

WM_W600_Register Manual

V1.0

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

Address: Fucheng Road, Haidian District, Beijing 67 Yindu Building 18 Floor

Phone: + 86-10-62161900

company website: www.winnermicro.com



Beijing Lianshengde Microelectronics Co., Ltd.

Document modification record

table of Contents

Document modification record	1
table of Contents	2
Figure catalog	13
Table directory	15
1 introduction	twenty two
1.1 Preparation purpose	twenty two
1.2 References	twenty two
2 feature	twenty three
3 Overview	twenty four
4 Chip structure	25
4.1 Chip structure	25
4.2 Bus structure	26
4.3 Clock structure	28
4.4 Address space	29
4.4.1 SRAM	30
4.4.2 Flash	30
4.4.3 Bit zone	30
4.5 Start configuration	31
5 Clock and reset module	33
5.1 Functional Overview	33
5.2 Main features	33
5.3 Functional description	33
5.3.1 Clock gating	33
5.3.2 Clock adaptive shutdown	33
5.3.3 Function reset	33
5.3.4 Clock divider	34
5.3.5 Debug function control	35
5.4 Register description	36
5.4.1 Register list	36
5.4.2 Software clock gating enable register	36
5.4.3 Software clock mask register	38
5.4.4 Software reset control register	38

5.4.5 Clock divider configuration register	41
5.4.6 Debug control register	42
5.4.7 I2S Clock Control Register.....	42
6 DMA Module	44
6.1 Functional Overview	44
6.2 Main features	44
6.3 Functional description	44
6.3.1 DMA aisle	44
6.3.2 DMA data flow	44
6.3.3 DMA Cycle mode	45
6.3.4 DMA Transmission mode	45
6.3.5 DMA Peripheral selection	45
6.3.6 DMA Linked list mode	46
6.3.7 DMA Interrupt	46
6.4 Register description	46
6.4.1 Register list	46
6.4.2 Interrupt mask register	47
6.4.3 Interrupt status register	48
6.4.4 UART Select register	48
6.4.5 DMA Source address register	48
6.4.6 DMA Destination address register	49
6.4.7 DMA Loop source start address register	49
6.4.8 DMA Loop destination start address register	49
6.4.9 DMA Loop length register	49
6.4.10 DMA Channel control register	49
6.4.11 DMA Mode selection register	50
6.4.12 DMA Data Flow Control Register	50
6.4.13 DMA Transfer Bytes Register	51
6.4.14 DMA Link list entry address register	51
6.4.15 DMA Current destination address register	52
7 General hardware encryption module	53
7.1 Functional Overview	53
7.2 Main features	53

7.3 Functional description	53
7.3.1 SHA1 encryption	53
7.3.2 MD5 encryption.....	53
7.3.3 RC4 encryption	53
7.3.4 DES encryption	53
7.3.5 3DES encryption	53
7.3.6 AES encryption	54
7.3.7 CRC encryption	54
7.3.8 PRNG Pseudo-random number	54
7.4 Register description	54
7.4.1 Register list	54
7.4.2 Configuration register	55
7.4.3 Control register	57
7.4.4 Status register	57
8 RSA Encryption module.....	58
8.1 Functional Overview	58
8.2 Main features	58
8.3 Functional description	58
8.3.1 Modular multiplication function	58
8.4 Register description	58
8.4.1 Register list	58
8.4.2 data X register	58
8.4.3 data Y register	58
8.4.4 data M register.....	59
8.4.5 data D register	59
8.4.6 RSA Control Register.....	59
8.4.7 parameter MC register	59
8.4.8 parameter N register	60
9 GPIO Module	61
9.1 Functional Overview	61
9.2 Main features	61
9.3 Functional description	61
9.4 Register description	62

9.4.1 Register list	62
9.4.2 GPIO Data register	63
9.4.3 GPIO Data enable register	63
9.4.4 GPIO Direction control register	64
9.4.5 GPIO Up and down control register	64
9.4.6 GPIO Multiplex select register	65
9.4.7 GPIO Multiplex select register 1	66
9.4.8 GPIO Multiplex select register 0	66
9.4.9 GPIO Interrupt trigger mode configuration register	66
9.4.10 GPIO Interrupt edge trigger mode configuration register	67
9.4.11 GPIO Interrupt upper and lower edge trigger configuration register	67
9.4.12 GPIO Interrupt enable configuration register	68
9.4.13 GPIO Bare Interrupt Status Register	68
9.4.14 GPIO Interrupt status register after masking	69
9.4.15 GPIO Interrupt Clear Control Register	69
 10 high speed SPI Device controller	70
10.1 Functional Overview	70
10.2 Main features	70
10.3 Functional description	70
10.3.1 SPI Brief Introduction	70
10.3.2 SPI work process	71
10.4 Register description	71
10.4.1 HSPI Register list of internal operation of the chip.....	71
10.4.2 Host side access HSPI Controller register list	73
10.4.3 high speed SPI Device controller interface timing	76
 11 SDIO Device controller	85
11.1 Functional Overview	85
11.2 Main features	85
11.3 Functional description	85
11.3.1 SDIO bus	85
11.3.2 SDIO command	85
11.3.3 SDIO Internal storage	86
11.4 Register description	87

11.4.1 Register list	87
11.4.2 SDIO Fn0 register.....	87
11.4.3 SDIO Fn1 register.....	96
12 HSPI/SDIO Wrapper Controller.....	103
12.1 Functional Overview	103
12.2 Main features	103
12.3 Functional description	103
12.3.1 Upstream data receiving function	103
12.3.2 Downstream data movement function	104
12.4 Register description	105
12.4.1 Register list	105
12.4.2 WRAPPER Interrupt status register	106
12.4.3 WRAPPER Interrupt Configuration Register	106
12.4.4 WRAPPER Uplink command ready register	106
12.4.5 WRAPPER Down Command buf Ready register	106
12.4.6 SDIO TX Link enable register.....	107
12.4.7 SDIO TX Link address register.....	107
12.4.8 SDIO TX Enable Register.....	107
12.4.9 SDIO TX Status Register.....	108
12.4.10 SDIO RX Link enable register	108
12.4.11 SDIO RX Link address register	108
12.4.12 SDIO RX Enable register	108
12.4.13 SDIO RX Status register	109
12.4.14 WRAPPER CMD BUF Base address register	109
12.4.15 WRAPPER CMD BUF SIZE register	109
13 SPI Controller.....	110
13.1 Functional Overview	110
13.2 Main features	110
13.3 Functional description	110
13.3.1 Master-slave	110
13.3.2 Multiple modes support	110
13.3.3 Efficient data transmission	111
13.4 Register description	111

13.4.1 Register list	111
13.4.2 Channel configuration register	111
13.4.3 SPI Configuration register	113
13.4.4 Clock configuration register	116
13.4.5 Mode configuration register	116
13.4.6 Interrupt control register	117
13.4.7 Interrupt status register	118
13.4.8 SPI Status register	119
13.4.9 SPI Timeout register	120
13.4.10 Data transmission register.....	120
13.4.11 Transfer Mode Register.....	121
13.4.12 Data Length Register.....	122
13.4.13 Data Receive Register.....	123
14 I2C Controller	124
14.1 Functional Overview	124
14.2 Main features	124
14.3 Functional description	124
14.3.1 Transmission rate selection	124
14.3.2 Controllable interruption, start and stop	124
14.3.3 Fast output and detection signal	124
14.4 Register description	125
14.4.1 Register list	125
14.4.2 Clock Divider Register_ 1	125
14.4.3 Clock Divider Register_ 2	125
14.4.4 Control register	126
14.4.5 Data register	126
14.4.6 Transceiver control register	126
14.4.7 TXR Read register	127
14.4.8 CR Read register	128
15 I2S Controller	129
15.1 Functional Overview	129
15.2 Main features	129
15.3 Functional description	129

15.3.1 Multiple modes support	129
15.3.2 Zero crossing detection	129
15.3.3 Efficient data transmission	129
15.4 Register description	130
15.4.1 Register list	130
15.4.2 Control register	130
15.4.3 Interrupt mask register	133
15.4.4 Interrupt flag register	135
15.4.5 Status register	137
15.4.6 Data send register	138
15.4.7 Data receive register	138
16 UART Module	139
16.1 Functional Overview	139
16.2 Main features	139
16.3 Functional description	139
16.3.1 UART Baud rate	139
16.3.2 UART Data Format	140
16.3.3 UART Hardware flow control	141
16.3.4 UART DMA transmission	141
16.3.5 UART Interrupt	142
16.4 Register description	142
16.4.1 Register list	142
16.4.2 Data Flow Control Register	142
16.4.3 Automatic hardware flow control register	143
16.4.4 DMA Set register	143
16.4.5 FIFO Control register	144
16.4.6 Baud rate control register	144
16.4.7 Interrupt mask register	145
16.4.8 Interrupt status register	145
16.4.9 FIFO Status register	146
16.4.10 TX Start address register	146
16.4.11 RX Start Address Register.....	147
17 UART&7816 Module	148

17.1 Functional Overview	148
17.2 Main features	148
17.3 UART Functional description	148
17.4 7816 Functional description	149
17.4.1 7816 Introduction	149
17.4.2 7816 interface	149
17.4.3 7816 Configuration	149
17.4.4 7816 Clock configuration	149
17.4.5 7816 Rate setting	150
17.4.6 7816 Power-on reset	151
17.4.7 7816 Warm reset	151
17.4.8 7816 Deactivation process	152
17.4.9 7816 data transmission	152
17.4.10 UART&7816 DMA transmission.....	153
17.4.11 UART&7816 Interrupt	153
17.5 Register description	153
17.5.1 Register list	153
17.5.2 Data Flow Control Register	154
17.5.3 Automatic hardware flow control register	155
17.5.4 DMA Set register	156
17.5.5 FIFO Control register	156
17.5.6 Baud rate control register	157
17.5.7 Interrupt mask register	157
17.5.8 Interrupt status register	157
17.5.9 FIFO Status register	159
17.5.10 TX Start address register	159
17.5.11 RX Start Address Register.....	159
17.5.12 7816 Guard Time Register.....	159
17.5.13 7816 Time-out register.....	160
18 Time r Module	161
18.1 Functional Overview	161
18.2 Main features	161
18.3 Functional description	161

18.3.1 Timing function	161
18.3.2 Delay function	161
18.4 Register description	162
18.4.1 Register list	162
18.4.2 standard us Configuration Register.....	162
18.4.3 Timer control register	162
18.4.4 Timer 1 Timing configuration register	163
18.4.5 Timer 2 Timing configuration register	163
18.4.6 Timer 3 Timing configuration register	163
18.4.7 Timer 4 Timing configuration register	163
18.4.8 Timer 5 Timing configuration register	163
18.4.9 Timer 6 Timing configuration register	164
19 Power management module	165
19.1 Functional Overview	165
19.2 Main features	165
19.3 Functional description	165
19.3.1 Full chip power control	165
19.3.2 Wake-up mode	165
19.3.3 Timer0 Timer	166
19.3.4 Real-time clock function	166
19.3.5 32K Clock source switching and calibration	166
19.4 Register description	166
19.4.1 Register list	166
19.4.2 PMU Control register	167
19.4.3 PMU Timer 0	168
19.4.4 PMU Interrupt source register	168
20 Real-time clock module	169
20.1 Functional Overview	169
20.2 Main features	169
20.3 Functional description	169
20.3.1 Timing function	169
20.3.2 Timing function	169
20.4 Register description	170

20.4.1 Register list	170
20.4.2 RTC Configuration register 1.....	170
20.4.3 RTC Configuration register 2.....	170
twenty one Watchdog module	171
21.1 Functional Overview	171
21.2 Main features	171
21.3 Functional description	171
21.3.1 Timing function	171
21.3.2 Reset function	171
21.4 Register description	171
21.4.1 Register list	171
21.4.2 WDG Time value load register	172
21.4.3 WDG Current Value Register.....	172
21.4.4 WDG Control Register.....	172
21.4.5 WDG Interrupt Clear Register.....	172
21.4.6 WDG Interrupt source register.....	173
21.4.7 WDG Interrupt Status Register.....	173
twenty two PWM Controller	174
22.1 Functional Overview	174
22.2 Main features	174
22.3 Functional description	174
22.3.1 Input signal capture	174
22.3.2 DMA Transmission capture number	174
22.3.3 Support single and automatic loading mode	175
22.3.4 Multiple output modes	175
22.4 Register description	175
22.4.1 PWM Register list.....	175
22.4.2 Clock Divider Register_ 01	176
22.4.3 Clock Divider Register_ twenty three	176
22.4.4 Control register	176
22.4.5 Period register	178
22.4.6 Cycle number register	179
22.4.7 Compare register	180

22.4.8 Dead zone control register	181
22.4.9 Interrupt control register	181
22.4.10 Interrupt Status Register.....	182
22.4.11 aisle 0 Capture Register.....	183
22.4.12 Brake control register.....	183
22.4.13 Clock Divider Register_ 4.....	184
22.4.14 aisle 4 Control register_ 1.....	185
22.4.15 aisle 4 Capture Register.....	186
22.4.16 aisle 4 Control register_ 2.....	186
twenty three QFLASH Controller	189
23.1 Functional Overview	189
23.2 Main features	189
23.3 Functional description	189
23.4 Register description	189
23.4.1 Register list	189
23.4.2 Command Information Register	189
23.4.3 Command start register	190
23.5 QFLASH Common instructions	190
twenty four appendix 1. Chip pin definition	191
24.1 Chip pinout	191
24.2 Chip pin multiplexing relationship	192
statement	194

Figure catalog

Fig 1 W600 Chip structure diagram	25
Fig 2 W600 Bus structure diagram	26
Fig 3 W600 Clock structure	28
Fig 4 Memory bit zone mapping relationship	31
Fig 5 System clock frequency division relationship	34
Fig 6 Host computer SPI Send and receive data format	77
Fig 7 HSPI Register read operation (big endian mode)	77
Fig 8 HSPI Register write operation (big endian mode)	77
Fig 9 Register read operation (little-endian mode)	78
Fig 10 Register write operation (little-endian mode)	78
Fig 11 Port read operation (big endian mode).....	78
Fig 12 Port write operation (big endian mode)	78
Fig 13 Port read operation (little-endian mode)	78
Fig 14 Port write operation (little-endian mode)	79
Fig 15 CPOL=0, CPHA=0.....	79
Fig 16 CPOL=0, CPHA=1.....	80
Fig 17 CPOL=1, CPHA=0.....	80
Fig 18 CPOL=1, CPHA=1.....	81
Fig 19 the Lord SPI Handle interruption process	82
Fig 20 Downstream data flow chart	82
Fig twenty one Downstream command flow chart	83
Fig twenty two Upstream data (command) flow chart	84
Fig 23 SDIO Internal storage mapping	87
Fig 24 CCCR Register storage structure	88
Fig 25 FBR1 Register structure	88
Fig 26 CIS Storage space structure	88
Fig 27 SDIO receive BD Descriptor	104
Fig 28 SDIO send BD Descriptor	105
Fig 29 UART Data length	140
Fig 30 UART Stop bit	140
Fig 31 UART Parity bit	141
Fig 32 UART Hardware flow control connection	141

Fig 33 7816 Connection diagram	149
Fig 34 7816 Power-on reset timing	151
Fig 35 7816 Warm reset	151
Fig 36 7816 Deactivation process	152
Fig 37 7816 data transmission	152
Fig 38 W600 Chip pinout	191

Table directory

table 1 AHB-1 List of bus masters	26
table 2 AHB-1 List of bus slaves	27
table 3 AHB-2 List of bus masters	27
table 4 AHB-2 List of bus slaves	27
table 5 Address space distribution	29
table 6 Start configuration	32
table 7 System clock frequency division parameter configuration	35
table 8 Clock reset module register list	36
table 9 Software clock gating enable register	36
table 10 Software clock mask register	38
table 11 Software reset control register	38
table 12 Clock divider configuration register	41
table 13 Debug control register	42
table 14 I2S Clock control register	42
table 15 DMA Address assignment	44
table 16 DMA Register list	46
table 17 DMA Interrupt mask register	47
table 18 DMA Interrupt status register	48
table 19 UART Select register	48
table 20 DMA Source address register	48
table 21 DMA Destination address register	49
table 22 DMA Loop source start address register	49
table 23 DMA Loop destination start address register	49
table 24 DMA Loop length register	49
table 25 DMA Channel control register	49
table 26 DMA Mode selection register	50
table 27 DMA Data Flow Control Register	50
table 28 DMA Transfer Bytes Register	51
table 29 DMA Link list entry address register	51
table 30 DMA Current destination address register	52
table 31 Encryption module register list	54
table 32 Encryption module configuration register	55

table 33 Encryption module control register	57
table 34 Encryption module status register	57
table 35 RSA Register list	58
table 36 RSA data X register	58
table 37 RSA data Y register	59
table 38 RSA data M register	59
table 39 RSA data D register	59
table 40 RSA Control register	59
table 41 RSA parameter MC register	59
table 42 RSA parameter N register	60
table 43 GPIOA Register list	62
table 44 GPIOB Register list	62
table 45 GPIOA Data register	63
table 46 GPIOB Data register	63
table 47 GPIOA Data enable register	63
table 48 GPIOB Data enable register	63
table 49 GPIOA Direction control register	64
table 50 GPIOB Direction control register	64
table 51 GPIOA Up and down control register	64
table 52 GPIOB Up and down control register	64
table 53 GPIOA Multiplex select register	65
table 54 GPIOB Multiplex select register	65
table 55 GPIOA Multiplex select register 1	66
table 56 GPIOB Multiplex select register 1	66
table 57 GPIOA Multiplex select register 0	66
table 58 GPIOB Multiplex select register 0	66
table 59 GPIOA Interrupt trigger mode configuration register	66
table 60 GPIOB Interrupt trigger mode configuration register	67
table 61 GPIOA Interrupt edge trigger mode configuration register	67
table 62 GPIOB Interrupt edge trigger mode configuration register	67
table 63 GPIOA Interrupt upper and lower edge trigger configuration register	67
table 64 GPIOB Interrupt upper and lower edge trigger configuration register	67
table 65 GPIOA Interrupt enable configuration register	68

table 66 GPIOB Interrupt enable configuration register	68
table 67 GPIOA Bare Interrupt Status Register	68
table 68 GPIOB Bare Interrupt Status Register	68
table 69 GPIOA Interrupt status register after masking	69
table 70 GPIOB Interrupt status register after masking	69
table 71 GPIOA Interrupt Clear Control Register	69
table 72 GPIOB Interrupt Clear Control Register	69
table 73 HSPI Internal access register	71
table 74 HSPI FIFO Clear register	71
table 75 HSPI Configuration register	72
table 76 HSPI Mode configuration register	72
table 77 HSPI Interrupt Configuration Register	72
table 78 HSPI Interrupt status register	73
table 79 HSPI Data upload length register	73
table 80 HSPI Interface configuration register (master access)	73
table 81 HSPI Get data length register	74
table 82 HSPI Send data flag register	74
table 83 HSPI Interrupt Configuration Register	75
table 84 HSPI Interrupt status register	75
table 85 HSPI Data port 0	75
table 86 HSPI Data port 1	75
table 87 HSPI Command port 0	76
table 88 HSPI Command port 1	76
table 89 SDIO CCCR Register and FBR1 Register list	89
table 90 SDIO Fn1 Address mapping	96
table 91 SDIO Fn1 Part of the register (for HOST access)	97
table 92 SDIO AHB Bus register	98
table 93 WRAPPER Controller register	105
table 94 WRAPPER Interrupt status register	106
table 95 WRAPPER Interrupt Configuration Register	106
table 96 WRAPPER Uplink command ready register	106
table 97 WRAPPER Down Command buf Ready register	106
table 98 SDIO TX Link enable register	107

table 99 SDIO TX Link address register	107
table 100 SDIO TX Enable register	107
table 101 SDIO TX Status register	108
table 102 SDIO RX Link enable register	108
table 103 SDIO RX Link address register	108
table 104 SDIO RX Enable register	108
table 105 SDIO RX Status register	109
table 106 WRAPPER CMD BUF Base address register	109
table 107 WRAPPER CMD BUF SIZE register	109
table 108 SPI Register list	111
table 109 SPI Channel configuration register	111
table 110 SPI Configuration register	113
table 111 SPI Clock Configuration Register.....	116
table 112 SPI Mode configuration register	116
table 113 SPI Interrupt control register	117
table 114 SPI Interrupt status register	118
table 115 SPI Status register	119
table 116 SPI Timeout register	120
table 117 SPI Data send register	120
table 118 SPI Transfer mode register	121
table 119 SPI Data length register	122
table 120 SPI Data receive register	123
table 121 I2C Register list	125
table 122 I2C Clock Divider Register_ 1	125
table 123 I2C Clock Divider Register_ 2	125
table 124 I2C Control register	126
table 125 I2C Data register	126
table 126 I2C Transceiver control register	126
table 127 I2C TXR Read register	127
table 128 I2C CR Read register	128
table 129 I2S Register list	130
table 130 I2S Control register	130
table 131 I2S Interrupt mask register	133

table 132 I2S Interrupt flag register	135
table 133 I2S Status register	137
table 134 I2S Data send register	138
table 135 I2S Data receive register	138
table 136 UART Register list	142
table 137 UART Data Flow Control Register	142
table 138 UART Automatic hardware flow control register	143
table 139 UART DMA Set register	143
table 140 UART FIFO Control register	144
table 141 UART Baud rate control register	144
table 142 UART Interrupt mask register	145
table 143 UART Interrupt status register	145
table 144 UART FIFO Status register	146
table 145 UART TX Start address register	146
table 146 UART RX Start address register	147
table 147 7816 Rate setting	150
table 148 UART&7816 Register list	153
table 149 UART&7816 Data Flow Control Register	154
table 150 UART&7816 Automatic hardware flow control register	155
table 151 UART&7816 DMA Set register	156
table 152 UART&7816 FIFO Control register	156
table 153 UART&7816 Baud rate control register	157
table 154 UART&7816 Interrupt mask register	157
table 155 UART&7816 Interrupt status register	157
table 156 UART&7816 FIFO Status register	159
table 157 UART&7816 TX Start address register	159
table 158 UART&7816 RX Start address register	159
table 159 7816 Protection time register	159
table 160 7816 Time-out register	160
table 161 Timer Register list	162
table 162 Timer standard us Configuration register	162
table 163 Timer Timer control register	162
table 164 Timer 1 Timing configuration register	163

table 165 Timer 2 Timing configuration register	163
table 166 Timer 3 Timing configuration register	163
table 167 Timer 4 Timing configuration register	163
table 168 Timer 5 Timing configuration register	163
table 169 Timer 6 Timing configuration register	164
table 170 PMU Register list	166
table 171 PMU Control register	167
table 172 PMU Timer 0 register	168
table 173 PMU Interrupt source register	168
table 174 RTC Register list	170
table 175 RTC Configuration register 1	170
table 176 RTC Configuration register 2	170
table 177 WDG Register list	171
table 178 WDG Timing value loading register	172
table 179 WDG Current Value Register	172
table 180 WDG Control Register	172
table 181 WDG Interrupt Clear Register	172
table 182 WDG Interrupt source register	173
table 183 WDG Interrupt Status Register	173
table 184 PWM Register list	175
table 185 PWM Clock Divider Register_01	176
table 186 PWM Clock Divider Register_twenty three	176
table 187 PWM Control register	176
table 188 PWM Period register	178
table 189 PWM Cycle number register	179
table 190 PWM Compare register	180
table 191 PWM Dead zone control register	181
table 192 PWM Interrupt control register	181
table 193 PWM Interrupt status register	182
table 194 PWM aisle 0 Capture register	183
table 195 PWM Brake control register	183
table 196 PWM Clock Divider Register_4	184
table 197 PWM aisle 4 Control register_1	185

table 198 PWM Tao 4 Capture register	186
table 199 PWM aisle 4 Control register_2.....	186
table 200 QFLASH Controller Register List.....	189
table 201 QFLASH Command information register	189
table 202 QFLASH Command Start Register.....	190
table 203 QFALSH Common commands	190
table 204 Chip pin multiplexing relationship	192

1 introduction

1.1 Preparation purpose

W600 Chip is an embedded product launched by Lianshengde Microelectronics Wi-Fi SoC chip. The chip has a high degree of integration and requires few peripheral devices.

