglitch strength
3	DEGEN	R/W	0	deglitch enable
2	LEDPOL	R/W	1	qdec led polarity
1	LEDEN	R/W	0	qdec led enable
0	QDECEN	R/W	0	qdec enable

14.4.14 qdec2_ctrl1

Address：0x4000a884

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

Bits	Name	Type	Reset	Description
31:25	RSVD			
24:16	LEDPED	R/W	0	Period in us the LED is switched on prior to sampling
15:4	RSVD			

Bits	Name	Type	Reset	Description
3	IS	R/W	0	input a/b swap
2	RPT	R/W	0	rpt option 0: Count time only if sample change 1: Continue time
1	SPL	R/W	0	spl option 0: Stop sample if rpt_rdy 1: Continue sample
0	ACC	R/W	1	acc option 0: Stop accumulate if overflow 1: Continue accumulate

14.4.15 qdec2_value

Address：0x4000a888

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SPLVAL		RSVD								ACC2VAL			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC1VAL															

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:28	SPLVAL	R	0	Sample value. Direction of last change 00: no change 01: clockwise 11: counter-clockwise 10: Error
27:20	RSVD			
19:16	ACC2VAL	R	0	Double error accumulation (0 15)
15:0	ACC1VAL	R	0	Sample accumulation (-32768 32767) clockwise +1, counter-clockwise -1

14.4.16 qdec2_int_en

Address：0x4000a890

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3	OFEN	R/W	0	overflow interrupt enable
2	DREN	R/W	0	double error interrupt enable
1	SREN	R/W	0	sample interrupt enable
0	RREN	R/W	1	report interrupt enable

14.4.17 qdec2_int_sts

Address：0x4000a894

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3	OFSTS	R	0	ACC1 or ACC2 overflow
2	DRSTS	R	0	ACC2 double error
1	SRSTS	R	0	Event being generated for every new sample value written to the SAMPLE register
0	RRSTS	R	0	Non-null report ready

14.4.18 qdec2_int_clr

Address：0x4000a898

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

Bits	Name	Type	Reset	Description
31:4	RSVD			

Bits	Name	Type	Reset	Description
3	OFCL	W1C	0	overflow interrupt clear
2	DRCL	W1C	0	double error interrupt clear
1	SRCL	W1C	0	sample interrupt clear
0	RRCL	W1C	0	report interrupt clear

15 KeyScan

15.1 KYS introduction

The KYS (Key Scan) module is used to scan the matrix keyboard to obtain key values. It periodically scans the keyboard connection pins and generates an interrupt as soon as it finds a key press.

15.2 KYS main features

- Configurable number of rows and columns, up to 8 rows*20 columns of matrix keyboard
- Store up to four key values
- Support key interrupt

15.3 KYS function description

15.3.1 Configurable number of rows and columns

The number of rows and columns of the matrix keyboard can be configured through the bits <ROW_NUM> and <COL_NUM> of the register KS_CTRL, and the configuration value is the actual value minus one. The maximum number of rows supports 8 rows, and the maximum number of columns supports 20 columns.

15.3.2 GPIO selection

Since the button scan is fixed from the GPIO pins with the functions of ROW_0 and COL_0, the row pins need to be selected sequentially from the GPIO with the function of ROW_0, that is, from any one of GPIO0/GPIO8/GPIO16/GPIO24. The column pins need to be selected in order from the GPIO with the function of COL_0, that is, from any one of GPIO0/GPIO20.

15.3.3 Key value

The key value will be stored in the register KEYCODE_VALUE, every 8 bits is a key value, and the first key value will be stored in the lowest 8 bits. When the bit corresponding to the serial number in the register KEYCODE_CLR is

set to one, the corresponding key value and interrupt flag bit will be cleared. The row number corresponding to the key value = key value% total number of rows, the column number corresponding to the key value = key value / total number of rows.

15.3.4 Interrupt

When a key press is detected, the interrupt flag bit will be set and an interrupt will be generated at the same time.

15.4 Register description

Name	Description
ks_ctrl	Keyscan control
ks_int_en	Keyscan interrupt enable
ks_int_sts	Keyscan interrupt status
keycode_clr	Keycode clear
keycode_value	Keycode value

15.4.1 ks_ctrl

Address：0x4000a900

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD								COLNUM				RSVD	ROWNUM		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						RCEXT		DEGCNT				DEG EN	GHEN	RSVD	KSEN

Bits	Name	Type	Reset	Description
31:25	RSVD			
24:20	COLNUM	R/W	5'd19	col_num + 1
19	RSVD			
18:16	ROWNUM	R/W	3'd7	row_num + 1
15:10	RSVD			
9:8	RCEXT	R/W	2'd3	idle duration between column scans
7:4	DEGCNT	R/W	0	deglitch count
3	DEGEN	R/W	0	deglitch

Bits	Name	Type	Reset	Description
2	GHEN	R/W	0	ghost key event detection
1	RSVD			
0	KSEN	R/W	0	Keyscan en

15.4.2 ks_int_en

Address：0x4000a910

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

Bits	Name	Type	Reset	Description
31:1	RSVD			
0	KSITEN	R/W	1	Keyscan interrupt enable

15.4.3 ks_int_sts

Address：0x4000a914

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3:0	KCVALID	R	0	Keycode valid

15.4.4 keycode_clr

Address：0x4000a918

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

Bits	Name	Type	Reset	Description
31:4	RSVD			
3:0	KCCLR	W1C	0	Keycode clear

15.4.5 keycode_value

Address：0x4000a91c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KC3								KC2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KC1								KC0							

Bits	Name	Type	Reset	Description
31:24	KC3	R	8'hff	Col = keycode / (row_num+1) Row = keycode
23:16	KC2	R	8'hff	Col = keycode / (row_num+1) Row = keycode
15:8	KC1	R	8'hff	Col = keycode / (row_num+1) Row = keycode
7:0	KC0	R	8'hff	Col = keycode / (row_num+1) Row = keycode

16 I2S

16.1 I2S introduction

I2S is an interface standard for the transmission of digital audio data, and two groups (left and right channel) data are transmitted in a sequential manner.

I2S separates the clock signal and the data signal for transmission, so that the receiving end does not need to restore the clock from the data signal, thereby reducing the design difficulty of the receiving end.

16.2 I2S main features

- Support master mode and slave mode
- Support Left-justified/Right-justified/DSP and other data formats
- Support 8/16/24/32 bit data width
- In addition to mono/dual-channel mode, four-channel mode is also supported
- Support dynamic mute switching function
- The width of the data transmission FIFO is 32 bits and the depth is 16
- The width of the data receiving FIFO is 32 bits and the depth is 16

16.3 I2S function description

Table 16.1: I2S pin list

Name	Type	Description
I2Sx_DI	input	Serial data input
I2Sx_DO	output	Serial data output
I2Sx_BCLK	input/output	Synchronous transmission clock, output when used as a master, and input when used as a slave
I2Sx_FS	input/output	Data start/end signal, output when used as master, input when used as slave

16.4 Register description

Name	Description
i2s_config	I2S configuration
i2s_int_sts	I2S interrupt status
i2s_bclk_config	I2S clock configuration
i2s_fifo_config_0	I2S FIFO configuration0
i2s_fifo_config_1	I2S DMA FIFO configuration
i2s_fifo_wdata	I2S FIFO write data
i2s_fifo_rdata	I2S FIFO read data
i2s_io_config	I2S IO configuration

16.4.1 i2s_config

Address：0x4000aa00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD						OFSEN	OFSCNT					MONORCH	ENDIAN	MODE	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATASIZE		FRAMSIZE		RSVD			FSCHMODE	FS4C	FS1T	MUTE	MONO	RXDEN	TXDEN	SLAEN	MASEN

Bits	Name	Type	Reset	Description
31:26	RSVD			

Bits	Name	Type	Reset	Description
25	OFSEN	R/W	1'b0	Offset enable 1'b0: Disabled, 1'b1: Enabled
24:20	OFSCNT	R/W	5'd0	Offset cycle count (unit: cycle of I2S BCLK) 5'd0: 1 cycle 5'd1: 2 cycles ...
19	MONORCH	R/W	1'b0	RX mono mode channel select signal 1'b0: L-channel 1'b1: R-channel
18	ENDIAN	R/W	1'b0	Data endian (bit reverse) 1'b0: MSB goes out first, 1'b1: LSB goes out first
17:16	MODE	R/W	2'd0	2'd0: Left-Justified, 2'd1: Right-Justified, 2'd2: DSP, 2'd3: Reserved
15:14	DATASIZE	R/W	2'd1	Data bit width of each channel 2'd0: 8, 2'd1: 16, 2'd2: 24, 2'd3: 32 (bits)
13:12	FRAMSIZE	R/W	2'd1	Frame size of each channel 2'd0: 8, 2'd1: 16, 2'd2: 24, 2'd3: 32 (cycles)
11:9	RSVD			
8	FSCHMODE	R/W	1'b0	1'b0: FS 2-channel mode, 1'b1: FS 3-channel mode (DSP mode only) Note: cr_fs_3ch_mode & cr_fs_4ch_mode should NOT be enabled at the same time Note: cr_mono_mode & cr_fifo_lr_merge will be invalid in 3-channel mode Note: When 3-channel mode is enabled, frame_size must equal data_size
7	FS4C	R/W	1'b0	1'b0: FS 2-channel mode, 1'b1: FS 4-channel mode (DSP mode only) Note: cr_fs_3ch_mode & cr_fs_4ch_mode should NOT be enabled at the same time Note: When 4-channel mode is enabled, frame_size must equal data_size
6	FS1T	R/W	1'b0	1'b0: FS high/low is even, 1'b1: FS only asserts for 1 cycle
5	MUTE	R/W	1'b0	1'b0: Normal mode, 1'b1: Mute mode
4	MONO	R/W	1'b0	1'b0: Stereo mode, 1'b1: Mono mode Note: csr_mono_mode & csr_fifo_lr_merge should NOT be enabled at the same time
3	RXDEN	R/W	1'b0	Enable signal of I2S RXD signal

Bits	Name	Type	Reset	Description
2	TXDEN	R/W	1'b0	Enable signal of I2S TXD signal
1	SLAEN	R/W	1'b0	Enable signal of I2S Slave function, cannot enable both csr_i2s_m_en & csr_i2s_s_en
0	MASEN	R/W	1'b0	Enable signal of I2S Master function, cannot enable both csr_i2s_m_en & csr_i2s_s_en

16.4.2 i2s_int_sts

Address：0x4000aa04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD					FER EN	RXF EN	TXF EN	RSVD							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					FER MASK	RXF MASK	TXF MASK	RSVD					FER INT	RXF INT	TXF INT

Bits	Name	Type	Reset	Description
31:27	RSVD			
26	FEREN	R/W	1'b1	Interrupt enable of i2s_fer_int
25	RXFEN	R/W	1'b1	Interrupt enable of i2s_rxf_int
24	TXFEN	R/W	1'b1	Interrupt enable of i2s_txf_int
23:11	RSVD			
10	FERMASK	R/W	1'b1	Interrupt mask of i2s_fer_int
9	RXFMASK	R/W	1'b1	Interrupt mask of i2s_rxf_int
8	TXFMASK	R/W	1'b1	Interrupt mask of i2s_txf_int
7:3	RSVD			
2	FERINT	R	1'b0	I2S TX/RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
1	RXFINT	R	1'b0	I2S RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto-cleared when data is popped
0	TXFINT	R	1'b0	I2S TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed

16.4.3 i2s_bclk_config

Address：0x4000aa10

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:16	BDIVH	R/W	12'd1	I2S BCLK active high period (unit: cycle of i2s_clk)
15:12	RSVD			
11:0	BDIVL	R/W	12'd1	I2S BCLK active low period (unit: cycle of i2s_clk)

16.4.4 i2s_fifo_config_0

Address：0x4000aa80

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					24BLJ	LREC	LRMG	RUD FLOW	RO FLOW	TUD FLOW	TO FLOW	RXF CLR	TXF CLR	DMA RXEN	DMA TXEN

Bits	Name	Type	Reset	Description
31:11	RSVD			
10	24BLJ	R/W	1'b0	FIFO 24-bit data left-justified mode 1'b0: Right-justified, 8'h0, data[23:0] 1'b1: Left-justified, data[23:0], 8'h0 Note: Valid only when cr_data_size = 2'd2 (24-bit)
9	LREC	R/W	1'b0	The position of L/R channel data within each entry is exchanged if this bit is enabled Can only be enabled if data size is 8 or 16 bits and csr_fifo_lr_merge is enabled

Bits	Name	Type	Reset	Description
8	LRMG	R/W	1'b0	Each FIFO entry contains both L/R channel data if this bit is enabled Can only be enabled if data size is 8 or 16 bits Note: cr_fifo_lr_merge &cr_mono_mode should NOT be enabled at the same time Note: cr_fifo_lr_merge &cr_fifo_l_shift should NOT be enabled at the same time
7	RUDFLOW	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	ROFLOW	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	TUDFLOW	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	TOFLOW	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	RXFCLR	W1C	1'b0	Clear signal of RX FIFO
2	TXFCLR	W1C	1'b0	Clear signal of TX FIFO
1	DMARXEN	R/W	1'b0	Enable signal of dma_rx_req/ack interface
0	DMATXEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface

16.4.5 i2s_fifo_config_1

Address：0x4000aa84

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	RXTH	R/W	4'd0	RX FIFO threshold, dma_rx_req will not be asserted if tx_fifo_cnt is less than this value
23:20	RSVD			
19:16	TXTH	R/W	4'd0	TX FIFO threshold, dma_tx_req will not be asserted if tx_fifo_cnt is less than this value
15:13	RSVD			
12:8	RXCNT	R	5'd0	RX FIFO available count
7:5	RSVD			
4:0	TXCNT	R	5'd16	TX FIFO available count

16.4.6 i2s_fifo_wdata

Address：0x4000aa88

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

Bits	Name	Type	Reset	Description
31:0	FIWD	W	x	I2S FIFO write data

16.4.7 i2s_fifo_rdata

Address：0x4000aa8c

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

Bits	Name	Type	Reset	Description
31:0	FIRD	R	32'h0	I2S FIFO read data

16.4.8 i2s_io_config

Address：0x4000aafc

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DEG EN	DEGCNT			BCLK INV	FS INV	RXD INV	TXD INV

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	DEGEN	R/W	1'b0	Deglitch enable (for all th input pins) 1'b0: Disabled, 1'b1: Enabled

Bits	Name	Type	Reset	Description
6:4	DEGCNT	R/W	3'd0	Deglitch cycle count (unit: cycle of I2S kernel clock) 3'd0: 1 cycle 3'd1: 2 cycles ...
3	BCLKINV	R/W	1'b0	Inverse BCLK signal 0: No inverse, 1: Inverse
2	FSINV	R/W	1'b0	Inverse FS signal 0: No inverse, 1: Inverse
1	RXDINV	R/W	1'b0	Inverse RXD signal 0: No inverse, 1: Inverse
0	TXDINV	R/W	1'b0	Inverse TXD signal 0: No inverse, 1: Inverse

17 Emac

17.1 Emac introduction

The EMAC module is a 10/100Mbps Ethernet MAC (Ethernet Media Access Controller) compatible with IEEE 802.3. It includes status and control register group, transceiver module, transceiver buffer descriptor group, host interface, MDIO, physical layer chip (PHY) interface.

The status and control register group contains the status bits and control bits of the EMAC, which is the interface with the user program, and is responsible for controlling data receiving and sending and reporting the status.

The transceiver module is responsible for obtaining the data frame from the designated memory location according to the control word in the transceiver descriptor, adding the preamble, CRC, and expanding the short frame before sending it out through the PHY; Or receive data from the PHY, and put the data into the designated memory according to the transmit and receive buffer descriptor. Configure related event flags after sending and receiving. If the event interrupt is enabled, the interrupt request will be sent to the host for processing.

The MDIO and MII/RMII interfaces are responsible for communicating with the PHY, including reading and writing PHY registers and sending and receiving data packets.

