



# BL702/704/706

## 参考手册

*Version: 1.2*

*Copyright @ 2023*

[www.bouffalolab.com](http://www.bouffalolab.com)

# 目录

1 系统和存储器概述 . . . . .	20
1.1 简介 . . . . .	20
1.2 主要特征 . . . . .	20
1.3 功能描述 . . . . .	20
2 复位和时钟 . . . . .	23
2.1 简介 . . . . .	23
2.2 复位源 . . . . .	23
2.3 时钟源 . . . . .	24
3 GLB . . . . .	26
3.1 简介 . . . . .	26
3.2 功能描述 . . . . .	26
3.2.1 时钟管理 . . . . .	26
3.2.2 复位管理 . . . . .	26
3.2.3 总线管理 . . . . .	27
3.2.4 内存管理 . . . . .	27
3.2.5 GPIO 概述 . . . . .	28
3.2.6 GPIO 主要特点 . . . . .	28
3.2.7 GPIO 功能描述 . . . . .	28
3.2.8 GPIO 功能设定 . . . . .	29
3.2.9 GPIO 输出设置 . . . . .	33
3.2.10 GPIO 输入设置 . . . . .	33
3.2.11 GPIO 可选功能设置 . . . . .	33
3.2.12 GPIO 中断设置 . . . . .	33
3.3 寄存器描述 . . . . .	34
3.3.1 GPIO_CFGCTL0 . . . . .	35
3.3.2 GPIO_CFGCTL1 . . . . .	36

3.3.3	GPIO_CFGCTL2 . . . . .	37
3.3.4	GPIO_CFGCTL3 . . . . .	38
3.3.5	GPIO_CFGCTL4 . . . . .	39
3.3.6	GPIO_CFGCTL5 . . . . .	40
3.3.7	GPIO_CFGCTL6 . . . . .	41
3.3.8	GPIO_CFGCTL7 . . . . .	42
3.3.9	GPIO_CFGCTL8 . . . . .	43
3.3.10	GPIO_CFGCTL9 . . . . .	44
3.3.11	GPIO_CFGCTL10 . . . . .	45
3.3.12	GPIO_CFGCTL11 . . . . .	46
3.3.13	GPIO_CFGCTL12 . . . . .	47
3.3.14	GPIO_CFGCTL13 . . . . .	48
3.3.15	GPIO_CFGCTL14 . . . . .	49
3.3.16	GPIO_CFGCTL15 . . . . .	50
3.3.17	GPIO_CFGCTL30 . . . . .	52
3.3.18	GPIO_CFGCTL32 . . . . .	54
3.3.19	GPIO_CFGCTL34 . . . . .	55
3.3.20	GPIO_CFGCTL35 . . . . .	58
3.3.21	GPIO_INT_STAT1 . . . . .	58
3.3.22	GPIO_INT_CLR1 . . . . .	58
3.3.23	GPIO_INT_MODE_SET1 . . . . .	59
3.3.24	GPIO_INT_MODE_SET2 . . . . .	59
3.3.25	GPIO_INT_MODE_SET3 . . . . .	59
3.3.26	GPIO_INT_MODE_SET4 . . . . .	60
3.3.27	GPIO_INT2_MASK1 . . . . .	60
3.3.28	GPIO_INT2_STAT1 . . . . .	61
3.3.29	GPIO_INT2_CLR1 . . . . .	61
3.3.30	GPIO_INT2_MODE_SET1 . . . . .	61
3.3.31	GPIO_INT2_MODE_SET2 . . . . .	62
3.3.32	GPIO_INT2_MODE_SET3 . . . . .	62
3.3.33	GPIO_INT2_MODE_SET4 . . . . .	63
4	ADC . . . . .	64
4.1	简介 . . . . .	64
4.2	主要特点 . . . . .	64
4.3	功能描述 . . . . .	65
4.3.1	ADC 引脚和内部信号 . . . . .	65
4.3.2	ADC 通道 . . . . .	66
4.3.3	ADC 时钟 . . . . .	67
4.3.4	ADC 转换模式 . . . . .	68

4.3.5	ADC 结果 . . . . .	68
4.3.6	ADC 异常中断 . . . . .	69
4.3.7	ADC FIFO 与阈值中断 . . . . .	69
4.3.8	ADC 设置流程 . . . . .	70
4.3.9	VBAT 测量 . . . . .	71
4.3.10	TSEN 测量 . . . . .	71
4.4	寄存器描述 . . . . .	71
4.4.1	gpadc_config . . . . .	72
4.4.2	gpadc_dma_rdata . . . . .	73
4.4.3	gpdac_config . . . . .	74
4.4.4	gpdac_dma_config . . . . .	75
4.4.5	gpdac_dma_wdata . . . . .	75
4.4.6	gpadc_reg_cmd . . . . .	76
4.4.7	gpadc_reg_config1 . . . . .	79
4.4.8	gpadc_reg_config2 . . . . .	82
4.4.9	gpadc_reg_scn_pos1 . . . . .	84
4.4.10	gpadc_reg_scn_pos2 . . . . .	84
4.4.11	gpadc_reg_scn_neg1 . . . . .	85
4.4.12	gpadc_reg_scn_neg2 . . . . .	86
4.4.13	gpadc_reg_status . . . . .	87
4.4.14	gpadc_reg_isr . . . . .	87
4.4.15	gpadc_reg_raw_result . . . . .	88
4.4.16	gpadc_reg_define . . . . .	88
5	DAC . . . . .	89
5.1	简介 . . . . .	89
5.2	主要特点 . . . . .	89
5.3	功能描述 . . . . .	89
5.4	寄存器描述 . . . . .	91
5.4.1	gpadc_config . . . . .	92
5.4.2	gpadc_dma_rdata . . . . .	93
5.4.3	gpdac_config . . . . .	93
5.4.4	gpdac_dma_config . . . . .	94
5.4.5	gpdac_dma_wdata . . . . .	95
5.4.6	gpdac_ctrl . . . . .	95
5.4.7	gpdac_actrl . . . . .	96
5.4.8	gpdac_bctrl . . . . .	97
5.4.9	gpdac_ctrl . . . . .	97
5.4.10	gpdac_actrl . . . . .	98
5.4.11	gpdac_bctrl . . . . .	99

6 DMA . . . . .	100
6.1 简介 . . . . .	100
6.2 主要特征 . . . . .	100
6.3 功能描述 . . . . .	101
6.3.1 工作原理 . . . . .	101
6.3.2 DMA 通道配置 . . . . .	102
6.3.3 外设支持 . . . . .	102
6.3.4 链表模式 . . . . .	103
6.3.5 DMA 中断 . . . . .	104
6.4 传输模式 . . . . .	104
6.4.1 内存到内存 . . . . .	104
6.4.2 内存到外设 . . . . .	104
6.4.3 外设到内存 . . . . .	105
6.4.4 外设到外设 . . . . .	105
6.5 寄存器描述 . . . . .	105
6.5.1 DMA_IntStatus . . . . .	108
6.5.2 DMA_IntTCStatus . . . . .	108
6.5.3 DMA_IntTCClear . . . . .	108
6.5.4 DMA_IntErrorStatus . . . . .	109
6.5.5 DMA_IntErrClr . . . . .	109
6.5.6 DMA_RawIntTCStatus . . . . .	110
6.5.7 DMA_RawIntErrorStatus . . . . .	110
6.5.8 DMA_EnbldChns . . . . .	111
6.5.9 DMA_SoftBReq . . . . .	111
6.5.10 DMA_SoftSReq . . . . .	111
6.5.11 DMA_SoftLBReq . . . . .	112
6.5.12 DMA_SoftLSReq . . . . .	112
6.5.13 DMA_Config . . . . .	112
6.5.14 DMA_Sync . . . . .	113
6.5.15 DMA_C0SrcAddr . . . . .	113
6.5.16 DMA_C0DstAddr . . . . .	114
6.5.17 DMA_C0LLI . . . . .	114
6.5.18 DMA_C0Control . . . . .	114
6.5.19 DMA_C0Config . . . . .	115
6.5.20 DMA_C1SrcAddr . . . . .	117
6.5.21 DMA_C1DstAddr . . . . .	117
6.5.22 DMA_C1LLI . . . . .	117
6.5.23 DMA_C1Control . . . . .	118
6.5.24 DMA_C1Config . . . . .	119

6.5.25	DMA_C2SrcAddr . . . . .	119
6.5.26	DMA_C2DstAddr . . . . .	120
6.5.27	DMA_C2LLI . . . . .	120
6.5.28	DMA_C2Control . . . . .	120
6.5.29	DMA_C2Config . . . . .	121
6.5.30	DMA_C3SrcAddr . . . . .	122
6.5.31	DMA_C3DstAddr . . . . .	122
6.5.32	DMA_C3LLI . . . . .	123
6.5.33	DMA_C3Control . . . . .	123
6.5.34	DMA_C3Config . . . . .	124
6.5.35	DMA_C4SrcAddr . . . . .	125
6.5.36	DMA_C4DstAddr . . . . .	125
6.5.37	DMA_C4LLI . . . . .	125
6.5.38	DMA_C4Control . . . . .	126
6.5.39	DMA_C4Config . . . . .	127
6.5.40	DMA_C5SrcAddr . . . . .	127
6.5.41	DMA_C5DstAddr . . . . .	128
6.5.42	DMA_C5LLI . . . . .	128
6.5.43	DMA_C5Control . . . . .	128
6.5.44	DMA_C5Config . . . . .	129
6.5.45	DMA_C6SrcAddr . . . . .	130
6.5.46	DMA_C6DstAddr . . . . .	130
6.5.47	DMA_C6LLI . . . . .	131
6.5.48	DMA_C6Control . . . . .	131
6.5.49	DMA_C6Config . . . . .	132
6.5.50	DMA_C7SrcAddr . . . . .	133
6.5.51	DMA_C7DstAddr . . . . .	133
6.5.52	DMA_C7LLI . . . . .	133
6.5.53	DMA_C7Control . . . . .	134
6.5.54	DMA_C7Config . . . . .	135
7	L1C . . . . .	136
7.1	简介 . . . . .	136
7.2	主要特征 . . . . .	137
7.3	功能描述 . . . . .	137
7.3.1	TCM 与 Cache RAM 资源相互变换 . . . . .	137
7.3.2	缓存 (Cache) . . . . .	137
7.4	寄存器描述 . . . . .	138
7.4.1	I1c_config . . . . .	139
7.4.2	hit_cnt_lsb . . . . .	140

7.4.3	hit_cnt_msb . . . . .	140
7.4.4	miss_cnt . . . . .	140
8	IR . . . . .	141
8.1	简介 . . . . .	141
8.2	主要特征 . . . . .	141
8.3	功能描述 . . . . .	141
8.3.1	固定协议接收 . . . . .	141
8.3.2	脉冲宽度接收 . . . . .	143
8.3.3	普通发送模式 . . . . .	143
8.3.4	脉冲宽度发送 . . . . .	143
8.3.5	载波调制 . . . . .	143
8.3.6	IR 中断 . . . . .	144
8.4	寄存器描述 . . . . .	144
8.4.1	irtx_config . . . . .	145
8.4.2	irtx_int_sts . . . . .	146
8.4.3	irtx_data_word0 . . . . .	147
8.4.4	irtx_data_word1 . . . . .	147
8.4.5	irtx_pulse_width . . . . .	147
8.4.6	irtx_pw . . . . .	148
8.4.7	irtx_swm_pw_0 . . . . .	149
8.4.8	irtx_swm_pw_1 . . . . .	149
8.4.9	irtx_swm_pw_2 . . . . .	149
8.4.10	irtx_swm_pw_3 . . . . .	150
8.4.11	irtx_swm_pw_4 . . . . .	150
8.4.12	irtx_swm_pw_5 . . . . .	151
8.4.13	irtx_swm_pw_6 . . . . .	151
8.4.14	irtx_swm_pw_7 . . . . .	151
8.4.15	irrx_config . . . . .	152
8.4.16	irrx_int_sts . . . . .	153
8.4.17	irrx_pw_config . . . . .	153
8.4.18	irrx_data_count . . . . .	154
8.4.19	irrx_data_word0 . . . . .	154
8.4.20	irrx_data_word1 . . . . .	155
8.4.21	irrx_swm_fifo_config_0 . . . . .	155
8.4.22	irrx_swm_fifo_rdata . . . . .	156
9	SPI . . . . .	157
9.1	简介 . . . . .	157
9.2	主要特征 . . . . .	157

9.3 功能描述 . . . . .	158
9.3.1 时钟控制 . . . . .	158
9.3.2 主设备持续传输模式 . . . . .	158
9.3.3 接收过滤功能 . . . . .	158
9.3.4 接收去差错功能 . . . . .	159
9.3.5 从模式超时机制 . . . . .	159
9.3.6 I/O 传输模式 . . . . .	159
9.3.7 DMA 传输模式 . . . . .	159
9.3.8 SPI 中断 . . . . .	160
9.4 寄存器描述 . . . . .	160
9.4.1 spi_config . . . . .	161
9.4.2 spi_int_sts . . . . .	162
9.4.3 spi_bus_busy . . . . .	164
9.4.4 spi_prd_0 . . . . .	164
9.4.5 spi_prd_1 . . . . .	165
9.4.6 spi_rxd_ignr . . . . .	165
9.4.7 spi_sto_value . . . . .	166
9.4.8 spi_fifo_config_0 . . . . .	166
9.4.9 spi_fifo_config_1 . . . . .	167
9.4.10 spi_fifo_wdata . . . . .	167
9.4.11 spi_fifo_rdata . . . . .	168
10 UART . . . . .	169
10.1 简介 . . . . .	169
10.2 主要特征 . . . . .	169
10.3 功能描述 . . . . .	170
10.3.1 数据格式描述 . . . . .	170
10.3.2 基本架构图 . . . . .	170
10.3.3 时钟源 . . . . .	170
10.3.4 波特率设定 . . . . .	171
10.3.5 发送器 . . . . .	171
10.3.6 接收器 . . . . .	172
10.3.7 自动波特率检测 . . . . .	172
10.3.8 硬件流控 . . . . .	173
10.3.9 LIN 传输模式 . . . . .	173
10.3.10 DMA 传输模式 . . . . .	173
10.3.11 UART 中断 . . . . .	174
10.4 寄存器描述 . . . . .	174
10.4.1 utx_config . . . . .	175
10.4.2 urx_config . . . . .	176

10.4.3	uart_bit_prd . . . . .	177
10.4.4	data_config . . . . .	177
10.4.5	utx_ir_position . . . . .	178
10.4.6	urx_ir_position . . . . .	178
10.4.7	urx_rto_timer . . . . .	179
10.4.8	uart_sw_mode . . . . .	179
10.4.9	uart_int_sts . . . . .	180
10.4.10	uart_int_mask . . . . .	181
10.4.11	uart_int_clear . . . . .	181
10.4.12	uart_int_en . . . . .	182
10.4.13	uart_status . . . . .	183
10.4.14	sts_urx_abr_prd . . . . .	183
10.4.15	uart_fifo_config_0 . . . . .	184
10.4.16	uart_fifo_config_1 . . . . .	184
10.4.17	uart_fifo_wdata . . . . .	185
10.4.18	uart_fifo_rdata . . . . .	185
11	I2C . . . . .	187
11.1	简介 . . . . .	187
11.2	主要特征 . . . . .	187
11.3	功能描述 . . . . .	187
11.3.1	起始和停止条件 . . . . .	188
11.3.2	数据传输格式 . . . . .	188
11.3.3	仲裁 . . . . .	189
11.4	I2C 时钟设定 . . . . .	190
11.5	I2C 配置流程 . . . . .	190
11.5.1	配置项 . . . . .	190
11.5.2	读写标志位 . . . . .	191
11.5.3	从设备地址 . . . . .	191
11.5.4	从设备寄存器地址 . . . . .	191
11.5.5	从设备寄存器地址长度 . . . . .	191
11.5.6	数据 . . . . .	191
11.5.7	数据长度 . . . . .	191
11.5.8	使能信号 . . . . .	191
11.6	FIFO 管理 . . . . .	192
11.7	搭配使用 DMA . . . . .	192
11.7.1	DMA 发送流程 . . . . .	193
11.7.2	DMA 接收流程 . . . . .	193
11.8	中断 . . . . .	194

11.9 寄存器描述 . . . . .	194
11.9.1 i2c_config . . . . .	195
11.9.2 i2c_int_sts . . . . .	196
11.9.3 i2c_sub_addr . . . . .	197
11.9.4 i2c_bus_busy . . . . .	198
11.9.5 i2c_prd_start . . . . .	198
11.9.6 i2c_prd_stop . . . . .	199
11.9.7 i2c_prd_data . . . . .	199
11.9.8 i2c_fifo_config_0 . . . . .	200
11.9.9 i2c_fifo_config_1 . . . . .	200
11.9.10 i2c_fifo_wdata . . . . .	201
11.9.11 i2c_fifo_rdata . . . . .	201
12 PWM . . . . .	202
12.1 简介 . . . . .	202
12.2 主要特征 . . . . .	202
12.3 功能描述 . . . . .	202
12.3.1 时钟与分频器 . . . . .	202
12.3.2 脉冲产生原理 . . . . .	203
12.3.3 PWM 中断 . . . . .	204
12.4 寄存器描述 . . . . .	204
12.4.1 pwm_int_config . . . . .	206
12.4.2 pwm0_clkdiv . . . . .	206
12.4.3 pwm0_thre1 . . . . .	206
12.4.4 pwm0_thre2 . . . . .	207
12.4.5 pwm0_period . . . . .	207
12.4.6 pwm0_config . . . . .	208
12.4.7 pwm0_interrupt . . . . .	209
12.4.8 pwm1_clkdiv . . . . .	209
12.4.9 pwm1_thre1 . . . . .	209
12.4.10 pwm1_thre2 . . . . .	210
12.4.11 pwm1_period . . . . .	210
12.4.12 pwm1_config . . . . .	211
12.4.13 pwm1_interrupt . . . . .	212
12.4.14 pwm2_clkdiv . . . . .	212
12.4.15 pwm2_thre1 . . . . .	212
12.4.16 pwm2_thre2 . . . . .	213
12.4.17 pwm2_period . . . . .	213
12.4.18 pwm2_config . . . . .	214
12.4.19 pwm2_interrupt . . . . .	215

12.4.20 pwm3_clkdiv . . . . .	215
12.4.21 pwm3_thre1 . . . . .	215
12.4.22 pwm3_thre2 . . . . .	216
12.4.23 pwm3_period . . . . .	216
12.4.24 pwm3_config . . . . .	217
12.4.25 pwm3_interrupt . . . . .	218
12.4.26 pwm4_clkdiv . . . . .	218
12.4.27 pwm4_thre1 . . . . .	218
12.4.28 pwm4_thre2 . . . . .	219
12.4.29 pwm4_period . . . . .	219
12.4.30 pwm4_config . . . . .	220
12.4.31 pwm4_interrupt . . . . .	221
<b>13 TIMER . . . . .</b>	<b>222</b>
<b>13.1 简介 . . . . .</b>	<b>222</b>
<b>13.2 主要特征 . . . . .</b>	<b>223</b>
<b>13.3 功能描述 . . . . .</b>	<b>223</b>
<b>13.3.1 8-bit 分频器 . . . . .</b>	<b>223</b>
<b>13.3.2 通用定时器工作原理 . . . . .</b>	<b>224</b>
<b>13.3.3 看门狗定时器工作原理 . . . . .</b>	<b>225</b>
<b>13.3.4 报警设定 . . . . .</b>	<b>225</b>
<b>13.3.5 看门狗报警 . . . . .</b>	<b>225</b>
<b>13.4 寄存器描述 . . . . .</b>	<b>226</b>
<b>13.4.1 TCCR . . . . .</b>	<b>228</b>
<b>13.4.2 TMR2_0 . . . . .</b>	<b>228</b>
<b>13.4.3 TMR2_1 . . . . .</b>	<b>229</b>
<b>13.4.4 TMR2_2 . . . . .</b>	<b>229</b>
<b>13.4.5 TMR2_0 . . . . .</b>	<b>229</b>
<b>13.4.6 TMR2_1 . . . . .</b>	<b>230</b>
<b>13.4.7 TMR2_2 . . . . .</b>	<b>230</b>
<b>13.4.8 TCR2 . . . . .</b>	<b>230</b>
<b>13.4.9 TCR3 . . . . .</b>	<b>231</b>
<b>13.4.10 TMSR2 . . . . .</b>	<b>231</b>
<b>13.4.11 TMSR3 . . . . .</b>	<b>232</b>
<b>13.4.12 TIER2 . . . . .</b>	<b>232</b>
<b>13.4.13 TIER3 . . . . .</b>	<b>233</b>
<b>13.4.14 TPLVR2 . . . . .</b>	<b>233</b>
<b>13.4.15 TPLVR3 . . . . .</b>	<b>233</b>
<b>13.4.16 TPLCR2 . . . . .</b>	<b>234</b>
<b>13.4.17 TPLCR3 . . . . .</b>	<b>234</b>

13.4.18 WMER . . . . .	235
13.4.19 WMR . . . . .	235
13.4.20 WVR . . . . .	236
13.4.21 WSR . . . . .	236
13.4.22 TICR2 . . . . .	237
13.4.23 TICR3 . . . . .	237
13.4.24 WICR . . . . .	238
13.4.25 TCER . . . . .	238
13.4.26 TCMR . . . . .	239
13.4.27 TILR2 . . . . .	239
13.4.28 TILR3 . . . . .	240
13.4.29 WCR . . . . .	240
13.4.30 WFAR . . . . .	241
13.4.31 WSAR . . . . .	241
13.4.32 TCVWR2 . . . . .	242
13.4.33 TCVWR3 . . . . .	242
13.4.34 TCVSYN2 . . . . .	242
13.4.35 TCVSYN3 . . . . .	243
13.4.36 TCDR . . . . .	243
14 QDEC . . . . .	244
14.1 简介 . . . . .	244
14.2 主要特征 . . . . .	244
14.3 功能描述 . . . . .	245
14.4 寄存器描述 . . . . .	246
14.4.1 qdec0_ctrl0 . . . . .	246
14.4.2 qdec0_ctrl1 . . . . .	248
14.4.3 qdec0_value . . . . .	248
14.4.4 qdec0_int_en . . . . .	249
14.4.5 qdec0_int_sts . . . . .	250
14.4.6 qdec0_int_clr . . . . .	250
14.4.7 qdec1_ctrl0 . . . . .	251
14.4.8 qdec1_ctrl1 . . . . .	252
14.4.9 qdec1_value . . . . .	253
14.4.10 qdec1_int_en . . . . .	253
14.4.11 qdec1_int_sts . . . . .	254
14.4.12 qdec1_int_clr . . . . .	254
14.4.13 qdec2_ctrl0 . . . . .	255
14.4.14 qdec2_ctrl1 . . . . .	256
14.4.15 qdec2_value . . . . .	257

14.4.16 qdec2_int_en . . . . .	258
14.4.17 qdec2_int_sts . . . . .	258
14.4.18 qdec2_int_clr . . . . .	259
15 KeyScan . . . . .	260
15.1 简介 . . . . .	260
15.2 主要特征 . . . . .	260
15.3 功能描述 . . . . .	260
15.3.1 可配置的行列数 . . . . .	260
15.3.2 GPIO 选择 . . . . .	260
15.3.3 键值 . . . . .	261
15.3.4 中断 . . . . .	261
15.4 寄存器描述 . . . . .	261
15.4.1 ks_ctrl . . . . .	261
15.4.2 ks_int_en . . . . .	262
15.4.3 ks_int_sts . . . . .	263
15.4.4 keycode_clr . . . . .	263
15.4.5 keycode_value . . . . .	264
16 I2S . . . . .	265
16.1 简介 . . . . .	265
16.2 主要特征 . . . . .	265
16.3 功能描述 . . . . .	265
16.4 寄存器描述 . . . . .	266
16.4.1 i2s_config . . . . .	266
16.4.2 i2s_int_sts . . . . .	268
16.4.3 i2s_bclk_config . . . . .	269
16.4.4 i2s_fifo_config_0 . . . . .	269
16.4.5 i2s_fifo_config_1 . . . . .	270
16.4.6 i2s_fifo_wdata . . . . .	271
16.4.7 i2s_fifo_rdata . . . . .	271
16.4.8 i2s_io_config . . . . .	272
17 Emac . . . . .	273
17.1 简介 . . . . .	273
17.2 主要特征 . . . . .	273
17.3 功能描述 . . . . .	274
17.4 时钟 . . . . .	275
17.5 收发缓冲描述符 (BD, Buffer Descriptor) . . . . .	275
17.6 PHY 交互 . . . . .	275
17.7 编程流程 . . . . .	276
17.7.1 PHY 初始化 . . . . .	276

17.7.2	发送数据帧 . . . . .	276
17.7.3	接收数据帧 . . . . .	277
17.8	寄存器描述 . . . . .	278
17.8.1	MODE . . . . .	278
17.8.2	INT_SOURCE . . . . .	280
17.8.3	INT_MASK . . . . .	282
17.8.4	IPGT . . . . .	283
17.8.5	PACKETLEN . . . . .	283
17.8.6	COLLCONFIG . . . . .	284
17.8.7	TX_BD_NUM . . . . .	285
17.8.8	MIIMODE . . . . .	285
17.8.9	MIICOMMAND . . . . .	286
17.8.10	MIIADDRESS . . . . .	287
17.8.11	MIITX_DATA . . . . .	287
17.8.12	MIIRX_DATA . . . . .	287
17.8.13	MIISTATUS . . . . .	288
17.8.14	MAC_ADDR0 . . . . .	288
17.8.15	MAC_ADDR1 . . . . .	289
17.8.16	HASH0_ADDR . . . . .	289
17.8.17	HASH1_ADDR . . . . .	290
17.8.18	TXCTRL . . . . .	290
18	USB . . . . .	291
18.1	简介 . . . . .	291
18.2	主要特征 . . . . .	291
18.3	功能描述 . . . . .	291
18.3.1	USB 使用步骤 . . . . .	291
18.3.2	部分寄存器配置及功能描述 . . . . .	292
18.3.3	USB 枚举阶段中断处理流程 . . . . .	293
18.3.4	各传输事务的寄存器操作流程 . . . . .	294
18.4	寄存器描述 . . . . .	297
18.4.1	usb_config . . . . .	299
18.4.2	usb_lpm_config . . . . .	300
18.4.3	usb_resume_config . . . . .	301
18.4.4	usb_frame_no . . . . .	301
18.4.5	usb_error . . . . .	302
18.4.6	usb_int_en . . . . .	303
18.4.7	usb_int_sts . . . . .	304
18.4.8	usb_int_mask . . . . .	306
18.4.9	usb_int_clear . . . . .	307