Price ratio is high. Apply to IoT (Smart home) various smart products in the field. Highly integrated Wi-Fi Function is its main function; in addition, the

Chip integration Cortex-M3 Kernel, built-in QFlash , SDIO , SPI , UART , GPIO , I²C , PWM , I²S , 7816 Equivalent interface

Support a variety of hardware encryption and decryption algorithms.

This document mainly describes W600 The internal structure of the chip, the information of each functional module and the detailed register usage information; it is the driver for developers to develop applications.

The main reference materials of the program and application. Provided by Lianshengde Microelectronics SDK There are open source implementations of various functions in it, developers can refer to them.

Corresponding examples of drivers and applications to increase understanding of chip functions and register descriptions. This document is not correct Wi-Fi Partially sent

Memory description.

1.2 References

W600 For chip packaging parameters, electrical characteristics, RF parameters, etc., please refer to "W600 Chip Product Specification";

W600 Chip integrated ROM program, ROM The program provides download firmware, MAC Address reading and writing, Wi-Fi Parameters reading and writing functions,

For more information, please refer to "WM_W600_ROM Function Brief";

W600 The chip is built in 1Mbytes QFlash Memory, as a storage space for codes and parameters. This document provides QFlash Basic

Operation information. If you need more than the scope of this document, you need to refer to QFlash manual;

W600 Chip adoption ARM Cortex-M3 core, M3 Related function introduction, development materials, etc. can refer to ARM Company release information;

For more information, please refer to Lianshengde Microelectronics website (<http://www.winnermicro.com/>).

2 feature

- Chip package
 - QFN32 Package, 5mm x 5mm .
 - Support most 17 Pc GPIO Mouth, each IO The port has a rich multiplexing relationship. With input and output configuration options, some GPIO Drive capacity reached 24mA .
- Chip integration
 - integrated Cortex-M3 Processor, highest 80MHz
 - integrated 288KB SRAM
 - integrated 1MB FLASH
 - integrated 8 aisle DMA Controller, support 16 Hardware applications, support software linked list management
 - integrated PA/LNA/TR-Switch
 - integrated 32.768KHz Clock oscillator
 - Integrated voltage detection circuit
 - integrated LDO
 - Integrated power-on reset circuit
 - WIFI Protocols and functions
 - stand by GB15629.11-2006 , IEEE802.11 b/g/n ;
 - stand by WMM/WMM-PS/WPA/WPA2/WPS
 - stand by WiFi Direct ;
 - stand by EDCA Channel access method;
 - stand by 20/40M Bandwidth working mode;
 - stand by STBC , GreenField , Short-GI , Support reverse transmission;
 - stand by RIFS Frame interval
 - stand by AMPDU , AMSDU ;
 - stand by 802.11n MCS 0~7 , MCS32 The transmission rate of the physical layer, the transmission rate is up to 150Mbps;
 - stand by HT-immediate Compressed BlockAck , normal ACK , no ACK Response method
 - stand by CTS to self ;
 - stand by AP Features; AP with STA use simultaneously;
 - in BSS In the network, it supports multiple multicast networks and supports different encryption methods for each multicast network. 32 Multicast network and access network
 - STA encryption;
 - BSS Network support as AP When in use, the sum of the support sites and groups is 32 A
 - Receive sensitivity:
 - 20MHz MCS7@-71dBm @10%PER ;
 - 40MHz MCS7@-67dBm @10%PER ;
 - 54Mbps@-73dBm @10%PER ;
 - 11Mbps@-86dBm @8%PER ;
 - 1Mbps@-94dBm @8%PER ;
 - Support a variety of different received frame filtering options;
 - Support monitoring function;
- Chip interface
 - integrated 1 Pc SDIO2.0 Device Controller, support SDIO 1 Bit/ 4 Bit/ SPI Three operating modes, operating clock range 0~50MHz
 - integrated 2 Pc UART Interface, support RTS/CTS , Baud rate range 1200bps~2Mbps
 - integrated 1 High speed SPI Slave interface, working clock range 0~50MHz
 - integrated 1 Pc SPI Master/slave interface, the working clock of master device is up to 20MHz , The highest support from the device 6Mbps Data transfer rate
 - Integrate one I2C Controller, support 100/400Kbps rate
 - integrated PWM Controller, support 5 road PWM Output alone or 2 road PWM Enter. Maximum output frequency 20MHz , Maximum input frequency 20MHz
 - Integrated duplex I2S Controller, support 32KHz To 192KHz I2S Interface codec
 - integrated 1 Pc 7816 Interface, compatible UART Interface support ISO-7816-3 T=0.T=1 Mode; support EVM2000 protocol
 - Support multiple hardware encryption and decryption modes, including RSA/AES/RC4/DES/3DES/RC4/SHA1/M D5/CRC8/CRC16/CRC32/PRNG
 - Power supply and power consumption
 - 3.3V Single power supply;
 - stand by Wi-Fi Energy-saving mode power management;
 - Support work, sleep, standby, shutdown working modes;
 - Standby power consumption is less than 10uA ;

3 Overview

This chip is a wireless local area network that supports multiple interfaces and multiple protocols 802.11n (1T1R) of SOC chip. The SOC Chip integrated RF transceiver front end RF Transceiver , CMOS PA Power amplifier, baseband processor/media access control, SDIO , SPI , UART , GPIO Low power consumption WLAN chip.

W600 Chip support GB15629.11-2006 , IEEE802.11 b/g/ n Agreement and support STBC , Green Field , Short-GI , Reverse transmission, RIFS Frame interval, AMPDU , AMSDU , T-immediate Compressed Block Ack , normal ACK , no ACK , CTS to self Rich protocols and operations.

W600 Integrated RF transceiver front end, A/D with D/A converter. It supports DSSS (Direct sequence spread spectrum) and OFDM

(Orthogonal frequency division multiplexing) modulation mode, with data descrambling capability, supports a variety of different data transmission rates. Transceiver equipped in the analog front end of the transceiver AGC The function enables the chip system to obtain the best performance. W600 The chip also includes a built-in enhanced signal monitor, which can largely eliminate the effects of multipath effects.

W600 In terms of security, the chip not only supports national standards WAPI Encryption, also supports international standards WEP , TKIP , CCMP Encryption, these hardware components enable the data transmission system based on the chip to still obtain data transmission performance similar to that of non-encrypted communication when performing confidential communication.

W600 In addition to chip support IEEE802.11 as well as Wi-Fi In addition to the energy-saving operations specified in the agreement, it also supports customized energy-saving solutions. The chip 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 define different energy-saving schemes according to their own usage scenarios.

W600 Chip integrated high performance 32 Embedded processor, a lot of memory resources, and rich peripheral interfaces make it easy for users to apply the chip to the secondary development of specific products.

W600 Chip support AP Function, which can be realized at the same time 5 Pcs SSID Network to achieve 5 Independent AP Function. Support the establishment of multi-multicast network functions. Can be achieved as STA While joining other networks, you act as AP set up BSS The function of the network.

W600 Chip support WPS In this way, users can use one-click operation to realize a fully encrypted network and ensure the security of information.

W600 Multi-function chip and high integration guarantee WLAN The system does not require excessive off-chip circuits and external memory.

4 Chip structure

4.1 Chip structure

The following figure describes W600 The overall structure of the chip, the core part includes Cortex-M3 CPU , 288KB SRAM with 16KB ROM Empty storage between. PMU Some of the constant power supply modules as chips provide power-on timing management, start-up clock, and real-time clock functions. Provides a wealth of Peripheral functions and hardware encryption and decryption functions. Wi-Fi Partially integrated MAC , BB with RF .

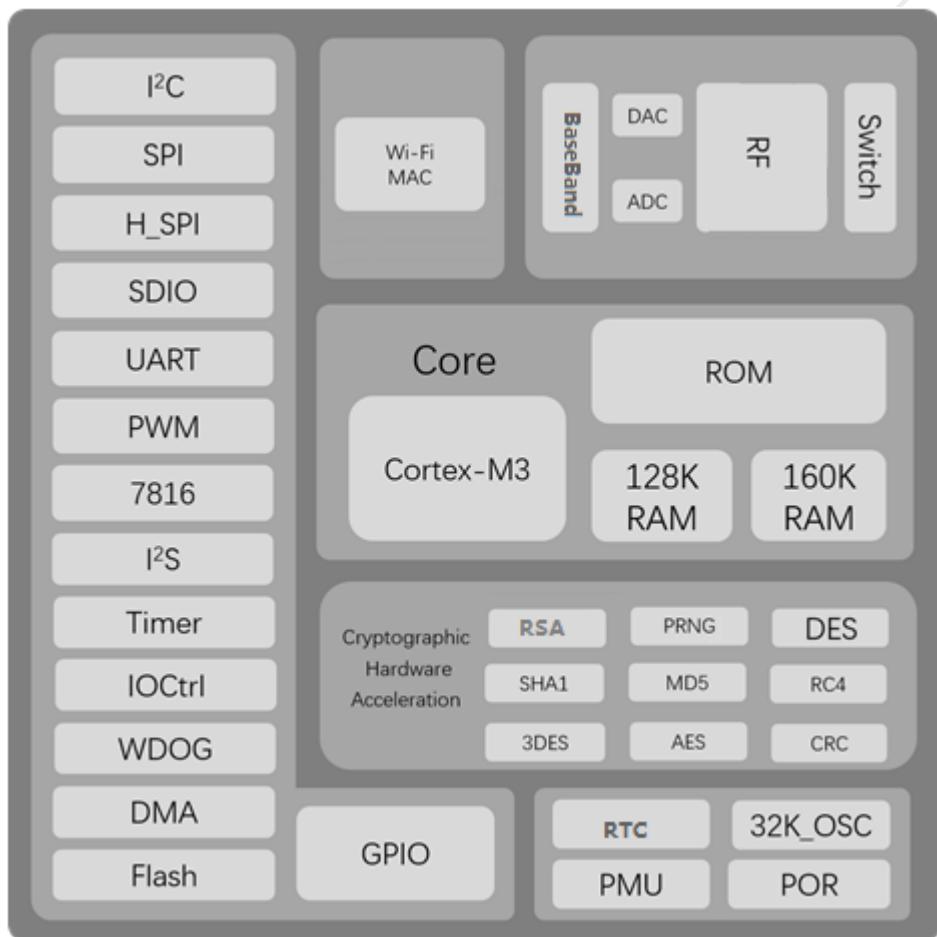


Fig 1 W600 Chip structure diagram

4.2 Bus structure

W600 The chip is composed of a two-level bus, as shown in the following figure

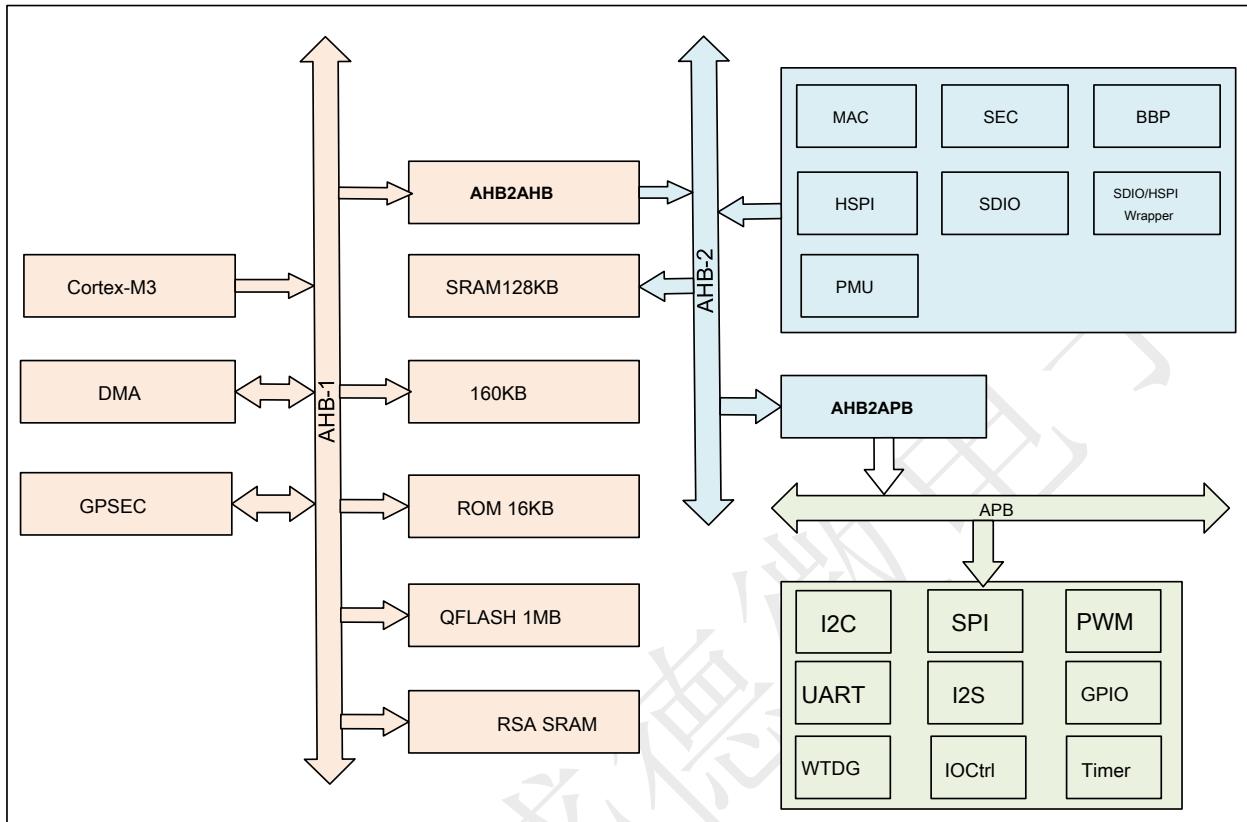


Fig 2 W600 Bus structure diagram

(1) AHB-1 bus

This level bus has four master devices-namely Cortex-M3 , DMA , GPSEC as well as 5 Slave devices.

Cortex-M3 based on ARMV7-M Architecture design, use Thumb-2 The instruction set adopts Harvard structure and has independent instruction bus and data bus. The bus clock works fastest 80MHz Frequency can be configured as 40MHz , Or lower, please refer to the clock division section for the clock configuration

table 1 AHB-1 List of bus masters

Main device	Features
CPU	Complete chip register configuration, memory management and use, and complete 802.11MAC protocol.
DMA	Support independent of the linked list structure 8 aisle DMA Module, support on chip 16 Hardware DMA Request source.
GPSEC	Universal encryption module, support DES/3DESSHA1/AES/MD5/RC4/CRC/PRDN . Encrypt and write back data blocks in the specified memory space automatically.

table 2 AHB-1 List of bus slaves

Slave device	Features
ROM	ROM Used to store CPU Initialize firmware after power on. Mainly complete the initial configuration of the chip register space. After completing the above work, CPU Control over FLASH The firmware stored in.
AHB2AHB	carry out CPU Bus clock domain to BusMatrix2 Conversion of access by the master device in the bus clock domain. The clock domain must be homologous, and CPU Clock with BusMatrix2 The ratio of the clock frequency is M : 1 , M Is greater than or equal to 1 Integer.
FLASH	Store firmware codes and operating parameters.
SRAM 160KB	It can be used to store instructions or data, and the firmware can use this memory as needed.
RSA	Support the longest 2048bit of RSA Encryption and decryption

(2) AHB-2 bus

This bus has 4 Master devices, 3 Slave devices, use crossbar Connection structure, can realize different master equipment to different slave equipment

Simultaneous access to increase bandwidth. The bus clock works fastest 40MHz The frequency can be configured to be lower as needed.

table 3 AHB-2 List of bus masters

Main device	Features
MAC	802.11MAC Control protocol processing module. The operations on the bus mainly include operations such as sending read data, receiving write data, and sending back write completion descriptors.
SEC	Security module, complete sending and receiving data encryption and decryption and moving. When sending, the data and MAC The descriptor is moved to the specified location and the encryption is completed; when receiving, the received data and MAC The receiving descriptor is moved to the specified location and decryption is completed.
AHB2AHB	AHB-1 Bus to AHB-2 Conversion of bus master access to the bus.
SDIO/HSPI	Pass the host SDIO2.0 Device controller or high speed SPI The access from the device controller to the chip is converted to AHB Bus signals, and access to content memory and register space.

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

table 4 AHB-2 List of bus slaves

Slave device	Features
SRAM 128KB	Used to store upstream and downstream data cache, SDIO/SPI/UART The interface uses this RAM Used as Data cache
Configuration Register config	uration space, high-speed module configuration registers are uniformly addressed here.
APB	All low-speed modules access space, various low-speed modules are used APB Bus connection.

4.3 Clock structure

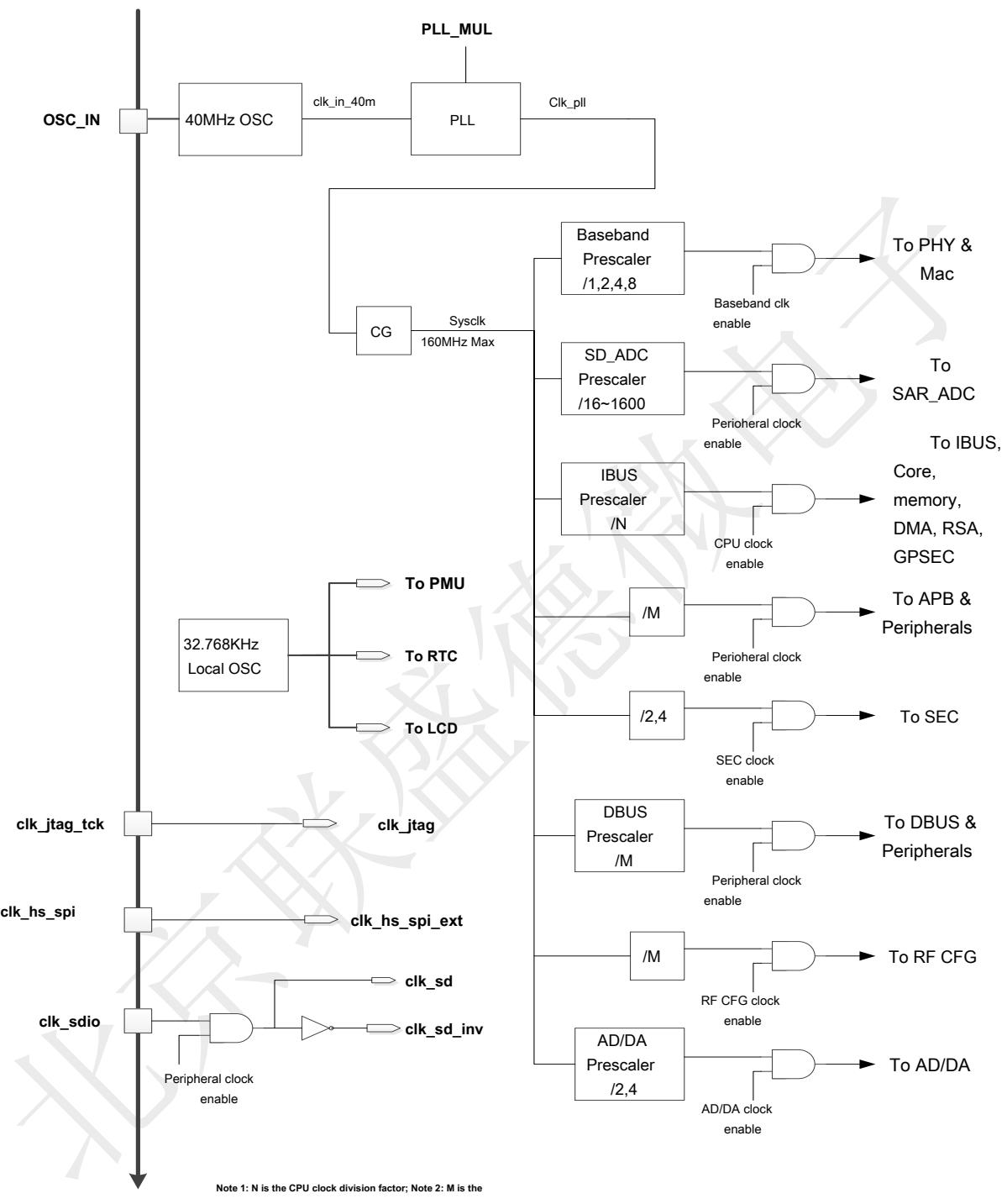


Fig 3 W600 Clock structure

4.4 Address space

Cortex-M3 Addressable 4G Spaces are code area, memory area, on-chip peripherals, off-chip storage area, off-chip peripherals and system peripheral area.

table 5 Address space distribution

Bus slave	Address space breakdown		Remarks
ROM	0x00000000 ~ 0x0003FFFF		Store the firmware code
FLASH	0x08000000 ~ 0x080FFFFFF		As a dedicated instruction memory, the firmware stores the operating area
SRAM	0x20000000 ~ 0x20027FFF		Program memory area
MAC RAM	0x20028000 ~ 0x20047FFF		SDIO/H_SPI/UART Data cache
CONFIG	0x40000000 ~ 0x40001FFFF	0x40000000 ~ 0x400001FF	SDIO Configuration space
		0x40000200 ~ 0x400002FF	SPI Configuration space
		0x40000300 ~ 0x400003FF	SDIO Wrapper Configuration space
		0x40000400 ~ 0x400005FF DMA Configuration space	
		0x40000600 ~ 0x400006FF	PMU Configuration space
		0x40000700 ~ 0x400007FF Clock versus Reset Configuration space	
		0x40000800 ~ 0x400009FF	
		0x4000A00 ~ 0x400013FF	
		0x40001400 ~ 0x400017FF	Wi-Fi Key register
		0x40001800 ~ 0x40001FFF	
		0x40002000 ~ 0x400021FF	FLASH Controller Configuration space
		0x40002200 ~ 0x400028FF RSA Configuration space	
		0x40003200 ~ 0x400033FF GPSEC Configuration space	
APB	0x40010000 ~ 0x4003C000	0x40010000 ~ 0x400101FF	I ² C Master
		0x40010200 ~ 0x400103FF	
		0x40010400 ~ 0x400107FF	SPI Master
		0x40010800 ~ 0x400109FF UART0	
		0x40010A00 ~ 0x40010BFF UART1	
		0x40010C00 ~ 0x40010DFF GPIOA	
		0x40010E00 ~ 0x40010FFF Timer 0x40011000	
		~ 0x400111FF WDOG 0x40011200 ~	
		0x400113FF GPIOB 0x40011400 ~	
		0x400115FF 0x40011600 ~ 0x400117FF	
		0x40011800 ~ 0x400119FF	
			PWM
		0x40011A00 ~ 0x40011BFF	I ² S
		0x40011C00 ~ 0x40011DFF	7816/UART2
		0x40014000 ~ 0x4000BFFF	
		0x4001C000 ~ 0x4003BFFF 0x40013C00 ~	
		0x5FFFFFFF RSV	

4.4.1 SRAM

W600 Built-in 288KB SRAM . among them 160KB Mount Level 1 AHB On the bus, 128KB Mount on level 2 AHB On the bus. CPU

Wait for the primary bus device to access all memory areas, but devices on the secondary bus can only access the secondary bus 128KB Memory.

