CH567 Datasheet

Overview

CH567 is a high-performance 32-bit RISC reduced instruction set microcontroller, built-in 192KB FLASH memory, 32KB SRAM and 32KB DataFlash. The chip is integrated with two sets of independent high-speed USB2.0 master/slave controller, 4 groups of SD controllers, encryption algorithm modules, 4 groups of UART interfaces, 7 groups of PWMs, 3 groups of timers and other peripheral resources, which can be widely used for various embedded applications.

Function

- **Core:**
  - 32-bit RISC reduced instruction set core
  - Highest frequency of 120MHz
- **Memory:**
  -192KB-byte program memory, support write protection
  - 32KB-byte SRAM
  - 32KB-byte DataFlash
- **Double USB2.0 high-speed receiver-transmitter (built-in PHY):**
  - High-speed Host / Device mode
  - Support control/batch/interrupt/synchronous transmission
  - Support double buffer PINT-PONG mechanism
  - Support DMA
- **4 groups of independent SD controllers:**
  - Support single-wire, 4-wire, 8-wire communication mode
  - Support SD/TF card, SDIO card and eMMC card, etc.
  - Built-in FIFO
  - Support AES and SM4 Algorithms
  - Provide 8 encryption and decryption modes
  - Support DMA
- **Timer:**
  - 3 sets of 26-bit timers
  - Support functions of signal width sample / edge capture, PWM adjustable output and count
  - TMR1 and TMR2 support DMA
- **Universal asynchronous receiver/transmitter (UART):**
  - 4 groups of independent UARTs, compatible with 16C550
  - The highest baud rate is 6Mbps
  - Built-in FIFO, multiple trigger levels
- **PWMX:**
  - Expand 4 sets of PWM output
  - Adjustable duty cycle
- **SPI:**
  - 2 sets of SPI interfaces: one supports Master and Slave mode, and the other supports only Master mode
  - Built-in FIFO
  - SPI0 supports DMA
- **LED screen interface:**
  - Support 1/2/4-channel data lines
  - Built-in FIFO, support double buffer
  - Support DMA
- **Active parallel port:**
  - 8-bit data, 15-bit address bus
  - Adjustable timing sequence
- **Low power:**
  - Sleep mode
  - Support some GPIOs or USB signal wake up
- **General-purpose I/O port:**
  - 30 GPIOs
  - 8 pins can be set level or edge interrupt
  - Some pins have multiplexing and mapping functions
- **ID Number of chip:**
  - Unique 64bit ID identification number
- **Power supply:**
  - 3.0---3.6V (3.3V±10%)
- **Package: LQFP48**

Application

Security storage, home security, USB-related applications, monitoring, alarm systems, printers, scanners and other application control.

Relevant information can be downloaded from the website: www.wch.cn

Chapter 1 Pin Information

1.1 Pin Configuration

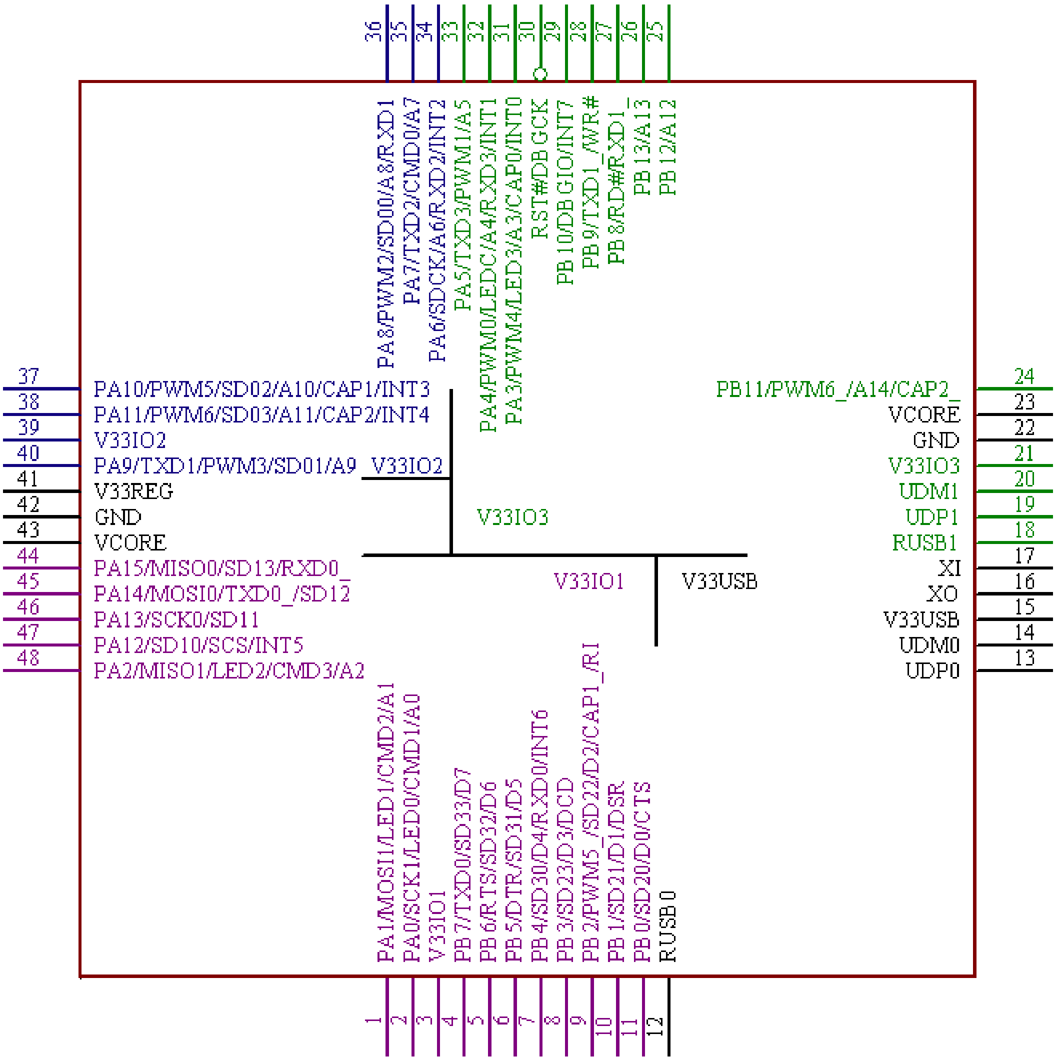


Figure 2-1 LQFP48 Package Pin Arrangement

1.2 Pin Description

Pin No.	Pin Name	Pin Type	Main Function (after reset)/ Multiplexing Function and Mapping	Function Description
1	PA1	I/O	PA1 /MOSI1/LED1/CMD2/A1	PA1: General purpose bidirectional digital I/O pin. MOSI1: SPI1 serial data pin, host output. LED1: LED serial data line 1. CMD2: SD2 controller command signal line. A1: Parallel port address line 1.
2	PA0	I/O	PA0 /SCK1/LED0/CMD1/A0	PA0: General purpose bidirectional digital I/O pin. SCK1: SPI1 serial clock pin, host clock output. LED0: LED serial data line 0.

				CMD1: SD1 controller command signal line. A0: Parallel port address.
3	V33IO1	P	V33IO1	1 set of 3.3V power supplies for peripherals, requires an external 0.1uF capacitor.
4	PB7	I/O	PB7 /TXD0/SD33/D7	PB7: General purpose bidirectional digital I/O pin. TXD0: UART0 serial data output. SD33: SD3 controller data line 3. D7: Parallel data line 7.
5	PB6	I/O	PB6 /RTS/SD32/D6	PB6: General purpose bidirectional digital I/O pin. RTS: MODEM output signal of UART0; request to send. SD32: SD3 controller data line 2. D6: Parallel data line 6.
6	PB5	I/O	PB5 /DTR/SD31/D5	PB5: General purpose bidirectional digital I/O pin. DTR: MODEM output signal of UART0; data terminal is ready. SD31: SD3 controller data line 1. D5: Parallel data line 5.
7	PB4	I/O	PB4 /SD30/D4/RXD0/INT6	PB4: General purpose bidirectional digital I/O pin. SD30: SD3 controller data line 0. D4: Parallel data line 4. RXD0: UART0 serial data input. INT6: IO interrupt 6.
8	PB3	I/O	PB3 /SD23/D3/DCD	PB3: General purpose bidirectional digital I/O pin. SD23: SD2 controller data line 3. D3: Parallel data line 3. DCD: MODEM input signal of UART0; carrier detection.
9	PB2	I/O	PB2 /PWM5_/SD22/D2/CAP1_/RI	PB2: General purpose bidirectional digital I/O pin. PWM5_: PWM5 function mapping. SD22: SD2 controller data line 2. D2: Parallel data line 2. CAP1_: CAP1 function mapping. RI: MODEM input signal of UART0; ring indicator.
10	PB1	I/O	PB1 /SD21/D1/DSR	PB1: General purpose bidirectional digital I/O pin. SD21: SD2 controller data line 1. D1: Parallel data line 1. DTR: MODEM input signal of UART0; data device is ready.
11	PB0	I/O	PB0 /SD20/D0/CTS	PB1: General purpose bidirectional digital I/O pin. SD20: SD2 controller data line 0. D0: Parallel data line 0. CTS: MODEM input signal of UART0; clear to send.
12	RUSB0	I/O	RUSB0	It is required to be connected with a 12KΩ resistor to ground for USB0-PHY.
13	UDP0	USB	UDP0(UD0+)	USB0 bus D+ data line.
14	UDM0	USB	UDM0(UD0-)	USB0 bus D- data line.
15	V33USB	P	V33USB	USB0 peripheral power supply.
16	XO	I/O	XO	Crystal oscillator inverted output

17	XI	I	XI	Crystal oscillator input
18	RUSB1	I/O	RUSB1	It is required to be connected with a 12KΩ resistor to ground for USB1-PHY.
19	UDP1	USB	UDP1(UD1+)	USB1 bus D+ data line.
20	UDM1	USB	UDM1(UD1-)	USB1 bus D- data line.
21	V33IO3	P	V33IO3	3 sets of 3.3V power supply for USB1 and peripheral, requires an external 0.1uF capacitor.
22	GND	P	GND	Ground: common ground, 0V reference point.
23	VCORE	P	VCORE	Core power supply, must be connected to VCORE and an external 0.1uF capacitor is required.
24	PB11	I/O	PB11 /PWM6_/A14/CAP2_	PB11: General purpose bidirectional digital I/O pin. PWM6_: PWM6 function mapping. A14: Parallel port address line 14. CAP2_: CAP2 function mapping.
25	PB12	I/O	PB12 /A12	PB12: General purpose bidirectional digital I/O pin. A12: Parallel port address line 12.
26	PB13	I/O	PB13 /A13	PB13: General purpose bidirectional digital I/O pin. A13: Parallel port address line 13.
27	PB8	I/O	PB8 /RD#/RXD1_	PB9: General purpose bidirectional digital I/O pin. RD#: Parallel port read control line. RXD1_: RXD pin mapping of UART1.
28	PB9	I/O	PB9 /TXD1_/WR#	PB9: General purpose bidirectional digital I/O pin. TXD1_: TXD pin mapping of UART1. WR#: Parallel port write control line.
29	PB10	I/O	PB10 /DBGIO/INT7	PB10: General purpose bidirectional digital I/O pin. DBGIO: Data input and output port of simulation and debugging interface. INT7: IO interrupt 7.
30	RST#	I	RST# /DBGCK	RST#: External reset input pin, active low, built-in pull-up resistor. DBGCK: Clock input port of simulation and debugging interface.
31	PA3	I/O	PA3 /PWM4/LED3/A3/CAP0 /INT0	PA3: General purpose bidirectional digital I/O pin. PWM4: Pulse width modulation output channel 4. LED3: LED serial data line 3. A3: Parallel port address line 3. CAP0: Timer0 capture input pin. INT0: IO interrupt 0.
32	PA4	I/O	PA4 /PWM0/LEDC/A4/RXD3 /INT1	PA4: General purpose bidirectional digital I/O pin. PWM0: Pulse width modulation output channel 0. LEDC: LED serial data line. A4: Parallel port address line 4. RXD3: UART3 serial data input. INT1: IO interrupt 1.
33	PA5	I/O	PA5 /TXD3/PWM1/A5	PA5: General purpose bidirectional digital I/O pin. TXD3: UART3 serial data output. PWM1: Pulse width modulation output channel 1. A4: Parallel port address line 5.
34	PA6	I/O	PA6	PA6: General purpose bidirectional digital I/O pin.

			/SDCK/A6/RXD2/INT2	SDCK: SD controller clock line output. A6: Parallel port address line 6. RXD2: UART2 serial data input. INT2: IO interrupt 2.
35	PA7	I/O	PA7 /TXD2/CMD0/A7	PA7: General purpose bidirectional digital I/O pin. TXD2: UART2 serial data output. CMD0: SD0 controller command signal line. A7: Parallel port address line 7.
36	PA8	I/O	PA8 /PWM2/SD00/A8/RXD1	PA8: General purpose bidirectional digital I/O pin. PWM2: Pulse width modulation output channel 2. SDOO: SD0 controller data line 0. A8: Parallel port address line 8. RXD1: UART1 serial data input.
37	PA10	I/O	PA10 /PWM5/SD02/A10/CAP1 /INT3	PA10: General purpose bidirectional digital I/O pin. PWM5: Pulse width modulation output channel 5. SDO2: SD0 controller data line 2. A10: Parallel port address line 10. CAP1: Timer 1 capture input pin. INT3: IO interrupt 3.
38	PA11	I/O	PA11 /PWM6/SD03/A11/CAP2 /INT4	PA11: General purpose bidirectional digital I/O pin. PWM6: Pulse width modulation output channel 6. SD03: SD0 controller data line 3. A11: Parallel port address line 11. CAP2: Timer2 capture input pin. INT4: IO interrupt 4.
39	V33IO2	P	V33IO2	2 sets of 3.3V power supplies for peripherals, requires an external 0.1uF capacitor.
40	PA9	I/O	PA9 TXD1/PWM3/SD01/A9	PA9: General purpose bidirectional digital I/O pin. TXD1: UART1 serial data output. PWM3: Pulse width modulation output channel 3. SD01: SD0 controller data line 1. A9: Parallel port address line 9.
41	V33REG	P	V33REG	Power supply voltage regulator input 3.3V power, requires an external 0.1uF capacitor.
42	GND	P	GND	Ground: common ground, 0V reference point.
43	VCORE	P	VCORE	Core power supply output, and an external 3.3uF or 10uF capacitor is required.
44	PA15	I/O	PA15 /MISO0/SD13/RXD0_	PA15: General purpose bidirectional digital I/O pin. MIOS0: SPI0 serial data pin, host input/slave output. SD13: SD1 controller data line 3. RXD0_: RXD pin mapping of UART0.
45	PA14	I/O	PA14 /MOSI0/TXD0_/SD12	PA14: General purpose bidirectional digital I/O pin. MOSI0: SPI0 serial data pin, host output/slave input. TXD0_: TXD pin mapping of UART0. SD12: SD1 controller data line 2.
46	PA13	I/O	PA13 /SCK0/SD11	PA13: General purpose bidirectional digital I/O pin. SCK0: SPI0 serial clock pin, host output/slave input. SD11: SD1 controller data line 1.
47	PA12	I/O	PA12	PA12: General purpose bidirectional digital I/O pin.

			/SD10/SCS/INT5	SD10: SD1 controller data line 0. SCS: Chip selection input pin of SPI0 slave. INT5: IO interrupt 5.
48	PA2	I/O	PA2 /MISO1/LED2/CMD3/A2	PA2: General purpose bidirectional digital I/O pin. MISO1: SPI1 serial data pin, host input; serial data input and output pin in SPI1 simplex mode. LED2: LED serial data line 2. CMD3: SD3 controller command signal line. A2: Parallel port address line 2.

ATTENTION:

(1). I: Input; O: Output; P: Power.

(2) USB: USB signal.

(3) The priority of multiplexing functions of the pins in the table are arranged in order from high to low (excluding the main function GPIO function)

Remarks :

In order to be compatible with the power supply system of external devices, CH567 will divide the power supply of peripherals for management and IOs and provide multiple sets of power pins. In the above pin description, the pins marked with different colors belong to different power domains, and the assignments are as follows:

Font color: Same as power supply V33REG/V33USB

Font color: Same as power supply V33IO1

Font color: Same as power supply V33IO2

Font color: Same as power supply V33IO3

Chapter 2 System Structure and Memory

2.1 System Structure

The following figure shows the system structure block diagram of CH567 chip.

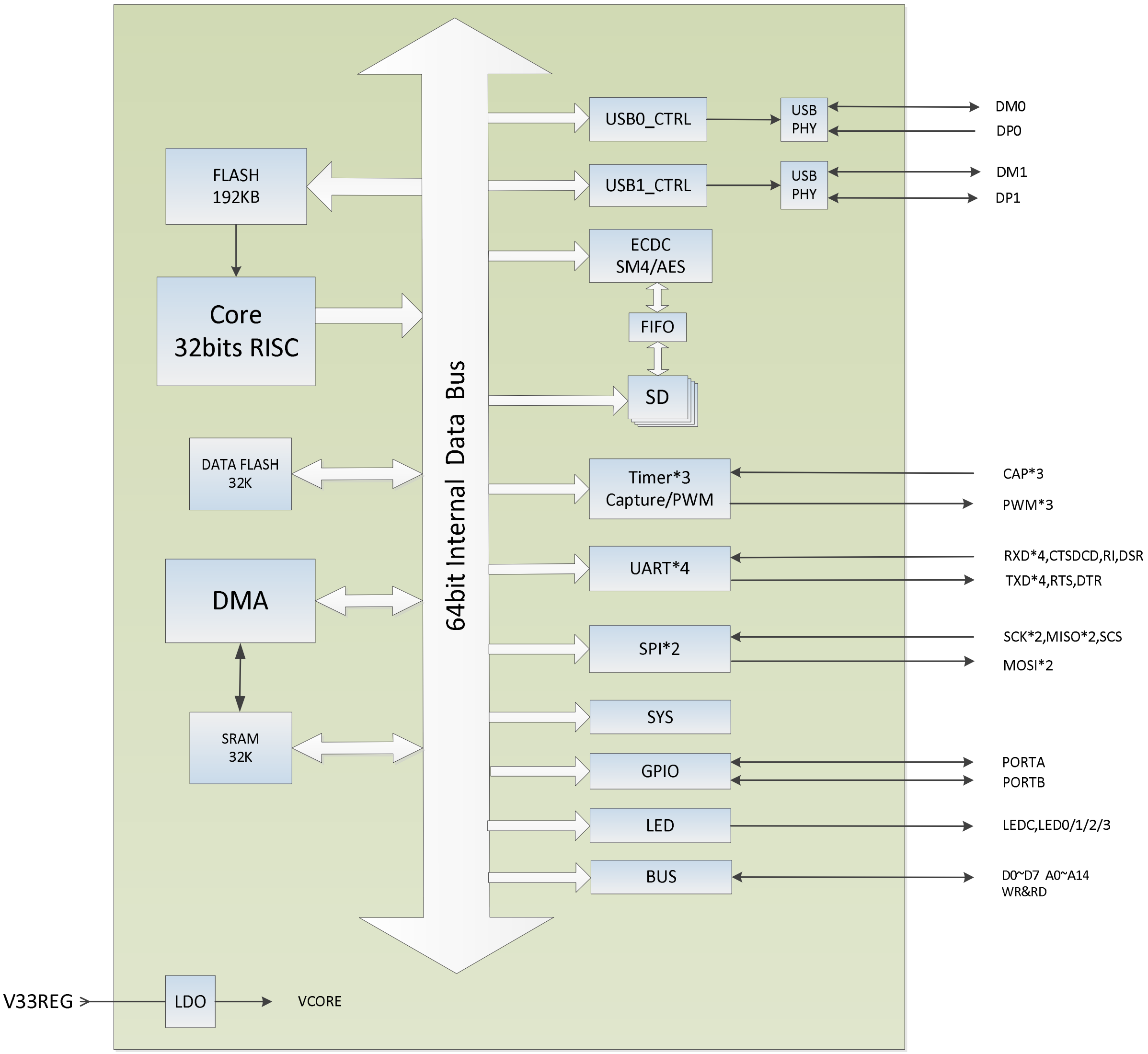


Figure 2-1 CH567 Internal Structure Block Diagram

The CPU core, DMA arbitration controller, SRAM and various peripheral modules are mounted on the 64bit system bus of CH567. The DMA controller can be used for modules such as USB, SATA, SD, SPI0, LED and TIMER of peripherals.

2.2 Memory Mapping

CH567 includes a 4GB address space, and the memory map mainly contains several different areas, as shown in the figure below.

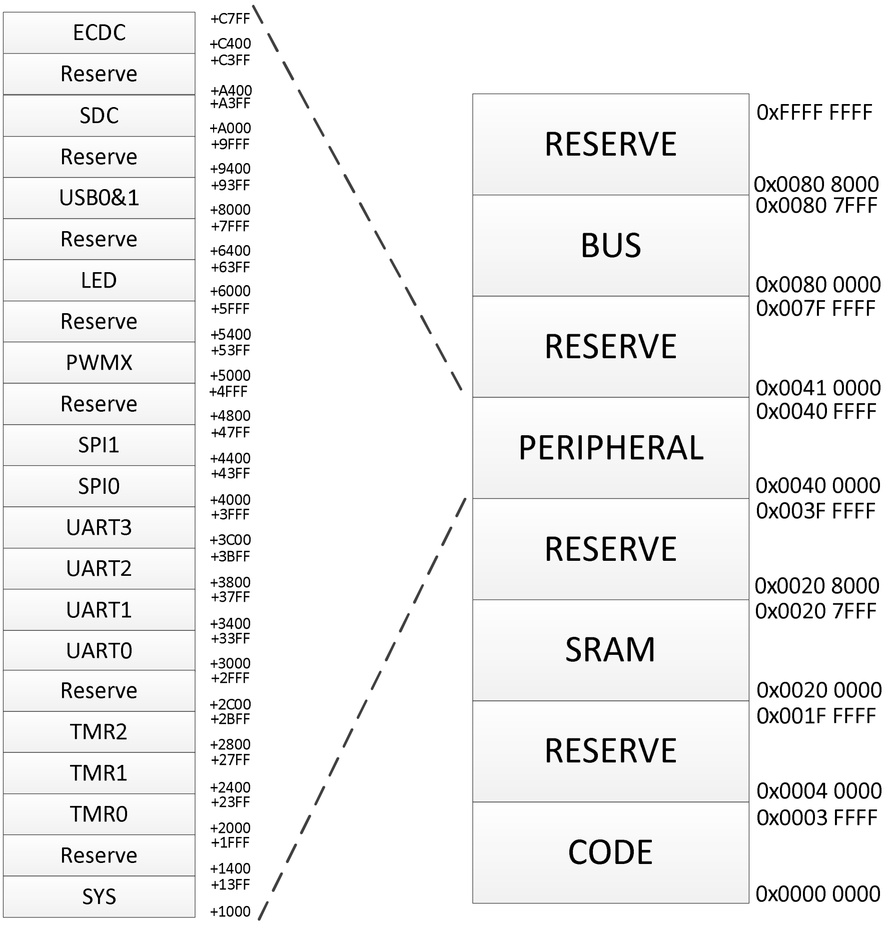


Figure 2-2 Memory Mapping

2.3 Memory Mapping

The address range of each memory mapping area is shown in the table below:

Table 2-1 Memory Mapping Area Address

Address range	Usage	Description
0x0000 0000-0x0003 FFFF	On-chip non-volatile memory	Flash memory (256KB)
0x0004 0000-0x001F FFFF	Reserved	-
0x0020 0000-0X0020 7FFF	On-chip SRAM, usually used to store data	32KB
0x0020 8000-0x003F FFFF	Reserved	-
0x0040 0000-0x0040 FFFF	Various peripherals	Multiple peripheral modules
0x0041 0000-0x007F FFFF	Reserved	-
0x0080 0000-0x0080 7FFF	External system bus	32KB
0x0080 8000-0xFFFF 7FFF	Reserved	-

2.4 Peripheral Address Assignment

CH567 mainly contains the following peripherals. Each peripheral occupies a certain address space, and the actual access address of peripheral register is: base address + offset address. In the following chapters, the address of the register is described in detail. The following table shows the assignment of base address of each peripheral.

Table 2-2 Peripheral Base Address Assignment

Peripheral No.	Peripheral name	Peripheral base address

1	SYS	0x0040 1000
2	TMR0	0x0040 2000
3	TMR1	0x0040 2400
4	TMR2	0x0040 2800
5	UART0	0x0040 3000
6	UART1	0x0040 3400
7	UART2	0x0040 3800
8	UART3	0x0040 3C00
9	SPI0	0x0040 4000
10	SPI1	0x0040 4400
11	PWMX	0x0040 5000
12	LED	0x0040 6000
13	USB0	0x0040 8000
14	USB1	0x0040 9000
15	SDC	0x0040 A000
16	ECDC	0x0040 C400

The following table is the explanation of "Access" in the register description in the subsequent chapters:

Abbreviation	**Description**
RF	The read value is fixed, which is not affected by reset.
RO	Read only.
WO	Write only (the read value is 0 or invalid).
RZ	Read-only, automatically cleared after read operation.
WZ	Write to clear.
RW	Readable, writable.
RW1	Cleared by reading/writing 1.
WA	Write-only (in safety mode), the read value is 0 or invalid.
RWA	Write in read/safety mode.

Chapter 3 System Control

3.1 Power Control

CH567 needs an external supply voltage of 3.3V. In the package pin description of Figure 1-1, multiple sets of power and ground are provided externally. Internal power management is provided with multi-power-domain grouping mode, which can connect different power systems according to the peripheral resources used.

After the system or power supply is reset, CH567 is in run state. When the CPU does not need to continue to run, or some functional modules do not need to be used, the clock or independent supply of these modules can be turned off, to reduce power consumption.

3.2 Reset Control

CH567 supports 3 types of reset forms, namely power-on reset, external manual reset and internal software reset. System will reload the configuration information and reload the program code into the RAM buffer after reset. The load time is about 8.8mS.

The register R8_GLOB_RESET_KEEP is only reset when power-on reset, and it is not affected by other types of reset.

3.2.1 Power-on Reset

When the power supply voltage is lower than the power-on reset threshold Vpot, CH567 will be reset. The figure below shows the power-on reset of CH567.

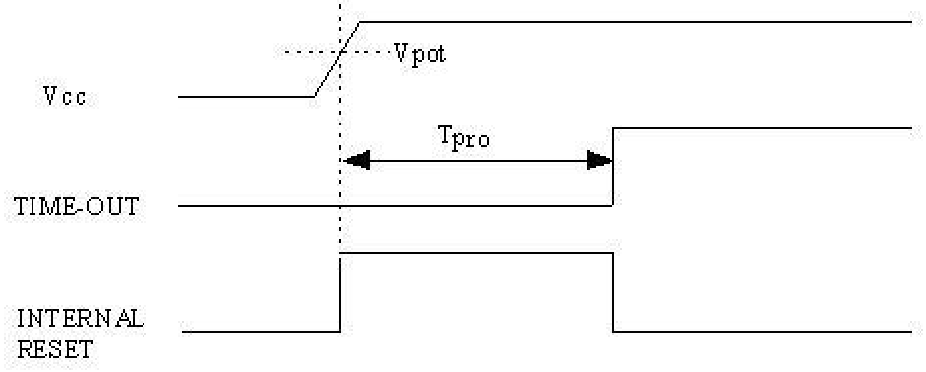


Figure 3-1 Power-on Reset

3.2.2 External Manual Reset

The external manual reset is generated by the low level applied to the RST# pin from the external. When the reset low level duration is greater than the minimum reset pulse width (Trst), the chip will be triggered to reset.

