BL602/604

Reference Manual

version：1.2

www.bouffalolab.com

Contents

1 System and memory overview 7
1.1 Introduction 7
1.2 Main features 7
1.3 Function description 7
1.4 Interrupt source 9
2 Reset and clock 11
2.1 Introduction 11
2.2 Reset source 11
2.3 Clock source 12
3 GLB 14
3.1 Introduction 14
3.2 GLB function description 14
3.2.1 Clock 14
3.2.2 Reset 14
3.2.3 Bus 14
3.2.4 Memory 15
3.2.5 GPIO overview 15
3.2.6 GPIO main features 15
3.2.7 GPIO function description 15
3.2.8 GPIO function 16
3.2.9 GPIO output 18
3.2.10 GPIO input 19

3.2.11 GPIO optional function . . . 19
3.2.12 GPIO interrupt . . . 19
3.3 Register description . . . 19
3.3.1 clk_cfg0 . . . 20
3.3.2 clk_cfg2 . . . 21
3.3.3 clk_cfg3 . . . 21
3.3.4 GPADC_32M_SRC_CTRL . . . 22
3.3.5 GPIO_CFGCTL0 . . . 23
3.3.6 GPIO_CFGCTL1 . . . 23
3.3.7 GPIO_CFGCTL2 . . . 24
3.3.8 GPIO_CFGCTL3 . . . 25
3.3.9 GPIO_CFGCTL4 . . . 26
3.3.10 GPIO_CFGCTL5 . . . 27
3.3.11 GPIO_CFGCTL6 . . . 28
3.3.12 GPIO_CFGCTL7 . . . 28
3.3.13 GPIO_CFGCTL8 . . . 29
3.3.14 GPIO_CFGCTL9 . . . 30
3.3.15 GPIO_CFGCTL10 . . . 31
3.3.16 GPIO_CFGCTL11 . . . 32
3.3.17 GPIO_CFGCTL12 . . . 33
3.3.18 GPIO_CFGCTL13 . . . 33
3.3.19 GPIO_CFGCTL14 . . . 34
4 ADC . . . 36
4.1 Introduction . . . 36
4.2 ADC main features . . . 36
4.3 ADC functional description . . . 37
4.3.1 ADC pins and internal signals . . . 38
4.3.2 ADC channel . . . 38
4.3.3 ADC clock . . . 39
4.3.4 ADC conversion mode . . . 40
4.3.5 ADC consequence . . . 40
4.3.6 ADC interrupt . . . 41
4.3.7 ADC FIFO . . . 42

4.3.8 ADC configuration process . . . 42
4.3.9 VBAT measurement . . . 43
4.3.10 TSEN measurement . . . 43
4.4 Register description . . . 44
4.4.1 gpadc_config . . . 45
4.4.2 gpadc_dma_rdata . . . 46
4.4.3 gpadc_reg_cmd . . . 46
4.4.4 gpadc_reg_config1 . . . 49
4.4.5 gpadc_reg_config2 . . . 51
4.4.6 gpadc_reg_scn_pos1 . . . 52
4.4.7 gpadc_reg_scn_pos2 . . . 53
4.4.8 gpadc_reg_scn_neg1 . . . 53
4.4.9 gpadc_reg_scn_neg2 . . . 54
4.4.10 gpadc_reg_status . . . 54
4.4.11 gpadc_reg_isr . . . 55
4.4.12 gpadc_reg_result . . . 55
4.4.13 gpadc_reg_raw_result . . . 56
4.4.14 gpadc_reg_define . . . 56
5 DAC . . . 57
5.1 Introduction . . . 57
5.2 Main features . . . 57
5.3 Function description . . . 57
5.4 Register description . . . 58
5.4.1 gpdac_config . . . 58
5.4.2 gpdac_dma_config . . . 59
5.4.3 gpdac_dma_wdata . . . 60
6 DMA . . . 61
6.1 Introduction . . . 61
6.2 DMA main features . . . 61
6.3 DMA functional description . . . 62
6.3.1 DMA transactions . . . 62
6.3.2 DMA channel configuration . . . 64
6.3.3 Peripheral support . . . 64

6.3.4 Linked List Mode . . . 65
6.3.5 DMA interrupt . . . 66
6.4 Transmission mode . . . 66
6.4.1 Memory to memory . . . 66
6.4.2 Memory to peripheral . . . 66
6.4.3 Peripheral to memory . . . 67
6.5 Register description . . . 67
6.5.1 DMA_IntStatus . . . 68
6.5.2 DMA_IntTCStatus . . . 69
6.5.3 DMA_IntTCClear . . . 69
6.5.4 DMA_IntErrorStatus . . . 70
6.5.5 DMA_IntErrClr . . . 70
6.5.6 DMA_RawIntTCStatus . . . 70
6.5.7 DMA_RawIntErrorStatus . . . 71
6.5.8 DMA_EnbldChns . . . 71
6.5.9 DMA_SoftBReq . . . 71
6.5.10 DMA_SoftSReq . . . 72
6.5.11 DMA_SoftLBReq . . . 72
6.5.12 DMA_SoftLSReq . . . 72
6.5.13 DMA_Config . . . 73
6.5.14 DMA_Sync . . . 73
6.5.15 DMA_C0SrcAddr . . . 73
6.5.16 DMA_C0DstAddr . . . 74
6.5.17 DMA_C0LLI . . . 74
6.5.18 DMA_C0Control . . . 74
6.5.19 DMA_C0Config . . . 75
6.5.20 DMA_C1SrcAddr . . . 76
6.5.21 DMA_C1DstAddr . . . 77
6.5.22 DMA_C1LLI . . . 77
6.5.23 DMA_C1Control . . . 77
6.5.24 DMA_C1Config . . . 78
6.5.25 DMA_C2SrcAddr . . . 79
6.5.26 DMA_C2DstAddr . . . 79

6.5.27 DMA_C2LLI . . . 80
6.5.28 DMA_C2Control . . . 80
6.5.29 DMA_C2Config . . . 81
6.5.30 DMA_C3SrcAddr . . . 82
6.5.31 DMA_C3DstAddr . . . 82
6.5.32 DMA_C3LLI . . . 83
6.5.33 DMA_C3Control . . . 83
6.5.34 DMA_C3Config . . . 84
7 L1C . . . 86
7.1 Introduction . . . 86
7.2 Main features . . . 87
7.3 Function description . . . 87
7.3.1 Mutual conversion between TCM and Cache RAM resources . . . 87
7.3.2 Cache . . . 87
7.4 Register description . . . 88
7.4.1 l1c_config . . . 89
7.4.2 hit_cnt_lsb . . . 89
7.4.3 hit_cnt_msb . . . 89
7.4.4 miss_cnt . . . 90
8 IR . . . 91
8.1 Introduction . . . 91
8.2 IR main features . . . 91
8.3 Function description . . . 91
8.3.1 Fixed receiving protocol . . . 91
8.3.2 Pulse width reception . . . 93
8.3.3 Normal sending mode . . . 93
8.3.4 Pulse width transmission . . . 93
8.3.5 Carrier modulation . . . 94
8.3.6 IR interrupt . . . 94
8.4 Register description . . . 94
8.4.1 irtx_config . . . 95
8.4.2 irtx_int_sts . . . 96
8.4.3 irtx_data_word0 . . . 96

8.4.4 irtx_data_word1 . . . 97
8.4.5 irtx_pulse_width . . . 97
8.4.6 irtx_pw . . . 98
8.4.7 irtx_swm_pw_0 . . . 98
8.4.8 irtx_swm_pw_1 . . . 99
8.4.9 irtx_swm_pw_2 . . . 99
8.4.10 irtx_swm_pw_3 . . . 100
8.4.11 irtx_swm_pw_4 . . . 100
8.4.12 irtx_swm_pw_5 . . . 100
8.4.13 irtx_swm_pw_6 . . . 101
8.4.14 irtx_swm_pw_7 . . . 101
8.4.15 irrx_config . . . 102
8.4.16 irrx_int_sts . . . 102
8.4.17 irrx_pw_config . . . 103
8.4.18 irrx_data_count . . . 103
8.4.19 irrx_data_word0 . . . 104
8.4.20 irrx_data_word1 . . . 104
8.4.21 irrx_swm_fifo_config_0 . . . 104
8.4.22 irrx_swm_fifo_rdata . . . 105
9 SPI . . . 106
9.1 Introduction . . . 106
9.2 Main features . . . 106
9.3 Function description . . . 106
9.3.1 Clock control . . . 106
9.3.2 Master continuous transmission mode . . . 107
9.3.3 Acceptance filtering function . . . 107
9.3.4 Receive error correction . . . 108
9.3.5 Slave mode timeout mechanism . . . 108
9.3.6 I/O transfer mode . . . 108
9.3.7 DMA transfer mode . . . 108
9.3.8 SPI interrupt . . . 109
9.4 Register description . . . 109
9.4.1 spi_config . . . 110

9.4.2 spi_int_sts . . . 111
9.4.3 spi_bus_busy . . . 112
9.4.4 spi_prd_0 . . . 113
9.4.5 spi_prd_1 . . . 113
9.4.6 spi_rxd_ignr . . . 113
9.4.7 spi_sto_value . . . 114
9.4.8 spi_fifo_config_0 . . . 114
9.4.9 spi_fifo_config_1 . . . 115
9.4.10 spi_fifo_wdata . . . 115
9.4.11 spi_fifo_rdata . . . 116
10 UART . . . 117
10.1 Introduction . . . 117
10.2 Main features . . . 117
10.3 Function description . . . 118
10.3.1 Data format description . . . 118
10.3.2 Basic architecture diagram . . . 118
10.3.3 Clock source . . . 118
10.3.4 Baud rate setting . . . 119
10.3.5 Transmitter . . . 120
10.3.6 receiver . . . 120
10.3.7 Automatic baud rate detection . . . 121
10.3.8 Hardware flow control . . . 122
10.3.9 DMA transfer mode . . . 122
10.3.10 UART interrupt . . . 122
10.4 Register description . . . 123
10.4.1 utx_config . . . 124
10.4.2 urx_config . . . 125
10.4.3 uart_bit_prd . . . 125
10.4.4 data_config . . . 126
10.4.5 utx_ir_position . . . 126
10.4.6 urx_ir_position . . . 127
10.4.7 urx_rto_timer . . . 127
10.4.8 uart_int_sts . . . 127

10.4.9 uart_int_mask . . . 128
10.4.10 uart_int_clear . . . 129
10.4.11 uart_int_en . . . 129
10.4.12 uart_status . . . 130
10.4.13 sts_urx_abr_prd . . . 130
10.4.14 uart_fifo_config_0 . . . 130
10.4.15 uart_fifo_config_1 . . . 131
10.4.16 uart_fifo_wdata . . . 132
10.4.17 uart_fifo_rdata . . . 132
11 I2C . . . 133
11.1 Introduction . . . 133
11.2 Main features . . . 133
11.3 Function description . . . 133
11.3.1 Start and stop conditions . . . 134
11.3.2 Data transmission format . . . 134
11.3.3 Arbitration . . . 135
11.4 I2C clock setting . . . 136
11.5 I2C configuration process . . . 136
11.5.1 Configuration item . . . 136
11.5.2 Read and write flags . . . 137
11.5.3 Slave address . . . 137
11.5.4 Slave device address . . . 137
11.5.5 Slave device address length . . . 137
11.5.6 Data . . . 137
11.5.7 Data length . . . 137
11.5.8 Enable signal . . . 137
11.6 FIFO management . . . 138
11.7 Using DMA . . . 139
11.7.1 DMA transmission process . . . 139
11.7.2 DMA receiving process . . . 139
11.8 Interrupt . . . 140
11.9 Register description . . . 140
11.9.1 i2c_config . . . 141

11.9.2 i2c_int_sts . . . 141
11.9.3 i2c_sub_addr . . . 143
11.9.4 i2c_bus_busy . . . 143
11.9.5 i2c_prd_start . . . 144
11.9.6 i2c_prd_stop . . . 144
11.9.7 i2c_prd_data . . . 144
11.9.8 i2c_fifo_config_0 . . . 145
11.9.9 i2c_fifo_config_1 . . . 146
11.9.10 i2c_fifo_wdata . . . 146
11.9.11 i2c_fifo_rdata . . . 146
12 PWM . . . 148
12.1 Introduction . . . 148
12.2 Main features . . . 148
12.3 Function description . . . 148
12.3.1 Clock and divider . . . 148
12.3.2 Pulse generation principle . . . 149
12.3.3 PWM interrupt . . . 150
12.4 Register description . . . 150
12.4.1 pwm_int_config . . . 151
12.4.2 pwm0_clkdiv . . . 152
12.4.3 pwm0_thre1 . . . 152
12.4.4 pwm0_thre2 . . . 152
12.4.5 pwm0_period . . . 153
12.4.6 pwm0_config . . . 153
12.4.7 pwm0_interrupt . . . 154
12.4.8 pwm1_clkdiv . . . 154
12.4.9 pwm1_thre1 . . . 155
12.4.10 pwm1_thre2 . . . 155
12.4.11 pwm1_period . . . 155
12.4.12 pwm1_config . . . 156
12.4.13 pwm1_interrupt . . . 156
12.4.14 pwm2_clkdiv . . . 157
12.4.15 pwm2_thre1 . . . 157

12.4.16 pwm2_thre2 . . . 157
12.4.17 pwm2_period . . . 158
12.4.18 pwm2_config . . . 158
12.4.19 pwm2_interrupt . . . 159
12.4.20 pwm3_clkdiv . . . 159
12.4.21 pwm3_thre1 . . . 160
12.4.22 pwm3_thre2 . . . 160
12.4.23 pwm3_period . . . 160
12.4.24 pwm3_config . . . 161
12.4.25 pwm3_interrupt . . . 161
12.4.26 pwm4_clkdiv . . . 162
12.4.27 pwm4_thre1 . . . 162
12.4.28 pwm4_thre2 . . . 162
12.4.29 pwm4_period . . . 163
12.4.30 pwm4_config . . . 163
12.4.31 pwm4_interrupt . . . 164
13 TIMER . . . 165
13.1 Introduction . . . 165
13.2 Main features . . . 166
13.3 Function description . . . 166
13.3.1 8-bit divider . . . 166
13.3.2 General timer operating mode . . . 167
13.3.3 Watchdog timer operating mode . . . 168
13.3.4 Alarm setting . . . 168
13.3.5 Watchdog alarm . . . 168
13.4 Register description . . . 169
13.4.1 TCCR . . . 171
13.4.2 TMR2_0 . . . 171
13.4.3 TMR2_1 . . . 172
13.4.4 TMR2_2 . . . 172
13.4.5 TMR3_0 . . . 172
13.4.6 TMR3_1 . . . 173
13.4.7 TMR3_2 . . . 173

13.4.8 TCR2 . . . 173
13.4.9 TCR3 . . . 174
13.4.10 TMSR2 . . . 174
13.4.11 TMSR3 . . . 174
13.4.12 TIER2 . . . 175
13.4.13 TIER3 . . . 175
13.4.14 TPLVR2 . . . 176
13.4.15 TPLVR3 . . . 176
13.4.16 TPLCR2 . . . 177
13.4.17 TPLCR3 . . . 177
13.4.18 WMER . . . 177
13.4.19 WMR . . . 178
13.4.20 WVR . . . 178
13.4.21 WSR . . . 179
13.4.22 TICR2 . . . 179
13.4.23 TICR3 . . . 180
13.4.24 WICR . . . 180
13.4.25 TCER . . . 180
13.4.26 TCMR . . . 181
13.4.27 TILR2 . . . 181
13.4.28 TILR3 . . . 182
13.4.29 WCR . . . 183
13.4.30 WFAR . . . 183
13.4.31 WSAR . . . 183
13.4.32 TCVWR2 . . . 184
13.4.33 TCVWR3 . . . 184
13.4.34 TCVSYN2 . . . 184
13.4.35 TCVSYN3 . . . 185
13.4.36 TCDR . . . 185
14 Revision history . . . 186

List of Figures

2.1 Reset source 21
2.2 Clock Block Diagram 22

3.1 GPIO Basic Struct 25

4.1 ADC block diagram 46
4.2 ADC Clock 48

6.1 DMA architecture 72
6.2 LLI architecture 74

7.1 L1c architecture 95
7.2 Cache architecture 97

8.1 nec logical 101
8.2 nec 101
8.3 rc5 logical 101
8.4 rc5 102

9.1 SPI clock 116
9.2 SPI ignore 117

10.1 UART data 127
10.2 UART clock 128
10.3 UART sample 129
10.4 UART fixed character mode 130
10.5 UART flow control 131

11.1 I2C stop/start condition 143
11.2 Master transmission 144

11.3 Master tx and slave rx 144
11.4 Master rx and slave tx 144
11.5 Tx and Rx together 145

12.1 Pwm 158

13.1 Timer block diagram 174
13.2 Watchdog timer block diagram 175
13.3 Timer Preload 176
13.4 Watchdog timing 177
13.5 Watchdog alarm mechanism 178

List of Tables

1.1 Bus connection 17
1.2 Address mapping 17
1.2 Address mapping 18
1.3 Interrupt distribution 18
1.3 Interrupt distribution 19

3.1 Pin description 26
3.1 Pin description 27

4.1 ADC internal signals 47
4.2 ADC external pins 47
4.3 Meaning of ADC conversion result 50

7.1 WayDisable settings 96

11.1 Pin lists 142

12.1 Duty Cycle Parameters 159

14.1 Document revision history 195

1 System and memory overview

1.1 Introduction

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

1.2 Main features

- RISC-V 32-bit with floating point
- Multi-layer 32-bit AHB bus architecture
- 96KB high-speed memory
- 180KB system memory
- 128KB read-only memory
- Off-chip memory Flash

1.3 Function description

The BL602 bus connection and address access are summarized as follows: The bus master includes CPU, SDIO, DMA, encryption engine, and debug interface. The bus includes memory, peripherals, WiFi / BLE. Except the encryption engine can only access the memory, all other bus masters can access all bus slaves.

Table 1.1: Bus connection

Slave/Master	CPU	SDIO	DMA	encryption engine	Debug interface
memory	V	V	V	V	V
Peripheral	V	V	V	-	V
WiFi/BLE	V	V	V	-	V

The address access mainly distinguishes ”memory” or ”peripheral” by [27:24], and the [31:28] bits can be ignored. The memory space is consecutive addresses 0x2008000 ~ 0x204BFFF (272KB SRAM), the read-only memory address is 0x1000000, and the deep sleep memory address is 0x0010000. The off-chip space address is 0x3000000 (maximum support 16MB Flash). The peripheral space is 0x0000000 ~ 0x000F000.

Table 1.2: Address mapping

Name	Address	Size	Description
WRAM	0x42030000	112KB	Wireless SRAM memory
RETRAM	0x40010000	4KB	Deep sleep memory (RAM reserved)
HBN	0x4000F000	4KB	Deep Sleep Control (Hibernation)
PDS	0x4000E000	4KB	Sleep control (power-down sleep)
SDU	0x4000D000	4KB	SDIO control
DMA	0x4000C000	4KB	DMA control
QSPI	0x4000B000	4KB	Flash control
IRR	0x4000A600	256B	Infrared remote control
TIMER	0x4000A500	256B	Timer control
PWM	0x4000A400	256B	Pulse width modulation control
I2C	0x4000A300	256B	I2C control
SPI	0x4000A200	256B	SPI master / slave control
UART1	0x4000A100	256B	UART control
UART0	0x4000A000	256B	UART control
L1C	0x40009000	4KB	Cache control
eFuse	0x40007000	4KB	eFuse memory control
TZ2	0x40006000	4KB	Trust zone isolation
TZ1	0x40005000	4KB	Trust zone isolation
SEC	0x40004000	4KB	Security engine
GPIP	0x40002000	4KB	Universal DAC/ADC/ACOMP interface control

Table 1.2: Address mapping

Name	Address	Size	Description
MIX	0x40001000	4KB	Mixed signal register
GLB	0x40000000	4KB	Global register
RAM	0x22020000 /0x42020000	64KB	On-chip memory.If used as data memory, use address 0x42020000 for access; if used as program memory, use address 0x22020000 for access
XIP	0x23000000	16MB	XIP flash
TCM1	0x22014000 /0x42014000	48KB	Cache memory.If used as data memory, use address 0x42014000 for access; if used as program memory, use address 0x22014000 for access
TCM0	0x22008000 /0x42008000	48KB	Cache memory.If used as data memory, use address 0x42008000 for access; if used as program memory, use address 0x22008000 for access
ROM	0x21000000	128KB	Read-only memory

1.4 Interrupt source

BL602/BL604 contains a total of 18 interrupt sources. The interrupt sources and corresponding interrupt numbers are shown in the following table:

Table 1.3: Interrupt distribution

Interrupt source		Number	Description
L1C	L1C_BMX_ERR	IRQ_NUM_BASE+2	L1C BMX Error Interrupt
	L1C_BMX_TO	IRQ_NUM_BASE+3	L1C BMX Timeout Interrupt
DMA	DMA_BMX_ERR	IRQ_NUM_BASE+8	DMA BMX Error Interrupt
IR	IRTX	IRQ_NUM_BASE+19	IR TX Interrupt
	IRRX	IRQ_NUM_BASE+20	IR RX Interrupt
ADC	GPADC_DMA	IRQ_NUM_BASE+25	GPADC_DMA Interrupt
SPI	SPI	IRQ_NUM_BASE+27	SPI Interrupt
UART	UART0	IRQ_NUM_BASE+29	UART0 Interrupt
	UART1	IRQ_NUM_BASE+30	UART1 Interrupt
I2C	I2C	IRQ_NUM_BASE+32	I2C Interrupt
PWM	PWM	IRQ_NUM_BASE+34	PWM Interrupt

Table 1.3: Interrupt distribution

Interrupt source		Number	Description
TIMER	TIMER_CH0	IRQ_NUM_BASE+36	Timer Channel 0 Interrupt
	TIMER_CH1	IRQ_NUM_BASE+37	Timer Channel 1 Interrupt
	TIMER_WDT	IRQ_NUM_BASE+38	Watch Dog Interrupt
GPIO	GPIO_INT0	IRQ_NUM_BASE+44	GPIO Interrupt
PDS	PDS_WAKEUP	IRQ_NUM_BASE+50	PDS Wakeup Interrupt
HBN	HBN_OUT0	IRQ_NUM_BASE+51	Hibernate out 0 Interrupt
	HBN_OUT1	IRQ_NUM_BASE+52	Hibernate out 1 Interrupt

Note: The IRQ_NUM_BASE is 16, and the interrupt numbers 0-15 are reserved for RISC-V interrupts.

2 Reset and clock

2.1 Introduction

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

2.2 Reset source

The reset sources are as follows:

- Hardware reset: reset via pins
  - Pin global reset (PAD_EXT_RST = 1-> 0): all logic will reset and return to the initial state(for QFN40)
  - Pin power reset (CHIP_EN = 0-> 1): similar to power management reset
  - Power management reset: The chip is restored from power failure, and the HBN logic resets the chip system
- Watchdog reset
  - When the watchdog alarm triggers a reset signal, the reset management unit will reset the chip system after necessary preparations, and the internal logic of the watchdog will record the status of the watchdog reset
- Software reset: local or partial reset according to software setting register
  - Software initial reset (reg_ctrl_pwron_rst): The rising edge of this register is triggered by software to reset the chip system
  - Software CPU reset (reg_ctrl_cpu_reset): The rising edge of this register is triggered by software to reset the CPU part of the system
  - Retain necessary logic processing such as power management unit, perform chip system reset
  - Software module reset: Set software reset according to the requirements of specific modules

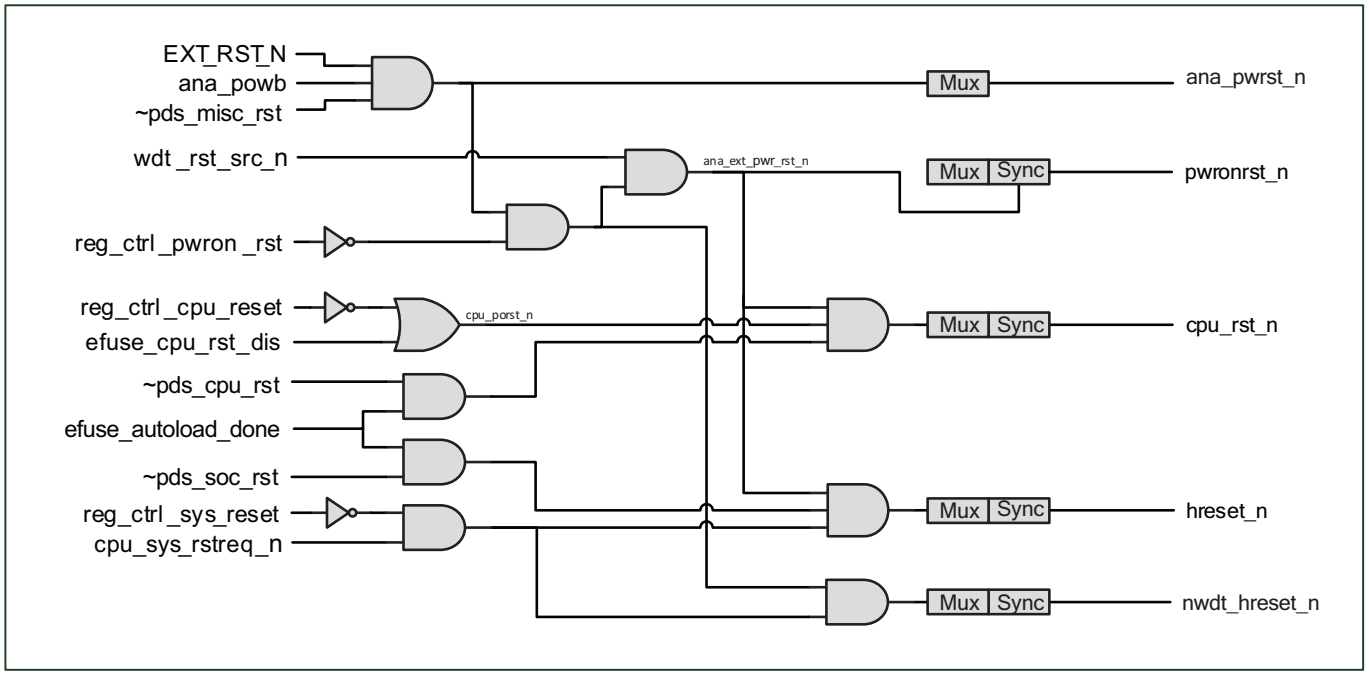


Figure 2.1: Reset source

2.3 Clock source

Clock source contains:

- XTAL：External crystal clock, according to system requirements, the frequency can be selected from 24, 32, 38.4, 40MHz.
- XTAL32K：External crystal clock, frequency 32kHz
- RC32K ： RC oscillator clock, 32kHz, provides calibration
- RC32M ： RC oscillator clock, frequency 32MHz, provides calibration
- PLL ： Phase-locked loop clock, internal system high-speed clock, the highest frequency supports 192MHz

The clock control unit distributes the clock from the oscillator to the core and peripheral devices. By selecting the system clock source, dynamic frequency divider, clock configuration, sleep using 32KHz clock to achieve low power clock management.