4.4.2 Flash

4.4.2.1 QFlash

W600 Internal integration 1MBytes QFlash . Through on-chip integration 32KB cache achieve XIP Way in QFlash To execute the program.

While the program is running, CPU First from Cache Read instructions in 8Bytes One line way from QFlash

Read instructions, deposit Cache Inside. Therefore, when continuously running the code size is less than 32K Time, CPU Will not need to QFlash Read instructions,

at this time CPU Can run at a higher frequency. The above method is the operation mode of reading instructions. Image of RO Duan will use this method

Style operation. No user intervention is required during this process.

QFlash Can also store data when the user program needs to read and write QFlash Internal data, you need to pass the built-in QFlash Controller to operate

Make, QFlash Provide corresponding address, instruction and other registers to assist the user to achieve the desired operation. Please refer to the specific description QFlash control

Corresponding to the controller.

The user needs to note that when the program reads or writes data, there is no need to perform operations such as status judgment and waiting, because QFlash control

The controller itself will make the judgment. when QFlash When the controller returns, it indicates that the reading or writing has been completed.

4.4.2.2 SPI Flash

W600 In addition to chip support 6PIN of QFlash Outside the interface (built-in PIN , Unpackaged), also supports low speed SPI Interface access. The SPI

The maximum working frequency of the interface can reach 20MHz , Support master-slave function, detailed description reference SPI The interface corresponds to the chapter.

W600 The chip program can only be found in QFlash The corresponding space is executed. But when users need more storage space, you can

SPI Interface expansion Flash . User data, parameters, OTA Image Etc. can be placed outside SPI Flash space.

4.4.3 Bit zone

W600 Support bit band operation (Bit-Band Operations) .

4.4.3.1 Bit band operation

Bit-band operations use ordinary loading (load)/storage(store) Instruction to single BIT Read and write, which is equivalent to the bit band area

Every BIT All have an alias, access to the alias to replace the specified BIT Access. Designated BIT Mapping with alias

The process is completed directly by the kernel without manual intervention.

4.4.3.2 Cortex-M3 Bit zone operating area

- SRAM The lowest 1M range

- The lowest in the internal and external peripheral area 1M range '

4.4.3.3 The relationship between the address of the bit band and the alias of the bit band

for SRAM A certain bit of the area, remember the address of the byte where it is located as A , The bit number is n ($0 \leq n \leq 7$), the mapped alias area address

for:

$$\text{AliasAddr} = 0x2200000 + ((A - 0x20000000) * 8 + n) * 4 = 0x22000000 + (A - 0x20000000) * 32 + n * 4$$

For a certain bit of the on-chip peripheral bit zone, write the address of the byte where it is located as A, The bit number is n($0 \leq n \leq 7$) , The mapped alias area

The address is:

$$\text{AliasAddr} = 0x42000000 + ((A - 0x40000000) * 8 + n) * 4 = 0x42000000 + (A - 0x40000000) * 32 + n * 4$$

In the above formula, "4" Means a word is 4 Bytes, "8" Indicates that there is one byte 8 Bits.

The following figure shows the mapping relationship between the memory bit zone address and the bit zone alias zone address

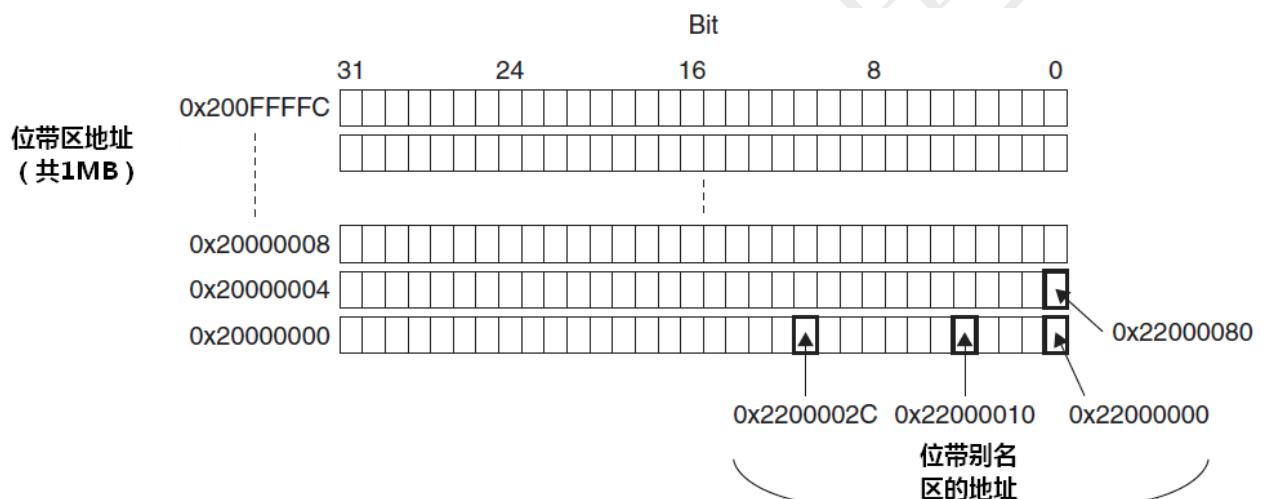


Fig 4 Memory bit zone mapping relationship

4.4.3.4 W600 Operable area

- 0X2000_0000-0X2004_8000
- 0X4000_0000-0X4001_3C00

4.5 Start configuration

W600 After the chip is powered on, CPU Will start execution ROM Firmware in, load Flash The user at the specified address Image .

ROM The firmware will read when it starts running BootMode(PA0) Pin, according to the signal of the pin to determine the start state:

table 6 Start configuration

BootMode	Starting conditions	Boot mode
high		Normal startup process
low	Continue< 30ms , The quick test mode is invalid	Normal startup process
	Continue< 30ms , And (PA1 &PA4&PB8&PB12 = 0110) Enter test mode	
	Continuous>= 30ms	Enter function mode

Note:

Test mode: chip test function, users can not operate.

Functional mode: Enter ROM Basic functions implemented, for example: downloading firmware, programming MAC Address, etc., detailed information reference " WM_W600_ROM Function Brief_V1.1.pdf »

usually, BootMode The pins should be used in the production or debugging phase. During the production phase, users BootMode The pin keeps pulling low 30ms Above, enter the function mode, you can quickly burn Flash jobs.

In the scenario of product rework or repair, the chip is not entered " Maximum security level " (For a description of the security level, please refer to " WM_W600_ROM Function Brief "), you can enter the function mode through this pin to erase the old Image , Write new Image .

During the commissioning phase, regardless of any malfunction of the firmware, you can BootMode The pin keeps pulling low 30ms Above, enter the serial download function and burn the new firmware.

5 Clock and reset module

5.1 Functional Overview

The clock and reset module complete the software control of the chip clock and reset system. Clock control includes clock frequency conversion, clock shutdown and adaptive gating; reset control includes system and sub-module soft reset control.

5.2 Main features

- Support clock shutdown of each module
- Supports adaptive shutdown of some module clocks
- Support software reset of each module
- stand by CPU Frequency setting
- stand by ADC/DAC Loopback test
- stand by I2S Clock settings

5.3 Functional description

5.3.1 Clock gating

By configuring the clock gating enable register CLK_GATE_EN Can control the clock shutdown of the specified function, so as to shut down a certain module Functional purpose.

In order to provide the flexibility of firmware to control the system power consumption, 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, the digital logic and clock tree of the module will stop working, which can reduce the dynamic power consumption of the system.

The corresponding register of the switch of each module SW_CLKG_EN Detailed description.

5.3.2 Clock adaptive shutdown

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

Please do not change the configuration, otherwise it may be in the configuration PMU Abnormal system caused by function.

5.3.3 Function reset

The chip provides the soft reset function of each subsystem, through setting SW_RST_CTRL corresponding BIT for 0 A subsystem reset can be achieved.

However, the reset state will not be cleared automatically. SW_RST_CTRL corresponding BIT position 1 .

Soft reset function does not reset CPU and WatchDog .

In this register, the APB/BUS1/BUS2(correspond APB Bus, system bus, and data bus) are not recommended for

The system access device is abnormal.

5.3.4 Clock divider

W600 System adoption 40MHz Crystal as the system clock source, the system is built-in DPLL , Fixed output 160MHz Clock as system-wide

Clock source (as shown below).

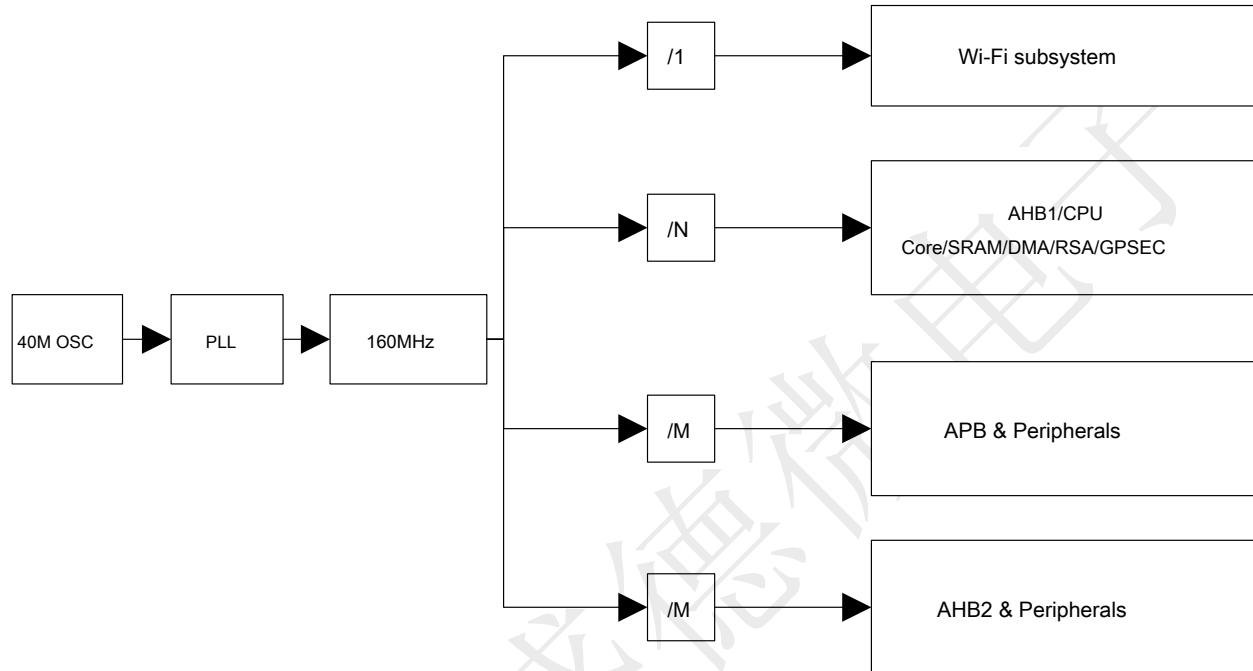


Fig 5 System clock frequency division relationship

System bus clock and CPU The clock is consistent, the clock of the data bus is fixed at WLAN Root clock 1/4 .

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

Settings are provided in this module CPU Clock with WLAN The function of the root clock is for firmware to adjust the system performance and power consumption.

Set up SYS_CLK_DIV Registered BIT[3:0] adjustable CPU Clock division factor. CPU The source clock for clock division is DPLL.

, The output is fixed at 160MHz . CPU The default value of the clock divider is 2 , which is CPU The default operating frequency is 160MHz of 2 Crossover,

which is 80MHz . When you need to adjust CPU When the clock is required, this parameter can be reconfigured.

Set up SYS_CLK_DIV Registered BIT[7:4] adjustable WLAN Clock division factor. The default division factor is 1 , That is wrong DPLL.

of 160MHz Output frequency division, get 160MHz Clock, given as the root node clock WLAN (WLAN Continue to divide the frequency to get more

Detailed low-frequency clock for WLAN System use).

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

When not needed WLAN When the system is working, you can WLAN The root clock is reduced, which reduces the system dynamic power consumption.

W600 in CPU/WLAN The clock division factor cannot be arbitrarily configured, otherwise, the configuration may not take effect, the following table is available

Frequency division configuration and corresponding system clock:

table 7 System clock frequency division parameter configuration

SYS_CLK_DIV		corresponding System clock (M HZ)		
BIT[7:4]	BIT[3:0]	CPU	WLAN	APB
1	2	80	160	40
1	4	40	160	40
4	8	20	40	10
5	10	16	32	8
5	4	40	32	8
4	4	40	40	10
5	5	32	32	8
8	8	20	20	5
10	10	16	16	4

When changing the system clock configuration, you need to pay attention: the ratio of the system bus to the data bus needs to be maintained at M : 1 ,among them M is an integer,

The minimum is 1 . When changing the system clock configuration, you also need to update the register at the same time SYS_CLK_DIV of BIT [11:8]. Set right

Scale factor. Otherwise, access to the data bus will get abnormal data.

SYS_CLK_DIV of[27:12] Settings provided SAR_ADC Divide factor of working frequency, to 40M Divide the clock source. Minute

The frequency coefficient is the assigned frequency division value.

SYS_CLK_DIV of BIT[28] For configuration RSA The clock frequency of the core operation of the module can be selected 40MHz or 80MHz .

When reconfiguration is required cpu_clk_divider , wlan_clk_divider , bus2_syncdn_factor , sdadc_fdiv Need to be set

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

I2S_CLK_CTRL provided I2S The module's clock configuration function.

5.3.5 Debug function control

Users can set DEBUG_CTRL Value of BIT5 To enable and disable JTAG Functional purpose.

5.4 Register description

5.4.1 Register list

table 8 Clock reset module register list

Offset address	name	abbreviation	access	description	Reset value
0X0000	Software clock gating enable register SW_CLK_EN	KG_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	Keep				
0X000C	Software reset control register	SW_RST_CTRL	RW	Software configuration reset module	0X01FF_FFFF
0X0010	Clock divider configuration register	SYS_CLK_DIV	RW	Configure clock division ratio	0X0000_2212
0X0014	Debug control register	DEBUG_CTRL	RW	Configuration ADC/DAC Loopback test	0X0000_0000
0X0018	I2S Clock control register	I2S_CLK_CTRL	RW	Configuration I2S clock	0X0000_0000

5.4.2 Software clock gating enable register

table 9 Software clock gating enable register

Bit access		Instructions	Reset value
[31:15]		Keep	
[14]	RW	soft_7816_gate_en Configuration 7816/uart2 Gating of module clock, default 7816 Module gate open 1'b0 : 7816 Module clock is off 1'b1 : 7816 Clock on	1'b1
[13]	RW	soft_gpsec_gate_en Configuration gpsec Gating of module clock, default gpsec Module gate open 1'b0 : gpsec Module clock is off 1'b1 : gpsec Clock on	1'b1
[12]	RW	soft_rsa_gate_en Configuration RSA Clock gating, default RSA Gating on 1'b0 : RSA Module clock is off 1'b1 : RSA Clock on	1'b1
[11]	RW	soft_i2s_gate_en Configuration i2s Clock gating, default i2s Gating on 1'b0 : i2s Clock off 1'b1 : i2s Clock on	1'b1
[10]	RW	soft_lcd_gate_en Configuration lcd Clock gating, default lcd Gating on	1'b1

		1'b0 : lcd Clock off 1'b1 : lcd Clock on	
[9]	RW	Soft_pwm_gate_en Configuration pwm Clock gating, default pwm Gating on 1'b0 : pwm Clock off 1'b1 : pwm Clock on	1'b1
[8]	RW	soft_sd_adc_gate_en Configuration sd_adc_ Clock gating, default sd_adc_ Gating on 1'b0 : sd_adc_ Clock off 1'b1 : sd_adc_ Clock on	1'b1
[7]	RW	soft_gpio_gate_en Configuration GPIO Clock gating, default GPIO Gating on 1'b0 : GPIO Clock off 1'b1 : GPIO Clock on	1'b1
[6]	RW	soft_timer_gate_en Configuration timer Clock gating, default timer Gating on 1'b0 : timer Clock off 1'b1 : timer Clock on	1'b1
[5]	RW	soft_rf_cfg_gate_en : For internal use, please do not modify Configuration rf_cfg Clock gating, default rf_cfg Gating on 1'b0 : rf_cfg Clock off 1'b1 : rf_cfg Clock on	1'b1
[4]	RW	soft_dma_gate_en Express supply dma Whether the clock in the clock domain is off 1'b0 : dma Clock off 1'b1 : dma Clock on	1'b1
[3]	RW	soft_ls_spi_gate_en Configure low speed spi Clock gating, default low speed spi Gating on 1'b0 : Low speed spi Clock off 1'b1 : Low speed spi Clock on	1'b1
[2]	RW	soft_uart1_gate_en Configuration uart1 Clock gating, default uart1 Gating on 1'b0 : uart1 Clock off 1'b1 : uart1 Clock on	1'b1
[1]	RW	soft_uart0_gate_en	1'b1

		Configuration uart0 Clock gating, default uart0 Gating on 1'b0 : uart0 Clock off 1'b1 : uart0 Clock on	
[0]	RW	soft_i2c_gate_en Configuration i2c Clock gating, default i2c Gating on 1'b0 : i2c Clock off 1'b1 : i2c Clock on	1'b1

5.4.3 Software clock mask register

table 10 Software clock mask register

Bit access		Instructions	Reset value
[6]	RW	soft_cpu_clk_gt_mask Express supply CPU Clock domain (including CPU , bus1 , ROM , SRAM) Whether the clock can be adaptive Off when CPU Need to enter WFI When in status, don't set adaptive shutdown) 1'b0 : Allows adaptive shutdown and opening 1'b1 : Do not allow adaptive shutdown and opening	1'b1
[5: 2]	RW	Keep, For internal use, please do not modify	
[1]	RW	soft_sdioahb_clk_gt_mask Express supply sdio ahb Whether the clock in the clock domain can be turned off adaptively 1'b0 : Allows adaptive shutdown and opening 1'b1 : Do not allow adaptive shutdown and opening	1'b1
[0]	RW	soft_pmu_clk_gt_mask pll There is a gating unit after the output clock. The register configuration is used to indicate whether PMU Shut down. 1'b0 :allow PMU Turn off the gating unit, thereby turning off the clock 1'b1 : Not allowed PMU Turn off the gating unit	1'b0

5.4.4 Software reset control register

table 11 Software reset control register

Bit access		Instructions	Reset value
[31:25]		Keep	
[twenty four]	RW	soft_RST_7816_n Software reset 7816/uart2 Module 1'b0 : Reset 1'b1 : Reset release	1'b1
[twenty three]	RW	soft_RST_gpsec_n	1'b1

		Software reset gpsec Module 1'b0 : Reset 1'b1 : Reset release	
[twenty two]	RW	soft_RST_RSA_n Software reset RSA Module 1'b0 : Reset 1'b1 : Reset release	1'b1
[twenty one]	RW	soft_RST_I2S_n Software reset I2S Module 1'b0 : Reset 1'b1 : Reset release	1'b1
[20]	RW	soft_RST_LCD_n Software reset LCD Module 1'b0 : Reset 1'b1 : Reset release	1'b1
[19]	RW	soft_RST_PWM_n Software reset PWM Module 1'b0 : Reset 1'b1 : Reset release	1'b1
[18]	RW	soft_RST_IO_CTRL_n Software reset IO_CTRL Module 1'b0 : Reset 1'b1 : Reset release	1'b1
[17]	RW	soft_RST_SAR_ADC_n Software reset SAR_ADC Module 1'b0 : Reset 1'b1 : Reset release	1'b1
[16]	RW	soft_RST_TIMER_n Software reset TIMER Module 1'b0 : Reset 1'b1 : Reset release	1'b1
[15]	RW	soft_RST_GPIO_n Software reset GPIO Module 1'b0 : Reset 1'b1 : Reset release	1'b1
[14]	RW	soft_RST_RF_CFG_n	1'b1

		Software reset configuration RF The register module (For internal use, please do not modify) 1'b0 : Reset 1'b1 : Reset release	
[13]	RW	soft_RST_SPIS_N Software reset high speed spi Module 1'b0 : Reset 1'b1 : Reset release	1'b1
[12]	RW	soft_RST_SPIM_N Software reset low speed spi 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 : Reset release	1'b1
[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

