Air105 chip data sheet

For questions or suggestions, go to https://gitee.com/openLuat/LuatOS report an issue

revision history

Date Revision 1.00 First draft		describe	author
1.10 Revision 2022-01-14	completed 2022-01-10		Wendal Wangxuefeng

Table of contents

graph index

table index

1 Memory and bus architecture

1.1 System Architecture

The chip system consists of the following units:

System part:

ÿ 32-bit processor
ÿ 640KB SRAM
ÿ 4MB Flash
ÿ System peripherals

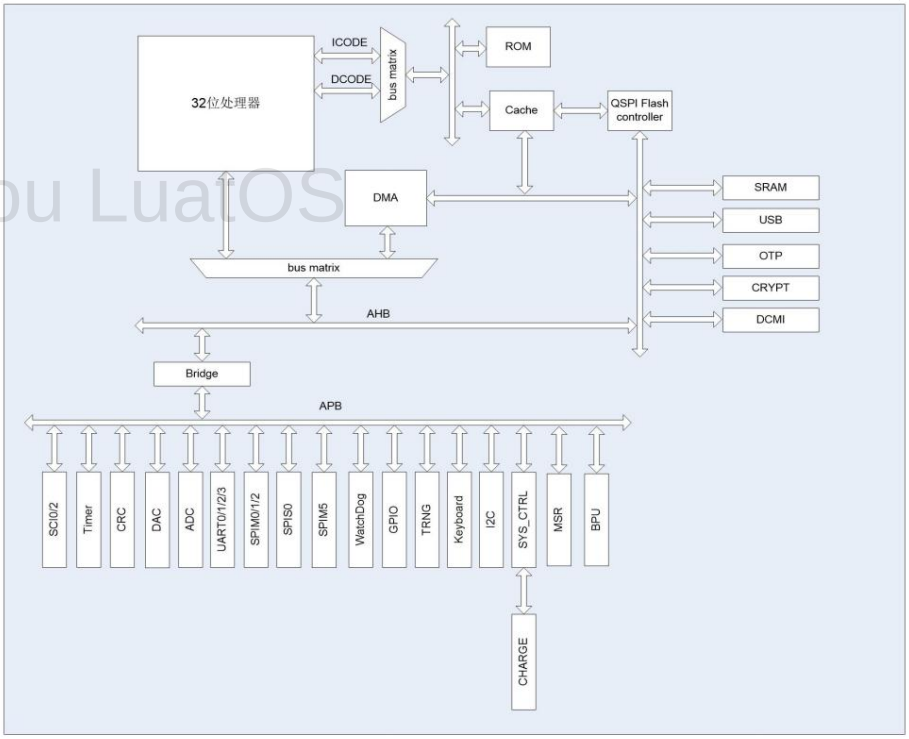


Figure 1- 1 System block diagram

1.2 Memory Map

1.2.1 Peripheral Address Mapping

The following table is the CPU overall address mapping table, see the corresponding chapter for the peripheral address mapping.

Table 1- 1 Peripheral Address Mapping Table

ÿÿÿÿ	Peripherals	bus
0x4000_0000-0x4000_03FF	SSC	AHB
0x4000_0800-0x4000_0BFF	DMA	
0x4000_0C00-0x4000_0FFF	USB	
0x4000_1000-0x4000_13FF	LCD	
0x4000_8000-0x4000_BFFF	OTP	
0x4006_0000-0x4006_FFFF	DCMI	
0x4008_0000-0x4008_FFFF	CACHE	
0x400A_2000-0x400A_2FFF	QSPI	
0x400A_3000-0x400A_3FFF	SPIM5	
0x4001_0000-0x4001_0FFF	SCI0	APB0

0x4001_2000-0x4001_2FFF	CRC	
0x4001_3000-0x4001_3FFF	Timer0	
0x4001_4000-0x4001_4FFF	ADC	
0x4001_5000-0x4001_5FFF	SCI2	
0x4001_6000-0x4001_6FFF	UART0	
0x4001_7000-0x4001_7FFF	UART1	
0x4001_8000-0x4001_8FFF	SPIM1	
0x4001_9000-0x4001_9FFF	SPIM2	
0x4001_A000-0x4001_AFFF	SPIM0	
0x4001_B000-0x4001_BFFF	SPIS0	
0x4001_C000-0x4001_CFFF	Watchdog	
0x4001_D000-0x4001_DFFF	GPIO	
0x4001_E000-0x4001_EFFF	TRNG	
0x4001_F000-0x4001_FFFF	SYS_CTRL	
0x4002_0000-0x4002_FFFF	MSR	APB1
0x4003_0000-0x4003_7FFF	BPU	APB2
0x4004_4000- 0x4004_4FFF	UART2	
0x4004_5000-0x4004_5FFF	UART3	
0x4004_8000-0x4004_8FFF	KEYBOARD	
0x4004_9000-0x4004_9FFF	I2C0	

1.2.2 Embedded RAM

The Air105 has a built-in 640KB of static SRAM. SRAM can be accessed in bytes, halfwords (16 bits) or words (32 bits). SRAM start address: 0x2000_0000

Peripherals	Address	capacity
SRAM	range 0x2000_0000-0x2009_FFFF	640KB

1.2.3 Bit segment access

The memory map includes two bit-band areas. These two bitfield areas map each word in the aliased memory area to the bitfield memory area A bit in the memory area, writing a word in the alias memory area has the same effect as performing a read-modify-write operation on the target bit in the bit-segment area.

Peripheral registers and SRAM are mapped into a bit field area, which allows write and read operations to be performed on a single bit field.

The following mapping formula shows how each word in the alias area corresponds to the corresponding bit in the bitband area:

bit_word_addr = bit_band_base + (byte_offset×32) + (bit_number×4)

in:

bit_word_addr is the address of a word in the aliased memory area that maps to a certain target bit.

bit_band_base is the starting address of the alias band.

byte_offset is the sequence number in the bit field of the byte containing the target bit

bit_number is the position of the target bit (0-31)

example:

The following example shows how to map bit 2 of the byte at SRAM address 0x20000300 in the alias area:

0x22006008 = 0x22000000 + (0x300×32) + (2×4).

A write operation to address 0x22006008 has the same effect as a read-modify-write operation to bit 2 of the address 0x20000300 byte in SRAM.

the same effect.

Reading address 0x22006008 returns the value of bit 2 (0x01 or 0x00) of byte 0x20000300 in SRAM.

2 Chip Feature Description

2.1 Electrical Characteristics

Table 2- 1 Limit parameters

parameter	illustrate	scope	unit
Iddpd	Shutdown current	--	nA
Tamb	Operating temperature	-40~+85	°C
Tstg	stored temperature	-40~+125	°C
Ground	land	-0.3~0.3	V
Voh	Digital output high level	VDD -0.3 ~ VDD+0.3	V
Vol	digital output low	<0.4	V
Ioh	Source current (PA2/3/4/5, PC6/7/8/9)	27 (@3V)	mA
	Source current (other IO) Sink current	16 (@3V)	mA
Iol	(PA2/3/4/5, PC6/7/8/9)	27 (@0.5V)	mA
	Sink current (other	16 (@0.5V)	mA
Vih	IO) digital input high level	ÿ0.7×VDD	V
ViL	digital input low	ÿ0.3×VDD	V

Table 2- 2 Electrical Characteristics

parameter	**Conditions (-40°C to** +85°C)	value		unit
		minimum	maximum	
VCC		3.6	5.5	V
CHARGE_VCC		4.7	5.4	V
AVD33		2.7	3.6	V
VDD33		2.7	3.6	V
VBAT33		2	3.6	V
Vol	VDD=3.3V	-	0.4	V
Voh	VDD=3.3V	VDD – 0.3	-	V
VIH	VDD=3.3V	0.7×VDD	-	V
VIL	VDD=3.3V	-	0.3×VDD	V

Table 2- 3 Safety-related characteristics

sensor	illustrate	scope	unit
Temperature Sensor	High temperature detection range	95~115	°C
	Low temperature detection range	-30~-45	°C
Voltage sensor	Main power supply voltage high voltage detection range	4.0±0.1	V
	Mains voltage low voltage detection range	2.8±0.1	V
	Battery voltage high voltage detection range	4.0±0.1	V
	Battery voltage low voltage detection range	1.9±0.1	V
clock frequency sensor	12M clock frequency detection range	12±50%	MHz
	32K clock frequency detection range	32±50%	KHz

Table 2- 4 Power consumption list

Operating mode	illustrate	Power consumption	unit
RUN	ÿ All peripherals are fully open ÿ core 204MHz, HCLK 102MHz, PCLK 51MHz ÿ core 192MHz, HCLK 96MHz, PCLK 48MHz ÿ core 168MHz, HCLK 84MHz, PCLK 42MHz ÿ All peripherals are turned off ÿ core 204MHz, HCLK 102MHz, PCLK 51MHz ÿ core 192MHz, HCLK 96MHz, PCLK 48MHz ÿ core 168MHz, HCLK 84MHz, PCLK 42MHz	 67 64 59 43 41 38	mA
CPU Sleep All peripherals are turned off at 204@MHz		19 mA	
Deep Sleep ÿ Support IO low level wake-up		450	uA
VBAT	ÿ CHARGE_VBAT is left floating Internal sensor fully open All internal sensors off ÿ CHARGE_VBAT is connected to lithium battery Internal sensor fully open, 5.6uA@CHARGE_VBAT Internal sensor fully closed, 5.6uA@CHARGE_VBAT	 2.1 1.7 0.7 0.3	uA

2.2 Pin Definition

Table 2- 5 Air105 QFN88 10mm×10mm package pin definition

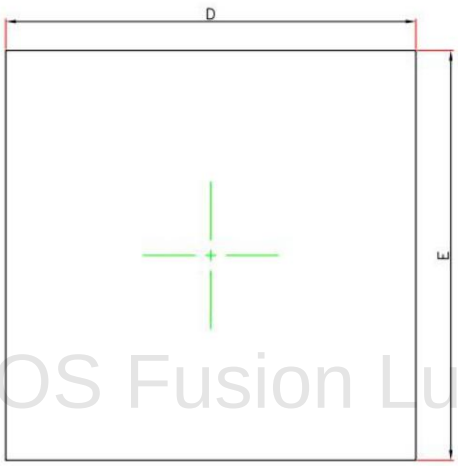
PIN No.	PID Name	ALT0	ALT1 (default)	ALT2	ALT3	Remark
1	CVCC					IC card power supply
2	VDD33					
3	VCC					5V to 3.3V LDO input
4	VDD33_OUT					5V to 3.3V output
5	POWER_KEY					switch button
6	PA7	SCI0_VCCEN	PA[7] PWM7		SPI1_CSN	
7	PA6	SCI0_DET	PA[6] PWM6		SPI1_SCK	
8	PB2	SPI2_SCK	PB[2] PWM2		UART2_RX/IrDA_IN	
9	PB3	SPI2_CSN	PB[3] PWM3		UART2_TX/IrDA_OUT	
10	PB4	SPI2_MOSI	PB[4] PWM4		UART2_CTS	
11	PB5	SPI2_MISO	PB[5] PWM5		UART2_RTS	
12	PE6	UART3_CTS	PE[6]	I2C0_SCL		

13	PE7	UART3_RTS	PE[7]	I2C0_SDA		
14	PE8		PE[8]	UART3_RX/IrDA_IN		
15	PE9		PE[9]	UART3_TX/IrDA_OUT		
16	PE10		PE[10] UART3_CTS			
17	PE11		PE[11] UART3_RTS			
18	PA0	UART0_RX/IrDA_IN	PA[0] PWM0			
19	PA1	UART0_TX/IrDA_OUT	PA[1] PWM1			Serial download pin
20	PA2	UART0_CTS	PA[2] PWM2		I2C0_SCL	
twenty one	PA3	UART0_RTS	PA[3] PWM3		I2C0_SDA	
twenty two	PB0	I2C0_SCL	PB[0] PWM0		XTAL32K	
twenty three	PB1	I2C0_SDA	PB[1] PWM1		CLK_24M	Configurable output 24M
twenty four	CHARGE_VBAT					CHARGE power output, Connect the battery
25	CHARGE_VCC					CHARGE power input
26	PD1		PD[1]		DCMIS_DATA0	
27	PD2		PD[2]		DCMIS_DATA1	
28	PD3		PD[3]		DCMIS_DATA2	
29	PD8		PD[8]	UART2_RX/IrDA_IN	DCMIS_DATA3	
30	PD9		PD[9]	UART2_TX/IrDA_OUT	DCMIS_DATA4	
31	PD10		PD[10] KeyBoard7		DCMIS_DATA5	
32	PD11		PD[11] KeyBoard8		DCMIS_DATA6	
33	PE0		PE[0]	KeyBoard4	DCMIS_DATA7	
34	PD6	UART1_CTS	PD[6]		DCMIS_DATA8	
35	PD7	UART1_RTS	PD[7]		DCMIS_DATA9	
36	PC6		PC[6] PWM4		DCMIS_DATA10	
37	PC7		PC[7] PWM5		DCMIS_DATA11	
38	PC8		PC[8] PWM6		DCMIS_DATA12	
39	PC9		PC[9] PWM7		DCMIS_DATA13	
40	PE1		PE[1]	KeyBoard5	DCMIS_VSYNC	
41	PE2		PE[2]	KeyBoard6	DCMIS_HSYNC	
42	PE3		PE[3]		DCMIS_PIX_CLK	
43	PB12	SPI0_CLK	PB[12] PWM3		UART1_RX/IrDA_IN	

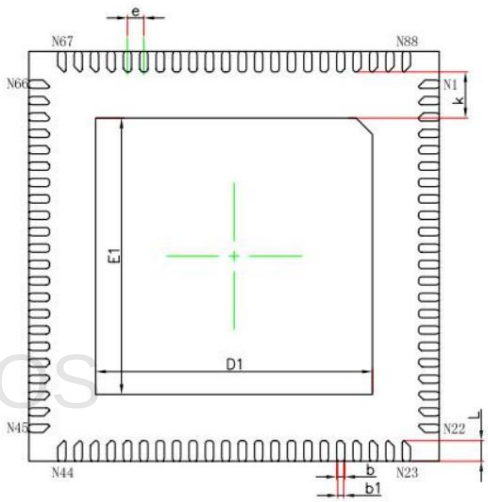
44	VSS					chip ground
45	PB13	SPI0_CSN	PB[13] PWM4		UART1_TX/IrDA_OUT	
46	PB14	SPI0_MOSI	PB[14] PWM5		UART1_CTS	
47	PB15	SPI0_MISO	PB[15] PWM6		UART1_RTS	
48	PC12		PC[12] SPI0_SCK		SPI5_MISO	
49	PC13		PC[13] SPI0_CSN		SPI5_MOSI	
50	PC14		PC[14] SPI0_MOSI		SPI5_CSN	
51	PC15		PC[15] SPI0_MISO		SPI5_CLK	
52	VDD33					
53	PD13	UART2_TX/IrDA_OUT	PD[13]	KeyBoard1		
54	PD12	UART2_RX/IrDA_IN	PD[12] KeyBoard0			
55	PD15	UART2_RTS	PD[15]	KeyBoard3		
56	PD14	UART2_CTS	PD[14] KeyBoard2			
57	NC					
58	NC					
59	NC					
60	NC					
61	REFP					Connect 1uF capacitor
62	PC5		PC[5] ADC_IN6		CLK_27P12	Configurable output 27.12M
63	PC4	TCK	PC[4] ADC_IN5		XTAL32K	
64	PC3	TMS	PC[3] ADC_IN4		UART1_RTS	
65	PC1	TDI	PC[1] ADC_IN2/DAC		UART1_TX/IrDA_OUT	
66	PC0	TRST	PC[0] ADC_IN1		UART1_RX/IrDA_IN	
67	VDD25					Connect 1uF capacitor to ground
68	DN					USB DN
69	DP					USB-DP
70	VBUS					USB VBUS
71	VDD33					
72	XO12M					XTAL 12MHz Output
73	XI12M					XTAL 12MHz Input
74	VDD12					Connect 1uF capacitor to ground

75	AVD33					
76	XI32					XTAL 32KHz Input
77	XO32					XTAL 32KHz Output
78	NC					
79	NC					
80	NC					
81	NC					
82	NC					
83	NC					
84	VBAT33					Button Battery
85	PA5		PA[5] PWM5		CLK_24M	Configurable output 24M
86	PA8	SCI0_CLK	PA[8]		SPI1_MOSI	When multiplexing as IO, you must first Turn on the power of the IC card and input the The high level of the output signal is the IC Card output level
87	PA9	SCI0_RSTN	PA[9]		SPI1_MISO	
88	PA10	SCI0_IO	PA[10]			

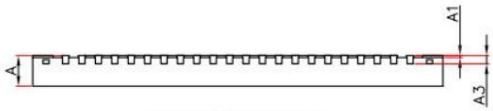
2.3 Package Information



TOP VIEW



SIDE VIEW



BOTTOM VIEW

Symbol	Dimensions In Millimeters		Dimensions In Inches	
	MIN.	MAX.	MIN.	MAX.
A	0.700	0.800	0.028	0.031
A1	0.000	0.050	0.000	0.002
A3	0.203REF.		0.008REF.	
D	9.900	10.100	0.390	0.398
E	9.900	10.100	0.390	0.398
D1	6.650	6.850	0.262	0.270
E1	6.650	6.850	0.262	0.270
k	1.125REF.		0.044REF.	
b	0.150	0.250	0.006	0.010
b1	0.150REF.		0.006REF.	
e	0.400BSC.		0.016BSC.	
L	0.400	0.600	0.016	0.024

Figure 2- 1Air105 QFN88 10mm*10mm package size

3 System Control (SYSCTRL)

3.1 Soft reset

The system provides software reset operation, write "1" to the corresponding operation bit of the software reset register (SOFT_RST1 or SOFT_RST2) to realize the corresponding module reset operation, and reset to "0" by hardware after writing "1". Some module resets need to clear the protection lock register (LOCK_R) in advance After the reset operation, please refer to the register description for details.

3.2 Clock

The chip supports internal 12MHz oscillator and external 12MHz crystal. The precision of the integrated 12MHz crystal is ±2%. After PLL After frequency multiplication, it provides input for the system. The frequency of the PLL clock after frequency multiplication can be configured by software, and its frequency can be configured as: 108MHz, 120MHz, 132MHz, 144MHz, 156MHz, 168MHz, 180MHz, 192MHz, 204MHz.

After the chip is powered on and reset, the entire ROM boot process is based on the internal 12MHz oscillator. If the manufacturer needs When using an external 12MHz crystal, manual switching to external 12MHz operation is required.

The entire safe area of the chip works based on the internal 32KHz, and the RTC works based on the internal OSC 32K by default, which can be switched by software to use external External XTAL 32K work, support internal or external 32KHz output.

The Air105 clock tree is as follows:

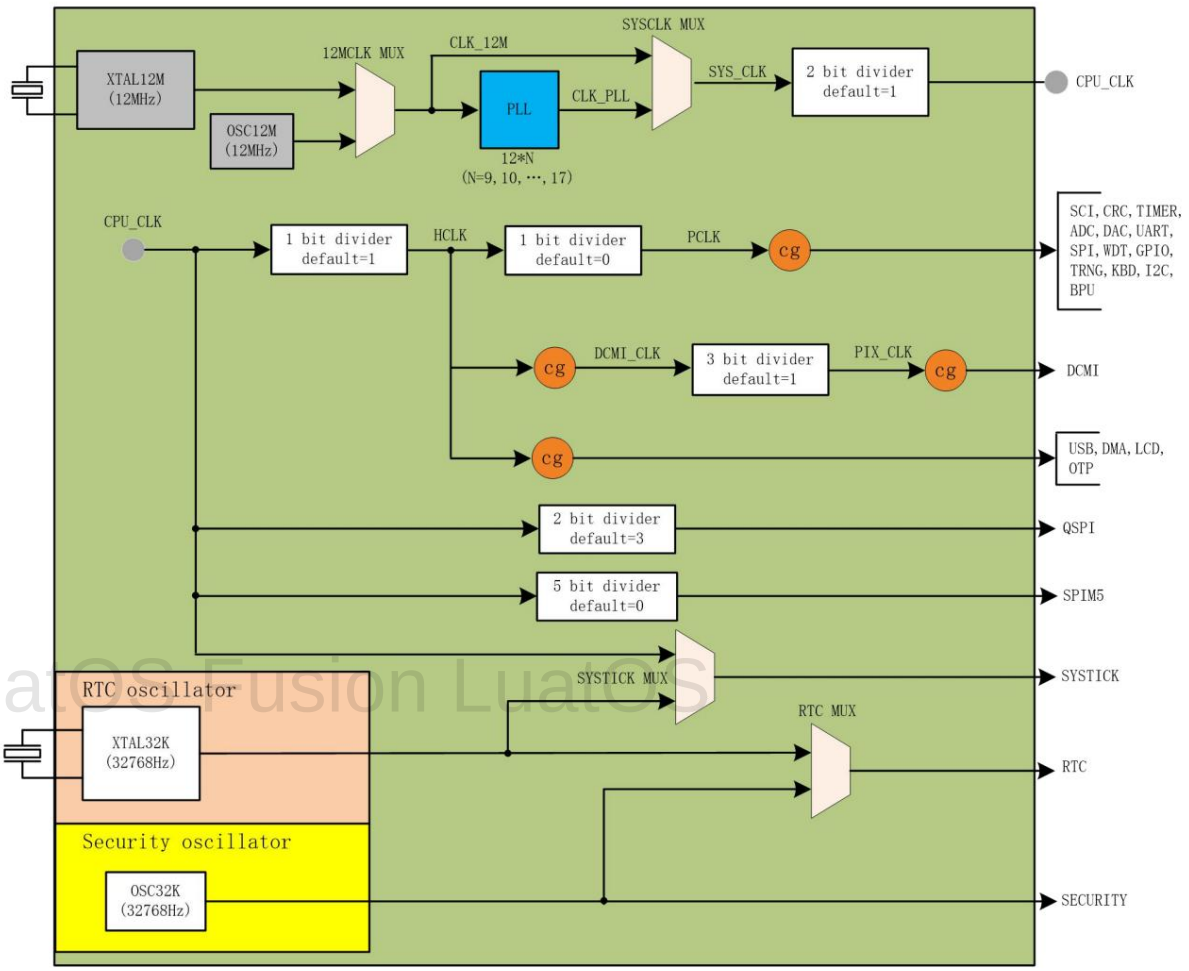


Figure 3- 1 Air105 Clock Tree

3.2.1 External clock source

The external system clock of the chip is required to be 12MHz, and the external real-time clock is required to be 32KHz.

3.2.2 PLL clock

The external clock is multiplied by the PLL to obtain the PLL clock, which can be configured by register FREQ_SEL to generate 9 different PLL clocks, as follows: ÿ 108MHz ÿ 120MHz ÿ 132MHz ÿ 144MHz ÿ 156MHz ÿ 168MHz ÿ 180MHz ÿ 192MHz ÿ 204MHz

3.2.3 FCLK clock

The PLL clock is divided by the clock frequency division unit as the FCLK clock. The PLL clock is divided by register: ÿ FCLK= PLL clock ÿ FCLK= PLL clock/2 ÿ FCLK= PLL clock/4

3.2.4 HCLK clock

The FCLK clock is divided by the clock frequency division unit as the HCLK clock. Divide the FCLK clock by register: ÿ HCLK= FCLK (when the FCLK frequency is less than or equal to 102MHz) ÿ HCLK= FCLK / 2

3.2.5 PCLK clock

The HCLK clock is divided by the clock frequency division unit as the PCLK clock. The HCLK clock is divided by the register: ÿ PCLK = HCLK / 2

3.2.6 Peripheral Clock Management

The system provides a clock gating control register (CG_CTRL) to manage peripheral clocks. The user can turn it on and off through this register Corresponds to the peripheral clock. After the peripheral clock is turned off, the peripheral will no longer run, and the system power consumption can be controlled more flexibly through this register.

3.3 Low Power Control

After a system or power reset, the safety CPU is running. When the CPU does not need to continue to run, can take advantage of a variety of low power consumption mode to save power, such as waiting for an external event to occur. Power consumption can be reduced in the following ways: ÿ Reduce system, AHB and APB bus clocks (FREQ_SEL) ÿ Turn off peripheral clocks (CG_CTRL) that are not used on AHB and APB buses ÿ Sleep mode (CPU core is stopped, all peripherals are still running)

ÿ Deep sleep mode (CPU core and peripherals are stopped)

Power state transition diagram:

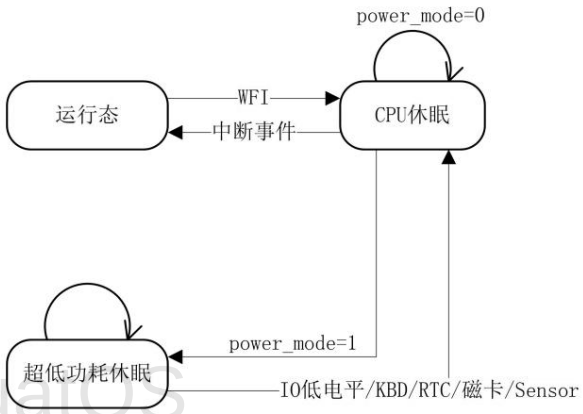


Figure 3- 2Air105 power consumption state transition diagram

Before executing the WFI instruction, the FREQ_SEL.power_mode register must be configured to select CPU sleep or ultra-low power sleep. When power_mode = 0, CPU sleep is executed, and any interrupt can wake up the CPU; when power_mode = 1, ultra-low power is executed. Power consumption sleep, first turn off the CPU clock to enter the CPU sleep mode, and then turn off the PLL to enter the ultra-low power sleep mode, in the ultra-low power state Only IO low level, KBD, RTC, card swiping, and Sensor attacks can switch the chip from the ultra-low power sleep state to the CPU sleep state. Wake up the CPU after generating an interrupt.

The wake-up source of ultra-low power sleep is controlled by four registers: WKUP_TYPE_EN, WKUP_P0_EN~WKUP_P2_EN system.

3.4 Peripheral Control

The system control area contains a peripheral control register (PHER_CTRL), which realizes the control of some peripherals. The control contents are as follows:

ÿ SPI0 working mode selection

ÿ SCI0/2 VCCEN signal active level selection

ÿ SCI0/2 card detection signal active level selection

3.5 Register Description

3.5.1 Address Mapping Table

SYSCTRL base address table

Address	Base	Peripherals	bus
range 0x4001_F000-0x4001_FFFF	address 0x4001_F000	SYSCTRL	APB0

Table 3- 1 SYSCTRL register table

Offset address register	name	Width (bit) Reset value	
0x00	FREQ_SEL	32	0x200D395A
0x04	CG_CTRL1	32	0x04100000
0x08	CG_CTRL2	32	0x00000000
0x0C	SOFT_RST1	32	0x00000000
0x10	SOFT_RST2	32	0x00000000

0x14	LOCK_R	32	0xF0000000
0x18	PHER_CTRL	32	0x00100000
0x1C-0x28	SYS_RSVD	32	0x00000000
0x2C	HCLK_1MS_VAL	32	0x0000BB80
0x30	PCLK_1MS_VAL	32	0x00005DC0
0x34	ANA_CTRL	32	0x0002C601
0x38	DMA_CHAN	32	0x0A0B0203
0x3C	SCI0_GLF	32	0x20060000
0x40-0x48 Reserved		32	0x00000000
0x4C	LDO25_CR	32	0x00000028
0x50	DMA_CHAN1	32	0x0A0B0203
0x54-0xFC reserved	reserved reserved	32	0x00000000
0100		32	0x07F88800
0104		32	0x0000000C
0108	USBPHY_CR1	32	0x204921AE
010C	USBPHY_CR2	32	0x40000000
0110	USBPHY_CR3	32	0x00000000
0114	ISO7816_CR1	32	0x00000080
	reserved 0118	32	0x00000000
011C	CHG_CTRL	32	0x00000000
0120-0200 Reserved	0204	32	0x00000000
	RSVD_POR	32	0x00000000
0208-03EC Reserved		32	0x00000000
03F0	CALIB_CSR	32	0xAB000080
03F4	DBG_CR	32	0x00000000
03F8	CHIP_ID	32	0x03070000

3.5.2 Clock Frequency Select Register (FREQ_SEL)

Offset address: 0x0000 Reset value: 0x200D395A

31	30	29	28	27	26	25	twenty four
pix_clk_freq			reserved		power_mode		
rw			ro		rw		

twenty three	22	twenty one	20	19	18	17	16
Reserved			xtal_sel				
ro			rw				

15	14	13	12	11	10	9	8
reserved		sys_ck_src ck12m_src		reserved		cpu_freq_sel	
ro		rw	rw	ro		rw	

7	6	5	4	3	2	1	0
reserved			hclk_freq_sel	reserved			pclk_freq_sel
ro			rw	ro			rw

Bit	Name	W/R	Description
31:29	pix_clk_freq	W/R -	pix_clk generation for DCMI 0: reserved 1: (clk_dcmis)/4 2: (clk_dcmis)/6

			and so on
28:27	reserved	-	reserved
26:24	power_mode	W/R	Low power mode selection: 3'h0: Turn off the CPU clock; 3'h1:DEEP SLEEP (clock off, ROM optional power down, SRAM can optionally enter retain); 3'h2~3'h7: reserved; Note: Before entering the low power mode, first configure this register to select the power consumption mode, and then enter through the **WFI** command of the processor .
23:21	reserved	-	reserved
20:16	xtal_sel	W/R	When clk_sys_src is 1, the system clock frequency is selected: 5'h00~5'h07: reserved; 5'h08: 108MHz; 5'h09: 120MHz; 5'h0a: 132MHz; 5'h0b: 144MHz; 5'h0c: 156MHz; 5'h0d: 168MHz; 5'h0e: 180MHz; 5'h0f: 192MHz; 5'h10: 204MHz; 5'h11~5'h1f: reserved;
15:14	reserved	-	reserved
13	sys_ck_src	W/R	System main clock source selection: 0: Before the system clock comes from the PLL; 1: After the system clock comes from the PLL
12	ck12m_src	W/R	12MHz clock source selection: 0: off-chip XTAL; 1: on-chip OSC
11:10	reserved	-	reserved
9:8	cpu_freq_sel	W/R	CPU CLOCK frequency selection control word (reference is PLL output CLOCK): 00: PLL output frequency; 01: 2 division of PLL output frequency; 1x: 4 division of PLL output frequency;
7:5	reserved	-	reserved
4	hclk_freq_sel	W/R	HCLK frequency selection control word (reference is CPU CLOCK): 0: No frequency division (when the CPU CLOCK frequency is less than or equal to 102MHz) 1: divide by two;
3:1	reserved	-	reserved
0	pclk_freq_sel	W/R	PCLK divider value (based on HCLK): 0: PCLK is divided by 4 based on HCLK; 1: PCLK is divided by 4 on the basis of HCLK;

3.5.3 Clock Gating Control Register 1 (CG_CTRL1)

Offset address: 0x0004 Reset value: 0x04100000

31	30	29	28	27	26	25 twenty four	
trng_cg_en	adc_cg_en	crc_cg_en	reserved	kbd_cg_en	bpu_cg_en		reserved
rw	rw	rw	ro	rw	rw	ro	
twenty three	twenty two	twenty one	20	19	18	17	16
dcmis_cg_en	csi2_cg_en	timer_cg_en	gpio_cg_en	Reserve i2c0_cg_en		Reserve sci2_cg_en	
rw	rw	rw	rw	ro	rw	ro	rw
15	14	13	12	11	10	9	8

Reserve	sci0_cg_en	spi5_cg_en	reserved	spi2_cg_en	spi1_cg_en	spi0_cg_en
ro	rw	rw	ro	rw	rw	rw
7	6	5	4 3	2	1	0

reserved	uart3_cg_en	uart2_cg_en	uart1_cg_en	uart0_cg_en
ro	rw	rw	rw	rw

Bit	Name	W/R	Description
31	trng_cg_en	W/R	Random number block gated clock enable: 0: Disable; 1: Enable
30	adc_cg_en	W/R	ADC gate clock enable: 0: Disable; 1: Enable
29	crc_cg_en	W/R	CRC Gated Clock Enable: 0: Disable; 1: Enable
28	reserved	- reserved	
27	kbd_cg_en	W/R	Keyboard module gated clock enable: 0: Disable; 1: Enable
26	bpu_cg_en	W/R	Battery powered module gated clock enable: 0: Disable; 1: Enable
25:24	reserved	- reserved	
twenty three	dcmis_cg_en	W/R	DCMIS gate enable 0: Disable; 1: Enable
twenty two	reserved	- reserved	
twenty one	timer_cg_en	W/R	Timer gate clock enable: 0: Disable; 1: Enable
20	gpio_cg_en	W/R	GPIO module gate clock enable: 0: Disable; 1: Enable
19	reserved	- reserved	
18	i2c0_cg_en	W/R	I2C0 gated clock enable: 0: Disable; 1: Enable
17	reserved	- reserved	
16	sci2_cg_en	W/R	SCI2 gated clock enable: 0: Disable; 1: Enable
15	reserved	- reserved	
14	sci0_cg_en	W/R	SCI0 gated clock enable: 0: Disable; 1: Enable
13	spi5_cg_en	W/R	SPI5 gated clock enable: 0: Disable; 1: Enable
12:11	reserved	- reserved	
10	spi2_cg_en	W/R	SPI2 gated clock enable: 0: Disable; 1: Enable
9	spi1_cg_en	W/R	SPI1 gated clock enable: 0: Disable; 1: Enable
8	spi0_cg_en	W/R	SPI0 gated clock enable: 0: Disable; 1: Enable
7:4	reserved	- reserved	
3	uart3_cg_en	W/R	UART3 gated clock enable: 0: Disable; 1: Enable
2	uart2_cg_en	W/R	UART2 gated clock enable: 0: Disable; 1: Enable

1	uart1_cg_en	W/R	UART1 gated clock enable: 0: Disable; 1: Enable
0	uart0_cg_en	W/R	UART0 gated clock enable: 0: Disable; 1: Enable

3.5.4 Clock Gating Control Register 2 (CG_CTRL2)

Offset address: 0x0008 Reset value: 0x00000000

31	30	29	28	27	26	25	twenty four
reserved		dma_cg_en	usb_cg_en	reserved			
ro		rw	rw	ro			

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
reserved		qr_cg_en	reserved	otp_cg_en	gpu_cg_en	lcd_cg_en	crypt_cg_en
ro		rw	ro	rw	rw	rw	rw

Bit	Name	W/R	Description
31:30	reserved	RO reserved bit	
29	dma_cg_en	W/R	DMA gated clock enable: 0: Disable; 1: Enable
28	usb_cg_en	W/R	USB gated clock enable: 0: Disable; 1: Enable
27:6	reserved	RO reserved bit	
5	qr_cg_en	W/R	Image coprocessor gated clock enable: 0: Disable; 1: Enable
4	reserved	RO reserved bit	
3	otp_cg_en	W/R	OTP gated clock enable: 0: Disable; 1: Enable
2	gpu_cg_en	W/R	GPU Gated Clock Enable: 0: Disable; 1: Enable
1	lcd_cg_en	W/R	LCD gate clock enable: 0: Disable; 1: Enable
0	crypt_cg_en	W/R	Algorithm acceleration engine gated clock enable: 0: Disable; 1: Enable

3.5.5 Software Reset Register 1 (SOFT_RST1)

Offset address: 0x000C Reset value: 0x00000000

31	30	29	28	27	26	25	twenty four
srst_trng	srst_adc	srst_crc	reserved	srst_kbd		reserved	
wc	wc	wc	ro	wc		ro	

twenty three	twenty two	twenty one	20	19	18	17	16
srst_dcmis reserved	srst_timer	0	srst_gpio reserved	srst_i2c0 reserved	srst_sci2		
wc	ro	wc	wc	ro	wc	ro	wc
15	14	13	12	11	10		8
Reserve srst_sci0		srst_spi5	reserved		srst_spi2	9 srst_spi1	srst_spi0
ro	wc	wc	ro		wc	wc	wc
7	6	5	4	3	2	1	0
reserved				srst_uart3	srst_uart2	srst_uart1	srst_uart0
ro				wc	wc	wc	wc

For all soft reset signals, write 1 to the corresponding bit to perform a soft reset operation (the hardware will automatically return to 0 after writing 1, and software does not need to be cleared to 0)

Bit	Name	W/R	Description
31	srst_trng	WC	random number soft reset signal
30	srst_adc	WC	ADC module soft reset signal
29	srst_crc	WC	CRC module soft reset signal
28	reserved	-	reserved
27	srst_kbd	WC	KBD module soft reset signal
26:24	reserved	-	reserved
twenty three	srst_dcmis	WC	DCMIS soft reset signal
twenty two	reserved	-	reserved
twenty one	srst_timer0	WC	Timer0 soft reset signal
20	srst_gpio	WC	GPIO module soft reset signal
19	reserved	-	reserved
18	srst_i2c0	WC	I2C0 soft reset signal
17	reserved	-	reserved
16	srst_sci2	WC	SCI2 soft reset signal
15	reserved	-	reserved
14	srst_sci0	WC	SCI0 soft reset signal
13	srst_spi5	WC	SPI5 soft reset signal
12:11	reserved	-	reserved
10	srst_spi2	WC	SPI2 soft reset signal
9	srst_spi1	WC	SPI1 soft reset signal
	srst_spi0	WC	SPI0 soft reset signal
8	reserved	RO	reservation
7:4 3	2 srst_uart3	WC	UART3 soft reset signal
2	1 srst_art	WC	UART2 soft reset signal
1		WC	UART1 soft reset signal
0		WC	UART0 soft reset signal

3.5.6 Software Reset Register 2 (SOFT_RST2)

Offset address: 0x0010 Reset value: 0x00000000

31 30		29	28	27	26	25	twenty four
srst_glb	srst_cm3	srst_dma	srst_usb	srst_cache reserved	srst_pramc	reserved	
wc	wc	wc	wc	wc	ro	wc	ro
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							
15	14	13	12	11	10	9	8

31	30	29	28	27	26	25	twenty four
reserved							
ro							

23	22	21	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
reserved		srst_qr	reserved		srst_gpu	1 srst_lcd	srst_crypt
ro		wc	ro		wc	wc	wc

For all soft reset signals, write 1 to the corresponding bit to perform a soft reset operation (the hardware will automatically return to 0 after writing 1, and software does not need to be cleared to 0)

31	30	29	28	27	26	25	twenty four
srst_glb	srst_cm3	srst_dma	srst_usb	srst_cache	Reserved		
wc	wc	wc	wc	wc	ro		

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
reserved		srst_qr	reserved		srst_gpu	srst_lcd	srst_crypt
ro		wc	ro		wc	wc	wc

Bit	Name	W/R	Description
31	srst_glb	The global soft reset signal of the digital part of the WC chip, used in conjunction with the lock signal	
30	srst_cm3	WC CPU core soft reset signal, used in conjunction with the lock signal	
29	srst_dma	WCDMA soft reset signal, used in conjunction with lock signal	
28	srst_usb	WC	USB interface soft reset signal, used in conjunction with lock signal Note: This bit needs to be cleared by software
27	srst_cache	WC CACHE soft reset signal	
26:6	reserved	- reserved	
5	srst_qr	WC image coprocessor soft reset signal	
4:3	reserved	RO reservation	
2	srst_gpu	RO	GPU soft reset
1	srst_lcd	WC LCD interface soft reset signal	
0	srst_crypt	WC Large number operation acceleration engine soft reset signal	

3.5.7 Protection Lock Register (LOCK_R)

Offset address: 0x0014 Reset value: 0xF0000000

31	30	29	28	27	26	25	twenty four
srst_glb_lock	srst_cm3_lock	srst_dma_lock	srst_usb_lock	reserved			
rw	rw	rw	rw	ro			

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
reserved							
ro							

Bit	Name	W/R	Description
31	srst_glb_lock	W/R	Chip global soft reset protection control bit: 0: Allow software to perform global soft reset; 1: Software is not allowed to perform global soft reset;
30	srst_cm3_lock	W/R	CPU core soft reset protection control bits: 0: Allow software to perform CPU core soft reset; 1: Software is not allowed to perform CPU core soft reset;
29	srst_dma_lock	W/R	DMA soft reset protection control bits: 0: Allow software to perform DMA soft reset; 1: Software is not allowed to perform DMA soft reset;
28	srst_usb_lock	W/R	USB soft reset protection control bits: 0: Allow software to perform USB soft reset; 1: Software is not allowed to perform USB soft reset;
27:0	reserved	RO reservation	

3.5.8 Peripheral Control Register (PHER_CTRL)

Offset address: 0x0018 Reset value: 0x00000000

31	30	29	28	27	26	25	twenty four
reserved							spi0_slv_sel
ro							rw
twenty three	twenty two	twenty one	20	19	18	17	16
Reserve	sci2_vccen_inv	Reserve	sci0_vccen_inv	Reserve	sci2_cdet_inv	Reserve	sci0_cdet_inv
ro	rw	ro	rw	ro	rw	ro	Rw
15	14	13	12	11	10	9	8
reserved							
ro							
7	6	5	4	3	2	1	0
reserved							
ro							

Bit	Name	W/R -	Description
31:25	reserved	Reserved	
twenty four	spi0_slv_sel	W/R	SPI0 working mode selection: 0: master mode; 1: slave mode
twenty three	reserved	- reserved	
twenty two	sci2_vccen_inv	W/R	SCI2 VCCEN signal active level selection 0: high valid; 1: low valid
twenty one	reserved	- reserved	
20	sci0_vccen_inv	W/R	SCI0 VCCEN signal active level selection 0: high valid; 1: low valid
19	reserved	RO reservation	
18	sci2_cdet_inv	W/R	SCI2 card detection signal effective level selection 0: high valid; 1: low valid
17	reserved	- reserved	
16	sci0_cdet_inv	W/R	SCI0 card detection signal effective level selection 0: high valid; 1: low valid

15:0	reserved	RO reservation

3.5.9 Cycle corresponding to HCLK 1ms (HCLK_1MS_VAL)

Offset address: 0x002C Reset value: 0x0000BB80

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
val_1ms_hclk															

Bit	Name	W/R	Description
31:17		-	reserved
16:0	Reserve val_1ms_pclk	RO	is based on the cycle value required to clock 1ms on HCLK

3.5.10 Cycle corresponding to PCLK 1ms (PCLK_1MS_VAL)

Offset address: 0x0030 Reset value: 0x00005DC0

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

15	14	13	12	11	10	9	8 7	6	5	4	3	2	1	0
val_1ms_pclk														

Bit	Name	W/R	Description
31:18		-	reserved
17:0	Reserve val_1ms_pclk	RO	is based on the cycle value required to clock 1ms on PCLK

3.5.11 Analog Control Register (ANA_CTRL)

Offset address: 0x0034 Reset value: 0x0002C601

31	30	29	28	27	26	25	twenty four
reserved							
ro							

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
rom_pd_en	reserved	usb33_pd_en	usb12_pd_en	reserved			
rw	ro	rw	rw	ro			

Bit	Name	W/R	Description
31:8	reserved	-	Reserved
7	rom_pd_en	RW	rom power off enable in deep sleep mode: 0: no power down; 1: Power down;

6	reserved	- reserved	
5	usb33_pd_en	RW	USB 3.3V power down enable in deep sleep mode: 0: no power down; 1: Power down;
4	usb12_pd_en	RW	USB 1.2V power down enable in deep sleep mode: 0: no power down; 1: Power down;
3:0	reserved	- reserved	

3.5.12 DMA channel selection (DMA_CHAN)

Offset address: 0x0038 Reset value: 0x0A0B0203

31	30	29	28	27	26	25	twenty four
reserved				dma_ch3_if			
ro				rw			
twenty three	twenty two	twenty one	20	19	18	17	16
reserved				dma_ch2_if			
ro				rw			
15	14	13	12	11	10	9	8
reserved				dma_ch1_if			
ro				rw			
7	6	5	4	3	2	1	0
reserved				dma_ch0_if			
ro				rw			

Bit	Name	W/R -	Description
31:30	reserved	Reserved	
29:24	dma_ch3_if	W/R	DMA hardware handshaking interface select for channel 3: 5'h0: DCMI; 5'h1: LCD; 5'h2: UART0 TX; 5'h3: UART0 RX; 5'h4: UART1 TX; 5'h5: UART1 RX; 5'h6: DAC; 5'ha: SPI0 TX; 5'hb: SPI0 RX; 5'hc: SPI1 TX; 5'hd: SPI1 RX; 5'he: SPI2 TX; 5'hf: SPI2 RX; 5'h14: UART2 TX; 5'h15: UART2 RX; 5'h16: UART3 TX; 5'h17: UART3 RX; 5'h18: I2C TX; 5'h19: I2C RX; 5'h1A: QSPI 5'h20: SPI5 RX; 5'h21: SPI5 TX;
23:22		- reserved	
21:16	Reserve dma_ch2_if	W/R DMA hardware handshaking interface select for	

			channel 2: 5'h0: DCMI; 5'h1: LCD; 5'h2: UART0 TX; 5'h3: UART0 RX; 5'h4: UART1 TX; 5'h5: UART1 RX; 5'h6: DAC; 'ha: SPI0 TX; 5'hb: SPI0 RX; 5'hc: SPI1 TX; 5'hd: SPI1 RX; 5'he: SPI2 TX; 5'hf: SPI2 RX; 5'h14: UART2 TX; 5' h15: UART2 RX; 5'h16: UART3 TX; 5'h17: UART3 RX; 5'h18: I2C TX; 5'h19: I2C RX; 5'h1A: QSPI 5'h20: SPI5 RX; 5'h21: SPI5 TX; - reserved
15:14	reserved		
13:8	dma_ch1_if	W/R	DMA hardware handshaking interface select for channel 1: 5'h0: DCMI; 5'h1: LCD; 5'h2: UART0 TX; 5'h3: UART0 RX; 5'h4: UART1 TX; 5'h5: UART1 RX; 5 'h6: DAC; 5'ha: SPI0 TX; 5'hb: SPI0 RX; 5'hc: SPI1 TX; 5'hd: SPI1 RX; 5'he: SPI2 TX; 5'hf: SPI2 RX; 5'h14 : UART2 TX; 5'h15: UART2 RX; 5'h16: UART3 TX; 5'h17: UART3 RX; 5'h18: I2C TX; 5'h19: I2C RX; 5'h1A: QSPI 5'h20: SPI5 RX ; 5'h21: SPI5 TX; - reserved
7:6	reserved		
5:0	dma_ch0_if	W/R	DMA hardware handshaking interface select for channel 0: 5'h0: DCMI; 5'h1: LCD; 5'h2: UART0 TX; 5'h3: UART0 RX; 5'h4: UART1 TX; 5'h5: UART1 RX; 5 'h6: DAC;

			5'ha: SPI0 TX; 5'hb: SPI0 RX; 5'hc: SPI1 TX; 5'hd: SPI1 RX; 5'he: SPI2 TX; 5'hf: SPI2 RX; 5'h14: UART2 TX; 5'h15: UART2 RX; 5'h16: UART3 TX; 5'h17: UART3 RX; 5'h18: I2C TX; 5'h19: I2C RX; 5'h1A: QSPI 5'h20: SPI5 RX; 5'h21: SPI5 TX;

3.5.13 SCI0 Glitch Filtering Configuration (SCI0_GLF)

Offset address: 0x003C Reset value: 0x20060000

31 30 card_norm	29	28	27	26	25	twenty four
al_glf_byp ass	card_norm al_bypass	reserved				
rw	rw	ro				

twenty three	twenty two	twenty one	20	19	18 17 card_normal_glf	16
reserved						
ro					rw	

15	14	13	12	11	10	9	8
card_normal_glf							
rw							

7	6	5	4	3	2	1	0
card_normal_glf							
rw							

Bit	Name	W/R	Description
31	card_normal_glf_byp ass	RW	card_normal signal (0: overcurrent, 1: normal) glitch filtering bypass 1: Turn off the glitch filtering of card_normal 0: Turn on glitch filtering
30	card_normal_bypass	RW	card_noraml does not participate in the generation of the card detected signal. card_noraml is fed back by the analog, indicating that the 7816 card voltage is normal
29:20	reserved	-	reserved
19:0	card_normal_glf	W/R	card_normal glitch filtering maximum count, the counter is counting at Glitches that appear before the maximum value will be filtered out

3.5.14 2.5V LDO Control Register (LDO25_CR)

Offset address: 0x004C Reset value: 0x00000028

31	30	29	28	27	26	25	twenty four
reserved							
ro							

twenty three	twenty two	twenty one	20	19	18	17	16

reserved							
ro							
15	14	13	12	11	10	9	8
reserved							
ro							
7	6	5	4	3	2	1	0
reserved		otp_pd_en	ldo25_pd_en	reserved			
ro		rw	rw	ro			

Bit	Name	W/R	Description
31:6	reserved	RO reserved bit	
5	otp_pd_en	RW	OTP power off enable 0: OTP power on; 1: OTP power off Remarks: After turning on the power, you need to ensure that it can be accessed for more than 2us
4	ldo25_pd_en	RW	LDO25 power-off enable in low power mode Note: If this enable is turned on, the corresponding USB, OTP will also power down Note: After turning on the power, you need to ensure that it is more than 10us to access the USB
3:0	reserved	RO reserved bit, cannot be modified	

3.5.15 DMA channel 4~7 selection (DMA_CHAN1)

Offset address: 0x0050 Reset value: 0x0A0B0203

31	30	29	28	27	26	25	twenty four
reserved		dma_ch7_if					
ro		rw					
twenty three	twenty two	twenty one	20	19	18	17	16
reserved		dma_ch6_if					
ro		rw					
15	14	13	12	11 10 dma_ch5_if		9	8
reserved							
ro		rw					
7	6	5	4	3	2	1	0
reserved		dma_ch4_if					
ro		rw					

Bit	Name	W/R	Description
31:30	reserved	RO reservation	
29:24	dma_ch7_if	W/R	DMA hardware handshaking interface select for channel 7: 5'h0: DCMI; 5'h1: LCD; 5'h2: UART0 TX; 5'h3: UART0 RX; 5'h4: UART1 TX; 5'h5: UART1 RX; 5'h6: DAC; 5'ha: SPI0 TX; 5'hb: SPI0 RX;

			5'hc: SPI1 TX; 5'hd: SPI1 RX; 5'he: SPI2 TX; 5'hf: SPI2 RX; 5'h14: UART2 TX; 5'h15: UART2 RX; 5'h16: UART3 TX; 5 'h17: UART3 RX; 5'h18: I2C TX; 5'h19: I2C RX; 5'h1A: QSPI 5'h20: SPI5 RX; 5'h21: SPI5 TX;
23:22	reserved	RO reservation	
21:16	dma_ch6_if	W/R	DMA hardware handshaking interface select for channel 6: 5'h0: DCMI; 5'h1: LCD; 5'h2: UART0 TX; 5'h3: UART0 RX; 5'h4: UART1 TX; 5'h5: UART1 RX; 5 'h6: DAC; 5'ha: SPI0 TX; 5'hb: SPI0 RX; 5'hc: SPI1 TX; 5'hd: SPI1 RX; 5'he: SPI2 TX; 5'hf: SPI2 RX; 5'h14 : UART2 TX; 5'h15: UART2 RX; 5'h16: UART3 TX; 5'h17: UART3 RX; 5'h18: I2C TX; 5'h19: I2C RX; 5'h1A: QSPI 5'h20: SPI5 RX ; 5'h21: SPI5 TX;
15:14	reserved	RO reservation	
13:8	dma_ch5_if	W/R	DMA hardware handshaking interface select for channel 5: 5'h0: DCMI; 5'h1: LCD; 5'h2: UART0 TX; 5'h3: UART0 RX; 5'h4: UART1 TX; 5'h5: UART1 RX; 5 'h6: DAC; 5'ha: SPI0 TX; 5'hb: SPI0 RX; 5'hc: SPI1 TX; 5'hd: SPI1 RX; 5'he: SPI2 TX; 5'hf: SPI2 RX; 5'h14 : UART2 TX; 5'h15: UART2 RX; 5'h16: UART3 TX; 5'h17: UART3 RX;

			5'h18: I2C TX; 5'h19: I2C RX; 5'h1A: QSPI 5'h20: SPI5 RX; 5'h21: SPI5 TX;
7:6	reserved	RO reservation	
5:0	dma_ch4_if	W/R	DMA hardware handshaking interface select for channel 4: 5'h0: DCMI; 5'h1: LCD; 5'h2: UART0 TX; 5'h3: UART0 RX; 5'h4: UART1 TX; 5'h5: UART1 RX; 5'h6: DAC; 5'ha: SPI0 TX; 5'hb: SPI0 RX; 5'hc: SPI1 TX; 5'hd: SPI1 RX; 5'he: SPI2 TX; 5'hf: SPI2 RX; 5'h14: UART2 TX; 5'h15: UART2 RX; 5'h16: UART3 TX; 5'h17: UART3 RX; 5'h18: I2C TX; 5'h19: I2C RX; 5'h1A: QSPI 5'h20: SPI5 RX; 5'h21: SPI5 TX;

3.5.16 USB Control Register (USBPHY_CR1)

Offset address: 0x0108 Reset value: 0x 204921AE

31	30	29	28	27	26	25	twenty four
reserved		stop_ck_for_suspend	reserved				
ro		rw	ro				

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
reserved							
ro							

Bit	Name	W/R	Description
31:30	reserved	- reserved bit	
29	stop_ck_for_suspend	RW	Turn off clk_core enable after USB enters suspend mode 0: not closed; 1: closed
28:0	reserved	- reserved bits	

3.5.17 7816 Control Register (ISO7816_CR1)

Offset address: 0x0114 Reset value: 0x00000080

31 30	29	28	27	26	25	twenty four
reserved						
ro						

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
7816_vsel		reserved					
rw		ro					

Bit	Name	W/R	Description
31:8	reserved	- reserved bits	
7:6	7816_vsel	RW	Card Supply Voltage Selection 00: reserved 01: Reserved 10:3V 11:1.8V
5:0	reserved	- reserved bits	

3.5.18 Charge Status Register (CHG_CTRL)

Offset address: 0x011C Reset value: 0x00000000

31 30	29	28	27	26	25	twenty four
reserved	chg_state		reserved			
ro	ro		ro			

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
reserved							
ro							

Bit	Name	W/R	Description
31:30	reserved	- reserved bits	
29:28	chg_state	RO	charging: 11: full 10: Constant current charging 01: trickle charging 00: undervoltage
27:0	reserved	- reserved bits	

3.5.19 Calibration Control Register (CALIB_CSR)

Offset address: 0x03F0 31 Reset value: 0xAB000080

30	29	28	27	26	25	twenty four
reserved						
ro						

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
32k_cal_sel	reserved				osc12m_usb_cal_en	osc12m_cal_done	osc12m_cal_en
rw	ro				rw	rc	rw

7	6	5	4	3	2	1	0
reserved							
ro							

Bit	Name	W/R -	Description
31:16		reserved bit	
15		RW Calibration reference source selection, set to 1 to select XTAL32K	
14:11	reserved32k_cal_sel reserved	- reserved bits	
10	osc12m_usb_cal_en RW Use USB to calibrate 12M enable, automatically clear to 0 after calibration is completed		
9	osc12m_cal_done	RC 12M Calibration Completion Mark	
8	osc12m_cal_en	RW 12M calibration enable, automatically clear to 0 after calibration is completed	
7:0	reserved - reserved bits		

4 General Purpose Input Output (GPIO)

4.1 GPIO function description

Each group of GPIOs on the chip contains 4 32-bit registers and 1 group of 5-bit registers:

ÿ Data Register (Px_IODR)

ÿ Set/Reset Register (Px_BSRR)

ÿ Direction register (Px_OEN)

ÿ Pull-up resistor enable register (Px_PUE).

The upper 16 bits of Px_IODR are used as input registers (Px_IDR) (read only), and the lower 16 bits are used as output registers (Px_ODR) (read Write)

The upper 16 bits of Px_BSRR are used as reset registers, and the lower 16 bits are used as set registers

Each pin of a GPIO port can be configured to work in several ways:

ÿ Input mode (input high impedance, input pull-up)

ÿ Open-drain output

ÿ Push-pull output

GPIO working mode configuration diagram:

Working mode	Px_IODR	Px_BSRR do	Px_OEN 1	Px_PUE
input High-	data_r	not use do		0
impedance input	data_r	not use do	1	1
Pull-up Open-	data_r	not use	data_w 0	0
drain output Push-pull output	data_rw	data_w		0

Note:

data_r: data read operation

data_w: data write operation

data_rw: data read and write operations

4.1.1 General Purpose I/O (GPIO)

When configured as an output, the value written to the output data register will be output to the corresponding I/O. The input data register is displayed on the APB Capture data on I/O.

There is an internal pull-up on all GPIO pins, which can be controlled by the pull-up enable register (Px_PUE).

The default state of all general IO after reset is pull-up input, the resistance value is 51Kÿ

4.1.2 Dedicated I/O (GPIO)

The three I/Os PA8, PA9, and PA10 are the I/Os used by the SCI0 module. If they are used as ordinary I/Os, the power of the 7816 module must be turned on. normal work. PA10 needs to open the internal pull-up resistor to maintain the high level, and needs to close the internal pull-up resistor to maintain the low level. Whether to open the internal pull-up resistor is controlled by the pull-up enable register (Px_PUE).

Note: PA6 is a common I/O and is not powered by the 7816 module.

4.1.3 Individual Bit Set or Clear

The set/reset operation of individual I/Os can be achieved through the set/reset register (Px_BSRR).

4.1.4 External Interrupts

The GPIO is configured to generate an interrupt request to the CPU for state changes on the pin.

GPIO external interrupt response type:

ÿ Rising edge interrupt

ÿ Falling edge interrupt

ÿ Double edge interrupt

4.1.5 External wake-up events

All GPIO pins of the chip support ultra-low power wake-up. GPIO only supports low-level wake-up. The IO wake-up source can be configured through the WKUP_P0_EN~ WKUP_P2_EN registers.

4.1.6 I/O function multiplexing

Peripheral and I/O multiplexing can be configured through the multiplexing control register (Px_ALT).

ÿ I/O multiplexing as needed

ÿ After multiplexing as functional peripherals, there is no need to configure the I/O working mode (input high-impedance, input pull-up, open-drain output, push-pull output),

After the multiplexing configuration is completed, the system will automatically enter the corresponding I/O working mode.

4.2 GPIO registers

4.2.1 Address Mapping Table

GPIO base address table

address range	base address	Peripherals	bus
0x4001_D000-0x4001_DFFF	0x4001_D000	GPIO	APB0

Table 4- 1 GPIO register table

offset address	register name	Width (bit)	Reset value	Notes
0x00	PA_IODR	32	0xFFFF0000	GPIOA
0x04	PA_BSRR	32	0x00000000	
0x08	PA_OEN	32	0x0000FFFF	
0x0C	PA_PUE	32	0x0000FFFF	
0x10	PB_IODR	32	0xFFFF0000	GPIOB
0x14	PB_BSRR	32	0x00000000	
0x18	PB_OEN	32	0x0000FFFF	
0x1C	PB_PUE	32	0x0000FFFF	
0x20	PC_IODR	32	0xFFFF0000	GPIOC
0x24	PC_BSRR	32	0x00000000	
0x28	PC_OEN	32	0x0000FFFF	
0x2C	PC_PUE	32	0x0000FFFF	
0x30	PD_IODR	32	0xFFFF0000	GPIOD
0x34	PD_BSRR	32	0x00000000	
0x38	PD_OEN	32	0x0000FFFF	
0x3C	PD_PUE	32	0x0000FFFF	
0x40	PE_IODR	32	0xFFFF0000	GPIOE
0x44	PE_BSRR	32	0x00000000	
0x48	PE_OEN	32	0x0000FFFF	
0x4C	PE_PUE	32	0x0000FFFF	
0x50	PF_IODR	32	0xFFFF0000	GPIOF

0x54	PF_BSRR	32	0x00000000	
0x58	PF_OEN	32	0x0000FFFF	
0x5C	PF_PUE	32	0x0000FFFF	
0x60-0x110	RSVD	32	0x00000000	reserved
0x114	INTP5_STA	32	0x00000000	
0x118	INTP4_STA	32	0x00000000	
0x11C	INTP3_STA	32	0x00000000	
0x120	INTP2_STA	32	0x00000000	
0x124	INTP1_STA	32	0x00000000	
0x128	INTP0_STA	32	0x00000000	
0x12C - 0x17C RSVD		32	0x00000000	reserved
0x180	PA_ALT	32	0x55555555	
0x184	PB_ALT	32	0x55555555	
0x188	PC_ALT	32	0x55555555	ALT
0x18C	PD_ALT	32	0x55555555	
0x190	PE_ALT	32	0x55555555	
0x194	PF_ALT	32	0x55555555	
0x198 - 0x1FC RSVD		32	0x00000000	reserved
0x200	SYS_CR1	32	0xFFFF0000	
0x204-0x21C	RSVD	32	0x00000000	reserved
0x220	WKUP_TYPE_EN	32	0x00007801	
0x224	WKUP_P0_EN	32	0x00000000	WKUP
0x228	WKUP_P1_EN	32	0x00000000	
0x22C	WKUP_P2_EN	32	0x00000000	
0x230-0x7FC	RSVD	32	0x00000000	reserved
0x800 0x804	PA_INTP_TYPE	32	0x00000000	
0x808 0x80C	PA_INTP_STA	32	0x00000000	
0x810 0x814	PB_INTP_TYPE	32	0x00000000	
0x818 0x81C	PB_INTP_STA	32	0x00000000	
0x820 0x824	PC_INTP_TYPE	32	0x00000000	
0x828 0x82C	PC_INTP_STA	32	0x00000000	
0x830-0x83C	PD_INTP_TYPE	32	0x00000000	INTP
	PD_INTP_STA	32	0x00000000	
	PE_INTP_TYPE	32	0x00000000	
	PE_INTP_STA	32	0x00000000	
	PF_INTP_TYPE	32	0x00000000	
	PF_INTP_STA	32	0x00000000	
	RSVD	32	0x00000000	reserved

4.2.2 Data Register (Px_IODR) (x= A..F)

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

Bit	Name	W/R	Description
31:16	Px_IDR	R	GPIO x input data register
15:0	Px_ODR	W/R	GPIO x Output Data Register

4.2.3 Set/Reset Register (Px_BSRR) (x= A..F)

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

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

Bit	Name	W/R	Description
31:16	Px_BR	WO	GPIOx corresponds to the bit reset register, when writing 1 to the corresponding bit When the corresponding bit of Px_ODR is cleared
15:0	Px_BS	WO	GPIOx corresponds to the bit set register, when writing 1 to the corresponding bit When the corresponding bit of Px_ODR is set to 1

4.2.4 Direction Register (Px_OEN) (x= A..F)

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

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

Bit	Name	W/R	Description
31:16	reserved	-	
15:0	Px_OEN	W/R	GPIOx corresponds to the bit output enable register: 1: input; 0: output.