18.4.10 ep1_config . . . . .	309
18.4.11 ep2_config . . . . .	310
18.4.12 ep3_config . . . . .	311
18.4.13 ep4_config . . . . .	312
18.4.14 ep5_config . . . . .	313
18.4.15 ep6_config . . . . .	314
18.4.16 ep7_config . . . . .	315
18.4.17 ep0_fifo_config . . . . .	316
18.4.18 ep0_fifo_status . . . . .	316
18.4.19 ep0_tx_fifo_wdata . . . . .	317
18.4.20 ep0_rx_fifo_rdata . . . . .	318
18.4.21 ep1_fifo_config . . . . .	318
18.4.22 ep1_fifo_status . . . . .	319
18.4.23 ep1_tx_fifo_wdata . . . . .	320
18.4.24 ep1_rx_fifo_rdata . . . . .	320
18.4.25 ep2_fifo_config . . . . .	320
18.4.26 ep2_fifo_status . . . . .	321
18.4.27 ep2_tx_fifo_wdata . . . . .	322
18.4.28 ep2_rx_fifo_rdata . . . . .	322
18.4.29 ep3_fifo_config . . . . .	323
18.4.30 ep3_fifo_status . . . . .	324
18.4.31 ep3_tx_fifo_wdata . . . . .	324
18.4.32 ep3_rx_fifo_rdata . . . . .	325
18.4.33 ep4_fifo_config . . . . .	325
18.4.34 ep4_fifo_status . . . . .	326
18.4.35 ep4_tx_fifo_wdata . . . . .	327
18.4.36 ep4_rx_fifo_rdata . . . . .	327
18.4.37 ep5_fifo_config . . . . .	327
18.4.38 ep5_fifo_status . . . . .	328
18.4.39 ep5_tx_fifo_wdata . . . . .	329
18.4.40 ep5_rx_fifo_rdata . . . . .	329
18.4.41 ep6_fifo_config . . . . .	330
18.4.42 ep6_fifo_status . . . . .	331
18.4.43 ep6_tx_fifo_wdata . . . . .	331
18.4.44 ep6_rx_fifo_rdata . . . . .	332
18.4.45 ep7_fifo_config . . . . .	332
18.4.46 ep7_fifo_status . . . . .	333
18.4.47 ep7_tx_fifo_wdata . . . . .	334
18.4.48 ep7_rx_fifo_rdata . . . . .	334

---

18.4.49 xcvr_if_config . . . . .	334
19 版本信息 . . . . .	337

## 插图

2.1 复位源 . . . . .	24
2.2 时钟架构 . . . . .	25
3.1 GPIO 基本框图 . . . . .	29
4.1 ADC 基本框图 . . . . .	65
4.2 ADC 时钟 . . . . .	67
5.1 DAC 基本框图 . . . . .	90
6.1 DMA 框图 . . . . .	101
6.2 LLI 框架 . . . . .	103
7.1 LIC 架构 . . . . .	136
7.2 缓存架构 . . . . .	138
8.1 NEC 逻辑波形 . . . . .	142
8.2 NEC 协议波形 . . . . .	142
8.3 RC5 逻辑波形 . . . . .	142
8.4 RC5 协议波形 . . . . .	143
9.1 SPI 时序图 . . . . .	158
9.2 SPI Ignore 波形图 . . . . .	159
10.1 UART 数据格式 . . . . .	170
10.2 UART 时钟 . . . . .	170
10.3 UART 采样波形图 . . . . .	171
10.4 UART 固定字符模式波形图 . . . . .	172
10.5 UART 硬件流控图 . . . . .	173
11.1 I2C 起始和停止条件 . . . . .	188

---

11.2 I2C 数据传输格式 . . . . .	188
11.3 主发送和从接收的时序 . . . . .	189
11.4 主接收和从发送的时序 . . . . .	189
11.5 同时传输数据波形示意图 . . . . .	190
12.1 PWM 波形图 . . . . .	203
13.1 定时器框图 . . . . .	222
13.2 看门狗定时器框图 . . . . .	223
13.3 定时器在 PreLoad 模式下工作时序 . . . . .	224
13.4 Watchdog 工作时序 . . . . .	225
13.5 看门狗报警机制 . . . . .	226
14.1 QDEC 功能框图 . . . . .	245
17.1 EMAC 框图 . . . . .	274
18.1 USB 中断触发方式 . . . . .	293
18.2 USB 通信方式 . . . . .	295

## 表格

1.1 总线连接 . . . . .	20
1.2 地址映像 . . . . .	21
1.3 中断源 . . . . .	22
3.1 软件复位功能表 . . . . .	26
3.2 GPIO 功能表 1 . . . . .	30
3.3 GPIO 功能表 2 . . . . .	30
3.4 GPIO 功能表 3 . . . . .	32
4.1 ADC 内部信号 . . . . .	65
4.2 ADC 外部引脚 . . . . .	66
4.3 ADC 转换结果含义 . . . . .	69
5.1 内部参考电压 . . . . .	91
7.1 WayDisable 的设定 . . . . .	137
11.1 I2C 引脚 . . . . .	187
12.1 占空比参数 . . . . .	204
16.1 I2S 引脚 . . . . .	265
17.1 传输信号 . . . . .	275
18.1 寄存器配置 1 . . . . .	295
18.2 寄存器配置 2 . . . . .	296
19.1 文档版本修改信息 . . . . .	337

## 系统和存储器概述

### 1.1 简介

芯片内处理器采用 32-bit 带浮点的 RISC-V。搭配高速缓存 (详见 L1C 章节), 达到优质的运算效率。处理器外部为多层 32-bit AHB 架构, 具有低功耗、低延迟、高弹性的特性。内存部分包含高速紧耦合内存以及缓存和系统共享内存。片外存储器支持 Flash 扩充。

### 1.2 主要特征

- RISC-V 32-bit 带浮点
- 多层 32-bit AHB 总线架构
- 132KB RAM
- 192KB ROM
- 片外存储器 Flash

### 1.3 功能描述

BL702/704/706 总线连接与地址访问总结如下: 总线主机包括 CPU, 以太网, DMA, 加密引擎, 调试接口。总线从机包括内存, 外设, Zigbee/BLE。除了以太网和加密引擎只能访问内存外, 其余总线主机皆可访问所有总线从机。

表 1.1: 总线连接

从/主	CPU	以太网	DMA	加密引擎	调试接口
内存	V	V	V	V	V
外设	V	-	V	-	V
Zigbee/BLE	V	-	V	-	V

地址访问主要以 [27:24] 来区分“存储”或“外设”，可忽略 [31:28]。内存空间是连续的 X2010000~X202FFFF (128KB SRAM)，只读内存 X1000000，深度睡眠内存 X0010000。片外空间是 X3000000 (最大支持 8MB Flash)。外设空间是 X0000000~X000F000。

表 1.2: 地址映像

目标	开始地址	大小	描述
RETRAM	0x40010000	4KB	深度睡眠内存（保留 RAM）
HBN	0x4000F000	4KB	深度睡眠控制（休眠）
PDS	0x4000E000	4KB	睡眠控制（掉电睡眠）
USB	0x4000D800	1KB	USB 控制
EMAC	0x4000D000	2KB	EMAC 控制
DMA	0x4000C000	4KB	DMA 控制
QSPI	0x4000B000	4KB	闪存/pSRAM QSPI 控制
I2S	0x4000AA00	256B	I2S 控制
KYS	0x4000A900	256B	Key-Scan 控制
QDEC2	0x4000A880	64B	求积译码器控制
QDEC1	0x4000A840	64B	求积译码器控制
QDEC0	0x4000A800	64B	求积译码器控制
IRR	0x4000A600	256B	红外遥控器
TIMER	0x4000A500	256B	计时器控制
PWM	0x4000A400	256B	脉冲宽度调制控制
I2C	0x4000A300	256B	I2C 控制
SPI	0x4000A200	256B	SPI 主/从控制
UART1	0x4000A100	256B	UART 控制
UART0	0x4000A000	256B	UART 控制
L1C	0x40009000	4KB	缓存控制
eFuse	0x40007000	4KB	eFuse 存储器控制
SEC	0x40004000	4KB	安全引擎
GPIP	0x40002000	4KB	通用 DAC / ADC / ACOMP 接口控制
MIX	0x40001000	4KB	混合信号寄存器
GLB	0x40000000	4KB	全局寄存器
pSRAM	0x24000000	8MB	pSRAM 存储器
XIP	0x23000000	8MB	XIP 闪存
OCRAM	0x22020000	64KB	片上存储器
DTCM	0x22014000	48KB	数据高速缓存
ITCM	0x22010000	16KB	指令高速缓存
ROM	0x21000000	192KB	只读存储器

中断源共 64 个，列举如下，电平或边沿触发由 CPU 配置，可屏蔽使用。

表 1.3: 中断源

编号	讯号源
54~63	wireless
53	brown-out
51~52	hbn_irq
50	pds_int
47~49	wireless
44	gpio_irq
40~42	qdec_int
39	kys_int
35~38	timer_irq
34	pwm_int
32	i2c_int
30	uart1_irq
29	uart0_irq
27	spi_int
26	efuse_int
25	adc_int
23	flash_int
22	emac_int
21	usb_int
19~20	ir_remote_int
18	i2s_int
15	dma_int
9~14	sec_eng_int
8	err_int
5~6	rf_int
0~4	err_int

## 2.1 简介

芯片内部包含的复位源：硬件重置，看门狗重置，软件复位。芯片内包含多个时钟源：XTAL，DLL，RC。搭配分频等配置送至各模块。

## 2.2 复位源

复位源包含

- 硬件重置: 通过管脚进行重置
  - 管脚电源重置 (**PU\_CHIP = 0-> 1**): 类似电源上电重置
  - 电源上电重置: 芯片从断电中复电, **HBN** 逻辑将芯片系统进行重置
- 看门狗重置
  - 当看门狗报警触发重置信号时, 重置管理单元将在必要准备后, 重置芯片系统, 看门狗内部逻辑会记录看门狗重置的状态
- 软件复位: 通过软件设置寄存器进行全局或局部复位
  - 软件初始重置 (**reg\_ctrl\_pwron\_rst**): 通过软件将该位置 1, 进行芯片系统重置
  - 软件 CPU 复位 (**reg\_ctrl\_cpu\_reset**): 通过软件将该位置 1, 进行 CPU 部分系统复位
  - 软件系统复位 (**reg\_ctrl\_sys\_reset**): 通过软件将该位置 1, 保留必要的逻辑处理如电源管理单元, 进行芯片部分系统复位
  - 软件模块复位: 根据特定模块的需求, 设置软件复位

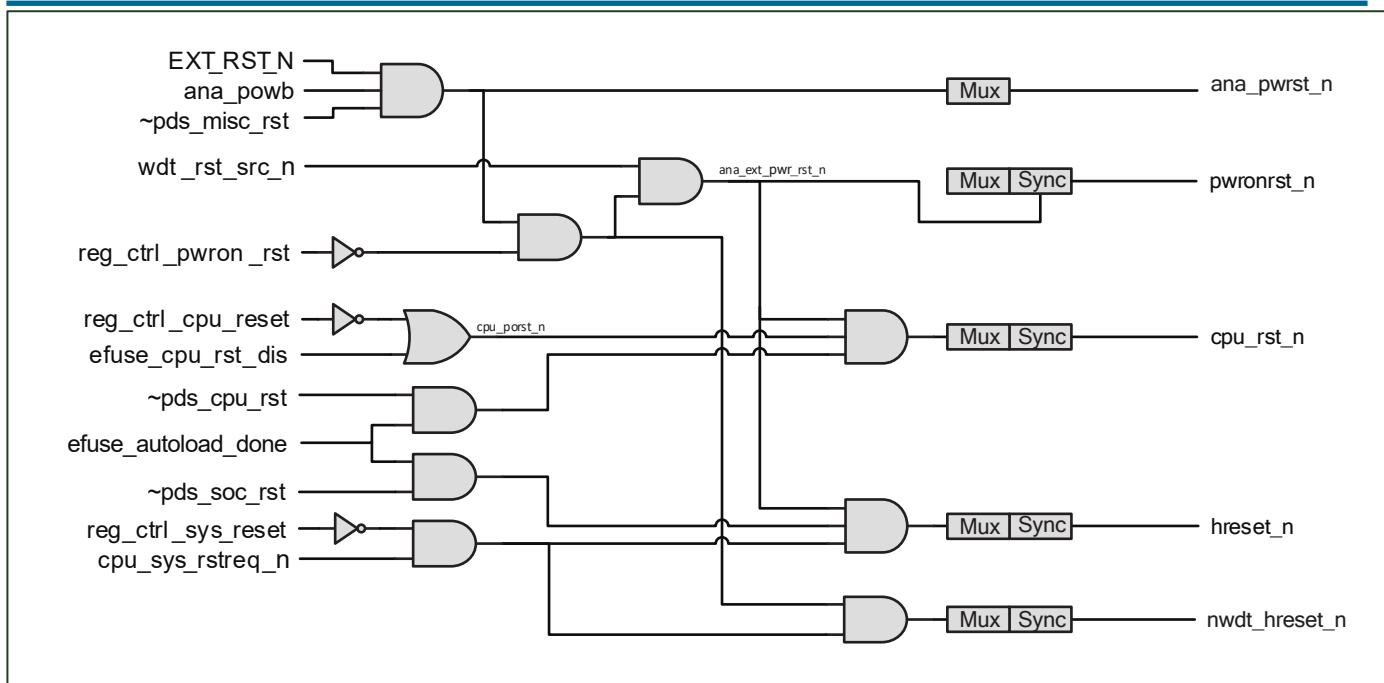


图 2.1: 复位源

## 2.3 时钟源

时钟源包含：

- XTAL：外部晶振时钟，主要支持频率 32MHz
- XTAL32K：外部晶振时钟，频率 32KHz
- RC32M：RC 振荡器时钟，频率 32MHz，提供校准
- DLL：锁相回路时钟，内部系统高速时钟，最高频率支持 144MHz

时钟控制单元将来自振荡器的时钟分配给内核和外围设备。可通过选择系统时钟源，动态分频器，时钟配置，睡眠使用 32KHz 时钟，以达到低功耗时钟管理。

外围设备时钟包括: Flash、USB2.0、Ethernet、UART、I2C、I2S、SPI、PWM、IR-remote、QDEC、KeyScan、ADC、DAC。

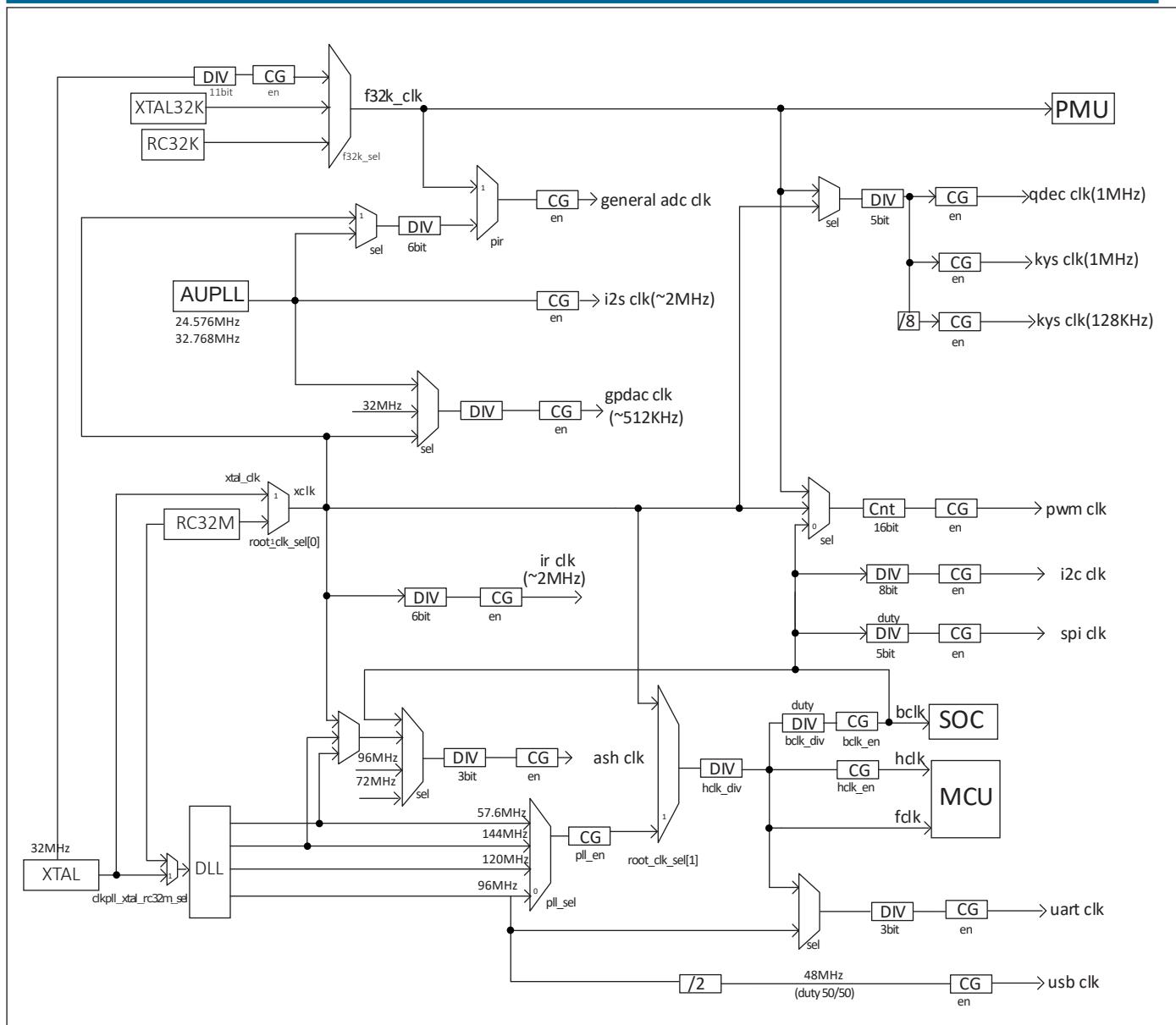


图 2.2: 时钟架构

## 3.1 简介

GLB(Global Register) 是芯片通用全局设定模块，主要包含了时钟管理、复位管理、总线管理、内存管理以及 GPIO 管理等功能。

## 3.2 功能描述

### 3.2.1 时钟管理

时钟管理功能主要用于设定处理器、总线、各个外设的时钟，通过该模块可以设定上述模块工作的时钟源，时钟分频等，同时也可以实现对上述模块时钟的门控，以达到系统低功耗的目的。详细设定可以参考复位和时钟章节。

### 3.2.2 复位管理

提供各个外设模块的单独复位功能以及芯片复位功能。芯片复位包括：

- CPU 复位：仅仅复位 CPU 模块，程序会重新运行，外设不会被复位
- 系统复位：各个外设和 CPU 会被复位，但是 AON 域的相关寄存器不会被复位
- 上电复位：整个系统包括 AON 域的相关寄存器都会被复位

应用程序可以根据需要，选择使用对应的复位方式。

表 3.1: 软件复位功能表

BL702	RST_- PIN/HBN/WDT/POWER ON	SW.Reset	CPU/PDS/SYS	PDS/CPU	PDS
CPU	✓			✓	
bus	✓		✓		
glb	✓	swrst_s1[0]			

表 3.1: 软件复位功能表 (continued)

BL702	RST_- PIN/HBN/WDT/POWER ON	SW.Reset	CPU/PDS/SYS	PDS/CPU	PDS
gpip	✓	swrst_s1[2]	✓		
I1c	✓	swrst_s1[9]	✓	✓	
dma	✓	swrst_s1[12]	✓		
emac	✓	swrst_s1[13]	✓		
usb	✓	swrst_s1a[12]	✓		✓
pds		swrst_s1[14]			
uart0	✓	swrst_s1a[0]	✓		
uart1	✓	swrst_s1a[1]	✓		
spi	✓	swrst_s1a[2]	✓		
I2C	✓	swrst_s1a[3]	✓		
pwm	✓	swrst_s1a[4]	✓		
timer	✓	swrst_s1a[5]	✓		
irr	✓	swrst_s1a[6]	✓		
qdec0	✓	swrst_s1a[8]	✓		
qdec1	✓	swrst_s1a[8]	✓		
qdec2	✓	swrst_s1a[8]	✓		
kys	✓	swrst_s1a[9]	✓		
i2s	✓	swrst_s1a[10]	✓		

### 3.2.3 总线管理

提供总线的仲裁设定以及总线出错设定，可以设定在总线出错时是否产生中断，并提供出错总线地址信息，方便用户调试程序。

### 3.2.4 内存管理

提供各个内存模块在芯片系统进入低功耗模式时的功耗管理，包含两种设定模式：

- **retention** 模式：在该模式下，内存上的数据可以保存，但是在退出低功耗模式之前，无法读写。
- **sleep** 模式：在该模式下，内存的数据会丢失，仅用于降低系统功耗。

### 3.2.5 GPIO 概述

GPIO 管理功能提供 GPIO 控制寄存器，实现软件对 GPIO 属性的配置，使用户能够方便地操作 GPIO。每个 GPIO 可以配置为 software GPIO 或其它复用功能。在每个功能下，提供设置上拉，下拉，浮空三种端口状态(配置为模拟功能时必须设置为浮空)，此外 GPIO 还提供中断功能，可以配置为上升沿触发，下降沿触发、高电平触发或者低电平触发。每个 GPIO 可以拥有两组中断配置，两组中断配置可以同时起效，例如 GPIO0 的 INT0 配置为上升沿触发、INT1 配置为下降沿触发，则最终效果为 GPIO 双边沿都会触发中断。

### 3.2.6 GPIO 主要特点

- 可以配置为 software GPIO 功能
- 可以配置为其它复用功能，搭配外设功能使用，在配置为模拟功能时，必须设置为浮空
- 可以设置输入或输出模式，并设定为上拉，下拉或者浮空
- 可以设置驱动能力，以提供更大的输出电流
- 可以设置施密特触发器功能，提供简单硬件防抖功能

### 3.2.7 GPIO 功能描述

每个 GPIO 可以通过软件配置为：

- 复用功能：I2S、SPI、I2C、UART、PWM、USB、SWGPIO..... 多达 24 种 function
- InputEnable/OutputEnable：输入、输出、高阻 (ie=0,oe=0)
- PullUp/PullDown：上拉、下拉、浮空 (pu=0,pd=0)
- drive strength：0、1、2、3 四个档位，数值越大，驱动能力越强
- smt 触发器：smt enable、smt disable，用以防止触发阈值附近的抖动

当 GPIO 复用功能配置为 SWGPIO input 时还可以为其配置中断触发方式，且每个 GPIO 可以有两个中断模式，两个中断模式均可以单独/同时起效：

- 中断模式 1：上升沿触发、下降沿触发、高电平触发、低电平触发
- 中断模式 2：上升沿触发、下降沿触发、高电平触发、低电平触发

GPIO 模块基本框图如下图所示。

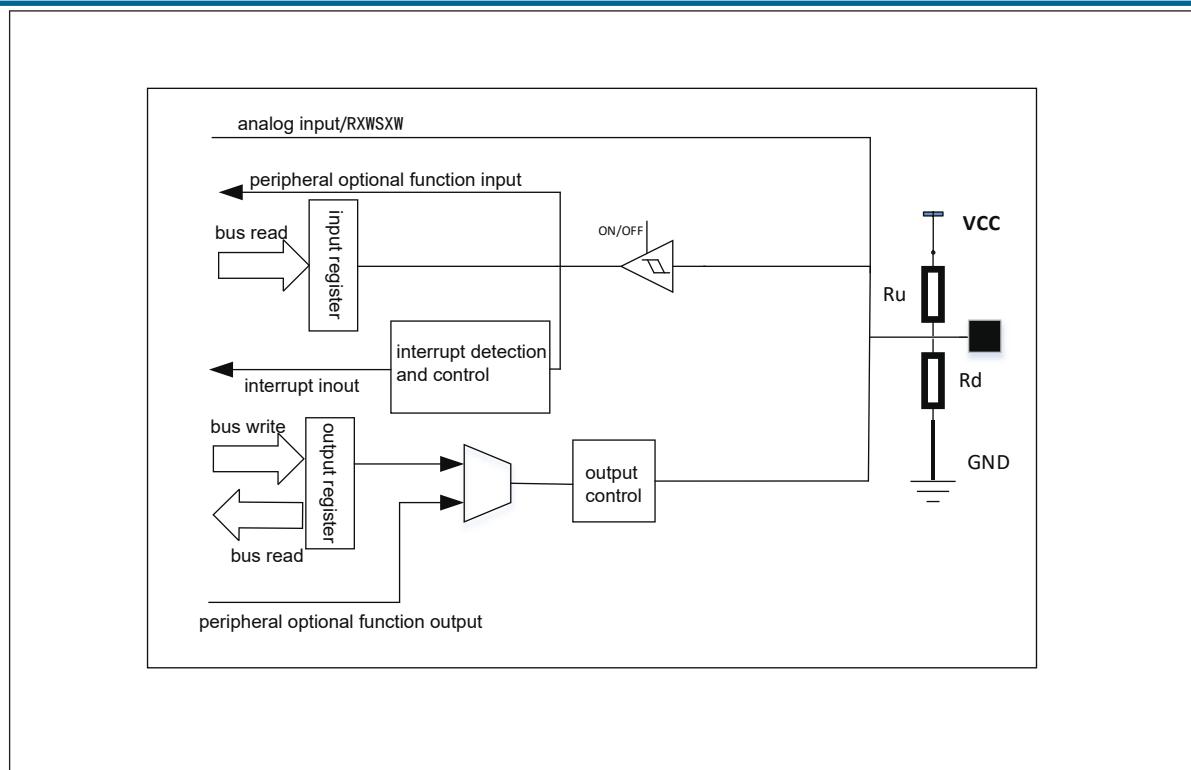


图 3.1: GPIO 基本框图

### 3.2.8 GPIO 功能设定

GPIO 的功能通过 `GPIO_CFGCTL` 寄存器组设定，主要设定项包括：

- `func_sel`: 选择 GPIO 功能
- `pu`: 选择是否上拉
- `pd`: 选择是否下拉
- `drv`: 设定驱动能力
- `smt`: 选择是否使能施密特触发器
- `ie`: 设定输入使能
- `oe`: 设定输出使能

GPIO 可以设定的功能包括：

- `Flash/QSPI`: 设定 GPIO 为 QSPI 功能，可以连接 Flash，作为程序存储/运行介质
- `SPI`: 设定 GPIO 为 SPI 功能
- `I2C`: 设定 GPIO 为 I2C 功能
- `UART`: 设定 GPIO 为 UART 功能

- PWM: 设定 GPIO 为 PWM 功能
- ANA: 设定 GPIO 为 Analog 功能
- SWGPIO: 设定 GPIO 为通用 IO 功能
- JTAG: 设定 GPIO 为 JTAG 功能
- 其它复用功能

