WM_W800_Register Manual

V2.1

Beijing Lianshengde Microelectronics Co., Ltd. (winner micro)

Address: 18th Floor, Yindu Building, No. 67, Fucheng Road, Haidian District, Beijing

Tel: +86-10-62161900

Company website: www.winnermicro.com

Document modification record

Version revision time		revision history	author	audit
V0.1	2019-09-25	create	chengsheng	
V0.2	2020-05-13	Revised the legacy description error;	Chengsheng	
V1.0	2020-7-20	Update digital function interface	Ray	
V2.0	2020-8-20	Improve high-speed interface function	Ray	
V2.1	2020-09-08	Added description of SD_ADC module	chengsheng	

content

Figure Catalog

table directory

1 Introduction

1.1 Purpose of writing

The W800 chip is an embedded Wi-Fi SoC chip launched by Lianshengde Microelectronics. The chip is highly integrated, requires less peripheral devices, and is cost-effective

high. It is suitable for various smart products in the field of IoT (smart home). Highly integrated Wi-Fi and Bluetooth 4.2 Combo functions are its main features;

In addition, the chip integrates XT804 core, built-in QFlash, SDIO, SPI, UART, GPIO, I²C, PWM, I²S, 7816, LCD,

Interfaces such as Touch Sensor, support a variety of hardware encryption and decryption algorithms. In addition, the chip MCU contains a security kernel that supports code security permission settings.

The whole system supports firmware encrypted storage, firmware signature, security debugging, security upgrade and other security measures to improve product security features.

This document mainly describes the internal structure of the W800 chip, information on each functional module and detailed register usage information;

The main reference for the application. There are open source implementations of various functions in the SDK provided by Lianshengde Microelectronics, and developers can refer to the corresponding drivers

Programs, application examples to increase understanding of chip functions and register descriptions. There are no register descriptions for the Wi-Fi/BT part of this document.

1.2 References

For information on W800 chip package parameters, electrical characteristics, RF parameters, etc., please refer to "W800 Chip Product Specifications";

The W800 chip integrates the ROM program. The ROM program provides functions such as downloading firmware, MAC address reading and writing, and Wi-Fi parameter reading and writing.

For information, please refer to "WM_W800_ROM Function Brief";

The W800 chip has a built-in 2Mbytes QFlash memory, which is used as a storage space for codes and parameters. This document provides basic QFlash operations

for information. If there are requirements beyond the scope of this document, you need to refer to the QFlash manual;

W800 chip adopts Hangzhou Pingtou Ge XT804 core, 804 related function introduction, development materials, etc. can refer to Pingtou Ge company release information;

For more information, please refer to WinnerMicro's website (http://www.winnermicro.com/).

2 Features

ÿ Chip Packaging

- ÿ QFN32 package, 4mm x 4mm.

ÿ Chip integration

- ÿ Integrated XT804 processor, up to 240MHz
- ÿ Integrated 288KB SRAM
- ÿ Integrated 2MB FLASH
- ÿ Integrated 8-channel DMA controller, supports 16 hardware applications, and supports software linked list management
- ÿ Integrated PA/LNA/TR-Switch
- ÿ Integrated 32.768KHz clock oscillator
- ÿ Integrated voltage detection circuit
- ÿ Integrated LDO
- ÿ Integrated power-on reset circuit

ÿ Chip interface

- ÿ Integrate 1 SDIO2.0 Device controller, support SDIO 1-bit/4-bit/SPI three operation modes, working clock range 0~50MHz

ÿ Integrate 1 SDIO 2.0 HOST controller, support SDIO and SD card operation, working clock range 0~50MHz

ÿ Integrate a QSPI PSRAM interface, support a maximum capacity of 64MB PSRAM, and a maximum operating clock frequency of 80MHz;

ÿ Integrate 5 UART interfaces, support RTS/CTS, baud rate range 1200bps~2Mbps

ÿ Integrate a high-speed SPI slave interface, the operating clock range is 0~50MHz

ÿ Integrate 1 SPI master/slave interface, the working clock of the master device is up to 20MHz, and the slave device supports up to 6Mbps data transfer rate

ÿ Integrate an I2C controller, support 100/400Kbps rate

ÿ Integrated PWM controller, support 5-channel PWM

Single output or 2 PWM inputs. Maximum output frequency 20MHz, maximum input frequency 20MHz

ÿ Integrated duplex I2S controller, support 32KHz to 192KHz I2S interface codec

ÿ Integrate one 7816 interface, compatible with UART interface, support ISO-7816-3 T=0.T=1 mode; support

EVM2000 Protocol

ÿ Support a variety of hardware encryption and decryption modes, including

RSA/AES/RC4/DES/3DES/RC4/SHA1/MD5/CRC8/CRC16/CRC32/TRNG

ÿ Integrate one differential, or two single-ended 16bit ADC interfaces;

ÿ Integrate 11-way Touch Sensor;

ÿ Support up to 17 GPIO ports, each IO port has

Rich reuse relationships. With input and output configuration options.

ÿ WIFI protocol and function

ÿSupport GB15629.11-2006, IEEE802.11 b/g/n;

ÿ Support WMM/WMM-PS/WPA/WPA2/WPS

ÿSupport WiFi Direct;

ÿ Support EDCA channel access mode;

ÿ Support 20/40M bandwidth working mode;

ÿ Support STBC, GreenField, Short-GI, support reverse transmission;

ÿ Support RIFS frame interval;

ÿ Support AMPDU, AMSDU;

ÿ Support 802.11n MCS 0~7, MCS32 physical layer transmission rate gear, the transmission rate is up to 150Mbps;

ÿ Support HT-immediate Compressed BlockAck, normal ACK, no ACK response mode;

ÿ Support CTS to self;

ÿ Support AP function; AP and STA are used at the same time;

ÿ In the BSS network, multiple multicast networks are supported, and each multicast network supports different encryption methods.

32 multicast networks and network access STA encryption;

ÿ When the BSS network supports as an AP, the total number of supported sites and groups is 32;

ÿ Receive Sensitivity:

ÿ 20MHz MCS7@-71dBm@10%PERÿ

ÿ 40MHz MCS7 @ -67dBm @ 10% PER ÿ

ÿ 54Mbps @ -73dBm @ 10% PER ÿ

ÿ 11Mbps@-86dBm@8%PERÿ

ÿ 1Mbps @ -96dBm @ 8% PER ÿ

ÿ Support a variety of different received frame filtering options;

ÿ Support monitoring function;

ÿ Bluetooth protocol and function

ÿ Integrated Bluetooth baseband processor/coprocessor, support BT/BLE4.2 protocol

ÿ Support various rates of DR/EDR;

ÿ Support BLE 1Mbps rate;

ÿ Power supply and power consumption

ÿ 3.3V single power supply;

ÿ Support Wi-Fi power saving mode power management;

ÿ Support work, sleep, standby, shutdown working modes;

ÿ Standby power consumption is less than 15uA;

3 Overview

This chip is a SOC chip that supports multi-interface and multi-protocol wireless local area network 802.11n (1T1R). The SOC chip integrates

RF Transceiver, CMOS PA Power Amplifier, Baseband Processor/Media Access Control, SDIO, SPI,

Low-power WLAN chip with interfaces such as UART and GPIO.

W800 chip supports GB15629.11-2006, IEEE802.11 b/g/n protocol, and supports STBC, Green Field,

Short-GI, Reverse Transmission, RIFS Interframe Interval, AMPDU, AMSDU, T-immediate Compressed Block Ack,

Rich protocols and operations such as normal ACK, no ACK, and CTS to self.

The W800 chip integrates the RF transceiver front-end, A/D and D/A converters. It supports DSSS (Direct Sequence Spread Spectrum) as well as OFDM

(Orthogonal Frequency Division Multiplexing) modulation mode, with data descrambling capability, supports a variety of different data transmission rates. in the analog front end of the transceiver

The equipped transceiver AGC function enables the system-on-a-chip to obtain the best performance. The W800 chip also includes a built-in enhanced signal monitor,

The influence of multipath effect can be largely eliminated.

In terms of security, the W800 chip not only supports the national standard WAPI encryption, but also supports the international standard WEP, TKIP, CCMP encryption,

These hardware components enable data transmission systems based on the chip to obtain data similar to those of non-encrypted communications during secure communications.

data transmission performance.

In addition to supporting the energy-saving operations specified by IEEE802.11 and Wi-Fi protocols, the W800 chip also supports user-customized energy-saving solutions. chip

It supports four working modes: work, sleep, standby and shutdown, so that the entire system can achieve low power consumption, and it is convenient for users to adapt to their own use fields.

different scenarios for energy saving.

The W800 chip integrates a high-performance 32-bit embedded processor, a large number of memory resources, and a wealth of peripheral interfaces, which are convenient for users.

It is easy to apply the chip to the secondary development of a specific product.

The W800 chip supports AP function, which can realize the establishment of 5 SSID networks at the same time, and realize the function of 5 independent APs. Support for creating multiple

Multicast network function. It can realize the function of establishing a BSS network as an AP while joining other networks as a STA.

The W800 chip supports the WPS method, allowing users to implement an encrypted complete network with one-click operation to ensure the security of information

sex.

The multi-function and high integration of W800 chip ensures that the WLAN system does not need too many off-chip circuits and external memory.

4 Chip Structure

4.1 Chip structure

The following figure describes the overall structure of the W800 chip, the core part includes XT804 CPU, 288KB SRAM and 20KB ROM storage space.

As the constant power supply module of the chip, the PMU part provides power-on sequence management, start-up clock, and real-time clock functions. Provides a wealth of peripherals function and hardware encryption and decryption functions. The Wi-Fi part integrates MAC, BB and RF.

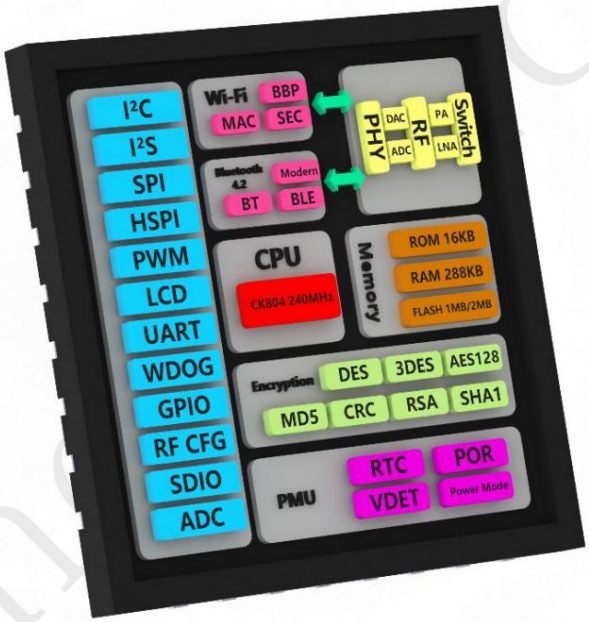


Figure 1 W800 chip structure diagram

4.2 Bus Structure

The W800 chip consists of a two-level bus, as shown in the figure below

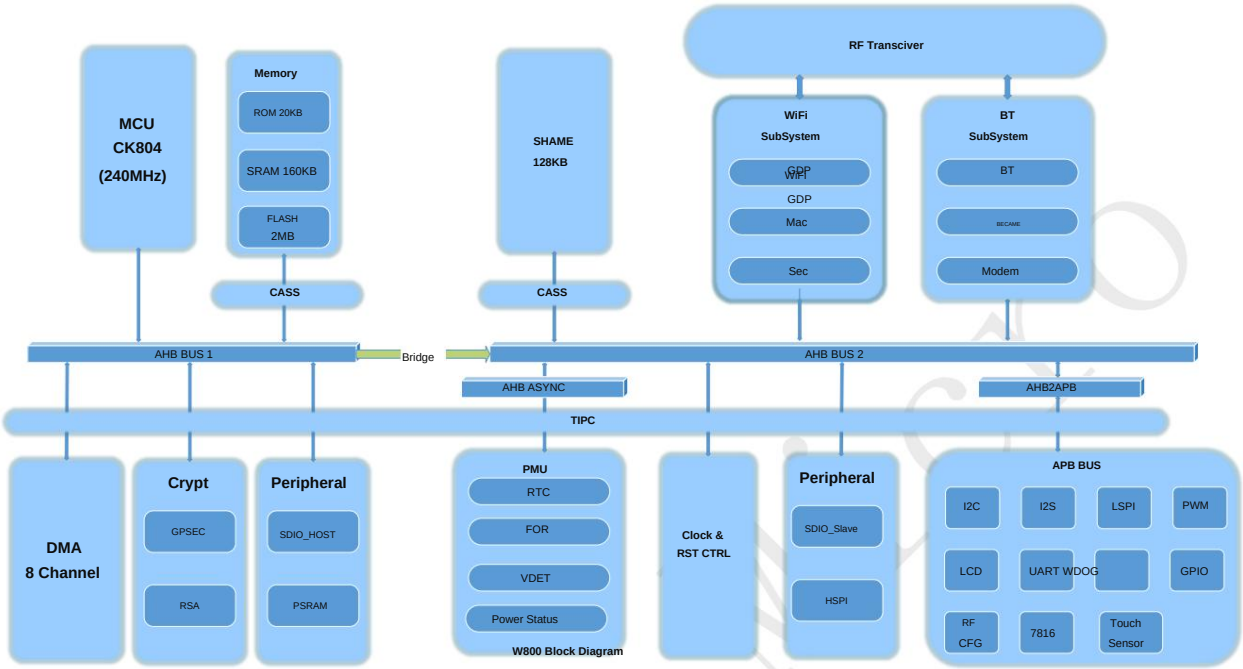


Figure 2 W800 bus structure diagram

(1) AHB-1 bus

This bus has four masters - XT804, DMA, GPSEC and 5 slaves.

XT804 is a 32-bit energy-efficient embedded CPU core oriented to the control field. It adopts 16/32-bit mixed coding instruction system.

A streamlined and efficient 3-stage pipeline.

The XT804 provides a variety of configurable functions, including hardware floating point unit, on-chip cache, DSP acceleration unit, trusted protection technology,

On-chip tightly coupled IP, etc., users can configure according to application needs. In addition, XT804 provides multi-bus interface, supports system bus, finger

Flexible configuration of the order bus and data bus. XT804 has made special acceleration for interrupt response, and the interrupt response delay is only 13 cycles.

The bus clock operates at the fastest frequency of 240MHz and can be configured to 240/160/120/80/40MHz, or lower.

Table 1 AHB-1 bus master list

main device	Features
CPU	Complete chip register configuration, memory management and use, and complete 802.11MAC protocol. Highest Operating frequency 240MHz
DMA supports an independent 8-channel DMA module with a linked list structure, and supports 16 on-chip hardware DMA request sources.	
GPSEC	Universal encryption module, supports DES/3DESSHA1/AES/MD5/RC4/CRC/PRDN. automatic completion Data blocks in the specified memory space are encrypted and written back.

Table 2 AHB-1 bus slave device list

Slave function	
ROOM	ROM is used to store the initialization firmware after the CPU is powered on. It mainly completes the initial configuration of the chip register space and other work. After completing the above work, the CPU control The control is given to the firmware stored in FLASH.
AHB2AHB	Complete the conversion of CPU bus clock domain to BusMatrix2 bus clock domain master access. Require The clock domains must be of the same source, and the ratio of the CPU clock to the BusMatrix2 clock frequency is M:1, M is an integer greater than or equal to 1.
FLASH stores firmware code and operating parameters.	
SRAM 160KB can be used to store instructions or data, and firmware can use this memory as needed.	
RSA	Supports RSA encryption and decryption operations up to 2048bit

GPSEC	general encryption/decryption module, supports SHA1/AES/MD5/RC4/CRC/TRNG. autocomplete Encrypt/decrypt and write back data blocks in a given memory space.
SDIO_HOST	SDIO 2.0 standard SDIO HOST controller; SDIO interface peripherals can be accessed through this interface. The SDIO interface clock is obtained by dividing the bus clock and supports up to 50MHz.
PSRAM_CTRL	PSRAM controller for QSPI interface. An external PSRAM can be accessed through this controller. QSPI connection The port clock is obtained by dividing the frequency of the bus clock, and it supports up to 80MHz clock.

(2) AHB-2 bus

This bus has 4 master devices and 3 slave devices. Using the crossbar connection structure, it can realize different master devices to different slave devices. access at the same time, thereby increasing the bandwidth. The bus clock operates at the fastest frequency of 40MHz and can be configured to be lower as required.

Table 3 AHB-2 bus master list

main device	Features
MAC	802.11MAC control protocol processing module. Operations on the bus mainly include sending read data Data, receive write data, and send completion descriptor write-back operations.
SEC	The security module completes the encryption, decryption and movement of the sent and received data. When sending, the data and The MAC descriptor is moved to the specified location and encryption is completed; when receiving, the received data and The MAC receives the descriptor and moves to the specified location and completes the decryption.
AHB2AHB	Transition of bus master access from the AHB-1 bus to the AHB-2 bus.
SDIO/HSPI	Connect the host to the chip through the SDIO2.0 device controller or the high-speed SPI slave device controller. Accesses translate to AHB bus signals and access content memory and register space.

Each master device adopts a fixed priority, and the priority decreases from top to bottom.

Table 4 AHB-2 bus slave device list

slave device	Features
SRAM 128KB	Used to store upstream and downstream data buffer, SDIO/SPI/UART interface uses this RAM as data cache
Configuration register	configuration space, high-speed module configuration registers are uniformly addressed here.
	All low-speed modules of APB access the space, and various low-speed modules are connected using APB bus.
BT_CORE	Bluetooth controller.

4.3 Clock Structure

W800 uses 24/40MHz crystal as SoC clock source, built-in 1 DPLL output 480MHz, supply

Used by CPU, system bus, data bus and WiFi system; on-chip additional built-in 32.768KHZ RC oscillator for PMU

and LCD module use. An overview of the clock structure is shown in the figure below.