4.2.5 Pull-Up Enable Register (Px_PUE) (x= A..F)

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

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

Bit	Name	W/R	Description
31:16	reserved	-	
15:0	Px_PUE	W/R	GPIOx corresponding bit pull-up enable

4.2.6 Interrupt Status Register (INTPx_STA) (x= A..F)

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

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

Bit	Name	W/R	Description
31:1	reserved	-	
0	intx_state	RC	Interrupt x Status Register: 0: No interrupt is generated; 1: Interrupt is generated

4.2.7 GPIOx multiplexing control register (Px_ALT) (x=A..F)

31 30 29 28 27 26 25 24 23 22 21 20 19						18	17 16
px15_alt 15	px14_alt 13	px13_alt 11	px12_alt 9	px11_alt 7	px10_alt 5	px9_alt 3	px8_alt
14	12	10	8	6	4	2	1 0
px7_alt	px6_alt	px5_alt	px4_alt	px3_alt	px2_alt	px1_alt	px0_alt

Bit	Name	W/R	Description
31:30	px15_alt	W/R	Px[15] Multiplexing Control Register
29:28	px14_alt	W/R	Px[14] Multiplexing Control Register
27:26	px13_alt	W/R	Px[13] Multiplexing Control Register
25:24	px12_alt	W/R	Px[12] Multiplexing Control Register
23:22	px11_alt	W/R	Px[11] Multiplexing Control Register
21:20	px10_alt	W/R	Px[10] Multiplexing Control Register
19:18	px9_alt	W/R	Px[9] Multiplexing Control Register
17:16	px8_alt	W/R	Px[8] Multiplexing Control Register
15:14	px7_alt	W/R	Px[7] Multiplexing Control Register
13:12	px6_alt	W/R	Px[6] Multiplexing Control Register
11:10	px5_alt	W/R	Px[5] Multiplexing Control Register
9:8	px4_alt	W/R	Px[4] Multiplexing Control Register
7:6	px3_alt	W/R	Px[3] Multiplexing Control Register
5:4	px2_alt	W/R	Px[2] Multiplexing Control Register
3:2	px1_alt	W/R	Px[1] Multiplexing Control Register
1:0	px0_alt	W/R	Px[0] Multiplexing Control Register

4.2.8 Ultra-low power wake-up type control register (WKUP_TYPE_EN)

31	30	29	28	27	26	25	twenty four
reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
15	14	13	12	11	10	9	8
Reserve	sensor_wk_en	Reserve	rtc_wk_en	kbd_wk_en	reserved		
7	6	5	4	3	2	1	0
reserved							gpio_wkup_type

Bit	Name	W/R -	Description
31:15		reserved bit	
14	reserved	RW	Sensor interrupt wake-up enable
13	sensor_wk_en	RW	reserved bit
12	reserved	RW	RTC interrupt wake-up enable
11	rtc_wk_en	RW	KBD interrupt wake-up enable
10:1	kbd_wk_en reserved	-	reserved
0	gpio_wkup_type	RW	Wake-up source GPIO type and enable control bit: 0: wake up directly; 1: wake up after filtering 1 32K clock cycle glitch

4.2.9 Ultra-low power wake-up source enable 0 (WKUP_P0_EN)

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

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

Bit	Name	W/R	Description
31:16	pb_wk_en	W/R	PB ultra-low power wakeup pb_wk_en[0] corresponds to PB[0], pb_wk_en[1] corresponds to PB[1]
15:0	pa_wk_en	W/R	PA ultra-low power wake-up pa_wk_en[0] corresponds to PA[0], and pa_wk_en[1] corresponds to PA[1]

4.2.10 Ultra-low power wake-up source enable 1 (WKUP_P1_EN)

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

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

Bit	Name	W/R	Description
31:16	pd_wk_en	W/R	PD ultra-low power wake-up pd_wk_en[0] corresponds to PD[0], pd_wk_en[1] corresponds to PD[1]
15:0	pc_wk_en	W/R	PC ultra-low power wakeup pc_wk_en[0] corresponds to PC[0], pc_wk_en[1] corresponds to PC[1]

4.2.11 Ultra-low power wake-up source enable 2 (WKUP_P2_EN)

31	30 29 28 27 26 25 24 23	twenty two	21 20 19 pf_wk_en	18	17	16

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

Bit	Name	W/R	Description
31:24	reserved	- reserved bit	
23:16	pf_wk_en	W/R	PF ultra-low power wake-up pf_wk_en[0] corresponds to PF[0], pf_wk_en[1] corresponds to PF[1]
15:0	pe_wk_en	W/R	PE ultra-low power wake-up pe_wk_en[0] corresponds to PE[0], pe_wk_en[1] corresponds to PE[1]

4.2.12 Interrupt Type Control Register (Px_INTP_TYPE) (x=A..F)

31 30 29 28 27 26 25 24 23 22 21 20 19						18	17	16
px15_int_type	px14_int_type	px13_int_type	px12_int_type	px11_int_type	px10_int_type	Px9_int_type	Px8_int_type	

15	14	13	12	11 10	9 8	7 6	5	4	2	1	0
Px7_int_type	Px6_int_type	Px5_int_type	Px4_int_type	Px3_int_type	Px2_int_type	3 px1_int_type	px0_int_type				

Bit	Name	W/R	Description

31:30	px15_int_type	W/R	Px15 interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
29:28	px14_int_type	W/R	Px[14] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
27:26	px13_int_type	W/R	Px[13] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
25:24	px12_int_type	W/R	Px[12] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
23:22	px11_int_type	W/R	Px[11] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
21:20	px10_int_type	W/R	Px[10] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
19:18	px9_int_type	W/R	Px[9] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
17:16	px8_int_type	W/R	Px[8] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
15:14	px7_int_type	W/R	Px[7] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
13:12	px6_int_type	W/R	Px[6] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
11:10	px5_int_type	W/R	Px[5] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
9:8	px4_int_type	W/R	Px[4] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
7:6	px3_int_type	W/R	Px[3] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
5:4	px2_int_type	W/R	Px[2] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
3:2	px1_int_type	W/R	Px[1] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt
1:0	px0_int_type	W/R	Px[0] Interrupt type and enable control bit: 00: No interrupt; 01: Rising edge interrupt; 10 : falling edge interrupt; 11 : double edge interrupt

4.2.13 Interrupt Status Register (Px_INTP_STA) (x=A..F)

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

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

Bit	Name	W/R	Description
31:16	reserved	-	
15:0	px_int_state	WC	GPIOx interrupt status register, a bit of 1 represents the corresponding pin An interrupt is generated, otherwise there is no interrupt. After the interrupt flag is set, the Hold until the corresponding bit is written 1, the interrupt flag is cleared

5 CRC calculation unit (CRC)

5.1 Introduction to CRC

The Cyclic Redundancy Check Calculation Unit (CRC) obtains any 16/32-bit CRC calculation result according to the selected production polynomial. In other applications, CRC technology is mainly used to verify the correctness and integrity of data transmission or data storage.

5.2 Main Features of CRC

ÿ Support CRC-16/32 two CRC calculation methods

ÿ Support CRC-16 polynomial: 0x8005 and 0x1021

ÿ Support CRC-32 polynomial: 0x04C11DB7

ÿ Data is entered in bytes

ÿ Input data can be configured to be reversed by hardware

ÿ The output data can be configured to be reversed by hardware

ÿ Support to specify the initial value of CRC calculation

The block diagram of the CRC calculation unit is as follows:

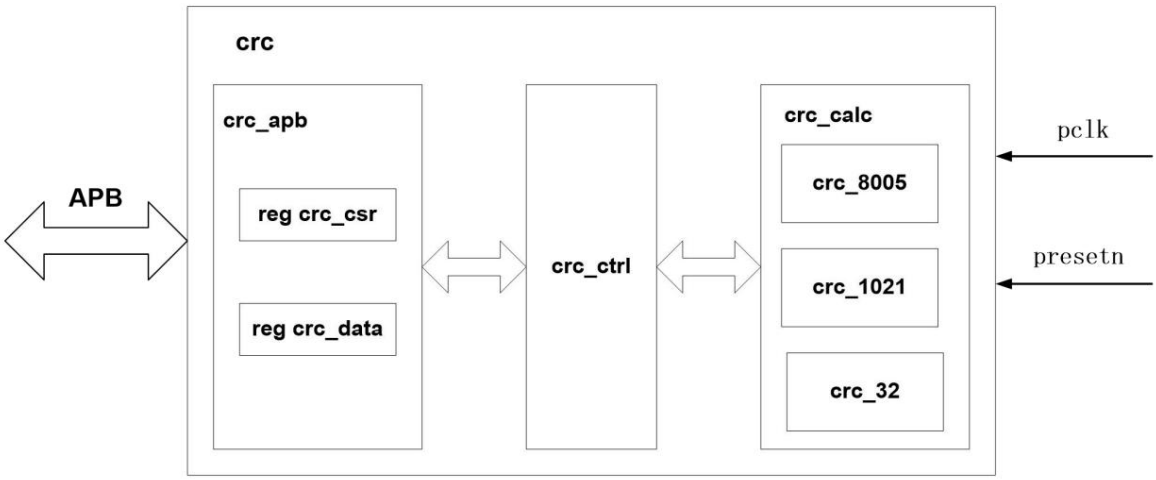


Figure 5- 1 CRC unit block diagram

5.3 CRC function description

The CRC unit contains a 32-bit data input register

ÿ When writing to it, it is used as an 8-bit input register (only the lower 8 bits are valid)

ÿ When it is read, it is used as an output register, and its effective bit width is controlled by the CRC control status register (CRC_CSR) Decide.

After configuring the control status register (CRC_CSR) and the CRC calculation initial value register (CRC_INI) as needed, The data register is continuously written. After the write operation of the data that needs to be verified is completed, the CRC data register is read and obtained. CRC check value.

The flow chart of CRC operation is as follows:

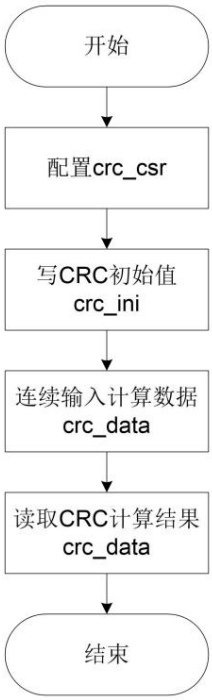


Figure 5- 2 CRC operation flow chart

5.4 CRC register

5.4.1 Address Mapping Table

CRC peripheral base address table

Address	base	Peripherals	bus
range 0x4001_2000-0x4001_2FFF	address 0x4001_2000	CRC	APB0

Table 5- 1 CRC register table

Offset address register	name	Width (bit)	Reset value
0x00	CRC_CSR	32	0x00000000
0x04	CRC_INI	32	0x00000000
0x08	CRC_DATA	32	0x00000000

5.4.2 Control Status Register (CRC_CSR)

31	30	29	28	27	26	25	twenty four
reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
15	14	13	12	11	10	9	8
reserved							
7	6	5	4	3	2	1	0
reserved			xor_out_sel	rev_out_sel	rev_in_sel	type_sel	poly_sel

Bit	Name	W/R	Description
31:5	reserved	-	
4	xor_out_sel	W/R	XOR the CRC calculation result with 0xffff; (this step happens after rev_out_sel)

			1: XOR with 0xffff; 0: The calculation result is output directly.
3	rev_out_sel	W/R	The high and low bits of the CRC calculation result are reversed; 1: reverse; 0: No inversion.
2	rev_in_sel	W/R	CRC 8-bit input size is reversed for calculation, such as bit7 as bit0 participates in the operation, bit6 is used as bit1, and so on. 1: reverse; 0: No inversion.
1	type_sel	W/R	CRC type selection; 1: CRC32; 0: CRC16.
0	poly_sel	W/R	Polynomial selection of CRC16, when CRC32 is selected, this bit is invalid. 1: 0x1021; 0: 0x8005.

5.4.3 Initial value register (CRC_INI)

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

Bit	Name	W/R	Description
31:0	crc_ini	W/R	initial value for CRC calculation; When calculating CRC16, the lower 16 bits are valid.

5.4.4 Data Register (CRC_DATA)

CRC_DATA_IN input (write operation)

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

Bit	Name	W/R	Description
31:8	-	-	reserved bits
7:0	reserved din	W	CRC data input, calculated in units of 1byte.

CRC_DATA_OUT output (read operation)

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

Bit	Name	W/R	Description
31:0	dout	R	CRC calculation result output; When CRC16 is used, only bits [15:0] are valid.

6 True Random Number Generator (TRNG)

6.1 Introduction to TRNG

The TRNG unit is used to generate a sequence of true random numbers.

6.2 Main Features of TRNG

ÿ One job generates 128-bit true random number sequence;

ÿ Configurable to generate CPU interrupt request after random number generation;

The block diagram of the TRNG computing unit is as follows:

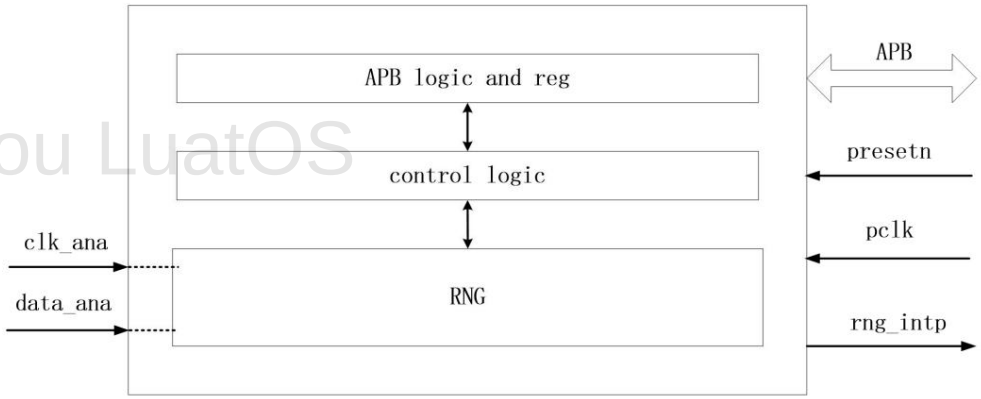


Figure 6- 1 Block diagram of TRNG unit

6.3 TRNG function description

1 Configuration control status register (TRNG_CSR) and analog control register (TRNG_ANA)

2 Clear TRNG_CSR[s128] to "0", and the TRNG unit starts to generate random numbers

3 Polling TRNG_CSR[s128], the hardware is set to "1" to indicate that the random number generation is completed (in interrupt mode, the random number is generated after the random number is generated. interrupt).

4 Continuously read TRNG_DATA 4 times to obtain 128bit random number

5 Loop ÿ~ÿ to get more random numbers

6.4 TRNG register

6.4.1 Address Mapping Table

TRNG peripheral base address table

address range	Base	Peripherals	bus
0x4001_E000-0x4001_EFFF	address 0x4001_E000	TRNG	APB0

Table 6- 1 TRNG register table

Offset address register name 0x00		Width (bit) Reset value	
	TRNG_CSR 0x04	32	0x00000020
	TRNG_DATA 0x0C	32	0x00000000
	TRNG_AMA 0x10	32	0x000FF486
	TRNG_PN 0x14	32	0x69D84C18
	RNG_INDEX	32	0x00000000

6.4.2 Control Status Register (RNG_CSR)

31	30	29	28	27	26	25	twenty four
Reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			intp_en	Reserve	rng0_attack	Reserve	rng0_s128

Bit	Name	W/R -	Description
31:5	reserved	reserved bit	
4	intp_en	W/R	0: No interrupt is generated; 1: Generate an interrupt.
3	reserved	- reserved space	
2	rng0_attack	W/R	RNG0 1: 47 consecutive 0s or 1s detected, there is an attack 0: no attack
1	reserved	- reserved space	
0	rng0_s128	W/R	RNG0, 128-bit random number status bit, set to 1 by hardware and cleared by software to 0, clear After 0, a new 128-bit random number generation starts automatically: 0: 128-bit random number is not generated; 1: A 128-bit random number has been generated.

6.4.3 Data Register (RNG_DATA)

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

Bit	Name	W/R	Description
31:0	data	R	32-bit random number, it needs to be read after s128 is set to 1, 4 times in a row Read this register to get a 128-bit random number.

6.4.4 Analog Control Register (RNG_AMA)

31	30	29	28	27	26	25	twenty four
reserved			ana_out_en	20	reserved		
twenty three	twenty two	twenty one	19	18	17	16	
reserved							
15	14	13	12	11	10	9	8
pd_trng				reserved			
7	6	5	4	3	2	1	0
reserved							

Bit	Name	W/R	Description
31:29	reserved	RO reservation	
28	ana_out_en	W/R	RNG0 1: Direct analog output

			0: output after data post-processing
27:16		-	reserved
15:12	reserved	W/R	random source PD signal
11:0	pd_trng reserved	-	reserved

6.4.5 Pseudo-Random Sequence Register (RNG_PN)

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

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

Bit	Name	W/R	Description
31:0	PN	RW	32-bit pseudo-random sequence, updated once per system cycle. The initial value of this register is configurable.

6.4.6 RNG FIFO Index (RNG_INDEX)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1 0
Reserved														rng0_index

Bit	Name	W/R	Description
31	fifo_rd_ov	WC	FIFO read overflow flag, when the newly generated 128bit random number is read After that, read the fifo again, this bit is set to 1, and the write operation is cleared to 0
30:2		-	reserved space
1:0	Reserve rng0_index	RO	RNG0 FIFO read depth

7 CACHE Module (CACHE)

7.1 Introduction to CACHE

Cache is used to improve the efficiency of the processor fetching instructions from low-speed memory, and is an intermediate medium between the two. The principle is based on the procedural bureau
The principle of partiality is to cache some instructions or data through small-capacity and fast memory to reduce the processor's impact on slow large-capacity memory.
access times, thereby improving processor efficiency.

7.2 CACHE function description

The CPU fetches instructions from the Cache through CodeBus, and the Cache reads data from the Flash Controller through the AHB Master. Cache
The maximum space for accessing Flash is 16MB.

7.3 CACHE register description

7.3.1 Address Mapping Table

Register Base Address Table

Address	base	Peripherals	bus
range 0x4008_0000-0x4008_FFFF	address 0x4008_0000	CACHE_CTRL	AHB

Table 7- 1 CACHE register table

Offset address register name		Width (bit) Reset value	
0x00	CACHE_I0	32	0x00000000
0x04	CACHE_I1	32	0x00000000
0x08	CACHE_I2	32	0x00000000
0x0C	CACHE_I3	32	0x00000000
0x10	CACHE_K0	32	0x00000000
0x14	CACHE_K1	32	0x00000000
0x18	CACHE_K2	32	0x00000000
0x1C	CACHE_K3	32	0x00000000
0x20	CACHE_CS	32	0x00000000
0x24	CACHE_REF	32	0x00000000
0x28-0x3C Reserved		32	0x00000000
0x40	CACHE_CONFIG	32	0x5A5A0000
0x44-0x70 Reserved	0x74	32	0x00000000
	CACHE_SADDR 0x78	32	0x00000000
	CACHE_EADDR	32	0x00000000

7.3.2 Initial vector register (CACHE_Ix) (x=0…3)

ÿ **Name:** CACHE_Ix
ÿ **Size:** 32 bits
ÿ **Address Offset:**
for x = 0, 0x00
for x = 1, 0x04
for x = 2, 0x08
for x = 3, 0x0C

ÿ **Read/write access:** write only

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

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

Bit	Name	W/R	Description
31:0	ivx	WO	Initial vector register x (x=0…3), key_gen must be A5 This register is writable when

7.3.3 Key Register (CACHE_Kx) (x=0…3)

ÿ **Name:** CACHE_Kx
ÿ **Size:** 32 bits
ÿ **Address Offset:**

for x = 0, 0x10
for x = 1, 0x14
for x = 2, 0x18
for x = 3, 0x1C

ÿ **Read/write access:** write only

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

Bit	Name	W/R	Description
31:0	keyx	WO	Key register x (x=0…3), when key_gen must be A5 This register is writable

7.3.4 Control Register (CACHE_CS)

ÿ **Name:** CACHE_CS
ÿ **Size:** 32 bits
ÿ **Address Offset: 0x20**
ÿ **Read/write access:** read/write

31	30	29	28	27	26	25	twenty four
key_gen_start	key_gen_err	cache_busy					

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							

15	14	13	12	11	10	9	8
reserved							

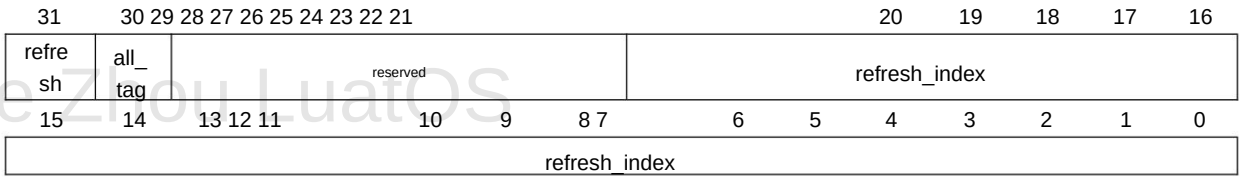
7	6	5	4	3	2	1	0
key_gen							

Bit	Name	W/R	Description
31	key_gen_start	RW	After it is set to 1, the round key generation is started, and the software is set to 1 to complete the round key generation. After completion, the hardware is cleared to 0 (key_gen must be A5, and the cache is not resolved This bit can only be set to 1 by hardware.
30	key_gen_err	RW	0: No key generation error 1: The key generation cannot be performed, which is caused by the following two situations: 1) key_gen is not equal to A5, 2) the cache is decrypting the data , start key generation

			Set to 1 by hardware, cleared by software
29	cache_busy	RO	1: cache is fetching from Flash 0: cache does not fetch instructions from Flash
28:8	reserved	RO Reserved bit, read as 0.	
7:0	key_gen	RW	Equal to A5: key generation mode, KEY/IV can be written Not equal to A5: decryption mode, KEY/IV cannot be written

7.3.5 CACHE Refresh Control Register (CACHE_REF)

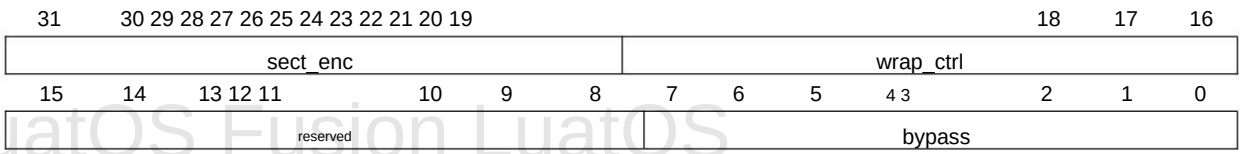
- **Name:** CACHE_REF
- **Size:** 32 bits
- **Address Offset: 0x24**
- **Read/write access:** read/write



Bit	Name	W/R	Description
31	refresh	RW	After setting 1, refresh refresh_index to specify the Cache TAG, and the refresh is complete Automatically clear to 0 after
30	all_tag	RW	0: Refresh only according to refresh_index 1: Refresh all TAGs Set by software and cleared by hardware
29:24	rsvd	RO Reserved bit, read as 0.	
23:0	refresh_index	RW refresh address, corresponding to 23:0-bit of CODE BUS address	

7.3.6 CACHE Configuration Register (CACHE_CONFIG)

- **Name:** CACHE_CONFIG
- **Size:** 32 bits
- **Address Offset: 0x40**
- **Read/write access:** read/write



Bit	Name	W/R	Description
31:17	sect_enc	RW	When it is equal to A5, and bypass is closed, the solution of some areas is opened. secret function When it is not equal to A5, and bypass is closed, the decryption of the whole area is enabled. Function
23:16	wrap_ctrl	RW	When equal to A5, read AHB WRAP of QSPI Controller turn on When not equal to A5, read AHB WRAP of QSPI Controller take off
15:8	reserved	- reserved bits	
7:0	bypass	RW	Equal to A5: bypass Not equal to A5: decryption mode

7.3.7 Area Decryption Start Address Register (CACHE _SADDR)

- **Name:** CACHE_SADDR
- **Size:** 32 bits
- **Address Offset:** 0x74
- **Read/write access:** read/write

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

Bit	Name	W/R	Description
31:0	saddr	RW	Area decryption start address, when saddr<read address<eaddr, correct Flash reads data for decryption

7.3.8 Area Decryption End Address Register (CACHE _EADDR)

- **Name:** CACHE_EADDR
- **Size:** 32 bits
- **Address Offset:** 0x78
- **Read/write access:** read/write

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

Bit	Name	W/R	Description
31:0	eaddr	RW	Area decryption end address, when saddr<read address<eaddr, right Flash reads data for decryption

8 OTP control module (OTP_CTRL)

8.1 Introduction to OTP

OTP is a special memory with a single write operation. When the OTP leaves the factory, the internal data is initialized and the bits are all "1". The write operation can only write the internal bit from "1" to "0", but cannot change from "0" to "1".

OTP operable address range is 0x40009400~0x40009FFF, a total of 3KB

8.2 OTP function description

8.2.1 OTP read-only lock

OTP provides zone write protection and zone write protection lock functions.

Area write protection:

When the OTP area write protection bit is "0", the corresponding area can be programmed/erased. When it is "1", the corresponding area can only be read and locked. The write protection lock bit of the OTP area is "0" ", the corresponding area write protection bit can be modified; when it is "1", the corresponding area write protection bit remains in the existing state and cannot be modified.

8.2.2 OTP programming operation protection

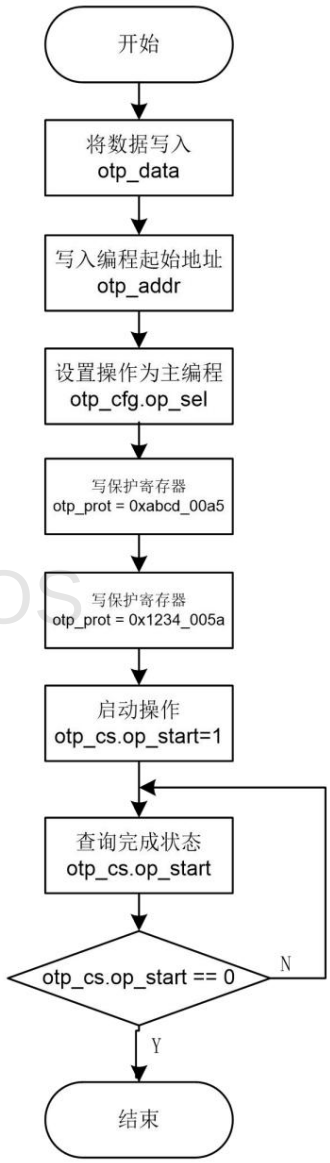
In order to prevent the misoperation of the OTP by the user program, when the OTP starts the program/erase operation, it needs to perform a fixed register operation, and then start the program/erase operation enable. Before programming/erase operations, it is necessary to perform 2 consecutive write operations to OTP_PROT, the first write: 0xABCD00A5, and the second write: 0x1234005A. After the two write operations are completed, the program/erase operation enable is performed immediately. If there are other FCU operations during the period, the start program/erase operation enable is regarded as invalid.

The operation process is as follows:

OTP_PROT = 0xABCD00A5;

OTP_PROT = 0x1234005A; After completing the above 2 register operations, start the program/erase operation enable

8.2.3 OTP programming operation



8.3 OTP_CTRL register

8.3.1 Address Mapping Table

Register Base Address Table

Address	base	Peripherals	bus
range 0x4000_8000-0x4000_BFFF	address 0x4000_8000	OTP_CTRL	AHB

Table 8- 1OTP_CTRL register table

Offset	Register name width (bit)		reset value
address 0x0000-0x1FFF reserved		32	0x00000000
0x2000	OTP_CFG	32	0x00000008
0x2004	OTP_CS	32	0x80000000
0x2008	OTP_PROT	32	0x00000000
0x200C	OTP_ADDR	32	0x00000000
0x2010	OTP_PDATA	32	0x00000000
0x2014	OTP_RO	32	0x00000000
0x2018	OTP_ROL	32	0x00000000
0x201C	RSVD	32	0x00000000

0x2020	OTP_TIM	32	0x00000000
0x2024	OTP_TIM_EN	32	0x00000000

8.3.2 OTP Configuration Register (OTP_CFG)

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

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

Bit	Name	W/R	Description
31:2	reserved	-	reserved bit
1:0	op_sel	RW	00: Programming 01: Sleep 10/11: Wake Up Writable only when op_start is 0.

8.3.3 OTP Control Status Register (OTP_CS)

31	30 29 28 27 26 25 24 23	twenty two	21 20 19	18	17	16
rd_ready	reserved					

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

Bit	Name	W/R	Description
31	rd_ready	RO	1: OTP read operation is available 0: OTP is in programming/sleep state and cannot be read
30:4	reserved	-	Reserved bit, read as 0.
3:1	illegal	RW	When the operation is completed, it is set by hardware and cleared by software: 000: There is no abnormality in this operation; 001: Read OTP in program/sleep state 010: Program the read-only area 011: The programming range exceeds the OTP address range 100: Program operation in sleep state 101: wake up in non-sleep state
0	op_start	RW	Start the operation of OTP, set the software to 1 and clear the hardware to 0, and the software can pass This bit queries whether the operation is complete.

8.3.4 OTP Boot Protection Register (OTP_PROT)

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

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

Bit	Name	W/R	Description
31:0	prot	WC	Enable protection control bit.

			To start OTP programming, 3 consecutive write operations are required, followed by The sequence is as follows: write prot 0xabcd_00a5, write prot 0x1234_005a, Set op_start to 1.

8.3.5 OTP Program Erase Address Register (OTP_ADDR)

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

Bit	Name	W/R	Description
31:0	op_addr	RW	programming address

8.3.6 OTP programming data register (OTP_PDATA)

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

Bit	Name	W/R	Description
31:0	pdata	W/R	Used to store 32bit data programming data. Writable only when op_start is 0. After starting hibernation, this register will be cleared to **0**

8.3.7 OTP Main Memory Region Read-Only Region Register (OTP_RO)

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

Bit	Name	W/R	Description
31:0	ro	W/R	Divide the entire OTP main memory area into 32 areas according to 256B, each Each area is controlled by 1bit of ro. 0: Programmable/eraseable. 1: Read only.

8.3.8 OTP Main Memory Read-Only Lock Register (OTP_ROL)

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

Bit	Name	W/R	Description
31:0	ro_lock	W/R	Used to lock the configuration of the OTP_RO register, once locked, The corresponding bit of otp_ro cannot be modified, and the lock is released after reset.

			0: Not locked. 1: Locked, after setting 1, software cannot clear 0, only hardware after reset Clear to 0. ro_lock[0] corresponds to ro[0], ro_lock[1] corresponds to ro[1], and so on push.

8.3.9 OTP Timing Register (OTP_TIM)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved					cycles_10ns			cycles_1us							

Bit	Name	W/R	Description
31:11	rsvd	RO	Reserved bit, read as 0.
10:8	cycles_10ns	RW	Determine the clock cycle used for 10ns, such as 60M, the cycle is 16.67ns, write 1 to this register. Writable only when op_start is 0.
7:0	cycles_1us	RW	Determine the clock cycle used for 1us, such as 60M, the cycle is 16.67ns, minus 1ns as a margin for insurance, so it should be sent The register is configured as 1000ns/15ns. If there is a remainder, add 1 to get 67. Writable only when op_start is 0.

8.3.10 OTP Timing Enable Register (OTP_TIM_EN)

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

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

Bit	Name	W/R	Description
31:8	reserved	-	Reserved bit, read as 0.
7:0	tim_en	W/R	When this register is equal to A5, use otp_tim as timing reference, Otherwise automatically generated by the system

9 Keyboard Control Unit (KCU)

9.1 Introduction to KCU

The keyboard control unit is used for scanning and identifying the keys of the keyboard, and has the security feature of anti-electromagnetic attack.

Schematic diagram of the module:

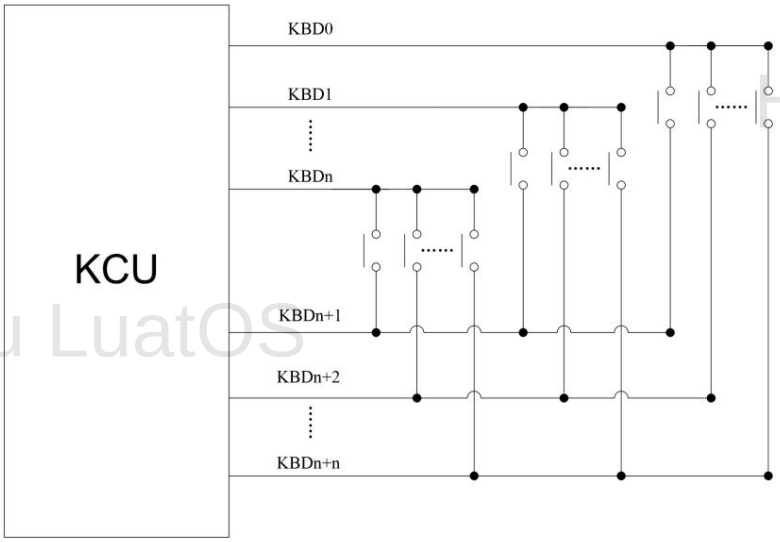


Figure 9- 1 Schematic diagram of KCU module

9.2 KCU Features

ÿ Can provide a maximum array of 4x5, 20 keys;

ÿ Configurable key debounce (Debounce) time;

ÿ 4 key cache registers;

ÿ Button press (Push) and button release (release) detection;

ÿ Anti-electromagnetic attack random keyboard scanning;

ÿ The port of the keyboard array has a built-in pull-up in the chip IO;

ÿ Multiple keys are not supported.

9.3 Functional Description

Configuration process:

ÿ Configure keyboard GPIO pin pull-up resistor enable

ÿ KCU enable disabled: KCU_CTRL1[0] = 0

ÿ Wait for KCU to be inactive: while (KCU_CTRL1[31])

ÿ Configure keyboard input/output and key debounce time: KCU_CTRL0

ÿ Configure keyboard interrupt: KCU_CTRL1

ÿ KCU enable on: KCU_CTRL1[0] = 1

9.4 KCU register

9.4.1 Address Mapping Table

address range	base address	Peripherals	bus

0x4004_8000-0x4004_8FFF	0x4004_8000	KCU	APB3

Table 9- 1 KCU register table

address	register name	Width (bit)	Reset value
0x00	KCU_CTRL0	32	0x00000600
0x04	KCU_CTRL1	32	0x00000002
0x08	KCU_STATUS	32	0x00000000
0x0C	KCU_EVENT	32	0x00000000
0x10	KCU_RNG	32	0x00000000

9.4.2 Control Register 0 (KCU_CTRL0)

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

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

Bit	Name	W/R	Description
31:12	reserved	-	
11:9	range	W/R	Determine the random range of the Debounce Time of the button; 000: 5ms~75ms; 001: 17.5ms~75ms; 010: 27.5ms~75ms; 011: 37.5ms~75ms; 100: 45ms~75ms; 101: 50ms~75ms; 110: 60ms~75ms; 111: 70ms~75ms. After selecting the range, such as 45ms~75ms, the actual debounce time The maximum possible yield is 75x2 = 150ms. Note: This value can only be modified when kcu_runing is 0.
8:0	out_en	W/R	Bit 0 corresponds to port 0, bit 1 corresponds to port 1, and so on. 0: The port is input; 1: The port is output. Note: This value can only be modified when kcu_runing is 0.

9.4.3 Control Register 1 (KCU_CTRL1)

31	30	29	28	27	26	25	twenty four
kcu_runing 23	Reserved						

twenty two	twenty one	20	19	18	17	16
reserve						

15	14	13	12	11	10	9	8
reserve							

7	6	5	4	3	2	1	0
reserve				overrun_ie	release_ie	push_ie	kbd_en

Bit	Name	W/R	Description
31	kcu_runing	W/R	KCU working status: 1: KCU is scanning the keyboard;

			0: The KCU has stopped scanning.
30:4	reserve	-	Reserved bit, read as 0.
3	overrun_ie	W/R	0: 4 key_event registers are full, and there is a key event, no generate an interrupt; 1: When the 4 key_event registers are full and there is a key event, the Interrupted. Note: This value can only be modified when kcu_runing is 0.
2	release_ie	W/R	0: When the button is released, the release event will not be stored key_event register, and no interrupt is generated; 1: When the button is released, the release event is stored in key_event register and generate an interrupt. Note: This value can only be modified when kcu_runing is 0.
1	push_ie	W/R	0: When the key is pressed, the push event will not be stored in key_event register, and no interrupt is generated; 1: When the key is pressed, the push event is stored in the key_event register device, and an interrupt is generated. Note: This value can only be modified when kcu_runing is 0.
0	kbd_en	W/R	0: The keyboard stops scanning; 1: Start keyboard scan mode.

9.4.4 Status Register (KCU_STATUS)

31	30	29	28	27	26	25	twenty four
Reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
reserve							
15	14	13	12	11	10	9	8
reserve				new3	push3	new2	push2
			4		2 1 push_is overrun_is		0
7 new1	6 push1	5 new0	push0	3 release_is			is

Bit	Name	W/R	Description
31:12	reserved	-	
11	new3	R	0: The keys stored in push3, out3, and in3 are old events. Already read; 1: The keys stored in push3, out3, in3 are new events, the software Not read. This bit is cleared to 0 after reading kcu_event;
10	push3	R	0: release event; 1: push event. This bit is cleared to 0 after reading kcu_event;
9	new2	R	0: The keys stored in push2, out2, and in2 are old events. Already read; 1: The keys stored in push2, out2, and in2 are new events. Not read. This bit is cleared to 0 after reading kcu_event;
8	push2	R	0: release event; 1: push event.

			This bit is cleared to 0 after reading kcu_event;
7	new1	R	0: The keys stored in push1, out1, and in1 are old events. Already read; 1: The keys stored in push1, out1, and in1 are new events, and the software Not read. This bit is cleared to 0 after reading kcu_event;
6	push1	R	0: release event; 1: push event. This bit is cleared to 0 after reading kcu_event;
5	new0	R	0: The keys stored in push0, out0, and in0 are old events. Already read; 1: The keys stored in push0, out0, and in0 are new events. Not read. This bit is cleared to 0 after reading kcu_event;
4	push0	R	0: release event; 1: push event. This bit is cleared to 0 after reading kcu_event;
3	release_is	R	0: No release interrupt is generated; 1: A release interrupt is generated; This bit is cleared to 0 after reading kcu_event; When release_intp_en is 0, this bit is always 0.
2	push_is	R	0: No push interrupt is generated; 1: A push interrupt is generated; This bit is cleared to 0 after reading kcu_event; When push_intp_en is 0, this bit is always 0.
1	overrun_is	R	0: No overrun interrupt is generated; 1: There is an overrun interrupt; This bit is cleared after reading kcu_status; When overrun_intp_en is 0, this bit is always 0.
0	is	R	0: No interrupt is generated; 1: An interrupt is generated.

9.4.5 KCU button cache register (KCU_EVENT)

31 30 29 28 27 26 25 24 23 22 21 20 19			18 17 16
out3	in3	out2	in2
15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
out1	in1	out0	in0

Bit	Name	W/R	Description
31:28	out3	RC	Event 3, on which output pin the key occurs, the value corresponds to the corresponding The pin number is automatically cleared to 0 after reading.
27:24	in3	RC	Event 3, on which input pin the key occurs, the value corresponds to the corresponding The pin number is automatically cleared to 0 after reading.
23:20	out2	RC	Event 2, on which output pin the key occurs, the value corresponds to the corresponding The pin number is automatically cleared to 0 after reading.
19:16	in2	RC	Event 2, on which input pin the key occurs, the value corresponds to the corresponding The pin number is automatically cleared to 0 after reading.
15:12	out1	RC	Event 1, which output pin the key occurs on, the value corresponds to the corresponding The pin number is automatically cleared to 0 after reading.

11:8	in1	RC	Event 1, on which input pin the key occurs, the value corresponds to the corresponding The pin number is automatically cleared to 0 after reading.
7:4	out0	RC	Event 0, which output pin the key occurs on, the value corresponds to the corresponding The pin number is automatically cleared to 0 after reading.
3:0	in0	RC	Event 0, on which input pin the key occurs, the value corresponds to the corresponding The pin number is automatically cleared to 0 after reading.

9.4.6 KCU PN initialization register (KCU_RNG)

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

Bit	Name	W/R	Description
31:0	rng_ini	RW	KCU random value initialization register

10 Real Time Clock (RTC)

10.1 Introduction to RTC

The real-time clock is an independent timer. The RTC module has a set of counters that count continuously. RTC module and RTC related configuration The setting registers are all in the battery power domain, that is, the main power failure has no effect on the RTC, and the RTC still maintains normal counting.

10.2 RTC Features

ÿ Use seconds as the timing unit (second interrupt is generated by configuration)

ÿ CPU independent interrupt source

ÿ 32-bit alarm setting register, used to generate interrupt signal or wake up CPU at set time

ÿ RTC real-time clock unit, support alarm interrupt generation

ÿ 32K clock can be output from chip pins

10.3 RTC registers

10.3.1 Address Mapping Table

RTC base address table

address range	Base	Peripherals	bus
0x4003_0000 - 0x4003_00B8	address 0x4003_00A0	RTC	APB2

Table 10- 1 RTC register table

Offset address register	name 0x00A0	Width (bit) Reset value	
	RTC_CS	32 0x00000008	
0x00A4	RTC_REF	32 0x00000000	
0x00A8	RTC_ARM	32 0x0000FFFF	
0x00AC	RTC_TIM	32	0x00000000
0x00B0	RTC_INTCLR	32	0x00000000
0x00B4	OSC32K_CR	32	0x00000060
0x00B8	RTC_ATTA_TIM	32	0x00000000

10.3.2 RTC Control Status Register (RTC_CS)

Offset address: 0x00A0 Reset value: 0x00000008

31	30	29	28	27	26	25	twenty four
reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
15	14	13	12	11	10	9	8
reserved							
7	6	5	4	3	2	1	0
reserved			rtc_clr	rtc_rdy	rtc_ien	time_rd_lock	intp_rd_bit

Bit	Name	W/R	Description
31:5	reserved	RO reserved	
4	rtc_clr	W/R	0: RTC counts normally, the current count can be read from the rtc_tim register value;

			1: The RTC counter is cleared to 0, that is, the rtc_tim register is cleared to 0. The software is set to 1 and the hardware is cleared to 0, that is, the RTC is in the counting state after power-on.
3	rtc_rdy	RO	Before each "battery power domain power-on" software operates on the RTC, it needs to be query rtc_rdy status, 0: Indicates that the RTC is being reset and cannot be operated on the RTC; 1: Indicates that the RTC reset is over, and the RTC can be read and written.
2	rtc_ien	W/R	RTC interrupt enable. 0: disable interrupt; 1: Enable interrupt.
1	time_rd_lock	W/R	The RTC current value reads the lock bit, when the CPU wants to read the current RTC When the count value of the To 0 this bit.
0	intp_rd_bit	RO	RTC interrupt flag, this bit can be set to 1 only after the interrupt is enabled 0: no interruption; 1: There is an interrupt.

10.3.3 RTC count initial value register (RTC_REF)

Offset address: 0x00A4 Reset value: 0x00000000

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

Bit	Name	W/R	Description
31:0	rtc_ref	W/R	The RTC timing initial value register is only used to save the timing initial value. Not involved in timing. Actual timing value=rtc_ref+rtc_tim;

10.3.4 RTC Alarm Setting Register (RTC_ARM)

Offset address: 0x00A8 Reset value: 0x0000FFFF

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

Bit	Name	W/R	Description
31:0	rtc_arm	W/R	RTC alarm clock setting register, generated when rtc_tim=rtc_arm break;

10.3.5 RTC current count value register (RTC_TIM)

Offset address: 0x00AC Reset value: 0x00000000

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

Bit	Name	W/R	Description

31:0	rtc_tim	RO	The current count value of the RTC 32bit counter. Before reading this register rtc_cs.time_rd_lock needs to be set to 1, and needs to be set to 0 after reading. Actual timing value=rtc_ref+rtc_tim.

10.3.6 RTC Interrupt Clear Register (RTC_INTCLR)

Offset address: 0x00B0 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:1		RO	reserved bit
0	reserved rtc_intclr	WC	A write to this register clears the interrupt.

10.3.7 32K Clock Calibration Control Register (OSC32K_CR)

Offset address: 0x00B4 Reset value: 0x00000060

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

15 14 13 12 11 10 9 8	Reserved	6	7	osc32k_cal_done	5	4	3	1	0	2
		osc32k_cal_en								osc32k_cal_word

Bit	Name	W/R	Description
31:8	reserved	RO	
7	osc32k_cal_done	RC	32KHz Oscillator Calibration Completion Indicator
6	osc32k_cal_en	RW	32KHz oscillator calibration enable, when calibration is completed, this bit automatically clear;
5:0	osc32k_cal_word	RW	32KHz oscillator calibration control word;

10.3.8 RTC attack time record register (RTC_ATTA_TIM)

Offset address: 0x00B8 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:0	rtc_atta_tim	RO	When the attack occurs, this register records the time of the RTC at that time

11 Watchdog (WDT)

11.1 Watchdog Peripheral Clock

The watchdog peripheral clock is provided by PCLK, that is, the watchdog peripheral clock frequency is equal to the PLCK clock frequency.

11.2 Counter (Counter)

The watchdog counter (DWT_CCVR: Watchdog Timer Current Counter Value Register) is a down counter, that is, the counter value is decremented from the preset value until the value is 0. When the counter counts to 0, the watchdog generates a system reset or interrupt according to the set mode. The watchdog operating in the interrupt mode will generate a watchdog interrupt when the watchdog counter reaches 0 for the first time, and reset the watchdog counter to the preset value, but will not generate a system reset. After that, the watchdog counter will enter the next round of countdown, and the user must feed the dog or clear the interrupt during this counting process, otherwise the system will reset after the count reaches 0.

The user can reset the counter by resetting the register (WDT_CRR: Counter Restart) at any time before the reset occurs. Register) to write 0x76 to reset the counter to the default value. Complete the "Feed the dog" action.

11.3 Counter Preset Values (Timeout Period Values)

The watchdog timer preset value is saved by WDT_RLD (Watchdog Timer Reload Value Register), and the user can set the watchdog timer timeout time through this register. Writing 0x76 to the WDT_CRR register will reset the setting of the DWT_CCVR register to this default value to complete the "feed the dog" operation.

11.4 Enable Watchdog (WatchDog Enable)

The watchdog is turned on by the watchdog control register (WDT_CR: Watchdog Timer Control Register). When WDT_CR[0] = 1, the watchdog is turned on. Once the watchdog enable is turned on, it cannot be turned off.

11.5 System Resets/Interrupts (System Resets)

The watchdog includes 2 modes:

WDT_CR[1] = 0: System reset mode WDT_CR[1] = 1: Interrupt mode System reset mode: After the watchdog counter counts to 0, the system resets immediately. Interrupt mode: When the watchdog counter counts to 0 for the first time, a watchdog interrupt will be generated (the interrupt source is the non-maskable interrupt NMI), and the watchdog counter will be reset to the preset value, but a system reset will not be generated . After that, the watchdog counter will enter the next round of down counting. The user must feed the dog or clear the interrupt during this counting process, otherwise the system will reset after the count reaches 0.

The watchdog running in the interrupt mode, in addition to using the normal dog feeding method (resetting the watchdog counter), can also complete the dog feeding by clearing the watchdog interrupt flag. The watchdog interrupt can be cleared in the following two ways: 1. Reset the watchdog counter (feed the dog)

After writing 0x76 to the WDT_CRR register, the hardware automatically loads the value of the WDT_RLD register into the DWT_CCVR register to complete the "feeding the dog" operation. 2. Read the watchdog interrupt clear register (WDT_EOI) Read the WDT_EOI register to clear the watchdog interrupt flag.

11.6 Register Description

11.6.1 Address Mapping Table

WDT base address list

address range	Base	Peripherals	bus
0x4001_C000-0x4001_CFFF	address 0x4001_C000	WDT	APB0

Table 11- 1 WDT register table

Offset address register name 0x00		Width (bit) Reset value	
	WDT_CR	32 0x00000000	
0x04	reserved	32	0x00000000
0x08	WDT_CCVR	32	0x0000FFFF
0x0C	WDT_CCR	32	0x00000000
0x10	WDT_STAT	32	0x00000000
0x14	WDT_EOI	32	0x00000000
0x18	reserved	32	0x00000000
0x1C	WDT_RLD	32	0xFFFFFFFF

11.6.2 Watchdog Control Register (WDT_CR)

ÿ **Name:** Control Register ÿ
Size: 32bits
ÿ **Address Offset:** 0x00
ÿ **Read/write access:** read/write

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

Bit	Name	R/W	Description
31:2	reserved	- A read operation returns 0	
1	RMOD	R/W	Response mode Select Watchdog Timeout Acknowledge Mode 0 = Generate system reset 1 = A system interrupt is generated on the first watchdog timeout, and a system interrupt is generated on the second watchdog timeout. If the interrupt is not cleared before death, a system reset occurs. Reset value: 0x00
0	WDT_EN	R/W	WDT enable Watchdog enable operation bit. After the watchdog is enabled, it cannot be closed, and the system Reset does not affect watchdog enable. 0 = Watchdog is off 1 = Watchdog is on Reset value: 0x00

11.6.3 Watchdog Counter (WDT_CCVR)

ÿ **Name:** Current Counter Value Register

ÿ **Size: 32bits**
ÿ **Address Offset:** 0x08
ÿ **Read/write access:** read

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

Bit	Name	R/W	Description
31:0	WDT_CCVR	R	For the read operation of this register, the read value is the count value corresponding to the read time. Reset value: 0xFFFF

11.6.4 Watchdog Counter Reset Register (WDT_CRR)

ÿ **Name:** Counter Restart Register ÿ
Size: 32 bits
ÿ **Address Offset:** 0x0c
ÿ **Read/write access:** write

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

Bit	Name	R/W	Description
31:8	reserved	- read operations return 0	
7:0	WDT_CRR W		This register is used to reset the watchdog counter value. To prevent accidental operation, write The input value must be 0x76. The return value of this register is 0 Reset value: 0x00

11.6.5 Watchdog Interrupt Status Register (WDT_STAT)

ÿ **Name:** Interrupt Status Register ÿ
Size: 32 bit ÿ **Address Offset:** 0x10

ÿ **Read/write access:** read

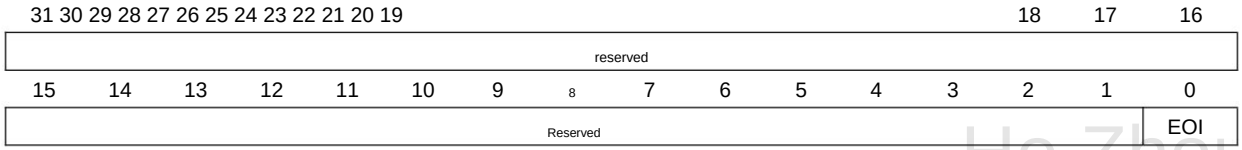
31 30 29 28 27 26 25 24 23 22 21 20 19	18	17	16
reserved			
15 14 13 12 11 10 9 8 7 6 5 4 3 2	1		0
reserved			STAT

Bit	Name	R/W	Description
31:1	reserved	- read operations return 0	
0	STAT	R	Interrupt Status This bit is used to indicate the interrupt status of the watchdog 1 = Watchdog interrupt generated 0 = Watchdog interrupt is not generated Reset value: 0x00

11.6.6 Watchdog Interrupt Clear Register (WDT_EOI)

ÿ **Name:** Interrupt Clear Register ÿ **Size:**
32 bit ÿ **Address Offset:** 0x14

ÿ **Read/write access:** read



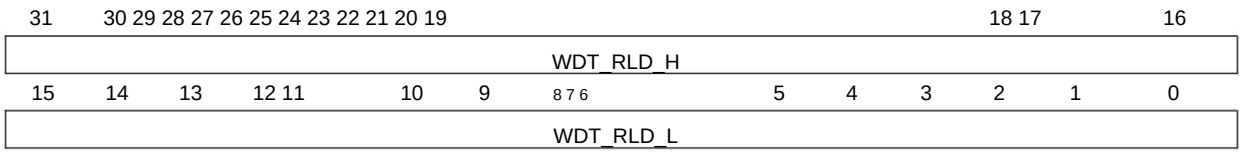
Bit	Name	R/W	Description
31:1	reserved	-	A read operation returns 0
0	EOI	R	Clears the watchdog interrupt Clear the watchdog interrupt and reset the watchdog counter (WDT_CCVR). Reset value: 0x00

11.6.7 Watchdog Preset Value Register (WDT_RLD)

ÿ **Name:** Reload Value Register ÿ
Size: *32bits*

ÿ **Address Offset:** 0x1C

ÿ **Read/write access:** read/write



Bit	Name	R/W	Description
31:0	WDT_RLD	R/W	Stores the watchdog counter preset value Reset value: 0xFFFFFFFF

12 timer (TIMER)

12.1 Introduction to Timers

ÿ 1 Timer unit, including 8 independent timers (Timer0, Timer1, Timer2, Timer3, Timer4, Timer5, Timer6, Time7).

ÿ The 8 timer interrupt sources are independent, and each timer occupies a separate interrupt source ÿ The timer adopts the down-counting method ÿ Each timer unit timer supports the PWM mode Frequency as timer clock source ÿ PWM single trigger (one shot) function

12.2 Timer peripheral clock

The timer peripheral clock is provided by PCLK, that is, the timer clock frequency is equal to the PCLK peripheral clock frequency

12.3 General Purpose Timer

12.3.1 Two modes of the general-purpose timer

In free-running and user-defined modes, when the timer is enabled, the count value is changed from The TimerNLoadCount register is loaded.

The load value is selected according to the selected mode: User-defined mode:

The timer count value is loaded into the TimerNLoadCount register setting. Using user mode can generate a timer interrupt of a fixed time. Free-runing mode:

The timer count value is loaded with its maximum allowable value, which is 0xFFFFFFFF. The user can reprogram or disable the timer interrupt before the timer generates an interrupt (timer counter counts to 0). Using this mode, the timer generates only one interrupt. After the interrupt is generated, the count value is reset to 0xFFFFFFFF and counts down, but no further interrupt is generated.

12.3.2 Interrupt Handling

After the timer generates an interrupt, the user can clear the timer interrupt status by reading TimerNEOI or TimersEOI.

TimerNEOI: Only clear the interrupt status on the corresponding timer interrupt source.

TimersEOI: Clears the timer interrupt status in this timer unit.

12.4 PWM Mode

The eight independent timers of the Timer unit can be programmed to generate

PWM signals. When the user sets the PWM bit in TimerNControlReg to "1", the timer enters the PWM working mode. At this time PWM By TimerNLoadCount2 and TimerNLoadCount registers respectively control the high-level and low-level cycle inversion output.

12.4.1 PWM operating mode

Set the PWM bit in TimerNControlReg to "1", and the timer will work in PWM mode after it is enabled.

12.4.2 PWM period and duty cycle setting

The PWM signal frequency and duty cycle can be configured in the following ways:

ÿ Width of HIGH period = (TimerNLoadCount2 + 1) * PCLK_ Period
ÿ Width of LOW period = (TimerNLoadCount + 1) * PCLK_ Period

12.5 Register Description

12.5.1 Address Mapping Table

TIMER base address list

Address	base	Peripherals	bus
range 0x4001_3000-0x4001_3FFF	address 0x4001_3000	Timer	APB0

Table 12- 1 TIMER register table

offset address	register name	Width (bit) Reset value	
0x00 0x04	Timer0LoadCount	32	0x00000000
0x08	Timer0CurrentValue	32	0xFFFFFFFF
0x0C	Timer0ControlReg	32	0x00000000
0x10	Timer0EOI	32	0x00000000
0x14	Timer0IntStatus	32	0x00000000
0x18	Timer1LoadCount	32	0x00000060
0x1C	Timer1CurrentValue	32	0xFFFFFFFF
0x20	Timer1ControlReg	32	0x00000000
0x24	Timer1EOI	32	0x00000000
0x28	Timer1IntStatus	32	0x00000000
0x2C	Timer2LoadCount	32	0x00000060
0x30	Timer2CurrentValue	32	0xFFFFFFFF
0x34	Timer2ControlReg	32	0x00000000
0x38	Timer2EOI	32	0x00000000
0x3C	Timer2IntStatus	32	0x00000000
0x40	Timer3LoadCount	32	0x00000060
0x44	Timer3CurrentValue	32	0xFFFFFFFF
0x48	Timer3ControlReg	32	0x00000000
0x4C	Timer3EOI	32	0x00000000
0x50	Timer3IntStatus	32	0x00000000
0x54	Timer4LoadCount	32	0x00000060
0x58	Timer4CurrentValue	32	0xFFFFFFFF
0x5C	Timer4ControlReg	32	0x00000000
0x60	Timer4EOI	32	0x00000000
0x64	Timer4IntStatus	32	0x00000000
0x68	Timer5LoadCount	32	0x00000060
0x6C	Timer5CurrentValue	32	0xFFFFFFFF
0x70	Timer5ControlReg	32	0x00000000
0x74	Timer5EOI	32	0x00000000
0x78	Timer5IntStatus	32	0x00000000
0x7C	Timer6LoadCount	32	0x00000060
0x80	Timer6CurrentValue	32	0xFFFFFFFF
0x84	Timer6ControlReg	32	0x00000000
0x88	Timer6EOI	32	0x00000000
0x8C	Timer6IntStatus	32	0x00000000
0x90	Timer7LoadCount	32	0x00000060
0x94	Timer7CurrentValue	32	0xFFFFFFFF
0x98	Timer7ControlReg	32	0x00000000
	Timer7EOI	32	0x00000000

0x9C	Timer7IntStatus	32	0x00000000
0xA0	TimersIntStatus	32	0x00000000
0xA4	TimersEOI	32	0x00000000
0xA8	TimersRawIntStatus	32	0x00000000
0xAC	Reserved	32	0x00000000
0xB0	Timer0LoadCount2	32	0x00000000
0xB4	Timer1LoadCount2	32	0x00000000
0xB8	Timer2LoadCount2	32	0x00000000
0xBC	Timer3LoadCount2	32	0x00000000
0xC0	Timer4LoadCount2	32	0x00000000
0xC4	Timer5LoadCount2	32	0x00000000
0xC8	Timer6LoadCount2	32	0x00000000
0xCC	Timer7LoadCount2	32	0x00000000

12.5.2 Auto-Reload Counter (TimerNLoadCount) (N=0…7)

ÿ **Name: TimerN Load Count Register**
ÿ **Size:** 32 bits
ÿ **Address Offset:**

for N = 0, 0x00
for N = 1, 0x14
for N = 2, 0x28
for N = 3, 0x3C
for N = 4, 0x50
for N = 5, 0x64
for N = 6, 0x78
for N = 7, 0x8C

ÿ **Read/write access:** read/write

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

Bit	Name	R/W	Description
31:0	TimerN Load Count	R/W	This value is automatically loaded into TimerN to count.

12.5.3 Auto-Reload Counter 2 (TimerNLoadCount2) (N=0…7)

ÿ **Name: TimerN Load Count Register 2**
ÿ **Size:** 32 bits
ÿ **Address Offset:**

for N = 0, 0xB0
for N = 1, 0xB4
for N = 2, 0xB8
for N = 3, 0xBC
for N = 4, 0xC0
for N = 5, 0xC4
for N = 6, 0xC8
for N = 7, 0xCC

ÿ **Read/write access:** read/write

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

TimerN Load Count2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TimerN Load Count2															

Bit	Name	R/W	Description
31:0	TimerN Load Count2	R/W	When the timer operates in PWM mode, this value is automatically loaded into TimerN count in. When this value is loaded into TimerN, during this count the PWM The output remains high.

12.5.4 Current counter value (TimerNCurrentValue) (N=0…7)

ÿ **Name: TimerN Current Value Register**
ÿ **Size:** 32 bits
ÿ **Address Offset:**
for N = 0, 0x04
for N = 1, 0x18
for N = 2, 0x2C
for N = 3, 0x40
for N = 4, 0x54
for N = 5, 0x68
for N = 6, 0x7C
for N = 7, 0x90
ÿ **Read/write access:** read only

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

Bit	Name	R/W	Description
31:0	TimerN CurrentValue	R	TimerN current count value.

12.5.5 Control Register (TimerNControlReg) (N=0…7)

ÿ **Name: TimerN Control Register**
ÿ **Size:** 32 bits
ÿ **Address Offset:**
for N = 0, 0x08
for N = 1, 0x1C
for N = 2, 0x30
for N = 3, 0x44
for N = 4, 0x58
for N = 5, 0x6C
for N = 6, 0x80
for N = 7, 0x94
ÿ **Read/write access:** read/write

31	30	29	28	27	26	25	24	23	22	21	Reserved	20	19	18	17	16
15	14	13	12	11	10	9	8									
reserved																

7	6	5	4	3	2	1	0
reserved		TIM_Reload	PWM-Oneshot	PWM IntMask		Mode	Enable

Bit	Name	R/W	Description
31:6	reserved	- reserved	
5	TIM_Reload	R/W	PWM Oneshot Reload TimerNLoadCount2 register value
4	PWM-Oneshot R/W		PWM Oneshot enable bit: 0-PWM Oneshot mode off 1-PWM Oneshot mode on
3	PWM	R/W	PWM enable bits: 0-PWM mode off 1-PWM mode on
2	IntMask	R/W	Interrupt mask bits: 0 - interrupt on 1 - interrupt mask
1	Mode	R/W	General timer working mode: 0 - Free-runing mode: 1-User-defined mode (user-defined):
0	Enable	R/W	Timer enable bits: 0 - off 1-Open

12.5.6 Interrupt Clear Register (TimerNEOI) (N=0…7)

- **Name: TimerN End-of-Interrupt Register**
- **Size:** 32 bits
- **Address Offset:**
  - for N = 0, 0x0C
  - for N = 1, 0x20
  - for N = 2, 0x34
  - for N = 3, 0x48
  - for N = 4, 0x5c
  - for N = 5, 0x70
  - for N = 6, 0x84
  - for N = 7, 0x98
- **Read/write access:** read only

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

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

Bit	Name	R/W	Description
31:1	reserved	-	
0	TimerNEOI	RC	A read of this bit clears the timer interrupt status. The return value is 0

12.5.7 Interrupt Status Register (TimerNIntStatus) (N=0…7)

- **Name: TimerN Interrupt Status Register**
- **Size:** 32bits
- **Address Offset:**

for N = 0, 0x10
for N = 1, 0x24
for N = 2, 0x38
for N = 3, 0x4c
for N = 4, 0x60
for N = 5, 0x74
for N = 6, 0x88
for N = 7, 0x9c

ÿ **Read/write access:** read only

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

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

Bit	Name	R/W	Description
31:1	reserved	-	
0	TimerNInt Status	R	Timer interrupt flag bit, this bit is "1" when the timer generates an interrupt.

12.5.8 Global Interrupt Clear Register (TimersEOI)

ÿ **Name: Timers End-of-Interrupt Register**
ÿ **Size:** 32 bits
ÿ **Address Offset: 0xA4**
ÿ **Read/write access:** read only

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

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

Bit	Name	R/W	Description
31:1	reserved	-	
0	TimersEOI	RC	A read of this bit clears the timer unit interrupt status

12.5.9 Global Raw Interrupt Status Register (TimersRawIntStatus)

ÿ **Name: Timers Interrupt Status Register**
ÿ **Size: 32** bits
ÿ **Address Offset: 0xa8**
ÿ **Read/write access:** read only

The interrupt flag value of this register is not affected by IntMask in register TimerNControlReg.

The value of the register TimersIntStatus is the value of the register TimersRawIntStatus after IntMask. When TimersIntStatus is valid bit Set to "1", the Timer sends an interrupt request to the CPU.

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

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

reserved	RawIntStatus

Bit	Name	R/W	Description
31:1	reserved	-	
0	RawIntStatus	R	The original interrupt flag bit of the timer unit, this bit is "1" when the timer generates an interrupt.

13 Universal Asynchronous Receiver/Transmitter (UART)

13.1 Introduction to UART

The Universal Asynchronous Receiver Receiver (UART) provides a flexible way to communicate with external devices using the industry standard NRZ asynchronous serial data format Full-duplex data exchange between devices. The UART utilizes a baud rate generator to provide a wide range of baud rate options.

The Universal Asynchronous Receiver/Transmitter (UART) supports one-way communication, duplex communication, and the IrDA (Infrared Data Association) SIR ENDEC specification, as well as modem (CTS/RTS) operation. Used in conjunction with DMA, high-speed data communication can be achieved.