为了最大限度的满足客户需求，每个 GPIO 基本上都可以选择上述可选功能，当选择某个可选功能时，GPIO 与对应的功能信号如下表所示：

表 3.2: GPIO 功能表 1

GPIO	CLK_OUT	Flash_PSRAM	I2S	SPI0	I2C	UART	PWM
GPIO0	clk_out[0]		I2S0_BCLK	SPI_0_MOSI	I2C0_SCL	UART_SIG0	PWM[0]
GPIO1	clk_out[1]		I2S0_FS	SPI_0_MISO	I2C0_SDA	UART_SIG1	PWM[1]
GPIO2	clk_out[0]		I2S0_DIO/I2S0_DO	SPI_0_SS	I2C0_SCL	UART_SIG2	PWM[2]
GPIO3	clk_out[1]		I2S0_RCLK_O/I2S0_DI	SPI_0_SCLK	I2C0_SDA	UART_SIG3	PWM[3]
GPIO4	clk_out[0]		I2S0_BCLK	SPI_0_MOSI	I2C0_SCL	UART_SIG4	PWM[4]
GPIO5	clk_out[1]		I2S0_FS	SPI_0_MISO	I2C0_SDA	UART_SIG5	PWM[0]
GPIO6	clk_out[0]		I2S0_DIO/I2S0_DO	SPI_0_SS	I2C0_SCL	UART_SIG6	PWM[1]
GPIO7	clk_out[1]		I2S0_RCLK_O/I2S0_DI	SPI_0_SCLK	I2C0_SDA	UART_SIG7	PWM[2]
GPIO8	clk_out[0]		I2S0_BCLK	SPI_0_MOSI	I2C0_SCL	UART_SIG0	PWM[3]
GPIO9	clk_out[1]		I2S0_FS	SPI_0_MISO	I2C0_SDA	UART_SIG1	PWM[4]
GPIO10	clk_out[0]		I2S0_DIO/I2S0_DO	SPI_0_SS	I2C0_SCL	UART_SIG2	PWM[0]
GPIO11	clk_out[1]		I2S0_RCLK_O/I2S0_DI	SPI_0_SCLK	I2C0_SDA	UART_SIG3	PWM[1]
GPIO12	clk_out[0]		I2S0_BCLK	SPI_0_MOSI	I2C0_SCL	UART_SIG4	PWM[2]
GPIO13	clk_out[1]		I2S0_FS	SPI_0_MISO	I2C0_SDA	UART_SIG5	PWM[3]
GPIO14	clk_out[0]		I2S0_DIO/I2S0_DO	SPI_0_SS	I2C0_SCL	UART_SIG6	PWM[4]
GPIO15	clk_out[1]		I2S0_RCLK_O/I2S0_DI	SPI_0_SCLK	I2C0_SDA	UART_SIG7	PWM[0]
GPIO16	clk_out[0]		I2S0_BCLK	SPI_0_MOSI	I2C0_SCL	UART_SIG0	PWM[1]
GPIO17	clk_out[1]	SF_IO_0/SF2_CS2	I2S0_FS	SPI_0_MISO	I2C0_SDA	UART_SIG1	PWM[2]
GPIO18	clk_out[0]	SF_IO_1	I2S0_DIO/I2S0_DO	SPI_0_SS	I2C0_SCL	UART_SIG2	PWM[3]
GPIO19	clk_out[1]	SF_CS	I2S0_RCLK_O/I2S0_DI	SPI_0_SCLK	I2C0_SDA	UART_SIG3	PWM[4]
GPIO20	clk_out[0]	SF_IO_3	I2S0_BCLK	SPI_0_MOSI	I2C0_SCL	UART_SIG4	PWM[0]
GPIO21	clk_out[1]	SF_CLK	I2S0_FS	SPI_0_MISO	I2C0_SDA	UART_SIG5	PWM[1]
GPIO22	clk_out[0]	SF_IO_2	I2S0_DIO/I2S0_DO	SPI_0_SS	I2C0_SCL	UART_SIG6	PWM[2]
GPIO23	clk_out[1]	SF2_IO_2	I2S0_RCLK_O/I2S0_DI	SPI_0_SCLK	I2C0_SDA	UART_SIG7	PWM[3]
GPIO24	clk_out[0]	SF2_IO_1	I2S0_BCLK	SPI_0_MOSI	I2C0_SCL	UART_SIG0	PWM[4]
GPIO25	clk_out[1]	SF2_CS	I2S0_FS	SPI_0_MISO	I2C0_SDA	UART_SIG1	PWM[0]
GPIO26	clk_out[0]	SF2_IO_3	I2S0_DIO/I2S0_DO	SPI_0_SS	I2C0_SCL	UART_SIG2	PWM[1]
GPIO27	clk_out[1]	SF2_CLK	I2S0_RCLK_O/I2S0_DI	SPI_0_SCLK	I2C0_SDA	UART_SIG3	PWM[2]
GPIO28	clk_out[0]	SF2_IO_0	I2S0_BCLK	SPI_0_MOSI	I2C0_SCL	UART_SIG4	PWM[3]
GPIO29	clk_out[1]		I2S0_FS	SPI_0_MISO	I2C0_SDA	UART_SIG5	PWM[4]
GPIO30	clk_out[0]		I2S0_DIO/I2S0_DO	SPI_0_SS	I2C0_SCL	UART_SIG6	PWM[0]
GPIO31	clk_out[1]		I2S0_RCLK_O/I2S0_DI	SPI_0_SCLK	I2C0_SDA	UART_SIG7	PWM[1]

表 3.3: GPIO 功能表 2

GPIO	Analog	SWGPIO	JTAG	Ether_Mac	QDEC
GPIO0		REG_GPIO[0]	E21_TMS/E21_TCK	MII_REF_CLK	qdec0_a

**表 3.3: GPIO 功能表 2(continued)**

GPIO	Analog	SWGPI	JTAG	Ether_Mac	QDEC
GPIO1		REG_GPIO[1]	E21_TDI/E21_TDO	MII_TXD[0]	qdec0_b
GPIO2		REG_GPIO[2]	E21_TCK/E21_TMS	MII_TXD[1]	qdec0_led
GPIO3		REG_GPIO[3]	E21_TDO/E21_TDI		qdec1_a
GPIO4		REG_GPIO[4]	E21_TMS/E21_TCK		qdec1_b
GPIO5		REG_GPIO[5]	E21_TDI/E21_TDO		qdec1_led
GPIO6		REG_GPIO[6]	E21_TCK/E21_TMS		qdec2_a
GPIO7	USB_DP/gpip_ch[6]/gpdac_vref_ext	REG_GPIO[7]	E21_TDO/E21_TDI	MII_RXD[0]	qdec2_b
GPIO8	USB_DM/gpip_ch[0]	REG_GPIO[8]	E21_TMS/E21_TCK	MII_RXD[1]	qdec2_led
GPIO9	pmip_dc_tp/clkpll_dc_tp/gpip_ch[7]	REG_GPIO[9]	E21_TDI/E21_TDO		qdec0_a
GPIO10	MICBIAS	REG_GPIO[10]	E21_TCK/E21_TMS		qdec0_b
GPIO11	gpip_ch[3]	REG_GPIO[11]	E21_TDO/E21_TDI		qdec0_led
GPIO12	gpip_ch[4]	REG_GPIO[12]	E21_TMS/E21_TCK		qdec1_a
GPIO13		REG_GPIO[13]	E21_TDI/E21_TDO		qdec1_b
GPIO14	gpip_ch[5]/atest_out_0	REG_GPIO[14]	E21_TCK/E21_TMS		qdec1_led
GPIO15	gpip_ch[1]/atest_out_1	REG_GPIO[15]	E21_TDO/E21_TDI		qdec2_a
GPIO16		REG_GPIO[16]	E21_TMS/E21_TCK		qdec2_b
GPIO17	gpip_ch[2]/psw_irrcv	REG_GPIO[17]	E21_TDI/E21_TDO		qdec2_led
GPIO18	gpip_ch[8]	REG_GPIO[18]	E21_TCK/E21_TMS	RMII_MDC	qdec0_a
GPIO19	gpip_ch[9]	REG_GPIO[19]	E21_TDO/E21_TDI	RMII_MDIO	qdec0_b
GPIO20	gpip_ch[10]	REG_GPIO[20]	E21_TMS/E21_TCK	RMII_RXERR	qdec0_led
GPIO21	gpip_ch[11]	REG_GPIO[21]	E21_TDI/E21_TDO	RMII_TX_EN	qdec1_a
GPIO22	leddrv[0]	REG_GPIO[22]	E21_TCK/E21_TMS	RMII_RX_DV	qdec1_b
GPIO23	leddrv[1]/flash_pull_out[0]	REG_GPIO[23]	E21_TDO/E21_TDI		qdec1_led
GPIO24	flash_pull_out[1]	REG_GPIO[24]	E21_TMS/E21_TCK	RMII_MDC	qdec2_a
GPIO25	flash_pull_out[2]	REG_GPIO[25]	E21_TDI/E21_TDO	RMII_MDIO	qdec2_b
GPIO26	flash_pull_out[3]	REG_GPIO[26]	E21_TCK/E21_TMS	RMII_RXERR	qdec2_led
GPIO27	flash_pull_out[4]	REG_GPIO[27]	E21_TDO/E21_TDI	RMII_TX_EN	qdec0_a
GPIO28	flash_pull_out[5]	REG_GPIO[28]	E21_TMS/E21_TCK	RMII_RX_DV	qdec0_b
GPIO29		REG_GPIO[29]	E21_TDI/E21_TDO		qdec0_led
GPIO30		REG_GPIO[30]	E21_TCK/E21_TMS		qdec1_a
GPIO31		REG_GPIO[31]	E21_TDO/E21_TDI		qdec1_b

表 3.4: GPIO 功能表 3

GPIO	SWGPI
GPIO17	pad_irrx_i,irrxgpsi=1
GPIO18	pad_irrx_i,irrxgpsi=2
GPIO19	pad_irrx_i,irrxgpsi=3
GPIO20	pad_irrx_i,irrxgpsi=4
GPIO21	pad_irrx_i,irrxgpsi=5
GPIO22	pad_irrx_i,irrxgpsi=6
GPIO23	pad_irrx_i,irrxgpsi=7
GPIO24	pad_irrx_i,irrxgpsi=8
GPIO25	pad_irrx_i,irrxgpsi=9
GPIO26	pad_irrx_i,irrxgpsi=10
GPIO27	pad_irrx_i,irrxgpsi=11
GPIO28	pad_irrx_i,irrxgpsi=12
GPIO29	pad_irrx_i,irrxgpsi=13
GPIO30	pad_irrx_i,irrxgpsi=14
GPIO31	pad_irrx_i,irrxgpsi=15

当使用 IR 功能时，需要将 GPIO 设置为 SWGPIO，并设置 irrxgpsi 寄存器（GPIO17-GPIO31 可以作为 IR 引脚）

在上述表格中，当选择 UART 功能时，只是选择了 UART 的一个信号，并没有指定该引脚的具体功能（比如是 UART TX 还是 UART RX），还需要通过 UART\_SIGX\_SEL(X=0-7) 进一步选择具体的 UART 信号及对应的功能。每一个 UART\_SIGX\_SEL 可以选择的信号包括：

- 0 : UART0\_RTS
- 1 : UART0\_CTS
- 2 : UART0\_TXD
- 3 : UART0\_RXD
- 4 : UART1\_RTS
- 5 : UART1\_CTS
- 6 : UART1\_TXD
- 7 : UART1\_RXD

以 GPIO0 为例，当 fun\_sel 选择 UART 的时候，GPIO0 选择的是 UART\_SIG0，在默认情况下 UART\_SIG0\_SEL 的值是 0，也就是 UART0\_RTS，即 GPIO 是 UART0\_RTS 功能。如果应用程序想把 GPIO 作为 UART1\_TXD，那只要

把 `UART_SIG0_SEL` 设置为 6, 那么 GPIO0 的功能就是 `UART1_TXD`。

### 3.2.9 GPIO 输出设置

通过设定 `func_sel` 为 SWGPIO, GPIO 可以作为普通 GPIO 的输入/输出, 将 `IE` 设置为 0, `OE` 设置为 1, 就可以将 GPIO 配置为输出功能, 输出的数值通过 `GPIO_O` 寄存器组设定。当 `GPIO_O` 对应 Bit 设置为 0 时, GPIO 输出低电平, 当 `GPIO_O` 对应 Bit 设置为 1 时, GPIO 输出高电平。可以通过 DRV 控制位设置输出能力。

### 3.2.10 GPIO 输入设置

通过设定 `func_sel` 为 SWGPIO, 将 `IE` 设置为 1, `OE` 设置为 0, 就可以将 GPIO 配置为输入功能, 可以通过 SMT 控制位设置是否使能施密特触发器, 通过 `PD`, `PU` 控制位设置上拉下拉属性。外部输入的数值, 可以通过读取 `GPIO_I` 寄存器对应的 Bit 获取到。

### 3.2.11 GPIO 可选功能设置

通过设定 `func_sel` 为对应的外设功能, 可以实现 GPIO 与外设的连接, 实现外设的输入输出, 从 GPIO 的基础功能框图可以看出, 当选择可选功能时, 需要将 `IE` 设置为 1, `OE` 设置为 0, 也就是断开普通 GPIO 的输出控制功能。这样, 对于固定输入功能的外设, 外设的 `OE` 信号始终为 0, 从而实现输入功能; 对于固定输出的外设, 其 `OE` 信号始终为 1, 从而实现输出是被外设控制, 而此时的输入信号就是输出信号, 但是不会被正在输出的外设采集; 当外设既需要输入又要输出时, 通过控制外设 `OE` 信号就可以实现输入输出。即: 对于除了 SWGPIO 以外的 `function`, 作为输出方向的功能时, `IE` 和 `OE` 的配置值不影响功能, 但是作为输入方向时, `IE` 必须置一而 `OE` 的配置不影响功能; 当作为 SWGPIO 时, `IE` 和 `OE` 都需要正确配置。

### 3.2.12 GPIO 中断设置

要使用 GPIO 的中断功能, 需要先将 GPIO 设置为输入模式, 中断触发模式通过 `GPIO_INT_MODE_SET` 寄存器组进行设定。可以设定的中断模式包括:

- 下降沿触发中断
- 上升沿触发中断
- 低电平触发中断
- 高电平触发中断

每个 GPIO 都可以设定为中断功能, 是否使能某个 GPIO 中断可以通过 `GPIO_INT_MASK` 寄存器进行设定, 中断产生时, 在中断函数中可以通过 `GPIO_INT_STAT` 寄存器获取到产生中断的 GPIO 引脚号, 同时可以通过 `GPIO_INT_CLR` 清除掉对应的中断信号。

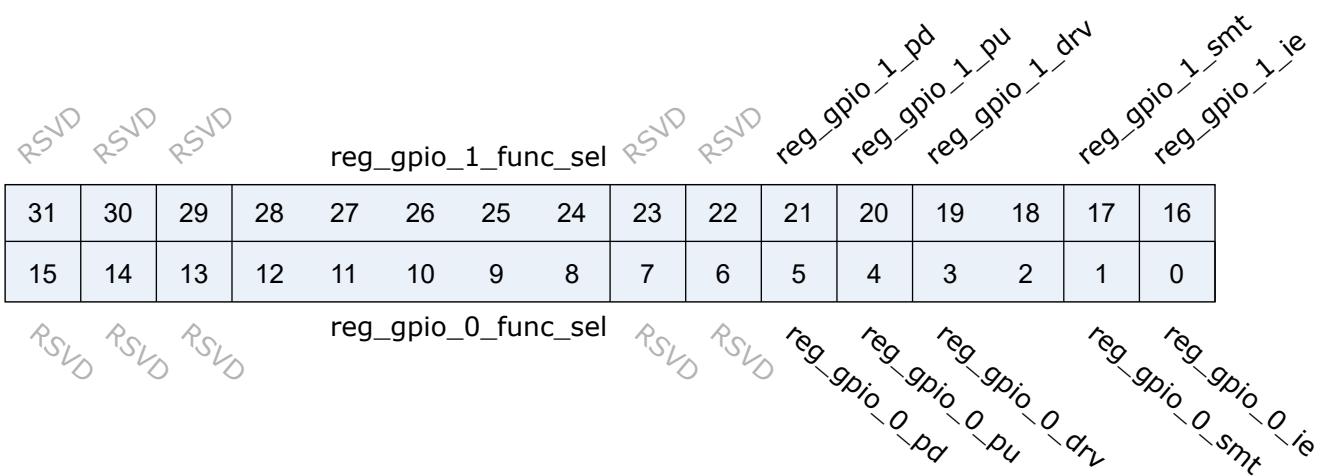
### 3.3 寄存器描述

名称	描述
GPIO_CFGCTL0	
GPIO_CFGCTL1	
GPIO_CFGCTL2	
GPIO_CFGCTL3	
GPIO_CFGCTL4	
GPIO_CFGCTL5	
GPIO_CFGCTL6	
GPIO_CFGCTL7	
GPIO_CFGCTL8	
GPIO_CFGCTL9	
GPIO_CFGCTL10	
GPIO_CFGCTL11	
GPIO_CFGCTL12	
GPIO_CFGCTL13	
GPIO_CFGCTL14	
GPIO_CFGCTL15	
GPIO_CFGCTL30	
GPIO_CFGCTL32	
GPIO_CFGCTL34	
GPIO_CFGCTL35	
GPIO_INT_MASK1	
GPIO_INT_STAT1	
GPIO_INT_CLR1	
GPIO_INT_MODE_SET1	
GPIO_INT_MODE_SET2	
GPIO_INT_MODE_SET3	
GPIO_INT_MODE_SET4	

名称	描述
GPIO_INT2_MASK1	
GPIO_INT2_STAT1	
GPIO_INT2_CLR1	
GPIO_INT2_MODE_SET1	
GPIO_INT2_MODE_SET2	
GPIO_INT2_MODE_SET3	
GPIO_INT2_MODE_SET4	

### 3.3.1 GPIO\_CFGCTL0

地址: 0x40000100

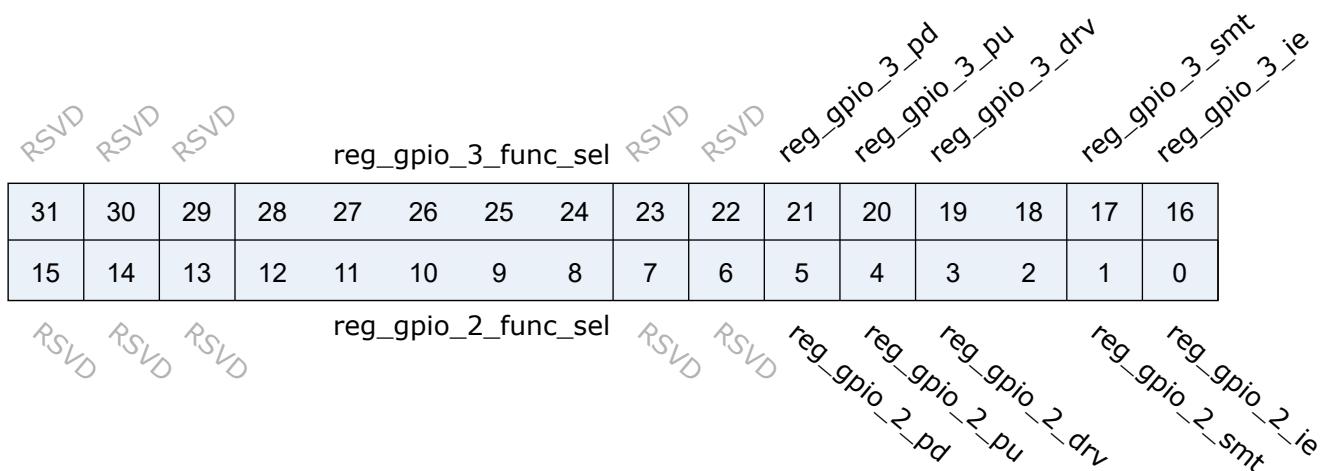


位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_1_func_sel	r/w	5'hE	GPIO Function Select (Default : JTAG)
23:22	RSVD			
21	reg_gpio_1_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_1_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_1_drv	r/w	0	GPIO Driving Control
17	reg_gpio_1_smt	r/w	1	GPIO SMT Control
16	reg_gpio_1_ie	r/w	1	GPIO Input Enable

位	名称	权限	复位值	描述
15:13	RSVD			
12:8	reg_gpio_0_func_sel	r/w	5'hE	GPIO Function Select (Default : JTAG)
7:6	RSVD			
5	reg_gpio_0_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_0_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_0_drv	r/w	0	GPIO Driving Control
1	reg_gpio_0_smt	r/w	1	GPIO SMT Control
0	reg_gpio_0_ie	r/w	1	GPIO Input Enable

### 3.3.2 GPIO\_CFGCTL1

地址: 0x40000104

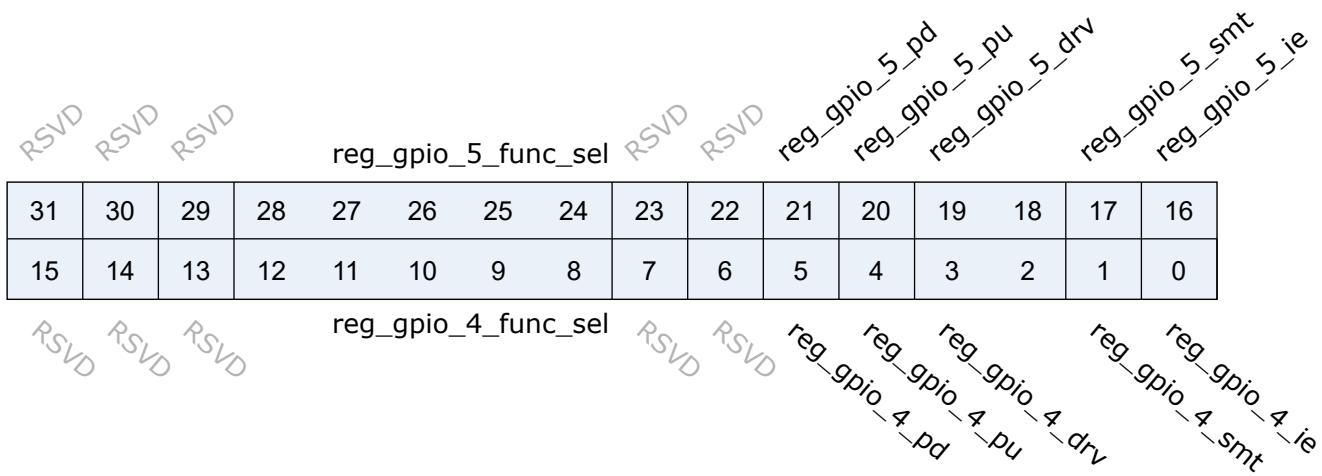


位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_3_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	reg_gpio_3_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_3_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_3_drv	r/w	0	GPIO Driving Control
17	reg_gpio_3_smt	r/w	1	GPIO SMT Control
16	reg_gpio_3_ie	r/w	1	GPIO Input Enable
15:13	RSVD			

位	名称	权限	复位值	描述
12:8	reg_gpio_2_func_sel	r/w	5'hE	GPIO Function Select (Default : JTAG)
7:6	RSVD			
5	reg_gpio_2_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_2_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_2_drv	r/w	0	GPIO Driving Control
1	reg_gpio_2_smt	r/w	1	GPIO SMT Control
0	reg_gpio_2_ie	r/w	1	GPIO Input Enable

### 3.3.3 GPIO\_CFGCTL2

地址: 0x40000108

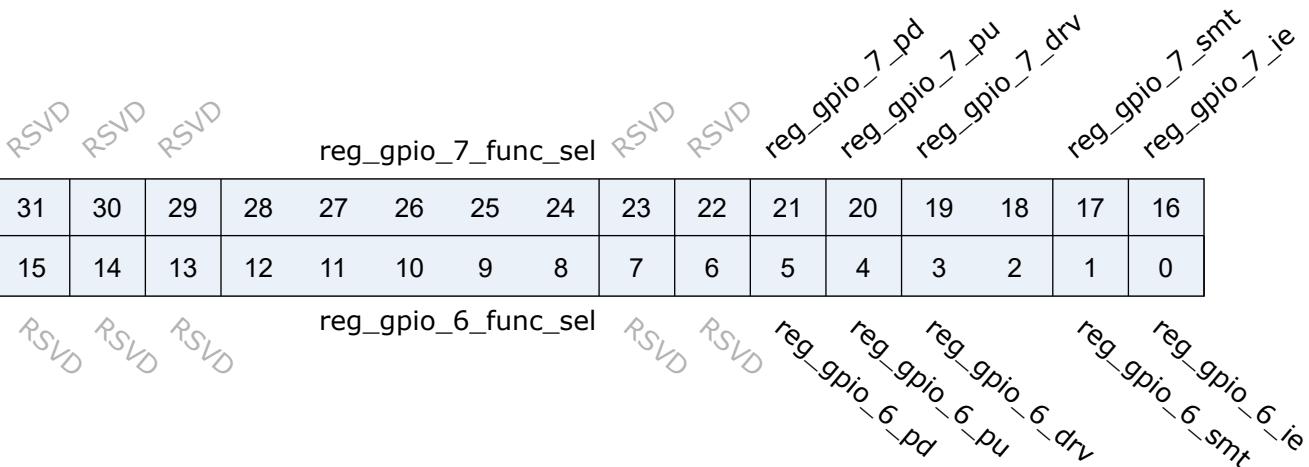


位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_5_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	reg_gpio_5_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_5_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_5_drv	r/w	0	GPIO Driving Control
17	reg_gpio_5_smt	r/w	1	GPIO SMT Control
16	reg_gpio_5_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_4_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO)

位	名称	权限	复位值	描述
7:6	RSVD			
5	reg_gpio_4_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_4_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_4_drv	r/w	0	GPIO Driving Control
1	reg_gpio_4_smt	r/w	1	GPIO SMT Control
0	reg_gpio_4_ie	r/w	1	GPIO Input Enable

### 3.3.4 GPIO\_CFGCTL3

地址: 0x4000010c

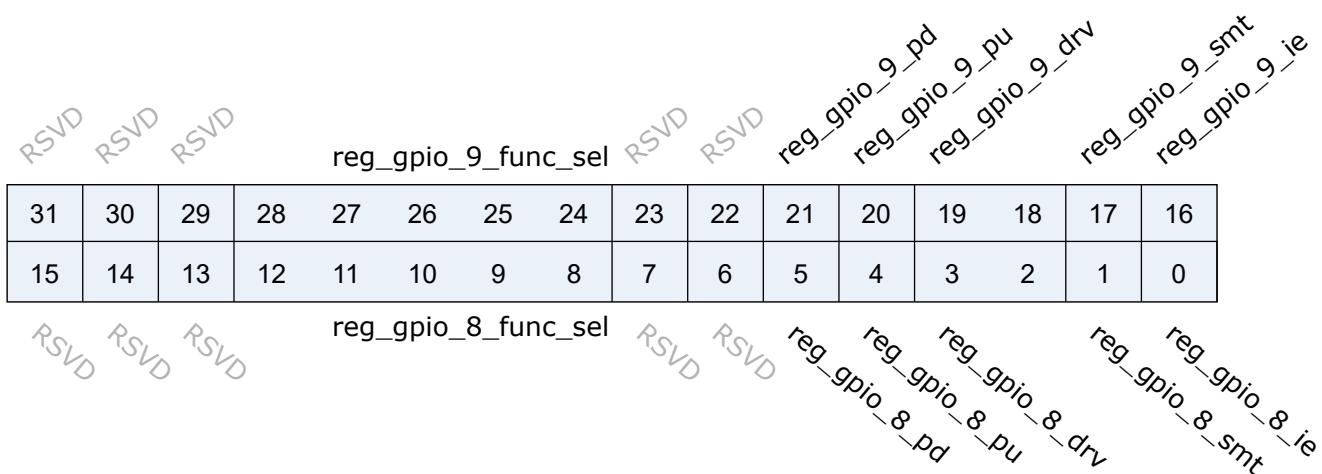


位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_7_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO)
23:22	RSVD			
21	reg_gpio_7_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_7_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_7_drv	r/w	0	GPIO Driving Control
17	reg_gpio_7_smt	r/w	1	GPIO SMT Control
16	reg_gpio_7_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_6_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			