		1'b1 : Reset release	
[5]	RW	soft_rst_mem_mng_n Software reset mem_mng Module(For internal use, please 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 Software reset sdio ahb Clock domain module 1'b0 : Reset 1'b1 : Reset release	1'b1
[2]	RW	soft_rst_sec_n Software reset security module (For internal use, please do not modify) 1'b0 : Reset 1'b1 : Reset release	1'b1
[1]	RW	soft_rst_mac_n Software reset mac Module(For internal use, please do not modify) 1'b0 : Reset 1'b1 : Reset release	1'b1
[0]	RW	soft_rst_bbp_n Software reset bbp Module(For internal use, please do not modify) 1'b0 : Reset 1'b1 : Reset release	1'b1

5.4.5 Clock divider configuration register

table 12 Clock divider configuration register

Bit access		Instructions	Reset value
[31]	RW	divide_freq_en When reconfiguration is required cpu_clk_divider , wlan_clk_divider , bus2_syncdn_factor , sdadc_fdiv When this register is set, the hardware automatically updates the above four parameters to the divider, and then clears this register. 1'b0 : Frequency division factor has taken effect 1'b1 : Requires hardware to update the crossover parameters	1'b0

		Note: When configuring the frequency division factor here, when Divide_freq_en When valid, all factors must already be valid	
[30:29]		Keep	
[28]	RW	rsa_clk_sel RSA Clock selection: 1'b0: 40MHz 1'b1: 80MHz	1'b0
[27:12]	RW	sdadc_fdiv clk_sar_adc Frequency division factor: To 40M Divide the clock source. The frequency division factor is the assigned frequency division value.	16'h20
[11: 8]	RW	bus2_syncdn_factor bus1 with bus2 The clock ratio of should be N : 1 among them N It is an integer. In actual adjustment, it mainly depends on CPU Working frequency and bus2 The ratio of the clock frequency. Due to default cpu use 80MHz clock, bus2 use 40MHz Clock, then N=2	4'h2
[7: 4]	RW	wlan_clk_divider From PLL After dividing the clock out, give it to wlan system. This register is the frequency division factor, the factor >= 1 . The default division factor is 1 , That is wrong pll of 160MHz Output frequency division, get 160MHz Clock as root Node clock to wlan (wlan Continue to divide the frequency to get a more detailed low-frequency clock); Note: Secondary bus and APB The clock is wlan Root node clock divided by four	4'b1
[3: 0]	RW	cpu_clk_divider From PLL After dividing the clock out, give it to CPU . This register is the frequency division factor, the factor >= 2 . The default division factor is 2 , That is, after reset release, PLL Exported 160MHz clock 2 Divide, give to cpu Is 80MHz clock. When you need to adjust cpu When the clock is needed, this parameter can be reconfigured	4'h2

5.4.6 Debug control register

table 13 Debug control register

Bit access		Instructions	Reset value
[5]	RW	JTAG Enable 1'b0: Forbid JTAG Debugging function 1'b1: Enable JTAG Debugging function	1'b0
[4:0]	RW	Keep, Do not modify	1'b0

5.4.7 I2S Clock control register

table 14 I2S Clock control register

Bit access		Instructions	Reset value

[31:18]		Keep	
[17: 8]	RW	<p>BCLKDIV</p> <p>BCLK Distributor: $F_{BCLK} = F_{I2SCLK} / BCLKDIV$</p> <p>Note: If not selected EXTAL_EN While using internal PLL then $F_{I2SCLK} = F_{CPU}$ (versus CPU frequency) .</p> <p>Same rate).</p> <p>Suppose $F_{CPU} = 160MHz$, Enable WXTAL_EN Time $F_{I2SCLK} =$ External crystal frequency,</p> <p>$BCLKDIV = \text{round} (F_{I2SCLK}/(Fs*W*F))$</p> <p>among them Fs Is the sampling frequency of the audio data, W Is word width</p> <p>When the data is mono F = 1 ;</p> <p>When the data is stereo F = 2 .</p> <p>For example, if you use internal PLL And the data width is twenty four Bit, the format is stereo, and the sampling frequency is 128KHz , BCLKDIV Should be configured as $(160*10e6/128*10e3*24*2)= 10'h1a$.</p>	10'b0
[7: 2]	RW	<p>MCLKDIV</p> <p>If you select an external clock, then the MCLK The frequency divider is used to generate the appropriate MCLK frequency .</p> <p>$F_{mclk} = F_{I2SCLK} / (2 * MCLKDIV) ;$</p> <p>when MCLKDIV = 0 Time F_{I2SCLK} Is an external clock;</p> <p>when MCLKDIV >= 1 Time $F_{mclk} = F_{I2SCLK} ;$</p> <p>note: F_{mclk} Should be configured as $256 * fs$,among them fs Is the sampling frequency.</p>	6'b0
[1]	RW	<p>MCLKEN</p> <p>MCLK Enable switch</p> <p>1'b0 : Prohibited MCLK</p> <p>1'b1 :Enable MCLK</p>	1'b0
[0]	RW	<p>EXTAL_EN</p> <p>External clock selection, choose to use internal I2S Block clock or external clock</p> <p>1'b0 : Internal clock</p> <p>1'b1 : External clock</p> <p>Note: When using an external clock, the external clock must be $2 * N * 256 fs$,among them fs Is the sampling frequency, N must</p> <p>Must be an integer.</p>	1'b0

6 DMA Module

6.1 Functional Overview

DMA Used to provide high-speed data transfer between peripherals and memory and between memory and memory. Can be done without any CPU operating

In the case of DMA Move data quickly. So saved CPU Resources without affecting CPU Perform other instructions.

DMA Mount on AHB On the bus, up to support 8 aisle, 16 A hardware peripheral request source supports linked list structure and register control.

6.2 Main features

- Amba2.0 Standard bus interface, 8 road DMA aisle
- Support based on memory linked list structure DMA operating
- stand by 16 Hardware peripheral request source
- stand by 1 , 4-burst Operating mode
- stand by byte , half-word , word Unit transfer operation
- Support source and destination addresses unchanged or in ascending order or can be configured to cycle in a predefined address range
- Supports data transfer from memory to memory, memory to peripherals, and peripherals to memory

6.3 Functional description

6.3.1 DMA aisle

W600 Total support 8 road DMA aisle, DMA The channels do not interfere with each other and can run simultaneously. Request different data streams can choose not Same DMA aisle.

Each DMA The channels are allocated in different register address offset segments, you can directly select the address segment of the corresponding channel for configuration and use.

The register configuration of different channels is completely the same.

table 15 DMA Address assignment

DMA Base address	0x4000 0400
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

8 road DMA The channel can realize a unidirectional data transmission link between the source and the destination.

DMA The source and destination addresses can be set to DMA After the operation is completed, there are three modes: unchanged, incremental or circular:

- DMA_CTRL[2:1] Control source address every time DMA Change mode after operation;
- DMA_CTRL[4:3] Control destination address every time DMA Change mode after operation.

DMA Can be set byte , half-word , word Transport unit, the number of final transport data is an integer multiple of the transport unit, through

DMA_CTRL[6:5] To set.

DMA able to pass burst Set how many units of data to move at a time, pass DMA_CTRL[7] To choose a move 1 or 4 Single

Bits of data if DMA_CTRL[6:5] Set as word , burst Set as 4 , Then each time 4 Pc word The data.

DMA Can be set to start every time DMA Transmission Byte Number, max 65535 Byte ,by DMA_CTRL[23:8] To set.

6.3.3 DMA Cycle mode

DMA Circular address mode refers to setting DMA After the source and destination addresses of the data, after the data movement reaches the set cycle boundary, it will jump

To the start address of the loop, it is executed in a loop until it reaches the set transfer byte.

The source and destination addresses of the circular address mode are required SRC_WRAP_ADDR with DEST_WRAP_ADDR Register to set, and through

Past WRAP_SIZE To set the length of the loop.

6.3.4 DMA Transmission mode

DMA stand by 3 Various transmission modes:

- Memory to memory

Both the source address and the destination address are configured as memory addresses that need to be transferred, DMA_MODE[0] Set as 0 , Software way.

- Memory to peripheral

The source address is set to the memory address, and the destination address is set to the peripheral address, DMA_MODE[0] Set as 1 , Hardware method,

DMA_MODE[5:2] Select the peripheral used.

- Peripheral to memory

The source address is set to the peripheral address, and the destination address is set to the memory address, DMA_MODE[0] Set as 1 , Hardware method,

DMA_MODE[5:2] Select the peripheral used.

6.3.5 DMA Peripheral selection

When using a peripheral to memory or memory to peripheral transfer method, in addition to the corresponding peripherals need to be set to DMA TX or RX

outer, DMA_MODE[5:2] You also need to select the corresponding peripheral.

Note: because UART Mouth 3 In UART use DMA When you need to pass UART_CH[1:0] To select the corresponding

UART .

6.3.6 DMA Linked list mode

DMA Support linked list working mode. Through the linked list mode, we are DMA When moving the current data in the linked list memory, you can advance down

Fill the data in a linked list, DMA After moving the current linked list, it is judged that the next linked list is valid, and the next linked list can be directly transported.

data. It can be effectively improved through a linked list DMA with CPU The efficiency of coordination.

Linked list operation mode: through DMA_MODE[1] Register settings DMA Work for the linked list, and then DESC_ADDR Register set

Set to the starting address of the linked list structure, and then pass CHNL_CTRL Register enable DMA . when DMA Processing complete the current memory

After moving, the software notifies by setting a valid flag DMA There are still valid data in the linked list, DMA Valid mark according to linked list

Process the next data to be moved.

6.3.7 DMA Interrupt

DMA Transfer complete or burst Can generate interrupts, INT_MASK Register can be masked DMA The interrupt corresponding to the channel.

when DMA After the corresponding interrupt is generated, you can pass INT_SRC The register queries the current interrupt status, indicating what is currently being generated

Off, the corresponding status bit needs to be written by software 1 clear 0 .

6.4 Register description

6.4.1 Register list

table 16 DMA Register list

Offset address	name	abbreviation	access	description	Reset value
0X0000	Interrupt mask register	INT_MASK	RW Set to be blocked DMA Interrupt		0X0000_FFFF
0X0004	Interrupt status register	INT_SRC	RW Indicate current DMA Out of state		0X0000_0000
0X0008	UART Select register	UART_CH	RW UART Which peripheral to choose UART 0X0000_0000		
0X000C	Keep				
DMA CHNL0 registers					
0X0010	DMA Source address register	SRC_ADDR	RW DMA Source address of transmission		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 In loop mode DMA Transmission source address 0X0000_0000		
0X001C	DMA Loop destination start address register DEST_WRAP_ADDR		RW In loop mode DMA Transmission destination site		0X0000_0000
0X0020	DMA Loop length register	WRAP_SIZE	RW In loop mode DMA Cycle boundary		0X0000_0000
0X0024	DMA Channel control register	CHNL_CTRL	RW Current channel DMA Start and stop		0X0000_0000
0X0028	DMA Mode selection register	DMA_MODE	RW Set up DMA Way of working		0X0000_0000
0X002C	DMA Data Flow Control Register	DMA_CTRL	RW Set up DMA Transport data stream		0X0000_0000
0X0030	DMA Transfer Bytes Register	DMA_STATUS	RO Get the number of bytes currently transferred		0X0000_0000
0X0034	DMA Link list entry address register DESC_ADDR		RW DMA Link list address entry address setting 0X0000_0000		
0X0038	DMA Current destination address register	CUR_DEST_ADDR	RO current DMA Operating address		0X0000_0000

DMA CHNL1 registers	
0X0040- 0X0068	with DMA CHNL0 registers
DMA CHNL2 registers	
0X0070- 0X0098	with DMA CHNL0 registers
DMA CHNL3 registers	
0X00A0- 0X00C8	with DMA CHNL0 registers
DMA CHNL4 registers	
0X00D0- 0X00F8	with DMA CHNL0 registers
DMA CHNL5 registers	
0X0100- 0X0128	with DMA CHNL0 registers
DMA CHNL6 registers	
0X0130- 0X0158	with DMA CHNL0 registers
DMA CHNL7 registers	
0X0160- 0X0188	with DMA CHNL0 registers

6.4.2 Interrupt mask register

table 17 DMA Interrupt mask register

Bit access		Instructions	Reset value
[31:16]		Keep	
[15]	RW	channel7 transfer_done interrupt masking, high effective.	1'b1
[14]	RW	channel7 burst_done interrupt masking, high effective.	1'b1
[13]	RW	channel6 transfer_done interrupt masking, high effective.	1'b1
[12]	RW	channel6 burst_done interrupt masking, high effective.	1'b1
[11]	RW	channel5 transfer_done interrupt masking, high effective.	1'b1
[10]	RW	channel5 burst_done interrupt masking, high effective.	1'b1
[9]	RW	channel4 transfer_done interrupt masking, high effective.	1'b1
[8]	RW	channel4 burst_done interrupt masking, high effective.	1'b1
[7]	RW	channel3 transfer_done interrupt masking, high effective.	1'b1
[6]	RW	channel3 burst_done interrupt masking, high effective.	1'b1
[5]	RW	channel2 transfer_done interrupt masking, high effective.	1'b1
[4]	RW	channel2 burst_done interrupt masking, high effective.	1'b1
[3]	RW	channel1 transfer_done interrupt masking, high effective.	1'b1
[2]	RW	channel1 burst_done interrupt masking, high effective.	1'b1
[1]	RW	channel0 transfer_done interrupt masking, high effective.	1'b1

[0]	RW	channel0 burst_done Interrupt masking, high effective.	1'b1
-----	----	--	------

6.4.3 Interrupt status register

table 18 DMA Interrupt status register

Bit access		Instructions	Reset value
[31:16]		Keep	
[15]	RW	channel7 transfer_done Interrupt status, write 1 clear 0 . DMA An interrupt is generated when the transfer is complete.	1'b0
[14]	RW	channel7 burst_done Interrupt status, write 1 clear 0 . DMA burst Completion generates an interrupt.	1'b0
[13]	RW	channel6 transfer_done Interrupt status, write 1 clear 0 . DMA An interrupt is generated when the transfer is complete.	1'b0
[12]	RW	channel6 burst_done Interrupt status, write 1 clear 0 . DMA burst Completion generates an interrupt.	1'b0
[11]	RW	channel5 transfer_done Interrupt status, write 1 clear 0 . DMA An interrupt is generated when the transfer is complete.	1'b0
[10]	RW	channel5 burst_done Interrupt status, write 1 clear 0 . DMA burst Completion generates an interrupt.	1'b0
[9]	RW	channel4 transfer_done Interrupt status, write 1 clear 0 . DMA An interrupt is generated when the transfer is complete.	1'b0
[8]	RW	channel4 burst_done Interrupt status, write 1 clear 0 . DMA burst Completion generates an interrupt.	1'b0
[7]	RW	channel3 transfer_done Interrupt status, write 1 clear 0 . DMA An interrupt is generated when the transfer is complete.	1'b0
[6]	RW	channel3 burst_done Interrupt status, write 1 clear 0 . DMA burst Completion generates an interrupt.	1'b0
[5]	RW	channel2 transfer_done Interrupt status, write 1 clear 0 . DMA An interrupt is generated when the transfer is complete.	1'b0
[4]	RW	channel2 burst_done Interrupt status, write 1 clear 0 . DMA burst Completion generates an interrupt.	1'b0
[3]	RW	channel1 transfer_done Interrupt status, write 1 clear 0 . DMA An interrupt is generated when the transfer is complete.	1'b0
[2]	RW	channel1 burst_done Interrupt status, write 1 clear 0 . DMA burst Completion generates an interrupt.	1'b0
[1]	RW	channel0 transfer_done Interrupt status, write 1 clear 0 . DMA An interrupt is generated when the transfer is complete.	1'b0
[0]	RW	channel0 burst_done Interrupt status, write 1 clear 0 . DMA burst Completion generates an interrupt.	1'b0

6.4.4 UART Select register

table 19 UART Select register

Bit access		Instructions	Reset value
[31: 2]		Keep	
[1: 0]	RW	uart dma channel select: 2'h0: uart0 Module dma Channel access dma 2'h1: uart1 Module dma Channel access dma 2'h2: uart2/7816 Module dma Channel access dma 2'h3: Keep	2'h0

6.4.5 DMA Source address register

table 20 DMA Source address register

Bit access		Instructions	Reset value
[31: 0]	RW	In acyclic mode, DMA Source address, peripheral address or memory address	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, DMA Transport destination address, peripheral address or memory address	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 loop mode, DMA Starting address of the source address to be moved, peripheral address or memory address	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 loop mode, DMA Starting address of the destination address, peripheral address or memory address	32'h0

6.4.9 DMA Loop length register

table 24 DMA Loop length register

Bit access		Instructions	Reset value
[31:16]	RW	In loop mode, DMA Destination address cycle length. DMA Increment the handling data from the starting address, when the number of bytes of handling data reaches this set value, it will jump to the loop starting address, and continue to carry the data from the starting address	16'h0
[15: 0]	RW	In loop mode, DMA Source address cycle length.	16'h0

6.4.10 DMA Channel control register

table 25 DMA Channel control register

Bit access		Instructions	Reset value
[31: 2]		Keep	
[1]	RW	dma_stop stop dma Operation, high efficiency. DMA Will complete the current burst Stop after operation and clear at the same time chnl_on . The software should be based on chnl_on for 0 determine dma It has completely stopped.	1'b0
[0]	RW	chnl_on Start current channel DMA Conversion, highly effective.	1'b0

		Dma After the conversion is completed or the setting is stopped, it is automatically cleared 0 .	
--	--	--	--

6.4.11 DMA Mode selection register

table 26 DMA Mode selection register

Bit access		Instructions	Reset value
[31: 7]		Keep	
[6]	RW	<p>chain_link_en</p> <p>Effective in linked list mode, indicating dma After processing the first linked list, whether to continue to read and process subsequent linked lists. If 1 , Then update the list next_desc_addr And continue to read the next linked list until the linked list valid for 0 ; If 0 , The process stops after the current linked list is completed.</p>	1'b0
[5: 2]	RW	<p>dma_sel</p> <p>When transferring peripherals, select the current DMA The peripheral type of the channel.</p> <p>16 Pcs dma_req's Choice.</p> <p>4'h0 : uart rx dma req</p> <p>4'h1 : uart tx dma req</p> <p>4'h2 : pwm_cap0_req</p> <p>4'h3 : pwm_cap1_req</p> <p>4'h4 : LS_SPI rx dma req</p> <p>4'h5 : LS_SPI tx dma req</p> <p>4'h6 : SD_ADC chnl0 req</p> <p>4'h7 : SD_ADC chnl1 req</p> <p>4'h8 : SD_ADC chnl2 req</p> <p>4'h9 : SD_ADC chnl3 req</p> <p>4'ha : SD_ADC chnl4 req</p> <p>4'hb : SD_ADC chnl5 req</p> <p>4'hc : SD_ADC chnl6 req</p> <p>4'hd : SD_ADC chnl7 req</p> <p>4'he: I2S RX req</p> <p>4'hf: I2S TX req</p>	4'h0
[1]	RW	<p>chain_mode</p> <p>1'b0 : Use normal mode</p> <p>1'b1 : Use linked list mode</p>	1'b0
[0]	RW	<p>dma_mode</p> <p>1'b0 : Software mode.</p> <p>1'b1 : Hardware mode.</p>	1'b0

6.4.12 DMA Data Flow Control Register

table 27 DMA Data Flow Control Register

Bit access		Instructions	Reset value
[31:24]		Keep	

[23: 8]	RW	total_byte Total operations byte Number. Need to communicate with data_size The configuration is consistent, that is, if it is a word operation, it should be configured as 4 An integer multiple of; if it is a halfword operation, it should be configured as 2 An integer multiple of.	16'h0
[7]	RW	burst_size Set up DMA How many units of data are moved at a time 1'b0 : burst for 1 1'b1 : burst for 4 When the last time burst When the size exceeds the number of remaining transmissions, use burst The size is the size of the remaining data.	1'b0
[6: 5]	RW	data_size Set up DMA Handling unit 2'h0 : byte 2'h1 : half_word 2'h2 : word 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 jumps to the loop start address when the defined loop boundary is reached.	2'h0
[twenty one]	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 the jump to the loop start address is reached when the defined loop boundary is reached.	2'h0
[0]	RW	auto_reload When finished current DMA After handling, automatically press the current DMA Configure again next time DMA Handling.	1'b0

6.4.13 DMA Transfer Bytes Register

table 28 DMA Transfer Bytes Register

Bit access		Instructions	Reset value
[31:16]		Keep	
[15: 0]	RW	transfer_cnt The number of bytes currently being transferred. Every time dma (chnl_on Set 1)clear 0 And restart counting.	16'h0

6.4.14 DMA Link list entry address register

table 29 DMA Link list entry address register

Bit access		Instructions	Reset value
[31: 0]	RW	<p>desc_addr</p> <p>When the linked list is enabled, it is used as the entry address of the linked list. After each link list is completed, the base address of the next link list is updated to this register.</p>	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	<p>current_dest_addr</p> <p>current DMA Operation destination address.</p> <p>When the software stops dma Can be learned by looking at this register dma The destination address to be operated.</p>	32'h0

7 General hardware encryption module

7.1 Functional Overview

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

Time; support SHA1/MD5/RC4/DES/3DES/AES/CRC/PRNG .

7.2 Main features

- stand by SHA1/MD5/RC4/DES/3DES/AES/CRC/PRNG Encryption Algorithm
- DES/3DES stand by ECB with CBC Two modes
- AES stand by ECB , CBC with CTR Three modes
- CRC stand by CRC8 , CRC16_MODBUS , CRC16_CCITT with CRC32 Four modes
- CRC Support input/output reverse
- SHA1/MD5/CRC Support continuous multi-packet encryption
- Pseudo-random numbers support continuous generation 16 Bit sum 32 Bit random number, support seed seed

7.3 Functional description

7.3.1 SHA1 encryption

Can perform hardware on continuous multi-packet data SHA1 Calculation, the calculation result is stored in the register, the encryption result of the previous packet can be used as the next packet

Initial value.

7.3.2 MD5 encryption

Can perform hardware on continuous multi-packet data MD5 Calculation, the calculation result is stored in the register, the encryption result of the previous packet can be used as the next packet

Initial value.

7.3.3 RC4 encryption

stand by RC4 Encrypt and decrypt.

7.3.4 DES encryption

stand by DES Encryption and decryption, support ECB with CBC Two modes.

7.3.5 3DES encryption

stand by 3DES Encryption and decryption, support ECB with CBC Two modes.

7.3.6 AES encryption

stand by AES Encryption and decryption, support ECB , CBC with CTR Three modes.

7.3.7 CRC encryption

Can perform hardware on continuous multi-packet data CRC Calculation, the calculation result is stored in the register, the encryption result of the previous packet can be used as the next packet

Initial value, support CRC8 , CRC16_MODBUS , CRC16_CCITT with CRC32 Four modes, support input/output reverse.