13.2 Serial Infrared Protocol (IrDA 1.0 SIR Protocol)

13.2.1 Introduction of Serial Infrared Protocol

Infrared 1.0 serial infrared mode supports bidirectional data transmission of infrared device data. IrDA 1.0 SIR mode maximum baud rate 115.2K.

The data format is similar to the data format of the standard serial port. Each data byte contains the following

parts: 1. 1 bit is used as the start bit 2, the next 8 bits are the data bit 3, and at least 1 bit is the end bit

The transmission data bits are fixed. Parity bits are not supported, and there is only 1 stop bit in this mode. Using the LCR register to set the number of bits of data and enable the parity bit has no effect.

IrDA pulse definition: 1.

When a pulse signal is generated, it means logic 0;

2. When there is no pulse signal, it means logic 1;

The pulse width of IrDA is only 3/16 of the pulse width of the common serial port; the start bit of byte transmission is represented by 1 pulse. However, on the IrDA receiver, due to the photodiode being excited by infrared light, the phase of the data received by the UART and the actual transmitted data will be reversed, that is, the high level changes to a low level. The level of the transistor must be inverted and converted by the UART peripheral input to the level currently required by the UART.

13.2.2 SIR Mode Enable

UART peripheral IrDA 1.0 SIR mode, enabled by mode control register MCR[6] = 0.

13.2.3 SIR Mode Operation Features

In IrDA SIR mode, data transmission can only be in half-duplex mode. This is mainly due to the specification of the IrDA SIR physical layer that there must be a 10ms delay between transmission and reception. The 10ms delay is software controlled.

13.3 Receive/Transmit FIFO

13.3.1 Receive/Transmit FIFO Introduction

The UART peripheral can be configured to use the FIFO to send and receive data. Each UART peripheral includes 16bytes of independent receive and transmit FIFOs.

13.3.2 Receive/Transmit FIFO

After the user enables the FIFO, the data written by the CPU to the THR register is stored in the transmit FIFO, and the data received by the UART peripheral is stored in the receive FIFO. Users can obtain valid data in FIFO through UART status register (USR) and other related registers

number or FIFO status. You can also configure the UART interrupt to process the data in the FIFO after generating an interrupt request to the CPU under certain conditions.

13.3.3 Receive/transmit FIFO interrupt usage

Trigger threshold

configuration: The user can configure the receive/transmit FIFO data interrupt using the FIFO Control Register (FCR) or its corresponding shadow register Triggered threshold. When the data in the receive/transmit FIFO meets the set threshold, the corresponding enabled FIFO interrupt will be triggered.

Receive FIFO interrupt:

FIFO mode is enabled, ERBFI interrupt is enabled, and the "receive data valid interrupt" is triggered when the amount of data received in the receive FIFO meets the set threshold.

Transmit FIFO interrupt:

FIFO mode is enabled, ETBEI interrupt is enabled, and the programmable THRE interrupt mode is enabled (IER[7] = 1), when the amount of data remaining in the transmit FIFO meets the set threshold, the "transmit register empty interrupt" is triggered . Trigger "Character Timeout Interrupt" when data is not fetched by timeout.

13.3.4 Receive/Transmit FIFO Access Mode

The UART peripheral provides a FIFO access mode, which is used for program debugging. In the FIFO access mode, the CPU can write the receive FIFO and read the transmit FIFO.

Enter access mode:

FAR[0] = 1, FIFO access mode (FIFO Access mode) is enabled and turned on, and the transmit and receive FIFOs are cleared by hardware after the enable is turned on. Transmit FIFO test: In the FIFO access mode, the data written into the transmit FIFO will not be shifted and sent, but will always be reserved in the transmit FIFO. The user can read the data in the FIFO by reading the TFR register to test the correctness of the data. Receive FIFO test: In FIFO access mode, the user writes data to the receive FIFO by writing the RFW (Receive FIFO Write) register, where RFW[9] is used to test the generation of frame errors, and RFW[8] is used to test A verification error occurs. Data can be read back normally by the receive FIFO. Since normal operation is prohibited in FIFO access mode, data must be manually written to the receive FIFO, and no external reception is possible.

13.4 UART peripheral clock

The UART peripheral clock is provided by the internal PCLK, that is, the UART peripheral clock frequency is equal to the PCLK clock frequency.

13.5 Interrupt

After the UART peripheral is generated, the user can obtain the interrupt type through the IIR register.

UART peripherals can generate interrupt types as follows:

ÿ Modem status interrupt ÿ Transmit register empty interrupt

ÿ Receive data valid interrupt ÿ Line status interrupt ÿ

UART busy interrupt ÿ Character timeout interrupt

13.6 Programmable THRE Interrupt

The UART peripheral can be implemented by programming the THRE interrupt mode.

The THRE interrupt mode setting relationship is as follows:

THRE interrupt	Programmable THRE Interrupt Mode Enable (IER[7])	FIFO enable	THRE interrupt enable (IER[1])
No interruption	-	-	×
Generated when the THR register is empty interrupt	-	×	ÿ
Both THR and transmit FIFO are empty interrupt	×	ÿ	ÿ
Transmit FIFO data arrival or low Interrupt when threshold is set	ÿ	ÿ	ÿ

The empty threshold (FCR[5:4]) that can be set in the transmit FIFO is as follows:

- ÿ empty
- ÿ 2chars
- ÿ ¼ Full
- ÿ ½ Full

13.7 DMA Support

Using DMA function for UART peripheral can effectively reduce system interruption and improve data transmission efficiency. Each UART peripheral can Use 2 DMA channels for receiving and sending data respectively.

The UART FIFO must be used when the UART peripheral uses the DMA function, and the data volume of the UART in each receive/transmit FIFO meets the trigger level. Send a request to the DMA after sending the threshold.

The UART FIFO must be enabled when the UART uses DMA, ie (FCR[0] = 1).

1. DMA transmission request of UART peripheral:

ÿ The request signal is valid in the following cases:

a) FIFO enable is on (FCR[0] = 1) and THRE interrupt mode enable is off (IRE[7] = 0), the transmit FIFO is null.

b) FIFO enable is on (FCR[0] = 1) and THRE interrupt mode enable is on (IRE[7] = 1), transmit FIFO number The amount of data reaches or falls below the set threshold.

ÿ The request signal is invalid in the following cases:

FIFO enable is turned on (FCR[0] = 1), the DMA channel continuously writes to the transmit FIFO until the transmit FIFO is full, the request Signal is invalid.

2. DMA reception request of UART peripheral:

ÿ The request signal is valid in the following cases:

a) The FIFO enable is turned on (FCR[0] = 1), and the data in the receive FIFO reaches or exceeds the set threshold.

b) FIFO enable is turned on (FCR[0] = 1), the byte data timeout (character timeout) occurs in the FIFO, no need Enable ERBFI (IER[0] = 1) interrupt.

ÿ The request signal is invalid in the following cases:

FIFO enable is turned on (FCR[0] = 1), the DMA channel continuously reads the receive FIFO until the data in the receive FIFO is lower than the set value. The set threshold, the request signal is invalid.

13.8 Register Description

13.8.1 Address Mapping Table

UARTx (x=0...1) base address list

Address	Base	Peripherals	bus
range 0x4001_6000-0x4001_6FFF	address	UART0	APB0
0x4001_7000-0x4001_7FFF	0x4001_6000	UART1	
0x4004_4000-0x4004_4FFF	0x4001_7000	UART2	APB3
0x4004_5000-0x4004_5FFF	0x4004_4000 0x4004_5000	UART3	

Table 13- 1 UART register table

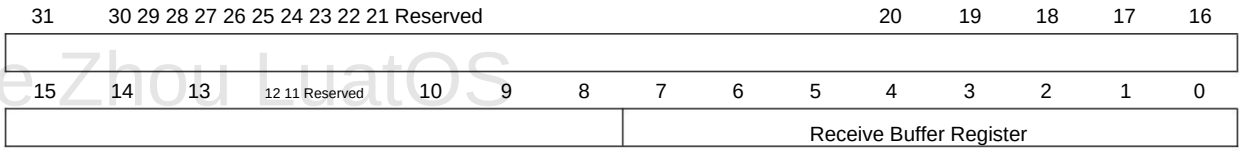
Offset address register name 0x00		Width (bit) Reset value	
	DLL/THR/RBR 0x04	32	0x00000000
	DLH/IER	32	0x00000000
0x08	IIR/FCR	32	IIR = 0x00000001 FCR = 0x00000000
0x0C	LCR	32	0x00000000
0x10	MCR	32	0x00000000
0x14	LSR	32	0x00000060
0x18	reservation	32	0x00000000
0x1C	SCR	32	0x00000000
0x20	reservation	32	0x00000000
0x24	reservation	32	0x00000000
0x28	reservation	32	0x00000000
0x2C	reservation	32	0x00000000
0x30	SRBR/STHR	32	0x00000000
0x34	SRBR/STHR	32	0x00000000
0x38	SRBR/STHR	32	0x00000000
0x3C	SRBR/STHR	32	0x00000000
0x40	SRBR/STHR	32	0x00000000
0x44	SRBR/STHR	32	0x00000000
0x48	SRBR/STHR	32	0x00000000
0x4C	SRBR/STHR	32	0x00000000
0x50	SRBR/STHR	32	0x00000000
0x54	SRBR/STHR	32	0x00000000
0x58	SRBR/STHR	32	0x00000000
0x5C	SRBR/STHR	32	0x00000000
0x60	SRBR/STHR	32	0x00000000
0x64	SRBR/STHR	32	0x00000000
0x68	SRBR/STHR	32	0x00000000
0x6C	SRBR/STHR	32	0x00000000
0x70	FAR	32	0x00000000
0x74	TFR	32	0x00000000
0x78	RFW	32	0x00000000
0x7C	USR	32	0x00000006
0x80	TFL	32	0x00000000
0x84	RFL	32	0x00000000
0x88	SRR	32	0x00000000
0x8C	SRTS	32	0x00000000
0x90	SBCR	32	0x00000000
0x94	SDMAM	32	0x00000000
0x98	SFE	32	0x00000000
0x9C	SRT	32	0x00000000

0xA0	STET	32	0x00000000
0xA4	HTX	32	0x00000000
0xA8	DMASA	32	0x00000000
0xAC~0xFC reserved		32	0x00000000

13.8.2 Receive Buffer Register (RBR)

ÿ **Name:** Receive Buffer Register ÿ **Size:**
32 bits
ÿ **Address Offset:** 0x00
ÿ **Read/write access:** read-only

The RBR register can only be accessed when the DLAB bit is cleared to 0.

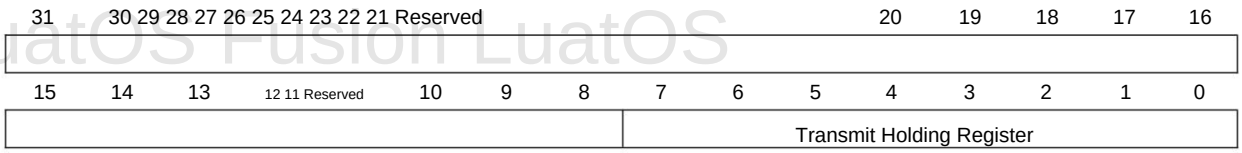


Bit	Name	R/W	Description
7:0	Receive Buffer Register	RW	receive data register The UART peripheral is used to receive data in serial or infrared mode. After the UART peripheral successfully receives data, the DR bit in the LSR register is "1". In non-FIFO mode, the data that is not read in time will be replaced by the data received next time overwrites, and generates an overflow error flag. In FIFO mode, newly received data is stored in the FIFO header. If the FIFO starts with If it is full, subsequent received data will be discarded and an overflow error flag will be generated. Reset value: 0x00

13.8.3 Transmit Holding Register (THR)

ÿ **Name:** Transmit Holding Register ÿ **Size:**
32 bits
ÿ **Address Offset:** 0x00
ÿ **Read/write access:** write-only

The THR register can only be accessed when the DLAB bit is cleared to 0.

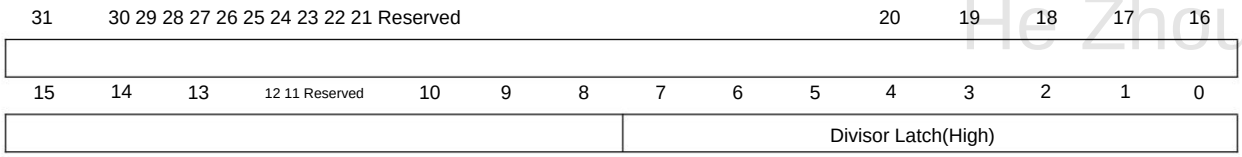


Bit	Name	R/W	Description
7:0	Transmit Holding Register	W	transmit data register The UART peripheral is used to send data in serial or infrared mode. In non-FIFO mode, the data transmission hold bit (THRE) is set to "1", and the transmission data The data register is empty, the data transmission holding bit (THRE) is cleared to "0", and the transmission data The data register is not empty. In FIFO mode, data can be written continuously when the FIFO is not full, Data that continues to be written when the FIFO is full will be lost. Reset value: 0x00

13.8.4 Divide Register_High (DLH)

- **Name:** Divisor Latch High
- **Size:** 32 bits
- **Address Offset:** 0x04
- **Read/write access:** read/write

The DLH register is only available when the DLAB bit in the LCR register is set and the UART is not busy (USRT bit0 is 0).
access.

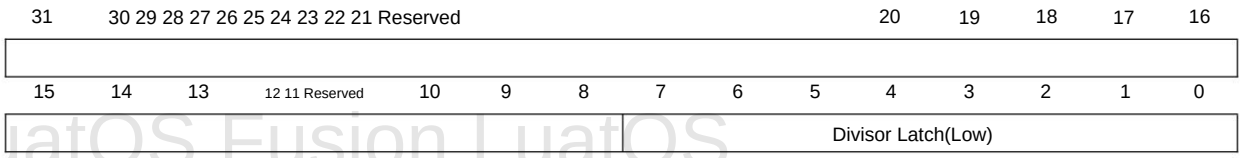


Bit	Name	R/W	Description
7:0	Divisor Latch (High)	R/W	Baud rate divider register high bit. Baud rate=input clock/(16*frequency division baud rate register value), when UART input clock is the PCLK clock period When the value of the baud rate divider register (DLL and DLH) is set to 0, the baud rate The clock is turned off and the serial port cannot communicate. After setting the DLH register, after 8 baud rate clocks, the UART will according to receive or send operation. Reset value: 0x00

13.8.5 Divide Register_Low (DLL)

- **Name:** Divisor Latch Low
- **Size:** 32 bits
- **Address Offset:** 0x00
- **Read/write access:** read/write

The DLL register is only available when the DLAB bit in the LCR register is set and the UART is not busy (USRT bit0 is 0).
access.



Bit	Name	R/W	Description
7:0	Divisor Latch (Low)	R/W	Baud rate divider register low bits. Baud rate=input clock/(16*frequency division baud rate register value), when UART input clock is the PCLK clock period When the value of the baud rate divider register (DLL and DLH) is set to 0, the baud rate The clock is turned off and the serial port cannot communicate. After setting the DLH register, after 8 baud rate clocks, the UART will according to receive or send operation. Reset value: 0x00

13.8.6 Interrupt Enable Register (IER)

ÿ **Name:** Interrupt Enable Register ÿ **Size:**
32 bits
ÿ **Address Offset:** 0x04
ÿ **Read/write access:** read/write

The IER register can only be accessed after the DLAB bit in the LCR register is cleared to 0.

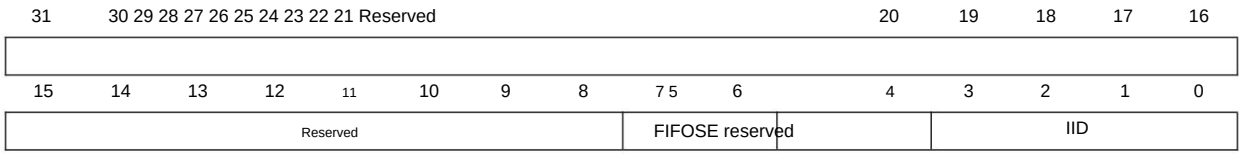
31 30 29 28 27 26 25 24	23 22 21 20 Reserved	19	18	17	16

15 14 13 12	11	10 9 8	7	6 5 4	3	2	1	0
reserved			PTIME	reserved	EDSSI	ELSI	ETBEI	ERBFI

Bit	Name	R/W	Description
7	PTIME	R/W	Programmable THRE Interrupt Mode Enable Used to set the THRE interrupt mode, used to control whether to generate a THRE interrupt 0 = disabled 1 = enabled Reset value: 0x00
6:4	reserved	-	-
3	EDSSI	R/W	Enable Modem Status Interrupt Enable Modem status interrupt to control whether Modem status interrupt is generated. The interrupt priority is 4 (prioritize the corresponding high-priority interrupt). 0 = disabled 1 = enabled Reset value: 0x00
2	ELSI	R/W	Enable Receiver Line Status Interrupt Enable Line state interrupt to control whether Line state interrupt is generated. The interrupt priority level is 1 (prioritize the corresponding high-priority interrupt). 0 = disabled 1 = enabled Reset value: 0x00
1	ETBEI	R/W	Enable Transmit Holding Register Empty Interrupt Enable send register empty interrupt, used to control whether to generate send register empty break. The interrupt priority level is 3 (prioritize the corresponding high-priority interrupt). 0 = disabled 1 = enabled Reset value: 0x00
0	ERBFI	R/W	Enable Received Data Available Interrupt Enable receive data valid interrupt, used to control whether to generate receive data valid interrupt and receive byte timeout interrupt (in FIFO mode or FIFO enabled). The interrupt priority level is 2 (prioritize the corresponding high-priority interrupt). . 0 = disabled 1 = enabled Reset value: 0x00

13.8.7 Interrupt Flag Register (IIR)

ÿ **Name:** Interrupt Identity Register ÿ **Size:**
32 bits
ÿ **Address Offset:** 0x08
ÿ **Read/write access:** read-only



Bit	Name	R/W	Description
7:6	FIFOSE	R	FIFO State Enabled FIFO enable status. 00 = disabled 11 = enabled Reset value: 0x00
4:5	reserved	-	-
3:0	IID	R	Interrupt ID UART interrupt flag, corresponding to the interrupt type (bit representation) 0000 – Modem status interrupted 0001 – no interrupt occurred 0010 – Transmit Register Empty Interrupt 0100 – Receive data valid interrupt 0110 – Line status interrupted 0111 – Busy Interrupt 1100 – Character timeout interrupt (receive FIFO enabled) Reset value: 0x01

interrupt number	Interrupt description			
Binary Indicates Response Priority	Interrupt Type		interrupt source	interrupt clear operation
0001	-	none	none	
0110	1	Line state interrupted	Receive overload error, check error error, framing error, or received break interrupt	Read Line Interrupt Status Register device (LSR)
0100	2	Receive data valid interrupt	ÿIn non-FIFO mode: receive after valid data ÿIn FIFO mode: receive The amount of data in the FIFO reaches the set After setting the threshold	ÿIn non-FIFO mode: read Fetch receive buffer register (RBR) ÿIn FIFO mode: read receive data in the FIFO, make the number in the receive FIFO The amount of data is reduced to the trigger threshold the following
1100	2	Byte timeout interrupt	In FIFO mode, receive There is valid data in the FIFO and After the last valid data received 4 unit time not received data. 1 unit time: 1 byte count time received	Read receive data register (RBR)
0010	3	Transmit register empty interrupt	ÿIRE register PTIME bit Set to 0 (IRE[7] = 0): When the transmit holding register is empty ÿIRE register PTIME bit Set to 1 (IRE[7] = 1): The data in the transmit FIFO is lower than the setting Trigger an interrupt at a certain threshold	If the interrupt source is ÿ, give THR register write data or Turn off send null interrupt enable Write if interrupt source is ÿ Send FIFO until data Above the set trigger threshold or turn off the send null interrupt enable

			(PTIME in the IRE register set 1)	can
0000	4	Modem status interrupt	Modem status register related position 1. If automatic control flow is enabled, then a change in CTS will not cause interrupt	Read Modem Status Register
0111	5	busy interrupt	When the UART is busy, try Try to write to the LCR register operate	Read USR register

13.8.8 FIFO Control Register (FCR)

ÿ **Name:** FIFO Control Register ÿ **Size:** 32 bits
ÿ **Address Offset:** 0x08
ÿ **Read/write access:** write-only

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

15 14 13 12 11 10 9	8	7 6	5 4	3	2	1	0
reserved		RCVR TET		DMAM	XFIFOR	RFIFOR	FIFOE

Bit	Name	R/W	Description
7:6	RCVR	W	Receiver Trigger Receive FIFO full trigger threshold setting. When the data in the receive FIFO exceeds the set threshold, it will trigger the receive data valid interrupt. In automatic flow control mode, this threshold is used to determine the received rts_n signal state (in RTC_FCT is turned off). In DMA mode, the UART sends DMA after the amount of data in the FIFO triggers the threshold ask. The following are the supported threshold types. 00 – 1 character in the FIFO 01 – FIFO ¼ full 10 – FIFO ½ full 11 – FIFO 2 less than full Reset value: 0x00
5:4	TET	W	TX Empty Trigger Transmit FIFO empty threshold setting. When the PTIME enable is turned on (IER[7] = 1), the data in the transmit FIFO is equal to or low At this threshold, the THRE interrupt will be triggered. In DMA mode, the UART sends DMA after the amount of data in the FIFO triggers the threshold ask. The following are the supported threshold types. 00 – FIFO empty 01 – 2 characters in the FIFO 10 – FIFO ¼ full 11 – FIFO ½ full Reset value: 0x00
3	DMAM	W	DMA Mode In case the extra DMA handshake signal is not selected (DMA_EXTRE = NO)

			In this case, it is used to determine the mode of DMA send request and receive request signal. 0 = Do not use DMA 1 = use DMA Reset value: 0x00
2	XFIFOR W		XMIT FIFO Reset Transmit FIFO reset. Resets the transmit FIFO related status information and clears the transmit FIFO. reset will make The DMA TX request signal is invalid. After being set to "1", it is cleared to "0" by hardware. Reset value: 0x00
1	RFIFOR W		RCVR FIFO Reset Receive FIFO reset. Resets the receive FIFO related status information and clears the receive FIFO. reset will make The DMA RX request signal is invalid. After being set to "1", it is cleared to "0" by hardware. Reset value: 0x00
0	FIFO	W	FIFO Enable Enable receive and transmit FIFOs. An enable operation causes the receive and transmit FIFOs to reset. Reset value: 0x00

13.8.9 Line Control Register (LCR)

ÿ **Name:** Line Control Register ÿ
Size: 32 bits
ÿ **Address Offset:** 0x0C
ÿ **Read/write access:** read/write

31	30 29 28 27 26	25 24	23	twenty two	21 20 19	18	17	16
Reserved								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved								DLAB	BC	SP	EPS	PEN	STOP	DLS	

Bit	Name	R/W	Description
31:8	reserved	- read returns 0	
7	DLAB	R/W	Divisor Latch Access Bit DLL and DLH read and write operations enable bits. DLL/DLH is multiplexed with other register addresses, which is used as an operation switch. 0- Operate on the corresponding multiplexed register 1- Operate on DLL/DLH After it is set to "1", it needs to be cleared to "0" by software. Write operation requires USR[0] to be 0, that is, the UART peripheral is not in a busy state. Reset value: 0x00
6	BC	R/W	Break Control Bit Generate output break signal to receiving device, this bit is 1, UART peripheral output The pin will be forced to output a logic 0 signal. In non-loopback mode (MCR[4] = 0), UART mode (MCR[4] = 0), The sout output will be forced low straight. Infrared mode (MCR[4] = 1), sout holds Continuously output pulse signal. After this bit is cleared to 0, the output stops. In loopback mode (MCR[4] = 1), the output break signal is internally looped back looped to the receiver, the sout output is forced low.

			Reset value: 0x00
5	SP	R/W	Stick Parity Force a check digit value When PEN, EPS, SP are set to 1, the parity bit will output logic 0. When PEN and SP are set to 1 and EPS is set to 0, the parity bit will output logic 1. Write operation requires USR[0] to be 0, that is, the UART peripheral is not in a busy state. 0 = disable 1 = enable Reset value: 0x00
4	EPS	R/W	Even Parity Select Select the parity check method. Write operation requires USR[0] to be 0, that is, the UART peripheral is not in a busy state. 0 = Odd 1 = Even Reset value: 0x00
3	PEN	R/W	Parity Enable Parity Check Enable Write operation requires USR[0] to be 0, that is, the UART peripheral is not in a busy state. 0 = disable 1 = enable Reset value: 0x00
2	STOP	R/W	STOP Bits Selection of the number of stop bits. Write operation requires USR[0] to be 0, that is, the UART peripheral is not in a busy state. 0 = 1 stop bit 1 = 1.5 stop bit when LCR[1:0]=0 in DLS, 2 otherwise stop bit Reset value: 0x00
1:0	DLS	R/W	Data Length Select number of data bits. Write operation requires USR[0] to be 0, that is, the UART peripheral is not in a busy state. 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits Reset value: 0x00

13.8.10 Modem Control Register (MCR)

ÿ **Name:** Modem Control Register ÿ **Size:** 32 bits

ÿ **Address Offset:** 0x10

ÿ **Read/write access:** read/write

31 30 29 28 27 26 25 24 23 22	twenty one	20	19	18	17	16
reserved						

15	14 13	12	11	10 9	8	7	6	5	4	3	2	1	0
reserved							SIRE	AFCE	LB	OUT2	OUT1	RTS	DTR

Bit	Name	R/W	Description
31:7	reserved	-	A read operation returns 0
6	SIRE	R/W	SIR Mode Enable

			Infrared mode enabled. Write operation requires USR[0] to be 0, that is, the UART peripheral is not in a busy state. 0 = disable 1 = enable Reset value: 0x00 If the SIR mode is enabled, the MCR should be configured before configuring the LCR register. Make the appropriate configuration.
5	AFCE	R/W	Auto Flow Control Enable Automatic flow control enabled. 0 = disable 1 = enable Reset value: 0x00
4	LB	R/W	LookBack Bit Diagnostic mode enabled. This mode is used for testing. In UART mode (MCR[6] = 0), the data on the sout output remains high level, the data output by sout is internally looped back to the input of sin. in this mode Interrupts are available. In loopback mode, modem control inputs (dsr_n, cts_n, ri_n, dcd_n) Not connected, modem control output (dtr_n, rts_n, out1_n, out2_n) are all within loopback to the modem control input. In infrared mode (MCR[6] = 1), the sout output remains low and the sout output The output is internally looped back to the sin input after the level is inverted. Reset value: 0x00
3	OUT2	R/W	OUT2 Output of Output2 pin (out2_n). 0 = output logic 1 1 = output logic 0 In LookBack mode (MCR[4] = 1), out2_n remains high, This position is now an input to the inner loop. Reset value: 0x00
2	OUT1	R/W	OUT1 Output of the Output1 pin (out1_n). 0 = output logic 1 1 = output logic 0 In LookBack mode (MCR[4] = 1), out1_n remains high, This position is now an input to the inner loop. Reset value: 0x00
1	RTS	R/W	Request to Send RTC output. The specific control methods are as follows: 1. Automatic flow control is not enabled (MCR[5]=0): This bit directly controls the output of the RTS pin (Request to Send). set When 1, the RTS pin outputs an active level (low level), and when it is cleared to 0, the RTS pin Output fail level (high). 2. Automatic flow control enabled (MCR[5]=1) and FIFO enabled (FCR[0] = 1): This bit acts as the RTC enable bit. When set to 1, the RTS pin output enable is turned on, When cleared to 0, the RTS pin output enable is disabled. The output level of the RTS pin is controlled by the receive FIFO threshold trigger. When the data is lower than the threshold, the RTS pin outputs a valid level (low level). When the received data is equal to or higher than the threshold, the RTS pin outputs an inactive level (high flat).

			Automatic flow control FIFO: enable off (MCR[5]=0) / enable close (MCR[5]=0) → RTS: direct control Automatic flow control FIFO: enable shutdown (MCR[5]=0) / enable open (FCR[0] = 1) → RTS: direct control Automatic flow control FIFO: enable open (MCR[5]=1) / enable close (MCR[5]=0) → RTS: -- (Automatic flow control mode formula must have FIFO support) Automatic flow control FIFO: enable open (MCR[5]=1) / enable open (FCR[0] = 1) → RTS: enable control Reset value: 0x00
0	DTR	R/W	Data Terminal Ready Data terminal ready state output The setting written to the register is the logical inverse of the value output by the pin 0 = dtr_n logic 1 1 = dtr_n logic 0 The data port ready output is used to notify the modem that the UART communication is established. Reset value: 0x00

13.8.11 Line Status Register (LSR)

ÿ **Name:** Line Status Register ÿ
Size: 32 bits
ÿ **Address Offset:** 0x14
ÿ **Read/write access:** read-only

31	30	29	28	27	26	25	24	23	Reserved	twenty two	twenty one	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved								RFE	TEMT	THRE	BI	FE	PE	OE	DR

Bit	Name	R/W -	Description
31:8	reserved	A read operation returns 0	
7	RFE	RC_R	Receiver FIFO Error bit Receive FIFO error flag When the FIFO is enabled (FCR[0] = 1), this bit is used to indicate the receive At least one byte of data in the FIFO has a parity error, framing error or break state. 0 = RX FIFO has no errors 1 = RX FIFO has an error After the error byte is at the top of the receive FIFO and there are no more errors in the receive FIFO In the case of byte data, read the LSR register to clear this bit to 0. Reset value: 0x00
6	TEMT	R	Transmitter Empty bit ÿIn non-FIFO mode (FCR[0] = 0): When this bit is 1, it means the transmit shift register (TSR) and transmit holding register (THR) is also empty. ÿFIFO mode (FCR[0] = 1): A 1 in this bit indicates that both the transmit shift register (TSR) and the FIFO are empty.

			in: abbreviation TSR: Transmitter Shift Register THR: Transmitter Holding Register If the above setting conditions are not met, the hardware is cleared to 0. Reset value: 0x01
5	THRE	R	Transmit Holding Register Empty bit. The THRE interrupt is enabled. When THRE is set to 1, the THRE interrupt will be triggered. ÿTHRE interrupt mode is cleared to 0 (IER[7] = 0): This bit is used to indicate that the THR or transmit FIFO is empty (FIFO enable is active). When data is output to TSR by THR or transmit FIFO and there is no new data Writing to either the THR or the transmit FIFO causes this bit to be set. ÿTHRE interrupt mode is set to 1 (IER[7] = 1) and FIFO enable is valid (FCR[0] = 1): The function of this bit changes to the state that the transmit FIFO threshold is triggered, and is no longer used. It means that the THR is empty, and the transmit FIFO threshold is set by FCR[5:4]. This bit is set to 1 after the amount of data in the send FIFO triggers the threshold. If the above setting conditions are not met, the hardware is cleared to 0. Reset value: 0x01
4	BI	RC_R	Break Interrupt bit Break interrupt flag This bit is used to indicate that the UART peripheral receives the break sequence signal from the peer device. number data. If the UART peripheral receives a break signal, the break signal Byte reception is considered to be a 0 for each bit. If the break signal remains Even if the transmission time exceeds 1 time, it is only received as 1 byte data. In UART mode (MCR[6] = 0), if the input pin remains at logic 0 The duration is greater than the sum of the time of start bit, data bit, check bit and stop bit, The position 1. In SIR mode (MCR[6] = 1), if the Logic 0 pulse duration is greater than the start bit, data bit, parity bit and stop bit and, the position 1. In non-FIFO mode (FCR[0] = 0): This bit is set to 1 immediately after the break signal data byte is received In FIFO mode (FCR[0] = 1): This bit is set when the break signal data byte is received and is at the top of the queue. Set to 1. Note: When a break signal is received when the receive FIFO is full, it will generate a FIFO overload occurred. At this time, other additional information of the byte data (check letter information, framing error information, and break information) are discarded. This bit is cleared by reading the LSR register. Reset value: 0x00
3	FE	RC_R	Framing Error bit frame error flag This bit is used to indicate a framing error in data reception. When the UART peripheral is connected

			A framing error occurs when the receiver does not receive a valid stop bit. In FIFO mode, each byte of data is received with a framing error message. The framing error flag is set when the byte data that caused the framing error is at the top of the receive FIFO. When debug occurs, the UART peripheral attempts to reproduce synchronization. The UART peripheral will assume that the error is due to the start bit of the next byte and continue to receive subsequent bits. It should be noted that the frame error flag will be set to 1 after receiving a break signal data, that is to say, the frame error flag will also be set when the break flag is set. Because the break signal is represented by a continuous logic 0 level, a framing error is bound to occur. 0 = No framing error 1 = There is a framing error Reading the LSR register clears this bit. Reset value: 0x00 Parity Error bit Parity error flag bit
2	PE	RC_R	When the verification enable is valid (LCR[3] = 1), this is used to indicate the verification error that occurs during data reception. In FIFO mode, each byte of data is received with a parity error message. The parity error flag is set to 1 when the byte data that caused the parity error is at the top of the receive FIFO. It should be noted that when the check enable is valid (LCR[3] = 1) and the check type is Odd (LCR[4] = 0), the check error flag will be reset when a break signal data is received. Set to 1, that is, when the break flag is set, it will also cause the check error flag to be set. 0 = No parity error 1 = Parity error Read LSR register to clear this bit. Reset value: 0x00 Overrun error bit Overrun error flag bit
1	OE	RC_R	This bit is used to indicate an overload error. An overload error occurs when new data is received and previous data is not read in time. ÿIn non-FIFO mode (FCR[0] = 0): 1 new byte data is received and the previous data in the BRB is not read in time, the flag is set to 1. If an overload occurs, the data previously in the BRB is overwritten. ÿIn FIFO mode (FCR[0] = 1): When a new byte of data is received when the receive FIFO is full, an overload error will occur and the flag will be set to 1. In the event of an overload, the UART peripheral reserves the previously received data in the FIFO and discards the currently received data. 0 = No overload error 1 = There is an overload error Reading the LSR register clears this bit. Reset value: 0x00 Data Ready bit Data valid flag
0	DR	R	This bit is used to indicate that at least 1 valid data has been received in the BRB or receive FIFO.

			0 = no valid data 1 = has valid data The hardware is cleared to 0 after the following operations: ÿIn non-FIFO mode (FCR[0] = 0), read the BRB. ÿIn FIFO mode (FCR[0] = 1), read the receive FIFO until until the receive FIFO is empty. Reset value: 0x00

13.8.12 Modem Status Register (MSR)

ÿ **Name:** Modem Status Register ÿ **Size:** 32 bits

ÿ **Address Offset:** 0x18

ÿ **Read/write access:** read-only

Bits 0, 1, 2, and 3 are used to indicate that the modem control input changes, and the input changes correspond to position 1. If modem in IER If the status interrupt enable is turned on, the corresponding interrupt will be generated, otherwise the generated interrupt will be ignored. Because in the synchronous modem signal version the During the process, the modem control input will be reset, the reset value is 0, and the value becomes 1 after the reset is completed, so even if the signals in the modem are Invalid, the above bits will also be set to 1 after reset. Read the MSR register after reset to prevent unnecessary interrupt is generated.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved								DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS

Bit	Name	R/W	Description
31:8	reserved	- A read operation returns 0	
7	DCD	R	Data Carrier Detect DCD status flag This bit is used to indicate the current state of the modem control line DCD. when the location 1, indicating that the Modem has sensed the "Data Carrier". 0 = DCD input is inactive (high) 1 = DCD input pin is active (low level) In loopback mode (MCR[4] = 1), DCD and OUT2 (MCR[3]) are state is the same. Reset value: 0x00
6	RI	R	Ring Indicator RI status flag This bit is used to indicate the state of the current modem control line RI. When the position is 1, the table indicates that the Modem has received a "telephone ringing signal" signal). 0 = RI input is inactive (high) 1 = RI input pin is active (low level) In loopback mode (MCR[4] = 1), RI and OUT1 (MCR[2]) status same. Reset value: 0x00
5	DSR	R	Data Set Ready

			DSR status flag This bit is used to indicate the current state of the modem control line DSR. when the location 1, means that the data sent by Modem to UART peripheral is ready. 0 = DSR input is inactive (high) 1 = DSR input is active (low) In loopback mode (MCR[4] = 1), DSR and DTR (MCR[0]) states same. Reset value: 0x00
4	CTS	R	Clear to Send CTS status flag This bit is used to indicate the current state of the modem control line CTS. when the location 1, indicates that the UART peripheral receives the Modem's request data signal (RTS). 0 = CTS input is inactive (high) 1 = CTS input is active (low level) In loopback mode (MCR[4] = 1), CTS and RTS (MCR[1]) states same. Reset value: 0x00
3	DDCD	RC_R	Delta Data Carrier Detect DDCD status flag This bit is used to indicate that the current state of the modem control line DCD has changed. This bit is set to indicate that the state of the DCD has changed since the last MSR read. 0 = The state of the DCD has not changed since the last MSR read 1 = The state of the DCD has changed since the last MSR read Clear this bit on MSR read operation, in loopback mode (MCR[4] = 1), DDCD indicates the state change of OUT2 (MCR[3]). Note that DDCD is set to 1 when: The DDCD state is 0, the DCD signal is valid and a reset is generated, if the DCD is a remain active until reset is complete, then DDCD will be set. Reset value: 0x00
2	TERI	RC_R	Trailing Edge of Ring Indicator TERI status flag This bit is used to indicate that the current state of the modem control line RI has changed (by Active low state becomes inactive high state). This position is 1, indicating that the upper The RI state changes after the next MSR read. 0 = The state of RI has not changed since the last MSR read 1 = The state of RI has changed since the last MSR read Clear this bit on MSR read operation, in loopback mode (MCR[4] = 1), TERI indicates the state change of OUT1 (MCR[2]) (active low state to high inactive state). Reset value: 0x00
1	DDSR	RC_R	Delta Data Set Ready DDSR status flag bit This bit is used to indicate that the current state of the modem control line DSR has changed. This bit is set to 1 to indicate that the state of the DSR has changed since the last read of the MSR. 0 = The state of the DSR has not changed since the last MSR read 1 = The state of the DSR has changed since the last MSR read Clear this bit on MSR read operation, in loopback mode (MCR[4] = 1), DDSR indicates the state change of DTR (MCR[0]).

			Note that DDSR is set to 1 when: DDSR state is 0, DSR signal is valid and reset is generated, if DSR is one remain active until reset is complete, then DDSR will be set. Reset value: 0x00
0	DCTS	RC_R	Delta Clear to Send DCTS status flag This bit is used to indicate that the current state of the modem control line CTS has changed. Should Bit 1 to indicate that the state of the CTS has changed since the last MSR read. 0 = The state of the CTS has not changed since the last MSR read 1 = The state of the CTS has changed since the last MSR read Clear this bit on MSR read operation, in loopback mode (MCR[4] = 1), DCTS Indicates the state change of the CTS (MCR[1]). Note that DCTS is set to 1 when: DCTS state is 0, CTS signal is valid and reset is generated, if CTS is a remain active until reset is complete, then DCTS will be set to 1. Reset value: 0x00

13.8.13 FIFO Access Enable Register (FAR)

ÿ **Name:** FIFO Access Register ÿ
Size: 32 bits
ÿ **Address Offset:** 0x70
ÿ **Read/write access:** read/write

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

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

Bit	Name	R/W	Description
31:1	reserved	-	-
0	FAR	R/W	FIFO Access Register FIFO access enable This bit is used for FIFO testing and controls whether the FIFO can be accessed by the user. When the enable is turned on, the user can write to the receive FIFO and read the transmit FIFO. Enable Off, the user can only access the FIFO through RBR and THR. ÿ 0 = FIFO access enable is off 1 = FIFO access enable is on Note: When FIFO access is enabled for on/off operation, receive and transmit The control part of the FIFO is reset and the FIFO is also considered empty. Reset value: 0x00

13.8.14 Read Transmit FIFO Register (TFR)

ÿ **Name:** Transmit FIFO Read
ÿ **Size:** 32 bits
ÿ **Address Offset:** 0x74
ÿ **Read/write access:** read-only

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

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

Bit	Name	R/W	Description
31:8	reserved	-	-
7:0	TFRD	R	Transmit FIFO Read Read transmit FIFO data This bit is valid only when the FIFO access enable is turned on (FAR[0] = 1). When the FIFO function is turned on, a read operation on this bit will return the transmit FIFO queue Data at the top of the column. Continuous read operations will take out the data in the FIFO in turn, The data read in each read operation is the top data of the current transmit FIFO queue. When the FIFO function is turned off, a read operation of this bit will return the data in the THR. Reset value: 0x00

13.8.15 Write Receive FIFO Register (RFW)

- **Name:** Receive FIFO Write
- **Size:** 32 bits
- **Address Offset:** 0x78
- **Read/write access:** write-only

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

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

Bit	Name	R/W	Description
31:10	reserved	-	-
9	RFFE	W	Receive FIFO Framing Error. Receive FIFO Framing Error command This bit is valid only when the FIFO access enable is turned on (FAR[0] = 1). When the FIFO function is turned on, this bit is used to write framing error information to the receive FIFO. When the FIFO function is disabled, this bit is used to write framing error information to the RBR. Reset value: 0x00
8	RFPE	W	Receive FIFO Parity Error. Receive FIFO Parity Error Command This bit is valid only when the FIFO access enable is turned on (FAR[0] = 1). When the FIFO function is turned on, this bit is used to write check error information to the receive FIFO. When the FIFO function is disabled, this bit is used to write the checksum error information to the RBR. Reset value: 0x00
7:0	RFWD	W	Receive FIFO Write Data Write receive FIFO data

			This bit is valid only when the FIFO access enable is turned on (FAR[0] = 1). When the FIFO function is turned on, the data written to this bit will be pushed into the receive FIFO. Data written to this bit each time will be pushed into the receive FIFO the next time this bit is written middle. When the FIFO function is turned off, the data written to this bit will be pushed into the RBR. Reset value: 0x00

13.8.16 UART Status Register (USR)

ÿ **Name:** UART Status Register ÿ
Size: 32 bits
ÿ **Address Offset:** 0x7C
ÿ **Read/write access:** read-only

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

15 14 13 12 11 10 9 8 7 6 5	4	3	2	1	0
reserved	RFF	RFNE	TFE	TFNF	BUSY

Bit	Name	R/W	Description
31:5	reserved	-	-
4	RFF	R	Receive FIFO Full Receive FIFO full (completely full) flag. 0 = Receive FIFO is not full 1 = Receive FIFO is full This bit is cleared when the receive FIFO is no longer full. Reset value: 0x00
3	RFNE	R	Receive FIFO Not Empty Receive FIFO not empty flag. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the receive FIFO is empty. Reset value: 0x00
2	TFE	R	Transmit FIFO Empty Transmit FIFO is empty (completely empty) flag. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the transmit FIFO is not empty. Reset value: 0x00
1	TFNF	R	Transmit FIFO Not Full Transmit FIFO not full flag. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the transmit FIFO is full. Reset value: 0x00
0	BUSY	R	UART Busy UART busy flag This bit is 1 when the serial transfer is in progress, 0 means the UART peripheral is idle or

			Inactive. 0 = UART peripheral is idle or inactive 1 = UART peripheral is busy (serial data transfer in progress) Either of the following conditions will cause this bit to be set, i.e. the UART peripheral is busy: 1. The serial port is sending 2. The FIFO access enable is not turned on (FAR[0] = 0), and the baud rate divider is not 0 (DLH, DLL ÿ 0), divider access enable is disabled (LCR.DLAB = 0) And there is transmit data in THR 3. The serial port is receiving data 4. The FIFO access enable is not turned on (FAR[0] = 0) and there is a reception in the RBR The data Reset value: 0x00

13.8.17 Transmit FIFO Data Volume (TFL)

ÿ **Name:** Transmit FIFO Level ÿ **Size: 32bits**

ÿ **Address Offset:** 0x80

ÿ **Read/write access:** read-only

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

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

Bit	Name	R/W	Description
31:4	reserved	-	-
3:0	TFL	R	Transmit FIFO Level The current number of data in the transmit FIFO. Reset value: 0x00

13.8.18 Receive FIFO Data Volume (RFL)

ÿ **Name:** Receive FIFO Level ÿ **Size: 32bits**

ÿ **Address Offset:** 0x84

ÿ **Read/write access:** read-only

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

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

Bit	Name	R/W	Description
31:4	reserved	-	-
3:0	RFL	R	Receive FIFO Level The current number of data in the receive FIFO. Reset value: 0x00

13.8.19 Soft Reset Register (SRR)

ÿ **Name:** Software Reset Register ÿ
Size: 32 bits
ÿ **Address Offset:** 0x88
ÿ **Read/write access:** write-only

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

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

Bit	Name	R/W	Description
31:3	reserved	-	-
2	XFR	W	XMIT FIFO Reset. Transmit FIFO reset This operation bit is the shadow operation bit of FCR[2]. When the user only resets the transmit FIFO When the transmit FIFO is reset through the FCR register, the previous FCR register will be overwritten. Register settings, use this shadow operation bit to avoid the above situation. the operation The control portion of the transmit FIFO is reset and the transmit FIFO is initially empty. reset Deasserts the DMA TX request signal. This bit is cleared by hardware. Reset value: 0x00
1	RFR	W	RCVR FIFO Reset Receive FIFO reset This operation bit is the shadow operation bit of FCR[1]. When the user only resets the receive FIFO When the receive FIFO is reset through the FCR register, the previous FCR register will be overwritten. Register settings, use this shadow operation bit to avoid the above situation. the operation The control portion of the receive FIFO is reset and the receive FIFO is initially empty. reset Deasserts the DMA TX request signal. This bit is cleared by hardware. Reset value: 0x00
0	UR	W	UART Reset UART reset This bit operation resets the UART peripheral and requires 2 execution clock cycles. Wait for both pclk and sclk to complete reset. The reset flag is cleared immediately after the reset is completed. Chi. Reset value: 0x00

13.8.20 Send Request Shadow Register (SRTS)

ÿ **Name:** Shadow Request to Send
ÿ **Size:** 32 bits
ÿ **Address Offset:** 0x8C
ÿ **Read/write access:** read/write

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

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

reserved	SRTS

Bit	Name	R/W	Description
31:1	reserved	-	-
0	SRTS	R/W	Shadow Request to Send RTS output shadow register This operation bit is the shadow operation bit of MCR[1] (RTS). This bit is used to control the UART peripheral RTC output. The specific control methods are as follows: 1. Automatic flow control is not enabled (MCR[5]=0): This bit directly controls the output of the RTS pin (Request to Send). set When 1, the RTS pin outputs an active level (low level), and when it is cleared to 0, the RTS pin Output fail level (high). 2. Automatic flow control enabled (MCR[5]=1) and FIFO enabled (FCR[0] = 1): This bit acts as the RTC enable bit. When set to 1, the RTS pin output enable is turned on, When cleared to 0, the RTS pin output enable is disabled. The output level of the RTS pin is controlled by the receive FIFO threshold trigger. When the data is lower than the threshold, the RTS pin outputs a valid level (low level). When the received data is equal to or higher than the threshold, the RTS pin outputs an inactive level (high flat).

Automatic flow control	FIFO	RTS	Note
enable off MCR[5]=0	enable close MCR[5]=0	direct control	
enable shutdown MCR[5]=0	enable open FCR[0] = 1	direct control	
enable open MCR[5]=1	enable close MCR[5]=0	--	Automatic flow control mode formula must have FIFO support
enable open MCR[5]=1	enable open FCR[0] = 1	enable control	

Reset value: 0x00

13.8.21 Break Signal Control Shadow Register (SBCR)

ÿ **Name:** Shadow Break Control Register ÿ **Size:** 32 bits
ÿ **Address Offset:** 0x90
ÿ **Read/write access:** read/write

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

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

Bit	Name	R/W	Description
31:1	reserved	-	-
0	SBCR	R/W	Shadow Break Control Bit Break Control Shadow Register

			This operation bit is the shadow operation bit of LCR[6] (Break Control bit). pass This bit eliminates read-modify-write operations on the LCR. This bit is used to generate the output break signal to the receiving device, this bit is 1, the UART Peripheral output pins force a logic 0 signal. When in non-loopback mode (MCR[4] = 0), UART mode (MCR[4] = 0), the sout output will be forced low straight. Infrared mode (MCR[4] = 1), sout continuously outputs pulse signals. After this bit is cleared to 0, the output stops. When in loopback mode (MCR[4] = 1), the output break signal is Loopback to the receiver, the sout output is forced low. Reset value: 0x00

13.8.22 DMA Mode Shadow Register (SDMAM)

- **Name:** Shadow DMA Mode
- **Size:** 32 bits
- **Address Offset:** 0x94
- **Read/write access:** read/write

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

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

Bit	Name	R/W	Description
31:1	reserved	-	-
0	SDMAM	R/W	Shadow DMA Mode. DMA Mode Shadow Register This operation bit is the shadow operation bit of FCR[3] (DMA mode bit). This bit can avoid affecting the FCR if only the DMA mode bits are modified content previously set. This bit is not selected in the extra DMA handshake signal (DMA_EXTRE = NO) The case is used to determine the mode of DMA send request and receive request signal. 0 = Do not use DMA 1 = use DMA Reset value: 0x00

13.8.23 FIFO Enable Shadow Register (SFE)

- **Name:** Shadow FIFO Enable
- **Size:** 32 bits
- **Address Offset:** 0x98
- **Read/write access:** read/write

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

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

Bit	Name	R/W	Description
31:1	reserved	-	-
0	SFE	R/W	Shadow FIFO Enable FIFO enable shadow register This operation bit is the shadow operation bit of FCR[0] (FIFO enable bit). This bit can avoid affecting the FCR when only the FIFO enable bit is modified previously set content. Used to enable receive and transmit FIFOs. Receive and transmit when this bit transmit changes FIFO will be reset. Reset value: 0x00

13.8.24 Receive FIFO Full Threshold Set Shadow Register (SRT)

ÿ **Name:** Shadow RCVR Trigger ÿ **Size:**
32 bits
ÿ **Address Offset:** 0x9C
ÿ **Read/write access:** read/write

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

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

Bit	Name	R/W	Description
31:2	reserved	-	-
1:0	SRT	R/W	Shadow RCVR Trigger Threshold setting shadow register for receive FIFO full trigger. This operation bit is the shadow operation bit of FCR[7:6] (RCVR Trigger). This bit avoids affecting the FCR if only the receive FIF threshold is modified previously set content. When the data in the receive FIFO exceeds the set threshold, it will trigger the receive data valid interrupt. In automatic flow control mode, this threshold is used to determine the received rts_n signal state (in RTC_FCT is turned off). In DMA mode, the UART sends DMA after the amount of data in the FIFO triggers the threshold ask. The following are the supported threshold types. 00 – 1 character in the FIFO 01 – FIFO ¼ full 10 – FIFO ½ full 11 – FIFO 2 less than full Reset value: 0x00

13.8.25 Transmit FIFO Empty Threshold Set Shadow Register (STET)

ÿ **Name:** Shadow TX Empty Trigger ÿ **Size:**
32 bits
ÿ **Address Offset:** 0xA0
ÿ **Read/write access:** read/write

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

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

Bit	Name	R/W	Description
31:2	reserved	-	-
1:0	STET	R/W	Shadow TX Empty Trigger The transmit FIFO empty threshold sets the shadow register. This operation bit is the shadow operation bit of FCR[5:4] (TX Empty Trigger). This bit can avoid affecting the FCR if only the transmit FIF threshold is modified content previously set. When the PTIME enable is turned on (IER[7] = 1), the data in the transmit FIFO is equal to or low At this threshold, the THRE interrupt will be triggered. In DMA mode, the UART sends DMA after the amount of data in the FIFO triggers the threshold ask. The following are the supported threshold types. 00 – FIFO empty 01 – 2 characters in the FIFO 10 – FIFO ¼ full 11 – FIFO ½ full Reset value: 0x00

13.8.26 Transmit Halt Register (HTX)

- **Name:** Halt TX
- **Size:** 32 bits
- **Address Offset:** 0xA4
- **Read/write access:** read/write

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

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

Bit	Name	R/W	Description
31:1	reserved	-	-
0	HTX	R/W	Halt TX Transmit Pause Register This register is used for UART testing and is manipulated to stop UART transmission. In this way, when the FIFO is enabled, the transmit FIFO can be written by the user. Full. 0 = stop TX enable off 1 = Stop TX enable on Reset value: 0x00

13.8.27 DMA Soft Acknowledge (DMASA)

- **Name:** DMA Software Acknowledge
- **Size:** 32 bits
- **Address Offset:** 0xA8
- **Read/write access:** write

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

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

Bit	Name	R/W	Description
31:1	reserved	-	-
0	DMASA	W	DMA Software Acknowledge DMA Soft Acknowledge If an error occurs during the DMA transfer and needs to be terminated, it can be Use this register to generate a DMA soft acknowledge. For example, if the DMA channel enable is off, the UART needs to clear its request signal, then the operation of this register will make TX request, TX single, RX The request and RX single signals are invalid. The operation of this register does not need to be used Cleared to 0 by the household, and cleared to 0 by the hardware. Reset value: 0x00

14 I2C interface

14.1 Introduction to I2C

The I2C (Inter-Chip) bus interface connects the microcontroller and the serial I2C bus. It provides multi-master capability and controls all I2C bus specific timing, protocol, arbitration and timing. Both standard and express modes are supported. Depending on the needs of a particular device, DMA can be used to offload the CPU.

14.2 Main Features of I2C

ÿ The I2C clock is provided by PCLK, that is, I2C_CLK = PCLK ÿ Multi-
master function: the module can be used as both a master device and a slave
device I2C slave/master function support ÿ I2C multiple status, interrupt and error
detection ÿ Generate and detect 7-bit/10-bit address and general call) ÿ Support
I2C protocol SDA HOLD/SETUP time setting ÿ DMA support

14.3 I2C function description

The I2C module receives and transmits data, and converts the data from serial to parallel, or parallel to serial. Interrupts can be enabled or disabled. The interface is connected to the I2C bus through the data pin (SDA) and the clock pin (SCL). Allows connection to standard (up to 100kHz) or fast (up to 400kHz) I2C bus.

14.3.1 I2C peripheral clock (I2C_CLK)

The I2C peripheral clock is provided by PCLK without frequency division, that is, I2C_CLK = PCLK

Define the following parameters:

I2C_CLK: I2C peripheral clock frequency

I2C_CLK_PERIOD: I2C peripheral clock cycle time

14.3.2 Receive/Transmit FIFO

The I2C peripheral contains 2 separate receive and transmit FIFOs with a depth of 8. The CPU writes the register IC_DATA_CMD, the number of According to the data written into the sending FIFO, the CPU reads the register IC_DATA_CMD, and the data is taken out from the receiving FIFO.

The receive and transmit FIFOs have independent interrupt threshold settings, and I2C sends an interrupt request to the CPU when the data meets the set threshold. The receive and transmit FIFOs have independent DMA threshold settings. When the data meets the set threshold, I2C sends out the corresponding DMA request. The set threshold needs to be set in combination with the DMA Burst (MSIZE) value, see DMA "SRC_MSIZE/DEST_MSIZE parameter setting" chapter

14.3.3 START and STOP signal generation

The I2C peripheral works in the master mode. When the CPU writes data to the IC_DATA_CMD data register, the I2C generates a START signal.

No. When the IC_DATA_CMD[9] bit that the CPU writes data to IC_DATA_CMD is 1, the I2C will generate a STOP signal. In the process of sending, IC_DATA_CMD[9] is not set to "1", and I2C will not generate a STOP signal, even if the sending FIFO is empty.

The I2C peripheral works in slave mode and does not generate START and STOP signals.

14.3.4 I2C Protocol

In master mode, the I2C interface initiates data transfers and generates clock signals. Serial data transfers always begin with a START condition and end with a STOP condition. Both the START and STOP conditions are generated under software control in master mode. In slave mode, the I2C interface can recognize its own address (7-bit or 10-bit) and general call address. Software can control to enable or disable the recognition of general call addresses. Data and addresses are transmitted in 8 bits/byte, high order first. 1 or 2 bytes following the start condition is the address (1 byte for 7-bit mode, 2 bytes for 10-bit mode). Addresses are only sent in master mode. During the ninth clock after eight clocks of a byte transfer, the receiver must send an acknowledge bit (ACK) back to the transmitter. Refer to the image below.

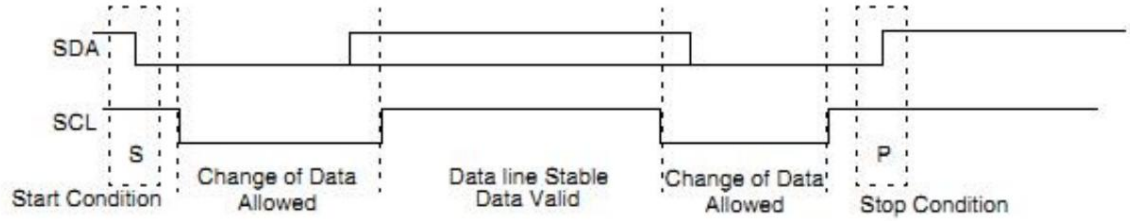


Figure 14- 1 I2C protocol

14.3.5 I2C Master Mode

The operation process is as follows:

1. IC_ENABLE[0] = 0 to turn off the I2C peripheral enable
2. Set the working speed mode through IC_CON, set the working mode as the master
mode 3. Write IC_TAR to set the target slave address, which can be set according to IC_TAR[12] Mode is 7 address or 10 address
4. Set IC_SS_SCL_HCNT/IC_SS_SCL_LCNT or IC_SS_SCL_LCNT according to the operating speed mode
IC_FS_SCL_HCNT/IC_FS_SCL_LCNT Baud rate 5. Set
IC_SDA_SETUP/IC_SDA_HOLD parameters as needed

8. IC_ENABLE[0] = 1 Enable I2C peripheral enable

14.3.6 I2C Slave Mode

1. IC_ENABLE[0] = 0 Disable I2C peripheral enable 2. Set
the working mode to slave mode through IC_CON, and set the slave address response mode (7 address or 10 address)
3. Write IC_SAR to set the local slave device address 4 ÿ Set the trigger threshold of the transceiver FIFO through
IC_RX_TL/IC_TX_TL 5. Open the required interrupt through IC_INTR_MASK

6. IC_ENABLE[0] = 1 Enable I2C peripheral enable

14.3.7 I2C Glitch Suppression

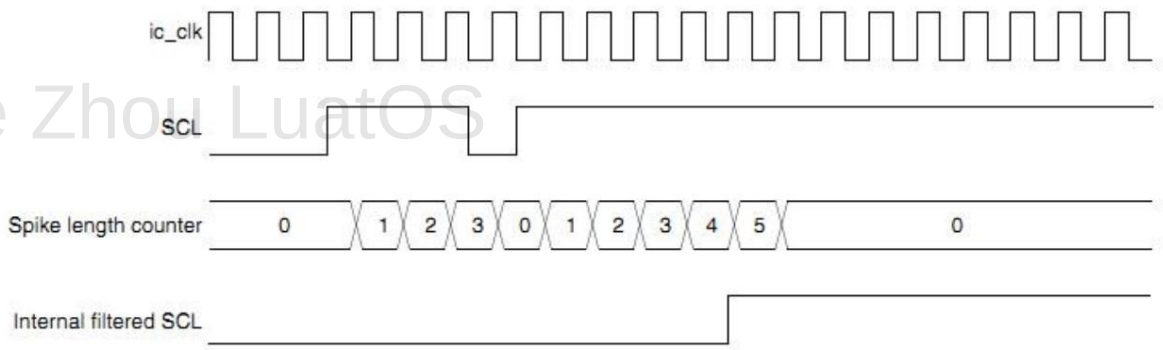
The I2C peripheral suppresses glitches on the I2C bus through the IC_FS_SPKLEN register.

The I2C peripheral contains a programmable glitch suppression unit, which is used to match the requirements of the "standard", "fast" I2C mode. glitch suppression The controller uses a counter to monitor the SCL and SDA level status, and checks whether the level maintains a preset number of times before the input level is sampled. IC_CLK cycle. The SCL and SDA signals have their own independent counters. The user can program the preset number, after calculating the preset number IC_CLK frequency and peak width need to be considered.

When the level changes, the respective counters are started. Details are as follows:

1. The input level signal remains stable until the counter accumulates to the set value. At this time, the state of the internal level signal changes New is the new input state, and the glitch suppression counter is reset and stopped. Counter until the next level send change again start up.
2. Before the counter accumulates to the set value, the input level signal changes. At this time, the state of the internal level signal does not change. At the same time the glitch suppression counter is reset and stopped. The counter starts again until the next level sends a change.

The suppression process is as follows:



The glitch suppressor register is 8 wide. This register can only be operated when the I2C peripheral is disabled. The minimum write value is 1, if Attempt to write a value less than 1, the result is set to 1 by default.

Suppressed glitch width = IC_FS_SPKLEN * I2C_CLK_PERIOD

14.3.8 I2C baud rate setting

Each speed mode of the I2C peripheral contains 2 baud rate setting registers

Standard mode baud rate setting register:

IC_SS_SCL_HCNT

IC_SS_SCL_LCNT

Fast Mode Baud Rate Setting Register:

IC_FS_SCL_HCNT

IC_FS_SCL_LCNT

IC_xx_SCL_HCNT: baud rate high period count

IC_xx_SCL_LCNT: baud rate low period count

Define the following parameters:

SCL_PERIOD: baud rate cycle time

SCL_PERIOD_Ht: baud rate high level hold time

SCL_PERIOD_Lt: Baud rate low level hold time

According to the above definition, the calculation is as follows:

SCL_PERIOD = SCL_PERIOD_Ht + SCL_PERIOD_Lt

SCL_PERIOD_Ht = IC_xx_SCL_HCNT * I2C_CLK_PERIOD

SCL_PERIOD_Lt = IC_xx_SCL_LCNT * I2C_CLK_PERIOD

SCL_PERIOD = (IC_xx_SCL_HCNT + IC_xx_SCL_LCNT) * I2C_CLK_PERIOD

Minimum requirements for baud rate high/low level:

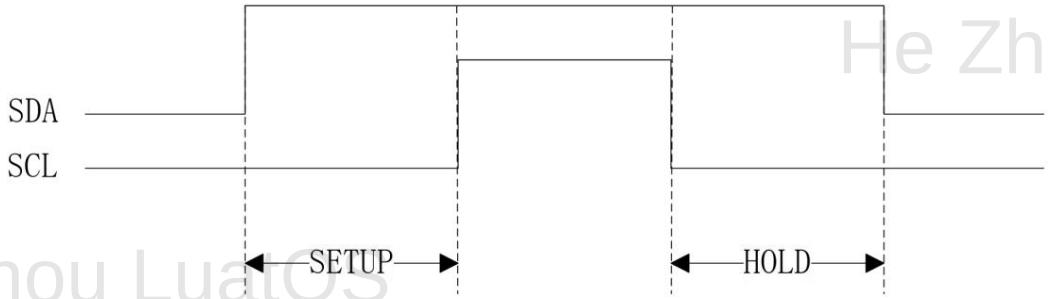
IC_SS_SCL_LCNT/ IC_FS_SCL_LCNT must be greater than IC_FS_SPKLEN +7

IC_SS_SCL_HCNT / IC_FS_SCL_HCNT must be greater than IC_FS_SPKLEN +5

14.3.9 SDA SETUP/HOLD time setting

The I2C peripheral provides IC_SDA_SETUP and IC_SDA_HOLD registers to set the SDA SETUP and HOLD time

SETUP/HOLD schematic diagram:



See "I2C bus protocol specification" for detailed time parameter requirements of SDA SETUP and HOLD.

IC_SDA_SETUP: SETUP time period count

IC_SDA_HOLD: HOLD time period count

SETUP time = IC_SDA_SETUP * I2C_CLK_PERIOD

HOLD time = IC_SDA_HOLD * I2C_CLK_PERIOD

14.3.10 DMA operations

The I2D peripheral supports DMA data operations to reduce CPU usage.