位	名称	权限	复位值	描述
5	reg_gpio_6_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_6_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_6_drv	r/w	0	GPIO Driving Control
1	reg_gpio_6_smt	r/w	1	GPIO SMT Control
0	reg_gpio_6_ie	r/w	1	GPIO Input Enable

### 3.3.5 GPIO\_CFGCTL4

地址: 0x40000110

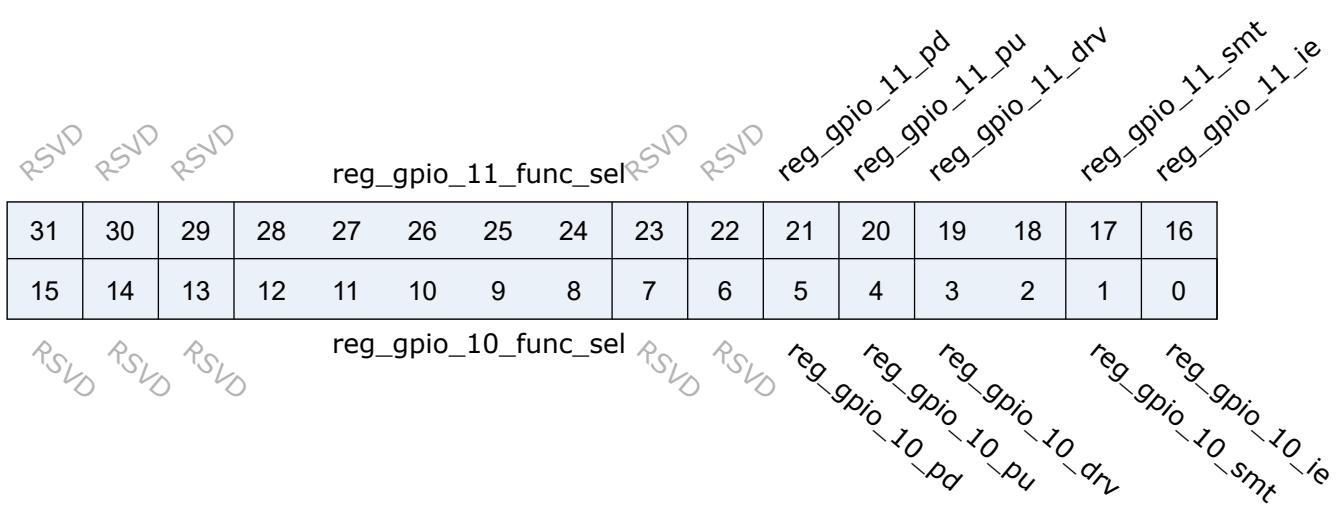


位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_9_func_sel	r/w	5'hE	GPIO Function Select (Default : JTAG )
23:22	RSVD			
21	reg_gpio_9_pd	r/w	0	GPIO Pull Down Control (Use this bit if reg_en_hw_pu_pd := 0 (0x4000F014[16]))
20	reg_gpio_9_pu	r/w	0	GPIO Pull Up Control (Use this bit if reg_en_hw_pu_pd := 0 (0x4000F014[16]))
19:18	reg_gpio_9_drv	r/w	0	GPIO Driving Control
17	reg_gpio_9_smt	r/w	1	GPIO SMT Control(Useless, IE depend on reg_aon_pad_ie_smt[0] : 0x4000F014[8])
16	reg_gpio_9_ie	r/w	0	GPIO Input Enable (Useless, IE depend on reg_aon_pad_ie_smt[0] : 0x4000F014[8])
15:13	RSVD			

位	名称	权限	复位值	描述
12:8	reg_gpio_8_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_8_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_8_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_8_drv	r/w	0	GPIO Driving Control
1	reg_gpio_8_smt	r/w	1	GPIO SMT Control
0	reg_gpio_8_ie	r/w	1	GPIO Input Enable

### 3.3.6 GPIO\_CFGCTL5

地址: 0x40000114

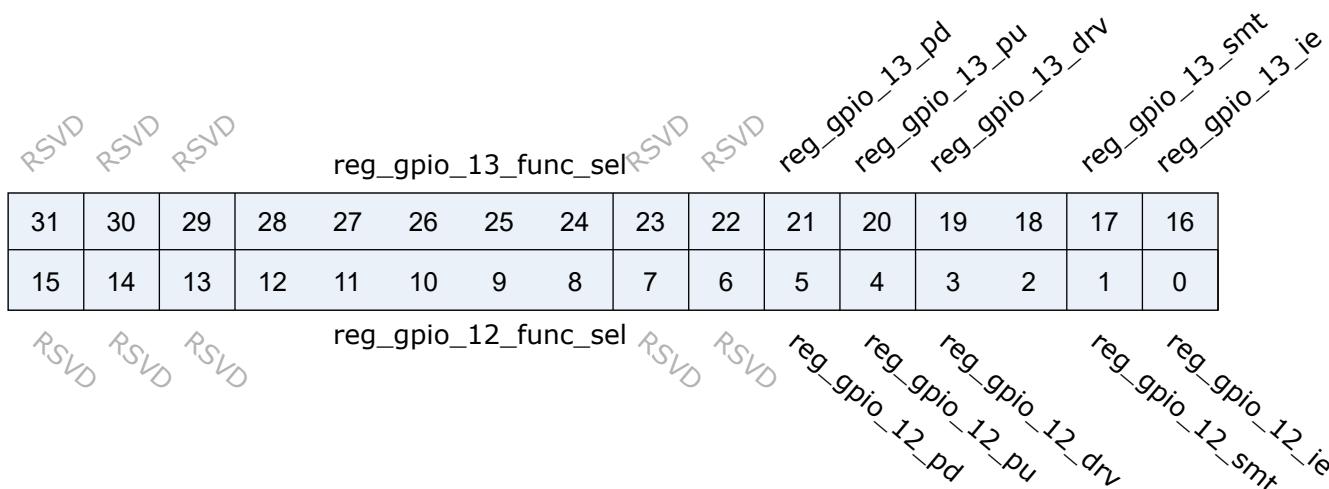


位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_11_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
23:22	RSVD			
21	reg_gpio_11_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_11_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_11_drv	r/w	0	GPIO Driving Control
17	reg_gpio_11_smt	r/w	1	GPIO SMT Control(Useless, IE depend on reg_aon_pad_ie_smt[2] : 0x4000F014[10])
16	reg_gpio_11_ie	r/w	1	GPIO Input Enable (Useless, IE depend on reg_aon_pad_ie_smt[2] : 0x4000F014[10])

位	名称	权限	复位值	描述
15:13	RSVD			
12:8	reg_gpio_10_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_10_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_10_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_10_drv	r/w	0	GPIO Driving Control
1	reg_gpio_10_smt	r/w	1	GPIO SMT Control(Useless, IE depend on reg_aon_pad_ie_smt[1] : 0x4000F014[9])
0	reg_gpio_10_ie	r/w	1	GPIO Input Enable (Useless, IE depend on reg_aon_pad_ie_smt[1] : 0x4000F014[9])

### 3.3.7 GPIO\_CFGCTL6

地址: 0x400000118

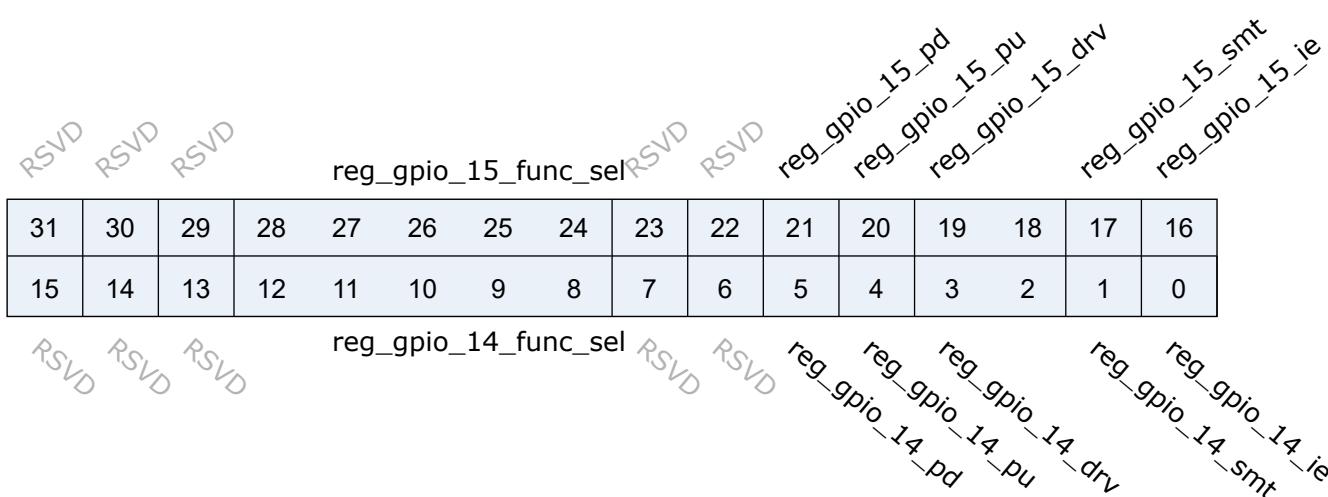


位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_13_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
23:22	RSVD			
21	reg_gpio_13_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_13_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_13_drv	r/w	0	GPIO Driving Control

位	名称	权限	复位值	描述
17	reg_gpio_13_smt	r/w	1	GPIO SMT Control(Useless, IE depend on reg_aon_pad_ie_smt[4] : 0x4000F014[12])
16	reg_gpio_13_ie	r/w	1	GPIO Input Enable (Useless, IE depend on reg_aon_pad_ie_smt[4] : 0x4000F014[12])
15:13	RSVD			
12:8	reg_gpio_12_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_12_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_12_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_12_drv	r/w	0	GPIO Driving Control
1	reg_gpio_12_smt	r/w	1	GPIO SMT Control(Useless, IE depend on reg_aon_pad_ie_smt[3] : 0x4000F014[11])
0	reg_gpio_12_ie	r/w	1	GPIO Input Enable (Useless, IE depend on reg_aon_pad_ie_smt[3] : 0x4000F014[11])

### 3.3.8 GPIO\_CFGCTL7

地址: 0x4000011c

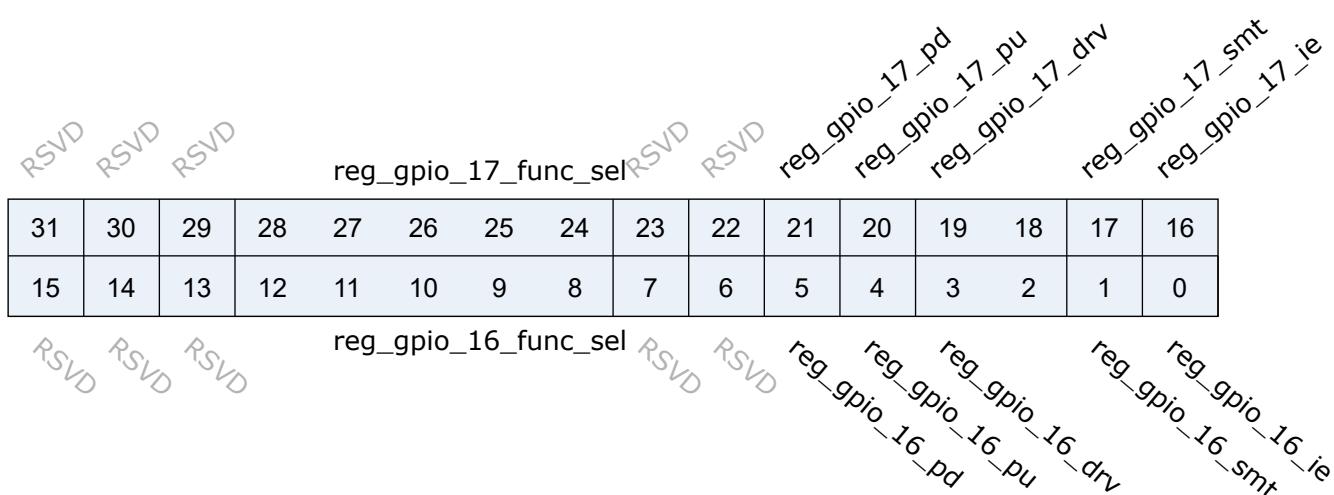


位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_15_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
23:22	RSVD			

位	名称	权限	复位值	描述
21	reg_gpio_15_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_15_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_15_drv	r/w	0	GPIO Driving Control
17	reg_gpio_15_smt	r/w	1	GPIO SMT Control
16	reg_gpio_15_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_14_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_14_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_14_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_14_drv	r/w	0	GPIO Driving Control
1	reg_gpio_14_smt	r/w	1	GPIO SMT Control
0	reg_gpio_14_ie	r/w	1	GPIO Input Enable

### 3.3.9 GPIO\_CFGCTL8

地址: 0x40000120

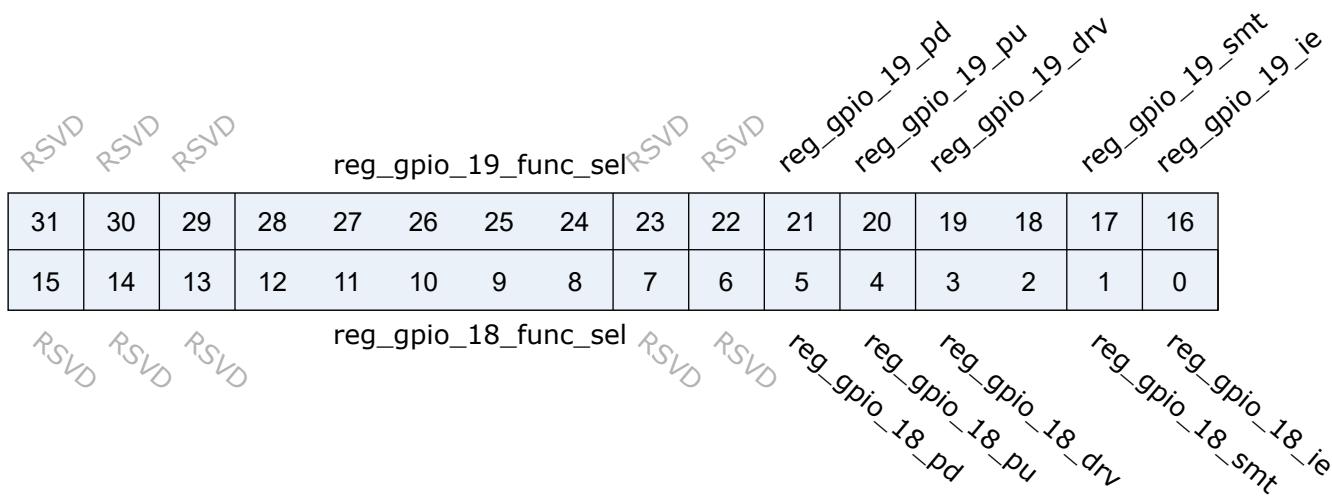


位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_17_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
23:22	RSVD			

位	名称	权限	复位值	描述
21	reg_gpio_17_pd	r/w	0	GPIO Pull Down Control (Use this bit if reg_en_hw_pu_pd := 0 (0x4000F014[16]))
20	reg_gpio_17_pu	r/w	0	GPIO Pull Up Control (Use this bit if reg_en_hw_pu_pd := 0 (0x4000F014[16]))
19:18	reg_gpio_17_drv	r/w	0	GPIO Driving Control
17	reg_gpio_17_smt	r/w	1	GPIO SMT Control
16	reg_gpio_17_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_16_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_16_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_16_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_16_drv	r/w	0	GPIO Driving Control
1	reg_gpio_16_smt	r/w	1	GPIO SMT Control
0	reg_gpio_16_ie	r/w	1	GPIO Input Enable

### 3.3.10 GPIO\_CFGCTL9

地址: 0x40000124

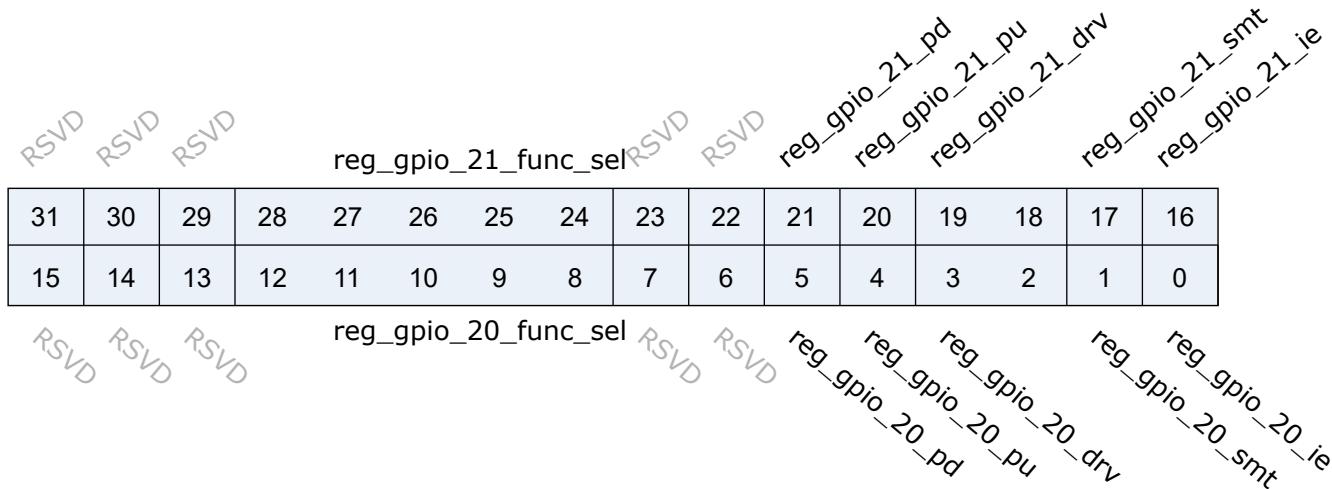


位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_19_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )

位	名称	权限	复位值	描述
23:22	RSVD			
21	reg_gpio_19_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_19_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_19_drv	r/w	0	GPIO Driving Control
17	reg_gpio_19_smt	r/w	1	GPIO SMT Control
16	reg_gpio_19_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_18_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_18_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_18_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_18_drv	r/w	0	GPIO Driving Control
1	reg_gpio_18_smt	r/w	1	GPIO SMT Control
0	reg_gpio_18_ie	r/w	1	GPIO Input Enable

### 3.3.11 GPIO\_CFGCTL10

地址: 0x40000128



位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_21_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )

位	名称	权限	复位值	描述
23:22	RSVD			
21	reg_gpio_21_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_21_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_21_drv	r/w	0	GPIO Driving Control
17	reg_gpio_21_smt	r/w	1	GPIO SMT Control
16	reg_gpio_21_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_20_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_20_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_20_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_20_drv	r/w	0	GPIO Driving Control
1	reg_gpio_20_smt	r/w	1	GPIO SMT Control
0	reg_gpio_20_ie	r/w	1	GPIO Input Enable

### 3.3.12 GPIO\_CFGCTL11

地址: 0x4000012c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	reg_gpio_23_func_sel	RSVD	RSVD	reg_gpio_23_pd	reg_gpio_23_pu	reg_gpio_23_drv	reg_gpio_23_smt	reg_gpio_23_ie					
RSVD	RSVD	RSVD	reg_gpio_22_func_sel	RSVD	RSVD	reg_gpio_22_pd	reg_gpio_22_pu	reg_gpio_22_drv	reg_gpio_22_smt	reg_gpio_22_ie					

位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_23_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )

位	名称	权限	复位值	描述
23:22	RSVD			
21	reg_gpio_23_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_23_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_23_drv	r/w	0	GPIO Driving Control
17	reg_gpio_23_smt	r/w	1	GPIO SMT Control
16	reg_gpio_23_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_22_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_22_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_22_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_22_drv	r/w	0	GPIO Driving Control
1	reg_gpio_22_smt	r/w	1	GPIO SMT Control
0	reg_gpio_22_ie	r/w	1	GPIO Input Enable

### 3.3.13 GPIO\_CFGCTL12

地址: 0x40000130

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	reg_gpio_25_func_sel	RSVD	RSVD	reg_gpio_25_pd	reg_gpio_25_pu	reg_gpio_25_drv	reg_gpio_25_smt	reg_gpio_25_ie					
RSVD	RSVD	RSVD	reg_gpio_24_func_sel	RSVD	RSVD	reg_gpio_24_pd	reg_gpio_24_pu	reg_gpio_24_drv	reg_gpio_24_smt	reg_gpio_24_ie					

位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_25_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )

位	名称	权限	复位值	描述
23:22	RSVD			
21	reg_gpio_25_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_25_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_25_drv	r/w	0	GPIO Driving Control
17	reg_gpio_25_smt	r/w	1	GPIO SMT Control
16	reg_gpio_25_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_24_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_24_pd	r/w	1	GPIO Pull Down Control
4	reg_gpio_24_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_24_drv	r/w	0	GPIO Driving Control
1	reg_gpio_24_smt	r/w	1	GPIO SMT Control
0	reg_gpio_24_ie	r/w	1	GPIO Input Enable

### 3.3.14 GPIO\_CFGCTL13

地址: 0x40000134

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

RSVD      RSVD      RSVD      reg\_gpio\_27\_func\_sel      RSVD      RSVD      reg\_gpio\_27\_pd  
 RSVD      RSVD      RSVD      reg\_gpio\_27\_pu      reg\_gpio\_27\_drv      reg\_gpio\_27\_smt  
 reg\_gpio\_27\_ie

RSVD      RSVD      RSVD      reg\_gpio\_26\_func\_sel      RSVD      RSVD      reg\_gpio\_26\_pd  
 reg\_gpio\_26\_pu      reg\_gpio\_26\_drv      reg\_gpio\_26\_smt      reg\_gpio\_26\_ie

位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_27_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )

位	名称	权限	复位值	描述
23:22	RSVD			
21	reg_gpio_27_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_27_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_27_drv	r/w	0	GPIO Driving Control
17	reg_gpio_27_smt	r/w	1	GPIO SMT Control
16	reg_gpio_27_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_26_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_26_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_26_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_26_drv	r/w	0	GPIO Driving Control
1	reg_gpio_26_smt	r/w	1	GPIO SMT Control
0	reg_gpio_26_ie	r/w	1	GPIO Input Enable

### 3.3.15 GPIO\_CFGCTL14

地址: 0x40000138

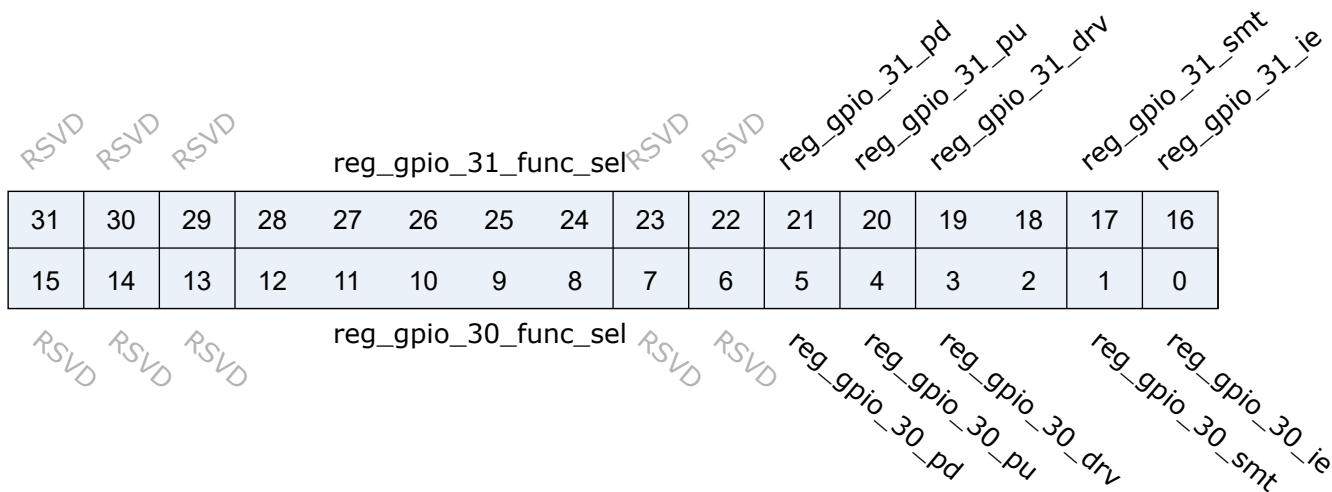
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	reg_gpio_29_func_sel	RSVD	RSVD	reg_gpio_29_pd	reg_gpio_29_pu	reg_gpio_29_drv	reg_gpio_29_smt	reg_gpio_29_ie					
RSVD	RSVD	RSVD	reg_gpio_28_func_sel	RSVD	RSVD	reg_gpio_28_pd	reg_gpio_28_pu	reg_gpio_28_drv	reg_gpio_28_smt	reg_gpio_28_ie					

位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_29_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )

位	名称	权限	复位值	描述
23:22	RSVD			
21	reg_gpio_29_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_29_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_29_drv	r/w	0	GPIO Driving Control
17	reg_gpio_29_smt	r/w	1	GPIO SMT Control
16	reg_gpio_29_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_28_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_28_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_28_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_28_drv	r/w	0	GPIO Driving Control
1	reg_gpio_28_smt	r/w	1	GPIO SMT Control
0	reg_gpio_28_ie	r/w	1	GPIO Input Enable

### 3.3.16 GPIO\_CFGCTL15

地址: 0x4000013c



位	名称	权限	复位值	描述
31:29	RSVD			
28:24	reg_gpio_31_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )

位	名称	权限	复位值	描述
23:22	RSVD			
21	reg_gpio_31_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_31_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_31_drv	r/w	0	GPIO Driving Control
17	reg_gpio_31_smt	r/w	1	GPIO SMT Control
16	reg_gpio_31_ie	r/w	1	GPIO Input Enable
15:13	RSVD			
12:8	reg_gpio_30_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO )
7:6	RSVD			
5	reg_gpio_30_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_30_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_30_drv	r/w	0	GPIO Driving Control
1	reg_gpio_30_smt	r/w	1	GPIO SMT Control
0	reg_gpio_30_ie	r/w	1	GPIO Input Enable
-1:22	RSVD			
21	reg_gpio_33_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_33_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_33_drv	r/w	0	GPIO Driving Control
17	reg_gpio_33_smt	r/w	1	GPIO SMT Control
16	reg_gpio_33_ie	r/w	1	GPIO Input Enable
15:6	RSVD			
5	reg_gpio_32_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_32_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_32_drv	r/w	0	GPIO Driving Control
1	reg_gpio_32_smt	r/w	1	GPIO SMT Control
0	reg_gpio_32_ie	r/w	1	GPIO Input Enable
-1:22	RSVD			
21	reg_gpio_35_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_35_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_35_drv	r/w	0	GPIO Driving Control
17	reg_gpio_35_smt	r/w	1	GPIO SMT Control

位	名称	权限	复位值	描述
16	reg_gpio_35_ie	r/w	1	GPIO Input Enable
15:6	RSVD			
5	reg_gpio_34_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_34_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_34_drv	r/w	0	GPIO Driving Control
1	reg_gpio_34_smt	r/w	1	GPIO SMT Control
0	reg_gpio_34_ie	r/w	1	GPIO Input Enable
-1:22	RSVD			
21	reg_gpio_37_pd	r/w	0	GPIO Pull Down Control
20	reg_gpio_37_pu	r/w	0	GPIO Pull Up Control
19:18	reg_gpio_37_drv	r/w	0	GPIO Driving Control
17	reg_gpio_37_smt	r/w	1	GPIO SMT Control
16	reg_gpio_37_ie	r/w	1	GPIO Input Enable
15:6	RSVD			
5	reg_gpio_36_pd	r/w	0	GPIO Pull Down Control
4	reg_gpio_36_pu	r/w	0	GPIO Pull Up Control
3:2	reg_gpio_36_drv	r/w	0	GPIO Driving Control
1	reg_gpio_36_smt	r/w	1	GPIO SMT Control
0	reg_gpio_36_ie	r/w	1	GPIO Input Enable

### 3.3.17 GPIO\_CFGCTL30

地址: 0x40000180

reg\_gpio\_31\_i  
reg\_gpio\_30\_i  
reg\_gpio\_29\_i  
reg\_gpio\_28\_i  
reg\_gpio\_27\_i  
reg\_gpio\_26\_i  
reg\_gpio\_25\_i  
reg\_gpio\_24\_i  
reg\_gpio\_23\_i  
reg\_gpio\_22\_i  
reg\_gpio\_21\_i  
reg\_gpio\_20\_i  
reg\_gpio\_19\_i  
reg\_gpio\_18\_i  
reg\_gpio\_17\_i  
reg\_gpio\_16\_i

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

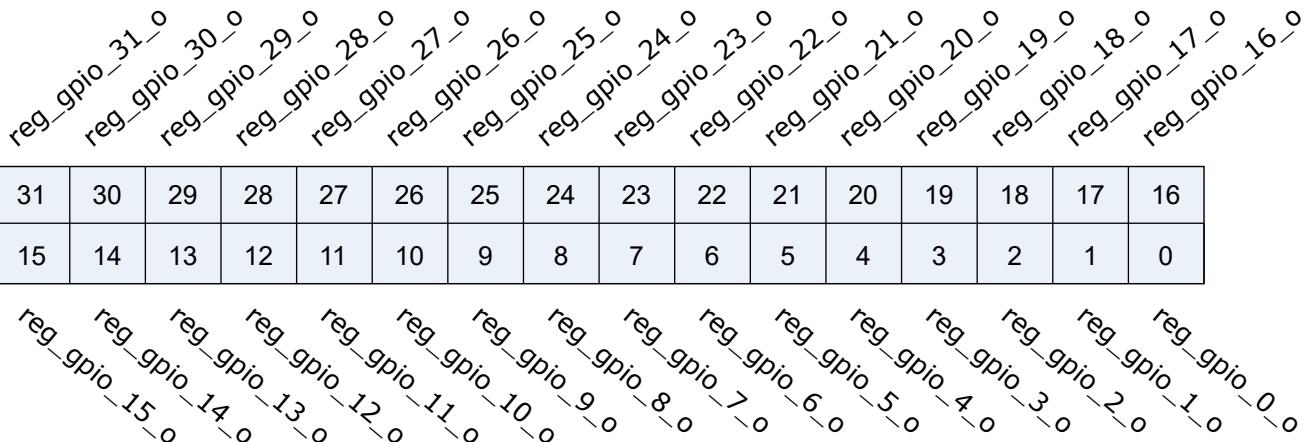
reg\_gpio\_15\_i  
reg\_gpio\_14\_i  
reg\_gpio\_13\_i  
reg\_gpio\_12\_i  
reg\_gpio\_11\_i  
reg\_gpio\_10\_i  
reg\_gpio\_9\_i  
reg\_gpio\_8\_i  
reg\_gpio\_7\_i  
reg\_gpio\_6\_i  
reg\_gpio\_5\_i  
reg\_gpio\_4\_i  
reg\_gpio\_3\_i  
reg\_gpio\_2\_i  
reg\_gpio\_1\_i  
reg\_gpio\_0\_i

位	名称	权限	复位值	描述
31	reg_gpio_31_i	r	0	Register Controlled GPIO Input value
30	reg_gpio_30_i	r	0	Register Controlled GPIO Input value
29	reg_gpio_29_i	r	0	Register Controlled GPIO Input value
28	reg_gpio_28_i	r	0	Register Controlled GPIO Input value
27	reg_gpio_27_i	r	0	Register Controlled GPIO Input value
26	reg_gpio_26_i	r	0	Register Controlled GPIO Input value
25	reg_gpio_25_i	r	0	Register Controlled GPIO Input value
24	reg_gpio_24_i	r	0	Register Controlled GPIO Input value
23	reg_gpio_23_i	r	0	Register Controlled GPIO Input value
22	reg_gpio_22_i	r	0	Register Controlled GPIO Input value
21	reg_gpio_21_i	r	0	Register Controlled GPIO Input value
20	reg_gpio_20_i	r	0	Register Controlled GPIO Input value
19	reg_gpio_19_i	r	0	Register Controlled GPIO Input value
18	reg_gpio_18_i	r	0	Register Controlled GPIO Input value
17	reg_gpio_17_i	r	0	Register Controlled GPIO Input value
16	reg_gpio_16_i	r	0	Register Controlled GPIO Input value
15	reg_gpio_15_i	r	0	Register Controlled GPIO Input value
14	reg_gpio_14_i	r	0	Register Controlled GPIO Input value
13	reg_gpio_13_i	r	0	Register Controlled GPIO Input value
12	reg_gpio_12_i	r	0	Register Controlled GPIO Input value
11	reg_gpio_11_i	r	0	Register Controlled GPIO Input value
10	reg_gpio_10_i	r	0	Register Controlled GPIO Input value
9	reg_gpio_9_i	r	0	Register Controlled GPIO Input value
8	reg_gpio_8_i	r	0	Register Controlled GPIO Input value
7	reg_gpio_7_i	r	0	Register Controlled GPIO Input value
6	reg_gpio_6_i	r	0	Register Controlled GPIO Input value
5	reg_gpio_5_i	r	0	Register Controlled GPIO Input value
4	reg_gpio_4_i	r	0	Register Controlled GPIO Input value
3	reg_gpio_3_i	r	0	Register Controlled GPIO Input value
2	reg_gpio_2_i	r	0	Register Controlled GPIO Input value
1	reg_gpio_1_i	r	0	Register Controlled GPIO Input value

位	名称	权限	复位值	描述
0	reg_gpio_0_i	r	0	Register Controlled GPIO Input value

### 3.3.18 GPIO\_CFGCTL32

地址: 0x40000188

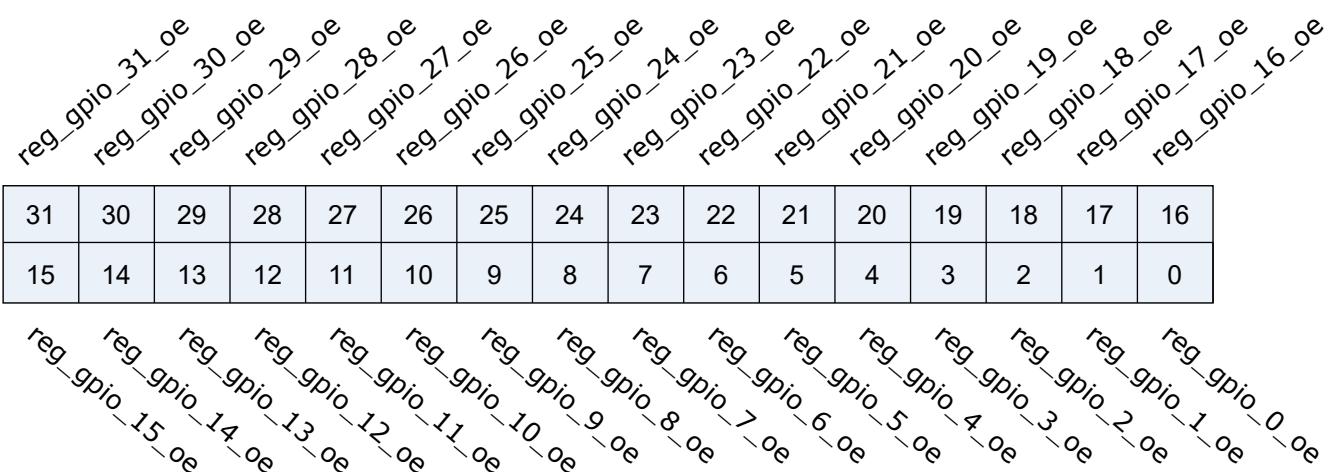


位	名称	权限	复位值	描述
31	reg_gpio_31_o	r/w	0	Register Controlled GPIO Output Value
30	reg_gpio_30_o	r/w	0	Register Controlled GPIO Output Value
29	reg_gpio_29_o	r/w	0	Register Controlled GPIO Output Value
28	reg_gpio_28_o	r/w	0	Register Controlled GPIO Output Value
27	reg_gpio_27_o	r/w	0	Register Controlled GPIO Output Value
26	reg_gpio_26_o	r/w	0	Register Controlled GPIO Output Value
25	reg_gpio_25_o	r/w	0	Register Controlled GPIO Output Value
24	reg_gpio_24_o	r/w	0	Register Controlled GPIO Output Value
23	reg_gpio_23_o	r/w	0	Register Controlled GPIO Output Value
22	reg_gpio_22_o	r/w	0	Register Controlled GPIO Output Value
21	reg_gpio_21_o	r/w	0	Register Controlled GPIO Output Value
20	reg_gpio_20_o	r/w	0	Register Controlled GPIO Output Value
19	reg_gpio_19_o	r/w	0	Register Controlled GPIO Output Value
18	reg_gpio_18_o	r/w	0	Register Controlled GPIO Output Value
17	reg_gpio_17_o	r/w	0	Register Controlled GPIO Output Value
16	reg_gpio_16_o	r/w	0	Register Controlled GPIO Output Value

位	名称	权限	复位值	描述
15	reg_gpio_15_o	r/w	0	Register Controlled GPIO Output Value
14	reg_gpio_14_o	r/w	0	Register Controlled GPIO Output Value
13	reg_gpio_13_o	r/w	0	Register Controlled GPIO Output Value
12	reg_gpio_12_o	r/w	0	Register Controlled GPIO Output Value
11	reg_gpio_11_o	r/w	0	Register Controlled GPIO Output Value
10	reg_gpio_10_o	r/w	0	Register Controlled GPIO Output Value
9	reg_gpio_9_o	r/w	0	Register Controlled GPIO Output Value
8	reg_gpio_8_o	r/w	0	Register Controlled GPIO Output Value
7	reg_gpio_7_o	r/w	0	Register Controlled GPIO Output Value
6	reg_gpio_6_o	r/w	0	Register Controlled GPIO Output Value
5	reg_gpio_5_o	r/w	0	Register Controlled GPIO Output Value
4	reg_gpio_4_o	r/w	0	Register Controlled GPIO Output Value
3	reg_gpio_3_o	r/w	0	Register Controlled GPIO Output Value
2	reg_gpio_2_o	r/w	0	Register Controlled GPIO Output Value
1	reg_gpio_1_o	r/w	0	Register Controlled GPIO Output Value
0	reg_gpio_0_o	r/w	0	Register Controlled GPIO Output Value

### 3.3.19 GPIO\_CFGCTL34

地址: 0x400000190



位	名称	权限	复位值	描述
31	reg_gpio_31_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
30	reg_gpio_30_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
29	reg_gpio_29_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
28	reg_gpio_28_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
27	reg_gpio_27_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
26	reg_gpio_26_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
25	reg_gpio_25_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
24	reg_gpio_24_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
23	reg_gpio_23_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
22	reg_gpio_22_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
21	reg_gpio_21_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
20	reg_gpio_20_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
19	reg_gpio_19_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
18	reg_gpio_18_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
17	reg_gpio_17_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
16	reg_gpio_16_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
15	reg_gpio_15_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
14	reg_gpio_14_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)

位	名称	权限	复位值	描述
13	reg_gpio_13_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
12	reg_gpio_12_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
11	reg_gpio_11_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
10	reg_gpio_10_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
9	reg_gpio_9_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
8	reg_gpio_8_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
7	reg_gpio_7_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
6	reg_gpio_6_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
5	reg_gpio_5_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
4	reg_gpio_4_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
3	reg_gpio_3_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
2	reg_gpio_2_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
1	reg_gpio_1_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)
0	reg_gpio_0_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO)

### 3.3.20 GPIO\_CFGCTL35

地址: 0x40000194

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

位	名称	权限	复位值	描述
31:0	reg_gpio_int_mask1	r/w	32'hFFFFFF	reg_gpio_int_mask[31:0]

### 3.3.21 GPIO\_INT\_STAT1

地址: 0x400001a8

gpio\_int\_stat1

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

gpio\_int\_stat1

位	名称	权限	复位值	描述
31:0	gpio_int_stat1	r	0	gpio_int_stat[31:0]

### 3.3.22 GPIO\_INT\_CLR1

地址: 0x400001b0

reg\_gpio\_int\_clr1

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

reg\_gpio\_int\_clr1

位	名称	权限	复位值	描述
31:0	reg_gpio_int_clr1	r/w	0	reg_gpio_int_clr[31:0]

### 3.3.23 GPIO\_INT\_MODE\_SET1

地址: 0x400001c0

*RSVD RSVD*

reg\_gpio\_int\_mode\_set1

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

reg\_gpio\_int\_mode\_set1

位	名称	权限	复位值	描述
31:30	RSVD			
29:0	reg_gpio_int_mode_set1	r/w	0	reg_gpio9_int_mode[2:0], ..... reg_gpio1_int_mode[2:0], reg_gpio0_int_mode

### 3.3.24 GPIO\_INT\_MODE\_SET2

地址: 0x400001c4

*RSVD RSVD*

reg\_gpio\_int\_mode\_set2

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

reg\_gpio\_int\_mode\_set2

位	名称	权限	复位值	描述
31:30	RSVD			
29:0	reg_gpio_int_mode_set2	r/w	0	reg_gpio19_int_mode[2:0], ..... reg_gpio11_int_mode[2:0], reg_gpio10_int_mode

### 3.3.25 GPIO\_INT\_MODE\_SET3

地址: 0x400001c8

*RSVD RSVD*

reg\_gpio\_int\_mode\_set3

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

reg\_gpio\_int\_mode\_set3

位	名称	权限	复位值	描述
31:30	RSVD			
29:0	reg_gpio_int_mode_set3	r/w	0	reg_gpio29_int_mode[2:0], ..... reg_gpio21_int_mode[2:0], reg_gpio20_int_mode

### 3.3.26 GPIO\_INT\_MODE\_SET4

地址: 0x4000001cc

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

reg\_gpio\_int\_mode\_set4

位	名称	权限	复位值	描述
31:6	RSVD			
5:0	reg_gpio_int_mode_set4	r/w	0	reg_gpio31_int_mode[2:0], reg_gpio30_int_mode

### 3.3.27 GPIO\_INT2\_MASK1

地址: 0x4000001d0

reg_gpio_int2_mask1																			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				

reg\_gpio\_int2\_mask1

位	名称	权限	复位值	描述
31:0	reg_gpio_int2_mask1	r/w	32'hFFFFFF	reg_gpio_int2_mask[31:0]

### 3.3.28 GPIO\_INT2\_STAT1

地址: 0x400001d4

gpio\_int2\_stat1

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

gpio\_int2\_stat1

位	名称	权限	复位值	描述
31:0	gpio_int2_stat1	r	0	gpio_int2_stat[31:0]

### 3.3.29 GPIO\_INT2\_CLR1

地址: 0x400001d8

reg\_gpio\_int2\_clr1

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

reg\_gpio\_int2\_clr1

位	名称	权限	复位值	描述
31:0	reg_gpio_int2_clr1	r/w	0	reg_gpio_int2_clr[31:0]

### 3.3.30 GPIO\_INT2\_MODE\_SET1

地址: 0x400001dc

RSVD  
RSVD

reg\_gpio\_int2\_mode\_set1

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

reg\_gpio\_int2\_mode\_set1

位	名称	权限	复位值	描述
31:30	RSVD			

位	名称	权限	复位值	描述
29:0	reg_gpio_int2_mode_set1	r/w	0	reg_gpio9_int2_mode[2:0], ..... reg_gpio1_int2_mode[2:0], reg_gpio0_int2_mode

### 3.3.31 GPIO\_INT2\_MODE\_SET2

地址: 0x4000001e0

RSVD RSVD

reg\_gpio\_int2\_mode\_set2

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

reg\_gpio\_int2\_mode\_set2

位	名称	权限	复位值	描述
31:30	RSVD			
29:0	reg_gpio_int2_mode_set2	r/w	0	reg_gpio19_int2_mode[2:0], ..... reg_gpio11_int2_mode[2:0], reg_gpio10_int2_mode

### 3.3.32 GPIO\_INT2\_MODE\_SET3

地址: 0x4000001e4

RSVD RSVD

reg\_gpio\_int2\_mode\_set3

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

reg\_gpio\_int2\_mode\_set3

位	名称	权限	复位值	描述
31:30	RSVD			
29:0	reg_gpio_int2_mode_set3	r/w	0	reg_gpio29_int2_mode[2:0], ..... reg_gpio21_int2_mode[2:0], reg_gpio20_int2_mode

### 3.3.33 GPIO\_INT2\_MODE\_SET4

地址: 0x400001e8

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

reg\_gpio\_int2\_mode\_set4

位	名称	权限	复位值	描述
31:6	RSVD			
5:0	reg_gpio_int2_mode_set4	r/w	0	reg_gpio31_int2_mode[2:0],reg_gpio30_int2_mode
-1:12	RSVD			
11	reg_usb_use_ctrl	r/w	1'b1	1: USB XCVR use on-chip usb controller ; 0: USB XCVR usb off-chip usb controller
10:0	RSVD			

## 4.1 简介

芯片内置一个 12bits 的逐次逼近式模拟数字转换器 (ADC)，支持 12 路外部模拟输入和若干内部模拟信号选择。ADC 支持以下四种模式：单次单通道转换、连续单通道转换、单次多通道转换和连续多通道转换模式。转换结果为 12/14/16bits 左对齐模式。ADC 拥有深度为 32 的 FIFO，支持多种中断和 DMA 操作。ADC 除了用于普通模拟信号测量外，还可以用于测量供电电压，此外 ADC 还可以通过测量内/外部二极管电压用于温度检测。

## 4.2 主要特点

- 高性能
  - 可以选择 12/14/16bits 转换结果输出
  - ADC 最大工作时钟为 2MHZ
  - 支持 2.0V,3.2V 可选内部参考电压
  - 支持 DMA 将转换结果搬运到内存
  - 支持单次单通道转换、连续单通道转换、单次多通道转换和连续多通道转换模式四种模式
  - 支持单端与差分两种输入模式
  - 支持抖动补偿
  - 支持用户自行设定转换结果偏移值
- 模拟通道数
  - 12 路外部模拟通道
  - 2 路 DAC 内部通道
  - 1 路 VBAT/2 通道
  - 1 路 TSEN 通道

## 4.3 功能描述

ADC 模块基本框图如图所示。

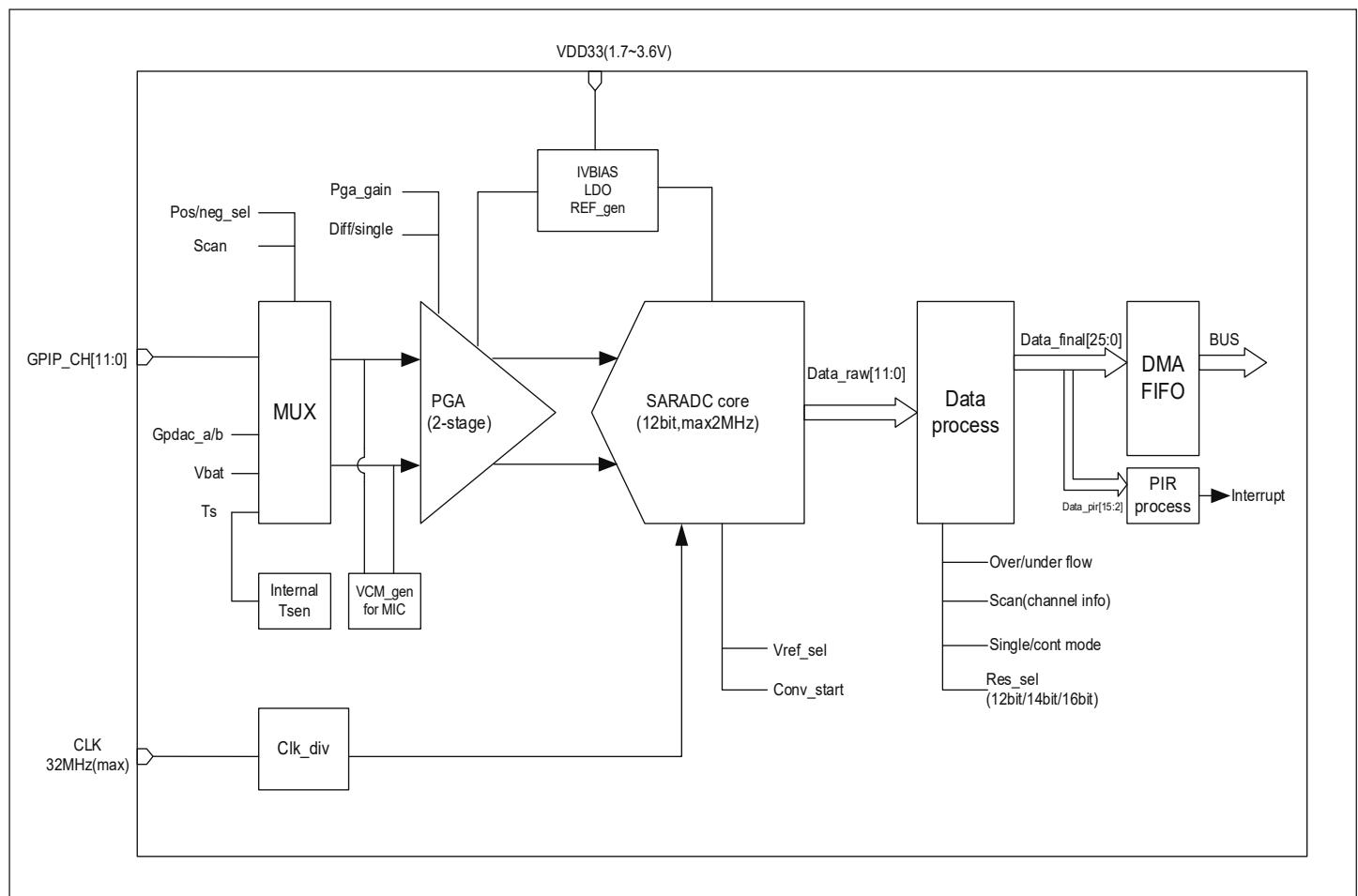


图 4.1: ADC 基本框图

ADC 模块包含五大部分：分别为前端输入通道选择器，程控放大器，ADC 采样模块，数据处理模块以及 FIFO。输入通道选择器用于选择需要采样的通道，既包含外部模拟信号，也包含内部模拟信号。程控放大器用于对输入信号做进一步处理，可以根据输入信号的特点进行设定，以便得到更准确的转换值。ADC 采样模块是最主要的功能模块，通过逐次比较的方式，实现模拟信号到数字信号的转换。转换结果的精度为 12bits，数据处理模块负责进一步处理转换的结果，包括添加通道信息等。最后得到的数据会被推送到最后端的 FIFO 中。

### 4.3.1 ADC 引脚和内部信号

表 4.1: ADC 内部信号

内部信号	信号类型	信号描述
VBAT/2	Input	从电源引脚分压过来的电压信号
TSEN	Input	内部温度传感器输出电压

表 4.1: ADC 内部信号 (continued)

内部信号	信号类型	信号描述
VREF	Input	内部模拟模块参考电压
DACOUTA	Input	DAC 模块输出
DACOUTB	Input	DAC 模块输出

表 4.2: ADC 外部引脚

外部引脚	信号类型	信号描述
VDDA	Input	模拟模块供电电压正极
VSSA	Input	模拟模块供电电压负极
ADC_CHX	Input	模拟输入引脚, 总共 12 路

### 4.3.2 ADC 通道

ADC 采样可以选择的通道包括外部模拟引脚的输入信号和芯片内部可选信号, 具体包括:

- ADC CH0
- ADC CH1
- ADC CH2
- ADC CH3
- ADC CH4
- ADC CH5
- ADC CH6
- ADC CH7
- ADC CH8
- ADC CH9
- ADC CH10
- ADC CH11
- VDDA
- VSSA

- DAC OUTA
- DAC OUTB
- VBAT/2
- TSEN
- VREF
- GND

需要注意的是，如果选择 VBAT/2 或 TSEN 作为输入待采信号，需要把 `gpadc_vbat_en` 或 `gpadc_ts_en` 置位。ADC 模块可以支持单端输入或者差分输入，如果是单端输入模式，负极输入通道需要选择 GND。