CRC32 Calculated as follows:

1 , CRC-32: 0x04C11DB7

$$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$$

Commonly used ZIP, RAR, IEEE 802 LAN/FDDI, IEEE 1394, PPP-FCS And other agreements.

2 , CRC-16: Supports two polynomials

2.1: 0X1021

$$X^{16} + X^{12} + X^5 + 1$$

Commonly used ISO HDLC, ITU X.25, V.34/V.41/V.42, PPP-FCS CCITT And other agreements.

2.2: 0X8005

$$X^{16} + X^{15} + X^2 + 1$$

Commonly used USB, ANSI X3.28, SIA DC-07 And other agreements.

3 , CRC-8 : 0X207

$$x^8+x^2+x^1+1$$

7.3.8 PRNG Pseudo-random number

Can be produced continuously 16 Bit or 32 Bit pseudo-random numbers, supporting random seeds, random results stored in registers.

7.4 Register description

7.4.1 Register list

table 31 Encryption module register list

Offset address	name	abbreviation	access	description	Reset value
0X0000	Source address register	SRC_ADDR	RW RC4/SHA1/AES/DES/3DES/CRC/MD5 Multiplex source address		0X0000_0000
0X0004	Destination address register	DEST_ADDR	RW RC4/AE/S/DES/3DES Multiplex destination address		0X0000_0000
0X0008	Configuration register	GPSEC_CFG	RW General	hardware encryption module configuration register	0X0000_0000
0X000C	Control register	GPSEC_CTRL	RW General	hardware encryption module control register	0X0000_0000
0X0010	Secret key 0 Low register	KEY00	RW Key0	low 32 First input key (RC4/AES/DES/3DES), reuse CRC Ci	0X0000_0000
0X0014	Secret key 0 High register	KEY01	RW Key0	high 32 First input key (RC4/AES/DES/3DES)	0X0000_0000

0X0018 Secret key 1 Low register	KEY10	RW Key1	low 32 Second input key (RC4/AES//3DES)	0X0000_0000
0X001C Secret key 1 High register	KEY11	RW Key1	high 32 Second input key (RC4/AES//3DES)	0X0000_0000
0X0020 Secret key 2 Low register	KEY20	RW Key2	low 32 Third input key (3DES), Reuse iv1 low 32 Bit input initial vector (AES)	0X0000_0000
0X0024 Secret key 2 High register	KEY21	RW Key2	high 32 Third input key (3DES), Reuse iv1 high 32 Bit input initial vector (AES)	0X0000_0000
0X0028 Initial vector 0 Low register IV00		RW	IV0 low 32 Bit input initial vector (AES/DES/3DES)	0X0000_0000
0X002C Initial vector 0 High register IV01		RW	IV0 high 32 Bit input initial vector (AES/DES/3DES)	0X0000_0000
0X0030 Status register	GPSEC_STS	RW General	hardware encryption 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	0XEFC0_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	RW RNG	Output	0X0000_0000

7.4.2 Configuration register

table 32 Encryption module configuration register

Bit access		Instructions	Reset value
[31]		Keep	
[30]	RW	RNG start 1'b0 : Do not start RNG 1'b1 : start up RNG	1'b0
[29]	RW	RNG Load_seed Automatic hardware cleaning 0 1'b0 : The random number generator will default to zero as the seed to generate a random number with the corresponding number of digits 1'b1 : Start random number generation after seed loading is completed	1'b0
[28]	RW	RNG switch Control the number of digits for generating random numbers, 1'b0 : 16 Bit 1'b1 : 32 Bit	1'b0
[27]	RW	des_soft_reset des Hardware reset automatically after soft reset 0 1'b0 : Does not generate a soft reset and does not change the current state 1'b1 : The encryption algorithm is reset to the initial state by the software	1'b0
[26]	RW	aes_soft_reset aes Hardware reset automatically after soft reset 0	1'b0

		1'b0 : Does not generate a soft reset and does not change the current state 1'b1 : The encryption algorithm is reset to the initial state by the software	
[25]	RW	rc4_soft_reset rc4 Hardware reset automatically after soft reset 0 1'b0 : Does not generate a soft reset and does not change the current state 1'b1 : The encryption algorithm is reset to the initial state by the software	1'b0
[twenty four]	RW	crc_datarev 1'b0 : CRC Input data is not reversed 1'b1 : CRC Input data reverse	1'b0
[twenty three]	RW	crc_chksrev 1'b0 : CRC The output is not reversed 1'b1 : CRC The output is reversed	1'b0
[22:21]	RW	sub_mode Algorithm type sub-mode selection: 2'b00 : DES/AES Cryptographic algorithm ECB Mode, reusable CRC Algorithmic CRC8 mode 2'b01 : 3DES/AES Cryptographic algorithm CBC , Reusable CRC Algorithmic CRC16_0 mode 2'b10 : AES Cryptographic algorithm CTR Mode, reusable CRC Algorithmic CRC16_1 mode 2'b11 : CRC Algorithmic CRC32	2'b0
[20]	RW	encrypt_decrypt RC4/AES/DES/3DES Algorithm encryption or decryption mode selection: 1'b0 : encryption 1'b1 : Decrypt	1'b0
[19]	RW	gpsec_int_mask 1'b0 : Do not block the encryption/decryption completion interrupt 1'b1 : Blocking encryption/decryption completion interrupt	1'b0
[18:16]	RW	cypher_mode Types of cryptographic algorithms 3'b000 : RSV 3'b001 : RC4 3'b010 : SHA1 3'b011 : AES 3'b100 : DES 3'b101 : 3DES 3'b110 : CRC 3'b111 : MD5	3'b0
[15: 0]	RW	total_byte A total of encryption and decryption operations are required byte Number.	16'h0

7.4.3 Control register

table 33 Encryption module control register

Bit access		Instructions	Reset value
[31: 2]		Keep	
[1]	RW	<p>sec_stop</p> <p>Stop the current encryption and decryption operation</p> <p>1'b0 :invalid</p> <p>1'b1 : Stop encryption/decryption</p>	1'b0
[0]	RW	<p>sec strt</p> <p>Start encryption and decryption, complete the encryption and decryption operation byte After the number, the hardware is automatically cleared 0</p> <p>1'b0 : Do not start encryption/decryption</p> <p>1'b1 : Start encryption/decryption</p>	1'b0

7.4.4 Status register

table 34 Encryption module status register

Bit access		Instructions	Reset value
[31: 17]		Keep	
[16]	RW	<p>int_flag</p> <p>Software write 1 Clear</p> <p>1'b0 : Does not generate encryption/decryption completion interrupt</p> <p>1'b1 : Generate encryption/decryption completion interrupt</p>	1'b0
[15: 0]	RO	<p>transfer_cnt</p> <p>The number of bytes currently encrypted. Clear every time you turn on encryption and decryption again 0 And restart counting.</p>	16'h0

8 RSA Encryption module

8.1 Functional Overview

RSA Computational hardware coprocessor, provided Montgomery(FIOS Algorithm) Modular multiplication operation function. Cooperate RSA Software library implementation RSA algorithm. stand by 128 In place 2048 Bit mode multiplication.

8.2 Main features

- stand by 128 In place 2048 Bit modular multiplication (modulo multiplication length is 32 Integer multiples)
- stand by D*D ; X*Y ; D*Y ; X*X Wait 4 Modular multiplication mode

8.3 Functional description

8.3.1 Modular multiplication function

RSA Computational hardware coprocessor, provided Montgomery (FIOS Algorithm) Modular multiplication operation function. Cooperate RSA Software library RSA algorithm. stand by 128 In place 2048 Bit mode multiplication.

8.4 Register description

8.4.1 Register list

table 35 RSA Register list

Offset address	name	Abbreviated access		description	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 Corresponding data X (2048bit)'S buffer, corresponding haddr Value is 0000h~00fch . The corresponding rules are as follows:

table 36 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 Corresponding data Y (2048bit)'S buffer, corresponding haddr Value is 0100h~01fch . The corresponding rules are as follows:

table 37 RSA data Y register

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

8.4.4 data M register

MBUF Corresponding data M (2048bit)'S buffer, corresponding haddr Value is 0200h~02fch . The corresponding rules are as follows:

table 38 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 Corresponding data D (2048bit)'S buffer, corresponding haddr Value is 0300h~03fch . The corresponding rules are as follows:

table 39 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 32bit register.

table 40 RSA Control register

Bit access		Instructions	Reset value
[31: 6]		Keep	
[5]	RW	Modulation start control bit. Software write "1" Start the modular multiplication operation, after the operation is completed, the hardware will automatically clear "0".	
[4]	RW	Provide soft reset function, high efficiency. Software write "1" Perform a soft reset. After the reset is completed, the hardware is automatically cleared "0". 1. Setting parameters MC with N for 0 . 2. After starting the modular multiplication (bit5 Set 1), place this "1" , Will terminate the current operation (when bit0 Go high, indicating that the soft reset command is completed and the operation is terminated), but the internal data buffer (X , Y , M , D) Part of the operation results that have been completed will be retained.	
[twenty three]	RW	Modular mode selection. 2'b00 : X = D*D mod M 2'b01 : D = X*Y mod M 2'b10 : X = D*Y mod M 2'b11 : D = X*X mod M	2'b0
[1]	RW	Keep	1'b0
[0]	RW	The modular multiplication operation completes the identification and is highly effective. Hardware "1" , Software clear "0" . Software write "1" is valid.	

8.4.7 parameter MC register

table 41 RSA parameter MC register

Bit access		Instructions	Reset value

[31: 0]	WO	RSAMC Corresponding parameters MC (32bit). Reset value 0 . The readout value is full 0 .	32'h0
---------	----	--	-------

8.4.8 parameter N register

RSAN Corresponding parameters N (7bit). N The value is the modulo length divided by 32 Value. Ie if you call 1024bit The modular multiplication operation needs to be set N = 32 . When writing to this register, take low 7 Bit is valid data, when read, high 25 Bit is 0 . Reset value 0 .

table 42 RSA parameter N register

Bit access		Instructions	Reset value
[31: 7]		Keep	
[6: 0]	RW	RSAN Corresponding parameters N (7bit). N The value is the modulo length divided by 32 Value.	7'h0

9 GPIO Module

9.1 Functional Overview

GPIO The controller implements a software pair GPIO Attribute configuration allows users to operate conveniently 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

- stand by GPIO Software configuration
- stand by GPIO Interrupt configuration
- Provide up to 17 Pcs GPIO Available

9.3 Functional description

W600 Provided in GPIO Divided into two groups, one group is GPIOA , A group is GPIOB , GPIOA with GPIOB Register start address is not

Same, but same function.

When the user wants to IO As software controlled GPIO If used, will GPIO The corresponding position in the multiplex selection register is 0 which is can.

GPIO The direction control register is used to control GPIO Direction, 1 Corresponding GPIO As an output pin, 0 Corresponding GPIO

As an input pin.

GPIO The up and down control register is used to control the corresponding IO Up and down function. This register is active low, set to 0 Means open the corresponding IO

The up and down function of is set to 1 Indicates that the up and down functions are turned off. Each IO There is only one up and down state, IO See the attributes IO

Reuse table.

GPIO Data register indicates input when set to input state IO Can be written when set to output state 1 or 0 Finger

set IO Output level. This register is subject to GPIO The control of the data enable register is only GPIO The data enable register is set to 1

when, GPIO Data registers can be read and written.

GPIO The module provides input signal detection function. By configuration GPIO Interrupt related registers can realize high and low level detection and up and down

Edge transition detection. When corresponding IO The input signal meets the preset conditions, such as high level trigger or rising edge trigger, etc.

trigger GPIO Interrupt, report to MCU deal with. MCU The corresponding interrupt status needs to be cleared to avoid false triggering of the interrupt.

9.4 Register description

9.4.1 Register list

table 43 GPIOA Register list

Offset address	name	abbreviation	access	description	Reset value
0X0000	GPIO Data register	GPIO_DATA	RW Read and write GPIO Current data		0X180B
0X0004	GPIO Data enable register	GPIO_DATA_E_N	RW Configuration	GPIO_DATA Enable bit	0xFFFF
0X0008	GPIO Direction control register	GPIO_DIR	RW Configuration	GPIO direction	0X0000
0X000C	GPIO Up and down control register	GPIO_PULL_E_N	RW Configuration	GPIO Up and down	0xFFFF
0X0010	GPIO Multiplex select register	GPIO_AF_SEL	RW Configuration	GPIO Multiplexing function enable bit	0xFFFF
0X0014	GPIO Multiplex select register 1	GPIO_SF_S1	RW	GPIO Multiplex function selection bit high address bit	0X0000
0X0018	GPIO Multiplex select register 0	GPIO_AF_S0	RW	GPIO Multiplex function selection bit low address bit	0X0000
0X0020	GPIO Interrupt trigger mode configuration register	GPIO_IS	RW Configuration	GPIO Interrupt trigger	0X0000
0X0024	GPIO Interrupt edge trigger mode configuration register	GPIO_IBE	RW Configuration	GPIO Interrupt edge trigger mode	0X0000
0X0028	GPIO Interrupt upper and lower edge trigger configuration register	GPIO_IEV	RW Configuration	GPIO Trigger on the upper and lower edges of the interrupt or high and low Level trigger	0X0000
0X002C	GPIO Interrupt enable configuration register	GPIO_IE	RW Configuration	GPIO Interrupt enable	0X0000
0X0030	GPIO Bare Interrupt Status Register	GPIO_RIS	RO Inquire	GPIO Bare interrupt status (MASK before) 0X0000	
0X0034	GPIO Interrupt status register after masking	GPIO_MIS	RO Inquire	GPIO Interrupt status after masking (MASK Rear)	0X0000
0X0038	GPIO Interrupt Clear Control Register	GPIO_IC	WO control	GPIO Interrupt clear	0X0000

table 44 GPIOB Register list

Offset address	name	abbreviation	access	description	Reset value
0X0000	GPIO Data register	GPIO_DATA	RW Read and write GPIO Current data		0X0000_7304
0X0004	GPIO Data enable register	GPIO_DATA_E_N	RW Configuration	GPIO_DATA Enable bit	0X7FFF_FFFF
0X0008	GPIO Direction control register	GPIO_DIR	RW Configuration	GPIO direction	0X0000_0000
0X000C	GPIO Up and down control register	GPIO_PULL_E_N	RW Configuration	GPIO Up and down	0xFFFF_FFFF
0X0010	GPIO Multiplex select register	GPIO_AF_SEL	RW Configuration	GPIO Multiplexing function enable bit	0xFFFF_FFFF
0X0014	GPIO Multiplex select register 1	GPIO_SF_S1	RW	GPIO Multiplex function selection bit high address bit 0X0000_0000	
0X0018	GPIO Multiplex select register 0	GPIO_AF_S0	RW	GPIO Multiplex function selection bit low address bit 0X0000_0000	

0X0020	GPIO Interrupt trigger mode configuration register GPIO_IS		RW Configuration	GPIO Interrupt trigger	0X0000_0000
0X0024	GPIO Interrupt edge trigger mode configuration register	GPIOIBE	RW Configuration	GPIO Interrupt edge trigger mode 0X0000_0000	
0X0028	GPIO Interrupt upper and lower edge trigger configuration register	GPIOIEV	RW Configuration	GPIO Trigger on the upper and lower edges of the interrupt	0X0000_0000
0X002C	GPIO Interrupt enable configuration register	GPIOIE	RW Configuration	GPIO Interrupt enable	0X0000_0000
0X0030	GPIO Bare Interrupt Status Register	GPIORIS	RO Inquire	GPIO Bare interrupt status (MASK before)	0X0000_0000
0X0034	GPIO Interrupt status register after masking GPIO_MIS		RO Inquire	GPIO Interrupt status after masking (MASK Rear)	0X0000_0000
0X0038	GPIO Interrupt Clear Control Register	GPIOIC	WO control	GPIO Interrupt clear	0X0000_0000

9.4.2 GPIO Data register

table 45 GPIOA Data register

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO Current data, every BIT With the corresponding GPIO Line correspondence	16'h180b

table 46 GPIOB Data register

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO Current data, every BIT With the corresponding GPIO Line correspondence	32'h7304

9.4.3 GPIO Data enable register

table 47 GPIOA Data enable register

Bit access		Instructions	Reset value
[15: 0]	RW	<p>correspond GPIO_DATA of BIT Enable bit, only corresponding BIT for 1 Time, right GPIO_DATA Corresponding</p> <p>The operation is effective, every BIT With the corresponding GPIO Line corresponding, 1'bx :</p> <p>[x] = 0, GPIO_DATA[x] Not readable and writable</p> <p>[x] = 1, GPIO_DATA[x] Can read and write</p>	16'hffff

table 48 GPIOB Data enable register

Bit access		Instructions	Reset value
[31: 0]	RW	<p>correspond GPIO_DATA of BIT Enable bit, only corresponding BIT for 1 Time, right GPIO_DATA Corresponding</p> <p>The operation is effective, every BIT With the corresponding GPIO Line corresponding, 1'bx :</p>	32'h7fff_ffff

		[x] = 0, GPIO_DATA[x] Not readable and writable [x] = 1, GPIO_DATA[x] Can read and write	
--	--	---	--

9.4.4 GPIO Direction control register

table 49 GPIOA Direction control register

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO Direction control, every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] For input [x] = 1, GPIO[x] For output	16'h0

table 50 GPIOB Direction control register

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO Direction control, every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] For input [x] = 1, GPIO[x] For output	32'h0

9.4.5 GPIO Up and down control register

table 51 GPIOA Up and down control register

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO Up and down control, every BIT With the corresponding GPIO Line corresponding, 1'bx : Note: This register is active low [x] = 0, GPIO[x] There are up and down [x] = 1, GPIO[x] No pull-down One GPIO There is only one up and down state, IO Properties see IO Reuse table	16'hffff

table 52 GPIOB Up and down control register

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO Up and down control, every BIT With the corresponding GPIO Line corresponding, 1'bx : Note: This register is active low [x] = 0, GPIO[x] There are up and down [x] = 1, GPIO[x] No pull-down	32'hffff_ffff

		One GPIO There is only one up and down state, IO Properties see IO Reuse table	
--	--	--	--

9.4.6 GPIO Multiplex select register

table 53 GPIOA Multiplex select register

Bit access		Instructions	Reset value
[15: 0]	RW	<p>GPIO Multiplexing function enable bit, every BIT Corresponding to GPIO Whether the reuse function is turned on, 1'bx :</p> <p>[x] = 0, GPIO[x] Multiplexing function off</p> <p>[x] = 1, GPIO[x] Multiplexing function turned on</p> <p>[x] = 1 , The multiplexing function depends on GPIO_AF_S1 with GPIO_AF_S0 Two registers correspond BIT status.</p> <p>S1.[x] = 0 , S0.[x] = 0 , Multiplexing function 1(opt1)</p> <p>S1.[x] = 0 , S0.[x] = 1 , Multiplexing function 2(opt2)</p> <p>S1.[x] = 1 , S0.[x] = 0 , Multiplexing function 3(opt3)</p> <p>S1.[x] = 1 , S0.[x] = 1 , Multiplexing function 4(opt4)</p> <p>[x] = 0 Time, if GPIO_DIR[x] = 0 , And GPIO_PULL_EN[x] = 1 ,then GPIO Reuse as opt6 simulation IO Features</p> <p>IO For the multiplexing function, see the chip pin multiplexing relationship</p>	16'hffff

table 54 GPIOB Multiplex select register

Bit access		Instructions	Reset value
[31: 0]	RW	<p>GPIO Multiplexing function enable bit, every BIT Corresponding to GPIO Whether the reuse function is turned on, 1'bx :</p> <p>[x] = 0, GPIO[x] Multiplexing function off</p> <p>[x] = 1, GPIO[x] Multiplexing function turned on</p> <p>[x] = 1 , The multiplexing function depends on GPIO_AF_S1 with GPIO_AF_S0 Two registers correspond BIT status.</p> <p>S1.[x] = 0 , S0.[x] = 0 , Multiplexing function 1(opt1)</p> <p>S1.[x] = 0 , S0.[x] = 1 , Multiplexing function 2(opt2)</p> <p>S1.[x] = 1 , S0.[x] = 0 , Multiplexing function 3(opt3)</p> <p>S1.[x] = 1 , S0.[x] = 1 , Multiplexing function 4(opt4)</p> <p>[x] = 0 Time, if GPIO_DIR[x] = 0 , And GPIO_PULL_EN[x] = 1 ,then GPIO Reuse as opt6 simulation IO Features</p>	32'hffff_ffff

		IO For the multiplexing function, see the chip pin multiplexing relationship	
--	--	--	--

9.4.7 GPIO Multiplex select register 1

table 55 GPIOA Multiplex select register 1

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO Multiplex function selection bit high address bit, and GPIO_AF_S0 Determine the multiplexing function together IO For the multiplexing function, see the chip pin multiplexing relationship	16'h0

table 56 GPIOB Multiplex select register 1

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO Multiplex function selection bit high address bit, and GPIO_AF_S0 Determine the multiplexing function together IO For the multiplexing function, see the chip pin multiplexing relationship	32'h0

9.4.8 GPIO Multiplex select register 0

table 57 GPIOA Multiplex select register 0

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO Multiplex function selection bit low address bit, and GPIO_AF_S1 Determine the multiplexing function together How to configure see GPIO_AF_SEL Register description	16'h0

table 58 GPIOB Multiplex select register 0

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO Multiplex function selection bit low address bit, and GPIO_AF_S1 Determine the multiplexing function together How to configure see GPIO_AF_SEL Register description	32'h0

9.4.9 GPIO Interrupt trigger mode configuration register

table 59 GPIOA Interrupt trigger mode configuration register

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO The interrupt trigger mode of each BIT With the corresponding GPIO Line corresponding, 1'b x : [x] = 0, GPIO[x] Interrupt is edge triggered	16'h0

		[x] = 1, GPIO[x] Interrupt is level triggered	
--	--	---	--

table 60 GPIOB Interrupt trigger mode configuration register

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO The interrupt trigger mode of each BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] Interrupt is edge triggered [x] = 1, GPIO[x] Interrupt is level triggered	32'h0

9.4.10 GPIO Interrupt edge trigger mode configuration register

table 61 GPIOA Interrupt edge trigger mode configuration register

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO Interrupt edge trigger mode, every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] Edge-triggered interrupt mode GPIO_IEV Decide [x] = 1, GPIO[x] Both edges trigger interrupt	16'h0

table 62 GPIOB Interrupt edge trigger mode configuration register

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO Interrupt edge trigger mode, every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] Edge-triggered interrupt mode GPIO_IEV Decide [x] = 1, GPIO[x] Both edges trigger interrupt	32'h0

9.4.11 GPIO Interrupt upper and lower edge trigger configuration register

table 63 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, every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] Interrupt is triggered by low level or falling edge [x] = 1, GPIO[x] Interrupt is triggered by high level or rising edge	16'h0

table 64 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, every BIT With the corresponding GPIO Line corresponding, 32'h0	32'h0

		1'bx : [x] = 0, GPIO[x] Interrupt is triggered by low level or falling edge [x] = 1, GPIO[x] Interrupt is triggered by high level or rising edge	
--	--	--	--

9.4.12 GPIO Interrupt enable configuration register

table 65 GPIOA Interrupt enable configuration register

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO Interrupt enable control, every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] Interrupt disability [x] = 1, GPIO[x] Interrupt enable	16'h0

table 66 GPIOB Interrupt enable configuration register

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO Interrupt enable control, every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] Interrupt disability [x] = 1, GPIO[x] Interrupt enable	32'h0

9.4.13 GPIO Bare Interrupt Status Register

table 67 GPIOA Bare Interrupt Status Register

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO Bare interrupt status (MASK Before), every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] No interruption [x] = 1, GPIO[x] Interrupted	16'h0

table 68 GPIOB Bare Interrupt Status Register

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO Bare interrupt status (MASK Before), every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] No interruption [x] = 1, GPIO[x] Interrupted	32'h0

9.4.14 GPIO Interrupt status register after masking

table 69 GPIOA Interrupt status register after masking

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO Interrupt status after masking (MASK After), every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] No interrupt is generated (MASK Rear) [x] = 1, GPIO[x] Interrupt generation (MASK Rear)	16'h0

table 70 GPIOB Interrupt status register after masking

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO Interrupt status after masking (MASK After), every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, GPIO[x] No interrupt is generated (MASK Rear) [x] = 1, GPIO[x] Interrupt generation (MASK Rear)	32'h0

9.4.15 GPIO Interrupt Clear Control Register

table 71 GPIOA Interrupt Clear Control Register

Bit access		Instructions	Reset value
[15: 0]	RW	GPIO Interrupt clear control, every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, No action [x] = 1, Clear GPIO[x] Interrupted state	16'h0

table 72 GPIOB Interrupt Clear Control Register

Bit access		Instructions	Reset value
[31: 0]	RW	GPIO Interrupt clear control, every BIT With the corresponding GPIO Line corresponding, 1'bx : [x] = 0, No action [x] = 1, Clear GPIO[x] Interrupted state	32'h0

10 high speed SPI Device controller

10.1 Functional Overview

Compatible with universal SPI Physical layer protocol, by stipulating the data format to interact with the host, the host can access the device at high speed, the highest support

Working frequency is 50MHZ .