IC_DMA_CR register:

This register is used to enable the SPI transceiver DMA function, with 2bit control bit operations respectively.

IC_DMA_TDLR/IC_DMA_RDLR registers:

IC_DMA_TDLR/IC_DMA_RDLR register is used to control I2C to generate request condition for DMA.

ÿ When the data in the transmit FIFO meets the set value of IC_DMA_TDLR, the I2C is triggered to request DMA, and the DMA After the corresponding request, according to the configuration of the corresponding channel, the data volume of Burst (MSIZE) is written to the transmit FIFO at one time. data.

ÿ When the data in the receiving FIFO meets the IC_DMA_RDLR setting value, the I2C is triggered to request DMA, and the DMA After the corresponding request, according to the corresponding channel configuration, the data volume of Burst (MSIZE) is taken out from the transmit FIFO at one time. data.

For specific settings, please refer to the chapter "SRC_MSIZE/DEST_MSIZE Parameter Setting" in DMA.

14.4 I2C register description

14.4.1 Address Mapping Table

List of I2C base addresses

Address	base	Peripherals	bus
range 0x4004_9000-0x4004_9FFF	address 0x4004_9000	I2C0	APB3

Table 14- 1 I2C register table

Offset address register name		Width (bit) Reset value	
0x00	IC_CON	32	0x0000007D

0x04	IC_TAR	32	0x00001055
0x08	IC_SAR	32	0x00000055
0x0C	IC_HS_MADDR	32	0x00000000
0x10	IC_DATA_CMD	32	0x00000000
0x14	IC_SS_SCL_HCNT	32	0x00000190
0x18	IC_SS_SCL_LCNT	32	0x000001D6
0x1C	IC_FS_SCL_HCNT	32	0x0000003C
0x20	IC_FS_SCL_LCNT	32	0x00000082
0x24	reserved reserved	32	0x00000000
0x28		32	0x00000000
0x2C	IC_INTR_STAT	32	0x00000000
0x30	IC_INTR_MASK	32	0x000008FF
0x34	IC_RAW_INTR_STAT	32	0x00000000
0x38	IC_RX_TL	32	0x00000000
0x3C	IC_TX_TL	32	0x00000000
0x40	IC_CLR_INTR	32	0x00000000
0x44	IC_CLR_RX_UNDER	32	0x00000000
0x48	IC_CLR_RX_OVER	32	0x00000000
0x4C	IC_CLR_TX_OVER	32	0x00000000
0x50	IC_CLR_RD_REQ	32	0x00000000
0x54	IC_CLR_TX_ABRT	32	0x00000000
0x58	IC_CLR_RX_DONE	32	0x00000000
0x5C	IC_CLR_ACTIVITY	32	0x00000000
0x60	IC_CLR_STOP_DET	32	0x00000000
0x64	IC_CLR_START_DET	32	0x00000000
0x68	IC_CLR_GEN_CALL	32	0x00000000
0x6C	IC_ENABLE	32	0x00000000
0x70	IC_STATUS	32	0x00000006
0x74	IC_TXFLR	32	0x00000000
0x78	IC_RXFLR	32	0x00000000
0x7C	IC_SDA_HOLD	32	0x00000001
0x80	IC_TX_ABRT_SOURCE	32	0x00000000
0x84	IC_SLV_DATA_NACK_ONLY	32	0x00000000
0x88	IC_DMA_CR	32	0x00000000
0x8C	IC_DMA_TDLR	32	0x00000000
0x90	IC_DMA_RDLR	32	0x00000000
0x94	IC_SDA_SETUP	32	0x00000064
0x98	IC_ACK_GENERAL_CALL	32	0x00000001
0x9C	IC_ENABLE_STATUS	32	0x00000000
0xA0	IC_FS_SPKLEN	32	0x00000005

14.4.2 I2C Control Register (IC_CON)

ÿ **Name: I2C Control Register**
ÿ **Size: 32bits**
ÿ **Address Offset: 0x00**

The write operation of this register must be performed in the state where the I2C peripheral is turned off, and the write operation is invalid in other cases.

31	30	29	28	27	2 6	2 5	twenty four

reserve							
twenty three	twenty two	twenty one	20	19	18	17	16
reserve							
15	14	13	12	11	10	9	8
reserve							
7	6	5	4	3	twenty one		0
guarantee Keep	IC_SLAVE_DISABLE	IC_RESTART_EN	IC_10BITADDR_MASTER	IC_10BITADDR_SLAVE	SPEED		MASTER_MODE

Bit	Name	R/W	Description
31:7	reserve	-	
6	IC_SLAVE_DISABLE R/W		I2C slave mode enable bit. 0 - open from mode 1-Slave mode off
5	IC_RESTART_EN	R/W	RESTART signal support in I2C master mode 0 - RESTART signal not supported 1- Support RESTART signal
4	IC_10BITADDR_MASTER	R	Address is mode in I2C master mode The address in I2C master mode is set by IC_TAR[12], this bit is only used for to get the status of IC_TAR[12] 0-7bit address mode 1-10bit address mode
3	IC_10BITADDR_SLAVE	R/W	Address bit pattern in I2C slave mode 0-7bit address mode 1-10bit address mode
2:1	SPEED	R/W	I2C transfer rate mode 1-standard mode (0 to 100kbit/s) 2-fast mode (400kbit/s)
0	MASTER_MODE	R/W	I2C master mode enable bit 0 - master mode off 1- Master mode on

I2C master-slave mode configuration table:

IC_SLAVE_DISABLE (IC_CON[6])	MASTER_MODE IC_CON[0]	State
0	0	Slave Device
0	1	Config Error
	0	Config Error
1 1	1	Master Device

14.4.3 I2C Target Address Register (IC_TAR)

- **Name: I2C Target Address Register**
- **Size: 32 bits**
- **Address Offset: 0x04**
- **Read/Write Access: Read/Write**

This register is only in I2C peripheral off state (IC_ENABLE[0]=0) or I2C idle state (IC_ENABLE[0]=1 and IC_STATUS[5]=0 and IC_CON[0]=1 and IC_STATUS[2]=1) write operation, otherwise write is invalid.

31	30	29	28	27	26	25	twenty four
reserve							
twenty three	twenty two	twenty one	20	19	18	17	16
reserve							
15	14	13	12	11	10	9	8
reserve			IC_10BITADDR_MASTER	SPECIAL	GC_OR_START	IC_TAR	
7	6	5	4	3	2	1	0
IC_TAR							

Bit	Name	R/W	Description
31:13	reserve	-	
12	IC_10BITADDR_MASTER	R/W	Address is mode in I2C master mode 0-7bit address mode 1-10bit address mode
11	SPECIAL	R/W	I2C address call (General Call) and start byte (START BYTE) command order support 0 - not supported 1- Support
10	GC_OR_START	R/W	I2C address call (General Call) / start byte (START BYTE) selection choice 0-address call 1=start byte Address call: After making an address call, by reading The TX_ABRT bit in the IC_RAW_INTR_STAT register gets the call result. The address call mode will remain until SPECIAL is cleared to "0".
9:0	IC_TAR	R/W	I2C target address I2C works in master mode, slave device target address. IC_TAR is ignored when I2C makes an address call, and when a start byte is sent, the CPU 1 write to IC_TAR is required.

14.4.4 I2C Slave Address Register (IC_SAR)

- ÿ **Name: I2C Slave Address Register**
- ÿ **Size: 32 bits**
- ÿ **Address Offset: 0x08**
- ÿ **Read/Write Access: Read/Write**

The write operation of this register must be performed in the state where the I2C peripheral is turned off, and the write operation is invalid in other cases.

31	30	29	28	27	26	25	twenty four
reserve							
twenty three	twenty two	twenty one	20	19	18	17	16
reserve							
15	14	13	12	11	10	9	8
						IC_SAR	
7	6	5	4	3	2	1	0
IC_SAR							

Bit	Name	W/R	Description
31:10	reserved	-	

9:8	IC_SAR	W/R	I2C Slave Address I2C works in slave mode, when the master device operates on the I2C bus address for matching.

14.4.5 I2C Receive/Transmit Data Command Register (IC_DATA_CMD)

- **Name: Rx/Tx Data Buffer and Command Register**
- **Size: 32 bits**
- **Address Offset: 0x10**
- **Read/Write Access: Read/Write**

The write operation of this register must be performed in the state where the I2C peripheral is turned off, and the write operation is invalid in other cases.

31	30	29	28	27	26	25	twenty four
reserve							
twenty three	twenty two	twenty one	20	19	18	17	16
reserve							
15	14	13	12	11	10	9	8
Reserved					RESTART STOP		CMD
7	6	5	4	3	2	1	0
DATA							

Bit	Name	W/R	Description
31:11	reserve	-	
10	RESTART	W	RESTART signal control bit Specify yes after the current byte is sent or before the next byte is received No RESTART signal is generated. When IC_RESTART_EN in IC_CON is "1" be effective. 1 - Generate RESTART signal 0 - RESTART signal is generated only when the transmission direction changes in IC_CON IC_RESTART_EN = 1, I2C generates RESTART. IC_RESTART_EN = 0, RESTART signal by STOP + START signal instead.
9	STOP	W	STOP signal control bit Specify yes after the current byte is sent or before the next byte is received No STOP signal is generated. 1- Generate STOP signal 0 - No STOP signal is generated
8	CMD	W	I2C read and write control bits 1 = read operation 0=write operation
9:8	DATA	W/R	I2C data field Write operation data will be put into the TX FIFO, read operation by RX FIFO fetch data.

14.4.6 I2C Standard Rate Mode SCL High Counter (IC_SS_SCL_HCNT)

- **Name: Standard Speed I2C Clock SCL High Count Register**
- **Size: 32 bits**

ÿ **Address Offset: 0x14**
ÿ **Read/Write Access: Read/Write**

The write operation of this register must be performed in the state where the I2C peripheral is turned off, and the write operation is invalid in other cases.

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

Bit	Name	W/R	Description
31:16	reserve	-	
15:0	IC_SS_SCL_HCNT	W/R	I2C SCL high period counter I2C works in standard speed mode, the The register value determines the SCL high level hold time, the register is the smallest The value is 6.

14.4.7 I2C Standard Rate Mode SCL Low Counter (IC_SS_SCL_LCNT)

ÿ **Name: Standard Speed I2C Clock SCL Low Count Register**
ÿ **Size: 32 bits**
ÿ **Address Offset: 0x18**
ÿ **Read/Write Access: Read/Write**

The write operation of this register must be performed in the state where the I2C peripheral is turned off, and the write operation is invalid in other cases.

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

Bit	Name	W/R	Description
31:16	reserve	-	
15:0	IC_SS_SCL_LCNT	W/R	I2C SCL low period counter I2C works in standard speed mode, the The register value determines the SCL low level hold time, the register is the smallest The value is 6.

14.4.8 I2C Fast Mode SCL High Counter (IC_FS_SCL_HCNT)

ÿ **Name: Fast Speed I2C Clock SCL High Count Register**
ÿ **Size: 32 bits**
ÿ **Address Offset: 0x1c**
ÿ **Read/Write Access: Read/Write**

The write operation of this register must be performed in the state where the I2C peripheral is turned off, and the write operation is invalid in other cases.

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

Bit	Name	W/R	Description
31:16	reserve	-	
15:0	IC_FS_SCL_HCNT	W/R	I2C SCL high period counter I2C works in high speed (fast speed) mode, the value of this register Determines the SCL high level hold time, the minimum value of the register is 8.

14.4.9 I2C Fast Mode SCL Low Counter (IC_FS_SCL_LCNT)

- **Name: Fast Speed I2C Clock SCL Low Count Register**
- **Size: 32bits**
- **Address Offset: 0x20**
- **Read/Write Access: Read/Write**

The write operation of this register must be performed in the state where the I2C peripheral is turned off, and the write operation is invalid in other cases.

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

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

Bit	Name	W/R	Description
31:16	reserve	-	
15:0	IC_FS_SCL_LCNT	W/R	I2C SCL low period counter I2C works in fast speed mode, the value of this register Determines the holding time of SCL low level, the minimum value of the register is 8.

14.4.10 I2C Interrupt Status Register (IC_INTR_STAT)

- **Name: I2C Interrupt Status Register**
- **Size: 32 bits**
- **Address Offset: 0x2C**
- **Read/Write Access: Read**

Each bit of this register corresponds to each bit of the IC_INTR_MASK register. Each interrupt in this register is read by reading the corresponding The interrupt clear register is cleared. The original interrupt status is maintained by the IC_RAW_INTR_STAT register and is not sent by IC_INTR_MASK. memory impact.

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
reserve															

15	14	13	12	11	10	9	8
reserve				R_GEN_CALL	R_START_DET	R_STOP_DET	R_ACTIVITY

7	6	5	4	3	2	1	0
R_RX_DONE	R_TX_ABRT	R_RD_REQ	R_TX_EMPTY	R_TX_OVER	R_RX_FULL	R_RX_OVER	R_RX_UNDER

Bit	Name	W/R	Description
31:12	reserve	-	
11	R_GEN_CALL	R	I2C interrupt status

10	R_START_DET	R	The corresponding bit of the I2C interrupt mask register is "1" and a corresponding interrupt is generated,
9	R_STOP_DET	R	Then it should be set to "1".
8	R_ACTIVITY	R	The interrupt flag is cleared by reading the corresponding interrupt clear register.
7	R_RX_DONE	R	"0".
6	R_TX_ABRT	R	
5	R_RD_REQ	R	See IC_RAW_INTR_STAT for the description marked as corresponding
4	R_TX_EMPTY	R	
3	R_TX_OVER	R	
2	R_RX_FULL	R	
1	R_RX_OVER	R	
0	R_RX_UNDER	R	

14.4.11 I2C Interrupt Mask Register (IC_INTR_MASK)

- ÿ **Name: I2CInterrupt Mask Register**
- ÿ **Size: 32 bits**
- ÿ **Address Offset: 0x30**
- ÿ **Read/Write Access: Read/Write**

The bits in this register are used to mask each interrupt status in the interrupt status register.

31	30	29	28	27	26	25	twenty four
reserve							
twenty three	twenty two	twenty one	20	19	18	17	16
reserve							
15	14	13	12	11	10	9	8
reserve				M_GEN_CALL	M_START_DET	M_STOP_DET	M_ACTIVITY
7	6	5	4	3	2	1	0
M_RX_DONE	M_TX_ABRT	M_RD_REQ	M_TX_EMPTY	M_TX_OVER	M_RX_FULL	M_RX_OVER	M_RX_UNDER

Bit	Name	W/R	Description
31:12	reserve	-	
11	M_GEN_CALL	W/R	I2C Interrupt Mask
10	M_START_DET	W/R	0 - mask interrupt
9	M_STOP_DET	W/R	1- Turn on interrupts
8	M_ACTIVITY	W/R	
7	M_RX_DONE	W/R	See IC_RAW_INTR_STAT for the description marked as corresponding
6	M_TX_ABRT	W/R	
5	M_RD_REQ	W/R	
4	M_TX_EMPTY	W/R	
3	M_TX_OVER	W/R	
2	M_RX_FULL	W/R	
1	M_RX_OVER	W/R	
0	M_RX_UNDER	W/R	

14.4.12 I2C Raw Interrupt Status Register (IC_RAW_INTR_STAT)

- ÿ **Name: I2CRaw Interrupt Status Register**
- ÿ **Size: 32 bits**

ÿ **Address Offset: 0x34**

ÿ **Read/Write Access: Read**

The use of each bit in this register is not affected by the masked interrupt register, and the corresponding bit is "1" when the corresponding condition occurs.

31	30	29	28	27	26	25	twenty four
reserve							
twenty three	twenty two	twenty one	20	19	18	17	16
reserve							
15	14	13	12	11	10	9	8
reserve				GEN_CALL	START_DET	STOP_DET	ACTIVITY
7	6	5	4	3	2	1	0
RX_DONE	TX_ABRT	RD_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER		RX_UNDER

Bit	Name	W/R	Description
31:12	reserve	-	
11	GEN_CALL	R	Address Call (General Call) Interrupted 0 - no call to address received 1- Received address call Clear "0" operation: 1. Read the corresponding interrupt clear register 2. Turn off I2C The data received by the address call is saved to the Rx buffer
10	START_DET	R	START signal interrupt 0 - No START or RESTART signal detected 1- START or RESTART letter detected Clear "0" operation: 1. Read the corresponding interrupt clear register
9	STOP_DET	R	STOP signal interrupt 0 - STOP signal not detected 1 - STOP signal detected Clear "0" operation: 1. Read the corresponding interrupt clear register
8	ACTIVITY	R	ACTIVITY interrupt 0 - I2C is not detected in ACTIVITY 1 - I2C detected in ACTIVITY Clear "0" operation: 1. Turn off I2C 2. Read the corresponding interrupt clear register 3. Read the global interrupt clear register 4. System reset This bit is "1" and remains set to "1" unless it is cleared to "0".
7	RX_DONE	R	In I2C slave mode, the other host receives complete interrupt 0 - The receiving host of the other party has not completed 1- The receiving host of the other party is completed

			Clear "0" operation: 1. Read the corresponding interrupt clear register I2C is in slave mode when the other host completes receiving data, no ack is generated signal, this bit is "1", which occurs when the data transfer is complete.
6	TX_ABRT	R	send termination interrupt 0 - no send termination detected 1 - Send termination detected When the transmission of data in the transmit FIFO cannot be completed, this bit is "1", I2C Termination occurs in both master-slave mode. Send termination reason save In the transmit termination source register (IC_TX_ABRT_SOURCE)
5	RD_REQ	R	Master request interrupt in I2C slave mode 0 - no host request detected 1 - Host request detected
4	TX_EMPTY	R	Transmit FIFO empty interrupt 0 - Transmit FIFO not empty 1 - Transmit FIFO empty This position is when the data in the transmit FIFO is empty or below the trigger threshold "1" Clear "0" operation: This bit is cleared to "0" by
3	TX_OVER	R	hardware Transmit FIFO overflow interrupt 0 - The transmit FIFO has not overflowed 1 - Transmit FIFO overflow In the process of I2C sending, the CPU writes data to IC_DATA_CMD Trigger transmit FIFO overflow interrupt after causing FIFO overflow
2	RX_FULL	R	Receive FIFO full interrupt 0 - the receive FIFO is not full 1 - Receive FIFO full This bit is set to 1 when the data in the receive FIFO is full or above the trigger threshold Clear "0" operation: This bit is cleared to "0" by
1	RX_OVER	R	hardware Receive FIFO overflow interrupt 0 - The receive FIFO has not overflowed 1 - Receive FIFO overflow Receiving data in I2C causes FIFO overflow, and subsequent received data is lost after overflow. lose.
0	RX_UNDER	R	Receive FIFO underflow 0 - The receive FIFO has not underflowed 1 - Receive FIFO underflow When the receive FIFO data is empty, the CPU tries to Read operation, this bit is "1".

14.4.13 I2C Receive FIFO Threshold Register (IC_RX_TL)

- **Name: I2CReceive FIFO Threshold Register**
- **Size: 32bits**
- **Address Offset: 0x38**
- **Read/Write Access: Read/Write**

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

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

Bit	Name	W/R	Description
31:8	reserve	-	
7:0	RX_TL	W/R	Receive FIFO threshold The RX_FULL interrupt is set when the received data exceeds this threshold (RX_FULL[2] in IC_RAW_INTR_STAT) Value range 0-7 0 - An overflow interrupt is generated when there is 1 valid data in the receive FIFO 1- An overflow interrupt is generated when there are 2 valid data in the receive FIFO 2- An overflow interrupt is generated when there are 3 valid data in the receive FIFO 3- An overflow interrupt is generated when there are 4 valid data in the receive FIFO 4- An overflow interrupt is generated when there are 5 valid data in the receive FIFO 5- An overflow interrupt is generated when there are 6 valid data in the receive FIFO 6- An overflow interrupt is generated when there are 7 valid data in the receive FIFO 7- An overflow interrupt is generated when there are 8 valid data in the receive FIFO

14.4.14 I2C Transmit FIFO Threshold Register (IC_TX_TL)

- **Name: Transmit FIFO Threshold Register**
- **Size: 32 bits**
- **Address Offset: 0x3c**
- **Read/Write Access: Read/Write**

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

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

Bit	Name	W/R	Description
31:8	reserve	-	
7:0	TX_TL	W/R	Transmit FIFO Threshold The TX_EMPTY interrupt is set when the received data is lower than the set threshold (RX_FULL[4] in IC_RAW_INTR_STAT) Value range 0-7 0 - Overrun interrupt is generated when there is less than 1 valid data in the transmit FIFO 1 - Overrun interrupt is generated when there are less than 2 valid data in the transmit FIFO

			2- An overflow interrupt is generated when there are less than 3 valid data in the transmit FIFO 3- An overflow interrupt is generated when there are less than 4 valid data in the transmit FIFO 4- An overflow interrupt is generated when there are less than 5 valid data in the transmit FIFO 5- An overflow interrupt is generated when there are less than 6 valid data in the transmit FIFO 6- An overflow interrupt is generated when there are less than 7 valid data in the transmit FIFO 7- An overflow interrupt is generated when there are less than 8 valid data in the transmit FIFO

14.4.15 I2C Global Interrupt Clear Register (IC_CLR_INTR)

- **Name: Clear Combined and Individual Interrupt Register**
- **Size: 32 bits**
- **Address Offset: 0x40**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
reserve							
twenty three	twenty two	twenty one	20	19	18	17	16
reserve							
15	14	13	12	11	10	9	8
reserve							
7	6	5	4	3	2	1	0
Reserved							CLR_INTR

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_INTR	R	global interrupt clear The read operation clears the interrupt, this bit only clears the interrupt that can be cleared by software, Does not include interrupt status that is automatically cleared by hardware.

14.4.16 I2C Receive FIFO Underflow Interrupt Clear Register (IC_CLR_RX_UNDER)

- **Name: Clear RX_UNDER Interrupt Register**
- **Size: 32bits**
- **Address Offset: 0x44**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
Reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CLR_RX_UNDER

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_RX_UNDER	R	Clear receive FIFO underflow interrupt

			Clear RX_UNDE (IC_RAW_INTR_STAT[0]) interrupt status state, a read operation clears the interrupt.

14.4.17 I2C Receive FIFO Overrun Interrupt Clear Register (IC_CLR_RX_OVER)

- **Name: Clear RX_OVER Interrupt Register**
- **Size: 32 bits**
- **Address Offset: 0x48**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
Reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
reserve							CLR_RX_OVE R

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_RX_OVER	R	Receive FIFO overflow interrupt clear Clear RX_OVER (IC_RAW_INTR_STAT[1]) interrupt status, A read operation clears the interrupt.

14.4.18 I2C Transmit FIFO Overrun Interrupt Clear Register (IC_CLR_TX_OVER)

- **Name: Clear TX_OVER Interrupt Register**
- **Size: 32 bits**
- **Address Offset: 0x4c**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
reserve							
twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
reserve							CLR_TX_OVE R

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_TX_OVER	R	Transmit FIFO overflow interrupt clear Clear RX_OVER (IC_RAW_INTR_STAT[3]) interrupt status, A read operation clears the interrupt.

14.4.19 I2C Slave Mode Read Request Interrupt Clear Register (IC_CLR_RD_REQ)

- **Name: Clear RD_REQ Interrupt Register**
- **Size: 32 bits**
- **Address Offset: 0x50**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
Reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CLR_RD_REQ

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_RD_REQ	R	Slave mode read request interrupt clear Clear RD_REQ (IC_RAW_INTR_STAT[5]) interrupt status, A read operation clears the interrupt.

14.4.20 I2C Transmit Termination Interrupt Clear Register (IC_CLR_TX_ABRT)

- **Name: Clear TX_ABRT Interrupt Register**
- **Size: 32 bits**
- **Address Offset: 0x54**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
reserve							
twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
reserve							CLR_TX_ABRT

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_TX_ABRT	R	Transmit Terminate Interrupt Clear Clear TX_ABRT (IC_RAW_INTR_STAT[6]) interrupt status, A read operation clears the interrupt.

14.4.21 I2C Slave Mode Transmit Complete Interrupt Clear Register (IC_CLR_RX_DONE)

- **Name: Clear RX_DONE Interrupt Register**
- **Size: 32bits**
- **Address Offset: 0x58**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
Reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
reserve							CLR_RX_DONE

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_RX_DONE	R	Slave Mode Transmit Complete Interrupt Clear Clear RX_DONE (IC_RAW_INTR_STAT[7]) interrupt status state, a read operation clears the interrupt.

14.4.22 I2C ACTIVITY Interrupt Clear Register (IC_CLR_ACTIVITY)

- **Name: Clear Clear ACTIVITY Interrupt Register**
- **Size: 32 bits**
- **Address Offset: 0x5c**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
Reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CLR_ACTIVITY

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_ACTIVITY	R	ACTIVITY Interrupt Clear Clear ACTIVITY (IC_RAW_INTR_STAT[8]) interrupt status state, a read operation clears the interrupt. Only I2C is already in a non-ACTIVITY state, this register A read operation can clear the ACTIVITY interrupt status, otherwise The ACTIVITY interrupt is reset to '1'.

14.4.23 I2C STOP Interrupt Clear Register (IC_CLR_STOP_DET)

- **Name: Clear STOP_DET Interrupt Register**
- **Size: 32 bits**
- **Address Offset: 0x60**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
Reserved							

twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CLR_STOP_DET

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_STOP_DET	R	STOP interrupt clear Clear STOP_DET (IC_RAW_INTR_STAT[9]) interrupt status state, a read operation clears the interrupt.

14.4.24 I2C START Interrupt Clear Register (IC_CLR_START_DET)

- **Name: Clear START_DET Interrupt Register**
- **Size: 32bits**
- **Address Offset: 0x64**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
Reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CLR_START_DET

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_START_DET	R	START interrupt clear Clear START_DET (IC_RAW_INTR_STAT[10]) in In the off state, a read operation clears the interrupt.

14.4.25 I2C Address Broadcast Interrupt Clear Register (IC_CLR_GEN_CALL)

- **Name: Clear GEN_CALL Interrupt Register**
- **Size: 32bits**
- **Address Offset: 0x68**
- **Read/Write Access: Read**

31	30	29	28	27	26	25	twenty four
Reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
reserve							CLR_GEN_CALL

Bit	Name	W/R	Description
31:1	reserve	-	
0	CLR_GEN_CALL	R	Address Broadcast Interrupt Clear Clear GEN_CALL (IC_RAW_INTR_STAT[11]) interrupt state, a read operation clears the interrupt.

14.4.26 I2C Enable Register (IC_ENABLE)

- **Name: I2C Enable Register**
- **Size: 32 bits**
- **Address Offset: 0x6C**
- **Read/Write Access: Read/Write**

31	30	29	28	27	26	25	twenty four
reserve							
twenty three	twenty two	twenty one	20	19	18	17	16
reserve							
15	14	13	12	11	10	9	8
reserve							
7	6	5	4	3	2	1	0
reserve						ABORT	ENABLE

Bit	Name	W/R	Description
31:2	reserve	-	
1	ABORT	W/R	Software terminates the transfer 0-ABORT is started or completed 1-ABORT operation in progress I2C operates in master mode, and I2C transfers are terminated by software. ABORT can only be done when I2C ENABLE has been set to "1" operation, otherwise the ABORT operation is ignored. The ABORT bit cannot be cleared to "0" by software, and the ABORT operation is completed by hardware. Automatic cleaning after operation. During an ABORT operation, the hardware issues STOP signal, clear the TX FIFO, and set the ABORT interrupt bit to "1".
0	ENABLE	W/R	I2C enable bit 0 - I2C is off (both TX and RX FIFOs are empty) 1-I2C open An I2C shutdown operation will be accompanied by the following states: 1. TX FIFO and RX FIFO are cleared 2. IC_INTR_STAT will remain until I2C enters IDLE state

14.4.27 I2C Status Register (IC_STATUS)

- **Name: I2C Status Register**
- **Size: 32 bits**
- **Address Offset: 0x70**
- **Read/Write Access: Read**

Bit bit "1" in this register will not cause an interrupt request.

When I2C is off:

ÿ Bits 1 and 2 are set to 1

ÿ Bits 3 and 4 are set to 0

When I2C is off and in IDLE state:

ÿ Bits 5 and 6 are set to 0

31	30	29	28	27	26	25	twenty four
Reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserve	SLV_ACTIVITY	MST_ACTIVITY	RFF	RFNE	TFE	TFNF	ACTIVITY

Bit	Name	W/R	Description
31:7	reserved	-	
6	SLV_ACTIVITY	R	I2C Slave Final State Machine (Slave Finite State Machine) (FSM) 0 - Slave mode FSM state is inactive 1- Slave mode FSM state to Active state The Activity state means that the I2C FSM is in the non-IDLE state.
5	MST_ACTIVITY	R	I2C Master Finite State Machine (FSM) 0 - Main mode FSM state is inactive 1-Main mode FSM state is Active state The Activity state means that the I2C FSM is in the non-IDLE state.
4	RFF	R	Receive FIFO full 0 - Receive FIFO is not full 1 - Receive FIFO full Whether this bit is set or not is not affected by IC_RX_TL
3	RFNE	R	Receive FIFO is not empty 0 - Receive FIFO empty (no data in FIFO) 1-Receive FIFO is not empty (1 or more data in FIFO) Whether this bit is set or not is not affected by IC_RX_TL
2	TFE	R	Transmit FIFO empty 0 - Transmit FIFO is not empty (1 or more data in FIFO) 1 - Transmit FIFO empty (no data in FIFO) Whether this bit is set or not is not affected by IC_TX_TL
1	TFNF	R	Transmit FIFO not full 0 - Transmit FIFO is full 1 - Transmit FIFO is not full Whether this bit is set or not is not affected by IC_TX_TL
0	ACTIVITY	R	I2C Final State Machine (Slave Finite State Machine) (FSM) 0 - inactive state

			1-Activity state The Activity state means that the I2C FSM is in the non-IDLE state.

14.4.28 I2C transmit FIFO data volume register (IC_TXFLR)

- **Name: I2C Transmit FIFO Level Register**
- **Size: 32 bits**
- **Address Offset: 0x74**
- **Read/Write Access: Read**

The amount of valid data in the I2C transmit FIFO is cleared to "0" in the following cases:

- Disable I2C
- Transmit abort

The register value increases when data is written to the FIFO, and decreases when the I2C fetches data from the FIFO.

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

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

Bit	Name	W/R	Description
31:4	reserve	-	
3:0	TXFLR	R	Transmit FIFO data volume Number of valid data contained in the transmit FIFO

14.4.29 I2C Receive FIFO Data Volume Register (IC_RXFLR)

- **Name: I2C ReceiveFIFO Level Register**
- **Size: 32bits**
- **Address Offset: 0x78**
- **Read/Write Access: Read**

The amount of valid data in the I2C receive FIFO is cleared to "0" in the following cases:

- Disable I2C
- Transmit abort

The register value increases when I2C receives data into the FIFO, and decreases when the user fetches data from the FIFO.

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

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

Bit	Name	W/R	Description
31:4	reserve	-	
3:0	RXFLR	R	Receive FIFO data volume Number of valid data contained in the receive FIFO

14.4.30 I2C SDA HOLD Time Register (IC_SDA_HOLD)

- **Name: SDA Hold Time Length Register**
- **Size: 32bits**
- **Address Offset: 0x7C**
- **Read/Write Access: Read/Write**

The register value is used to control the time that SDA needs to be held after the falling edge of SCL occurs, and the time is in units of PCLK cycles. I2C in master mode
The minimum value is 1 PCLK cycle, and the minimum is 7 PCLK cycles in slave mode.

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

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

Bit	Name	W/R	Description
31:16	reserved	-	
15:0	IC_SDA_HOLD	W/R	Set the SDA hold time, the unit time is PCLK cycle

14.4.31 I2C Transmit Termination Source Register (IC_TX_ABRT_SOURCE)

- **Name: I2C Transmit Abort Source Register**
- **Size: 32bits**
- **Address Offset: 0x80**
- **Read/Write Access: Read**

Except for register bit 9 (TX_ABRT), all other bits can be read by reading IC_CLR_TX_ABRT or IC_CLR_INTR.
line clear.
Clear register bit 9 (TX_ABRT) must meet the following conditions:
ÿ RESTART must be valid (IC_CON[5]=1)
ÿ SPECIAL (IC_TAR[11]) or GC_OR_START (IC_TAR[10]) must be cleared to "0".
After the above conditions are met, the read operation of IC_CLR_TX_ABRT or IC_CLR_INTR can be cleared.
The TX_ABRT bit is automatically set after division.

31	30	29	28	27	26	25	twenty four
TX_FLUSH_CNT							

twenty three	twenty two	twenty one	20	19	18	17	16
reserve							ABRT_USER_ABRT

15	14	13	12	11	10	9	8
ABRT_SLVRD_INTX	ABRT_SLV_ARBLOST	ABRT_SLVFLUSH_TXFIFO	ARB_LOST	ABRT_MASTER_DIS	ABRT_10B_RD_NORSTRT	ABRT_SBYTE_NORSTRT	ABRT_HS_NORSTRT

7	6	5	4	3	2	1	0
ABRT_SBYTE_ACKDET	ABRT_HS_ACKDET	ABRT_GCALL_READ	ABRT_GCALL_NOACK	ABRT_TXDATA_NOACK	ABRT_10ADDR2_NOACK	ABRT_10ADDR1_NOACK	ABRT_7B_ADDR_NOACK

Bit	Name	W/R	Mode	Description
31:24	TX_FLUSH_CNT	R	Master mode is used to record when a TX_ABRT event occurs	value in post-TXFLR
23:17	reserve	-	-	
16	ABRT_USER_ABRT	R	Master mode user actively terminates transfer	

15	ABRT_SLVRD_INTX	R	In slave mode, when the I2C is used as a slave device, the processor rings In response to a remote master data read request, User to IC_DATA_CMD register Write 1 to CMD (bit 8)
14	ABRT_SLV_ARBLOST	R	Slave mode transmits as a slave device to the master device in I2C The bus was lost during data.
13	ABRT_SLVFLUSH_TXFIF O	R	Slave mode receives external when I2C is a slave device Host read request command, if sent at this time If there is data in the FIFO, this bit is "1". In
12	ARB_LOST	R	master/slave mode, when I2C is the master device, the I2C bus is in the middle. Failed Slave-transmitter total when I2C is a slave Line Arbitration Lost
11	ABRT_MASTER_DIS	R	Master/Slave mode In the case of I2C master mode off, use user tries I2C master mode operation
10	ABRT_10B_RD_NORSTR T	R	Master mode is enabled and disabled at IC_RESTART_EN (IC_CON[5] = 0) simultaneously master Send 10 address mode read command
9	ABRT_SBYTE_NORSTRT	R	Master mode is enabled and disabled at IC_RESTART_EN (IC_CON[5] = 0) while the user tries Try sending START BYTE
8	ABRT_HS_NORSTRT	R	Master mode is enabled and disabled at IC_RESTART_EN (IC_CON[5] = 0) while the user tries Try using the main mode to upload in high-speed mode input data
7	ABRT_SBYTE_ACKDET	R	Master mode I2C works in master mode to transmit START BYTE, ACK received
6	ABRT_HS_ACKDET	R	Master mode I2C works in high-speed master mode to transmit START BYTE, ACK received
5	ABRT_GCALL_READ	R	Master Mode I2C Working in Master Mode Sending Address Calls (General Call) while the user is sending After the address call is completed Set IC_DATA_CMD[9] to "1" and try to send send read command
4	ABRT_GCALL_NOACK	R	Master Mode I2C Working in Master Mode Sending Address Calls (General Call), there is no slave on the bus device response
3	ABRT_TXDATA_NOACK	R	Main Mode This bit value is used in main mode. I2C works in master mode and can receive The other party's address is ACK, but when sending data, No ACK response was received.
2	ABRT_10ADDR2_NOACK	R	Master mode I2C works in 10 address mode, the second Byte address has no slave ACK response
1	ABRT_10ADDR1_NOACK	R	Master mode I2C works in 10 address mode, the first Byte address has no slave ACK response
0	ABRT_7B_ADDR_NOACK	R	Master mode I2C works in 7 address mode, sending ground address command byte, no slave ACK answer

14.4.32 I2C slave mode data NACK response register (IC_SLV_DATA_NACK_ONLY)

- **Name: Generate Slave Data NACK Register**
- **Size: 32bits**
- **Address Offset: 0x84**
- **Read/Write Access: Read/Write**

I2C works in slave mode and is used to generate a NACK signal during data transmission.

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

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

Bit	Name	W/R	Description
31:1	reserved	-	
0	NACK	W/R	I2C works in slave mode and is used during data transfer A NACK signal is generated. 1 - Generate NACK after byte data reception is complete 0 - Normal mode generates NACK/ACK

14.4.33 I2C DMA Control Register (IC_DMA_CR)

- **Name: DMA Control Register**
- **Size: 32bits**
- **Address Offset: 0x88**
- **Read/Write Access: Read/Write**

Operations on this register are not affected by I2C enable.

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

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

Bit	Name	W/R	Description
31:2	reserved	-	
1	TDMAE	W/R	Transmit DMA enable 0 - send DMA off 1 - Transmit DMA on
0	RDMAE	W/R	Receive DMA enable 0 - receive DMA off 1 - Receive DMA on

14.4.34 I2C DMA Transmit Data Threshold Register (IC_DMA_TDLR)

- **Name: DMA Transmit Data Level Register**
- **Size: 32bits**
- **Address Offset: 0x8C**
- **Read/Write Access: Read/Write**

The operation of this register is not affected by whether I2C is enabled or not.

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

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

Bit	Name	W/R	Description
31:3	reserved	-	
2:0	IC_DMA_TDLR	W/R	DMA transmit threshold When the amount of data in the transmit FIFO is equal to or less than the DMA transmit Threshold, I2C will request DMA, request DMA Write data to the I2C transmit FIFO.

14.4.35 I2C DMA Receive Data Threshold Register (IC_DMA_RDLR)

- **Name: DMA Receive Data Level Register**
- **Size: 32bits**
- **Address Offset: 0x90**
- **Read/Write Access: Read/Write**

The operation of this register is not affected by whether I2C is enabled or not.

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

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

Bit	Name	W/R	Description
31:3	reserved	-	
2:0	IC_DMA_RDLR	W/R	DMA receive threshold When the amount of data in the receive FIFO is equal to or greater than the DMA transmit Threshold, I2C will request DMA, request DMA Get the data from the receive FIFO in I2C.

14.4.36 I2C SDA SETUP Time Setting Register (IC_SDA_SETUP)

- **Name: SDA Setup Register**
- **Size: 32 bits**
- **Address Offset: 0x94**
- **Read/Write Access: Read/Write**

The register value is used to control the time delay between the rising edge of SCL and the valid of SDA (see I2C Bus Specification for details).
tSU:DAT), the time is in PCLK cycles. This register must be programmed with a value greater than or equal to 2.

This register is valid only when IC_ENABLE[0] = 0.

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

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

reserve	IC_SDA_SETUP

Bit	Name	W/R	Description
31:8	reserved	-	
7:0	IC_SDA_SETUP	W/R	SDA SETUP settings The register value is used to control the time between the rising edge of SCL and the valid period of SDA time delay (see tSU:DAT in I2C Bus Specification for details), The time is measured in PCLK cycles. This register must be programmed with a value greater than or equal to 2.

14.4.37 I2C Address Call Response Register (IC_ACK_GENERAL_CALL)

- **Name: ACK General Call Register**
- **Size: 32 bits**
- **Address Offset: 0x98**
- **Read/Write Access: Read/Write**

The register controls I2C to respond to the address call (General Call) with NACK or ACK.

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

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

Bit	Name	W/R	Description
31:1	reserved	-	
0	ACK_GEN_CALL	W/R	Address Call (General Call) Response 0 - does not respond to address calls 1 - Response to address call

14.4.38 I2C Enable Status Register (IC_ENABLE_STATUS)

- **Name: I2C Enable Status Register**
- **Size: 32 bits**
- **Address Offset: 0x9C**
- **Read/Write Access: Read**

The register is used to reflect the state of I2C after IC_ENABLE[0] is set from 1 to 0.

When IC_ENABLE[0] = 1: SLV_RX_DATA_LOST and SLV_DISABLED_WHILE_BUSY are forced to "0", IC_EN Forced to "1".

When IC_ENABLE[0] = 0: The values of SLV_RX_DATA_LOST and SLV_DISABLED_WHILE_BUSY are valid until IC_EN is set to "1" by hardware.

31	30	29	28	27	26	25	twenty four
Reserved							

twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0

reserve	SLV_RX_DATA_L OST	SLV_DISABLED_WHILE_ BUSY	IC_E N

Bit	Name	W/R	Description
31:3	reserved	-	
2	SLV_RX_DATA_LOST	W/R	Data loss in slave mode When I2C is in slave mode IC_ENABLE[0] is set by "1" "0", reception is terminated and there is at least 1 in the receive FIFO In the case of valid data, this bit is set to "1".
1	SLV_DISABLED_WHILE_B USY	W/R	I2C Slave Mode Busy Shutdown When I2C works in slave mode and is busy IC_ENABLE[0] is set to "0" by "1", this bit is "1".
0	IC_EN	W/R	I2C enable state 0-I2C is off 1-I2C is on

14.4.39 I2C Standard/Fast Mode Glitch Length Register (IC_FS_SPKLEN)

- **Name: I2C SS and FS Spike Suppression Limit Register**
- **Size: 32 bits**
- **Address Offset: 0xa0**
- **Read/Write Access: Read/Write**

The glitch filter parameter is introduced in the I2C Bus Specification related to tSP.

This register is valid only when IC_ENABLE[0] = 0.

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

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

Bit	Name	W/R	Description
31:8	reserved	-	
7:0	IC_FS_SPKLEN	W/R	Glitches in SDA/SCL larger than the set value will be detected by the I2C logic Cell filtering Minimum value is 1

15 Serial Peripheral Interface (SPI)

15.1 Introduction to SPI

The Serial Peripheral Interface (SPI) allows the chip to communicate with external devices in a half/full duplex, synchronous, serial manner. This interface can be configured into master mode and provide the communication clock (SCK) for external slave devices. The interface can also work in a multi-master configuration.

15.2 Main Features of SPI

ÿ The SPI clock is provided by PCLK, that is, SPI_CLK = PCLK ÿ Support the protocol Motorola Serial Peripheral Interface (SPI) ÿ Support the protocol Texas Instruments Serial Protocol (SSP) Transceiver FIFO, can be configured with transceiver FIFO interrupt threshold ÿ SPI0 supports master or slave operation (master/slave address is different) ÿ 4 to 16-bit data frame format selection ÿ Support full duplex, half duplex mode ÿ Slave mode supports CS pulled low Continuous reception ÿ Multiple transceivers, error interrupt detection ÿ DMA support

15.3 SPI function description

15.3.1 SPI peripheral clocks and requirements

The SPI clock is provided by PCLK, i.e. SPI_CLK = PCLK

SPI_CLK: SPI access clock frequency
SPI_M_CLK: SPI master mode bus clock frequency
SPI_S_CLK: SPI slave mode bus clock frequency

Theoretical Clock Requirements:

SPI_CLK >= 2 x SPI_M_CLK
SPI_CLK >= 10 x SPI_S_CLK

15.3.2 Receive/Transmit FIFO

The SPI peripheral contains 2 separate receive and transmit FIFOs with a depth of 16.

The CPU writes the register DR, the data is written into the transmit FIFO, the CPU reads the register DR, and the data is fetched from the receive FIFO out.

The receive and transmit FIFOs have independent interrupt threshold settings. When the data meets the set threshold, the SPI sends an interrupt request to the CPU. The receive and transmit FIFOs have independent DMA threshold settings. When the data meets the set threshold, the SPI sends out the corresponding DMA request. The set threshold needs to be set in combination with the DMA Burst (MSIZE) value, see DMA "SRC_MSIZE/DEST_MSIZE parameter setting" chapter

15.3.3 Interrupt Types

The SPI peripheral supports the following interrupt types: ÿ Transmit FIFO Empty Interrupt

Transmit FIFO empty interrupt, generate an interrupt when the amount of data in the transmit FIFO meets the set threshold

ÿ Transmit FIFO Overflow Interrupt Transmit FIFO
overflow interrupt, write operation to DR when transmit FIFO data is full will cause transmit FIFO overflow interrupt ÿ Receive FIFO
Full Interrupt

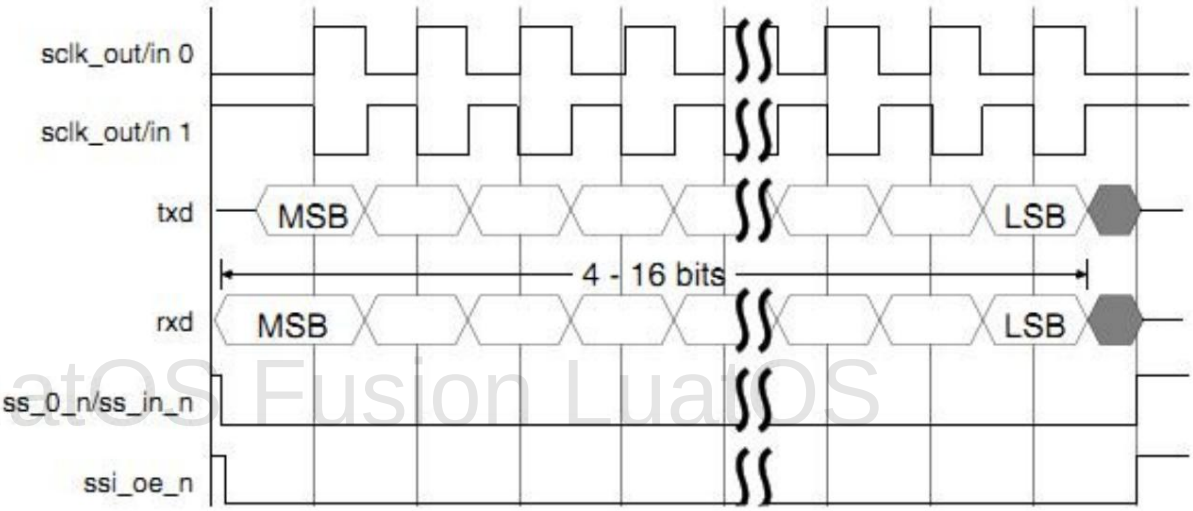
Receive FIFO slow interrupt, when the amount of data in the receive FIFO meets the set threshold, an interrupt is generated

ÿ Receive FIFO Overflow Interrupt Receive FIFO
overflow interrupt, in the case that the receive FIFO data is full, the SPI peripheral receives new data and causes the receive FIFO
overflow interrupt. ÿ Receive FIFO Underflow Interrupt Receive FIFO underflow interrupt, the receive FIFO is already empty. The read
operation of the receive FIFO will trigger the receive FIFO underflow interrupt ÿ Multi-Master Contention Interrupt Multi-master bus arbitration
interrupt, SPI works in live mode and occupies the total first, at this time, other master devices select the current SPI peripheral and
transmit data. ÿ Combined Interrupt Request The above interrupt types pass through the interrupt mask register or the operation value

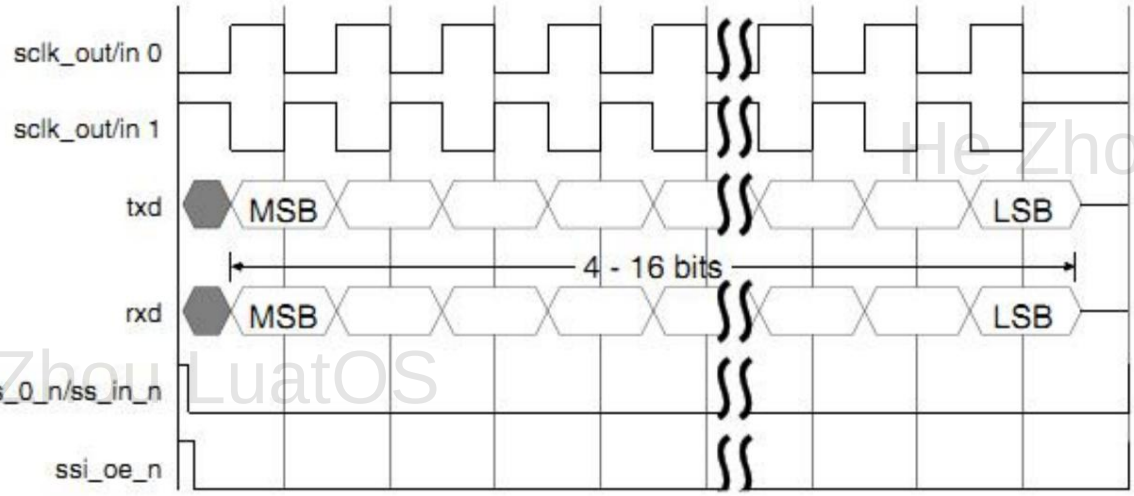
15.3.4 Motorola SPI Common Protocol

There are four communication modes supported by the commonly used Motorola SPI communication protocol, which can realize full-duplex communication. The system works in mode 0 by default when it is powered on.

SCPH = 0:



SCPH = 1:



sclk_out/ in: bus clock, out: SPI output CLK for master device. in: SPI is the slave input CLK sclk_out/ in = 0: (CPHA) = 0
sclk_out/ in = 1: (CPHA) = 1 ss_0_n/ss_in_n: chip select signal, s_0_n: output chip select when SPI is the master device.
s_in_n: input chip select when SPI is master device ss_oe_n: output enable option when SPI is slave mode.

The 4 communication formats specified by the SPI protocol
are described as follows: ÿ Mode 0: Clock polarity (CPOL) = 0, clock phase (CPHA) = 0, the idle state of the serial synchronous clock in this mode is
low level, the chip will The first transition edge (rising edge) of the serial synchronization clock is sampled, the chip defaults to this mode; ÿ
Mode 1: Clock polarity (CPOL) = 0, clock phase (CPHA) = 1, serial synchronization in this mode idle state of the clock
If the state is low, the chip will sample on the second transition edge (falling edge) of the serial synchronization clock;
ÿ Mode 2: Clock polarity (CPOL) = 1, clock phase (CPHA) = 0, in this mode the idle state of the serial synchronous clock is high, the chip will be on
the first transition edge of the serial synchronous clock (falling edge) sampling;
ÿ Mode 3: Clock polarity (CPOL) = 1, clock phase (CPHA) = 1, the serial synchronization clock in this mode is idle
If the state is high, the chip will sample on the second transition (rising edge) of the serial synchronization clock.

15.3.5 SPI Master/Slave Mode Selection

In SPCU, SPI0 includes 2 groups of register groups, which are used to realize master mode (SPIMx) and slave mode (SPISx) respectively. The two groups of register groups have the same structure and different addresses. The SPI0 peripheral operating mode is switched using the corresponding bit of PHER_CTRL in the SYSCTRL register. SPI1~SPI4 only support master mode (SPIMx).

When working in master mode, the corresponding initialization of the SPI and data transceiver operations are completed by SPIMx. When working in slave mode, the corresponding SPI initialization and data reception operations are completed by SPISx

Enable SPI0 clock gating and make it in master mode, the specific process is as follows:

// CG_CTRL is the clock gating register, see the System Control (SYSCTL) chapter for details
CG_CTRL |= 1<<8
// PHER_CTRL is the peripheral control register, see the System Control (SYSCTL) chapter for details
PHER_CTRL &= ~(1<<24)

Enable SPI0 clock gating and make it in slave mode, the specific process is as follows:

// CG_CTRL is the clock gating register, see the System Control (SYSCTL) chapter for details
CG_CTRL |= 1<<8
// PHER_CTRL is the peripheral control register, see the System Control (SYSCTL) chapter for details
PHER_CTRL |= 1<<24

15.3.6 SPI Master Mode Configuration

The master mode uses the SPIMx register set to configure
the configuration process as follows: 1. Configure
PHER_CTRL to make the SPI in master mode 2. Configure
SPIMx->SSIENR = 0, SPI enable and disable 3. Configure SPIMx->
IMR= 0, disable interrupt 4, Configure SPIMx->CTRLR0, configure
transmission mode, frame mode, clock polarity, clock phase, and communication data width 5. Configure SPIMx->BAUDR, configure required baud rate 6. Configure SPIMx->SER, initialize chip select signal 7. Configure SPIMx->RXFTLR and SPIMx->TXFTLR, configure transceiver FIFO interrupt trigger threshold 8. Configure SPIMx->IMR, enable related interrupts 9. Configure SPIMx->SSIENR = 1, enable SPI enable

15.3.7 SPI master mode data transmission and reception

The master mode uses the SPIMx register set for data transmission/
reception. Data transmission/reception process: 1. Determine whether
the transmit FIFO is full. 2. If the transmit FIFO is not full, write data to
SPIMx->DR register. Data will be sent to the corresponding chip select slave device. 3. If SPIMx->SER does not enable any slave device, data is sent after SPIMx->SER is enabled. 4. When the transmit FIFO empty interrupt occurs, write data to SPIMx->DR into the transmit FIFO. When receive FIFO full occurs, read SPIMx->DR to read data from transmit FIFO. 5. The data transfer is completed, and the SPI returns to the non-busy state

15.3.8 SPI Slave Mode Configuration

Slave mode uses the SPISx register set to configure the
configuration process as follows: 1. Configure
PHER_CTRL to make SPI in slave mode 2. Configure SPISx-
>SSIENR = 0, SPI enable and disable 3. Configure SPISx-> IMR= 0,
disable interrupt 4. Configure SPISx->CTRLR0, configure transmission
mode, frame mode, clock polarity, clock phase, slave output enable 5. Configure SPISx->RXFTLR and SPISx->TXFTLR, configure transceiver FIFO interrupt trigger threshold 6. Configure SPISx->IMR , turn on related interrupts 7. Configure SPISx->SSIENR = 1, SPI enable is turned on

15.3.9 SPI slave mode data transmission and reception

Slave mode uses SPISx register set for data transmission/reception.
Data transmission/reception process: 1. When the SPI chip select is
valid, data transmission begins. 2. When transmit FIFO empty interrupt
occurs, write data to SPISx->DR to transmit FIFO. When receive FIFO full occurs, read SPISx->DR to read data from transmit FIFO.

3. The data transfer is completed, and the SPI returns to the non-busy state

15.3.10 DMA operations

The SPI peripheral supports DMA data operations to reduce CPU usage.

DMACR register:

This register is used to enable the SPI transceiver DMA function, and there are 2bit control bit operations respectively.

DMATDLR/DMARDLR registers:

The DMATDLR/DMARDLR registers are used to control the SPI to generate request conditions for DMA.

When the data in the transmit FIFO meets the DMATDLR setting value, the SPI is triggered to request the DMA.

According to the corresponding channel configuration, the data of the burst (MSIZE) data is written into the transmit FIFO at one time.

When the data in the receiving FIFO meets the set value of DMARDLR, the SPI is triggered to request the DMA.

According to the configuration of the corresponding channel, the data of the burst (MSIZE) data is taken out from the transmit FIFO at one time.

For specific settings, please refer to the chapter "SRC_MSIZE/DEST_MSIZE Parameter Setting" in DMA.

15.4 SPI register description

15.4.1 Address Mapping Table

SPIx (x=0...2) base address list

address range	base address	Peripherals	bus
0x4001_A000-0x4001_AFFF	0x4001_A000	SPIM0	APB0
0x4001_B000-0x4001_BFFF	0x4001_B000	SPIS0	
0x4001_8000-0x4001_8FFF	0x4001_8000	SPIM1	
0x4001_9000-0x4001_9FFF	0x4001_9000	SPIM2	
0x4001_0000-0x4004_0FFF	0x4004_0000	SPIM3	APB3
0x4004_1000-FFF	0x4004_1000	SPIM4	

Table 15- 1 SPI register table

Offset address register name		Width (bit)	Reset value
0x00	CTRLR0	32	0x00000007
0x04	CTRLR1	32	0x00000000
0x08	SSIENR	32	0x00000000
0x0C	MWCR	32	0x00000000
0x10	SER	32	0x00000000
0x14	BAUDR	32	0x00000000
0x18	TXFTLR	32	0x00000000
0x1C	RXFTLR	32	0x00000000
0x20	TXFLR	32	0x00000000
0x24	RXFLR	32	0x00000000
0x28	SR	32	0x00000006
0x2C	IMR	32	0x0000003F
0x30	ISR	32	0x00000000
0x34	RISR	32	0x00000000
0x38	TXOICR	32	0x00000000
0x3C	RXOICR	32	0x00000000
0x40	RXUICR	32	0x00000000
0x44	MSTICR	32	0x00000000
0x48	ICR	32	0x00000000
0x4C	DMACR	32	0x00000000
0x50	DMATDLR	32	0x00000000
0x54	DMARDLR	32	0x00000000
0x58	reserved	32	0xFFFFFFFF
0x5C	reservation	32	0x3332322A
0x60	DR	32	0x00000000
0x64-0xEC reserved		32	0x00000000

0xF0	RX_SAMPLE_DLY	32	0x00000000

15.4.2 Control Register 0 (CTRLR0)

- **Name: Control Register 0**
- **Size:** 32 bits
- **Address Offset: 0x0**
- **Read/write access:** read/write

31 30 29 28 27	26	25 24	twenty three	22 21 20 19 18 17	16
reserved					

15 14 13 12 11	10	9 8	7	6	5 4	3 2 1 0
CFS	SRL	SLV_OE	TMOD	SCPOL	SCPH	FRF	DFS

Bit	Name	W/R	Description
31:16	reserved	-	
15:12	CFS	W/R	control frame size Microwire frame format, please refer to the following table for setting values: 0000 1-bit control word 0001 2-bit control word 0010 3-bit control word 0011 4-bit control word 0100 5-bit control word 0101 6-bit control word 0110 7-bit control word 0111 8-bit control word 1000 9-bit control word 1001 10-bit control word 1010 11-bit control word 1011 12-bit control word 1100 13-bit control word 1101 14-bit control word 1110 15-bit control word 1111 16-bit control word
11	SRL	W/R	shift register ring This bit is only used for testing, this bit is "1" to output the shift register automatically connected to the input shift register. 0 - normal operating mode 1- Test the operating mode
10	SLV_OE	W/R	Enable from output This bit is only used for the module working in slave mode, when working in master mode The formula is invalid for this bit operation 0 - open from send 1 - Close from send
9:8	TMOD	W/R	transfer mode 00 - send and receive mode 01 - send only 10 - Receive only 11-EEPROM mode
7	SCPOL	W/R	Clock Polarity

			0 - In idle state, SCK remains low; 1 - In idle state, SCK remains high.
6	SCPH	W/R	clock phase 0 - data sampling starts from the first clock edge; 1 - Data sampling starts on the second clock edge.
5:4	FRF	W/R	Protocol Frame Format 00-Motorola SPI 01-Texas Instruments SSP 10 - National Semiconductors Microwire 11-Reserved
3:0	DFS	W/R	data frame size Refer to the table below for the setting values:

0000	reserved
0001	reserved
0010	reserved
0011	4-bit data size
0100	5-bit data size
0101	6-bit data size
0110	7-bit data size
0111	8-bit data size
1000	9-bit data size
1001	10-bit data size
1010	11-bit data size
1011	12-bit data size
1100	13-bit data size
1101	14-bit data size
1110	15-bit data size
1111	16-bit data size

15.4.3 Control Register 1 (CTRLR1)

- **Name: Control Register 1**
- **Size:** 32 bits
- **Address Offset: 0x04**
- **Read/write access:** read/write

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

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

Bit	Name	R/W	Description
31:16	reserved	-	
15:0	NDF	R/W	number of data frames After TMOD = 10 or TMOD = 11, this register is used to set the continuous Number of data frames received.

15.4.4 ENABLE REGISTER (SSIENR)

- **Name: SSI Enable Register**
- Size: 32bits

ÿ **Address Offset: 0x08**
ÿ **Read/write access:** read/write

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

Bit	Name	W/R	Description
31:1	reserved	-	
0	SSIENR	W/R	enable register 0-SPI off 1-SPI open

15.4.5 Microwire Control Register (MWCR)

ÿ **Name: Microwire Control Register**
ÿ **Size:** 32bits
ÿ **Address Offset: 0x0C**
ÿ **Read/write access:** read/write

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

Bit	Name	W/R	Description
31:3	reserved	-	
2	MHS	W/R	Microwire handshake This bit is only valid when the SPI is configured in master mode, slave mode This bit is ignored. This bit is used to enable the Microwire "busy/ready" handshake interface mouth. When the enable is turned on, the SPI sends the last 1bit data or After the control command, the ready state of the remote slave device will be detected.
1	MDD	W/R	Microwire direction control bits 0 - receive external device data 1- Send data to external device
0	MWMOD	W/R	Microwire transmission mode 0 - discontinuous transmission 1 - Continuous transmission

15.4.6 Slave Select Register (SER)

ÿ **Name: Slave Enable Register**
ÿ **Size:** 32bits
ÿ **Address Offset: 0x10**
ÿ **Read/write access:** read/write

31 30 29		28	27	26	25 24 23	twenty two	twenty one	20	19	18	17	16

reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved																s0

Bit	Name	W/R	Description
31:1	reserved	-	
0	s0	W/R	Slave Select Register This register is only valid in SPI master mode, and in slave mode neglect. 0 - Invalid slave device 1- Valid from the device

15.4.7 Baud Rate Register (BAUDR)

- **Name: Baud Rate Select**
- **Size:** 32bits
- **Address Offset: 0x14**
- **Read/write access:** read/write

31	30	29	28	27	26	25	24	23	Reserved	twenty two	twenty one	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SCKDV																

Bit	Name	W/R	Description
31:16	reserved	-	
15:0	SCKDV	W/R	Baud rate register Baud rate = PCLK/SCKDV SCKDV range 2 to 65534 (even numbers only) This register operation is valid when the SPI enable is turned on.

15.4.8 Transmit FIFO Threshold Register (TXFTLR)

- **Name: Transmit FIFO Threshold Level**
- **Size: 32bits**
- **Address Offset: 0x18**
- **Read/Write Access: Read/Write**

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

Bit	Name	W/R	Description
31:4	reserved	-	
3:0	TXFTLR	W/R	Transmit FIFO full interrupt threshold Refer to the table below for the setting values: 0 The amount of data in the transmit FIFO is 0, triggering an interrupt

			1 Data in transmit FIFO is less than 1 trigger interrupt
			2 Data in transmit FIFO is less than 2 trigger interrupt
			3 Data in transmit FIFO is less than 3 trigger interrupt
			4 Data in transmit FIFO is less than 4 trigger interrupt
			5 Data in transmit FIFO is less than 5 trigger interrupt
			6 Data in transmit FIFO is less than 6 trigger interrupt
			7 Data in transmit FIFO is less than 7 trigger interrupt
			8 Data in transmit FIFO is less than 8 trigger interrupt
			9 Data in transmit FIFO is less than 9 trigger interrupt
			10 Less than 10 data in transmit FIFO trigger interrupt
			11 Less than 11 data in transmit FIFO trigger interrupt
			12 Data in transmit FIFO is less than 12 trigger interrupt
			13 Less than 13 data in transmit FIFO trigger interrupt
			14 Less than 14 data in transmit FIFO trigger interrupt
			15 Less than 15 data in transmit FIFO trigger interrupt

15.4.9 Receive FIFO Threshold Register (RXFTLR)

- **Name: Receive FIFO Threshold Level**
- **Size: 32bits**
- **Address Offset: 0x1C**
- **Read/Write Access: Read/Write**

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

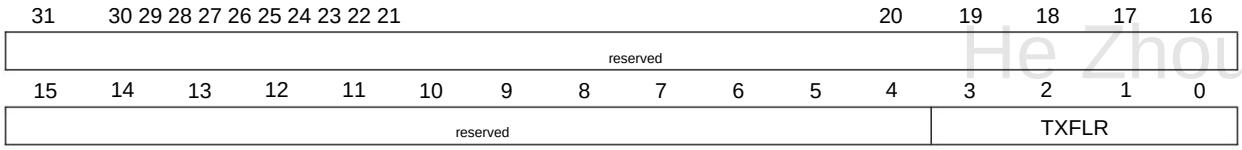
Bit	Name	W/R	Description
31:4	reserved	-	
3:0	RXFTLR	W/R	Receive FIFO Empty Interrupt Threshold Refer to the table below for the setting values:

0 Receive 1 or more data in the FIFO to trigger an interrupt
1 Receive 2 or more data in the FIFO to trigger an interrupt
2 Receive 3 or more data in the FIFO to trigger an interrupt
3 Receive 4 or more data in the FIFO to trigger an interrupt
4 Receive 5 or more data in the FIFO to trigger an interrupt
5 Receive 6 or more data in the FIFO to trigger an interrupt
6 Receive 7 or more data in the FIFO to trigger an interrupt
7 Receive 8 or more data in FIFO to trigger interrupt
8 Receive 9 or more data in FIFO to trigger interrupt
9 Receive 10 or more data in the FIFO to trigger an interrupt
10 Receive 11 or more data in the FIFO to trigger an interrupt
11 Receive 12 or more data in the FIFO to trigger an interrupt
12 Receive 13 or more data in the FIFO to trigger an interrupt
13 Receive 14 or more data in the FIFO to trigger an interrupt
14 Receive 15 or more data in the FIFO to trigger an interrupt
15 Receive data in FIFO 16 trigger interrupts

15.4.10 Transmit FIFO Data Volume Register (TXFLR)

- **Name: Transmit FIFO Level Register**
- **Size: 32bits**
- **Address Offset: 0x20**
- **Read/Write Access: Read**

The register value increases when data is written to the FIFO, and decreases when the SPI fetches data from the FIFO.