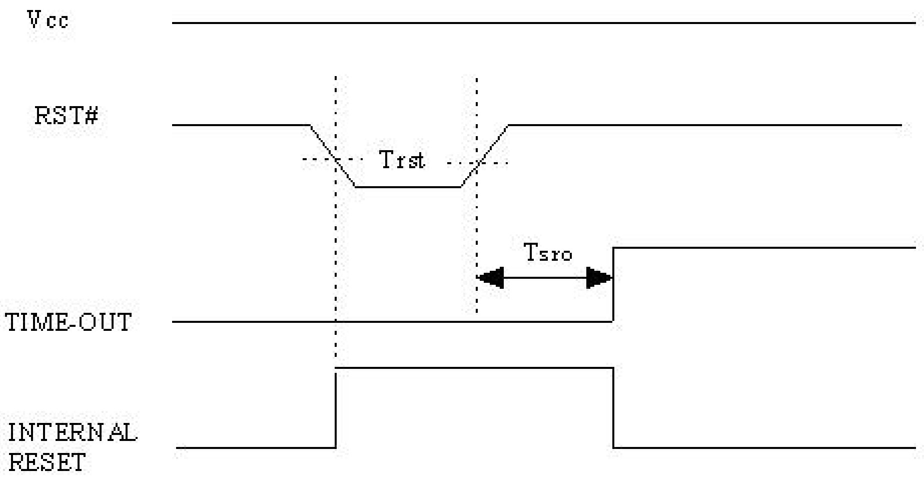


Figure 3-2 External Reset

3.2.3 Internal Software Reset

The chip is provided with the internal software reset function, so that no external intervention is required to perform software reset in some specific situations. Set the bit RB_SOFTWARE_RESET of global reset configuration register (R8_RST_WDOG_CTRL) to 1, to realize software reset. This bit will be automatically cleared.

3.2.4 Reset Feature

Please refer to the timing parameter table in section 15.4 for reset parameters.

3.3 Register Description

System control related register physical base address: 0x0040 1000

Table 3-1 List of Clock and CPU Control Related Registers

Name	Offset address	Description	Reset value
R8_SAFE_ACCESS_SIG	0x00	Secure access flag register	8h00
R8_CHIP_ID	0x01	Chip ID register	8h68
R8_SAFE_ACCESS_ID	0x02	Secure access ID register	8h02
R8_GLOB_ROM_CFG	0x04	ROM configuration register	8h80
R8_RST_BOOT_STAT	0x05	BOOT status register	8hC1
R8_RST_WDOG_CTRL	0x06	Reset register	8h00
R8_GLOB_RESET_KEEP	0x07	Reset keep register	8h00
R8_SLP_WAKE_CTRL	0x0E	Wake-up control register	8h00
R8_SLP_POWER_CTRL	0x0F	Low-power power management register	8h00

Security access flag register (R8_SAFE_ACCESS_SIG):

Bit	Name	Access	Description	Reset value
[7:0]	R8_SAFE_ACCESS_SIG	WO	Secure access flag register. Some registers (access attribute is RWA) are protected registers, read/write operation can be performed only after entering the secure access mode. Write 0x57 first and then 0xA8 to this register, to enter the secure access	00h

			mode. And the time is limited to about 110 main clock cycles (Tsys), and it will be automatically protected beyond the time limit.	
[6:4]	RB_SAFE_ACC_TIMER	RO	Secure access time, fixed at 128 Tsys	0
[1:0]	RB_SAFE_ACC_MODE	RO	Current secure access mode status: 11: Secure mode, registers with RWA attribute can be accessed; Other: Non-secure mode;	0

Chip ID register (R8_CHIP_ID)

Bit	Name	Access	Description	Reset value
[7:0]	R8_CHIP_ID	RF	Fixed value of 67h, used to identify the chip.	67h

Secure access ID register (R8_SAFE_ACCESS_ID):

Bit	Name	Access	Description	Reset value
[7:0]	R8_SAFE_ACCESS_ID	RF	Fixed value of 02h.	02h

ROM configuration register (R8_GLOB_ROM_CFG):

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RWA	Reserved, must write 01b to [7:6], the read is 0.	0000b
3	RB_ROM_CODE_WE	RWA	Flash ROM code and data area erase/program enable bit: 1: Programmable/erasable; 0: Write protection	0
2	RB_ROM_DATA_WE	RWA	Flash ROM data area erase/program enable bit: 1: Programmable/erasable; 0: Write protection.	0
1	RB_CODE_RAM_WE	RWA	Code RAM area write enable bit: 1: Write enabled; 0: Write protection.	0
0	RB_ROM_EXT_RE	RO	External programmer read Flash ROM enable bit: 1: Read enabled; 0: Read protection.	0

BOOT status register (R8_RST_BOOT_STAT)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved.	11b
5	RB_BOOT_LOADER	RO	Boot loader status: 1: Boot loader status (Boot-Loader); 0: User program status.	0
4	RB_CFG_DEBUG_EN	RO	Debugging enable control bit: 1: Enable; 0: Disable.	0

3	RB_CFG_BOOT_EN	RO	Boot program enable control bit: 1: Enable; 0: Disable.	0
2	RB_CFG_RESET_EN	RO	External reset enable control bit: 1: External input low-level signal reset; 0: Disable.	0
[1:0]	RB_RESET_FLAG	RO	Last reset flag, as shown in Table 3-2.	1

Table 3-2 Last Reset Flag Description

RB_RESET_FLAG	Reset Flag Description
00b	Software reset, source: RB_SOFTWARE_RESET=1 and RB_BOOT_LOADER=0.
01b	Power on reset, source: Chip operating voltage is lower than the threshold voltage.
11b	Manual reset, source: RST# pin input low level.

Reset register (R8_RST_WDOG_CTRL)

Bit	Name	Access	Description	Reset value
[7:1]	Reserved	RO	Reserved, must write 01b to [7:6].	00h
0	RB_SOFTWARE_RESET	WA/ WZ	System software reset; cleared automatically: 1: System reset; 0: No action	0

Reset keep register (R8_GLOB_RESET_KEEP)

Bit	Name	Access	Description	Reset value
[7:0]	R8_GLOB_RESET_KEEP	RW	Reset keep register. The value of this register is not affected by manual reset, software reset or watchdog reset.	00h

Wake-up control register (R8_SLP_WAKE_CTRL)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved.	0
4	RB_SLP_GPIO_WAKE	RWA	GPIO port wake-up enable control bit: 1: Enable; 0: Disable.	0
[3:2]	Reserved	RO	Reserved.	0
1	RB_SLP_USB1_WAKE	RWA	USB1 wake-up enable control bit: 1: Enable; 0: Disable.	0
0	RB_SLP_USB0_WAKE	RWA	USB0 wake-up enable control bit: 1: Enable; 0: Disable.	0

Low-power power management register (R8_SLP_POWER_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_SLP_STANDBY	RWA	Low-power mode control bit; automatically	0

			cleared after entering the mode: 1: Request core low power; 0: No action.	
[6:2]	Reserved	RO	Reserved.	0
1	RB_SLP_USB1_PWRDN	RWA	USB1 power control bit: 1: Power off; 0: Power on normally.	0
0	RB_SLP_USB0_PWRDN	RWA	USB0 power control bit: 1: Power off; 0: Power on normally.	0

3.4 Low Power Mode and Wake-up

In the low-power state (RB_SLP_STANDBY bit is set to 1), the PLL will stop working, the internal clock of CH567 will be suspended, the CPU will not work or respond to any interrupts. However, if CPU starts working after waking up, and it is found that the wake-up event is also an interrupt event (for example, a certain GPIO wakes up and GPIO interrupt is generated), it will be treated as an interrupt.

In order to reduce power, the physical PHY module (such as USB) that is not used during the low power period shall be turned off before entering the low power state. Set RB_SLP_USB0_PWRDN bit to 1, and set RB_SLP_USB1_PWRDN bit to 1. In addition, GPIO pins cannot be in floating state, and need to be set to output state or input state with an external fixed level. If there is no external input with a fixed level, they need to be set to input state under internal pull-down mode.

In low power mode, CH567 only supports the wake-up of part of GPIO or USB, please refer to R8_SLP_WAKE_CTRL register.

There are 8 GPIO pins that support wake-up, which are 8 pins that support GPIO interrupt. GPIO wake-up event source is the same as the GPIO interrupt event source, but only triggered by level (R8_GPIO_INT_MODE is not required). When the bit corresponding to R8_GPIO_INT_POLAR is 0, a low level occurs on GPIO pin, then it will wake up. When the bit corresponding to R8_GPIO_INT_POLAR is 1, a high level occurs on GPIO pin, then it will wake up.

Take GPIO port PA3 wake-up as an example, the configuration is as follows:

RB_GPIO_PA3_IP=0;
RB_GPIO_PA3_IE=1;
RB_SLP_GPIO_WAKE=1;

When a low level occurs at PA3 port, a wake-up event will be generated. After CH567 exits the low-power mode, it will trigger the GPIO interrupt of PA3 port.

Chapter 4 Clock Control

4.1 Clock Block Diagram

The internal clock structure of CH567 is shown in the figure below:

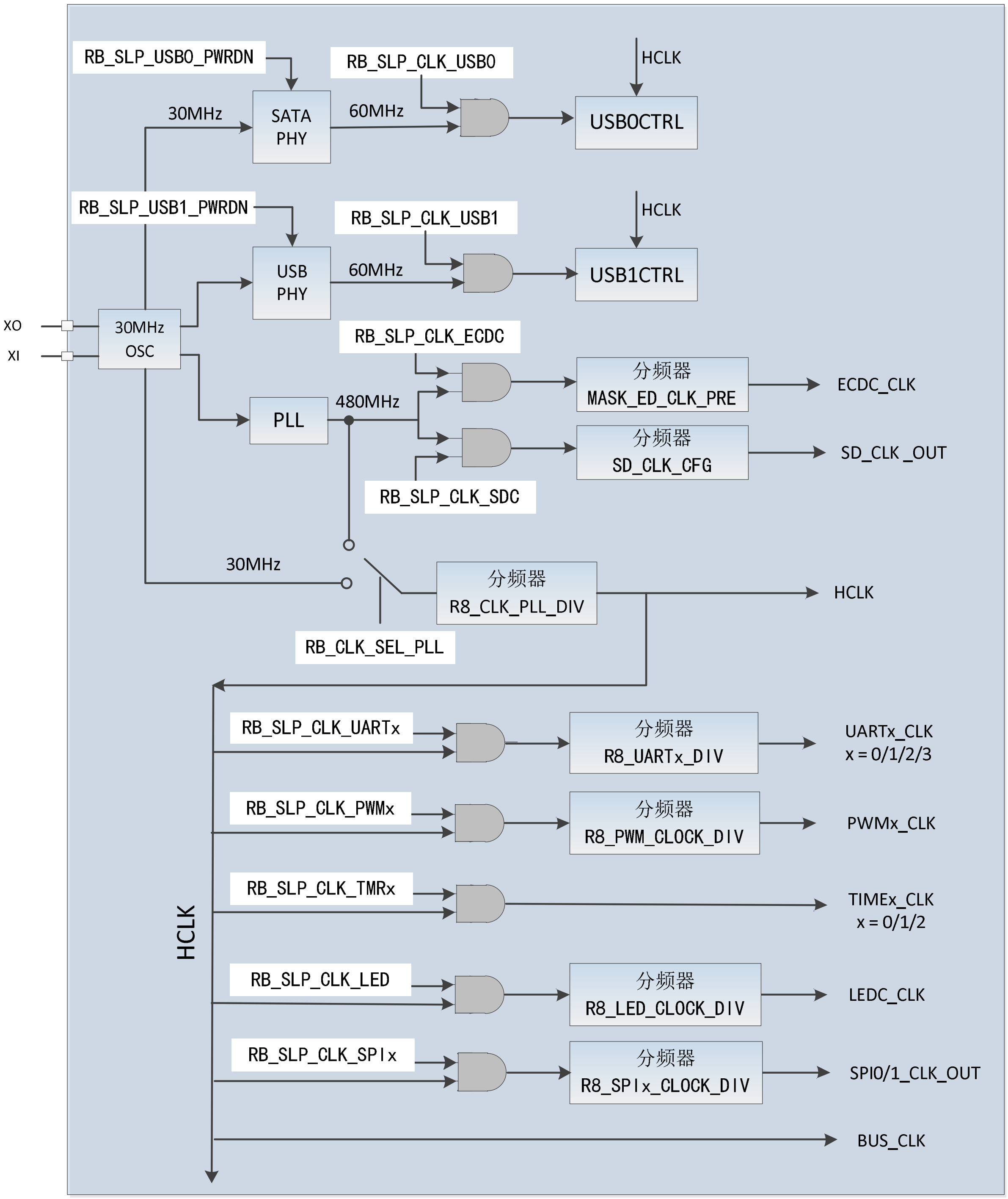


Figure 4-1 Clock Structure Block Diagram

After the external clock is sent to CH567, it will be connected to USB-PHY to generate the clock frequency required by USB controller, and generate the frequency multiplication clock of 480MHz through the PLL module. Get the clock frequency of 30MHz or 480MHz before frequency division through the clock source selection control bit (RB_CLK_SEL_PLL). At this time, the clock frequency will get the system clock Fsys (HCLK) through the frequency divider (R8_CLK_PLL_DIV), namely the main clock of CPU, with a range of 2MHz- 120MHz.

Each peripheral module clock has a corresponding clock register control bit, which can be turned on or off independently. In order to reduce the power of chip, you can turn off the function module clocks that are not used.

4.2 Register Description

Clock control related register physical base address: 0x0040 1000

Table 4-1 List of Clock Control Related Registers

Name	Offset address	Description	Reset value
R8_CLK_PLL_DIV	0x08	PLL output clock divider register	8h42
R8_CLK_CFG_CTRL	0x0A	Clock configuration register	8h80
R8_SLP_CLK_OFF0	0x0C	Sleep control register 0	8h00
R8_SLP_CLK_OFF1	0x0D	Sleep control register 1	8h00

PLL output clock divider register (R8_CLK_PLL_DIV)

Bit	Name	Access	Description	Reset value
[7:0]	R8_CLK_PLL_DIV	RWA	The lower 4 bits are valid, must write 01b to [7:6], and the minimum value is 2.	8h42

Clock configuration register (R8_CLK_CFG_CTRL)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RWA	Reserved, must write 01b to [7:6].	100000b
1	RB_CLK_SEL_PLL	RWA	Clock source selection: 1:PLL 480MHz; 0: External crystal oscillator of 30MHz.	0
0	RB_CLK_PLL_SLEEP	RWA	PLL sleep control bit: 1: PLL sleep; 0: PLL works normally.	0

Sleep control register 0 (R8_SLP_CLK_OFF0)

Bit	Name	Access	Description	Reset value
7	RB_SLP_CLK_UART3	RWA	UART3 clock control bit: 1: UART3 clock is turned off; 0: UART3 clock is turned on.	0
6	RB_SLP_CLK_UART2	RWA	UART2 clock control bit: 1: UART2 clock is turned off; 0: UART2 clock is turned on.	0
5	RB_SLP_CLK_UART1	RWA	UART1 clock control bit: 1: UART1 clock is turned off; 0: UART1 clock is turned on.	0
4	RB_SLP_CLK_UART0	RWA	UART0 clock control bit: 1: UART0 clock is turned off;	0

			0: UART0 clock is turned on.	
3	RB_SLP_CLK_PWMX	RWA	PWM clock control bit: 1: PWM clock is turned off; 1: PWM clock is turned on.	0
2	RB_SLP_CLK_TMR2	RWA	TIMER2 clock control bit: 1: TIMER2 clock is turned off; 0: TIMER2 clock is turned on.	0
1	RB_SLP_CLK_TMR1	RWA	TIMER1 clock control bit: 1: TIMER1 clock is turned off; 0: TIMER1 clock is turned on.	0
0	RB_SLP_CLK_TMR0	RWA	TIMER0 clock control bit: 1: TIMER0 clock is turned off; 0: TIMER0 clock is turned on.	0

Sleep control register 1 (R8_SLP_CLK_OFF1)

Bit	Name	Access	Description	Reset value
7	RB_SLP_CLK_ECDC	RWA	ECDC (encryption/decryption module) clock control bit: 1: ECDC clock is turned off; 1: ECDC clock is turned on.	0
6	Reserved	RO	Reserved.	0
5	RB_SLP_CLK_USB1	RWA	USB clock control bit: 1: USB clock is turned off; 1: USB clock is turned on.	0
4	RB_SLP_CLK_USB0	RWA	USB0 clock control bit: 1: USB0 clock is turned off; 0: USB0 clock is turned on.	0
3	RB_SLP_CLK_LED	RWA	LEDC clock control bit: 1: LEDC clock is turned off; 1: LEDC clock is turned on.	0
2	RB_SLP_CLK_SDC	RWA	SDC clock control bit: 1: SDC clock is turned off; 1: SDC clock is turned on.	0
1	RB_SLP_CLK_SPI1	RWA	SPI1 clock control bit: 1: SPI1 clock is turned off; 0: SPI1 clock is turned on.	0
0	RB_SLP_CLK_SPI0	RWA	SPI0 clock control bit: 1: SPI0 clock is turned off; 0: SPI0 clock is turned on.	0

4.3 System Clock Configuration

External crystal oscillator clock: Fosc = 30MHz;

PLL frequency multiplication clock: Fpll = 480MHz;

1. Select PLL clock source: Fsrc = RB_CLK_SEL_PLL? PLL_FREQ : OSC_FREQ;
2. System clock calculation: Fsys = Fsrc / R8_CLK_PLL_DIV, (2MHz - 120MHz).

When the system is powered on, 30MHz is selected as the PLL clock source by default, the frequency division factor is 2, and the default main frequency is 15MHz.

Chapter 5 General purpose and Multiplexing I/O

5.1 Introduction to GPIO

The system is equipped with 2 sets of GPIO ports of PA and PB, with a total of 30 general purpose input/output pins, and some pins have multiplexing and mapping functions. Each GPIO port has a 32-bit direction configuration register (R32_Px_DIR), a 32-bit data input register (R32_Px_PIN), a 32-bit data output register (R32_Px_OUT), a 32-bit data clear register (R32_Px_CLR), a 32-bit pull-up configuration register (R32_Px_PU), a 32-bit open drain output and input pull-down configuration register (R32_Px_PD), a 32-bit I/O drive capability configuration register (R32_Px_DRV) and a 32-bit Schmitt trigger enable configuration register (R32_Px_SMT).

In the PA port, the PA[0]-PA[15] bits are valid, corresponding to the 16 GPIO pins on the chip. In the PB port, the PB[0]-PB[13] bits are valid, corresponding to the 14 GPIO pins on the chip. Among them, 8 GPIO ports have interrupt function and can realize sleep and wake-up function.

Each I/O port bit can be freely programmed, but the I/O port register must be accessed by 8-bit, 16-bit or 32-bit words. If the multiplexing function of pin is not enabled, it will be used as a general-purpose I/O port.

The following figure is a block diagram of the internal structure of GPIO:

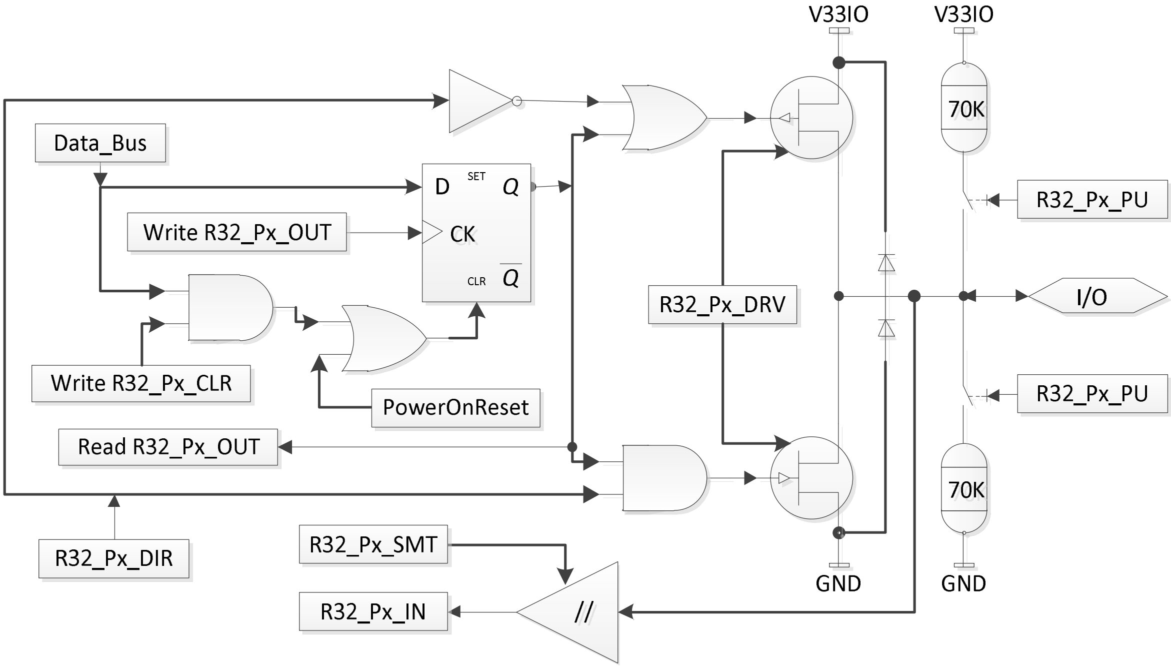


Figure 5-1 IO Internal Structure Block Diagram

5.2 Register Description

GPIO related register physical base address: 0x0040 1000

Table 5-1 List of GPIO Related Registers

Name	Offset address	Description	Reset value
R8_GPIO_INT_STATUS	0x1C	GPIO interrupt flag register	8h00
R8_GPIO_INT_ENABLE	0x1D	GPIO interrupt enable register	8h00
R8_GPIO_INT_MODE	0x1E	GPIO interrupt trigger mode register	8h00
R8_GPIO_INT_POLAR	0x1F	GPIO interrupt polarity register	8h00

R32_PA_DIR	0x40	PA port direction setting register	32h0000 0000
R32_PA_PIN	0x44	PA port data input register	32hxxxx xxxx
R32_PA_OUT	0x48	PA port data output register	32h0000 0000
R32_PA_CLR	0x4C	PA port output clear register	32h0000 0000
R32_PA_PU	0x50	PA port pull-up enable register	32h0000 0000
R32_PA_PD	0x54	PA port open-drain output and input pull-down configuration register	32h0000 0000
R32_PA_DRV	0x58	PA port drive capability configuration register	32h0000 0000
R32_PA_SMT	0x5C	PA port Schmitt trigger enable configuration register	32h0000 0000
R32_PB_DIR	0x60	PB port direction setting register	32h0000 0000
R32_PB_PIN	0x64	PB port data input register	32hxxxx 8000
R32_PB_OUT	0x68	PB port data output register	32h0000 0000
R32_PB_CLR	0x6C	PB port output clear register	32h0000 0000
R32_PB_PU	0x70	PB port pull-up configuration register	32h0000 0000
R32_PB_PD	0x74	PB port open-drain output and input pull-down configuration register	32h0000 0000
R32_PB_DRV	0x78	PB port drive capability configuration register	32h0000 0000
R32_PB_SMT	0x7C	PB port Schmitt trigger enable configuration register	32h0000 0000
R8_PORT_PIN	0x12	Multiplexing and remapping configuration register	8h00

GPIO interrupt flag register (R8_GPIO_INT_STATUS)

Bit	Name	Access	Description	Reset value
7	RB_GPIO_PB10_IS	RW1	PB10 pin interrupt flag bit, cleared by writing 1: 1: An interrupt is generated; 0: No interrupt is generated.	0
6	RB_GPIO_PB4_IS	RW1	PB4 pin interrupt flag bit, cleared by writing 1: 1: An interrupt is generated; 0: No interrupt is generated.	0
5	RB_GPIO_PA12_IS	RW1	PA12 pin interrupt flag bit, cleared by writing 1: 1: An interrupt is generated; 0: No interrupt is generated.	0
4	RB_GPIO_PA11_IS	RW1	PA11 pin interrupt flag bit, cleared by writing 1: 1: An interrupt is generated; 0: No interrupt is generated.	0
3	RB_GPIO_PA10_IS	RW1	PA10 pin interrupt flag bit, cleared by writing 1: 1: An interrupt is generated; 0: No interrupt is generated.	0
2	RB_GPIO_PA6_IS	RW1	PA6 pin interrupt flag bit, cleared by writing 1: 1: An interrupt is generated; 0: No interrupt is generated.	0
1	RB_GPIO_PA4_IS	RW1	PA4 pin interrupt flag bit, cleared by writing 1: 1: An interrupt is generated;	0

			0: No interrupt is generated.	
0	RB_GPIO_PA3_IS	RW1	PA3 pin interrupt flag bit, clear by writing 1: 1: An interrupt is generated; 0: No interrupt is generated.	0

GPIO interrupt enable register (R8_GPIO_INT_ENABLE)

Bit	Name	Access	Description	Reset value
7	RB_GPIO_PB10_IE	RW	PB10 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
6	RB_GPIO_PB4_IE	RW	PB4 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
5	RB_GPIO_PA12_IE	RW	PA12 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
4	RB_GPIO_PA11_IE	RW	PA11 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	RB_GPIO_PA10_IE	RW	PA10 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	RB_GPIO_PA6_IE	RW	PA6 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	RB_GPIO_PA4_IE	RW	PA4 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	RB_GPIO_PA3_IE	RW	PA3 pin interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

GPIO interrupt trigger mode register (R8_GPIO_INT_MODE)

Bit	Name	Access	Description	Reset value
7	RB_GPIO_PB10_IM	RW	PB10 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
6	RB_GPIO_PB4_IM	RW	PB4 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
5	RB_GPIO_PA12_IM	RW	PA12 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
4	RB_GPIO_PA11_IM	RW	PA11 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
3	RB_GPIO_PA10_IM	RW	PA10 pin interrupt mode selection bit:	0

			1: Edge trigger; 0: Level trigger.	
2	RB_GPIO_PA6_IM	RW	PA6 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
1	RB_GPIO_PA4_IM	RW	PA4 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0
0	RB_GPIO_PA3_IM	RW	PA3 pin interrupt mode selection bit: 1: Edge trigger; 0: Level trigger.	0

GPIO interrupt polarity register (R8_GPIO_INT_POLAR)

Bit	Name	Access	Description	Reset value
7	RB_GPIO_PB10_IP	RW	PB10 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
6	RB_GPIO_PB4_IP	RW	PB4 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
5	RB_GPIO_PA12_IP	RW	PA12 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
4	RB_GPIO_PA11_IP	RW	PA11 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
3	RB_GPIO_PA10_IP	RW	PA10 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
2	RB_GPIO_PA6_IP	RW	PA6 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
1	RB_GPIO_PA4_IP	RW	PA4 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0
0	RB_GPIO_PA3_IP	RW	PA3 pin interrupt polarity selection bit: 1: High level/rising edge; 0: Low level/falling edge.	0

PA port direction setting register (R32_PA_DIR)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_DIR	RW	Current input/output direction control of PA pin: 1: The pin direction is output mode; 0: The pin direction is input mode.	0

PA port input data register (R32_PA_PIN)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_PIN	RO	PA pin level status: 1: The pin inputs high level; 0: The pin inputs low level; The value of this bit is valid only when the corresponding bit of the direction register (R32_PA_DIR) is 0.	0

PA port output data register (R32_PA_OUT)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_OUT	RW	PA pin output level status: 1: The pin outputs high level; 0: The pin outputs low level; The value of this bit is valid only when the corresponding bit of the direction register (R32_PA_DIR) is 1.	0

PA port bit clear register (R32_PA_CLR)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_CLR	WZ	PA hold/clear data output control: 1: The pin outputs low level; 0: No effect.	0

PA port pull-up configuration register (R32_PA_PU)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_PU	RW	PA pin pull-up function enable control: 1: Enable pin pull-up function; 0: Disable pin pull-up function.	0

PA port open-drain output and input pull-down configuration register (R32_PA_PD)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_PD	RW	When the corresponding bit of direction register (R32_PA_DIR) is configured as 1 (ie output mode): 1: Enable open-drain output function of this pin; 1: Disable open-drain output function of this pin. When the corresponding bit of direction register	0

			(R32_PA_DIR) is configured as 0 (ie input mode): 1: Enable the pull-down function of this pin; 1: Disable the pull-down function of this pin;	

PA port drive capability configuration register (R32_PA_DRV)

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_DRV	RW	PA pin output drive capability control: 1: The maximum drive current is 16mA; 0: The maximum drive current is 8mA.	0

PA port Schmitt trigger enable configuration register (R32_PA_SMT):

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	R32_PA_SMT	RW	PA pin Schmitt trigger function control: 1: Enable the Schmitt trigger input function or low slope output function of this pin; 0: Disable the Schmitt trigger input function or low slope output function of this pin.	0

PB port direction setting register (R32_PB_DIR)

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	R32_PB_DIR	RW	Current input/output direction control of PB pin: 1: The pin direction is output mode; 0: The pin direction is input mode.	0

PB port input data register (R32_PB_PIN)

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	R32_PB_PIN	RO	Current level status of PB pin: 1: The pin inputs high level; 0: The pin inputs low level. The value of this bit is valid only when the corresponding bit of the direction register (R32_PB_DIR) is 0.	0

PB port output data register (R32_PB_OUT)

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	R32_PB_OUT	RW	PB pin output level status: 1: The pin outputs high level;	0

			1: The pin outputs low level. The value of this bit is valid only when the corresponding bit of the direction register (R32_PB_DIR) is 1.	