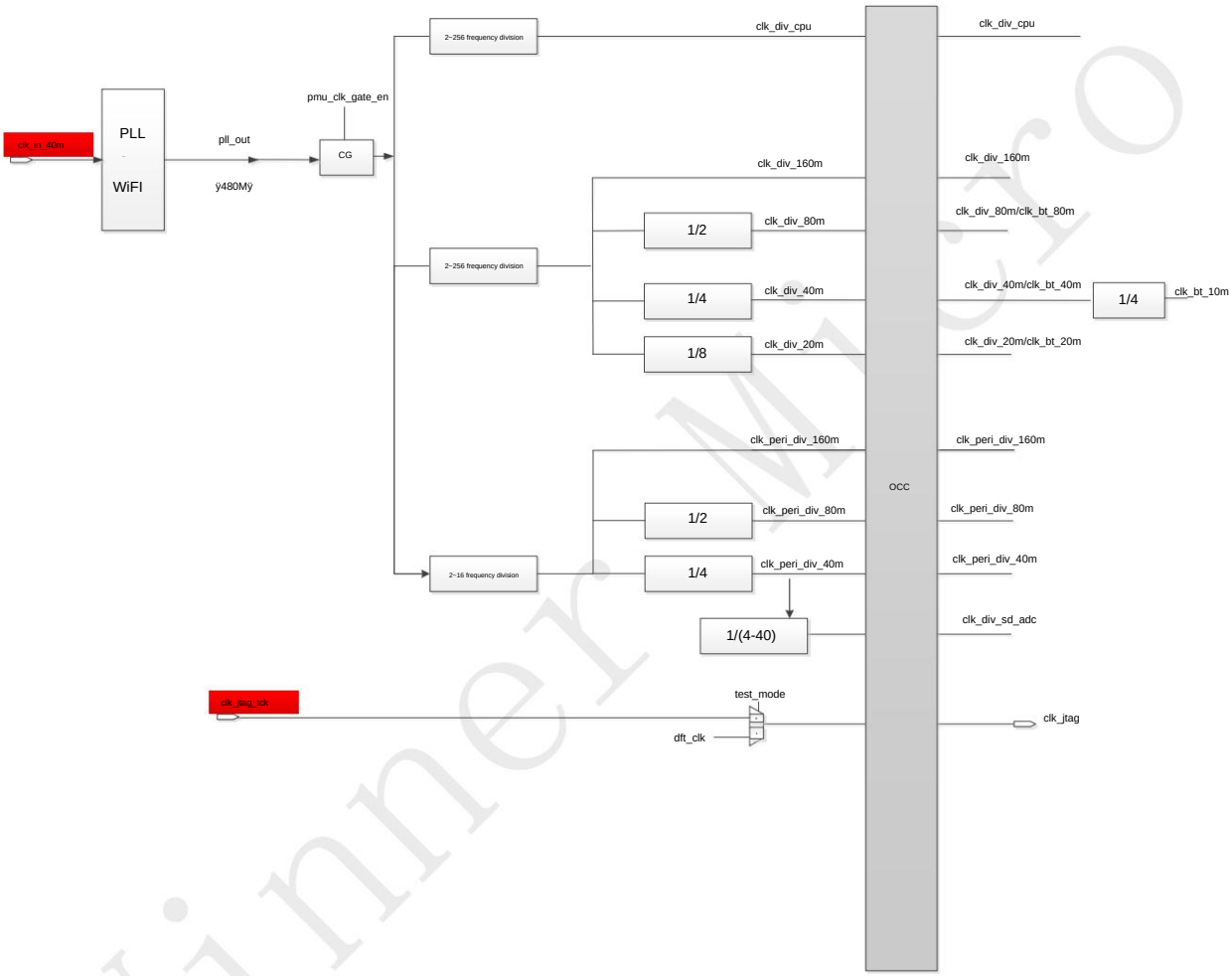
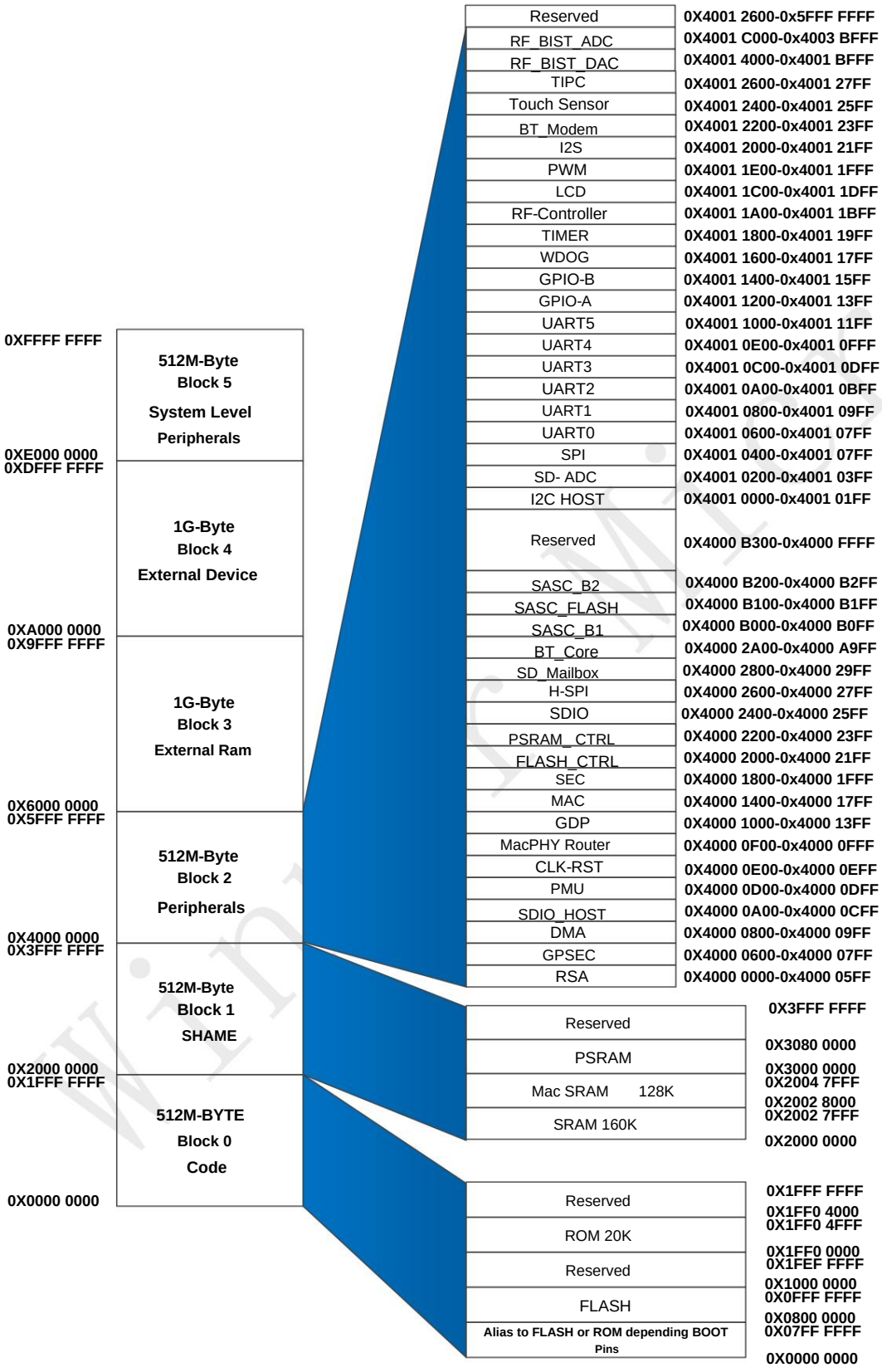


Figure 3 W800 clock structure

4.4 Address space



W800 Address Mapping

XT804 supports 4G storage space, which is divided into 6 blocks as shown in the figure above, which are code area, memory area, on-chip peripherals, and off-chip storage.

area, off-chip peripherals and system peripherals area. According to requirements, the on-chip storage space of w800 is mapped to the first three areas as shown in Figure 3.

table 5 Detailed division of bus device address space

bus from equipment	BootMode=0	Address space breakdown	Remark
ROM 0x0000 0000 ~ 0x0004 FFFF			Store the solidified firmware code
FLASH 0x0800 0000 ~ 0x0FFF FFFF			Stored as a dedicated instruction device.
SRAM 0x2000 0000 ~ 0x2002 7FFFF			Firmware memory and instruction storage Area
Mac RAM 0x2002 8000 ~ 0x2004 7FFFF			SDIO/H-SPI/UART data cache
PSRAM	0x3000 000~0x30800000		Peripheral memory
CONFIG 0x4000 0000 ~ 0x4000 2FFF		0x4000 0000 ~ 0x4000 05FF	RSA configuration space
		0x4000 0600 ~ 0x4000 07FF	GPSEC configuration space
		0x4000 0800 ~ 0x4000 09FF	DMA configuration space
		0x4000 0A00 ~ 0x4000 0CFF	SDIO_HOST configuration is empty between
		0x4000 0D00 ~ 0x4000 0DFF	PMU configuration space

		0x4000 0E00 ~ 0x4000 0EFF	Clock and Reset configuration space
		0x4000 0F00 ~ 0x4000 0FFF	MacPHY Router configuration space
		0x4000 1000 ~ 0x4000 13FF	BBP configuration space
		0x4000 1400 ~ 0x4000 17FF	MAC configuration space
		0x4000 1800 ~ 0x4000 1FFF	SEC configuration space
		0x4000 2000 ~ 0x4000 21FF	FLASH Controller configuration space
		0x4000 2200 ~ 0x4000 23FF	PSRAM_CTRL configuration space
		0x4000 2400 ~ 0x4000 25FF	SDIO Slave configuration is empty between
		0x4000 2600 ~ 0x4000 27FF	H-SPI configuration space
		0x4000 2800 ~ 0x4000 29FF	SD Wrapper configuration space
		0x4000 2A00 ~ 0x4000 A9FF	BT Core configuration space
		0x4000 B000 ~ 0x4000 B0FF	CASS-B1 Level 1 Bus Memory Security Configuration Mode piece
		0x4000 B100 ~ 0x4000 B1FF	CASS-Flash Flash Security Configuration Module

		0x4000 B200 ~ 0x4000 B2FF	CASS-B2 Secondary bus memory security configuration module piece
APB	0x4001 0000 ~ 0x 4001 C000	0x4001 0000 ~ 0x4001 01FF	I2C master
		0x4001 0200 ~ 0x4001 03FF	Sigma ADC
		0x4001 0400 ~ 0x4001 07FF	SPI master
		0x4001 0600 ~ 0x4001 07FF	UART0
		0x4001 0800 ~ 0x4001 09FF	UART1
		0x4001 0A00 ~ 0x4001 0BFF	UART2
		0x4001 0C00 ~ 0x4001 0DFF	UART3
		0x4001 0E00 ~ 0x4001 0FFF	UART4
		0x4001 1000 ~ 0x4001 11FF	UART5
		0x4001 1200 ~ 0x4001 13FF	GPIO-A
		0x4001 1400 ~ 0x4001 15FF	GPIO-B
		0x4001 1600 ~ 0x4001 17FF	WatchDog
		0x4001 1800 ~ 0x4001 19FF	Timer
		0x4001 1A00 ~ 0x4001 1BFF	RF_Controller
		0x4001 1C00 ~ 0x4001 1DFF	LCD
		0x4001 1E00 ~ 0x4001 1FFF	PWM

		0x4001 2000 ~ 0x4001 22FF	I2S
		0x4001 2200 ~ 0x4001 23FF	BT modem
		0x4001 2400 ~ 0x4001 25FF	Touch Sensor
		0x4001 2600 ~ 0x4001 25FF	TIPC Interface Security Settings
		0x4001 4000 ~ 0x4000 BFFF	RF_BIST DAC transmit RAM
		0x4001 C000 ~ 0x4003 BFFF	RF_BIST ADC receive RAM
		0x4001 3C00 ~ 0x5FFF FFFF	RSV

4.4.1 SRAM

W800 has built-in 288KB SRAM. Among them, 160KB is mounted on the first-level AHB bus, and 128KB is mounted on the second-level AHB bus. CPU

Devices on the first-level bus can access all memory areas, but devices on the second-level bus can only access 128KB of memory on the second-level bus.

4.4.2 Flash

4.4.2.1 QFlash

W800 integrates 2MBytes QFlash inside. The XIP method is implemented on the QFlash through the integrated 32KB cache inside the chip.

sequence. When the program is running, the CPU first reads the instructions from the Cache.

QFlash reads the instruction and stores it in the Cache. Therefore, when the continuous running code size is less than 32K, the CPU will not need to read from QFlash

Fetch instructions, at which time the CPU can run at a higher frequency. The above method is the operation mode of the read command, and the RO segment of the entire Image will be

Operate in this way. This process requires no user intervention.

QFlash can also store data. When the user program needs to read and write data in QFlash, it needs to use the built-in QFlash controller.

Operation, QFlash provides the corresponding address, instruction and other registers to assist the user to achieve the desired operation. For specific description, please refer to QFlash

The controller corresponds to the chapter.

Users need to pay attention that when the program reads or writes data, there is no need to perform state judgment, wait and other operations, because the QFlash control

The device itself will judge. When the QFlash controller returns, the read or write is complete.

4.4.2.2 SPI Flash

In addition to supporting 6PIN QFlash interface (built-in PIN, not packaged), W800 chip also supports low-speed SPI interface access. The SPI

The maximum operating frequency of the interface can reach 20MHz, and it supports the master-slave function. For a detailed description, please refer to the corresponding chapter of the SPI interface.

4.4.3 PSRAM

W800 has a built-in PSRAM controller with SPI/QSPI interface, supports external PSRAM device access with a maximum capacity of 64Mb, and provides bus mode of PSRAM read, write and erase operations. The maximum read and write speed is 80MHz. When the storage capacity needs to be expanded, the off-chip PSRAM can be used to expand Fill code storage space or data storage space. PSRAM also supports XIP execution of programs, and CPU Cache also supports cache data in PSRAM.

4.5 Startup Configuration

After the W800 chip is powered on, the CPU will start to execute the firmware in the ROM and load the user image at the specified address in the Flash.

When the ROM firmware starts to run, it will read the BootMode (PA0) pin, and judge to enter the boot state according to the signal of the pin:

Table 6 Startup configuration

BootMode	start condition	boot mode
high		normal startup process
Low	Continuous <30ms, quick test mode is invalid	normal startup process
	Last >=30ms	Enter functional mode
Note: Test mode: chip test function, user cannot operate. Function mode: Enter the basic functions implemented by ROM, such as: downloading firmware, programming MAC address, etc. For details, please refer to		

"WM_W800_ROM Function Brief.pdf"

Typically, the BootMode pins should be used in production or debug stages. During the production phase, the user continuously pulls the BootMode pin

If it is low for more than 30ms, it enters the function mode, and can quickly burn the Flash.

In the scenario of product rework or repair, the chip does not enter the "highest security level" (for a description of the security level, please refer to

"WM_W800_ROM Function Brief"), you can enter the function mode through this pin, erase the old Image, write the new

Imageÿ

In the debugging stage, no matter what the firmware is faulty, you can enter the serial port by continuously pulling the BootMode pin down for more than 30ms

Download function, burn new firmware.

5 Clock and reset module

5.1 Function overview

The clock and reset module completes the software control of the chip clock and reset system. Clock control includes clock frequency conversion, clock shutdown and self-adaptation should be gated; reset control includes soft reset control of the system and sub-modules.

5.2 Main Features

- ÿ Supports clock shutdown of each module
- ÿ Support some modules clock adaptive shutdown
- ÿ Support software reset of each module
- ÿ Support CPU frequency setting
- ÿ Support ADC/DAC loopback test
- ÿ Support I2S clock setting

5.3 Functional Description

5.3.1 Clock Gating

By configuring the clock gating enable register CLK_GATE_EN, the clock of the specified function can be controlled to turn off, so as to turn off the function of a certain module. able purpose.

In order to provide the flexibility of firmware to control the power consumption of the system, the clock and reset module provides the clock gating function of each module in the system. when closed When the clock of the corresponding module is stopped, the digital logic and clock tree of the module will stop working, which can reduce the dynamic power consumption of the system.

The switch of each module corresponds to the detailed description of the register SW_CLKG_EN.

5.3.2 Clock Adaptive Shutdown

The chip adaptively shuts down the clocks of certain functional modules according to the transition of certain internal states.

Users, please do not change the configuration, otherwise it may cause system abnormality when configuring the PMU function.

5.3.3 Function reset

The chip provides the soft reset function of each subsystem, and the subsystem reset can be achieved by setting the corresponding BIT of SW_RST_CTRL to 0.

However, the reset state will not be cleared automatically, and the corresponding BIT bit of SW_RST_CTRL needs to be set to 1 to resume normal operation.

The soft reset function does not reset the CPU and WatchDog.

In this register, the reset operation of APB/BUS1/BUS2 (corresponding to APB bus, system bus and data bus) is not recommended, which will cause the system

The system access device is abnormal.

5.3.4 Clock division

The W800 system uses 40MHz/24MHz crystal as the system clock source, the system has built-in DPLL, and the fixed output 480MHz clock is used as the clock source.

The clock source of the whole system (as shown in the figure below).

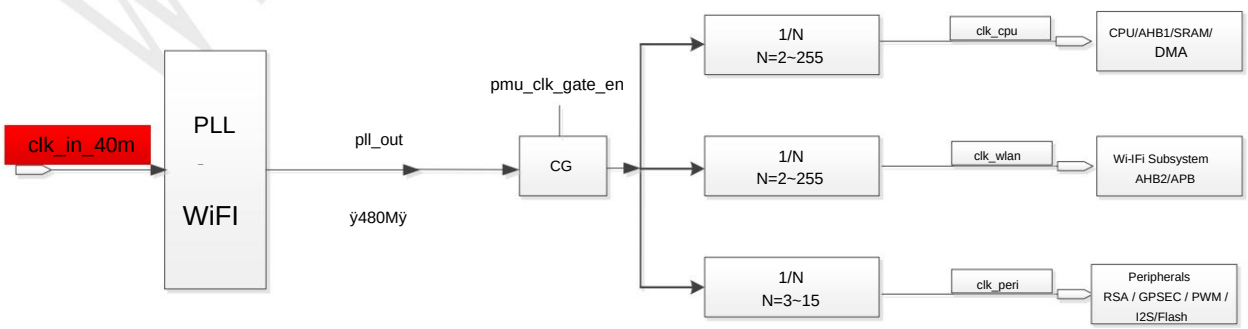


Figure 4 System clock frequency division relationship

The clock of the system bus is consistent with the CPU clock, and the clock of the data bus is fixed to 1/4 of the WLAN root clock.

The WLAN root clock is also the clock source of the entire WLAN system.

This module provides the function of setting CPU clock and WLAN root clock for firmware to adjust system performance and power consumption.

Set the BIT[7:0] of the SYS_CLK_DIV register to adjust the CPU clock frequency division factor. The source clock of the CPU clock frequency division is the DPLL

output, fixed at 480MHz. The default value of the CPU clock frequency division factor is 6, that is, the default operating frequency of the CPU is 480MHz divided by 6.

That is 80MHz. This parameter can be reconfigured when the clock required by the CPU needs to be adjusted.

The CLK_PERI clock provides the root clock of the running clock of the encryption module in the SoC system, and the root clock of the running clock of some interfaces, more than

Such as PWM interface, I2S interface, Flash interface clock. This clock is also derived by dividing the 480MHz output from the DPLL. normal working condition

The lower frequency should be fixed to divide by 3 to get the CLK_PERI root clock of 160MHz. Divide by 2 and divide by 4 by CLK_PERI root clock

80MHz and 40MHz are provided for encryption module and interface module.

Set the BIT[15:8] of the SYS_CLK_DIV register to adjust the WLAN clock frequency division factor. The default frequency division factor is 3, that is, for DPLL

The 480MHz output frequency is divided by 3, and the 160MHz clock is obtained, which is sent to the WLAN as the root node clock (the WLAN continues to divide the frequency to obtain more

For the detailed low frequency clock used by the WLAN system.

Note: If you want the WLAN system to work normally, the WLAN root clock needs to be kept at 160MHz, otherwise the WLAN system will fail.

When the WLAN system is not required to work, the WLAN root clock can be lowered to reduce the dynamic power consumption of the system.

When changing the system clock configuration, it should be noted that the ratio of the system bus to the data bus needs to be maintained at M:1, where M is an integer,

The minimum is 1. When changing the system clock configuration, it is also necessary to update the BIT [23:16] of the register SYS_CLK_DIV at the same time, set the correct ratio

example coefficient. Otherwise, accessing the data bus will result in abnormal data.

[15:8] of SYS_CLK_SEL provides the frequency division factor for setting the operating frequency of SAR_ADC, which is divided by 40M as the clock source. Frequency division

The number is the assigned frequency division value.

BIT[4] of SYS_CLK_SEL is to configure the clock frequency selection of the core operation of the RSA module, which can be 80MHz or 160MHz.

BIT[5] is to configure the clock frequency selection of the core operation of the GPSEC module, which can be 80MHz or 160MHz.

BIT[6] is to configure the clock frequency selection of the external bus of the FLASH module, which can be 40MHz or 80MHz.

When you need to reconfigure cpu_clk_divider, wlan_clk_divider, bus2_syncdn_factor, sdadc_fdiv, you need to set

Bit BIT[31] of SYS_CLK_DIV, the hardware automatically updates the above four parameters to the frequency divider, and then clears BIT[31].

I2S_CLK_CTRL provides the clock configuration function of the I2S module.

5.3.5 Debug function control

The user can enable and disable the JTAG function by setting the value of DEBUG_CTRL (SYS_CLK_SEL-BIT[16]).