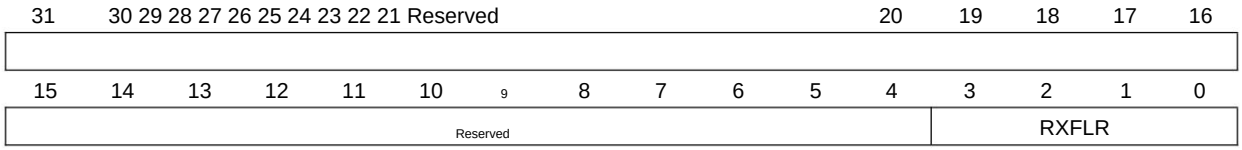


Bit	Name	W/R	Description
31:4	reserved	-	
3:0	TXFLR	R	transmit FIFO data amount Number of valid data contained in the transmit FIFO

15.4.11 Receive FIFO Data Volume Register (RXFLR)

- **Name: Receive FIFO Level Register**
- **Size: 32bits**
- **Address Offset: 0x20**
- **Read/Write Access: Read**

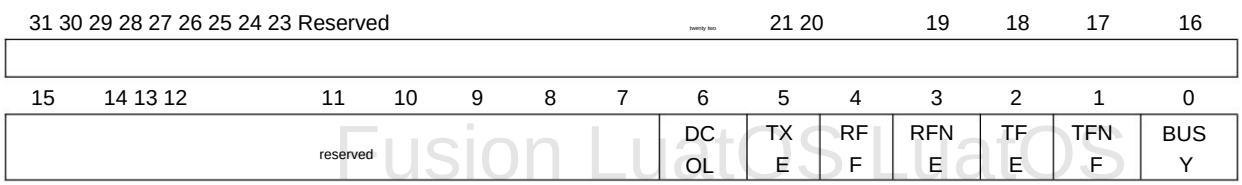
The register value increases when data is written to the FIFO, and decreases when the SPI fetches data from the FIFO.



Bit	Name	W/R	Description
31:4	reserved	-	
3:0	RXFLR	R	Receive FIFO data volume Number of valid data contained in the receive FIFO

15.4.12 Status Register (SR)

- **Name: Status Register**
- **Size: 32 bits**
- **Address Offset: 0x28**
- **Read/Write Access: Read**



Bit	Name	W/R	Description
31:7	reserved	-	
6	DCOL	R	data collision error 0 - no error 1-Transfer data collision error Reading the register clears this bit.
5	TXE	R	transmission error 0 - no error 1 - Transmission error Reading the register clears this bit.
4	RFF	R	Receive FIFO full 0 - Receive FIFO is not full 1 - Receive FIFO full When the FIFO is not full, it is automatically cleared to "0" by hardware.
3	RFNE	R	The receiving FIFO is not empty. 0 - Receive FIFO empty 1 - The receive FIFO is not empty Read FIFO by software Clear "0"
2	TFE	R	Transmit FIFO empty 0 - Transmit FIFO not empty 1 - Transmit FIFO empty When the FIFO is not empty, it will be automatically cleared to "0" by
1	TFNF	R	hardware. The transmit FIFO is not full. 0 - Transmit FIFO full 1 - Transmit FIFO is not full When the FIFO is full, it is automatically cleared by hardware to
0	BUSY	R	"0" busy status 0-SPI in EDLE or off state 1-SPI is in the active state of transmitting data

15.4.13 Interrupt Mask Register (IMR)

- **Name: Interrupt Mask Register**
- **Size: 32bits**
- **Address Offset: 0x2C**
- **Read/Write Access: read/write**

Used to mask or enable SPI interrupt sources.

31	30	29	28	27	26	25	twenty four
reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
15	14	13	12	11	10	9	8
reserved							
7	6	5	4	3	2	1	0

reserved	MSTIM	RXFIM	RXOIM	RXUIM	TXOIM	TXEIM			

Bit	Name	W/R	Description
31:6	reserved	-	
5	MSTIM	W/R	Multi-master contention interrupt masking 0 - Disable multi-master contention interrupts 1-Allow multi-master contention interrupts
4	RXFIM	W/R	Receive FIFO full interrupt mask 0 - disable receive FIFO full interrupt 1 - Enable receive FIFO full interrupt
3	RXOIM	W/R	Receive FIFO overflow interrupt mask 0 - disable receive FIFO overflow interrupt 1 - Enable receive FIFO overflow interrupt
2	RXUIM	W/R	Receive FIFO underflow interrupt mask 0 - disable receive FIFO underflow interrupt 1 - Enable receive FIFO underflow interrupt
1	TXOIM	W/R	Transmit FIFO Overflow Interrupt Mask 0 - disable transmit FIFO overflow interrupt 1 - Enable transmit FIFO overflow interrupt
0	TXEIM	W/R	Transmit FIFO Empty Interrupt Mask 0 - disable transmit FIFO empty interrupt 1 - Enable transmit FIFO empty interrupt

15.4.14 Interrupt Status Register (ISR)

- **Name: Interrupt Status Register**
- **Size: 32bits**
- **Address Offset: 0x30**
- **Read/Write Access: read**

31	30	29	28	27	26	25	twenty four
reserved							

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							

15	14	13	12	11	10	9	8
reserved							

7	6	5	4	3	2	1	0
reserved		MSTIS	RXFIS	RXOIS	RXUIS	TXOIS	TXEIS

Bit	Name	W/R	Description
31:6	reserved	-	
5	SSTIS	R	Multimaster contention interrupt status 0 - no multimaster contention interrupt is generated 1- Generate a multi-master contention interrupt
4	RXFIS	R	Receive FIFO full interrupt status 0 - no receive FIFO full interrupt is generated 1 - Generate receive FIFO full interrupt
3	RXOIS	R	Receive FIFO overflow interrupt status 0 - no receive FIFO overflow interrupt is generated 1 - generate receive FIFO overflow interrupt
2	RXUIS	R	Receive FIFO underflow interrupt status

			0 - no receive FIFO underflow interrupt is generated 1 - Generate receive FIFO underflow interrupt
1	TXOIS	R	Transmit FIFO overflow interrupt status 0 - no transmit FIFO overflow interrupt is generated 1 - Generate transmit FIFO overflow interrupt
0	TXEIS	R	Transmit FIFO Empty Interrupt Status 0 - no transmit FIFO empty interrupt is generated 1 - Generate transmit FIFO empty interrupt

15.4.15 Original Interrupt Status Register (ISR)

- **Name: Raw Interrupt Status Register**
- **Size: 32bits**
- **Address Offset: 0x34**
- **Read/Write Access: read**

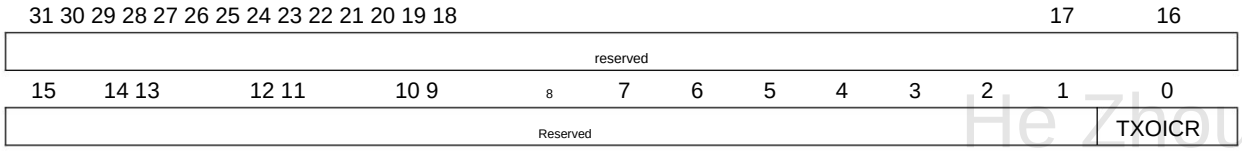
The state value of this register is different from the Interrupt Status Register (ISR) in that the value of this register is not affected by the Interrupt Mask Register (IMR) control.

31	30	29	28	27	26	25	twenty four
reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
15	14	13	12	11	10	9	8
reserved							
7	6	5	4	3	2	1	0
reserved		MSTIR	RXFIR	RXOIR RXUIR	TXOIR		TXEIR

Bit	Name	W/R	Description
31:6	reserved	-	
5	SSTIR	R	Multimaster contention interrupt status 0 - no multimaster contention interrupt is generated 1- Generate a multi-master contention interrupt
4	RXFIR	R	Receive FIFO full interrupt status 0 - no receive FIFO full interrupt is generated 1 - Generate receive FIFO full interrupt
3	RXOIR	R	Receive FIFO overflow interrupt status 0 - no receive FIFO overflow interrupt is generated 1 - Generate receive FIFO overflow interrupt
2	RXUIR	R	Receive FIFO underflow interrupt status 0 - no receive FIFO underflow interrupt is generated 1 - Generate receive FIFO underflow interrupt
1	TXOIR	R	Transmit FIFO overflow interrupt status 0 - no transmit FIFO overflow interrupt is generated 1 - Generate transmit FIFO overflow interrupt
0	TXEIR	R	Transmit FIFO Empty Interrupt Status 0 - no transmit FIFO empty interrupt is generated 1 - Generate transmit FIFO empty interrupt

15.4.16 Transmit FIFO Overrun Interrupt Clear Register (TXOICR)

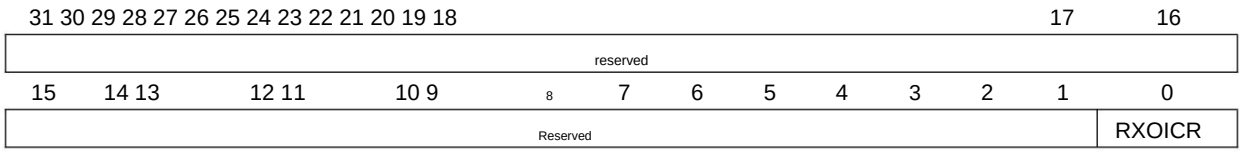
- **Name: Transmit FIFO Overflow Interrupt Clear Register**
- **Size: 32bits**
- **Address Offset: 0x38**
- **Read/Write Access: read**



Bit	Name	W/R	Description
31:1	reserved	-	
0	TXOICR	R	Transmit FIFO overflow interrupt clear Clear interrupt on read operation

15.4.17 Receive FIFO Overrun Interrupt Clear Register (RXOICR)

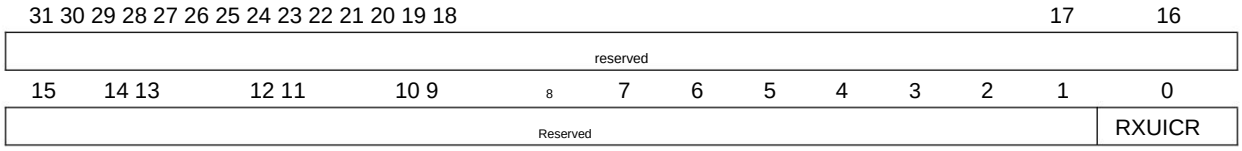
- **Name: Receive FIFO Overflow Interrupt Clear Register**
- **Size: 32bits**
- **Address Offset: 0x38**
- **Read/Write Access: read**



Bit	Name	W/R	Description
31:1	reserved	-	
0	RXOICR	R	Receive FIFO overflow interrupt clear Clear interrupt on read operation

15.4.18 Receive FIFO Underflow Interrupt Clear Register (RXUICR)

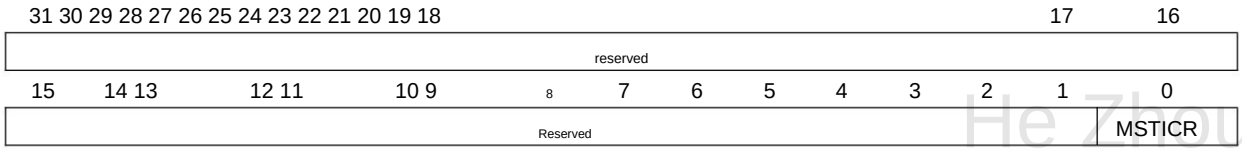
- **Name: Receive FIFO Underflow Interrupt Clear Register**
- **Size: 32bits**
- **Address Offset: 0x40**
- **Read/Write Access: read**



Bit	Name	W/R	Description
31:1	reserved	-	
0	RXUICR	R	Receive FIFO underflow interrupt clear Clear interrupt on read operation

15.4.19 Multi-Master Contention Interrupt Clear Register (MSTICR)

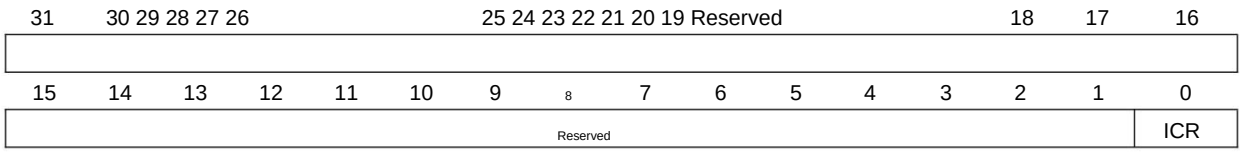
- **Name: Multi-Master Interrupt Clear Register**
- **Size: 32bits**
- **Address Offset: 0x44**
- **Read/Write Access: read**



Bit	Name	W/R	Description
31:1	reserved	-	
0	MSTICR	R	Multi-master contention interrupt clear Clear interrupt on read operation

15.4.20 Global Interrupt Clear Register (ICR)

- **Name: Interrupt Clear Register**
- **Size: 32bits**
- **Address Offset: 0x48**
- **Read/Write Access: read**

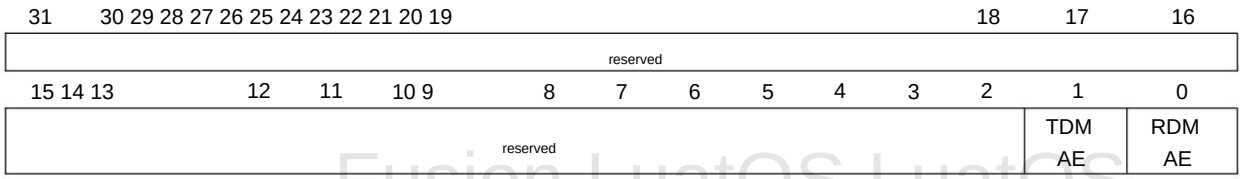


Bit	Name	W/R	Description
31:1	reserved	-	
0	ICR	R	global interrupt clear Clears the above 4 interrupt states by reading it

15.4.21 DMA Control Register (DMACR)

- **Name: DMA Control Register**
- **Size: 32 bits**
- **Address Offset: 0x4C**
- **Read/Write Access: read/write**

Operations on this register are not affected by the SPI enable.

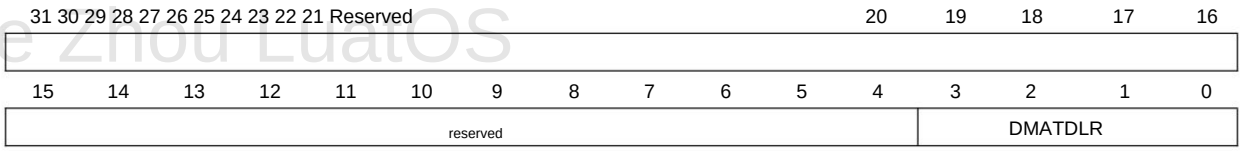


Bit	Name	W/R	Description

31:2	reserved	-	
1	TDMAE	W/R	Transmit DMA enable 0 - send DMA off 1 - Transmit DMA on
0	RDMAE	W/R	Receive DMA enable 0 - receive DMA off 1 - Receive DMA on

15.4.22 DMA Transmit Data Threshold Register (DMATDLR)

- **Name: DMA Transmit Data Level**
- **Size: 32bits**
- **Address Offset: 0x50**
- **Read/Write Access: Read/Write**



Bit	Name	W/R	Description
31:4	reserved	-	
3:0	DMATDLR	W/R	DMA transmit threshold When the amount of data in the transmit FIFO is equal to or less than the DMA transmit threshold, SPI will request DMA to request DMA to send FIFO to SPI write data in. Refer to the table below for the setting values:

0	The amount of data in the transmit FIFO is 0
1	The data in the transmit FIFO is less than or equal to 1
2	The data in the transmit FIFO is less than or equal to 2
3	The data in the transmit FIFO is less than or equal to 3
4	The data in the transmit FIFO is less than or equal to 4
5	The data in the transmit FIFO is less than or equal to 5
6	The data in the transmit FIFO is less than or equal to 6
7	The data in the transmit FIFO is less than or equal to 7
8	The data in the transmit FIFO is less than or equal to 8
9	The data in the transmit FIFO is less than or equal to 9
10	The data in the transmit FIFO is less than or equal to 10
11	The data in the transmit FIFO is less than or equal to 11
12	The data in the transmit FIFO is less than or equal to 12
13	The data in the transmit FIFO is less than or equal to 13
14	The data in the transmit FIFO is less than or equal to 14
15	The data in the transmit FIFO is less than or equal to 15

15.4.23 DMA Receive Data Threshold Register (DMARDLR)

- **Name: DMA Receive Data Level**
- **Size: 32bits**
- **Address Offset: 0x54**

ÿ **Read/Write Access: Read/Write**

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

Bit	Name	W/R	Description
31:4	reserved	-	
3:0	DMARDLR	W/R	DMA receive threshold When the amount of data in the receive FIFO is equal to or greater than the DMA transmit threshold, The SPI will send a request to the DMA, requesting the DMA to take out the SPI and receive FIFO data. Refer to the table below for the setting values:

0	The data in the receive FIFO is 1 or more
1	The data in the receive FIFO is 2 or more
2	The data in the receive FIFO is 3 or more
3	The data in the receive FIFO is 4 or more
4	The data in the receive FIFO is 5 or more
5	The data in the receive FIFO is 6 or more
6	The data in the receive FIFO is 7 or more
7	The data in the receive FIFO is 8 or more
8	The number of data in the receive FIFO is 9 or more
9	The data in the receive FIFO is 10 or more
10	The data in the receive FIFO is 11 or more
11	The data in the receive FIFO is 12 or more
12	The data in the receive FIFO is 13 or more
13	The number of data in the receive FIFO is 14 or more
14	The data in the receive FIFO is 15 or more
15	The data in the receive FIFO is 16

15.4.24 Data Register (DR)

ÿ **Name: Data Register**
ÿ **Size: 32bits**
ÿ **Address Offset: 0x60**
ÿ **Read/Write Access: read/write**

The data register is a 16bit wide read and write buffer. When a read operation is performed on it, the data is taken out through the receive FIFO. when the During a write operation, data is written into the transmit FIFO. Writes are only possible when SSI_EN = 1.

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

Bit	Name	W/R	Description
31:16	reserved	-	
15:0	DR	W/R	SPI Data Register

			When writing data, the data must be right-aligned data, when reading data Data is automatically right-aligned. Read = receive data FIFO Write = transmit data FIFO

15.4.25 Receive Sample Delay Register (RX_SAMPLE_DLY)

- ÿ **Name: Rx Sample Delay Register**
- ÿ **Size: 32bits**
- ÿ **Address Offset: 0xfc**
- ÿ **Read/Write Access: read/write**

This register is used to delay the sampling point time backward from the standard receive sampling time point. The delay time is measured in PCLK cycles. exist
This register can be read and written after SPI is enabled.

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

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

Bit	Name	W/R	Description
31:8	reserved	-	
7:0	RSD	W/R	Sample Delay Register This register is used to delay the standard receive sampling time point Late sampling point time. Delay time is in units of PCLK cycles bit.

16 high-speed SPI interface

16.1 Introduction to High Speed SPI Interface

This module is a high-speed SPI Master interface module that supports standard SPI

16.2 Main Features of High Speed SPI Interface

ÿ The high-speed SPI clock is provided by FCLK, that is, HSPI_CLK = FCLK ÿ
Supports up to 32 frequency division ÿ Supports the protocol Motorola Serial Peripheral
Interface (SPI) ÿ Contains hardware implementation of transceiver FIFO, with a depth
of 64 ÿ Independent hardware transceiver FIFO, can be configured with transceiver
FIFO interrupt threshold ÿ Support full-duplex, half-duplex mode ÿ Multiple transceivers,
error interrupt detection ÿ DMA support

16.3 High-speed SPI interface function description

16.3.1 Receive/Transmit FIFO

HSPI contains 2 independent 64x8bit receive and transmit FIFOs.

The CPU writes the register DR, the data is written into the transmit FIFO, the CPU reads the register DR, and the data is fetched from the receive FIFO out.

The receive and transmit FIFOs have independent interrupt threshold settings. When the data meets the set threshold, the SPI sends an interrupt request to the CPU.

The receive and transmit FIFOs have independent DMA threshold settings. When the data meets the set threshold, the SPI sends out the corresponding DMA request.

16.3.2 Interrupt Types and Interrupt Flags

The HSPI master supports the following interrupt types:

ÿ Tramsmit Done Interrupt

Send complete interrupt, when all the data in the transmit FIFO is sent out, an interrupt is generated.

ÿ Transmit Interrupt

Transmit FIFO interrupt, which is generated when the amount of data in the transmit FIFO is less than the set lower threshold or greater than the set upper threshold break.

ÿ Receive Interrupt

Receive FIFO interrupt, generated when the amount of data in the receive FIFO is less than the set lower threshold or greater than the set upper threshold break.

HSPI master interrupt flag:

ÿ Transmit FIFO Arrive Full Interrupt

Send FIFO will be full flag, when the amount of data in the send FIFO is greater than the set upper threshold, the send FIFO will be full flag will be generated.

ÿ Transmit FIFO Full Interrupt

Transmit FIFO full flag, when the amount of data in the transmit FIFO is greater than the set upper threshold, and the amount of data in the transmit FIFO Greater than the FIFO depth, the transmit FIFO full flag is generated.

ÿ Transmit FIFO Arrive Empty Interrupt

The sending FIFO will be empty flag, when the amount of data in the sending FIFO is less than the set lower limit threshold, the sending FIFO will be empty flag will be generated.

ÿ Transmit FIFO Empty Interrupt

Transmit FIFO empty flag, when the amount of data in the transmit FIFO is less than the set lower threshold, and the amount of data in the transmit FIFO When 0, the transmit FIFO empty flag is generated.

ÿ Receive FIFO Arrive Full Interrupt

Receiving FIFO will be full flag, when the amount of data in the receiving FIFO is greater than or equal to the set upper threshold, it will generate the receiving FIFO will be full logo.

ÿ Receive FIFO Full Interrupt

Receiving FIFO full flag, when the amount of data in the receiving FIFO is greater than the set upper threshold, and the amount of data in the receiving FIFO is large When the FIFO depth is reached, the receive FIFO full flag is generated.

ÿ Receive FIFO Arrive Empty Interrupt

The receiving FIFO will be empty flag, when the amount of data in the receiving FIFO is less than the set lower limit threshold, the receiving FIFO will be empty flag will be generated.

ÿ Receive FIFO Empty Interrupt

Receiving FIFO empty flag, when the amount of data in the receiving FIFO is less than the set lower threshold, and the amount of data in the receiving FIFO When it is 0, the receive FIFO empty flag is generated.

16.4 Register Description

16.4.1 Address Mapping Table

address range	Base	Peripherals	bus
0x400A_3000-0x400A_3FFF	address 0x400A_3000	SPIM5	AHB

Table 16- 1SPIM5 register table

offset address	Register name width (bit)		reset value
0x20	CTRL0	32	0x01000000
0x24	reserved	32	0x00000000
0x28	FLOW_STATUS	32	0x00000000
0x2C	FIFO_CTRL	32	0x003f003f
0x30	RX_DATA	32	0x00000000
0x34	TX_DATA	32	0x00000000
0x38	STATUS	32	0x00000606
0x3C	CTRL1	32	0x00000000

0x40	FIFO_STATUS	32	0x00000000
0x44	DMA_CTRL	32	0x00000000
0x48	TX_INT	32	0x00000000
0x4C	RX_INT	32	0x00000000

16.4.2 Control Register 0 (CTRL0)

31	30	29	28	27	26	25	twenty four
reserved			sample_edge_sel			default_output	master_enable
ro			rw			rw	rw
twenty three	twenty two	twenty one	20	19	18	17	16
reserved			rx_dma_en	reserved			tx_dma_en
ro			rw	ro			rw
15	14	13	12	11	10	9	8
rx_done_int_en	tx_done_int_en	int_en	modsel			lsbfe	cpol
rw	rw	rw	rw			rw	rw
7	6	5	4	3	2	1	0
cpha	reserved				mdiv_en	tx_enable	master_busy
rw	ro				rw	rw	rc

Bit	Name	W/R	Description
31-30	reserved	-- reserved bit	
29:26	sample_edge_sel	RW	0: Receive data at the normal sampling edge 1: Receive data after the normal sampling edge delay 1 cycle 2: Receive data after the normal sampling edge delay 2 cycle
25	default_output	RW The default value when sending the first word	
24	master_enable	RW master enabled.	
23:21	reserved	-- reserved bits	
20	rx_dma_en	RW receive dma enable switch	
19:17	reserved	-- reserved bits	
16	tx_dma_en	RW send dma enable switch	
15	rx_done_int_en	RW	receive interrupt enable 1: When the received fifo depth is greater than rx_af_th or less than rx_ae_th, spi interrupt will be generated 0: No interrupt is generated
14	tx_done_int_en	RW	Transmit interrupt enable 1: When the sending fifo depth is greater than tx_af_th or less than tx_ae_th, spi interrupt will be generated 0: No interrupt is generated
13	int_en	RW	spi interrupt enable signal 0: No interrupt is generated 1: Generated when spi_master ends or an error occurs in tx/rx interrupt signal
12:10	modsel	RW	Mode selection: 3'b000/001: Single-wire full-duplex mode.
9	lsbfe	RW	endian selection 0: Little endian, left high and right low. 1: big endian

8	cpol	RW polarity selection	
7	cpha	RW phase selection	
6:3		RO reserved bit	
2	reserved mdiv_en	RW Frequency division enable signal. When not enabled, the default frequency is divided by 2.	
1	tx_enable	RW	tx enable switch, after it is turned on, spi_master can turn on the TX function can.
0	busy	RC is currently configured with the CTRL0 register flag bit, read the register.	

16.4.3 Status Register (FLOW_STATUS)

31	30	29	28	27	26	25	twenty four
reserved							
ro							

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
Reserved							spi_done
ro							rc

Bit	Name	W/R	Description
31-1		-- reserved bit	
0	reserved spi_done	RC SPI transfer ends	

16.4.4 FIFO Control Register (FIFO_CTRL)

31	30	29	28	27	26	25	twenty four
reserved			rx_ae_th				
ro			rw				

twenty three	twenty two	twenty one	20	19	18	17	16
reserved			rx_af_th				
ro			rw				

15	14	13	12	11	10	9	8
reserved			tx_ae_th				
ro			rw				

6 7 tx_fifo_clr		5	4	3	2	1	0
rx_fifo_clr		tx_af_th					
rw	rw	rw					

Bit	Name	W/R	Description
31-30		-- reserved bit	
29-24	Reserved	RW Receive FIFO will be empty threshold	
23-22	rx_ae_th	-- reserved bits	
21-16	Reserved	RW Receive FIFO will be full threshold	
15-14	rx_af_th	-- reserved bits	
13-8	Reserved	RW Transmit FIFO will be empty threshold	
7	tx_ae_th	RW Write 1 to clear the transmit FIFO	
6	tx_fifo_clr	RW Write 1 to clear the receive FIFO	
5-0	rx_fifo_clr tx_af_th	RW Transmit FIFO will be full threshold	

16.4.5 Receive FIFO Read Register (RX_DATA)

31	30	29	28	27	26	25	twenty four
rdata							
ro							
twenty three	twenty two	twenty one	20	19	18	17	16
rdata							
ro							
15	14	13	12	11	10	9	8
rdata							
ro							
7	6	5	4	3	2	1	0
rdata							
ro							

Bit	Name	W/R	Description
31-0	rdata	RO SPI	Master read FIFO data entry

16.4.6 Transmit FIFO Read Register (TX_DATA)

31	30	29	28	27	26	25	twenty four
tdata							
wo							
twenty three	twenty two	twenty one	20	19	18	17	16
tdata							
wo							
15	14	13	12	11	10	9	8
tdata							
wo							
7	6	5	4	3	2	1	0
tdata							
wo							

Bit	Name	W/R	Description
31-0	tdata	WO SPI	Master write FIFO data entry

16.4.7 Status Register (STATUS)

31	30	29	28	27	26	25	twenty four
reserved							
ro							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved		push_af_rx	push_full_rx	push_error_rx	pop_empty_rx	pop_ae_rx	pop_error_rx
ro		ro	ro	ro	ro	ro	ro

7	6	5	4	3	2	1	0
reserved		push_af_tx	push_full_tx	push_error_tx	pop_empty_tx	pop_ae_tx	pop_error_tx
ro		ro	ro	ro	ro	ro	ro

Bit	Name	W/R	Description
31-14		--	reserved bit
13	reserved	RO	receiving fifo push reaches the threshold
12	push_af_rx	RO	receives fifo push full
11	push_full_rx	RO	receives fifo push error
10	push_error_rx	RO	receives fifo and sends empty
	pop_empty_rx pop_ae_rx	RO	receive fifo sending reaches threshold
	pop_error_rx	RO	receive fifo error
9 8 7-6	reserved	--	reserved bits
5	push_af_tx	RO	sending fifo push reaches the threshold
4	push_full_tx	RO	sends fifo push full
3	push_error_tx	RO	send fifo push error
2	pop_empty_tx	RO	send fifo send empty
1	pop_ae_tx	RO	send fifo send reaches threshold
0	pop_error_tx	RO	send fifo error

16.4.8 Control Register 1 (CTRL1)

31	30	29	28	27	26	25	twenty four
reserved							
ro							

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro						rw	

15	14	13	12	11	10	9	8
Reserved	freq_div					rx_dlen	
	rw					rw	

7	6	5	4	3	2	1	0
rx_dlen							
rw							

Bit	Name	W/R	Description
31-15	reserved	--	reserved bit
14-10	freq_div	RW	Clock divider: 0,1: divide by 2 2: divide by 4 3: divide by 6 4: divide by 8 ... 16: 32 division ... Divider = freq_div * 2.
9-0	rx_dlen	RW	The master receives the data length, 0: 1 byte 1: 2 bytes 2: 3 bytes 3: 4 bytes 4: 5 bytes ... >=127: 128 bytes

16.4.9 FIFO Status Register (FIFO_STATUS)

31	30	29	28	27	26	25	twenty four
reserved							
ro							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							
15	14	13	12	11	10	9	8
reserved				rx_pop_depth			
ro				ro			
	6	5	4	3	2	1	0
7 rx_pop_de pth				tx_push_depth			
ro				ro			

Bit	Name	W/R	Description
31-14		--	reserved bit
13-7	reserved	RO	The amount of data in the receive FIFO
6-0	rx_pop_depth tx_push_depth	RO	transmits the amount of data in the FIFO

16.4.10 DMA Control Register (DMA_CTRL)

31	30	29	28	27	26	25	twenty four
reserved							
ro							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							
15	14	13	12	11 10 rx_fifo_thr		9	8
reserved							
ro				rw			
	6	5	4		2	1	0
7 rx_fifo_thr				3 tx_fifo_thr			
rw				rw			

Bit	Name	W/R	Description
31:14		--	reserved bit
13:7	Reserved	RW	DMA access to receive FIFO threshold
6:0	rx_fifo_thr tx_fifo_thr	RW	DMA access transmit FIFO threshold

16.4.11 Transmit Interrupt Register (TX_INT)

31	30	29	28	27	26	25	twenty four
reserved							
ro							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							
15	14	13	12	11	10	9	8

reserved
ro

7	6	5	4	3	2	1	0
reserved				tx_full_int	tx_af_int	tx_empty_ int	tx_ae_int
ro				rc	rc	rc	rc

Bit	Name	W/R	Description
31:4		--	reserved bit
3	Reserved	RC	transmit FIFO full interrupt
2	tx_full_int	RC	transmit FIFO will be full interrupt
1	tx_af_int	RC	transmit FIFO empty interrupt
0	tx_empty_int tx_ae_int	RC	transmit FIFO will be empty interrupt

16.4.12 Receive Interrupt Register (RX_INT)

31	30	29	28	27	26	25	twenty four
reserved							
ro							

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
reserved				rx_full_int	rx_af_int	rx_empty_ int	rx_ae_int
ro				rc	rc	rc	rc

Bit	Name	W/R	Description
31:4		--	reserved bit
	Reserved	RC	receive FIFO full interrupt
3 2	rx_full_int	RC	receive FIFO will be full interrupt
1	rx_af_int	RC	receive FIFO empty interrupt
0	rx_empty_int rx_ae_int	RC	receive FIFO will be empty interrupt

17 DMA Controller (DMAC)

17.1 Introduction to DMA

Direct Memory Access (DMA) is used to provide high-speed data transfers between peripherals and memory or between memory and memory lose. Data can be moved quickly through DMA without CPU intervention, which saves CPU resources for other operations.

The DMA controller has 8 channels, each channel is dedicated to managing memory-to-memory, memory-to-peripheral, and peripheral-to-memory accesses. beg. Each channel has independent priority settings.

17.2 Main Features of DMA

ÿ 8 independent configurable channels ÿ 8 independent hardware DMA requests (DMA hard handshake interface), each channel also supports software start (DMA soft handshake interface) Interface)

ÿ 8 channels independent priority setting ÿ Support data transmission of multiple widths ÿ Each channel has its own independent interrupt signal and mark Multi-Block Transmission

17.3 Peripheral Types

Peripherals that use DMA transfers can be divided into two categories: ÿ Memory peripherals ÿ Non-memory peripherals

Memory peripherals:

The "DMA handshake interface" is not required between the memory peripheral and the DMA controller, so when the DMA channel is enabled, the DMA does not require a request signal to start the DMA transfer. At the same time, because there is no handshake interface, SRC_MSIZE and DEST_MSIZE have unlimited effects on memory peripherals.

Non-memory peripherals:

Non-memory peripherals, such as: UART, I2C, SPI, etc., all contain request signals for direct communication with DMA, that is, DMA handshake interface. When a non-memory peripheral performs DMA transfer, it will control the data transfer through a direct handshake interface with the DMA. DMA interface is divided into two types: hard handshake interface and soft handshake interface

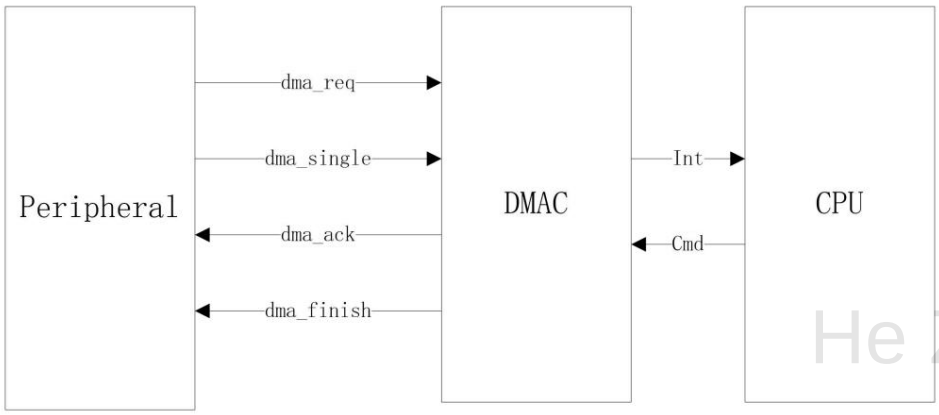
17.4 DMA Handshake Interface

DMA provides two DMA request handshake interfaces: ÿ Hard handshake interface ÿ Soft handshake interface Users can configure each channel register to select the handshake interface mode through the DMA register.

17.4.1 DMA Hard Handshake Interface

After the user configures the corresponding peripheral DMA enable, configures the corresponding trigger conditions, and configures the corresponding DMA channel as a hard handshake signal, the DMA request Seeking signal triggering will be done automatically by hardware according to conditions.

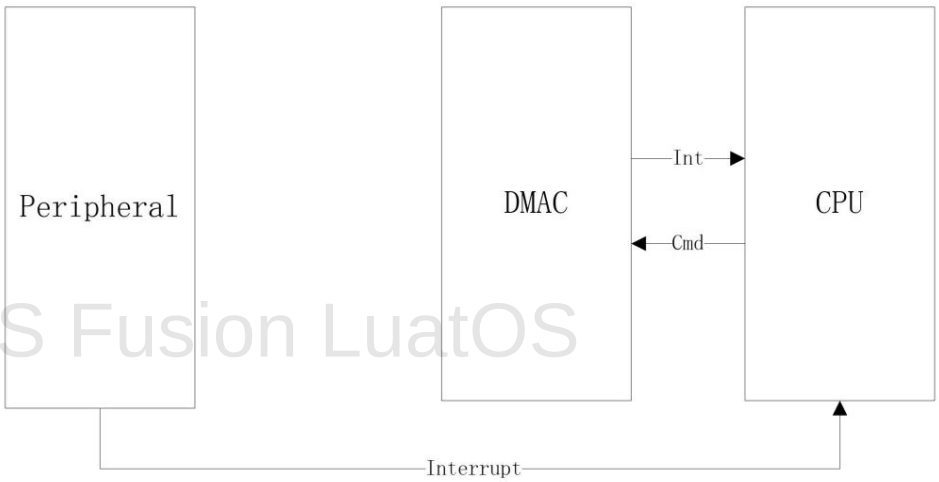
The DMA controller hard handshake interface communicates with the corresponding peripherals through five signals, dma_req, dma_single, dma_ack and dma_finish. interact. All peripherals with DMA function in Air105 include the following interface signals. .



Name	Description
dma_req	Burst transaction request from peripheral The peripheral sends a burst transfer request to the DMA
dma_single	Single transfer status Peripheral single transfer status
dma_ack	DMA reply signal DMA completes 1 transfer on the AHB bus (Burst or single) Rear peripheral response signal
dma_finish	DMA transfer complete Peripheral response signal after DMA completes block (BLOCK) transfer

17.4.2 DMA Soft Handshake Interface

When the peripheral has no DMA interface or the user does not use the peripheral interface, the DMA soft handshake interface can be used to operate DMA data transfer. Register interface and operation details "soft handshake interface register" description.



ÿ DMA request register: ReqSrcReg/ReqDstReg

Generate a DMA request, such as ReqSrcReg[x] is set to 1, that is, the source device generates a request for DMA channel x

ÿ DMA single request register: SglReqSrcReg/SglReqDstReg

Set the DMA request type, such as ReqSrcReg[x] is 0, the source device generates a "burst" transfer request for DMA channel x, is 1, the source device generates a "single" transfer request for DMA channel x

ÿ DMA last request register: LstSrcReg/LstDstReg

Set the last DMA request of this block transfer. For example, LstSrcReg[x] is 1, indicating that this DMA request is the source device. standby last request to DMA channel x

ÿ Register writing sequence:

Before writing ReqSrcReg/ReqDstReg, ensure that SglReqSrcReg/SglReqDstReg and LstSrcReg/LstDstReg have been written correctly

Example of register usage: ÿ The source device generates 1 "burst" data transfer to DMA channel 2:

```
SglReqSrcReg[2] = 0;
LstSrcReg[2] = 0;
ReqSrcReg[2] = 1;
```

ÿ The source device generates 1 "single" data transfer to DMA channel 2:

```
SglReqSrcReg[2] = 1;
LstSrcReg[2] = 0;
ReqSrcReg[2] = 1;
```

ÿ The source device generates the last "single" data transfer for DMA channel 2 in this DMA block transfer

```
SglReqSrcReg[2] = 1;
LstSrcReg[2] = 1;
ReqSrcReg[2] = 1;
```


17.5 DMA Interrupts

17.5.1 Interrupt Types

Each channel of the DMA contains 5 interrupt types:

ÿ Block transfer complete interrupt (IntBlock – Block Transfer Complete Interrupt) After the DMA channel completes the block transfer of data to the target device, the interrupt is set.

ÿ Destination Data Transfer Complete Interrupt (IntDstTran – Destination Transaction Complete Interrupt) The interrupt is set after the DMA channel completes one ("single" or "burst") data transfer to the target device on the AHB bus.

ÿ Error Interrupt (IntErr – Error Interrupt) During a DMA transfer, the interrupt is set when the DMA receives an error acknowledgement from the AHB bus.

ÿ Source data transfer complete interrupt (IntSrcTran – Source Transaction Complete Interrupt) The interrupt is set after the DMA channel has completed a single ("single" or "burst") data transfer to the source device on the AHB bus. ÿ DMA transfer complete interrupt (IntTfr – DMA Transfer Complete Interrupt) After the DMA completes the data transfer to the target device, the interrupt is set.

17.5.2 Interrupt Register

5 groups of interrupt registers correspond to each interrupt type, and the bit in each register corresponds to 1 channel.

ÿ Raw (Raw) interrupt status registers: RawBlock, RawDstTran, RawErr, RawSrcTran, RawTfr DMA interrupt events are saved in the original interrupt status register

ÿ Interrupt status registers: StatusBlock, StatusDstTran, StatusErr, StatusSrcTran, StatusTfr The interrupt status not masked by the interrupt mask register is saved to the interrupt status register

ÿ Interrupt mask register: MaskBlock, MaskDstTran, MaskErr, MaskSrcTran, MaskTfr Enable or disable DMA channel interrupt status

ÿ Interrupt clear register: ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, ClearTfr Write "1" to the interrupt control register to clear the corresponding channel and generate an interrupt

ÿ StatusInt

The interrupt flag bit is the result of the interrupt flag in the interrupt status register group after the "OR" operation.

17.6 Data transfer rules

The three parameters of the DMA register related to the data type and transfer amount are:

Transfer data block size: BLOCK_TS
Burst data transfer size: SRC_MSIZE, DEST_MSIZE
Data bit width: SRC_TR_WIDTH, DST_TR_WIDTH

17.6.1 Memory to memory

The DMA transfer from memory to memory is determined by DMA BLOCK_TS and SRC_TR_WIDTH.

The memory peripheral has no handshake interface, and the data transmission will not be controlled during the data transmission.

Total transfer amount Bytes = BLOCK_TS * (SRC_TR_WIDTH / 8)

17.6.2 Memory to Peripheral/Peripheral to Memory

In the case of memory-to-peripheral/peripheral-to-memory, memory and DMA are directly without handshake interface library, memory-to-DMA and memory-to-memory

The situation is the same from memory to memory.

Data transfer between DMA and non-memory peripherals is affected by the size of the FIFO, so the DMA controls the data transfer.

DMA uses SRC_MSIZE/DEST_MSIZE to limit the amount of data transferred in one "burst" (Burst) transfer.

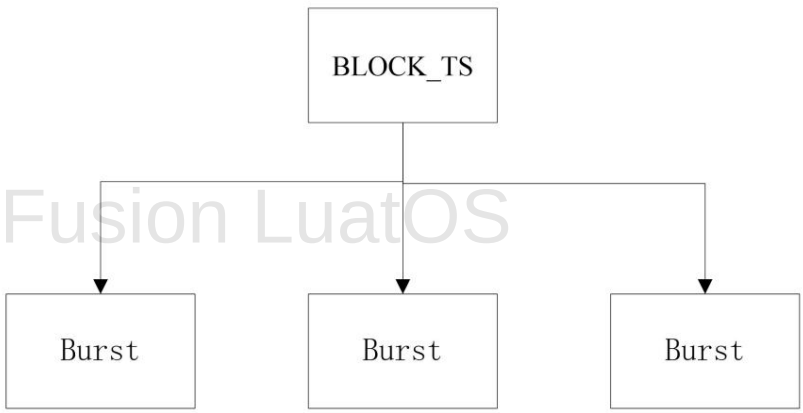
ÿ If DMA BLOCK_TS is an integer multiple of SRC_MSIZE/DEST_MSIZE, only "burst" will be generated during the entire transfer process.

transmission.

Example parameter configuration:

DMA BLOCK_TS = 12
SRC_MSIZE/DEST_MSIZE = 4

According to the above configuration, the DMA transfer control flow is as follows:



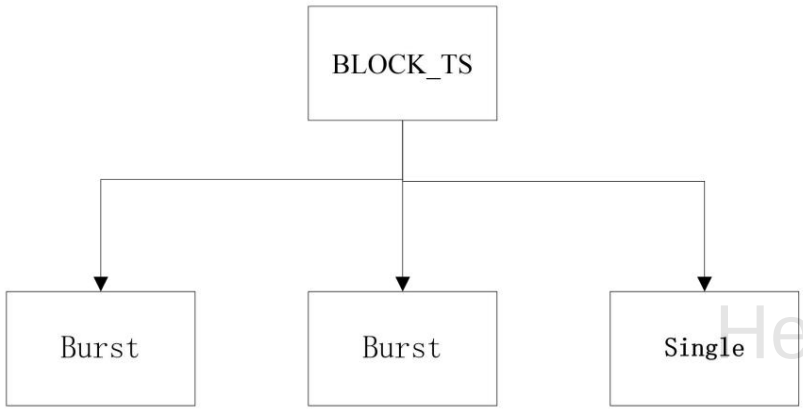
ÿ If DMA BLOCK_TS is not an integer multiple of SRC_MSIZE/DEST_MSIZE, use "single" in the last remaining transfers

"Single" transfer is completed. "Single" transfer only transfers SRC_TR_WIDTH/DST_TR_WIDTH single data at a time

Example parameter configuration:

DMA BLOCK_TS = 12
SRC_MSIZE/DEST_MSIZE = 5
SRC_TR_WIDTH/DST_TR_WIDTH=1

According to the above configuration, the DMA transfer control flow is as follows:



17.6.3 SRC_MSIZE/DEST_MSIZE parameter setting

The SRC_MSIZE/DEST_MSIZE parameters are used in non-memory peripherals.

SRC_MSIZE/DEST_MSIZE are related to non-memory peripheral receive and transmit FIFO settings.

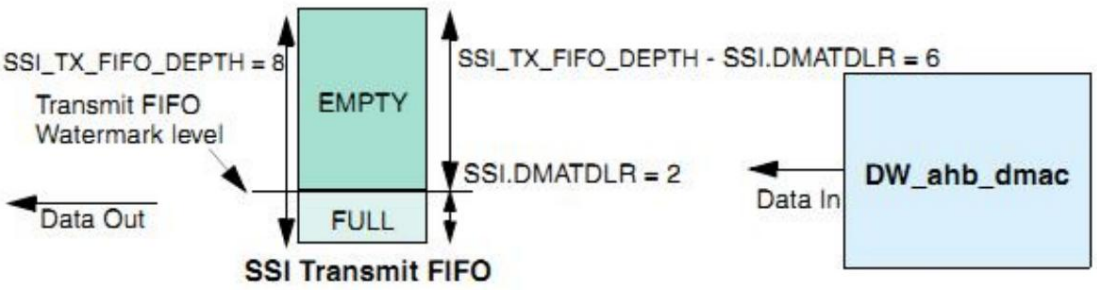
ÿ When the non-memory peripheral is used as the target device, after the data in the non-memory peripheral transmit FIFO reaches the set threshold, it sends the data to the DMA.

When a request is issued, the DMA will write data to the transmit FIFO according to the amount of data in DEST _MSIZE. Send FIFO at this time

The remaining space should be greater than the DEST_MSIZE value to meet the amount of data carried by the DMA channel at one time, if it is less than it will cause

Corresponds to peripheral FIFO overflow error.

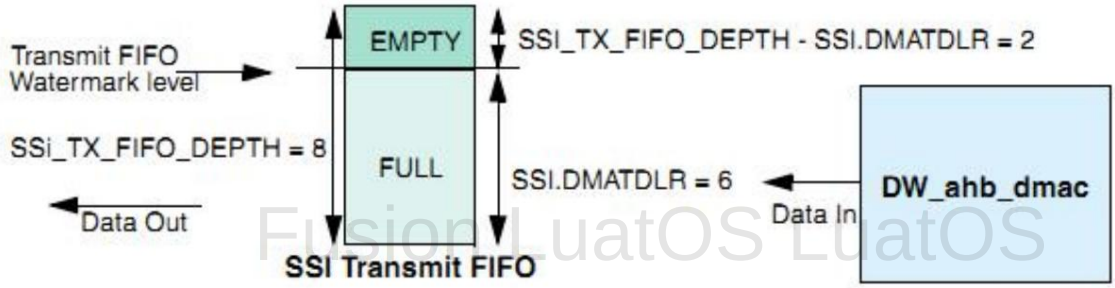
Reasonable security settings:



parameter	Notes
SSI.DMATDLR = 2	SSI peripheral transmit FIFO threshold
DMA.CTLx.DEST_MSIZE = 6 SSI_TX_FIFO_DEPTH -SSI.DMATDLR = 6 SSI_TX_FIFO_DEPTH	DEST_MSIZE is equal to the size of the remaining FIFO space
= 8 DMA.CTLx.BLOCK_TS = 30	Total transmit FIFO depth
	block size

Unreasonable settings:

DMA.CTLx.DEST_MSIZE > SSI_TX_FIFO_DEPTH -SSI.DMATDLR



parameter	Notes
SSI.DMATDLR = 6	SSI peripheral transmit FIFO threshold
DMA.CTLx.DEST_MSIZE = 4 SSI_TX_FIFO_DEPTH -SSI.DMATDLR = 2	DEST_MSIZE is less than FIFO remaining space size
SSI_TX_FIFO_DEPTH = 8 DMA.CTLx.BLOCK_TS = 30	Total transmit FIFO depth
	block size

ÿ When the non-memory peripheral is used as the source device, the non-memory peripheral sends data to the DMA after the data in the receiving FIFO reaches the set threshold.

At this time, the DMA will fetch data from the receive FIFO according to the amount of data in SRC_MSIZE. Therefore in the receive FIFO The set threshold should be greater than the SRC_MSIZ value to meet the amount of data read by the DMA channel once, if it is less than the data read once amount will cause the corresponding peripheral to generate a FIFO underflow interrupt.

17.7 Multi-Block Mode

Table 17- 1 DMA Multi-Block Mode

Transmission type	LLPx. LOC	LLP_ SRC_E N	RELOA D_SRC	LLP_ DST_E N	RELOA D_DST	CTLx, LLPx Update	SARx Update	DARx Update	Write Back
1.Single-bloc k or last transfer of multi-block	0	0	0	0	0	Manual editing Procedure	None	None	No
2.Auto-reload multi-block transfer with contiguous SAR	0	0	0	0	1	Automatically add load initial value	Contig uous	Auto-Reload	No
3.Auto-reload multi-block transfer with contiguous DAR	0	0	1	0	0	Automatically add load initial value	Auto-Reload	Contig uous	No
4.Auto-reload multi-block transfer	0	0	1	0	1	Automatically add load initial value	Auto-Reload	Auto-Reload	No
5. Single-block or last transfer of multi-block	1	0	0	0	0	Manual editing Procedure	None	None	Yes
6. Linked list multi-block transfer with contiguous SAR	1	0	0	1	0	Automatically add load next list item	Contig uous	Linked List	Yes
7. Linked list multi-block transfer with auto-reload SAR	1	0	1	1	0	Automatically add load next list item	Auto-Reload	Linked List	Yes
8. Linked list multi-block transfer with contiguous DAR	1	1	0	0	0	Automatically add load next list item	Linked List	Contig uous	Yes
9. Linked list multi-block transfer with	1	1	0	0	1	Automatically add download next	Linked List	Auto-Reload	Yes

auto-reload DAR 10. Linked list multi-block transfer	1	1	0	1	0	linked list item Automatically add load next list item	Linked List	Linked List	Yes

17.8 DMA register description

17.8.1 Address Mapping Table

DMA base address list

Address	base	Peripherals	bus
range 0x4000_0800-0x4000_0BFF	address 0x4000_0800	DMA	AHB

Table 17- 2 DMA register table

offset	register name	Width (bit) Reset value 32		register field
address 0x000	SAR0	0x00000000		DMA_Channel_0
0x004	reserved	32	0x00000000	
0x008	DAR0	32	0x00000000	
0x00C	reserved	32	0x00000000	
0x010	LLP0	32	0x00000000	
0x014		32	0x00000000	
reserved 0x018	CTL0_L	32	0x00304801	
0x01C	CTL0_H	32	0x00000002	
0x020 - 0x03F Reserved x 8 0x040		32x8	0x00000000	
	CFG0	32	0x00000E00	
0x044 - 0x057 Reserved x 5 0x058		32x5	0x00000000	
	SAR1	32	0x00000000	DMA_Channel_1
0x05C	reserved	32	0x00000000	
0x060	DAR1	32	0x00000000	
0x064	reserved	32	0x00000000	
0x068	LLP1	32	0x00000000	
0x06C		32	0x00000000	
reserved 0x070	CTL1_L	32	0x00304801	
0x074	CTL1_H	32	0x00000002	
0x078 - 0x097 Reserved x 8 0x098		32x8	0x00000000	
	CFG1	32	0x00000E20	
0x09C - 0x0AF Reserved x 5 0x0B0		32x5	0x00000000	
	SAR2	32	0x00000000	DMA_Channel_2
0x0B4	reserved	32	0x00000000	
0x0B8	DAR2	32	0x00000000	
0x0BC	reserved	32	0x00000000	
0x0C0	LLP2	32	0x00000000	
0x0C4		32	0x00000000	
reserved 0x0C8	CTL2_L	32	0x00304801	
0x0CC	CTL2_H	32	0x00000002	
0x0D0 - 0x0EF Reserved x 8 0x0F0		32x8	0x00000000	
	CFG2	32	0x00000E40	
0x0F4 - 0x107 Reserved x 5		32x5	0x00000000	
0x108	SAR3	32	0x00000000	DMA_Channel_3
0x10C	reserved	32	0x00000000	
0x110	DAR3	32	0x00000000	

0x114	reserved	32	0x00000000	
0x118	LLP3	32	0x00000000	
0x11C		32	0x00000000	
reserved 0x120	CTL3_L	32	0x00304801	
0x124	CTL3_H	32	0x00000002	
0x128 - 0x147 Reserved x 8 0x148		32x8	0x00000000	
	CFG3	32	0x00000E60	
0x14C - 0x15F Reserved x 5 0x160		32x5	0x00000000	
	SAR4	32	0x00000000	
0x164	reserved	32	0x00000000	
0x168	DAR4	32	0x00000000	
0x16C	reserved	32	0x00000000	
0x170	LLP4	32	0x00000000	
0x174		32	0x00000000	DMA_Channel_4
reserved 0x178	CTL4_L	32	0x00304801	
0x17C	CTL4_H	32	0x00000002	
0x180 - 0x19F Reserved x 8 0x1A0		32x8	0x00000000	
	CFG4	32	0x00000E60	
0x1A4 - 0x1B7 Reserved x 5 0x1B8		32x5	0x00000000	
	SAR5	32	0x00000000	
0x1BC	reserved	32	0x00000000	
0x1C0	DAR5	32	0x00000000	
0x1C4	reserved	32	0x00000000	
0x1C8	LLP5	32	0x00000000	
0x1CC		32	0x00000000	DMA_Channel_5
reserved 0x1D0	CTL5_L	32	0x00304801	
0x1D4	CTL5_H	32	0x00000002	
0x1D8 - 0x1F7 Reserved x 8 0x1F8		32x8	0x00000000	
	CFG5	32	0x00000E60	
0x1FC - 0x20F Reserved x 5 0x210		32x5	0x00000000	
	SAR6	32	0x00000000	
0x214	reserved	32	0x00000000	
0x218	DAR6	32	0x00000000	
0x21C	reserved	32	0x00000000	
0x220	LLP6	32	0x00000000	
0x224		32	0x00000000	DMA_Channel_6
reserved 0x228	CTL6_L	32	0x00304801	
0x22C	CTL6_H	32	0x00000002	
0x230 - 0x24F Reserved x 8 0x250		32x8	0x00000000	
	CFG6	32	0x00000E60	
0x254 - 0x267 Reserved x 5 0x268		32 x 5	0x00000000	
	SAR7	32	0x00000000	
0x26C	reserved	32	0x00000000	
0x270	DAR7	32	0x00000000	
0x274	reserved	32	0x00000000	
0x278	LLP7	32	0x00000000	
0x27C	Reserved	32	0x00000000	DMA_Channel_7
0x280	CTL7_L	32	0x00304801	
0x284 0x288 - 0x2A7 Reserved x 8	CTL7_H	32	0x00000002	
0x2A8		32x8	0x00000000	
	CFG7	32	0x00000E60	
0x2AC - 0x2BF Reserved x 5 0x2C0		32x5	0x00000000	
	RawTfr	32	0x00000000	DMA_Interrupt

0x2C4	reserved	32	0x00000000	
0x2C8	RawBlock	32	0x00000000	
0x2CC	Reserved	32	0x00000000	
0x2D0	RawSrcTran	32	0x00000000	
0x2D4	Reserved	32	0x00000000	
0x2D8	RawDstTran	32	0x00000000	
0x2DC	Reserved	32	0x00000000	
0x2E0	RawErr	32	0x00000000	
0x2E4	reserved	32	0x00000000	
0x2E8	StatusTfr	32	0x00000000	
0x2EC	Reserved	32	0x00000000	
0x2F0	StatusBlock	32	0x00000000	
0x2F4	Reserved	32	0x00000000	
0x2F8	StatusSrcTran	32	0x00000000	
0x2FC	Reserved	32	0x00000000	
0x300	StatusDstTran	32	0x00000000	
0x304	Reserved	32	0x00000000	
0x308	StatusErr	32	0x00000000	
0x30C	Reserved	32	0x00000000	
0x310	MaskTfr	32	0x00000000	
0x314	Reserved	32	0x00000000	
0x318	MaskBlock	32	0x00000000	
0x31C	reserved	32	0x00000000	
0x320	MaskSrcTran	32	0x00000000	
0x324	Reserved	32	0x00000000	
0x328	MaskDstTran	32	0x00000000	
0x32C	Reserved	32	0x00000000	
0x330	MaskErr	32	0x00000000	
0x334	reserved	32	0x00000000	
0x338	ClearTfr	32	0x00000000	
0x33C	Reserved	32	0x00000000	
0x340	ClearBlock	32	0x00000000	
0x344	Reserved	32	0x00000000	
0x348	ClearSrcTran	32	0x00000000	
0x34C	Reserved	32	0x00000000	
0x350	ClearDstTran	32	0x00000000	
0x354	Reserved	32	0x00000000	
0x358	ClearErr	32	0x00000000	
0x35C	Reserved	32	0x00000000	
0x360	StatusInt	32	0x00000000	
0x364	Reserved	32	0x00000000	
0x368	ReqSrcReg	32	0x00000000	Soft_HandShake
0x36C	reserved	32	0x00000000	
0x370	ReqDstReg	32	0x00000000	
0x374	reserved	32	0x00000000	
0x378	SglReqSrcReg	32	0x00000000	
0x37C	reserved	32	0x00000000	
0x380	SglReqDstReg	32	0x00000000	
0x384	reserved	32	0x00000000	
0x388	LstSrcReg	32	0x00000000	
0x38C	reserved	32	0x00000000	
0x390	LstDstReg	32	0x00000000	
0x394	reserved	32	0x00000000	

0x398	DmaCfgReg	32	0x00000000	DMA_Control
0x39C	reserved	32	0x00000000	
0x3A0	ChEnReg	32	0x00000000	
0x3A4	reserved	32	0x00000000	

DMA register area division table

ÿÿÿÿÿÿÿÿ	Register Field Base Address	Register Field	Peripherals
0x4000_0800-0x4000_0857	0x4000_0800	DMA_Channel_0	DMA
0x4000_0858-0x4000_08AF	0x4000_0858	DMA_Channel_1	
0x4000_08B0-0x4000_0907	0x4000_08B0	DMA_Channel_2	
0x4000_0908-0x4000_095F	0x4000_0908	DMA_Channel_3	
0x4000_0960-0x4000_09B7	0x4000_0960	DMA_Channel_4	
0x4000_09B8-0x4000_0A0F	0x4000_09B8	DMA_Channel_5	
0x4000_0A10-0x4000_0A67	0x4000_0A10	DMA_Channel_6	
0x4000_0A68-0x4000_0ABF	0x4000_0A68	DMA_Channel_7	
0x4000_0AC0-0x4000_0B67	0x4000_0AC0	DMA_Interrupt	
0x4000_0B68-0x4000_0B97	0x4000_0B68	Soft_HandShake	
0x4000_0B98-0x4000_0BB7	0x4000_0B98	DMA_Control	
0x4000_0BB8-0x4000_0BFF	0x4000_0BB8 Reserved		

17.8.2 DMAC Configuration Register (DmaCfgReg_L)

- ÿ **Name: DMAC Configuration Register**
- ÿ **Size: 32 bits**
- ÿ **Address Offset: 0x398**
- ÿ **Read/Write Access: Read/Write**

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

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

Bit	Name	W/R	Description
31:1	reserved	-	
0	DMA_EN	W/R	DMA peripherals can 0 - DMA peripheral off 1-DMA peripheral on

17.8.3 DMAC Channel Enable Register (ChEnReg)

- ÿ **Name: DW_ahb_dmac Channel Enable Register**
- ÿ **Size: 32 bits**
- ÿ **Address Offset: 0x3a0**
- ÿ **Read/Write Access: Read/Write**

CHx_EN_WE protects the enable operation of each channel. When writing to CHx_EN, it is necessary to write to the corresponding CHx_EN_WE "1".

For example: to enable channel 0, write 0x01x1 to ChEnReg, only the channel 0 operation is valid, and the other x corresponding bit operations are invalid.