Peripheral clock includes: Flash、UART、I2C、SPI、PWM、IR-remote、ADC、DAC.

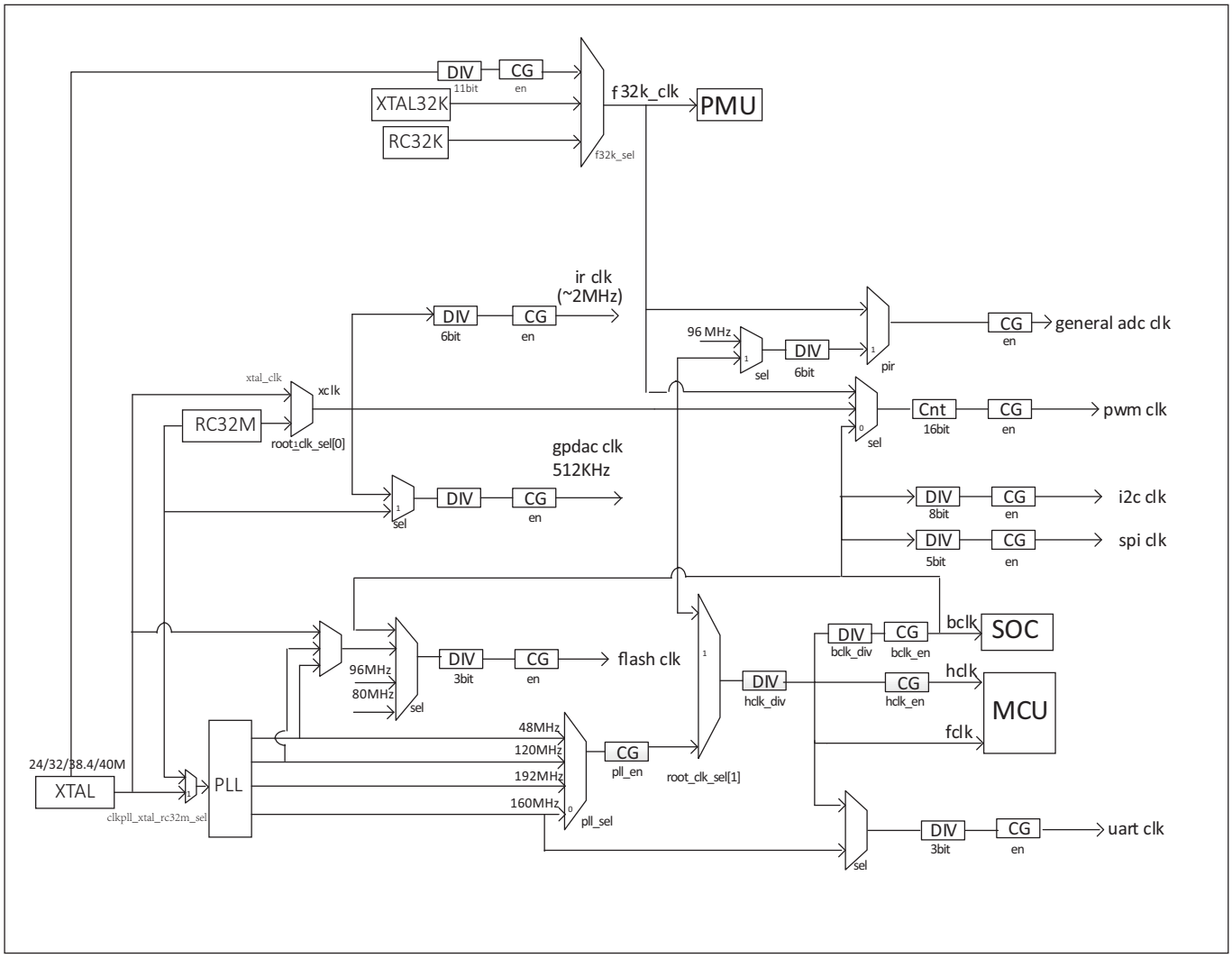


Figure 2.2: Clock Block Diagram

3 GLB

3.1 Introduction

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

3.2 GLB function description

3.2.1 Clock

The clock management function is mainly used to set the clock of the processor, bus, and various peripherals. This module can set the clock source, clock frequency division, etc. of the module's work, and can also achieve the gate control of the module's clock to achieve the purpose of low power consumption of the system.

For detailed settings, please refer to the relevant chapter of the system clock.

3.2.2 Reset

Provide individual reset function for each peripheral and chip reset function.

The chip reset includes:

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

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

3.2.3 Bus

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

3.2.4 Memory

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

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

3.2.5 GPIO overview

The GPIO management function provides GPIO control registers to realize the configuration of GPIO attributes by software, so that users can conveniently operate GPIO. Each GPIO can be configured as three modes of input, output and optional function. In each mode (except for analog optional functions), it provides three port states: pull-up, pull-down, and floating. In addition, GPIO also provides interrupt functions, which can be configured as rising edge trigger, falling edge trigger, or High/low level trigger.

3.2.6 GPIO main features

- It can be configured as a normal input / output function. In this mode, pull-up, pull-down or floating input/output can be set.
- It can be configured as an optional function and used with peripheral functions. In this mode, pull-up and pull-down can also be set. When using the analog function, it must be set to floating.
- The drive capability can be set to provide greater output current.
- Schmitt trigger function can be set to provide simple hardware anti-shake function.

3.2.7 GPIO function description

Each GPIO can be configured by software as:

- Floating input
- Pull-up input
- Pull down input
- Pull-up interrupt input
- Pull-down interrupt input
- Floating interrupt input
- Pull-up output
- Pull-down output

- Floating output
- Analog input optional function
- Analog output optional function
- Digital optional functions

The basic block diagram of the GPIO module is shown below:

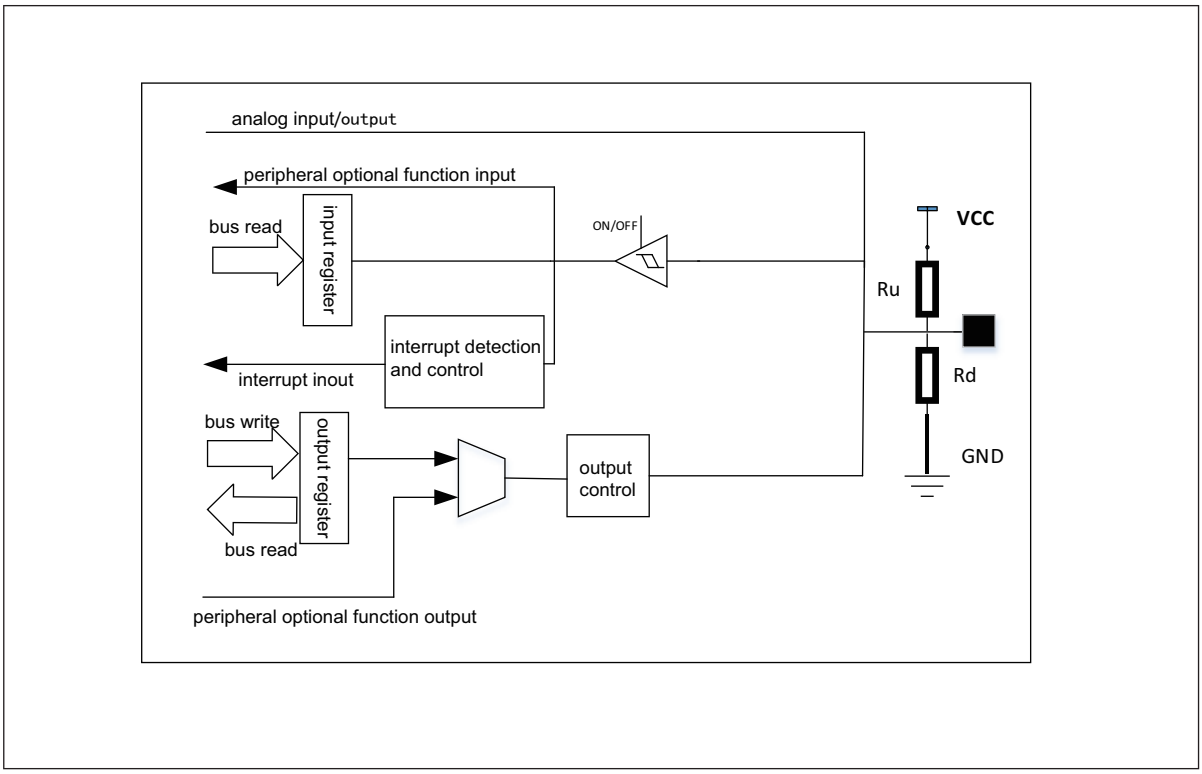


Figure 3.1: GPIO Basic Struct

3.2.8 GPIO function

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

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

The functions that GPIO can set include:

- Flash/QSPI：set GPIO as QSPI function, can be connected to Flash as program storage / run medium
- SPI: set GPIO as SPI function
- I2C: set GPIO to I2C function
- UART: set GPIO as UART function
- PWM: set GPIO to PWM function
- ANA: set GPIO to Analog function
- SWGPIO: set GPIO as general IO function
- JTAG: set GPIO to JTAG function

In order to meet the needs of customers as much as possible, each of the GPIOs can basically select the above optional functions. When selecting an optional function, the GPIO and corresponding function signals are shown in the following table:

Table 3.1: Pin description

GPIO	SDIO	FLASH	SPI	I2C	UART	PWM	Analog	SWGPIO	JTAG
GPIO0	CLK	D1	MISO	SCL	SIG0	CH0		SWGPIO0	TMS
GPIO1	CMD	D2	MOSI	SDA	SIG1	CH1		SWGPIO1	TDI
GPIO2	DAT0	D2	SS	SCL	SIG2	CH2		SWGPIO2	TCK
GPIO3	DAT1	D3	SCLK	SDA	SIG3	CH3		SWGPIO3	TDO
GPIO4	DAT2		MISO	SCL	SIG4	CH4	CH1	SWGPIO4	TMS
GPIO5	DAT3		MOSI	SDA	SIG5	CH0	CH4	SWGPIO5	TDI
GPIO6			SS	SCL	SIG6	CH1	CH5	SWGPIO6	TCK
GPIO7			SCLK	SDA	SIG7	CH2		SWGPIO7	TDO
GPIO8			MISO	SCL	SIG0	CH3		SWGPIO8	TMS
GPIO9			MOSI	SDA	SIG1	CH4	CH6/7	SWGPIO9	TDI
GPIO10			SS	SCL	SIG2	CH0	MICBIAS/CH8/9	SWGPIO10	TCK
GPIO11			SCLK	SDA	SIG3	CH1	IROUT/CH10	SWGPIO11	TDO
GPIO12			MISO	SCL	SIG4	CH2	ADC_VREF/CH0	SWGPIO12	TMS
GPIO13			MOSI	SDA	SIG5	CH3	CH3	SWGPIO13	TDI
GPIO14			SS	SCL	SIG6	CH4	CH2	SWGPIO14	TCK
GPIO15			SCLK	SDA	SIG7	CH0	PSWIROUT/CH11	SWGPIO15	TDO
GPIO16			MISO	SCL	SIG0	CH1		SWGPIO16	TMS

Table 3.1: Pin description

GPIO	SDIO	FLASH	SPI	I2C	UART	PWM	Analog	SWGPIO	JTAG
GPIO17		D3	MOSI	SDA	SIG1	CH2	DC_TP_OUT	SWGPIO17	TDI
GPIO18		D2	SS	SCL	SIG2	CH3		SWGPIO18	TCK
GPIO19		D1	SCLK	SDA	SIG3	CH4		SWGPIO19	TDO
GPIO20		D0	MISO	SCL	SIG4	CH0		SWGPIO20	TMS
GPIO21		CS	MOSI	SDA	SIG5	CH1		SWGPIO21	TDI
GPIO22		CLK_OUT	SS	SCL	SIG6	CH2		SWGPIO22	TCK

In the above table, when the UART function is selected, only one signal of the UART is selected, and the specific function of the pin is not specified (such as UART TX or UART RX). It is also necessary to use UART_SIGX_SEL(X = 0-7) to select specific UART signals and corresponding functions.

The signals that can be selected for each UART_SIGX_SEL include:

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

Take GPIO0 as an example, when fun_sel selects UART, GPIO0 selects UART_SIG0. By default, the value of UART_SIG0_SEL is 0, which is UART0_RTS, that is, GPIO is UART0_RTS function. If the application wants to use GPIO as UART1_TXD, as long as UART_SIG0_SEL is set to 6, then the function of GPIO0 is UART1_TXD.

3.2.9 GPIO output

By setting func_sel to SWGPIO, GPIO can be used as the input / output of ordinary GPIO. Setting ie to 0 and oe to 1 can configure GPIO as an output function. The output value is set through the GPIO_O register group.

When the corresponding bit of GPIO_O is set to 0, the GPIO output is low, and when the corresponding bit of GPIO_O is set to 1, the GPIO output is high. The output capability can be set via the drv control bit.

3.2.10 GPIO input

Set func_sel to SWGPIO, set ie to 1, and oe to 0. The user can configure the GPIO as an input function, set whether to enable the Schmitt trigger through the smt control bit, and set the pull-down property through the pd, pu control bit.

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

3.2.11 GPIO optional function

Setting func_sel as the corresponding peripheral function can realize the connection between GPIO and peripherals, and realize the input and output of peripherals. As can be seen from the basic functional block diagram of GPIO, when selecting optional functions, it is necessary to set ie to 1, oe Set to 0, that is to disconnect the output control function of ordinary GPIO.

In this way, for peripherals with fixed input functions, the OE signal of the peripheral is always 0 to implement the input function; for peripherals with fixed output, the OE signal is always 1 so that the output is controlled by the peripheral. At this time, The input signal is the output signal, but it will not be collected by the output peripheral. When the peripheral needs both input and output, the input and output can be realized by controlling the peripheral OE signal.

3.2.12 GPIO interrupt

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

- Interrupt on rising edge
- Interrupt on falling edge
- High level trigger interrupt
- Low level trigger interrupt

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

3.3 Register description

Name	Description
clk_cfg0	Clock configuration-processor, bus
clk_cfg2	Clock configuration-UART,Flash
clk_cfg3	Clock configuration-I2C,SPI
GPADC_32M_SRC_CTRL	Clock configuration-GPADC

Name	Description
GPIO_CFGCTL0	GPIO0, GPIO1 configuration
GPIO_CFGCTL1	GPIO2, GPIO3 configuration
GPIO_CFGCTL2	GPIO4, GPIO5 configuration
GPIO_CFGCTL3	GPIO6, GPIO7 configuration
GPIO_CFGCTL4	GPIO8, GPIO9 configuration
GPIO_CFGCTL5	GPIO10, GPIO11 configuration
GPIO_CFGCTL6	GPIO12, GPIO13 configuration
GPIO_CFGCTL7	GPIO14, GPIO15 configuration
GPIO_CFGCTL8	GPIO16, GPIO17 configuration
GPIO_CFGCTL9	GPIO18, GPIO19 configuration
GPIO_CFGCTL10	GPIO20, GPIO21 configuration
GPIO_CFGCTL11	GPIO22, GPIO23 configuration
GPIO_CFGCTL12	GPIO24, GPIO25 configuration
GPIO_CFGCTL13	GPIO26, GPIO27 configuration
GPIO_CFGCTL14	GPIO28 configuration

3.3.1 clk_cfg0

Address：0x40000000

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

Bits	Name	Type	Reset	Description
31:28	GLBID	R	4'h6	
27:24	RSVD			
23:16	BCLKDIV	R/W	0	bclk divide from hclk
15:8	HCLKDIV	R/W	0	hclk divide from root clock (clock source selected by hbn_root_clk_sel)

Bits	Name	Type	Reset	Description
7:6	RCSEL	R	0	root clock selection from HBN (0: RC32M 1: XTAL 2/3: PLL others)
5:4	PLLSEL	R/W	0	pll clock selection (0: 48MHz 1: 120MHz 2: 160MHz 3: 192MHz)
3:0	RSVD			

3.3.2 clk_cfg2

Address：0x40000008

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

Bits	Name	Type	Reset	Description
31:24	DMAEN	R/W	8'hff	CH0, 1, 2, AHBm, AHBs, Rqs
23:14	RSVD			
13:12	SFSEL	R/W	2'd2	Flash Clock Select (0: 120M, 1:80M, 2:HCLK, 3:96M)
11	SFEN	R/W	1	Flash Clock Enable
10:8	SFDIV	R/W	3'd3	Flash Clock Divider (Selected Flash Clock)/(N+1)
7	HUCSEL	R	0	uart clock selection from HBN (0: root clock 1: PLL 160M)
6:5	RSVD			
4	UARTEN	R/W	1	UART Clock Enable
3	RSVD			
2:0	UARTDIV	R/W	3'd7	UART Clock Divider (root clock or 160M)/(N+1) (clock source selected by hbn_uart_clk_sel)

3.3.3 clk_cfg3

Address：0x4000000c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD							I2C EN	I2CDIV							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							SPI EN	RSVD			SPIDIV				

Bits	Name	Type	Reset	Description
31:25	RSVD			
24	I2CEN	R/W	1	I2C Master Clock Out Enable
23:16	I2CDIV	R/W	8'd255	I2C Master Clock Out Divider (Freq_of_BCLK/(N+1))
15:9	RSVD			
8	SPIEN	R/W	1	SPI Clock Enable (Default : Enable)
7:5	RSVD			
4:0	SPIDIV	R/W	5'd3	SPI Clock Divider (BUS_CLK/(N+1)), default BUS_CLK/4

3.3.4 GPADC_32M_SRC_CTRL

Address：0x400000a4

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

Bits	Name	Type	Reset	Description
31:9	RSVD			
8	GADCDIV	R/W	1	GPADC 32M Clock Dvider Enable
7	GADCSEL	R/W	0	GPADC Clock Source Select. 0: 96MHz, 1: xclk
6	RSVD			
5:0	GADCDIV	R/W	6'd2	GPADC 32M Clock Divider (96M)/(N+1) , default : 96M/3 = 32M

3.3.5 GPIO_CFGCTL0

Address：0x40000100

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP1FUNC	R/W	4'h1	GPIO Function Select (Default : SDIO)
23:22	RSVD			
21	GP1PD	R/W	0	GPIO Pull Down Control
20	GP1PU	R/W	0	GPIO Pull Up Control
19:18	GP1DRV	R/W	0	GPIO Driving Control
17	GP1SMT	R/W	1	GPIO SMT Control
16	GP1IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP0FUNC	R/W	4'h1	GPIO Function Select (Default : SDIO)
7:6	RSVD			
5	GP0PD	R/W	0	GPIO Pull Down Control
4	GP0PU	R/W	0	GPIO Pull Up Control
3:2	GP0DRV	R/W	0	GPIO Driving Control
1	GP0SMT	R/W	1	GPIO SMT Control
0	GP0IE	R/W	1	GPIO Input Enable

3.3.6 GPIO_CFGCTL1

Address：0x40000104

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP3FUNC	R/W	4'h1	GPIO Function Select (Default : SDIO)
23:22	RSVD			
21	GP3PD	R/W	0	GPIO Pull Down Control
20	GP3PU	R/W	0	GPIO Pull Up Control
19:18	GP3DRV	R/W	0	GPIO Driving Control
17	GP3SMT	R/W	1	GPIO SMT Control
16	GP3IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP2FUNC	R/W	4'h1	GPIO Function Select (Default : SDIO)
7:6	RSVD			
5	GP2PD	R/W	0	GPIO Pull Down Control
4	GP2PU	R/W	0	GPIO Pull Up Control
3:2	GP2DRV	R/W	0	GPIO Driving Control
1	GP2SMT	R/W	1	GPIO SMT Control
0	GP2IE	R/W	1	GPIO Input Enable

3.3.7 GPIO_CFGCTL2

Address：0x40000108

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP5FUNC	R/W	4'h1	GPIO Function Select (Default : SDIO)
23:22	RSVD			
21	GP5PD	R/W	0	GPIO Pull Down Control
20	GP5PU	R/W	0	GPIO Pull Up Control
19:18	GP5DRV	R/W	0	GPIO Driving Control
17	GP5SMT	R/W	1	GPIO SMT Control
16	GP5IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP4FUNC	R/W	4'h1	GPIO Function Select (Default : SDIO)
7:6	RSVD			
5	GP4PD	R/W	0	GPIO Pull Down Control
4	GP4PU	R/W	0	GPIO Pull Up Control
3:2	GP4DRV	R/W	0	GPIO Driving Control
1	GP4SMT	R/W	1	GPIO SMT Control
0	GP4IE	R/W	1	GPIO Input Enable

3.3.8 GPIO_CFGCTL3

Address：0x4000010c

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP7FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP7PD	R/W	0	GPIO Pull Down Control
20	GP7PU	R/W	0	GPIO Pull Up Control

Bits	Name	Type	Reset	Description
19:18	GP7DRV	R/W	0	GPIO Driving Control
17	GP7SMT	R/W	1	GPIO SMT Control
16	GP7IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP6FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP6PD	R/W	0	GPIO Pull Down Control
4	GP6PU	R/W	0	GPIO Pull Up Control
3:2	GP6DRV	R/W	0	GPIO Driving Control
1	GP6SMT	R/W	1	GPIO SMT Control
0	GP6IE	R/W	1	GPIO Input Enable

3.3.9 GPIO_CFGCTL4

Address：0x40000110

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP9FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP9PD	R/W	0	GPIO Pull Down Control
20	GP9PU	R/W	0	GPIO Pull Up Control
19:18	GP9DRV	R/W	0	GPIO Driving Control
17	GP9SMT	R/W	1	GPIO SMT Control
16	GP9IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP8FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)

Bits	Name	Type	Reset	Description
7:6	RSVD			
5	GP8PD	R/W	0	GPIO Pull Down Control
4	GP8PU	R/W	0	GPIO Pull Up Control
3:2	GP8DRV	R/W	0	GPIO Driving Control
1	GP8SMT	R/W	1	GPIO SMT Control
0	GP8IE	R/W	1	GPIO Input Enable

3.3.10 GPIO_CFGCTL5

Address：0x40000114

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP11FUNC	R/W	4'hE	GPIO Function Select (Default : JTAG)
23:22	RSVD			
21	GP11PD	R/W	0	GPIO Pull Down Control
20	GP11PU	R/W	0	GPIO Pull Up Control
19:18	GP11DRV	R/W	0	GPIO Driving Control
17	GP11SMT	R/W	1	GPIO SMT Control
16	GP11IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP10FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP10PD	R/W	0	GPIO Pull Down Control
4	GP10PU	R/W	0	GPIO Pull Up Control
3:2	GP10DRV	R/W	0	GPIO Driving Control
1	GP10SMT	R/W	1	GPIO SMT Control

Bits	Name	Type	Reset	Description
0	GP10IE	R/W	1	GPIO Input Enable

3.3.11 GPIO_CFGCTL6

Address：0x40000118

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD				GP13FUNC				RSVD		GP13 PD	GP13 PU	GP13DRV		GP13 SMT	GP13 IE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				GP12FUNC				RSVD		GP12 PD	GP12 PU	GP12DRV		GP12 SMT	GP12 IE