#### 4.3.3 ADC 时钟

ADC 模块的工作时钟来源如下图所示。

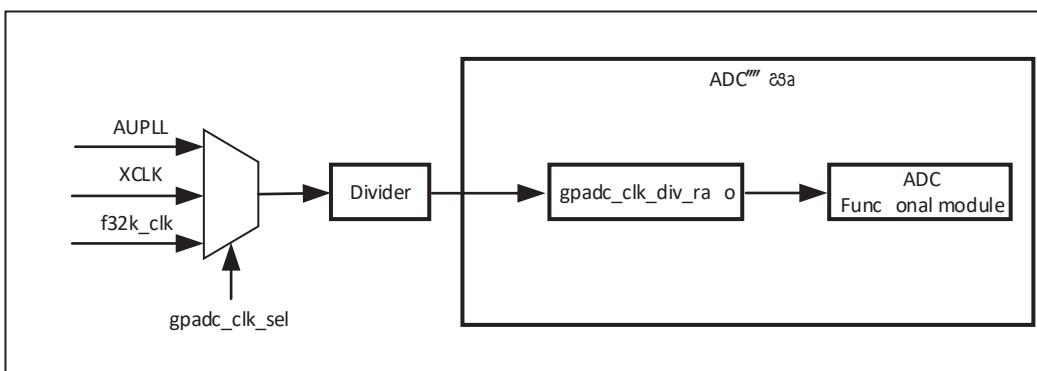


图 4.2: ADC 时钟

ADC 的时钟源可以选择来自 Audio PLL, XCLK 或者 f32k, 时钟源的选择在 GLB 模块中设定，选择的同时可以通过分频器对时钟源进行分频。在一般的电压测量应用下，会选择使用 XCLK 作为时钟源。如果是在音频应用下，可以使用 Audio PLL 去产生诸如 8KHZ,44.1KHZ 这样的常用采样时钟。f32k 是低频时钟，为 MCU 睡眠的时候提供唤醒的时钟。

在 ADC 模块内部，提供了一个时钟分频，可以对输入时钟进行 1/4/8/12/16/20/24/32 分频。用户可以根据实际采样需求，自行调整 ADC 的时钟源和分频系数，注意 ADC 的最大输入时钟是 2MHZ。

`gpadc_32m_clk_div` 分频寄存器宽度为 6bits, 最大分频为 64, 分频公式为  $f_{out} = f_{source}/(gpadc\_32m\_clk\_div + 1)$ 。  
`gpadc_clk_div_ratio` 分频寄存器位于 ADC 模块内部，宽度为 3bits，其分频值定义如下：

<code>ADC_CLK_DIV_1,</code>	<i>/*!&lt; ADC clock:on 32M clock is 32M */</i>
<code>ADC_CLK_DIV_4,</code>	<i>/*!&lt; ADC clock:on 32M clock is 8M */</i>

(continues on next page)

(continued from previous page)

```
ADC_CLK_DIV_8,          /*!< ADC clock:on 32M clock is 4M */
ADC_CLK_DIV_12,         /*!< ADC clock:on 32M clock is 2.666M */
ADC_CLK_DIV_16,         /*!< ADC clock:on 32M clock is 2M */
ADC_CLK_DIV_20,         /*!< ADC clock:on 32M clock is 1.6M */
ADC_CLK_DIV_24,         /*!< ADC clock:on 32M clock is 1.333M */
ADC_CLK_DIV_32,         /*!< ADC clock:on 32M clock is 1M */
```

用户若想调整 ADC 的输入时钟，将有四种方式。

1. 切换时钟源，XTAL 默认是 32MHZ，Audio PLL(可配置为 11.288MHZ 或 11.2896MHZ)。
2. 使用在时钟模块中长度为 6BITS 的分频器。
3. 使用 ADC 模块内部分频器，可选分频为 1/4/8/12/16/20/24/32 分频。
4. 通过配置 `gpadc_res_sel` 寄存器，改变 OSR 的值，达成分频效果。如果 OSR=256，实际等效 ADC 的输入时钟做了 256 分频。

假设时钟源选择选择 Audio PLL=11.2896MHZ，GLB 分频选择配置为 1，ADC 内部分频器选择 ADC\_CLK\_DIV\_4，OSR=128 则最终的时钟输出为  $f_{out} = 11289600 / (1 + 1) / 4 / 128 = 11025\text{Hz}$

#### 4.3.4 ADC 转换模式

ADC 支持单通道转换和扫描转换两种模式，在单通道转换模式下，用户需要通过 `gpadc_pos_sel` 选择正极输入通道，通过 `gpadc_neg_sel` 选择负极输入通道，同时把 `gpadc_cont_conv_en` 控制位设置为 0，表示单通道转换，然后设置 `gpadc_conv_start` 控制位启动转换即可。

在扫描转换模式下，`gpadc_cont_conv_en` 控制位需要设置为 1，ADC 根据 `gpadc_scan_length` 控制位设定的转换通道个数，依次按照 `gpadc_reg_scn_posX(X=1, 2)` 和 `gpadc_reg_scn_negX(X=1, 2)` 寄存器组所设定的通道顺序，逐个进行转换，转换的结果会自动推入 ADC 的 FIFO。`gpadc_reg_scn_posX(X=1, 2)` 和 `gpadc_reg_scn_negX(X=1, 2)` 寄存器组所设定的通道可以相同，这也就意味着用户可以实现对一个通道进行多次采样转换。

ADC 的转换结果一般都是放入 FIFO 中，ADC 模块不提供转换完成中断，用户需要根据实际转换通道数，设定 FIFO 接收数据阈值中断，通过 FIFO 的阈值中断，作为 ADC 转换完成中断。

#### 4.3.5 ADC 结果

`gpadc_raw_data` 寄存器存放了 ADC 的原始结果，在单端模式下，数据有效位是 12bits，无符号位，在差分模式下，最高位为符号位，剩下 11bits 代表转换的结果。

`gpadc_dma_rdata` 寄存器存放了 ADC 的结果，这个结果里包含了 ADC 结果，符号位和通道信息，数据格式如下：

表 4.3: ADC 转换结果含义

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
含义	正极通道号				负极通道号				转换结果																	

转换结果的 bit21-bit25 是正极通道号, bit16-bit20 是负极通道号, bit0-bit15 是转换的数值。

gpadc\_res\_sel 控制位可以设定转换结果的位数为 12 位, 14 位, 和 16 位, 其中 14 位和 16 位是多次采样提高精度得到的结果, 其可以设置的值如下:

- 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

ADC 转换结果为左对齐模式, 当选择 12 位时, 转换结果的 bit15-bit4 有效, 当选择 14 位时, 转换结果的 bit15-bit2 有效, 当选择 16 位时, 转换结果的 bit15-bit0 有效。同样, 在差分模式下, 最高为是符号位, 也就是, 当选择 14 位时, bit15 是符号位, bit14-bit2 是转换结果, bit14 是 MSB, 在单端模式下, 没有符号位, 也就是, 当选择 12 位时, bit15-bit4 是转换结果, bit15 是 MSB。

在实际使用中, ADC 的结果一般都是放入 FIFO, 这在多通道扫描模式下尤为重要, 所以用户一般都是从 ADC FIFO 获取转换结果, ADC FIFO 的数据格式 gpadc\_dma\_rdata 寄存器中数据格式相同。

#### 4.3.6 ADC 异常中断

ADC 模块在正极采样饱和和负极采样饱和时可以产生中断, 可以通过 gpadc\_pos\_satur\_mask, gpadc\_neg\_satur\_mask 屏蔽各自中断, 当中断产生时, 可以通过 gpadc\_pos\_satur, 和 gpadc\_neg\_satur 寄存器查询中断状态, 同时可以通过 gpadc\_pos\_satur\_clr 和 gpadc\_neg\_satur\_clr 清除中断。该功能可以用来判断输入电压是否异常。

#### 4.3.7 ADC FIFO 与阈值中断

ADC 模块拥有深度为 32 的 FIFO, 数据宽度为 26bits, 当 ADC 完成转换后, 会自动将结果推入到 FIFO。ADC 的 FIFO 有如下状态和中断管理功能:

- FIFO Overrun 中断
- FIFO Underrun 中断
- FIFO 阈值中断

当 FIFO 已经填满, 但是用户没有通过 DMA 或者直接访问寄存器的方式把读取取走, 此时又有数据进入 FIFO, 那么此时模块会产生一个 FIFO Overrun 中断。当 FIFO 为空, 但是用户依旧向 FIFO 请求数据, 那么此时模块会产生 FIFO

Underrun 中断。

用户可以配置 FIFO 的阈值寄存器 `gpadc_fifo_thl`, 选择 FIFO 产生中断的阈值, 1、4、8、16 四种选择可选。如果 ADC 的 FIFO 数量达到设定的阈值数, 那么会产生阈值中断。

当中断产生时, 可以通过对应的 `clear` 位将中断标志清除掉。

利用 ADC 的 FIFO 用户可以实现三种模式获取数据: 查询模式, 中断模式, DMA 模式

查询模式

CPU 轮询 ADC FIFO 的长度, 当 FIFO 的长度不为空的时候, 说明 FIFO 中存在有效数据, CPU 可以从 FIFO 中读出这些数据。

中断模式

利用 FIFO 的阈值中断, 当中断产生, ADC 的数据个数到达阈值, CPU 可以在中断服务函数中可以读取 ADC FIFO 的长度, 将其全部读出。

**DMA** 模式

用户设定 `dmaen` 控制位, 可以配合 DMA 完成转换数据到内存的搬运, 在使用 DMA 模式时, 通过 `fifothl` 设置 ADC FIFO 发送 DMA 请求的数据个数阈值, DMA 在收到请求时, 会自动根据用户设定的参数, 从 FIFO 搬运指定个数的结果到对应的内存。

#### 4.3.8 ADC 设置流程

设置 **ADC** 时钟

根据 ADC 转换速度需求, 确定 ADC 的工作时钟, 设定 GLB 模块的 ADC 时钟源和分频, 结合 `clkdivrt`, 确定最终 ADC 模块的工作时钟频率。

根据使用的通道设置 **GPIO**

根据使用的模拟引脚, 确定使用的通道号, 初始化对应的 GPIO 为模拟功能, 需要注意的是, 在设定 GPIO 为模拟输入的时候, 不要设置 GPIO 的上拉或者下拉, 需要设置为浮空输入。

设定要转换的通道

根据使用的模拟通道和转换模式, 设定对应的通道寄存器, 对于单通道转换, 在 `possel` 和 `negsel` 寄存器中设置转换的通道信息。对于多通道扫描模式, 根据要扫描通道数目和扫描顺序, 设定 `scalen,scpX` 和 `scnX`。

设定数据读取方式

根据 ADC FIFO 介绍的读取数据方式, 选择使用的模式, 设置对应的寄存器。如果使用 DMA, 同样需要配置 DMA 的一个通道, 配合 ADC FIFO 完成数据的搬运。

启动转换

最后设置 `ressel` 选择数据转换结果的精度, 最后设置 `gben=1, cvst=1` 就可以启动 ADC 开始转换。当转换完成, 需要再次转换时, 需要将 `cvst` 设置为 0, 再设置为 1, 以便再次触发转换。

### 4.3.9 VBAT 测量

这里的 VBAT/2 测量的是芯片 VDD33 的电压，而不是外部的比如锂电池的电压，如果需要测量锂电池等供电源头的电压，可以将电压分压，然后输入 ADC 的 GPIO 模拟通道，测量 VDD33 的电压可以减少 GPIO 的使用。

ADC 模块测量的 VBAT/2 电压是经过分压的，实际输入到 ADC 模块的电压是 VDD33 的一半，即  $VBAT/2=VDD33/2$ 。由于电压经过分压，为了得到较高的精确度，建议 ADC 的参考电压选择 2V，采用单端模式，正极输入电压选择 VBAT/2，负极输入电压选择 GND，同时将 `vbaten` 设置为 1，启动转换后，将对应的转换结果乘以 2 就可以得到 VDD33 电压。

### 4.3.10 TSEN 测量

ADC 可以测量内部二极管或者外部二极管电压值，而二极管的压差和温度有关，所以通过测量二极管的电压，可以计算得到环境温度，我们称之为 Temperature Sensor，简称 TSEN。

TSEN 的测量原理是通过一个二极管上面测量两次不同大小的电流产生的电压差  $\Delta V$  随着温度的变化拟合的曲线，无论外部或者内部二极管的测量，最终输出的值和温度有关，都可以表示成  $\Delta(ADC\_out)=7.753T+X$ ，当我们知道了电压值，也就知道了温度  $T$ 。这里的  $X$  是一个偏移值，可以作为标准值，在实际使用前，我们需要确定  $X$ 。芯片厂商会在芯片出厂前，在标准温度下，例如室温 25 度，测量  $\Delta(ADC\_out)$ ，从而得到  $X$ 。在用户使用的时候，只要根据公式  $T=[\Delta(ADC\_out)-X]/7.753$ ，就可以得到温度  $T$ 。

在使用 TSEN 时，建议把 ADC 设置成 16bits 模式，通过多次采样以减少误差，参考电压选择 2V 以提高精度，设置 `tsen` 为 1 以便启动 TSEN 功能，如果选择内部二极管，`tsxten=0`，如果选择外部二极管，`tsxten=1`，根据实际情况选择正向输入通道，如果是内部二极管，选择 TSEN 通道，如果是外部，选择对应的模拟 GPIO 通道，负极输入端选择 GND。在上述设定完毕后，设置 `tsdc=0`，启动测量，得到测量结果  $V0$ ，再设置 `tsdc=1`，启动测量，得到测量结果  $V1$ ， $\Delta(ADC\_out)=V1-V0$ ，根据公式  $T=[\Delta(ADC\_out)-X]/7.753$ ，得到温度  $T$ 。

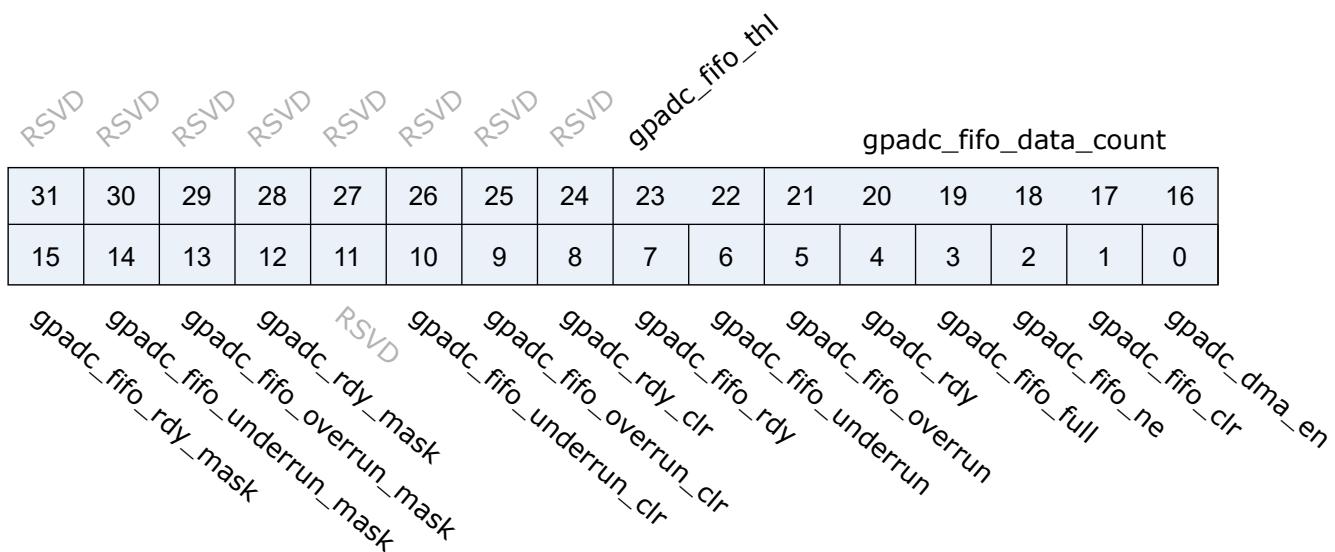
## 4.4 寄存器描述

名称	描述
<code>gpadc_config</code>	GPADC configuration
<code>gpadc_dma_rdata</code>	GPADC DMA read data
<code>gpdac_config</code>	GPDAC configuration
<code>gpdac_dma_config</code>	GPDAC dma configuration
<code>gpdac_dma_wdata</code>	GPDAC dma write data
<code>gpadc_reg_cmd</code>	GPADC register configuration 0
<code>gpadc_reg_config1</code>	GPADC register configuration 1
<code>gpadc_reg_config2</code>	GPADC register configuration 2
<code>gpadc_reg_scn_pos1</code>	adc conversation sequence 1
<code>gpadc_reg_scn_pos2</code>	adc conversation sequence 2

名称	描述
gpadc_reg_scn_neg1	adc conversation sequence 3
gpadc_reg_scn_neg2	adc conversation sequence 4
gpadc_reg_status	GPADC register status
gpadc_reg_isr	GPADC register operation
gpadc_reg_raw_result	GPADC register raw result
gpadc_reg_define	GPADC register define

#### 4.4.1 gpadc\_config

地址: 0x40002000



位	名称	权限	复位值	描述
31:24	RSVD			
23:22	gpadc_fifo_thl	r/w	2'd0	fifo threshold 2'b00: 1 data 2'b01: 4 data 2'b10: 8 data 2'b11: 16 data
21:16	gpadc_fifo_data_count	r	6'd0	fifo data number
15	gpadc_fifo_rdy_mask	r/w	1'b1	write 1 mask

位	名称	权限	复位值	描述
14	gpadc_fifo_underrun_mask	r/w	1'b0	write 1 mask
13	gpadc_fifo_overrun_mask	r/w	1'b0	write 1 mask
12	gpadc_rdy_mask	r/w	1'b0	write 1 mask
11	RSVD			
10	gpadc_fifo_underrun_clr	w1c	1'b0	Write 1 to clear flag
9	gpadc_fifo_overrun_clr	w1c	1'b0	Write 1 to clear flag
8	gpadc_rdy_clr	w1c	1'b0	Write 1 to clear flag
7	gpadc_fifo_rdy	r	1'b0	FIFO ready interrupt flag
6	gpadc_fifo_underrun	r	1'b0	FIFO underrun interrupt flag
5	gpadc_fifo_overrun	r	1'b0	FIFO overrun interrupt flag
4	gpadc_rdy	r	1'b0	Conversion data ready interrupt flag
3	gpadc_fifo_full	r	1'b0	FIFO full flag
2	gpadc_fifo_ne	r	1'b0	FIFO not empty flag
1	gpadc_fifo_clr	w1c	1'b0	FIFO clear signal
0	gpadc_dma_en	r/w	1'b0	GPADC DMA enable

#### 4.4.2 gpadc\_dma\_rdata

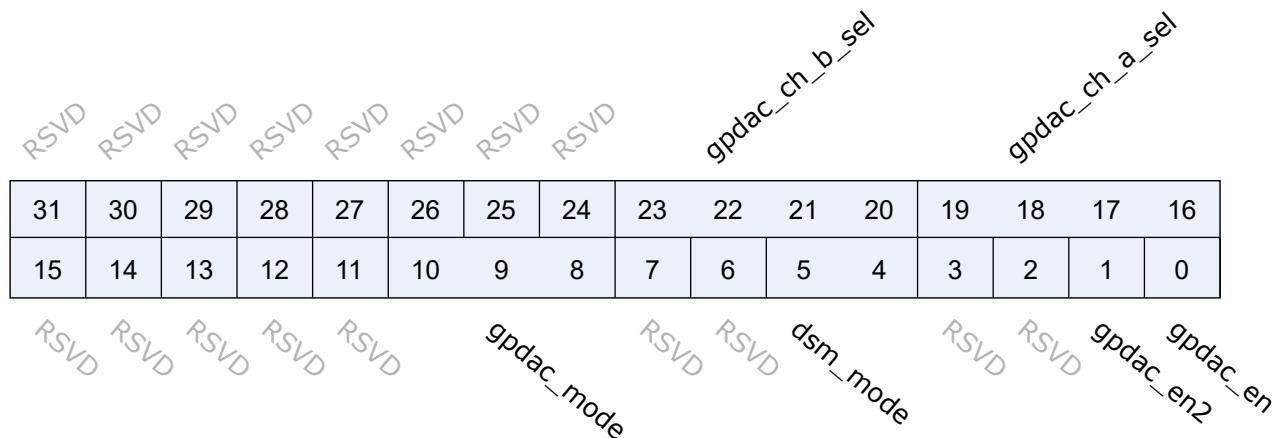
地址: 0x40002004

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

位	名称	权限	复位值	描述
31:26	RSVD			
25:0	gpadc_dma_rdata	r	26'd0	GPADC final conversion result stored in the FIFO

#### 4.4.3 gpdac\_config

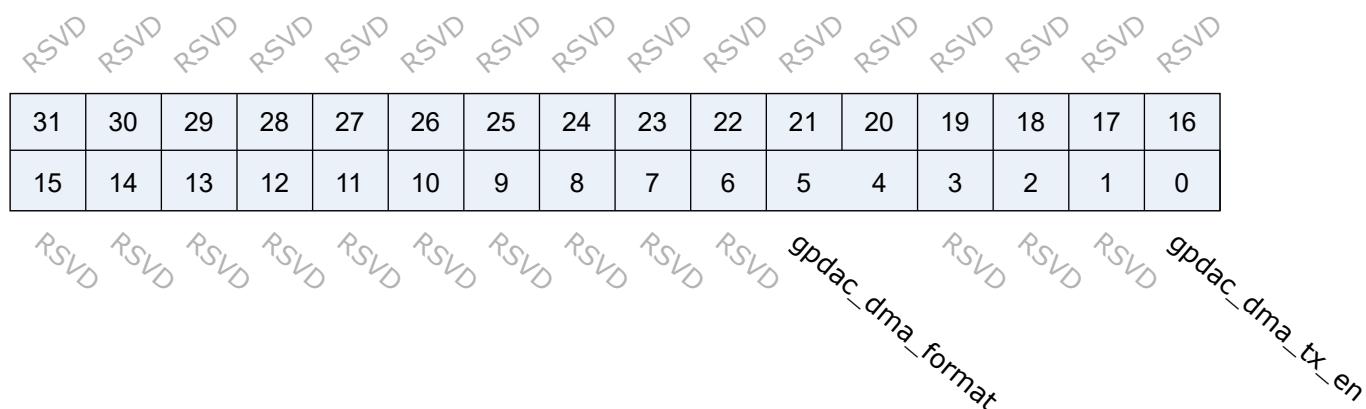
地址: 0x40002040



位	名称	权限	复位值	描述
31:24	RSVD			
23:20	gpdac_ch_b_sel	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	gpdac_ch_a_sel	r/w	0	Channel A Source Select 0: Reg 1: DMA 2: DMA + Filter 3: Sin Gen
15:11	RSVD			
10:8	gpdac_mode	r/w	0	0:32k, 1:16k, 3:8k, 4:512k(for DMA only)
7:6	RSVD			
5:4	dsm_mode	r/w	0	0:bypass, 1:dsm order=1, 2: dsm order=2
3:2	RSVD			
1	gpdac_en2	r/w	0	GPDAC enable 2 (for B channel)
0	gpdac_en	r/w	0	GPDAC enable

#### 4.4.4 gpdac\_dma\_config

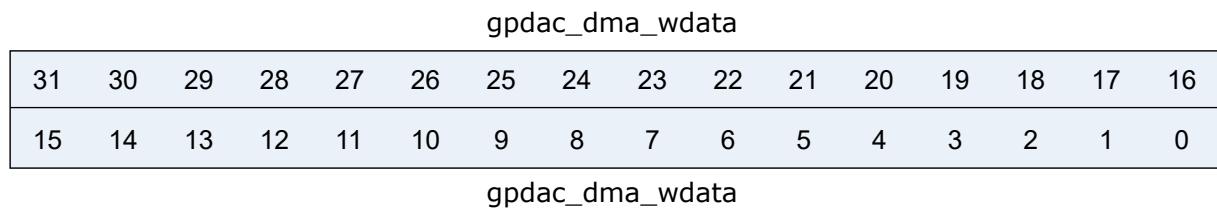
地址: 0x40002044



位	名称	权限	复位值	描述
31:6	RSVD			
5:4	gpdac_dma_format	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	gpdac_dma_tx_en	r/w	0	GPDAC DMA TX enable

#### 4.4.5 gpdac\_dma\_wdata

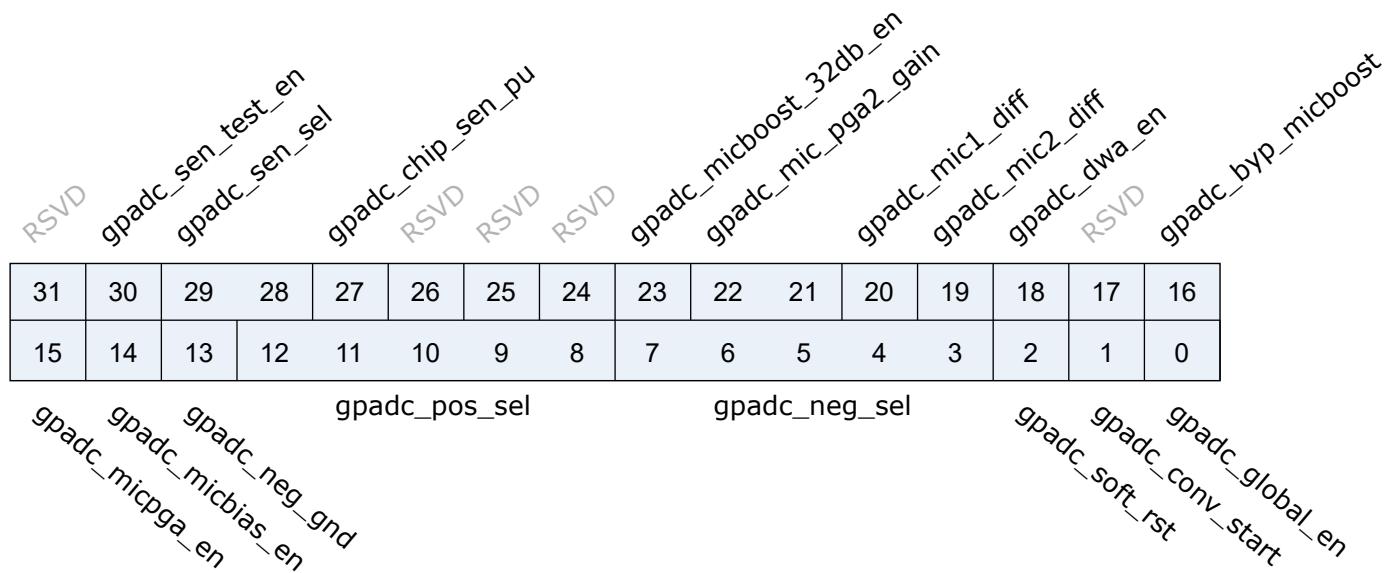
地址: 0x40002048



位	名称	权限	复位值	描述
31:0	gpdac_dma_wdata	w	x	GPDAC DMA TX data

#### 4.4.6 gpadc\_reg\_cmd

地址: 0x4000f90c



位	名称	权限	复位值	描述
31	RSVD			
30	gpadc_sen_test_en	r/w	1'b0	enable sensor dc test mux
29:28	gpadc_sen_sel	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
27	gpadc_chip_sen_pu	r/w	1'b0	enable chip sensor test 1'b0: disable 1'b1: enable
26:24	RSVD			
23	gpadc_micboost_32db_en	r/w	1'b0	micboost 32db enable 1'b0: 16dB 1'b1: 32dB
22:21	gpadc_mic_pga2_gain	r/w	2'h0	mic_pga2_gain 2'h0: 0dB 2'h1: 6dB 2'h2: -6dB 2'h3: 12dB
20	gpadc_mic1_diff	r/w	1'b0	mic1 diff enable 1'b0: single 1'b1: diff