17.2 Emac main features

- Compatible with the MAC layer functions defined by IEEE 802.3
- Support MII and RMII interface PHY defined by IEEE 802.3
- Interaction with PHY through MDIO
- Support 10Mbps and 100Mbps Ethernet
- Support half duplex and full duplex
- In full duplex mode, support automatic flow control and control frame generation
- Support collision detection and retransmission in half-duplex mode

- Support CRC generation and verification
- Data frame preamble generation and removal
- When sending, automatically expand short data frames
- Detect too long or too short data frame (length limit)
- Can transmit long data frames (> standard Ethernet frame length)
- Automatically discard packets that exceed the number of retransmissions or the frame gap is too small
- Broadcast packet filtering
- Internal RAM for storing up to 128 BD (Buffer Descriptor)
- Various event flags sent/received
- Generate a corresponding interrupt when an event occurs

17.3 Emac function description

The composition of the EMAC module is shown in the figure below.

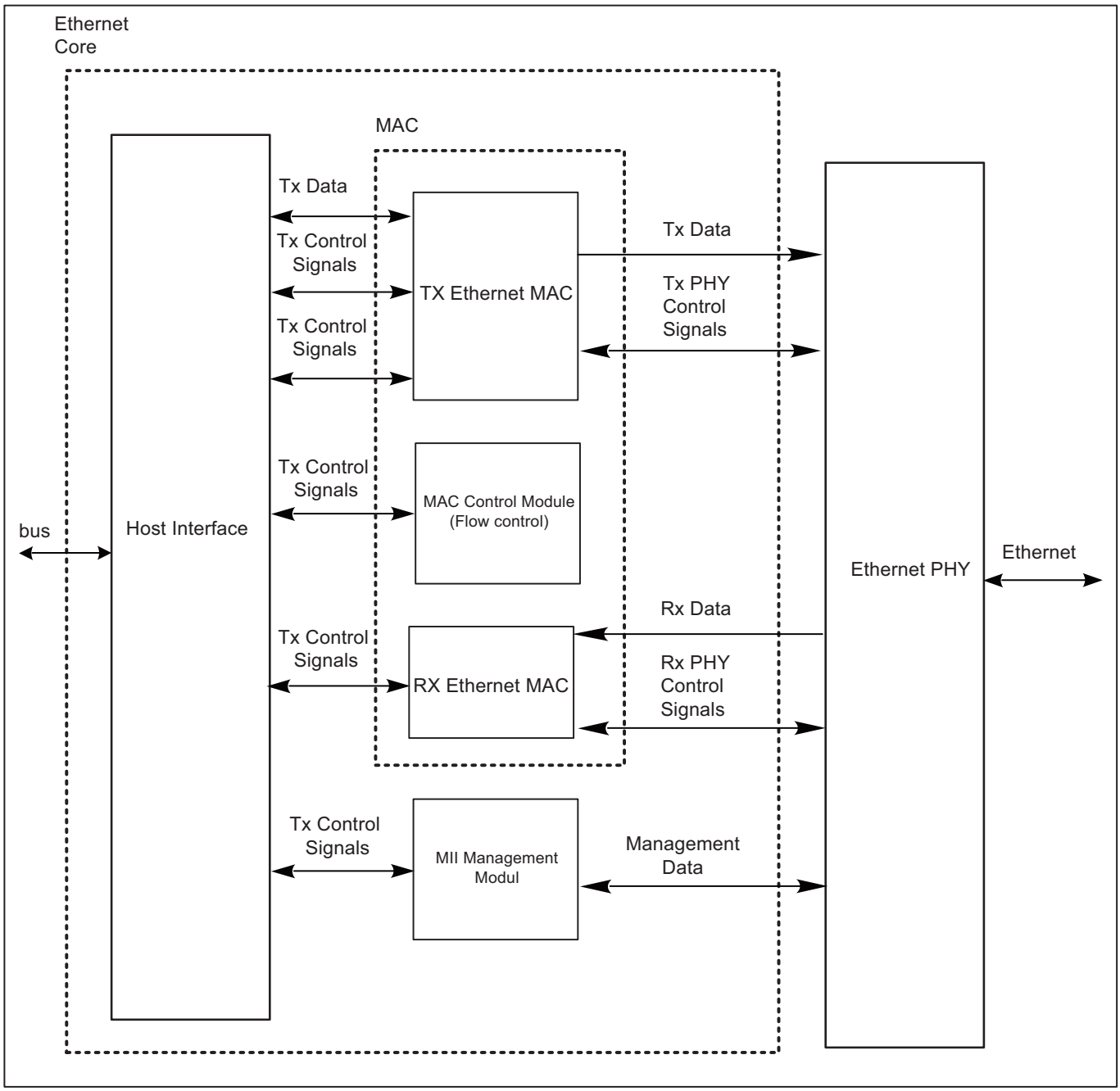

Figure 17.1: EMAC architecture

The module's control register can read and write the PHY register through MDIO to realize configuration, select mode (half/full duplex), initiate negotiation and other operations.

The receiving module filters and checks the received data frame: whether there is a legal preamble, FCS, length, etc. And according to the descriptor, the data frame is stored in the designated buffer address.

The sending module obtains data from the memory according to the data buffer descriptor, adds preamble, FCS, pad, etc., and then sends the data according to the CSMA/CD protocol.

If CRS is detected, the retry will be delayed.

The send and receive buffer descriptor group is connected to the external RAM, which is used to store the Ethernet data frames sent and received. Each descriptor contains the corresponding control status word and the corresponding buffer memory address. There are 128 groups of descriptors, which can be flexibly allocated for sending or receiving.

17.4 Emac clock

The EMAC module needs a clock for synchronous transmission and reception (at 100Mbps, 25MHz (MII) or 50MHz (RMII); at 10Mbps, 2.5MHz). This clock must be synchronized between EMAC and PHY.

17.5 Send and receive buffer descriptor (BD, Buffer Descriptor)

The transceiver buffer descriptor is used to provide the association between the EAMC and the data frame buffer address information, to control the transceiver data frame, and to provide the transceiver status prompt.

Each descriptor is composed of two consecutive words (32 bits). The low address word provides the length, control and status bits of the data frame contained in the buffer; the high address word is a memory pointer.

Specific BD description can refer to the register description chapter.

Note that: For BD, you need to write in word.

The EMAC module supports 128 BDs, which are shared by the sending/receiving logic and can be freely combined. But the sending BD always occupies the previous continuous area (the number is specified by the TXBDNUM field in the MAC_TX_BD_NUM register).

EMAC processes the sending/receiving BD in the order of BD, until it encounters the BD marked as WR, it wraps around to send/receive the respective first BD.

17.6 PHY interaction

The PHY interaction register group provides the commands and data communication methods needed for PHY interaction. EMAC controls the working mode of PHY through MDIO and ensures that the two match (rate, full/half duplex).

The data packet interacts between EMAC and PHY through the MII/RMII interface, and can be selected by RMII_EN in the EMAC mode register (EMAC_MODE). When this bit is 1, select RMII mode, otherwise it is MII mode.

Both MII and RMII modes support the 10Mbps and 100Mbps transmission rates specified in the IEEE 802.3u standard.

The transmission signal description of MII and RMII is shown in the table below.

Table 17.1: Transmission signal

Name	MII	RMII
EXTCK_EREFCK	ETXCK: send clock signal	EREFCK: reference clock
ECRS	ECRS: carrier detection	-

Table 17.1: Transmission signal

Name	MII	RMII
ECOL	ECOL: collision detection	-
ERXDV	ERXDV: data valid	ECRSDV: Carrier detect/data valid
ERX0-ERX3	ERX0-ERX3: 4-bit receive data	ERX0-ERX1: 2-bit receive data
ERXER	ERXER: Receive error indication	ERXER: Receive error indication
ERXCK	ERXCK: Receive clock signal	-
ETXEN	ETXEN: transmit enable	ETXEN: transmit enable
ETX0-ETX3	ETX0-ETX3: 4-bit transmit data	ETX0-ETX1: 2-bit transmit data
ETXER	ETXER: Send error indication	-
EMDC	MDIO Clock	MDIO Clock
EMDIO	MDIO Data Input Output	MDIO Data Input Output

The RMII interface has fewer pins, and a 2-bit data line is used for receiving and sending. At a rate of 100Mbps, a 50MHz reference clock is required.

17.7 Programming process

17.7.1 PHY initialization

- According to the PHY type, set the RMII_EN bit in the EMAC_MODE register to select the appropriate connection method
- Set the MAC address of EMAC to EMAC_MAC_ADDR0 and EMAC_MAC_ADDR1
- Set the appropriate clock for the MDIO part by programming the field CLKDIV in the EMAC_MIIMODE register
- Set the corresponding PHY address to the FIAD field of the register EMAC_MIIADDRESS
- According to the PHY manual, send commands through the EMAC_MIICOMMAND and EMAC_MIITX_DATA registers
- The data read from the PHY will be stored in the EMAC_MIIRX_DATA register
- The status of interaction with PHY commands can be queried through the EMAC_MIISTATUS register

After the basic interaction is completed, the PHY should enter the auto-negotiation state. After the negotiation is completed, program the mode to the FULLD bit in the EMAC_MODE register according to the negotiation result.

17.7.2 Send data frame

- Configure bit fields such as data frame format and interval in the EMAC_MODE register
- By configuring the TXBDNUM field in the EMAC_TX_BD_NUM register to specify the number of BDs used for transmission, the remaining BDs are RX BDs
- Prepare the data frame to be sent in the memory
- Fill in the address of the data frame into the data pointer field (word 1) corresponding to the sent BD
- Clear the status flag in the control and status field (wrod 0) corresponding to the sent BD, and set the control field (CRC enable, PAD enable, interrupt enable, etc.)
- Write the length of the data frame and set the RD field to inform EMAC that this BD data needs to be sent; if necessary, set the IRQ bit to enable interrupts
- In particular, if it is the last BD sent, the WR bit needs to be set, and EMAC will ”wrap around” to the first sent BD for processing after processing this BD
- If there are multiple BDs to be sent, repeat the steps of setting BD to fill all sending BDs
- If you need to enable the transmit interrupt, you also need to configure the TX related bits in the EMAC_INT_MASK register
- Configure the TXEN bit in the EMAC_MODE register to enable transmission
- If the interrupt is enabled, in the interrupt sent, the current BD can be obtained through the TXBDNUM field in the EMAC_TX_BD_NUM register
- Perform corresponding processing according to the current BD status word
- For the BD whose data has been sent, the RD bit in the control field will be cleared by hardware and will not be sent again; after filling in new data, set RD, and this BD can be used for sending again

17.7.3 Receive data frame

- Configure bit fields such as data frame format and interval in the EMAC_MODE register
- By configuring the TXBDNUM field in the EMAC_TX_BD_NUM register to specify the number of BDs used for transmission, the remaining BDs are RX BDs
- The area in the memory that is ready to receive data
- Fill in the address of the data frame into the data pointer field (word 1) corresponding to the received BD
- Clear the status flag in the control and status field (wrod 0) corresponding to the sending BD, and set the control field (interrupt enable, etc.)
- Write the receivable data frame length and set the E bit field to inform EMAC that the BD is idle and can be used for data reception; if necessary, set the IRQ bit to enable interrupts

- In particular, if it is the last valid receiving BD, the WR bit needs to be set, and EMAC will ”wrap around” to the first receiving BD for processing after processing this BD
- If there are multiple BDs available to receive data, repeat the steps of setting BD to fill all BDs
- If you need to enable the receive interrupt, you also need to configure the RX related bits in the EMAC_INT_MASK register
- Configure the RXEN bit in the EMAC_MODE register to enable reception
- If the interrupt is enabled, in the received interrupt, the current BD can be obtained through the RXBDNUM field in the EMAC_TX_BD_NUM register
- Perform corresponding processing according to the current BD status word
- For the received BD, the E bit in the control field will be cleared by hardware and will not be used for receiving again; the data needs to be taken away, and E is set, this BD can be used for receiving again

17.8 Register description

Name	Description
MODE	EMAC configuration
INT_SOURCE	EMAC transmit control
INT_MASK	EMAC interrupt mask
IPGT	Inter packet gap
PACKETLEN	Frame length
COLLCONFIG	Collision configuration
TX_BD_NUM	TX buffer descriptors number
MIIMODE	Management Data configuration
MIICOMMAND	Trigger command
MIIADDRESS	Register address
MIITX_DATA	Control data to be written to PHY
MIIRX_DATA	Received data from PHY
MIISTATUS	MIIM I/F status
MAC_ADDR0	Ethernet MAC address0
MAC_ADDR1	Ethernet MAC address1
HASH0_ADDR	Lower 32-bit of HASH register
HASH1_ADDR	Upper 32-bit of HASH register

Name	Description
TXCTRL	TX control

17.8.1 MODE

Address：0x4000d000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD														RMII EN	RECS MALL
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAD	HUG EN	CRC EN	RSVD		FULL D	RSVD			IFG	PRO	RSVD	BRO	NO PRE	TXEN	RXEN

Bits	Name	Type	Reset	Description
31:18	RSVD			
17	RMIIEN	R/W	1'b0	RMII mode enable 0: MII PHY I/F is used 1: RMII PHY I/F is used
16	RECSMALL	R/W	1'b0	Receive small frame enable 0: Frames smaller than MINFL are ignored. 1: Frames smaller than MINFL are accepted.
15	PAD	R/W	1'b1	Padding enable 0: Do not add pads to frames shorter than MINFL. 1: Add pads to short frames, until the length equals MINFL.
14	HUGEN	R/W	1'b0	Huge frames enable 0: The maximum frame length is MAXFL. All additional bytes are dropped. 1: Frame size is not limited by MAXFL and can be up to 64K bytes.
13	CRCEN	R/W	1'b1	CRC Enable 0: TX MAC does not append CRC field. 1: TX MAC will append CRC field to every frame.
12:11	RSVD			
10	FULLD	R/W	1'b0	Full duplex 0: Half duplex mode. 1: Full duplex mode.
9:7	RSVD			

Bits	Name	Type	Reset	Description
6	IFG	R/W	1'b0	Inter frame gap check 0: IFG is verified before each frame be received. 1: All frames are received regardless to IFG requirement.
5	PRO	R/W	1'b0	Promiscuous mode enable 0: The destination address is checked before receiving. 1: All frames received regardless of the address.
4	RSVD			
3	BRO	R/W	1'b1	Broadcast address enable 0: Reject all frames containing the broadcast address unless the PRO bit is asserted. 1: Receive all frames containing broadcast address.
2	NOPRE	R/W	1'b0	No preamble mode 0: 7-byte preamble will be sent. 1: No preamble will be sent.
1	TXEN	R/W	1'b0	Transmit enable 0: Transmitter is disabled. 1: Transmitter is enabled. If TX_BD_NUM equals 0x0 (zero buffer descriptors are used), then the transmitter is disabled regardless of TXEN.
0	RXEN	R/W	1'b0	Receiver enable 0: Receiver is disabled. 1: Receiver is enabled. If TX_BD_NUM equals 0x80 (all buffer descriptors are used for TX), then the receiver is disabled regardless of RXEN.