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP13FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP13PD	R/W	0	GPIO Pull Down Control
20	GP13PU	R/W	0	GPIO Pull Up Control
19:18	GP13DRV	R/W	0	GPIO Driving Control
17	GP13SMT	R/W	1	GPIO SMT Control
16	GP13IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP12FUNC	R/W	4'hE	GPIO Function Select (Default : JTAG)
7:6	RSVD			
5	GP12PD	R/W	0	GPIO Pull Down Control
4	GP12PU	R/W	0	GPIO Pull Up Control
3:2	GP12DRV	R/W	0	GPIO Driving Control
1	GP12SMT	R/W	1	GPIO SMT Control
0	GP12IE	R/W	1	GPIO Input Enable

3.3.12 GPIO_CFGCTL7

Address：0x4000011c

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP15FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP15PD	R/W	0	GPIO Pull Down Control
20	GP15PU	R/W	0	GPIO Pull Up Control
19:18	GP15DRV	R/W	0	GPIO Driving Control
17	GP15SMT	R/W	1	GPIO SMT Control
16	GP15IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP14FUNC	R/W	4'hE	GPIO Function Select (Default : JTAG)
7:6	RSVD			
5	GP14PD	R/W	0	GPIO Pull Down Control
4	GP14PU	R/W	0	GPIO Pull Up Control
3:2	GP14DRV	R/W	0	GPIO Driving Control
1	GP14SMT	R/W	1	GPIO SMT Control
0	GP14IE	R/W	1	GPIO Input Enable

3.3.13 GPIO_CFGCTL8

Address：0x40000120

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP17FUNC	R/W	4'hE	GPIO Function Select (Default : JTAG)
23:22	RSVD			
21	GP17PD	R/W	0	GPIO Pull Down Control
20	GP17PU	R/W	0	GPIO Pull Up Control
19:18	GP17DRV	R/W	0	GPIO Driving Control
17	GP17SMT	R/W	1	GPIO SMT Control
16	GP17IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP16FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP16PD	R/W	0	GPIO Pull Down Control
4	GP16PU	R/W	0	GPIO Pull Up Control
3:2	GP16DRV	R/W	0	GPIO Driving Control
1	GP16SMT	R/W	1	GPIO SMT Control
0	GP16IE	R/W	1	GPIO Input Enable

3.3.14 GPIO_CFGCTL9

Address：0x40000124

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP19FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP19PD	R/W	0	GPIO Pull Down Control
20	GP19PU	R/W	0	GPIO Pull Up Control

Bits	Name	Type	Reset	Description
19:18	GP19DRV	R/W	0	GPIO Driving Control
17	GP19SMT	R/W	1	GPIO SMT Control
16	GP19IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP18FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP18PD	R/W	0	GPIO Pull Down Control
4	GP18PU	R/W	0	GPIO Pull Up Control
3:2	GP18DRV	R/W	0	GPIO Driving Control
1	GP18SMT	R/W	1	GPIO SMT Control
0	GP18IE	R/W	1	GPIO Input Enable

3.3.15 GPIO_CFGCTL10

Address：0x40000128

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP21FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP21PD	R/W	0	GPIO Pull Down Control
20	GP21PU	R/W	0	GPIO Pull Up Control
19:18	GP21DRV	R/W	0	GPIO Driving Control
17	GP21SMT	R/W	1	GPIO SMT Control
16	GP21IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP20FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)

Bits	Name	Type	Reset	Description
7:6	RSVD			
5	GP20PD	R/W	0	GPIO Pull Down Control
4	GP20PU	R/W	0	GPIO Pull Up Control
3:2	GP20DRV	R/W	0	GPIO Driving Control
1	GP20SMT	R/W	1	GPIO SMT Control
0	GP20IE	R/W	1	GPIO Input Enable

3.3.16 GPIO_CFGCTL11

Address：0x4000012c

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP23FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP23PD	R/W	0	GPIO Pull Down Control
20	GP23PU	R/W	0	GPIO Pull Up Control
19:18	GP23DRV	R/W	0	GPIO Driving Control
17	GP23SMT	R/W	1	GPIO SMT Control
16	GP23IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP22FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP22PD	R/W	0	GPIO Pull Down Control
4	GP22PU	R/W	0	GPIO Pull Up Control
3:2	GP22DRV	R/W	0	GPIO Driving Control
1	GP22SMT	R/W	1	GPIO SMT Control

Bits	Name	Type	Reset	Description
0	GP22IE	R/W	1	GPIO Input Enable

3.3.17 GPIO_CFGCTL12

Address：0x40000130

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP25FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP25PD	R/W	0	GPIO Pull Down Control
20	GP25PU	R/W	0	GPIO Pull Up Control
19:18	GP25DRV	R/W	0	GPIO Driving Control
17	GP25SMT	R/W	1	GPIO SMT Control
16	GP25IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP24FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP24PD	R/W	1	GPIO Pull Down Control
4	GP24PU	R/W	0	GPIO Pull Up Control
3:2	GP24DRV	R/W	0	GPIO Driving Control
1	GP24SMT	R/W	1	GPIO SMT Control
0	GP24IE	R/W	1	GPIO Input Enable

3.3.18 GPIO_CFGCTL13

Address：0x40000134

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

Bits	Name	Type	Reset	Description
31:28	RSVD			
27:24	GP27FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	GP27PD	R/W	0	GPIO Pull Down Control
20	GP27PU	R/W	0	GPIO Pull Up Control
19:18	GP27DRV	R/W	0	GPIO Driving Control
17	GP27SMT	R/W	1	GPIO SMT Control
16	GP27IE	R/W	1	GPIO Input Enable
15:12	RSVD			
11:8	GP26FUNC	R/W	4'hB	GPIO Function Select (Default : SWGPIO)
7:6	RSVD			
5	GP26PD	R/W	0	GPIO Pull Down Control
4	GP26PU	R/W	0	GPIO Pull Up Control
3:2	GP26DRV	R/W	0	GPIO Driving Control
1	GP26SMT	R/W	1	GPIO SMT Control
0	GP26IE	R/W	1	GPIO Input Enable

3.3.19 GPIO_CFGCTL14

Address：0x40000138

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

Bits	Name	Type	Reset	Description
31:6	RSVD			
5	GP28PD	R/W	0	GPIO Pull Down Control
4	GP28PU	R/W	0	GPIO Pull Up Control
3:2	GP28DRV	R/W	0	GPIO Driving Control
1	GP28SMT	R/W	1	GPIO SMT Control
0	GP28IE	R/W	1	GPIO Input Enable

4 ADC

4.1 Introduction

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

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

4.2 ADC main features

- High performance
  - 12-bit, 14-bit or 16-bit conversion result output
  - ADC conversion time:fastest 0.5us for 12-bit resolution
  - 1.8V, 3.3V optional reference voltage
  - DMA support
  - Two working modes: single-channel conversion and multi-channel scanning
  - Two input modes: single-ended and differential
  - Support jitter compensation
  - User can set conversion result offset value
  - Scanning mode supports up to 1M, non-scanning mode is 2M
- Analog channels
  - 12 external analog channels

- 2 DAC internal channels
- 1 VBAT / 2 channel
- 1 TSEN channel

4.3 ADC functional description

The basic block diagram of the ADC is shown below.

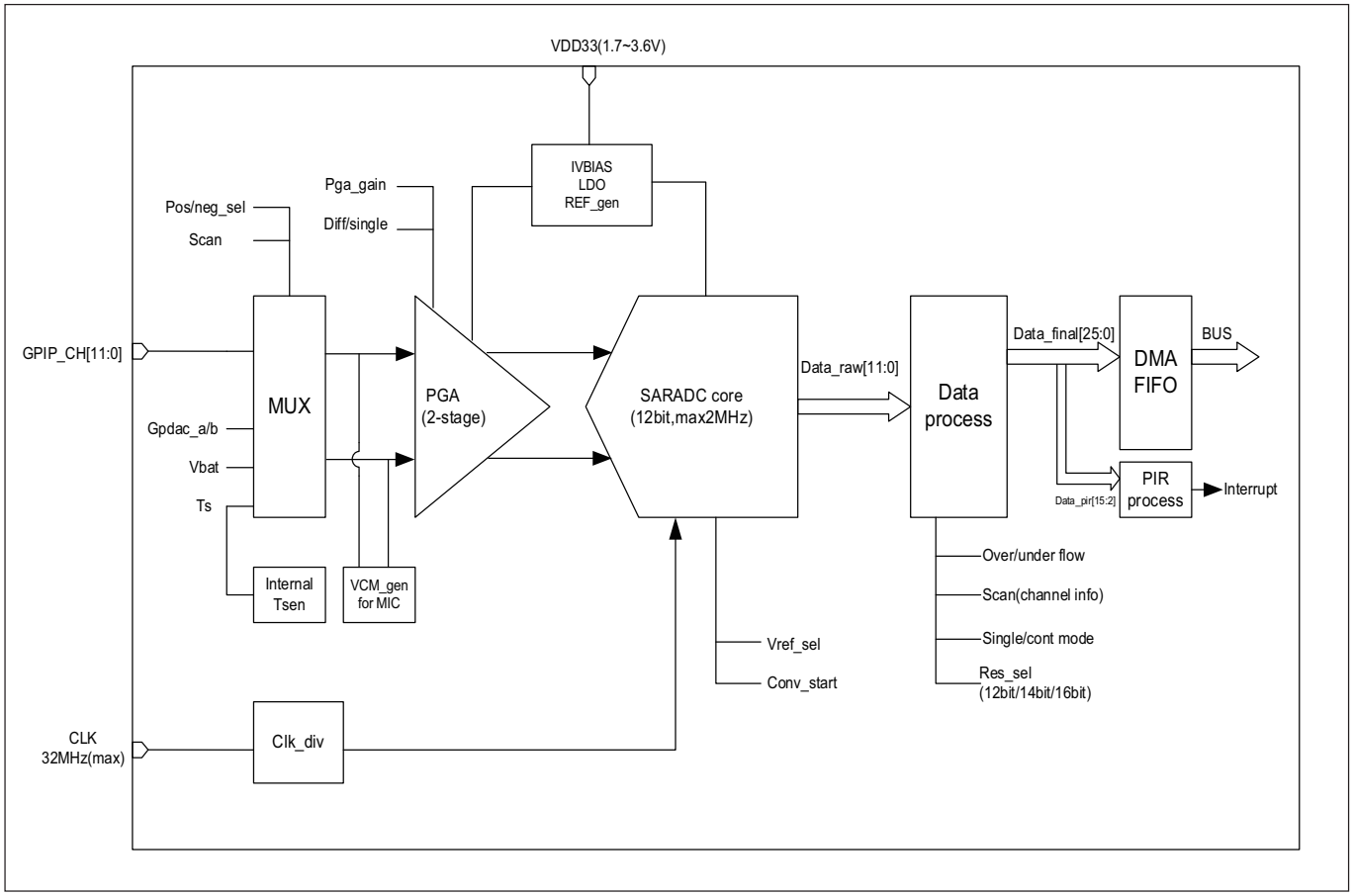


Figure 4.1: ADC block diagram

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

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

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

pushed into the FIFO.

4.3.1 ADC pins and internal signals

Table 4.1: ADC internal signals

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

Table 4.2: ADC external pins

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

4.3.2 ADC channel

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

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

- ADC CH7
- ADC CH8
- ADC CH9
- ADC CH10
- ADC CH11
- DAC OUTA
- DAC OUTB
- VBAT/2
- TSEN
- VREF
- GND

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

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

4.3.3 ADC clock

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

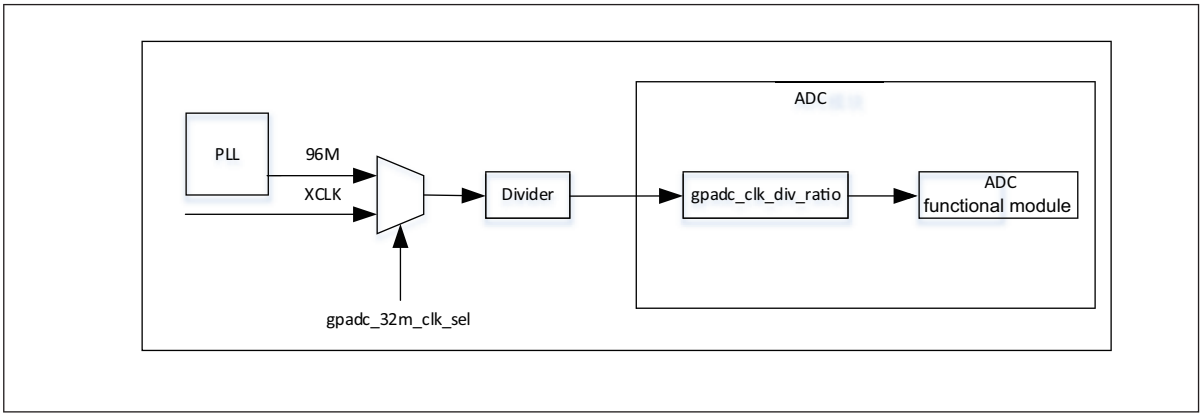


Figure 4.2: ADC Clock

The ADC clock source can select 96M, XTAL or internal RC32M from the PLL. The clock source selection is set in the GLB module. At the same time, the GLB module also provides the clock frequency division. By default, the ADC clock source is 96M. The frequency is 2, and the clock to the ADC module is 32M.

Inside the ADC module, a clock frequency division is provided. The default is 16 frequency division, so the internal

clock of the ADC module is 2M by default. Users can adjust the ADC's clock source and various frequency division coefficients according to actual sampling requirements.

The gpadc_32m_clk_div divider register width is 6 bits, and the maximum divider is 64. Frequency division formula: fout = fsource / (gpadc_32m_clk_div + 1).

The gpadc_clk_div_ratio frequency division register is located inside the ADC module and has a width of 3 bits. The frequency division value is defined as follows:

- 3'b000: div=1
- 3'b001: div=4
- 3'b010: div=8
- 3'b011: div=12
- 3'b100: div=16
- 3'b101: div=20
- 3'b110: div=24
- 3'b111: div=32

4.3.4 ADC conversion mode

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

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

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

ADC conversion results are generally placed in the FIFO. Users need to set the FIFO receive data threshold interrupt based on the actual number of conversion channels. The FIFO threshold interrupt is used as the ADC conversion completion interrupt.

4.3.5 ADC consequence

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

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

Table 4.3: Meaning of ADC conversion result

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

bit21-bit25 of the conversion result is the positive channel number, bit16-bit20 is the negative channel number, and bit0-bit15 is the converted value.

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

The values that can be set are as follows:

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

The ADC conversion result is left-justified. When 12 bits are selected, bit15-bit4 of the conversion result is valid. When 14 bits are selected, bit15-bit2 of the conversion result is valid. When 16 bits are selected, bit15-bit0 of the conversion result is valid.

Similarly, in the differential mode, the highest is the sign, that is, when 14 bits are selected, bit15 is the sign bit, bit14-bit2 is the conversion result, and bit14 is the MSB.

In single-ended mode, there is no sign bit, that is, when 12 bits are selected, bit15-bit4 is the conversion result and bit15 is the MSB.

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

4.3.6 ADC interrupt

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

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

4.3.7 ADC FIFO

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

- FIFO full status
- FIFO is not empty
- FIFO Overrun interrupt
- FIFO Underrun interrupt

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

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

Query mode

The CPU polls the gpadc_rdy bit. When this control bit is set, it indicates that there is valid data in the FIFO. The CPU can obtain the number of FIFO data according to gpadc_fifo_data_count and read these data from the FIFO.

Interrupt mode

The CPU sets gpadc_rdy_mask to 0, and the ADC will generate an interrupt when there is data in the FIFO. The user can use the interrupt function to obtain the number of FIFO data according to gpadc_fifo_data_count and read these data from the FIFO. Then set gpadc_rdy_clr to clear the interrupt.

DMA mode

The user sets the gpadc_dma_en control bit, which can cooperate with DMA to complete the transfer of data to memory. When using the DMA mode, the gpadc_fifo_thl is used to set the threshold of the number of data sent by the ADC FIFO by the FIFO. When the DMA receives the request, it will automatically transfer the specified number of results from the FIFO to the corresponding memory according to the parameters set by the user.

4.3.8 ADC configuration process

Setting the ADC clock

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

Set GPIO according to the channel used

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

Set the channel to be converted

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

For single-channel conversion, set the converted channel information in the gpadc_pos_sel and gpadc_neg_sel registers.

For multi-channel scanning mode, set gpadc_scan_length, gpadc_reg_scn_posX and gpadc_reg_scn_negX according to the number of scanning channels and scanning order.

Set the data reading method

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

Start conversion

Finally set gpadc_res_sel to select the precision of the data conversion result. Finally set gpadc_global_en = 1 and gpadc_conv_start = 1 to start the ADC to start conversion.

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

4.3.9 VBAT measurement

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

The VBAT/2 voltage measured by the ADC module is after a partial pressure.The actual input voltage to the ADC module is half of VDD33, that is, VBAT/2 = VDD33/2. Because the voltage is divided, in order to obtain higher accuracy, it is recommended that the reference voltage of the ADC is 1.8V, single-ended mode is used, the positive input voltage is VBAT/2, the negative input voltage is GND, and Gpadc_vbat_en is set to 1 to start.

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

4.3.10 TSEN measurement

The ADC can measure the internal diode or external diode voltage value, and the voltage difference between the diode and temperature is related, so by measuring the voltage of the diode, the ambient temperature can be calculated. We

call it Temperature Sensor, referred to as TSEN.

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

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

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

When using TSEN, it is recommended to set the ADC to 16bits mode, reduce the error by multiple sampling, and select 1.8V as the reference voltage to improve accuracy. Set gpadc_ts_en to 1 to enable the TSEN function. If the internal diode is selected, gpadc_tsext_sel = 0. External diode, gpadc_tsext_sel = 1, select the forward input channel according to the actual situation.

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

4.4 Register description

Name	Description
gpadc_config	GPADC configuration
gpadc_dma_rdata	GPADC DMA read data
gpadc_reg_cmd	GPADC configuration register
gpadc_reg_config1	GPADC configuration register1
gpadc_reg_config2	GPADC configuration register2
gpadc_reg_scn_pos1	GPADC converation sequence 1
gpadc_reg_scn_pos2	GPADC converation sequence 2
gpadc_reg_scn_neg1	GPADC converation sequence 3
gpadc_reg_scn_neg2	GPADC converation sequence 4
gpadc_reg_status	GPADC status register
gpadc_reg_isr	GPADC status flag register
gpadc_reg_result	GPADC result register

Name	Description
gpadc_reg_raw_result	GPADC raw result register
gpadc_reg_define	GPADC define register

4.4.1 gpadc_config

Address：0x40002000

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

Bits	Name	Type	Reset	Description
31:24	RSVD			
23:22	FIFOTHL	R/W	2'd0	fifo threshold 2'b00: 1 data 2'b01: 4 data 2'b10: 8 data 2'b11: 16 data
21:16	FIFODACN	R	6'd0	fifo data number
15	RSVD			
14	FURM	R/W	1'b0	write 1 mask
13	FORM	R/W	1'b0	write 1 mask
12	RDYM	R/W	1'b0	write 1 mask
11	RSVD			
10	URCL	R/W	1'b0	Write 1 to clear flag
9	ORCL	R/W	1'b0	Write 1 to clear flag
8	RDYCLR	R/W	1'b0	Write 1 to clear flag
7	RSVD			
6	FIFOUR	R	1'b0	FIFO underrun interrupt flag
5	FIFOOR	R	1'b0	FIFO overrun interrupt flag
4	RDY	R	1'b0	Conversion data ready interrupt flag
3	FIFOFULL	R	1'b0	FIFO full flag

Bits	Name	Type	Reset	Description
2	FIFONE	R	1'b0	FIFO not empty flag
1	FIFOCLR	W1C	1'b0	FIFO clear signal
0	DMAEN	R/W	1'b0	GPADC DMA enbale

4.4.2 gpadc_dma_rdata

Address：0x40002004

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

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

4.4.3 gpadc_reg_cmd

Address：0x4000f90c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD	STEN	SENSEL		CSPU	RSVD			MBEN	MPG		M1D	M2D	DWEN	RSVD	BMB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPEN	MBIEN	NG	POSSEL					NEGSEL					SRST	CSTA	GEN

Bits	Name	Type	Reset	Description
31	RSVD			
30	STEN	R/W	1'b0	enable sensor dc test mux
29:28	SENSEL	R/W	2'h0	selected output current channel and measurement channel 2'h0: 1st channel 2'h1: 2nd channel 2'h2: 3rd channel 2'h3: 4th channel

Bits	Name	Type	Reset	Description
27	CSPU	R/W	1'b0	enable chip sensor test 1'b0: disable 1'b1: enable
26:24	RSVD			
23	MBEN	R/W	1'b0	micboost 32db enable 1'b0: 16dB 1'b1: 32dB
22:21	MPG	R/W	2'h0	mic_pga2_gain 2'h0: 0dB 2'h1: 6dB 2'h2: -6dB 2'h3: 12dB
20	M1D	R/W	1'b0	mic1 diff enable 1'b0: single 1'b1: diff
19	M2D	R/W	1'b0	mic2 diff enable 1'b0: single 1'b1: diff
18	DWEN	R/W	1'b0	dwa enable 1'b0: dwa disable 1'b1: dwa enable
17	RSVD			
16	BMB	R/W	1'b0	micboost amp bypass 1'b0: not bypass 1'b1: bypass
15	MPEN	R/W	1'b0	micpga enable 1'b0: micpga disable 1'b1: miapga enable
14	MBIEN	R/W	1'b0	enable micbias 1'b0: micbias power down 1'b1: miabias power on
13	NG	R/W	1'b0	set negative input of adc to ground 1'b0: disable 1'b1: enable

Bits	Name	Type	Reset	Description
12:8	POSSEL	R/W	5'hf	select adc positive input in none-scan mode 5'h0 gpip_ch[0] 5'h1 gpip_ch[1] 5'h2 gpip_ch[2] 5'h3 gpip_ch[3] 5'h4 gpip_ch[4] 5'h5 gpip_ch[5] 5'h6 gpip_ch[6] 5'h7 gpip_ch[7] 5'h8 gpip_ch[8] 5'h9 gpip_ch[9] 5'h10 gpip_ch[10] 5'h11 gpip_ch[11] 5'h12 daca 5'h13 dacb 5'h14 temp_p 5'h16 vref 5'h18 vbat/2 5'h23 31 avss
7:3	NEGSEL	R/W	5'hf	select adc negative input in none-scan mode 5'h0 gpip_ch[0] 5'h1 gpip_ch[1] 5'h2 gpip_ch[2] 5'h3 gpip_ch[3] 5'h4 gpip_ch[4] 5'h5 gpip_ch[5] 5'h6 gpip_ch[6] 5'h7 gpip_ch[7] 5'h8 gpip_ch[8] 5'h9 gpip_ch[9] 5'h10 gpip_ch[10] 5'h11 gpip_ch[11] 5'h12 daca 5'h13 dacb 5'h14 temp_p 5'h16 vref 5'h18 vbat/2 5'h23 31 avss
2	SRST	R/W	1'b0	user reset the whole block 1'h0: not reset 1'h1: reset
1	CSTA	R/W	1'b0	1'h0: stop converation 1'h1: start converation
0	GEN	R/W	1'b0	1'h0: disable ADC 1'h1: enable ADC

4.4.4 gpadc_reg_config1

Address：0x4000f910

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD	V18SEL		V11SEL		DTEN	SCEN	SCLEN				CDRD			CAIV	RSVD
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											RSSEL			CTCV	OCEN

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

Bits	Name	Type	Reset	Description
24:21	SCLEN	R/W	4'h0	select scan mode length 4'b0000 : select gpadc_scan_pos_0 and gpadc_scan_-neg_0 4'b0001 : select gpadc_scan_pos_1 and gpadc_scan_-neg_1 4'b0010 : select gpadc_scan_pos_2 and gpadc_scan_-neg_2 4'b0011 : select gpadc_scan_pos_3 and gpadc_scan_-neg_3 4'b0100 : select gpadc_scan_pos_4 and gpadc_scan_-neg_4 4'b0101 : select gpadc_scan_pos_5 and gpadc_scan_-neg_5 4'b0110 : select gpadc_scan_pos_6 and gpadc_scan_-neg_6 4'b0111 : select gpadc_scan_pos_7 and gpadc_scan_-neg_7 4'b1000 : select gpadc_scan_pos_8 and gpadc_scan_-neg_8 4'b1001 : select gpadc_scan_pos_9 and gpadc_scan_-neg_9 4'b1010 : select gpadc_scan_pos_10 and gpadc_scan_-neg_10 4'b1011 : select gpadc_scan_pos_11 and gpadc_scan_-neg_11
20:18	CDRD	R/W	3'h3	analog 32M clock division ratio 3'b000: div=1 3'b001: div=4 3'b010: div=8 3'b011: div=12 3'b100: div=16 3'b101: div=20 3'b110: div=24 3'b111: div=32
17	CAIV	R/W	1'b0	analog clock 2M inverted
16:5	RSVD			