PB port bit clear register (R32_PB_CLR)

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	R32_PB_CLR	WZ	PB hold/clear data output control: 1: The pin outputs low level; 0: No effect.	0

PB port pull-up configuration register (R32_PB_PU)

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	R32_PB_PU	RW	PB pin pull-up function enable control: 1: Enable pin pull-up function; 0: Disable pin pull-up function.	0

PB port open-drain output and input pull-down configuration register (R32_PB_PD)

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	R32_PB_PD	RW	When the corresponding bit of direction register (R32_PB_DIR) is configured as 1 (ie output mode): 1: Enable open-drain output function of this pin; 1: Disable open-drain output function of this pin; When the corresponding bit of direction register (R32_PB_DIR) is configured as 0 (ie input mode): 1: Enable pull-down function of this pin; 1: Disable pull-down function of this pin; If the corresponding bit of pull-up configuration register (R32_PB_PU) is also configured as 1, the input status weak hold function is enabled.	0

PB port drive capability configuration register (R32_PB_DRV)

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	R32_PB_DRV	RW	PB pin output drive capability control: 1: The maximum drive current is 16mA; 0: The maximum drive current is 8mA.	0

PB port Schmitt trigger enable configuration register (R32_PB_SMT):

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	R32_PB_SMT	RW	PB pin Schmitt trigger function control: 1: Enable the Schmitt trigger input function or low slope output function of this pin; 0: Disable the Schmitt trigger input function or low slope output function of this pin.	0

Multiplexing and remapping configuration register (R8_PORT_PIN)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved.	0
5	RB_PIN_UART1	RW	UART1 remapping configuration bit: 1: RXD1/TXD1 to PB8/PB9 pin; 0: RXD1/TXD1 to PA8/PA9 pin.	0
4	RB_PIN_UART0	RW	UART0 remapping configuration bit: 1: RXD0/TXD0 to PA15/PA14 pin; 0: RXD0/TXD0 to PB4/PB7 pin.	0
3	Reserved	RO	Reserved.	0
2	RB_PIN_TMR2	RW	TIMER2 remapping configuration bit: 1: TMR2/PWM6/CAP2 to PB11 pin; 0: TMR2/PWM6/CAP2 to PA11 pin.	0
1	RB_PIN_TMR1	RW	TIMER1 remapping configuration bit: 1: TMR1/PWM5/CAP1 to PB2 pin; 0: TMR1/PWM5/CAP1 to PA10 pin.	0
0	Reserved	RO	Reserved.	0

5.3 GPIO Multiplexing and Remapping

5.3.1 Multiplexing Function

Some I/O pins of chip have the function of multiplexing. After power on, all I/O pins have common I/O functions by default. After enabling different function modules, the corresponding pins are configured as corresponding function pins of each function module.

If a pin is multiplexed with multiple functions, and multiple functions are enabled, please refer to the function order in the "Multiplexing Function and Mapping" list in section 1.2 for the priority order of multiplexing function.

For example: If PA0 pin is multiplexed as /SCK1/LED0/CMD1, the clock function of SPI1 has priority, and the CMD1 function of SD1 controller has the lowest priority. In this way, among multiple multiplexing functions, the multiplexing functions with the relatively higher priority of the pin can be enabled while those with the lowest priority do not need to be used.

The following tables list the I/O pins used by each function module.

Table 5-2 Serial Peripheral Interface (SPI0)

Pin	GPIO	Function Description
SCS	PA12	Chip selection input pin of SPI0 slave
SCK0	PA13	SPI0 serial clock pin, host output/slave input

MOSI0	PA14	SPI0 serial data pin, host output/slave input
MISO0	PA15	SPI0 serial data pin, host input/slave output

Table 5-3 Serial Peripheral Interface (SPI1)

Pin	GPIO	Function Description
SCK1	PA0	SPI1 serial clock output pin
MOSI1	PA1	SPI1 serial data output pin (only host function)
MISO1	PA2	SPI1 serial data input pin (only host function)

Table 5-4 Universal Asynchronous Receiver-Transmitter (UART0)

Pin	GPIO	Function Description
RXD0	PB4	UART0 receiver input pin
TXD0	PB7	UART0 transmitter output pin
RXD0_	PA15	RXD pin function mapping of UART0
TXD0_	PA14	TXD pin function mapping of UART0
DTR	PB5	MODEM signal of UART0; data terminal is ready
RTS	PB6	MODEM signal of UART0; request to send
CTS	PB0	MODEM signal of UART0; clear to send
DSR	PB1	MODEM signal of UART0; data device is ready
RI	PB2	MODEM signal of UART0; ring indicator
DCD	PB3	MODEM signal of UART0; carrier detection

Table 5-5 Universal Asynchronous Receiver-Transmitter (UART1-3)

Pin	GPIO	Function Description
RXD1	PA8	UART1 receiver input pin
TXD1	PA9	UART1 transmitter output pin
RXD1_	PB8	RXD pin function mapping of UART1
TXD1_	PB9	TXD pin function mapping of UART1
RXD2	PA6	UART2 receiver input pin
TXD2	PA7	UART2 transmitter output pin
RXD3	PA4	UART3 receiver input pin
TXD3	PA5	UART3 transmitter output pin

Table 5-6 SD Controller (SD0-3)

Pin	GPIO	Function Description
SDCK	PA6	SD0-3 clock signal pin
CMD0	PA7	SD0 command signal pin
SD00	PA8	SD0 data signal 0 pin
SD01	PA9	SD0 data signal 1 pin
SD02	PA10	SD0 data signal 2 pin
SD03	PA11	SD0 data signal 3 pin
CMD1	PA0	SD1 command signal pin
SD10	PA12	SD1 data signal 0 pin
SD11	PA13	SD1 data signal 1 pin
SD12	PA14	SD1 data signal 2 pin

SD13	PA15	SD1 data signal 3 pin
CMD2	PA1	SD2 command signal pin
SD20	PB0	SD2 data signal 0 pin
SD21	PB1	SD2 data signal 1 pin
SD22	PB2	SD2 data signal 2 pin
SD23	PB3	SD2 data signal 3 pin
CMD3	PA2	SD3 command signal pin
SD30	PB4	SD3 data signal 0 pin
SD31	PB5	SD3 data signal 1 pin
SD32	PB6	SD3 data signal 2 pin
SD33	PB7	SD3 data signal 3 pin

Table 5-7 LED Control Card

Pin	GPIO	Function Description
LED0	PA0	LED serial data 0 pin
LED1	PA1	LED serial data 1 pin
LED2	PA2	LED serial data 2 pin
LED3	PA3	LED serial data 3 pin
LEDC	PA4	LED serial clock pin

Table 5-8 ISP Download (ISP)

Pin	GPIO	Function Description
SCS	PA12	ISP download chip select input pin
SCK	PA13	ISP download clock input pin
MOSI	PA14	ISP download data input pin
MISO	PA15	ISP download data output pin
RST	RST#	ISP download reset input pin

5.3.2 Remapping

In order to optimize the peripheral number of the chip package, some multiplexing functions can be remapped to other pins. Remapping of the pin can be realized by means of setting the multiplexing and mapping register R8_PORT_PIN.

CH567 supports remapping of UART0, TIMER1 and TIMER2 peripheral pins, please refer to the following table for details:

Table 5-9 Remapping Pin

Peripheral Function	Default Pin	Remapping Pin
UART0	PB4/PB7	PA15/PA14
UART1	PA8/PA9	PB8/PB9
TIMER1/PWM5	PA10	PB2
TIMER2/PWM6	PA11	PB11

Chapter 6 Serial Peripheral Interface (SPI)

6.1 Introduction to SPI

SPI is a full-duplex serial interface. It can handle multiple masters and slaves connected to the specified bus. During data communication, there can only be one master and one slave for communication on the bus. Usually SPI interface consists of 4 pins: SPI chip selection pin (SCS), SPI clock pin (SCK), SPI serial data pin MISO (master input/slave output pin) and SPI serial data pin MOSI (master output/slave input pin).

CH567 chip has 2 SPI interfaces, and their respective characteristics are as follows:

SPI0 features:

(1). Support master mode and device mode;
(2). Compatible with Serial Peripheral Interface (SPI) specification;
(3). Support the data transmission mode 0 and mode 3;
(4). 8-bit data transmission mode;
(5). The clock frequency is close to half of Fsys;
(6). 8-byte FIFO;
(7). The device mode supports the first byte as command mode or data stream mode;
(8). Support DMA data transmission.

SPI1 features:

(1). Only support master mode;
(2). Support the data transmission mode 0 and mode 3;
(3). 8-bit data transmission mode;
(4). The maximum clock frequency is close to half of Fsys;
(5). 8-byte FIFO.

6.2 Register Description

SPI0 related register physical start address: 0x0040 4000
SPI1 related register physical start address: 0X0040 4400

Table 6-1 List of SPI0 Related Registers

Name	Offset address	Description	Reset value
R8_SPI0_CTRL_MOD	0x00	SPI0 mode configuration register	8h00
R8_SPI0_CTRL_CFG	0x01	SPI0 configuration register	8h00
R8_SPI0_INTER_EN	0x02	SPI0 interrupt enable register	8h00
R8_SPI0_CLOCK_DIV R8_SPI0_SLAVE_PRE	0x03	SPI0 clock frequency division register in master mode SPI0 preset data register in device mode	8h10
R8_SPI0_BUFFER	0x04	SPI0 data buffer	8hxx
R8_SPI0_RUN_FLAG	0x05	SPI0 working status register	8h00
R8_SPI0_INT_FLAG	0x06	SPI0 interrupt flag register	8h00
R8_SPI0_FIFO_COUNT	0x07	SPI0 transceiver FIFO counter register	8hxx
R16_SPI0_TOTAL_CNT	0x0C	SPI0 transceiver data length register	16hxxxx
R8_SPI0_FIFO	0x10	SPI0 FIFO register	8hxx
R8_SPI0_FIFO_COUNT1	0x13	SPI0 transceiver FIFO counter register	8hxx
R16_SPI0_DMA_NOW	0x14	Current address of SPI0 DMA buffer	16hxxxx
R16_SPI0_DMA_BEG	0x18	Start address of SPI0 DMA buffer	16hxxxx

R16_SPI0_DMA_END	0x1C	End address of SPI0 DMA buffer	16hxxxx

Table 6-2 List of SPI1 Related Registers

Name	Offset address	Description	Reset value
R8_SPI1_CTRL_MOD	0x00	SPI1 mode configuration register	8h00
R8_SPI1_CTRL_CFG	0x01	SPI1 configuration register	8h00
R8_SPI1_INTER_EN	0x02	SPI1 interrupt enable register	8h00
R8_SPI1_CLOCK_DIV	0x03	SPI1 clock frequency division register in master mode	8hxx
R8_SPI1_BUFFER	0x04	SPI1 data buffer	8hxx
R8_SPI1_RUN_FLAG	0x05	SPI1 working status register	8h00
R8_SPI1_INT_FLAG	0x06	SPI1 interrupt flag register	8h00
R8_SPI1_FIFO_COUNT	0x07	SPI1 transceiver FIFO counter register	8hxx
R16_SPI1_TOTAL_CNT	0x0C	SPI1 transceiver data length register	16hxxxx
R8_SPI1_FIFO	0x10	SPI1 FIFO register	8hxx
R8_SPI1_FIFO_COUNT1	0x13	SPI1 transceiver FIFO counter register	8hxx

SPI mode configuration register (R8_SPIx_CTRL_MOD) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_SPI_MISO_OE	RW	MISO pin output enable bit (can be used for data line switching direction in 2-wire mode): 1: MISO pin output enabled; 0: MISO pin output disabled.	0
6	RB_SPI_MOSI_OE	RW	MOSI pin output enable bit: 1: MOSI pin output enabled; 0: MOSI pin output disabled.	0
5	RB_SPI_SCK_OE	RW	SCK pin output enable bit 1: SCK pin output enabled; 0: SCK pin output disabled.	0
4	RB_SPI_FIFO_DIR	RW	FIFO direction setting bit: 1: Input mode (read data in host mode); 0: Output mode (write data in host mode).	0
3	RB_SPI_SLV_CMD_MOD	RW	SPI0 device mode first byte configuration bit, only for SPI0: 1: First byte command mode; 0: Data stream mode. In the first byte command mode, it will be regarded as a command code when receiving the first-byte data after valid SPI chip selection, and RB_SPI_IF_FST_BYTE bit of interrupt flag register will be set to 1, This bit is only valid in device mode.	0
3	RB_SPI_MST_SCK_MOD	RW	Host clock sampling mode configuration bit: 1: Mode 3 (SCK is at high level in idle mode); 0: Mode 0 (SCK is at low level in idle mode). This bit is only valid in host mode.	0
2	RB_SPI_2WIRE_MOD	RW	2-wire or 3-wire SPI mode configuration bit,	0

			only for SPI0, SPI1 does not need this bit: 1: 2-wire mode (SCK, MISO); 0: 3-wire mode (SCK, MOSI, MISO).	
1	RB_SPI_ALL_CLEAR	RW	FIFO register and counter register clear bit: 1: Force to clear; 0: Not clear.	1
0	RB_SPI_MODE_SLAVE	RW	SPI0 master/slave mode selection bit, only for SPI0 1: Device mode; 0: Host mode. Note: SPI1 does not support device mode.	0

SPI configuration register (R8_SPIx_CTRL_CFG) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved.	0
5	RB_SPI_BIT_ORDER	RW	SPI data bit order selection bit: 1: Low byte is in front; 0: High byte is in front.	0
4	RB_SPI_AUTO_IF	RW	Enable the function of automatically clearing flag bit RB_SPI_IF_BYTE_END when accessing BUFFER/FIFO: 1: Enable; 0: Disable.	0
3	Reserved	RO	Reserved.	0
2	RB_SPI_DMA_LOOP	RW	SPI0 DMA address loop enable bit: 1: Enable DMA address loop function; 0: Disable DMA address loop function. Note: Not supported by SPI1.	0
1	Reserved	RO	Reserved.	0
0	RB_SPI_DMA_ENABLE	RW	SPI0 DMA enable/disable bit, only supported by SPI0: 1: Enable DMA; 0: Disable DMA.	0

Note: If the DMA address loop mode function is enabled, when the DMA address is added to the set end address, it will automatically loop to the set first address, without re-setting the DMA start address register (R16_SPI0_DMA_BEG) and DMA end address register (R16_SPI0_DMA_END).

SPI interrupt enable register (R8_SPIx_INTER_EN) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_SPI_IE_FST_BYTE	RW	The first byte receiving interrupt enable bit in slave mode, only supported by SPI0: 1: Enable receiving the first byte interrupt; 0: Disable receiving the first byte interrupt. To enable this function, you need to set SPI to device mode, and meanwhile it is required to set RB_SPI_SLV_CMD_MOD bit to 1, thus entering the first byte command mode.	0

[6:5]	Reserved	RO	Reserved.	0
4	RB_SPI_IE_FIFO_OV	RW	FIFO overflow interrupt enable bit, only supported by SPI0: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	RB_SPI_IE_DMA_END	RW	DMA end interrupt enable bit, only supported by SPI0: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	RB_SPI_IE_FIFO_HF	RW	More than half FIFO interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	RB_SPI_IE_BYTE_END	RW	SPI single byte transmission completion interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	RB_SPI_IE_CNT_END	RW	SPI all byte transmission completion interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

SPI master mode clock divider register (R8_SPIx_CLOCK_DIV) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPI_CLOCK_DIV	RW	Frequency division factor in master mode, the minimum value of is 2 SPI clock frequency = main frequency/frequency division factor.	10h

Preset data register in SPI slave mode (R8_SPI0_SLAVE_PRE)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPI0_SLAVE_PRE	RW	Preset return data in SPI0 slave mode. Used to receive the return data after first byte of data. Note: Not supported by SPI1	10h

SPI data buffer (R8_SPIx_BUFFER) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPIx_BUFFER	RW	SPI data transmit and receive buffer	xx

SPI working status register (R8_SPIx_RUN_FLAG) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_SPI_SLV_SELECT	RO	SPI0 slave mode selection status bit: 1: Device mode; 0: Host mode.	0

			Note: Not supported by SPI1	
6	RB_SPI_SLV_CS_LOAD	RO	After SPI0 slave mode chip selection, first load status bit: 1: Loading is completed; 0: Not completed (the preload value can be modified). Note: Not supported by SPI1	0
5	RB_SPI_FIFO_READY	RO	FIFO ready status bit: 1: FIFO is ready; 0: FIFO is not ready.	0
4	RB_SPI_SLV_CMD_ACT	RO	Command receive completion status bit in SPI0 slave mode, namely to completing the exchange of first-byte data: 1: The first byte exchange is completed; 1: The first byte exchange is not completed. Note: Not supported by SPI1	0
[3:0]	Reserved	RO	Reserved.	0

SPI interrupt flag register (R8_SPIx_INT_FLAG) (x=0/1)

Bit	Name	Access	Description	Reset value
7	RB_SPI_IF_FST_BYTE	RW1	First byte receive flag bit in SPI0 device mode: 1: The first byte is received; 0: The first byte is not received. Note: Not supported by SPI1.	0
6	RB_SPI_FREE	RO	Current SPI idle status bit: 1: SPI is currently in idle state; 0: SPI is currently in non-idle state.	0
5	Reserved	RO	Reserved.	0
4	RB_SPI_IF_FIFO_OV	RW1	SPI0 FIFO overflow flag bit, only supported by SPI0: 1: FIFO overflow; 0: FIFO not overflow. Note: Not supported by SPI1.	0
3	RB_SPI_IF_DMA_END	RW1	SPI0 DMA end flag bit, only supported by SPI0: 1: DMA transmission is ended; 0: DMA transmission is not ended. Note: Not supported by SPI1.	0
2	RB_SPI_IF_FIFO_HF	RW1	FIFO data more than half flag bit: 1: Data reaches half of the FIFO buffer; 0: Data does not reach half of the FIFO buffer; Note: If RB_SPI_FIFO_DIR=1, receive data, trigger when FIFO count >= 4; If RB_SPI_FIFO_DIR=0, transmit data, trigger when FIFO count <4.	0
1	RB_SPI_IF_BYTE_END	RW1	SPI single byte transmission completion flag bit:	0

			1: SPI single byte transmission is completed; 0: SPI single byte transmission is not completed.	
0	RB_SPI_IF_CNT_END	RW1	SPI all byte transmission completion flag bit: 1: SPI all byte transmission is completed; 0: SPI all byte transmission is not completed.	0

SPI transceiver FIFO count register (R8_SPIx_FIFO_COUNT) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPIx_FIFO_COUNT	RW	Count of bytes in the current FIFO.	xx

SPI transceiver FIFO count register (R8_SPIx_FIFO_COUNT1) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPIx_FIFO_COUNT1	RW	Count of bytes in the current FIFO, it is equivalent to the register R8_SPIx_FIFO_COUNT.	xx

SPI transceiver data total length register (R16_SPIx_TOTAL_CNT) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	R16_SPIx_TOTAL_CNT	RW	The total number of bytes of SPI data receiver and transmit, and the lower 12 bits are valid. At most 4095 bytes can be transmitted at a time when DMA is used.	0

SPI FIFO register (R8_SPIx_FIFO) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:0]	R8_SPIx_FIFO	RO/ WO	SPI FIFO register. The FIFO size is 8 bytes.	0

Register R8_SPIx_BUFFER and register R8_SPIx_FIFO are SPI data related registers. The main difference is: after the latter reads one byte of data, as it reads from FIFO, the value of length register (R16_SPI_TOTAL_CNT) is automatically reduced by 1; after the former reads one byte, the value of length register remains unchanged.

Current address of SPI0 DMA buffer (R16_SPI0_DMA_NOW)

Bit	Name	Access	Description	Reset value
[15:0]	R16_SPI0_DMA_NOW	RW	Current address of DMA buffer, only supported by SPI0. The DMA operation can be judged by querying this value.	xxxx

Start address of SPI0 DMA buffer (R16_SPI0_DMA_BEG)

Bit	Name	Access	Description	Reset

				value
[15:0]	R16_SPI0_DMA_BEG	RW	Start address of DMA buffer, only supported by SPI0. Point to the start address of SPI0 receive and transmit data buffer.	xxxx

End address of SPI0 DMA buffer (R16_SPI0_DMA_END)

Bit	Name	Access	Description	Reset value
[15:0]	R16_SPI0_DMA_END	RW	End address of DMA buffer, only supported by SPI0. Point to the end address of SPI0 receive and transmit data buffer.	xxxx

6.3 SPI Transmission Format

SPI supports two transmission formats, mode 0 and mode 3, which can be selected by setting RB_SPI_MST_SCK_MOD bit of SPI mode configuration register (R8_SPIx_CTRL_MOD).

The data transmission format is shown in the figure below:

Mode 0: RB_SPI_MST_SCK_MOD = 0

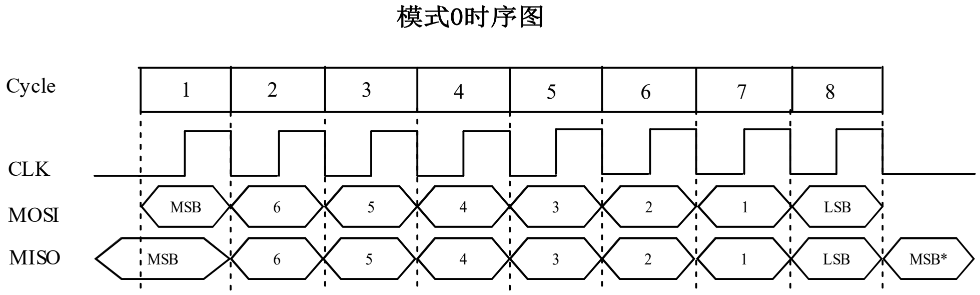


Figure 6-1 Transmission Format of SPI Mode 0

Mode 3: RB_SPI_MST_SCK_MOD = 1

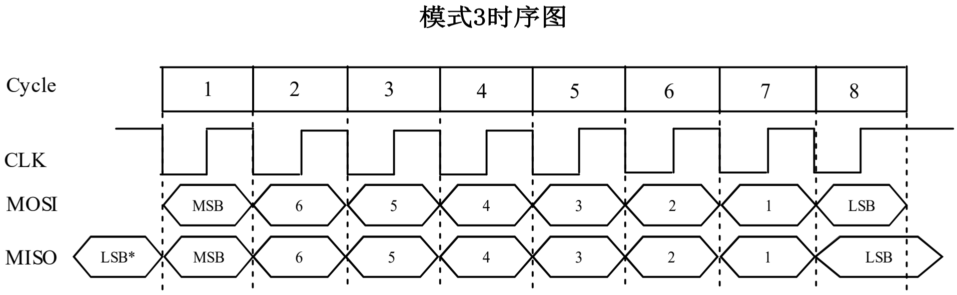


Figure 6-2 Transmission Format of SPI Mode 3

6.4 SPI Configuration

6.4.1 SPI Master Mode

In SPI master mode, serial clock is generated on SCK pin, and the chip selection pin can be specified as any I/O pin.

Configuration Steps:

(1). Set SPI master mode clock divider register (R8_SPIx_CLOCK_DIV), to configure SPI clock speed;

(2). Set the RB_SPI_MODE_SLAVE bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 0, to configure SPI as the master mode;

(3). Set the RB_SPI_SLV_CMD_MOD bit of SPI mode configuration register (R8_SPIx_CTRL_MOD), and set it to mode 0 or mode 3 according to the requirements of the connected device;

(4). Set the RB_SPI_FIFO_DIR bit of SPI mode configuration register (R8_SPIx_CTRL_MOD), to configure the FIFO direction. If it is 1, the current FIFO direction is data input; if it is 0, the current FIFO direction is data output.

(5). Set the RB_SPI_MOSI_OE bit and RB_SPI_SCK_OE bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 1, and set the RB_SPI_MISO_OE bit to 0, and set the bits corresponding to the MOSI pin and SCK pin in the PA port direction register (R32_PB_DIR) to 1, the bit corresponding to MISO pin to 0, to configure the MOSI pin and SCK pin direction as output, and configure the MISO pin direction as input;

Data transmit process:

(1). Set the RB_SPI_FIFO_DIR bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 0, to configure the current FIFO direction as output;

(2). Write R16_SPIx_TOTAL_CNT register, to set the length of the data to be transmitted;

(3). Write R8_SPIx_FIFO register, write the data to be transmitted to FIFO. If R8_SPI0_FIFO_COUNT is less than FIFO size, data can continue to be written to FIFO;

(4). After all data is written to FIFO, wait until R16_SPIx_TOTAL_CNT register becomes 0, it indicates that the data transmission is completed. If only one-byte data is transmitted, you can also wait until R8_SPI0_FIFO_COUNT becomes 0, it indicates that there is no data in FIFO and the data transmission is completed.

Data receive process:

(1). Set the RB_SPI_FIFO_DIR bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 1, to configure the current FIFO direction as input;

(2). Write R16_SPIx_TOTAL_CNT register, to set the length of the data to be received;

(3). Wait until R8_SPIx_FIFO_COUNT register is not 0, it indicates that the returned data is received;

(4). The value read in R8_SPIx_FIFO is the received data.

6.4.2 SPI Slave Mode

Only SPI0 supports the slave mode. In the slave mode, SCK pin is used to receive the serial clock of the connected SPI host.

Configuration Steps:

(1). Set the RB_SPI_MODE_SLAVE bit of SPI0 mode configuration register (R8_SPI0_CTRL_MOD) to 1, to configure SPI0 as the slave mode;

(2). Set the RB_SPI_SLV_CMD_MOD bit of SPI0 mode configuration register (R8_SPI0_CTRL_MOD) as required;

(3). Set the RB_SPI_FIFO_DIR bit of SPI0 mode configuration register (R8_SPI0_CTRL_MOD), to configure the FIFO direction. If it is 1, the current FIFO direction is data input. If it is 0, the current FIFO direction is data output.