31	30	29	28	27	26	25	twenty four
reserved							

twenty three	twenty two	twenty one	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
CH7_EN _WE	CH6_EN _WE	CH5_EN _WE	CH4_EN _WE	CH3_EN_ WE	CH2_EN_ WE	CH1_EN_ WE	CH0_EN_ WE
7	6	5	4	3	2	1	0
CH7_EN	CH6_EN	CH5_EN	CH4_EN	CH3_EN	CH2_EN	CH1_EN	CH0_EN

Bit	Name	W/R	Description
31:16	reserved	-	Reserved
15	CH7_EN_WE	W	DMAC channel 7 enable write protection bit
14	CH6_EN_WE	W	DMAC channel 6 enable write protection bit
13	CH5_EN_WE	W	DMAC channel 5 enable write protection bit
12	CH4_EN_WE	W	DMAC channel 4 enable write protection bit
11	CH3_EN_WE	W	DMAC channel 3 enable write protection bit
10	CH2_EN_WE	W	DMAC channel 2 enable write protection bit
9	CH1_EN_WE	W	DMAC channel 1 enable write protection bit
8	CH0_EN_WE	W	DMAC channel 0 enable write protection bit
7	CH7_EN	W/R	DMAC channel 7 enable bit 0 - channel off 1 - channel open
6	CH6_EN	W/R	DMAC channel 6 enable bit 0 - channel off 1 - channel open
5	CH5_EN	W/R	DMAC channel 5 enable bit 0 - channel off 1 - channel open
4	CH4_EN	W/R	DMAC channel 4 enable bit 0 - channel off 1 - channel open
3	CH3_EN	W/R	DMAC channel 3 enable bit 0 - channel off 1 - channel open
2	CH2_EN	W/R	DMAC channel 2 enable bit 0 - channel off 1 - channel open
1	CH1_EN	W/R	DMAC channel 1 enable bit 0 - channel off 1 - channel open
0	CH0_EN	W/R	DMAC channel 0 enable bit 0 - channel off 1 - channel open

17.8.4 Channel x Source Address Register (SARx) (x=0..7)

- **Name: Source Address Register for Channel x**
- **Size: 32 bits (upper 32 bits are reserved)**
- **Address Offset: for x = 0 to 7:**
  - **SAR0 – 0x000**
  - **SAR1 – 0x058**
  - **SAR2 – 0x0b0**

SAR3 – 0x108
SAR4 – 0x160
SAR5 – 0x1B8
SAR6 – 0x210
SAR7 – 0x268

ÿ **Read/Write Access: Read/Write**

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

Bit	Name	W/R	Description
31:0	SARx		W/R Channel x Source Address Register

17.8.5 Channel x Destination Register (DARx) (x=0..7)

ÿ **Name: Destination Address Register for Channel x**
ÿ **Size: 32 bits (upper 32 bits are reserved)**
ÿ **Address Offset: for x = 0 to 7:**

DAR0 – 0x008
DAR1 – 0x060
DAR2 – 0x0b8
DAR3 – 0x110
DAR4 – 0x168
DAR5 – 0x1C0
DAR6 – 0x218
DAR7 – 0x270

ÿ **Read/Write Access: Read/Write**

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

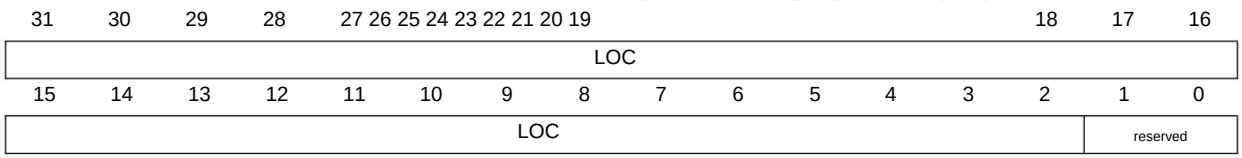
Bit	Name	W/R	Description
31:0	DARx		W/R Channel x Destination Register

17.8.6 Channel x Linked List Pointer Register (LLPx) (x=0..7)

ÿ **Name: Destination Address Register for Channel x**
ÿ **Size: 32 bits (upper 32 bits are reserved)**
ÿ **Address Offset: for x = 0 to 7:**

LLP0 – 0x010
LLP 1 – 0x068
LLP 2 – 0x0c0
LLP 3 – 0x118
LLP 4 – 0x170
LLP 5 – 0x1c8
LLP6 – 0x220
LLP7 – 0x278

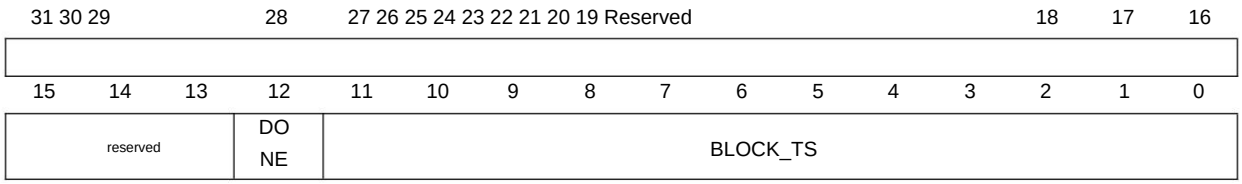
ÿ **Read/Write Access: Read/Write**



Bit	Name	W/R	Description
31:2	LOC	RW	The starting address of the memory corresponding to the next linked list item (LLI), the starting point The lower 2bits of the address do not take effect (the address is fixed to 32bit alignment)
1:0	reserved	RW	Reserved bit, value cannot be changed

17.8.7 Channel x Control Register (CTLx_H) (x=0..7)

- **Name: Control Register for Channel x**
- **Size: 64 bits**
- **Address Offset: for x = 0 to 7:**
  - **CTL0 – 0x01C**
  - **CTL1 – 0x074**
  - **CTL2 – 0x0CC**
  - **CTL3 – 0x124**
  - **CTL4 – 0x17C**
  - **CTL5 – 0x1D4**
  - **CTL6 – 0x22C**
  - **CTL7 – 0x284**
- **Read/Write Access: Read/Write**



Bit	Name	W/R	Description
31:13	reserved	-	reserved bit
12	DONE	W/R	After the write-back function is enabled, this flag is set after the block transfer is completed. At the same time, the register value will be written to the control corresponding to the linked list item Register CTL_H. The software can query the LLI.CTL_H.Done flag to judge a block Whether the transfer is complete
11:0	BLOCK_TS	W/R	DMA transfer data block size, ie XXX_TR_WIDTH Number of unit data When DMA transfers data, it is regarded as unit data, 1 DMA The size of the transmission data block is XXX_TR_WIDTH The number of bits of data. 1 DMA transfer bytes (Byte) = BLOCK_TS * XXX_TR_WIDTH / 8

17.8.8 Channel x Control Register (CTLx_L) (x=0..7)

- **Name: Control Register for Channel x**
- **Size: 64 bits**
- **Address Offset: for x = 0 to 7:**

CTL0 – 0x018
CTL1 – 0x070
CTL2 – 0x0c8
CTL3 – 0x120
CTL4 – 0x178
CTL5 – 0x1D0
CTL6 – 0x228
CTL7 – 0x280

ÿ **Read/Write Access: Read/Write**

31	30	29	28	27	26	25 twenty four
reserved			LLP_SRC_EN	LLP_DST_EN	reserved	

23	twenty two	twenty one	20	19	18	17	16
Reserve	TT_FC			Reserved			

15	14	13	12	11	10	9	8
SRC_MSIZE 6 7		DEST_MSIZE 4			SINC		

5			3	2	1	0
DINC	SRC_TR_WIDTH		DST_TR_WIDTH			INT_EN

Bit	Name	W/R	Description
31:29	reserved	- reserved bit	
28	LLP_SRC_EN	W/R	Multi-blockchain source enable bit 1: enable
27	LLP_DST_EN	W/R	Multi-blockchain destination enable bit 1: enable
26:23	reserved	W/R reserved bit	
22:20	TT_FC	W/R	Transmission method and stream controller selection The setting value and the corresponding table of transmission mode: Setting value transmission method 000 Memory to Memory 001 Memory to Peripheral 010 Peripheral to Memory
19:17	reserved	-	
16:14	SRC_MSIZE	W/R	Source data burst transfer volume (Source Burst Transaction Length) In 1 burst transmission of source data, SRC_TR_WIDTH is transmitted , that is, how many SRC_TR_WIDTH are transmitted. * Source data 1 burst transfer Byte = SRC_MSIZE SRC_TR_WIDTH / 8 Set value and data volume corresponding table: set value / The amount of data 000 1 001 4 010 8 011 16 100 32 101 64 110 128 111 256
13:11	DEST_MSIZE	W/R	Destination Burst Transaction Length)

			In 1 burst transmission of source data, DST_TR_WIDTH is transmitted , that is, how many DST_TR_WIDTH are transmitted. * Target data 1 burst transfer Byte = DEST_MSIZE DST_TR_WIDTH / 8 The set value and data volume correspondence table is the same as SRC_MSIZE
10:9	SINC	W/R	Source address increment mode 00 = Address self-incrementing 01 = address is decremented 1x = the address does not change
8:7	DINC	W/R	Target address increment mode 00 = Address self-incrementing 01 = address is decremented 1x = the address does not change
6:4	SRC_TR_WIDTH	W/R	source data bit width Setting value and data bit width corresponding table:
3:1	DST_TR_WIDTH	W/R	target data bit width The setting value and data bit width correspond to the same as SRC_TR_WIDTH
0	INT_EN	W/R	DMA interrupt master control bits 0 - disable DMA all interrupts 1- Turn on DMA for all interrupts

Set value	data bit width
000	8
001	16
010	32
011	64
100	128
101	256
11x	256

17.8.9 Channel x Configuration Register (CFGx_L) (x=0..7)

- **Name: Configuration Register for Channel x**
- **Size: 32 bits (upper 32 bits are reserved)**
- **Address Offset: for x = 0 to 7:**
  - **CFG0 – 0x040**
  - **CFG1 – 0x098**
  - **CFG2 – 0x0f0**
  - **CFG3 – 0x148**
  - **CFG4 – 0x1A0**
  - **CFG5 – 0x1F8**
  - **CFG6 – 0x250**
  - **CFG7 – 0x2A8**
- **Read/Write Access: Read/Write**

31	30	29	28	27	26	25	twenty four
RELOAD_ DST	RELOAD_ SRC	reserved					
twenty three	twenty two	twenty one	20	19	18	17	16

reserved				SRC_HS_POL	DST_HS_POL	reserved	
15	14	13	12	11	10	9	8
reserved				HS_SEL_SRC	HS_SEL_DST	FIFO_EMPTY	CH_SUSP
7	6	5	4	3	2	1	0
CH_PRIOR				reserved			

Bit	Name	W/R	Description
31	RELOAD_DST	W/R	Autoload destination address In multi-block transfer mode, DARx registers after each block transfer is completed. The processor automatically loads its initial value, and a new block transfer is initialized Initialize 1: Enable; 0: Disable
30	RELOAD_SRC	W/R	Autoload source address In multi-block transfer mode, the SARx register is The processor automatically loads its initial value, and a new block transfer is initialized Initialize 1: Enable; 0: Disable
29:20	reserved	- reserved bits	
19	SRC_HS_POL	W/R	Source device handshake signal valid polarity 0-high effective 1 - active low
18	DST_HS_POL	W/R	Target device handshake signal valid polarity 0-high effective 1 - active low
17:12	reserved	- reserved bits	
11	HS_SEL_SRC	W/R	Source device soft/hard handshake interface selection 0 - hard handshake 1- Soft handshake
10	HS_SEL_DST	W/R	Target device soft/hard handshake interface selection 0 - hard handshake 1- Soft handshake
9	FIFO_EMPTY	R	Channel FIFO Status 0 - Channel FIFO is not full 1-channel FIFO full
8	CH_SUSP	W/R	Channel Suspend 0 - do not hang 1 - Suspend DMA channel to get data from source device
7:5	CH_PRIOR	W/R	channel priority Setting value range: 0 < CH_PRIOR < 7 (number of channels - 1) Exceeding the setting range will result in an abnormal error
4:0	reserved	- reserved bits	

17.8.10 Source Interrupt Status Register

- **Name: Interrupt Raw Status Registers**
- **Size: 32 bits**
- **Address Offset:**
  - **RawTfr – 0x2c0**
  - **RawBlock – 0x2c8**

RawSrcTran – 0x2d0
RawDstTran – 0x2d8
RawErr – 0x2e0

ÿ **Read/Write Access: Read/Write**

Source interrupt status registers include: RawBlock, RawDstTran, RawErr, RawSrcTran, RawTfr

For details, see the "DMA Interrupt" chapter.

The above register structures and corresponding channels are the same, but the operation addresses are different.

The status value in the source interrupt status register is not affected by the interrupt mask register (Mask).

By writing "1" to the corresponding interrupt clear register, the flag bit in the corresponding source interrupt status register is cleared.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved								CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

Bit	Name	W/R	Description
31:8	reserved	- Reserved	
7	CH7	W/R	Source interrupt flag bit, write operation is usually used for testing, normal mode write operation is not recommended. 0 - no interrupt is generated 1 - generate an interrupt
6	CH6	W/R	
5	CH5	W/R	
4	CH4	W/R	
3	CH3	W/R	
2	CH2	W/R	
1	CH1	W/R	
0	CH0	W/R	

17.8.11 Interrupt Status Register

ÿ **Name: Interrupt Status Registers**
ÿ **Size: 32 bits**
ÿ **Address Offset:**
StatusTfr – 0x2c0
StatusBlock – 0x2c8
StatusSrcTran – 0x2d0
StatusDstTran – 0x2d8
StatusErr – 0x2e0
ÿ **Read/Write Access: Read/Write**

Interrupt status registers include: StatusBlock, StatusDstTran, StatusErr, StatusSrcTran, StatusTfr

For details, see the "DMA Interrupt" chapter.

The above register structures and corresponding channels are the same, but the operation addresses are different.

The status value in the interrupt status register will be affected by the interrupt mask register (Mask). When the bit bit of the interrupt status register is "1", the peripheral An interrupt signal will be generated to the CPU.

By writing "1" to the corresponding interrupt clear register, the flag bit in the corresponding source interrupt status register is cleared.

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

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

reserved	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0

Bit	Name	W/R	Description
31:8	reserved	-	
7	CH7	R	Interrupt status flag bit. 0 - no interrupt is generated 1 - generate an interrupt
6	CH6	R	
5	CH5	R	
4	CH4	R	
3	CH3	R	
2	CH2	R	
1	CH1	R	
0	CH0	R	

17.8.12 Interrupt Mask Register

- **Name: Interrupt Mask Registers**
- **Size: 32 bits**
- **Address Offset:**
  - **MaskTfr – 0x310**
  - **MaskBlock – 0x318**
  - **MaskSrcTran – 0x320**
  - **MaskDstTran – 0x328**
  - **MaskErr – 0x330**
- **Read/Write Access: Read/Write**

Interrupt mask registers include: MaskBlock, MaskDstTran, MaskErr, MaskSrcTran, MaskTfr

For details, see the "DMA Interrupt" chapter.

The above register structures and corresponding channels are the same, but the operation addresses are different.

The interrupt mask register (Mask) is used to mask DMA interrupts and affect the interrupt status register (Status) value.

Each interrupt mask bit corresponds to a write protection operation.

For example: turn on the channel 0 interrupt, write 0x01 to the MaskBlock, only the channel 0 operation is valid, and the other x corresponding bit operations are invalid.

31	30	29	28	27	26	25	24	23	twenty two	21	20	19	18	17	16
reserved															

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

Bit	Name	W/R	Description
31:16	reserved	- reserved	
15	CH7_WE	W	Interrupt Mask Write Protect Bit 0-CHx write operation is invalid 1-CHx write operation is valid
14	CH6_WE	W	
13	CH5_WE	W	
12	CH4_WE	W	
11	CH3_WE	W	
10	CH2_WE	W	
9	CH1_WE	W	
8	CH0_WE	W	
7	CH7	W/R Interrupt status flag.	

6	CH6	W/R	0 - interrupt off
5	CH5	W/R	1- Interrupt on
4	CH4	W/R	
3	CH3	W/R	
2	CH2	W/R	
1	CH1	W/R	
0	CH0	W/R	

17.8.13 Interrupt Status Register

- **Name: Interrupt Clear Registers**
- **Size: 32 bits**
- **Address Offset:**
  - **ClearTfr – 0x2c0**
  - **ClearBlock – 0x2c8**
  - **ClearSrcTran – 0x2d0**
  - **ClearDstTran – 0x2d8**
  - **ClearErr – 0x2e0**
- **Read/Write Access: Read/Write**

Interrupt status registers include: ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, ClearTfr

For details, see the "DMA Interrupt" chapter.

The above register structures and corresponding channels are the same, but the operation addresses are different.

Write "1" to each interrupt clear register, the flag bit in the source interrupt register will be cleared to "0" in the same cycle.

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

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

Bit	Name	W/R	Description
31:8	reserved	- reserved	
7	CH7	W	Interrupt status flag bit. 0 - invalid 1 - clear interrupt
6	CH6	W	
5	CH5	W	
4	CH4	W	
3	CH3	W	
2	CH2	W	
1	CH1	W	
0	CH0	W	

17.8.14 Combined Interrupt Status Register (ChEnReg)

- **Name: Combined Interrupt Status Register**
- **Size: 32 bits**
- **Address Offset: 0x360**
- **Read/Write Access: Read/Write**

The following interrupt status bits are the status value of each interrupt status register or the result after operation, that is, the interrupt status or operation result of each of the 5 channels.

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

Bit	Name	W/R	Description
31:5	reserved	-	
4	ERR	R	error status interrupt
3	DSTT	R	Target data transfer complete interrupt
2	SRCT	R	source data transfer complete interrupt
1	BLOCK	R	block transfer interrupt
0	TFR	R	transfer complete interrupt

17.8.15 Soft Handshake Interface Register

- **Name: Interrupt Mask Registers**
- **Size: 32 bits**
- **Address Offset:**

  ReqSrcReg – 0x368
  ReqDstReg – 0x370
  SglReqSrcReg – 0x378
  SglReqDstReg – 0x380
  LstSrcReg – 0x388
  LstDstReg – 0x390
- **Read/Write Access: Read/Write**

Soft handshake interface registers: ReqSrcReg, ReqDstReg, SglReqSrcReg, SglReqDstReg, LstSrcReg, LstDstReg

For the function description, see the chapter "DMA Soft Handshake Interface"

The above register structures and corresponding channels are the same, but the operation addresses are different.

Each interrupt mask bit corresponds to a write protection operation.

For example: generate source data DMA request corresponding to channel 0, write 0x0101 to ReqSrcReg, in which only channel 0 operation is valid, and the other x pairs
Bit operation is invalid.

31 30 29 28 27 26 25 24 23 22 21 20																			19	18	17	16
reserved																						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
CH7_WE	CH6_WE	CH5_WE	CH4_WE	CH3_WE	CH2_WE	CH1_WE	CH0_WE	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0							

Bit	Name	W/R	Description
31:16	reserved	-	
15	CH7_WE	W	DMA request write protection bit 0-CHx write operation is invalid 1-CHx write operation is valid
14	CH6_WE	W	
13	CH5_WE	W	
12	CH4_WE	W	
11	CH3_WE	W	
10	CH2_WE	W	
9	CH1_WE	W	

8	CH0_WE	W	
7	CH7	W/R	DMA request bit. 0 - invalid 1- Generate the request
6	CH6	W/R	
5	CH5	W/R	
4	CH4	W/R	
3	CH3	W/R	
2	CH2	W/R	
1	CH1	W/R	
0	CH0	W/R	

18 USB ports

18.1 Introduction to USB

The USB peripheral implements the interface between the USB2.0 full-speed bus and the APB1 bus

USB peripherals support USB suspend/resume operation and can stop the device clock for low power consumption

USB2.0 Device

18.2 Main Features of USB

ÿ Comply with the technical specifications of USB2.0 full-speed devices

ÿ CRC (Cyclic Redundancy Check) generation/checking, reverse non-return-to-zero (NRZI) encoding/decoding and bit stuffing

18.3 USB Function Description

The USB module provides a USB-compliant communication connection between the PC host and the functions implemented by the microcontroller. PC host and microcontroller

Data transfer between USB devices is accomplished by sharing a dedicated data buffer that can be directly accessed by USB peripherals.

18.4 USB register description

The registers of the USB module have the following three categories:

ÿ General-purpose registers: interrupt registers and control registers

ÿ Endpoint class registers: endpoint configuration register and status register

ÿ Buffer description table type register: a register used to determine the storage address of data packets

The base address of the buffer description table type register is specified by the USB_BTABLE register, and the base address of all other registers is the USB module.

The base address of the block is 0x4000 5C00. The same address alignment is used for packet buffer memory starting at 0x4000 6000.

18.4.1 Address Mapping Table

List of USB base addresses

Address	Base	Peripherals	bus
range 0x4000_0C00-0x4000_0FFF	address 0x4000_0C00	USB	AHB

Table 18- 1 USB register table

Offset address	register name	Width (bit)	Reset value
0x00	FADDR	8	0x00
0x01	POWER	8	0x20
0x02	INTRTX	16	0x0000
0x04	INTRRX	16	0x0000
0x06	INTRTXE	16	0xFFFF
0x08	INTRRXE	16	0xFFFE
0x0A	INTRUSB	8	0x00
0x0B	INTRUSBE	8	0x06
0x0C	FRAME	16	0x0000
0x0E	INDEX	8	0x00
0x0F	TESTMODE	8	0x00
0x10	TXMAXP	16	0x0000
0x12	TXCSRL	8	0x00
0x13	TXCSRH	8	0x00
0x14	RXMAXP	16	0x0000

0x16	RXCSRL	8	0x00
0x17	RXCSRH	8	0x00
0x18	RXCOUNT	16	0x0000
0x1A	TXTYPE	8	0x00
0x1B	TXINTERVAL	8	0x00
0x1C	RXTYPE	8	0x00
0x1D	RXINTERVAL	8	0x00
0x1F	FIFOSIZE	8	Configuration Dependent
0x20-0x5F	FIFOx	32	0x00000000
0x60 0x61	DEVCTL	8	0x80
0x62 0x63	MISC	8	0x00
0x64 0x66	TXFIFOSZ	8	0x00
0x68 0x6C	RXFIFOSZ	8	0x00
0x70-0x77	TXFIFOADD	16	0x0000
	RXFIFOADD	16	0x0000
	VCONTROL/VSTATUS	32	-
	HWVERS	32	Version dependent
	-	-	-
0x78	EPINFO	8	Implementation dependent
0x79	RAMINFO	8	Implementation dependent
0x7A	LINKINFO	8	0x5C
0x7B	VPLEN	8	0x3C
0x7C	HS_EOF1	8	0x80
0x7D	FS_EOF1	8	0x77
0x7E	LS_EOF1	8	0x72
0x7F	SOFT_RST	8	0x00
0x80+8*n	TXFuncAddr	8	0x00
0x82+8*n	TXHubAddr	8	0x00
0x83+8*n	TXHubPort	8	0x00
0x84+8*n	RxFuncAddr	8	0x00
0x86+8*n	RxHubAddr	8	0x00
0x87+8*n	RxHubPort	8	0x00
0x100+16*n TXMAXP 0x102+16*n		16	0x0000
TXCSRL 0x103+16*n TXCSRH		8	0x00
0x104+16*n RXMAXP 0x106+16*n		8	0x00
RXCSRL 0x107+16*n RXCSRH		16	0x0000
0x108+16*n RXCOUNT		8	0x00
0x10A+16*n TXBTYPE 0x +16*n		8	0x00
TXINTERVAL 0x10C+16*n		16	0x0000
RXTYPE 0x10D+16*n		8	0x00
RXINTERVAL		8	0x00
		8	0x00
		8	0x00
0x10F+16*n FIFOSIZE		8	Configuration Dependent
0x200	DMA_INTR		0x00
0x204	DMA_CNTL	8	0x0000
0x208	DMA_ADDR	16	0x00000000
0x20C	DMA_COUNT	32 32	0x00000000

0x300+2*n	RqPktCount	16	0x0000
0x340	RxDPktBufDis	16	0x0000
0x342	TxDPktBufDis	16	0x0000
0x344	C_T_UCH	16	Various
0x346	C_T_HSRTN	16	Various
0x348	C_T_HSBT	16	0x0000
0x360	LPM_ATTR	16	0x00
0x362	LPM_CNTRL	8	0x00
0x363	LPM_INTREN	8	0x00
0x364	LPM_INTR	8	0x00
0x365	LPM_FADDR	8	0x00

18.4.2 USB Common Registers

FADDR

Offset address: 0x0000 Reset value: 0x00

Note: Only valid in peripheral mode

D7	D6	D5	D4	D3	D2	D1	D0
0	Function Address						
r	rw						

Bit	Name	W/R	Description
D7	-		- Not used, read always returns 0
D6:D0	Func Addr	W/R	When used in device mode (DevCtl.D2=0), write via The address received by the SET_ADDRESS command, followed by the Token The package will automatically use this address.

POWER

Offset address: 0x0001 Reset value: 0x20

	D7	D6	D5	D4	D3	D2	D1	D0
	ISO Update	Soft Conn	HS Enab	HS Mode	Reset	Resume	Suspend Mode	Enable SuspendM
Peripera	rw	rw	rw	r	r	rw	r	rw
Host	-	-	rw	r	rw	rw	w	rw

Bit	Name	W/R	Description
D7	ISO Update	W/R	Setting this bit will wait to receive a signal from when TxPktRdy is set. Send packets after SOF Tokens. If waiting for SOF Token During the period, an IN Token is received first, and a 0 length will be sent of data packets. Note: This bit is only valid in device mode. and only works with etc. **Isocbronous transfers**
D6	Soft Conn	W/R	When software connect/disconnect function is enabled, set this bit to enable USB D+/D- line, clear this bit USB D+/D- line is trying. Note: This bit is only valid in device mode.
D5	HS Enab	W/R	not currently supported
D4	HS Mode	R	does not currently support
D3	Reset	Host:W/R	Change to 1 when a Reset signal is detected on the bus.

		Peripheral mode:RO	Note: Change to readable and writable in **Host Mode ;** in **Peripheral** Read-only in **Mode .**
D2	Resume	W/R	When the device is in Suspend mode, setting this bit will generate Resume signal. In Peripheral Mode, it should be within 10ms (the most Clear this bit after 15ms); in Host Mode it should be within 20ms After clearing this bit.
D1	Suspend Mode	R	In Host Mode, set this bit to enter Suspend Mode; in In Peripheral Mode, entering Suspend Mode will set this bit. Reading the interrupt register or setting the Resume bit above will Clear this bit.
D0	Enable SuspendM	W/R	Setting this bit enables the SUSPENDM output.

INTRTX

Offset address: 0x0002 Reset value: 0x0000

IntrTx is a 16-bit read-only register used to indicate the current interrupt on endpoint 0 and Tx endpoints 1-7.

Note: The interrupt bit will always be 0 for endpoints that are not configured. A simultaneous read of this register will clear all active interrupts indicator bit.

D15	D14	D13	D12	D11	D10	D9	D8
reserved							
ro							
D7	D6	D5	D4	D3	D2	D1	D0
EP7 Tx	EP6 Tx	EP5 Tx	EP4 Tx	EP3 Tx	EP2 Tx	EP1 Tx	EP0
r	r	r	r	r	r	r	r

Bit	Name	W/R	Description
D15-D8	reserved	RO	reserved
D7	EP7 Tx	RO	Tx endpoint 7 interrupt
D6	EP6 Tx	RO	Tx endpoint 6 interrupt
D5	EP5 Tx	RO	Tx endpoint 5 interrupt
D4	EP4 Tx	RO	Tx endpoint 4 interrupt
D3	EP3 Tx	RO	Tx endpoint 3 interrupt
D2	EP2 Tx	RO	Tx endpoint 2 interrupt
D1	EP1 Tx	RO	Tx endpoint 1 interrupt
D0	EP0	RO	endpoint 0 interrupt

INTRRX

Offset address: 0x0004 Reset value: 0x0000

IntrRx is a 16-bit read-only register used to indicate the current interrupt occurred at Rx endpoints 1-7.

Note: The interrupt bit will always be 0 for endpoints that are not configured. A simultaneous read of this register will clear all active interrupts indicator bit.

D15	D14	D13	D12	D11	D10	D9	D8
reserved							
ro							
D7	D6	D5	D4	D3	D2	D1	D0
EP7 Rx	EP6 Rx	EP5 Rx	EP4 Rx	EP3 Rx	EP2 Rx	EP1 Rx	-
r	r	r	r	r	r	r	r

Bit	Name	W/R	Description

D15-D8	reserved	RO	reserved
D7	EP7 Rx	RO	Rx endpoint 7 interrupt
D6	EP6 Rx	RO	Rx endpoint 6 interrupt
D5	EP5 Rx	RO	Rx endpoint 5 interrupt
D4	EP4 Rx	RO	Rx endpoint 4 interrupt
D3	EP3 Rx	RO	Rx endpoint 3 interrupt
D2	EP2 Rx	RO	Rx endpoint 2 interrupt
D1	EP1 Rx	RO	Rx endpoint 1 interrupt
D0	-	RO	Unused

INTRTXE

Offset Address: 0x0006 Reset value: 0xFFFF

IntrTxE is a 16-bit register that provides interrupt enable for the IntrTx register. Writing 1 to the corresponding Bit will enable the corresponding interrupt,

When the interrupt event occurs, the corresponding Bit bit of IntrTx will be set and the interrupt will be triggered. When the corresponding Bit bit is 0, when the interrupt event occurs

It will also set the bit corresponding to intrTx but will not trigger an interrupt.

Note: Bits that are not configured will always be 0.

D15	D14	D13	D12	D11	D10	D9	D8
reserved							
ro							
D7	D6	D5	D4	D3	D2	D1	D0
EP7 Tx	EP6 Tx	EP5 Tx	EP4 Tx	EP3 Tx	EP2 Tx	EP1 Tx	EP0
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D15-D8	reserved	RO	reserved
D7	EP7 Tx	W/R	Tx endpoint 7 interrupt enable
D6	EP6 Tx	W/R	Tx endpoint 6 interrupt enable
D5	EP5 Tx	W/R	Tx endpoint 5 interrupt enable
D4	EP4 Tx	W/R	Tx endpoint 4 interrupt enable
D3	EP3 Tx	W/R	Tx endpoint 3 interrupt enable
D2	EP2 Tx	W/R	Tx endpoint 2 interrupt enable
D1	EP1 Tx	W/R	Tx endpoint 1 interrupt enable
D0	EP0	W/R	Endpoint 0 Interrupt Enable

INTRRXE

Offset address: 0x0008 Reset value: 0xFFFF

IntrRxE is a 16-bit register that provides interrupt enable for the IntrRx register. Writing 1 to the corresponding Bit will enable the corresponding interrupt,

When the interrupt event occurs, the corresponding Bit bit of IntrRx will be set and the interrupt will be triggered. When the corresponding Bit bit is 0, when the interrupt event occurs

The bit corresponding to intrRx will also be set but the interrupt will not be triggered.

Note: Bits that are not configured will always be 0.

D15	D14	D13	D12	D11	D10	D9	D8
reserved							
ro							
D7	D6	D5	D4	D3	D2	D1	D0
EP7 Rx	EP6 Tx	EP5 Tx	EP4 Tx	EP3 Tx	EP2 Tx	EP1 Tx	-
rw	rw	rw	rw	rw	rw	rw	r

Bit	Name	W/R	Description

D15-D8	reserved	RO	reserved
D7	EP7 Rx	W/R	Rx endpoint 7 interrupt enable
D6	EP6 Rx	W/R	Rx endpoint 6 interrupt enable
D5	EP5 Rx	W/R	Rx endpoint 5 interrupt enable
D4	EP4 Rx	W/R	Rx endpoint 4 interrupt enable
D3	EP3 Rx	W/R	Rx endpoint 3 interrupt enable
D2	EP2 Rx	W/R	Rx endpoint 2 interrupt enable
D1	EP1 Rx	W/R	Rx endpoint 1 interrupt enable
D0	-	RO	Unused

INTRUSB

Offset address: 0x000A Reset value: 0x00

D7	D6	D5	D4	D3	D2	D1	D0
VBus Error	Sess Req	Discon	Conn	SOF	Reset/ Babble	Resume	Suspend
ro	ro	ro	ro	ro	ro	ro	ro

Bit	Name	W/R	Description
D7	VBus Error	R	During a session, set when VBus is below the valid VBus threshold **bit. Only valid as an 'A' device.**
D6	Sess Req	R	Set when a Session Request signal is detected. only when set as 'A' Valid when ready.
D5	Discon	R	Set when a device disconnection is detected in Host Mode; Peripheral Mode Set when the next session ends. exist Valid for all types of transfer speeds.
D4	Conn	R	Set when a device connection is detected. Only in **Host** Mode Valid for all types of transfer speeds.
D3	SOF	R	Set when a new frame (Frame) starts
D2	Reset	R	In Peripheral Mode when a Reset signal is detected on the bus time set.
	Babble	R	Set when Babble is detected in Host Mode. Note: Only after the first **SOF** is **sent (active).**
D1	Resume	R	is in Suspend Mode. Resume signal detected on the bus Position.
D0	Suspend	R	Set when a Suspend signal is detected on the bus. Note: Only valid in Peripheral Mode.

INTRUSBE

Offset address: 0x000B Reset value: 0x06

IntrUSBE is an 8-bit register that provides interrupt enable for each interrupt in IntrUSB.

D7	D6	D5	D4	D3	D2	D1	D0
VBus Error	Sess Req	Discon	Conn	SOF	Reset /Babble	Resume	Suspend
rw	rw	rw	rw	rw	rw	rw	rw

FRAME

Offset address: 0x000C Reset value: 0x0000

Frame is a 16-bit read-only register used to hold the latest received frame number (Frame number).

D15	…	D11	D10	…	D0
0 0 0 0 0			(MSB) (LSB)	Frame Number	
r	…	r	r	…	r

INDEX

Offset address: 0x000E Reset value: 0x00

Each TX endpoint, each receiving terminal has its own set of control/status registers located between 100H-1FFh. Another set of TX Control/status and a set of receive control/status and a set of receive control/status registers appear at 10H-19H.

Index is a four-bit register used to determine which endpoint control/status registers are accessed.

D3	D2	D1	D0
(MSB)	Selected Endpoint		(LSB)
rw	rw	rw	rw

The endpoint number should be written to the index register before accessing 10h-19h of the endpoint control/status register to ensure proper control/status The registers will appear memory mapped.

TESTMODE

Offset Address: 0x0F Reset Value: 0x00

TESTMODE is an 8-bit register, which mainly applies Air105 to the four test modes described in the USB2.0 specification for high-speed running one

D7	D6	D5	D4	D3	D2	D1	D0
Force_Host	FIFO_Access	Force_FS	Force_HS	Test_Packet	Test_K	Test_J	Test_SE0_NAK
rw	w	rw	rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D7	Force_Host	W/R	The CPU sets this bit to indicate that when the core enters host mode, the session bit is set, not Whether it is connected to a peripheral or not. In the CID input state, HostDisconnect and LineState signals are ignored. The core will then remain in host mode, Until the session bit is cleared, even if the device is disconnected, if the Force_Host bit remains keep setting, will re-enter host mode the next time the session bit is set, here mode, the status of the HOSTDISCON signal from the PHY can be changed from bit 7 Reading the DevCtl register, the operating speed is determined from the Force_FS bits as follows:
D6	FIFO_Access	W	CPU sets this bit to endpoint 0 Tx FIFO transmits data packets to endpoint 0 receive FIFO, it will be automatically cleared
D5	Porce_FS	W/R	CPU sets this bit or with 7+ bits or forces Air105 to receive at full speed A USB reset mode
D4	Porce_HS	W/R	CPU set this bit or with 7 or more or force Air105 to receive high speed A USB reset mode
D3	Test_Packet	W/R	The CPU sets this bit to enter the Test Packet test mode. In this mode, The MH1903 repeatedly sends 53-byte test packets on this bus, in which The form is defined in the Universal Serial Bus Specification, Revision 2.0, Section 7.1.20

Force_FS	running speed
0	low speed
1	full speed
1	undefined

			Note: Test packets have a fixed format and must be loaded into the endpoint **0 FIFO** Test mode was entered before
D2	Test_K	W/R	CPU sets this bit to enter Test_K test mode, in this mode, Air105 Send K continuously on the bus
D1	Test_J	W/R	CPU sets this bit to enter Test_J test mode, in this mode, Air105 Continuously send J on the bus
D0 Test_sE0_NAK W/R			The CPU sets this bit to enter Test_SE0_NAK test mode, in this mode , the Air105 remains in high-speed mode, but responds to any valid IN token.

DEVCTL

Offset Address: 0x60 Reset Value: 0x80

DevCtl is an 8-bit register used to select whether the Air105 runs in peripheral mode or in host mode and monitors USB for control VBUS line. If the PHY is suspended, no PHY clock (XCLK) is received and VBUS is not sampled

D7	D6	D5	D4	D3	D2	D1	D0
B-Device	FSDev	LSDev	VBus[1]	VBus[0]	Host Mode	Host Req	Session
r	r	r	r	r	r	rw	rw

Bit	Name	W/R	Description
D7	B-Device	W/R	This read-only bit indicates whether the Air105 operates as an "A" device or a "B" device, 0=>'A' device, 1=>'B' device. Only active sessions are in progress, when no sessions are in progress When the role is determined, the session bit is set and read. NOTE: If the core is in Force_Host mode (ie the session has started and Testmode.D7=1), this bit will indicate the HOSTDISCON from the PHY input signal status
D6	FSDev	R	When this read-only bit is set, a full-speed device is detected connected to this port. Only There is valid in host mode
D5	LSDev	R	When this read-only bit is set, a low-speed device is detected connected to this port. Only There is valid in host mode
D4-D3 VBus[1:0]		R	The HESE read-only bits encode the current VBUS level as follows:
D2 Host Mode		R	This read-only bit is set when the Air105 acts as a host
D1	Host Req	W/R	When set, Air105 will enter suspend mode when starting host negotiation. when the host Cleared after negotiation is complete.
D0	Session	W/R	When operating as an 'A' device, this bit is set or cleared by the CPU to start or End the session. When operating as a 'B' device, this bit is set by the Air105 clears at the beginning/end of the session. It is also used by CPU1 to initiate a session request protocol discussion. When Air105 is in suspend mode, this bit can be cleared by CPU to execute software disconnect Note: Clearing this bit when the core is not halted will result in undefined behavior

D4 D3		Meaning
0 0		Below SESSIONEND
0	1	Above SESSIONEND, below AValid
1	0	Above AValid, below VBusValid
1	1	Above VBusValid

MISC

Offset address: 0x60 Reset value: 0x00

The MISC register is an 8-bit register that contains various common configuration bits, including the RX/TX earlier than DMA enable bits.

D7	D6	D5	D4	D3	D2	D1	D0
Unused						tx_edma	rx_edma
r	r	r	r	r	r	rw	rw

Bit	Name	W/	Description
D7-D2 Unused		RR	These bits are reserved
D1	tx_edma	W/R	1'b0: DMA_REQ signal, all IN endpoints will be re-pulled high when MAXP Bytes are written to an endpoint, which is a late mode. 1'b1: DMA_REQ signal, all IN endpoints will be re-pulled high when MAXP-8 Bytes have been written to the endpoint, which is the early mode.
D0	rx_edma	W/R	1'b0: DMA_REQ signal, all OUT endpoints will be pulled high again when MAXP Bytes are read to the endpoint, which is a late mode. 1'b1: DMA_REQ signal, when all OUT endpoints will be pulled high again MAXP-8 bytes are read to the endpoint, which is the earlier mode.

18.4.3 USB Indexed registers

CSR0L

CSR0L is an 8-bit register that provides control and status bits for endpoint 0.

Note: The interpretation of this register depends on whether the Air105 is acting as a peripheral or as a host. Users should also be aware that when reading registers state as the return value when the result of writing to the register.

CSROL in peripheral mode:

D7	D6
Serviced SetupEnd (self-clearing)	Serviced RxPktRdy (self-clearing)
w	w

D5	D4	D3	D2	D1	D0
SendStall (self-clearing)	SetupEnd	DataEnd (self-clearing)	SentStall	TxPktRdy (self-clearing)	RxPktRdy
w	r	w	rw	rw	r

Bit	Name	W/R	Description
D7	ServicedSetupEnd	W	The CPU writes a 1 to this bit to clear the SetupEnd bit. It will be cleared automatically.
D6	ServicedRxPktRdy	W	The CPU writes a 1 to this bit to clear the RxPktRdy bit. It will be cleared automatically.
D5	SendStall	W	The CPU writes a 1 to this bit to terminate the current transaction. STALL handshake will be sent sent, then the bit will be automatically cleared
D4	SetupEnd	R	The DATAEND bit has been set before the control transaction, this bit will be set. An interrupt will be generated at this time and the FIFO will be flushed. This bit is written by the CPU to 1 to SetupEnd bit is cleared.
D3	DataEnd	W	CPU sets this bit: 1. When setting TxPktRdy to last packet 2. Clear RxPktRdy when the last packet is unloaded 3. Set to TxPktRdy zero-length data packets It will be cleared automatically.
D2	SentStall	W/R	Set this bit when STALL handshake is sent, the CPU should clear this bit
D1	TxPktRdy	W/R	This bit is set after the CPU loads the packet into the FIFO, it is automatically cleared when A packet is sent and an interrupt is generated (if enabled)

D0	RxPktRdy	R	This bit is set when a data packet is received. When this bit is set, a an interruption. The CPU clears this bit by setting the ServicedRxPktRdy bit

CSROL in host mode:

D7	D6	D5	D4	D3	D2	D1	D0
NAK Timeout	StatusPkt	ReqPkt	Error	SetupPkt	RxStall	TxPktRdy	RxPktRdy
rc	rw	rw	rc	rc	rc	rw	rc

Bit	Name	W/R	Description
D7	NAK Timeout	C/R	When endpoint 0 receives a NAK response longer than the time set in the NAKLimit0 register this bit is set. The CPU should clear this bit to allow the endpoint to continue
D6	StatusPkt	W/R	CPU sets this bit at the same time as TxPktRdy or ReqPkt bit, executes Status stage transactions. Setting this bit ensures that the data toggle is set to 1, such that DATA1 packets are used for status phase transactions
D5	ReqPkt	W/R	CPU sets this bit to request an IN transaction, this bit is cleared when RxPktRdy is set
D4	Error	C/R	The C/R party has made three attempts and there is no response from the perimeter to execute the transaction, the The bit will be set and an interrupt will be generated. The CPU should clear this bit.
D3	SetupPkt		When the C/R CPU sets this bit as the TxPktRdy bit, it sends a transaction SETUP token in place of OUT token. Note: This bit is also cleared when data is toggled.
D2	RxStall	C/R	This bit is set when a STALL handshake is received. The CPU should clear this bit
D1	TxPktRdy	W/R	This bit is set after the CPU loads the data packet into the FIFO, when the data packet is sent it will Automatically cleared when an interrupt is generated (if enabled).
D0	RxPktRdy	C/R	This bit is set when a data packet is received, and an interrupt is generated at this time (if enabled can). The CPU clears this bit when a packet is read in the FIFO

CSR0H

CSR0H is an 8-bit register that provides control and status bits for endpoint 0.

Note: The interpretation of this register depends on whether the Air105 is acting as a peripheral or as a host. Users should also be aware that when reading registers state as the return value when the result of writing to the register.

CSROL in peripheral mode:

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	-	-	FlushFIFO (self_clearing)
r	r	r	r	r	r	r	w

Bit	Name	W/R	Description
D7-D1	-	R	not used, returns 0 on read
D0	FlushFIFO	W	The CPU writes a 1 to this bit to buffer the next transmit/read from endpoint 0 FIFO open data packet to start reading. FIFO pointer is reset, TxPktRdy/ RxPktRdy bits (and below) are cleared. Note: When the TxPktRdy/ **RxPktRdy bits are set** FlushFIFO is only used when it is set, at other times this may lead to data corruption

CSROL in host mode:

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	Dis Ping	Data Toggle Wr.Enable (self-clearing)	Data Toggle	FlushFIFO (self-clearing)
r	r	r	r	rw	w	rw	w

Bit	Name	W/R	Description
D7-D4	-	R	not used, returns 0 on read
D3	Dis Ping	W/R	The CPU writing a 1 to this bit will show that the core will not be at high speed in data and status Issue a PING instruction when control is transferred
D2	Data Toggle Wr.Enable	W	The CPU writes 1 to this bit, it will trigger the data state with the end bit of 0 and record it down (see the data bit is triggered, the same below). Once the new value is written, This bit is automatically cleared
D1	Data Toggle	W/R	When read by W/R, this bit indicates that endpoint 0 data will trigger. If D10 is high, the Bits may be written with the desired settings for data triggering. If D10 is low, Any value written to this bit will be ignored
D0	FlushFIFO	W	The CPU writes a 1 to this bit to flush the next packet sent from the endpoint 0 FIFO send/read. The FIFO pointer is reset, the TxPktRdy/ RxPktRdy bits (starting with down) is cleared **Note: When the TxPktRdy/** RxPktRdy bits are set, the FlushFIFO is The only ones that are used, at other times, may cause data corruption

COUNT0

COUNT0 is a 7-bit read-only register that indicates the number of bytes of data received in the endpoint 0 FIFO. When RxPktRdy (CSR0.DO) is set, returns the change of value as the content of change in FIFO.

D6	D5	D4	D3	D2	D1	D0
(MSB)			Endpoint 0 Rx Count			(LSB)
r	r	r	r	r	r	r

TYPE0

Note: in HOST mode!

D7	D6
Speed	
rw	rw

Bit	Name	W/R	Description
D7-D6	Speed	W/R	The operating speed of the target device: 00: Not used (Note: if checked, the target will be considered using the same connection speed as the core Heart) 10: All 11: low

CONFIGDATA

CONFIGDATA is an 8-bit read-only register that returns information about the selected core configuration.

D7	D6	D5	D4	D3	D2	D1	D0
MPRxE	MPTxE	BigEndian	HBRxE	HBTxE	DynFIFO	SoftConE	UTMI DataWidth
r	r	r	r	r	r	r	r

Bit	Name	W/R	Description
D7	MPRxE	R	When set to '1', automatic merging of bulk packets is selected
D6	MPTxE	R	When set to "1", automatic splitting of bulk packets is selected

D5	BigEndian	R is always 0, indicating little endian ordering
D4	HBRxE	R When set bit "1" indicates high bandwidth receive ISO endpoint support selection
D3	HBTxE	R When set bit "1" indicates high bandwidth receive ISO endpoint support selection
D2	DynFIFO Sizing	R When setting bit "1" indicates dynamic FIFO resizing option is selected
D1	SoftConE	R is always "1" for soft connect/disconnect
D0 UTMI DataWidth		R means select UTML + data width. Always 0, which means 8 bits

NAKLIMIT0

Note: in HOST mode

NAKLIMIT0 is a 5-bit register that sets the number of frames after endpoint 0 times out to receive the number of NAK response streams. (Can for other terminal equivalent setup and RxInterval registration via TxInterval)

The number of selected frames is 2(m-1). (where m is the value set in the register, valid values are 2-16), if the host receives a NAK

The endpoint will stall when there are more response targets than the limit set by the register.

Note: A value of 0 or 1 disables the NAK timeout function

D4	…	D0
Endpoint 0 NAK Limit(m)		(LSB)
rw	…	rw

TXMAXP

Offset address: 0x100+16*n Reset value: 0x0000

The TxMax register defines the maximum amount of data that can be transferred in a single operation through the selected TX endpoint. each TX endpoint (except endpoint 0) all have a TxMaxP register

D12/15		D11	D10	…	D0
m-1			(MSB)	Maximum Payload/transaction	(LSB)
rw	…	rw	rw	…	rw

This register can be set to a value of up to 1024 bytes, but isochronous transfers for interrupt and full-speed operations are subject to the packet size on the USB specification Small and batch constraints

With endpoint batching enabled in relation to the packet splitting option, the multiplier m can reach 32 and limit the maximum number of "USB" packets target (that is, a packet transmitted over USB), individual packets of the specified payload should be placed separately in the FIFO prior to transmission. (like

If endpoints related to packet splitting options are not enabled, D15-D13 are not implemented and D12-D11 (if included) are ignored)

NOTE: The data packet is required to be an exact multiple of the payload, indicated by bits 10 to 0, which itself needs to be 8, 16, 32, 64 or (in In the case of high-speed transmission) 512 bytes.

m can only be 2 or 3 (equivalent to bit 11, respectively, for endpoints that run synchronously/interrupt in high-speed mode and related to high-bandwidth option enablement)

sets and bit 12 sets) and specifies the maximum number of such transactions that can occur in a single microframe if either bit 11 or bit 12 is not zero, the Air105 will automatically split any data written to the FIFO into up to 2 or 3 USB packets, each containing the specified payload (or smaller). The maximum payload is 1024 bytes per transaction, thus allowing up to 3072 bytes to be transmitted per frame.

(Bits 11 and 12 are ignored in full speed mode or when isochronous transfer high bandwidth is not enabled).

The value written to this register represents the total amount of data (specified payload * M), which must not exceed the FIFO size set by the TX endpoint,

And should not exceed half the size of the FIFO when double buffering is required.

TXCSRL

Offset address: 0x102+16*n Reset value: 0x00

TXCSRL is an 8-bit register that provides control and status bits for transmission through the currently selected TX endpoint. TX per configuration Endpoints (excluding Endpoint 0) have a TXCSRL register

Note: The interpretation of this register depends on whether the Air105 is acting as a peripheral or as a host, the user should also pay attention when reading the register to reflect the status state as the return value when the result of writing to the register.

In peripheral mode:

D7	D6	D5	D4	D3	D2	D1	D0
IncompTx ClrDataTog SentStall			SentStall	FlushFIOFO (self-clearing)	UnderRun FIFO	NotEmpty	TxPktRdy
rc	w	rc	rw	w	rc	rc	rw

Bit	Name	W/R	Description
D7	IncompTx	C/R	When the endpoint is being used for high bandwidth, this bit is set to indicate that large packets have been Was split into 2 or 3 packets but no command to send all parts was received. Note: This bit will always return 0 on any asynchronous transfer
D6	ClrDataTog	W	CPU writes 1 to this bit to trigger endpoint data reset to 0
D5	SentStall	C/R	This bit is set when STALL handshake is sent. FIFO is flushed and TxPktRdy bit Cleared (see below), the CPU clears this bit
D4	SentStall	W/R	CPU writes 1 to this bit to issue STALL handshake instruction, CPU clears this bit to terminate the failure speed status. Note: This bit is not used by any endpoint for isochronous transfers
D3	FlushFIOFO	W	CPU writes 1 to this bit to flush the endpoint FIFO with the latest packet. FIFO pointer complex bit, the TxPktRdy bit (and later) is cleared and an interrupt is generated, while TxPktRdy can be set to abort a packet currently being loaded into the FIFO. Note: **FlushFIFO is only used when TxPktRdy is set, and may be used at other times lead to data corruption. Also note that if the FIFO is double buffered, the FlushFIFO** May need to be set twice to completely clear the FIFO
D2	UnderRun		USB sets this bit when C/R receives a command that TxPktRdy is not set. CPU should should clear this bit
D1	FIFO NotEmpty	C/R	USB sets this bit when there is at least one packet in the TX FIFO
D0	TxPktRdy		This bit is set after the W/R CPU loads a packet into the FIFO. When a packet is sent It is then automatically cleared, at which point an interrupt (if enabled) is generated. TxPktRdy also will automatically clear the second packet previously loaded into the FIFO for double buffering

In host mode:

D7	D6
NAK Timeout/ IncompTx	ClrDataTog
rc	w

D5	D4	D3	D2	D1	D0
SentStall	SentStall	FlushFIOFO (self-clearing)	UnderRun	FIFO NotEmpty	TxPktRdy
rc	rw	w	rc	rc	rw

Bit	Name	W/R	Description
D7	NAK Timeout/ IncompTx	C/R	only batch ports: the response time of receiving NAK at the TX endpoint is longer than the set This bit is set when the NAK limit of the TxInterval register is long. CPU clear Clear this bit to allow the endpoint to continue. Only high bandwidth interrupt endpoints: this bit will be set if a packet is sent to the receiver When the device is not responding
D6	ClrDataTog	W	CPU writes 1 to this bit to trigger endpoint data reset to 0
D5	SentStall	C/R	Set this bit when STALL handshake is received. This bit is set, any When the DMA request in the row is stopped, the FIFO is completely flushed and the TxPktRdy bit is cleared zero (see below), the CPU clears this bit
D4	SentStall	W/R	is set at TxPktRdy bit, send a SETUP command instead of OUT command At the same time, the CPU sets this bit. Note: Setting this bit also clears the data toggle
D3	FlushFIOFO	W	CPU writes 1 to this bit to flush the endpoint FIFO with the latest packet. FIFO pointer complex

			bit, the TxPktRdy bit (and later) is cleared and an interrupt is generated, while TxPktRdy can be set to abort a packet currently being loaded into the FIFO. Note: **FlushFIFO is only used when TxPktRdy is set, and may be used at other times lead to data corruption. Also note that if the FIFO is double buffered, the FlushFIFO** May need to be set twice to completely clear the FIFO
D2	UnderRun	R/C has 3 attempts to send a packet and no handshake message is received when the USB device	Set this bit. An interrupt is generated when this bit is set, TxPktRdy is cleared, the FIFO Completely refreshed. The CPU should clear this bit only if the endpoint is operating in bulk or interrupt mode Time
D1	FIFO NotEmpty	R/C USB sets this bit when there is at least one packet in the TX FIFO	
D0	TxPktRdy	This bit is set after the W/R CPU loads a packet into the FIFO. When a packet is sent	It is then automatically cleared, at which point an interrupt (if enabled) is generated. TxPktRdy also will automatically clear the second packet previously loaded into the FIFO for double buffering

TXCSRH

Offset Address: 0x103+16*n Reset value: 0x00

TXCSRH is an 8-bit register that provides additional control for transmission through the currently selected TX endpoint. Each TX endpoint (not Including endpoint 0) are configured with a TXCSRH register.

Note: The interpretation of this register depends on whether the Air105 is acting as a peripheral or host. The user should also be aware that when reading registers reflect the status of the The return value when the result of writing to the register.

In peripheral mode:

D7	D6	D5	D4	D3	D2	D1	D0
AutoSet	ISO	Mode	DMAReqEnab	FrcDataTog	DMAReqMode	-	-
rw	rw	rw	rw	rw	rw	r	r

Bit	Name	W/R	Description
D7	AutoSet	W/R	If the CPU sets this bit, TxPktRdy will be loaded into the TX after the largest packet Automatically set when FIFO. If the loaded packet is smaller than the maximum packet, TxPktRdy Manual setup required Note: Not set for any high bandwidth isochronous endpoint and high bandwidth interrupt endpoint
D6	ISO	W/R	CPU sets this bit to initiate a TX endpoint isochronous transfer and clear the TX endpoint's bulk Enable or disable transfers. Note: This bit only works in peripheral mode, in host mode, it always returns 0
D5	Mode	W/R	CPU setting this bit makes the endpoint point to TX, clearing this bit makes the endpoint point to RX **Note: This bit only works when the same endpoint FIFO is used for TX and RX transactions**
D4	DMAReqEnab	W/R	CPU sets this bit to enable TX endpoint by DMA request
D3	FrcDataTog	W/R	CPU sets this bit to force endpoint data to trigger switching, regardless of whether it is received or not ACK, the data packets in the FIFO are cleared. This can be communicated via a synchronous endpoint rate feedback to interrupt the TX endpoint.
D2	DMAReqMode	R	CPU sets this bit to select DMA request mode 1 and clears it to select DMA request Mode 0 Note: This bit cannot be cleared at the same time as and before DMAReqEnab is cleared
D1-D0	-	R	not used, always returns 0

In host mode:

D7	D6	D5	D4	D3	D2	D1	D0
AutoSet	-	Mode	DMAReqEnab	FrcDataTog	DMAReqMode	Data Toggle Wr.Enable (self-clearing)	Data Toggle
rw	r	rw	rw	rw	rw	w	rw

Bit	Name	W/R	Description
D7	AutoSet	W/R	If the CPU sets this bit, TxPktRdy will be loaded into the TX after the largest packet Automatically set when FIFO. If the loaded packet is smaller than the maximum packet, TxPktRdy Manual setup required Note: Not set for any high bandwidth isochronous endpoint and high bandwidth interrupt endpoint
D6	-	R	not used, always returns 0
D5	Mode	W/R	CPU setting this bit makes the endpoint point to TX, clearing this bit makes the endpoint point to RX **Note: This bit only works when the same endpoint FIFO is used for TX and RX transactions**
D4	DMAReqEnab	W/R	CPU sets this bit to enable TX endpoint by DMA request
D3	FrcDataTog	W/R	CPU sets this bit to force endpoint data to trigger switching, regardless of whether it is received or not ACK, the data packets in the FIFO are cleared. This can be communicated via a synchronous endpoint rate feedback to interrupt the TX endpoint.
D2	DMAReqMode	W/R	CPU sets this bit to select DMA request mode 1 and clears it to select DMA request Mode 0 Note: This bit cannot be cleared at the same time as and before DMAReqEnab is cleared
D1	Data Toggle Write Enable	W	CPU writes 1 to this bit to make TX endpoint data trigger write (see Data trigger bit, below same), this bit will be automatically cleared once a new value is written.
D0	Data Toggle	W/R	When read, this bit indicates the current state of the TX endpoint data trigger. If D1 is high, This bit may be written with the desired settings for data triggering. If D1 is low, write this Any value of the bit will be ignored

RXMAXP

Offset Address: 0x104+16*n The Reset value: 0x0000

RXMAXP register defines the maximum amount of data that can be manipulated by the selected receiving endpoint in a single transfer. each
Receive endpoints (except endpoint 0) have an RXMAXP register.

D12/15	…	D11	D10	…	D0
m-1			(MSB)	Maximum Payload/transaction	(LSB)
rw	…	rw	rw	…	rw

Bits 10:0 define the maximum payload (bytes) transferred in one transaction. The set value can be up to 1024 bytes, but is limited by
The USB specification places a batch limit on the packet size of isochronous transfers for interrupt and full-speed operation.

With endpoint batching enabled in relation to the packet splitting option, the multiplier m can be up to 32 and the limit is merged into the FIFO specified
Maximum number of USB single packets of payload. (D15-D13 are not implemented if the group split option is not enabled,
D12-D11 (if included) are ignored)

The value written from bit 0 to bit 10 (multiplied by m in the case of high-bandwidth isochronous transfers) must match the relevant endpoints described in the standard endpoint
Point to the value given in the wMaxPacketSize field. Mismatches can lead to unexpected results.

The value written to this register represents the total amount of data (specified payload * M), which must not exceed the FIFO size of the RX endpoint,
If double buffered, it should not exceed half the FIFO size.

Note: RxMaxP must be set to an even number of bytes for DMA mode 1 that generates interrupts

RXCSRL

Offset address: 0x106+16*n Reset value: 0x00

RXCSRL is an 8-bit register that provides control and status bits for the current transfer through the selected receiving endpoint. connection for each configuration
Receive endpoints (excluding endpoint 0) are configured with an RXCSRL register.

Note: The interpretation of this register depends on whether the Air105 is acting as a peripheral or host. The user should also be aware that when reading registers reflect the status of the
The return value when the result of writing to the register.

In peripheral mode:

D7	D6	D5
ClrDataTog	SentStall	SendStall
w	rc	rw

D4	D3	D2	D1	D0
FlushFIFO (self-clearing)	DataError	OverRun	FIFOFull (self-clearing)	RxPktRdy
w	r	rc	r	rc

Bit	Name	W/R	Description
D7	ClrDataTog	W	CPU writes 1 to this bit to trigger data, reset to 0
D6	SentStall	C/R	This bit is set when STALL handshake is sent, CPU should clear this bit
D5	SendStall	W/R	CPU writes 1 to this bit to issue STALL handshake, CPU clears this bit to terminate stall condition state. Note: This bit does not have any endpoints being used for synchronous transfer effects
D4	FlushFIFO	W	CPU writes 1 to this bit to receive and read the next buffered packet from the endpoint FIFO, The FIFO is reset and the RxPktRdy bit (and below) is cleared. **Note: If the FIFO is double buffered, the FlushFIFO may need to be set twice to be fully clear FIFO**
D3	DataError	R	When the packet contains CRCRxPktRdy is set or a bit stuffing error occurs This bit is set when RxPktRdy is cleared Note: This bit operates only in IOS mode, in batch mode it always returns 0
D2	OverRun	C/R	This bit is set when the OUT packet cannot be loaded into the RxFIFO. The CPU should clear this bit. Note: This bit operates only in IOS mode, in batch mode it always returns 0
D1	FIFOFull	R	This bit is set when no data is packed into the RxFIFO
D0	RxPktRdy		This bit is set when the C/R receives a data packet, and is cleared by the CPU when the packet is unloaded in the FIFO bit, will generate an interrupt

In host mode:

D7	D6	D5
ClrdataTog	RxStall	ReqPkt
w	rc	rw

D4	D3	D2	D1	D0
FlushFIFO (self-clearing)	DataError/ NAK Timeout	Error	FIFOFull (self-clearing)	RxPktRdy
w	rc	rc	r	rc

Bit	Name	W/R	Description
D7	ClrdataTog	W	CPU writes 1 to this bit to trigger data, reset to 0
D6	RxStall	C/R	This bit is set when STALL handshake is sent, CPU should clear this bit
D5	ReqPkt	W/R	CPU writes a 1 to this bit to request a transaction. This bit is cleared when RxPktRdy is set zero
D4	FlushFIFO	W	CPU writes 1 to this bit to receive and read the next buffered packet from the endpoint FIFO, The FIFO is reset and the RxPktRdy bit (and below) is cleared. **Note: FlushFIFO is only used when RxPktRdy is set, at other times, it may result in data corruption. If the FIFO is double buffered, the FlushFIFO may** Need to set twice to fully clear the FIFO
D3	DataError/ NAK Timeout	C/R	works in ISO mode, when the packet contains CRCRxPktRdy is set or sent This bit is set when a bit stuffing error RxPktRdy is cleared. In batch mode, When a NAK response is received that is less than the NAK limit set by the RxInterval register This bit will be set during long pauses. The CPU clears this bit to allow the endpoint to continue. However, if double-packet buffering is enabled alone, no further transmission is allowed. in this situation In this case, the reqpkt bit should also be set at the same cycle to clear this bit
D2	Error	C/R	This bit is set by the USB when 3 attempts to receive data have been made but no packet has been received. The CPU clears this bit and an interrupt is generated when this bit is set.

D1	FIFOFull	R This bit is set when no data is packed into the RxFIFO
D0	RxPktRdy	This bit is set when the C/R receives a data packet, and is cleared by the CPU when the packet is unloaded in the FIFO bit, will generate an interrupt

RXCSRH

Offset address: 0x107+16*n Reset value: 0x00

RXCSRH is an 8-bit register that provides additional control and status bit transfers through the currently selected receiving terminal. each receiving end All endpoints (excluding endpoint 0) are configured with an RXCSRH register.

Note: The interpretation of this register depends on whether the Air105 is acting as a peripheral or host. The user should also be aware that when reading registers reflect the status of the The return value when the result of writing to the register.