10.2 Main features

- Compatible with universal SPI protocol
- Selectable level interrupt signal
- Maximum support 50Mbps rate
- Simple frame format, full hardware analysis and DMA

10.3 Functional description

10.3.1 SPI Brief Introduction

SPI Work in master-slave mode, usually there is a master device and one or more slave devices, need at least 4 Root line, in fact 3 Root can also (single To transfer). 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 is 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 a predetermined enable signal (high Potential or low potential), the master chip is only valid for the operation of the slave chip. This allows multiple connections on the same bus SPI Equipment becomes possible.

In addition to the above 4 After the signal line, HSPI Also added an extra INT Line, when the slave device has data to upload, it generates a download

The interruption of the falling edge enables the active reporting of data.

SPI Communication is done through data exchange, data is transmitted bit by bit, by SCLK Provide clock pulses, SDI , SDO Zeki

Data transmission is completed by this pulse. Data output through SDO Line, the data changes on the rising or falling edge of the clock, on the next falling

The edge or rising edge is read. To complete a one-bit data transfer, the input also uses the same principle. Therefore, at least 8 Changes in the sub-clock signal (on Along the edge and the bottom edge), to complete 8 Bit data transmission.

SCLK The signal line is controlled by the master device, and the slave device cannot control the signal line. Based on SPI Among the devices, there is at least one master device.

10.3.2 SPI work process

Inside the chip HSPI Yes and wrapper The controller works together, wrapper Controller internal integration DMA ,by DMA achieve HSPI

internal FIFO Data exchange with the internal cache of the chip. This operation is realized by hardware, software does not need to care about the process of sending and receiving data,

Only need to configure the sending and receiving data list, and operation wrapper The corresponding register of the controller.

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

10.4 Register description

10.4.1 HSPI Register list of internal operation of the chip

table 73 HSPI Internal access register

Offset address	name	abbreviation	access	description	Reset value
0X0000	HSPI FIFO Clear register	CLEAR_FIFO	RW	Clear Tx with Rx FIFO Of the content, meanwhile Circuits that will synchronously reset the system clock domain	0X0000_0000
0X0004	HSPI Configuration register	SPI_CFG	RW	Configuration SPI The transmission mode and the size of the end Set	0X0000_0000
0X0008	HSPI Mode configuration register	MODE_CFG	RW	Configuration ahb master When accessing the bus burst length	0X0000_0000
0X000C	HSPI Interrupt Configuration Register	SPI_INT_CPU_MASK	RW	Configure whether 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 74 HSPI FIFO Clear register

Bit access		Instructions	Reset value
[31: 1]	RO	Keep	
[0]	RW	<p>Clear FIFOs , Clear Tx with Rx FIFO The content of the system will simultaneously reset the circuit of the system clock domain (this (Except the registers in the list)</p> <p>0 : Do not clear FIFO</p> <p>1 : Clear effective</p> <p>Set by software, cleared by hardware</p> <p>Note: If you want to reset the entire circuit, you need to use the asynchronous reset pin of this module: rst_n</p>	1'b0

10.4.1.2 HSPI Configuration register

table 75 HSPI Configuration register

Bit access		Instructions	Reset value
[31: 4]	RO	Keep	
[3]	RW	Bigendian, spi The interface supports the selection of the size of the data. 0 : Support small data transmission 1 : Support big-endian data transmission	1'b0
[2]	RW	spi_tx_always_drive 0 : spi The output is only valid when the chip select is valid, other times it is high impedance 1 : spi The 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 in IDLE Time polarity 0 : SCK IDLE When 0 1 : SCK IDLE When 1	1'b0

10.4.1.3 HSPI Mode configuration register

table 76 HSPI Mode configuration register

Bit access		Instructions	Reset value
[31: 1]	RO	Keep	
[0]	RW	Burst len , ahb master When accessing the bus burst length 0 : burst len for 1 word 1 : burst len for 4 word The recommended setting is 4 Word burst Transmission, so spi When the interface frequency is high, continuous flow can be guaranteed	1'b0

10.4.1.4 HSPI Interrupt Configuration Register

table 77 HSPI Interrupt Configuration Register

Bit access		Instructions	Reset value
[31: 2]	RO	Keep	
[1]	RW	IntEnRxOverrun , RxOverrun Interrupt enable 0 : Rx FIFO overflow Interrupt enable 1 : Rx FIFO overflow Interruption	1'b1

[0]	RW	IntEnTxUnderrun , TxUnderrun Interrupt enable 0 : Tx FIFO underflow interrupt enable 1 : Tx FIFO underflow interrupt is not enabled	1'b1
-----	----	---	------

10.4.1.5 HSPI Interrupt status register

table 78 HSPI Interrupt status register

Bit access		Instructions	Reset value
[31: 2]	RO	Keep	
[1]	RW	RxOverrun 0 : Rx FIFO overflow 1 : Rx FIFO overflow write 1 Clear	1'b0
[0]	RW	TxUnderrun 0 : Tx FIFO underflow 1 : Tx FIFO underflow write 1 Clear	1'b0

10.4.1.6 HSPI Data upload length register

table 79 HSPI Data upload length register

Bit access		Instructions	Reset value
[31:16]	RO	Keep	
[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 the word. If the upload length is less than the whole word, round up.	16'h0

10.4.2 Host side access HSPI Controller register list

The host side is fixed SPI Command format access SPI Interface register. The command length is fixed at one byte, and the data length is fixed at two bytes.

table 80 HSPI Interface configuration register (master access)

Offset address	name	abbreviation	access	description	Reset value
0X02	Get data length register	RX_DAT_LEN	RO When uploading data, spi Host used to get slave device Data length		0X0000
0X03	Send data flag register	TX_BUFF_AVAIL	RO When sending data from master to slave, it is used to judge whether it is possible to download data or commands		0X0000

0X04 Keep		RSV	RO		
0X05 Interrupt Configuration Register		SPI_INT_HOST_MASK RW Whether to block interrupts			0X0000
0X06 Interrupt status register		SPI_INT_HOST_STTS	RO Interrupt	status register, spi The host queries this bit Identify whether there is data upload	0X0000
0X07 Keep		RSV	RO		
0X00 Data port 0		DAT_PORT0	RW	Spi The host sends data to the slave device through this register port, and the previous data frame is sent to use this port	
0X10 Data port 1		DAT_PORT1	RW	Spi The host sends data to the slave device through this register port, and the last data frame is sent to use this port	
0X01 Command port 0		DN_CMD_PORT0	WO	Spi The host sends command data to the slave device through this register, and the previous command data is used to use this port	
0X11 Command port 1		DN_CMD_PORT1	WO	Spi The host sends command data to the slave device through this register, and the last frame of command data is used to use this port	

10.4.2.1 HSPI Get data length register

table 81 HSPI Get data length register

Bit	access	Instructions	Reset value
[15: 0]	RO	<p>spi Host read-only register, mainly used to learn from when uploading data device How much data is read</p> <p>However, in this module, the upload length is an integer multiple of the word. If the upload length value is not a whole word, the host will round up when reading, that is, the redundant bytes of the multi-read part</p>	16'h0

10.4.2.2 HSPI Send data flag register

table 82 HSPI Send data flag register

Bit	access	Instructions	Reset value
[15: 2]	RO	Keep	
[1]	RO	<p>tx_cmdbuf_avail</p> <p>Flag sending cmd of buff Whether available, if available, the host can deliver cmd .</p> <p>0 :send buff unavailable</p> <p>1 :send buff Available</p>	1'b0
[0]	RO	<p>tx_buff_avail</p> <p>Flag sending buff Whether available, if available, the host can deliver data.</p> <p>0 :send buff unavailable</p>	1'b0

		1 : send buff Available	
--	--	-------------------------	--

10.4.2.3 HSPI Interrupt Configuration Register

table 83 HSPI Interrupt Configuration Register

Bit	access	Instructions	Reset value
[15: 1]	RO	Keep	
[0]	RO	IntMaskup_dat_cmd_rdy Interrupt mask 0 : The interrupt is not masked and an interrupt 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 84 HSPI Interrupt status register

Bit	access	Instructions	Reset value
[15: 1]	RO	Keep	
[0]	RO	up_dat_cmd_rdy to SPI Status register generated by the host 0 : The data or command is not ready 1 : The data or command is ready Readable	1'b0

10.4.2.5 HSPI Data port 0

table 85 HSPI Data port 0

Bit	access	Instructions	Reset value
	RW	SPI The host uses this register port and device For data transmission, write data to this register to send data, and read data from this register to upload data. If the frame being transmitted needs multiple transmissions to complete, the last transmission uses the register port DAT_PORT1 , Other uses DAT_PORT0 .	

10.4.2.6 HSPI Data port 1

table 86 HSPI Data port 1

Bit	access	Instructions	Reset value
	RW	SPI The host uses this register port and device For data transmission, write data to this register to send data, and read data from this register to upload data. If the frame being transmitted needs multiple transmissions to complete	

		Successful, the last transfer uses the register port DAT_PORT1 , Other uses DAT_PORT0 .	
--	--	---	--

10.4.2.7 HSPI Command port 0

table 87 HSPI Command port 0

Bit	access	Instructions	Reset value
	RW	<p>SPI The host uses this register port and device Perform command interaction, write a number to this register, and then you can send a command make. If the command being transmitted needs to be completed by multiple transmissions, the last transmission uses the register port DN_CMD_PORT1 , Other uses DN_CMD_PORT0 .</p> <p>Note: This window is only used to issue commands negotiated by the driver and firmware.</p>	

10.4.2.8 HSPI Command port 1

table 88 HSPI Command port 1

Bit	access	Instructions	Reset value
	RW	<p>SPI The host uses this register port and device Perform command interaction, write a number to this register, and then you can send a command make. If the command being transmitted needs to be completed by multiple transmissions, the last transmission uses the register port DN_CMD_PORT1 , Other uses DN_CMD_PORT0 .</p> <p>Note: This window is only used to issue commands negotiated by the driver and firmware.</p>	

10.4.3 high speed SPI Device controller interface timing

Main description SPI Read and write timing, and master SPI How to HSPI Perform data interaction.

10.4.3.1 Data Format

The data format is divided into two parts, the command field and the data field, as shown below. The fixed length of the command field is 8bit , The length of the data field depends on the access obje

Different, different lengths, see below for details.

The highest command domain bit For reading and writing flags, the rest 7bit Is the address.

- 0 From behind 7bit Read data at address
- 1 Means backward 7bit Address write data

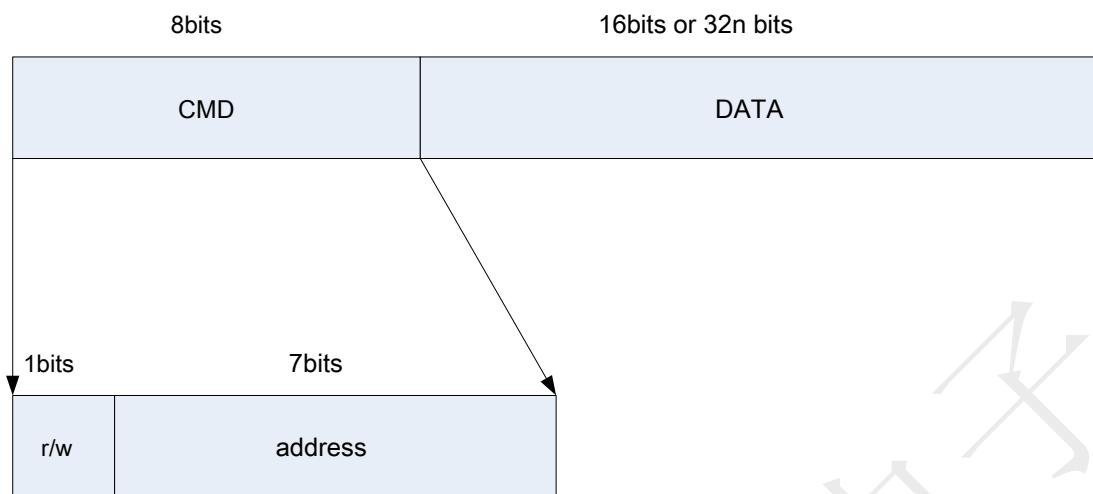


Fig 6 Host computer SPI Send and receive data format

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

Through port (data port 0 , Data port 1 , Command port 0 And command port 1) Transmission data, the data field length is 32bit Integer

Times

The following figure is the timing diagram of the read and write interface configuration registers. The default configuration of the slave device is little-endian mode.

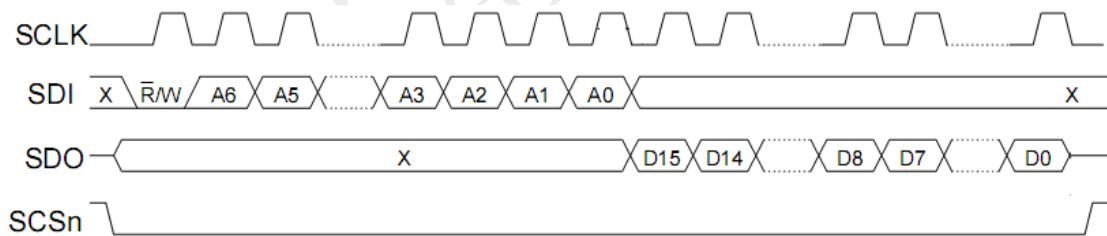


Fig 7 HSPI Register read operation (big endian mode)

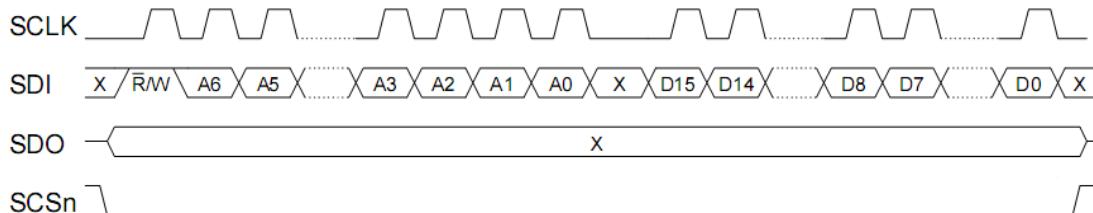


Fig 8 HSPI Register write operation (big endian mode)

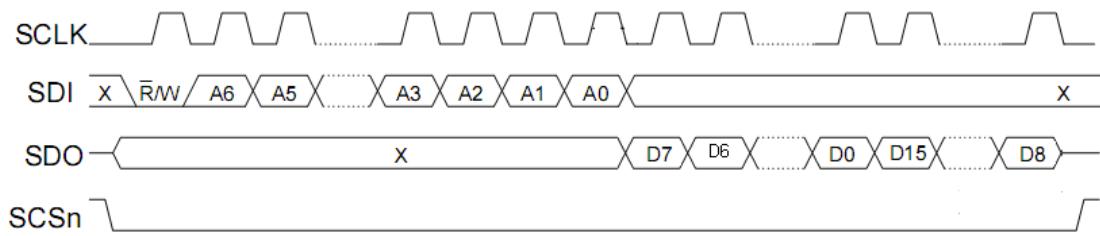


Fig 9 Register read operation (little-endian mode)

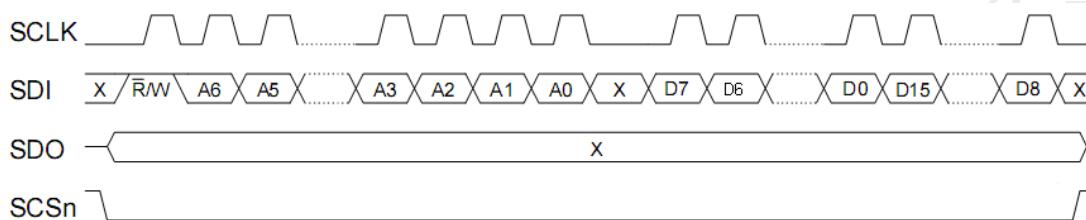


Fig 10 Register write operation (little-endian mode)

The following figure is the timing diagram of reading and writing data, the length of the data field is 32bit It is an integer multiple of, and the figure shows that only one word is transmitted.

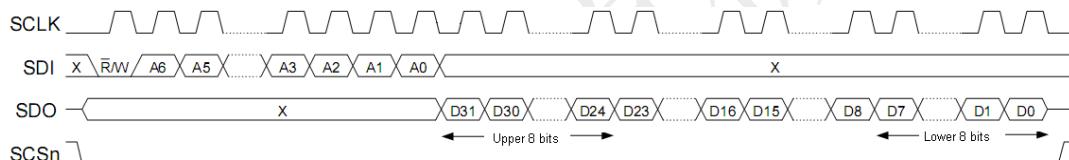


Fig 11 Port read operation (big endian mode)

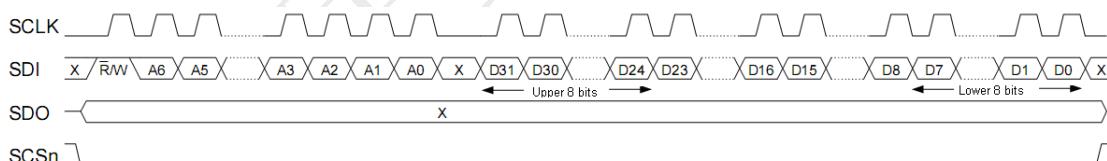


Fig 12 Port write operation (big endian mode)

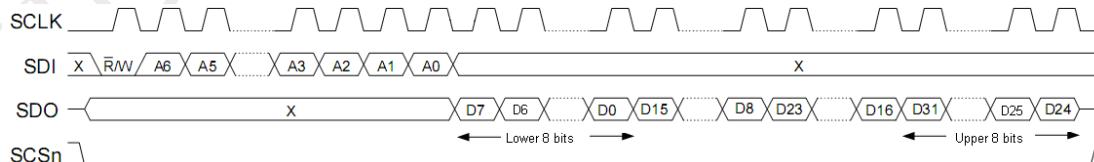


Fig 13 Port read operation (little-endian mode)

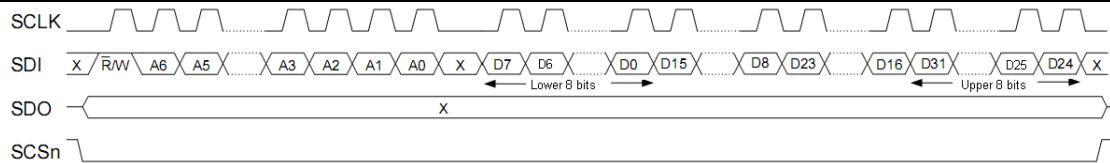


Fig 14 Port write operation (little-endian mode)

Note: There can be no waiting time between the command and the data, that is, after the command field is transmitted, the data can be transmitted immediately, when there is no need for extra idle clock or free time. A time delay is also possible, but no idle clock can appear.

10.4.3.2 Timing

This module supports half-duplex. The timing that can be supported is divided into different phases according to the clock phase and sampling point. 4 Species. The following timing is just given the relationship between the four species.

Phase and sampling relationship. It should be noted that the chip supports (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 can be transmitted immediately, when there is no need for extra idle clock or free time. A time delay is also possible, but no idle clock can appear.