17.8.2 INT_SOURCE

Address：0x4000d004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									RXC	TXC	BUSY	RXE	RXB	TXE	TXB

Bits	Name	Type	Reset	Description
31:7	RSVD			

Bits	Name	Type	Reset	Description
6	RXC	R/W	1'b0	Receive control frame This bit indicates that the control frame was received. It is cleared by writing 1 to it. Bit RXFLOW in the CTRLMODE register must be set to 1 in order to get the RXC bit set.
5	TXC	R/W	1'b0	Transmit control frame This bit indicates that a control frame was transmitted. It is cleared by writing 1 to it. Bit TXFLOW in the CTRLMODE register must be set to 1 in order to get the TXC bit set.
4	BUSY	R/W	1'b0	Busy This bit indicates that RX packet is being received and there is no empty buffer descriptor to use. It iscleared by writing 1 to it. This bit appears regardless to the IRQ bits in the Receive Buffer Descriptor.
3	RXE	R/W	1'b0	Receive error This bit indicates that an error occurred while receiving data (overrun, receiver error, dribble nibble, too long, >64K, CRC error, bus error or late collision. It is cleared by writing 1 to it. This bit appears only when IRQ bit is set in the Receive Buffer Descriptor.
2	RXB	R/W	1'b0	Receive frame This bit indicates that a frame was received. It is cleared by writing 1 to it. This bit appears only when IRQ bit is set in the Receive Buffer Descriptor.
1	TXE	R/W	1'b0	Transmit error This bit indicates that a buffer was not transmitted due to a transmit error (underrun, retransmission limit, late collision, bus error or defer time-out). It is cleared by writing 1 to it. This bit appears only when IRQ bit is set in the Transmit Buffer Descriptor.
0	TXB	R/W	1'b0	Transmit buffer This bit indicates that a buffer has been transmitted. It is cleared by writing 1 to it. This bit appears only when IRQ bit is set in the Transmit Buffer Descriptor.

17.8.3 INT_MASK

Address：0x4000d008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									RXCM	TXCM	BM	RXEM	RXBM	TXEM	TXBM

Bits	Name	Type	Reset	Description
31:7	RSVD			
6	RXCM	R/W	1'b1	Receive control frame mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
5	TXCM	R/W	1'b1	Transmit control frame mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
4	BM	R/W	1'b1	Busy mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
3	RXEM	R/W	1'b1	Receive error mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
2	RXBM	R/W	1'b1	Receive frame mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
1	TXEM	R/W	1'b1	Transmit error mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
0	TXBM	R/W	1'b1	Transmit buffer mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked

17.8.4 IPGT

Address：0x4000d00c

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

Bits	Name	Type	Reset	Description
31:7	RSVD			
6:0	IPGT	R/W	7'h18	Inter packet gap The recommended value is 0x18 (24 clock cycles), which equals 9.6 us for 10 Mbps and 0.96 us for 100 Mbps mode

17.8.5 PACKETLEN

Address：0x4000d018

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

Bits	Name	Type	Reset	Description
31:16	MINFL	R/W	16'h40	Minimum frame length The minimum Ethernet packet is 64 bytes long (0x40). To receive small packets, assert the RECSMALL bit or change the MINFL value. To transmit small packets, assert the PAD bit or change the MINFL value.
15:0	MAXFL	R/W	16'h600	Maximum frame length The maximum Ethernet packet is 1518 bytes long. To support this and to have some additional space for tags, a default maximum packet length equals to 1536 bytes (0x600). For bigger packets, you can assert the HUGEN bit or increase the value of MAXFL field.

17.8.6 COLLCONFIG

Address：0x4000d01c

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

Bits	Name	Type	Reset	Description
31:20	RSVD			
19:16	MAXRET	R/W	4'hF	Maximum retry This field specifies the maximum number of consequential retransmission attempts after the collision is detected. When the maximum number has been reached, the TX MAC reports an error and stops transmitting the current packet. According to the Ethernet standard, the MAXRET default value is set to 0xf (15).
15:6	RSVD			
5:0	COLLVAL	R/W	6'h3F	Collision valid This field specifies a collision time window. A collision that occurs later than the time window is reported as a "Late Collisions" and transmission of the current packet is aborted.

17.8.7 TX_BD_NUM

Address：0x4000d020

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

Bits	Name	Type	Reset	Description
31	RSVD			
30:24	RXBDPTR	R	7'h0	RX buffer descriptors (BD) pointer, pointing at the RXBD currently being used
23	RSVD			
22:16	TXBDPTR	R	7'h0	TX buffer descriptors (BD) pointer, pointing at the TXBD currently being used
15:8	RSVD			

Bits	Name	Type	Reset	Description
7:0	TXBDNUM	R/W	8'h40	TX buffer descriptors (BD) number Number of TX BD. TX and RX share 128 (0x80) descriptors, so the number of RX BD equals 0x80 - TXBDNUM. The maximum number of TXBDNUM is 0x80. Values greater then 0x80 cannot be written into this register.

17.8.8 MIIMODE

Address：0x4000d028

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

Bits	Name	Type	Reset	Description
31:9	RSVD			
8	MINOPRE	R/W	1'b0	No preamble for Management Data (MD) 0: 32-bit preamble will be sent. 1: No preamble will be sent.
7:0	CLKDIV	R/W	8'h64	Clock divider for Management Data Clock (MDC) The source clock is bus clock and can be divided by any even number.

17.8.9 MIICOMMAND

Address：0x4000d02c

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

Bits	Name	Type	Reset	Description
31:3	RSVD			

Bits	Name	Type	Reset	Description
2	WCTRDATA	R/W	1'b0	Write control data, setting this bit to 1 will trigger the command (auto cleared) Note: [2]/[1]/[0] cannot be asserted at the same time, execute one command at a time
1	RSTA	R/W	1'b0	Read status, setting this bit to 1 will trigger the command (auto cleared) Note: [2]/[1]/[0] cannot be asserted at the same time, execute one command at a time
0	SCAS	R/W	1'b0	Scan status, setting this bit to 1 will trigger the command (auto cleared) Note: [2]/[1]/[0] cannot be asserted at the same time, execute one command at a time

17.8.10 MIIADDRESS

Address：0x4000d030

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

Bits	Name	Type	Reset	Description
31:13	RSVD			
12:8	RGAD	R/W	5'h0	Register Address
7:5	RSVD			
4:0	FIAD	R/W	5'h0	PHY Address

17.8.11 MIITX_DATA

Address：0x4000d034

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	CTRLDATA	R/W	16'h0	Control Data to be written to PHY

17.8.12 MIIRX_DATA

Address：0x4000d038

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	PRSD	R	16'h0	Received Data from PHY

17.8.13 MIISTATUS

Address：0x4000d03c

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

Bits	Name	Type	Reset	Description
31:2	RSVD			
1	MIIMBUSY	R	1'b0	MIIM I/F busy signal 0: The MIIM I/F is ready. 1: The MIIM I/F is busy.
0	MIIMLF	R	1'b0	MIIM I/F link fail signal

17.8.14 MAC_ADDR0

Address：0x4000d040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACB2								MACB3							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACB4								MACB5							

Bits	Name	Type	Reset	Description
31:24	MACB2	R/W	8'd0	Ethernet MAC address byte 2
23:16	MACB3	R/W	8'd0	Ethernet MAC address byte 3
15:8	MACB4	R/W	8'd0	Ethernet MAC address byte 4
7:0	MACB5	R/W	8'd0	Ethernet MAC address byte 5

17.8.15 MAC_ADDR1

Address：0x4000d044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACB0								MACB1							

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:8	MACB0	R/W	8'd0	Ethernet MAC address byte 0
7:0	MACB1	R/W	8'd0	Ethernet MAC address byte 1

17.8.16 HASH0_ADDR

Address：0x4000d048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HASH0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HASH0															

Bits	Name	Type	Reset	Description
31:0	HASH0	R/W	32'h0	Lower 32-bit of HASH register

17.8.17 HASH1_ADDR

Address：0x4000d04c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HASH1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HASH1															

Bits	Name	Type	Reset	Description
31:0	HASH1	R/W	32'h0	Upper 32-bit of HASH register

17.8.18 TXCTRL

Address：0x4000d050

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

Bits	Name	Type	Reset	Description
31:17	RSVD			
16	TXPRQ	R/W	1'b0	TX Pause Request Writing 1 to this bit starts sending control frame and is automatically cleared to zero.
15:0	TXPTV	R/W	16'h0	TX Pause Timer Value The value that is sent in the pause control frame.

18 USB

18.1 USB introduction

USB (Universal Serial Bus), fully supports USB1.1 full-speed devices, and partially supports USB2.0.

18.2 USB main features

- Support USB full speed device-mode
- Supports 8 bidirectional endpoints: EP0 can be configured as control/bulk/interrupt/synchronization endpoints, EP1-EP7 can be configured as bulk/interrupt/synchronization endpoints
- Each endpoint contains TX and RX FIFOs with a depth of 64 bytes and supports DMA
- Support internal transceiver
- Support suspend/resume
- Support LPM
- Support multiple USB interrupt configurations

18.3 USB function description

18.3.1 USB steps

1. Configure the internal transceiver, the corresponding addresses are 0x40000228 and 0x4000022C
2. Configure usb_config and epx_config of each endpoint
3. Configure USB interrupt related registers
4. Configure USB DMA related (optional)
5. Configure GPIO as USB function (internal transmitter—-function is 10)

6. 0x40000228[20]usb_enum is set to 1, so that the host recognizes that the USB device is inserted and triggers the enumeration process

18.3.2 Part of the register configuration and function description

- swrdy:Read only, only when this bit is 0, can write 1 to cr_usb_ep0_sw_rdy
- crsr:Writing 1 is automatically cleared. When the software allows the next packet to reply with ACK, write 1 to this field, and only the next packet will reply with ACK. For OUT/IN transaction data, it will be received or sent from FIFO (FIFO release once)
- e0snko:It needs to be set to 1, which means that the OUT transaction will reply NAK by default, and the data will not enter the FIFO (FIFO is not released)
- e0snki:It needs to be set to 1, which means that the IN transaction will reply NAK by default, and the data will not be sent from the FIFO (FIFO is not released)
- e0ss:Writing 1 is automatically cleared. Write 1 to this field when the software allows the next packet to reply STALL, then only the next packet will reply STALL
- epxdit:”Token package => trigger xxx_cmd_int => data package => trigger xxx_done_int => handshake package”
- epxcit:”Token package => trigger xxx_cmd_int => data package => trigger xxx_done_int => handshake package”
- ep0odit:”Token package => trigger xxx_cmd_int => data package => trigger xxx_done_int => handshake package”
- ep0ocit:”Token package => trigger xxx_cmd_int => data package => trigger xxx_done_int => handshake package”
- ep0idit:”Token package => trigger xxx_cmd_int => data package => trigger xxx_done_int => handshake package”
- ep0icit:”Token package => trigger xxx_cmd_int => data package => trigger xxx_done_int => handshake package”
- ep0sdit:”Token package => trigger xxx_cmd_int => data package => trigger xxx_done_int => handshake package”
- ep0scit:”Token package => trigger xxx_cmd_int => data package => trigger xxx_done_int => handshake package”
- exrs:Read only. Only when this bit is 0, can write 1 to cr_epx_rdy
- exr:Writing 1 is automatically cleared. When the software allows the next packet to reply ACK, write 1 to this field, then only the next packet will reply ACK (FIFO release once)
- exn:It needs to be set to 1, which means that the transaction will reply NAK by default. IN/OUT depends on the transmission direction configuration of the current endpoint (FIFO is not released)
- exs:Set when the software wants to suspend this endpoint, after setting this endpoint will always reply STALL

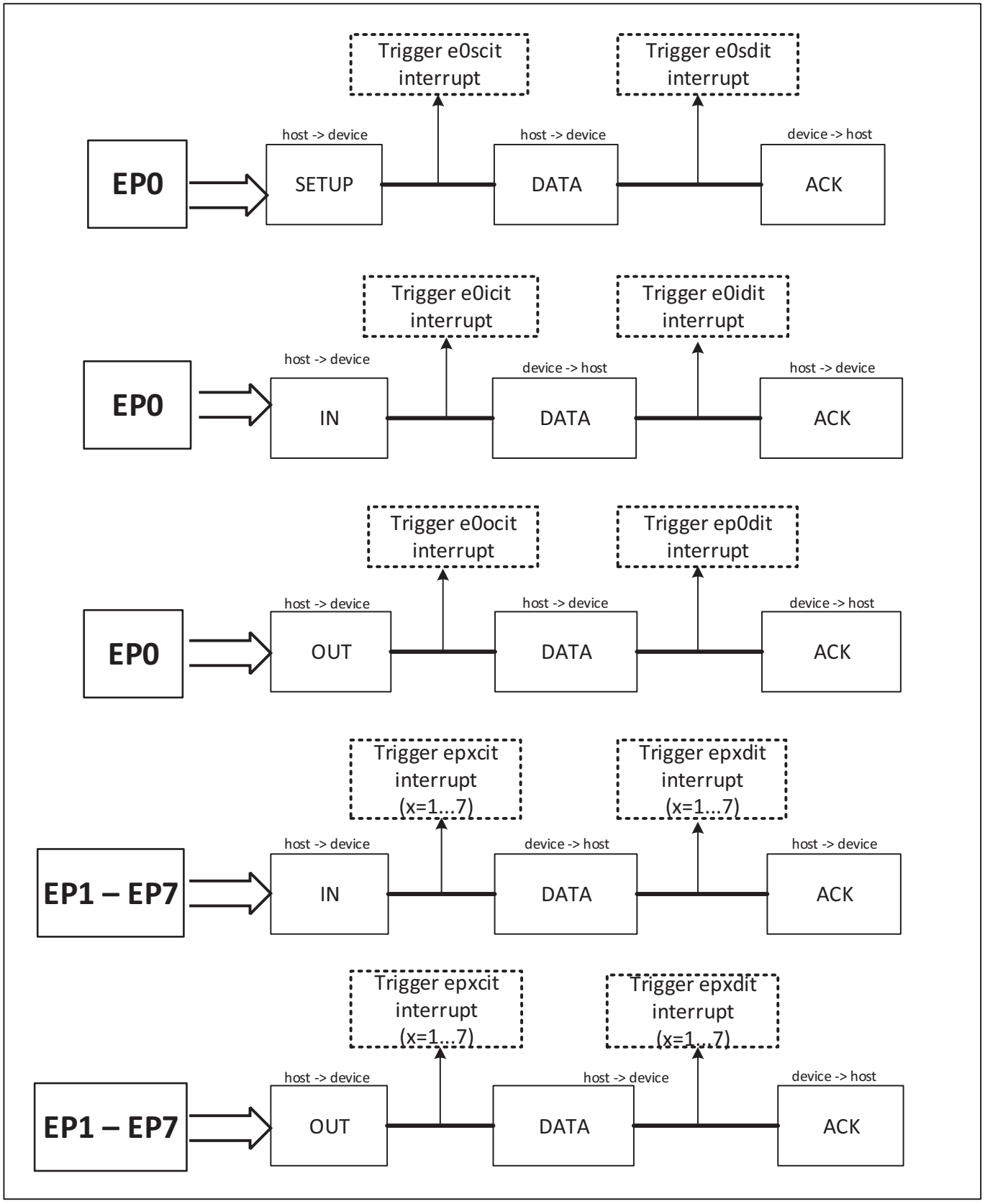

Figure 18.1: USB interrupt trigger mode

18.3.3 USB enumeration phase interrupt processing flow

1. The first is a reset of more than 10ms, which will trigger the reset interrupt.
2. When the reset ends, the reset end interrupt will be triggered.
3. The SETUP transaction, IN transaction, and OUT transaction of the enumeration process will trigger e0sdit, e0icit, and ep0dit respectively.
4. After the enumeration is over, the IN transaction and OUT transaction between the host and the specific endpoint EPx will trigger epxcit and epxdit respectively.