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

4.4.5 gpadc_reg_config2

Address：0x4000f914

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDCR	DLYSEL			PGA1GAIN			PGA2GAIN			TESTSEL			ATEN	BSEL	CM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CM	PCEN	PEN	POCAL				PVCM		TSEN	DDM	VBEN	VRFS	DIFM	RSVD	

Bits	Name	Type	Reset	Description
31	TDCR	R/W	1'b0	tsen diode current
30:28	DLYSEL	R/W	3'h0	adc conversion speed
27:25	PGA1GAIN	R/W	3'h0	3'h0: disable 3'h1: gain=1 3'h2: gain=2 3'h3: gain=4 3'h4: gain=8 3'h5: gain=16 3'h6: gain=32 3'h7: gain=32
24:22	PGA2GAIN	R/W	3'h0	3'h0: disable 3'h1: gain=1 3'h2: gain=2 3'h3: gain=4 3'h4: gain=8 3'h5: gain=16 3'h6: gain=32 3'h7: gain=32
21:19	TESTSEL	R/W	3'h0	select test point 0 7

Bits	Name	Type	Reset	Description
18	ATEN	R/W	1'b0	Analog test enable.
17	BSEL	R/W	1'b0	adc analog portion low power mode select 1'h0: Full biasing current 1'h1: Half biasing current
16:15	CM	R/W	2'h3	2'b11 all off 2'b11 Vref AZ on 2'b11 Vref AZ and PGA chop on 2'b11 Vref AZ and PGA chop+RPC on
14	PCEN	R/W	1'b0	enable pga input vcm bias
13	PEN	R/W	1'b0	1'h0: disable PGA 1'h1 enable PGA
12:9	POCAL	R/W	4'h8	pga offset calibration
8:7	PVCM	R/W	2'h2	Audio PGA output common mode control 2'b00: cm=1V 2'b11: cm=1.2V 2'b11: cm=1.4V 2'b11: cm=1.6V
6	TSEN	R/W	1'b0	1'h0: disable temperature sensor 1'h1: enable temperature sensor
5	DDM	R/W	1'b0	1'h0: internal diode mode 1'h1: external diode mode
4	VBEN	R/W	1'b0	1'h0: disable VBAT sensor 1'h1 enable VBAT sensor
3	VRFS	R/W	1'b0	ADC reference select 1'h0 3.3V 1'h1 1.8V
2	DIFM	R/W	1'b0	1'h0 single-ended 1'h1 differential
1:0	RSVD			

4.4.6 gpadc_reg_scn_pos1

Address：0x4000f918

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SCANP5					SCANP4					SCAN3			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCAN3		SCANP2					SCANP1					SCANP0			

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:25	SCANP5	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
24:20	SCANP4	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
19:15	SCAN3	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
14:10	SCANP2	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
9:5	SCANP1	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
4:0	SCANP0	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel

4.4.7 gpadc_reg_scn_pos2

Address：0x4000f91c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SCANP11					SCANP10					SCAN9			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCAN9		SCANP8					SCANP7					SCANP6			

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:25	SCANP11	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
24:20	SCANP10	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
19:15	SCAN9	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
14:10	SCANP8	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
9:5	SCANP7	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
4:0	SCANP6	R/W	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel

4.4.8 gpadc_reg_scn_neg1

Address：0x4000f920

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SCANN5					SCANN4					SCAN3			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCAN3		SCANN2					SCANN1					SCANN0			

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:25	SCANN5	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
24:20	SCANN4	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
19:15	SCAN3	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
14:10	SCANN2	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
9:5	SCANN1	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
4:0	SCANN0	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel

4.4.9 gpadc_reg_scn_neg2

Address：0x4000f924

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		SCANN11					SCANN10					SCAN9			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCAN9		SCANN8					SCANN7					SCANN6			

Bits	Name	Type	Reset	Description
31:30	RSVD			
29:25	SCANN11	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
24:20	SCANN10	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
19:15	SCAN9	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
14:10	SCANN8	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
9:5	SCANN7	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
4:0	SCANN6	R/W	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel

4.4.10 gpadc_reg_status

Address：0x4000f928

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

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

4.4.11 gpadc_reg_isr

Address：0x4000f92c

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

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

4.4.12 gpadc_reg_result

Address：0x4000f930

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

Bits	Name	Type	Reset	Description
31:26	RSVD			

Bits	Name	Type	Reset	Description
25:0	DATAOUT	R	26'h1EF0000	ADC finial conversion result data, after calibration and signed/unsigned process

4.4.13 gpadc_reg_raw_result

Address：0x4000f934

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

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

4.4.14 gpadc_reg_define

Address：0x4000f938

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

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

5 DAC

5.1 Introduction

The chip has a built-in 10bits digital-to-analog converter (DAC) with a FIFO depth of 1, and supports 2 DAC modulation outputs.

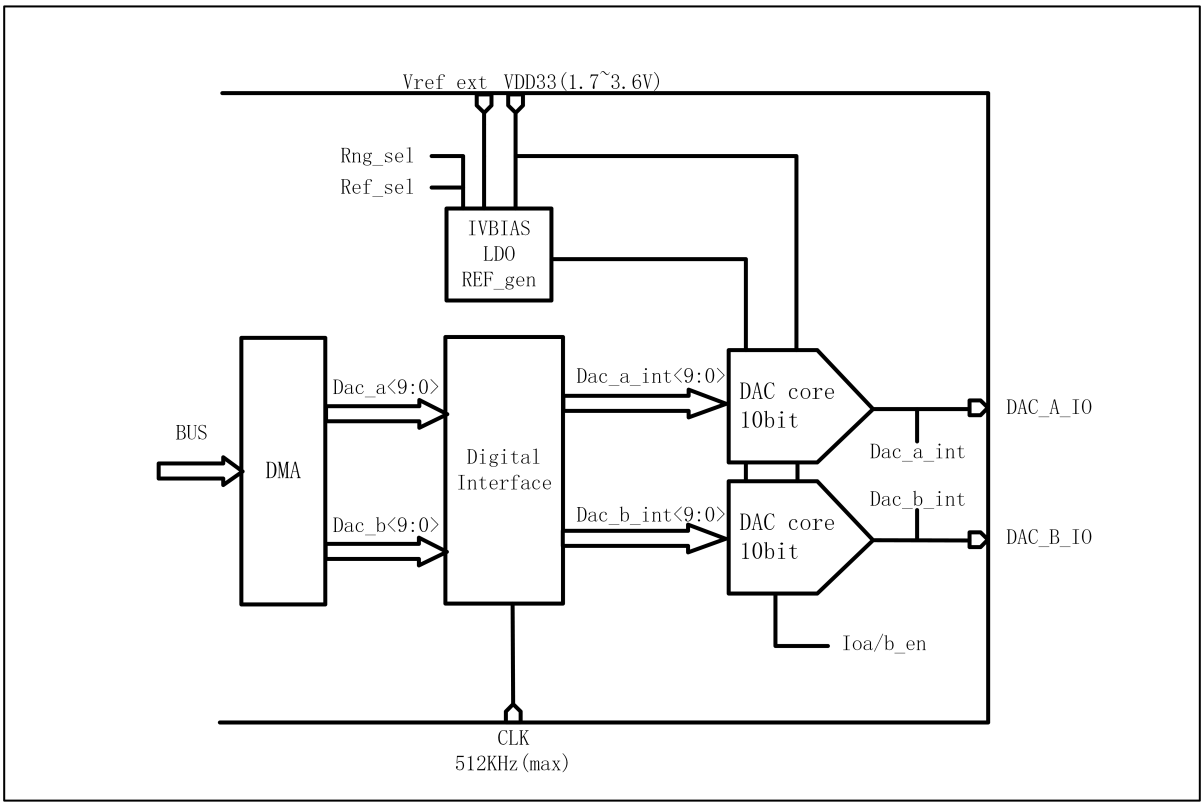
Can be used for audio playback, transmitter voltage modulation.

5.2 Main features

- DAC modulation accuracy is 10-bits
- DAC input clock can be selected as 32k, 16k, 8k or 512k
- Support DMA to transfer memory to DAC modulation register
- Support dual channel playback DMA transport mode
- The output pin of DAC is fixed to ChannelA as GPIO13, Channel as GPIO14

5.3 Function description

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



- DAC module supports up to two modulation outputs
- DAC module supports dual-channel DMA data transfer mode
- DAC module supports a DMA data interface with a length of 32-bit, in which the high 16 bits will be modulated on the pins of ChannelA, and the low 16 bits will be modulated on the pins of ChannelB

5.4 Register description

Name	Description
gpdac_config	GPDAC configuration
gpdac_dma_config	GPDAC DMA configuration
gpdac_dma_wdata	GPDAC DMA write data

5.4.1 gpdac_config

Address：0x40002040

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

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

5.4.2 gpdac_dma_config

Address：0x40002044

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

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

5.4.3 gpdac_dma_wdata

Address：0x40002048

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

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

6 DMA

6.1 Introduction

DMA (Direct Memory Access) is a memory access technology that can independently read and write system memory directly without processor intervention. Under the same degree of processor load, DMA is a fast data transfer method. The DMA controller has 4 channels, which manage the data transfer between peripheral devices and memory to improve bus efficiency.

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

6.2 DMA main features

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

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

6.3 DMA functional description

6.3.1 DMA transactions

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

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

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

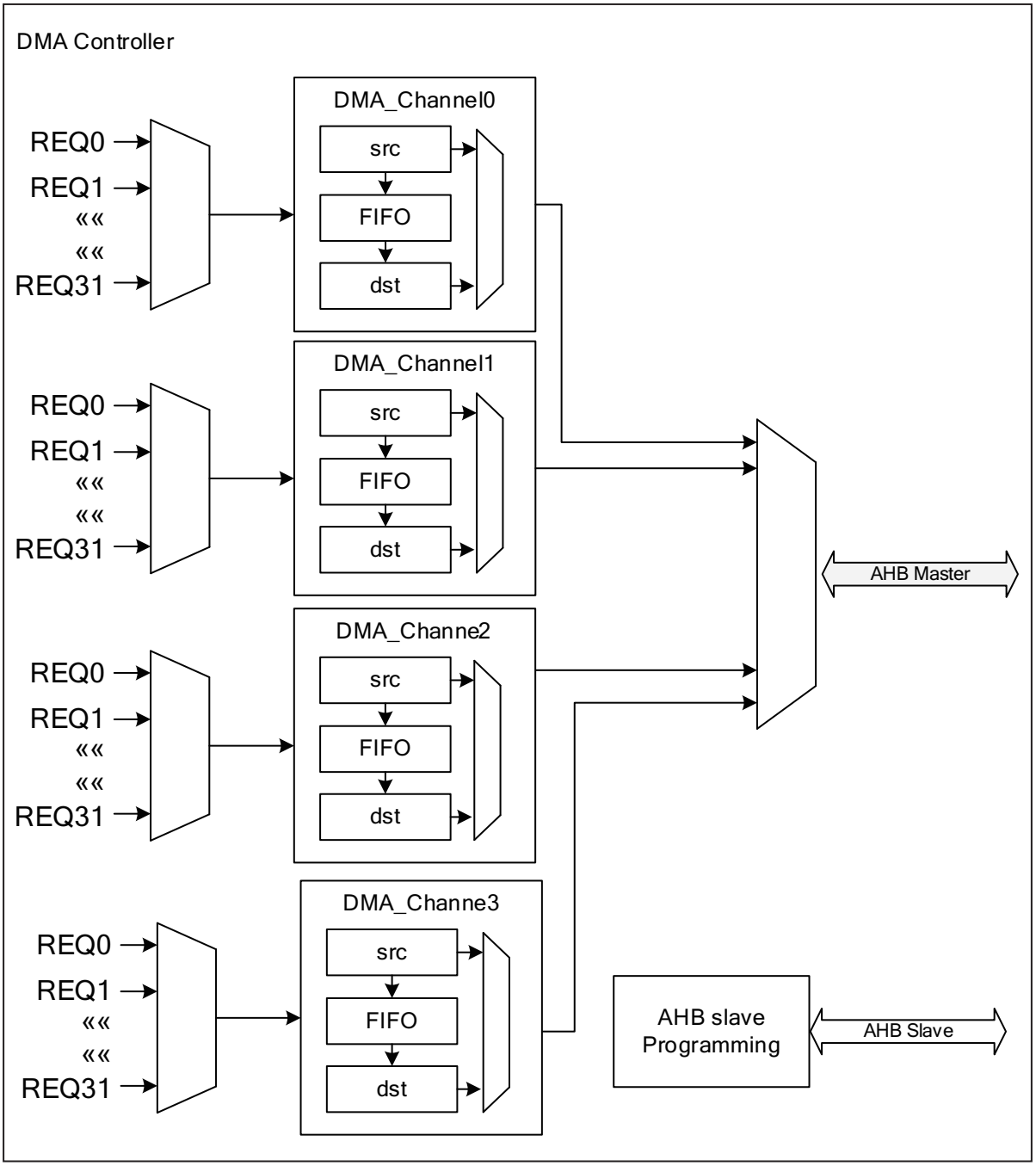


Figure 6.1: DMA architecture

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

6.3.2 DMA channel configuration

DMA supports 4 channels in total, each channel does not interfere with each other and can run at the same time. The following is the configuration process of DMA channel x:

1. Set 32-bit source address in DMA_C0SrcAddr register
2. Set the 32-bit target address in the DMA_C0DstAddr register
3. Configure SI (source) and DI (destination) in the DMA_C0Control register to set whether to enable the automatic address accumulation mode. When set to 1, enable the automatic address accumulation mode
4. Set the transmission data width by configuring the STW (source) and DTW (destination) bits in the DMA_C0Control register. The width options are single-byte, double-self-knot, and four-byte.
5. Burst type, which can be set by configuring the SBS (source) and DBS (destination) bits in the DMA_C0Control register. The configuration options are Single, INCR4, INCR8, INCR16
6. Special attention should be paid to the configured combination. A single burst cannot exceed 16 bytes.
7. Set the data transmission length range: 0-4095

6.3.3 Peripheral support

The SrcPeripheral (source) and DstPeripheral (destination) are configured to determine the peripherals that the current DMA cooperates with. The relationship is 0-3: UART / 6-7: I2C / 10-11: SPI / 22-23: ADC / DAC

UART uses DMA to transfer data

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

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

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

I2C uses DMA to transfer data

The configuration is as follows:

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

SPI uses DMA to transfer data

The configuration is as follows:

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

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

ADC0/1 uses DMA to transfer data

The configuration is as follows:

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

6.3.4 Linked List Mode

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

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

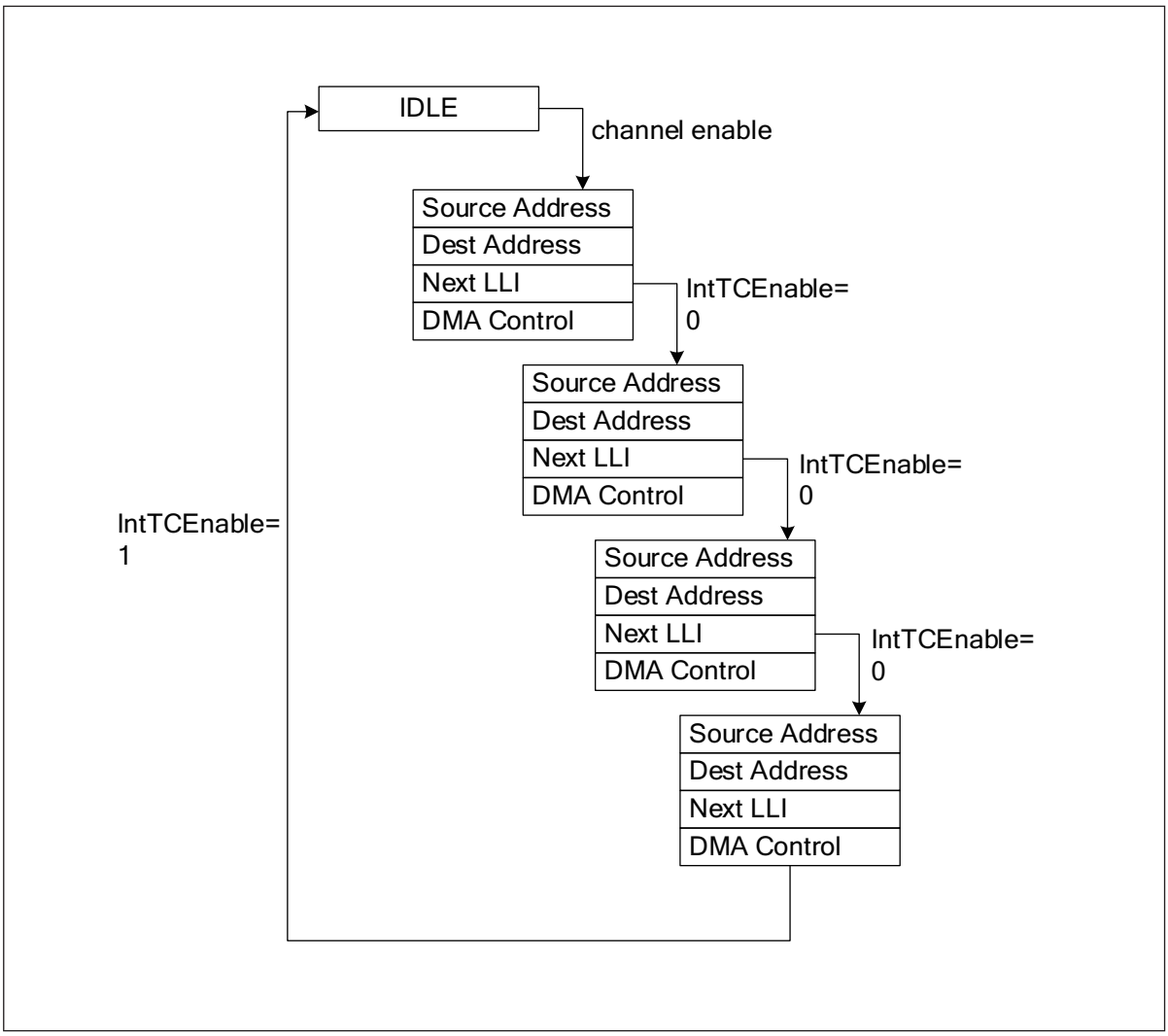


Figure 6.2: LLI architecture

6.3.5 DMA interrupt

- DMA_INT_TCOMPLETED
  - Data transmission completed interrupt. When a data transmission is completed, this interrupt will be entered.
- DMA_INT_ERR
  - Data transmission error interrupt, when an error occurs during data transmission, this interrupt will be entered

6.4 Transmission mode

6.4.1 Memory to memory

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

The specific configuration process is as follows:

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

6.4.2 Memory to peripheral

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

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

The specific configuration process is as follows:

1. Set the value of the register DMA_C0SrcAddr to the memory address of the source
2. Set the value of the register DMA_C0DstAddr to the target peripheral address
3. Select the transfer mode and set the value of the DMA_C0Config [FLOWCTRL] bit to 1 to select the memory-to-

peripheral mode

4. Set the value of the corresponding bit in the DMA_C0Control register: set the DI and SI bits to 1 to enable the automatic address accumulation mode, the DTW and STW bits set the transmission width of the source and destination, and the DBS and SBS bits set the burst type of the source and destination
5. Select the appropriate channel, enable DMA, and complete the data transfer

6.4.3 Peripheral to memory

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

The specific configuration process is as follows:

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

6.5 Register description

Name	Description
DMA_IntStatus	Interrupt status
DMA_IntTCStatus	Interrupt terminal count request status
DMA_IntTCClear	Terminal count request clear
DMA_IntErrorStatus	Interrupt error status
DMA_IntErrClr	Interrupt error clear
DMA_RawIntTCStatus	Status of the terminal count interrupt prior to masking
DMA_RawIntErrorStatus	Status of the error interrupt prior to masking
DMA_EnbldChns	Channel enable status
DMA_SoftBReq	Software burst request

Name	Description
DMA_SoftSReq	Software single request
DMA_SoftLBReq	Software last burst request
DMA_SoftLSReq	Software last single request
DMA_Config	DMA general configuration
DMA_Sync	DMA request asynchronous setting
DMA_C0SrcAddr	Channel DMA source address
DMA_C0DstAddr	Channel DMA Destination address
DMA_C0LLI	Channel DMA link list
DMA_C0Control	Channel DMA bus control
DMA_C0Config	Channel DMA configuration
DMA_C1SrcAddr	Channel DMA source address
DMA_C1DstAddr	Channel DMA Destination address
DMA_C1LLI	Channel DMA link list
DMA_C1Control	Channel DMA bus control
DMA_C1Config	Channel DMA configuration
DMA_C2SrcAddr	Channel DMA source address
DMA_C2DstAddr	Channel DMA Destination address
DMA_C2LLI	Channel DMA link list
DMA_C2Control	Channel DMA bus control
DMA_C2Config	Channel DMA configuration
DMA_C3SrcAddr	Channel DMA source address
DMA_C3DstAddr	Channel DMA Destination address
DMA_C3LLI	Channel DMA link list
DMA_C3Control	Channel DMA bus control
DMA_C3Config	Channel DMA configuration

6.5.1 DMA_IntStatus

Address：0x4000c000

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

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

6.5.2 DMA_IntTCStatus

Address：0x4000c004

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

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

6.5.3 DMA_IntTCClear

Address：0x4000c008

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

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

6.5.4 DMA_IntErrorStatus

Address：0x4000c00c

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

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

6.5.5 DMA_IntErrClr

Address：0x4000c010

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

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

6.5.6 DMA_RawIntTCStatus

Address：0x4000c014

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

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

Bits	Name	Type	Reset	Description

6.5.7 DMA_RawIntErrorStatus

Address：0x4000c018

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

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

6.5.8 DMA_EnbldChns

Address：0x4000c01c

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

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

6.5.9 DMA_SoftBReq

Address：0x4000c020

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

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

6.5.10 DMA_SoftSReq

Address：0x4000c024

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

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

6.5.11 DMA_SoftLBReq

Address：0x4000c028

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

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

6.5.12 DMA_SoftLSReq

Address：0x4000c02c

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

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

6.5.13 DMA_Config

Address：0x4000c030

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

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

6.5.14 DMA_Sync

Address：0x4000c034

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

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

6.5.15 DMA_C0SrcAddr

Address：0x4000c100

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

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

6.5.16 DMA_C0DstAddr

Address：0x4000c104

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

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

6.5.17 DMA_C0LLI

Address：0x4000c108

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

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

6.5.18 DMA_C0Control

Address：0x4000c10c

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

Bits	Name	Type	Reset	Description
31	TCIEN	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROTECT	R/W	0	Protection.
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25	RSVD			
24	IMTMMODE	R/W	0	In Memory-to-memory mode, Set this bit high when Src data size is larger than Dst.
23:21	DTW	R/W	3'b010	Destination transfer width: 8/16/32
20:18	STW	R/W	3'b010	Source transfer width: 8/16/32
17:15	DBS	R/W	3'b001	Destination burst size: 1/4/8/16
14:12	SBS	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TS	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.19 DMA_C0Config

Address：0x4000c110

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

Bits	Name	Type	Reset	Description
31:30	RSVD			

Bits	Name	Type	Reset	Description
29:20	LLICOUNT	R	0	LLI counter. Increased 1 each LLI run. Cleared 0 when config Control.
19	RSVD			
18	HALT	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	ACTIVE	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	LOCK	R/W	0	Lock.
15	TCIM	R/W	0	Terminal count interrupt mask.
14	IEM	R/W	0	Interrupt error mask.
13:11	FLOWCTRL	R/W	0	000: Memory-to-memory (DMA) 001: Memory-to-peripheral (DMA) 010: Peripheral-to-memory (DMA) 011: Source peripheral-to-Destination peripheral (DMA) 100: Source peripheral-to-Destination peripheral (Destination peripheral) 101: Memory-to-peripheral (peripheral) 110: Peripheral-to-memory (peripheral) 111: Source peripheral-to-Destination peripheral (Source peripheral)
10:6	DSTPH	R/W	0	Destination peripheral. [23:22] DAC/ADC [11:10] SPI TX/RX [7: 6] I2C TX/RX [3: 0] UART1 TX/RX ; UART0 TX/RX
5:1	SRCPH	R/W	0	Source peripheral.
0	CHEN	R/W	0	Channel enable.