位	名称	权限	复位值	描述
19	gpadc_mic2_diff	r/w	1'b0	mic2 diff enable 1'b0: single 1'b1: diff
18	gpadc_dwa_en	r/w	1'b0	dwa enable 1'b0: dwa disable 1'b1: dwa enable
17	RSVD			
16	gpadc_byp_micboost	r/w	1'b0	micboost amp bypass 1'b0: not bypass 1'b1: bypass
15	gpadc_micpga_en	r/w	1'b0	micpga enable 1'b0: micpga disable 1'b1: miapga enable
14	gpadc_micbias_en	r/w	1'b0	enable micbias 1'b0: micbias power down 1'b1: miabias power on
13	gpadc_neg_gnd	r/w	1'b0	set negative input of adc to ground 1'b0: disable 1'b1: enable

位	名称	权限	复位值	描述
12:8	gpadc_pos_sel	r/w	5'hf	select adc positive input in none-scan mode 5 'h0 gpio0 5'h1 gpio1 5'h2 gpio2 5 'h3 gpio3 5'h4 gpio4 5'h5 gpio5 5 'h6 gpio6 5'h7 gpio7 5'h8 gpio8 5 'h9 gpio9 5'h10 gpio10 5'h11 gpio11 5 'h12 daca 5'h13 dacb 5'h14 temp_p 5 'h15 temp_n 5'h16 vref 5'h17 atest 5 'h18 vbat/2 5'h19 vp3_diode 5'h20 vp2_diode 5 'h21 vp1_diode 5'h22 vp0_diode 5'h23 31 avss

位	名称	权限	复位值	描述
7:3	gpadc_neg_sel	r/w	5'hf	select adc positive input in none-scan mode 5 'h0 gpio0 5'h1 gpio1 5'h2 gpio2 5 'h3 gpio3 5'h4 gpio4 5'h5 gpio5 5 'h6 gpio6 5'h7 gpio7 5'h8 gpio8 5 'h9 gpio9 5'h10 gpio10 5'h11 gpio11 5 'h12 daca 5'h13 dacb 5'h14 temp_p 5 'h15 temp_n 5'h16 vref 5'h17 atest 5 'h18 vbat/2 5'h19 vn3_diode 5'h20 vn2_diode 5 'h21 vn1_diode 5'h22 vn0_diode 5'h23 31 avss
2	gpadc_soft_RST	r/w	1'b0	user reset the whole block 1'h0: not reset 1'h1: reset
1	gpadc_conv_start	r/w	1'b0	1'h0: stop converation 1'h1: start converation
0	gpadc_global_en	r/w	1'b0	1'h0: disable ADC 1'h1: enable ADC

#### 4.4.7 gpadc\_reg\_config1

地址: 0x4000f910

RSVD	RSVD	RSVD	RSVD	RSVD	gpadc_dither_en	gpadc_scan_en	gpadc_scan_length	gpadc_clk_div_ratio	gpadc_clk_ana_inv	RSVD
31	30	29	28	27	26	25	24	23	22	21
15	14	13	12	11	10	9	8	7	6	5
					gpadc_lowv_det_en	gpadc_vcm_hyst_sel	RSVD	RSVD	gpadc_res_sel	gpadc_cal_os_en
										gpadc_cont_conv_en

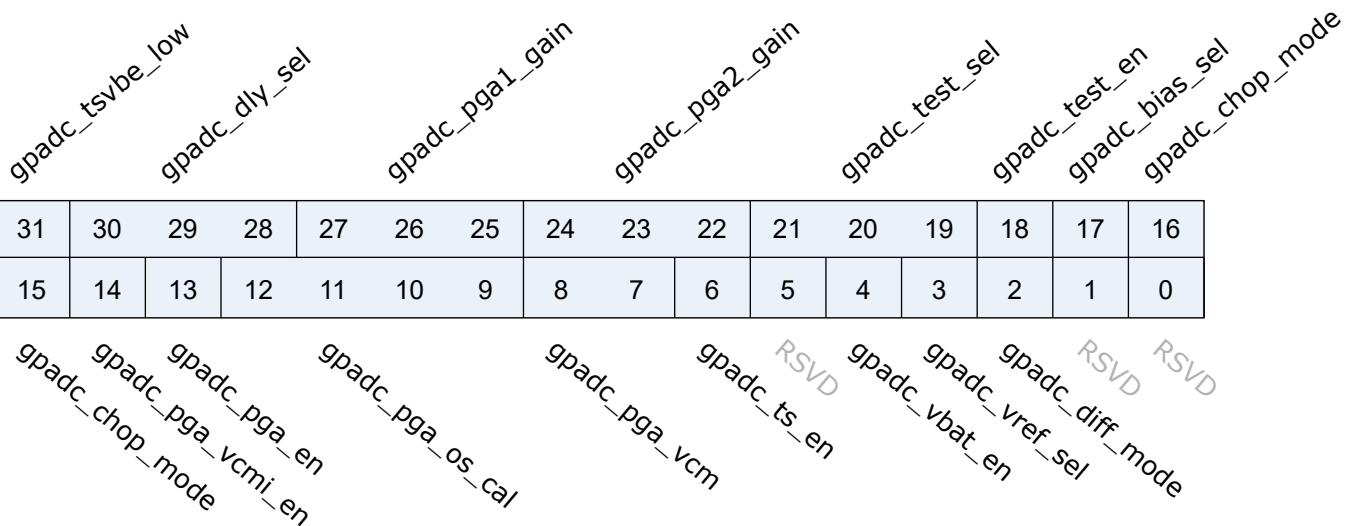
位	名称	权限	复位值	描述
31	RSVD			
30:29	gpadc_v18_sel	r/w	2'h0	internal vdd18 select
28:27	gpadc_v11_sel	r/w	2'h0	internal vdd11 select
26	gpadc_dither_en	r/w	1'h0	Dither compensation enable
25	gpadc_scan_en	r/w	1'h0	select scan mode enable: 0: select gpadc_pos/neg_sel;1: select : select gpadc_scan_pos_x and gpadc_scan_neg_x

位	名称	权限	复位值	描述
24:21	gpadc_scan_length	r/w	4'h0	<p>select scan mode length</p> <p>4'b0000 : select gpadc_scan_pos_0 and gpadc_scan_neg_0</p> <p>4'b0001 : select gpadc_scan_pos_1 and gpadc_scan_neg_1</p> <p>4'b0010 : select gpadc_scan_pos_2 and gpadc_scan_neg_2</p> <p>4'b0011 : select gpadc_scan_pos_3 and gpadc_scan_neg_3</p> <p>4'b0100 : select gpadc_scan_pos_4 and gpadc_scan_neg_4</p> <p>4'b0101 : select gpadc_scan_pos_5 and gpadc_scan_neg_5</p> <p>4'b0110 : select gpadc_scan_pos_6 and gpadc_scan_neg_6</p> <p>4'b0111 : select gpadc_scan_pos_7 and gpadc_scan_neg_7</p> <p>4'b1000 : select gpadc_scan_pos_8 and gpadc_scan_neg_8</p> <p>4'b1001 : select gpadc_scan_pos_9 and gpadc_scan_neg_9</p> <p>4'b1010 : select gpadc_scan_pos_10 and gpadc_scan_neg_10</p> <p>4'b1011 : select gpadc_scan_pos_11 and gpadc_scan_neg_11</p>
20:18	gpadc_clk_div_ratio	r/w	3'h3	<p>analog 32M clock division ratio</p> <p>3'b000: div=1</p> <p>3'b001: div=4</p> <p>3'b010: div=8</p> <p>3'b011: div=12</p> <p>3'b100: div=16</p> <p>3'b101: div=20</p> <p>3'b110: div=24</p> <p>3'b111: div=32</p>
17	gpadc_clk_ana_inv	r/w	1'b0	analog clock 2M inverted
16:11	RSVD			
10	gpadc_lowv_det_en	r/w	1'b0	Low power supply detected enable
9	gpadc_vcm_hyst_sel	r/w	1'b0	pga vcm hysteresis select when vcm_sel_en is enabled
8	gpadc_vcm_sel_en	r/w	1'b0	pga vcm selected when lowv_det_en is enable
7:5	RSVD			

位	名称	权限	复位值	描述
4:2	gpadc_res_sel	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 mode 16KS/s) 3'b100 16bit 7.8125KS/s, OSR=256 (voice mode 8KS/s)
1	gpadc_cont_conv_en	r/w	1'b1	To enable continuous conversion 1'h0: one shot conversion 1'h1: continuous conversion
0	gpadc_cal_os_en	r/w	1'b0	offset calibration enable

#### 4.4.8 gpadc\_reg\_config2

地址: 0x4000f914

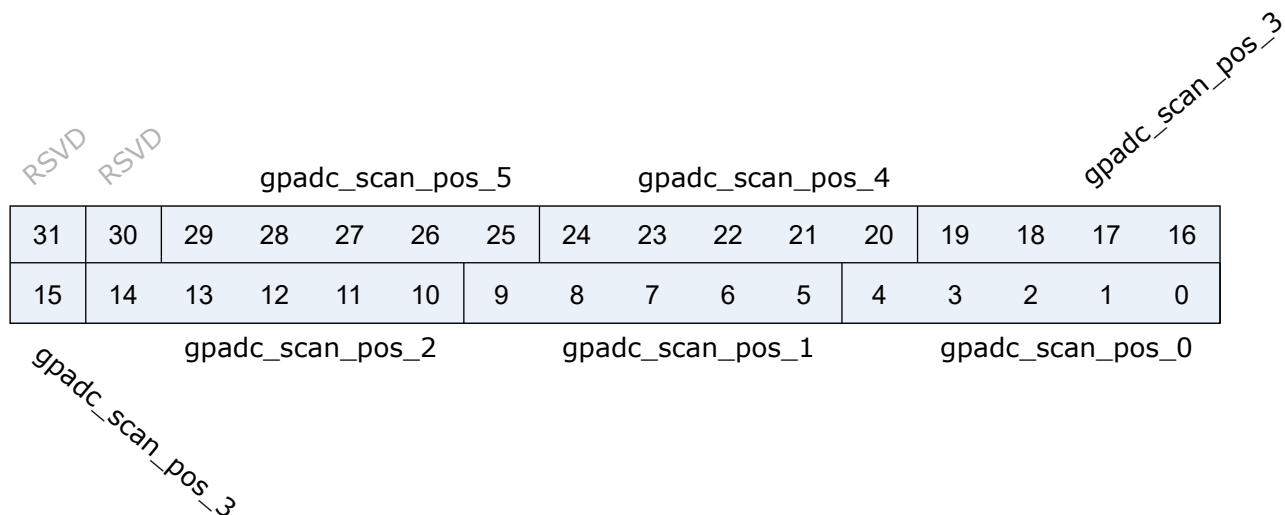


位	名称	权限	复位值	描述
31	gpadc_tsvbe_low	r/w	1'b0	tsen diode current
30:28	gpadc_dly_sel	r/w	3'h0	adc conversion speed
27:25	gpadc_pga1_gain	r/w	3'h0	3'h0: disable 3'h1: gain=1 3'h2: gain=2 3'h3: gain=4 3'h4: gain=8 3'h5: gain=16 3'h6: gain=32 3'h7: gain=32

位	名称	权限	复位值	描述
24:22	gpadc_pga2_gain	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	gpadc_test_sel	r/w	3'h0	select test point 0~7
18	gpadc_test_en	r/w	1'b0	Analog test enable.
17	gpadc_bias_sel	r/w	1'b0	adc analog portion low power mode select 1'h0: bandgap system 1'h1:aon bandgap
16:15	gpadc_chop_mode	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	gpadc_pga_vcmi_en	r/w	1'b0	enable pga input vcm bias
13	gpadc_pga_en	r/w	1'b0	1'h0: disable PGA 1'h1 enable PGA
12:9	gpadc_pga_os_cal	r/w	4'h8	pga offset calibration
8:7	gpadc_pga_vcm	r/w	2'h2	Audio PGA output common mode control 2'b00: cm=1.3V 2'b11: cm=1.4V 2'b11: cm=1.5V 2'b11: cm=1.6V
6	gpadc_ts_en	r/w	1'b0	1'h0: disable temperature sensor 1'h1: enable temperature sensor
5	gpadc_tsext_sel	r/w	1'b0	1'h0: internal diode mode 1'h1: external diode mode
4	gpadc_vbat_en	r/w	1'b0	1'h0: disable VBAT sensor 1'h1 enable VBAT sensor
3	gpadc_vref_sel	r/w	1'b0	ADC reference select 1'h0 3.2V 1'h1 2.0V
2	gpadc_diff_mode	r/w	1'b0	1'h0 single-ended 1'h1 differential
1:0	RSVD			

#### 4.4.9 gpadc\_reg\_scn\_pos1

地址: 0x4000f918



位	名称	权限	复位值	描述
31:30	RSVD			
29:25	gpadc_scan_pos_5	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
24:20	gpadc_scan_pos_4	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
19:15	gpadc_scan_pos_3	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
14:10	gpadc_scan_pos_2	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
9:5	gpadc_scan_pos_1	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
4:0	gpadc_scan_pos_0	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel

#### 4.4.10 gpadc\_reg\_scn\_pos2

地址: 0x4000f91c

RSVD		RSVD		gpadc_scan_pos_11				gpadc_scan_pos_10				gpadc_scan_pos_9			
31		30		29 28 27 26 25				24 23 22 21 20				19 18 17 16			
15		14		13 12 11 10				9 8 7 6 5				4 3 2 1 0			
gpadc_scan_pos_8				gpadc_scan_pos_7				gpadc_scan_pos_6				gpadc_scan_pos_9			

位	名称	权限	复位值	描述
31:30	RSVD			
29:25	gpadc_scan_pos_11	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
24:20	gpadc_scan_pos_10	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
19:15	gpadc_scan_pos_9	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
14:10	gpadc_scan_pos_8	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
9:5	gpadc_scan_pos_7	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel
4:0	gpadc_scan_pos_6	r/w	5'hf	definition is the same as adc_reg_cmd.adc_pos_sel

#### 4.4.11 gpadc\_reg\_scn\_neg1

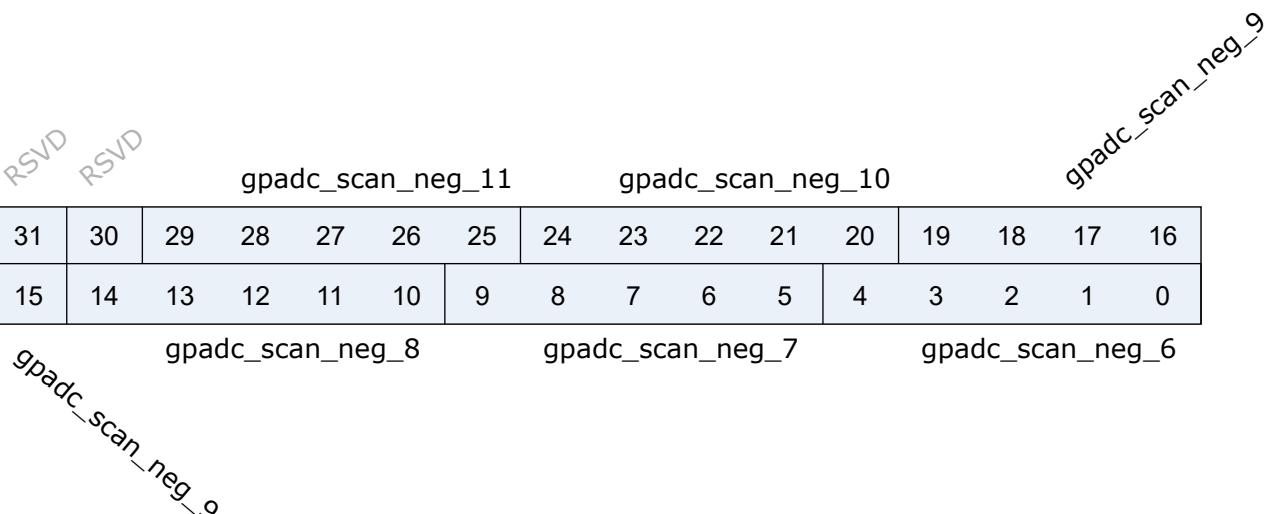
地址: 0x4000f920

RSVD		RSVD		gpadc_scan_neg_5				gpadc_scan_neg_4				gpadc_scan_neg_3			
31		30		29 28 27 26 25				24 23 22 21 20				19 18 17 16			
15		14		13 12 11 10				9 8 7 6 5				4 3 2 1 0			
gpadc_scan_neg_2				gpadc_scan_neg_1				gpadc_scan_neg_0				gpadc_scan_neg_3			

位	名称	权限	复位值	描述
31:30	RSVD			
29:25	gpadc_scan_neg_5	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
24:20	gpadc_scan_neg_4	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
19:15	gpadc_scan_neg_3	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
14:10	gpadc_scan_neg_2	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
9:5	gpadc_scan_neg_1	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
4:0	gpadc_scan_neg_0	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel

#### 4.4.12 gpadc\_reg\_scn\_neg2

地址: 0x4000f924



位	名称	权限	复位值	描述
31:30	RSVD			
29:25	gpadc_scan_neg_11	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
24:20	gpadc_scan_neg_10	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
19:15	gpadc_scan_neg_9	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
14:10	gpadc_scan_neg_8	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
9:5	gpadc_scan_neg_7	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel
4:0	gpadc_scan_neg_6	r/w	5'hf	definition is the same as adc_reg_cmd.adc_neg_sel



#### 4.4.13 gpadc\_reg\_status

地址: 0x4000f928

位	名称	权限	复位值	描述
31:16	gpadc_reserved	r/w	16'h0	
15:1	RSVD			
0	gpadc_data_rdy	r	1'b0	ADC final conversion data ready

#### 4.4.14 gpadc\_reg\_isr

地址: 0x4000f92c

位	名称	权限	复位值	描述
31:10	RSVD			
9	gpadc_pos_satur_mask	r/w	1'h0	write 1 mask
8	gpadc_neg_satur_mask	r/w	1'h0	write 1 mask
7:6	RSVD			
5	gpadc_pos_satur_clr	r/w	1'b0	Write 1 to clear flag

位	名称	权限	复位值	描述
4	gpadc_neg_satur_clr	r/w	1'b0	Write 1 to clear flag
3:2	RSVD			
1	gpadc_pos_satur	r	1'b0	ADC data positive side saturation interrupt flag
0	gpadc_neg_satur	r	1'b0	ADC data negative side saturation interrupt flag

#### 4.4.15 gpadc\_reg\_raw\_result

地址: 0x4000f934

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

位	名称	权限	复位值	描述
31:12	RSVD			
11:0	gpadc_raw_data	r	12'h0	ADC Raw data

#### 4.4.16 gpadc\_reg\_define

地址: 0x4000f938

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

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	gpadc_os_cal_data	r/w	16'h0	User defined or self calculated offset data 16-bit signed

## 5.1 简介

芯片内置一个 10bits 的数字模拟转换器 (DAC) ,FIFO 深度为 1, 支持 2 路 DAC 调制输出。可用于音频播放, 常规的模拟信号调制。

## 5.2 主要特点

- DAC 调制精度为 10bits
- DAC 的输入时钟可选为 32M 或者 Audio PLL
- 支持 DMA 从内存将数据搬运至 DAC 调制寄存器
- 支持双声道播放模式
- DAC 的输出引脚固定, ChannelA 固定为 GPIO11, ChannelB 固定为 GPIO17
- DAC 的参考电压可选择内部或者外部

## 5.3 功能描述

DAC 模块基本框图如图所示。

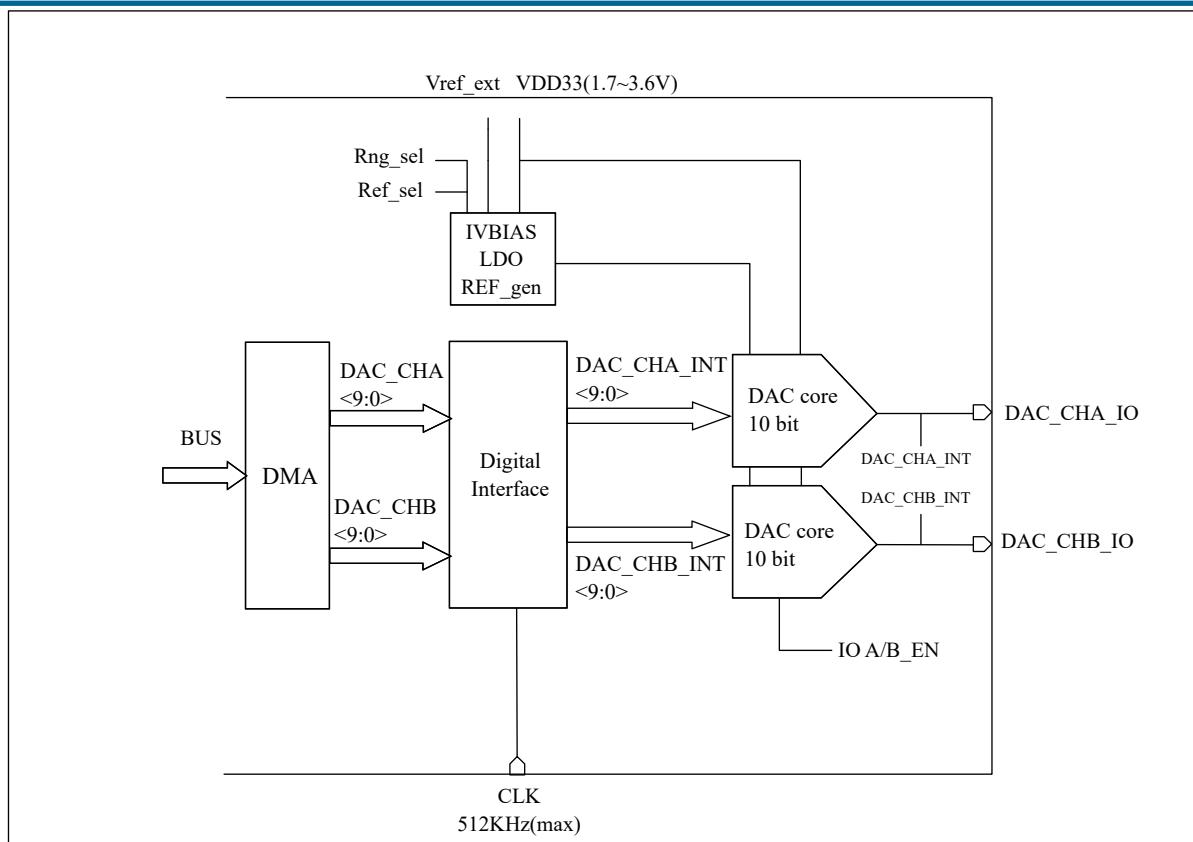


图 5.1: DAC 基本框图

DAC 模块包含两路 DAC 调制电路, 以及调制模拟信号相关的电源电路, 用户可以通过 Ref\_Sel 来选择 DAC 的参考电压是外部/内部, Ref\_Rng 来选择内部参考电压源。DAC 的调制数据可以由 CPU 直接写入 DAC 调制寄存器 (0x40000314 中的 GLB\_GPDAC\_A\_DATA、GLB\_GPDAC\_B\_DATA), 也可以由 DMA 搬运至 gpdac\_dma\_wdata (0x40002048) 寄存器。

#### DAC 的数据写入方式

CPU 直接写入 GLB\_GPDAC\_A\_DATA、GLB\_GPDAC\_B\_DATA 寄存器完成调制, 或者使用 DMA, 将需要调制的数据搬运到 gpdac\_dma\_wdata 中。

#### DMA 的搬运模式

gpdac\_dma\_wdata (0x40002048) 是一个 32BITS 的寄存器, 默认含义为, 32BITS 值按顺序全部调制在 ChannelA 引脚上, 也可配置为高 16 位默认是对应 Channel B 的模拟电压输出, 低 16 位对应 Channel A 的模拟电压输出。注意无论是 32/16 位调制, 都只有低 10 位有效, 因为 DAC 的最大调制精度为 10BITS。用户可以通过配置 gpdac\_dma\_format 寄存器来修改 DMA 搬运的高低字节含义。

gpdac\_dma\_format 为 0, 则 DMA 搬运进 gpdac\_dma\_wdata 的数据全部依次调制在 Channel A, 调制顺序为 {A0},{A1},{A2},...。gpdac\_dma\_format 为 1, 则 DMA 搬运进 gpdac\_dma\_wdata 的数据高 16 位调制在 Channel B, 低 16 位调制在 Channel A。调制顺序为 {B0,A0},{B1,A1},{B2,A2},...。这样的特性在立体声播放中非常有作用。gpdac\_dma\_format 为 2, 则 DMA 搬运进 gpdac\_dma\_wdata 的数据全部调制在 Channel A, 但调制的顺序为 {A1,A0},{A3,A2},{A5,A4},...。

## DAC 外部参考电压选择

用户可以通过配置 `gpdac_ref_sel` (0x40000308[8]) 来配置选择外部参考电压还是内部参考电压。

如果选择内部参考电压，配置如下表所示。

表 5.1: 内部参考电压

<code>gpdac_a_rng</code>	<code>gpdac_ref_sel</code>	输出范围
00	0	0.2-1
01/10	0	0.225-1.425
11	0	0.2-1.8