5.4 Register Description

5.4.1 Register List

Table 7 Clock Reset Module Register List

offset address	name	abbreviation	access	describe	reset value
0X0000 Software Clock Gating Enable Register SW_CLKG_EN			RW Whether the software configuration module turns off the clock		0X0000_7FFF
0X0004 Software clock mask register SW_CLK_MASK RW				Whether the software configuration module is adaptively shut down Bell	0X0000_007E
0X0008 Reserved					
0X000C Software reset control register SW_RST_CTRL			RW software configuration reset module		0X01FF_FFFF
0X0010 Clock divider configuration register		SYS_CLK_DIV	RW configures the clock divider ratio		0X0000_2212
0X0014 Debug Control Register		DEBUG_CTRL	RW Configure ADC/DAC loopback test		0X0000_0000
0X0018	I2S Clock Control Register	I2S_CLK_CTRL	RW configure the I2S clock		0X0000_0000
0X001C Reset Status Register		RESET_STATUS RW		View CPU soft reset and Watchdog reset state	0x0000_0000

5.4.2 Software Clock Gating Enable Register

Table 8 Software Clock Gating Enable Register

bit access		Instructions	reset value
[31:22] RO		Reserve	
[21]	RW	soft_touch_gate_en Configure the gate control of the touch_sensor module clock. By default, the touch_sensor module gate control is enabled. 0: touch_sensor module clock is off	1'b1

		1: touch_sensor clock on	
[20]	RW	soft_bt_gate_en Configure the gate control of the BT./BLE module clock. By default, the gate control of the BT/BLE module is enabled. 0: BT/BLE module clock is off 1: BT/BLE clock on	1'b1
[19]	RW	soft_qsram_gate_en Configure the gating of the clock of the qspi_ram module. By default, the gating of the qspi_ram module is enabled. 0: qspi_ram module clock is off 1: qspi_ram clock on	1'b1
[18]	RW	soft_sdio_m_gate_en Configure the gating of the clock of the sdio_master module. By default, the gating of the sdio_master module is enabled. 0: sdio_master module clock is off 1: sdio_master clock on	1'b1
[17]	RW	soft_gpsec_gate_en Configure gpsec module clock gating, gpsec module gating is enabled by default 0: gpsec module clock is off 1: gpsec clock on	1'b1
[16]	RW	soft_rsa_gate_en Configure the gating of the RSA clock, the default RSA gating is enabled 0: RSA module clock is off 1: RSA clock on	1'b1
[15]	RW	soft_i2s_gate_en	1'b1

		Configure i2s clock gating, i2s gating is enabled by default 0: i2s clock is off 1: i2s clock on	
[14]	RW	soft_lcd_gate_en Configure the gating of the lcd clock, the default lcd gating is enabled 0: LCD clock off 1: LCD clock on	1'b1
[13]	RW	Soft_pwm_gate_en Configure the gating of the pwm clock, the default pwm gating is enabled 0: pwm clock off 1: pwm clock on	1'b1
[12]	RW	soft_sd_adc_gate_en Configure the gating of sd_adc_clock, the default sd_adc_gating is enabled 0: sd_adc_clock off 1: sd_adc_clock on	1'b1
[11]	RW	soft_gpio_gate_en Configure the gating of the GPIO clock, the default GPIO gating is enabled 0: GPIO clock off 1: GPIO clock on	1'b1
[10]	RW	soft_timer_gate_en Configure the gate control of the timer clock, the default timer gate control is enabled 0: timer clock off	1'b1

		1: timer clock on	
[9]	RW	soft_rf_cfg_gate_en: used internally, do not modify Configure the gating of the rf_cfg clock, the default rf_cfg gating is enabled 1'b0: rf_cfg clock off 1'b1: rf_cfg clock on	1'b1
[8]	RW	soft_dma_gate_en Indicates whether the clock supplied to the dma clock domain is turned off 1'b0: dma clock off 1'b1: dma clock on	1'b1
[7]	RW	soft_ls_spi_gate_en Configure the gating of the low-speed spi clock, the default low-speed spi gating is enabled 1'b0: low speed spi clock off 1'b1: low speed spi clock on	1'b1
[6]	RW	soft_uart5_gate_en Configure the gate control of uart5, the default uart5 is enabled 0: uart5 is off 1: uart5 is on	1'b1
[5]	RW	soft_uart4_gate_en Configure the gate control of uart4, the default uart4 is open 0: uart4 is off 1: uart4 is on	1'b1
[4]	RW	soft_uart3_gate_en	1'b1

		Configure the gate of uart3, the default uart3 is enabled 0: uart3 is off 1: uart3 is enabled	
[3]	RW	soft_uart2_gate_en Configure the gate control of uart2, the default uart2 is enabled 0: uart2 is off 1: uart2 is enabled	1'b1
[2]	RW	soft_uart1_gate_en Configure the gating of the uart1 clock, the default uart1 gating is enabled 1'b0: uart1 clock off 1'b1: uart1 clock on	1'b1
[1]	RW	soft_uart0_gate_en Configure the gating of the uart0 clock, the default uart0 gating is enabled 1'b0: uart0 clock off 1'b1: uart0 clock on	1'b1
[0]	RW	soft_i2c_gate_en Configure i2c clock gating, i2c gating is enabled by default 1'b0: i2c clock off 1'b1: i2c clock on	1'b1

5.4.3 Software Clock Mask Register

Table 9 Software Clock Mask Register

bit access		Instructions	reset value
[6]	RW	soft_cpu_clk_gt_mask Indicates whether the clock supplied to the CPU clock domain (including CPU, bus1, ROM, SRAM) can be adaptive Shutdown (when the **CPU needs to enter the WFI state, do not set the adaptive shutdown)** 1'b0: Allows adaptive turn-off and turn-on 1'b1: Adaptive turn-off and turn-on are not allowed	1'b1
[5 : 2] RW		**Reserved for internal use, do not modify**	
[1]	RW	soft_sdioahb_clk_gt_mask Indicates whether the clock supplied to the sdio ahb clock domain can be turned off adaptively 1'b0: Allows adaptive turn-off and turn-on 1'b1: Adaptive turn-off and turn-on are not allowed	1'b1
[0]	RW	soft_pmu_clk_gt_mask There is a gate control unit after the clock output by pll, which is configured by this register to indicate whether it is allowed to be turned off by the PMU. 1'b0: Allows the PMU to shut down the gate unit, thereby shutting down the clock 1'b1: PMU is not allowed to shut down the gate unit	1'b0

5.4.4 Software Reset Control Register

Table 10 Software Reset Control Register

bit access		Instructions	reset value
[31]	RW	soft_touch_rst_n Software reset touch_sensor module 0: reset	1'b1

		1: Reset release	
[30]	RW	soft_rst_flash_n Software reset the flash controller module 0: reset 1: Reset release	1'b1
[29]	RW	soft_rst_bt_n Software reset BT module 0: reset 1: Reset release	1'b1
[28]	RW	soft_rst_qspi_ram_n Software reset qspi_ram module 0: reset 1: Reset release	1'b1
[27]	RW	soft_rst_sdio_m_n Software reset sdio_master module 0: reset 1: Reset release	1'b1
[26]	RW	soft_rst_gpsec_n software reset gpsec module 1'b0: reset 1'b1: reset release	1'b1
[25]	RW	soft_rst_rsa_n	1'b1

		Software reset RSA module 1'b0: reset 1'b1: reset release	
[24]	RW	soft_rst_i2s_n Software reset i2s module 1'b0: reset 1'b1: reset release	1'b1
[23]	RW	soft_rst_lcd_n software reset lcd module 1'b0: reset 1'b1: reset release	1'b1
[22]	RW	soft_rst_pwm_n software reset pwm module 1'b0: reset 1'b1: reset release	1'b1
[21]	RW	soft_rst_sar_adc_n Software reset sar_adc module 1'b0: reset 1'b1: reset release	1'b1
[20]	RW	soft_rst_timer_n Software reset timer module 1'b0: reset	1'b1

		1'b1: reset release	
[19]	RW	soft_rst_gpio_n software reset gpio module 1'b0: reset 1'b1: reset release	1'b1
[18]	RW	soft_rst_rf_cfg_n Software reset to configure **the RF register module (for internal use, do not modify)** 1'b0: reset 1'b1: reset release	1'b1
[17]	RW	soft_rst_spis_n Software reset high-speed spi module 1'b0: reset 1'b1: reset release	1'b1
[16]	RW	soft_rst_spim_n Software reset low speed spi module 1'b0: reset 1'b1: reset release	1'b1
[15]	RW	soft_rst_uart5_n Software reset on-chip uart5 module 1'b0: reset 1'b1: reset release	1'b1
[14]	RW	soft_rst_uart4_n	1'b1

		Software reset on-chip uart4 module 1'b0: reset 1'b1: reset release	
[13]	RW	soft_rst_uart3_n Software reset on-chip uart3 module 1'b0: reset 1'b1: reset release	1'b1
[12]	RW	soft_rst_uart2_n Software reset on-chip uart2 module 1'b0: reset 1'b1: reset release	1'b1
[11]	RW	soft_rst_uart1_n Software reset on-chip uart1 module 1'b0: reset 1'b1: reset release	1'b1
[10]	RW	soft_rst_uart0_n Software reset on-chip uart0 module 1'b0: reset 1'b1: reset release	1'b1
[9]	RW	soft_rst_i2c_n Software reset on-chip i2c module 1'b0: reset	1'b1

		1'b1: reset release	
[8]	RW	soft_rst_bus2_n Software reset on-chip bus2 module 1'b0: reset 1'b1: reset release	1'b1
[7]	RW	soft_rst_bus1_n Software reset on-chip bus1 module 1'b0: reset 1'b1: reset release	1'b1
[6]	RW	soft_rst_apb_n software reset abp bridge module 1'b0: reset 1'b1: reset release	1'b1
[5]	RW	soft_rst_mem_mng_n Software reset mem_mng **module (internal use, do not modify)** 1'b0: reset 1'b1: reset release	1'b1
[4]	RW	soft_rst_dma_n software reset dma module 1'b0: reset 1'b1: reset release	1'b1
[3]	RW	soft_rst_sdio_ahb_n	1'b1

		software reset sdio ahb clock domain module 1'b0: reset 1'b1: reset release	
[2]	RW	soft_rst_sec_n **Software reset security module (internal use, do not modify)** 1'b0: reset 1'b1: reset release	1'b1
[1]	RW	soft_rst_mac_n Software reset mac **module (internal use, do not modify)** 1'b0: reset 1'b1: reset release	1'b1
[0]	RW	soft_rst_bbp_n Software reset bbp **module (internal use, do not modify)** 1'b0: reset 1'b1: reset release	1'b1

5.4.5 Clock Divider Configuration Register

Table 11 Clock Divider Configuration Register

bit access		Instructions	reset value
[31]	RW	divide_freq_en When it is necessary to reconfigure cpu_clk_divider, wlan_clk_divider, bus2_syncdn_factor, When sdadc_fdiv, set this register, the hardware will automatically update the above four parameters to the frequency divider, and then clear this register	1'b0

		register. 1'b0: The frequency division factor has taken effect 1'b1: Require hardware to update frequency division parameters Note: When the dividing factor is configured here, when Divide_freq_en is valid, all factors must be valid	
[30:28]		Reserve	
[27:24] RW		Peripheral_divider 160M clock division factor: Divided by the DPLL as the clock source. The frequency division factor is the assigned frequency division value. The divided output should be 160MHz. The DPLL output is 480MHz and should be configured to 3.	4'h3
[23: 16] RW		bus2_syncdn_factor The clock ratio between bus1 and bus2 should be N:1 Among them, N is an integer. In the actual adjustment, it mainly depends on the ratio of the operating frequency of the CPU and the clock frequency of bus2. Since the default cpu uses 80MHz clock and bus2 uses 40MHz clock, then N=2	8'h2
[15 : 8] RW		wlan_clk_divider The clock from the PLL is divided and sent to the wlan system. This register is the frequency division factor, which is >=2. The default frequency division factor is 3, that is, the 480MHz output of pll is divided by 3 to obtain a 160MHz clock, which is used as the root The node clock is sent to wlan (wlan continues to divide the frequency to obtain a more detailed low-frequency clock); Note 1: If the WLAN system needs to work normally, this clock needs to be fixed at 160MHz; if the WLAN system is turned off, Then this clock can be downclocked to save power. This clock must not be configured higher than 160MHz. Note 2: The secondary bus clock and APB clock are divided by 4 for this clock;	8'h3

[7 : 0] RW		cpu_clk_divider The clock from the PLL is divided and sent to the CPU. This register is the frequency division factor, which is >=2. The default frequency division factor is 6, that is, after the reset is released, the 480MHz clock output by the PLL is divided by 6 and sent to the cpu is the 80MHz clock. When you need to adjust the clock required by the cpu, you can reconfigure this parameter	8'h6

5.4.6 Debug Control Register

Table 12 Clock Select Register

bit access		Instructions	reset value
[16]	RW	JTAG enable 1'b0: Disable JTAG debugging 1'b1: Enable JTAG debug function	1'b0
[15ÿ8] RW		sd_adc_div sigma-delta ADC clock division factor: Divide by 40MHz as the clock source. The frequency division factor is the assigned frequency division value. After configuring this register, Divide_freq_en in the register clk_divider must be configured to take effect;	8'd10
[7]	RW	RSV	1'b0
[6]	RW	qflash_clk_sel QSPI_FLASH clock selection 1: Use 80MHz;	1'b0

		0: use 40MHz;	
[5]	RW	gpsec_sel GPSEC clock selection 1: Use 160MHz; 0: use 80MHz;	1'b0
[4]	RW	rsa_sel RSA clock selection 1: Use 160MHz; 0: use 80MHz;	1'b0
[3:0] RW		**reserved, do not modify**	4'd0

5.4.7 I2S Clock Control Register

Table 13 I2S Clock Control Register

bit access		Instructions	reset value
[31:18]		Reserve	
[17: 8] RW		BCLKDIV BCLK splitter: F_BCLK = F_I2SCLK / BCLKDIV Note: If EXTAL_EN is not selected and internal PLL is used then F_I2SCLK = F_CPU (same as CPU frequency) same). Assuming F_CPU = 160MHz, F_I2SCLK = external crystal frequency when WXTAL_EN is enabled, **BCLKDIV = round (F_I2SCLK/(Fs*W*F))** Where Fs is the sampling frequency of the audio data, W is the word width;	10'b0

		F = 1 when the data is mono; F = 2 when the data is stereo. For example, if the internal PLL is used and the data width is 24 bits, the format is stereo and the sampling frequency is 128KHz, BCLKDIV should be configured as (160 * 10e6 / 128 * 10e3 * 24 * 2ÿ= 10'h1aÿ	
[7 : 2] RW		MCLKDIV If an external clock is selected, this MCLK divider is used to generate the appropriate MCLK frequency. F_mclk = F_I2SCLK / (2 * MCLKDIV)ÿ F_I2SCLK is the external clock when MCLKDIV = 0; F_mclk = F_I2SCLK when MCLKDIV >= 1; Note: F_mclk should be configured as 256 * fs, where fs is the sampling frequency.	6'b0
[1]	RW	MCLKEN MCLK enable switch 1'b0: MCLK disabled 1'b1: enable MCLK	1'b0
[0]	RW	EXTAL_EN External clock selection, choose whether to use the internal I2S block clock or external clock 1'b0: Internal clock 1'b1: External clock Note: When using an external clock, the external clock must be 2 * N * 256 fs, where fs is the sampling frequency, N must be Must be an integer.	1'b0

5.4.8 Reset Status Register

Table 14 Reset Status Register

bit access		Instructions	reset value
[31:18]		Reserve	
[17: 8]	WHERE	CPU soft reset state clear Write 1 to clear CPU soft Reset Status.	1'b0
[7 : 2] WO		Wdog soft reset state clear Writing a 1 clears the Wdog Reset Status.	1'b0
[1]	RO	CPU soft reset state 1: The CPU has generated a soft reset; 0: The CPU does not generate a soft reset;	1'b0
[0]	RO	Wdog reset state 1: Wdog generated Reset; 0: Wdpg does not generate Reset	1'b0

6 DMA module

6.1 Function overview

DMA is used to provide high-speed data transfer between peripherals and memory and between memory and memory. Can operate without any CPU

Fast data movement via DMA without The CPU resources saved in this way do not affect the operations of other instructions performed by the CPU.

DMA is mounted on the AHB bus, supports up to 8 channels, 16 hardware peripheral request sources, and supports linked list structure and register control.

6.2 Main Features

- ÿ Amba2.0 standard bus interface, 8 DMA channels
- ÿ Support DMA operation based on memory linked list structure
- ÿ Support 16 hardware peripheral request sources
- ÿ Support 1, 4-burst operation mode
- ÿ Support byte, half-word, word as unit transfer operation
- ÿ Support source and destination address unchanged or sequentially incremented or configurable to cycle operations within a predefined address range
- ÿ Supports data transfer methods from memory to memory, memory to peripherals, and peripherals to memory

6.3 Functional Description

6.3.1 DMA channel

W800 supports a total of 8 DMA channels, DMA channels do not interfere with each other and can run at the same time. Request different data streams can choose different

DMA channel.

Each DMA channel is allocated in a different register address offset segment, and you can directly select the address segment of the corresponding channel for configuration and use. No

The register configuration method of the same channel is exactly the same.

Table 15 DMA address assignment

DMA base address	0x4000 0800
DMA_CH0	Offset (0x10~0x38)
DMA_CH1	Offset (0x40~0x68)
DMA_CH2	Offset (0x70~0x98)
…	…
DMA_CH7	Offset (0x160~0x188)

6.3.2 DMA data flow

Eight DMA channels enable unidirectional data transfer link between source and destination.

The source and destination addresses of the DMA can be set to remain unchanged, incremented or cyclic after each DMA operation is completed:

- ÿ DMA_CTRL[2:1] controls how the source address changes after each DMA operation;
- ÿ DMA_CTRL[4:3] controls how the destination address changes after each DMA operation.

DMA can set the handling unit of byte, half-word and word, and the final quantity of data to be handled is an integer multiple of the handling unit.

DMA_CTRL[6:5] to set.

DMA can set how many units of data to transfer each time through burst, and choose to transfer 1 or 4 units at a time through DMA_CTRL[7].

Bit data, if DMA_CTRL[6:5] is set to word and burst is set to 4, then 4 words of data are transferred each time.

DMA can set the number of Bytes to start DMA transfer each time, the maximum is 65535 Bytes, which can be set by DMA_CTRL[23:8].

6.3.3 DMA Cycling Mode

The DMA loop address mode means that after the source and destination addresses of the DMA are set, after the data transfer reaches the set loop boundary, it will jump to

The loop start address, and this loop executes until the set transfer byte is reached.

The source and destination addresses of the circular address mode need to be set with the SRC_WRAP_ADDR and DEST_WRAP_ADDR registers, and are set by the SRC_WRAP_ADDR and DEST_WRAP_ADDR registers.

WRAP_SIZE to set the loop length value.

6.3.4 DMA transfer mode

DMA supports 3 transfer modes:

ÿ RAM to RAM

Both the source address and the destination address are configured as memory addresses that need to be transferred, and DMA_MODE[0] is set to 0, in software mode.

ÿ Memory to Peripherals

The source address is set to the memory address, the destination address is set to the peripheral address, DMA_MODE[0] is set to 1, the hardware mode,

DMA_MODE[5:2] selects the peripheral used.

ÿ Peripherals to memory

The source address is set to the peripheral address, the destination address is set to the memory address, DMA_MODE[0] is set to 1, the hardware mode,

DMA_MODE[5:2] selects the peripheral used.

6.3.5 DMA peripheral selection

When using the transfer method of peripheral to memory or memory to peripheral, in addition to the corresponding peripheral needs to be set to DMA TX or RX,

DMA_MODE[5:2] also needs to select the corresponding peripheral.

Note: Because there are 3 UART ports in total, when UART uses DMA, it is necessary to select the corresponding port through UART_CH[1:0].

UARTÿ

6.3.6 DMA linked list mode

DMA supports linked list working mode. Through the linked list mode, when DMA transfers the current linked list memory data, we can advance to the next

The data is filled in each linked list. After the DMA finishes moving the current linked list, it judges that the next linked list is valid, and can directly move the data of the next linked list.

The linked list method can effectively improve the efficiency of DMA and CPU cooperation.

Linked list operation mode: Set the DMA to linked list working mode through the DMA_MODE[1] register, and then set the DESC_ADDR register

is the starting address of the linked list structure, and then enables DMA through the CHNL_CTRL register. When the DMA process finishes moving the current memory

After that, the software informs the DMA that there is still valid data in the linked list by setting the valid flag, and the DMA processes the data according to the valid flag of the linked list.

A data to be moved.

6.3.7 DMA Interrupts

An interrupt can be generated when the DMA transfer is completed or burst, and the INT_MASK register can mask the interrupt corresponding to the DMA channel.

When the corresponding DMA interrupt is generated, the current interrupt status can be queried through the INT_SRC register to indicate what is currently generating the interrupt.

The corresponding status bits need to be cleared by software by writing 1 to 1.