6.5.20 DMA_C1SrcAddr

Address：0x4000c200

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

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

6.5.21 DMA_C1DstAddr

Address：0x4000c204

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

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

6.5.22 DMA_C1LLI

Address：0x4000c208

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

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

6.5.23 DMA_C1Control

Address：0x4000c20c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I	PROT			DI	SI	RSVD		DWIDTH			SWIDTH			DBSIZE	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBSIZE	SBSIZE			TRANSIZE											

Bits	Name	Type	Reset	Description
31	I	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROT	R/W	0	Protection.
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25:24	RSVD			
23:21	DWIDTH	R/W	3'b010	Destination transfer width: 8/16/32
20:18	SWIDTH	R/W	3'b010	Source transfer width: 8/16/32
17:15	DBSIZE	R/W	3'b001	Destination burst size: 1/4/8/16
14:12	SBSIZE	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TRANSIZE	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.24 DMA_C1Config

Address：0x4000c210

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD													H	A	L
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITC	IE	FLOWCTRL			DSTPH					SRCPH					E

Bits	Name	Type	Reset	Description
31:19	RSVD			
18	H	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	A	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	L	R/W	0	Lock.
15	ITC	R/W	0	Terminal count interrupt mask.
14	IE	R/W	0	Interrupt error mask.

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

6.5.25 DMA_C2SrcAddr

Address：0x4000c300

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

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

6.5.26 DMA_C2DstAddr

Address：0x4000c304

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

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

6.5.27 DMA_C2LLI

Address：0x4000c308

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

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

6.5.28 DMA_C2Control

Address：0x4000c30c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I	PROT			DI	SI	RSVD		DWIDTH			SWIDTH			DBSIZE	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBSIZE	SBSIZE			TRANSIZE											

Bits	Name	Type	Reset	Description
31	I	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROT	R/W	0	Protection.
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.

Bits	Name	Type	Reset	Description
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25:24	RSVD			
23:21	DWIDTH	R/W	3'b010	Destination transfer width: 8/16/32
20:18	SWIDTH	R/W	3'b010	Source transfer width: 8/16/32
17:15	DBSIZE	R/W	3'b001	Destination burst size: 1/4/8/16
14:12	SBSIZE	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TRANSIZE	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.29 DMA_C2Config

Address：0x4000c310

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD													H	A	L
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITC	IE	FLOWCTRL			DSTPH					SRCPH					E

Bits	Name	Type	Reset	Description
31:19	RSVD			
18	H	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	A	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	L	R/W	0	Lock.
15	ITC	R/W	0	Terminal count interrupt mask.
14	IE	R/W	0	Interrupt error mask.

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

6.5.30 DMA_C3SrcAddr

Address：0x4000c400

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

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

6.5.31 DMA_C3DstAddr

Address：0x4000c404

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

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

6.5.32 DMA_C3LLI

Address：0x4000c408

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

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

6.5.33 DMA_C3Control

Address：0x4000c40c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I	PROT			DI	SI	RSVD		DWIDTH			SWIDTH			DBSIZE	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBSIZE	SBSIZE			TRANSIZE											

Bits	Name	Type	Reset	Description
31	I	R/W	0	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROT	R/W	0	Protection.
27	DI	R/W	1	Destination increment. When set, the Destination address is incremented after each transfer.

Bits	Name	Type	Reset	Description
26	SI	R/W	1	Source increment. When set, the source address is incremented after each transfer.
25:24	RSVD			
23:21	DWIDTH	R/W	3'b010	Destination transfer width: 8/16/32
20:18	SWIDTH	R/W	3'b010	Source transfer width: 8/16/32
17:15	DBSIZE	R/W	3'b001	Destination burst size: 1/4/8/16
14:12	SBSIZE	R/W	3'b001	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TRANSIZE	R/W	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

6.5.34 DMA_C3Config

Address：0x4000c410

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD													H	A	L
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITC	IE	FLOWCTRL			DSTPH					SRCPH					E

Bits	Name	Type	Reset	Description
31:19	RSVD			
18	H	R/W	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	A	R	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	L	R/W	0	Lock.
15	ITC	R/W	0	Terminal count interrupt mask.
14	IE	R/W	0	Interrupt error mask.

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

7 L1C

7.1 Introduction

The L1 Cache Controller is a unit module that is located outside the processor and is used to manage code or data buffers on the Flash and increase the speed of the CPU accessing the Flash. The architecture is as follows:

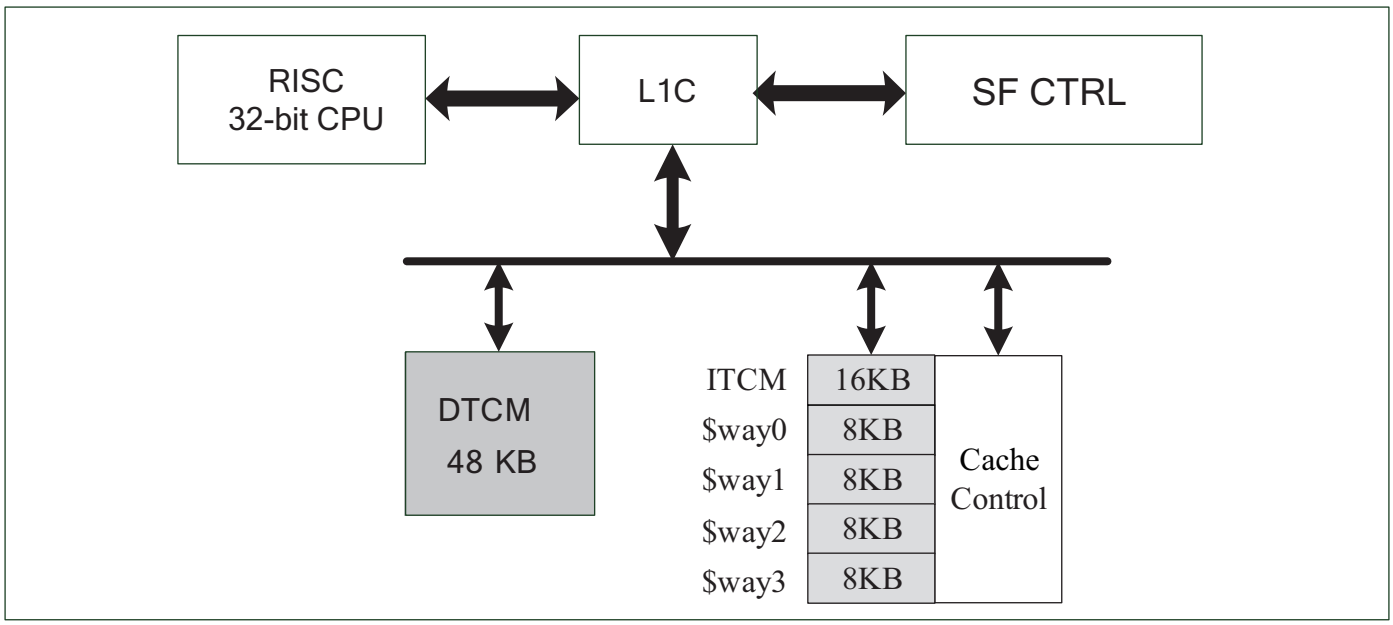


Figure 7.1: L1c architecture

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

7.2 Main features

- 4-way Set-Associative mapping
- Variable cache size
- Connect to TCM address space, can easily configure L1C space as TCM space
- Support cache performance statistics

7.3 Function description

7.3.1 Mutual conversion between TCM and Cache RAM resources

In order to increase the memory usage efficiency, the 32K RAM of the cache can be fully or partially adjusted to the TCM space, which is convenient for users to adjust the memory usage method and efficiency according to the actual situation.

The default size of the cache is 32K, divided into 4 ways, each way is 8K, the unit of adjustment is 1 way, which is 8K. The default size of ITCM is 16K. Through the setting of WayDisable, you can flexibly adjust the actual space size of Cache and ITCM.

Table 7.1: WayDisable settings

WayDisable	Cache	ITCM
none	32K	16K
one way	24K	24K
two way	16K	32K
three way	8K	40K
four way	0K	48K

7.3.2 Cache

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

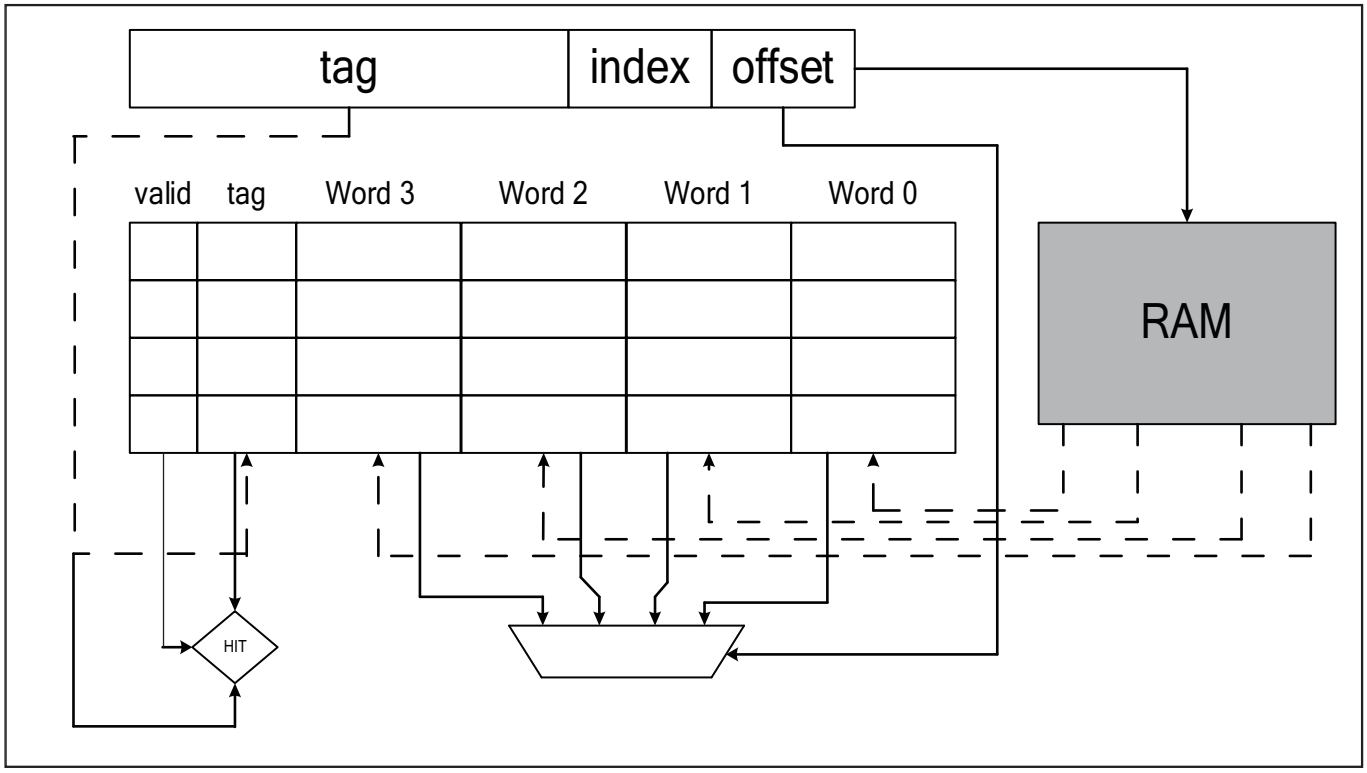


Figure 7.2: Cache architecture

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

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

7.4 Register description

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

7.4.1 l1c_config

Address：0x40009000

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

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

7.4.2 hit_cnt_lsb

Address：0x40009004

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

Bits	Name	Type	Reset	Description
31:0	CNTLSB	R	0	Hit counter low 32-bit

7.4.3 hit_cnt_msb

Address：0x40009008

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

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

7.4.4 miss_cnt

Address：0x4000900c

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

Bits	Name	Type	Reset	Description
31:0	MISSCNT	R	0	Miss counter

8 IR

8.1 Introduction

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

8.2 IR main features

- Receiving data with NEC, RC-5 protocol
- Receiving arbitrary format data in pulse width counting mode
- Powerful infrared waveform editing capabilities, which can emit waveforms conforming to various protocols
- Power settings of up to 15 gears to suit different power requirements
- Supports up to 64-bit data bits
- 64-byte receive FIFO
- Programmable carrier frequency and duty cycle
- The maximum operating frequency is 32MHz

8.3 Function description

8.3.1 Fixed receiving protocol

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

- NEC protocol

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

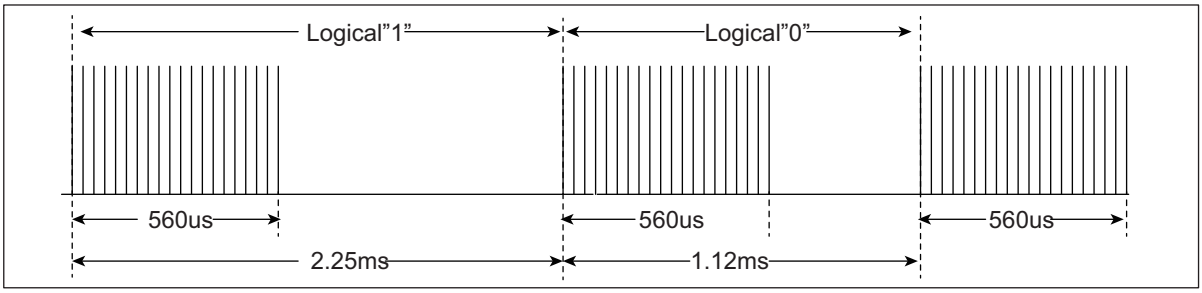


Figure 8.1: nec logical

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

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

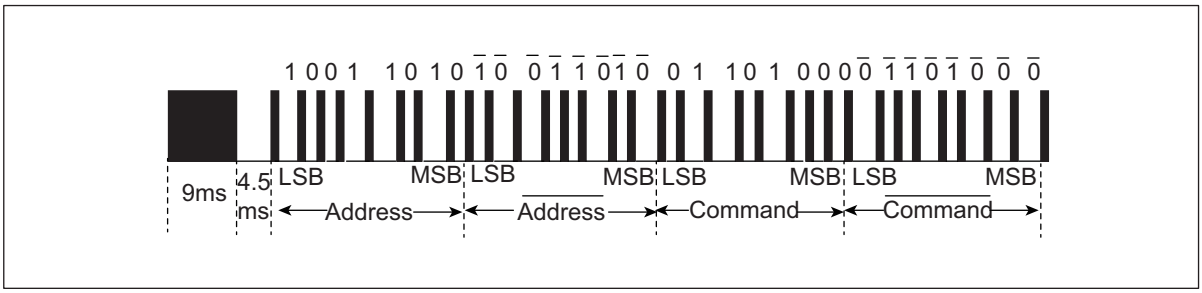


Figure 8.2: nec

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

- RC-5 protocol

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

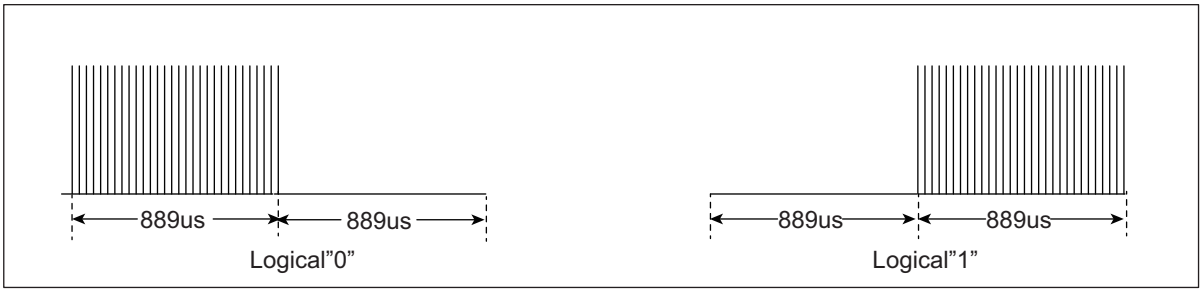


Figure 8.3: rc5 logical

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

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

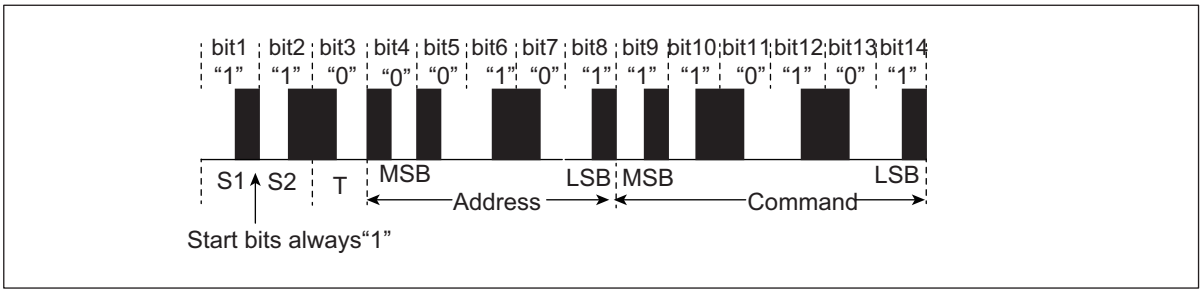


Figure 8.4: rc5

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

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

8.3.2 Pulse width reception

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

8.3.3 Normal sending mode

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

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

8.3.4 Pulse width transmission

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

8.3.5 Carrier modulation

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

8.3.6 IR interrupt

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

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

8.4 Register description

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

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

8.4.1 irtx_config

Address：0x4000a600

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

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

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

8.4.2 irtx_int_sts

Address：0x4000a604

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

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

8.4.3 irtx_data_word0

Address：0x4000a608

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

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

8.4.4 irtx_data_word1

Address：0x4000a60c

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

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

8.4.5 irtx_pulse_width

Address：0x4000a610

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

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

8.4.6 irtx_pw

Address：0x4000a614

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

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

8.4.7 irtx_swm_pw_0

Address：0x4000a640

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

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

8.4.8 irtx_swm_pw_1

Address：0x4000a644

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

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

8.4.9 irtx_swm_pw_2

Address：0x4000a648

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

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

8.4.10 irtx_swm_pw_3

Address：0x4000a64c

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

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

8.4.11 irtx_swm_pw_4

Address：0x4000a650

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

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

8.4.12 irtx_swm_pw_5

Address：0x4000a654

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

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

8.4.13 irtx_swm_pw_6

Address：0x4000a658

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

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

8.4.14 irtx_swm_pw_7

Address：0x4000a65c

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

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

8.4.15 irrx_config

Address：0x4000a680

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

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

8.4.16 irrx_int_sts

Address：0x4000a684

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

Bits	Name	Type	Reset	Description
31:25	RSVD			
24	RXEEN	R/W	1'b1	Interrupt enable of irrx_end_int

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

8.4.17 irrx_pw_config

Address：0x4000a688

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

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

8.4.18 irrx_data_count

Address：0x4000a690

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

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

8.4.19 irrx_data_word0

Address：0x4000a694

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

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

8.4.20 irrx_data_word1

Address：0x4000a698

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

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

8.4.21 irrx_swm_fifo_config_0

Address：0x4000a6c0

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

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

Bits	Name	Type	Reset	Description
2	RXFOF	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
1	RSVD			
0	RXFCLR	W1C	1'b0	Clear signal of RX FIFO

8.4.22 irrx_swm_fifo_rdata

Address：0x4000a6c4

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

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

9 SPI

9.1 Introduction

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

9.2 Main features

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

9.3 Function description

9.3.1 Clock control

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

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

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

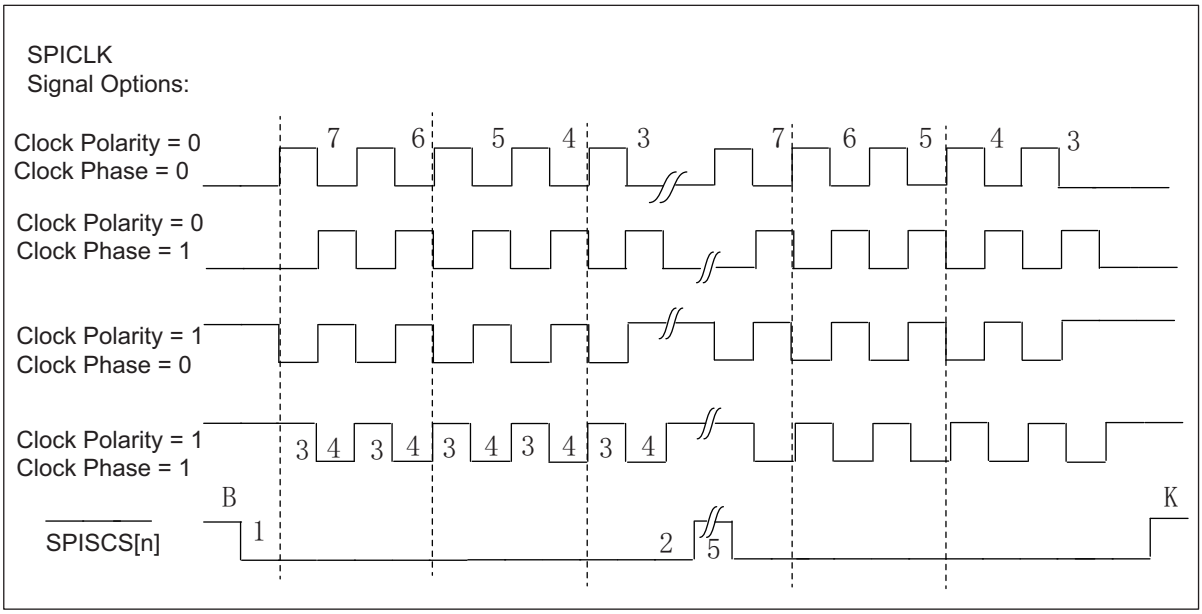


Figure 9.1: SPI clock

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

9.3.2 Master continuous transmission mode

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

9.3.3 Acceptance filtering function

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

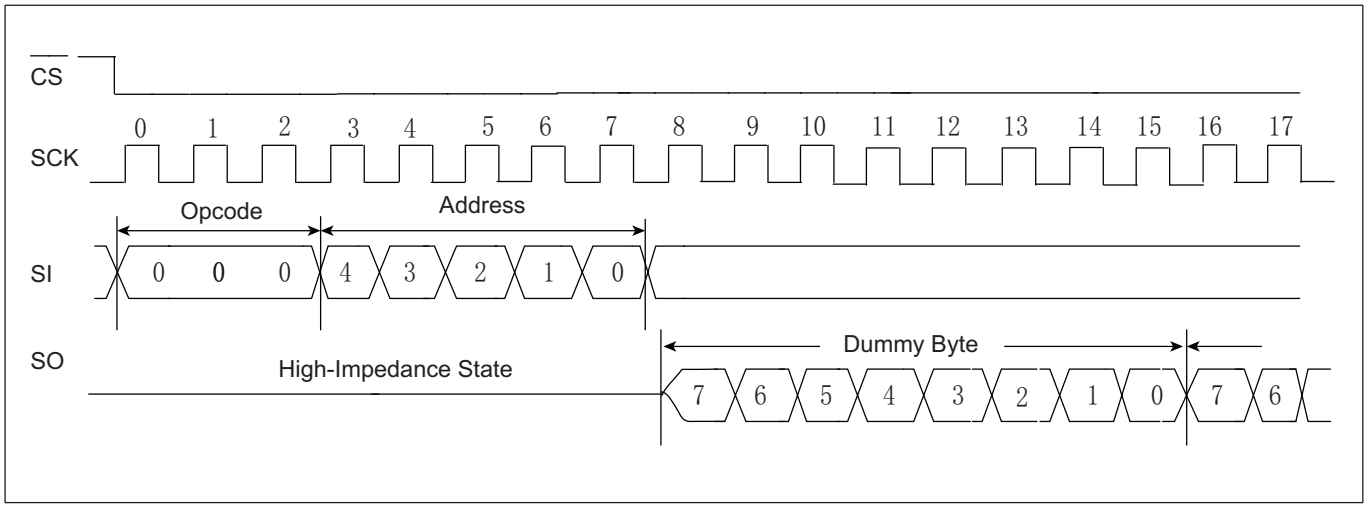


Figure 9.2: SPI ignore

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

9.3.4 Receive error correction

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

9.3.5 Slave mode timeout mechanism

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

9.3.6 I/O transfer mode

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

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

9.3.7 DMA transfer mode

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

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

9.3.8 SPI interrupt

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

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

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

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

9.4 Register description

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

Name	Description
spi_fifo_rdata	SPI FIFO read data

9.4.1 spi_config

Address：0x4000a200

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

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

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

9.4.2 spi_int_sts

Address：0x4000a204

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

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

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

9.4.3 spi_bus_busy

Address：0x4000a208

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

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

Bits	Name	Type	Reset	Description

9.4.4 spi_prd_0

Address：0x4000a210

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

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

9.4.5 spi_prd_1

Address：0x4000a214

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

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

9.4.6 spi_rxd_ignr

Address：0x4000a218

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

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

9.4.7 spi_sto_value

Address：0x4000a21c

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

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

9.4.8 spi_fifo_config_0

Address：0x4000a280

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	RFUF	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr

Bits	Name	Type	Reset	Description
6	RFOF	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	TFUF	R	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	TFOF	R	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	RFC	W1C	1'b0	Clear signal of RX FIFO
2	TFC	W1C	1'b0	Clear signal of TX FIFO
1	DMAREN	R/W	1'b0	Enable signal of dma_rx_req/ack interface
0	DMATEN	R/W	1'b0	Enable signal of dma_tx_req/ack interface

9.4.9 spi_fifo_config_1

Address：0x4000a284

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

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

9.4.10 spi_fifo_wdata

Address：0x4000a288

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

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

9.4.11 spi_fifo_rdata

Address：0x4000a28c

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

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

10 UART

10.1 Introduction

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

BL602 has two sets of UART ports (UART0 and UART1). By using with DMA, you can achieve efficient data communication.