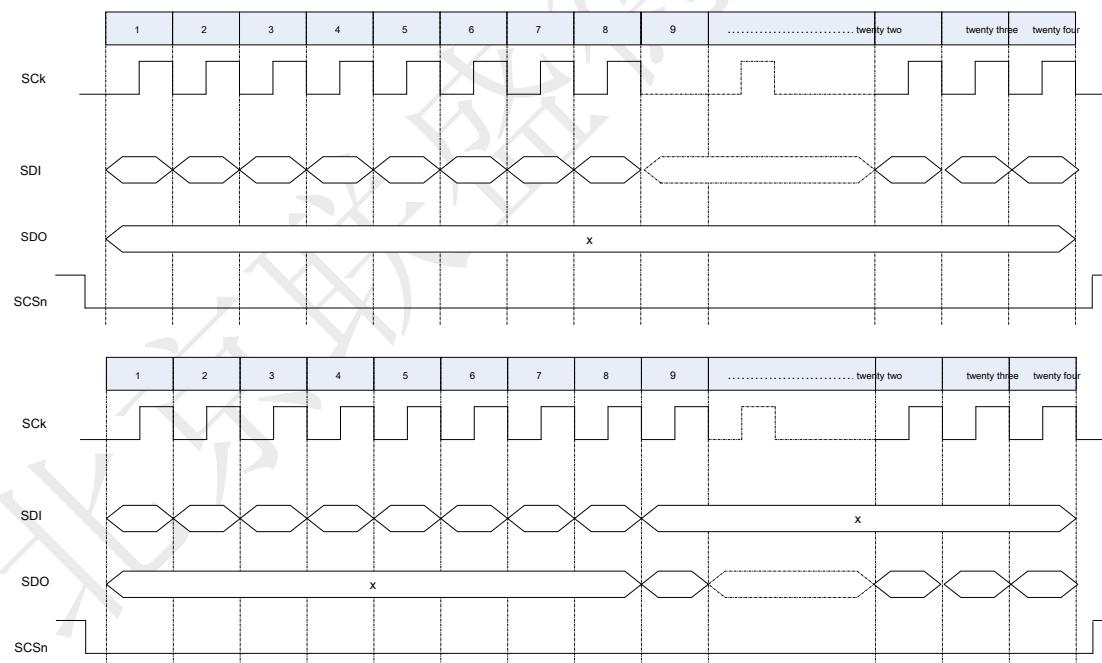


Fig 15 CPOL=0, CPHA=0

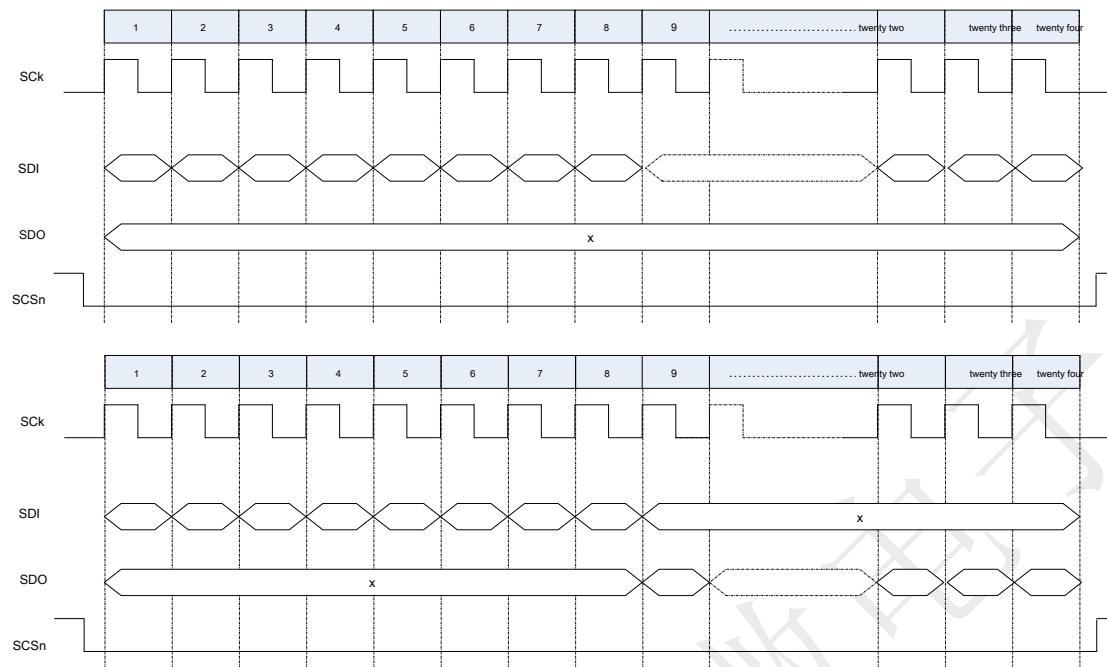


Fig 16 CPOL=0, CPHA=1

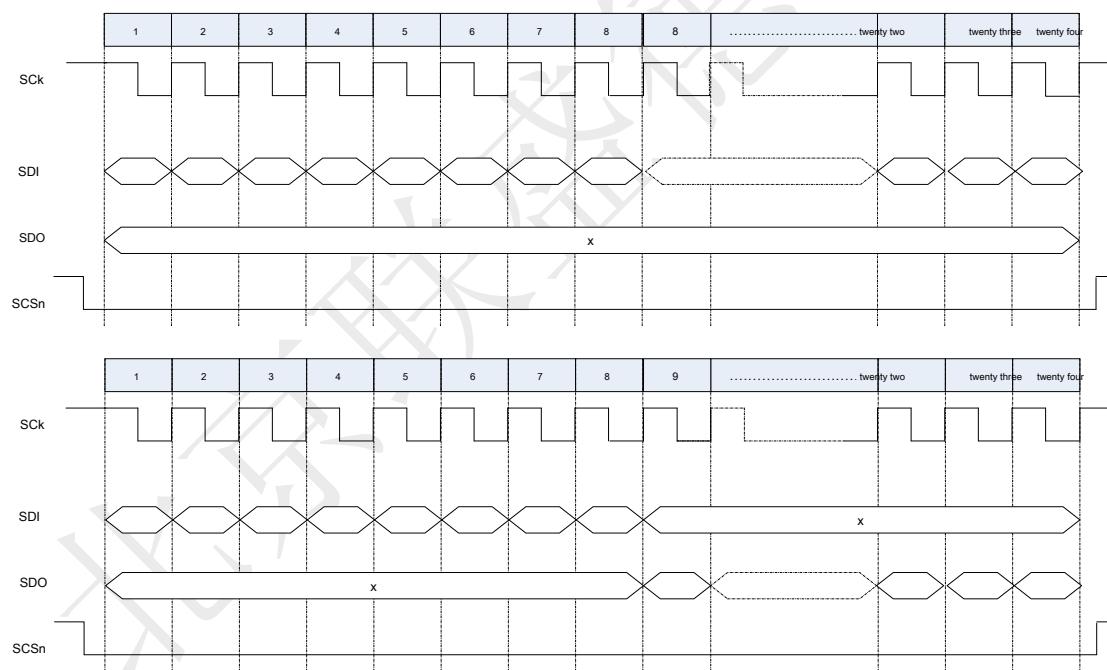


Fig 17 CPOL=1, CPHA=0

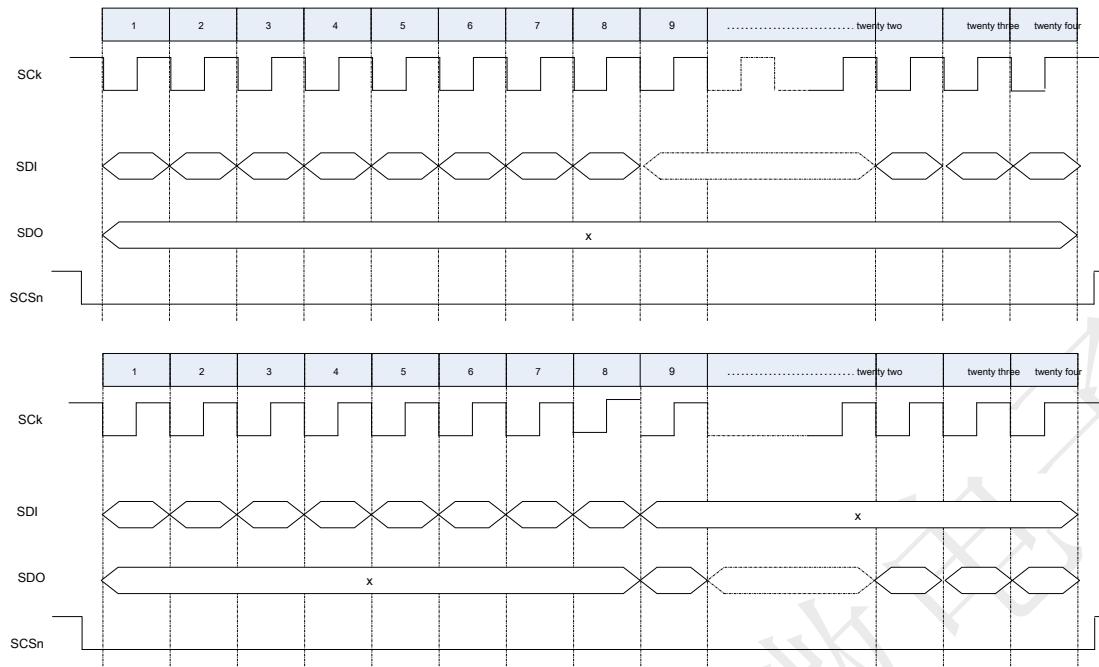


Fig 18 CPOL=1,CPHA=1

10.4.3.3 Interrupt

The interrupt signal is sent from the slave device to the master device, through SPI_INT Pin trigger, active low.

spi_int Main notice spi The host has data or commands to upload, spi The interface registers that the host cares about when processing interrupts are:

- SPI_INT_HOST_MASK
- SPI_INT_HOST_STTS
- RX_DAT_LEN

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

A new interrupt will be generated. The following figure is one way to handle interrupts.

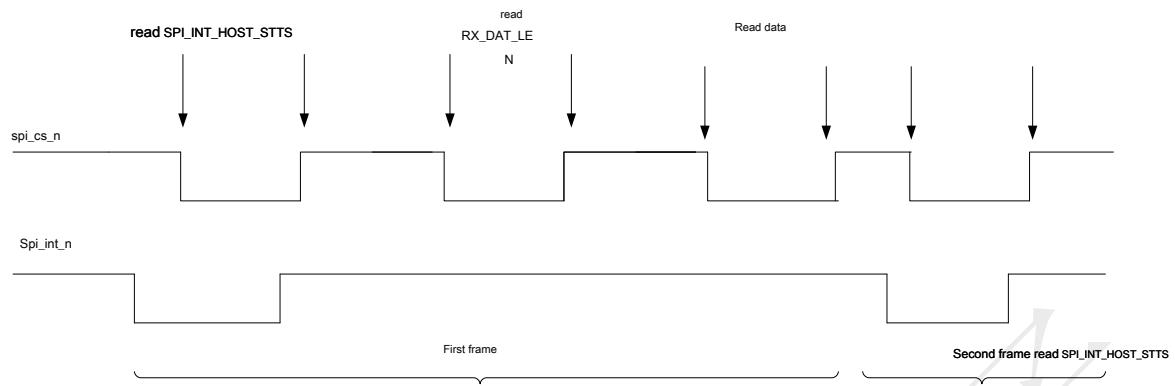


Fig 19 the Lord SPI Handle interruption process

10.4.3.4 the Lord SPI Send and receive data workflow

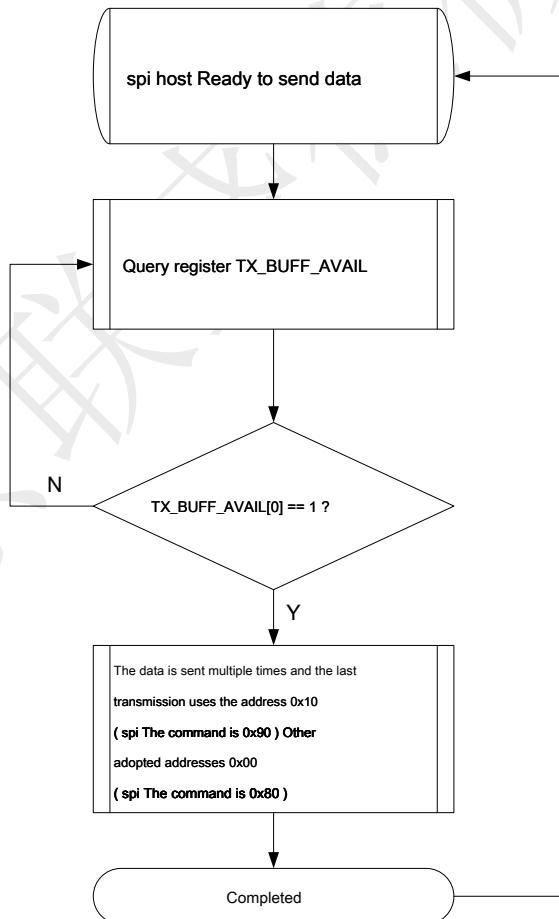


Fig 20 Downstream data flow chart

Note: The length of the data sent must be in word units, if it is not a whole word, fill in 0 Make up.

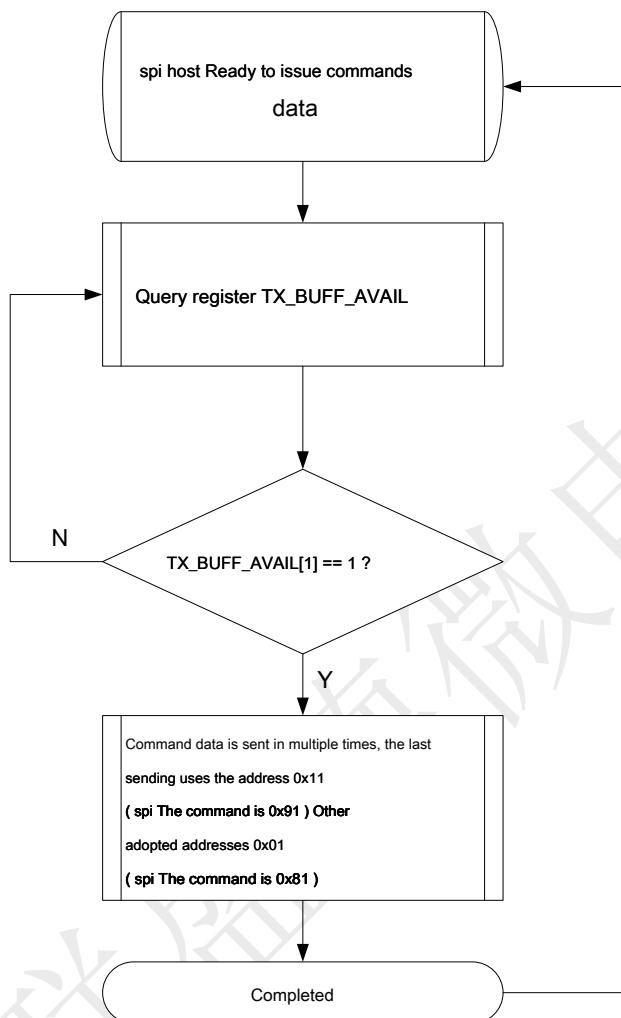


Fig twenty one Downstream command flow chart

Note: The length of the command issued must be in word units, if it is not a whole word, fill in 0 Make up.

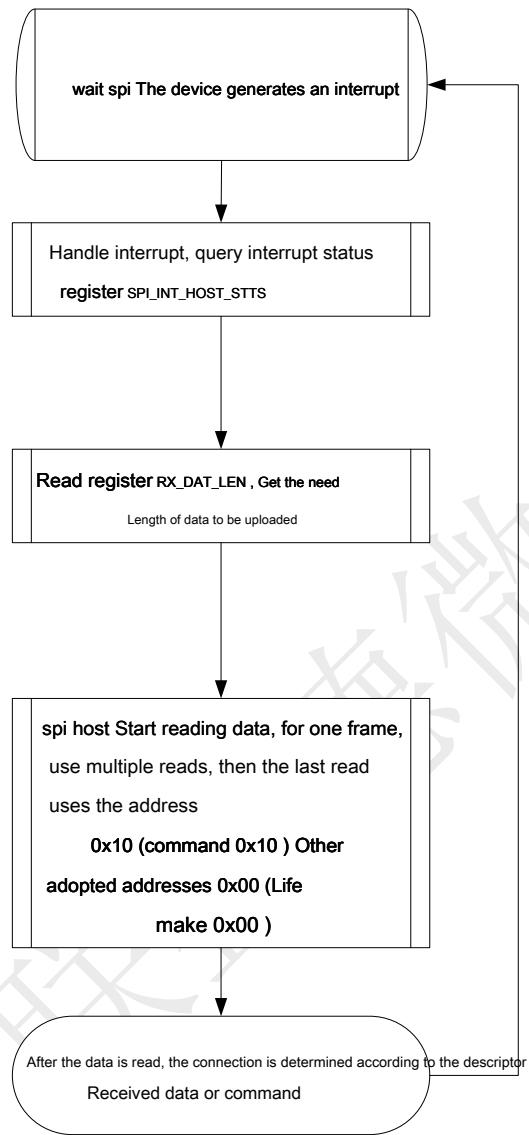


Fig twenty two Upstream data (command) flow chart

The flow of upstream data and upstream commands are the same.

It should be noted here that the length of the upstream data must be in units of words. If the effective length is not a whole word, the extra data at the end can be thrown away

Off.

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 a channel as required

Use either or both. The maximum data length of a command channel for one interaction is 256 Bytes, the maximum data length of one data channel interaction is

1500 byte. 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 Functional Overview

W600 Integrated SDIO The device-side interface, as a slave device, completes the interaction with the host data. Internally integrated 1024byte Asynchronous FIFO To complete the data interaction between the host and the chip.

11.2 Main features

- compatible SDIO Card specifications 2.0
- Support host rate 0~50MHz
- Maximum support 1024 Byte Block
- stand by 1 Bit SD with 4 Bit SD mode

11.3 Functional description

11.3.1 SDIO bus

SDIO Bus and USB The bus is similar, SDIO The bus also has two ends, one end is the host end, the other end is the device end, using HOSTDEVICE This design is to simplify DEVICE Design, all communications are made by HOST The end issued a command to start. in DEVICE As long as the end can be resolved HOST Command HOST Communicated, SDIO of HOST Can connect multiple DEVICE .

in SDIO In the bus definition, DAT1 The signal line is multiplexed as an interrupt line. in SDIO of 1BIT Under mode DAT0 Used to transfer data, DAT1 Used as an interrupt line. in SDIO of 4BIT Under mode DAT0-DAT3 Used to transfer data, where DAT1 Reuse as an interrupt line.

11.3.2 SDIO command

SDIO On the bus HOST Request DEVICE The terminal responds to the request, which includes data information in the request and the response:

- **Command:** The command used to start the transmission is HOST Send to DEVICE End, where the command is through CMD letter No. line transmission;
- **Response:** The response is DEVICE Returned, as Command Response. Also through CMD By wire
- **Data:** The data is transmitted in both directions. Can be set to 1 Line mode can also be set to 4 Line mode. The data is passed DAT0-DAT3 Signal line transmission.

SDIO Every operation of HOST in CMD Initiate one online CMD , For some CMD , DEVICE Need to return Response ,

Some are not required.

For read commands, first HOST Would DEVICE Send a command, followed by DEVICE Will return a handshake signal, at this time, when HOST

After receiving the response to the handshake signal, the data will be placed 4 Bit data line, it will follow along with the data transmission CRC Check code. when

After the entire read is transmitted, HOST Will send a command again to notify DEVICE After the operation, DEVICE At the same time will return a ring should.

For write commands, first HOST Would DEVICE Send a command, followed by DEVICE Will return a handshake signal, at this time, when HOST

After receiving the response to the handshake signal, the data will be placed 4 Bit data line, it will follow along with the data transmission CRC Check code. when

After the entire write is completed, HOST Will send a command again to notify DEVICE After the operation, DEVICE At the same time will return a ring should.

11.3.3 SDIO Internal storage

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

CIA The registers in I/O Port function, interrupt generation and port work information can be read and written through 0 Correct CIA Defined

Related operations. CIA Contains CCCR , FBR with CIS There are three aspects of information. among them CCCR Defined SDIO card

The common control register, the host side through the operation CCCR Yes SDIO Check the card and operate the port, CCCR the address of for 0X00-0XFF . FBR Defines the supported port functions 1 To port function 7 Operation, including the requirements and functions of each port,

Power control, etc., FBR 'S address is 0Xn00-0Xnff (among them n is the function port number). CIS Defined some information structure of the card, ground

Address is 0X1000-0X17FFF , CIS Public CIS And each of the function ports CIS Of which public CIS The initial address of CCCR

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

CIA The storage map is as follows.

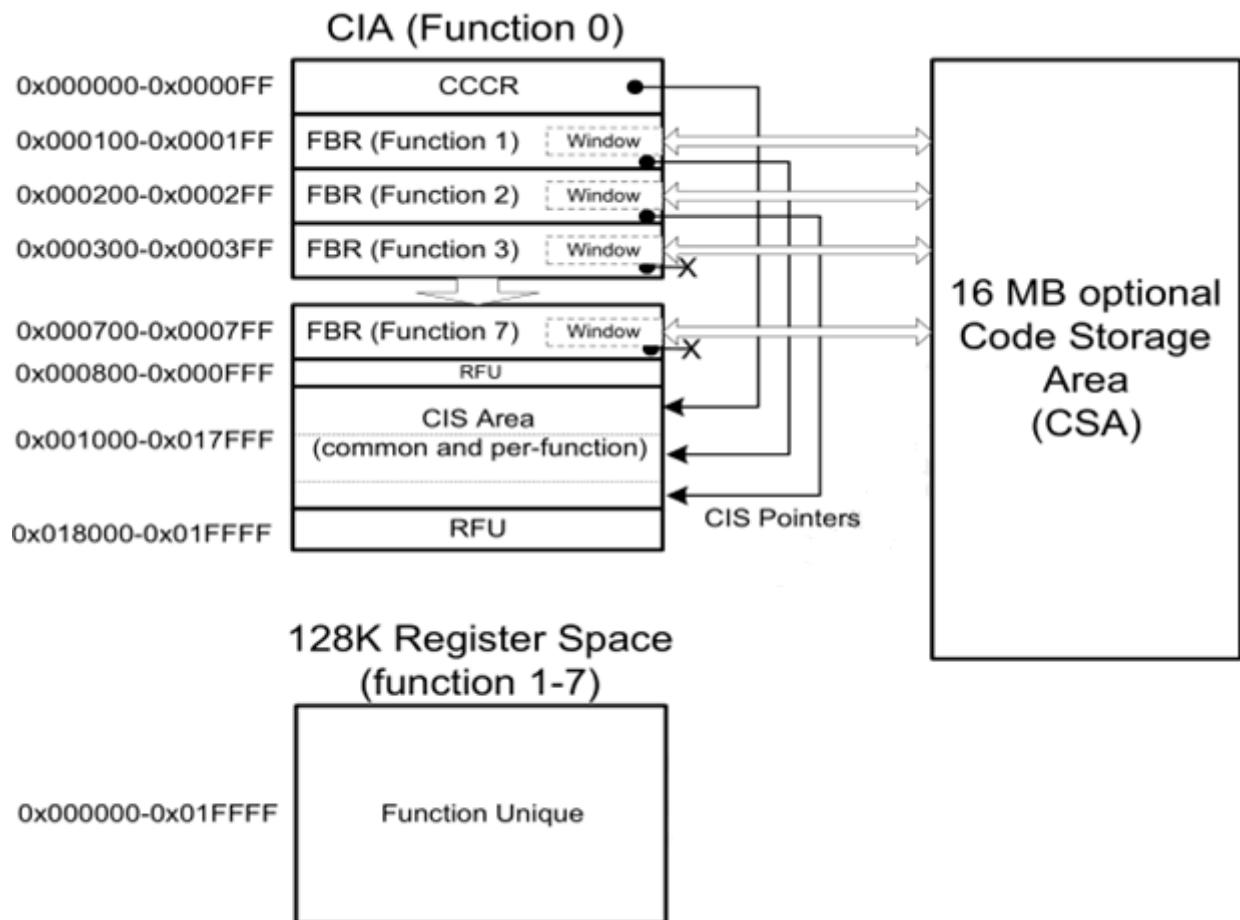


Fig 23 SDIO Internal storage mapping

CIA Refer to the following for the description of each register. To understand CIA , See SDIO Protocol specification.