18.3.4 Register operation flow of each transfer transaction

- **Control transmission-SETUP transaction data reception:**
  - Enter interrupt
  - Determine the e0sdit interrupt flag bit
  - What is stored in e0rfr is the data from the setup firm, which can be obtained by reading e0rfr according to e0rfc
  - Set a crsr to release subsequent transactions
  - Clear interrupt flag
  - Exit interrupt
- **Control transmission-IN transaction data transmission:**
  - Enter interrupt
  - Determine the e0idit interrupt flag bit
  - Wait for swrdy to be 0 before writing data to e0tfw
  - Set a crsr to release subsequent transactions
  - Clear interrupt flag
  - Exit interrupt
- **Control transmission-OUT transaction data reception:**
  - Enter interrupt
  - Determine the ep0dit interrupt flag bit
  - What is stored in e0rfr is the data from the OUT firm, which can be obtained by reading e0rfr according to e0rfc
  - Set a crsr to release subsequent transactions

  - Clear interrupt flag
  - Exit interrupt

- **EPx(x=1...7)——IN transaction data transmission:**
  - Enter interrupt
  - Determine the epxcit interrupt flag bit
  - Wait for exrs to be 0 before writing data to extfw (exs needs to be changed to 1 only when only 1 byte is sent)
  - Set an exr to release subsequent transactions
  - Clear interrupt flag
  - Exit interrupt
- **EPx(x=1...7)——OUT transaction data reception:**
  - Enter interrupt
  - Determine the epxdit interrupt flag bit
  - The data stored in exrfr is the data from the OUT firm, which can be obtained by reading exrfr according to exrfc
  - Set an exrs to release subsequent transactions
  - Clear interrupt flag
  - Exit interrupt

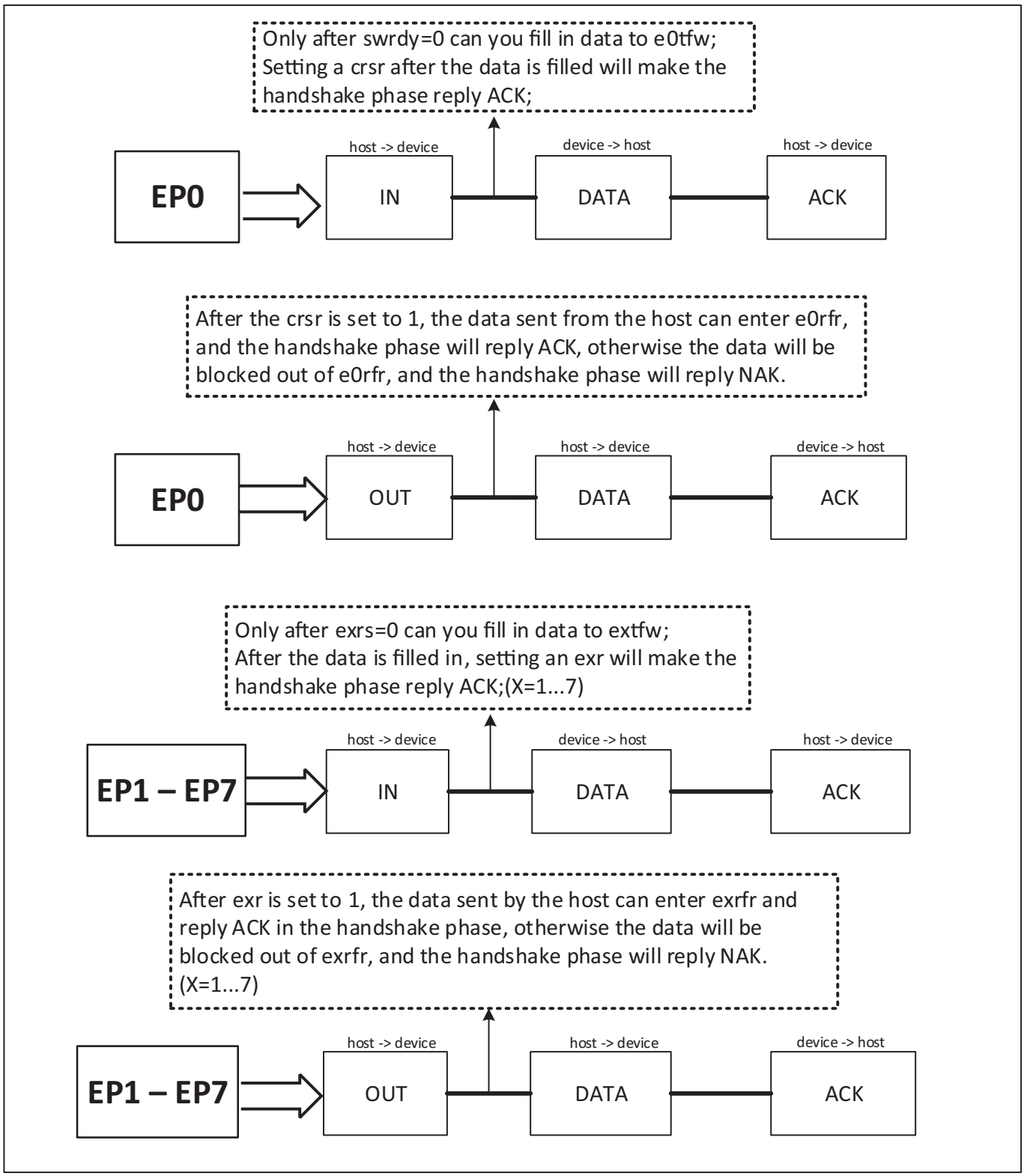

Figure 18.2: USB communication method

Recommended configuration of internal transmitter register:

Table 18.1: Register configuration 1

usb_xcvr	value
usb_rcv	read only
usb_vip	read only
usb_vim	read only
usb_bd	read only
pu_usb	0->1
usb_sus	0
usb_spd	1
usb_enum	0->1
usb_data_convert	0
usb_oeb	read only
usb_oeb_reg	1
usb_oeb_sel	0
usb_rout_pmos	3
usb_rout_nmos	3
pu_usb_ldo	0
usb_ldo_vfb	3

Table 18.2: Register configuration 2

usb_xcvr_config	value
usb_slewrate_p_rise	4
usb_slewrate_p_fall	3
usb_slewrate_m_rise	4
usb_slewrate_m_fall	3
usb_res_pullup_tune	2
reg_usb_use_ctrl	0
usb_str_drv	1
reg_usb_use_xcvr	1
usb_bd_vth	7
usb_v_hys_p	1

Table 18.2: Register configuration 2

usb_xcvr_config	value
usb_v_hys_m	1

Note: You need to set pu_usb and usb_enum to one when you are ready to open the internal transceiver.

18.4 Register description

Name	Description
usb_config	USB configuration
usb_lpm_config	USB lpm configuration
usb_resume_config	USB resume configuration
usb_frame_no	USB frame number
usb_error	USB error
usb_int_en	USB interrupt enable
usb_int_sts	USB interrupt status
usb_int_mask	USB interrupt mask
usb_int_clear	USB interrupt clear
ep1_config	EP1 configuration
ep2_config	EP2 configuration
ep3_config	EP3 configuration
ep4_config	EP4 configuration
ep5_config	EP5 configuration
ep6_config	EP6 configuration
ep7_config	EP7 configuration
ep0_fifo_config	EP0 fifo configuration
ep0_fifo_status	EP0 fifo status
ep0_tx_fifo_wdata	EP0 tx fifo write data
ep0_rx_fifo_rdata	EP0 rx fifo write data
ep1_fifo_config	EP1 fifo configuration

Name	Description
ep1_fifo_status	EP1 fifo status
ep1_tx_fifo_wdata	EP1 tx fifo write data
ep1_rx_fifo_rdata	EP1 rx fifo write data
ep2_fifo_config	EP2 fifo configuration
ep2_fifo_status	EP2 fifo status
ep2_tx_fifo_wdata	EP2 tx fifo write data
ep2_rx_fifo_rdata	EP2 rx fifo write data
ep3_fifo_config	EP3 fifo configuration
ep3_fifo_status	EP3 fifo status
ep3_tx_fifo_wdata	EP3 tx fifo write data
ep3_rx_fifo_rdata	EP3 rx fifo write data
ep4_fifo_config	EP4 fifo configuration
ep4_fifo_status	EP4 fifo status
ep4_tx_fifo_wdata	EP4 tx fifo write data
ep4_rx_fifo_rdata	EP4 rx fifo write data
ep5_fifo_config	EP5 fifo configuration
ep5_fifo_status	EP5 fifo status
ep5_tx_fifo_wdata	EP5 tx fifo write data
ep5_rx_fifo_rdata	EP5 rx fifo write data
ep6_fifo_config	EP6 fifo configuration
ep6_fifo_status	EP6 fifo status
ep6_tx_fifo_wdata	EP6 tx fifo write data
ep6_rx_fifo_rdata	EP6 rx fifo write data
ep7_fifo_config	EP7 fifo configuration
ep7_fifo_status	EP7 fifo status
ep7_tx_fifo_wdata	EP7 tx fifo write data
ep7_rx_fifo_rdata	EP7 rx fifo write data

18.4.1 usb_config

Address：0x4000d800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD			SW RDY	CRSR	E0S NKO	E0S NKI	E0SS	E0SIZE							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOSWADDR							E0SC	RSVD			RBEN	RSVD			USB EN

Bits	Name	Type	Reset	Description
31:29	RSVD			
28	SWRDY	R	1'b0	EP0 transaction ready status bit. Asserted with sw_rdy, and de-asserted when ACK is sent/received.
27	CRSR	W1C	1'b0	EP0 transaction ready. When NACK is enabled, asserting this bit will allow one packet to be transferred even if NACK is asserted Set 1 when a transaction needs to be released once (respond ACK)
26	E0SNKO	R/W	1'b0	EP0 OUT/SETUP transaction nack response (SW control mode) Note: Should NOT enable both ep0_sw_nack_out and ep0_sw_stall at the same time must be 1
25	E0SNKI	R/W	1'b1	EP0 IN transaction nack response (SW control mode) Note: Should NOT enable both ep0_sw_nack_in and ep0_sw_stall at the same time must be 1
24	E0SS	W1C	1'b0	EP0 stall response (SW control mode) Note: Should NOT enable both ep0_sw_nack_in/out and ep0_sw_stall at the same time Set 1 when a handshake needs to be stall once (respond STALL)
23:16	E0SIZE	R/W	8'd0	EP0 transfer size (SW control mode) max packet size is 64
15:9	EOSWADDR	R/W	7'd0	EP0 address (SW control mode) device address

Bits	Name	Type	Reset	Description
8	E0SC	R/W	1'b0	EP0 software control enable 1'b1: EP0 IN/OUT transaction is fully contolled by SW 1'b0: EP0 IN/OUT transaction is controlled by HW must be 1
7:5	RSVD			
4	RBEN	R/W	1'b1	Enable signal of ROM-based descriptors (don't care if ep0_-sw_ctrl is asserted) 1'b1: USB descriptors stored in ROM will be used 1'b0: SW should prepare the descriptors requested by HOST must be 0
3:1	RSVD			
0	USBEN	R/W	1'b0	Enable signal of USB function

18.4.2 usb_lpm_config

Address：0x4000d804

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPM STS	LPMATTR											RSVD			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												LPMRESP		RPUD	LPM EN

Bits	Name	Type	Reset	Description
31	LPMSTS	R	1'b0	LPM status bit
30:20	LPMATTR	R	11'h0	LPM attributes received in LPM packet
19:4	RSVD			
3:2	LPMRESP	R/W	2'd2	Response when LPM packet is received 2'd3: NYET 2'd2: STALL 2'd1: NACK 2'd0: ACK
1	RPUD	W1C	1'b0	Response update signal (for async concern) Assert this bit when cr_lpm_resp is updated
0	LPMEN	W1C	1'b0	LPM enable signal

18.4.3 usb_resume_config

Address：0x4000d808

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

Bits	Name	Type	Reset	Description
31	RSFC	R/W	1'b0	Force to output K-state
30:13	RSVD			
12	RSTG	W1C	1'b0	Resume K-state trigger
11	RSVD			
10:0	RSWD	R/W	11'd26	Resume K-state width (unit: 2.67us)

18.4.4 usb_frame_no

Address：0x4000d818

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

Bits	Name	Type	Reset	Description
31:20	RSVD			
19:16	EPNO	R	4'd0	Endpoint number of the current transaction
15:12	PID	R	4'd0	PID value of the current transaction
11	RSVD			
10:0	FRANM	R	11'h0	Current frame number

18.4.5 usb_error

Address：0x4000d81c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									CRC16 ERR	CRC5 ERR	PCSE	PSEO	IEEO	TTEO	UREO

Bits	Name	Type	Reset	Description
31:7	RSVD			
6	CRC16ERR	R	1'b0	Data CRC error occurs, cleared by cr_usb_err_clr, enabled by ueen
5	CRC5ERR	R	1'b0	Token CRC error occurs, cleared by cr_usb_err_clr, enabled by ueen
4	PCSE	R	1'b0	PID check sum error occurs, cleared by cr_usb_err_clr, enabled by ueen
3	PSEO	R	1'b0	PID sequence error occurs, cleared by cr_usb_err_clr, enabled by ueen
2	IEEO	R	1'b0	Invalid endpoint error occurs, cleared by cr_usb_err_clr, enabled by ueen
1	TTEO	R	1'b0	Transfer time-out error occurs, cleared by cr_usb_err_clr, enabled by ueen
0	UREO	R	1'b0	UTMI I/F RX error occurs, cleared by cr_usb_err_clr, enabled by ueen

18.4.6 usb_int_en

Address：0x4000d820

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UEEN	S3EN	LPEN	LWEN	RDEN	RSVD			EP7D EN	EP7C EN	EP6D EN	EP6C EN	EP5D EN	EP5C EN	EP4D EN	EP4C EN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP3D EN	EP3C EN	EP2D EN	EP2C EN	EP1D EN	EP1C EN	E0OD EN	E0OC EN	E0ID EN	E0IC EN	E0SD EN	E0SC EN	GDC EN	VTEN	UREN	SEN

Bits	Name	Type	Reset	Description
31	UEEN	R/W	1'b1	Interrupt enable of usb_err_int
30	S3EN	R/W	1'b0	Interrupt enable of sof_3ms_int SOF is absent for 3 times (3ms) interrupt enable

Bits	Name	Type	Reset	Description
29	LPEN	R/W	1'b0	Interrupt enable of lpm_pkt_int receive link power mangement packet interrupt enable
28	LWEN	R/W	1'b0	Interrupt enable of lpm_wkup_int receive link power mangement resume (wakeup) packet interrupt enable
27	RDEN	R/W	1'b0	Interrupt enable of usb_rend_int reset end interrupt enable
26:24	RSVD			
23	EP7DEN	R/W	1'b1	Interrupt enable of ep7_done_int
22	EP7CEN	R/W	1'b1	Interrupt enable of ep7_cmd_int
21	EP6DEN	R/W	1'b1	Interrupt enable of ep6_done_int
20	EP6CEN	R/W	1'b1	Interrupt enable of ep6_cmd_int
19	EP5DEN	R/W	1'b1	Interrupt enable of ep5_done_int
18	EP5CEN	R/W	1'b1	Interrupt enable of ep5_cmd_int
17	EP4DEN	R/W	1'b1	Interrupt enable of ep4_done_int
16	EP4CEN	R/W	1'b1	Interrupt enable of ep4_cmd_int
15	EP3DEN	R/W	1'b1	Interrupt enable of ep3_done_int
14	EP3CEN	R/W	1'b1	Interrupt enable of ep3_cmd_int
13	EP2DEN	R/W	1'b1	Interrupt enable of ep2_done_int
12	EP2CEN	R/W	1'b1	Interrupt enable of ep2_cmd_int
11	EP1DEN	R/W	1'b1	Interrupt enable of ep1_done_int
10	EP1CEN	R/W	1'b1	Interrupt enable of ep1_cmd_int
9	E0ODEN	R/W	1'b1	Interrupt enable of ep0_out_done_int
8	E0OCEN	R/W	1'b1	Interrupt enable of ep0_out_cmd_int
7	E0IDEN	R/W	1'b1	Interrupt enable of ep0_in_done_int
6	E0ICEN	R/W	1'b1	Interrupt enable of ep0_in_cmd_int
5	E0SDEN	R/W	1'b1	Interrupt enable of ep0_setup_done_int
4	E0SCEN	R/W	1'b1	Interrupt enable of ep0_setup_cmd_int
3	GDCEN	R/W	1'b1	Interrupt enable of get_dct_cmd_int set 0
2	VTEN	R/W	1'b1	Interrupt enable of vbus_tgl_int vbus voltage toggle interrupt enable, for detect equipment plug