10.2 Main features

- Full-duplex asynchronous communication
- Data bit length can be selected from 5/6/7/8 bits
- Stop bit length can be selected from 0.5/1/1.5/2 bits
- Supports odd/even/no parity bits
- Detects wrong start bit
- Multiple interrupt control
- Support hardware flow control (RTS / CTS)
- Convenient baud rate programming
- Configurable MSB / LSB priority transmission
- Normal / fixed character automatic baud rate detection
- 32-byte transmit / receive FIFO
- Support DMA transfer mode
- Maximum operating frequency is 160MHz

10.3 Function description

10.3.1 Data format description

Normal UART communication data is composed of a start bit, a data bit, a parity bit, and a stop bit. The BL602's UART supports configurable data bits, parity bits, and stop bits, all of which are set in the UTX_CONFIG and URX_CONFIG registers. The waveform of one frame of data is shown below:

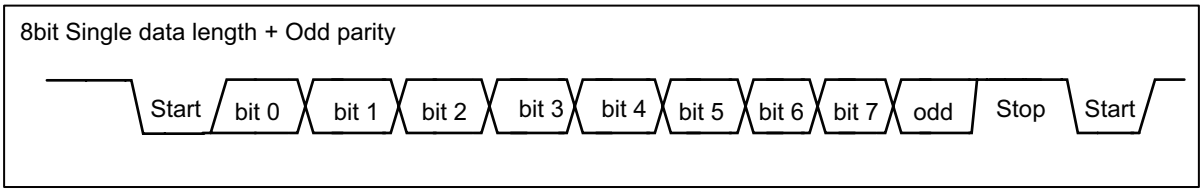


Figure 10.1: UART data

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

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

When <CR_UTX_PRT_EN> and <CR_URX_PRT_EN> are set, the data frame adds a parity bit after the data. <CR_-UTX_PRT_SEL> and <CR_URX_PRT_SEL> are used to select odd or even parity. When the receiver detects a parity error in the input data, a parity error interrupt is generated.

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

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

10.3.2 Basic architecture diagram

10.3.3 Clock source

The UART has two clock sources: 160MHz APB_CLK and FCLK. The frequency divider in the clock is used to divide the clock source and then generate a clock signal to drive the UART module. As shown below:

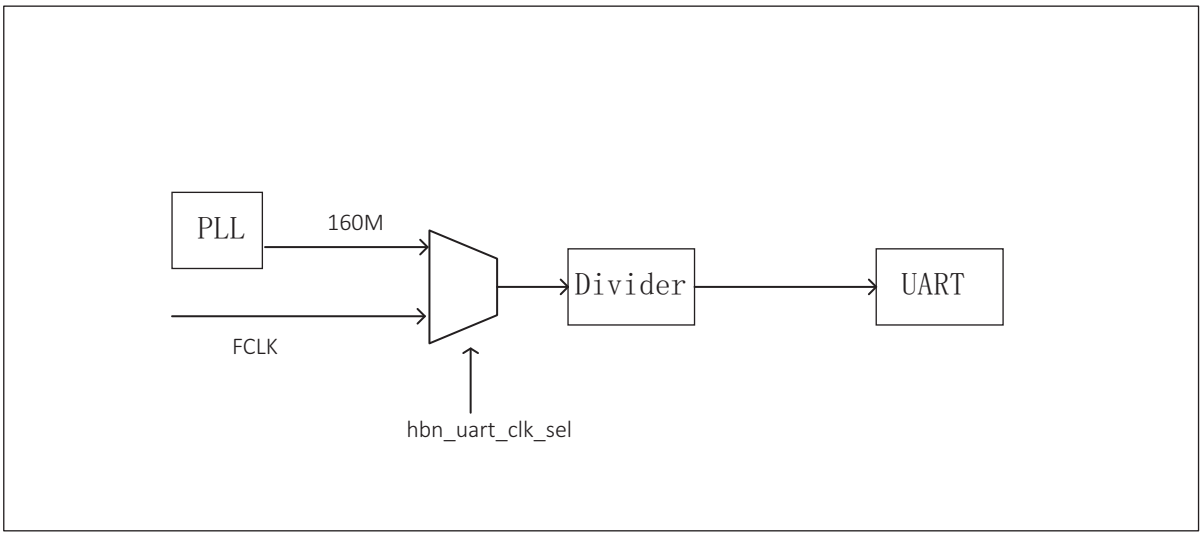


Figure 10.2: UART clock

10.3.4 Baud rate setting

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

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

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

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

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

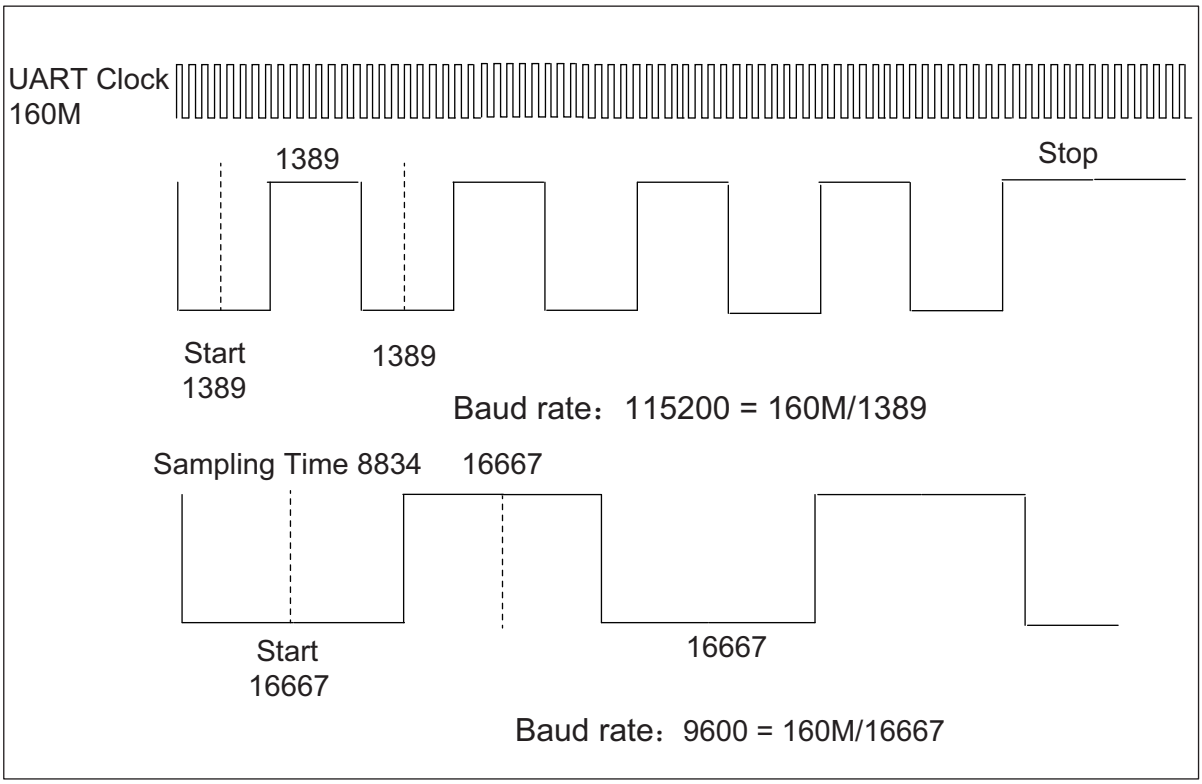


Figure 10.3: UART sample

10.3.5 Transmitter

The transmitter contains a 32-byte transmit FIFO to store the data to be transmitted. Software can write the TX FIFO through the APB bus, and can also move data into the TX FIFO through DMA. When the transmit enable bit is set, the data stored in the FIFO will be output from the TX pin. Software can choose to transfer data into TX FIFO through two methods: DMA or APB bus.

Software can check the status of the transmitter by querying the TX FIFO remaining free space count value in bit <TX_FIFO_CNT> of the register UART_FIFO_CONFIG_1. The transmitter's FreeRun mode is as follows:

- If the FreeRun mode is not turned on, the transmission behavior is terminated and an interrupt is generated when the transmission byte reaches the specified length. If you want to continue the transmission, you need to turn it off and then enable the transmission enable bit.
- If the FreeRun mode is turned on, the transmitter will transmit when there is data in the TX FIFO, and the transmitted byte will not terminate when it reaches the specified length.

10.3.6 receiver

The receiver contains a 32-byte receive FIFO to store the received data. Software can check the status of the receiver by querying the RX FIFO available data count value through the bit <RX_FIFO_CNT> in the register UART_FIFO_-CONFIG_1. The lower 8 bits of the URX_RTO_TIMER register are used to set a receive timeout threshold. When the receiver does not receive data beyond this time value, an interrupt will be triggered. Bits <CR_URX_DEG_EN> and

<CR_URX_DEG_CNT> of the URX_CONFIG register are used to enable the deburring function and set the threshold value, which controls the filtering part before UART sampling. The UART filters the glitches below the threshold width in the waveform and sends them for sampling.

10.3.7 Automatic baud rate detection

The UART module supports automatic baud rate detection. The detection is divided into two types, one is the general mode and the other is the fixed character mode. Each time the bit <CR_URX_ABR_EN> of the set register URX_-CONFIG is turned on, these two detection modes are enabled.

General mode

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

Fixed character mode

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

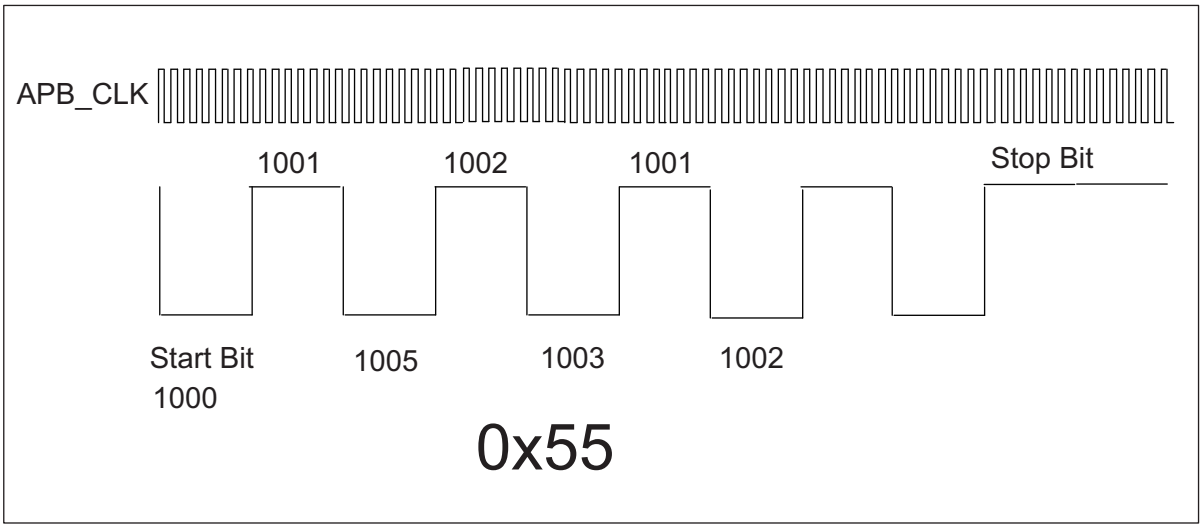


Figure 10.4: UART fixed character mode

For an unknown baud rate, the UART uses UART_CLK to count the start bit with a width of 1000 and the second bit with a width of 1001, which does not fluctuate more than 4 UART_CLK from the previous bit width. The UART will continue to count the third bit. The third bit is 1005. If the difference between the UART and the start bit exceeds 4, the test fails and the data is discarded. The UART compares the first 6 bits of the data bit with the start bit in turn.

The formula for calculating the detected baud rate is as follows:

Baud rate = source clock/(16-bit detection value + 1)

10.3.8 Hardware flow control

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

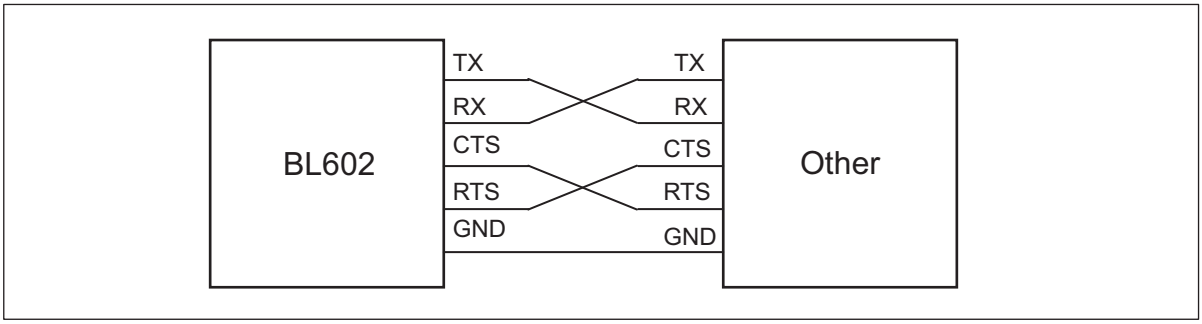


Figure 10.5: UART flow control

When using the hardware flow control function, the output signal RTS is high to request the other party to send data, and RTS is low to notify the other party to stop data transmission until the RTS returns to high. There are two ways for the hardware flow control of the transmitter.

- The bit <CR_URX_RTS_SW_MODE> of the URX_CONFIG register is equal to 0: pull down the RTS level when the amount of data in the RX FIFO is greater than <RX_FIFO_TH>.
- The bit <CR_URX_RTS_SW_MODE> of the URX_CONFIG register is equal to 1: The RTS level can be changed by configuring the bit <CR_URX_RTS_SW_VAL> of the URX_CONFIG register.

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

10.3.9 DMA transfer mode

The UART supports DMA transfer mode. To use this mode, you need to set the TX and RX FIFO thresholds through the bits <TX_FIFO_TH> and <RX_FIFO_TH> of the UART_FIFO_CONFIG_1 register. When this mode is enabled, the UART will check the TX / RX FIFO. Above the set threshold, a DMA request will be initiated, and the DMA will move data to the TX FIFO or out of the RX FIFO according to the setting.

10.3.10 UART interrupt

The UART has multiple interrupt control, including the following interrupt modes:

- TX transmission end interrupt

- RX transmission end interrupt
- TX FIFO request interrupt
- RX FIFO request interrupt
- RX timeout interrupt
- RX parity error interrupt
- TX FIFO overflow interrupt
- RX FIFO overflow interrupt

TX and RX can set a transmission length value through the upper 16 bits of the UTX_CONFIG and URX_CONFIG registers. When the number of bytes transmitted reaches this value, the corresponding TX / RX transmission end interrupt will be triggered. The TX / RX FIFO request interrupt will be triggered when its FIFO available count value is greater than the threshold set in the register UART_FIFO_CONFIG_1. When the condition is not met, the interrupt flag will be automatically cleared. The RX timeout interrupt is triggered when the receiver does not receive data beyond the timeout threshold, and the RX parity error interrupt occurs when a parity error occurs. If the TX / RX FIFO overflows or underflows, the corresponding overflow interrupt will be triggered. When the FIFO clear bit TX_-FIFO_CLR / RX_FIFO_CLR is set to 1, the corresponding FIFO will be cleared and the overflow interrupt flag will be automatically cleared.

Query the interrupt status through the register UART_INT_STS, and clear the interrupt by writing 1 to the corresponding bit in the register UART_INT_CLEAR.

10.4 Register description

Name	Description
utx_config	UART TX configuration register
urx_config	UART RX configuration register
uart_bit_prd	UART period control register
data_config	UART data configuration register
utx_ir_position	UART TX ir position control register
urx_ir_position	UART RX ir position control register
urx_rto_timer	RTO interrupt control register
uart_int_sts	UART interrupt status
uart_int_mask	UART interrupt mask
uart_int_clear	UART interrupt clear
uart_int_en	UART interrupt enable

Name	Description
uart_status	UART status control register
sts_urx_abr_prd	Auto baud detection control register
uart_fifo_config_0	UART FIFO configuration register0
uart_fifo_config_1	UART FIFO configuration register1
uart_fifo_wdata	UART FIFO write data
uart_fifo_rdata	UART FIFO read data

10.4.1 utx_config

Address：0x4000a000

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

Bits	Name	Type	Reset	Description
31:16	TXLEN	R/W	16'd0	Length of UART TX data transfer (Unit: character/byte) (Don't-care if cr_utx_frm_en is enabled)
15:14	RSVD			
13:12	TXBCNTP	R/W	2'd1	UART TX STOP bit count (unit: 0.5 bit)
11	RSVD			
10:8	TXBCNTD	R/W	3'd7	UART TX DATA bit count for each character
7	IRTXINV	R/W	1'b0	Inverse signal of UART TX output in IR mode
6	IRTXEN	R/W	1'b0	Enable signal of UART TX IR mode
5	TXPRSEL	R/W	1'b0	Select signal of UART TX parity bit 1: Odd parity 0: Even parity
4	TXPREN	R/W	1'b0	Enable signal of UART TX parity bit
3	RSVD			
2	FRMEN	R/W	1'b0	Enable signal of UART TX freerun mode (utx_end_int will be disabled)
1	CTSEN	R/W	1'b0	Enable signal of UART TX CTS flow control function

Bits	Name	Type	Reset	Description
0	EN	R/W	1'b0	Enable signal of UART TX function Asserting this bit will trigger the transaction, and should be de-asserted after finish

10.4.2 urx_config

Address：0x4000a004

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

Bits	Name	Type	Reset	Description
31:16	RXLEN	R/W	16'd0	Length of UART RX data transfer (Unit: character/byte) urx_end_int will assert when this length is reached
15:12	DEGCNT	R/W	4'd0	De-glitch function cycle count
11	DEGEN	R/W	1'b0	Enable signal of RXD input de-glitch function
10:8	RXBCNTD	R/W	3'd7	UART RX DATA bit count for each character
7	IRRXINV	R/W	1'b0	Inverse signal of UART RX input in IR mode
6	IRRXEN	R/W	1'b0	Enable signal of UART RX IR mode
5	RXPRSEL	R/W	1'b0	Select signal of UART RX parity bit 1: Odd parity 0: Even parity
4	RXPREN	R/W	1'b0	Enable signal of UART RX parity bit
3	ABREN	R/W	1'b0	Enable signal of UART RX Auto Baud Rate detection function
2	RTSSWV	R/W	1'b0	UART RX RTS output SW control value
1	RTSSWM	R/W	1'b0	UART RX RTS output SW control mode
0	EN	R/W	1'b0	Enable signal of UART RX function

10.4.3 uart_bit_prd

Address：0x4000a008

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

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

10.4.4 data_config

Address：0x4000a00c

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

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

10.4.5 utx_ir_position

Address：0x4000a010

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

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

10.4.6 urx_ir_position

Address：0x4000a014

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

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

10.4.7 urx_rto_timer

Address：0x4000a018

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

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

10.4.8 uart_int_sts

Address：0x4000a020

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	RFERINT	R	1'b0	UART RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
6	TFIN	R	1'b0	UART TX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
5	RPCEINT	R	1'b0	UART RX parity check error interrupt
4	RRTOINT	R	1'b0	UART RX Time-out interrupt
3	RFIN	R	1'b0	UART RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto-cleared when data is popped
2	TFIN	R	1'b0	UART TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed
1	REIN	R	1'b0	UART RX transfer end interrupt (set according to cr_urx_-len)
0	TEIN	R	1'b0	UART TX transfer end interrupt (set according to cr_utx_-len)

10.4.9 uart_int_mask

Address：0x4000a024

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

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

Bits	Name	Type	Reset	Description

10.4.10 uart_int_clear

Address：0x4000a028

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

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

10.4.11 uart_int_en

Address：0x4000a02c

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	RFER	R/W	1’b1	Interrupt enable of urx_fer_int
6	TFER	R/W	1’b1	Interrupt enable of utx_fer_int
5	RPCE	R/W	1’b1	Interrupt enable of urx_pce_int
4	RRTO	R/W	1’b1	Interrupt enable of urx_rto_int
3	RFIF	R/W	1’b1	Interrupt enable of urx_fifo_int

Bits	Name	Type	Reset	Description
2	TFIF	R/W	1'b1	Interrupt enable of utx_fifo_int
1	REND	R/W	1'b1	Interrupt enable of urx_end_int
0	TEND	R/W	1'b1	Interrupt enable of utx_end_int

10.4.12 uart_status

Address：0x4000a030

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

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

10.4.13 sts_urx_abr_prd

Address：0x4000a034

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

Bits	Name	Type	Reset	Description
31:16	ABRPRD	R	16'd0	Bit period of Auto Baud Rate detection using codeword 0x55
15:0	ABRPRDS	R	16'd0	Bit period of Auto Baud Rate detection using START bit

10.4.14 uart_fifo_config_0

Address：0x4000a080

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

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

10.4.15 uart_fifo_config_1

Address：0x4000a084

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

Bits	Name	Type	Reset	Description
31:29	RSVD			
28:24	RFITH	R/W	5'd0	RX FIFO threshold, dma_rx_req will not be asserted if tx_-fifo_cnt is less than this value
23:21	RSVD			
20:16	TFITH	R/W	5'd0	TX FIFO threshold, dma_tx_req will not be asserted if tx_-fifo_cnt is less than this value
15:14	RSVD			
13:8	RFICNT	R	6'd0	RX FIFO available count
7:6	RSVD			

Bits	Name	Type	Reset	Description
5:0	TFICNT	R	6'd32	TX FIFO available count

10.4.16 uart_fifo_wdata

Address：0x4000a088

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

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

10.4.17 uart_fifo_rdata

Address：0x4000a08c

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

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

11 I2C

11.1 Introduction

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

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

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

BL602 includes an I2C controller host, which can be flexibly configured with slaveAddr, subAddr, and data transmission to facilitate communication with slave devices. It provides 2 word depth fifo and provides interrupt functions. It can be used with DMA to improve efficiency and flexibly adjust clock frequency.