11.4 Register description

11.4.1 Register list

11.4.2 SDIO Fn0 register

Fn0 The register is SDIO The address range of the register specified in the agreement is: 0x00000-0x1FFFF Of 128K . The starting address is 0x00000 .

Fn0 The register consists of SDIO Host pass CMD52 Command to access, the offset address is the access address, 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
0x0D	Function Select	DF	RFU	RFU	RFU	FS3	FS2	FS1	FS0
0x0E	Exec Flags	EX7	EX6	EX5	EX4	EX3	EX2	EX1	EXM
0x0F	Ready Flags	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RFM
0x10-0x11	FN0 Block Size	I/O block size for Function 0							
0x12	Power Control	Reserved for Future Use (RFU)						EMPC	SMPC
0x13	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							

Fig 24 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)								
0x800-0xFFFF	Reserved for Future Use								

Fig 25 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							

Fig 26 CIS Storage space structure

11.4.2.1 SDIO CCCR Register and FBR1 Register list
table 89 SDIO CCCR Register and FBR1 Register list

Offset address	name	Abbreviated access		description	Reset value
0X00	CCCR/SDI Revision	SDIOx	RO	[3:0], Supportive CCCR/FBR format 4'h0 : CCCR/FBR Version 1.00 4'h1 : CCCR/FBR Version 1.10 4'h2 : CCCR/FBR Version 1.20 4'h3-4'hF : Rsv by CIA Register[3:0] To represent	4'h2
				[7:4], Supportive SDIO Protocol version 4'h0 : SDIO Version 1.00 4'h1 : SDIO Version 1.10 4'h2 : SDIO Version 1.20 (unreleased) 4'h3 : SDIO Version 2.00 4'h4-4'hF : Rsv by CIA Register[7:4] To represent	4'h3
0X01	SD specification Revision	SDx	RO	[3:0] , For support SD Protocol version 4'h0 : SD Physical Version 1.01 (March 2000) 4'h1 : SD Physical Version 1.10 (October 2004) 4'h2 : SD Physical Version 2.00 (May 2006) 4'h3-4'hF : Rsv by CIA Register[11:8] To represent	4'h2
				RO RFU	4'h0
0X02	I/O Enable	IOEx	RO	RFU	1'b0
			RW	[7:1] , Function Enable, bit1-bit7 Corresponding to 7 Pcf function ,correspond SD Host makes corresponding bit for 1 , Then the corresponding function , Otherwise function Do not Can work. Note: CIS0 , CIS1 as well as CSA Put on Fn1 In, at this time even Fn1 Not enabled, SD The 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 Total 7bit , Corresponding to 7 Pcf function 'S status, if corresponding bit for 1 , It means that function Can work. In this design, HC8051 By configuration program register of function ready Bit	7'b0

				for 1 , Make this register bit1=1 , Thereby marking Fn1 Can work normally. Note: yes CIS0 , CIS1 , CSA The read and write operations are independent of IOR1 , That is, even if IOR1=0 , You can also access the contents of these three storage spaces.	
0X04	Int Enable	IENM RW		[0], Interrupt enable signal 0 : Interrupts from Card cannot be given SD Host 1 :any function Can be sent to the host	1'b0
		IENx	RW	[7:1],functionx The interrupt is enabled. IEN1=0, Then come Fn1 Will not be sent to the host. IEN1=1 , Then allow Fn1 Interrupt to the host	7'b0
0X05	Int Pending		RO	[0], RFU	1'b0
		INTx	RO	[7:1],functionx Interruption hangs. INT1=0 ,No Fn1 Interruption is suspended INT1=1 , Fn1 There is an interrupt being suspended. Note: If IEN1 with IENM Not for 1 , The host will not receive a suspend interrupt	7'b0
0X06	I/O Abort	ASx	WO	[2:0], cancel IO Read or write to release the bus. cancel Fn1 Operation should be used CMD52 Write 3'b1 . The order is in SPI under not support.	3'b0
		RES	WO	[3], Soft reset signal 1: Reset SD Circuits in the clock domain, the bit Automatically cleared after setting, no special The door is cleared. This reset signal will not affect the current card protocol selection (SD or SPI Mode) without affecting CD Disable . Can only be used CMD52 operating.	1'b0
			RO	RFU	4'b0
0X07 Bus	Interface control	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], Allow continuous SPI Interrupt. in case SCSI for 1 , Then this register is used to allow SDIO stuck at SPI Mode, Give an interrupt at any time, no need to care at this time CS The status of the line.	1'b0
		SCSI	RO	[6] , Support continuous SPI Interrupt. If 1 , Indicating SDIO Card support SPI In the mode, an interrupt is given at any time, Without concern CS status. when program reg[2] for 1 , This register is set.	1'b1

		CD Disable	RW	[7] , Connect or disconnect CD/DAT[3](pin1) Up 10-90K Pull-up resistor. 0 : Connect the pull-up resistor 1 : Disconnect the pull-up resistor After power-on, this register is cleared, that is, a pull-up resistor is connected. The status of this register is not Will be SD The reset command in the protocol affects.	1'b0
0X08 Card	Capability	SDC	RO	[0], Supports execution during data transfer CMD52 command. in SPI Not supported in mode This register. when program_reg[3] for 1 , The register is 1	1'b1
		SMB	RO	[1], Express SDIO Card support CMD53 required Block Transmission mode. when program_reg[4] for 1 , The register is 1	1'b1
		SRW	RO	[2], Express SDIO Support read wait- Read Wait Control (RWC)operating. when program_reg[5] for 1 , The register is 1	1'b1
		SBS	RO	[3] , Express SDIO The card supports suspend/resume. If 0 , It does not support (0x0C-0x0F) register If 1 ,apart from Fn0 ,all function Will be based on SD The host asks to hang or Recovery when program_reg[6] for 1 , The register is 1	1'b1
		S4MI	RO	[4], Express SDIO Card support at 4bit many Block In data transmission mode, Health interruption. 0 : Not supported Block Interruption occurs between transmissions, in this case, as long as IENx=1 , SDIO Can still initiate an interrupt to the host in other interrupt cycles 1 : Supported in Block Interruption between transmissions when program_reg[7] for 1 , The register is 1	1'b1
		E4MI	RW	[5], The interrupt is enabled. allow 4bit many Block Mode, in two block Generated in the middle of data transmission to the host Interrupt. 0 : Not allowed 1 :allow Power-on reset or reset command will clear this register 0	1'b0
		LSC	RO	[6], 0 : Means SDIO The card is a full-speed device 1 : Means SDIO The card is a low-speed device when program_reg[8] for 1 , The register is 1	1'b0

		4BLS	RO	[7], 0 : Means SDIO Low speed mode device or not supported 4bit mode 1 : Means SDIO It is a low-speed mode device and supports 4bit mode when program_reg[9] for 1 , The register is 1	1'b1
0X09- 0X0B	Common CIS pointer		RO	[23:0], direction SDIO Card sharing CIS (CIS0) Of the starting address pointer. CIS0 package Contains information about the entire card. Its access space is: 0x001000-0x017FFF . Finger The needle is stored in little-endian format (LSB).	24'h001 000
0X0C Bus	Suspend	BS	RO	[0] , Bus status. 0 : Currently selected function No data bus used 1 : Currently selected function (use FSx Or use IO In the order function number) Is executing the command that will transfer data on the data line This register is used by the host to decide which function Data currently in use line. in case SDIO The card does not support the suspend recovery function, the register is 0. Any visit CIA Operation cannot be suspended, the register has been 1 ,even if BR send The register is 1 . SPI In mode, read-only, and 0.	1'b0
		BR	RW	RW Bus release Request/Status. This register is used to request the selected function (use FSx or CMD53 total function number Checked) Release the data bus and suspend related operating. If the host sets this register to 1 , The selected function Will temporarily stop According to the data transmission on the line, and suspend the current data operation command. BR Register Hold for 1 Until the release process is complete. once function Is suspended, the device clears zero BS , BR To notify the host. The host can read BR To monitor pending requests Execution status, if BR for 1 , The pending request is still being executed. Host can take the initiative to BR write 0 To cancel the pending request being executed. SPI In mode, read-only, and 0.	4'h0
			RO	[7:2], RFU	6'b0
0X0D Function	Select	FSx	RW	[3:0] , Used to select during suspend/resume operations function[0-7] . Two ways to write FSx : Correct CCCR carried out IO Write operation Newly launched IO The command will cause FSx Set to function number . in case function Currently suspended, to FSx Write this function of number , Then	4'b0

				<p>read FSx When function Data transfer operation. The returned value will be Currently selected function of number .</p> <p>Note: When reading FSx Time, if BS=0 ,then FSx 'S value is undefined.</p> <p>4'b0000 : Transaction of function 0 (CIA)</p> <p>4'b0001-4'b0111 : Transaction to functions 1-7</p> <p>4'b1000 : Transaction of memory in combo card</p> <p>4'b1001-4'b1111 : Not defined, reserved for future use</p>	
		RO	[3:1], RFU	3'b000	
	DF	RO	<p>[7] To restore the data flag. to FSx Write function number , Will reselect function Data transmission. Once the data transfer is restored, DF The register will indicate whether There is more data to be transmitted.</p> <p>0 :in function After being restored, there is no more data to transfer.</p> <p>1 :in function After being restored, there is more data to be transferred.</p> <p>DF Used in 4bit In mode, the interrupt cycle is controlled. If 1 ,in function Restore After the reset, there is more data to be transferred, in this case, the interrupt cycle is canceled. Such as If 0 , function After the data transfer is completed (in busy Case), here In this case, there is no data transmission after recovery, so the host can function After recovery, the start of the interruption cycle is monitored.</p>	1'b0	
0X0E Exec Flags	EXx	RO	<p>[7:0] , Execute the flag. The host passes these bit Decide all function[7-1]</p> <p>The status of the executed command. These registers can inform the host of a certain function Executing Command, so you cannot function Issue new commands.</p> <p>SPI In mode, read-only, and 0.</p>	8'h00	
0XF Ready Flags RFx		RO	<p>[7:0] , Read the flag. The host can know the right function[7-1]</p> <p>Read and write busy status. If one function A write transaction is being executed, corresponding to RFx bit Clearing marks function Busy, not ready to receive more data. in case One function Performing a read operation, corresponding to RFx bit Cleared, then marked Reading data is invalid, if it is 1 , It indicates that the read data can be transmitted.</p> <p>SPI Invalid, read-only, and 0</p>	8'h00	
0X10-0X11	FN0 Block Size	RW	[15:0], Fn0 corresponding Block When transmitting, Block size size. maximum 2048Byte , Minimal 1Byte . The storage method is small segment format (LSB)	16'h00	
0X12	Power Control	SMPC RO	<p>[0], Support host power consumption control.</p> <p>0 : SDIO The total current is less than 200mA , Even if all function Are valid (IOEx=1).</p> <p>EMPC , SPS , EPS All for 0 .</p>	1'b1	

				1 : SDIO The total current can exceed 200mA . EMPC , SPS , EPS effective.	
		EMPC RO		<p>[1], Host power consumption control is enabled.</p> <p>0 : SDIO Card total current is less than 200mA . SDIO Card automatic switching function(s) To Low current mode or not allowed function Enable, and ignore EPS 'S value, So that the card current is less than or equal to 200mA .</p> <p>1 : SDIO Card total current can exceed 200mA , And SPS with EPS effective. Host Use according to your ability to provide current FBR middle SPS , EPS as well as IOEx Make Capable of more current function .</p>	1'b0
		RO		[7:2], RFU	
0X13	High-Speed SHS	RO		<p>[0] , Indicating SDIO Card supports high speed</p> <p>0 : Does not support high speed</p> <p>1 : Support high speed</p>	1'b1
	EHS	RW		<p>[1] , High speed enable</p> <p>0 : SDIO The card works at the default speed, the highest frequency 25MHZ</p> <p>1 : SDIO The card can work in high-speed mode, the highest frequency 50MHZ</p>	1'b0
		RO		[7-2] , RFU	
0X14- 0XEF	RFU	RO		Reserved for Future Use (RFU)	8'b0
0XF0- 0xFF	Reserved for Vendors	RO		Area Reserved for Vendor Unique Registers	8'b0
0X100	I/O Device Interface Code	RO		<p>[3:0], Sign Fn1 For what kind of equipment.</p> <p>Pass register CIA[15:12] Programmable.</p> <p>4'h0 No SDIO standard interface supported by this function</p> <p>4'h1 This function supports the SDIO Standard UART</p> <p>4'h2 This function supports the SDIO Type-A for Bluetooth standard interface</p> <p>4'h3 This function supports the SDIO Type-B for Bluetooth standard interface</p> <p>4'h4 This function supports the SDIO GPS standard interface</p> <p>4'h5 This function supports the SDIO Camera standard interface</p> <p>4'h6 This function supports the SDIO PHS standard interface</p> <p>4'h7 This function supports the SDIO WLAN interface</p> <p>4'h8 This function supports the Embedded SDIO-ATA standard</p>	4'h7

			interface	
	RFU	RO	[5:4], RFU	2'b00
	Function supports CSA	RO	[6], 0 : Fn1 not support CSA 1 : Fn1 Support and have CSA Register CIA[16] program	1'b0
	Function CSA enable	RW	[7], 0: Disallow access CSA 1 : Allow access CSA	1'b0
0X101	Extended standard I/O device type code	RO	[7:0] , I/O Device Interface Code Extension of Pass through register CIA[24 : 17] program	8'b0
0X102	SPS	RO	[0], Mark Fn1 Is there a power consumption option 0 : No power consumption option 1 : There are two power consumption options, you can pass EPS select Pass through register CIA[25] program	1'b0
	EPS	RW	[1], Power consumption selection 0 : Fn1 Working in high current mode 1 : Fn1 Working in low current mode	1'b0
		RO	[7 : 2] , RFU	6'b0
0X103- 0X108		RO	RFU	0
0X109- 0X10B	Address pointer to function CIS1	RO	[16:0], Fn1 of CIS Address pointer, ie CIS1 , Indicating host access Fn1 of CIS starting address. The storage method is LSB Short format.	17'h020 00
		RO	[twenty three : 17], RFU	7'b0
0X10C- 0X10E	Address pointer to function CSA	RW	[23:0], direction CSA of 24bit Address pointer, the host passes CSA Access window access CSA After that, the pointer is automatically added 1. The address is stored in small segments (LSB)	24'h000 000
0X10F	Data access window to CSA	RW	[7:0] ,Correct CSA Read and write window. When writing to this address, the corresponding data will be communicated Write through this window CSA 24bit In the address indicated by the address pointer, read the address When operating, from 24bit CSA The address indicated by the address pointer reads the data The window is sent to the host.	8'h00
0X110- 0X111	Function1 IO Block Size	RW	[15:0] , 16bit 'S register, used to set IO block size . biggest block size for 2048Byte , The minimum is 1 .	16'b0

			The data is stored in little-endian mode (LSB).	
0X1000 - 0X1010	CIS0	RO Host access Fn0 of CIS Address space, that is, the host accesses through this address space segment CIS0. this SDIO Support the most in the card 17 Bytes CIS0		
0X2000 - 0X2133	CIS1	RO Host access Fn1 of CIS Address space, that is, the host accesses through this address space segment CIS1 . this SDIO Card support CIS1 The byte data is 55~308.		
RFU		RFU		

11.4.3 SDIO Fn1 register

Fn1 The register is SDIO Agreement assigned to function1 'S address space, its address range is: 0x00000~0x1FFFF Of 128K . Due to the inside of the chip AHB The bus address bit width is 32 Bit, SDIO Not available 17 The bit address directly accesses the inside of the chip, so in the design, an address mapping needs to be completed. The specific mapping relationship is as follows: (FN1 Access space)

table 90 SDIO Fn1 Address mapping

SDIO Host access window	Corresponds to the actual physical address space	Actual physical address space content
0X0000 ~ 0X00FF	0X0000 ~ 0X00FF	SDIO Module internal register address space.
0X1000 ~ 0X1FFF	Configurable	CIS0 Physical space, the specific physical space is configured by the firmware.
0X2000 ~ 0X2FFF	Configurable	CIS1 Physical space, the specific physical space is configured by the firmware.
0X4000 ~ 0X4FFF	Configurable	Downstream and upstream cmd Physical space The address is configured by the firmware.
0X5000 ~ 0X5FFF 0X15000 ~ 0X15FFF	variable	send buffer Address space, depending on sdio_txbd Instructions.
0X6000 ~ 0X7FFF 0X16000 ~ 0X17FFF	variable	receive buffer Address space, depending on sdio_rxbd Instructions.
0X8000 ~ 0X9FFF	0X0E000000 ~ 0X0E002000	AHB bus config Address space.
0XA000 ~ 0XBFFF	0X0F000000 ~ 0X0F002000	AHB bus APB Address space.

The driver should avoid accessing the space beyond the above range. Doing so may bring unexpected results. The first address space register is in SDIO Internally and only by SDIO HOST Access; access to other address spaces will be rooted

According to the description, it is mapped to other spaces inside the chip.

This section only introduces SDIO 0x0000 ~ 0x00FF Registers in the address space, these registers consist of SDIO Host pass CMD52 command

Direct access, the offset address is the access address, the function number is 1 .

table 91 SDIO Fn1 Part of the register (for HOST access)

Offset address name		Bit wide access	description		Reset value
0X00~0X03			RO	RSV	
0X04		[7:1]	RO	RSV	7'b0
	int_read_data	[0]	RW Upstream	data is interrupted. Highly effective, write 1 Cleared. Reading 0x1C Will also be cleared automatically 0 .	1'b0
0X05		[7:1]	RO	RSV	7'b0
	int_mask	[0]	RW correspond int_src	Shield enable signal. 1 Then shield the corresponding interrupt.	1'd0
0X06		[7:2]	RO	RSV	6'b0
	wlan_awake_stts	[0]	RO current	WLAN status: 1 for ACTIVE ; 0 for SLEEP .	1'b1
0X1C	dat_len0	[6:0]	RO High upstream data length 7bit . Total 12bit ,low 5bit in 0x1D in.	7'b0	
	dat_vld	[7]	RO	1'b1	1'b1
0X1D	dat_len1	[7:3]	RO Low upstream data length 5bit . Total 12bit ,high 7bit in 0x1C in.	5'b0	
		[2:0]	RO	RSV	3'b0
0X1E			RO	RSV	
0X1F		[7:2]	RO	RSV	6'b0
	down_cmdbuf_vld	[1]	RO Down Command buffer Available, 1 effective.		1'b1
	txbuf_vld	[0]	RO Downstream data buffer Available. 1 Valid, indicating that there is available delivery	buffer .	1'b0
0X20	wlan_wake_en	[0]	RW	SLEEP Under state SDIO The issued chip wake-up is enabled, highly effective. After the chip is woken up, this bit will be automatically cleared by hardware 0 .	1'b0
0X21		[7:1]	RO	RSV	7'b0
	fn1_RST	[0]	RW Soft reset, 1 effective.	Software write 1 After, (ie chip wlan Part of the circuit) is reset, write 0 Rear, function1 The reset is released.	1'b0
0X22		[7:1]	RO	RSV	7'b0
	fn1_recov	[0]	RW Error recovery is enabled, 1 Valid command response After the end, the bit	Automatically clear 0 . This function is used to complete fn0/fn1 io abort Same function, when exists cmd Abnormal or premature end command, you can set this register 1 , To be done io abort operating. Because in some bus driver In the version, io abort Command is restricted (ie This register has limited access to the address space) and does not allow user drivers to use it.	1'b0

				At this time write to this register 1 Can replace io abort Operate and produce phase The same effect.	
--	--	--	--	--	--

11.4.3.1 SDIO AHB Interface slave register

The following register, in SDIO Used when the slave device is initialized.

When transferring data, you need to wrapper The controller is used together, Wrapper Part of the controller reference HSPI Part of the documentation.

table 92 SDIO AHB Bus register

Offset address name		Bit wide access	description		Reset value
0X0000			RO	Rsv	
0X0004					
0X0008	CIS function0 address	[31:0]	RW	CIS0 Inside the system memory The offset address stored in. CIS0 Actual storage start address = 0x01000 (Start address of read command) + The offset address	32'b0
0X000C	CIS function1 address	[31:0]	RW	CIS1 Inside the system memory The offset address stored in. CIS1 Actual storage start address = 0x02000 (Start address of read command) + The offset address	32'b0
0X0010	CSA address	[31:0]	RW	Set access during firmware initialization CSA Cheap address, its principle and CIS The settings are the same. Not supported in this design CSA .	32'b0
0X0014	Read address	[31:0]	RW	Used to set DMA From memory The starting address for reading data. Cooperate Data Port The register can realize the internal memory Reading Work (ie the access address is 0x00+ Read address). In this design, This method is not used, so the default is 0	32'b0
0X0018	Write address	[31:0]	RW	Used to set DMA to memory The starting address for writing data. Cooperate Data Port The register can realize the internal memory Write Work (ie the access address is 0x00+ write address). In this design, This method is not used, so the default is 0	32'b0
0X001C	AHB Transfer count	[20:0]	RW	SDIO The device notifies the host that in the read data operation to be initiated, it is necessary How many bytes of data to read. [23:21] RFU	32'b0
0X001F		RO	--	rsv	
0X0020	SDIO Transfer count	[20:0]	RO	During a data transfer, from the host to SDIO The number of bytes delivered by the device. When the data transmission is completed, the high-speed device is issued internally through this register	32'b0