Bits	Name	Type	Reset	Description
1	UREN	R/W	1'b1	Interrupt enable of usb_reset_int reset interrupt enable
0	SEN	R/W	1'b1	Interrupt enable of sof_int SOF interrupt enable

18.4.7 usb_int_sts

Address：0x4000d824

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UEIT	S3IT	LPIT	LWIT	RDIT	RSVD			EP7DIT	EP7CIT	EP6DIT	EP6CIT	EP5DIT	EP5CIT	EP4DIT	EP4CIT
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP3DIT	EP3CIT	EP2DIT	EP2CIT	EP1DIT	EP1CIT	EP0DIT	E0OCIT	E0IDIT	E0ICIT	E0SDIT	E0SCIT	GDCIT	VTIT	URIT	SIT

Bits	Name	Type	Reset	Description
31	UEIT	R	1'b0	USB error occurs, check usb_error for detailed error type
30	S3IT	R	1'b0	SOF is absent for 3 ms
29	LPIT	R	1'b0	LPM packet is received
28	LWIT	R	1'b0	LPM resume (wakeup) signal is received
27	RDIT	R	1'b0	USB reset de-assert is triggered
26:24	RSVD			
23	EP7DIT	R	1'b0	EP7 IN or OUT command is finished
22	EP7CIT	R	1'b0	EP7 IN or OUT command is received
21	EP6DIT	R	1'b0	EP6 IN or OUT command is finished
20	EP6CIT	R	1'b0	EP6 IN or OUT command is received
19	EP5DIT	R	1'b0	EP5 IN or OUT command is finished
18	EP5CIT	R	1'b0	EP5 IN or OUT command is received
17	EP4DIT	R	1'b0	EP4 IN or OUT command is finished
16	EP4CIT	R	1'b0	EP4 IN or OUT command is received
15	EP3DIT	R	1'b0	EP3 IN or OUT command is finished
14	EP3CIT	R	1'b0	EP3 IN or OUT command is received
13	EP2DIT	R	1'b0	EP2 IN or OUT command is finished
12	EP2CIT	R	1'b0	EP2 IN or OUT command is received

Bits	Name	Type	Reset	Description
11	EP1DIT	R	1'b0	EP1 IN or OUT command is finished
10	EP1CIT	R	1'b0	EP1 IN or OUT command is received
9	EP0DIT	R	1'b0	EP0 OUT command is finished
8	E0OCIT	R	1'b0	EP0 OUT command is received
7	E0IDIT	R	1'b0	EP0 IN command is finished
6	E0ICIT	R	1'b0	EP0 IN command is received
5	E0SDIT	R	1'b0	EP0 SETUP command is finished
4	E0SCIT	R	1'b0	EP0 SETUP command is received
3	GDCIT	R	1'b0	GET_DESCRIPTOR command is received
2	VTIT	R	1'b0	VBUS detection is toggled, check 0x1FC[31] for vbus_detect status
1	URIT	R	1'b0	USB reset is triggered
0	SIT	R	1'b0	SOF is received

18.4.8 usb_int_mask

Address：0x4000d828

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UEM	S3M	LPM	LWM	RDM	RSVD			E7DM	E7CM	E6DM	E6CM	E5DM	E5CM	E4DM	E4CM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E3DM	E3CM	E2DM	E2CM	E1DM	E1CM	E0 ODM	E0 OCM	E0 IDM	E0 ICM	E0 SDM	E0 SCM	GDCM	VTM	URM	SM

Bits	Name	Type	Reset	Description
31	UEM	R/W	1'b1	Interrupt mask of usb_err_int
30	S3M	R/W	1'b1	Interrupt mask of sof_3ms_int
29	LPM	R/W	1'b1	Interrupt mask of lpm_pkt_int
28	LWM	R/W	1'b1	Interrupt mask of lpm_wkup_int
27	RDM	R/W	1'b1	Interrupt mask of usb_rend_int
26:24	RSVD			
23	E7DM	R/W	1'b1	Interrupt mask of ep7_done_int
22	E7CM	R/W	1'b1	Interrupt mask of ep7_cmd_int
21	E6DM	R/W	1'b1	Interrupt mask of ep6_done_int

Bits	Name	Type	Reset	Description
20	E6CM	R/W	1'b1	Interrupt mask of ep6_cmd_int
19	E5DM	R/W	1'b1	Interrupt mask of ep5_done_int
18	E5CM	R/W	1'b1	Interrupt mask of ep5_cmd_int
17	E4DM	R/W	1'b1	Interrupt mask of ep4_done_int
16	E4CM	R/W	1'b1	Interrupt mask of ep4_cmd_int
15	E3DM	R/W	1'b1	Interrupt mask of ep3_done_int
14	E3CM	R/W	1'b1	Interrupt mask of ep3_cmd_int
13	E2DM	R/W	1'b1	Interrupt mask of ep2_done_int
12	E2CM	R/W	1'b1	Interrupt mask of ep2_cmd_int
11	E1DM	R/W	1'b1	Interrupt mask of ep1_done_int
10	E1CM	R/W	1'b1	Interrupt mask of ep1_cmd_int
9	E0ODM	R/W	1'b1	Interrupt mask of ep0_out_done_int
8	E0OCM	R/W	1'b1	Interrupt mask of ep0_out_cmd_int
7	E0IDM	R/W	1'b1	Interrupt mask of ep0_in_done_int
6	E0ICM	R/W	1'b1	Interrupt mask of ep0_in_cmd_int
5	E0SDM	R/W	1'b1	Interrupt mask of ep0_setup_done_int
4	E0SCM	R/W	1'b1	Interrupt mask of ep0_setup_cmd_int
3	GDCM	R/W	1'b1	Interrupt mask of get_dct_cmd_int
2	VTM	R/W	1'b1	Interrupt mask of vbus_tgl_int
1	URM	R/W	1'b1	Interrupt mask of usb_reset_int
0	SM	R/W	1'b1	Interrupt mask of sof_int

18.4.9 usb_int_clear

Address：0x4000d82c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UEC	S3C	LPC	LWC	RDC	RSVD			E7DC	E7CC	E6DC	E6CC	E5DC	E5CC	E4DC	E4CC
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E3DC	E3CC	E2DC	E2CC	E1DC	E1CC	E0 ODC	E0 OCC	E0 IDC	E0 ICC	E0 SDC	E0 SCC	GDCC	VTC	URC	E0

Bits	Name	Type	Reset	Description
31	UEC	W1C	1'b0	Interrupt clear of usb_err_int

Bits	Name	Type	Reset	Description
30	S3C	W1C	1'b0	Interrupt clear of sof_3ms_int
29	LPC	W1C	1'b0	Interrupt clear of lpm_pkt_int
28	LWC	W1C	1'b0	Interrupt clear of lpm_wkup_int
27	RDC	W1C	1'b0	Interrupt clear of usb_rend_int
26:24	RSVD			
23	E7DC	W1C	1'b0	Interrupt clear of ep7_done_int
22	E7CC	W1C	1'b0	Interrupt clear of ep7_cmd_int
21	E6DC	W1C	1'b0	Interrupt clear of ep6_done_int
20	E6CC	W1C	1'b0	Interrupt clear of ep6_cmd_int
19	E5DC	W1C	1'b0	Interrupt clear of ep5_done_int
18	E5CC	W1C	1'b0	Interrupt clear of ep5_cmd_int
17	E4DC	W1C	1'b0	Interrupt clear of ep4_done_int
16	E4CC	W1C	1'b0	Interrupt clear of ep4_cmd_int
15	E3DC	W1C	1'b0	Interrupt clear of ep3_done_int
14	E3CC	W1C	1'b0	Interrupt clear of ep3_cmd_int
13	E2DC	W1C	1'b0	Interrupt clear of ep2_done_int
12	E2CC	W1C	1'b0	Interrupt clear of ep2_cmd_int
11	E1DC	W1C	1'b0	Interrupt clear of ep1_done_int
10	E1CC	W1C	1'b0	Interrupt clear of ep1_cmd_int
9	E0ODC	W1C	1'b0	Interrupt clear of ep0_out_done_int
8	E0OCC	W1C	1'b0	Interrupt clear of ep0_out_cmd_int
7	E0IDC	W1C	1'b0	Interrupt clear of ep0_in_done_int
6	E0ICC	W1C	1'b0	Interrupt clear of ep0_in_cmd_int
5	E0SDC	W1C	1'b0	Interrupt clear of ep0_setup_done_int
4	E0SCC	W1C	1'b0	Interrupt clear of ep0_setup_cmd_int
3	GDCC	W1C	1'b0	Interrupt clear of get_dct_cmd_int
2	VTC	W1C	1'b0	Interrupt clear of vbus_tgl_int
1	URC	W1C	1'b0	Interrupt clear of usb_reset_int
0	E0	W1C	1'b0	Interrupt clear of sof_int

18.4.10 ep1_config

Address：0x4000d840

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD												EPRS	ER	ENRE	ESRE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPT			EPD		EPMP										

Bits	Name	Type	Reset	Description
31:20	RSVD			
19	EPRS	R	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and de-asserted when ACK is sent/received.
18	ER	W1C	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	ENRE	R/W	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time must be 1
16	ESRE	R/W	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time Set 1 when endpoint needs to be suspend (respond STALL)
15:13	EPT	R/W	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	EPD	R/W	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	EPMP	R/W	11'd64	Endpoint max packet size

18.4.11 ep2_config

Address：0x4000d844

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD												E2RS	E2R	E2N	E2S
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E2T			E2D		E2S										

Bits	Name	Type	Reset	Description
31:20	RSVD			
19	E2RS	R	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and de-asserted when ACK is sent/received.
18	E2R	W1C	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	E2N	R/W	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time must be 1
16	E2S	R/W	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time Set 1 when endpoint needs to be suspend (respond STALL)
15:13	E2T	R/W	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	E2D	R/W	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	E2S	R/W	11'd64	Endpoint max packet size

18.4.12 ep3_config

Address：0x4000d848

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD												E3RS	E3RS	E3N	E3S
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E3T			E3D		E3S										

Bits	Name	Type	Reset	Description
31:20	RSVD			
19	E3RS	R	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and de-asserted when ACK is sent/received.
18	E3RS	W1C	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	E3N	R/W	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time must be 1
16	E3S	R/W	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time Set 1 when endpoint needs to be suspend (respond STALL)
15:13	E3T	R/W	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	E3D	R/W	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	E3S	R/W	11'd64	Endpoint max packet size

18.4.13 ep4_config

Address：0x4000d84c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD												E4RS	E4R	E4N	E4S
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E4T			E4D		E4S										

Bits	Name	Type	Reset	Description
31:20	RSVD			
19	E4RS	R	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and de-asserted when ACK is sent/received.

Bits	Name	Type	Reset	Description
18	E4R	W1C	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	E4N	R/W	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time must be 1
16	E4S	R/W	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time Set 1 when endpoint needs to be suspend (respond STALL)
15:13	E4T	R/W	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	E4D	R/W	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	E4S	R/W	11'd64	Endpoint max packet size

18.4.14 ep5_config

Address：0x4000d850

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD												E5RS	E5RS	E5N	E5S
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E5T			E5D		E5S										

Bits	Name	Type	Reset	Description
31:20	RSVD			
19	E5RS	R	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and deasserted when ACK is sent/received.
18	E5RS	W1C	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred

Bits	Name	Type	Reset	Description
17	E5N	R/W	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time must be 1
16	E5S	R/W	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time Set 1 when endpoint needs to be suspend (respond STALL)
15:13	E5T	R/W	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	E5D	R/W	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	E5S	R/W	11'd64	Endpoint max packet size

18.4.15 ep6_config

Address：0x4000d854

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD												E6RS	E6R	E6N	E6S
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E6T			E6D		E6S										

Bits	Name	Type	Reset	Description
31:20	RSVD			
19	E6RS	R	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and de-asserted when ACK is sent/received.
18	E6R	W1C	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	E6N	R/W	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time must be 1

Bits	Name	Type	Reset	Description
16	E6S	R/W	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time Set 1 when endpoint needs to be suspend (respond STALL)
15:13	E6T	R/W	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	E6D	R/W	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	E6S	R/W	11'd64	Endpoint max packet size

18.4.16 ep7_config

Address：0x4000d858

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD												E7RS	E7R	E7N	E7S
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E7T			E7D		E7S										

Bits	Name	Type	Reset	Description
31:20	RSVD			
19	E7RS	R	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and de-asserted when ACK is sent/received.
18	E7R	W1C	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	E7N	R/W	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time must be 1
16	E7S	R/W	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time Set 1 when endpoint needs to be suspend (respond STALL)

Bits	Name	Type	Reset	Description
15:13	E7T	R/W	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	E7D	R/W	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	E7S	R/W	11'd64	Endpoint max packet size

18.4.17 ep0_fifo_config

Address：0x4000d900

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E0 RFU	E0 RFO	E0 TFU	E0 TFO	E0 RFC	E0 TFC	E0 DREN	E0 DTEN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	E0RFU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	E0RFO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	E0TFU	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	E0TFO	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	E0RFC	W1C	1'b0	Clear signal of RX FIFO
2	E0TFC	W1C	1'b0	Clear signal of TX FIFO
1	E0DREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface Set 1 when use DMA for rx FIFO data transfer
0	E0DTEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface Set 1 when use DMA for tx FIFO data transfer

18.4.18 ep0_fifo_status

Address：0x4000d904

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E0 RFF	E0 RFE	RSVD							E0RFC						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E0 TFF	E0 TFE	RSVD							E0TFC						

Bits	Name	Type	Reset	Description
31	E0RFF	R	1'b0	RX FIFO full flag
30	E0RFE	R	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	E0RFC	R	7'd0	RX FIFO available count
15	E0TFF	R	1'b0	TX FIFO full flag
14	E0TFE	R	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	E0TFC	R	7'd64	TX FIFO available count

18.4.19 ep0_tx_fifo_wdata

Address：0x4000d908

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E0TFW							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E0TFW	W	x	EP0 TX FIFO

18.4.20 ep0_rx_fifo_rdata

Address：0x4000d90c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E0RFR							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E0RFR	R	8'h0	EP0 RX FIFO

18.4.21 ep1_fifo_config

Address：0x4000d910

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E1 RFU	E1 RFO	E1 TFU	E1 TFO	E1 RFC	E1 TFC	E1 DREN	E1 DTEN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	E1RFU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	E1RFO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	E1TFU	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	E1TFO	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	E1RFC	W1C	1'b0	Clear signal of RX FIFO
2	E1TFC	W1C	1'b0	Clear signal of TX FIFO
1	E1DREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface Set 1 when use DMA for rx FIFO data transfer
0	E1DTEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface Set 1 when use DMA for tx FIFO data transfer

18.4.22 ep1_fifo_status

Address：0x4000d914

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E1 RFF	E1 RFE	RSVD							E1RFC						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E1 TFF	E1 TFE	RSVD							E1TFC						

Bits	Name	Type	Reset	Description
31	E1RFF	R	1'b0	RX FIFO full flag
30	E1RFE	R	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	E1RFC	R	7'd0	RX FIFO available count
15	E1TFF	R	1'b0	TX FIFO full flag
14	E1TFE	R	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	E1TFC	R	7'd64	TX FIFO available count

18.4.23 ep1_tx_fifo_wdata

Address：0x4000d918

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E1TFW							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E1TFW	W	x	EP1 TX FIFO

18.4.24 ep1_rx_fifo_rdata

Address：0x4000d91c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E1RFR							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E1RFR	R	8'h0	EP1 RX FIFO

18.4.25 ep2_fifo_config

Address：0x4000d920

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E2 RFU	E2 RFO	E2 TFU	E2 TFO	E2 RFC	E2 TFC	E2 DREN	E2 DTEN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	E2RFU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	E2RFO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	E2TFU	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	E2TFO	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	E2RFC	W1C	1'b0	Clear signal of RX FIFO
2	E2TFC	W1C	1'b0	Clear signal of TX FIFO
1	E2DREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface Set 1 when use DMA for rx FIFO data transfer
0	E2DTEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface Set 1 when use DMA for tx FIFO data transfer