(4). Set the RB_SPI_MOSI_OE bit and RB_SPI_SCK_OE bit of SPI0 mode configuration register (R8_SPI0_CTRL_MOD) to 0, set the RB_SPI_MISO_OE bit to 1, and set the bits corresponding to the MOSI pin, SCK pin and SCS pin in the PA port direction register (R32_PB_DIR) to 0, set the bit corresponding to MISO pin to 1, to configure the MOSI pin, SCK pin and SCS pin direction as input, and configure the MISO pin direction as output. If the MISO pin is not configured as output, when chip selection is valid (low level), MISO will automatically enable output. It is recommended to set the MISO pin as input so that MISO does not output when chip selection is invalid, so that SPI bus can be shared when multiple devices work. Note: In SPI slave mode, I/O pin direction of MISO can be configured as output by the MISO pin direction, it can also be configured as output automatically during the valid SPI chip selection, but its output data is selected by RB_SPI_MISO_OE. When RB_SPI_MISO_OE is 1, output SPI data. When RB_SPI_MISO_OE is 0, output data of GPIO register.

(5). Optionally, set the preset data register in SPI0 slave mode (R8_SPI0_SLAVE_PRE), to be automatically loaded into the buffer for the first time after chip selection for external output. After 8 clocks (that is, the first byte of data exchange between the host and the slave is completed), the controller will obtain the first byte of data (command code) sent by the external SPI host, and the external SPI host obtains the preset data in R8_SPI0_SLAVE_PRE (status value) through exchange. The bit 7 of this register will be automatically loaded into the MISO pin during low-level SCK after the SPI chip selection is effective. For SPI mode 0 (CLK defaults to low level), if the bit 7 of R8_SPI0_SLAVE_PRE is preset, the external SPI master will obtain the preset value of bit 7 of R8_SPI0_SLAVE_PRE by inquiring the MISO pins when the SPI chip selection is effective but has no data transmission, thereby the value of bit 7 of R8_SPI0_SLAVE_PRE can be obtained only by validating SPI chip selection.

Data transmit process:

(1). Set the RB_SPI_FIFO_DIR bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 0, to configure FIFO direction as data output;

(2) Write the data to be transmitted to FIFO register (R8_SPIx_FIFO), and add 1 to the SPI transmit/receive data total length register (R16_SPIx_TOTAL_CNT). It is recommended to set R16_SPIx_TOTAL_CNT to a larger value directly. In this way, once there is data in FIFO, it will be transmitted automatically, and it will be automatically suspended when the FIFO is empty. It is not necessary to set R16_SPIx_TOTAL_CNT every time;

(3). If a single byte is sent, wait for the R16_SPIx_TOTAL_CNT register to be 0, and wait for all data to be sent. If multiple bytes are sent, you can write up to 8 data to the FIFO register (R8_SPIx_FIFO) at a time, and then wait for completing sending;

Data receive process:

(1). Set the RB_SPI_FIFO_DIR bit of SPI mode configuration register (R8_SPIx_CTRL_MOD) to 1, to configure the current FIFO direction as data input;

(2) Wait to query the SPI transmit/receive data total length register (R16_SPIx_TOTAL_CNT). If the register is not 0, data is received, and the received data can be obtained by reading the FIFO register (R8_SPIx_FIFO).

For receiving the data of single byte, FIFO is not required, and SPI data buffer register (R8_SPIx_BUFFER) can be read directly to obtain the current data given by the other party for starting transmission.

6.5 DMA Function

For CH567, only SPI0 has DMA function, but SPI1 does not have this function. By enabling the DMA function, receiving and transmiting SPI data can be realized more conveniently on the basis of reducing software intervention.

6.5.1 DMA Transmiting Data Configuration in SPI Master Mode

(1). According to Section 6.4.1, configure SPI0 as master mode;

(2). If it is needed to generate a DMA completion interrupt, set the RB_SPI_IE_DMA_END bit of SPI interrupt enable register (R8_SPIx_INTER_EN) to 1;

(3). Initialize the R16_SPI_DMA_BEG register as the start address of SPI data transmit buffer;

(4). Initialize the R16_SPI_DMA_END register as the end address of SPI data transmit buffer;

(5). Clear SPI interrupt status register (R8_SPIx_INT_FLAG);

(6). Initialize the R16_SPI_TOTAL_CNT register to the number of data to be sent, and start the transmission if DMA is enabled;

(7). If it is needed to enable the DMA address loop mode function, set the RB_SPI_DMA_LOOP bit of SPI DMA control register (R8_SPIx_CTRL_DMA) to 1;

(8). Set the RB_SPI_DMA_ENABLE bit of DMA control register (R8_SPIx_CTRL_DMA) of SPI to 1, to enable DMA to transmit data.

6.5.2 DMA Receiving Data Configuration in SPI Master Mode

(1). According to Section 6.4.1, configure SPI0 as master mode;

(2). If it is needed to generate DMA interrupt, set the RB_SPI_IE_DMA_END bit of register R8_SPI_INTER_EN to 1, to enable generating the DMA end interrupt;

(3). Initialize the R16_SPI_DMA_BEG register as the start address of SPI data receive buffer;

(4). Initialize the R16_SPI_DMA_END register as the end address of SPI data receive buffer;

(5). Clear SPI interrupt status register (R8_SPIx_INT_FLAG);

(6). Initialize the R16_SPI_TOTAL_CNT register to the number of data to be received, and start the transmission if DMA is enabled;

(7). If it is needed to enable the DMA address loop mode function, set the RB_SPI_DMA_LOOP bit of SPI DMA control register (R8_SPIx_CTRL_DMA) to 1;

(8). Set the RB_SPI_DMA_ENABLE bit of DMA control register (R8_SPIx_CTRL_DMA) of SPI to 1, and enable DMA to receive data. If DMA is enabled first, transmission will start automatically after R16_SPI_TOTAL_CNT is set.

Chapter 7 Universal Asynchronous Receiver-Transmitter (UART)

7.1 Introduction to UART

CH567 is equipped with 4 sets of full-duplex asynchronous serial ports, UART0/1/2/3, and they support full-duplex and half-duplex serial communication. Among them, UART0 is equipped with a transmit status pin for switching RS485, and supports MODEM signals CTS, DSR, RI, DCD, DTR and RTS.

UART features:

(1). Compatible with 16C550 asynchronous serial port and enhanced;

(2). Support 5, 6, 7 or 8 data bits and 1 or 2 stop bits;

(3). Support the parity check modes of odd, even, no check, blank 0 and flag 1;

(4). Programmable communication baud rate, support 115200bps and communication baud rate up to 6Mbps;

(5). Built-in 8-byte FIFO (first-in-first-out) buffer, support 4 FIFO trigger levels;

(6). UART0 supports MODEM modem signals CTS, DSR, RI, DCD, DTR and RTS, which can be converted to RS232 level;

(7). Support automatic handshake and automatic transmission rate control of hardware flow control signals CTS and RTS, compatible with TL16C550C;

(8). Support serial port frame error detection and Break line interval detection;

(9). Support full-duplex and half-duplex serial communication, and UART0 provides a transmit status pin for switching RS485;

7.2 Register Description

UART0 related register physical start address: 0x0040 3000

UART1 related register physical start address: 0x0040 3400

UART2 related register physical start address: 0x0040 3800

UART3 related register physical start address: 0x0040 3c00

Table 7-1 List of UART0 Related Registers

Name	Offset address	Description	Reset value
R8_UART0_MCR	0x00	MODEM control register	8h00
R8_UART0_IER	0x01	Interrupt enable register	8h00
R8_UART0_FCR	0x02	FIFO control register	8h00
R8_UART0_LCR	0x03	Line control register	8h00
R8_UART0_IIR	0x04	Interrupt recognition register	8h01
R8_UART0_LSR	0x05	Line status register	8hC0
R8_UART0_MSR	0x06	MODEM status register	8hx0
R8_UART0_RBR	0x08	Receive buffer register	8hxx
R8_UART0_THR	0x08	Transmit hold register	8hxx
R8_UART0_RFC	0x0A	Receiver FIFO count register	8hxx
R8_UART0_TFC	0x0B	Transmitter FIFO count register	8hxx
R16_UART0_DL	0x0C	Baud rate divisor latch	16hxxxx
R8_UART0_DIV	0x0E	Predivider divisor register	8hxx
R8_UART0_ADR	0x0F	Slave address register	8hFF

Table 7-2 List of UART1 Related Registers

Name	Offset address	Description	Reset value

R8_UART1_MCR	0x00	MODEM control register	8h00
R8_UART1_IER	0x01	Interrupt enable register	8h00
R8_UART1_FCR	0x02	FIFO control register	8h00
R8_UART1_LCR	0x03	Line control register	8h00
R8_UART1_IIR	0x04	Interrupt recognition register	8h01
R8_UART1_LSR	0x05	Line status register	8hC0
R8_UART1_RBR	0x08	Receive buffer register	8hxx
R8_UART1_THR	0x08	Transmit hold register	8hxx
R8_UART1_RFC	0x0A	Receiver FIFO count register	8hxx
R8_UART1_TFC	0x0B	Transmitter FIFO count register	8hxx
R16_UART1_DL	0x0C	Baud rate divisor latch	16hxxxx
R8_UART1_DIV	0x0E	Predivider divisor register	8hxx

Table 7-3 List of UART2 Related Registers

Name	Offset address	Description	Reset value
R8_UART2_MCR	0x00	MODEM control register	8h00
R8_UART2_IER	0x01	Interrupt enable register	8h00
R8_UART2_FCR	0x02	FIFO control register	8h00
R8_UART2_LCR	0x03	Line control register	8h00
R8_UART2_IIR	0x04	Interrupt recognition register	8h01
R8_UART2_LSR	0x05	Line status register	8hC0
R8_UART2_RBR	0x08	Receive buffer register	8hxx
R8_UART2_THR	0x08	Transmit hold register	8hxx
R8_UART2_RFC	0x0A	Receiver FIFO count register	8hxx
R8_UART2_TFC	0x0B	Transmitter FIFO count register	8hxx
R16_UART2_DL	0x0C	Baud rate divisor latch	16hxxxx
R8_UART2_DIV	0x0E	Predivider divisor register	8hxx

Table 7-4 List of UART3 Related Registers

Name	Offset address	Description	Reset value
R8_UART3_MCR	0x00	MODEM control register	8h00
R8_UART3_IER	0x01	Interrupt enable register	8h00
R8_UART3_FCR	0x02	FIFO control register	8h00
R8_UART3_LCR	0x03	Line control register	8h00
R8_UART3_IIR	0x04	Interrupt recognition register	8h01
R8_UART3_LSR	0x05	Line status register	8hC0
R8_UART3_RBR	0x08	Receive buffer register	8hxx
R8_UART3_THR	0x08	Transmit hold register	8hxx
R8_UART3_RFC	0x0A	Receiver FIFO count register	8hxx
R8_UART3_TFC	0x0B	Transmitter FIFO count register	8hxx
R16_UART3_DL	0x0C	Baud rate divisor latch	16hxxxx
R8_UART3_DIV	0x0E	Predivider divisor register	8hxx

Modulator-demodulator (MODEM) control register (R8_UARTx_MCR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
7	RB_MCR_HALF	RW	UART0 half-duplex receive/transmit mode	0

			enable/disable bit 1: Enter half-duplex receive/transmit mode, giving priority to transmission, receiving when not transmitting; 0: Disable half-duplex receive/transmit mode. Note: Only UART0 supports.	
6	RB_MCR_TNOW	RW	Being transmitting status (TNOW) output (DTR pin) enable bit of UART0: 1: Enable the DTR pin of UART0 to output the status of being transmitting TNOW, which can be used to control RS485 receive/transmit switching; 0: Disable. Note: Only UART0 supports.	0
5	RB_MCR_AU_FLOW_EN	RW	UART0 allow CTS and RTS hardware automatic flow control bit: 1: Allow CTS and RTS hardware automatic flow control; 0: Invalid. Note: Only UART0 supports. In the flow control mode, if this bit is 1, then UART will continuously send the next data only when it detects that the CTS pin input is valid (active low). Otherwise, UART will suspend transmission. And CTS input status change will not generate MODEM status interrupt when this bit is 1. If this bit is 1 and RTS is 1, UART will automatically validate the RTS pin (active low) when receiver FIFO is empty. UART will automatically invalidate the RTS pin when the number of received bytes reaches the trigger point of FIFO and will re-validate the RTS pin when the receiver FIFO is empty. The CTS pin can be connected to another RTS pin through the hardware automatic band rate control, and the RTS pin can be connected to another CTS pin.	0
4	RB_MCR_LOOP	RW	Test mode control bit of UART0 internal loop: 1: Enable the test mode of internal loop; 0: Disable the test mode of internal loop. In the test mode of the internal loop, all output pins of UART are invalid, TXD internally returns to RXD (i.e., the output of TSR internally returns to the input of RSR), RTS internally returns to CTS, DTR internally returns to DSR, OUT1 internal returns to RI and OUT2 internally returns to DCD Note: Only UART0 supports.	0

3	RB_MCR_OUT2	RW	UART interrupt request output control bit: 1: Enable; 0: Disable.	0
2	RB_MCR_OUT1	RW	User-defined MODEM control bit, and no actual output pin is connected: 1: Set to high level; 0: Set to low level. Note: Only UART0 supports.	0
1	RB_MCR_RTS	RW	RTS pin output valid control bit: 0: Enable RTS pin output valid (active low) 0: Disable RTS pin output valid Note: Only UART0 supports.	0
0	RB_MCR_DTR	RW	DTR pin output valid control bit: 1: Enable pin output valid (active low); 0: Disable pin output valid. Note: Only UART0 supports.	0

Interrupt enable register (R8_UARTx_IER) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
7	RB_IER_RESET	WZ	UART software reset control bit, cleared automatically: 1: Software resets UART; 0: Do not perform software reset.	0
6	RB_IER_TXD_EN	RW	UART TXD pin output enable bit: 1: Enable pin output; 0: Disable pin output.	0
5	RB_IER_RTS_EN	RW	RTS pin output enable bit of UART0: 1: Enable pin output; 0: Disable pin output. Note: Only UART0 supports.	0
4	RB_IER_DTR_EN	RW	DTR/TNOW pin output enable bit of UART0: 1: Enable pin output; 0: Disable pin output. Note: Only UART0 supports.	0
3	RB_IER_MODEM_CHG	RW	Modem input status change interrupt enable bit of UART0: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt. Note: Only UART0 supports.	0
2	RB_IER_LINE_STAT	RW	Receive line status interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	RB_IER_THR_EMPTY	RW	Transmit hold register empty interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	RB_IER_RECV_RDY	RW	Receive data interrupt enable bit:	0

			1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	

FIFO control register (R8_UARTx_FCR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value

[7:6]	RB_FCR_FIFO_TRIG	RW	Interrupt and hardware flow control trigger point set domain of receiver FIFO:
			00: 1 byte;
			01: 2 bytes;
			10: 4 bytes;
			11: 7 bytes.
			This domain is used to set the interrupt and hardware flow control trigger point of receiver FIFO. For example: 00 corresponds to 1 byte, that is, interrupt available for receiving data is generated when 1 byte is received, and RTS pin is automatically invalidated when hardware flow control is enabled.
				0

[5:3]	Reserved	RO	Reserved	0
2	RB_FCR_TX_FIFO_CLR	WZ	Transmitter FIFO data clear enable bit, cleared automatically: 1: Clear the data of transmitter FIFO (excluding TSR); 0: Not clear the data of transmitter FIFO.	0
1	RB_FCR_RX_FIFO_CLR	WZ	Receiver FIFO data clear enable bit, cleared automatically: 1: Clear the data of receiver FIFO (excluding RSR); 0: Not clear the data of receiver FIFO.	0
0	RB_FCR_FIFO_EN	RW	FIFO enable bit: 1: Enable FIFO, internal FIFO size is 8 bytes; 0: Disable FIFO. After disabling FIFO, it is 16C450 compatible mode, which means that there is only one byte in FIFO (RECV_TG1=0, RECV_TG0=0, FIFO_EN=1), and it is recommended to enable FIFO.	0

Line control register (R8_UARTx_LCR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
7	RB_LCR_DLAB/ RB_LCR_GP_BIT	RW	UART common bit (user-defined).	0
6	RB_LCR_BREAK_EN	RW	Forced to generate BREAK line interval enable bit: 1: Forced to generate BREAK line interval. 0: Not generate BREAK line interval.	0

[5:4]	RB_LCR_PAR_MOD	RW	Parity check bit format set domain
			00: Odd check;
			01: Even check;
			10: Flag bit (MARK, set to 1);
			11: Space bit (SPACE, cleared).
			This domain is valid only when RB_LCR_PAR_EN bit is 1.
				0

3	RB_LCR_PAR_EN	RW	Parity bit enable bit: 1: Allow to generate parity bit when transmitting and check parity bit when receiving; 0: No parity bit.	0
2	RB_LCR_STOP_BIT	RW	Stop bit format set bit: 0: One stop bit; 1: Two stop bits.	0

[1:0]	RB_LCR_WORD_SZ	RW	UART data length set domain:
			00: 5 data bits;
			01: 6 data bits;
			10: 7 data bits;
			11: 8 data bits.
				0

Interrupt identification register (R8_UARTx_IIR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:6]	RB_IIR_FIFO_ID	RO	UART FIFO enable status bit: 1: FIFO is enabled; 0: FIFO is not enabled.	0
[5:4]	Reserved	RO	Reserved.	0
[3:0]	RB_IIR_INT_MASK	RO	Interrupt flag domain: If RB_IIR_NO_INT bit is 0, an interrupt is generated, and it is needed to judge the interrupt source by reading this domain. See the table below for details	0
0	RB_IIR_NO_INT	RO	UART no interrupt flag bit: 1: No interrupt; 0: Interrupt.	1

The meaning of bit RB_IIR_NO_INT of interrupt recognition register (R8_UARTx_IIR) and each bit of RB_IIR_INT_MASK domain are shown in the following table:

Table 8-3 Meaning of RB_IIR_INT_MASK in IIR Register

IIR register bit				Precedence level	Interrupt Type	Interrupt sources	Clearing interrupt Method
IID3	IID2	IID1	NOINT				
0	0	0	1	None	No interrupt generated	No interrupt	

1	1	1	0	0	Bus address matching	Receive 1 data and it is UART bus address, and the address matches the preset slave value or the broadcast address. Note: This interrupt only applies to UART0.	Read IIR or disable multi-device mode
0	1	1	0	1	Receive line status	OVER_ERR, PAR_ERR, FRAM_ERR, BREAK_ERR	Read LSR
0	1	0	0	2	Receive data available	The number of bytes received reaches the trigger point of FIFO.	Read RBR
1	1	0	0	2	Receive data timeout	No next data is received when the time of four data is exceeded.	Read RBR
0	0	1	0	3	THR register empty	Transmit hold register empty, or RB_IER_THR_EMPTY bit is changed from 0 to 1 for triggering.	Read IIR or write THR
0	0	0	0	4	MODEM input change	Trigger by setting △CTS, △DSR, △RI and △DCD to 1.	Read MSR

Line status register (R8_UARTx_LSR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
7	RB_LSR_ERR_RX_FIFO	RO	Receiver FIFO error flag bit: 1: There is at least one PAR_ERR or FRAM_ERR or BREAK_ERR error in the receiver FIFO; 0: There is no error in receiver FIFO.	0
6	RB_LSR_TX_ALL_EMP	RO	Transmit hold register THR and transmit shift register TSR empty flag. 1: Both of them are empty; 0: Both of them are not empty.	1
5	RB_LSR_TX_FIFO_EMP	RO	Transmitter FIFO empty flag bit: 1: Transmitter FIFO is empty; 0: Transmitter FIFO is not empty.	1
4	RB_LSR_BREAK_ERR	RZ	BREAK line interval detection flag bit: 1: BREAK line interval is detected: 0: BREAK line interval is not detected:	0
3	RB_LSR_FRAME_ERR	RZ	Data frame error flag bit: 1: It means the frame error of the data being read from the receiver FIFO due to lack of a valid stop bit; 0: The data frame is correct.	0
2	RB_LSR_PAR_ERR	RZ	Parity error flag bit of received data: 1: It means that there is parity error of the data being read from the receiver FIFO; 0: Parity check is correct.	0
1	RB_LSR_OVER_ERR	RZ	Receiver FIFO buffer overflow flag bit: 1: Overflowed; 0: Not overflowed.	0
0	RB_LSR_DATA_RDY	RO	Flag bit of data received in receiver FIFO:	0

			1: There is data in FIFO; 0: No data. After reading all data in the FIFO, this bit will be automatically cleared.	

Modulator-demodulator (MODEM) status register (R8_UART0_MSR) (only supported by UART0)

Bit	Name	Access	Description	Reset value
7	RB_MSR_DCD	RO	DCD pin status bit: 1: DCD pin is valid (active low); 0: DCD pin is invalid (high level). Note: Only UART0 supports.	x
6	RB_MSR_RI	RO	RI pin status bit: 1: RI pin is valid (active low); 0: RI pin is invalid (high level). Note: Only UART0 supports.	x
5	RB_MSR_DSR	RO	DSR pin status bit: 1: DSR pin is valid (active low); 0: DSR pin is invalid (high level). Note: Only UART0 supports.	x
4	RB_MSR_CTS	RO	CTS pin status bit: 1: CTS pin is valid (active low); 0: CTS pin is invalid (high level). Note: Only UART0 supports.	x
3	RB_MSR_DCD_CHG	RZ	DCD pin input status change flag bit: 1: Change; 0: No change. Note: Only UART0 supports.	0
2	RB_MSR_RI_CHG	RZ	RI pin input status change flag bit: 1: Change; 0: No change. Note: Only UART0 supports.	0
1	RB_MSR_DSR_CHG	RZ	DSR pin input status change flag bit: 1: Change; 0: No change. Note: Only UART0 supports.	0
0	RB_MSR_CTS_CHG	RZ	CTS pin input status change flag bit: 1: Change; 0: No change. Note: Only UART0 supports.	0

Receive buffer register (R8_UARTx_RBR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_RBR	RO	Data receive buffer register. If the DATA_RDY bit of LSR is 1, the received data can be read from this register; If FIFO_EN is 1, the data received from UART shift register RSR will be firstly stored	x

			in the receiver FIFO, and then read out through the register	

Transmit hold register (R8_UARTx_THR) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_THR	WO	Transmit hold register. Transmitter FIFO is included, used to write the data to be transmitted. If FIFO_EN is 1, the written data will be firstly stored in the transmitter FIFO, and then output one by one through the transmit shift register TSR.	x

Receiver FIFO count register (R8_UARTx_RFC) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_RFC	RO	Data count in the current receiver FIFO. The maximum value is 8.	x

Transmitter FIFO count register (R8_UARTx_TFC) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_TFC	RO	Data count in the current transmitter FIFO. The maximum value is 8.	x

Baud rate divisor latch (R16_UARTx_DL) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UARTx_DL	RW	16-bit divisor, used to calculate the baud rate. Formula: Divisor = the internal reference clock of UART / 16 / the required communication baud rate. For example: If the internal reference clock of UART is 1.8432MHz and the required baud rate is 9600bps, then the divisor =1843200/16/9600=12.	x

Pre-frequency division divisor register (R8_UARTx_DIV) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UARTx_DIV	RW	Used to calculate the internal reference clock of UART, the lower 7 bits are valid.. Formula: Divisor = Fsys*2 / internal reference clock of UART, with a maximum value of 127. Example: If the system main clock is 96MHz and the divisor is 104, then the internal reference clock of UART is 1.846MHz, and	x

			the difference between it and the commonly used reference clock 1.8432 is 0.16%	

Slave address register (R8_UART0_ADR) Only used by UART0

Bit	Name	Access	Description	Reset value
[7:0]	R8_UART0_ADR	RW	UART0 slave address: FFh: Not used; Other: Slave address.	8hFF

R8_UART0_ADR presets the address of this computer as a slave, which is used to automatically compare the received addresses during multi-device communication, and generate an interrupt when the address matches or when the broadcast address 0FFH is received. Meanwhile, it is allowed to receive subsequent data packets. It is not allowed to receive any data before the address does not match. After sending data or rewriting to the R8_UART0_ADR register, stop receiving any data, until the address is matched again next time or the broadcast address is received.

When R8_UART0_ADR is 0FFH or RB_LCR_PAR_EN=0, the automatic comparison function of bus address is disabled.

When R8_UART0_ADR is not 0FFH and RB_LCR_PAR_EN=1, the automatic comparison function of bus address is enabled, and the following parameters should be configured: RB_LCR_WORD_SZ is 11b to select 8 data bits. For the case when the address byte is MARK (that is, the bit 9 of data byte is 0), RB_LCR_PAR_MOD should be set to 10b; for the case when the address byte is SPACE (that is, the bit 9 of data byte is 1), RB_LCR_PAR_MOD should be set to 11b.

7.3 UART Application

The UART0/1/2/3 output pins of CH567 chip are all at 3.3V LVCMOS level. The pins in asynchronous serial port mode include: data transmission pins (supported by UART0/1/2/3) and MODEM contact signal pins (only supported by UART0). Data transmission pins include: TXD pin and RXD pin, both of which are at high level by default. MODEM contact signal pins include: CTS pin, DSR pin, RI pin, DCD pin, DTR pin, RTS pin, all of which are at high level by default. All these MODEM communication signals can be used as general-purpose IO pins, controlled by the application program and their purposes can be defined.

4 sets of UARTs each has built-in independent receive/transmit buffer and 8-byte FIFO, they support simplex, half-duplex or full duplex asynchronous serial communication. Serial data includes 1 low-level start bit, 5/6/7/8 data bits, 0 or 1 additional check bit or flag bit, 1 or 2 high-level stop bits, and supports odd/even/mark/blank check. Common communication baud rates are supported: 1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K, 230.4K, 460.8K, 921.6K, 1.8432M, 2.7648M, 7.8125M, etc. The baud rate error of UART transmit signal is less than 0.2%, and the allowable baud rate error of UART receive signal is not greater than 2%.

7.3.1 Baud Rate Calculation

1) Calculate the reference clock, set the R8_UART0_DIV register, with the maximum value of 127;

2) Calculate the baud rate and set R16_UART0_DL register;

formula: Baud rate =Fsys * 2 / R8_UART0_DIV / 16 / R16_UART0_DL.

7.3.2 UART Transmission

"THR register empty" interrupt sent by UART (the low 4 bits of IIR register are 02H) refers to transmitter FIFO empty. The interrupt is cleared when the IIR is read, or the interrupt may be cleared when the next data is written to THR. If only one byte is written to THR, it will soon generate a request of "transmit hold register (THR) empty interrupt" as the byte is quickly transferred to the transmit shift register

(TSR) to start transmitting. At this point, the next data ready to be transmitted can be written. After all data in TSR is removed, UART transmission is completed and RB_LSR_TX_ALL_EMP bit of LSR changes to 1 and becomes valid.

In interrupt trigger mode, when UART transmit hold register THR empty interrupt is received, if FIFO is enabled, up to 8 bytes can be written to THR and FIFO at a time and will be transmitted automatically by the controller in sequence. If FIFO is disabled, only one byte can be written at a time. If no data needs to be transmitted, simply exit (the interrupts have been automatically cleared when IIR is read earlier).

In the query mode, it can judge whether the transmitter FIFO is empty according to RB_LSR_TX_FIFO_EMP bit of LSR. If this bit is 1, data can be written to THR and FIFO. If FIFO is enabled, up to 8 bytes can be written at a time.

7.3.3 UART Reception

UART receive data available interrupt (the low 4 bits of IIR are 04H) means that the number of existing data bytes in the receiver FIFO has reached or exceeded the FIFO trigger points set and selected by RB_FCR_FIFO_TRIG of FCR register. The interrupt is cleared when the data is read from RBR to cause the number of bytes in the FIFO below the FIFO trigger point.

UART receive data timeout interrupt (the low 4 bits of IIR are 0CH) means that there is at least one byte of data in the receiver FIFO, and it has waited for the time equivalent to the time for receiving 4 data when UART receives data last time and the MCU takes the data last time. The interrupt is cleared when a new data is received again, or the interrupt can also be cleared when the MCU reads RBR once. When receiver FIFO is empty, RB_LSR_DATA_RDY bit of LSR is 0. When there is data in receiver FIFO, RB_LSR_DATA_RDY bit is 1 and valid.