11.2 Main features

- Support host mode
- Support multi-master mode and arbitration function
- Flexible clock frequency adjustment
- The maximum operating frequency is 40MHz

11.3 Function description

Table 11.1: Pin lists

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

11.3.1 Start and stop conditions

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

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

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

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

The waveform diagram is as follows:

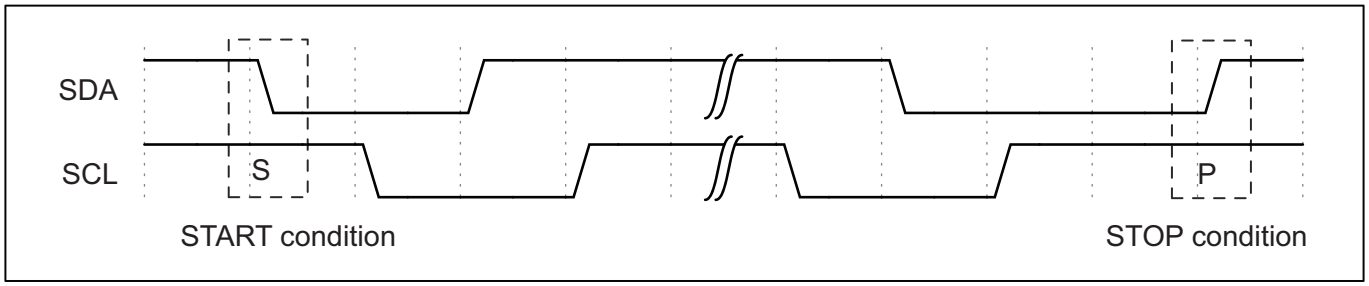


Figure 11.1: I2C stop/start condition

11.3.2 Data transmission format

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

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

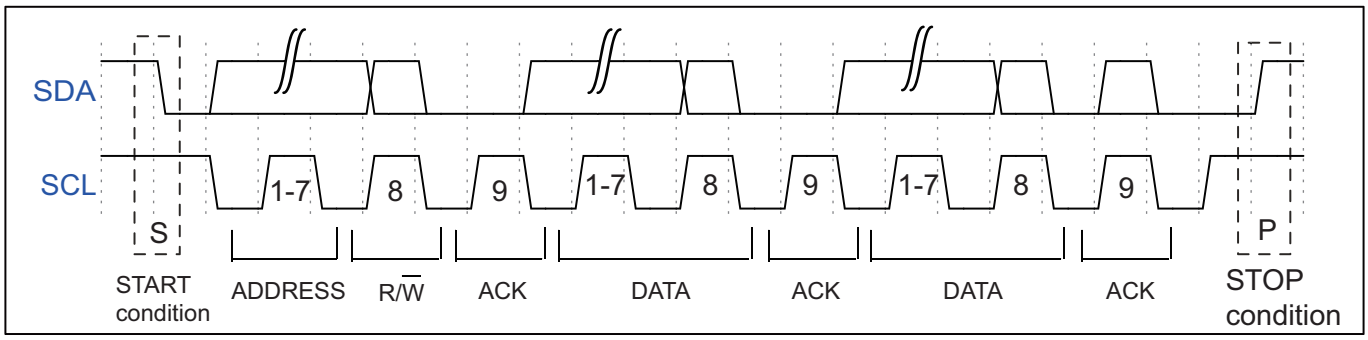


Figure 11.2: Master transmission

Timing of master transmission and slave reception

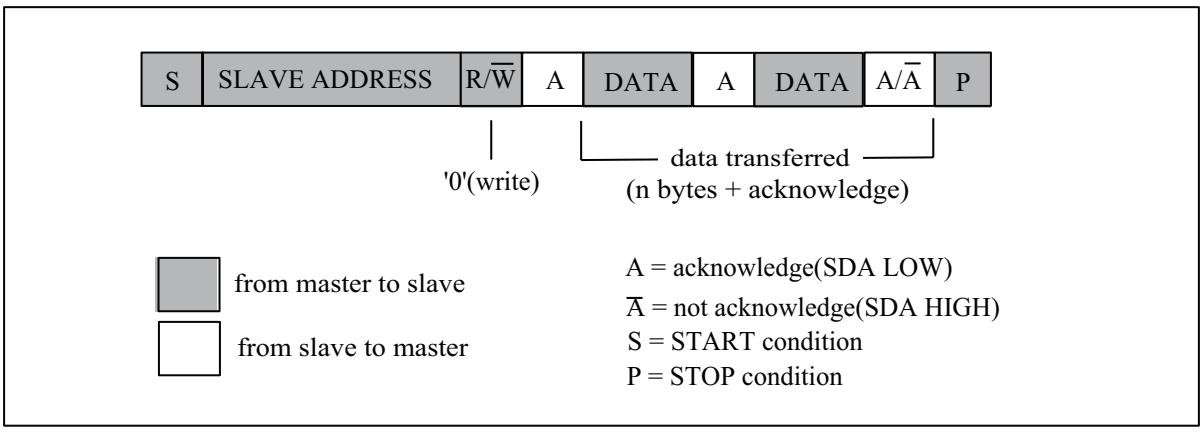


Figure 11.3: Master tx and slave rx

Timing of master receive and slave send

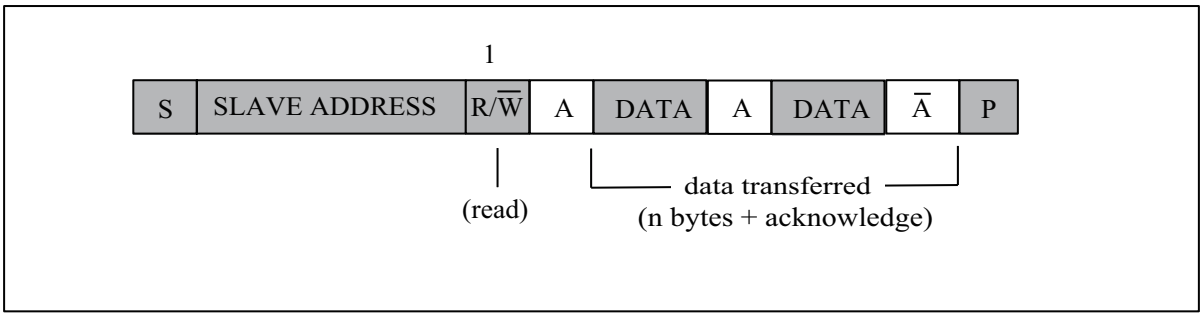


Figure 11.4: Master rx and slave tx

11.3.3 Arbitration

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

the bus is free.

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

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

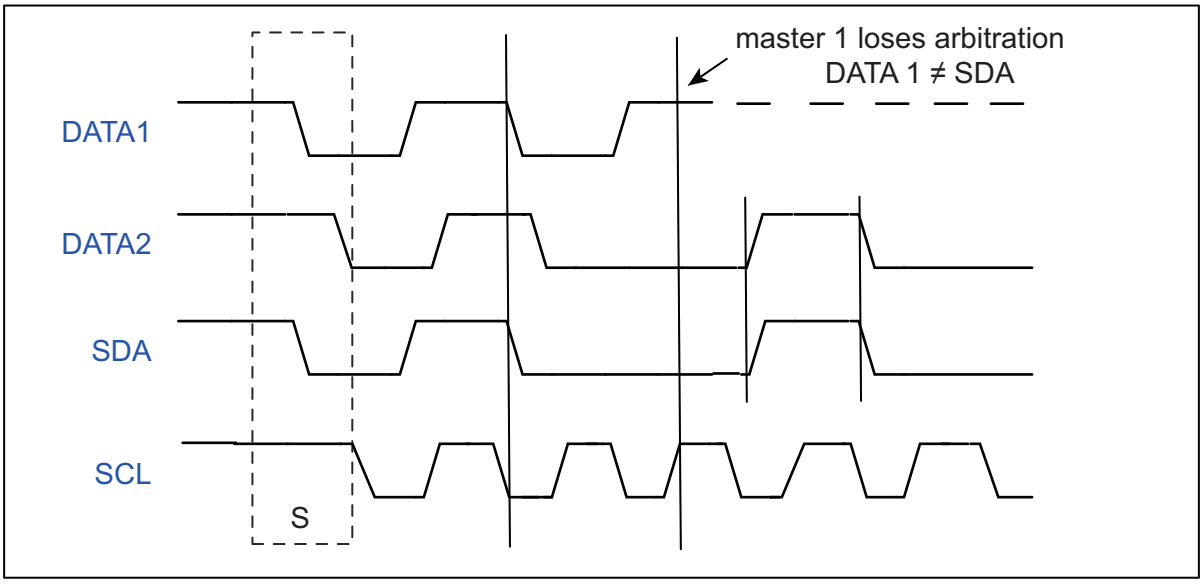


Figure 11.5: Tx and Rx together

11.4 I2C clock setting

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

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

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

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

11.5 I2C configuration process

11.5.1 Configuration item

- Read and write flags

- Slave address
- Slave device address
- Slave device address length
- Data (when sending, configure the data to be sent; when receiving, store the received data)
- Data length
- Enable signal

11.5.2 Read and write flags

I2C supports two working states: sending and receiving. Register I2C_CR_I2C_PKT_DIR indicates the sending or receiving status. When it is set to 0, it indicates the sending state, and when it is set to 1, it indicates the receiving state.

11.5.3 Slave address

Each slave device connected to I2C will have a unique address. Usually the address length is 7 bits. The slave device address will be written into the register I2C_CR_I2C_SLV_ADDR. I2C will automatically shift left by 1 bit before sending it from the device address. Transmit/receive direction bit on the low-order complement.

11.5.4 Slave device address

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

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

11.5.5 Slave device address length

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

11.5.6 Data

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

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

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

11.5.7 Data length

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

11.5.8 Enable signal

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

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

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

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

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

11.6 FIFO management

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

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

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

I2C FIFO status:

- RX FIFO underflow: When the data in the RX FIFO has been read or is empty, continue to read data from the RX

FIFO, the register RFIU will be set;

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

11.7 Using DMA

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

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

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

11.7.1 DMA transmission process

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

11.7.2 DMA receiving process

1. Configure the read and write flags to 1
2. Configure the slave device address
3. Configure Slave Device Address
4. Configure slave device address length
5. Data length
6. Set the enable signal register
7. Configure DMA transfer size
8. Configure the DMA source address transfer width (Note that when I2C is used with DMA, the source address transfer width needs to be set to 32bits and used in word alignment)
9. Configure DMA destination address transfer width
10. Configure the DMA source address as I2C RX FIFO address, I2C_FIFO_RDATA
11. Configure the DMA destination address as the memory address to store the received data
12. Enable DMA

11.8 Interrupt

I2C includes the following interrupts:

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

11.9 Register description

Name	Description
i2c_config	I2C configuration register
i2c_int_sts	I2C interrupt status
i2c_sub_addr	I2C sub-address configuration

Name	Description
i2c_bus_busy	I2C bus busy control register
i2c_prd_start	I2C length of start phase
i2c_prd_stop	I2C length of stop phase
i2c_prd_data	I2C length of data phase
i2c_fifo_config_0	I2C FIFO configuration register0
i2c_fifo_config_1	I2C FIFO configuration register1
i2c_fifo_wdata	I2C FIFO write data
i2c_fifo_rdata	I2C FIFO read data

11.9.1 i2c_config

Address：0x4000a300

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

Bits	Name	Type	Reset	Description
31:28	DEGCNT	R/W	4'd0	De-glitch function cycle count
27:24	RSVD			
23:16	PKTLEN	R/W	8'd0	Packet length (unit: byte)
15	RSVD			
14:8	SLVADDR	R/W	7'd0	Slave address for I2C transaction (target address)
7	RSVD			
6:5	SABC	R/W	2'd0	Sub-address field byte count 2'd0: 1-byte, 2'd1: 2-byte, 2'd2: 3-byte, 2'd3: 4-byte
4	SAEN	R/W	1'b0	Enable signal of I2C sub-address field
3	SCLSEN	R/W	1'b1	Enable signal of I2C SCL synchronization, should be enabled to support Multi-Master and Clock-Stretching (Normally should not be turned-off)

Bits	Name	Type	Reset	Description
2	DEGEN	R/W	1'b0	Enable signal of I2C input de-glitch function (for all input pins)
1	PKTDIR	R/W	1'b1	Transfer direction of the packet 1'b0: Write; 1'b1: Read
0	MEN	R/W	1'b0	Enable signal of I2C Master function Asserting this bit will trigger the transaction, and should be de-asserted after finish

11.9.2 i2c_int_sts

Address：0x4000a304

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD		FEREN	ARBEN	NAKEN	RXFEN	TXFEN	ENDEN	RSVD			ARBCLR	NAKCLR	RSVD		ENDCLR
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		FERMASK	ARBMASK	NAKMASK	RXFMASK	TXFMASK	ENDMASK	RSVD		FERINT	ARBINT	NAKINT	RXFINT	TXFINT	ENDINT

Bits	Name	Type	Reset	Description
31:30	RSVD			
29	FEREN	R/W	1'b1	Interrupt enable of i2c_fer_int
28	ARBEN	R/W	1'b1	Interrupt enable of i2c_arb_int
27	NAKEN	R/W	1'b1	Interrupt enable of i2c_nak_int
26	RXFEN	R/W	1'b1	Interrupt enable of i2c_rxf_int
25	TXFEN	R/W	1'b1	Interrupt enable of i2c_txf_int
24	ENDEN	R/W	1'b1	Interrupt enable of i2c_end_int
23:21	RSVD			
20	ARBCLR	W1C	1'b0	Interrupt clear of i2c_arb_int
19	NAKCLR	W1C	1'b0	Interrupt clear of i2c_nak_int
18:17	RSVD			
16	ENDCLR	W1C	1'b0	Interrupt clear of i2c_end_int
15:14	RSVD			
13	FERMASK	R/W	1'b1	Interrupt mask of i2c_fer_int
12	ARBMASK	R/W	1'b1	Interrupt mask of i2c_arb_int

Bits	Name	Type	Reset	Description
11	NAKMASK	R/W	1'b1	Interrupt mask of i2c_nak_int
10	RXFMASK	R/W	1'b1	Interrupt mask of i2c_rxf_int
9	TXFMASK	R/W	1'b1	Interrupt mask of i2c_txf_int
8	ENDMASK	R/W	1'b1	Interrupt mask of i2c_end_int
7:6	RSVD			
5	FERINT	R	1'b0	I2C TX/RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
4	ARBINT	R	1'b0	I2C arbitration lost interrupt
3	NAKINT	R	1'b0	I2C NACK-received interrupt
2	RXFINT	R	1'b0	I2C RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto-cleared when data is popped
1	TXFINT	R	1'b0	I2C TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed
0	ENDINT	R	1'b0	I2C transfer end interrupt

11.9.3 i2c_sub_addr

Address：0x4000a308

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

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

11.9.4 i2c_bus_busy

Address：0x4000a30c

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

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

11.9.5 i2c_prd_start

Address：0x4000a310

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

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

11.9.6 i2c_prd_stop

Address：0x4000a314

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

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

11.9.7 i2c_prd_data

Address：0x4000a318

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

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

11.9.8 i2c_fifo_config_0

Address：0x4000a380

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

Bits	Name	Type	Reset	Description
31:8	RSVD			
7	RFIU	R	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	RFIO	R	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr

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

11.9.9 i2c_fifo_config_1

Address：0x4000a384

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

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

11.9.10 i2c_fifo_wdata

Address：0x4000a388

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

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

11.9.11 i2c_fifo_rdata

Address：0x4000a38c

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

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

12 PWM

12.1 Introduction

Pulse width modulation (PWM) is an analog control method that modulates the bias of the transistor base or the grid of the MOS tube according to the change in the corresponding load. Therefore, the on-time of the transistor or the MOS tube is changed, and the output of the switch stabilized power supply is changed. This method can keep the output voltage of the power supply constant when the operating conditions change. It is a very effective technique for controlling analog circuits using digital signals from microprocessors. It is widely used in many fields from measurement and communication to power control and conversion.

12.2 Main features

- Supports 5-channel PWM signal generation
- Three clock sources can be selected (bus clock <bclk>, crystal clock <xtal_ck>, slow clock <32k>), with 16-bit clock divider
- Double threshold setting to increase pulse flexibility
- The maximum operating frequency is 40MHz

12.3 Function description

12.3.1 Clock and divider

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

A. bclk - Chip bus clock

B. XTAL - External crystal clock

C. f32k - System RTC clock

Each counter has its own 16-bit frequency divider. The selected clock can be divided by APB. The PWM counter will

use the divided clock as the counting cycle unit, and perform one action every time a counting cycle passes .

12.3.2 Pulse generation principle

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

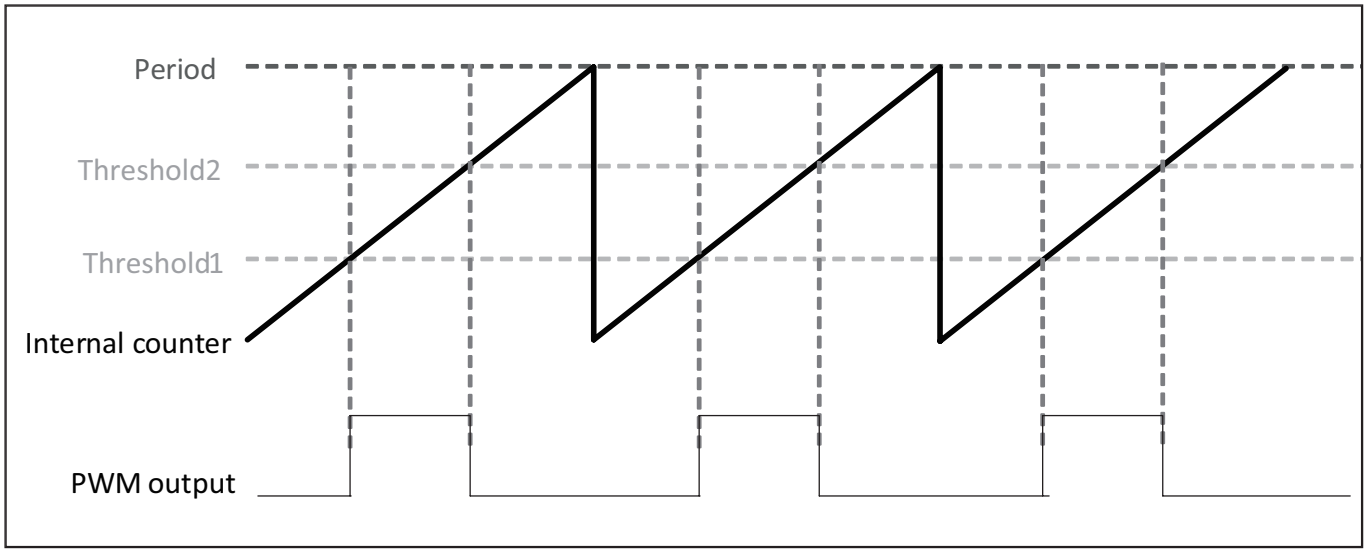


Figure 12.1: Pwm

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

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

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

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

example: The PWM source clock is 80MHz. To generate a 1kHz PWM wave with a duty ratio of 20%, the settings are as follows:

PWMn_CLK_DIV[15:0]=2

PWMn_PERIOD[15:0]=80000000/2/1000=40000

PWMn_THRE1[15:0]=0

PWMn_THRE2[15:0]=0+40000*20%=8000

12.3.3 PWM interrupt

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

Table 12.1: Duty Cycle Parameters

F/MHz	Supported duty cycle (n is an integer, and 2 <= n <= 65535^2)											
40	0%	50%	100%									
26.67	0%	33.33%	66.67%	100%								
20	0%	25%	50%	75%	100%							
16	0%	20%	40%	60%	80%	100%						
13.33	0%	16.67%	33.33%	50%	66.67%	83.33%	100%					
11.43	0%	14.29%	28.57%	42.86%	57.14%	71.43%	85.71%	100%				
10	0%	12.50%	25%	37.50%	50%	62.50%	75%	87.50%	100%			
8.89	0%	11.11%	22.22%	33.33%	44.44%	55.56%	66.67%	77.78%	88.89%	100%		
8	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	
•												
•												
•												
80/n	0/n	1/n	2/n	3/n	4/n	5/n	6/n	7/n	8/n	9/n	•••	n/n

12.4 Register description

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

Name	Description
pwm1_clkdiv	PWM1 clock division configuration register
pwm1_thre1	PWM1 first counter threshold configuration register
pwm1_thre2	PWM1 sencond counter threshold configuration register
pwm1_period	PWM1 period setting register
pwm1_config	PWM1 configuration register
pwm1_interrupt	PWM1 interrupt register
pwm2_clkdiv	PWM2 clock division configuration register
pwm2_thre1	PWM2 first counter threshold configuration register
pwm2_thre2	PWM2 sencond counter threshold configuration register
pwm2_period	PWM2 period setting register
pwm2_config	PWM2 configuration register
pwm2_interrupt	PWM2 interrupt register
pwm3_clkdiv	PWM3 clock division configuration register
pwm3_thre1	PWM3 first counter threshold configuration register
pwm3_thre2	PWM3 sencond counter threshold configuration register
pwm3_period	PWM3 period setting register
pwm3_config	PWM3 configuration register
pwm3_interrupt	PWM3 interrupt register
pwm4_clkdiv	PWM4 clock division configuration register
pwm4_thre1	PWM4 first counter threshold configuration register
pwm4_thre2	PWM4 sencond counter threshold configuration register
pwm4_period	PWM4 period setting register
pwm4_config	PWM4 configuration register
pwm4_interrupt	PWM4 interrupt register

12.4.1 pwm_int_config

Address：0x4000a400

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

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

12.4.2 pwm0_clkdiv

Address：0x4000a420

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

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

12.4.3 pwm0_thre1

Address：0x4000a424

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	THRE1	R/W	16'b0	PWM first counter threshold, can't be larger that pwm_-thre2

Bits	Name	Type	Reset	Description

12.4.4 pwm0_thre2

Address：0x4000a428

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

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

12.4.5 pwm0_period

Address：0x4000a42c

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

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

12.4.6 pwm0_config

Address：0x4000a430

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

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

12.4.7 pwm0_interrupt

Address：0x4000a434

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

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

12.4.8 pwm1_clkdiv

Address：0x4000a440

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

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

12.4.9 pwm1_thre1

Address：0x4000a444

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

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

12.4.10 pwm1_thre2

Address：0x4000a448

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

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

12.4.11 pwm1_period

Address：0x4000a44c

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

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

12.4.12 pwm1_config

Address：0x4000a450

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

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

12.4.13 pwm1_interrupt

Address：0x4000a454

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

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

12.4.14 pwm2_clkdiv

Address：0x4000a460

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

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

12.4.15 pwm2_thre1

Address：0x4000a464

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

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

Bits	Name	Type	Reset	Description

12.4.16 pwm2_thre2

Address：0x4000a468

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

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

12.4.17 pwm2_period

Address：0x4000a46c

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

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

12.4.18 pwm2_config

Address：0x4000a470

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

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

12.4.19 pwm2_interrupt

Address：0x4000a474

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

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

12.4.20 pwm3_clkdiv

Address：0x4000a480

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

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

12.4.21 pwm3_thre1

Address：0x4000a484

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

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

12.4.22 pwm3_thre2

Address：0x4000a488

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

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

12.4.23 pwm3_period

Address：0x4000a48c

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

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

12.4.24 pwm3_config

Address：0x4000a490

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

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

12.4.25 pwm3_interrupt

Address：0x4000a494

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

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

12.4.26 pwm4_clkdiv

Address：0x4000a4a0

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

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

12.4.27 pwm4_thre1

Address：0x4000a4a4

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

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

Bits	Name	Type	Reset	Description

12.4.28 pwm4_thre2

Address：0x4000a4a8

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

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

12.4.29 pwm4_period

Address：0x4000a4ac

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

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

12.4.30 pwm4_config

Address：0x4000a4b0

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

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

12.4.31 pwm4_interrupt

Address：0x4000a4b4

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

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

13 TIMER

13.1 Introduction

The chip has two 32-bit counters, each of which can independently control and configure its parameters and clock frequency.

There is a watchdog counter in the chip. Unpredictable software or hardware behavior may cause the application to malfunction. A watchdog timer can help the system recover from it. If the current time exceeds the predetermined time, but the dog is not fed or closed Timer, which can trigger interrupt or system reset according to the setting.

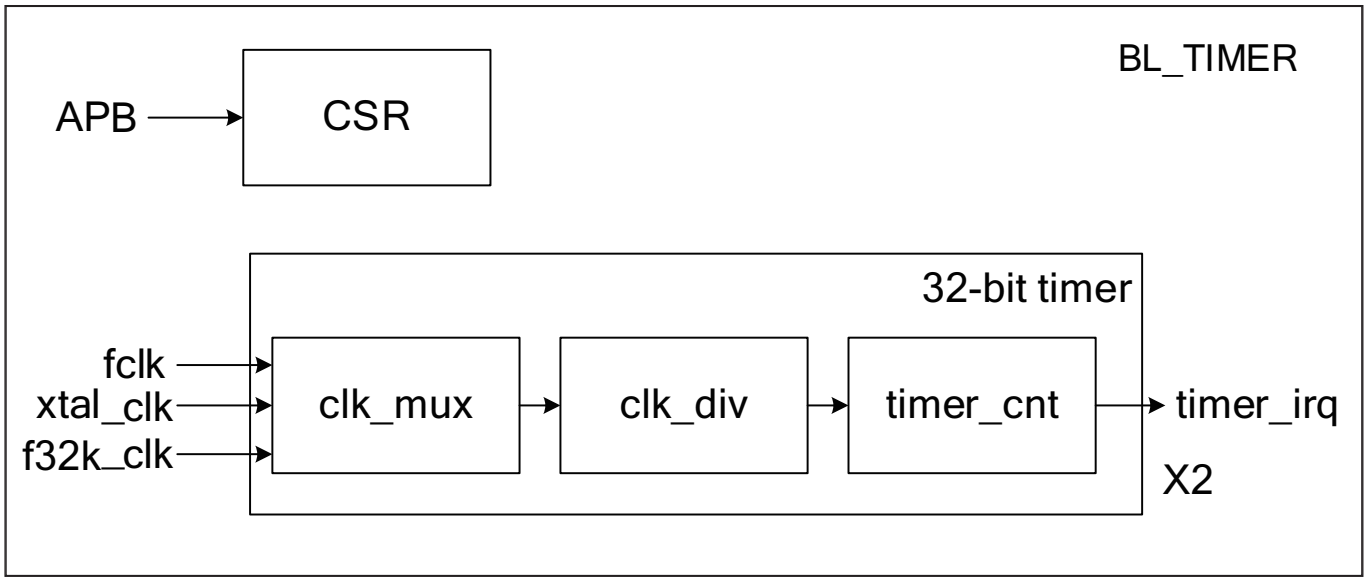


Figure 13.1: Timer block diagram

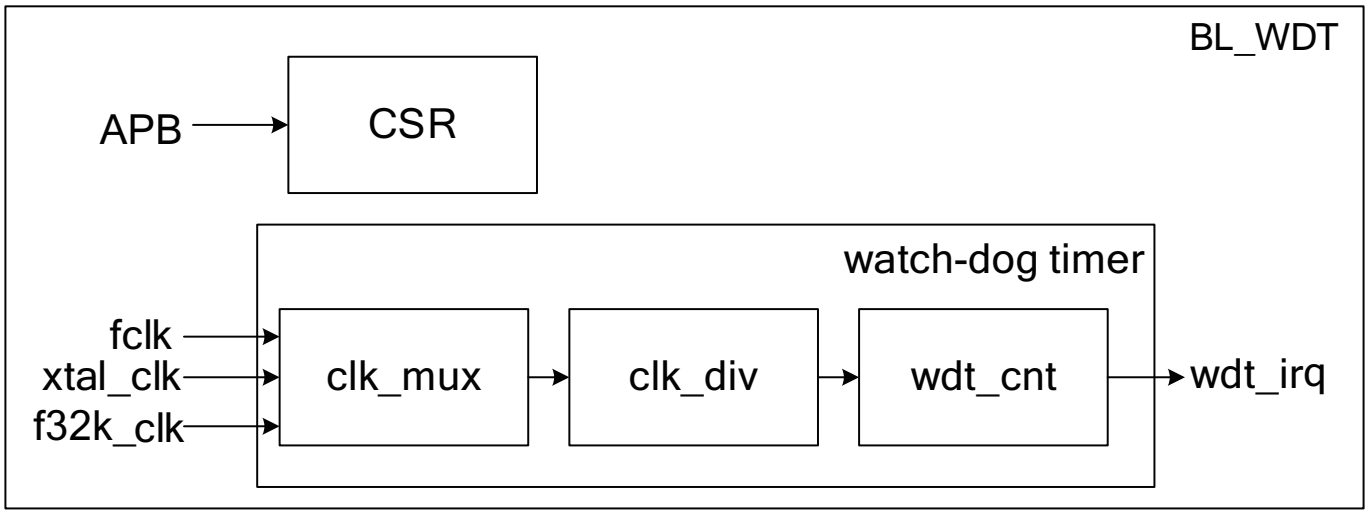


Figure 13.2: Watchdog timer block diagram

13.2 Main features

- Multiple clock sources, up to 160M clock
- 8-bit clock divider with a division factor of 1-256.
- Two 32-bit timers
- Each timer contains three alarm value settings, which can be set independently to alarm when each alarm value overflows
- Support Free Run mode and Pre_load mode
- 16-bit watchdog timer
- Supports write password protection to prevent system abnormalities caused by incorrect settings
- Support two watchdog overflow methods: interrupt or reset

13.3 Function description

13.3.1 8-bit divider

There are three types of Watchdog timer clocks:

- Fclk–System master clock
- 32K–32K clock
- Xtal–External crystal

There are four timer clock sources:

- Fclk–System master clock
- 32K–32K clock
- 1K–1K clock (32K frequency division)
- Xtal–External crystal

Each counter has its own 8-bit frequency divider. The selected clock can be divided by 1-256 through APB. Specifically, when it is set to 0, it means no frequency division, and when it is set to 1, it divides it by 2. , The maximum frequency division coefficient is 256, the counter will use the divided clock as the unit of the counting cycle, each time a counting cycle is increased by one.