In peripheral mode:

D7	D6	D5	D4	D3	D2	D1	D0
AutoClear	ISO	DMAReqEnab	DisNyet /PIDError	DMAReqMode	-	-	IncompRx
rw	rw	rw	rw	rw	r	r	rc

Bit	Name	W/R	Description
D7	AutoClear	W/R	When the RxMaxP byte packet is unloaded from the RX FIFO endpoint, RxPktRdy is When auto-cleared, the CPU sets this bit. When packets smaller than the largest packet are unloaded , RxPktRdy needs to be cleared manually. When using a DMA to unload the RXFIFO, Regardless of RxMaxP, data is read from RXFIFO in blocks of four bytes NOTE: Should not be set for high bandwidth isochronous endpoints
D6	ISO	W/R	CPU sets this bit to receive isochronous transfers from endpoints and clears this bit to receive Endpoint block or interrupt transfer
D5	DMAReqEnab	W/R	CPU sets this bit when DMA requests RX endpoint
D4	DisNyet /PIDError	W/R	Split/Break Transaction: The CPU sets this bit to disable sending the NYET handshake. set up After all successfully received packets including FIFO full point are set to ACK'd Note: This bit has effect only in high speed mode, in this mode it should Set all interrupt endpoints. ISO Transaction: The core sets this bit to indicate the PID error of the received packet
D3	DMAReqMode	W/R	CPU set this bit to select DMA request mode 1, clear this bit to select DMA request mode 0
D2-D1	-	R	not used, always returns 0
D0	IncompRx	C/R	This bit is set on a high-bandwidth sync/interrupt transfer if there is no receiver section due to Sub-data, RxFIFO packets are incomplete. This bit is when RxPktRdy is cleared also cleared Note: This bit always returns 0 on any asynchronous transfer

In host mode:

D7	D6	D5	D4	D3	D2	D1	D0
AutoClear	AutoReq	DMAReqEnab	PID Error	DMAReqMode	Data Toggle Wr. Enable (self-clearing)	Data Toggle	IncompRx
rw	rw	rw	r	rw	r	r	rc

Bit	Name	W/R	Description
D7	AutoClear	W/R	RxPktRdy when the RxMaxP byte packet is unloaded from the RX FIFO endpoint When cleared automatically, the CPU sets this bit. When the grouping is smaller than the largest grouping When uninstalled, RxPktRdy needs to be cleared manually. When using a DMA offload When RXFIFO, regardless of RxMaxP, data is read four bytes from RXFIFO piece

			NOTE: Should not be set for high bandwidth isochronous endpoints
D6	AutoReq	The ReqPkt bit is automatically set when the W/R RxPktRdy bit is cleared, and the CPU sets this bit.	Note: This bit is automatically cleared when a short message is received.
D5	DMAReqEnab	W/R CPU sets this bit when DMA requests RX endpoint	
D4	PID Error	R ISO transaction: The core sets this bit to indicate the PID error of the received packet	Split/Break Transaction: This bit setting is ignored
D3	DMAReqMode	W/R CPU set this bit to select DMA request mode 1, clear this bit to select	DMA request mode 0
D2	Data Toggle Wr. Enable	R CPU writes 1 to this bit, triggering the current state data write of endpoint 0 (see Data	trigger bit, the same below). This bit is automatically cleared once a new value is written.
D1	Data Toggle	R When read, this bit indicates the current state of endpoint 0 data trigger. if D10	high, this bit may be written to toggle the desired setting if D10 low, any value written to this bit will be ignored
D0	IncompRx	C/R This bit receives an incomplete packet during high-bandwidth isochronous or interrupted transfers	time setting. This bit is cleared when RxPktRdy is cleared. Note: This bit should not be set if the USB protocol is followed. (in any non-
			Under synchronous transmission, this bit always returns 0)

RXCOUNT

Offset address: 0x108+16*n Reset value: 0x0000

RXCOUNT is a 14-bit read-only register that reserves the number of data packets currently read by the Rx FIFO in the data byte. if
The number given will be used to combine the packets when the packet is sent as multiple packets.

Note: When RxPktRDY(RxCSR.D0) is set, the returned value changed to FIFO unload is only valid

D13	…	D0
(MSB)	Endpoint Rx Count	(LSB)
r	…	r

TXTYPE

Offset address: 0x10A+16*n Reset value: 0x00

TXTYPE is an 8-bit register written to the endpoint's target endpoint number, the transaction protocol for the currently selected TX endpoint, and its operation.
line speed. Each configured TX endpoint (endpoint 0 has its own register type, 1Ah) has a TxType register

D7	D6	D5	D4	D3	D2	D1	D0
*Speed		Protocol		Target Endpoint Number			
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D7-D6	Speed	W/R The operating speed of the target device when the core is configured with the multidrop option:	00: Not used (Note: if checked, the target will be considered to be used with the core with the same connection speed) 10: All 11: low These bits cannot be accessed when the kernel is not configured with the multidrop option
D5-D4	Protocol	W/R The CPU sets this selection to the protocol required by the receiving terminal:	00: Control 01: Sync 10: Bulk 11: Interrupt
D3-D0	Target Endpoint	W/R CPU should set this value to include return to Air105 during device enumeration	The endpoint number of the RX endpoint description

	Number		

TXINTERVAL

Offset address: 0x10B+16*n Reset value: 0x00

TXINTERVAL is an 8-bit register used for interrupt and synchronous transmission, which defines the polling interval for the current selectRx endpoint. for batch Quantum endpoint, after this register sets the frame, the endpoint should timeout to receive the number of NAK response streams.

D7	…	D0
Tx Polling Interval/NAK Limit (m)		
rw	…	rw

Transfer Type	Speed low	Valid values (m) 1 –	Interpretation
Interrupt	speed or full speed	255 1 – 16 2 –	The rotation interval is m frames
Isochronous	full speed full	16	The rotation interval is 2(m-1) frames
Bulk	speed		NAK is limited to 2(m-1) frames Note: 0 or 1 The value of disable NAK-timeout FUNC

RXTYPE

Offset address: 0x10C+16*n RXTYPE Reset value: 0x00

is an 8-bit register written to the endpoint target endpoint number, this transaction protocol is used for the currently selected receive endpoint, and its operation speed. Each configured receive endpoint (endpoint 0 has its own register type) has an RxType register

D7	D6	D5	D4	D3	D2	D1	D0
*Speed		Protocol		Target Endpoint Number			
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D7-D6	Speed	W/R	The operating speed of the target device when the core is configured with the multidrop option: 00: Not used (Note: if checked, the target will be considered to be used with the core same connection speed) 10: All 11: low These bits cannot be accessed when the kernel is not configured with the multidrop option
D5-D4	Protocol	W/R	The CPU sets this selection to the protocol required by the receiving terminal: 00: Control 01: Sync 10: Bulk 11: Interrupt
D3-D0	Target Endpoint Number	W/R	CPU should set this value to include return to Air105 at RX during device enumeration The endpoint number of the endpoint description

RXINTERVAL

Offset address: 0x10D+16*n Reset value: 0x00

RXINTERVAL is an 8-bit register used for interrupt and synchronous transfer, defining the polling interval for the current selectRx endpoint. for batch Quantum endpoint, after this register sets the frame, the endpoint should timeout to receive the number of NAK response streams. Every receiving endpoint (except endpoint 0) are configured with an RXINTERVAL register

D7	…	D0
Rx Polling Interval/NAK Limit (m)		

rw	…		rw

Transfer Type	Speed low	Valid values (m) 1 –	Interpretation
Interrupt	speed or full speed	255 1 – 16 2	The rotation interval is m frames
Isochronous	full speed full	– 16	The rotation interval is 2(m-1) frames
Bulk	speed		NAK limited to 2(m-1) frames Note: value of 0 or 1 Disable NAK Timeout FUNC

FIFOSIZE

Offset address: 0x10F+16*n Reset value: 0x00

FIFOSIZE is an 8-bit read-only register that returns the size of the FIFO associated with the selected additional TX/RX endpoint. lower nibble Line encodes the size of the selected TX endpoint FIFO, and the upper nibble encodes the selected receive endpoint FIFO size. 3-13 values Corresponds to a FIFO size of 2^n bytes (8-8192 bytes). If the endpoint has not been configured, a value of 0 is displayed, where TX and RX Endpoints share the same FIFO, the size of the RX FIFO will be encoded as 0xF

Note: This register only has this interpretation when the index register setting selects one of endpoints 1-15 and dynamic resizing is not selected. It has a special interpretation when the index register is set to select endpoint 0, and the returned result is invalid, where the dynamic FIFO size is used.

D7	…	D4	D3	…	D0
Rx FIFO Size			Tx FIFO Size		
r	…	r	r	…	r

18.4.4 FIFOx

Offset address: 0x20-0x5F Reset value: 0x00000000

This address range provides 16 addresses for the CPU to access the FIFO of each endpoint. Writing these addresses loads data into the TXFIFO. corresponding endpoint. From these addresses data can be read or unloaded from the corresponding endpoint in the RXFIFO

The address range is 20H-5Fh and the FIFOs are located on 32-bit doubleword boundaries (endpoints 0-20, endpoints 1-24, endpoint 15 at 5Ch)

Note: (i) The round-trip FIFO can be set to 8-bit, 16-bit or 32-bit as desired, and access any combination of allowed data provided

Access is sequential. However, transfers associated with a packet must be of the same width so that the data is consistent byte by byte,

Word processing or double word alignment. However, the last transfer may be longer than previous transfers to complete an odd byte or word transfer. fewer bytes

(ii) Depending on the size of the FIFO and the desired maximum packet size, the FIFO supports single-packet or double-packet buffering. but, Burst write of multiple groups that need to be set after writing is not supported

(iii) Endpoints 1-15 with STALL response or TX strike error, the related FIFO is completely flushed

18.4.5 Additional Multipoint Control/Status registers

TXFUNCADDR/RXFUNCADDR

Offset address: 0x80+8*n/0x84+8*n Reset value: 0x00/0x00

TXFUNCADDR/RXFUNCADDR is a 7-bit read/write register that records that the target function is accessed through the associated endpoint (EPN). ask for the address. TXFUNCADDR needs to be defined for each TX endpoint used, and RXFUNCADDR needs to be defined for each used RX Receive Endpoint Definition

Note: TXFUNCADDR must be defined for endpoint 0, endpoint 0 does not exist on the RXFUNCADDR register

D6	…	D0
Address of Target Function		
rw	…	rw

TXHUBADDR/RXHUBADDR

Offset address: 0x82+8*n/0x86+8*n Reset value: 0x00/0x00

Note: only in host mode

TXHUBADDR/RXHUBADDR are 8-bit read/write registers, like TxHubPort and RxHubPort, only need to be written through

The full-speed device of the USB2.0 hub is connected to the TX/RX endpoint EPN, and the necessary transactions are performed to convert between full-speed/low-speed transmission.

D7	D6	…	D0
Multiple Translators	Hub Address		
rw	rw	…	rw

TXHUBPORT/RXHUBPORT

Offset address: 0x83+8*n/0x87+8*n Reset value: 0x00/0x00

TXHUBPORT and RXHUBPORT only need to be written in which full-speed or low-speed devices carry the necessary through a high-speed USB2.0 hub

Transaction transitions are connected to the EPN of the TX/RX endpoint. In this case, these 7-bit read/write registers need to be used to record the

The endpoint is associated with the target USB2.0 hub port being accessed

D6	…	D0
Hub Port		
rw	…	rw

18.4.6 Additional Control/Status registers

VCONTROL

Offset Address: 0x68 Reset Value: - VCONTROL is an optional vendor

register included in the UTMI+ PHY of the configured chip, its size is also configurable,

Up to 32 bits. The structure of the registers is designed by the system designer. Although the user should note that the VCONTROL register is controlled by UTML+

canonical definition

VSTATUS

Offset Address: 0x68 Reset Value: - VSTATUS is an optional vendor

register included in the UTMI+ PHY of the configured chip, its size is also configurable, high

up to 32 bits. The structure of the registers is designed by the system designer. Although the user should note that the VSTATUS register is specified by the UTML+ specification

defined, but this register can be accessed at address 68H

Users should also note that:

(1) VSTATUS inputs a bus sampling every 6 XCLK cycles

(2) The delay between the PHY changing the VSTATUS input bus and the new value reads 2Hc+Xc from the VSTATUS register to Between 3HC+6Xc. where Hc is one cycle of XCLK

HWVERS

Offset address: 0x6C Reset value: Version dependent

The HWVERS register is a 16-bit read-only register from which information about the core hardware, especially the RTL version, is generated and returned

number (vxx.yyy) and version information

D15	D14	…	D10	D9	…	D0
RC	xx			yyy		
r	r	…	r	r	…	r

Bit	Name	W/R	Description
D15	RC		R set to '1' if it is a release candidate core RTL rather than a full version Core

D14-D10	xx	R major version number (range 0-31)
D9-D0	yyy	R minor version number (range 0-999)

18.4.7 Configuration registers

EPINFO

Offset Address: 0x78 This Reset value: Implementation dependent

8-bit read-only register allows the number of TX and RX endpoints included in the design to read back

D7	D6	D5	D4	D3	D2	D1	D0
RxEndPoints				TEndPoints			
r	r	r	r	r	r	r	r

Bit	Name	W/R	Description
D7-D4	RxEndPoints		The number of R received, implemented at the designed endpoint
D3-D0	TEndPoints		The number of R TX, implemented at the endpoint of the design

RAMINFO

Offset Address: 0x79 This Reset value: Implementation dependent

8-bit read-only register provides information on the width of the RAM

D7	D6	D5	D4	D3	D2	D1	D0
DMAChans				RamBits			
r	r	r	r	r	r	r	r

Bit	Name	W/R	Description
D7-D4	DMAChans	R	Number of DMA channels implemented in the design
D3-D0	RamBits	R	RAM address bus width

LINKINFO

Offset address: 0x7A Reset value: 0x5C

This 8-bit register allows to specify some delays

D7	D6	D5	D4	D3	D2	D1	D0
WTCON				WTID			
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D7-D4	WTCON	W/R	Sets the wait to apply for user connect/disconnect filtering in 533.3ns units device (default setting corresponds to 2.667us)
D3-D0	WTID	W/R	set delay from IDPULLUP application is asserted IDDIG is considered valid Units of 4.369ms (default setting corresponds to 52.43ms)

VPLEN

Offset address: 0x7B Reset value: 0x3C

This 8-bit register sets the duration of the VBUS pulse charging

D7	D6	D5	D4	D3	D2	D1	D0
VPLEN							
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D7-D0	VPLEN	W/R	Sets the duration of the VBUS pulse charge in 546.1 microsecond bit units (default The setting corresponds to 32.77ms)

FS_EOF1

Offset address: 0x7D Reset value: 0x77

The minimum time interval set by this 8-bit register can be between the start of the last transaction and the EOF full-speed transaction

D7	D6	D5	D4	D3	D2	D1	D0
(msb)			FS_EOF1				(lsb)
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D7-D0	FS_EOF1	W/R	sets the time for a full-speed transaction to stop EOF and start a new transaction in units of 533.3ns (The default setting corresponds to 63.46us)

LS_EOF1

Offset address: 0x7E Reset value: 0x72

This 8-bit register sets the minimum time interval between the start of the last transaction and the EOF low-speed transaction.

D7	D6	D5	D4	D3	D2	D1	D0
(msb)			LS_EOF1				(lsb)
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D7-D0	LS_EOF1	W/R	sets the time for a low-speed transaction to stop EOF and start a new transaction in 1.067ns bit (default setting corresponds to 121.6us)

SOFT_RST

Offset address: 0x7F Reset value: 0x00

This 8-bit register will deassert the output reset signals NRSTO and NRSTOX. This register clears itself and is reset by entering NRST

Unused						D1	D0
(msb)	SOFT_RST						(lsb)
						rw	rw

Bit	Name	W/R	Description
D7-D2	-	-	reserved, always returns 0
D1	NRSTX	W/R	The default value of this bit is 1'b0; when a 1 is written to this bit, the output NRSTXO will be Minimum delay of 7 cycles on the CLK input within the set (low). NRSTXO Will be asserted asynchronously and output synchronously, pulled high with respect to XCLK. the deposit device is self-clearing, and input NRST resets
D0	NRST	W/R	The default value of this bit is 1'b0; when a 1 is written to this bit, the output NRSTXO will be Minimum delay of 7 cycles on the CLK input within the set (low). NRSTXO Will be asserted asynchronously and output synchronously, pulled high with respect to XCLK. the deposit

			device is self-clearing, and input NRST resets

18.4.8 Extended registers

RQPKTCOUNT

Offset address: 0x300+2*n Reset value: 0x0000

For each receiving endpoint 1-15, Air105 provides a 16-bit register RQPKTCOUNT. This read/write register is used in the master
In machine mode, to specify the number of packets N to be transferred to the RX endpoint in blocks of one or more bulk packets of length MAXP.
The core uses the value recorded in this register to determine the issuance of the request where the AutoReq option (included in the RxCSR register) is set
set quantity.

D15	…	D0
(msb)	RqPktCount	(lsb)
rw	…	rw

Bit	Name	W/R	Description
D15-D0	RqPktCount	W/R	Sets the MAXP size of the number of packets transferred in chunks, when AutoReq is not set Only used in host mode

Double Buffered Packet Disable

18.4.8.1.1 RX DPKTBUFDIS

Offset address: 0x340 Reset value: 0x0000

RX DPKTBUFDIS is a 16-bit register that indicates which endpoints received have disabled double packet buffering for the Air105 product specification
Note: No configured endpoint bits can be asserted by writing a "1" to the respective register, but disabling bits will not have any significant effect

D15	D14	D13	D12	D11	D10	D9	D8
EP15 RxDis	EP14 RxDis	EP13 RxDis	EP12 RxDis	EP11 RxDis	EP10 RxDis	EP9 RxDis	EP8 RxDis
rw	rw	rw	rw	rw	rw	rw	rw

D7	D6	D5	D4	D3	D2	D1	D0
EP7 RxDis	EP6 RxDis	EP5 RxDis	EP4 RxDis	EP3 RxDis	EP2 RxDis	EP1 RxDis	EP0 RxDis
rw	rw	rw	rw	rw	rw	rw	r

Bit	Name	W/R	Description
D15	EP15 RxDis	W/R	Rx endpoint 15 interrupt
D14	EP14 RxDis	W/R	Rx endpoint 14 interrupt
D13	EP13 RxDis	W/R	Rx endpoint 13 interrupt
D12	EP12 RxDis	W/R	Rx endpoint 12 interrupt
D11	EP11 RxDis	W/R	Rx endpoint 11 interrupt
D10	EP10 RxDis	W/R	Rx endpoint 10 interrupt
D9	EP9 RxDis	W/R	Rx endpoint 9 interrupt
D8	EP8 RxDis	W/R	Rx endpoint 8 interrupt
D7	EP7 RxDis	W/R	Rx endpoint 7 interrupt
D6	EP6 RxDis	W/R	Rx endpoint 6 interrupt
D5	EP5 RxDis	W/R	Rx endpoint 5 interrupt
D4	EP4 RxDis	W/R	Rx endpoint 4 interrupt
D3	EP3 RxDis	W/R	Rx endpoint 3 interrupt
D2	EP2 RxDis	W/R	Rx endpoint 2 interrupt

D1	EP1 RxDis	W/R	Rx endpoint 1 interrupt
D0	Unused	R	reserved

18.4.8.1.2 TX DPKTBUFDIS

Offset address: 0x342 Reset value: 0x0000

TX DPKTBUFDIS is a 16-bit register that indicates which endpoint of the TX has disabled the double packet buffering feature of the Air105 product specification

Note: No configured endpoint bits can be asserted by writing a "1" to the respective register, but disabling bits will not have any significant effect

D15	D14	D13	D12	D11	D10	D9	D8
EP15 TxDis	EP14 TxDis	EP13 TxDis	EP12 TxDis	EP11 TxDis	EP10 TxDis	EP9 TxDis	EP8 TxDis
rw	rw	rw	rw	rw	rw	rw	rw

D7	D6	D5	D4	D3	D2	D1	D0
EP7 TxDis	EP6 TxDis	EP5 TxDis	EP4 TxDis	EP3 TxDis	EP2 TxDis	EP1 TxDis	EP0 TxDis
rw	rw	rw	rw	rw	rw	rw	r

Bit	Name	W/R	Description
D15	EP15 TxDis	W/R	Tx endpoint 15 interrupt
D14	EP14 TxDis	W/R	Tx endpoint 14 interrupt
D13	EP13 TxDis	W/R	Tx endpoint 13 interrupt
D12	EP12 TxDis	W/R	Tx endpoint 12 interrupt
D11	EP11 TxDis	W/R	Tx endpoint 11 interrupt
D10	EP10 TxDis	W/R	Tx endpoint 10 interrupt
D9	EP9 TxDis	W/R	Tx endpoint 9 interrupt
D8	EP8 TxDis	W/R	Tx endpoint 8 interrupt
D7	EP7 TxDis	W/R	Tx endpoint 7 interrupt
D6	EP6 TxDis	W/R	Tx endpoint 6 interrupt
D5	EP5 TxDis	W/R	Tx endpoint 5 interrupt
D4	EP4 TxDis	W/R	Tx endpoint 4 interrupt
D3	EP3 TxDis	W/R	Tx endpoint 3 interrupt
D2	EP2 TxDis	W/R	Tx endpoint 2 interrupt
D1	EP1 TxDis	W/R	Tx endpoint 1 interrupt
D0	EP0 TxDis	R	reserved

C_T_UCH

Offset address: 0x344 Reset value: Various

This register sets the delay to return the UTM to normal operating mode at the end of the high-speed resume signal (as master). Multiply this number by 4 represents the number of cycles of XCLK before the timeout occurs. That is, if XCLK is 30MHz, this number represents the 33.3ns number of time intervals. If XCLK is 60MHz, this number represents the number of 16.7ns interval before timeout. although

Although this bit is written by the host in the CLK field, the counter using this value in the XCLK field has no time-domain cross setting, and is registered here.

The value in the compiler is static, the default value is the value of the compiler directive of the same name located in the configuration file musbhsfc_xcfg.v.

D15	…	D0
C_T_UCH[15:8]		
rw	…	rw

Bit	Name	W/R	Description

D15-D0 C_T_UCH		The W/R restores the UTM normal operating mode from the delayed recovery signaling at the end of the high speed. default	The value is determined by compiler directives in the musbhsfc_xcfg.v file. if the host When the PHY data width is 16 bits, the default value is 2F3h (XCLK is 30MHz), When the PHY data width is 8 bits corresponding to 100us delay, the default value is 5E6H

18.4.9 DMA registers

DMA_INTR

Offset Address: 0x200 Reset value: 0x00

This register provides interrupts for each DMA channel. When read, this interrupt register is cleared. When any bit of this register is set, The output DMA_NINT is set low, causing an interrupt event to occur, as described in Section 17 (Description of the optional DMA controller) Bright. This register will be set in the DMA interrupt enable bit corresponding to the channel enable.

D7	D6	D5	D4	D3	D2	D1	D0
DMA_INTR[7:0]							
rw	rw	rw	rw	rw	r	r	r

Bit	Name	W/R	Description
D7	CH8 DMA_INTR	W/R	Channel 8DMA Interrupt
D6	CH7 DMA_INTR	W/R	Channel 7DMA interrupt
D5	CH6 DMA_INTR	W/R	Channel 6DMA interrupt
D4	CH5 DMA_INTR	W/R	Channel 5DMA interrupt
D3	CH4 DMA_INTR	W/R	Channel 4DMA interrupt
D2	CH3 DMA_INTR	R	Channel 3DMA interrupt
D1	CH2 DMA_INTR	R	Channel 2DMA interrupt
D0	CH1 DMA_INTR	R	Channel 1DMA interrupt

DMA_CNTL

Offset address: 0x204 Reset value: 0x0000

This register is only used when the Air105 is configured to use at least one internal DMA channel. This register provides DMA for each channel Transmission Control. Enable, transfer direction, transfer mode and DMA burst mode are all controlled by this register.

D10 D9	D8	D7 D6	D5	D4	D3	D2	D1	D0
DMA_BRSTM	DMA_ERR	DMAEP			DMAIE	DMAMODE	DMA_DIR	DMA_EN
rw rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D10-D9	DMA_BRSTM	W/R	Burst Mode: 00=Burst Mode 0: Burst of unspecified length 01=Burst Mode 1: INCR4 or unspecified length 10=Burst Mode 2: INCR8, INCR4 or unspecified length 11=Burst Mode 3: INCR16, INCR8, INCR4 or unspecified length
D8	DMA_ERR	W/R	Bus Error Bit: Input indicating that a bus error has been observed AHB_HRESPM[1:0]. This bit is cleared by software
D7-D4	DMAEP	W/R	This channel is assigned to this endpoint number
D3	DMAIE	W/R	DMA Interrupt Enable
D2	DMAMODE	W/R	This bit selects the DMA transfer mode: 0=DMA mode 0 transfer

			1=DMA mode 1 transfer
D1	DMA_DIR	W/R	This bit selects the DMA transfer direction: 0=DMA write (RX endpoint) 1 = DMA read (TX endpoint)
D0	DMA_EN	W/R	This bit will initiate a DMA transfer and will cause the transfer to begin

DMA_ADDR

Offset Address: 0x208 Reset value: 0x00000000

This register identifies the current memory address of the corresponding DAM channel. The initial memory address written to this register must have a value, Make its value modulo 4 equal to 0. That is: DMA_ADDR[1:0] must be equal to 2'b00. The lower two bits of this register are read-only and cannot be set by software to set. As a DMA transfer proceeds, the memory address is transferred as byte increments.

D31	D30	…	D1	D0
DMA_ADDR[31:0]				
rw	rw	…	rw	rw

Bit	Name	W/R	Description
D31-D0	DMA_ADDR	W/R	DMA memory address, please note: the initial memory address written to this register must be Has a value such that its value modulo 4 is equal to 0. That is: DMA_ADDR[1:0] must Must be equal to 2'b00. The lower two bits of this register are read-only and cannot be set by software set

DMA_COUNT

Offset address: 0x20C Reset value: 0x00000000

This register is used to identify the current DMA count of the transfer, and the software setting identifies the initial count of transfers for the entire transfer length. with count As progress, the count is decremented in bytes transferred.

D31	D30	…	D1	D0
DMA_ADDR[31:0]				
rw	rw	…	rw	rw

Bit	Name	W/R	Description
D31-D0	DMA_ADDR	W/R	DMA channel corresponding to DMA memory address. Note: If DMA is counting When the number is 0, the bus will not be requested and the DMA interrupt will be generated

18.4.10 Dynamic FIFO registers

The dynamic FIFO register is only available when the Air105 is configured not to use a dynamic FIFO size. There is a set of registers that each endpoint does not include 0 endpoint. Access to the index register address 0EH of these indices must set the corresponding end point.

TXFIFOSZ

Offset address: 0x62 Reset value: 0x00

TXFIFOSZ is a five-bit register that controls the size of the selected TX endpoint FIFO.

D4	D3	D2	D1	D0
DPB	SZ3	SZ2	SZ1	SZ0
rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D4	DPB		W/R defines whether double-buffered packets are supported: when '1', double-packet buffering is supported, when '0', Only supports single packet buffering
D3-D0	SZ[3:0]		W/R Maximum packet size allowed (any pre-split/high-band within the bulk FIFO wide packets form before transmission)

SZ[3:0]	packet size
0 0 0 0	8
0 0 0 1	16
0 0 1 0	32
0 0 1 1	64
0 1 0 0	128
1 0 1	256
0 0	512
0 1 1 1 1 1	1024
1 0 0	2048
1 0 1	4096

0 0 If DPB=0, the FIFO size is equal to this value, if DPB=1, the FIFO will be twice the value

RXFIFOSZ

Offset address: 0x63 Reset value: 0x00

RXFIFOSZ is a five-bit register that controls the size of the selected receive endpoint FIFO.

D4	D3	D2	D1	D0
DPB	SZ3	SZ2	SZ1	SZ0
rw	rw	rw	Rw	rw

Bit	Name	W/R	Description
D4	DPB		W/R defines whether double-buffered packets are supported: when '1', double-packet buffering is supported, when '0', Only supports single packet buffering
D3-D0	SZ[3:0]		W/R Maximum allowed packet size (below receive bulk/high bandwidth packets table after any combination in FIFO)

SZ[3:0]	packet size
0 0 0 0	8
0 0 0 1	16
0 0 1 0	32
0 0 1 1	64
0 1 0 0	128
0	256
0 1 1 1 1 0	512
0	1024
1 1 1 1	2048
1 0 0 0 1	4096

0 0 If DPB=0, the FIFO size is equal to this value, if DPB=1, the FIFO will be twice the value

TXFIFOADD

Offset address: 0x64 Reset value: 0x0000

TXFIFOADD is a 14-bit register that controls the start address of the selected Tx endpoint FIFO

D13	D12	…	D0

-	AD12	…	AD0
rw	rw	rw	rw

Bit	Name	W/R	Description
D13	-	W/R reserved for future use	
D12-D0 AD[12:0]		W/R starts the endpoint's FIFO address in 8-byte units as follows:	

AD[12:0]	initial address
0 0 0 0 0 0 0	
0 0 0 0 0 0 0 1	0008
0 0 0 0 0 0 0 1 0	0010
…	…
1 F F F	FFF8

RXFIFOADD

Offset address: 0x66 Reset value: 0x0000

RXFIFOADD is a 14-bit register that controls the start address of the selected RX endpoint FIFO

D13	D12	…	D0
-	AD12	…	AD0
rw	rw	rw	rw

Bit	Name	W/R	Description
D13	-	W/R reserved for future use	
D12-D0 AD[12:0]		W/R starts the endpoint's FIFO address in 8-byte units as follows:	

AD[12:0]	initial address
0 0 0 0 0 0 0	
0 0 0 0 0 0 0 1	
0 0 0 0 0 0 1 0	
…	…
1 F F F	FFF8

18.4.11LPM registers

LPM_ATTR

Offset address: 0x360 Reset value: 0x00

This register is used to define the properties of LPM transactions and sleep cycles. In host mode and peripheral mode, the meaning of this register is the same , but the data sources for host and peripheral are different as follows:

1) In peripheral mode, the value in this register will contain the last LPM transaction received and the equivalent properties are accepted. If the response is only After the ACK of the LPM transaction, the contents of this register and the LPM report are updated.

This register can be updated by software, otherwise, the register will hold the current value.

2) In master mode, software will establish the value in this register to define the LPM transaction that will be sent. These values will be inserted into The payload of the next LPM transaction.

D15	D14	D13	D12	D11	D10	D9	D8
EndPoint				Reserved			RmtWak
r	r	r	r	r	r	r	r

D7	D6	D5	D4	D3	D2	D1	D0
HIRD				LinkState			
r	r	r	r	r	r	r	r

Bit	Name	W/R	Description

D15-D12	EndPnt	R This is the endpoint of the LPM transaction token packet
D11-D9	Reserver	R reserved, always returns 0 when read
D8	RmtWak	R This bit is the remote wake-up enable bit RmtWak = 1'b0 : remote wakeup is not used; RmtWak = 1'b1: Remote wake-up function. This bit is set temporarily and applies only to the current suspend state. of the pause period After enumeration the negotiated remote wakeup function will apply
D7-D4	HIRD	R This is the host-initiated recovery duration. This value is the minimum time for host recovery, the The value in the register corresponds to the actual recovery time. Recovery time = 50 microseconds + HIRD * 75 microseconds. Results are between 50-1200 microseconds
D3-D0	LinkState	R This value is used by the host or peripheral to indicate that the LPM transaction charges a status of over-validation. link state=4'h0-reserved link state = 4'h1 - sleep state (L1) link state=4'h2-reserved link-state=4'h3-reserved

LPM_CNTRL

Offset address: 0x362 Reset value: 0x00

In peripheral mode:

D7	D6	D5	D4	D3	D2	D1	D0
			LPMNAK	LPMEN		LPMRES	LPMXMT
r	r	r	rw	rw		rw	rw

Bit	Name	W/R	Description
D7-D5	Reserved	R	reserved, always returns 0x00 when read
D4	LPMNAK	W/R	This bit is used to store the state of all endpoints to deal with all other transactions. Then the LPM transaction will become a NAK. This bit will suspend LPM at Air105 In this case, Air105 will continue to NAK until this bit is in software is cleared
D3-D2	LPMEN	W/R	This register is used to enable LPM in Air105. There are three levels of LPM that can Enabled, this will determine the Air105's response to LPM transactions. These three levels are Yes: ÿ 2'b00, 2'b10 - LPM extended transactions are not supported. in this case, Air105 will not respond to LPM transactions and transactions will time out ÿ 2'b01 - not supported but will extend LPM transactions. in this case, Air105 will respond to LPM transactions as STALL ÿ 2'b11-Air105 supports LPM extended transaction. In this case, Air105 will respond to a NYET or to be determined by the value of LPMXMT and other conditions ACK
D1	LPMRES	W/R	This bit is used by software to initiate recovery (remote wake-up). The difference in this bit is that, The classic RESUME bit of the power register (address 0x01.2), the RESUME Signal timing is controlled by hardware. By writing this bit in software, the resume signal will be set to 50 microseconds, this bit is automatically cleared
D0	LPMXMT	W/R	This bit is set by software to instruct Air105 to switch when the next LPM transaction is received to the state of L1. This bit is only valid if LPMEN is set to 2'b11. This bit can Set with LPMEN at the same cycle. If this bit is set to 1'b1 and LPMEN=2'b11, in Air105, you can deal with it in the following ways: ÿ If no data is waiting (all TX FIFOs are empty), the Air105 will respond with an ACK. In this case, this bit will automatically clear

			and software will generate an interrupt ÿ If data is waiting (at least one data resides in the TX FIFO), The Air105 will respond with a NYET. In this case, the bit will not automatically automatically cleared, but software will generate an interrupt

In host mode:

D7	D6	D5	D4	D3	D2	D1	D0
Reserved						LPMRES	LPMXMT
r	r	r	r	r	r	rw	rw

Bit	Name	W/R	Description
D7-D2	Reserved	R	reserved, always returns 0 when read
D1	LPMRES	W/R	This bit is used by software to initiate recovery from the L1 state. This bit differs from the power Classic RESUME bit of register (Address 0x01.2) and resume signal timing Controlled by hardware. Write this bit in software, the resume signal will be set to HIRD The time specified by the LPM_ATTR register in the field. This bit is automatically cleared remove
D0	LPMXMT	W/R	software sets this bit to send LPM transaction, this bit will be cleared automatically and will be Cleared immediately upon any instruction card or three timeouts

LPM_INTREN

Offset address: 0x363 Reset value: 0x00

LPM_INTREN is a 6-bit register that provides enable bits for interrupts in the LPM_INTR register. If this is registered
If a bit in the register sets bit 1, MC_NINT will acknowledge when the corresponding interrupt in the LPM_INTR register is set. If you send it here
A bit in the register is set to 0, the corresponding interrupt in the LPM_INTR register is still set but MC_NINT is not acknowledged (low power
flat). On reset, all bits of this register are reset to 0

D7	D6	D5	D4	D3	D2	D1	D0
RESERVED		LPMERREN	LPMRESEN	LPMACKEN	LPMNYEN	LPMSTEN	LPMTOEN
r	r	r	r	rw	rw	rw	rw

Bit	Name	W/R	Description
D7-D6	Reserved	R	reserved, always returns 0 when read
D5	LPMERREN	R	1'b0: disable LPMERR interrupt 1'b1: Enable LPMERR interrupt
D4	LPMRESEN	R	1'b0: disable LPMERR interrupt 1'b1: Enable LPMERR interrupt
D3	LPMACKEN	W/R	1'b0: disable LPMERR interrupt 1'b1: Enable LPMERR interrupt
D2	LPMNYEN	W/R	1'b0: disable LPMERR interrupt 1'b1: Enable LPMERR interrupt
D1	LPMSTEN	W/R	1'b0: disable LPMERR interrupt 1'b1: Enable LPMERR interrupt
D0	LPMTOEN	W/R	1'b0: disable LPMERR interrupt 1'b1: Enable LPMERR interrupt

LPM_INTR

Offset address: 0x364 Reset value: 0x00

LPM_INTR is a 7-bit register that provides the LPM power status. When a bit is set to 1, output MC_NINT is occupied (low)
The corresponding enable bit is also set to 1. If the corresponding enable bit is set to 0, the output MC_NINT is not asserted. When reset, send
All bits in the register are reset to 0. Cleared when this register is read.

In peripheral mode:

D7	D6	D5	D4	D3	D2	D1	D0
RESERVED		LPMERR	LPMRES	LPMNC	LPMACK	LPMNY	LPMST
r	r	r	r	r	r	r	R

Bit	Name	W/R	Description
D7-D6	Reserved	R	reserved, always returns 0X0 when read
D5	LKPMERR	R	This bit is set if an LPM transaction is received with an unsupported link state set. In this case, the response to the transaction will go to STALL, however LPM_ATTR The registers will be updated so that software can observe the presence of incompatible LPM packets payload.
D4	LPMRES	R	This bit is set if the Air105 is restored for any reason. This bit is registered with the power supply The RESUME bit of the device (Address 0x01) is mutually exclusive
D3	LPMNC	R	when LPM transaction is received and MUSBMDHRC due to data in RX FIFO etc. This bit is set when there is no response, which occurs under the following conditions: The LPMRESP field in the LMSRESP register is set to 2'b11, the The LPMXMT field has bit 1'b1 set and there is data pending in the Air105 TX FIFO.
D2	LPMACK	R	This bit is set when an LPM transaction is received and an ACK is generated by MUSBMDHRC setting, this will only happen under the following conditions: The LPMRESP field in the LMSRESP register is set to 2'b11, the The LPMXMT field has bit 1'b1 set and there is data pending in the Air105 TX FIFO.
D1	LPMNY	R	This bit is used when an LPM transaction is received and when MUSBMDHRC responds with a NYET be set to. This only happens when: The LPMRESP field in the LMRESP register is set to 2'b11 and LPMXMT Field set bit 1'b0
D0	LPMST	R	This bit is used when an LPM transaction is received and when MUSBMDHRC responds to a STALL is set, this will only happen if: The LPMRESP field in the LMRESP register is set to 2'b01

In host mode:

D7	D6	D5	D4	D3	D2	D1	D0
RESERVED			LPMRES	LPMNC	LPMNCK	LPMNY	LPMST
r	r	r	r	r	r	r	r

Bit	Name	W/R	Description
D7-D6	Reserved	R	reserved, always returns 0x0 when read
D5	LPMERR	R	This bit is received in response to an LPM transaction with a bit stuffing error or PID error is set. In this case, it cannot be paused and the state of the device is unknown
D4	LPMRES	R	This bit is set when the Air105 recovers for any reason. This bit is related to the power register (Address 0x01) RESUME bits are mutually exclusive
D3	LPMNC	R	This bit is set when an LPM transaction has been sent or failed to send, a timeout occurs, or there is The transaction will fail with three attempts to respond with an error
D2	LPMACK	R	This bit is set when an LPM transaction is sent and the device responds with an ACK
D1	LPMNY	R	This bit is set when an LPM transaction is sent and the device responds with a NYET
D0	LPMST	R	This bit is set when an LPM transaction is sent and the device responds with a STALL

LPM_FADDR

Preliminary understanding: only in host mode

Offset address: 0x365 Reset value: 0x00

LPM_FADDR is the function address that will be placed in the LPM payload

D7	D6	D5	D4	D3	D2	D1	D0
0	Function Address						
r	rw	rw	rw	rw	rw	rw	rw

Bit	Name	W/R	Description
D7	-	R	not used, always returns 0
D6-D0	LPM FADDR	W/R	The LPM function address

18.5 USB reset

18.5.1 In peripheral mode

When the Air105 acts as a peripheral and detects other reset conditions on USB, the unit will do the following:

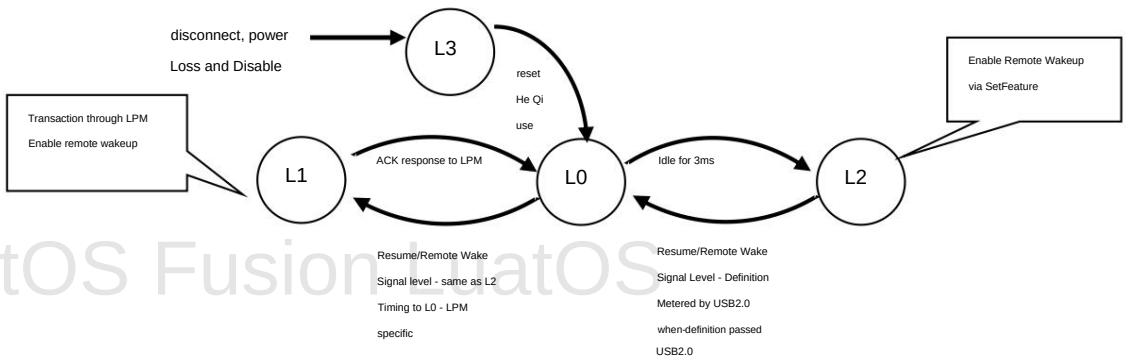
- ÿ Set FADDR to 0
- ÿ Set the index to 0
- ÿ Flush all endpoint FIFOs
- ÿ Clear all control/status registers
- ÿ Enable all endpoint interrupts
- ÿ Generate a reset interrupt

When the application software driver Air105 receives a reset interrupt, it should close all open pipes and wait for the start of bus enumeration.

18.6 Pause/Resume

With the introduction of connection power management, there are two basic methods for Air105 to suspend and resume. Both of these methods are as follows

The basic LPM transaction diagram shows



The process by which Air105 is suspended and resumed depends on whether the core is operating as a device or host and the method required for scraping.

18.6.1 Air105 operating as a peripheral

1) Enter pause mode. When operating as a peripheral, the Air105 monitors USB activity and goes into suspend when no activity occurs for 3ms model. If the suspend interrupt is enabled, an interrupt will be generated at this time and the SUSPENDM output will also go low (if enabled). At this point, the POWERDWN signal is also asserted to indicate that the application can save power by stopping CLK. POWERDWN remains active until power is removed from the bus (indicating that the device has been disconnected) or the signal is restored or reset Signal detected on the bus.

2) When the resume signal is generated on the bus, the first CLK must be restarted if necessary. Then Air105 will automatically exit the temporary

stop mode. If the interrupt enable is restored, an interrupt will be

generated 3) Remote wake-up initiated. If software wants to initiate a remote wakeup while the Air105 is in suspend mode, it should write the power register to set the resume bit (D2) to "1". (Note: If the CLK has been stopped, a restart is required, and a write may be required before that). This bit should be set aside for about 10ms before a software reset is reset to 0 (minimum 2ms, maximum 15ms). At this time the hub should take over the run signal on the USB Note: When the software initiated remote wakeup does not resume, an interrupt will be generated.

18.7 Connect/Disconnect

The specific behavior associated with connecting and disconnecting the Air105 is available both in host mode and in peer-to-peer communication peripheral mode.

18.7.1 In peripheral mode

When the Air105 operates in peripheral mode, the device is connected to the host without interruption. However, a disconnect interrupt (IntrlUSB.D5) is generated when the host aborts the session.

18.8 Planning scheme

This is seen in the following sections, the device controls the operations that the Air105 core needs to perform and the various parties that operate the core under the influence of noodle.

Throughout the discussion, the control unit is assumed to be a microcontroller running some firmware, but it can be customized with hard-wired logic blocks.

18.8.1 Soft connect/disconnect

If desired, the Air105 will allow it to be connected to a software-controlled USB bus. When soft connect/disconnect is

selected, then when Air105 works in peripheral mode, the UTMI+- together with Air105 standard PHY can be used between normal mode and non-driving mode by setting/clearing the 6th bit of the power register (toggle is determined to be a soft link bit). When the soft connect bit is set to 1, the PHY is placed in its normal mode and the D+/D- lines of the USB bus are enabled. At the same time, the Air105 is placed in the "Powered" state and will no longer respond to any USB signals, except for a USB reset. If this feature is enabled, the soft connect bit is 0, the physical layer is put in non-driving mode, D+ and D- are 3-state, and if it has been disconnected, the Air105 will present

other devices on the USB bus. After a hardware reset (NRST=0), the soft link is cleared to 0. Air105 will therefore appear short line until the software sets the soft connection bit to 1. The application software can then choose when to set the PHY to normal mode. A lengthy initialization process may be used to ensure that the initial

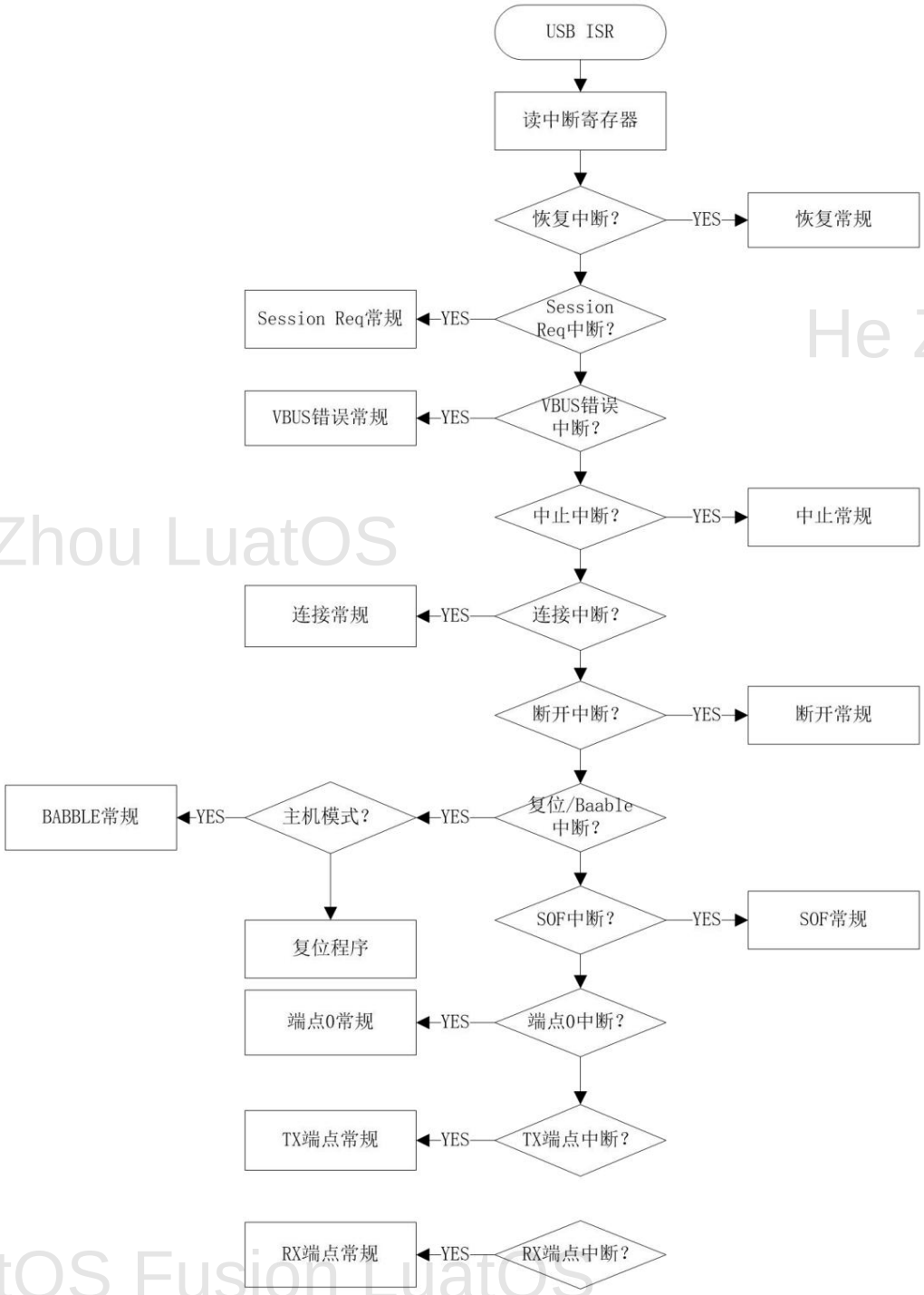
initialization of the system is complete and the system is ready to perform enumeration before connecting to the USB.

Once the soft connect bit has been set to 1, the software can also simulate a disconnect by clearing this bit.

18.8.2 USB Interrupt Handling

When the CPU is disconnected during a USB interrupt, the interrupt status register needs to be read to determine which endpoint caused the interrupt, and Jump to the corresponding program. If the interrupt is caused by multiple endpoints, endpoint 0 is served first, followed by other endpoints. The

following flow chart is given for the flow of the USB terminal service program:



18.9 VBUS Activity

The USB specification defines a series of thresholds related to the need for peer-to-peer communication devices:

ÿ VBUS is valid (required in 4.4 and 4.75)

ÿ Session valid for 'A' device (required between 0.8V and 2.1V)

ÿ Session ends (required between 0.2V and 0.8V)

(The actual threshold used in a particular device needs to be passed through a series of comparators, the external Air105 core. The external Air105 core takes Corresponding VBUSVALID, AVALID and SESSEND input high or low according to VBUS level setting)

Where these thresholds are critical, where the CPU controls the way the Air105 needs to respond depends on whether the device is an 'A' device or a 'B' device and the sender the occurrence of other events. The required actions are summarized as follows:

18.9.1 Operation as 'B' device

1) VBUS> session valid value (ie Vbus[1:0](DevCTL[D4:D3])=10B, session bit (DevCtl.D0) set). This indicates
Activity from "A" device. Air105 will set the session bit and take the DPPULLDOWN output low to disconnect the D+ line pull-down
resistance.

2) VBUS < session valid value, while session bit remains set (ie Vbus[1:0](DevCTL[D4:D3])=01B, session bit (DevCtl.D0)
set up). This indicates that the "A" device has lost power (or disconnected). Air105 will clear the session bit (DevCtl.D0) and generate
Generates a disconnect interrupt (IntrUSB.D5). The CPU ends the session.

3) VBUS< session valid value (ie Vbus[1:0](DevCTL[D4:D3])=00B). This is the next "B" device that can initiate a session please
requested conditions. If the session bit (DevCtl.D0) is set, the Air105 will first pulse 2 ms after SE0 on the bus
data line, then pulse VBUS (take CHRGVBUS high) to start the adjustment program.

18.10 Dynamic FIFOs

If desired, the Air105 can be configured with FIFOs of 128,256,512...1K 64K bytes, partition size, when the Air105
It can continue to be assigned to different endpoints during initialization.

Note: We strongly recommend that you only use this feature, where Air105 is used in devices that require different FIFO sizes in different environments. like
If the FIFO size does not need to be changed, it is best to set these sizes to the dynamic FIFO size options via the standard configuration options
Increases the size of the core, and requires more complex firmware to handle it.

The FIFO specifies the space allocated according to the different requirements of each TX and RX endpoint:

ÿ The starting address of the RAM block of the FIFO

ÿ The maximum size of the packet to be supported

ÿ Is double buffering necessary?

(The last two co-limited spaces need to be allocated to the FIFO)

These details can be specified by the following four registers, which are added to the Air105 register when the dynamic FIFO size option is specified
The index area of the map:

ADDR	Name	Description
62	TxFIFOsz	Tx endpoint FIFO size
63	RxFIFOsz	Rx endpoint FIFO size
64,65	TxFIFOadd	Tx endpoint FIFO address
66,67	RxFIFOadd	Rx endpoint FIFO address

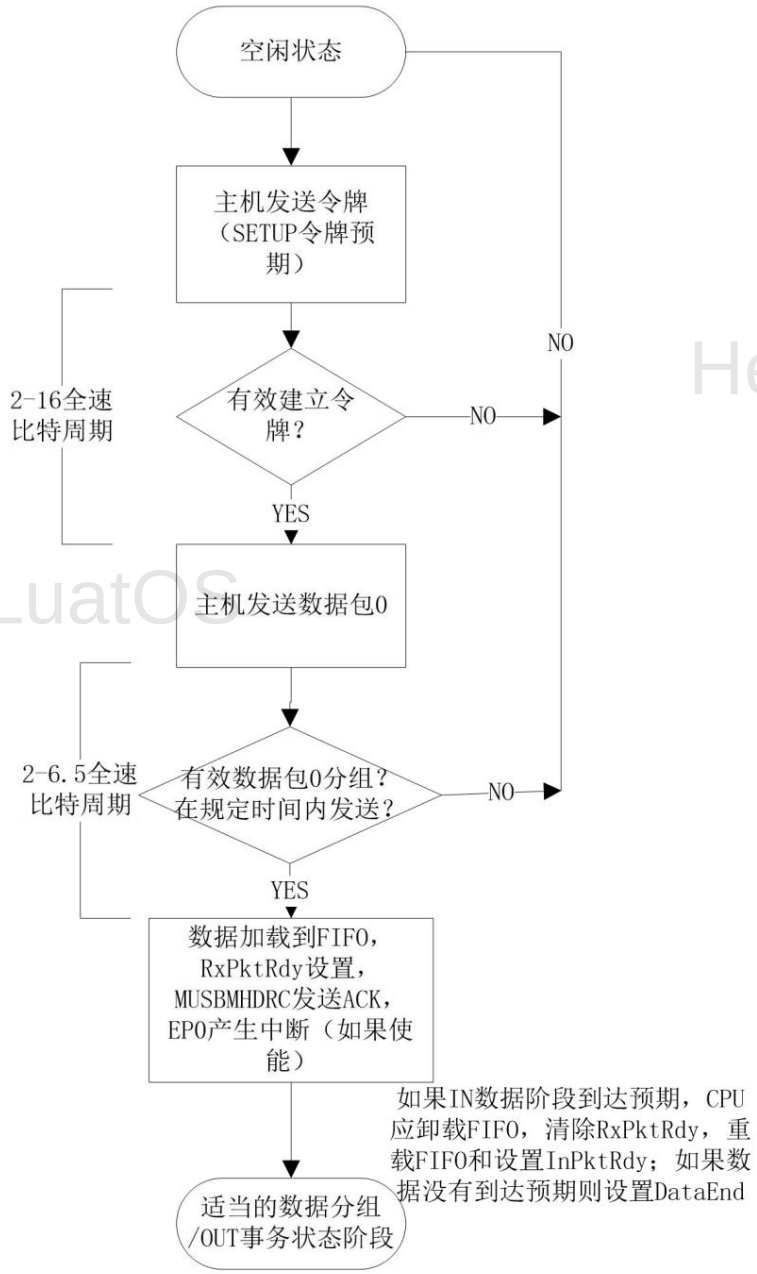
Note: (i) The option size for setting dynamic FIFO is only applicable to endpoints 1-15. Endpoint 0 of the FIFO has a fixed size (64 bytes)
and a fixed location (starting address 0)

(ii) It is the responsibility of the firmware (and the system designer). to ensure that all transmit and receive terminals are active. current usb
The configured RAM blocks are allocated to them at least as large as the endpoint sets the maximum packet size.