In the interrupt trigger mode, R8_UARTx_RFC register can be read to query the remaining data count in the current FIFO after receiving UART receive data timeout interrupt, read all data directly, or continuously query the RB_LSR_DATA_RDY bit of LSR. If this bit is valid, read the data until this bit becomes invalid. After receiving UART receive data available interrupt, you can read the number of bytes set by RB_FCR_FIFO_TRIG of FCR from RBR, and then directly read the data for the number of bytes, or you can read all the data in the current FIFO according to the RB_LSR_DATA_RDY bit and the R8_UARTx_RFC register.

In query mode, the MCU can judge whether the receiver FIFO is empty according to the RB_LSR_DATA_RDY bit of LSR, or read the R8_UARTx_RFC register to get the data count in the current FIFO and get all the data received by UART.

7.3.4 Hardware Flow Control

Hardware flow control includes automatic CTS (RB_MCR_AU_FLOW_EN of MCR is 1) and automatic RTS (RB_MCR_AU_FLOW_EN and RB_MCR_RTS of MCR are 1).

If automatic CTS is enabled, CTS pin must be valid before UART transmits data. UART transmitter detects CTS pin before transmitting the next data. When CTS pin state is valid, the transmitter transmits the next data. In order to ensure that transmitter stops sending the later data, CTS pin must be invalidated before the middle time of the last stop bit currently transmitted. The automatic CTS function reduces the interrupt applied to MCU system. When hardware flow control is enabled, a change of CTS pin level does not trigger a MODEM interrupt as the controller automatically controls the transmitter based on CTS pin status. If automatic RTS is enabled, RTS pin output will be valid only when there is enough space in FIFO to receive data, and RTS pin output is invalid when the receiver FIFO is full. RTS pin output will be valid if all the data in the receiver FIFO is taken or cleared. When the trigger point for the receiver FIFO is reached (the number of existing bytes in the receiver FIFO is not less than the number of bytes set by RB_FCR_FIFO_TRIG of FCR), RTS pin output is invalid, and the transmitter of the other side is allowed to send another data after RTS pin is invalid. Once the data in the receiver FIFO is emptied, RTS pin will be automatically re-enabled, so that the transmitter of the other side restores transmitting. If both automatic CTS and automatic RTS are enabled (both RB_MCR_AU_FLOW_EN and RB_MCR_RTS of MCR register are 1), one side will not transmit data unless there is enough space in the receiver FIFO of the other side when RTS pin of one side is connected to CTS pin of the other side. Therefore, the hardware flow control can avoid FIFO overflow and timeout errors during UART reception.

Chapter 8 General purpose Timer (TMRx)

8.1 Introduction to TMRx

CH567 chip is equipped with 3 26-bit timers, TMR0, TMR1 and TMR2, and the longest timing interval is 2^26 clock cycles. All timers support capture, PWM and interrupt functions. In addition, TMR1 and TMR2 support DMA functions.

Features:

(1). 3 26-bit timers, and the longest timing interval of each timer is 2^26 clock cycles;

(2). Each timer supports PWM function;

(3). Each timer supports capture function;

(4). Timer interrupt is supported by each timer, and among them, TMR1 and TMR2 support DMA and interrupt;

(5). The capture function can be set to level change capture function and high or low level hold time capture function;

(6). PWM function supports dynamically adjust PWM duty cycle settings;

8.2 Register Description

TMR0 related register physical start address: 0x0040 2000

TMR1 related register physical start address: 0x0040 2400

TMR2 related register physical start address: 0x0040 2800

Table 8-1 List of TMR0 Related Registers

Name	Offset address	Description	Reset value
R8_TMR0_CTRL_MOD	0x00	Mode setting register	8h02
R8_TMR0_INTER_EN	0x02	Interrupt enable register	8h00
R8_TMR0_INT_FLAG	0x06	Interrupt flag register	8h00
R8_TMR0_FIFO_COUNT	0x07	FIFO count register	8h00
R32_TMR0_COUNT	0x08	Current count value register	32h0000 0000
R32_TMR0_CNT_END	0x0C	Final count value setting register	32h0000 0000
R32_TMR0_FIFO	0x10	FIFO register	32h0000 0000

Table 8-2 List of TMR1 Related Registers

Name	Offset address	Description	Reset value
R8_TMR1_CTRL_MOD	0x00	Mode setting register	8h02
R8_TMR1_CTRL_DMA	0x01	DMA control register	8h00
R8_TMR1_INTER_EN	0x02	Interrupt enable register	8h00
R8_TMR1_INT_FLAG	0x06	Interrupt flag register	8h00
R8_TMR1_FIFO_COUNT	0x07	FIFO count register	8h00
R32_TMR1_COUNT	0x08	Current count value register	32h0000 0000
R32_TMR1_CNT_END	0x0C	Final count value register	32h0000 0000
R32_TMR1_FIFO	0x10	FIFO register	32h0000 0000
R16_TMR1_DMA_NOW	0x14	Current address of DMA buffer	16h0000
R16_TMR1_DMA_BEG	0x18	Start address of DMA buffer	16h0000
R16_TMR1_DMA_END	0x1C	End address of DMA buffer	16h0000

Table 8-3 List of TMR2 Related Registers

Name	Offset address	Description	Reset value
R8_TMR2_CTRL_MOD	0x00	Mode setting register	8h02
R8_TMR2_CTRL_DMA	0x01	DMA control register	8h00
R8_TMR2_INTER_EN	0x02	Interrupt enable register	8h00
R8_TMR2_INT_FLAG	0x06	Interrupt flag register	8h00
R8_TMR2_FIFO_COUNT	0x07	FIFO countregister	8h00
R32_TMR2_COUNT	0x08	Current count value register	32h0000 0000
R32_TMR2_CNT_END	0x0C	Final count value register	32h0000 0000
R32_TMR2_FIFO	0x10	FIFO register	32h0000 0000
R16_TMR2_DMA_NOW	0x14	Current address of DMA buffer	16h0000
R16_TMR2_DMA_BEG	0x18	Start address of DMA buffer	16h0000
R16_TMR2_DMA_END	0x1C	End address of DMA buffer	16h0000

Mode setting register (R8_TMRx_CTRL_MOD) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[7:6]	RB_TMR_CAP_EDGE	RW	In capture mode, edge trigger mode setting domain: 00: Disable trigger; 01: Capture the time between any edge changes; 10: Capture the time between falling edges; 11: Capture the time between rising edges;	0

[7:6]	RB_TMR_PWM_REPEAT	RW	PWM repeat mode setting domain:
			00: Repeat for 1 time;
			01: Repeat for 4 times;
			10: Repeat for 8 times;
			11: Repeat for 16 times.
				0

5	Reserved	RO	Reserved.	0
4	RB_TMR_CAP_COUNT	RW	Auxiliary selection; when RB_TMR_MODE_IN=1, set: 1: Count; 0: Capture.	0
4	RB_TMR_OUT_POLAR	RW	In PWM mode, output polarity setting bit: 1: Default high level, active low; 0: Default low level, active high;	0
3	RB_TMR_OUT_EN	RW	Timer output enable bit 1: Timer output is enabled; 0: Timer output is disabled.	0
2	RB_TMR_COUNT_EN	RW	Timer module enable bit: 1: Enable; 0: Disable.	0
1	RB_TMR_ALL_CLEAR	RW	FIFO and Count register clear bit of counter: 1: Force to clear; 0: No action.	1
0	RB_TMR_MODE_IN	RW	Timer mode setting bit: 1: Capture/count mode 0: Timing mode/PWM mode	0

Interrupt enable register (R8_TMRx_INTER_EN) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved.	x
4	RB_TMR_IE_FIFO_OV	RW	FIFO overflow interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	RB_TMR_IE_DMA_END	RW	DMA end interrupt enable bit, not supported by TMR0: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	RB_TMR_IE_FIFO_HF	RW	FIFO half use interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt. (capture fifo >=4 or PWM fifo <=3)	0
1	RB_TMR_IE_DATA_ACT	RW	In capture mode, level change interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt. In PWM mode, PWM completion interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	RB_TMR_IE_CYC_END	RW	In capture mode, capture timeout interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt. In PWM mode, PWM clock cycle end interrupt enable bit: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

Interrupt flag register (R8_TMRx_INT_FLAG) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved.	x
4	RB_TMR_IF_FIFO_OV	RW1	FIFO overflow flag bit: 1: Overflowed; 0: No overflow.	0
3	RB_TMR_IF_DMA_END	RW1	DMA completion flag bit, not supported by TMR0: 1: Completed; 0: Not completed.	0
2	RB_TMR_IF_FIFO_HF	RW1	FIFO half count flag bit: 1: FIFO has been half counted; 0: FIFO has not been half counted. (capture fifo >=4 or PWM fifo <=3)	0
1	RB_TMR_IF_DATA_ACT	RW1	In the capture mode, capture level change	0

			flag bit: 1: Capture the level change; 0: No level change is captured. In PWM mode, PWM trigger flag bit: 1: Triggered (PWM count reaches the specified value); 0: Not triggered.	
0	RB_TMR_IF_CYC_END	RW1	In capture mode, timeout flag bit: 1: Timeout; 0: Not timeout. In PWM mode, PWM cycle end flag bit: 1: End; 0: Not end. Timing mode: 1: Timing cycle ends; 0: Not end. Cleared by writing 1.	0

FIFO count register (R8_TMRx_FIFO_COUNT) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[7:0]	R8_TMRx_FIFO_COUNT	RO	Data byte count in the FIFO, with the maximum value of 8.	x

Current count value register (R32_TMRx_COUNT) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[31:0]	R32_TMRx_COUNT	RO	Current count value of counter.	x

Final count value setting register (R32_TMRx_CNT_END) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[31:0]	R32_TMRx_CNT_END	RW	In timer mode, number of timing clocks; In PWM mode, the total number of clocks in PWM cycle; In the capture mode, the number of captured timeout clock cycles; The maximum value is 67108864. Note: R32_TMRx_COUNT counts from 0, so the maximum value is R32_TMRx_CNT_END minus 1. Only the lower 26 bits are valid.	x

FIFO register (R32_TMRx_FIFO) (x=0/1/2)

Bit	Name	Access	Description	Reset value
[31:0]	R32_TMRx_FIFO	RO/ WO	FIFO data register, only the lower 26 bits are valid.	x

DMA control register (R8_TMRx_CTRL_DMA) (x=1/2)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved.	x
2	RB_TMR_DMA_LOOP	RW	DMA address loop function enable bit, not supported by TMR0: 1: Enable DMA address loop function; 0: Disable DMA address loop function. If the DMA address loop mode function is enabled, when the DMA address is added to the set end address, it will automatically loop to the start address set.	0
1	Reserved	RO	Reserved.	0
0	RB_TMR_DMA_ENABLE	RW	DMA function enable bit, not supported by TMR0: 1: Enable DMA; 0: Disable DMA.	0

DMA current buffer address (R16_TMRx_DMA_NOW) (x=1/2)

Bit	Name	Access	Description	Reset value
[15:0]	R16_TMRx_DMA_NOW	RW	Current address of DMA data buffer. It can be used to calculate the number of conversions, and the calculation method is: COUNT=(TMR_DMA_NOW-TMR_DMA_BEG)/4	xxxx

DMA start buffer address (R16_TMRx_DMA_BEG) (x=1/2)

Bit	Name	Access	Description	Reset value
[15:0]	R16_TMRx_DMA_BEG	RW	For the start address of DMA data buffer, 4 bytes must be aligned for the address. That is, in PWM data transmission or capture mode, the data captured starts from this buffer address.	xxxx

DMA end buffer address (R16_TMRx_DMA_END) (x=1/2)

Bit	Name	Access	Description	Reset value
[15:0]	R16_TMRx_DMA_END	RW	End address of DMA data buffer, 4 bytes must be aligned for the address. That is, in PWM data transmit or capture mode, the data captured ends in this buffer address.	xxxx

8.3 TMRx Function

8.3.1 Timing and Counting Function

There are 3 timers in CH567, each of which supports a maximum timing interval of 2^26 clock cycles.

If the system clock cycle is 96M, the longest time interval is: 10.4ns*2^26 ≈ 0.7s. If the system clock is lower than 96M, the time interval is longer.

3 timers have an independent interrupt.

The register with timing function is initialized as follows:

(1). Set register R32_TMRx_CNT_END to the time value that is needed for timing;

The specific calculation method is: Time = Fsys * R32_TMRx_CNT_END

(2). Set the RB_TMR_MODE_IN bit of register R8_TMRx_CTRL_MOD to 0, and RB_TMR_ALL_CLEAR to 0;

(3). Set the RB_TMR_COUNT_EN bit of register R8_TMRx_CTRL_MOD to 1, to start the timer function;

(4). At the end of timing interval, set the RB_TMR_IF_CYC_END bit of register R8_TMRx_INT_FLAG to 1, and cleared by writing 1.

8.3.2 PWM Function

3 timers of CH567 chip all have PWM function. The default output polarity of PWM can be set to high level or low level. The number of repeat times can be selected as 1, 4, 8 or 16. This repeat function can be combined with DMA to imitate the effect of DAC. The shortest time cycle for PWM output is 1 system clock cycle, and the duty cycle of PWM can be dynamically modified to imitate special waveforms, such as quasi-sine-wave.

PWM function operation:

It is needed to set the register (R32_TMRx_FIFO) and register (R32_TMRx_CNT_END) when PWM outputs, R32_TMRx_FIFO is the data register, R32_TMRx_CNT_END is the PWM total cycle register.

PWM operation steps are as follows:

(1). Set the PWM total cycle register R32_TMRx_CNT_END, the minimum value is 1, the value of this register must be greater than or equal to the value of R32_TMRx_FIFO register;

(2). Set the data register R32_TMRx_FIFO, the minimum value is 0, with the corresponding duty cycle of 0%, the maximum value is the same as that of R32_TMR_CNT_END, with the corresponding duty cycle of 100%, the calculation of duty cycle: R32_TMRx_FIFO/R32_TMRx_CNT_END. TMR1 and TMR2 support continuous dynamic data (DMA), which can imitate special waveforms;

(3). Clear the RB_TMR_MODE_IN bit of the mode setting register (R8_TMRx_CTRL_MOD) to 0 to enable PWM mode. At the same time, set the RB_TMR_ALL_CLEAR bit to 1 and then clear it to forcefully clear the FIFO and COUNT. Set the RB_TMR_OUT_POLAR bit to select the output polarity. If it is needed to set the number of repeat times, set the RB_TMR_PWM_REPEAT domain as needed.

(4). Set the RB_TMR_COUNT_EN and RB_TMR_OUT_EN bits of the mode setting register (R8_TMRx_CTRL_MOD) to 1 to enable the PWM function;

(5). Set the I/O pin corresponding to PWM as output;

(6). If it is needed to enable interrupts, set the corresponding interrupt enable register bit;

(7). After the PWM is completed, if the interrupt is turned on, the corresponding timer interrupt will be generated. At the same time, read the R8_TMRx_INT_FLAG register to know whether the PWM is completed and whether an error occurred during the PWM process;

For example: Set the RB_TMR_OUT_POLAR bit to 0, R32_TMRx_FIFO to 6, R32_TMRx_CNT_END to 18, the basic timing diagram of generating PWM is as follows, and its duty cycle is:

PWM duty cycle = R32_TMRx_FIFO/R32_TMRx_CNT_END = 1/3

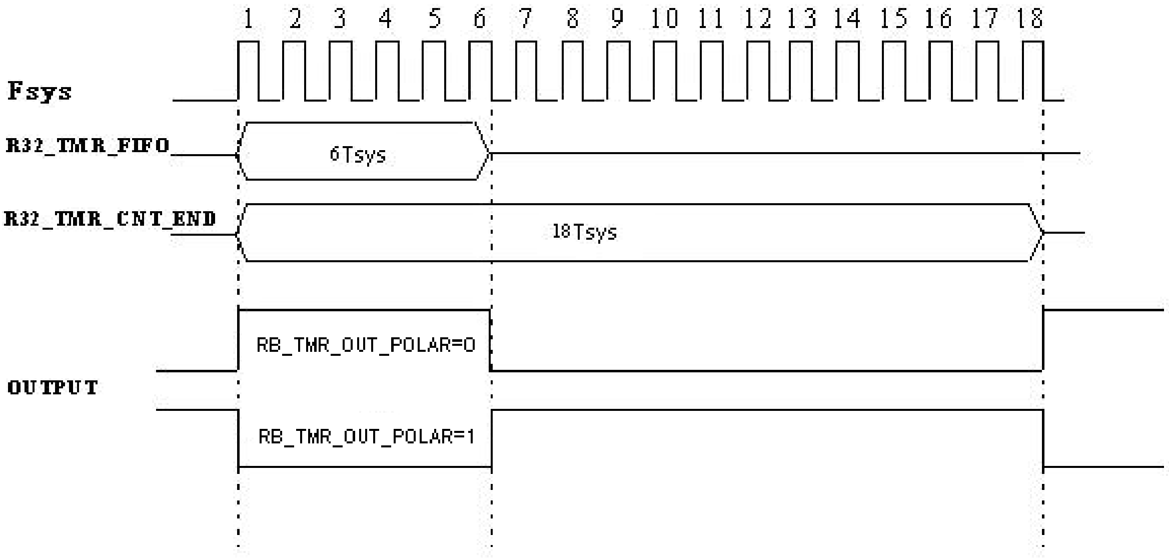


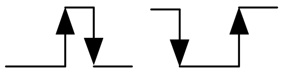
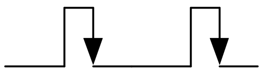
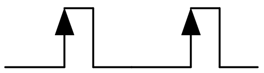
Figure 8-1 PWM Output Timing Diagram

If RB_TMR_PWM_REPEAT domain is set to 00, it means that the above process is repeated once, 01 means that the above process is repeated for 4 times, 10 means that the above process is repeated for 8 times, and 11 means that the above process is repeated for 16 times. After repeating, take the next data in FIFO and then continue.

8.3.3 Capture Function

3 timers of CH567 chip all have capture function, among which the capture functions of TMR1 and TMR2 support DMA data storage. Three capture modes can be selected: start triggering at any edge and end at any edge, start triggering at rising edge and end at rising edge, and start triggering at falling edge and end at falling edge. The following is the description of capture trigger mode:

Table 8-4 Description of Capture Trigger Mode

Capture Mode Selection Bit (RB_TMR_CATCH_EDGE)	Trigger Mode	Graphical representation
00	Disable capture	None
01	Edge trigger	
10	Falling edge to falling edge	
11	Rising edge to rising edge	

There are two trigger states in edge trigger mode, which can capture high level width or low level width. When the highest bit (bit 25) of the valid data in data register (R32_TMRx_FIFO) is 1, it indicates that high level is captured. Otherwise, low level is captured. If the bit25 of multiple sets of data is 1 (or 0), the width of the high (or low) level exceeds the timeout value, and multiple sets need to be accumulated.

In the trigger modes from falling edge to falling edge or from rising edge to rising edge, an input change cycle can be captured. When the highest bit (bit 25) of the valid data in data register (R32_TMRx_FIFO) is 0, one cycle is normally sampled. When it is 1, the input change period exceeds the timeout value R32_TMRx_CNT_END, and the latter set of data needs to be added and accumulated as a single input change cycle.

The specific description is shown in the figure below:

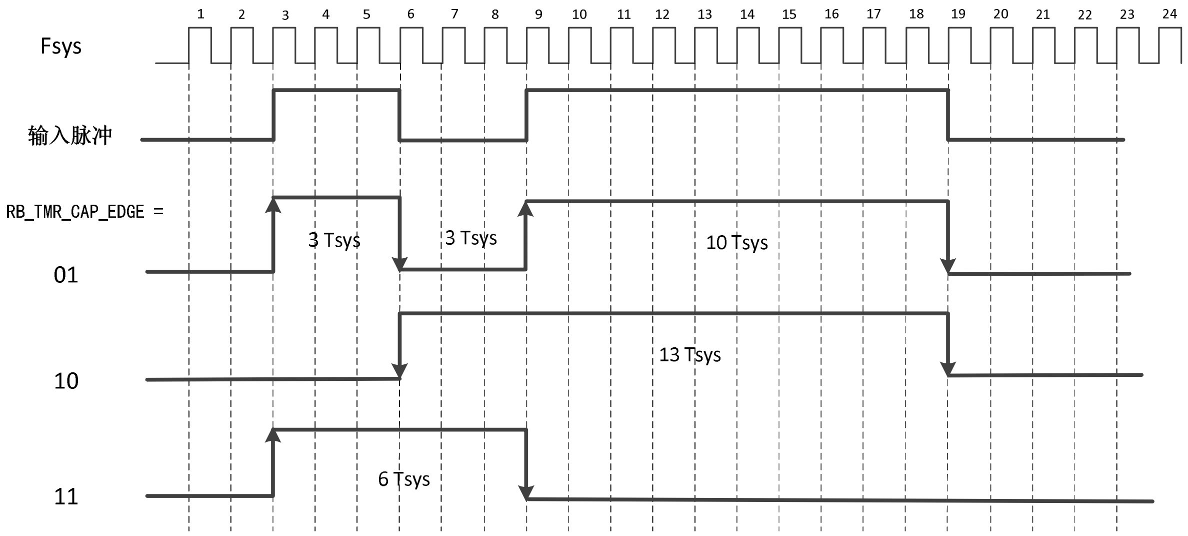


Figure 8-2 Taking System Clock Cycle as Capture Cycle

As shown in the figure above, sample once in each clock cycle,

When RB_TMR_CATCH_EDGE = 2b01, it is set as edge-triggered sampling, and the sampled time widths are 3Tsys, 3Tsys, 10Tsys;

When RB_TMR_CATCH_EDGE = 2b10, it is set as the sampling from falling edge to falling edge, and the sampled time width is 13Tsys;

When RB_TMR_CATCH_EDGE = 2b11, it is set as the sampling from rising edge to rising edge, and the sampled time width is 6Tsys.

Operation steps for capture mode:

(1). Set the register R32_TMRx_CNT_END to set the capture timeout time. The default maximum timeout time is 2^26 clock cycles. It is recommended to set a reasonable timeout to avoid no input change for a long time and avoid no time data. If no level change is detected within the maximum timeout time, set bit 25 of R32_TMRx_FIFO register to 1;

(2). Set the direction of the I/O pin corresponding to capture as input;

(3). Set the RB_TMR_MODE_IN bit of mode setting register (R8_TMRx_CTRL_MOD) to 1 and set RB_TMR_CAP_COUNT bit to 0, and clear the RB_TMR_ALL_CLEAR bit to clear FIFO and COUNT. At the same time, set the RB_TMR_CAP_EDGE control bit to select the capture mode;

(4). If interrupt needs to be enabled, set the corresponding bit of interrupt register R8_TMRx_INTER_EN to 1, to enable the corresponding interrupt;

(5). To save the captured data by DMA method (only supported by TMR1 and TMR2), you need to set RB_TMR_DMA_ENABLE bit of R8_TMRx_CTRL_DMA register to 1, to enable the DMA function, and set the register R16_TMRx_DMA_BEG to the first address of the buffer for storing the captured data, and set the register R16_TMRx_DMA_END as the end address of the buffer for storing captured data;

(6). Set the RB_TMR_COUNT_EN bit of register R8_TMRx_CTRL_MOD to 1, to enable timer module and start the capture function;

(7). After the capture is completed, the register R8_TMRx_INT_FLAG will generate the corresponding interrupt status. The data captured by default is stored in the register R32_TMRx_FIFO. If DMA data transmission is used, the captured data will be automatically stored in the data buffer set by DMA.

Chapter 9 PWM

9.1 Introduction to PWM Controller

CH567 is provided with 4 channels of PWM output, the duty cycle is adjustable, and the PWM cycle is fixed and 2 modes are available, and the operation is simple.

The extended PWM pin output is identified as PWM3/ PWM4/ PWM5/ PWM6, among which PWM5 and PWM6 support remap to PWM5_ and PWM6_ pins.

9.2 Register Description

PWM0/1/2/3 related register physical base address: 0x0040 5000

Table 9-1 List of PWM0/1/2/3 Related Registers

Name	Offset address	Description	Reset value
R8_PWM_CTRL_MOD	0x00	PWM mode control register	8h00
R8_PWM_CTRL_CFG	0x01	PWM configuration control register	8h00
R8_PWM_CLOCK_DIV	0x02	PWM clock frequency division register	8h00
R32_PWM_DATA	0x08	PWM0/1/2/3 data hold register	32h

PWM mode control register (R8_PWM_CTRL_MOD)

Bit	Name	Access	Description	Reset value
7	RB_PWM3_POLAR	RW	PWM3 output polarity control bit: 1: Default high level, active low; 0: Default low level, active high;	0
6	RB_PWM2_POLAR	RW	PWM2 output polarity control bit: 1: Default high level, active low; 0: Default low level, active high;	0
5	RB_PWM1_POLAR	RW	PWM1 output polarity control bit: 1: Default high level, active low; 0: Default low level, active high;	0
4	RB_PWM0_POLAR	RW	PWM0 output polarity control bit: 1: Default high level, active low; 0: Default low level, active high;	0
3	RB_PWM3_OUT_EN	RW	PWM3 output enable bit: 1: Enable; 0: Disable.	0
2	RB_PWM2_OUT_EN	RW	PWM2 output enable bit: 1: Enable; 0: Disable.	0
1	RB_PWM1_OUT_EN	RW	PWM1 output enable bit: 1: Enable; 0: Disable.	0
0	RB_PWM0_OUT_EN	RW	PWM0 output enable bit: 1: Enable; 0: Disable.	0

PWM configuration control register (R8_PWM_CTRL_CFG)

Bit	Name	Access	Description	Reset value
[7:1]	Reserved	RW	Reserved.	0
0	RB_PWM_CYCLE_SEL	RW	PWM cycle selection: 1: 255 clock cycles; 0: 256 clock cycles.	0

PWM clock divider register (R8_PWM_CLOCK_DIV)

Bit	Name	Access	Description	Reset value
[7:0]	R8_PWM_CLOCK_DIV	RW	PWM reference clock frequency division factor. Calculation: CLK=Fsys/R8_PWM_CLOCK_DIV.	0

PWM0/1/2/3 data hold register (R32_PWM_DATA)

Bit	Name	Access	Description	Reset value
[31:24]	R8_PWM3_DATA	RW	PWM3 data hold register.	xx
[23:16]	R8_PWM2_DATA	RW	PWM2 data hold register.	xx
[15:8]	R8_PWM1_DATA	RW	PWM1 data hold register.	xx
[7:0]	R8_PWM0_DATA	RW	PWM0 data hold register.	xx

9.3 PWM Configuration

1) Set PWM0-PWM3 pin direction as output; optionally, set the drive capability of corresponding I/O;

2) Set the register R8_PWM_CLOCK_DIV to calculate the clock reference frequency of PWM;

3) Set the PWM mode control register R8_PWM_CTRL_MOD, configure the output polarity of PWMx, and enable the corresponding PWMx (RB_PWMx_OUT_EN position 1) output;

4) Set the R8_PWM_CTRL_CFG register and R32_PWM_DATA register to configure the PWM duty cycle output.

Calculation formula:

PWMx duty cycle = R8_PWMx_DATA / (RB_PWM_CYCLE_SEL? 255 : 256)

Note: If the corresponding RB_PWMx_OUT_EN output enable is always on in the R8_PWM_CTRL_MOD register, the PWM waveform will be output continuously until RB_PWMx_OUT_EN is disabled.

Chapter 10 LED Screen Controller

10.1 Introduction to LED Controller

CH567 is equipped with an LED screen control card interface and built-in 4-byte FIFO, which supports DMA and interrupts, saves CPU processing time and supports 1/2/4-channel data line interfaces.