13.3.2 General timer operating mode

Each general-purpose timer includes three comparators, a counter and a preload register. When the clock source is set and the timer is started, the counter starts to count up. When the counter value is equal to the comparator, the comparison is performed. When the flag is set, a compare interrupt is generated.

The initial value of the counter depends on the timing mode. In FreeRun mode, the initial value of the counter is 0, and then counts up. When it reaches the maximum value, it starts counting from 0 again.

In PreLoad mode, the initial value of the counter is the value of the PreLoad register and then counts up. When the PreLoad condition is met, the value of the counter is set to the value of the PreLoad register, and then the counter starts to count up again. During the counting process, once the value of the counter matches one of the three comparators, the comparator's comparison flag will be set and a corresponding comparison interrupt can be generated.

If the value of the preload register is 10, the value of Comparator 0 is 13, the value of Comparator 1 is 16, and the value of Comparator 2 is 19, the working sequence of the timer in PreLoad mode is as follows:

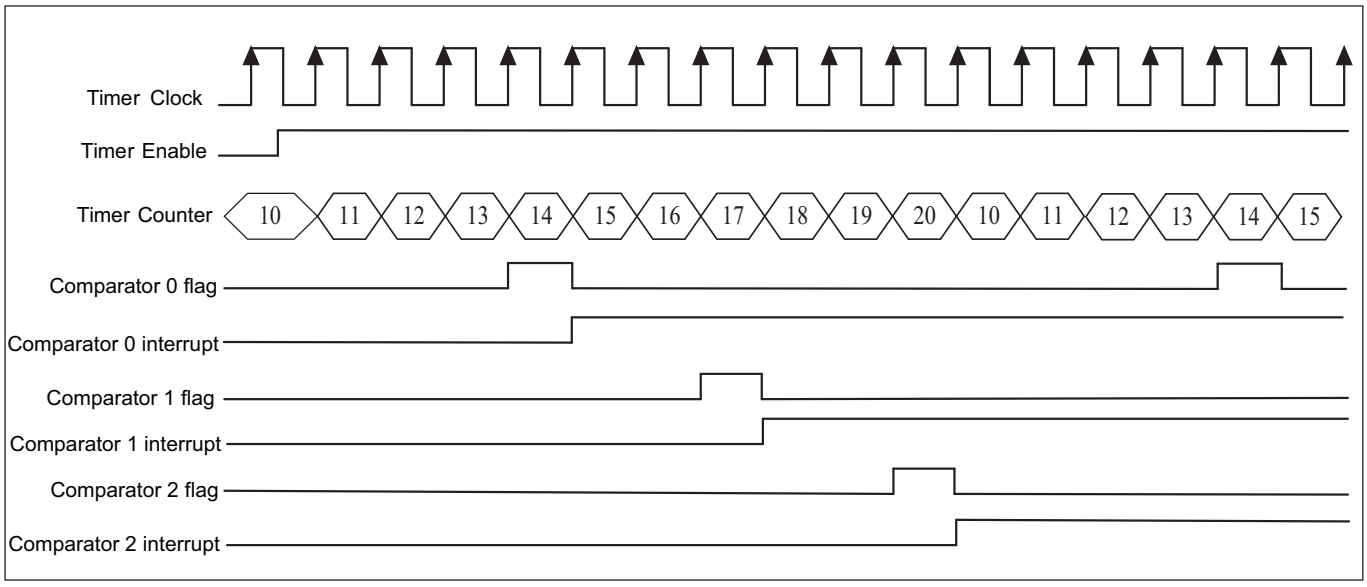


Figure 13.3: Timer Preload

In FreeRun mode, the timer working sequence is basically the same as PreLoad, the difference is that the counter will start to accumulate from 0 to the maximum value. The mechanism of the generated compare flags and compare interrupts is the same as in FreeRun mode.

13.3.3 Watchdog timer operating mode

The watchdog timer includes a counter and a comparator. The counter counts up from 0. If the counter is reset (feed the dog), it starts counting up from 0 again. When the counter value is equal to the comparator, a comparison interrupt signal or a system reset signal will be generated, and the user can choose to use one of them as required.

The watchdog counter is incremented by one in each counting cycle unit. Software can reset the watchdog counter to zero at any point in time through the APB.

If the value of the comparator is 6, the working sequence of Watchdog is shown in the figure below:

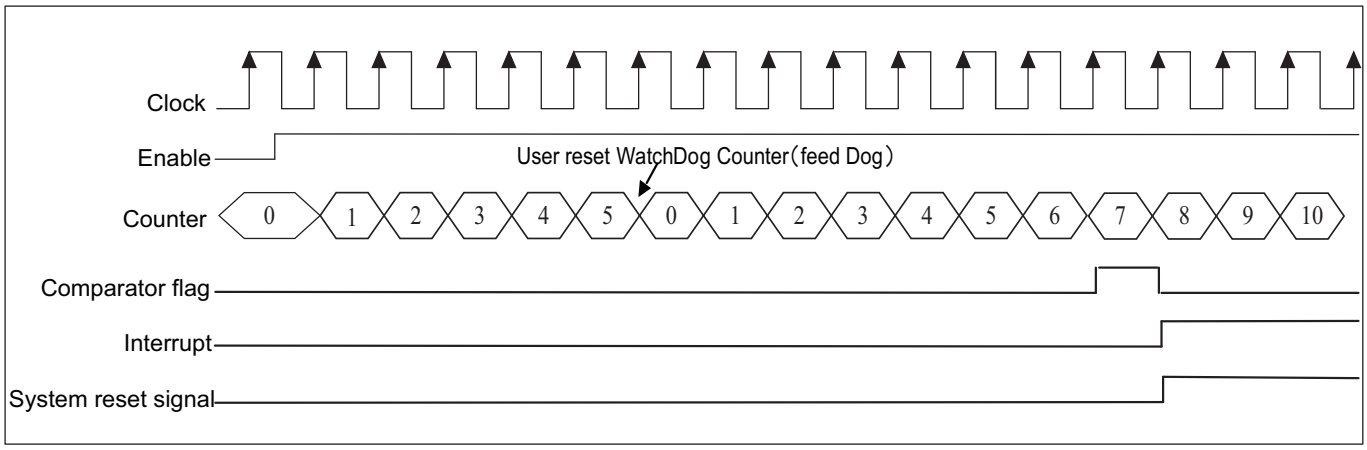


Figure 13.4: Watchdog timing

13.3.4 Alarm setting

Each counter has three comparison values, and can set whether each comparison value triggers an alarm interrupt. When the counter matches the comparison value and the setting will alarm, the counter will notify the processor through the interrupt.

The software can read through the APB whether an alarm has occurred and which comparison value triggered the alarm interrupt. When the alarm interrupt is cleared, the alarm status is also cleared simultaneously.

13.3.5 Watchdog alarm

A comparison value can be set for each counter. When the software fails to reset the watchdog counter to zero due to a system error, which causes the watchdog counter to exceed the comparison value, a watchdog alarm is triggered. There are two types of alarms. The first is to perform necessary actions through interrupt notification software. The second is to enter the system watchdog reset. When the watchdog reset is triggered, it will notify the system reset controller and prepare for system reset. When everything is ready, enter the system watchdog reset. It is worth noting

that software can read the WSR register through APB to know if a watchdog system reset has occurred.

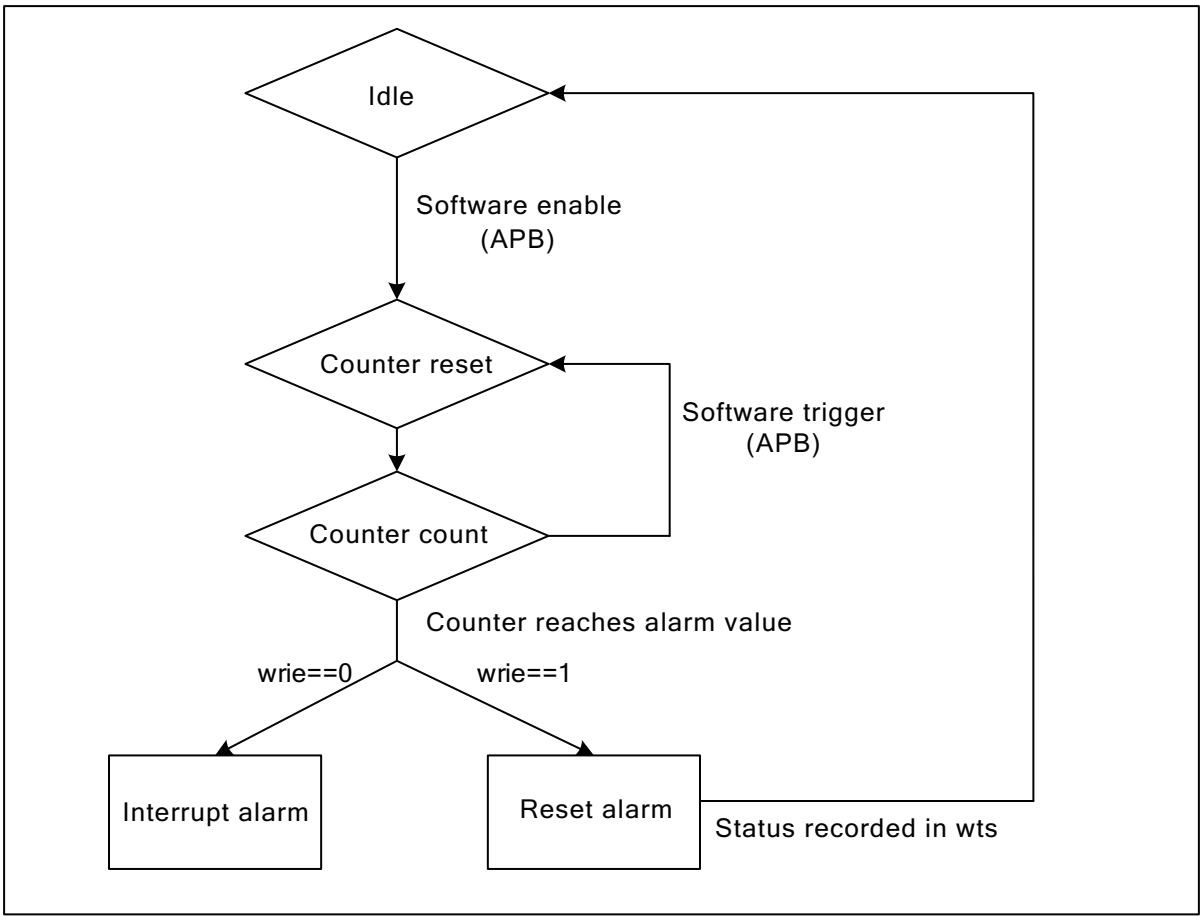


Figure 13.5: Watchdog alarm mechanism

13.4 Register description

Name	Description
TCCR	Timer clock source configuration register
TMR2_0	Timer2 match register 0
TMR2_1	Timer2 match register 1
TMR2_2	Timer2 match register 2
TMR3_0	Timer3 match register 0
TMR3_1	Timer3 match register 1
TMR3_2	Timer3 match register 2
TCR2	Timer2 counter register
TCR3	Timer3 counter register

Name	Description
TMSR2	Timer2 match register status
TMSR3	Timer3 match register status
TIER2	Timer2 match interrupt enable register
TIER3	Timer3 match interrupt enable register
TPLVR2	Timer2 pre-load value register
TPLVR3	Timer3 pre-load value register
TPLCR2	Timer2 pre-load control register
TPLCR3	Timer3 pre-load control register
WMER	WDT reset/interrupt mode register
WMR	WDT counter match value register
WVR	WDT counter value register
WSR	WDT timer reset indication register
TICR2	Timer2 Interrupt clear control register
TICR3	Timer3 Interrupt clear control register
WICR	WDT Interrupt clear register
TCER	Timer count enable register
TCMR	Timer count mode register
TILR2	Timer2 match interrupt mode register
TILR3	Timer3 match interrupt mode register
WCR	WDT timer count reset register
WFAR	WDT access key1 register
WSAR	WDT access key2 register
TCVWR2	Timer2 capture value of counter register
TCVWR3	Timer3 capture value of counter register
TCVSYN2	Timer2 synchronous value of counter register
TCVSYN3	Timer3 synchronous value of counter register
TCDR	WDT/Timer clock division register

13.4.1 TCCR

Address：0x4000a500

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						CSWDT		RSVD	CS2		RSVD	CS1		RSVD	

Bits	Name	Type	Reset	Description
31:10	RSVD			
9:8	CSWDT	R/W	2'd0	Clock Source for Timer #1/#2/#3/WDT 2'd0 - fclk 2'd1 - f32k_clk 2'd2 - 1 kHz 2'd3 - PLL 32MHz
7	RSVD			
6:5	CS2	R/W	2'd0	Clock Source for Timer #1/#2/#3/WDT 2'd0 - fclk 2'd1 - f32k_clk 2'd2 - 1 kHz 2'd3 - PLL 32MHz
4	RSVD			
3:2	CS1	R/W	2'd0	Clock Source for Timer #1/#2/#3/WDT 2'd0 - fclk 2'd1 - f32k_clk 2'd2 - 1 kHz 2'd3 - PLL 32MHz
1:0	RSVD			

13.4.2 TMR2_0

Address：0x4000a510

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR20															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR20															

Bits	Name	Type	Reset	Description
31:0	TMR20	R/W	32'hffffffff	Timer2 match register 0

13.4.3 TMR2_1

Address：0x4000a514

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR21															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR21															

Bits	Name	Type	Reset	Description
31:0	TMR21	R/W	32'hffffffff	Timer2 match register 1

13.4.4 TMR2_2

Address：0x4000a518

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR22															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR22															

Bits	Name	Type	Reset	Description
31:0	TMR22	R/W	32'hffffffff	Timer2 match register 2

13.4.5 TMR3_0

Address：0x4000a51c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR30															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR30															

Bits	Name	Type	Reset	Description
31:0	TMR30	R/W	32'hffffffff	Timer3 match register 0

13.4.6 TMR3_1

Address：0x4000a520

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR31															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR31															

Bits	Name	Type	Reset	Description
31:0	TMR31	R/W	32'hffffffff	Timer3 match register 1

13.4.7 TMR3_2

Address：0x4000a524

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TMR32															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR32															

Bits	Name	Type	Reset	Description
31:0	TMR32	R/W	32'hffffffff	Timer3 match register 2

13.4.8 TCR2

Address：0x4000a52c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCR2COUT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCR2COUT															

Bits	Name	Type	Reset	Description
31:0	TCR2COUT	R	32'h0	Timer2 counter register

13.4.9 TCR3

Address：0x4000a530

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCR3COUT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCR3COUT															

Bits	Name	Type	Reset	Description
31:0	TCR3COUT	R	32'h0	Timer3 counter register

13.4.10 TMSR2

Address：0x4000a538

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													T2MR2S	T2MR1S	T2MR0S

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	T2MR2S	R	1'b0	Timer2 match register 2 status/Clear interrupt would also clear this bit
1	T2MR1S	R	1'b0	Timer2 match register 1 status/Clear interrupt would also clear this bit
0	T2MR0S	R	1'b0	Timer2 match register 0 status/Clear interrupt would also clear this bit

13.4.11 TMSR3

Address：0x4000a53c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													T3MR2S	T3MR1S	T3MR0S

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	T3MR2S	R	1'b0	Timer3 match register 2 status/Clear interrupt would also clear this bit
1	T3MR1S	R	1'b0	Timer3 match register 1 status/Clear interrupt would also clear this bit
0	T3MR0S	R	1'b0	Timer3 match register 0 status/Clear interrupt would also clear this bit

13.4.12 TIER2

Address：0x4000a544

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TIER22	TIER21	TIER20

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TIER22	R/W	1'b0	Timer2 match register 2 interrupt enable register
1	TIER21	R/W	1'b0	Timer2 match register 1 interrupt enable register
0	TIER20	R/W	1'b0	Timer2 match register 0 interrupt enable register

13.4.13 TIER3

Address：0x4000a548

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TIER32	TIER31	TIER30

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TIER32	R/W	1'b0	Timer3 match register 2 interrupt enable register
1	TIER31	R/W	1'b0	Timer3 match register 1 interrupt enable register
0	TIER30	R/W	1'b0	Timer3 match register 0 interrupt enable register

13.4.14 TPLVR2

Address：0x4000a550

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPLVR2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPLVR2															

Bits	Name	Type	Reset	Description
31:0	TPLVR2	R/W	32'h0	Timer2 pre-load value register

13.4.15 TPLVR3

Address：0x4000a554

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TPLVR3															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPLVR3															

Bits	Name	Type	Reset	Description
31:0	TPLVR3	R/W	32'h0	Timer3 pre-load value register

13.4.16 TPLCR2

Address：0x4000a55c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														TPLCR2	

Bits	Name	Type	Reset	Description
31:2	RSVD			
1:0	TPLCR2	R/W	2'h0	Timer2 pre-load control register 2'd0 - No pre-load 2'd1 - Pre-load with match comparator 0 2'd2 - Pre-load with match comparator 1 2'd3 - Pre-load with match comparator 2

13.4.17 TPLCR3

Address：0x4000a560

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														TPLCR3	

Bits	Name	Type	Reset	Description
31:2	RSVD			
1:0	TPLCR3	R/W	2'h0	Timer3 pre-load control register 2'd0 - No pre-load 2'd1 - Pre-load with match comparator 0 2'd2 - Pre-load with match comparator 1 2'd3 - Pre-load with match comparator 2

13.4.18 WMER

Address：0x4000a564

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

Bits	Name	Type	Reset	Description
31:2	RSVD			
1	WRIE	R/W	1'b0	WDT reset/interrupt mode register 1'b0 - WDT expiration to generate interrupt 1'b1 - WDT expiration to generate reset source
0	WE	R/W	1'b0	WDT enable register

13.4.19 WMR

Address：0x4000a568

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	WMR	R/W	16'hffff	WDT counter match value register

13.4.20 WVR

Address：0x4000a56c

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	WVR	R	16'h0	WDT counter value register

Bits	Name	Type	Reset	Description

13.4.21 WSR

Address：0x4000a570

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

Bits	Name	Type	Reset	Description
31:1	RSVD			
0	WTS	R/W	1’b0	WDT timer reset indication, Indicates that reset was caused by the WDT. (Write)1’b0 - clear the WDT reset status (Write)1’b1 - no affect (Read)1’b0 - Watchdog timer did not cause reset because this bit was cleare (Read)1’b1 - Watchdog timer caused reset

13.4.22 TICR2

Address：0x4000a578

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TCLR22	TCLR21	TCLR20

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TCLR22	W	1’b0	Timer2 Interrupt clear for match comparator 2
1	TCLR21	W	1’b0	Timer2 Interrupt clear for match comparator 1
0	TCLR20	W	1’b0	Timer2 Interrupt clear for match comparator 0

13.4.23 TICR3

Address：0x4000a57c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TCLR32	TCLR31	TCLR30

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TCLR32	W	1'b0	Timer3 Interrupt clear for match comparator 2
1	TCLR31	W	1'b0	Timer3 Interrupt clear for match comparator 1
0	TCLR30	W	1'b0	Timer3 Interrupt clear for match comparator 0

13.4.24 WICR

Address：0x4000a580

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

Bits	Name	Type	Reset	Description
31:1	RSVD			
0	WICLR	W	1'b0	WDT Interrupt clear register

13.4.25 TCER

Address：0x4000a584

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TIM3 EN	TIM2 EN	RSVD

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TIM3EN	R/W	1'b0	Timer3 count enable
1	TIM2EN	R/W	1'b0	Timer2 count enable
0	RSVD			

13.4.26 TCMR

Address：0x4000a588

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TIM3 MODE	TIM2 MODE	RSVD

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TIM3MODE	R/W	1'b0	Timer1/2/3 count mode register 1'b0 - pre-load mode 1'b1 - free run mode
1	TIM2MODE	R/W	1'b0	Timer1/2/3 count mode register 1'b0 - pre-load mode 1'b1 - free run mode
0	RSVD			

13.4.27 TILR2

Address：0x4000a590

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TILR22	TILR21	TILR20

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TILR22	R/W	1'b0	Timer2 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt
1	TILR21	R/W	1'b0	Timer2 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt
0	TILR20	R/W	1'b0	Timer2 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt

13.4.28 TILR3

Address：0x4000a594

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD													TILR32	TILR31	TILR30

Bits	Name	Type	Reset	Description
31:3	RSVD			
2	TILR32	R/W	1'b0	Timer3 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt
1	TILR31	R/W	1'b0	Timer3 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt
0	TILR30	R/W	1'b0	Timer3 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt

Bits	Name	Type	Reset	Description

13.4.29 WCR

Address：0x4000a598

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

Bits	Name	Type	Reset	Description
31:1	RSVD			
0	WCR	W	1’b0	WDT timer count reset register

13.4.30 WFAR

Address：0x4000a59c

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	WFAR	W	16’b0	WDT access key1 - 16’hBABA

13.4.31 WSAR

Address：0x4000a5a0

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

Bits	Name	Type	Reset	Description
31:16	RSVD			
15:0	WSAR	W	16'b0	WDT access key2 - 16'hEB10

13.4.32 TCVWR2

Address：0x4000a5a8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCVWR2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCVWR2															

Bits	Name	Type	Reset	Description
31:0	TCVWR2	R	32'h0	Timer2 capture value of counter

13.4.33 TCVWR3

Address：0x4000a5ac

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCVWR3															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCVWR3															

Bits	Name	Type	Reset	Description
31:0	TCVWR3	R	32'h0	Timer3 capture value of counter

13.4.34 TCVSYN2

Address：0x4000a5b4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCVSYN2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCVSYN2															

Bits	Name	Type	Reset	Description
31:0	TCVSYN2	R	32'h0	Timer2 synchronous value of counter

13.4.35 TCVSYN3

Address：0x4000a5b8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCVSYN3															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCVSYN3															

Bits	Name	Type	Reset	Description
31:0	TCVSYN3	R	32'h0	Timer3 synchronous value of counter

13.4.36 TCDR

Address：0x4000a5bc

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WCDR								TCDR3							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCDR2								RSVD							

Bits	Name	Type	Reset	Description
31:24	WCDR	R/W	8'h0	WDT clock division value register
23:16	TCDR3	R/W	8'h0	Timer3 clock division value register
15:8	TCDR2	R/W	8'h0	Timer2 clock division value register
7:0	RSVD			

14 Revision history

Table 14.1: Document revision history

Date	Revision	Changes
2020/2/13	0.9	Initial release
2020/4/20	1.0	Add related content of HBN register
2020/8/26	1.1	Add ADC and DAC
2020/12/14	1.2	Add the introduction of interrupt sources and the maximum operating speed of peripherals