18.4.26 ep2_fifo_status

Address：0x4000d924

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E2 RFF	E2 RFE	RSVD							E2RFC						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E2 TFF	E2 TFE	RSVD							E2TFC						

Bits	Name	Type	Reset	Description
31	E2RFF	R	1'b0	RX FIFO full flag
30	E2RFE	R	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	E2RFC	R	7'd0	RX FIFO available count
15	E2TFF	R	1'b0	TX FIFO full flag
14	E2TFE	R	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	E2TFC	R	7'd64	TX FIFO available count

18.4.27 ep2_tx_fifo_wdata

Address：0x4000d928

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E2TFW							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E2TFW	W	x	EP2 TX FIFO

18.4.28 ep2_rx_fifo_rdata

Address：0x4000d92c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E2RFR							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E2RFR	R	8'h0	EP2 RX FIFO

18.4.29 ep3_fifo_config

Address：0x4000d930

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E3 RFU	E3 RFO	E3 TFU	E3 TFO	E3 RFC	E3 TFC	E3 DTEN	E3 DTEN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	E3RFU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	E3RFO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	E3TFU	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	E3TFO	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	E3RFC	W1C	1'b0	Clear signal of RX FIFO
2	E3TFC	W1C	1'b0	Clear signal of TX FIFO
1	E3DTEN	R/W	1'b0	Enable signal of dma_rx_req/ack interface Set 1 when use DMA for rx FIFO data transfer
0	E3DTEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface Set 1 when use DMA for tx FIFO data transfer

18.4.30 ep3_fifo_status

Address：0x4000d934

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E3 RFF	E3 RFE	RSVD							E3RFC						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E3 TFF	E3 TFE	RSVD							E3TFC						

Bits	Name	Type	Reset	Description
31	E3RFF	R	1'b0	RX FIFO full flag
30	E3RFE	R	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	E3RFC	R	7'd0	RX FIFO available count
15	E3TFF	R	1'b0	TX FIFO full flag
14	E3TFE	R	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	E3TFC	R	7'd64	TX FIFO available count

18.4.31 ep3_tx_fifo_wdata

Address：0x4000d938

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E3TFW							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E3TFW	W	x	EP3 TX FIFO

18.4.32 ep3_rx_fifo_rdata

Address：0x4000d93c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E3RFR							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E3RFR	R	8'h0	EP3 RX FIFO

18.4.33 ep4_fifo_config

Address：0x4000d940

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E4 RFU	E4 RFO	E4 TFU	E4 TFO	E4 RFC	E4 TFC	E4 DREN	E4 DTEN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	E4RFU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	E4RFO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	E4TFU	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	E4TFO	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	E4RFC	W1C	1'b0	Clear signal of RX FIFO
2	E4TFC	W1C	1'b0	Clear signal of TX FIFO
1	E4DREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface Set 1 when use DMA for rx FIFO data transfer
0	E4DTEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface Set 1 when use DMA for tx FIFO data transfer

18.4.34 ep4_fifo_status

Address：0x4000d944

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E4 RFF	E4 RFE	RSVD							E4RFC						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E4 TFF	E4 TFE	RSVD							E4TFC						

Bits	Name	Type	Reset	Description
31	E4RFF	R	1'b0	RX FIFO full flag
30	E4RFE	R	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	E4RFC	R	7'd0	RX FIFO available count
15	E4TFF	R	1'b0	TX FIFO full flag
14	E4TFE	R	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	E4TFC	R	7'd64	TX FIFO available count

18.4.35 ep4_tx_fifo_wdata

Address：0x4000d948

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E4TFW							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E4TFW	W	x	EP4 TX FIFO

18.4.36 ep4_rx_fifo_rdata

Address：0x4000d94c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E4RFR							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E4RFR	R	8'h0	EP4 RX FIFO

18.4.37 ep5_fifo_config

Address：0x4000d950

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E5 RFU	E5 RFO	E5 TFU	E5 TFO	E5 RFC	E5 TFC	E5 DREN	E5 DTEN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	E5RFU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	E5RFO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	E5TFU	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	E5TFO	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	E5RFC	W1C	1'b0	Clear signal of RX FIFO
2	E5TFC	W1C	1'b0	Clear signal of TX FIFO
1	E5DREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface Set 1 when use DMA for rx FIFO data transfer
0	E5DTEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface Set 1 when use DMA for tx FIFO data transfer

18.4.38 ep5_fifo_status

Address：0x4000d954

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E5 RFF	E5 RFE	RSVD							E5RFC						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E5 TFF	E5 TFE	RSVD							E5TFC						

Bits	Name	Type	Reset	Description
31	E5RFF	R	1'b0	RX FIFO full flag
30	E5RFE	R	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	E5RFC	R	7'd0	RX FIFO available count
15	E5TFF	R	1'b0	TX FIFO full flag
14	E5TFE	R	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	E5TFC	R	7'd64	TX FIFO available count

18.4.39 ep5_tx_fifo_wdata

Address：0x4000d958

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E5TFW							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E5TFW	W	x	EP5 TX FIFO

18.4.40 ep5_rx_fifo_rdata

Address：0x4000d95c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E5RFR							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E5RFR	R	8'h0	EP5 RX FIFO

18.4.41 ep6_fifo_config

Address：0x4000d960

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E6 RFU	E6 RFO	E6 TFU	E6 TFO	E6 RFC	E6 TFC	E6 DREN	E6 DTEN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	E6RFU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	E6RFO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	E6TFU	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	E6TFO	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	E6RFC	W1C	1'b0	Clear signal of RX FIFO
2	E6TFC	W1C	1'b0	Clear signal of TX FIFO
1	E6DREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface Set 1 when use DMA for rx FIFO data transfer
0	E6DTEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface Set 1 when use DMA for tx FIFO data transfer

18.4.42 ep6_fifo_status

Address：0x4000d964

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E6 RFF	E6 RFE	RSVD							E6RFC						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E6 TFF	E6 TFE	RSVD							E6TFC						

Bits	Name	Type	Reset	Description
31	E6RFF	R	1'b0	RX FIFO full flag
30	E6RFE	R	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	E6RFC	R	7'd0	RX FIFO available count
15	E6TFF	R	1'b0	TX FIFO full flag
14	E6TFE	R	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	E6TFC	R	7'd64	TX FIFO available count

18.4.43 ep6_tx_fifo_wdata

Address：0x4000d968

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E6TFW							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E6TFW	W	x	EP6 TX FIFO

18.4.44 ep6_rx_fifo_rdata

Address：0x4000d96c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E6RFR							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E6RFR	R	8'h0	EP6 RX FIFO

18.4.45 ep7_fifo_config

Address：0x4000d970

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E7 RFU	E7 RFO	E7 TFU	E7 TFO	E7 RFC	E7 TFC	E7 DREN	E7 DTEN

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	E7RFU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	E7RFO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	E7TFU	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	E7TFO	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	E7RFC	W1C	1'b0	Clear signal of RX FIFO
2	E7TFC	W1C	1'b0	Clear signal of TX FIFO
1	E7DREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface Set 1 when use DMA for rx FIFO data transfer
0	E7DTEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface Set 1 when use DMA for tx FIFO data transfer

18.4.46 ep7_fifo_status

Address：0x4000d974

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E7 RFF	E7 RFE	RSVD							E7RFC						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E7 TFF	E7 TFE	RSVD							E7TFC						

Bits	Name	Type	Reset	Description
31	E7RFF	R	1'b0	RX FIFO full flag
30	E7RFE	R	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	E7RFC	R	7'd0	RX FIFO available count
15	E7TFF	R	1'b0	TX FIFO full flag
14	E7TFE	R	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	E7TFC	R	7'd64	TX FIFO available count

18.4.47 ep7_tx_fifo_wdata

Address：0x4000d978

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E7TFW							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E7TFW	W	x	EP7 TX FIFO

18.4.48 ep7_rx_fifo_rdata

Address：0x4000d97c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								E7RFR							

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	E7RFR	R	8'h0	EP7 RX FIFO

19 LowPower

19.1 Introduction

Low power consumption is an important indicator for IoT applications. The chip's processor contains three power consumption modes, including working mode, idle power saving mode and sleep mode. You can select the appropriate power consumption mode according to the current application scenario, reduce chip power consumption and extend battery life.

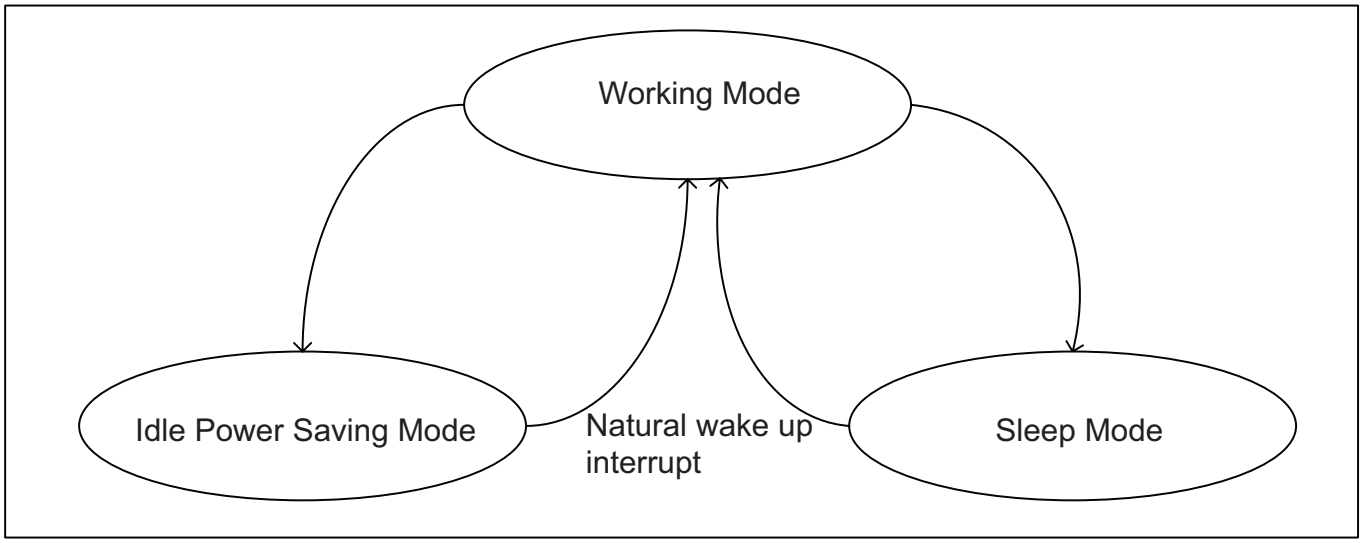

Figure 19.1: Low power mode

19.2 Main features

- Clock control: clock control of each peripheral in GLB, small range of power saving, faster response speed
- Sleep Control (PDS): Contains 9 levels of PDS0/1/2/3/4/5/6/7/31, large-scale power saving, medium response speed
- Sleep Control (HBN): Contains 3 levels of HBN0/1/2, global power saving, long response time

19.3 Function description

19.3.1 Power domain

There are 9 power domains in BL702, and the main functions of each power domain are as follows:

- PD_AON
  - HBN state machine controls power/isolation/reset/clock
  - Keep the internal voltage output selection
  - Pin wakeup control
- PD_AON_HBNRTC
  - Keep RC32K/XTAL32K control register
  - RTC can be used for wake-up or LED blinking
- PD_AON_HBNCORE
  - Part of the power control register
  - 4KB of HBN_RAM, used to save program data before entering PDS/HBN mode, the data will not disappear after entering PDS/HBN
  - PIR digital control, PIR is a pyroelectric infrared sensor, a peripheral in the HBN area, which can be used as a HBN wake-up source
- PD_CORE
  - HBN state machine controls power/isolation/reset/clock
  - 64KB reserved RAM
  - WIFI/BLE timer control
- PD_CORE_MISC_DIG
  - Peripherals (digital part)
- PD_CORE_MISC_ANA
  - Peripherals (analog part)
- PD_CPU
  - CPU/Cache Controller
  - ROM/High-speed RAM
- PD_BZ

  - Bluetooth and ZigBee

- PD_USB
  - USB

Each power domain is controlled by 9 different power modes. The specific control methods are shown in the following table:

Table 19.1: Power mode

NO	Scenario	AON	AON_-HBNRTC	AON_HB-NCORE	CORE	CORE_-MISC_DIG	CORE_-MISC_ANA	CPU	BZ	USB
1	Normal	ON	ON	ON	ON	ON	ON	ON	ON	ON
2	PDS0	ON	ON	ON	ON	ON	ON	ON	ON	ON
3	PDS1	ON	ON	ON	ON	ON	ON	ON	ON	OFF
4	PDS2	ON	ON	ON	ON	ON	ON	ON	OFF	ON
5	PDS3	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
6	PDS4	ON	ON	ON	ON	ON	ON	OFF	ON	ON
7	PDS5	ON	ON	ON	ON	ON	ON	OFF	ON	OFF
8	PDS6	ON	ON	ON	ON	ON	ON	OFF	OFF	ON
9	PDS7	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF
10	PDS31	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF
11	HBN0	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
12	HBN1	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
13	HBN2	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

19.3.2 Wakeup source

The chip supports multiple wake-up sources and can wake up from different power modes.

The CPU will not reset after the PDS0/1/2/3 mode wakes up, and the CPU will reset after the PDS4/5/6/7/31 and HBN0/1/2 modes wake up. After the CPU wakes up and resets, it can be learned from the event register that it is currently reset for what reason.

Each power domain has its own wake-up source, the wake-up source in the sleep hibernation state is the sum of the wake-up sources of the power domain that are still powered in this mode.

PD_AON has the following wake-up sources:

- always on pin wakeup (GPIO10~13)

PD_AON_HBNRTC has the following wake-up sources:

- hbn rtc timer wakeup

PD_AON_HBNCORE has the following wake-up sources:

- pir interrupt, bor interrupt, acomp0 interrupt, acomp1 interrupt

PD_CORE has the following wake-up sources:

- ble sleep timer wakeup、GPIO0~7 Pin Wakeup (select one)、pds timer wakeup

PD_CORE_MISC_DIG and PD_CORE_MISC_ANA have the following wake-up sources:

- kys interrupt、irrx interrupt、GPIO0~8 GPIO Pin Wakeup、GPIO14~31 GPIO Pin Wakeup

PD_USB has the following wake-up sources:

- usb wakeup interrupt

19.3.3 Power mode

Operating mode

The chip provides independent clock control of the processor and peripherals. The clock control of each module is introduced in the chapter of GLB and clock.

The software can control the clocks of the processors or peripherals that do not need to be used according to the current application scenarios. The reaction of the clock control is real-time. In this working mode, there is no need to worry about the response time.

Sleep mode(PDS)

The power-down mode has lower power consumption than the working mode. After entering the PDS mode, control the clock other than RTC (Real Time Clock), switch to the internal low-speed clock, and turn off the external crystal oscillator and PLL to achieve a more power-saving state, so entering and leaving this low-power mode will There is a time delay.

1. Enter sleep mode

The software can make this module enter the power-down mode through PDS configuration and wait for processing. After entering the waiting interrupt mode (WFI), the PDS module will trigger the clock control module to enter the gate clock operation and notify the analog circuit to turn off the PLL and external crystal oscillator.

2. Leave sleep mode

There are two ways to leave sleep mode. The first is that a specific interrupt or event interrupts the idle state during sleep. The second is that the sleep time of PDS_TIM is set by software, and both will trigger the PDS module to enter or leave low power consumption. mode. Note: Because it takes about 1ms to turn on the crystal oscillator, PDS provides the software to turn on the crystal oscillator in advance. This approach can speed up the PDS to wake up.

When the PDS module is ready to wake up, the module will notify the processor to leave the wait for interrupt mode (WFI) through an interrupt.