10.2 Register Description

LED related register physical base address: 0x0040 6000

Table 10-1 List of LED Related Registers

Name	Offset address	Description	Reset value
R8_LED_CTRL_MOD	0x00	LED mode configuration register	8h02
R8_LED_CLOCK_DIV	0x01	LED serial clock divider register	8hxx
R8_LED_STATUS	0x04	LED status register	8h00
R16_LED_FIFO	0x08	LED FIFO register	16hxxxx
R16_LED_DMA_CNT	0x10	LED DMA remaining count register	16hxxxx
R16_LED_DMA_MAIN	0x14	LED main buffer DMA address	16hxxxx
R16_LED_DMA_AUX	0x18	LED auxiliary buffer DMA address	16hxxxx

LED mode configuration register (R8_LED_CTRL_MOD)

Bit	Name	Access	Description	Reset value

[7:6]	RB_LED_CHAN_MOD	RW	LED channel mode setting domain:
			00: LED0, single channel output;
			01: LED0/1, dual channel output;
			10: LED0~3, 4-channel output;
			11: LED0~3, 4-channel output, among which the data of LED2/3 channels comes from auxiliary buffer.
				0

5	RB_LED_IE_FIFO	RW	FIFO half count interrupt enable: 1: FIFO count<=2 interrupt trigger; 0: Disable the corresponding interrupt.	0
4	RB_LED_DMA_EN	RW	LED DMA function and DMA interrupt enable: 1: Enable; 0: Disable.	0
3	RB_LED_OUT_EN	RW	LED signal output: 1: Enable; 0: Disable.	0
2	RB_LED_OUT_POLAR	RW	LED data output polarity control bit: 1: Flip output, data 0 outputs 1, data 1 outputs 0; 0: Direct output, data 0 outputs 0, data 1 outputs 1.	0
1	RB_LED_ALL_CLEAR	RW	Clear LED FIFO and counter: 1: Force to clear; 0: No action.	1
0	RB_LED_BIT_ORDER	RW	LED serial data bit sequence:	0

			1: High byte is the first; 0: Low byte is the first;	

LED serial clock divider register (R8_LED_CLOCK_DIV)

Bit	Name	Access	Description	Reset value
[7:0]	R8_LED_CLOCK_DIV	RW	LED control output clock frequency division factor. Calculation: CLK=Fsys/R8_LED_CLOCK_DIV.	xx

LED status register (R8_LED_STATUS)

Bit	Name	Access	Description	Reset value
7	RB_LED_IF_DMA_END	RW1	DMA transmission completion flag bit: 1: DMA transmission is completed; 0: Not completed. Cleared by writing 1 or R16_LED_DMA_CNT	0
6	RB_LED_FIFO_EMPTY	RO	FIFO empty status bit: 1: FIFO is empty; 0: FIFO is not empty.	0
5	RB_LED_IF_FIFO	RW1	FIFO half count interrupt flag bit: 1: FIFO count <= 2; 0: FIFO count > 2. Cleared by writing 1 or R16_LED_FIFO	0
4	RB_LED_CLOCK	O	Current LED clock signal level status: 1: High level; 0: Low level.	0
3	Reserved	RO	Reserved.	0
[2:0]	RB_LED_FIFO_COUNT	RO	The byte count in the current FIFO must be an even number.	0

LED FIFO register (R16_LED_FIFO)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_FIFO	WO	LED data FIFO entry, 16-bit write.	xxxx

LED DMA remaining count register (R16_LED_DMA_CNT)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_DMA_CNT	RW	Current DMA remaining word (16-bit) count of LED_DMA_MAIN main buffer area. It will automatically decrease after DMA is started and only the lower 12 bits are valid. Auxiliary buffers are not included.	xxxx

LED main buffer DMA address (R16_LED_DMA_MAIN)

Bit	Name	Access	Description	Reset value

[15:0]	R16_LED_DMA_MAIN	RW	DMA start address/current address of the main buffer area, is automatically increased after the initial value is preset.	xxxx

LED auxiliary buffer DMA address (R16_LED_DMA_AUX)

Bit	Name	Access	Description	Reset value
[15:0]	R16_LED_DMA_AUX	RW	DMA start address/current address of auxiliary buffer area, is automatically increased after the initial value is preset.	xxxx

10.3 LED Control Application

1) Set LEDC and the necessary LED0～LED3 pin directions as output, optionally, set the corresponding I/O drive capability;

2) Set R8_LED_CLOCK_DIV to select the LED output clock frequency;

3) Set the DMA start address R16_LED_DMA_MAIN to point to the buffer that is ready to output data, that is, the main buffer;

4) If LED channel mode 3 is selected, the auxiliary DMA start address R16_LED_DMA_AUX must be set to point to the auxiliary buffer;

5) Set the LED control register R8_LED_CTRL_MOD, select the channel mode, output polarity, bit sequence, enable interrupt and DMA functions, etc.;

6) Set the DMA count register R16_LED_DMA_CNT, to start DMA transmission, or transmit data by writing to FIFO;

7) Query or use interrupt processing to interrupt the corresponding status.

Chapter 11 USB Controller

11.1 Introduction to USB Controller

CH567 is embedded with two-channel USB2.0 controller and USB-PHY, with dual roles of host controller and USB device controller. When used as a host controller, they can support low-speed, full-speed and high-speed USB devices. When used as a device controller, it can be flexibly set to low-speed, full-speed or high-speed mode, to adapt to various applications.

The features of USB0 and USB1 controllers are as follows:

1) Support USB 2.0, USB 1.1 and USB 1.0;
2) Support USB Host functions and USB Device functions;
3) Host supports high-speed HUB;
4) The hardware can be configured as high-speed, full-speed and low-speed device;
5) Both the host and the device support control transmission, batch transmission, interrupt transmission, real-time/synchronous transmission;
6) Support directly access the data of each endpoint buffer by DMA;
7) Support suspend, remote wake-up and resume functions;
8) Except device endpoint 0, all other endpoints support the data packets up to 512 bytes, and some endpoints support double buffering.

11.2 Register Description

CH567 is integrated with double USB2.0 master-slave controller (built-in PHY), USB0 and USB1, and can be flexibly configured as host function or device function.

The USB0 and USB1 related registers of CH567 are divided into 3 parts, some of which are multiplexed in the host and device modes.

1) USB global register;
2) USB device controller register;
3) USB host controller register.

USB0 related register physical base address: 0x0040 8000
USB1 related register physical base address: 0x0040 9000

11.2.1 Global Register Description

Table 11-1 List of USB Global Registers

Name	Offset address	Description	Reset value
USBx_CTRL	00h	USB control register	8h06
USBx_INT_EN	02h	USB interrupt enable register	8h00
USBx_DEV_AD	03h	USB device address register	8h00
USBx_FRAME_NO	04h	USB frame number register	16h0000
USBx_SUSPEND	06h	USB suspend control register	8h00
USBx_SPPED_TYPE	08h	USB current speed type register	8h00
USBx_MIS_ST	09h	USB miscellaneous status register	8hxx10_1000b
USBx_INT_FG	0Ah	USB interrupt flag register	8h00
USBx_INT_ST	0Bh	USB interrupt status register	8h00xx_xxxxb
USBx_RX_LEN	0Ch	USB receive length register	16hxxxx

USB control register (USBx_CTRL) (x=0/1)

Bit	Name	Access	Description	Reset value
7	bUC_HOST_MODE	RW	USB working mode selection bit: 0: Device mode (DEVICE); 1: Host mode (HOST).	0

[6:5]	UC_SPEED_TYPE	RW	USB bus signal transmission rate selection bit:
			00: Full speed;
			01: High speed;
			10: Low speed.
				00b

4	bUC_DEV_PU_EN	RW	In device mode, USB device enable and internal pull-up resistor control bit: 1: Enable USB device transmission and enable internal pull-up resistor; 0: Disable.	0
3	bUC_INT_BUSY	RW	Automatic suspend enable bit before USB transmission completion interrupt flag is not cleared: 1: It will automatically suspend before the interrupt flag UIF_TRANSFER is not cleared. It will automatically respond to busy NAK in device mode, and will automatically suspend the subsequent transmission in host mode; 0: Not suspend.	0
2	bUC_RESET_SIE	RW	Software reset control bit of USB protocol processor: 1: Forcibly resetting the USB protocol processor (SIE), it needs to be cleared by software; 0: Not reset.	1
1	bUC_CLR_ALL	RW	1: Empty USB interrupt flag and FIFO, it needs software to clear; 0: Not empty.	1
0	bUC_DMA_EN	RW	DMA and DMA interrupt control bit of USB: 1: enable the DMA function and DMA interrupt; 0: Switch off DMA.	0

USB interrupt enable register (USBx_INT_EN) (x=0/1)

Bit	Name	Access	Description	Reset value
7	bUIE_DEV_SOF	RW	In USB device mode, receive SOF packet interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
6	bUIE_DEV_NAK	RW	In USB device mode, receive NAK interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
5	Reserved	RO	Reserved.	0

4	bUIE_FIFO_OV	RW	FIFO overflow interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	bUIE_HST_SOF	RW	In USB host mode, SOF timing interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	bUIE_SUSPEND	RW	USB bus suspend or wake-up event interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	bUIE_TRANSFER	RW	USB transfer completion interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	bUIE_DETECT	RW	In USB host mode, the USB device connection or disconnection event interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	bUIE_BUS_RST	RW	In USB device mode, USB bus reset event interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

USB device address register (USBx_DEV_AD) (x=0/1)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved.	0
[6:0]	MASK_USB_ADDR	RW	In host mode, it is the address of the USB device being operated or HUB address; In device mode, it is the address of the USB device.	00h

USB frame number register (USBx_FRAME_NO) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	USB_FRAME_NO	RO	Frame number, it refers to the frame number of the SOF packet to be sent in host mode, and it refers to the frame number of the SOF packet currently received in device mode. Among them, the lower 11 bits are the valid frame number, and the higher 3 bits are the micro frame number of high-speed mode.	0

USBx_FRAME_NO is a 16-bit register; among them, the lower 11 bits represent the SOF packet frame number, and the higher 3 bits represents the current micro-frame nomber, which can be used for interrupt, synchronous/real-time transmission when operating high-speed HUB.

USB suspend register (USBx_SUSPEND) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved.	0

1	bUS_RESUME	RW	Remote wake-up control bit: 1: Wake up the host remotely; 0: No action.	0
0	Reserved	RO	Reserved.	0

Note: When remote wake-up is required, pull the bUS_RESUME bit up and then down.

USB speed type register (USBx_SPEED_TYPE) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved.	0

[1:0]	USB_SPEED_TYPE	RO	In host mode, it refers to the speed type of the currently connected device. In device mode, it refers to the speed type of the current device.
			00: Full speed;
			01: High speed;
			10: Low speed.
				00b

Note: Different from UC_SPEED_TYPE in USBx_CTRL register, UC_SPEED_TYPE represents the highest speed expected. Assuming that in device mode, set UC_SPEED_TYPE to high speed. When the device is connected to a host at full speed, the actual speed type is full speed and it can be known by querying USB_SPEED_TYPE register. In host mode, set UC_SPEED_TYPE to high speed. When a device at full speed is connected, the actual communication speed is full speed and it can be known by querying the USBx_SPEED_TYPE register.

USB miscellaneous status register (USBx_MIS_ST) (x=0/1)

Bit	Name	Access	Description	Reset value
7	bUMS_SOF_PRES	RO	SOF packet indication status bit in USB host mode: 1: SOF packet will be sent, and it will be automatically delayed if there are other USB data packets; 0: No SOF packet will be sent.	x
6	bUMS_SOF_ACT	RO	SOF packet transmission status bit in USB host mode: 1: SOF packet is being sent out; 0: The transmission is completed or idle.	x
5	bUMS_SIE_FREE	RO	Idle status bit of USB protocol processor: 1: Idle protocol processor; 0: Busy, USB transmission is in progress.	1
4	bUMS_R_FIFO_RDY	RO	USB receiver FIFO data ready status bit: 1: Receiver FIFO is non-empty; 0: Receiver FIFO is empty.	0
3	bUMS_BUS_RESET	RO	USB bus reset status bit: 1: The current USB bus is at reset status; 0: The current USB bus is at non-reset status.	0
2	bUMS_SUSPEND	RO	USB suspend status bit: 1: The USB bus is in a suspended state, and there is no USB activity for a period of time; 0: USB bus is at non-suspended status.	0

1	bUMS_ATTACH	RO	USB device connection status bit of the port in USB host mode: 1: The port has been connected to a USB device; 0: No USB device is connected to the port.	0
0	bUMS_SPLIT_CAN	RO	In USB host mode, SPLIT packet transmission permission bit: 1: Allow to transmit SPLIT packet; 0: Disable transmission.	0

USB interrupt flag register (USBx_INT_FG) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved.	00b
4	UIF_FIFO_OV	RW1	USB FIFO overflow interrupt flag bit, write 1 to clear it: 1: FIFO overflow trigger; 0: No event.	0
3	UIF_HST_SOF	RW1	SOF timing interrupt flag bit in USB host mode, cleared by writing 1: 1: SOF transmission completion trigger; 0: No event.	0
2	UIF_SUSPEND	RW1	USB bus suspension or wake-up event interrupt flag bit, write 1 to clear it: 1: USB suspension event or wake-up event trigger; 0: No event.	0
1	UIF_TRANSFER	RW1	USB transfer completion interrupt flag bit, write 1 to clear it: 1: A USB transmission completion trigger; 0: No event.	0
0	UIF_DETECT	RW1	In USB host mode, USB device connection or disconnection event interrupt flag bit, write 1 to clear it: 1: USB device connection or disconnection trigger is detected; 0: No event.	0
0	UIF_BUS_RST	RW1	USB bus reset event interrupt flag bit in USB device mode, write 1 to clear it: 1: USB bus reset event trigger; 0: No event.	0

USB interrupt status register (USBx_INT_ST) (x=0/1)

Bit	Name	Access	Description	Reset value
7	bUIS_IS_NAK	RO	In USB device mode, NAK response status bit: 1: Respond to NAK during current USB transmission; 0: No NAK response.	0

6	bUIS_TOG_OK	RO	Current USB transmission DATA0/1 synchronization flag match status bit: 1: Synchronous; 0: Asynchronous.	0
[5:4]	MASK_UIS_TOKEN	RO	In device mode, the token PID of the current USB transfer transaction.	xxb
[3:0]	MASK_UIS_ENDP	RO	In device mode, the endpoint number of the current USB transfer transaction.	xxxxb
[3:0]	MASK_UIS_H_RES	RO	In host mode, respond PID identification of the current USB transmission transaction. 0000 indicates that device has no response or timeout. Other values indicate that respond PID.	xxxxb

MASK_UIS_TOKEN is used to identify the token PID of the current USB transmission transaction in USB device mode: 00 means OUT packet; 01 means SOF packet; 10 means IN packet; 11 means SETUP packet.

MASK_UIS_H_RES is only valid in host mode. In host mode, if the host transmits OUT/SETUP token packet, the PID is the handshake packet ACK/NAK/STALL/NYET, or the device has no response/timeout. If the host transmits IN token packet, the PID is the PID of the data packet (DATA0/DATA1/DATA2/MDATA) or the handshake packet PID.

USB receive length register (USBx_RX_LEN) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	USB_RX_LEN	RO	Current count of data received by the USB endpoint, among which the lower 11 bits are valid, and the higher 5 bits are fixed to 0.	xxxxh

11.2.2 Device Register Description

In USB device mode, each USB device controller of CH567 is equipped with 5 groups of bidirectional endpoints 0, 1, 2, 3, 4. The maximum data packet length of all endpoints except endpoint 0 is 512 bytes, and the maximum data packet length of endpoint 0 is 64 bytes.

Endpoint 0 is the default endpoint and supports control transmission. Transmission and reception share a 64-byte data buffer area.

Endpoint 1, endpoint 2, endpoint 3 each includes a transmission endpoint IN and a reception endpoint OUT. The transmission and reception each has a separate 512-byte or double 512-byte data buffer, support batch transmission, interrupt transmission, and real-time/synchronous transmission.

Endpoint 4 includes a transmission endpoint IN and a reception endpoint OUT. The transmission and reception each has a separate 512-byte data buffer, support batch transmission, interrupt transmission, and real-time/synchronous transmission.

Endpoint 0/1/2/3 can all set the UEPn_DMAx register to configure their DMA address respectively. Set the endpoint receive/transmit data buffer mode through UEP4_1_MODx and UEP2_3_MODx registers.

Each group of endpoints are equipped with receive/transmit control register UEPn_TX_CTRLx and UEPn_RX_CTRLx and transmit length register UEPn_T_LENx (n=0/1/2/3/4), which are used to set the synchronization trigger bit of this endpoint, the response to OUT transactions and IN transactions and the length of data to be sent.

As the necessary USB bus pull-up resistor of USB device, it can be set whether to be enabled by software at any time. When bUC_DEV_EN in USB control register USBx_CTRL is set to 1, the chip will set according to the speed of bUC_SPEED_TYPE, internally connect the pull-up resistor with the DP/DM pin of the USB bus and enable the USB device function.

When a USB bus reset, USB bus suspend or wake-up event is detected, or when the USB successfully processes data transmission or reception, the USB protocol processor will set corresponding interrupt flag. If the interrupt enable is switched on, the corresponding interrupt request will be also generated. The application program can directly query or query and analyze the interrupt flag register USBx_INT_FG in the USB interrupt service program, and perform corresponding processing according to UIF_BUS_RST and UIF_SUSPEND. In addition, if UIF_TRANSFER is valid, it is required to continue to analyze the USB interrupt status register USBx_INT_ST, and perform the corresponding processing according to the current endpoint number MASK_UIS_ENDP and the current transaction token PID identification MASK_UIS_TOKEN. If the synchronization trigger bit bUEP_R_TOG of OUT transaction of each endpoint is set in advance, you can judge whether the synchronization trigger bit of the data packet received matches the synchronization trigger bit of the endpoint through bUIS_TOG_OK. If the data is synchronized, the data is valid. If the data is not synchronized, the data should be discarded. After the USB transmission or reception interrupt is processed each time, the synchronization trigger bit of corresponding endpoint should be modified correctly to synchronize the data packet sent next time and detect whether the data packet received next time is synchronized. In addition, bUEP_AUTO_TOG can be set to automatically flip the corresponding synchronization trigger bit after successful transmission or reception.

The data to be sent by each endpoint is in their own buffer, and the length of the data to be sent is independently set in UEPn_T_LENx, and the length of one packet transmitted cannot exceed 512 bytes. The data received by each endpoint is in their own buffer, but the length of the data received is in the USB receive length register USBx_RX_LEN, and it can be distinguished according to the current endpoint number when the USB receives an interrupt. The maximum packet length that can be received by each endpoint needs to be written into the UEPn_MAX_LENx register in advance.

Table 11-2 List of USB Device Registers

Name	Offset address	Description	Reset value
UEP4_1_MODx	10h	Mode control register of endpoint 1/4	8h00
UEP2_3_MODx	11h	Mode control register of endpoint 2/3	8h00
UEP0_DMAx	14h	Start address of endpoint 0 buffer	16hxxxx
UEP1_DMAx	18h	Start address of endpoint 1 buffer	16hxxxx
UEP2_DMAx	1Ch	Start address of endpoint 2 buffer	16hxxxx
UEP3_DMA	20h	Start address of endpoint 3 buffer	16hxxxx
UEP0_MAX_LENx	24h	Endpoint 0 maximum length packet register	16hxxxx
UEP1_MAX_LENx	28h	Endpoint 1 maximum length packet register	16hxxxx
UEP2_MAX_LENx	2Ch	Endpoint 2 maximum length packet register	16hxxxx
UEP3_MAX_LENx	30h	Endpoint 3 maximum length packet register	16hxxxx
UEP4_MAX_LENx	34h	Endpoint 4 maximum length packet register	16hxxxx
UEP0_T_LENx	38h	Endpoint 0 transmit length register	16hxxxx
UEP0_TX_CTRLx	3Ah	Endpoint 0 transmit control register	8h00
UEP0_RX_CTRLx	3Bh	Endpoint 0 receive control register	8h00
UEP1_T_LENx	3Ch	Endpoint 1 transmit length register	16hxxxx
UEP1_TX_CTRLx	3Eh	Endpoint 1 transmit control register	8h00
UEP1_RX_CTRLx	3Fh	Endpoint 1 receive control register	8h00
UEP2_T_LENx	40h	Endpoint 2 transmit length register	16hxxxx
UEP2_TX_CTRLx	42h	Endpoint 2 transmit control register	8h00
UEP2_RX_CTRLx	43h	Endpoint 2 receive control register	8h00
UEP3_T_LENx	44h	Endpoint 3 transmit length register	16hxxxx
UEP3_TX_CTRLx	46h	Endpoint 3 transmit control register	8h00
UEP3_RX_CTRLx	47h	Endpoint 3 receive control register	8h00
UEP4_T_LENx	48h	Endpoint 4 transmit length register	16hxxxx

UEP4_TX_CTRLx	4Ah	Endpoint 4 transmit control register	8h00
UEP4_RX_CTRLx	4Bh	Endpoint 4 receive control register	8h00

Mode control register of USB endpoint 1/4 (UEP4_1_MODx) (x=0/1)

Bit	Name	Access	Description	Reset value
7	bUEP1_RX_EN	RW	1: Enable endpoint 1 reception (OUT); 0: Disable endpoint 1 reception.	0
6	bUEP1_TX_EN	RW	1: Enable endpoint 1 transmission (IN); 0: Disable endpoint 1 transmission.	0
5	Reserved	RO	Reserved.	0
4	bUEP1_BUF_MOD	RW	Endpoint 1 data buffer mode control bit.	0
3	bUEP4_RX_EN	RW	1: Enable endpoint 4 reception (OUT); 0: Disable endpoint 4 reception.	0
2	bUEP4_TX_EN	RW	1: Enable endpoint 4 transmission (IN); 0: Disable endpoint 4 transmission.	0
[1:0]	Reserved	RO	Reserved.	0

The data buffer modes of USB endpoint 0/4 are configured by a combination of bUEP4_RX_EN and bUEP4_TX_EN. Refer to the following table for the details:

Table 11-3 Endpoint 0/4 Buffer Mode

bUEP4_RX_EN	bUEP4_TX_EN	Description: arrange from low to high with UEP0_DMA as start address
0	0	Endpoint 0 single 64-byte transmit/receive shared buffer (IN and OUT), and endpoint 4 is disabled for transceiving.
1	0	Endpoint 0 single 64-byte transmit/receive shared buffer; endpoint 4 single 512-byte receive buffer (OUT).
0	1	Endpoint 0 single 64-byte transmit/receive shared buffer; endpoint 4 single 512-byte transmit buffer (IN).
1	1	Endpoint 0 single 64-byte transmit/receive shared buffers; endpoint 4 single 512-byte receive buffer (OUT); Endpoint 4 single 512-byte receive buffer (IN). All 1088 bytes are arranged as follows: UEP0_DMA+0 address: 64-byte start address of endpoint 0 transmi/receive shared buffer; UEP0_DMA+64 address: 512-byte start address of endpoint 4 receive buffer; UEP1_DMA+64+512 address: 512-byte start address of endpoint 4 transmit buffer.

USB endpoint 2/3 mode control register (UEP2_3_MODx) (x=0/1)

Bit	Name	Access	Description	Reset value
7	bUEP3_RX_EN	RW	1: Enable endpoint 3 reception (OUT); 0: Disable endpoint 3 reception.	0
6	bUEP3_TX_EN	RW	1: Enable endpoint 3 transmission (IN); 0: Disable endpoint 3 transmission.	0
5	Reserved	RO	Reserved.	0
4	bUEP3_BUF_MOD	RW	Endpoint 3 data buffer mode control bit.	0
3	bUEP2_RX_EN	RW	1: Enable endpoint 2 reception (OUT);	0

			0: Disable endpoint 2 reception.	
2	bUEP2_TX_EN	RW	1: Enable endpoint 2 transmission (IN); 0: Disable endpoint 2 transmission.	0
1	Reserved	RO	Reserved.	0
0	bUEP2_BUF_MOD	RW	Endpoint 2 data buffer mode control bit.	0

The data buffer modes of USB endpoint 1/2/3 are controlled by a combination of bUEPn_RX_EN, bUEPn_TX_EN and bUEPn_BUF_MOD (n=1/2/3) respectively, refer to the following table for details. Among them, in the double 512-byte buffer mode, the first 512-byte buffer will be selected based on bUEP_*_TOG=0 and the last 512-byte buffer will be selected based on bUEP_*_TOG=1 during USB data transmission, and bUEP_AUTO_TOG=1 is set to realize automatic switch.

Table 11-4 Endpoint n Buffer Mode (n=1/2/3)

bUEPn_RX_EN	bUEPn_TX_EN	bUEPn_BUF_MOD	Description: arrange from low to high with UEPn_DMA as start address
0	0	x	Endpoint is disabled, and the UEPn_DMA buffer is not used.
1	0	0	Single 512-byte receive buffer (OUT).
1	0	1	Double 512-byte receive buffer (OUT), selected by bUEP_R_TOG.
0	1	0	Single 512-byte transmit buffer (IN).
0	1	1	Double 512-byte transmit buffer (IN), selected by bUEP_T_TOG.
1	1	0	Single 512-byte receive buffer (OUT), and single 512-byte transmit buffer (IN).
1	1	1	Double 512-byte receive buffer (OUT), selected by bUEP_R_TOG. Double 512-byte transmit buffer (IN), selected by bUEP_T_TOG. All 2K bytes are arranged as follows: UEPn_DMA+0 address: Endpoint receive address when bUEP_R_TOG=0; UEPn_DMA+512 address: Endpoint receive address when bUEP_R_TOG=1; UEPn_DMA+1024 address: Endpoint tranmit address when bUEP_T_TOG=0; UEPn_DMA+1536 address: Endpoint transmit address when bUEP_T_TOG=1.

Start address of USB endpoint n buffer (UEPn_DMAx)(n=1/2/3) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	UEPn_DMA	RW	Start address of endpoint n buffer, the lowest 2 bits are fixed to 0 (4 bytes are aligned), and the highest 1 bit is fixed to 0.	xxxxh

Endpoint n maximum length packet register (UEPn_MAX_LENx) (n=1/2/3) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	UEPn_MAX_LEN	RW	Maximum packet length of data received by	xxxxh

			endpoint n.	

Note: This maximum packet length determines the maximum length of data that can be received by the endpoint. The data beyond this length will be discarded and DMA will not transmit it to the customized zone.