6.4 Register Description

6.4.1 Register List

Table 16 DMA register list

offset address	name	abbreviation	access	describe	reset value
0X0000	Interrupt Mask Register	INT_MASK	RW	sets the DMA interrupt that needs to be masked	0X0000_FFFF
0X0004	Interrupt Status Register	INT_SRC	RW	indicates the current DMA interrupt status	0X0000_0000
0X0008	DMA channel selection register	DMA_CH	RW	UART peripheral to select which UART	0X0000_0000

0X000C Reserved					
DMA CHNL0 registers					
0X0010	DMA Source Address Register	SRC_ADDR	RW	DMA transfer source address	0X0000_0000
0X0014	DMA Destination Address Register	DEST_ADDR	RW	DMA transfer destination address	0X0000_0000
0X0018	DMA loop source start address register	SRC_WRAP_ADDR	RW	DMA transfer source address in loop mode	0X0000_0000
0X001C	DMA loop destination start address register device	DEST_WRAP_ADD R	RW	DMA transfer destination in circular mode site	0X0000_0000
0X0020	DMA Cycle Length Register	WRAP_SIZE	RW	DMA cycle boundary in cycle mode	0X0000_0000
0X0024	DMA Channel Control Register	CHNL_CTRL	RW	Current channel DMA start and stop	0X0000_0000
0X0028	DMA Mode Select Register	DMA_MODE	RW	sets how DMA works	0X0000_0000
0X002C	DMA Data Flow Control Register	DMA_CTRL	RW	set DMA transfer data stream	0X0000_0000
0X0030	DMA transfer bytes register	DMA_STATUS	RO	Get the current number of bytes transferred	0X0000_0000
0X0034	DMA linked list entry address register	DESC_ADDR	RW	DMA linked list address entry address setting	0X0000_0000
0X0038	DMA current destination address register	CUR_DEST_ADDR	RO	The address of the current DMA operation	0X0000_0000
DMA CHNL1 registers					
0X0040 - 0X0068	ÿ DMA CHNL0 registers				
DMA CHNL2 registers					
0X0070 - 0X0098	ÿ DMA CHNL0 registers				
DMA CHNL3 registers					
0X00A0 - ÿ DMA CHNL0 registers					

0X00C8	
DMA CHNL4 registers	
0X00D0 - 0X00F8	ÿ DMA CHNL0 registers
DMA CHNL5 registers	
0X0100 - 0X0128	ÿ DMA CHNL0 registers
DMA CHNL6 registers	
0X0130 - 0X0158	ÿ DMA CHNL0 registers
DMA CHNL7 registers	
0X0160 - 0X0188	ÿ DMA CHNL0 registers

6.4.2 Interrupt Mask Register

Table 17 DMA Interrupt Mask Register

bit access		Instructions	reset value
[31:16]		Reserve	
[15]	RW	channel7 transfer_done interrupt mask, active high.	1'b1
[14]	RW	channel7 burst_done interrupt mask, active high.	1'b1
[13]	RW	channel6 transfer_done interrupt mask, active high.	1'b1
[12]	RW	channel6 burst_done interrupt mask, active high.	1'b1

[11]	RW	channel5 transfer_done interrupt mask, active high.	1'b1
[10]	RW	channel5 burst_done interrupt mask, active high.	1'b1
[9]	RW	channel4 transfer_done interrupt mask, active high.	1'b1
[8]	RW	channel4 burst_done interrupt mask, active high.	1'b1
[7]	RW	channel3 transfer_done interrupt mask, active high.	1'b1
[6]	RW	channel3 burst_done interrupt mask, active high.	1'b1
[5]	RW	channel2 transfer_done interrupt mask, active high.	1'b1
[4]	RW	channel2 burst_done interrupt mask, active high.	1'b1
[3]	RW	channel1 transfer_done interrupt mask, active high.	1'b1
[2]	RW	channel1 burst_done interrupt mask, active high.	1'b1
[1]	RW	channel0 transfer_done interrupt mask, active high.	1'b1
[0]	RW	channel0 burst_done interrupt mask, active high.	1'b1

6.4.3 Interrupt Status Register

Table 18 DMA Interrupt Status Register

bit access		Instructions	reset value
[31:16]		Reserve	
[15]	RW	channel7 transfer_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA transfer is complete.	1'b0
[14]	RW	channel7 burst_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA burst is complete.	1'b0
[13]	RW	channel6 transfer_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA transfer is complete.	1'b0
[12]	RW	channel6 burst_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA burst is complete.	1'b0
[11]	RW	channel5 transfer_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA transfer is complete.	1'b0

[10]	RW	channel5 burst_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA burst is complete.	1'b0
[9]	RW	channel4 transfer_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA transfer is complete.	1'b0
[8]	RW	channel4 burst_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA burst is complete.	1'b0
[7]	RW	channel3 transfer_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA transfer is complete.	1'b0
[6]	RW	channel3 burst_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA burst is complete.	1'b0
[5]	RW	channel2 transfer_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA transfer is complete.	1'b0
[4]	RW	channel2 burst_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA burst is complete.	1'b0
[3]	RW	channel1 transfer_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA transfer is complete.	1'b0
[2]	RW	channel1 burst_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA burst is complete.	1'b0
[1]	RW	channel0 transfer_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA transfer is complete.	1'b0
[0]	RW	channel0 burst_done interrupt status, write 1 to clear 0. An interrupt is generated when the DMA burst is complete.	1'b0

6.4.4 UART selection register

Table 19 UART select register

bit access		Instructions	reset value
[31: 24]		Reserve	
[23:8] RW		dma req clear: Write 1 to each bit to clear the corresponding dma req request. Self-cleaning. For example, writing 1 to bit 23 will clear the 15th corresponding dma request in dma_sel; Writing 1 to bit 8 will clear the 0th dma request in dma_sel - uart_rx_req;	16'd0
[2 : 0] RW		Uart dma channel selection: 3'd0: uart0 module dma channel access dma	3'h0

		3'd1: uart1 module dma channel access dma 3'd2: uart2/7816 module dma channel access dma 3'd3: uart3 module dma channel access dma 3'd4: uart4 module dma channel access dma 3'd5: uart5 module dma channel access dma	

6.4.5 DMA Source Address Register

Table 20 DMA Source Address Register

bit access		Instructions	reset value
[31: 0] RW		In acyclic mode, the source address, peripheral address or memory address of the DMA transfer	32'h0

6.4.6 DMA Destination Address Register

Table 21 DMA Destination Address Register

bit access		Instructions	reset value
[31: 0] RW		In acyclic mode, the destination address, peripheral address or memory address of DMA transfer	32'h0

6.4.7 DMA loop source start address register

Table 22 DMA loop source start address register

bit access		Instructions	reset value
[31: 0] RW		In circular mode, the starting address of the source address, peripheral address or memory address of the DMA transfer	32'h0

6.4.8 DMA loop destination start address register

Table 23 DMA loop destination start address register

bit access		Instructions	reset value
[31: 0] RW		In circular mode, the starting address, peripheral address or memory address of the destination address of DMA transfer	32'h0

6.4.9 DMA Cycle Length Register

Table 24 DMA Cycle Length Register

bit access		Instructions	reset value
[31:16] RW		In loop mode, the loop length of the DMA destination address. DMA moves data sequentially from the start address. When the number of bytes of data to be moved reaches this set value, it will jump Go to the cycle start address and continue to move data from the start address	16'h0
[15: 0] RW		In loop mode, the DMA source address loop length.	16'h0

6.4.10 DMA Channel Control Register

Table 25 DMA Channel Control Register

bit access		Instructions	reset value
[31: 2]		Reserve	
[1]	RW	dma_stop Stop dma operation, active high. The DMA stops after completing the current burst operation and clears chnl_on at the same time. software should be based on chnl_on is 0 to determine that the dma has completely stopped.	1'b0
[0]	RW	chnl_on	1'b0

		Start the current channel DMA conversion, active high. It is automatically cleared to 0 after Dma is completed and the conversion or setup stops.	

6.4.11 DMA Mode Select Register

Table 26 DMA Mode Select Register

bit access		Instructions	reset value
[31: 7]		Reserve	
[6]	RW	chain_link_en Valid in linked list mode, indicating whether dma continues to read and process subsequent chains after processing the first linked list surface. If it is 1, update the next_desc_addr in the linked list and continue reading the next linked list until the linked list where vld is 0; if it is 0, the processing will stop after completing the current linked list.	1'b0
[5 : 2]	RW	dma_sel Choice of 16 dma_reqs. 4'd0ÿuart rx dma req 4'd1ÿuart tx dma req 4'd2ÿpwm_cap0_req 4'd3ÿpwm_cap1_req 4'd4ÿLS_SPI rx dma req 4'd5ÿLS_SPI tx dma req 4'd6ÿSD_ADC chnl0 req 4'd7ÿSD_ADC chnl1 req	4'h0

		4'd8ÿSD_ADC chnl2 req 4'd9ÿSD_ADC chnl3 req 4'd10: I2S RX req 4'd11: I2S TX req 4'd12: SDIO_HOST req	
[1]	RW	chain_mode 1'b0: use normal mode 1'b1: use linked list mode	1'b0
[0]	RW	dma_mode 1'b0: Software mode. 1'b1: Hardware mode.	1'b0

6.4.12 DMA Data Flow Control Register

Table 27 DMA Data Flow Control Register

bit access		Instructions	reset value
[31:24]		Reserve	
[23: 8] RW		total_byte The total number of bytes to be operated on. It needs to be consistent with the data_size configuration, that is, if it is a word operation, then Should be configured as an integer multiple of 4; if it is a halfword operation, it should be configured as an integer multiple of 2.	16'h0
[7]	RW	burst_size Set how many units of data the DMA transfers at a time 1'b0: burst is 1	1'b0

		1'b1: burst is 4 When the last burst size exceeds the number of remaining transfers, use the burst size as the size of the remaining data small.	
[6 : 5] RW		data_size Set the handling unit for DMA 2'h0 ÿ byte 2'h1ÿhalf_word 2'h2ÿword 2'h3: reserved	2'h0
[4 : 3] RW		dest_addr_inc 2'h0: The destination address remains unchanged after each operation; 2'h1: The destination address is automatically accumulated after each operation. 2'h2: reserved 2'h3: Loop operation, the destination address is automatically accumulated after each operation, and it jumps to the start of the loop when it reaches the defined loop boundary. starting address.	2'h0
[2 : 1] RW		src_addr_inc 2'h0: The source address remains unchanged after each operation; 2'h1: The source address is automatically accumulated after each operation. 2'h2: reserved 2'h3: Loop operation, the source address is automatically accumulated after each operation, and jumps to the start of the loop when it reaches the defined loop boundary address.	2'h0
[0]	RW	auto_reload	1'b0

		When the current DMA transfer is completed, the next DMA transfer will be performed automatically according to the current DMA configuration.	

6.4.13 DMA transfer bytes register

Table 28 DMA Transfer Bytes Register

bit access		Instructions	reset value
[31:16]		Reserve	
[15: 0] RW		transfer_cnt The number of bytes currently transferred. Each time the dma is restarted (chnl_on is set to 1), it is cleared to 0, and the counting is restarted.	16'h0

6.4.14 DMA linked list entry address register

Table 29 DMA linked list entry address register

bit access		Instructions	reset value
[31: 0] RW		desc_addr When the linked list is enabled, it is used as the entry address of the linked list. After each transfer of the linked list is completed, the base of the next linked list is address is updated to this register.	32'h0

6.4.15 DMA current destination address register

Table 30 DMA current destination address register

bit access		Instructions	reset value
[31: 0] RO		current_dest_addr The destination address of the current DMA operation.	32'h0

		当软件停止 dma 时，可以通过查看此寄存器获悉 dma 将要操作的目的地址。	

7 Universal hardware encryption module

7.1 Function overview

The encryption module automatically completes the encryption of the source address space data of the specified length, and automatically writes the encrypted data back to the specified destination address after completion.

time; supports SHA1/MD5/RC4/DES/3DES/AES/CRC/TRNG.

7.2 Main Features

- ÿ Support SHA1/MD5/RC4/DES/3DES/AES/CRC/TRNG encryption algorithm
- ÿ DES/3DES supports ECB and CBC modes
- ÿ AES supports ECB, CBC and CTR modes
- ÿ CRC supports four modes: CRC8, CRC16_MODBUS, CRC16_CCITT and CRC32
- ÿ CRC supports input/output reverse
- ÿ SHA1/MD5/CRC supports continuous multi-packet encryption
- ÿ Built-in true random number generator, also supports seed to generate pseudo-random numbers

7.3 Functional Description

7.3.1 SHA1 encryption

Hardware SHA1 calculation can be performed on consecutive multiple packets of data, the calculation result is stored in the register, and the encryption result of the previous packet can be used as the encryption result of the next packet.

initial value.

7.3.2 MD5 encryption

Hardware MD5 calculation can be performed on consecutive multi-packet data, the calculation result is stored in the register, and the encryption result of the previous packet can be used as the initial value of the next packet.

initial value.

7.3.3 RC4 encryption

Supports RC4 encryption and decryption.

7.3.4 DES encryption

Support DES encryption and decryption, support ECB and CBC two modes.

7.3.5 3DES encryption

Support 3DES encryption and decryption, support ECB and CBC two modes.

7.3.6 AES encryption

Support AES encryption and decryption, support ECB, CBC and CTR three modes.

7.3.7 CRC encryption

The hardware CRC calculation can be performed on consecutive multi-packet data, the calculation result is stored in the register, and the encryption result of the previous packet can be used as the initial value of the next packet.

It supports four modes: CRC8, CRC16_MODBUS, CRC16_CCITT and CRC32, and supports input/output inversion.

The calculation formula of CRC32 is as follows:

1ÿCRC-32: 0x04C11DB7

X32+X26+X23+X22+X16+X12+X11+X10+X8+X7+X5+X4+X2+X+1

Commonly used in ZIP, RAR, IEEE 802 LAN/FDDI, IEEE 1394, PPP-FCS and other protocols.

2. CRC-16: supports two polynomials

2.1: 0X1021

X16 + X12 + X5 + 1

Commonly used in ISO HDLC, ITU X.25, V.34/V.41/V.42, PPP-FCS CCITT and other protocols.

2.2: 0X8005

X16 + X15 + X2 + 1

Commonly used in USB, ANSI X3.28, SIA DC-07 and other protocols.

3ÿCRC-8ÿ0X207

x8+x2+x1+1

7.3.8 TRNG Random Number Generator

A true random number generator module is integrated into the W800 system. Divided into analog module and digital post-processing module. The analog module outputs a random clock ad_trng_clks and random number ad_trng_dout, the digital post-processing module is used to eliminate the bias and autocorrelation of random numbers. Related control The control register is in the GPSEC register list.

The basic operation process is as follows:

1. Enable TRNG_EN and set TRNG_SEL to 1, so that GPSEC register 0x48 displays the output value of TRNG. mode at this time

   The pseudo module starts to output random clocks and random signals. The signal sampled by the first 8 clocks is used as the initial state of the LFSR to initialize the LFSR chain, the data sampled by each random clock is post-processed by the XOR CHAIN and LFSR registers, and then shifted and stored in the register. in the server TRNG_RANDOM.

2. Software can read random value through GPSEC register 0x48. Digital postprocessing when register TRNG_DIG_BYPASS is set to 1

   The module stops working and directly stores the output value of the analog module into the result register TRNG_RANDOM.

7.4 Register Description

7.4.1 Register List

Table 31 Encryption module register list

offset address	name	abbreviation	access	describe	reset value
0X0000 Source address register		SRC_ADDR	RW RC4/SHA1/AES/DES/3DES/CRC/MD 5 Multiplexing source address		0X0000_0000
0X0004 Destination address register		DEST_ADDR	RW RC4/AES/DES/3DES multiplexing destination address 0X0000_0000		
0X0008 Configuration register		GPSEC_CFG	RW Generic Hardware Cryptographic Module Configuration Register		0X0000_0000
0X000C Control Register		GPSEC_CTRL	RW Generic Hardware Cryptographic Module Control Register		0X0000_0000
0X0010 Key 0 low register KEY00			RW Key0	Low 32-bit first input key (RC4/AES/DES/3DES), multiplexed CRC There	0X0000_0000
0X0014 Key 0 high register KEY01			RW Key0	High 32-bit first input key ÿRC4/AES/DES/3DESÿ	0X0000_0000
0X0018 Key 1 low register KEY10			RW Key1	lower 32-bit second input key ÿRC4/AES//3DESÿ	0X0000_0000
0X001C Key 1 high register KEY11			RW Key1	High 32-bit second input key ÿRC4/AES//3DESÿ	0X0000_0000
0X0020 Key 2 low register		KEY20	RW Key2	The third input key of the lower 32 bits (3DES), multiplex iv1 low 32-bit input initial Vector (AES)	0X0000_0000

0X0024 Key 2 high register		KEY21	RW	Key2 High 32-bit third input key (3DES), multiplex iv1 high 32-bit input initial Vector (AES)	0X0000_0000
0X0028	Initial vector 0 low register device	IV00	RW	IV0 low 32-bit input initial vector ÿAES/DES/3DESÿ	0X0000_0000
0X002C	Initial vector 0 high order register device	IV01	RW	IV0 upper 32-bit input initial vector ÿAES/DES/3DESÿ	0X0000_0000
0X0030 Status Register		GPSEC_STS	RW	Generic Hardware Cryptographic Module Status Register	0X0000_0000
0X0034 Summary 0 register		SHA1-DIGEST0	RW	sha1-digest0/MD5-digest0	0X6745_2301
0X0038 Summary 1 Register		SHA1-DIGEST1	RW	sha1-digest1/MD5-digest1	0XEFCD_AB89
0X003C Summary 2 Register		SHA1-DIGEST2	RW	sha1-digest2/MD5-digest2	0X98BA_DCFE
0X0040 Summary 3 register		SHA1-DIGEST3	RW	sha1-digest3/MD5-digest3	0X1032_5476
0X0044 Summary 4 Register		SHA1-DIGEST4	RW	sha1-digest4 / CRC	0XC3D2_E1F0
0X0048 RNG_result		RNG_RESULT	RW	RNG output	0X0000_0000
0X004C	Key 3 low register Key30		RW	Key3 lower 32-bit third input key (RC4's 256bit mode)	0X0000_0000
0X0050	Key 3 low register Key31		RW	Key3 high 32-bit third input key (RC4's 256bit mode)	0X0000_0000
0X0054 TRNG configuration		PRINCIPAL_CLEAR	RW	True Random Number Generator Configuration Selection	0X40

7.4.2 Configuration Registers

Table 32 Cryptographic Module Configuration Registers

bit access		Instructions	reset value
[31]	RW	RC4_128_256 0: indicates that RC4 encryption/decryption is performed according to the 128bit block length; 1: Flag RC4 encryption/decryption is performed according to 256bit block length.	1'b0
[30]	RW	RNG start 1'b0: do not start RNG 1'b1: start RNG	1'b0
[29]	RW	RNG Load_seed Hardware automatically reset to 0 1'b0: The random number generator will be seeded with zero by default to generate a random number of the corresponding number of digits 1'b1: Start generating random numbers after the seed is loaded	1'b0
[28]	RW	RNG switch controls the number of bits to generate random numbers, 1'b0: 16 bits 1'b1: 32 bits	1'b0
[27]	RW	des_soft_reset des is automatically cleared to 0 by hardware after the soft reset is completed 1'b0: No soft reset is generated and the current state is not changed 1'b1: The encryption algorithm is reset to the initial state by software	1'b0
[26]	RW	aes_soft_reset aes is automatically cleared to 0 by hardware after the soft reset is completed 1'b0: No soft reset is generated and the current state is not changed	1'b0

		1'b1: The encryption algorithm is reset to the initial state by software	
[25]	RW	rc4_soft_reset rc4 After the soft reset is completed, the hardware is automatically cleared to 0 1'b0: No soft reset is generated and the current state is not changed 1'b1: The encryption algorithm is reset to the initial state by software	1'b0
[24]	RW	crc_datarev 1'b0: CRC input data is not reversed 1'b1: CRC input data reverse	1'b0
[23]	RW	crc_chksrev 1'b0: CRC output result is not reversed 1'b1: CRC output result is reversed	1'b0
[22:21] RW		sub_mode Algorithm type submode selection: 0: ECB mode of DES/AES encryption algorithm, CRC8 mode of CRC algorithm can be reused 1: CBC of DES/AES cipher algorithm, CRC16_0 mode of reusable CRC algorithm 2: CTR mode of AES encryption algorithm, CRC16_1 mode of CRC algorithm can be reused 3: MAC mode of AES encryption algorithm, CRC32 of CRC algorithm can be reused	2'b0
[20]	RW	encrypt_decrypt Encryption or decryption mode selection for RC4/AES/DES/3DES algorithm: 1'b0: encryption 1'b1: decrypt	1'b0
[19]	RW	gpsec_int_mask	1'b0

		1'b0: Do not mask encryption/decryption completion interrupt 1'b1: Shield encryption/decryption completion interrupt	
[18:16] RW		cypher_mode Cryptographic Algorithm Type 3'b000 ÿ RSV 3'b001ÿRC4 3'b010ÿSHA1 3'b011ÿAES 3'b100ÿDES 3'b101ÿ3DES 3'b110ÿCRC 3'b111ÿMD5	3'b0
[15: 0] RW		total_byte The total number of bytes required for encryption and decryption operations.	16'h0

7.4.3 TRNG Control Register

Table 33 TRNG Module Control Register

bit access		Instructions	reset value
[31: 7]		Reserve	
[6]	RW	TRNG_INT_MASK TRNG interrupt mask	1'b1

		0: TRNG module reports interrupt; 1: The TRNG module does not report interrupts;	
[5:3] RW		TRNG_CP TRNG module control signal	3'd0
[2]	RW	TRNG_DIG_BYPASS TRNG digital post-processing module bypass: 0: TRNG module for post-processing; 1: The TRNG module does not perform post-processing;	1'b0
[1]	RW	TRNG_SEL: RNG output selection signal. 0: Register 0x48 displays the output result of the pseudo-random module; 1: Register 0x48 displays the TRNG output result;	1'b0
[0]	RW	TRNG_EN: TRNG module enable signal. Highly effective. 0: TRNG module stops; 1: The TRNG module starts to work;	1'b0

7.4.4 Control Register

Table 34 Cryptographic Module Control Registers

bit access		Instructions	reset value
[31: 2]		Reserve	

[1]	RW	sec_stop Stop currently ongoing encryption and decryption operations 1'b0: invalid 1'b1: Encryption/decryption stop	1'b0
[0]	RW	sec_strt Start encryption and decryption, after completing the number of bytes of encryption and decryption operations, the hardware will automatically clear 0 1'b0: Do not start encryption/decryption 1'b1: start encryption/decryption	1'b0

7.4.5 Status Register

Table 35 Cryptographic Module Status Register

bit access		Instructions	reset value
[31: 17]		Reserve	
[16]	RW	int_flag Software write 1 to clear 1'b0: do not generate encryption/decryption completion interrupt 1'b1: Generate encryption/decryption completion interrupt	1'b0
[15: 0] RO		transfer_cnt The number of bytes currently encrypted. It is cleared to 0 each time the encryption and decryption is restarted, and the counting starts again.	16'h0

8 RSA encryption module

8.1 Function overview

RSA operation hardware coprocessor, providing Montgomery (FIOS algorithm) modular multiplication function. Implementing RSA Algorithm with RSA Software Library Law. 128-bit to 2048-bit modulo multiplication is supported.