3. Data retention

When entering the sleep mode, the data in the OCRAM area can automatically enter the retention state and be retained, and can exit the retention state automatically after waking up.

And in sleep mode, 4K HBN_RAM data will always be retained.

Sleep mode(HBN)

In the sleep mode, while retaining the AON (Always On) power supply, most of the chip logic is powered off (Vcore), and the internal circuit will not be awakened until an external event is received.

In the sleep mode, it can achieve the ultimate power saving state, but the response time required compared with the first two is also the longest. It is suitable for a state that does not need to work for a long time, and it can enter this state to extend battery life.

During the sleep period, most of the circuits will be powered off, and the corresponding register values and memory data will also disappear. Therefore, there is 4KB HBN_RAM inside the HBN. This memory will not be powered off when it is in hibernation. The data or state that the software needs to save can be copied to this memory before entering hibernation. When resuming from hibernation, it can be accessed directly from RAM. Data, usually can be used as a state record or quick data recovery (only HBN0 is valid).

19.4 Register description

Name	Description
HBN_CTL	HBN control
HBN_TIME_L	RTC timer compare
HBN_TIME_H	RTC timer compare
RTC_TIME_L	RTC time latched value
RTC_TIME_H	RTC time latched value
HBN_IRQ_MODE	HBN irq mode
HBN_IRQ_STAT	HBN irq state
HBN_IRQ_CLR	HBN irq clear
HBN_PIR_CFG	HBN pir configuration
HBN_PIR_VTH	PIR compare threshold
HBN_PIR_INTERVAL	PIR interval
HBN_MISC	HBN misc configuration

Name	Description
HBN_GLB	HBN glb configuration
HBN_SRAM	HBN ram control

19.4.1 HBN_CTL

Address：0x4000f000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HBNSTA				SRAMSLP	SRSOPT	PWONOPT	RTCDOPT	PODC	V11AONSL				V11RTSL		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V11RTSL	POFFVRT	POFFVCOR	SRST	POFFHRTC	RSVD	POFFHCOR	TRPM	HBNM	RTCCTL						

Bits	Name	Type	Reset	Description
31:28	HBNSTA	R	4'h0	SW polling until 0 (default 4'h3 @pwron)
27	SRAMSLP	R	1'b0	SW polling until 0 (default 1'b1 @pwron)
26	SRSOPT	R/W	0	
25	PWONOPT	R/W	0	
24	RTCDOPT	R/W	0	
23	PODC	R/W	1	Power On DCDC18 during Power On Sequence (require 400 800us)
22:19	V11AONSL	R/W	4'hA	VDD11_AON Voltage Out Select @ Enter hibernate
18:15	V11RTSL	R/W	4'hA	VDD11_RT Voltage Out Select @ Enter hibernate
14	POFFVRT	R/W	0	Set 1 to disable power off VDDCORE_RT at HBN mode (for low power)
13	POFFVCOR	R/W	0	Set 1 to disable power off VDDCORE at HBN mode (for debug)
12	SRST	R/W	0	soft reset
11	POFFHRTC	R/W	0	Power Off HBN RTC @ Enter hibernate
10	RSVD			
9	POFFHCOR	R/W	0	Power Off HBN Core @ Enter hibernate
8	TRPM	R	0	Boot Strap 0: Flash, 1: UART or SDIO
7	HBNM	W	0	Enter hibernate

Bits	Name	Type	Reset	Description
6:0	RTCCTL	R/W	7'h0	[6:4] Slow LED, x/0.25/0.5/1/2/4/8/16 seconds [3] rtc long time 0 353days (bit 39 13 compare) [2] rtc short time 0 488s (bit 23 0 compare) [1] rtc time 0 353days (bit 39 0 compare) [0] rtc enable

19.4.2 HBN_TIME_L

Address：0x4000f004

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

Bits	Name	Type	Reset	Description
31:0	HBNTIML	R/W	32'h0	RTC timer compare bit 31:0

19.4.3 HBN_TIME_H

Address：0x4000f008

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7:0	HBNTIMH	R/W	8'h0	RTC timer compare bit 39:32

19.4.4 RTC_TIME_L

Address：0x4000f00c

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

Bits	Name	Type	Reset	Description
31:0	RTCTIMLL	R	32'h0	RTC time latched value bit 31:0

19.4.5 RTC_TIME_H

Address：0x4000f010

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

Bits	Name	Type	Reset	Description
31	RTCTIML	W	0	RTC time latch for SW read
30:8	RSVD			
7:0	RTCTIMLH	R	8'h0	RTC time latched value bit 39:32

19.4.6 HBN_IRQ_MODE

Address：0x4000f014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD				PWUEN	PWUSEL			A1EN		A0EN		RSVD	BOREN	RSVD	HWPUPD
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			AONPADIE					HPAM					HPWMODE		

Bits	Name	Type	Reset	Description
31:28	RSVD			
27	PWUEN	R/W	0	Pin wakeup delay enable
26:24	PWUSEL	R/W	3'd3	Pin wakeup delay 1 7 sec

Bits	Name	Type	Reset	Description
23:22	A1EN	R/W	0	enable acomp1 interrupt [20] posedge [21] negedge
21:20	A0EN	R/W	0	enable acomp0 interrupt [20] posedge [21] negedge
19	RSVD			
18	BOREN	R/W	0	enable brown-out interrupt
17	RSVD			
16	HWPUPD	R/W	1	1: Pull GPIO9 and GPIO17 @ pwr_on and pwr_rst 0 : no pull
15:13	RSVD			
12:8	AONPADIE	R/W	5'b11111	AON_PAD IE/SMT (GPIO13 GPIO9)
7:3	HPAM	R/W	5'b0	mask hbn_pin_wakeup_event
2:0	HPWMODE	R/W	3'b101	hbn_pin_wakeup mode 000 : sync falling edge trigger 001 : sync rising edge trigger 010 : sync low level trigger 011 : sync high level trigger 100 : async falling edge trigger 101 : async rising edge trigger 110 : async low level trigger 111 : async high level trigger

19.4.7 HBN_IRQ_STAT

Address：0x4000f018

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

Bits	Name	Type	Reset	Description
31:0	IRQSTATE	R	0	[22] acomp1 [20] acomp0 [18] brown-out [17] irq_pir state [16] irq_rtc state [4:0] hbn_pin_wakeup state (GPIO13 GPIO9)

19.4.8 HBN_IRQ_CLR

Address：0x4000f01c

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

Bits	Name	Type	Reset	Description
31:0	IRQCLR	W	0	[22] irq_acomp1 clear [20] irq_acomp0 clear [18] irq_bor clear [17] irq_pir clear [16] irq_rtc clear [4:0] hbn_pin_wakeup clear (GPIO13 GPIO9)

19.4.9 HBN_PIR_CFG

Address：0x4000f020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						NO SYNC	CGEN	PIR EN	RSVD	PIRDIS		RSVD	LPF SEL	HPFSEL	

Bits	Name	Type	Reset	Description
31:10	RSVD			
9	NOSYNC	R/W	0	gpadc no sync
8	CGEN	R/W	0	gpadc force cgen=1
7	PIREN	R/W	0	pir enable
6	RSVD			
5:4	PIRDIS	R/W	0	pir disable [4] low -> high won't trigger interrupt [5] high -> low won't trigger interrupt
3	RSVD			
2	LPFSEL	R/W	0	0: /1. 1:/2

Bits	Name	Type	Reset	Description
1:0	HPFSEL	R/W	0	0: 1-z^-1, 1: 1-z^-2, 0: 2-z^-3

19.4.10 HBN_PIR_VTH

Address：0x4000f024

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

Bits	Name	Type	Reset	Description
31:14	RSVD			
13:0	PIRVYH	R/W	14'h3ff	PIR compare threshold

19.4.11 HBN_PIR_INTERVAL

Address：0x4000f028

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

Bits	Name	Type	Reset	Description
31:12	RSVD			
11:0	PIRINTERV	R/W	12'd2621	pir_lpf_sel = 0: 32768 / (pir_interval+1) Hz, default 12.5Hz (80ms) pir_lpf_sel = 1: 32768 / (pir_interval*2+1) Hz, default 6.25Hz (160ms)

19.4.12 HBN_MISC

Address：0x4000f02c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		PULLDWAON						RSVD		PULLUPAON					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												BOR OUT	PU BOR	BOR VTH	BOR SEL

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:24	PULLDWAON	R/W	6'h0	[5] : 1 : Turn on Flash PAD PD (GPIO28), Only Set this bit when Flash @ Deep Power Down Mode [4] : 1 : Turn on Flash PAD PD (GPIO27), Only Set this bit when Flash @ Deep Power Down Mode [3] : 1 : Turn on Flash PAD PD (GPIO26), Only Set this bit when Flash @ Deep Power Down Mode [2] : 1 : Turn on Flash PAD PD (GPIO25), Only Set this bit when Flash @ Deep Power Down Mode [1] : 1 : Turn on Flash PAD PD (GPIO24), Only Set this bit when Flash @ Deep Power Down Mode [0] : 1 : Turn on Flash PAD PD (GPIO23), Only Set this bit when Flash @ Deep Power Down Mode
23:22	RSVD			
21:16	PULLUPAON	R/W	6'h0	[5] : 1 : Turn on Flash PAD PU (GPIO28), Only Set this bit when Flash @ Deep Power Down Mode [4] : 1 : Turn on Flash PAD PU (GPIO27), Only Set this bit when Flash @ Deep Power Down Mode [3] : 1 : Turn on Flash PAD PU (GPIO26), Only Set this bit when Flash @ Deep Power Down Mode [2] : 1 : Turn on Flash PAD PU (GPIO25), Only Set this bit when Flash @ Deep Power Down Mode [1] : 1 : Turn on Flash PAD PU (GPIO24), Only Set this bit when Flash @ Deep Power Down Mode [0] : 1 : Turn on Flash PAD PU (GPIO23), Only Set this bit when Flash @ Deep Power Down Mode
15:4	RSVD			
3	BOROUT	R	1'h0	Brown Out Reset status
2	PUBOR	R/W	1'h0	Power up Brown Out Reset
1	BORVTH	R/W	1'h1	BOR threshold 0:2.0V, 1: 2.4V
0	BORSEL	R/W	1'h0	0: POR is independent of BOR, 1: POR is relevant to BOR

19.4.13 HBN_GLB

Address：0x4000f030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AIDOVEC				RTIDOVEC				RSVD				V11SOVSEL			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		CRSE	HBNRSEV							PU RC32	F32SEL		UART CLK	ROOTCLK	

Bits	Name	Type	Reset	Description
31:28	AIDOVEC	R/W	4'ha	aon ldo output voltage external control: 0:0.60V, 1:0.65V, 2:0.70V, 3:0.75V, 4:0.80V, 5:0.85V, 6:0.9V, 7:0.95V 8:1.0V, 9:1.05V, 10:1.1V, 11:1.15V, 12:1.2V, 13:1.25V, 14:1.3V, 15:1.35V
27:24	RTIDOVEC	R/W	4'ha	ldo output voltage external control: 0:0.60V, 1:0.65V, 2:0.70V, 3:0.75V, 4:0.80V, 5:0.85V, 6:0.9V, 7:0.95V 8:1.0V, 9:1.05V, 10:1.1V, 11:1.15V, 12:1.2V, 13:1.25V, 14:1.3V, 15:1.35V
23:20	RSVD			
19:16	V11SOVSEL	R/W	4'hA	vdd11soc output voltage selection
15:14	RSVD			
13	CRSE	R/W	1'b0	clear reset event
12:8	HBNRSEV	R	4'b0	[4] : bor_out_ event [3] : pwr_rst_n event [2] : sw_rst event [1] : ext_rst_n event [0] : por_out event
7:6		R/W	2'h1	ldo11_rt drive strength select
5	PURC32	R/W	1	0: Turn off rc32k during rtc power domain off, 1: Don't turn off rc32k
4:3	F32SEL	R/W	0	32KHz clock source selection (0: RC32K 1: XTAL 32K 3: DIG 32K)
2	UARTCLK	R/W	0	uart clock selection from HBN (0:fclk 1:96MHz)
1:0	ROOTCLK	R/W	0	root clock source selection (0: RC32M 1: XTAL 2/3: PLL)

19.4.14 HBN_SRAM

Address：0x4000f034

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	SLP	R/W	0	make HBN RAM Sleep
6	RET	R/W	0	make HBN RAM Retention
5:0	RSVD			

19.5 Register description

Name	Description
PDS_CTL	PDS control register
PDS_TIME1	PDS sleep time
PDS_INT	PDS interrupt
PDS_CTL2	PDS control register 2
PDS_CTL3	PDS control register 3
PDS_CTL4	PDS control register 4
pds_stat	PDS state
pds_ram1	PDS ram control
pds_gpio_set_pu_pd	PDS gpio set
pds_gpio_int	PDS gpio interrupt

19.5.1 PDS_CTL

Address：0x4000e000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTRLPLL		CTRLRF		IDOVOL				FSR CLK	PD ID11	NPW MASK	RSVD	RLPW CLK	IDO VSEL	RC32 PC	RSEN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEFO	PDXT	PWOF	WXRD	ISL EN	PUFS	MEM STBY	GATE CLK	CPFS	CIO	PD BGS	PDDC	SWF STA	XTFO	SLFE	PDSS

Bits	Name	Type	Reset	Description
31:30	CTRLPLL	R/W	2'b00	00：PDS don't Control PLL on/off 01：PDS Control PLL on/off depend on gate_clock signal 10：PDS Control PLL on/off depend on MISC isolation off/on 11：PDS Control PLL on/off when pds at idle state Note. When PDS Control PLL, the real PU signal will still wait until "xtal_rdy"
29:28	CTRLRF	R/W	2'b01	00：PDS don't control RF on/off 01：PDS control RF on/off depend on soc_en 10：PDS control RF on/off depend on BLE power on/off 11：PDS control RF on/off whe pds at idle state
27:24	IDOVOL	R/W	4'hA	LDO voltage value in PDS mode
23	FSRCLK	R/W	0	0：Don't Force SRAM CLK Enable 1：Force SRAM CLK Enable
22	PDID11	R/W	0	0：power on ldo11 during PDS 1：power down ldo11 during PDS
21	NPWMASK	R/W	0	pds start condition mask np_wfi
20	RSVD			
19	RLPWCLK	R/W	0	0：Control SRAM Low Power without CLK (Async) 1：Control SRAM Low Power with CLK (Sync)
18	IDOVSEL	R/W	0	PDS "SLEEP" control LDO voltage enable
17	RC32PC	R/W	0	1：RC32M always on @any state 0：RC32M on/off controlled by PDS state
16	RSEN	R/W	0	0：no pds_reset 1: pds_rst controlled by PDS
15	SEFO	R/W	0	0 :pds_soc_enb controlled by PDS 1 :pds_soc_enb always active
14	PDXT	R/W	1	0：don't_touch xtal during PDS 1：xtal power down during PDS

Bits	Name	Type	Reset	Description
13	PWOF	R/W	1	0：don’t_touch Power during PDS 1：Power off during PDS (each power domain can has its own control)
12	WXRD	R/W	0	0：Skip wait XTAL Ready before PDS Interrupt 1：wait XTAL Ready during before PDS Interrupt
11	ISLEN	R/W	1	0：don’t_touch Isolation during PDS (all power domain) 1：Isolation during PDS (each power domain can has its own control)
10	PUFS	R/W	1	if cr_pds_ctrl_pu_flash =0 or not in PDS31 state 0：SW Turn Off Flash 1: SW Turn on Flash
9	MEMSTBY	R/W	1	0：don’t_touch mem_stby during PDS 1：mem_stby during PDS (each power domain can has its own control)
8	GATECLK	R/W	1	0：don’t_touch clock gating during PDS (all power domain) 1：gate clock during PDS (each pwr domain has its own control)
7	CPFS	R/W	0	0：don’’t touch Flash power during all PDS (Flash Power Always on) 1：turn off Flash Power During PDS (same control with pds power off control)
6	CIO	R/W	0	1: allow PDS Control the GPIO IE/PU/PD at Sleep Mode
5	PDBGS	R/W	0	0：don’t_touch bg_sys during PDS 1：power down bg_sys during PDS
4	PDDC	R/W	0	0：don’t_touch dcdc18 during PDS 1：power down dcdc18 during PDS
3	SWFSTA	R/W	0	Save WIFI State Before Enter PDS
2	XTFO	R/W	0	0：don’t_touch xtal during PDS 1：power down xtal during PDS
1	SLFE	R/W	0	0:can be awakened by pds_timer 1:can’t be awakened by pds_timer
0	PDSS	W1P	0	Enter PDS