Endpoint n transmit length register (UEPn_T_LENx) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	UEPn_T_LEN	RW	Set the number of bytes of data to be sent by USB endpoint n, the lower 10 bits are valid, and the higher 6 bits are fixed to 0, with a maximum length of 512.	xxxxh

Endpoint n transmit control register (UEPn_TX_CTRLx) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
5	bUEP_AUTO_TOG	RW	Synchronization trigger bit auto flip enable control bit: 1: After the data is successfully transmitted, the corresponding synchronization trigger bit is automatically flipped; 0: Not flipped automatically, but can be switched manually. Only endpoint 1/2/3 supports, and real-time/synchronous transmission can only be switched manually.	0

[4:3]	MASK_UEP_T_TOG	RW	Synchronization trigger bit of the transmitter (processing IN transactions) of USB endpoint n
			00: Transmit DATA0;
			01: Transmit DATA1;
			10: Transmit DATA2;
			11: Transmit MDATA.
				0

2	bUEP_T_RES_NO	RW	1: No expected response, used to achieve real-time/synchronous transmission of non-endpoint 0. Ignore MASK_UEP_T_RES at this time; 0: Expected response.	0
[1:0]	MASK_UEP_T_RES	RW	Response control from endpoint n to IN transactions: 00: Data is ready and ACK is expected; 10: Response NAK or busy; 11: Response STALL or error.	0

Endpoint n receive control register (UEPn_RX_CTRLx) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0

5	bUEP_AUTO_TOG	RW	Synchronization trigger bit auto flip enable control bit: 1: After the data is successfully received, the corresponding synchronization trigger bit is automatically flipped; 0: It is not flipped automatically, but can be switched manually. Only endpoint 1/2/3 supports, and real-time/synchronous transmission can only be switched manually.	0

[4:3]	MASK_UEP_R_TOG	RW	Expected synchronization trigger bit of the receiver (processing OUT transactions) of USB endpoint n:
			00: Expected DATA0;
			01: Expected DATA1;
			10: Expected DATA2;
			11: Expected MDATA.
			It is invalid for real-time/synchronous transmission.
				0

2	bUEP_R_RES_NO	RW	1: No expected response, used to achieve real-time/synchronous transmission of non-endpoint 0. Ignore MASK_UEP_R_RES at this time; 0: Expected response.	0
[1:0]	MASK_UEP_R_RES	RW	Control of response from the receiver of USB endpoint n to OUT transactions: 00: Response ACK; - 10: Response NAK or busy; 11: Response STALL or error; 01: Response to NYET. It is invalid for real-time/synchronous transmission.	0

11.2.3 USB Host Register

In USB host mode, each host controller of CH567 is equipped with 1 set of bidirectional host endpoints, including a transmit endpoint OUT and a receive endpoint IN. The maximum data packet length is 512 bytes, they support control transmission, interrupt transmission, batch transmission and real-time/synchronous transmission.

Each USB transaction initiated by host endpoint always automatically sets the interrupt flag UIF_TRANSFER after processing. The application program can directly query or query and analyze the interrupt flag register USB_INT_FG in the USB interrupt service program, and perform corresponding processing according to each interrupt flag. In addition, if UIF_TRANSFER is valid, it is required to continue to analyze the USB interrupt status register USBx_INT_ST, and perform the corresponding processing according to the response PID identification MASK_UIS_H_RES of the current USB transmission transaction.

If the synchronization trigger bit bUH_R_TOG of IN transaction of host receive endpoint is set in advance, you can judge whether the synchronization trigger bit of the data packet received matches the synchronization trigger bit of the endpoint through bUIS_TOG_OK. If the data is synchronized, the data is valid. If the data is not synchronized, the data should be discarded. After the USB transmit or receive interrupt is processed each time, the synchronization trigger bit of corresponding host endpoint should be modified correctly to synchronize the data packet sent next time and detect whether the data packet received next time is synchronized. In addition, bUEP_AUTO_TOG can be set to automatically flip the corresponding synchronization trigger bit after successful transmission or reception.

USB host token setting register UH_EP_PID is used to set the endpoint number of the target device being operated and the token PID packet identification of the USB transmission transaction. The data corresponding to the SETUP token and OUT token is provided by the host transmit endpoint. The data to be transmitted is in the UH_TX_DMAx buffer, and the length of the data to be transmitted is set in UH_TX_LENx. The data corresponding to the IN token is returned by the target device to the host receive endpoint, the received data is stored in the UH_RX_DMAx buffer, and the received data length is stored in USBx_RX_LEN. The maximum packet length that can be received by the host endpoint needs to be written to the UH_RX_MAX_LENx register in advance.

Table 11-5 List of USB Host Related Registers

Name	Offset address	Description	Reset value
UHOST_CTRLx	01h	USB host control register	8h00
UH_EP_MODx	11h	USB host endpoint mode control register	8h00
UH_RX_DMAx	1Ch	USB host receive buffer area start address	16hxxxx
UH_TX_DMAx	20h	USB host transmit buffer area start address	16hxxxx
UH_RX_MAX_LENx	2Ch	USB host receive maximum length packet register	16hxxxx
UH_SETUPx	3Eh	USB host auxiliary setting register	8h00
UH_EP_PIDx	40h	USB host token setting register	8h00
UH_RX_CTRLx	43h	USB host receive endpoint control register	8h00
UH_TX_LENx	44h	USB host transmit length register	16hxxxx
UH_TX_CTRLx	46h	USB host transmit endpoint control register	8h00
UH_SPLIT_DATAx	48h	USB host transmit SPLIT packet data	16hxxxx

USB host control register (UHOST_CTRLx) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved.	0
2	bUH_TX_BUS_RESUME	RW	In host mode, the host wakes up the device.	0
1	bUH_TX_BUS_SUSPEND	RW	USB host transmit suspend signal.	0
0	bUH_TX_BUS_RESET	RW	USB host transmit bus reset signal.	0

Note: The suspend and reset time is determined by the hold time of high-level bUH_TX_BUS_SUSPEND and bUH_TX_BUS_RESET. If the host wakes up the device, it is determined by the edge method of bUH_TX_BUS_RESUME, so it is only required to pull bUH_TX_BUS_RESUME up and then down for wake-up.

USB host endpoint mode control register (UH_EP_MODx) (x=0/1)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	bUH_EP_TX_EN	RW	Host transmit endpoint transmit (SETUP/OUT) enable bit: 1: Enable endpoint transmission; 0: Disable endpoint transmission.	0

5	Reserved	RO	Reserved.	0
4	bUH_EP_TBUF_MOD	RW	Host transmit endpoint transmitg data buffer mode control bit.	0
3	bUH_EP_RX_EN	RW	Host receive endpoint receive (IN) enable bit: 1: Enable endpoint reception; 0: Disable endpoint reception.	0
[2:1]	Reserved	RO	Reserved.	00b
0	bUH_EP_RBUF_MOD	RW	USB host receive endpoint data buffer area mode control bit.	0

The data buffer modes of USB host transmit endpoint are controlled by a combination of bUH_EP_TX_EN and bUH_EP_TBUF_MOD, refer to the following table.

Table 11-6 Host transmit buffer Mode

bUH_EP_TX_EN	bUH_EP_TBUF_MOD	Description: Take UH_TX_DMA as start address
0	x	Endpoint is disabled, and UH_TX_DMA buffer is not used.
1	0	Single 512-byte transmit buffer (SETUP/OUT).
1	1	Double 512-byte transmit buffer, selected by bUH_T_TOG: When bUH_T_TOG=0, select the first 512 bytes of the buffer; When bUH_T_TOG=1, select the last 512 bytes of the buffer.

The data buffer modes of USB host receive endpoint are controlled by a combination of bUH_EP_RX_EN and bUH_EP_RBUF_MOD, refer to the following table.

Table 11-7 Host receive buffer Mode

bUH_EP_RX_EN	bUH_EP_RBUF_MOD	Description: Take UH_TX_DMA as start address
0	x	Endpoint is disabled, and UH_RX_DMA buffer is not used.
1	0	Single 512-byte receive buffer (IN).
1	1	Double 512-byte receive buffer, selected by bUH_R_TOG: When bUH_R_TOG=0, select the first 512 bytes of the buffer; When bUH_R_TOG=1, select the last 512 bytes of the buffer.

Start address of USB host receive buffer (UH_RX_DMAx) (x=0/1)

Bit	**Name**	**Access**	**Description**	**Reset value**
[15:0]	UH_RX_DMA	RW	Start address of host endpoint data receive buffer, the lowest 2 bits are fixed to 0 (4 bytes are aligned), and the highest 1 bit is fixed to 0.	xxxxh

Start address of USB host transmit buffer (UH_TX_DMAx) (x=0/1)

Bit	**Name**	**Access**	**Description**	**Reset value**
[15:0]	UH_TX_DMA	RW	Start address of host endpoint data transmit buffer, the lowest 2 bits are fixed to 0 (4 bytes are aligned), and the highest 1 bit is fixed to 0.	xxxxh

USB host receive maximum length packet register (UH_RX_MAX_LENx) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	UH_RX_MAX_LEN	RW	Maximum packet length of data received by the host endpoint.	xxxxh

Note: This maximum packet size determines the maximum length of data that can be received by the endpoint. The data beyond this length will be discarded and DMA will send it to the customized zone.

USB host auxiliary device register (UH_SETUPx) (x=0/1)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved.	0
6	bUH_SOF_EN	WO	Automatical generate SOF packet enable control bit: 1: The host automatically generates SOF packet; 0: No SOF packet is generated.	0
[5:0]	Reserved	RO	Reserved.	00

USB host token setting register (UH_EP_PIDx) (x=0/1)

Bit	Name	Access	Description	Reset value
[7:4]	MASK_UH_TOKEN	RW	Set the token PID packet identification of this USB transmission transaction.	0000b
[3:0]	MASK_UH_ENDP	RW	Set the endpoint number of the target device being operated this time.	0000b

USB host receive endpoint control register (UH_RX_CTRLx) (x=0/1)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved.	0b
6	bUH_R_DATA_NO	RW	1: Data packet is not expected, used for high-speed HUB operation in host mode; 0: Data packet expected (IN).	0
5	bUH_R_AUTO_TOG	RW	Synchronization trigger bit auto flip enable control bit: 1: After the data is successfully received, the corresponding expected synchronization trigger bit is automatically flipped; 0: It is not flipped automatically, but can be switched manually.	0

[4:3]	MASK_UH_R_TOG	RW	Synchronization trigger bit expected by the host receiver (processing IN transactions),
			00: Expected DATA0;
			01: Expected DATA1;
			10: Expected DATA2;
			11: Expected MDATA.
				0

2	bUH_R_RES_NO	RW	1: No response, used to achieve	0

			real-time/synchronous transmission of non-endpoint 0. Ignore MASK_UEP_R_RES at this time; 0: Transmit response after data is received successfully.	
[1:0]	MASK_UH_R_RES	RW	Control on response from the receiver of host to IN transactions: 00: Response ACK; - It is invalid for real-time/synchronous transmission.	00b

USB host transmit length register (UH_TX_LENx) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	UH_TX_LEN_H	RW	Set the number of bytes of data to be sent by USB host transmiy endpoint, only the lower 11 bits are valid, and the higher 5 bits are fixed to 0.	xxxxh

USB host transmit endpoint control register (UH_TX_CTRLx) (x=0/1)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved.	0b
6	bUH_T_DATA_NO	RW	1: No data packet is sent (PING/SPLIT); 0: Send data packet (OUT/SETUP).	0
5	bUH_T_AUTO_TOG	RW	Synchronization trigger bit auto flip enable control bit: 1: After the data is successfully tranmistted, the corresponding synchronization trigger bit is automatically flipped; 0: It is not flipped automatically, but can be switched manually.	0

[4:3]	MASK_UH_T_TOG	RW	Synchronization trigger bit prepared by USB host transmitter (processing SETUP/OUT transactions)
			00: Transmit DATA0;
			01: Transmit DATA1;
			10: Transmit DATA2;
			11: Transmit MDATA.
				00b

2	bUH_T_RES_NO	RW	1: No response, used to achieve real-time/synchronous transmission of non-endpoint 0. Ignore MASK_UEP_T_RES at this time; 0: Expect response after data is transmitted successfully.	0

[1:0]	MASK_UH_T_RES	RW	Response control bit from USB host transmitter to SETUP/OUT transaction
			00: Expected response ACK;
			10: Expected response NAK or busy;
				00b

			11: Expected response STALL or error; 01: Expected response to NYET. It is invalid for real-time/synchronous transmission.	

Data of SPLIT packet sent by USB host (UH_SPLIT_DATAx) (x=0/1)

Bit	Name	Access	Description	Reset value
[15:0]	UH_SPLIT_DATA	RW	Data content of SPLIT packet sent by the host endpoint, among which the lower 12 bits are valid, and the higher 4 bits are fixed to 0.	0xxxh

11.3 USB Device Mode Configuration

11.3.1 Basic Initial Configuration

1. Set bUC_HOST_MODE bit of USBx_CTRL register to 0, to configure USB device mode;

2. Set USBx_CTRL register, clear bUC_RESET_SIE and bUC_CLR_ALL bits to 0, set bUC_INT_BUSY and bUC_DMA_EN bits to 1, configure UC_SPEED_TYPE to select the speed of USB device. If it is set as a high-speed device, but the current host is at full speed, the controller will automatically slow down and switch to full speed, and the actual communication speed can be queried in the USBx_SPEED_TYPE register.

3. Clear the device address register USBx_DEV_AD and interrupt flag register USBx_INT_FG. Optional operation, enable the required interrupt, and write USBx_INT_EN register;

4. Configure the device endpoint data transceiver buffer mode register UEP4_1_MODx/ UEP2_3_MODx, and the transceiver control register UEPn_TX_CTRLx/ UEPn_RX_CTRLx;

5. Set the endpoint maximum packet receive length UEPn_MAX_LENx register and the endpoint data transceiver start address UEPn_DMAx;

6. Set bUC_DEV_PU_EN bit of USBx_CTRL register to 1, to enable USB device function.

11.4 USB Host Mode Configuration

11.4.1 Basic Initial Configuration

1. Set bUC_HOST_MODE bit of USBx_CTRL register to 1, to configure USB host mode;

2. Set the USBx_CTRL register, clear bUC_RESET_SIE and bUC_CLR_ALL bits to 0, set bUC_INT_BUSY and bUC_DMA_EN bits to 1, configure UC_SPEED_TYPE to select the speed of USB device. If it is set as a high-speed device, but the current connecting device is at full speed, the controller will automatically slow down and switch to full speed, and the actual communication speed can be queried in the USBx_SPEED_TYPE register.

3. Clear the device address register USBx_DEV_AD and interrupt flag register USBx_INT_FG. Optional operation, enable the required interrupt, and write USBx_INT_EN register;

4. Configure the host endpoint data transceiver buffer mode register UH_EP_MODx, and the transceiver control register UH_RX_CTRLx/UH_TX_CTRLx;

5. Set the host endpoint maximum packet receiving length UH_RX_MAX_LENx register and the host endpoint data transceiver start address register UH_RX_DMAx / UH_TX_DMAx;

6. Set the bUH_SOF_EN bit of UH_SETUPx register to 1, to enable the port to automatically transmit SOF packets.

Chapter 12 SD Controller and AES/SM4 Module

12.1 Introduction to SD Controller and AES/SM4 Module

CH567 chip is equipped with 4 independent SD controllers: SD0, SD1, SD2 and SD3. Compared with common controllers, they are provided with additional encryption/decryption algorithm module support, which can meet the data security requirements of the market.

The main features are as follows:

1) Support SD physical layer 1.0 and 2.0 specifications, support UHS-I SDR50 mode (forward compatible) of SD3.0 specifications;

2) Conform to 4.4 and 4.5.1 specifications of eMMC card, and compatible with 5.0 specifications and HS200 mode;

3) 4 controllers all support the single-wire and four-wire mode of eMMC card, and SD0 and SD2 support the single-wire, four-wire and eight-wire mode of eMMC card;

4) Support SD card, SDIO card, eMMC card and other devices that comply with SD protocol;

5) Support SD interface data for AES and SM4 algorithm encryption and decryption;

6) The 4 controllers work independently, support DMA and interrupts.

12.2 SD Register Description

CH567 is equipped with 4 independent SD controllers, and each controller has a similar control unit.

SD0 related register physical base address: 0x0040 A000

SD1 related register physical base address: 0x0040 A040

SD2 related register physical base address: 0x0040 A080

SD3 related register physical base address: 0x0040 A0C0

Table 12-1 List of SD Registers

Name	Offset address	Description	Reset value
SD_CLK_CFG	3Ch	Clock configuration register	16h0214
SDx_ARGUMENT	00h	Command parameter register	32h00000000
SDx_CMD_SET	04h	Command setting register	16h0000
SDx_RESPONSE0	08h	Response parameter register 0	32h00000000
SDx_RESPONSE1	0Ch	Response parameter register 1	32h00000000
SDx_RESPONSE2	10h	Response parameter register 2	32h00000000
SDx_RESPONSE3	14h	Response parameter register 3	32h00000000
SDx_WRITE_CONT	14h	Continue write start register	32h00000000
SDx_CTRL	18h	Control register	8h15
SDx_TOCNT	1Ch	Timeout count register	8h0C
SDx_STATUS	20h	Status register	32h00000000
SDx_INT_FG	24h	Interrupt flag register	16h0000
SDx_INT_EN	28h	Interrupt enable register	16h0000
SDx_DMA	2Ch	DMA start address register	16hxxxx
SDx_BLOCK_CFG	30h	Transmission block configuration register	32h00000000
SDx_TRAN_MODE	34h,	Transmission mode register	8h00

Clock configuration register (SD_CLK_CFG)

Bit	Name	Access	Description	Reset

				value
[16:10]	Reserved	RO	Reserved.	00h
9	bSDCLK_Mode	WO	Clock frequency mode selection bit: 1: High-speed mode, 25M-100MHz; 0: Low-speed mode, 400KHz.	1
8	bSDCLK_OE	WO	SD physical clock signal line output control bit: 1: Turn on, output the communication clock; 0: Turn off.	0
[7:5]	Reserved	RO	Reserved.	0
[4:0]	MASK_SD_CLK_PRE	WO	SD controller clock (SDCLK) division factor: When bSDCLK_Mode=1, then SDCLK = 480M/MASK_CLK_PRE; When bSDCLK_Mode =0, then SDCLK = 480M/MASK_CLK_PRE/64. Writing 1 is equivalent to turning off the SDC module sampling clock.	14h

Note: The clock configured by clock configuration register (SD_CLK_CFG) is shared by 4 SD card controllers.

Command parameter register (SDx_ARGUMENT) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[31:0]	SD_ARGUMENT	RW	SD/eMMC 32-bit command parameter register.	0

Command setting register (SDx_CMD_SET) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved.	0
11	bCHK_RESP_IDX	RW	Command index for check response: 1: Required; 0: Not required.	0
10	bCHK_RESP_CRC	RW	CRC for check response: 1: Required; 0: Not required.	0
[9:8]	MASK_RESP_TYPE	RW	Expected response type: 00b: No response; 01b: The response length is 136 bits; 10b: The response length is 48 bits; 11b: The response length is 48 bits, and it is R1b type response.	0
[7:6]	Reserved	RO	Reserved.	0
[5:0]	MASK_CMD_IDX	RW	The index number of the command being sent currently.	0

Response parameter register (SDx_RESPONSE) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value

[31:0]	SDx_RESPONSE0	RO	Response parameter register 0	0
[63:32]	SDx_RESPONSE1	RO	Response parameter register 1	0
[95:64]	SDx_RESPONSE2	RO	Response parameter register 2	0
[127:96]	SDx_RESPONSE3	RO	Response parameter register 3	0
[127:96]	SDx_WRITE_CONT	WO	Multiplexed SD_RESPONSE3 register, used to start write operation in the multi-block write process.	0

Note: When the response length is 136 bits, the effective data is 128 bits. When the response length is 48 bits, the effective data length is 32 bits. The SDx_RESPONSEx register is used to store the effective data parameters for response.

The SDx_RESPONSE3 register is multiplexed. The multiplexing of the register is: operation of writing this register in the process of continuously writing multiple blocks of data to the card using the CMD25 command, when the block interrupt is completed, when it is not required to change DMA address, to start the operation of writing data to SD. To change DMA address, write DMA address register to start to write data to SD, and it is not needed to start it by writing register.

Control register (SDx_CTRL) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved.	0
5	bSC_NEG_SAMPLE	RW	Cmd and Data signal line sample mode selection bit: 1: Sample on the falling edge; 1: Sample on the rising edge.	0
4	bSC_RST_DAT_LGC	RW	1: Reset the internal data receive/transmit logic, which needs to be cleared by software; 0: Work normally;	1
3	bSC_DMA_ENABLE	RW	DMA and DMA interrupt control bit of SD controller: 1: Enable DMA function and DMA interrupt; 0: Switch off DMA.	0
2	bSC_ALL_CLR	RW	1: Reset SD controller logic, which needs to be cleared by software; 0: Work normally;	1
[1:0]	DAT_LINE_WIDTH	RW	Data line width sampled by receive/transmit data logic (communication data line width): 00: The transceiver only uses dat[0], single data line; 01: The transceiver uses dat[3:0], 4 data lines; 10: The transceiver uses dat[7:0], 8 data lines; This value is only supported by the 0# and 2# controllers and used for 8-wire mode of eMMC card.	01b

Timeout control register (SDx_TOCNT) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved.	0

[3:0]	MASK_TOCNT	RW	Response/data timeout configuration: 0: Disable the internal timeout mechanism; Non-zero: Set the timeout time, and valid values are 0-12. Calculation method: SD card clock cycle * 4194304 * MASK_TOCNT. For example: If SDCLK cycle is 10ns at this time, write 12, and the timeout time is 10ns * (4194304) * (12) = 503ms.	Ch

Note: 1. The above data timeout includes the following 4 situations:

1) DAT[0] busy timeout after R1b response;

2) When writing a data block, DAT[0] busy timeout after CRC status;

3) When writing a data block, waiting for the CRC status timeout;

4) When reading a data block, waiting for the start bit timeout.

2. The command response also supports a timeout mechanism. If the response times out, it will be given by the SIF_RE_TMOUT interrupt in the interrupt register. The command timeout uses the maximum timeout value given by the protocol: 64 Tsdclk.

Status indication register (SDx_STATUS) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[32:18]	Reserved	RO	Reserved.	0
17	bST_DAT0_HI	RO	1: The current DAT0 line is at high level 0: Low level.	0
16	bST_CMD_HI	RO	1: The current CMD line is at high level; 0: Low level.	0
[15:0]	MASK_BLOCK_NUM	RO	It indicates the number of blocks that have been successfully transmitted in the current multi-block transmission operation.	0

Interrupt flag register (SDx_INT_FG) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved.	0
9	bSIF_SDIO_INT	RW1	SDIO card interrupt flag bit, write 1 to clear it: 1: SDIO card generates card interrupt; 0: No event.	0
8	bSIF_FIFO_OF	RW1	FIFO overflow interrupt flag bit, write 1 to clear it: 1: FIFO overflow trigger; 0: No event.	0
7	bSIF_BLOCK_GAP	RW1	Single block transmission completion flag bit, write 1 to clear it: 1: Single block reception/transmission completion trigger; 0: No event.	0
6	bSIF_TRANS_SC	RW1	Flag bit for completing transmitting the number of requested blocks, write 1 to clear it: 1: The transmission of number of requested	0

			blocks completed trigger; 0: No event.	
5	bSIF_TRANS_ER	RW1	Transmission CRC error flag bit, write 1 to clear it: 1: CRC error trigger; 0: No event.	0
4	bSIF_DATA_TMO	RW1	Data timeout flag bit, write 1 to clear it: 1: Data timeout trigger; 0: No event.	0
3	bSIF_CMD_DONE	RW1	Command completion flag bit, write 1 to clear it: 1: Transmit command, and receive the completion of response; 0: No event.	0
2	bSIF_RE_IDX_ER	RW	Response index number check error flag bit, write 1 to clear it: 1: Response index number check error trigger; 0: No event.	0
1	bSIF_RE_CRC_WR	RW	Response CRC check error flag bit, write 1 to clear it: 1: Response CRC check error trigger; 0: No event.	0
0	bSIF_RE_TMOUT	RW	Receive response timeout flag bit, write 1 to clear it: 1: Data timeout trigger; 0: No event.	0

Interrupt enable register (SDx_INT_EN) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
9	bSIE_SDIO_INT	RW	SDIO card interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
8	bSIE_FIFO_OF	RW	FIFO overflow is interrupted: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
7	bSIE_BLOCK_GAP	RW	Single block completion interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
6	bSIE_TRANS_SC	RW	Request block transmission completion interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
5	bSIE_TRANS_ER	RW	Block transmission CRC error interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
4	bSIE_DATA_TMO	RW	Data timeout interrupt: 1: Enable the corresponding interrupt;	0

			0: Disable the corresponding interrupt.	
3	bSIE_CMD_DONE	RW	Command completion interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	bSIE_RE_IDX_ER	RW	Response index check error interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	bSIE_RE_CRC_WR	RW	Response CRC check error interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	bSIE_RE_TMOUT	RW	Command response timeout interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

Data block DMA start address register (SDx_DMA) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[15:0]	SD_DMA	RW	Start address of read/write data buffer, the lower 3 bits are fixed to 0 (8 bytes are aligned).	0000h

Note: When reading data from SD, this register stores the read data in the start address of SRAM. When writing data to SD card, the data to be written is stored in the start address of SRAM.

If continuous multi-block read/write SD operations are performed, the user can write to the SDx_DMA register to change the DMA address as needed after the single block transmission is completed (bSIF_BLOCK_GAP).

When performing continuous multi-block writing, it is required to start the continued write operation by writing to the SDx_WRITE_CONT or SDx_DMA register after the single block transmission is completed. It is not required for multi-block read operation.

Transmission block configuration register (SDx_BLOCK_CFG) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved.	0
[27:16]	BLOCK_SIZE	RW	Single block transmission size (1-2048 bytes).	0
[15:0]	BLOCK_NUM	RW	Count of blocks (1~65535 blocks) to be transmitted by DMA this time, automatically cleared internally. If the number of blocks is not zero, enable receiving or transmitting.	0

Transmission mode register (SDx_TRAN_MODE) (x=0/1/2/3)

Bit	Name	Access	Description	Reset value
[7:3]	Reserved	RO	Reserved.	0
2	bTM_EMMC_BOOT	RW	Set the eMMC card transmission mode: 1: Boot mode; 0: Normal mode. Note: Only for eMMC card.	0
1	Reserved	RO	Reserved.	0
0	bTM_WR_SD	RW	Direction of DMA transmission: 1: From controller to SD;	0

			0: From SD to controller.	

12.3 SD Control Application

12.3.1 SD Command Transmit Operation:

1. Set the 32-bit SDx_ARGUMENT parameter register;

2. Set the 16-bit SDx_CMD_SET register;

3. Wait for the command transmit status, and query the SDx_INT_FG register. If the command is sent successfully, it will generate the command transmission success flag. Otherwise, it will generate CRC error, or timeout, or respond index error flag.

12.3.2 Operation of Reading SD Card multi-Block Data:

1. Set the SDx_DMA register, set SD0x_TRAN_MODE register to configure DMA transmission direction from SD to controller, set SD_BLOCK_CFG (the number of bytes received per block, and the number of blocks to be received by DMA), and the controller is now ready to start receiving the data blocks returned by the SD card.

2. Set the 32-bit SDx_ARGUMENT parameter register and SDx_CMD_SET register, to send CMD18 (command of reading multiple blocks).

3. Wait for the command to be sent.

4. After the controller has successfully received N blocks, transmission success interrupt may be generated (bSIF_TRANS_SC=1). If a transmission error occurs in the middle, a corresponding error interrupt will be generated. At this time, read the status register (SDx_STATUS) to know the number of blocks that have been successfully transmitted this time.

12.3.3 Operation of Writing SD Card multi-Block Data:

1. Set the 32-bit SDx_ARGUMENT parameter register and SDx_CMD_SET register, to send CMD25 (command of writing multiple blocks).

2. Wait for the command to be sent.

1. Set the SDx_DMA register, set SDx_TRAN_MODE register to configure the DMA transmission direction from controller to SD, set SDx_BLOCK_CFG (the number of bytes sent per block, and the number of blocks to be sent by the DMA), and the controller now starts to send the data block to SD card.

4. After the controller has successfully sent N blocks, transmission success interrupt may be generated (bSIF_TRANS_SC=1). If a transmission error occurs in the middle, a transmission error interrupt will be generated. At this time, read the status register (SDx_STATUS) to know the number of blocks that have been successfully transmitted this time.