8.2 Main Features

ÿ Support 128-bit to 2048-bit modulo multiplication (the modulo multiplication length is an integer multiple of 32 bits)

ÿ Support D*D; X*Y; D*Y; X*X and other 4 modulo multiplication modes

8.3 Functional Description

8.3.1 Modular multiplication function

RSA operation hardware coprocessor, providing Montgomery (FIOS algorithm) modular multiplication function. Implement RSA together with RSA software library algorithm. 128-bit to 2048-bit modulo multiplication is supported.

8.4 Register Description

8.4.1 Register List

Table 36 RSA register list

offset address name		Abbreviated access		describe	reset value
0X0000~0X00FC Data	X register	XBUF	RW Data	X Register	
0X0100~0X01FC Data	Y register	YBUF	RW Data	Y Register	
0X0200~0X02FC Data	M register	MBUF	RW Data	M Register	
0X0300~0X03FC Data	D register	DBUF	RW Data	D Register	

0X0400	RSA Control Register RSACON	RW		RSA Control Register	0X0000_0000
0X0404	Parameter MC register	RSAMC	WO	parameter MC register	0X0000_0000
0X0408	Parameter N register	RSAN	RW	parameter N register	0X0000_0000

8.4.2 Data X Register

XBUF corresponds to the buffer of data X (2048bit), and the corresponding haddr value is 0000h~00fch. The corresponding rules are as follows:

Table 37 RSA Data X Register

000h	004h	008h	.....	00f8h	00fch
X[31:0]	X[63:32]	X[95:64]	.....	X[2015:1984]	X[2047:2016]

8.4.3 Data Y register

YBUF corresponds to the buffer of data Y (2048bit), and the corresponding haddr value is 0100h~01fch. The corresponding rules are as follows:

Table 38 RSA Data Y Register

0100h	0104h	0108h	.....	01f8h	01fch
AND[31:0]	AND[63:32]	AND[95:64]	.....	AND[2015:1984]	AND[2047:2016]

8.4.4 Data M register

MBUF corresponds to the buffer of data M (2048bit), and the corresponding haddr value is 0200h~02fch. The corresponding rules are as follows:

Table 39 RSA Data M Register

0200h	0204h	0208h	.....	02f8h	02fch
M[31:0]	M[63:32]	M[95:64]	.....	M[2015:1984]	M[2047:2016]

8.4.5 Data D Register

DBUF corresponds to the buffer of data D (2048bit), and the corresponding haddr value is 0300h~03fch. The corresponding rules are as follows:

Table 40 RSA Data D Register

0300h	0304h	0308h	.....	03f8h	03fch
D[31:0]	D[63:32] D[95:64]			D[2015:1984]	D[2047:2016]

8.4.6 RSA Control Register

RSACON, RSA control register, the actual physical space is a 32bit register.

Table 41 RSA Control Register

bit access		Instructions	reset value
[31: 6]		Reserve	
[5]	RW	Modulo multiplication start control bit. The software writes "1" to start the modulo multiplication operation. After the operation is completed, the hardware automatically clears it to "0". 1'b0	
[4]	RW	Provides soft reset function, active high. The software writes "1" to perform a soft reset. After the reset is completed, the hardware automatically clears "0"ÿ 1. Set parameters MC and N to 0. 2. After the modulo multiplication is started (bit5 is set to 1), this bit will be set to 1, and the current operation will be terminated (when bit0 changes to 1). High, indicating that the soft reset command is executed and the operation is terminated), but the internal data buffer (X, Y, M, D) Part of the operation results that have been completed in the .	1'b0
[2 : 3] RW		Modulo multiplication mode selection. 2'b00ÿX = D*D mod M 2'b01ÿD = X*Y mod M 2'b10ÿX = D*Y mod M 2'b11ÿD = X*X mod M	2'b0
[1]	RW	Reserve	1'b0

[0]	RW	Modulo multiplication completion flag, high effective. Set to "1" by hardware and clear to "0" by software. Software writing "1" is invalid.	1'b0

8.4.7 Parameter MC register

Table 42 RSA parameter MC register

bit access		Instructions	reset value
[31:0] WO		RSAMC corresponds to the parameter MC (32bit). The reset value is all 0s. The read value is all 0s.	32'h0

8.4.8 Parameter N register

RSAN corresponds to parameter N (7bit). The N value is the modulo multiplied length divided by 32. That is, if you call the 1024bit modular multiplication operation, you need to set N = 32. When writing to this register, the lower 7 bits are taken as valid data, and when reading, the upper 25 bits are 0. The reset value is all 0s.

Table 43 RSA parameter N register

bit access		Instructions	reset value
[31: 7]		Reserve	
[6 : 0] RW		RSAN corresponds to parameter N (7bit). The N value is the modulo multiplied length divided by 32.	7'h0

9 GPIO module

9.1 Function overview

The GPIO controller implements the configuration of the GPIO properties by software, enabling users to conveniently operate the GPIO.

Each GPIO can be individually configured by software, set it as input port, output port, set its floating, pull-up, pull-down state,

Set its rising edge, falling edge, double edge, high level, low level interrupt trigger mode.

9.2 Main Features

- ÿ Support GPIO software configuration
- ÿ Support GPIO interrupt configuration
- ÿ Provides up to 48 GPIOs available

9.3 Functional Description

The GPIO provided in W800 is divided into two groups, one is GPIOA, the other is GPIOB. The starting addresses of GPIOA and GPIOB registers are different.

Same, but function the same.

When the user wants to use a specific IO as a software-controlled GPIO, set the corresponding position in the GPIO multiplexing selection register to 0, i.e.

Can.

The GPIO direction control register is used to control the direction of the GPIO, 1 means the corresponding GPIO is used as an output pin, 0 means the corresponding GPIO

as an input pin.

The GPIO pull-up and pull-down control registers are used to control the pull-up and pull-down functions of the corresponding IO.

The GPIO pull-up control register is active low, setting it to 0 means to enable the pull-up function of the corresponding IO, and setting it to 1 means to close the pull-up function.

For the attributes of IO, please refer to the IO multiplexing table.

The GPIO pull-down control register is active high. Setting it to 1 means to open the pull-down function of the corresponding IO, and setting it to 0 means to close the pull-down function.

For the attributes of IO, please refer to the IO multiplexing table.

When the GPIO data register is set to the input state, it represents the level of the input IO. When it is set to the output state, 1 or 0 can be written to specify

IO output level. This register is controlled by the GPIO data enable register, only when the GPIO data enable register is set to 1

time, the GPIO data register can be read and written.

The GPIO module provides input signal detection function. High and low level detection and upper and lower edges can be realized by configuring GPIO interrupt related registers

Jump detection. When the input signal corresponding to IO meets the preset conditions, such as high-level trigger or rising edge trigger, etc., it will trigger

GPIO interrupt is reported to MCU for processing. The MCU needs to clear the corresponding interrupt status to avoid false triggering of the interrupt.

9.4 Register Description

9.4.1 Register List

Table 44 GPIOA register list

offset address	name	Abbreviated access		describe	reset value
0X0000	GPIO data register	GPIO_DATA	RW	Read and write GPIO current data	0X180B
0X0004	GPIO data enable register	GPIO_DATA_EN	RW	Configure the enable bit of GPIO_DATA	0XFFFF
0X0008	GPIO direction control register	GPIO_DIR	RW	configure GPIO direction	0X0000
0X000C	GPIO Pull-Up Control Register	GPIO_PULL_EN	RW	configure GPIO pull-up	0XFFFF
0X0010	GPIO multiplexing selection register	GPIO_AF_SEL	RW	Configure GPIO alternate function enable bit	0XFFFF

0X0014	GPIO multiplexing selection register 1	GPIO_SF_S1	RW	GPIO alternate function selection bit high address bit	0X0000
0X0018	GPIO multiplexing select register 0	GPIO_AF_S0	RW	GPIO alternate function selection bit low address bit	0X0000
0X001C	GPIO pull-down control register	GPIO_DN_ENA	RW	Configure GPIO pull-down	0X0000
0X0020	GPIO interrupt trigger mode configuration register	GPIO_IS	RW	Configure the interrupt triggering method of GPIO	0X0000
0X0024	GPIO interrupt edge-triggered mode configuration register	GPIO_IBE	RW	Configure GPIO interrupt edge trigger mode	0X0000
0X0028	GPIO interrupt upper and lower edge trigger configuration register	GPIO_IEV	RW	Configure GPIO interrupt upper and lower edge trigger or high and low level trigger	0X0000
0X002C	GPIO interrupt enable configuration register	GPIO_IE	RW	configure GPIO interrupt enable	0X0000
0X0030	GPIO Bare Interrupt Status Register	GPIO_RIS	RO	Query GPIO raw interrupt status (before MASK)	0X0000
0X0034	GPIO masked interrupt status register	GPIO_MIS	RO	Query the interrupt status after GPIO masking (MASK Rear)	0X0000
0X0038	GPIO interrupt clear control register	GPIO_IC	WO	controls GPIO interrupt clearing	0X0000

Table 45 GPIOB register list

offset address	name	Abbreviated access		describe	reset value
0X0000	GPIO data register	GPIO_DATA	RW	Read and write GPIO current data	0X0000_7304
0X0004	GPIO data enable register	GPIO_DATA_EN	RW	Configure the enable bit of GPIO_DATA	0X7FFF_FFFF
0X0008	GPIO direction control register	GPIO_DIR	RW	configure GPIO direction	0X0000_0000
0X000C	GPIO Pull-Up Control Register	GPIO_PULL_EN	RW	configure GPIO pull-up	0XFFFF_FFFF

0X0010	GPIO multiplexing selection register	GPIO_AF_SEL	RW	Configure GPIO alternate function enable bit 0XFFFF_FFFF	
0X0014	GPIO multiplexing selection register 1	GPIO_SF_S1	RW	GPIO alternate function selection bit high address bit 0X0000_0000	
0X0018	GPIO multiplexing select register 0	GPIO_AF_S0	RW	GPIO alternate function selection bit low address bit 0X0000_0000	
0X001C	GPIO pull-down control register	GPIO_DN_ENA	RW	Configure GPIO pull-down	0X0000_0000
0X0020	GPIO interrupt trigger mode configuration register	GPIO_IS	RW	Configure the interrupt trigger mode of GPIO 0X0000_0000	
0X0024	GPIO interrupt edge-triggered mode configuration register	GPIO_IBE	RW	Configure GPIO interrupt edge trigger mode 0X0000_0000	
0X0028	GPIO interrupt upper and lower edge trigger configuration register	GPIO_IEV	RW	Configure GPIO interrupt to be edge-triggered or High and low level trigger	0X0000_0000
0X002C	GPIO interrupt enable configuration register	GPIO_IE	RW	configure GPIO interrupt enable	0X0000_0000
0X0030	GPIO Bare Interrupt Status Register	GPIO_RIS	RO	queries GPIO bare interrupt status (MASK forward)	0X0000_0000
0X0034	GPIO masked interrupt status register	GPIO_MIS	RO	Query the interrupt status after GPIO masking (after MASK)	0X0000_0000
0X0038	GPIO interrupt clear control register	GPIO_IC	WO	controls GPIO interrupt clearing	0X0000_0000

9.4.2 GPIO data register

Table 46 GPIOA data register

bit access		Instructions	reset value
[15: 0]	RW	GPIO current data, each BIT corresponds to the corresponding GPIO line	16'h180b

Table 47 GPIOB data register

bit access		Instructions	reset value
[31: 0]	RW	GPIO current data, each BIT corresponds to the corresponding GPIO line	32'h7304

9.4.3 GPIO Data Enable Register

Table 48 GPIOA data enable register

bit access		Instructions	reset value
[15: 0]	RW	Corresponding to the BIT enable bit of GPIO_DATA, only when the corresponding BIT is 1, the operation of the corresponding bit of GPIO_DATA It is valid only after the operation, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, GPIO_DATA[x] cannot be read or written [x] = 1, GPIO_DATA[x] can be read and written	16'hffff

Table 49 GPIOB data enable register

bit access		Instructions	reset value
[31: 0]	RW	Corresponding to the BIT enable bit of GPIO_DATA, only when the corresponding BIT is 1, the operation of the corresponding bit of GPIO_DATA It is valid only after the operation, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, GPIO_DATA[x] cannot be read or written [x] = 1, GPIO_DATA[x] can be read and written	32'h7fff_ffff

9.4.4 GPIO direction control register

Table 50 GPIOA direction control register

bit access		Instructions	reset value
[15: 0]	RW	GPIO direction control, each BIT corresponds to the corresponding GPIO line, 1'bx:	16'h0

		[x] = 0, GPIO[x] is input [x] = 1, GPIO[x] is output	

Table 51 GPIOB direction control register

bit access		Instructions	reset value
[31: 0]	RW	GPIO direction control, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, GPIO[x] is input [x] = 1, GPIO[x] is output	32'h0

9.4.5 GPIO pull-up and pull-down control registers

Table 52 GPIOA pull-up control register

bit access		Instructions	reset value
[15: 0]	RW	GPIO pull-up control, each BIT corresponds to the corresponding GPIO line, 1'bx: Note: This register is active low [x] = 0, GPIO[x] has pull-up [x] = 1, no pull-up on GPIO[x]	16'hffff

bit access		Instructions	reset value
[15: 0]	RW	GPIO pull-down control, each BIT corresponds to the corresponding GPIO line, 1'bx: Note: This register is active high [x] = 1, GPIO[x] has pull-down [x] = 0, GPIO[x] has no pull-down	16'h0000

Table 53 GPIOB pull-up and pull-down control registers

bit access		Instructions	reset value
[31: 0]	RW	GPIO pull-up control, each BIT corresponds to the corresponding GPIO line, 1'bx: Note: This register is active low [x] = 0, GPIO[x] has pull-up [x] = 1, no pull-up on GPIO[x]	32'hffff_ffff

bit access		Instructions	reset value
[31: 0]	RW	GPIO pull-down control, each BIT corresponds to the corresponding GPIO line, 1'bx: Note: This register is active high [x] = 1, GPIO[x] has pull-down [x] = 0, GPIO[x] has no pull-down	32'h0000_0000

9.4.6 GPIO multiplexing selection register

Table 54 GPIOA multiplexing selection register

bit access		Instructions	reset value
[15: 0]	RW	GPIO multiplexing function enable bit, each BIT corresponds to whether the corresponding GPIO multiplexing function is enabled, 1'bx:	16'hffff

		[x] = 0, GPIO[x] alternate function is disabled [x] = 1, GPIO[x] alternate function is enabled When [x] = 1, the alternate function depends on the corresponding BITs of the two registers GPIO_AF_S1 and GPIO_AF_S0 status. S1.[x] = 0, S0.[x] = 0, alternate function 1 (opt1) S1.[x] = 0, S0.[x] = 1, alternate function 2 (opt2) S1.[x] = 1, S0.[x] = 0, alternate function 3 (opt3) S1.[x] = 1, S0.[x] = 1, multiplexing function 4 (opt4) When [x] = 0, if GPIO_DIR[x] = 0, and GPIO_PULL_EN[x] = 1, the GPIO multiplexing is opt6 analog IO function For the IO multiplexing function, please refer to the chip pin multiplexing relationship	

Table 55 GPIOB multiplexing selection register

bit access		Instructions	reset value
[31: 0]	RW	GPIO multiplexing function enable bit, each BIT corresponds to whether the corresponding GPIO multiplexing function is enabled, 1'bx: [x] = 0, GPIO[x] alternate function is disabled [x] = 1, GPIO[x] alternate function is enabled When [x] = 1, the alternate function depends on the corresponding BITs of the two registers GPIO_AF_S1 and GPIO_AF_S0 status. S1.[x] = 0, S0.[x] = 0, alternate function 1 (opt1)	32'hffff_ffff

		S1.[x] = 0, S0.[x] = 1, alternate function 2 (opt2) S1.[x] = 1, S0.[x] = 0, alternate function 3 (opt3) S1.[x] = 1, S0.[x] = 1, multiplexing function 4 (opt4) When [x] = 0, if GPIO_DIR[x] = 0, and GPIO_PULL_EN[x] = 1, the GPIO multiplexing is opt6 analog IO function For the IO multiplexing function, please refer to the chip pin multiplexing relationship	

9.4.7 GPIO multiplexing selection register 1

Table 56 GPIOA multiplexing selection register 1

bit access		Instructions	reset value
[15: 0]	RW	The high address bit of the GPIO alternate function selection bit, together with GPIO_AF_S0 to determine the alternate function For the IO multiplexing function, please refer to the chip pin multiplexing relationship	16'h0

Table 57 GPIOB multiplexing selection register 1

bit access		Instructions	reset value
[31: 0]	RW	The high address bit of the GPIO alternate function selection bit, together with GPIO_AF_S0 to determine the alternate function For the IO multiplexing function, please refer to the chip pin multiplexing relationship	32'h0

9.4.8 GPIO multiplexing selection register 0

Table 58 GPIOA multiplexing selection register 0

bit access		Instructions	reset value
[15: 0]	RW	The low address bit of the GPIO alternate function selection bit, and GPIO_AF_S1 together determine the alternate function How to configure see GPIO_AF_SEL register description	16'h0

Table 59 GPIOB multiplexing selection register 0

bit access		Instructions	reset value
[31: 0]	RW	The low address bit of the GPIO alternate function selection bit, and GPIO_AF_S1 together determine the alternate function How to configure see GPIO_AF_SEL register description	32'h0

9.4.9 GPIO interrupt trigger mode configuration register

Table 60 GPIOA interrupt trigger mode configuration register

bit access		Instructions	reset value
[15: 0]	RW	The interrupt triggering method of GPIO, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, GPIO[x] interrupt is edge-triggered [x] = 1, GPIO[x] interrupt is level sensitive	16'h0