19.5.2 PDS_TIME1

Address：0x4000e004

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

Bits	Name	Type	Reset	Description
31:0	SLTIM	R/W	32'd3240	PDS Sleep Time (in units of 32K clock cycles)

19.5.3 PDS_INT

Address：0x4000e00c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUEV								WUSEN							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INCL	RSVD			PDMS	RDMS	RSVD	WUMS	CRSE	RSEV			PDIT	RDIT	RSVD	WKIT

Bits	Name	Type	Reset	Description
31:24	WUEV	R	0	Record Wakeup resason, and be clear by cr_pds_int_clr [31]: wakeup trigger by kys_int [30]: wakeup trigger by usb_wkup [29]: wakeup trigger by ble_slp_irq [28]: wakeup trigger by irrx_int [27]: wakeup trigger by gpio_irq [26]: wakeup trigger by hbn_irq_out[1] [25]: wakeup trigger by hbn_irq_out[0] [24]: wakeup trigger by pds_sleep_cnt=0
23:16	WUSEN	R/W	8'hFF	Mask PDS Wakeup Trigger [23]: enable wakeup trigger by kys_int [22]: enable wakeup trigger by usb_wkup [21]: enable wakeup trigger by ble_slp_irq [20]: enable wakeup trigger by irrx_int [19]: enable wakeup trigger by gpio_irq [17]: enable wakeup trigger by hbn_irq_out[1] [17]: enable wakeup trigger by hbn_irq_out[0] [16]: enable wakeup trigger by pds_sleep_cnt=0
15	INCL	R/W	0	pds interrupt & wakeup reason clear
14:12	RSVD			
11	PDMS	R/W	0	Mask pds pll done interrupt

Bits	Name	Type	Reset	Description
10	RDMS	R/W	0	Mask pds rf done interrupt
9	RSVD			
8	WUMS	R/W	0	Mask pds wakeup interrupt
7	CRSE	W	0	clear pds reset event
6:4	RSEV	R	0	[2] : pds_rst_n (pds reset) [1]: pwr_rst_n (hbn power on reset) [0]: hreset_n (Bus Reset)
3	PDIT	R	0	pu_pll_done interrupt
2	RDIT	R	0	pu_rf_done interrupt
1	RSVD			
0	WKIT	R	0	PDS Wakeup Interrupt

19.5.4 PDS_CTL2

Address：0x4000e010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD												FUSB GCLK	FBZ GCLK	RSVD	FNP GCLK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FUSB MSLP	FBZ MSLP	RSVD	FNP MSLP	FUSB RST	FBZP RST	RSVD	FNP PRST	FUSB ISEN	FBZ ISEN	RSVD	FNP ISEN	FUSB PWOF	FBZ PWOF	RSVD	FNP PWOF

Bits	Name	Type	Reset	Description
31:20	RSVD			
19	FUSBGCLK	R/W	0	manual force USB clock gated
18	FBZGCLK	R/W	0	manual force BZ clock gated
17	RSVD			
16	FNPGCLK	R/W	0	manual force NP clock gated
15	FUSBMSLP	R/W	0	manual force USB memory sleep
14	FBZMSLP	R/W	0	manual force BZ memory sleep
13	RSVD			
12	FNPMSLP	R/W	0	manual force NP memory sleep
11	FUSBRST	R/W	0	manual force USB pds reset
10	FBZPRST	R/W	0	manual force BZ pds reset

Bits	Name	Type	Reset	Description
9	RSVD			
8	FNPPRST	R/W	0	manual force NP pds reset
7	FUSBISEN	R/W	0	manual force USB isolation
6	FBZISEN	R/W	0	manual force BZ isolation
5	RSVD			
4	FNPISEN	R/W	0	manual force NP isolation
3	FUSBPWOF	R/W	0	manual force USB power off
2	FBZPWOF	R/W	0	manual force BZ power off
1	RSVD			
0	FNPPWOF	R/W	0	manual force NP power off

19.5.5 PDS_CTL3

Address：0x4000e014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD	MSIS	USB ISO	BLE ISO	BZ ISO	RSVD		NP ISO	RSVD							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	FBLE GCLK	FMIS GCLK	RSVD	FBLE MSTB	FMIS MSTB	RSVD	FBLE PRST	FMIS PRST	RSVD	FBLE ISO	RSVD		FBLE PWOF	FMIS PWOF	RSVD

Bits	Name	Type	Reset	Description
31	RSVD			
30	MSIS	R/W	1	1：make misc isolated at PDS Sleep state 0：make misc isolated at PDS Sleep state
29	USBISO	R/W	1	1：make usb isolated at PDS Sleep state 0：make usb isolated at PDS Sleep state
28	BLEISO	R/W	1	1：make bz_Ble isolated at PDS Sleep state 0：make bz_Ble isolated at PDS Sleep state
27	BZISO	R/W	1	1：make BZ isolated at PDS Sleep state 0：make BZ isolated at PDS Sleep state
26:25	RSVD			
24	NPISO	R/W	1	1：make NP isolated at PDS Sleep state 0：make NP isolated at PDS Sleep state
23:15	RSVD			

Bits	Name	Type	Reset	Description
14	FBLEGCLK	R/W	0	manual force BZ_BLE gate_clk
13	FMISGCLK	R/W	0	manual force MISC gate_clk
12	RSVD			
11	FBLEMSTB	R/W	0	manual force BZ_BLE mem_stby
10	FMISMSTB	R/W	0	manual force MISC mem_stby
9	RSVD			
8	FBLEPRST	R/W	0	manual force BZ_BLE pds_rst
7	FMISPRST	R/W	0	manual force MISC pds_rst
6	RSVD			
5	FBLEISO	R/W	0	manual force BZ_BLE iso_en
4:3	RSVD			
2	FBLEPWOF	R/W	0	manual force BZ_BLE pwr_off
1	FMISPWOF	R/W	0	manual force MISC pwr_off
0	RSVD			

19.5.6 PDS_CTL4

Address：0x4000e018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MISC DPOF	MISC APOF	RSVD		MISC GCLK	MISC MSTB	MISC RST	MISC PWOF	USB GCLK	USB MSTB	USB RST	USB PWOF	BLE GCLK	BLE MSTB	BLE RST	BLE PWOF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BZ GCLK	BZ MSTB	BZ RST	BZ PWOF	RSVD								NP GCLK	NP MSTB	NP RST	NP PWOF

Bits	Name	Type	Reset	Description
31	MISCDPOF	R/W	1	Use with bit[27] cr_pds_misc_pwr_off , if cr_pds_misc_-pwr_off =1, this bit : 1: Power Off PD_CORE_MISC_DIG when pds_state run at Power OFF 0 :Power On PD_CORE_MISC_DIG when pds_state run at Power OFF

Bits	Name	Type	Reset	Description
30	MISCAPOF	R/W	1	Use with bit[27] cr_pds_misc_pwr_off , if cr_pds_misc_-pwr_off =1, this bit : 1: Power Off PD_CORE_MISC_ANA when pds_state run at Power OFF 0 :Power On PD_CORE_MISC_ANA when pds_state run at Power OFF
29:28	RSVD			
27	MISCGCLK	R/W	1	1 : make core_misc clock gated at PDS Sleep state 0 : make core_misc clocking at PDS Sleep state
26	MISCMSTB	R/W	1	1 : make core_misc RAM @Retention at PDS Sleep state 0 : make core_misc RAM @ Normal at PDS Sleep state
25	MISCRST	R/W	1	1 : make core_misc reset at PDS Sleep state 0 : make core_misc not reset at PDS Sleep state
24	MISCPWOF	R/W	1	1 : make core_misc Power off at PDS Sleep state 0 : make core_misc power on at PDS Sleep state
23	USBGCLK	R/W	1	1 : make usb clock gated at PDS Sleep state 0 : make usb clocking at PDS Sleep state
22	USBMSTB	R/W	1	1 : make usb RAM @Retention at PDS Sleep state 0 : make usb RAM @ Normal at PDS Sleep state
21	USBRST	R/W	1	1 : make usb reset at PDS Sleep state 0 : make usb not reset at PDS Sleep state
20	USBPWOF	R/W	1	1 : make usb Power off at PDS Sleep state 0 : make usb power on at PDS Sleep state
19	BLEGCLK	R/W	1	1 : make bz_ble clock gated at PDS Sleep state 0 : make bz_ble clocking at PDS Sleep state
18	BLEMSTB	R/W	1	1 : make bz_ble RAM @Retention at PDS Sleep state 0 : make bz_ble RAM @ Normal at PDS Sleep state
17	BLERST	R/W	1	1 : make bz_ble reset at PDS Sleep state 0 : make bz_ble not reset at PDS Sleep state
16	BLEPWOF	R/W	1	1 : make bz_ble Power off at PDS Sleep state 0 : make bz_ble power on at PDS Sleep state
15	BZGCLK	R/W	1	1 : make BZ clock gated at PDS Sleep state 0 : make BZ clocking at PDS Sleep state
14	BZMSTB	R/W	1	1 : make BZ RAM @Retention at PDS Sleep state 0 : make BZ RAM @ Normal at PDS Sleep state
13	BZRST	R/W	1	1 : make BZ reset at PDS Sleep state 0 : make BZ not reset at PDS Sleep state

Bits	Name	Type	Reset	Description
12	BZPWOF	R/W	1	1：make BZ Power off at PDS Sleep state 0：make BZ power on at PDS Sleep state
11:4	RSVD			
3	NPGCLK	R/W	1	1：make NP clock gated at PDS Sleep state 0：make NP clocking at PDS Sleep state
2	NPMSTB	R/W	1	1：make NP RAM @Retention at PDS Sleep state 0：make NP RAM @ Normal at PDS Sleep state
1	NPRST	R/W	1	1：make NP reset at PDS Sleep state 0：make NP not reset at PDS Sleep state
0	NPPWOF	R/W	1	1：make NP Power off at PDS Sleep state 0：make NP power on at PDS Sleep state

19.5.7 pds_stat

Address：0x4000e01c

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

Bits	Name	Type	Reset	Description
31:18	RSVD			
17:16	PLLSTA	R	2'b00	ST_PDS_PLL_OFF = 2'b00 ; ST_PDS_PLL_SFREG = 2'b01 ; ST_PDS_PLL_PU = 2'b10 ; ST_PDS_PLL_RDY = 2'b11 ;
15:12	RSVD			
11:8	RFSTA	R	4'b0	ST_PDS_RF_OFF = 4'b0000 ; ST_PDS_PU_MBG = 4'b0001 ; ST_PDS_PU_LDO15RF = 4'b0011 ; ST_PDS_PU_SFREG = 4'b0111 ; ST_PDS_BZ_EN_AON = 4'b1111 ;
7:4	RSVD			

Bits	Name	Type	Reset	Description
3:0	STA	R	4'b0	ST_IDLE = 4'b0000; ST_ECG = 4'b1000; ST_ERST = 4'b1100; ST_EISO = 4'b1111; ST_POFF = 4'b0111; ST_PRE_BGON = 4'b0011; ST_PRE_BGON1 = 4'b0001; ST_BGON = 4'b0101; ST_CLK_SW_32M = 4'b0100; ST_PON_DCDC = 4'b0110; ST_PON_LDO11_MISC = 4'b1110; ST_PON = 4'b1010; ST_DISO = 4'b0010; ST_DCG = 4'b1101; ST_DRST = 4'b1011; ST_WAIT_EFUSE = 4'b1001;

19.5.8 pds_ram1

Address：0x4000e020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				RAMPGEN				RAMRET2N				RAMRET1N			

Bits	Name	Type	Reset	Description
31:12	RSVD			
11:8	RAMPGEN	R/W	4'h0	[3] : 48 64KB np_ram Low Power Enable [2] : 32 48KB np_ram Low Power Enable [1] : 16 32KB np_ram Low Power Enable [0] : 0 16KB np_ram Low Power Enable
7:4	RAMRET2N	R/W	4'h0	[3] : 48 64KB np_ram RET2N, 0: Active With cr_pds_ram_-pgen[3]=1 and cr_pds_ram_ret1n[3]=1 [2] : 32 48KB np_ram RET 2N, 0: Active With cr_pds_ram_-pgen[2]=1 and cr_pds_ram_ret1n[2]=1 [1] : 16 32KB np_ram RET 2N,0: Active With cr_pds_ram_-pgen[1]=1 and cr_pds_ram_ret1n[1]=1 [0] : 0 16KB np_ram RET_2N, 0: Active With cr_pds_ram_-pgen[0]=1 and cr_pds_ram_ret1n[0]=1

Bits	Name	Type	Reset	Description
3:0	RAMRET1N	R/W	4'hF	[3] : 48 64KB np_ram RET2N, 0: Active With cr_pds_ram_-pgen[3]=1 [2] : 32 48KB np_ram RET 2N, 0: Active With cr_pds_ram_-pgen[2]=1 [1] : 16 32KB np_ram RET 2N,0: Active With cr_pds_ram_-pgen[1]=1 [0] : 0 16KB np_ram RET_2N, 0: Active With cr_pds_ram_-pgen[0]=1

19.5.9 pds_gpio_set_pu_pd

Address：0x4000e030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		IO2823PU						RSVD		IO2823PD					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		IO2217PU						RSVD		IO2217PD					

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:24	IO2823PU	R/W	6'h0	pu, pd 00 : no PD/PU/IE on PAD 01 : Pulldown PAD 10 : Pullup PAD 11 : Active IE [0] : PAD_GPIO_23 [1] : PAD_GPIO_24 …. [5] : PAD_GPIO_28
23:22	RSVD			

Bits	Name	Type	Reset	Description
21:16	IO2823PD	R/W	6'h0	pu, pd 00 : no PD/PU/IE on PAD 01 : Pulldown PAD 10 : Pullup PAD 11 : Active IE [0] : PAD_GPIO_23 [1] : PAD_GPIO_24 …. [5] : PAD_GPIO_28
15:14	RSVD			
13:8	IO2217PU	R/W	6'h0	pu, pd 00 : no PD/PU/IE on PAD 01 : Pulldown PAD 10 : Pullup PAD 11 : Active IE [0] : PAD_GPIO_17 [1] : PAD_GPIO_18 …. [5] : PAD_GPIO_22
7:6	RSVD			
5:0	IO2217PD	R/W	6'h0	pu, pd 00 : no PD/PU/IE on PAD 01 : Pulldown PAD 10 : Pullup PAD 11 : Active IE [0] : PAD_GPIO_17 [1] : PAD_GPIO_18 …. [5] : PAD_GPIO_22

19.5.10 pds_gpio_int

Address：0x4000e040

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

Bits	Name	Type	Reset	Description
31:11	RSVD			
10:8	ITSEL	R/W	3'd0	Select VDDCORE GPIO as PDS Interrupt Source (GPIO0 7)
7	RSVD			
6:4	ITMD	R/W	3'b0	PDS Interrupt Mode
3	RSVD			
2	ITCL	R/W	1'b0	Clear PDS Interrupt
1	ITST	R	1'b0	PDS Interrupt Status
0	ITMS	R/W	1'b1	Mask PDS Interrupt

20 Revision history

Table 20.1: Document revision history

Date	Revision	Changes
2020/9/9	1.0	Initial release
2021/3/8	1.1	Modify the GPIO port corresponding to the DAC