12.4 AES/SM4 Module Function Description

CH567 is equipped with a built-in block cipher algorithm module, which supports two types of block cipher algorithms (AES and SM4), and electronic codebook (ECB) and counter (CTR) modes. There are totally 8 combinations as follows:

ECB mode and CTR mode with SM4 algorithm 128bit key;

ECB mode and CTR mode with AES algorithm 128bit key;

ECB mode and CTR mode with AES algorithm 192bit key;

ECB mode and CTR mode with AES algorithm 256bit key;

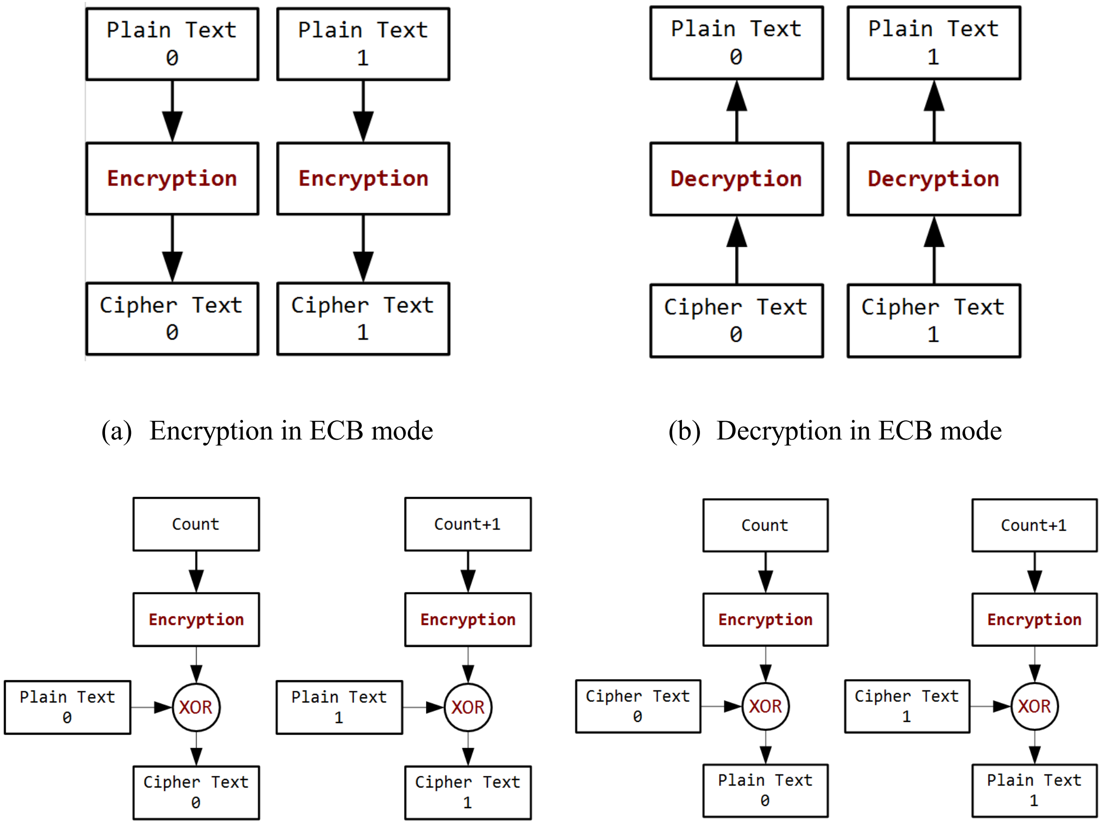
12.4.1 AES/SM4 Algorithm

The AES (Advanced Encryption Standard) algorithm is a block encryption method that uses a symmetric block cipher system, which is one of the most popular algorithms in symmetric key encryption. The SM4 block cipher algorithm is generally a special block cipher algorithm for wireless local area networks and trusted computers, and it can also be used for data encryption protection in other environments.

In the process of data encryption and decryption, the key needs to be loaded. For the AES algorithm, the user key is extended to 11×128/13×128/15×128-bit extended keys by setting the key length to 128/192/256 bits. While for SM4 algorithm, the 128-bit user key is extended to 32×32-bit extended key. These extended keys are stored in internal registers for use during encryption and decryption.

12.4.2 ECB and CTR Mode

AES/SM4 supports two modes, including electronic codebook (ECB) mode and counter (CTR) mode. Among them, the security performance of CTR mode is higher than that of ECB mode. The difference between the two is shown in Figure 13-1. In ECB mode, there is a one-to-one correspondence between plain text and cipher text, and the encrypted plain text is directly used as cipher text. While in CTR mode, a 128-bit counter value needs to be loaded in advance to encrypt the count value, and the encrypted count value is different from the plain text or used as the cipher text. It is worth noting that in CTR decryption mode, only the count value is encrypted, but not decrypted.



(a) Encryption in ECB mode

(b) Decryption in ECB mode

(c) Encryption in CTR mode

(d) Decryption in CTR mode

Figure 12-1 Encryption and Decryption Diagrams in ECB and CTR Modes

12.5 AES/SM4 Module Register Description

AES/SM4 module related register physical base address: 0x0040 c400

Table 12-2 List of AES/SM4 Module Registers

Name	Offset address	Description	Reset value
AES_SM4_CTRL	0x00	AES/SM4 control register	32h20

AES_SM4_INT_FG	0x04	AES/SM4 interrupt flag register	32h0
AES_SM4_KEY7	0x08	Key register 7	32hxxxxxxxx
AES_SM4_KEY6	0x0C	Key register 6	32hxxxxxxxx
AES_SM4_KEY5	0x10	Key register 5	32hxxxxxxxx
AES_SM4_KEY4	0x14	Key register 4	32hxxxxxxxx
AES_SM4_KEY3	0x18	Key register 3	32hxxxxxxxx
AES_SM4_KEY2	0x1C	Key register 2	32hxxxxxxxx
AES_SM4_KEY1	0x20	Key register 1	32hxxxxxxxx
AES_SM4_KEY0	0x24	Key register 0	32hxxxxxxxx
AES_SM4_IV3	0x28	Count value register 3	32hxxxxxxxx
AES_SM4_IV2	0x2C	Count value register 2	32hxxxxxxxx
AES_SM4_IV1	0x30	Count value register 1	32hxxxxxxxx
AES_SM4_IV0	0x34	Count value register 0	32hxxxxxxxx

AES/SM4 control register (AES_SM4_CTRL)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved.	0
16	bKEYE_ACT_IE	RW	Key expansion completion interrupt enable: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
[15:12]	Reserved	RO	Reserved.	0

[11:10]	MASK_Key_LEN	RW	Key length setting:
			00:128-bit;
			01:192-bit;
			10:256-bit;
			11: Reserved.
				0

9	bBCIPHER_MOD	RW	Block cipher mode selection bit: 1: CTR mode; 0: ECB mode.	0
8	bALGRM_MOD	RW	Algorithm mode selection bit: 1:AES; 0:SM4.	0
[7:6]	Reserved	RO	Reserved.	0
[5:4]	MASK_ED_CLK_PRE	RW	Encryption and decryption module clock frequency division factor, Calculation: EDclk=480M/ED_CLK_PRE. The minimum value is 2, and writing 1 is equivalent to turning off ECDC module operation clock.	10b
3	bEDMOD_SELT	RW	1: Decryption mode; 0: Encryption mode.	0
2	bRDDAT_ED_EN	RW	Enable to write SD data for encryption and decryption control bit: 1: Encryption and decryption; 0: No action.	0
1	bWRDAT_ED_EN	RW	Enable to read SD data for encryption and decryption control bit:	0

			1: Encryption and decryption; 0: No action.	
0	bKEYE_EN	RW	Key extension function enable control bit, high level pulse enable.	0

Note: When the bKEYE_EN bit is used, it needs to be set high and then low.

AES/SM4 interrupt flag register (AES_SM4_INT_FG)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved.	0
16	bKEYE_ACT_IF	RW1	Key extension completion interrupt flag bit, write 1 to clear it: 1: Key expansion completion trigger; 0: No event.	0
[15:0]	Reserved	RO	Reserved.	0

User key register group (AES_SM4_KEYn) (n=0-7)

Bit	Name	Access	Description	Reset value
[31:0]	AES_SM4_KEY7	RW	User key 223-256 bits.	x
[31:0]	AES_SM4_KEY6	RW	User key 192-223 bits.	x
[31:0]	AES_SM4_KEY5	RW	User key 160-191 bits.	x
[31:0]	AES_SM4_KEY4	RW	User key 128-159 bits.	x
[31:0]	AES_SM4_KEY3	RW	User key 96-127 bits.	x
[31:0]	AES_SM4_KEY2	RW	User key 64-95 bits.	x
[31:0]	AES_SM4_KEY1	RW	User key 32-63 bits.	x
[31:0]	AES_SM4_KEY0	RW	User key 0-31 bits.	x

Count value register group (AES_SM4_IVn) (n=0-3)

Bit	Name	Access	Description	Reset value
[31:0]	AES_SM4_IV3	RW	Count value of 96-127 bits.	x
[31:0]	AES_SM4_IV2	RW	Count value of 64-95 bits.	x
[31:0]	AES_SM4_IV1	RW	Count value of 32-63 bits.	x
[31:0]	AES_SM4_IV0	RW	Count value of 0-31 bits.	x

12.6 Data Storage Encryption and Decryption Applications

12.6.1 Data Encryption Function Configuration

1. Set AES/SM4 control register AES_SM4_CTRL: set bCLR_ALL_IF bit to 1, clear interrupt, select AES or SM4 algorithm, select ECB or CTR mode, and set key length. Note that SM4 algorithm only supports 128-bit key length;

2. Set user key register group and fill in the key. If ETC mode is used, it is also required to set the group value of count value register;

3. Set the bKEYE_EN of control register AES_SM4_CTRL, set it to 1 and then to 0, to enable the key expansion;

4. Query the interrupt flag register AES_SM4_INT_FG and wait for the key extension to complete the interrupt. Optionally, turn on the key expansion completion interrupt enable bit bKEYE_ACT_IE of control

register and wait for the interrupt to be triggered;

5. Clear the interrupt. Set the bEDMOD_SELT bit of the control register to 0, select the encryption mode, and set the bRDDAT_ED_EN bit to 1, to enable the encryption function when transmitting data from SRAM to SD, or set the bWRDAT_ED_EN bit to 1, to enable the encryption function when transmitting data from SD to SRAM.

12.6.2 Data Decryption Function Configuration

1. Set AES/SM4 control register AES_SM4_CTRL: set bCLR_ALL_IF bit to 1, clear interrupt, select AES or SM4 algorithm, select ECB or CTR mode, and set key length. Note that SM4 algorithm only supports 128-bit key length;

2. Set user key register group and fill in the key. If CTR mode is used, it is also required to set the group value of count value register;

3. Set the bKEYE_EN of control register AES_SM4_CTRL, set it to 1 and then to 0, to enable the key expansion;

4. Query the interrupt flag register AES_SM4_INT_FG and wait for the key extension to complete the interrupt. Optionally, turn on the key expansion completion interrupt enable bit bKEYE_ACT_IE of control register and wait for the interrupt to be triggered;

5. Clear the interrupt. Set the bEDMOD_SELT bit of the control register to 1, select the decryption mode, and set the bRDDAT_ED_EN bit to 1, to enable the encryption function when transmitting data from SRAM to SD, or set the bWRDAT_ED_EN bit to 1, to enable the decryption function when transmitting data from SD to SRAM.

Chapter 13 External Bus Interface (PARA)

13.1 Introduction to External Bus Interface

CH567 chip is equipped with an external bus controller that supports 8-bit bus width and address space ranges from 0x00800000 to 00807FFF, including read signal pin (PRD#), write signal pin (PWR#), 15-bit address pin (PA0-PA14) and 8-bit data pin (PB0-PB7). When the bus interface is not used, the above pins can be used as general purpose I/O ports. The interface supports static memory mapping components, including RAM, ROM, Flash, and some external I/O components, and it can automatically modify the pulse width of read/write signals as well as the creat and hold time of address and data.

External bus interface features:

(1). Support 8-bit bus width;

(2). Support up to 15-bit address bus width;

(3). Support static memory mapping components, including RAM, Flash and some external I/O components;

(4). Support dynamically modify pulse width, address and data setup and hold time of read/write signals, etc.;

13.2 Register Description

External bus interface related register physical start address: 0x0040 1000

Table 9-1 List of External Bus Interface Related Registers

Name	Offset address	Description	Reset value
R8_XBUS_CONFIG	0x10	External bus configuration register	8h00

External bus configuration register (R8_XBUS_CONFIG)

Bit	Name	Access	Description	Reset value
7	RB_XBUS_SETUP	RW	External bus data setup time: 1: 3 clock cycles; 0: 2 clock cycles.	0
6	RB_XBUS_HOLD	RW	External bus data hold time: 1: 3 clock cycles; 0: 2 clock cycles.	0

[5:4]	RB_XBUS_WIDTH	RW	External bus RD/WR valid time:
			00: 3 clock cycles;
			01: 5 clock cycles;
			10: 9 clock cycles;
			11: 16 clock cycles.
				0

[3:2]	RB_XBUS_ADDR_OE	RW	External bus address output enable: 00: No bus address output; 01: Address line PA[5:0]; 10: Address line PA[9:0]; 11: Address line PA[14:0].	0
1	Resered	RO	Reserved.	0
0	RB_XBUS_ENABLE	RW	External bus enable bit: 1: Enable;	0

			0: Disable.	

13.3 External Bus Interface Application

13.3.1 External Bus Interface

The external bus interface includes 1 read signal pin PRD# (PB8), 1 write signal pin PWR# (PB9), 15 address pins A0-A14 (PA0-PA14) and 8 data pins D0-D7 (PB0-PB7). The interface supports 8-bit data bus width.

8-bit wide memory interface:

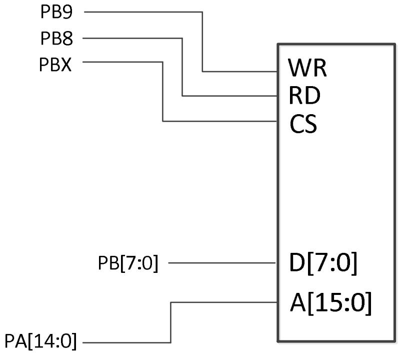


Figure 9-1 8-bit Bus Interface Connected to 8-bit Memory Chip

13.3.2 Typical Bus Timing

The following figure shows a typical external bus interface read/write access timing, Fsys is the bus clock frequency, RD is the read signal pin, and WR is the write signal pin.

For example, after setting RB_XBUS_WIDTH = 01b, RB_XBUS_HOLD = 0b, RB_XBUS_SETUP = 0b, the timing diagram is as follows:

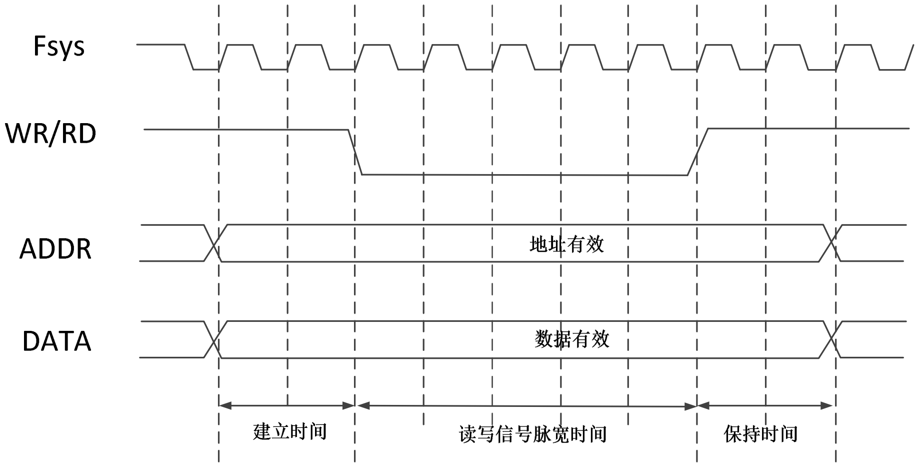


Figure 9-2 External Line Interface Read/Write Access Timing Diagram

The RB_XBUS_SETUP bit in the figure above is the data setup time. When it is set to 0, the external bus setup time is 2 clock cycles (Tsys). When it is set to 1, the external bus setup time is 3 clock cycles (Tsys). The RB_XBUS_HOLD bit is data hold time. When it is set to 0, the data hold time is 2 clock cycles (Tsys). When it is set to 1, the data hold time is 3 clock cycles (Tsys). RB_XBUS_WIDTH is the effective pulse width of bus data read and write, and 3 clocks, 5 clocks, 9 clocks or 16 clocks are optional.

13.3.3 External Bus Configuration

The main configuration steps are shown below:

(1). Configure the external bus configuration register (R8_XBUS_CONFIG), set the RB_XBUS_ENABLE bit to 1 to enable the external bus interface, and set the address bit control domain RB_XBUS_ADDR_OE as needed, to select the required addressing range;

(2). Configure the external bus configuration register (R8_XBUS_CONFIG), set the RB_XBUS_WIDTH bit control domain, to select the bus read/write clock pulse width, set the RB_XBUS_SETUP bit and RB_XBUS_HOLD bit, to select the bus setup time and bus hold time;

(3) Configure the corresponding pin direction of external bus interface.

The main configuration programs are shown below:

```
/* Configure external bus interface related registers, enable parallel port bus function, 15 address lines, bus read/write clock pulse width 5 clocks, bus setup time and hold time of 2 clocks each */
R8_XBUS_CONFIG = 0x1d;
/* Configure the corresponding pins of the external bus interface */
R32_PA_DIR |= 0x7FFF; // Can only use part of the address pins
R32_PB_DIR |= ( PWR | PRD);
```

Chapter 14 Interrupt

14.1 Interrupt Controller

CH567 chip supports multiple interrupt sources, with a total of 16 peripheral interrupt sources, including UART, SPI, TMR, USB, GPIO, etc.

Table 14-1 Interrupt Vector

Default priority order	Interrupt No.	Priority	Name	Description
High priority ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ Low priority	0	Level 2, can be set	SOFT	Software interrupt
	1	Level 2, can be set	TMR0	Timer0 interrupt
	2	Level 2, can be set	GPIO	GPIO port interrupt
	3	Level 2, can be set	SPI0	SPI0 interrupt
	4	Level 2, can be set	USB0	USB0 interrupt
	5	Level 2, can be set	TMR1	Timer1 interrupt
	6	Level 2, can be set	TMR2	Timer2 interrupt
	7	Level 2, can be set	UART0	UART0 interrupt
	8	Level 2, can be set	USB1	USB1 interrupt
	9	Level 2, can be set	SDC	SD controller interrupt
	10	Level 2, can be set	ECDC	Encryption/decryption controller interrupt
	11	Level 2, can be set	LED	LED controller interrupt
	12	Level 2, can be set	SPI1	SPI1 interrupt
	13	Level 2, can be set	UART1	UART1 interrupt
	14	Level 2, can be set	UART2	UART2 interrupt
	15	Level 2, can be set	UART3	UART3 interrupt

14.2 Interrupt Instruction

1) To use the interrupt function of CH567 chip, you need to add the following functions to the software code to enable the system interrupt control.

```
        void Interrupt_init(void);
```

2) Enable the corresponding interrupt control bit of interrupt enable register of corresponding peripheral module, to trigger the interrupt under the corresponding trigger condition.

3) Refer to the following example to write interrupt function. Taking TMR0 as example, the interrupt number is 1:

```
__attribute__( ( interrupt ( "id=1" ) ) ) void TIME0_Deal()
{
    static UINT16 j=0;
    if(R8_TMR0_INT_FLAG & RB_TMR_IF_CYC_END)
    {
        j++;
        if(j>300){
            printf("*\n");
            j = 0;
        }
        R8_TMR0_INT_FLAG |= RB_TMR_IF_CYC_END;   //Clear flag
    }
}
```

Chapter 15 Parameters

15.1 Absolute Maximum Value

Critical value or exceeding the absolute maximum value may cause the chip to work abnormally or even be damaged.

Table 15-1 Absolute Maximum Value Parameters

Name	Parameter description		Min.	Max.	Unit
TA	Ambient temperature during operation	V33REG=3.3V V33USB=3.3V V33IO1/2/3=3.3V	-40	85	℃
TS	Ambient temperature during storage		-55	125	℃
V33REG	System supply voltage (V33REG connects to power supply, GND to ground)		-0.4	4.2	V
V33USB	USB0 supply voltage (V33USB connects to power supply, GND to ground)		-0.4	4.2	V
V33IO1	Peripheral group 1 supply voltage (V33IO1 connects to power supply, GND to ground)		-0.4	4.2	V
V33IO2	Peripheral group 2 supply voltage (V33IO2 connects to power supply, GND to ground)		-0.4	4.2	V
V33IO3	Peripheral group 3 supply voltage (V33IO3 connects to power supply, GND to ground)		-0.4	4.2	V
VIO0	Voltage on input or output pins of V33USB power domain		-0.4	V33USB+0.4	V
VIO1	Voltage on input or output pins of V33IO1 power domain		-0.4	V33IO1+0.4	V
VIO2	Voltage on input or output pins of V33IO2 power domain		-0.4	V33IO2+0.4	V
VIO3	Voltage on input or output pins of V33IO3 power domain		-0.4	V33IO3+0.4	V

15.2 Electrical Parameters

Test conditions: TA=25℃, V33REG=3.3V, V33USB=3.3V, V33IO1/2/3=3.3V, Fsys=96MHz.

Table 15-2 Electrical Parameters

Name	Parameter description		Min.	Typ.	Max.	Unit
V33REG	System supply voltage	V33REG	2.7	3.3	3.6	V
V33USB	USB0 supply voltage	V33USB	3.0	3.3	3.6	V
V33IO1	Peripheral group 1 supply voltage	V33IO1	1.6	3.3	3.6	V
V33IO2	Peripheral group 2 supply voltage	V33IO2	1.6	3.3	3.6	V
V33IO3	Peripheral group 3 supply voltage	V33IO3	3.0	3.3	3.6	V
ICC	Total supply current during operation		20	55	150	mA
ISLP	Supply current when at low power status (I/O pin output with no-load or input with pull-down)		240	270	350	uA
VIL	Low-level input voltage (V33IO=3.3V)		-0.4	-	0.7	V
VIH	High-level input voltage (V33IO=3.3V)		2.0	-	V33IO+0.4	V

VIL18	Low-level input voltage (V33IO1/2=1.8V)	-0.4	-	0.5	V
VIH18	High-level input voltage (V33IO1/2=1.8V)	1.2	-	V33IO+0.4	V
VOL	Low level output voltage (6mA draw current)	-	-	0.4	V
VOH	High level output voltage (5mA output current)	V33IO-0.4	-	-	V
IUP	Input current at the input terminal with built-in pull-up resistor	25	45	80	uA
IDN	Input current at the input terminal with built-in pull-down resistor	-25	-45	-80	uA
Vpot	Voltage threshold for VCORE core power supply power-on reset	0.6	0.7	0.8	V

15.3 Static Current of Function Module

Test conditions: TA=25°C, V33REG=3.3V, V33USB=3.3V, V33IO1/2/3=3.3V.

Table 15-3 Dynamic current of Function Module

Frequency Function Module	30M	60M	96M	120M	Unit
USB0/USB1	0.2	0.4	0.65	0.79	mA
USB0-Phy	10				mA
USB1-Phy	15				mA
SDC-96MHz	12.45	13.53	14.84	15.65	mA
SDC-48MHz	6.28	7.39	8.64	9.49	mA
SDC-24MHz	4.16	5.22	6.5	7.32	mA
ECDC-240MHz	15.64	16.88	17.72	19.3	mA
ECDC-160MHz	10.99	12.25	13.79	14.69	mA
TMR+UART+SPI+PWM	0.17	0.33	0.51	0.7	mA
PLL	7				mA
Core+BUS+DMA	24	27	31	34	mA

15.4 Timing Parameters

Test conditions: TA=25°C, V33REG=3.3V, V33USB=3.3V, V33IO1/2/3=3.3V, Fsys=96MHz.

Table 15-4 Timing Parameters

Name	Parameter description	Min.	Typ.	Max.	Unit
Trst	Effective signal width of external reset input RST#	50	2*Tsys	-	ns
Tpro	Reset delay after power-on reset	22	32	50	mS
Tsro	Reset delay after external/software reset input + load time	8	8.8	10	mS
TWAK	Wake-up time to exit from low-power state	0.2	1	5	mS

Chapter 16 Package

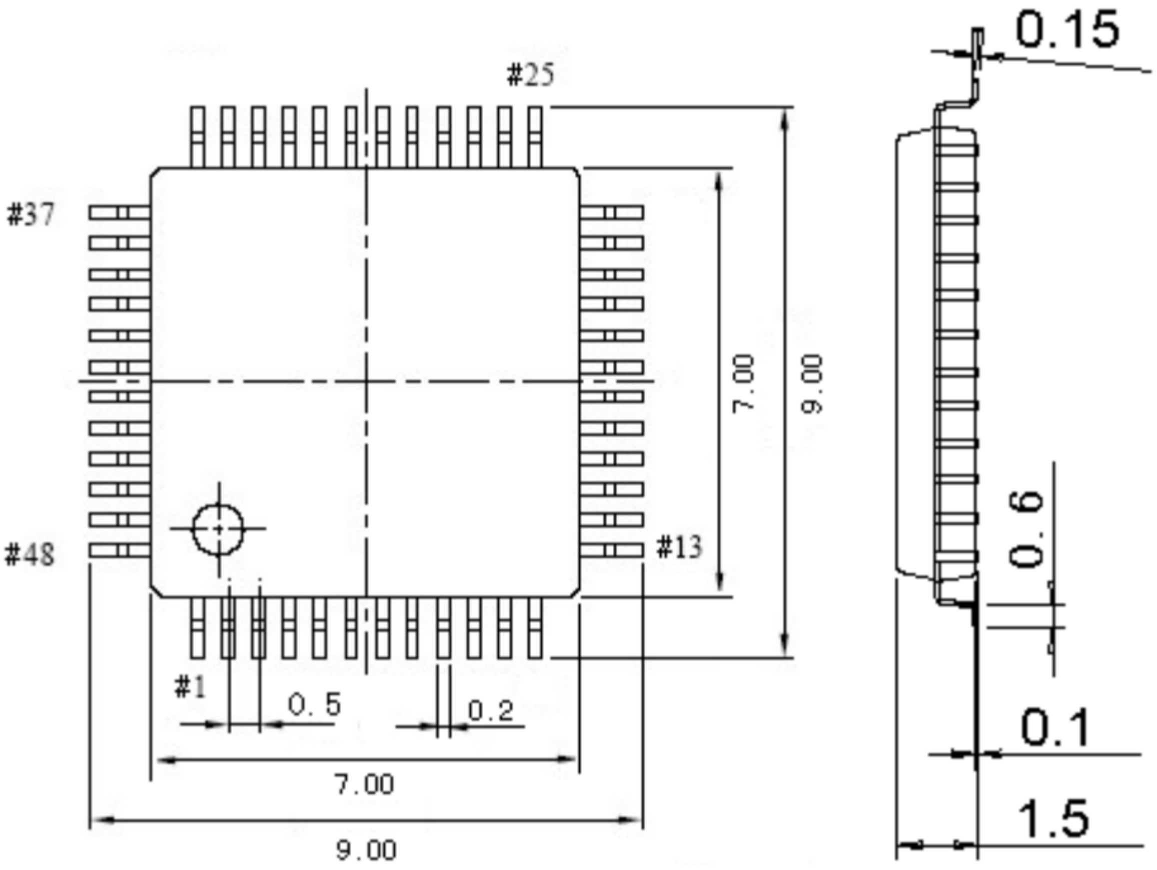
Chip package

Package	Width Of Plastic	Pitch Of Pin		Instruction Of Package	Ordering Information
LQFP-48	7*7mm	0.5mm	19.7mil	Standard LQFP48 pin patch	CH567L

Remarks :

The unit of dimension is mm (millimeters)

The pin center spacing is the nominal value, there is no error, and the other dimension error is not greater than ±0.5mm.



Chapter 17 Modification Record

Version	Date	Description
V1.0	May 10, 2017	First release
V1.1	October 29, 2018	Some error descriptions are modified