Table 61 GPIOB interrupt trigger mode configuration register

bit access		Instructions	reset value
[31: 0]	RW	The interrupt triggering method of GPIO, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, GPIO[x] interrupt is edge-triggered	32'h0

		[x] = 1, GPIO[x] interrupt is level sensitive	

9.4.10 GPIO Interrupt Edge Triggered Mode Configuration Register

Table 62 GPIOA Interrupt Edge Triggered Mode Configuration Register

bit access		Instructions	reset value
[15: 0]	RW	GPIO interrupt edge trigger mode, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, GPIO[x] edge-triggered interrupt mode is determined by GPIO_IEV [x] = 1, both edges of GPIO[x] trigger an interrupt	16'h0

Table 63 GPIOB Interrupt Edge Triggered Mode Configuration Register

bit access		Instructions	reset value
[31: 0]	RW	GPIO interrupt edge trigger mode, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, GPIO[x] edge-triggered interrupt mode is determined by GPIO_IEV [x] = 1, both edges of GPIO[x] trigger an interrupt	32'h0

9.4.11 GPIO interrupt upper and lower edge trigger configuration register

Table 64 GPIOA interrupt upper and lower edge trigger configuration register

bit access		Instructions	reset value
[15: 0]	RW	GPIO interrupt upper and lower edge trigger or high and low level trigger selection, each BIT corresponds to the corresponding GPIO line, 1'bxÿ [x] = 0, GPIO[x] interrupt is low level or falling edge triggered [x] = 1, GPIO[x] interrupt is high or rising edge triggered	16'h0

Table 65 GPIOB interrupt upper and lower edge trigger configuration register

bit access		Instructions	reset value
[31: 0]	RW	GPIO interrupt upper and lower edge trigger or high and low level trigger selection, each BIT corresponds to the corresponding GPIO line, 1'bxÿ [x] = 0, GPIO[x] interrupt is low level or falling edge triggered [x] = 1, GPIO[x] interrupt is high or rising edge triggered	32'h0

9.4.12 GPIO Interrupt Enable Configuration Register

Table 66 GPIOA Interrupt Enable Configuration Register

bit access		Instructions	reset value
[15: 0]	RW	GPIO interrupt enable control, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, GPIO[x] interrupt disabled [x] = 1, GPIO[x] interrupt enable	16'h0

Table 67 GPIOB Interrupt Enable Configuration Register

bit access		Instructions	reset value
[31: 0]	RW	GPIO interrupt enable control, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, GPIO[x] interrupt disabled [x] = 1, GPIO[x] interrupt enable	32'h0

9.4.13 GPIO Raw Interrupt Status Register

Table 68 GPIOA Raw Interrupt Status Register

bit access		Instructions	reset value
[15: 0]	RW	GPIO bare interrupt status (before MASK), each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, no interrupt is generated for GPIO[x] [x] = 1, GPIO[x] has an interrupt	16'h0

Table 69 GPIOB Raw Interrupt Status Register

bit access		Instructions	reset value
[31: 0]	RW	GPIO bare interrupt status (before MASK), each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, no interrupt is generated for GPIO[x] [x] = 1, GPIO[x] has an interrupt	32'h0

9.4.14 GPIO Masked Interrupt Status Register

Table 70 GPIOA Masked Interrupt Status Register

bit access		Instructions	reset value
[15: 0]	RW	Interrupt status after GPIO masking (after MASK), each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, no interrupt is generated for GPIO[x] (after MASK) [x] = 1, GPIO[x] interrupt is generated (after MASK)	16'h0

Table 71 GPIOB Masked Interrupt Status Register

bit access		Instructions	reset value
[31: 0]	RW	Interrupt status after GPIO masking (after MASK), each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, no interrupt is generated for GPIO[x] (after MASK) [x] = 1, GPIO[x] interrupt is generated (after MASK)	32'h0

9.4.15 GPIO Interrupt Clear Control Register

Table 72 GPIOA Interrupt Clear Control Register

bit access		Instructions	reset value
[15: 0]	RW	GPIO interrupt clear control, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, no action [x] = 1, clear GPIO[x] interrupt status	16'h0

Table 73 GPIOB Interrupt Clear Control Register

bit access		Instructions	reset value
[31: 0]	RW	GPIO interrupt clear control, each BIT corresponds to the corresponding GPIO line, 1'bx: [x] = 0, no action [x] = 1, clear GPIO[x] interrupt status	32'h0

10 High Speed SPI Device Controller

10.1 Function overview

Compatible with the general SPI physical layer protocol, by agreeing on the data format for interaction with the host, the host can access the device at high speed, up to The working frequency is 50MHZ.

10.2 Main Features

- ÿ Compatible with general SPI protocol
- ÿ Selectable level interrupt signal
- ÿ Support up to 50Mbps rate
- ÿ Simple frame format, full hardware parsing and DMA

10.3 Functional Description

10.3.1 Introduction to SPI Protocol

SPI works in master-slave mode, usually there is a master device and one or more slave devices, requiring at least 4 wires, in fact 3 wires are also possible (single when transferring). They are SDI (data input), SDO (data output), SCLK (clock), CS (chip select).

(1) SDI – Serial Data In, serial data input

(2) SDO – Serial Data Out, serial data output

(3) SCLK – Serial Clock, clock signal, generated by the master device

(4) CS – Chip Select, the slave device enable signal, controlled by the master device.

Among them, CS is the control signal of whether the slave chip is selected by the master chip, that is to say, only when the chip select signal is the predetermined enable signal (high

potential or low potential), the operation of the master chip is valid for this slave chip. This makes it possible to connect multiple SPI devices on the same bus.

In addition to the above four signal lines, HSPI also adds an INT line, which generates a drop when the slave device has data to upload.

The interruption of the edge realizes the active reporting of data.

SPI communication is accomplished through data exchange, the data is transmitted bit by bit, the clock pulse is provided by SCLK, SDI, SDO are based on

This pulse completes the data transfer. The data output goes through the SDO line, the data changes on the rising or falling edge of the clock, and the data changes on the following falling edge or

rising edge is read. To complete one-bit data transmission, the same principle is used for input. Therefore, at least 8 changes of the clock signal (the rising edge

and the lower edge is once), to complete the transmission of 8-bit data.

The SCLK signal line is controlled by the master device, and the slave device cannot control the signal line. In an SPI-based device, there is at least one master device.

10.3.2 SPI working process

The HSPI inside the chip works with the wrapper controller. The wrapper controller integrates DMA and is implemented through DMA.

Data exchange between HSPI internal FIFO and chip internal buffer. This operation is implemented in hardware, and software does not need to care about data transmission and reception

In the process, you only need to configure the sending and receiving data linked list, and operate the corresponding registers of the wrapper controller.

For a detailed introduction to the wrapper controller, please refer to the relevant chapters.

10.4 Register Description

10.4.1 Register List of Internal Operation of HSPI Chip

Table 74 HSPI Internal Access Registers

offset address	name	abbreviation	access	describe	reset value

0X0000	HSPI FIFO clear register CLEAR_FIFO		RW	clears the contents of the Tx and Rx FIFOs, while A circuit that synchronously resets the system clock domain	0X0000_0000
0X0004	HSPI Configuration Register	SPI_CFG	RW	configures the transmission mode and endianness of SPI set	0X0000_0000
0X0008	HSPI Mode Configuration Register MODE_CFG		RW	Configure ahb master to access the bus burst length	0X0000_0000
0X000C	HSPI Interrupt Configuration Register	SPI_INT_CPU_ MASK	RW	configuration interrupt is enabled	0X0000_0003
0X0010	HSPI Interrupt Status Register	SPI_INT_CPU_ STTS	RW	Get and clear interrupt status	0X0000_0000
0X0018	HSPI data upload length register RX_DAT_LEN		RW	Configure the length of data that can be uploaded	0X0000_0000

10.4.1.1 HSPI FIFO clear register

Table 75 HSPI FIFO clear register

bit access		Instructions	reset value
[31: 1] RO		Reserve	
[0]	RW	Clear FIFOs, clear the contents of the Tx and Rx FIFOs, and synchronously reset the circuit of the system clock domain (this Except for the registers in the list) 0: Do not clear the FIFO 1: Clear valid Set by software, cleared by hardware Note: If you want to reset the whole circuit, you need to use the asynchronous reset leg of this module: rst_n	1'b0

10.4.1.2 HSPI Configuration Register

Table 76 HSPI configuration registers

bit access		Instructions	reset value
[31: 4] RO		Reserve	
[3]	RW	Bigendian, spi interface supports big and small endian selection of data. 0: support small segment data transfer 1: Support big-endian data transmission	1'b0
[2]	RW	spi_tx_always_drive 0: The spi output is only valid when the chip select is valid, and it is high impedance at other times 1: spi output is always valid	1'b0
[1]	RW	SPI CPHA 0: Transmission mode A 1: Transmission mode B	1'b0
[0]	RW	SPI CPOL, SCK polarity at IDLE 0: 0 when SCK IDLE 1: 1 when SCK IDLE	1'b0

10.4.1.3 HSPI Mode Configuration Register

Table 77 HSPI Mode Configuration Register

bit access		Instructions	reset value
[31: 1] RO		Reserve	

[0]	RW	Burst len, the burst length when the ahb master accesses the bus 0: burst len is 1 word 1: burst len is 4 characters It is recommended to set the burst transmission of 4 words, so that when the frequency of the spi interface is high, the continuous flow can be guaranteed.	1'b0

10.4.1.4 HSPI Interrupt Configuration Register

Table 78 HSPI Interrupt Configuration Register

bit access		Instructions	reset value
[31: 2] RO		Reserve	
[1]	RW	IntEnRxOverrun, RxOverrun interrupt enable 0: Rx FIFO overflow interrupt enable 1: Rx FIFO overflow interrupt disabled	1'b1
[0]	RW	IntEnTxUnderrun, TxUnderrun interrupt enable 0: Tx FIFO underflow interrupt enable 1: Tx FIFO underflow interrupt disabled	1'b1

10.4.1.5 HSPI Interrupt Status Register

Table 79 HSPI Interrupt Status Register

bit access		Instructions	reset value
[31: 2] RO		Reserve	
[1]	RW	RxOverrun 0ÿRx FIFO overflow	1'b0

		1ýRx FIFO overflow Write 1 to clear	
[0]	RW	TxUnderrun 0ýTx FIFO underflow 1ýTx FIFO underflow Write 1 to clear	1'b0

10.4.1.6 HSPI data upload length register

Table 80 HSPI data upload length register

bit access		Instructions	reset value
[31:16] RO		Reserve	
[15: 0] RW		Rx_dat_len Indicates the length of data that can be uploaded, in bytes The upload length is an integer multiple of words. If the upload length is less than a whole word, it will be rounded up.	16'h0

10.4.2 Host side access HSPI controller register list

The host side accesses the SPI interface registers through a fixed SPI command format. The command length is fixed to one byte, and the data length is fixed to two byte.

Table 81 HSPI Interface Configuration Register (Master Access)

offset address	name	abbreviation	access	describe	reset value

0X02 Get data length register		RX_DAT_LEN	RO	When uploading data, the spi host is used to obtain data from The length of the data read by the device side	0X0000
0X03 Send data flag register		TX_BUFF_AVAIL	RO	When the master sends data to the slave, it is used to judge whether it can be Download data or commands	0X0000
0X04 Reserved		RSV	RO		
0X05 Interrupt configuration register		SPI_INT_HOST_MASK RW whether to mask the interrupt			0X0000
0X06 Interrupt Status Register		SPI_INT_HOST_STTS	RO	Interrupt status register, the spi host polls this bit Check if there is data to upload	0X0000
0X07 Reserved		RSV	RO		
0X00 Data port 0		DAT_PORT0	RW	The Spi master communicates to the slave device through this register port To send data, the previous data frame is sent using this port	
0X10 Data port 1		DAT_PORT1	RW	The Spi master communicates to the slave device through this register port To send data, send the last data frame to use the port	
0X01 Command port 0		DN_CMD_PORT0	WHERE	The Spi host sends the slave device through this register Command data, use the previous command data the port	
0X11 Command port 1		DN_CMD_PORT1	WHERE	The Spi host sends the slave device through this register Command data, send the last frame of command data to make use this port	

10.4.2.1 HSPI get data length register

Table 82 HSPI get data length register

Rank	access	Instructions	reset value
[15: 0] RO		spi host read-only register, when uploading data, it is mainly used to know how much data is read from the device side But in this module, the upload length is an integer multiple of words. If the upload length value is not an integer word, the host will read Round up when counting, that is, read some redundant bytes	16'h0

10.4.2.2 HSPI send data flag register

Table 83 HSPI send data flag register

Rank	access	Instructions	reset value
[15: 2] RO		Reserve	
[1]	RO	tx_cmdbuff_avail Indicates whether the buff of sending cmd is available, if available, the host can send cmd. 0: Send buff is not available 1: send buff available	1'b0
[0]	RO	tx_buff_avail Indicates whether the sending buff is available, if available, the host can send data. 0: Send buff is not available 1: send buff available	1'b0

10.4.2.3 HSPI Interrupt Configuration Register

Table 84 HSPI Interrupt Configuration Register

Rank	access	Instructions	reset value
[15: 1] RO		Reserve	
[0]	RO	IntMaskup_dat_cmd_rdy interrupt mask 0: Interrupts are not masked, interrupts can be generated 1: Interrupts are masked Note: It is recommended to use the host's own internal interrupt mask, which can improve efficiency.	1'b0

10.4.2.4 HSPI Interrupt Status Register

Table 85 HSPI Interrupt Status Register

Rank	access	Instructions	reset value
[15: 1] RO		Reserve	
[0]	RO	up_dat_cmd_rdy Status register for generating interrupt to SPI master 0: Data or command not ready 1: Data or command is ready Readable	1'b0

10.4.2.5 HSPI Data Port 0

Table 86 HSPI data port 0

Rank	access	Instructions	reset value
	RW	The SPI host transmits data through the register port and the device, and writes data to this register to send the data. Data can be uploaded by reading from this register. If the frame being transmitted requires multiple transmissions to complete If successful, the register port DAT_PORT1 is used for the last transfer, and DAT_PORT0 is used for the others.	

10.4.2.6 HSPI Data Port 1

Table 87 HSPI Data Port 1

Rank	access	Instructions	reset value
	RW	The SPI host transmits data through the register port and the device, and writes data to this register to send the data. Data can be uploaded by reading from this register. If the frame being transmitted requires multiple transmissions to complete If successful, the register port DAT_PORT1 is used for the last transfer, and DAT_PORT0 is used for the others.	

10.4.2.7 HSPI Command Port 0

Table 88 HSPI Command Port 0

Rank	access	Instructions	reset value
	RW	The SPI host interacts with the device through this register port, and writes data to this register to issue a command. make. If the command being transferred requires multiple transfers to complete, the last transfer takes the register Port DN_CMD_PORT1, others use DN_CMD_PORT0. Note: This window is only used to issue commands negotiated by the driver and firmware.	

10.4.2.8 HSPI Command Port 1

Table 89 HSPI Command Port 1

Rank	access	Instructions	reset value
	RW	The SPI host interacts with the device through this register port, and writes data to this register to issue a command. make. If the command being transferred requires multiple transfers to complete, the last transfer takes the register Port DN_CMD_PORT1, others use DN_CMD_PORT0. Note: This window is only used to issue commands negotiated by the driver and firmware.	

10.4.3 High Speed SPI Device Controller Interface Timing

Mainly describe the SPI read and write timing, and how the main SPI and HSPI interact with each other.

10.4.3.1 Data Format

The data format is divided into two parts: command field and data field, as shown in the figure below. The fixed length of the command field is 8 bits, and the length of the data field is based on the access object.
Different, different length, see below for details.

The highest bit of the command field is the read and write flag bit, and the other 7 bits are the address.

- ÿ 0 means read data from the following 7bit address
- ÿ 1 means write data to the next 7bit address

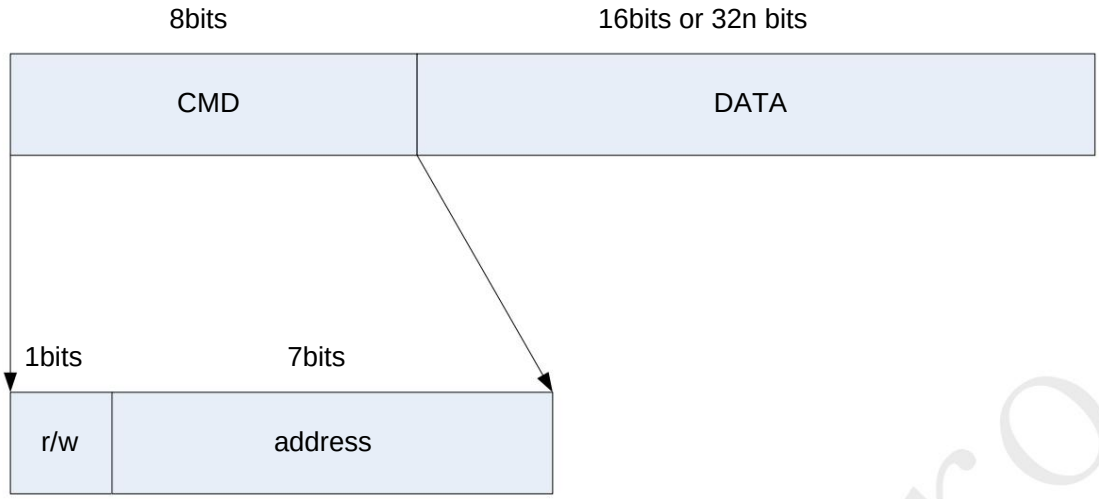


Figure 5 SPI transceiver data format of the host computer

The data field of this module only supports two lengths, the host computer SPI access interface configuration register (Table 2), the data field length is 16bit;

Transfer data through ports (data port 0, data port 1, command port 0 and command port 1), the data field length is an integer of 32 bits

times;

The following figure shows the timing diagram of reading and writing the interface configuration register. The default configuration of the slave device is little endian mode.

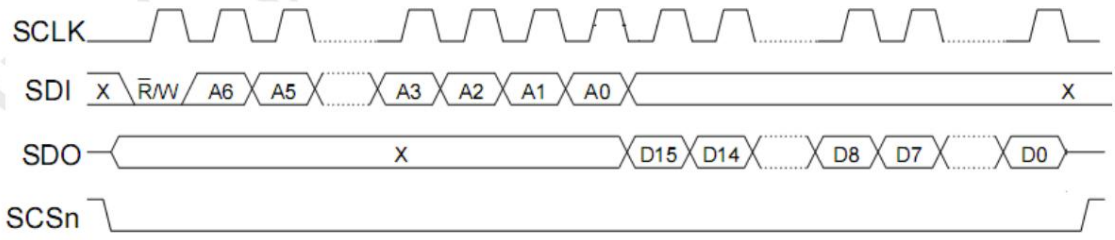


Figure 6 HSPI register read operation (big endian mode)

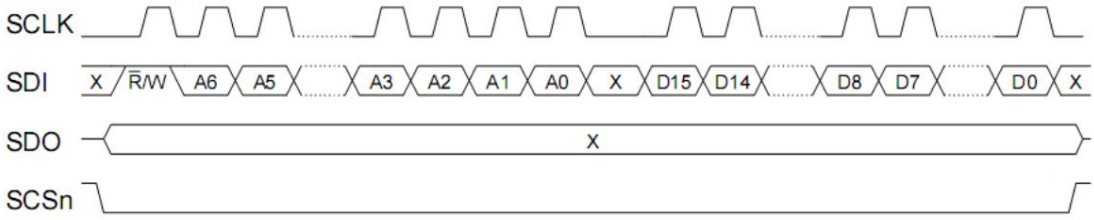


Figure 7 HSPI register write operation (big endian mode)

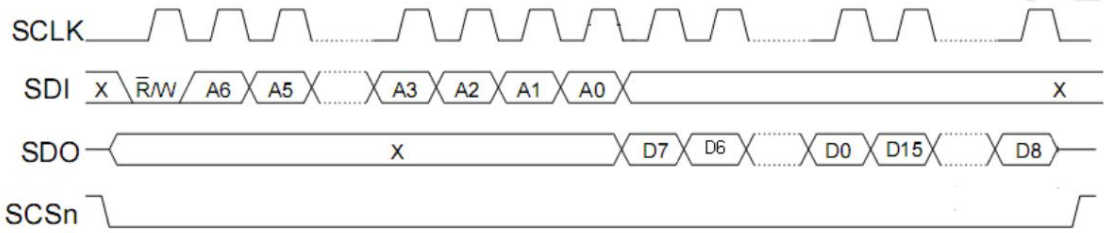


Figure 8 Register read operation (little endian mode)

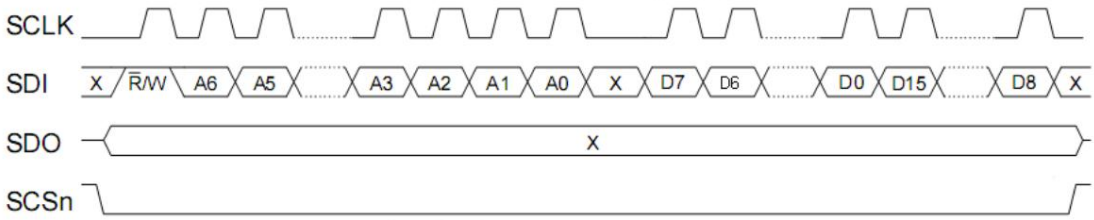


Figure 9 Register write operation (little endian mode)

The following figure is the sequence diagram of reading and writing data, the length of the data field is an integer multiple of 32bit, and the figure only transmits the length of one word.