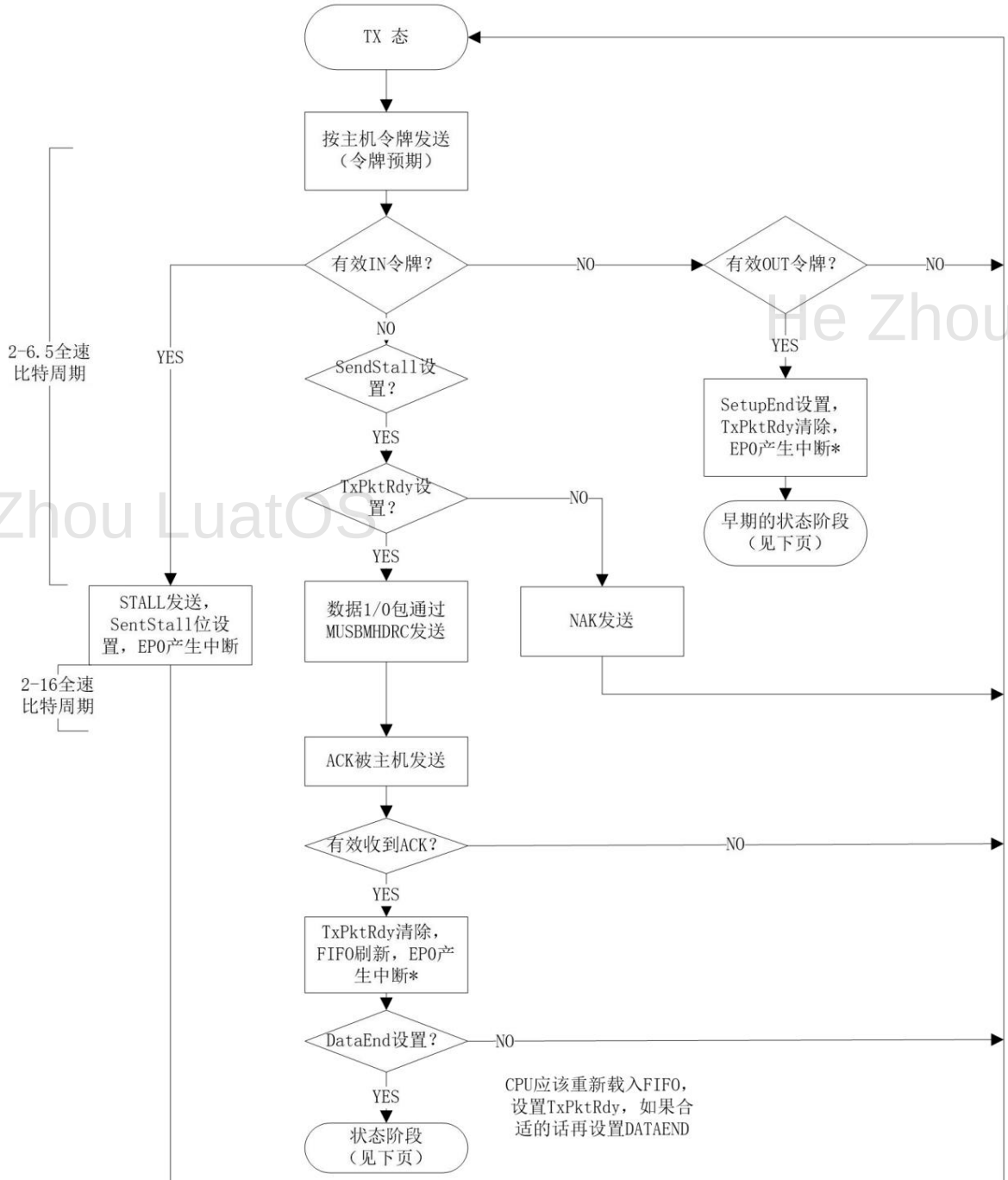
18.11 Transaction Flow as a Peripheral

18.11.1 Control Transactions

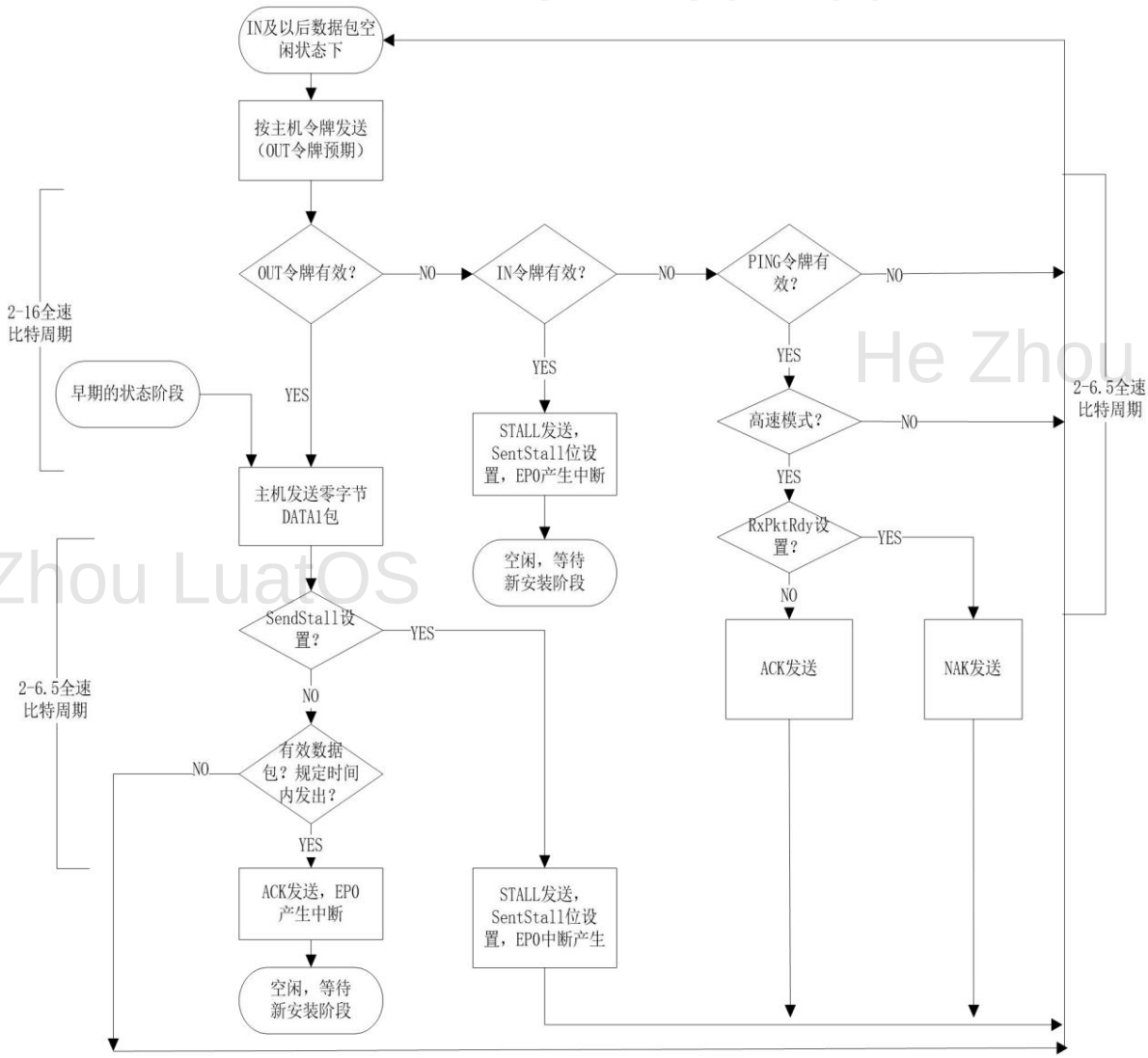
installation phase



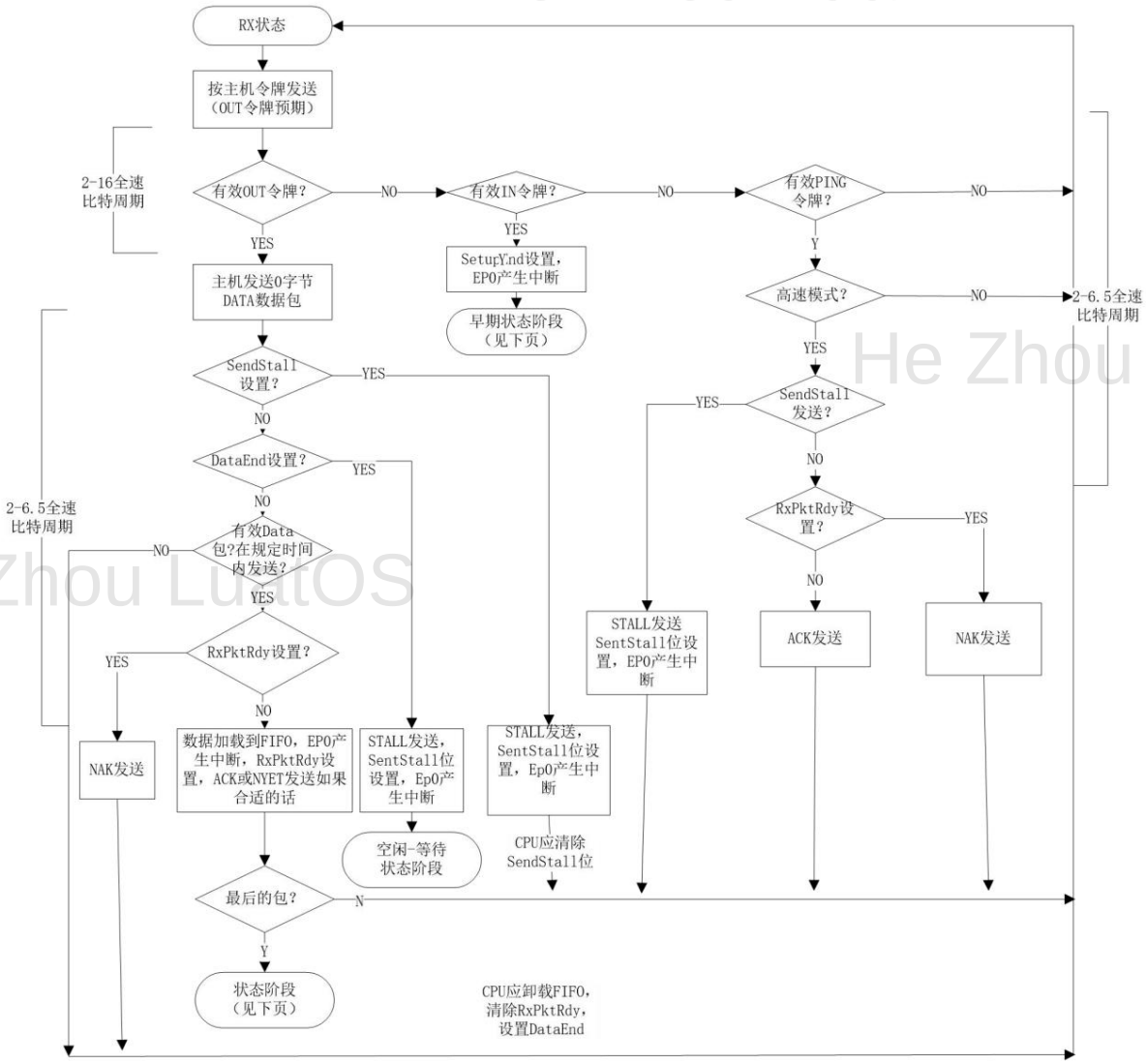
data



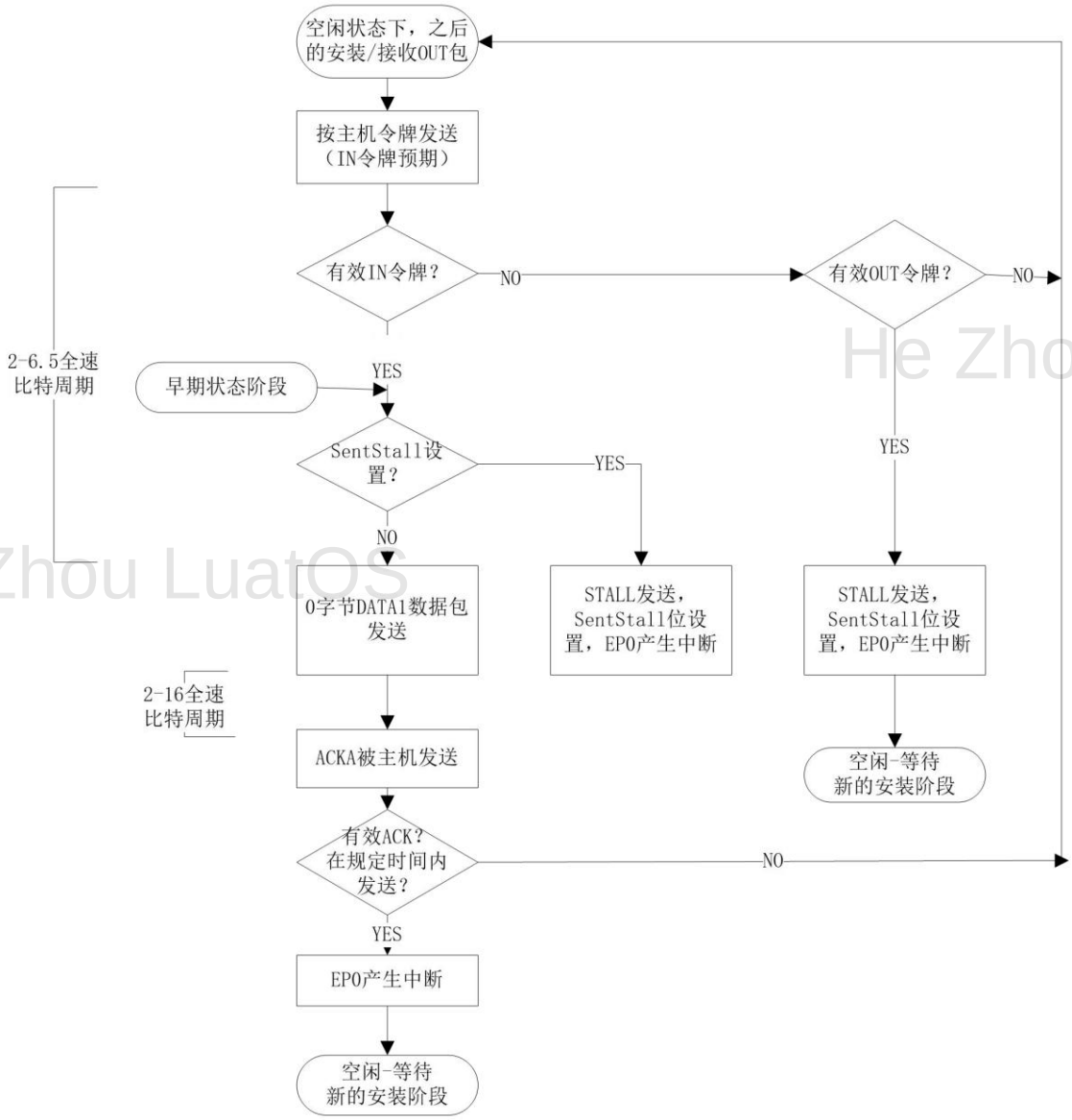
later state stage



OUT data status

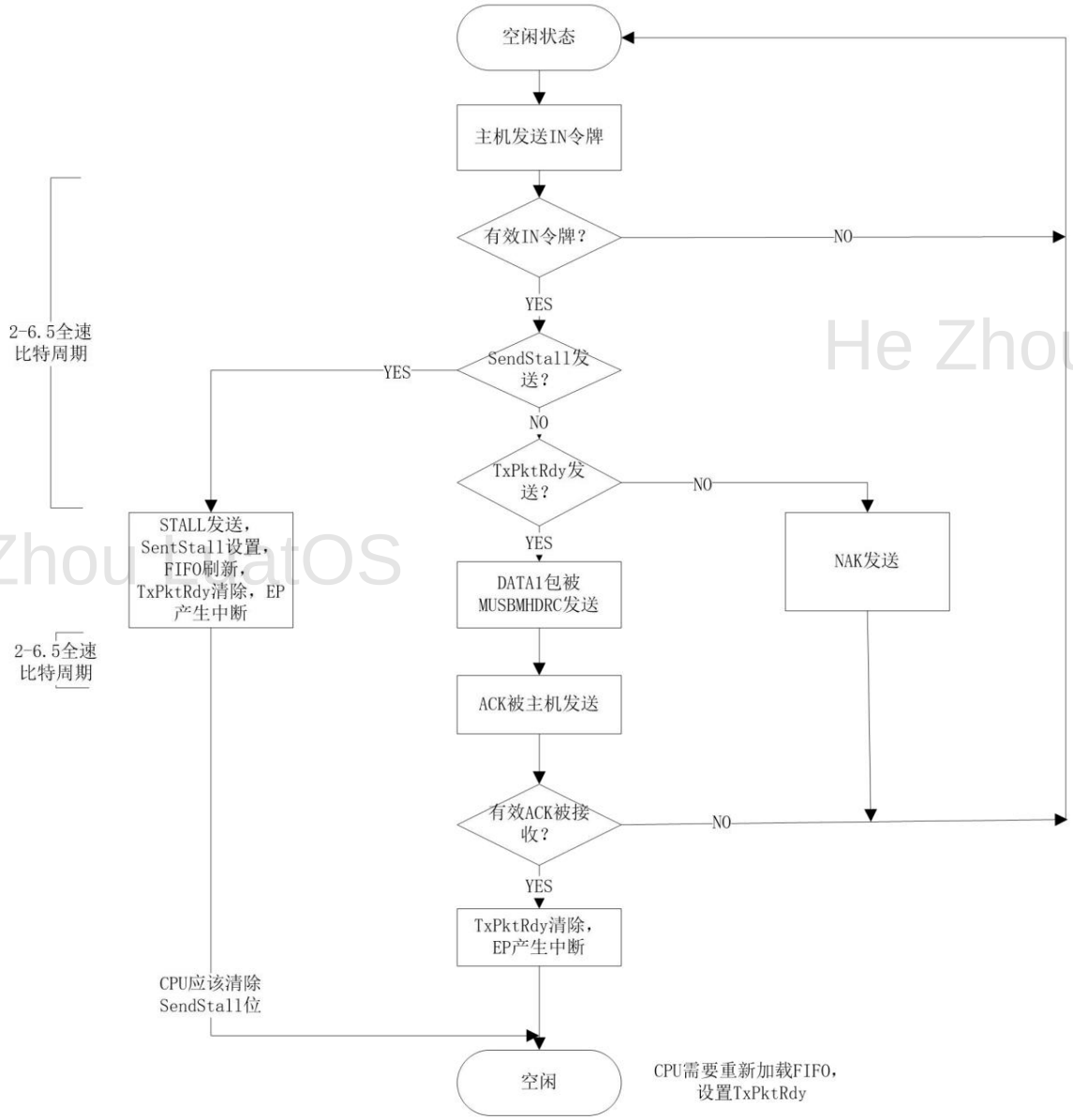


later state stage

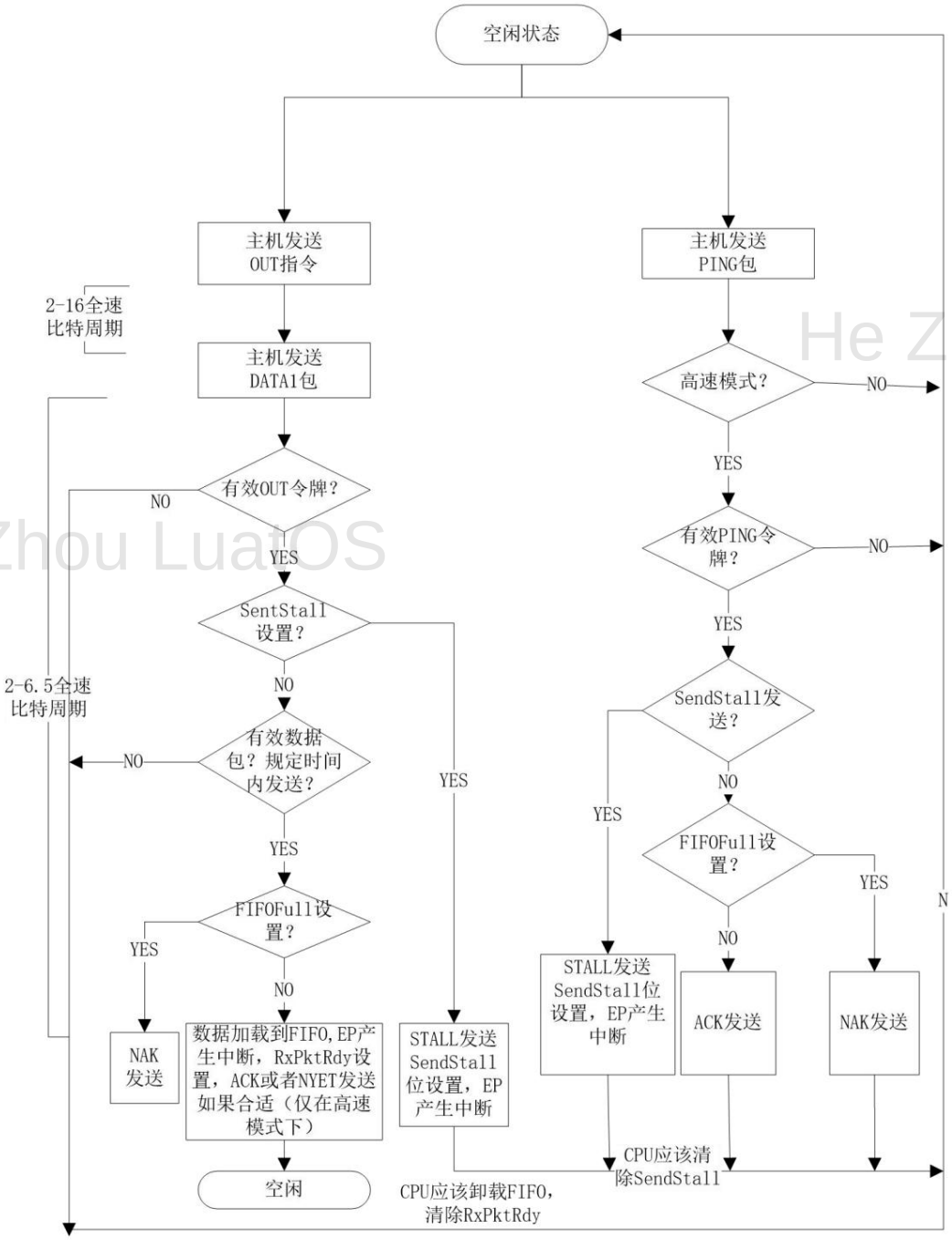


18.11.2 BULK/Low Bandwidth Interrupt Transactions

IN transaction

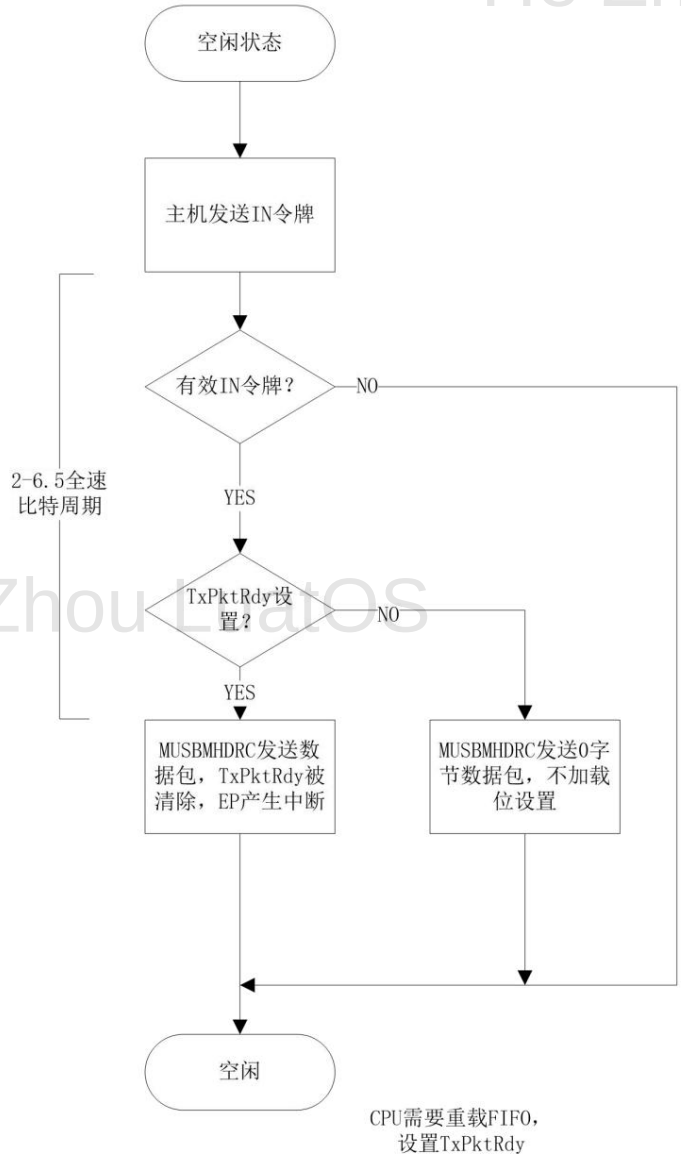


OUT transaction

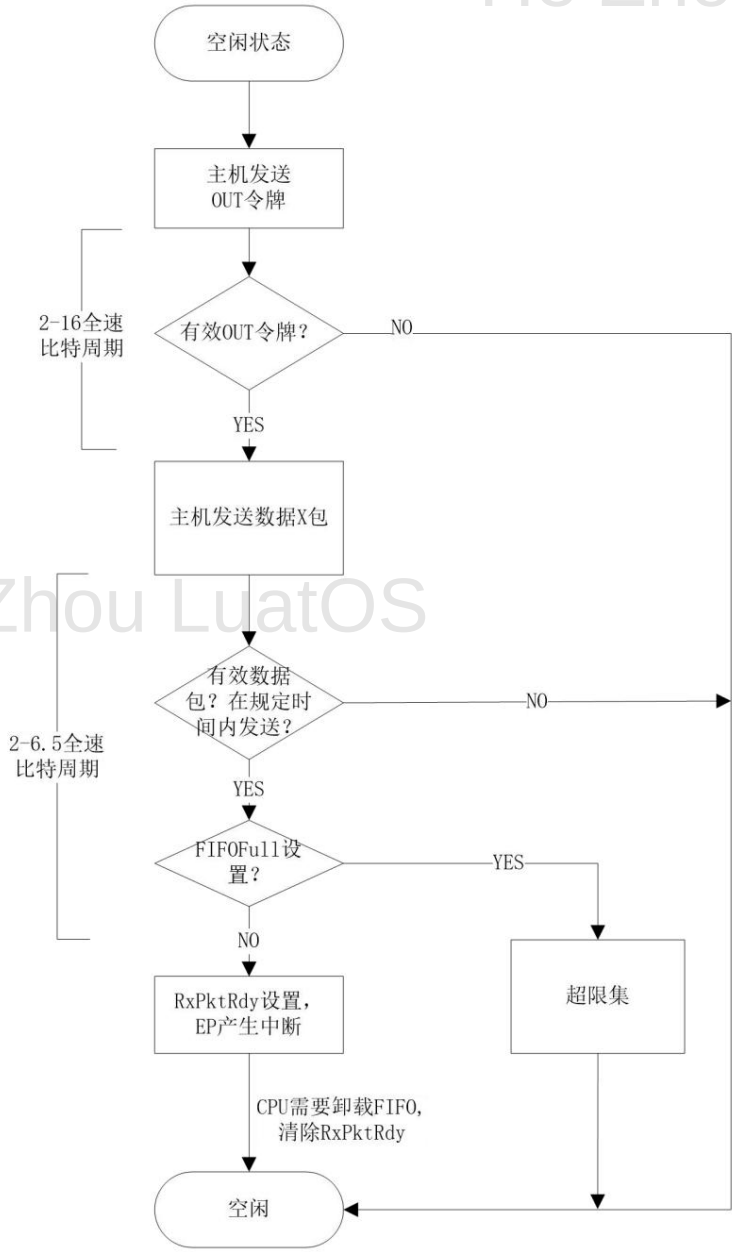


18.11.3 Full-Speed/Low-Bandwidth Isochronous Transactions

IN transaction

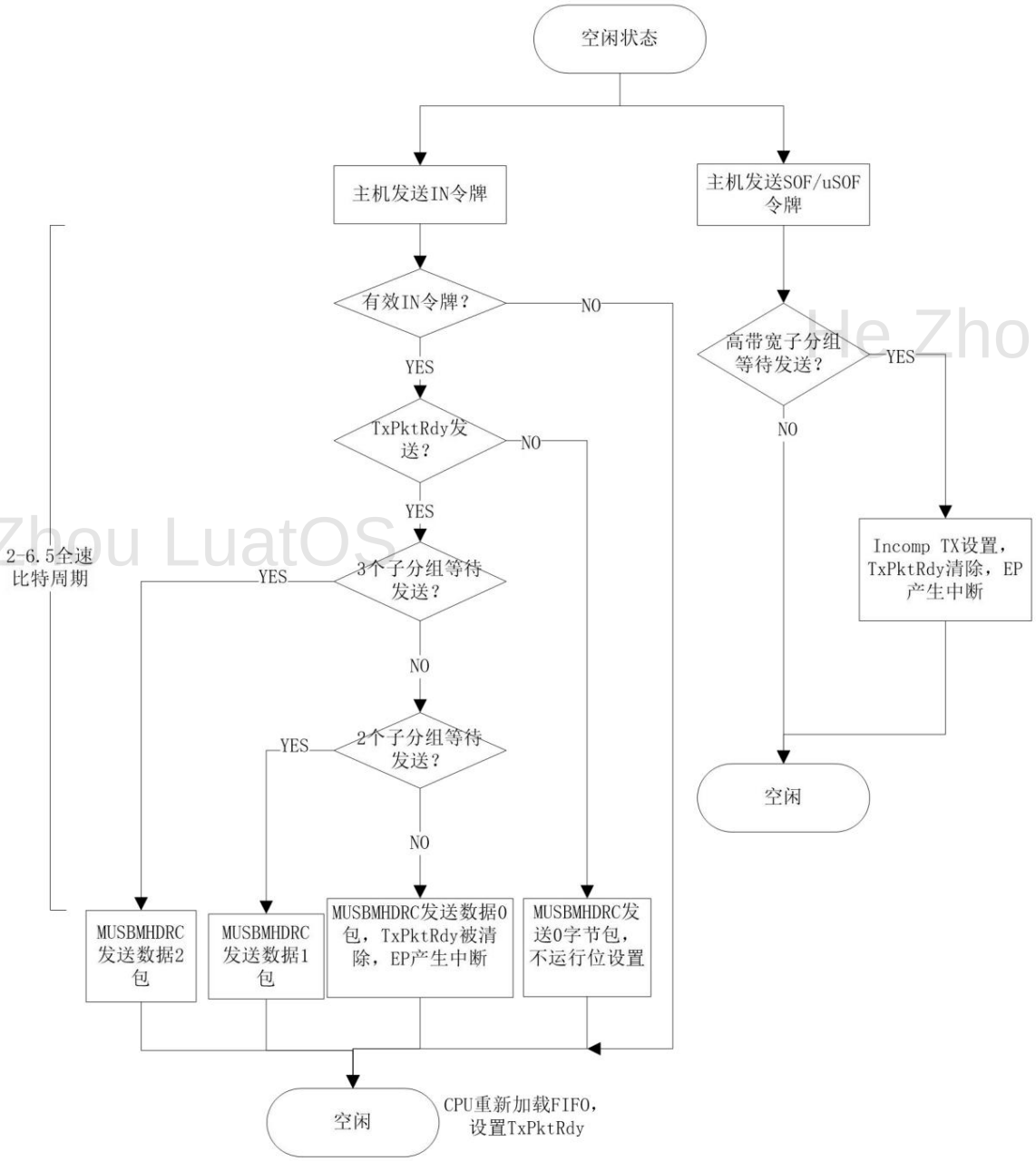


OUT transaction

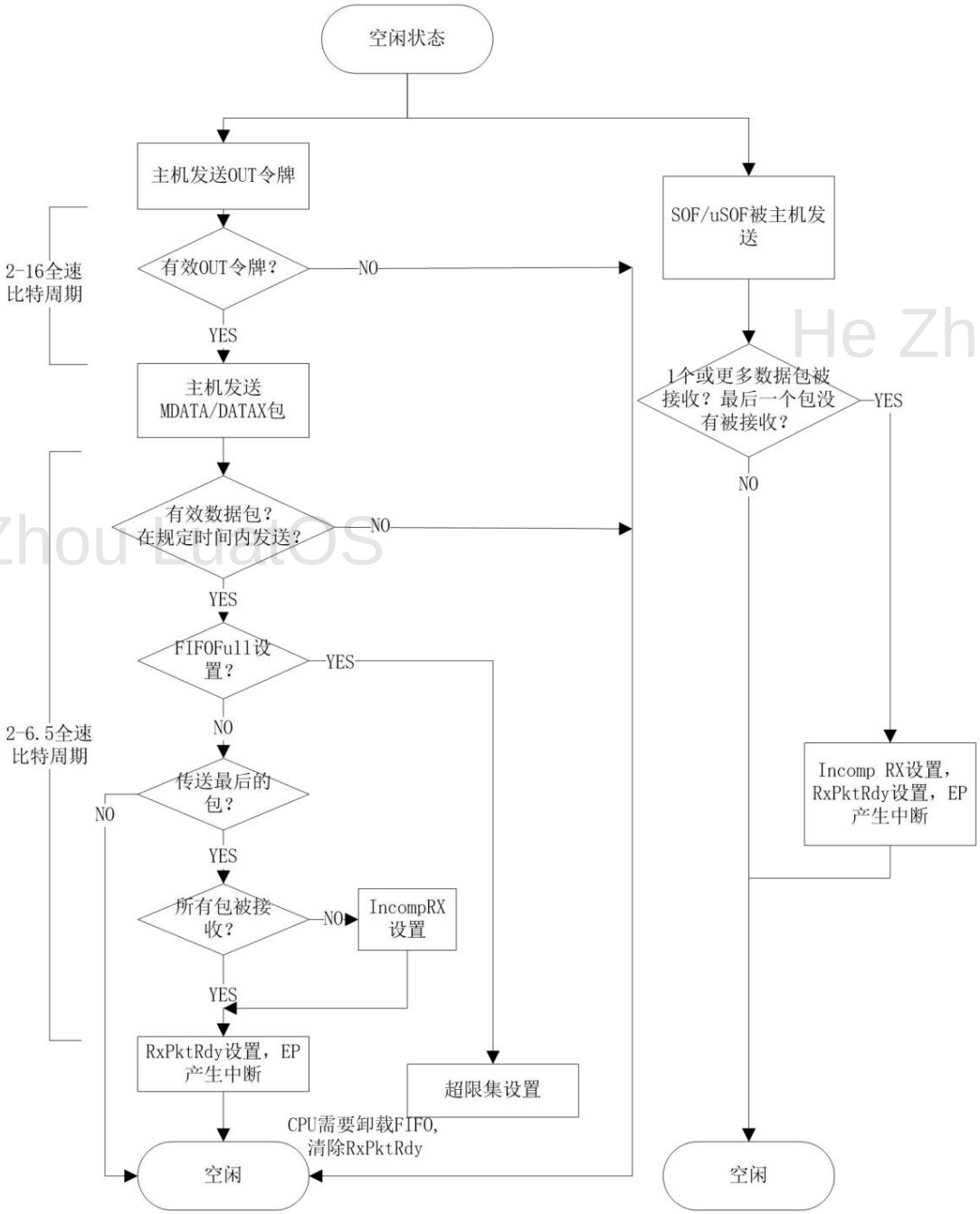


18.11.4 High-bandwidth transactions (sync/interrupt)

IN transaction



OUT transaction



19 Analog/Digital Conversion (ADC)

19.1 Introduction to ADCs

ADC sampling rate is 800KHz, sampling accuracy is 12 bits

19.2 ADC Characteristics

ÿ ADC has a total of 7 channels, the highest sampling rate is 857KHz, and the sampling accuracy is 12 bits

ÿ The reference voltage of ADC is 1.8V

ÿ ADC channel 0 internally collects the CHARGE_VBAT voltage fixedly, the acquisition voltage range is 0~5V, and the internal voltage divider is 900K:500K

ÿ The acquisition voltage range of ADC channel 1~channel 6 can be configured. When the internal voltage divider is enabled, the acquisition voltage range is 0~3.6V.

The acquisition voltage range is 0~1.8V during partial voltage

ÿ ADC channel 1~channel 5 internal integrated voltage divider voltage 558K:558K, channel 6 internal voltage divider 953K:558K

ÿ Unified switch of the internal voltage divider resistors of the 6 channels led by the ADC

19.3 ADC registers

19.3.1 Address Mapping Table

ADC base address table

Address	base	Peripherals	bus
range 0x4001_4000-0x4001_40FF	address 0x4001_4000	ADC	APB0

Table 19- 1 ADC register table

Offset address register name		Width (bit) Reset value	
0x0000	ADC_CR1	32	0x00000000
0x0004	ADC_SR	32	0x00000400
0x0008	ADC_FIFO	32	0x00000000
0x000C	ADC_DATA	32	0x00000000
0x0010	ADC_FIFO_FL	32	0x00000000
0x0014	ADC_FIFO_THR	32	0x00000000
0x0018	ADC_CR2	32	0x0000F040

19.3.2 ADC Control Register 1 (ADC_CR1)

Offset address: 0x0000 Reset value: 0x00000000

31 30 29 28 27 26 25	twenty four	23	twenty two	21 20 19 18 17 16					
Reserved									
15 14 13 12 11 10 9	8	7	6	5	4	3	2	1	0
reserved	pd_adc	reserved	samp_en	adc_ie	adc_samp_rate		adc_ch_sel		

Bit	Name	W/R	Description
31:9	reserved	RO	reserved bit, read value is 0
8	pd_adc	RW	ADC power done enable 0: start ADC 1: Turn off ADC
7	reserved	RO	reserved

6	samp_en	W/R	ADC sampling enable
5	adc_ie	W/R	ADC interrupt enable: 0: Disable; 1: Enable
4:3	adc_samp_rate	W/R	ADC rate selection: 2'b00: 857K; 2'b01: 428K; 2'b10: 214K; 2'b11: 53K
2:0	adc_ch_sel	W/R	ADC channel selection: 3'b000: Channel 0; 3'b001: Channel 1; 3'b010: Channel 2; 3'b011: Channel 3; 3'b100: Channel 4; 3'b101: Channel 5;

19.3.3 ADC Status Register (ADC_SR)

Offset address: 0x0004 Reset value: 0x00000400

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

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

Bit	Name	W/R	Description
31:2		RO	reserved bit, read value is 0
1	Reserve fifo_ov	RC	fifo overflow interrupt, indicating empty and read or full and write
0	adc_done_flag	RO	ADC operation completion flag: FIFO not enabled, read clear 1: completed; 0: not completed Enable FIFO, only clear to 0 when the number of FIFO data is less than the threshold 1: The number of FIFO data is greater than the threshold 0: The number of FIFO data is less than or equal to the threshold

19.3.4 ADC FIFO Control Register (ADC_FIFO)

Offset address: 0x0008 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:3		RO	Reserved bit, read as 0.
2	reserved	RW	fifo overflow interrupt enable, active high
1	fifo_ov_ie		After WC is set to 1, reset the FIFO, set to 1 by software, and clear to 0 by hardware
0	fifo_rst	fifo_en	RW ADC FIFO enable, active high

19.3.5 ADC Data Register (ADC_DATA)

Offset address: 0x000C Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:12		RO reserved	
11:0	reserved adc_data	RO ADC data, FIFO entry after FIFO is enabled	

19.3.6 ADC FIFO Level Register (ADC_FIFO_FL)

Offset address: 0x0010 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:4	reserved	RO reserved	
3:0	fl	RO reflects the number of data in the write operation FIFO	

19.3.7 ADC FIFO Threshold Setting Register (ADC_FIFO_THR)

Offset address: 0x0014 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:4	reserved	RO reserved	
3:0	thr	RO	FIFO threshold value 0: There is more than one data to generate an interrupt 1: There are more than 2 data to generate an interrupt 2: There are more than 3 data to generate an interrupt x: There are more than x+1 data to generate an interrupt

19.3.8 ADC Control Register 2 (ADC_CR2)

Offset address: 0x0018 Reset value: 0x0000F040

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

15	14	13	12 11 10 9 8 7 6 5 4 3 2 1 0
pre Keep	buffs er_ ctrl	ran ge_ sel	reserved

Bit	Name	W/R	Description
31:15	reserved	RO reserved bit	
14	buffer_ctrl	RW	Input BUFFER switch control 0: Input goes through Buffer 1: Input directly without going through Buffer

13	range_sel	RW	Acquisition voltage range selection: 0: 0~1.8V (channel 1~channel 6 internal voltage divider is not enabled) 1: 0~3.6V (enable channel 1~channel 6 internal voltage divider)
12:0	reserved	RO reserved bit	

20 Digital/Analog Conversion (DAC)

20.1 Introduction to DAC

The DAC is a 10-bit digital input, voltage output digital/analog converter.

20.2 Main Features of DAC

ÿ 1 channel

ÿ 10 digital input

ÿ Support DMA function

20.3 DAC Register

20.3.1 Address Mapping Table

DAC base address table

Address	base	Peripherals	bus
range 0x4001_4100-0x4001_41FF	address 0x4001_4100	DAC	APB0

Table 20- 1DAC register table

Offset	Register name width (bit) 32 32 32 32 32		reset value
address 0x0000	DAC_CR1		0x00000010
0x0004 0x0008	DAC_DATA		0x00000000
0x000C 0x0010	DAC_TIMER		0x00000000
	DAC_FIFO_FL		0x00000000
	DAC_FIFO_THR		0x00000000

20.3.2 DAC Control Register (DAC_CR1)

Offset address: 0x0000 Reset value: 0x00000010

31 30		29	28 27 26 25 twenty four
fifo_is	fifo_ov	dac_runin g	reserved

twenty three twenty two twenty one 20 19 18 17 16
reserved

15 14 13 12 11 10 9 8
reserved

7	6	4	3		1	0
reserved	5 dac_curr_s el	pd_dac	fifo_rst	2 dac_dma_ en	is_ie	ov_ie

Bit	Name	W/R	Description
31	fifo_is	RO	DAC FIFO threshold interrupt flag, cleared by hardware 1: The number in the DAC FIFO is less than or equal to the threshold value 0: The number in the DAC FIFO is greater than the threshold
30	fifo_ov	RC	fifo overflow interrupt, indicating empty and read or full and write, read clear to clear the flag bit

29	dac_runing	RO	DAC running flag, after the last data is read from FIFO, required After a period of time, the DAC can be sent to complete, so it can pass This bit confirms whether the transmission is complete 1: DAC is running 0: DAC stopped
28:6	reserved	RO reserved bit, read value is 0	
5	dac_curr_sel	RW	DAC working mode selection: 0: VOUTP connected to at least 20K resistor 1: VOUTP is connected to 2K resistor
4	pd_dac	RW	To turn on the ADC or DAC, both PD_ADC and PD_DAC need to be cleared to 0 0: Turn on DAC 1: Turn off the DAC
3	fifo_rst	After WC is set to 1, reset the FIFO, set to 1 by software, and clear to 0 by hardware	
2	dac_dma_en	W/R	DAC DMA enable, set to 1 to enable DMA when the counter matches the expected value Isochronous start DMA to transfer data to DAC
1	is_ie	W/R	DAC FIFO interrupt enable, set to 1 to enable FIFO threshold interrupt (does not affect ring fifo_is flag position 1)
0	ov_ie	W/R	FIFO overflow interrupt enable, set to 1 to enable FIFO overflow interrupt (does not affect fifo_ov flag position 1)

20.3.3 DAC Data Register (DAC_DATA)

Offset address: 0x0004 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:10		RO reserved	
9:0	Reserve dac_data	RO DAC data register, FIFO entry	

20.3.4 DAC Timer (DAC_TIMER)

Offset address: 0x0008 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:0	dac_timer	RW	Expected value of DAC timer, each count (dac_timer+1)*2 PCLKs fetches data from the DAC FIFO to initiate the conversion

20.3.5 DAC FIFO Level Register (DAC_FIFO_FL)

Offset address: 0x000C Reset value: 0x00000000

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

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

reserved	fl

Bit	Name	W/R	Description
31:4	reserved	RO	reserved
3:0	fl	RO	reflects the number of data in the write operation FIFO

20.3.6 DAC FIFO Threshold Setting Register (DAC_FIFO_THR)

Offset address: 0x0014 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:4	reserved	RO	reserved
3:0	thr	RO	FIFO threshold value The number of FIFO data is less than or equal to the threshold requesting DMA or generating The interrupt requires the CPU to write data, and the request ends if it is greater than the threshold

21 QSPI controller (QFlash_controller)

21.1 Introduction to QSPI Controller

The QSPI controller supports standard SPI, Dual SPI and Quad SPI communication.

21.2 QSPI Controller Features

ÿ Supports Single, Double and Quad I/O commands

ÿ Configurable DMA controller for access

ÿ Command parameters/format, SPI polarity/phase, etc. can be configured, with good compatibility

21.3 QSPI Controller Registers

21.3.1 Address Mapping Table

Register Base Address Table

address range	Base	Peripherals	bus
0x400A_2000-0x400A_2FFF	address 0x400A_2000	QFlash_controller	AHB

Table 21- 1QSPI Controller Register Table

Offset	register name	Width (bit) Reset value	
address	FCU_CMD	32	0x00000000
0x00 0x04	ADDRESS	32	0x00000000
0x08 0x0C	BYTE_NUM	32	0x00000000
0x10 0x14	WR_FIFO	32	0x00000000
0x18 0x1C	RD_FIFO	32	0x00000000
0x20 0x24	DEVICE_PARA	32	0x00A80283
0x28 0x2C	REG_WDATA	32	0x00000000
0x30 0x34	REG_RDATA	32	0x00000000
0x38 0x3C	INT_MASK	32	0x00000000
0x40	INT_UMMASK	32	0x00000000
	INT_MASK_STATUS	32	0x00000000
	INT_STATUS	32	0x00000000
	INT_RAWSTATUS	32	0x00000000
	INT_CLEAR	32	0x00000000
	CACHE_INTF_CMD	32	0xABB92BEB
	DMA_CNTL	32	0x00000000
	FIFO_CNTL	32	0x00000000

21.3.2 Command Register (FCU_CMD)

Offset address: 0x0000 Reset value: 0x00000000

31 30	29	28 27 command_code	26	25	twenty four
twenty three	twenty two	twenty one	20 19	18	17 16
reserved					
15 14	13	12	11	10	9 8
reserved					bus_mode
7 6	5	4	3	2	1 0

cmd_format done busy access_ack access_req

Bit	Name	W/R	Description
31:24	command_code	RW	command word
23:10	reserved	RO	reserved bit
9:8	bus_mode	RW	00: SPI mode, cmd-addr-data = 1-1-1 01: Q-out mode, cmd-addr-data = 1-1-4 10: Q-IO mode, cmd-addr-data = 1-4-4 11: QPI mode, cmd-addr-data = 4-4-4
7:4	cmd_format	RW	Command format: 0000: 8bit cmd 0001: 8bit cmd + 8bit read reg data 0010: 8bit cmd + 16bit read reg data 0011: 8bit cmd + 24bit read reg data 0100: 8bit cmd + 24bit dummy + 8bit write reg data 0101: 8bit cmd + 24bit address (release from power down, address is set to 0) + 8bit read reg data 0110: 8bit cmd + 24bit address (release from power down, address is set to 0) + 16bit read reg data 0111: 8bit cmd + 8bit write reg data 1000: 8bit cmd + 16bit write reg data 1001: 8bit cmd + 24bit address 1010: 8bit cmd + 24bit addr + read data… 1011: 8bit cmd + 24bit addr + dummy + read data... 1100: 8bit cmd + 24bit addr + 8bitM + dummy + read data... 1101: 8bit cmd + 24bit addr + programming data...
3	done	RO	current command completion flag
2	busy	RO	current command in progress
1	access_ack	RO	1: The current command requst is licensed 0: The current code bus is accessing, and the s-bus is not allowed to access Flash
0	access_req	RW	requests to access Flash

21.3.3 Address Register (ADDRESS)

Offset address: 0x0004 Reset value: 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
address													m7_0		

Bit	Name	W/R	Description
31:8	address		The Flash address where the RW command is executed
7:0	m7_0	RW	m7~m0: Whether to do continuous reading (M5-M4=(10))

21.3.4 Read Data Quantity Register (BYTE_NUM)

Offset address: 0x0008 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:29	reserved	RO	reserved bit
28:16	wr_byte_num	RW	Number of write data bytes after write command, 1-4096 bytes, write '0' table Show 4096
15:13	reserve	RO	reserved bit
12:0	rd_byte_num	RW	The number of read data bytes after read command, 1-4096 bytes, write '0' table Show 4096

21.3.5 Write Data FIFO Register (WR_FIFO)

Offset address: 0x000C Reset value: 0x00000000

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

Bit	Name	W/R	Description
31:0	wr_data	WO	Write data FIFO, 32 bits, send according to the actual data of wr_byte_num to Flash

21.3.6 Read Data FIFO Register (RD_FIFO)

Offset address: 0x0010 Reset value: 0x00000000

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

Bit	Name	W/R	Description
31:0	rd_data	RO	read data FIFO, 32 bits, if rd_byte_num is not an integer of 4 times, the LSB is filled with 0

21.3.7 DEVICE PARAMETER REGISTER (DEVICE_PARA)

Offset address: 0x0014 Reset value: 0x00A80283

31	30	29	28	27	26	25	twenty four											
one_us_count																		
twenty three	twenty two	twenty one	20	19	one_us_count								18			17		16
15	14	13	12	11									10			9		8
sample_dly	sample_pha		reserved													protocol		
7	6	5	dummy_cycles								4				2	1		0
											3	flash_ready	reserved			freq_sel		

Bit	Name	W/R	Description

31:16	one_us_count	RW counts the value of 1us with the controller clock cycle	
15	sample_dly	RW	Sampling Delay Time: 0: Delay 1 cycle to sample the returned Flash read data 1: Delay 2 cycles to sample the returned Flash read data
14	sample_pha	RW	Sampling clock phase selection: 0: Use the rising edge of the non-inverted clock to sample the Flash read data 1: Use the rising edge of the inverted clock to sample the Flash read data
13:10	reserved	RO reserved bit	
9:8	protocol	RW	00: CLPL (clock inactive low) 11: CHPH (clock inactive high) 01,10: reserved
7:4	dummy_cycles	RW	The number of dummy cycles includes the 8 bits after the 24-bit address M-code
3	flash_ready	RW	0: Flash device initialization is not completed, Cache read operation is suspended 1: Flash device initialization is complete, Cache can read Flash
2	reserved	RO reserved bit	
1:0	freq_sel	RW	00: qspi_clk = fclk 01: qspi_clk = fclk/2 10: qspi_clk = fclk/3 11: qspi_clk = fclk/4

21.3.8 FLASH register write data (REG_WDATA)

Offset address: 0x0018 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:0	reg_wdata	RW write register data	

21.3.9 FLASH register read data (REG_RDATA)

Offset address: 0x001C Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:0	reg_rdata	WO Read register data	

21.3.10 Interrupt MASK register (INT_MASK)

Offset address: 0x0020 Reset value: 0x00000000

31	30	29	28	27	26	25	twenty four
reserved							

twenty three	twenty two	twenty one	20	19	18	17	16

reserved							
15	14	13	12	11	10	9	8
reserved							
7	6						0
Reserve	tx_fifo_data_mask	5 rx_fifo_data_mask	4 tx_fifo_of_mask	3 tx_fifo_uf_mask	2 rx_fifo_of_mask	1 rx_fifo_uf_mask	done_int_mask

Bit	Name	W/R	Description
31:7	Reserve	RO	reserved bit
6	tx_fifo_data_mask	RW	Write 1 to disable the data interrupt of tx FIFO
5	rx_fifo_data_mask	RW	Write 1 to disable rx FIFO data interrupt
4	tx_fifo_of_mask	RW	write 1 to disable tx FIFO overflow interrupt
3	tx_fifo_uf_mask	RW	write 1 to disable tx FIFO underflow interrupt
2	rx_fifo_of_mask	RW	write 1 to disable rx FIFO overflow interrupt
1	rx_fifo_uf_mask	RW	write 1 to disable rx FIFO underflow interrupt
0	done_int_mask	RW	Write 1 to disable command completion interrupt

21.3.11 Interrupt UNMASK register (INT_UNMASK)

Offset address: 0x0024 Reset value: 0x00000000

31 30	29	28	27	26	25	twenty four
reserved						
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
15	14	13	12	11	10	9	8
reserved							
7	6						0
Reserve	tx_fifo_data_unmask	5 rx_fifo_data_unmask	4 tx_fifo_of_unmask	3 tx_fifo_uf_unmask	2 rx_fifo_of_unmask	1 tx_fifo_uf_unmask	done_int_unmask

Bit	Name	W/R	Description
31:7	reserved	RO	reserved bit
6	tx_fifo_data_unmask	RW	Write 1 to enable data interrupt of tx FIFO
5	rx_fifo_data_unmask	RW	Write 1 to enable data interrupt of rx FIFO
4	tx_fifo_of_unmask	RW	Write 1 to disable tx FIFO overflow interrupt
3	tx_fifo_uf_unmask	RW	Write 1 to disable tx FIFO underflow interrupt
2	rx_fifo_of_unmask	RW	Write 1 to disable rx FIFO overflow interrupt
1	rx_fifo_uf_unmask	RW	Write 1 to disable rx FIFO underflow interrupt
0	done_int_unmask	RW	Write 1 to disable command completion interrupt

21.3.12 Interrupt MASK Status Register (INT_MASK_STATUS)

Offset address: 0x0028 Reset value: 0x00000000

31 30	29	28	27	26	25	twenty four
reserved						
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
15	14	13	12	11	10	9	8
reserved							
7	6	5	4	3	2	1	0

Reserve	tx_fifo_data_mask	rx_fifo_data_mask	tx_fifo_of_mask	tx_fifo_uf_mask	rx_fifo_of_mask	rx_fifo_uf_mask	done_int_mask

Bit	Name	W/R	Description
31:7		RO	reserved bit
6	Reserved		Data interrupt enable status of RO tx FIFO
5	tx_fifo_data_mask		Data interrupt enable status of RO rx FIFO
4	rx_fifo_data_mask		RO tx FIFO overflow interrupt enable status
3	tx_fifo_of_mask		RO tx FIFO underflow interrupt enable status
2	tx_fifo_uf_mask		RO rx FIFO overflow interrupt enable status
1	rx_fifo_of_mask		RO rx FIFO underflow interrupt enable status
0	rx_fifo_uf_mask done_int_mask	RO	Command Completion Interrupt Enable Status

21.3.13 INTERRUPT STATUS REGISTER (INT_STATUS)

Offset address: 0x002C Reset value: 0x00000000

31	30	29	28	27	26	25	twenty four
reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
15	14	13	12	11	10	9	8
reserved							
7	6		4				0

Reserve	tx_fifo_data_status	5	rx_fifo_data_status	tx_fifo_of_status	3	tx_fifo_uf_status	2	rx_fifo_of_status	1	rx_fifo_uf_status	done_int_status

Bit	Name	W/R	Description
31: 7		RO	reserved bit
6	Reserved		Data interrupt status of RO tx FIFO
5	tx_fifo_data_status		Data interrupt status of RO rx FIFO
4	rx_fifo_data_status		RO tx FIFO overflow interrupt status
3	tx_fifo_of_status		RO tx FIFO underflow interrupt status
2	tx_fifo_uf_status		RO rx FIFO overflow interrupt status
1	rx_fifo_of_status		RO rx FIFO underflow interrupt status
0	rx_fifo_uf_status done_int_status	RO	command complete interrupt status

21.3.14 INTERRUPT RAW STATUS REGISTER (INT_RAWSTATUS)

Offset address: 0x0030 Reset value: 0x00000000

31	30	29	28	27	26	25	twenty four
reserved							
twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
15	14	13	12	11	10	9	8
reserved							
7	6		4				0

Reserve	tx_fifo_data_rawstatus	5	rx_fifo_data_rawstatus	tx_fifo_of_rawstatus	3	tx_fifo_uf_rawstatus	2	rx_fifo_of_rawstatus	1	rx_fifo_uf_rawstatus	done_int_rawstatus

Bit	Name	W/R	Description

31:7	reserved	RO reserved bit
6	tx_fifo_data_rawstatus	RO tx FIFO data interrupt status
5	rx_fifo_data_rawstatus	RO rx FIFO data interrupt status
4	tx_fifo_of_rawstatus	RO tx FIFO overflow interrupt status
3	tx_fifo_uf_rawstatus	RO tx FIFO underflow interrupt status
2	rx_fifo_of_rawstatus	RO rx FIFO overflow interrupt status
1	rx_fifo_uf_rawstatus	RO rx FIFO underflow interrupt status
0	done_int_rawstatus	RO command complete interrupt status

21.3.15 INTERRUPT CLEAR REGISTER (INT_CLEAR)

Offset address: 0x0034 31 Reset value: 0x00000000

30	29	28	27	26	25	twenty four
reserved						
twenty three	twenty two	twenty one	20	19	18	17	16
reserved						
15	14	13	12	11	10	9	8
reserved						
7	6		4		2	1	0
Reserve clr_tx_fifo_data		5 clr_rx_fifo_data	clr_tx_fifo_of	3 clr_tx_fifo_uf	clr_rx_fifo_of	clr_rx_fifo_uf	clr_done_int

Bit	Name	W/R	Description
31:7		RO	reserved bit
6	Reserved	RO	clear data interrupt of tx FIFO
	clr_tx_fifo_data	RO	clear data interrupt of rx FIFO
5 4	clr_rx_fifo_data	RO	clears tx FIFO overflow interrupt
3	clr_tx_fifo_of	RO	clears tx FIFO underflow interrupt
2	clr_tx_fifo_uf	RO	clears rx FIFO overflow interrupt
1	clr_rx_fifo_of	RO	clears rx FIFO underflow interrupt
0	clr_rx_fifo_uf clr_done_int	RO	clear command complete interrupt

21.3.16 CACHE Interface Command Register (CACHE_INTF_CMD)

Offset address: 0x0038 31 Reset value: 0x00000000

30	29	28 27	26	25	twenty four
cache_intf_reldscmd					
twenty three	twenty two	twenty one	20 19	18	17	16
cache_intf_dscmd					
15	14	13 12	11	10 9		8
reserved		cache_intf_rdcmd_bus_mode		cache_intf_rdcmd_format		
7	6	5	4	3	2	1	0
cache_intf_rdcmd							

Bit	Name	W/R	Description
31:24	cache_intf_reldscmd		The RW cache interface issues the command Flash release from deepsleep. Save 0xAB
23:16	cache_intf_dscmd		RW cache interface issues Flash deepsleep command, default 0xB9
15:14	reserved	RO	reserved bit

13:12	cache_intf_rdcmd_bus_mode	RW	cache reads Flash and Flash Deepsleep (release from DS) bus mode: 00: SPI mode, cmd-addr-data = 1-1-1 01: Q-out mode, cmd-addr-data = 1-1-4 10: Q-IO mode, cmd-addr-data = 1-4-4 11: QPI mode, cmd-addr-data = 4-4-4
11:8	cache_intf_rdcmd_format	RW	The command format of cache to read Flash: 1011: 8bit cmd + 24bit addr + 8bitM + dummy + read data…
7:0	cache_intf_rdcmd	RW	cache reads the command word of Flash, the default value is 0xEB 0xEB: quad-IO

21.3.17 DMA Control Register (DMA_CNTL)

Offset address: 0x003C Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:1	reserved	RO	reserved bit
0	tx_dma_en	RW	Enable TX DMA, after writing '1', if the tx FIFO capacity is less than 4, it will generate dma_req

21.3.18 FIFO Control Register (FIFO_CNTL)

Offset address: 0x0040 Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
tx_fifo_flush	reserved							reserved				tx_fifo_empty	tx_fifo_full	reserved	tx_fifo_level

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

Bit	Name	W/R	Description
31	tx_fifo_flush	RW	Write '1' to clear the TX FIFO
30:22	reserved	RO	reserved bit
21	tx_fifo_empty	RO	'1': TX FIFO is empty
20	tx_fifo_full	RO	'1': TX FIFO is full
19:16	tx_fifo_level	RO	TX FIFO data volume
15	rx_fifo_flush	RW	Write '1' to clear the RX FIFO
14:6	reserved	RO	reserved bit
5	rx_fifo_empty	RO	1': RX FIFO is empty
4	rx_fifo_full	RO	1': RX FIFO is full

3:0	rx_fifo_level	RO	RX FIFO data volume

22 DCMI interface (DCMIS)

22.1 Introduction to DCMI Interface

The Digital Camera Interface (DCMI) is a synchronous parallel interface capable of accepting external 8-bit, 10-bit, 12-bit or 14-bit CMOS High-speed data stream from the camera module. Different data formats are supported: video and compressed data. The interface contains up to 14 data line and a pixel clock line. The polarity of the pixel clock can be programmed so that data can be captured on the rising and falling edges of the pixel clock.

according to.

22.2 DCMI functional characteristics

ÿ Support 8-bit, 10-bit, 12-bit or 14-bit parallel interface

ÿ Support embedded code/external line sync and frame sync

ÿ Support continuous mode or snapshot mode

ÿ Support crop function

ÿ Supports the following data formats 8/10/12/14-bit progressive video: Monochrome or native Bayer format

22.3 DCMI register

22.3.1 Address Mapping Table

Register Base Address Table

Address	base	Peripherals	bus
range 0x4006_0000-0x4006_FFFF	address 0x4006_0000	DCMIS	AHB

Table 22- 1DCMI register table

offset address	register name	Width (bit)	Reset value
0x00	DCMI_CR	32	0x80000000
0x04	DCMI_SR	32	0x00000000
0x08	DCMI_RIS	32	0x00000000
0x0C	DCMI_IER	32	0x00000000
0x10	DCMI_MIS	32	0x00000000
0x14	DCMI_ICR	32	0x00000000
0x20	DCMI_CWSTRT	32	0x00000000
0x24	DCMI_CWSIZE	32	0x00000000
0x28	DCMI_DR	32	0x00000000

22.3.2 DCMI Control Register (DCMI_CR)

Offset address: 0x0000 Reset value: 0x80000000

31	30	29	28	27	26	25	twenty four
dma_tri_level 22					Reserved		
twenty three		21	20	19	18	17	16
reserved		fifo_flush OELS 13		LSM	OEBS	BSM	
15	14		12	11	10	9	8
Reserved ENABLE 6		reserved		EDM		FCRC	
7		5	4	3	2	1	0
VSPOL HSPOL PCLKPOL			reserved		CROP	CM CAPTUR E	

Bit	Name	W/R	Description
31:29	dma_tri_level	RW	DMA request trigger threshold: 0: reserved 1: DMA request is started when there is 1 word in the FIFO 2: DMA request is started when there are 2 words in the FIFO 3: reserved 4 (4 32bits, default value): There are 4 words in the FIFO to start DMA request
28:22	reserved	RO reserved bit	
twenty one	fifo_flush	RW	Write 1 to clear the DCMI receive FIFO. After the FIFO is cleared, this bit is automatically clear Writing 0 has no effect After writing fifo_flush, the software queries the bit status: 1: Indicates that the FIFO clearing operation is in progress and cannot receive images operate 0: Indicates that the FIFO clearing operation has been completed
20	OELS	RW	Parity row selection, used with LSM to select rows 0: Capture the first line from the frame, discard the second line 1: Capture the second line from the frame, discard the first line
19	LSM	RW	row selection mode 0: capture all lines 1: Capture 1 line every 2 lines
18	OEBS	RW	Parity byte selection, used with BSM to select bytes 0: Capture the first byte from frame/line, discard the second word Festival 1: Capture the second byte from the start of frame/line, discard the first word Festival
17:16	BSM	RW	BSM[1:0]: 00: Capture every received data 01: Capture 1 byte every 1 byte 10: Capture 1 byte every 4 bytes 11: Capture 2 bytes every 4 bytes This mode only works for EDM[1:0]=00
15	reserved	RO reserved bit	
14	ENABLE	RW	DCMI enable bit 0: DCMI is not enabled 1: DCMI enable
13:12	reserved	RO reserved bit	
11:10	EDM	RW	Extended data schema selection 2'b00: 8-bit data is obtained for each pixel clock 2'b01: Get 10 bits of data for each pixel clock 2'b10: Get 12-bit data for each pixel clock 2'b11: Each pixel clock gets 14 bits of data
9:8	FCRC	RW	Image frame capture rate control, valid only in continuous capture mode. 00: All frames are captured 01: Capture every other frame (reduce 50% bandwidth) 10: Capture 1 frame every 4 frames (75% bandwidth reduction) 11: reserved
7	VSPOL	RW	vertical sync polarity 0: VSYNC active low

			1: VSYNC active high
6	HSPOL	RW	Horizontal sync polarity 0: HSYNC active low 1: HSYNC active high
5	PCLKPOL	RW	Pixel clock polarity 0: Falling edge valid 1: Active on rising edge
4:3	reserved	RO reserved bit	
2	CROP	RW	CROP function 0: Disable 1: enable
1	CM	RW	capture mode 0: continue mode 1: Snapshot mode
0	CAPTURE	RW	capture enable 0: Disable 1: enable

22.3.3 DCMI Status Register (DCMI_SR)

Offset address: 0x0004 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:3	reserved	RO reserved bit	
2	FNE	RO FIFO non-empty judgment	
1	VSYNC	RO	VSYNC Pin Status 0: frame work 1: Inter-frame synchronization
0	HSYNC	RO	HSYNC Pin Status 0: row work 1: Inter-frame synchronization

22.3.4 DCMI Raw Interrupt Register (DCMI_RIS)

Offset address: 0x0008 Reset value: 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved										LINE_RIS	VSYNC_RIS	pre	Keep	OVR_RIS	FRAME_RIS

Bit	Name	W/R	Description
31:5	reserved	RO	reserved bit
4	LINE_RIS	RO	line raw break
3	VSYNC_RIS	RO	VSYNC Raw Interrupt
2	Rsvd	RO	reserved bit
1	OVR_RIS	RO	overflow raw interrupt
0	FRAME_RIS	RO	capture completes original interrupt

22.3.5 DCMI Interrupt Enable Register (DCMI_IER)

Offset address: 0x000C Reset value: 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3 2	1	0
reserved										LINE_IE	VSYNC_IE	pre Keep	OVR_IE	FRAME_IE

Bit	Name	W/R	Description
31:5	reserved	RO	reserved bit
4	LINE_IE	RW	line raw break 0: Interrupt disabled 1: Interrupt enable
3	VSYNC_IE	RW	VSYNC raw interrupt 0: Interrupt disabled 1: Interrupt enable
2	reserved	RO	reserved bit
1	OVR_IE	RW	overflow raw interrupt 0: Interrupt disabled 1: Interrupt enable
0	FRAME_IE	RW	Capture completes original interrupt 0: Interrupt disabled 1: Interrupt enable

22.3.6 DCMI Maskable Interrupt Status Register (DCMI_MIS)

Offset address: 0x0010 Reset value: 0x00000000

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

15	14	13	12	11	10	9	8 7	6	5	4	3	2	1	0
reserved									LINE_MIS	VSYNC_MIS	ERR_MIS	OVR_MIS	FRAME_MIS	

Bit	Name	W/R	Description
31:5	reserved	RO	reserved bit
4	LINE_MIS	RO	line raw break

			0: No interrupt is generated 1: An interrupt is generated
3	VSYNC_MIS	RO	VSYNC raw interrupt 0: No interrupt is generated 1: An interrupt is generated
2	ERR_MIS	RO	sync error raw interrupt 0: No interrupt is generated 1: An interrupt is generated
1	OVR_MIS	RO	overflow raw interrupt 0: No interrupt is generated 1: An interrupt is generated
0	FRAME_MIS	RO	Capture completes original interrupt 0: No interrupt is generated 1: An interrupt is generated

22.3.7 DCMI Interrupt Clear Register (DCMI_ICR)

Offset address: 0x0014 Reset value: 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved										LINE_ISC	VSYNC_ISC	ERR_ISC	OVR_ISC	FRAME_ISC	

Bit	Name	W/R	Description
31:5 Reserve		RO reserved bit	
4	LINE_ISC	WO write 1 line raw interrupt clear	
3	VSYNC_ISC	WO write 1VSYNC raw interrupt clear	
2	ERR_ISC	WO Write 1 Synchronization Error Raw Interrupt Clear	
1	OVR_ISC	WO write 1 overflow raw interrupt clear	
0	FRAME_ISC	WO Write 1 to capture complete original interrupt clear	

22.3.8 DCMI Crop Window Start Position Register (DCMI_CWSTRT)

Offset address: 0x0020 Reset value: 0x00000000

31 30 29 28 27 26 25 Reserved	15 14 Reserved	24 23 22 21 20 19	18	17	16
		VST			

	13	12	11	10	9	8	7 6	5	4	3	2	1	0
	VST							HOFFCNT					

Bit	Name	W/R	Description
31:29	reserved	RO reserved bit	
28:16	VST	RW	The vertical position, the value to indicate that the image capture starts from this line number, previous line counts are ignored 0x0000 => line 1 0x0001 => line 2

			0x0002 => line 3 ....
15:14	reserved	RO reserved bit	
13:0	HOFFCNT	RW	Horizontal position, representing the capture count, this value gives the value before the start of capture pixel clock count of

22.3.9 DCMI Crop Window Size Register (DCMI_CWSIZE)

Offset address: 0x0024 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:30	reserved	RO reserved bit	
29:16	VLINE	RW	Indicates the vertical line count, which gives the number of lines captured from the starting point 0x0000 => 1 line 0x0001 => 2 lines 0x0002 => 3 lines ...
15:14	reserved	RO reserved bit	
13:0	CAPCON	RW	Indicates the capture count, which gives the pixel captured at the beginning of the same line Clock width, it should word-align parallel interfaces of different widths. 0x0000 => 1 pixel 0x0001 => 2 pixels 0x0002 => 3 pixels

22.3.10 DCMI Data FIFO Entry Register (DCMI_DR)

Offset address: 0x0028 Reset value: 0x00000000

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

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

Bit	Name	W/R	Description
31:0	DATA	RO	Bits 31:24 Data byte 3 Bits 23:16 Data byte 2 Bits 15:8 Data byte 1 Bits 7:0 Data byte 0 The DCMI interface seals the received data before requesting DMA transfer. Installed in 32bit format, with a 4-word depth FIFO to reserve enough time for DMA transfer and effectively avoid DMA overflow

23 Charging module (CHARGE)

23.1 Introduction to Charging Module

Used to charge 3.7V lithium battery

ÿ Supports charging current up to 200mA

ÿ The fully charged voltage of the battery is 4.15±0.05V

ÿ After the battery is fully charged, the voltage will drop to 4.05V and the battery will be recharged

23.2Characteristics of charging module

The trickle charging voltage threshold is 2.9V. When it is lower than 2.9V, it adopts the trickle (small current) method to charge, and when it is higher than 2.9V, it adopts the constant current method. Charge. When the battery voltage is higher than 2.9V and the charging current is less than 1/10 of the constant current, it is considered to be fully charged, and the battery will no longer be charged.

The charging status can be queried through the CHG_CTRL register, and the charging current can be configured through the SEN_ANA0 register, as shown in the figure below.

23.3 Charge Status Register (CHG_CTRL)

Offset address: 0x011C Reset value: 0x00000000

31	30	29	28	27	26	25	twenty four
reserved		chg_state		reserved			
ro		ro		ro			

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							
ro							

15	14	13	12	11	10	9	8
reserved							
ro							

7	6	5	4	3	2	1	0
reserved							
ro							

Bit	Name	W/R	Description
31:30	reserved	-	reserved bits
29:28	chg_state	RO	charging: 11: full 10: Constant current charging 01: trickle charging 00: undervoltage
27:0	reserved	-	reserved bits

23.4 Analog Control Register 0 (SEN_ANA0)

Offset address: 0x0158 Reset value: 0x23500220

31	30	29	28	27	26	25	twenty four
soft_power r_down	xtal_current_sel		reserved	charge_current_sel			reserved

twenty three	twenty two	twenty one	20	19	18	17	16
reserved							

15	14	13	12	11	10	9	8
Reserve					rtc_32k_sel 2	reserved	

7	6	5	4	3	1	0
reserved						

Bit	Name	W/R	Description
31	soft_power_down	WO	set 1 soft shutdown
30:29	xtal_current_sel	RW	XTAL 32K current selection, 11 max
28	reserved	RO	reserved
27:25	charge_current_sel	RW	Maximum charging current selection: 000: 50mA 001: 60mA 010: 70mA 011: 80mA 100: 100mA 101: 120mA 110: 160mA 111: 200mA
24:11	reserved	RO	reserved
10	rtc_32k_sel	RW	RTC internal and external 32K selection 0: use internal 32K 1: Use external 32K
9:0	reserved	RO	reserved

24 switch circuit (POWER_KEY)

24.1 Introduction to switch circuit

The switch circuit realizes the switch function of the chip by controlling the internal 5V to 3.3V LDO to enable/disable.

24.2 Switch circuit characteristics

ÿ The power on/off circuit uses CHARGE_VBAT for power supply, and VBAT33 must be in place when using the power on/off function

ÿ When the chip is not powered on, the POWER_KEY is pulled high for 150ms to enable the internal LDO. After the chip is powered on, the POWER_KEY is pulled high for 7s to turn off
External LDO output

ÿ The software can query the state of the POWER_KEY key (corresponding to PF7), and can configure the interrupt enable of the POWER_KEY key (corresponding to PF7).

ÿ The reading state of PF7 is opposite to the actual level of the POWER_KEY pin, that is, if the POWER_KEY is not pressed (low level), the soft
The device reads the PF7 state as high

ÿ Software can control the shutdown of the internal 3.3V LDO

ÿ Software shutdown can be realized through the above two features

ÿ Support low power wake-up

ÿ When CHARGE_VCC is powered, the internal LDO directly enables the output of 3.3V and cannot be turned off

Remarks: The POWER_KEY key is usually low, press it and pull it high

24.3 Switch circuit registers

See the analog control register (SEN_ANA0) for the description of the register that software implements shutdown control.