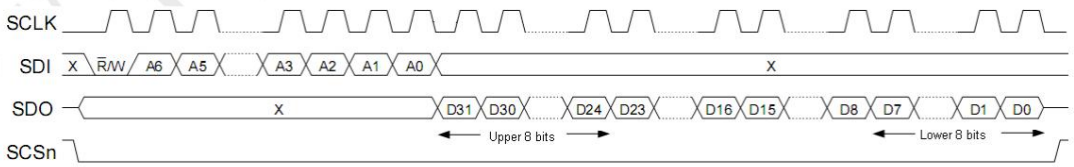


Figure 10 Port read operation (big endian mode)

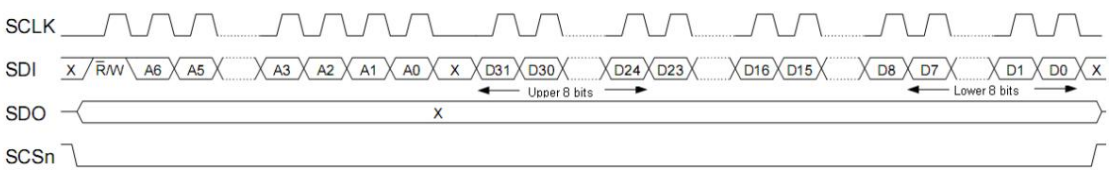


Figure 11 Port write operation (big endian mode)

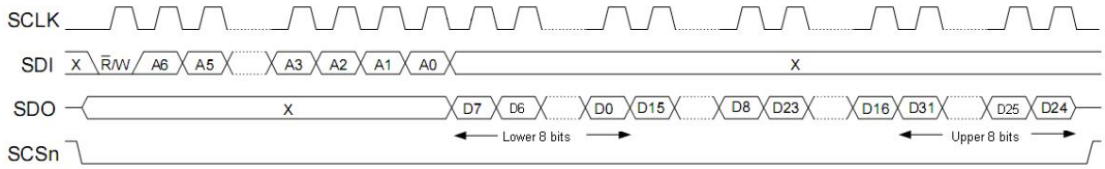


Figure 12 Port read operation (little endian mode)

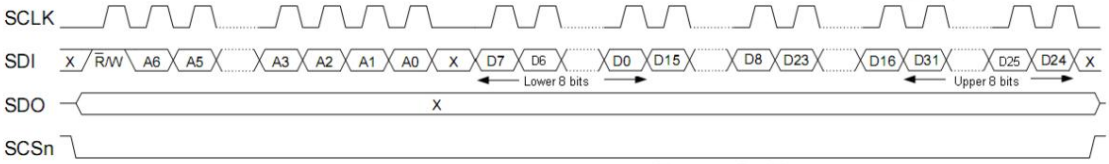


Figure 13 Port write operation (little endian mode)

Note: There can be no waiting time between the command and the data, that is, after the command field is transmitted, the data transmission can be followed, and there is no need for redundant idle clocks

or free time. A time delay is fine, but no idle clocks can appear.

10.4.3.2 Timing

This module supports half-duplex, and the supported timings are divided into 4 types according to the clock phase and sampling point. The following timings are given only for clocks

phase and sampling relationship. It should be noted that the chip supports it by default (CPOL=0, CPHA=0).

Note: There can be no waiting time between the command and the data, that is, after the command field is transmitted, the data transmission can be followed, and there is no need for redundant idle clocks

or free time. A time delay is fine, but no idle clocks can appear.

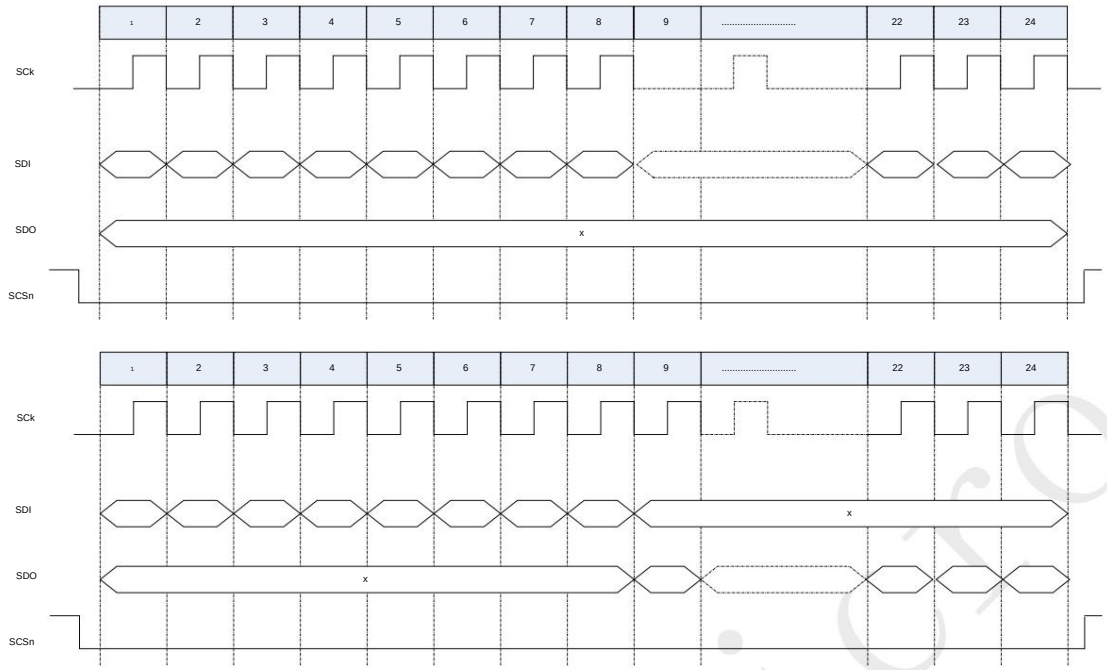


Figure 14 CPOL=0, CPHA=0

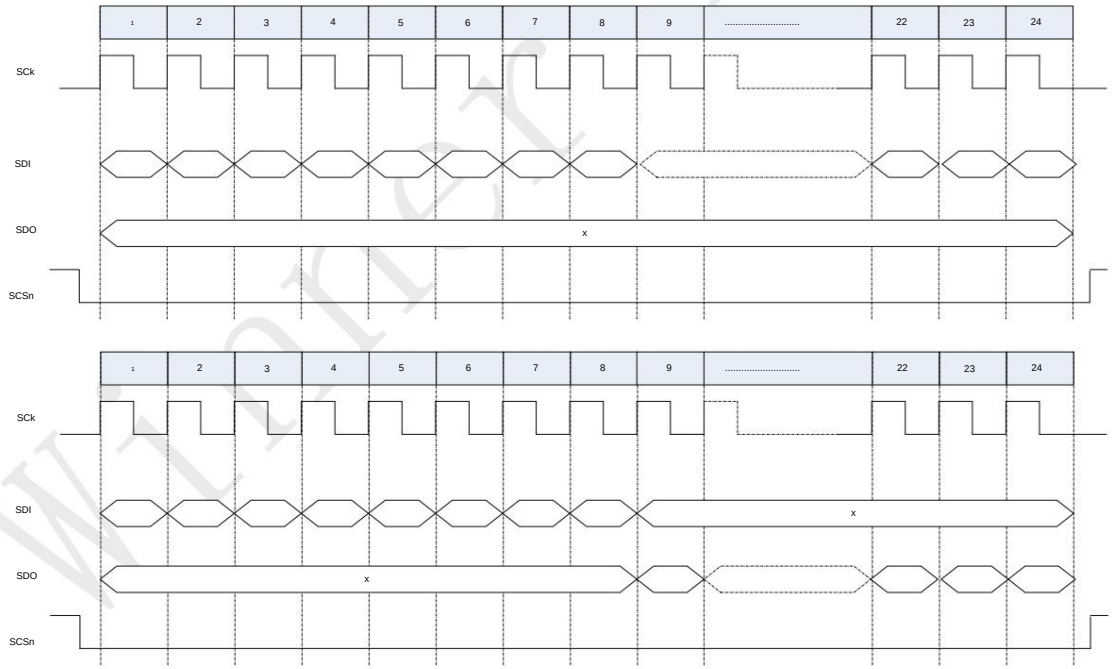


Figure 15 CPOL=0, CPHA=1

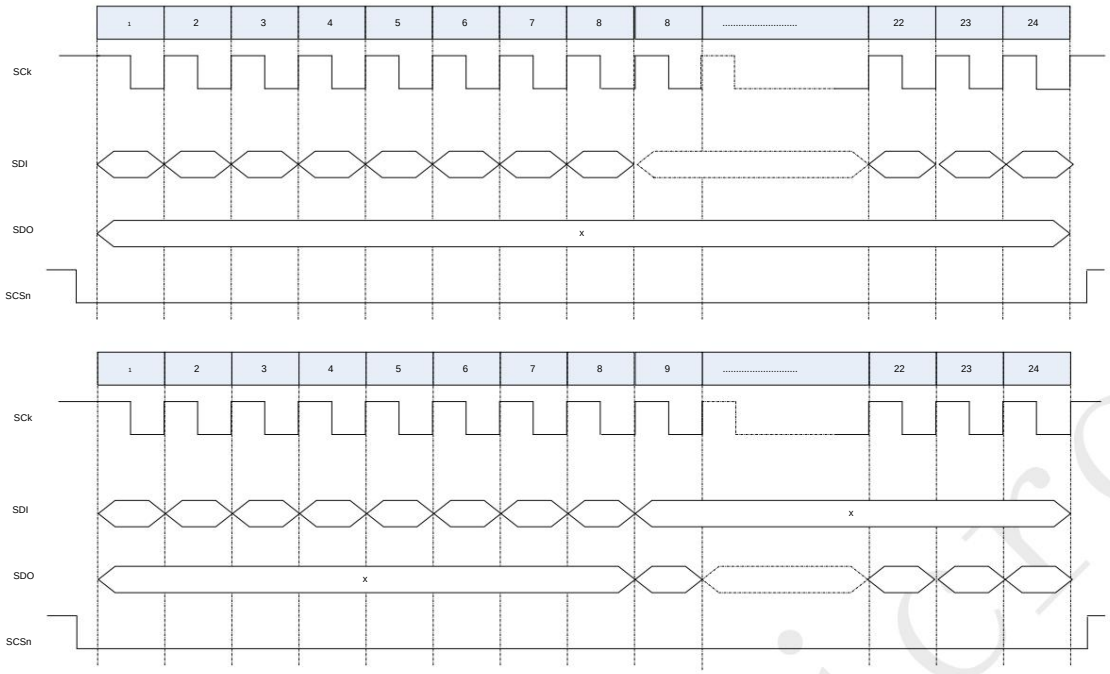


Figure 16 CPOL=1, CPHA=0

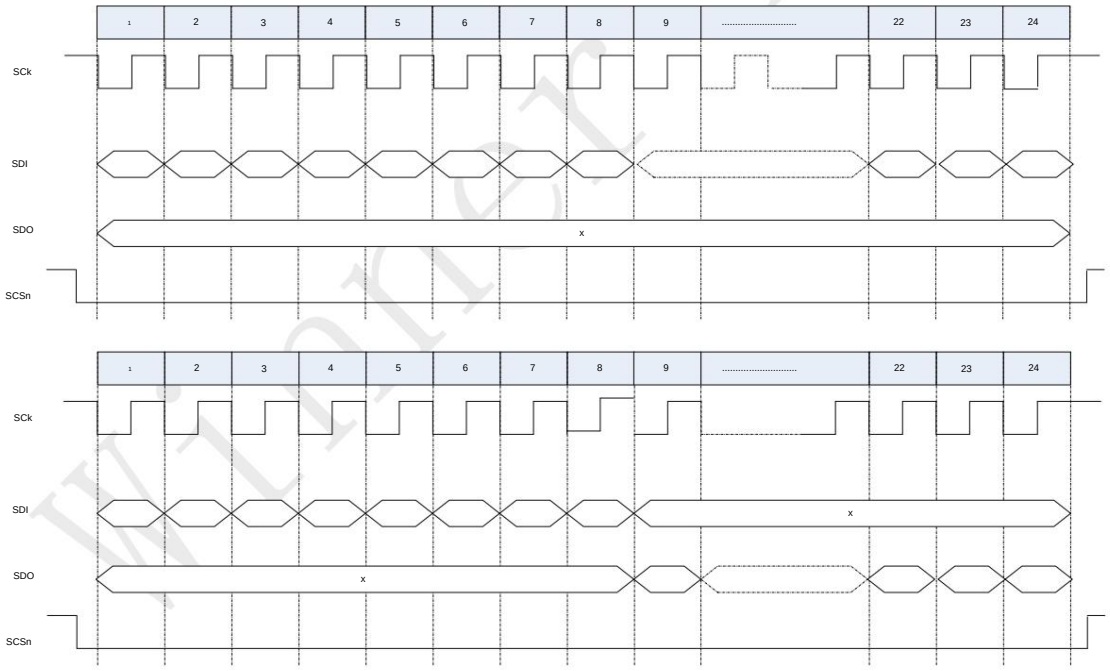


Figure 17 CPOL=1, CPHA=1

10.4.3.3 Interrupts

The interrupt signal is sent by the slave device to the master device, triggered by the SPI_INT pin, active low.

spi_int mainly informs the spi host that there is data or commands that need to be uploaded. The interface registers that the spi host cares about when processing interrupts are:

ÿ SPI_INT_HOST_MASK

ÿ SPI_INT_HOST_STTS

ÿ RX_DAT_LEN

Note: Each uploaded frame corresponds to an interrupt. Only after the frame transmission that needs to be uploaded is completed, if there are still frames to be uploaded, at this time, the

Generate a new interrupt. The diagram below is one way to handle interrupts.

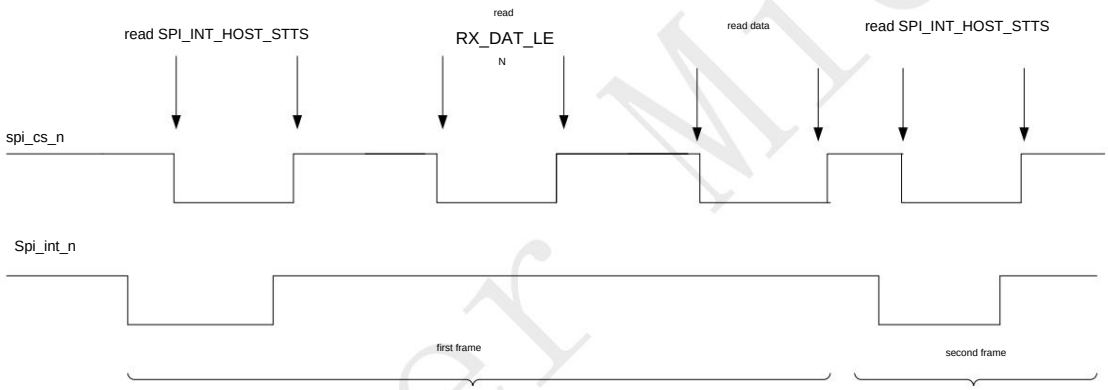


Figure 18 Main SPI processing interrupt flow

10.4.3.4 Main SPI Transceiver Data Workflow

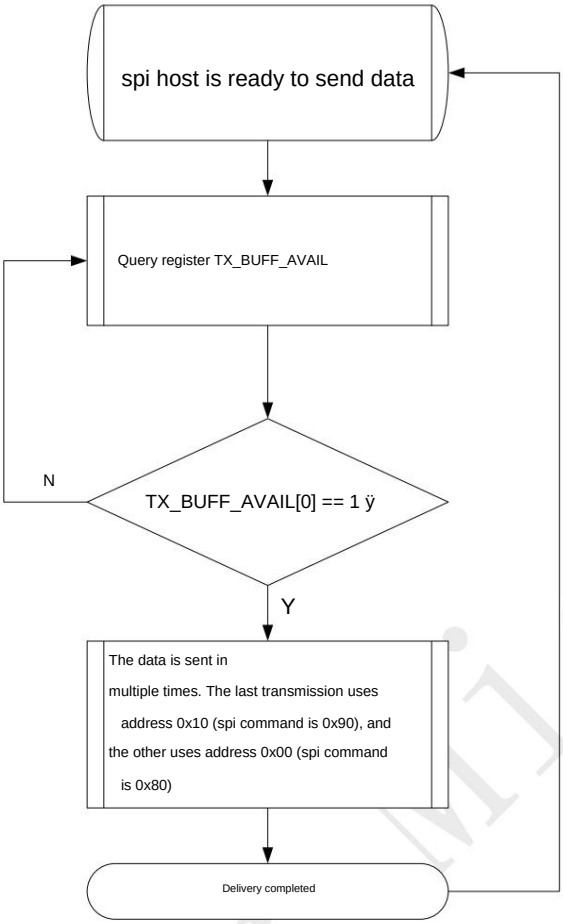


Figure 19 Downlink data flow chart

Note: The length of the data to be sent must be in word units. If it is not a whole word, fill in 0 and complete it.

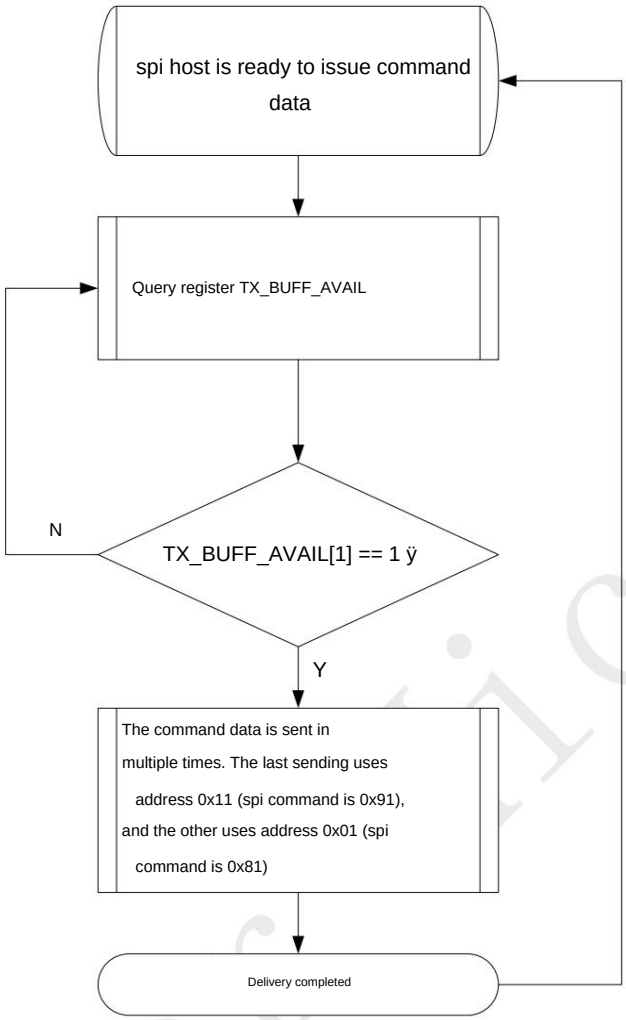


Figure 20 Downlink command flow chart

Note: The length of the issued command must be in word units. If it is not a whole word, fill in 0 and make up for it.

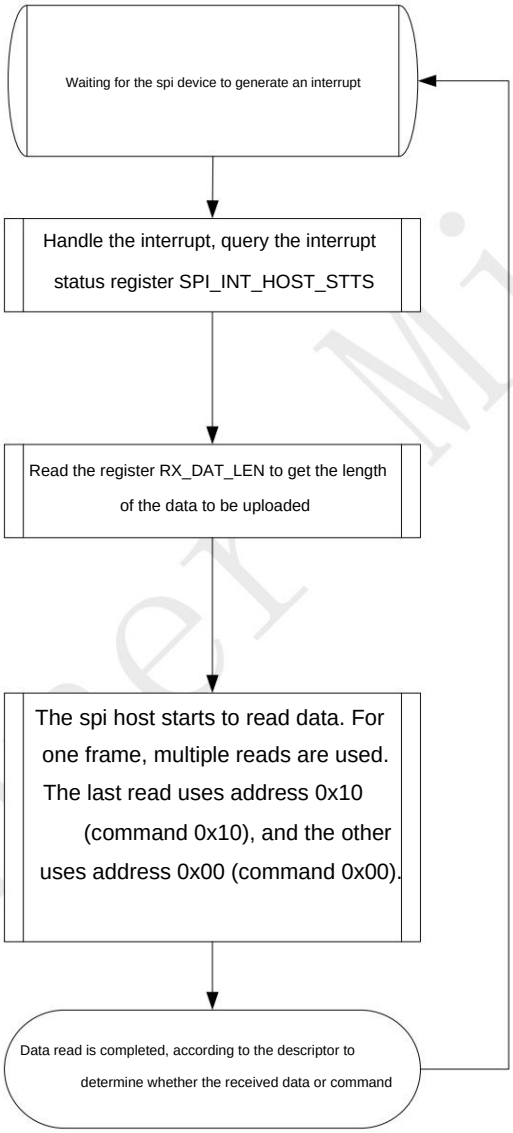


Figure 21 Upstream data (command) flow chart

Upstream data and upstream commands have the same process.

It should be noted here that the length of the upstream data must be in words. If the effective length is not an integer word, the excess data at the tail can be thrown away.

Lose.

It should be noted that there are two channels of data and commands between the master and slave to exchange data, and the user can choose any channel to use

use or both. The maximum data length of one exchange of command channel is 256 bytes, and the maximum length of one exchange of data channel is 256 bytes.

1500 bytes. The data length limit is controlled by the slave device. If the length exceeds the limit, the data structure of the slave device will be destroyed.

11 SDIO Device Controller

11.1 Function overview

W800 integrates the SDIO device interface, as a slave device, to complete the data interaction with the host. Internally integrated 1024byte asynchronous FIFO, completes the data interaction between the host and the chip.

11.2 Main Features

ÿ Compatible with SDIO Card Specification 2.0

ÿ Support host rate 0~50MHz

ÿ Support blocks up to 1024 bytes

ÿ Supports 1-bit SD and 4-bit SD modes

11.3 Functional Description

11.3.1 SDIO bus

The SDIO bus is similar to the USB bus. The SDIO bus also has two ends, one of which is the host side and the other side is the device side . The design of DEVICE is to simplify the design of DEVICE, and all communications are started by commands issued by the HOST side. exist As long as the DEVICE side can parse the HOST command, it can communicate with the HOST, and the SDIO HOST can connect multiple DEVICEÿ

In the SDIO bus definition, the DAT1 signal line is multiplexed as an interrupt line. In the 1BIT mode of SDIO, DAT0 is used to transmit data, DAT1 used as an interrupt line. In the 4BIT mode of SDIO, DAT0 -DAT3 are used to transmit data, and DAT1 is multiplexed as an interrupt line.

11.3.2 SDIO Commands

On the SDIO bus, the HOST side initiates the request, and then the DEVICE side responds to the request. The request and response will contain data information:

ÿ Command: The command used to start the transmission is sent from the HOST side to the DEVICE side, where the command is sent through the CMD message.

transmitted by the number line;

ÿ Response: The response is returned by DEVICE as the response of Command. It is also transmitted through the CMD line;

ÿ Data: Data is transmitted in both directions. Can be set to 1-wire mode or 4-wire mode. Data is passed through DAT0-

DAT3 signal line transmission.

Each operation of SDIO is initiated by HOST on the CMD line to initiate a CMD. For some CMDs, DEVICE needs to return Response,

Some don't.

For the read command, firstly HOST will send a command to DEVICE, and then DEVICE will return a handshake signal. At this time, when HOST

After receiving the response handshake signal, the data will be placed on the 4-bit data line, and the CRC check code will be followed when the data is transmitted. when the whole

After the read transfer is completed, HOST will send a command again to notify DEVICE that the operation is completed, and DEVICE will return a response at the same time.

For the write command, first HOST will send a command to DEVICE, and then DEVICE will return a handshake signal. At this time, when HOST

After receiving the response handshake signal, the data will be placed on the 4-bit data line, and the CRC check code will be followed when the data is transmitted. when the whole

After the write transfer is completed, HOST will send a command again to notify DEVICE that the operation is completed, and DEVICE will return a response at the same time.

11.3.3 SDIO Internal Storage

The SDIO device has a fixed storage map, including the general information area (CIA) and the special function area (function unique area).

The registers in CIA include I/O port function, interrupt generation and port work information, which can be defined by CIA through read and write function 0.

register for related operations. CIA includes CCCR, FBR and CIS three aspects of information. Among them, CCCR defines the public

Common control register, the host side can check the SDIO card and operate the port by operating CCCR. The address of CCCR is 0X00-

0xFF. FBR defines the supported operations of port function 1 to port function 7, including the requirements and functions of each port, power control, etc.

The address of FBR is 0Xn00-0Xnff (where n is the function port number). CIS defines some information structures of the card, the address is 0X1000-

0X17FFF, CIS has a public CIS and its own CIS for each functional port, where the initial address of the public CIS is in the CIS Pointer of CCCR

In the domain, the CIS of each port function is in the CIS Pointer domain of the FBR of each functional port.

The storage map of CIA is as shown below.

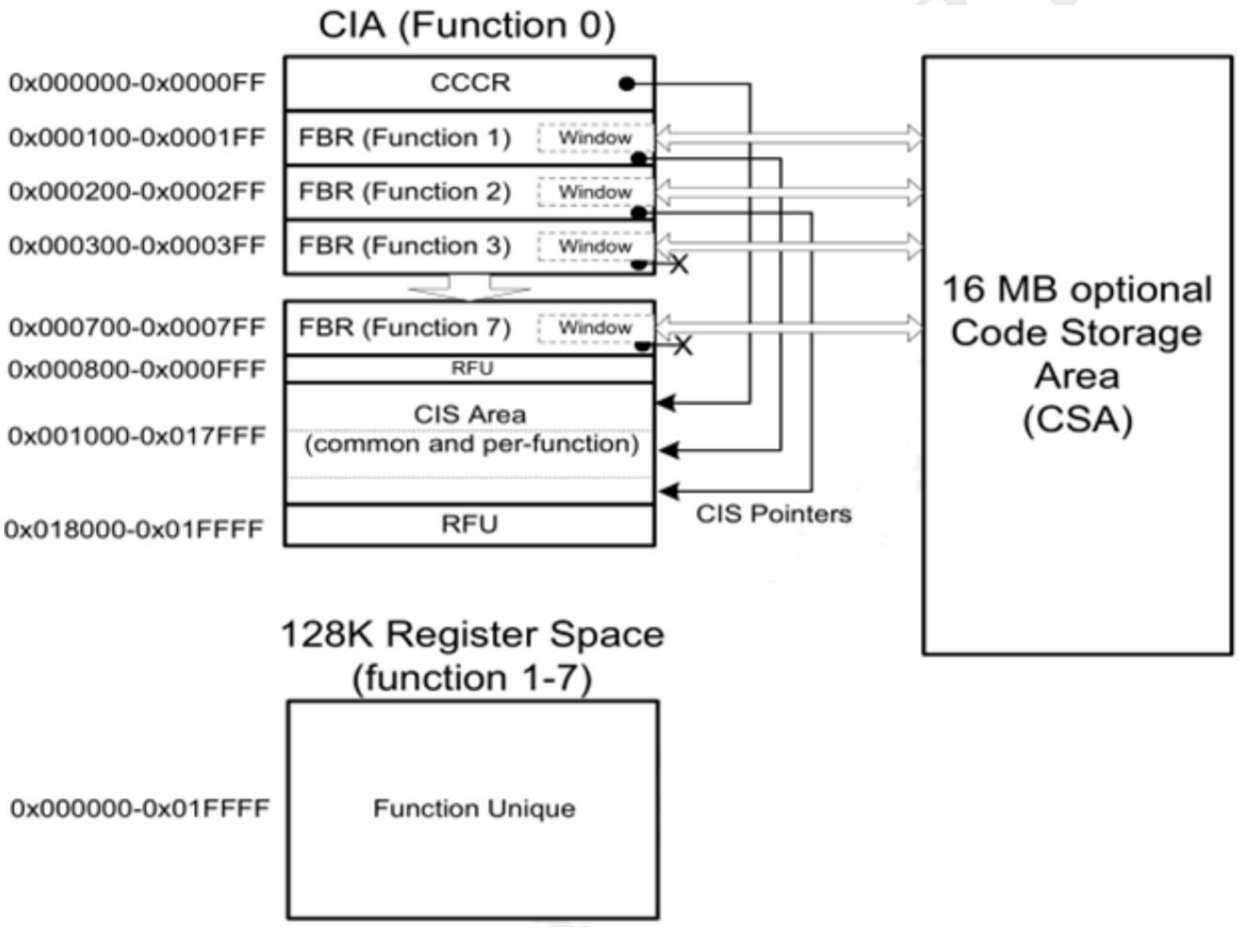


Figure 22 SDIO internal storage map

The description of each register in CIA refers to the following. For an in-depth look at CIA, see the SDIO Protocol Specification.

11.4 Register Description

11.4.1 Register List

11.4.2 SDIO Fn0 register

The Fn0 register is a register specified by the SDIO protocol, and its address range is: 0x00000~0x1FFFF, a total of 128K. The starting address is

0x00000ÿ

The Fn0 register is accessed by the SDIO host through the CMD52 command, the offset address is the access address, and the function number is 0.

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	CCCR/SDIO Revision	SDIO bit 3	SDIO bit 2	SDIO bit 1	SDIO bit 0	CCCR bit 3	CCCR bit 2	CCCR bit 1	CCCR bit 0
0x01	SD Specification Revision	RFU	RFU	RFU	RFU	SD bit 3	SD bit 2	SD bit 1	SD bit 0
0x02	I/O Enable	IOE7	IOE6	IOE5	IOE4	IOE3	IOE2	IOE1	RFU
0x03	I/O Ready	IOR7	IOR6	IOR5	IOR4	IOR3	IOR2	IOR1	RFU
0x04	Int Enable	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IENM
0x05	Int Pending	INT7	INT6	INT5	INT4	INT3	INT2	INT1	RFU
0x06	I/O Abort	RFU	RFU	RFU	RFU	RES	AS2	AS1	AS0
0x07	Bus Interface Control	CD Disable	SCSI	ECSI	RFU	RFU	RFU	Bus Width 1	Bus Width 0
0x08	Card Capability	4BLS	LSC	E4MI	S4MI	SBS	SRW	SMB	SDC
0x09-0x0B	Common CIS Pointer	Pointer to card's *common* Card Information Structure (CIS)							
0x0C	Bus Suspend	RFU	RFU	RFU	RFU	RFU	RFU	BR	BS
0x 0D	Function Select	DF	RFU	RFU	RFU	FS3	FS2	FS1	FS0
0x 0E	Exec Flags	EX7	EX6	EX5	EX4	EX3	EX2	EX1	EXM
0x 0F	Ready Flags	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RFM
0x10-0x11	FN0 Block Size	I/O block size for Function 0							
0x 12	Power Control	Reserved for Future Use (RFU)						EMPC	SMPC
0x 13	High-Speed	RFU	RFU	RFU	RFU	RFU	RFU	EHS	SHS
0x14-0xEF	RFU	Reserved for Future Use (RFU)							
0xF0-0xFF	Reserved for Vendors	Area Reserved for Vendor Unique Registers							

Figure 23 CCCR register storage structure

Address	7	6	5	4	3	2	1	0
0x100	Function 1 CSA enable	Function 1 supports CSA	RFU	RFU	Function 1 Standard SDIO Function interface code			
0x101	Function 1 Extended standard SDIO Function interface code							
0x102	RFU	RFU	RFU	RFU	RFU	RFU	EPS	SPS
0x103-0x108	Reserved for Future Use (RFU)							
0x109-0x10B	Pointer to Function 1 Card Information Structure (CIS)							
0x10C-0x10E	Pointer to Function 1 Code Storage Area (CSA)							
0x10F	Data access window to Function 1 Code Storage Area (CSA)							
0x110-0x111	I/O block size for Function 1							
0x112-0x1FF	Reserved for Future Use							
0x200-0x7FF	Function 2 to 7 Function Basic Information Registers (FBR)							
0x800-0xFFF	Reserved for Future Use							

Figure 24 FBR1 register structure

Address	7	6	5	4	3	2	1	0
0x0001000 - 0x017FFF	Card Common Card Information Structure (CIS) area for card common and all functions							
0x018000- 0x01FFFF	Reserved for Future Use							

Figure 25 CIS storage space structure

11.4.2.1 SDIO CCCR register and FBR1 register list

Table 90 SDIO CCCR register and FBR1 register list

offset address	name abbreviated access			describe	reset value
0X00	CCCR/SDI The Review	SDIOx RO		[3:0], indicates the supported CCCR/FBR format 4'h0ÿ CCCR/FBR Version 1.00 4'h1ÿCCCR/FBR Version 1.10 4'h2ÿCCCR/FBR Version 1.20 4'h3-4'hFÿRsv Represented by CIA Register[3:0]	4'h2

		CCCRx RO		[7:4], indicates the supported SDIO protocol version 4'h0ÿ SDIO Version 1.00 4'h1ÿ SDIO Version 1.10 4'h2ÿ SDIO Version 1.20 (unreleased) 4'h3ÿ SDIO Version 2.00 4'h4-4'hFÿ Rsv Represented by CIA Register[7:4]	4'h3
0X01	SD pecificatio n Revision	SDx	RO [3:0],	indicates the supported SD protocol version 4'h0ÿSD Physical Version 1.01 (March 2000) 4'h1ÿSD Physical Version 1.10 (October 2004) 4'h2ÿSD Physical Version 2.00 (May 2006) 4'h3-4'hFÿ Rsv Represented by CIA Register[11:8]	4'h2
			RO	RFU	4'h0
0X02	I/O Enable IOEx		RO	RFU	1'b0
			RW [7:1],	Function enable, bit1-bit7 correspond to 7 functions respectively, corresponding to If the SD host sets the corresponding bit to 1, the corresponding function is enabled, otherwise the corresponding function does not work. Note: CIS0, CIS1 and CSA are placed in Fn1, even if Fn1 is not enabled at this time, SD host can also read and write to these three areas (CIS0, CIS1 cannot Write).	7'b0
0X03	I/O Ready IORx		RO	RFU	1'b0

			RO	[7:1], IOR has a total of 7 bits, corresponding to the status of the 7 functions, if the corresponding bit If it is 1, it means that the function can work. In this design, HC8051 configures the function of program register The ready bit is 1, so that the bit1 of this register is set to 1, so that the flag Fn1 can be normal. Work. Note: Read and write operations to CIS0, CIS1, CSA are independent of IOR1, that is, even if If IOR1=0, the contents of these three storage spaces can also be accessed.	7'b0
0X04 Int Enable	IENM	RW	[0], interrupt enable signal 0: Interrupts from the card cannot be sent to the SD host 1: Any function interrupt can be sent to the host		1'b0
		IENx	RW	[7:1], interrupt enable of functionx. If IEN1=0, the interrupt from Fn1 will not be sent to the host. IEN1=1, then the interrupt of Fn1 is allowed to be sent to the host	7'b0
0X05 Int	Pending		RO	[0], RFU	1'b0
		INTx	RO	[7:1], Interrupt for functionx is pending. INT1=0, interrupts without Fn1 are pending INT1=1, Fn1 has an interrupt pending. Note: If IEN1 and IENM are not 1, the host will not receive pending interrupts	7'b0
0X06 I/O Abort	ASx		WO	[2:0], cancel IO read or write, thereby releasing the bus. To cancel the Fn1 operation, CMD52 should be used to write 3'b1. This command is under SPI not support.	3'b0
		RES	WO	[3], soft reset signal	1'b0

				1: Reset the circuit of the SD clock domain, this bit is automatically cleared after being set, no special Door clears. This reset signal will not affect the current card protocol selection (SD or SPI mode) without affecting CD Disable. Only use CMD52 operation.	
			RO	RFU	4'b0
0X07 Bus	Interface ontrol	Bus Width	RW [1:0], data line width 2'b00: 1bit data line mode 2'b10: 4bit data line mode Reset or power on, it will become 2'b00		2'b00
			RO	[4: 2], RFU	3'b000
		ECSI	RW [5], enable continuous SPI interrupt. If SCSI is 1, this register is used to enable SDIO card in SPI mode In this case, an interrupt is given at any time, and the state of the CS line does not need to be concerned at this time.		1'b0
		SCSI	RO [6], supports continuous SPI interrupts. If it is 1, it means that the SDIO card supports SPI mode, giving break without caring about the state of the CS. This register is set when progtam reg[2] is 1.		1'b1
		CD Disabl And	RW [7], connect or disconnect the 10-90K pull-up resistor on CD/DAT[3] (pin1). 0: connect pull-up resistor 1: Disconnect the pull-up resistor After power-up, this register is cleared, that is, a pull-up resistor is connected. The state of this register is not Will be affected by the reset command in the SD protocol.		1'b0
0X08 Card		SDC	RO[0], supports the execution of CMD52 commands during data transfer. 1'b1 is not supported in SPI mode		

	Capability			this register. When progtam reg[3] is 1, this register is 1	
		SMB	RO	[1], means that the SDIO card supports the Block transfer mode required by CMD53. When program_reg[4] is 1, this register is 1	1'b1
		SRW	RO	[2], indicates that SDIO supports read wait - Read Wait Control (RWC) operation. When program_reg[5] is 1, this register is 1	1'b1
		SBS	RO	[3] , means that the SDIO card supports suspend/resume. If 0, (0x0C-0x0F) registers are not supported If 1, except Fn0, all functions will be suspended according to SD host requirements or restore When program_reg[6] is 1, this register is 1	1'b1
		S4MI	RO	[4], indicates that the SDIO card supports the 4bit multi-block data transmission mode to the host An interrupt is generated. 0: Interrupts between block transfers are not supported, in this case, as long as IENx=1, SDIO can still initiate an interrupt to the host in other interrupt cycles 1: Support to generate interrupts between block transfers When program_reg[7] is 1, this register is 1	1'b1
		E4MI	RW	[5], interrupt enable. Allow 4bit multi-block mode, in the middle of two block data transmission to the host Interrupted. 0: not allowed 1: Allow	1'b0

				A power-on reset or reset command will clear this register to 0	
		LSC	RO	[6], 0: Indicates that the SDIO card is a full-speed device 1: Indicates that the SDIO card is a low-speed device When program_reg[8] is 1, this register is 1	1'b0
		4BLS	RO	[7], 0: Indicates that SDIO is a low-speed mode device or does not support 4bit mode 1: Indicates that SDIO is a low-speed mode device and supports 4bit mode When program_reg[9] is 1, this register is 1	1'b1
0X09- 0X0B	Common CIS pointer RO			[23:0], points to the starting address pointer of SDIO card shared CIS (CIS0). CIS0 package Contains information about the entire card. Its access space is: 0x001000-0x017FFF. Pointers are stored in little endian format (LSB).	24'h00 1000
0X0C Bus	Suspend	BS	RO	[0] , bus state. 0: The currently selected function does not use the data bus 1: The currently selected function (using FSx or using the function in the IO command number) is executing the command that will transmit data on the data line This register is used by the host to determine which function is currently using the data total String. If the SDIO card does not support the suspend-resume function, this register is 0. Any access to the CIA cannot be suspended, this register is always 1, even if the BR register is 1.	1'b0