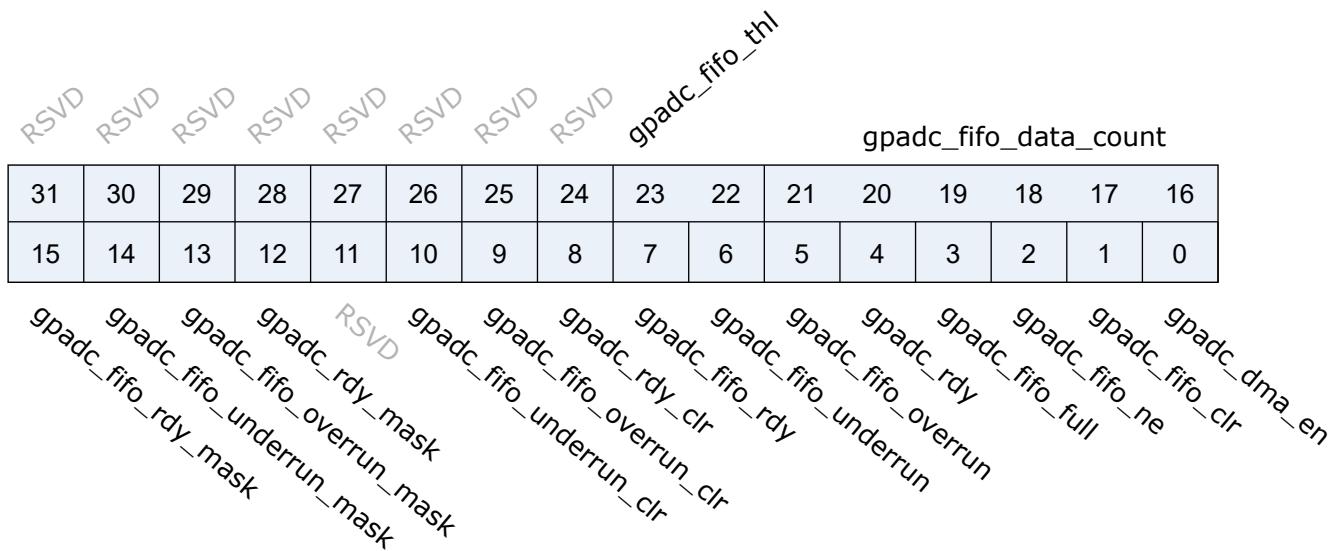
如果选择外部参考电压，请将外部电压连入固定的 GPIO7。

## 5.4 寄存器描述

名称	描述
<code>gpadc_config</code>	GPADC configuration
<code>gpadc_dma_rdata</code>	GPADC DMA read data
<code>gpdac_config</code>	GPDAC configuration
<code>gpdac_dma_config</code>	GPDAC dma configuration
<code>gpdac_dma_wdata</code>	GPDAC dma write data
<code>gpdac_ctrl</code>	GPDAC control
<code>gpdac_actrl</code>	GPDAC channelA control
<code>gpdac_bctrl</code>	GPDAC channelB control
<code>gpdac_ctrl</code>	GPDAC control
<code>gpdac_actrl</code>	GPDAC channelA control
<code>gpdac_bctrl</code>	GPDAC channelB control

### 5.4.1 gpadc\_config

地址: 0x40002000



位	名称	权限	复位值	描述
31:24	RSVD			
23:22	gpadc_fifo_thl	r/w	2'd0	fifo threshold 2'b00: 1 data 2'b01: 4 data 2'b10: 8 data 2'b11: 16 data
21:16	gpadc_fifo_data_count	r	6'd0	fifo data number
15	gpadc_fifo_rdy_mask	r/w	1'b1	write 1 mask
14	gpadc_fifo_underrun_-mask	r/w	1'b0	write 1 mask
13	gpadc_fifo_overrun_-mask	r/w	1'b0	write 1 mask
12	gpadc_rdy_mask	r/w	1'b0	write 1 mask
11	RSVD			
10	gpadc_fifo_underrun_clr	w1c	1'b0	Write 1 to clear flag
9	gpadc_fifo_overrun_clr	w1c	1'b0	Write 1 to clear flag
8	gpadc_rdy_clr	w1c	1'b0	Write 1 to clear flag
7	gpadc_fifo_rdy	r	1'b0	FIFO ready interrupt flag
6	gpadc_fifo_underrun	r	1'b0	FIFO underrun interrupt flag
5	gpadc_fifo_overrun	r	1'b0	FIFO overrun interrupt flag

位	名称	权限	复位值	描述
4	gpadc_rdy	r	1'b0	Conversion data ready interrupt flag
3	gpadc_fifo_full	r	1'b0	FIFO full flag
2	gpadc_fifo_ne	r	1'b0	FIFO not empty flag
1	gpadc_fifo_clr	w1c	1'b0	FIFO clear signal
0	gpadc_dma_en	r/w	1'b0	GPADC DMA enable

#### 5.4.2 gpadc\_dma\_rdata

地址: 0x40002004

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

位	名称	权限	复位值	描述
31:26	RSVD			
25:0	gpadc_dma_rdata	r	26'd0	GPADC final conversion result stored in the FIFO

#### 5.4.3 gpdac\_config

地址: 0x40002040

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

位	名称	权限	复位值	描述
31:24	RSVD			
23:20	gpdac_ch_b_sel	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	gpdac_ch_a_sel	r/w	0	Channel A Source Select 0: Reg 1: DMA 2: DMA + Filter 3: Sin Gen
15:11	RSVD			
10:8	gpdac_mode	r/w	0	0:32k, 1:16k, 3:8k, 4:512k(for DMA only)
7:6	RSVD			
5:4	dsm_mode	r/w	0	0:bypass, 1:dsm order=1, 2: dsm order=2
3:2	RSVD			
1	gpdac_en2	r/w	0	GPDAC enable 2 (for B channel)
0	gpdac_en	r/w	0	GPDAC enable

#### 5.4.4 gpdac\_dma\_config

地址: 0x40002044

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

位	名称	权限	复位值	描述
31:6	RSVD			
5:4	gpdac_dma_format	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	gpdac_dma_tx_en	r/w	0	GPDAC DMA TX enable

#### 5.4.5 gpdac\_dma\_wdata

地址: 0x400002048

gpdac\_dma\_wdata

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

gpdac\_dma\_wdata

位	名称	权限	复位值	描述
31:0	gpdac_dma_wdata	w	x	GPDAC DMA TX data

#### 5.4.6 gpdac\_ctrl

地址: 0x40000308

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

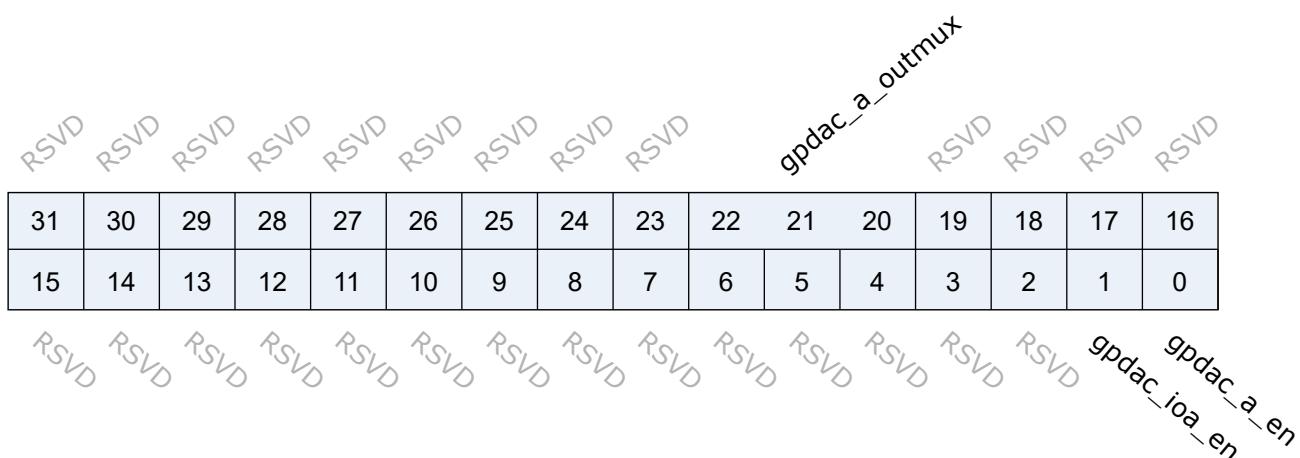
  

RSVD	gpdac_ref_sel	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	gpdacb_rstn_ana	gpdaca_rstn_ana						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

位	名称	权限	复位值	描述
31:9	RSVD			
8	gpdac_ref_sel	r/w	1'h0	Reference select 1'h0 Internal reference 1'h1 External reference
7:2	RSVD			
1	gpdacb_rstn_ana	r/w	1'h1	Soft reset for DAC channel B, active low
0	gpdaca_rstn_ana	r/w	1'h1	Soft reset for DAC channel A, active low

#### 5.4.7 gpdac\_actrl

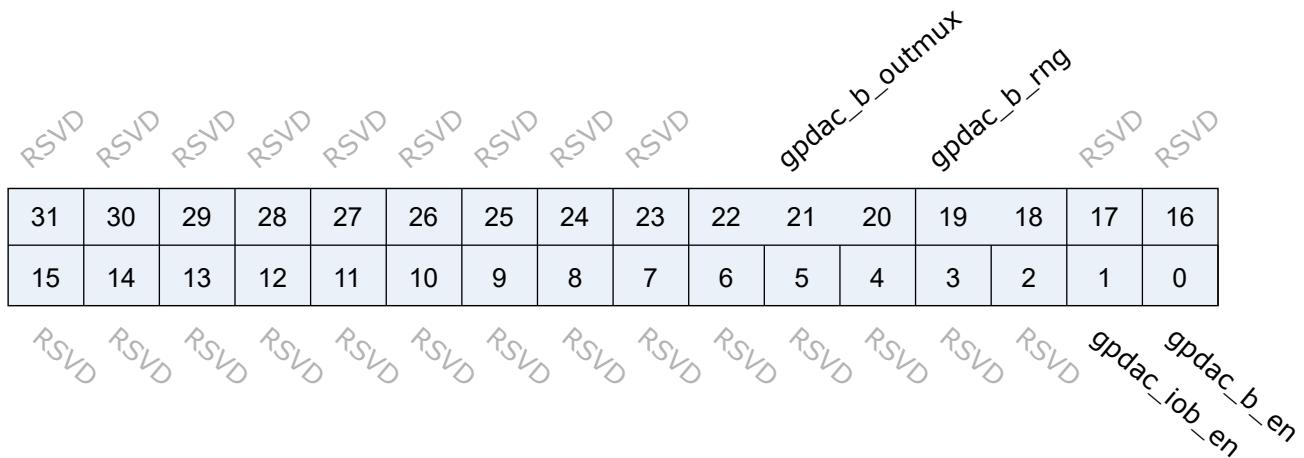
地址: 0x4000030c



位	名称	权限	复位值	描述
31:23	RSVD			
22:20	gpdac_a_outmux	r/w	3'h0	
19:18	gpdac_a_rng	r/w	2'h3	Output voltage range control with internal/external reference
17:2	RSVD			
1	gpdac_ioa_en	r/w	1'h0	Channel A conversion output to pad enable 1'h0 Disable channel A conversion result to GPIO 1'h1 Enable channel A conversion result to GPIO
0	gpdac_a_en	r/w	1'h0	Channel A enable/disable signal 1'h0 Disable channel A conversion. 1'h1 Enable channel A conversion

### 5.4.8 gpdac\_bctrl

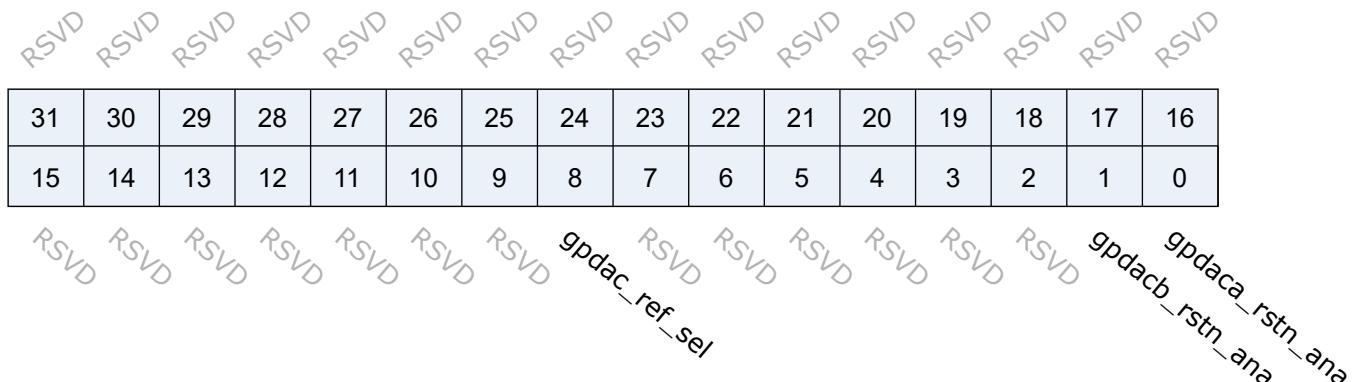
地址: 0x40000310



位	名称	权限	复位值	描述
31:23	RSVD			
22:20	gpdac_b_outmux	r/w	3'h0	
19:18	gpdac_b_rng	r/w	2'h3	
17:2	RSVD			
1	gpdac_iob_en	r/w	1'h0	channel B conversion output to pad enable 1'h0 Disable channel B conversion result to GPIO 1'h1 Enable channel B conversion result to GPIO
0	gpdac_b_en	r/w	1'h0	channel B enable/disable signal 1'h0 Disable channel B conversion. 1'h1 Enable channel B conversion

### 5.4.9 gpdac\_ctrl

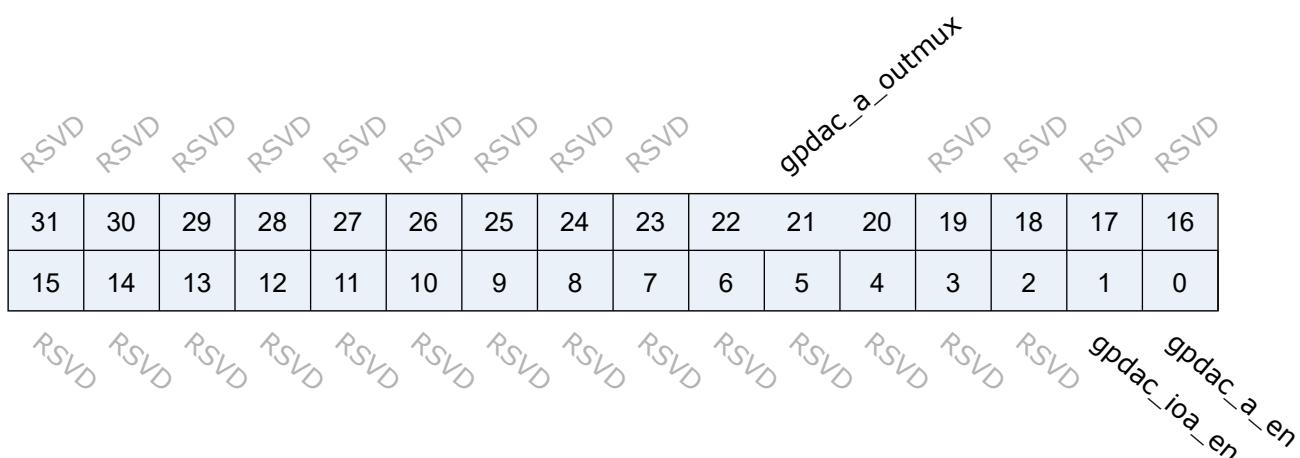
Address: 0x40000308



位	名称	权限	复位值	描述
31:9	RSVD			
8	gpdac_ref_sel	r/w	1'h0	Reference select 1'h0 Internal reference 1'h1 External reference
7:2	RSVD			
1	gpdacb_rstn_ana	r/w	1'h1	Soft reset for DAC channel B, active low
0	gpdaca_rstn_ana	r/w	1'h1	Soft reset for DAC channel A, active low

#### 5.4.10 gpdac\_actrl

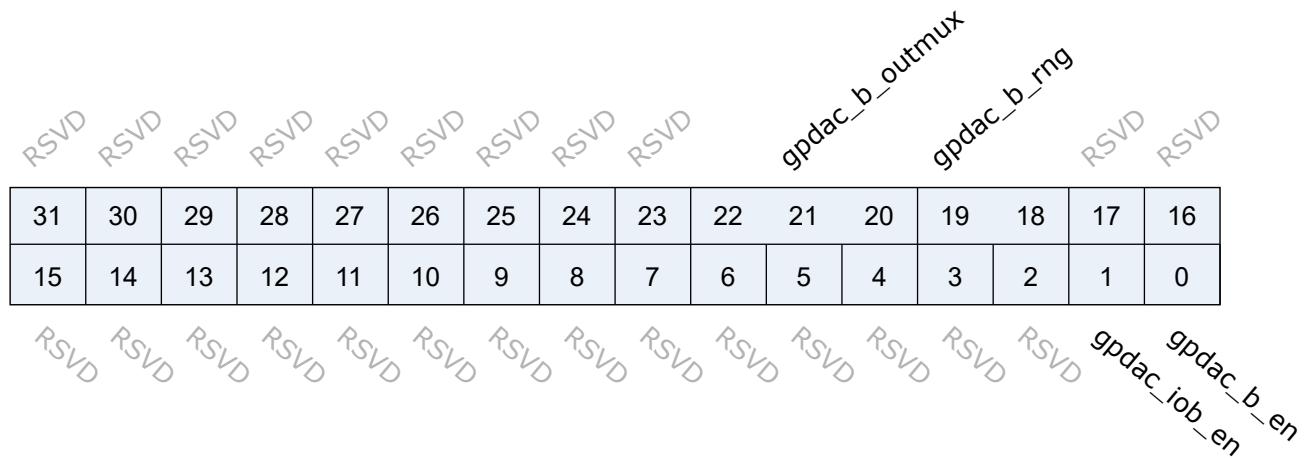
Address: 0x4000030c



位	名称	权限	复位值	描述
31:23	RSVD			
22:20	gpdac_a_outmux	r/w	3'h0	
19:18	gpdac_a_rng	r/w	2'h3	Output voltage range control with internal/external reference
17:2	RSVD			
1	gpdac_ioa_en	r/w	1'h0	Channel A conversion output to pad enable 1'h0 Disable channel A conversion result to GPIO 1'h1 Enable channel A conversion result to GPIO
0	gpdac_a_en	r/w	1'h0	Channel A enable/disable signal 1'h0 Disable channel A conversion. 1'h1 Enable channel A conversion

### 5.4.11 gpdac\_bctrl

Address: 0x40000310



位	名称	权限	复位值	描述
31:23	RSVD			
22:20	gpdac_b_outmux	r/w	3'h0	
19:18	gpdac_b_rng	r/w	2'h3	
17:2	RSVD			
1	gpdac_iob_en	r/w	1'h0	channel B conversion output to pad enable 1'h0 Disable channel B conversion result to GPIO 1'h1 Enable channel B conversion result to GPIO
0	gpdac_b_en	r/w	1'h0	channel B enable/disable signal 1'h0 Disable channel B conversion. 1'h1 Enable channel B conversion

## 6.1 简介

DMA(Direct Memory Access) 是一种内存存取技术，可以独立地直接读写系统内存，而不需处理器介入处理。在同等程度的处理器负担下，DMA 是一种快速的数据传送方式。DMA 控制器有 8 组独立专用通道，管理外围设备和内存之间的数据传输以提高总线效率。主要有四种类型传输包括内存至内存、内存至外设、外设至内存和外设至外设。并支持 LLI 链接列表功能。使用上由软件配置传输数据大小、数据源地址和目标地址。

## 6.2 主要特征

- 8 组独立专用通道
- 独立控制来源与目标存取宽度 (单字节、双字节、四字节)
- 每个通道独立作为读写缓存
- 每个通道可被独立的外设硬件触发或是软件触发
- 支持外设包括 UART、I2C、SPI、ADC、I2S、DAC
- 四种流程控制
  - DMA 流程控制，来源内存、目标内存
  - DMA 流程控制，来源内存、目标外设
  - DMA 流程控制，来源外设、目标内存
  - DMA 流程控制，来源外设、目标外设
- 支持 LLI 链表功能，提高 DMA 效率

## 6.3 功能描述

### 6.3.1 工作原理

当一个设备试图通过总线直接向另一个设备传输数据(一般是大批量的数据)时, 它会先向 CPU 发送 DMA 请求信号。外设通过 DMA 向 CPU 提出接管总线控制权的总线请求, CPU 收到该信号后, 在当前的总线周期结束后, 会按 DMA 信号的优先级和提出 DMA 请求的先后顺序响应 DMA 信号。CPU 对某个设备接口响应 DMA 请求时, 会让出总线控制权。于是在 DMA 控制器的管理下, 外设和存储器直接进行数据交换, 而不需 CPU 干预。数据传送完毕后, 设备会向 CPU 发送 DMA 结束信号, 交还总线控制权。

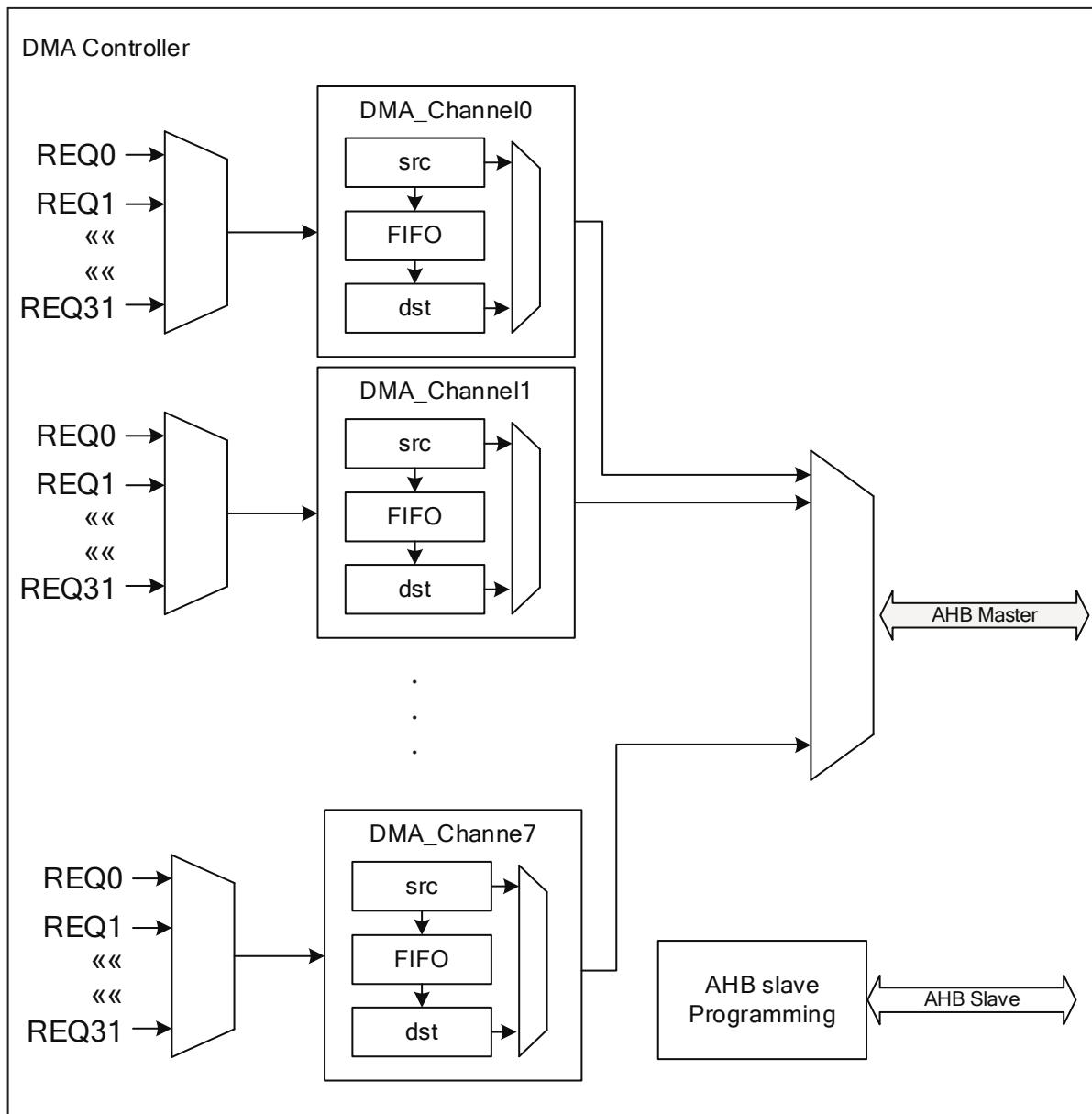


图 6.1: DMA 框图

DMA 包含一组 AHB Master 接口和一组 AHB Slave 接口。AHB Master 接口根据当前配置需求通过系统总线主动存取内存或是外设, 做为数据搬移的端口。AHB Slave 接口作为配置 DMA 的接口, 只支持 32-bit 存取。

### 6.3.2 DMA 通道配置

DMA 共支持 8 路通道，各通道之间互不干涉，可以同时运行，下面是 DMA 通道 x 的配置过程：

1. 在 DMA\_C0SrcAddr 寄存器中设置 32-bit 来源地址
2. 在 DMA\_C0DstAddr 寄存器中设置 32-bit 目标地址
3. 地址自动累加，可通过配置 DMA\_C0Control 寄存器中 SI(来源)、DI(目标) 设定是否开启地址自动累加模式，设置为 1 时，开启地址自动累加模式
4. 设置传输数据宽度，可通过配置 DMA\_C0Control 寄存器中 STW(来源)、DTW(目标) 位，宽度选项有单字节、双字节、四字节
5. Burst 型态，可通过配置 DMA\_C0Control 寄存器中 SBS(来源)、DBS(目标) 位来设置，配置选项有 Single、INCR4、INCR8、INCR16
6. 需要特别注意的是所配置的组合，单笔 burst 不能超过 16 字节
7. 设置数据传输长度的范围为：0-4095

### 6.3.3 外设支持

可通过配置 SrcPeripheral(来源) 和 DstPeripheral(目标) 来决定当前 DMA 配合的外设，关系为 0-5 : UART / 6-9 : I2C / 10-13 : SPI / 18-21 : I2S / 22: ADC / 23: DAC

#### UART 使用 DMA 传输数据

UART 发送数据包，使用 DMA 方式能大量减轻 CPU 处理的时间，使其 CPU 资源不被大量浪费，尤其在 UART 收发大量数据包（如高频率收发指令）时具有明显优势。

以 UART0 传输为例，配置过程如下：

1. 将寄存器 DMA\_C0Config[SRCPH] 位的值设置为 1，即将 Source peripheral 设置为 UART\_TX
2. 将寄存器 DMA\_C0Config[DSTPH] 位的值设置为 0，即将 Destination peripheral 设置为 UART\_RX

#### I2C 使用 DMA 传输数据

配置如下：

1. 将寄存器 DMA\_C0Config[SRCPH] 位的值设置为 7，即将 Source peripheral 设置为 I2C\_TX
2. 将寄存器 DMA\_C0Config[DSTPH] 位的值设置为 6，即将 Destination peripheral 设置为 I2C\_RX

#### SPI 使用 DMA 传输数据

配置如下：

1. 将寄存器 DMA\_C0Config[SRCPH] 位的值设置为 11，即将 Source peripheral 设置为 SPI\_TX
2. 将寄存器 DMA\_C0Config[DSTPH] 位的值设置为 10，即将 Destination peripheral 设置为 SPI\_RX

## ADC 使用 DMA 传输数据

配置如下：

1. 将寄存器 DMA\_C0Config[SRCPH] 位的值设置为 22，即将 Source peripheral 设置为 GPADC

## DAC 使用 DMA 传输数据

配置如下：

1. 将寄存器 DMA\_C0Config[SRCPH] 位的值设置为 23，即将 Source peripheral 设置为 GPDAC

### 6.3.4 链表模式

DMA 支持链表工作模式。在进行一次 DMA 读或写操作时，可以向下一条链表中填写数据，当完成当前链表的数据传输后，通过读取 DMA\_COLLI 寄存器的数值获取下一条链表的起始地址，直接传输下一条链表中的数据。保证 DMA 传输过程中连续不间断的工作，提高 CPU 和 DMA 的效率。

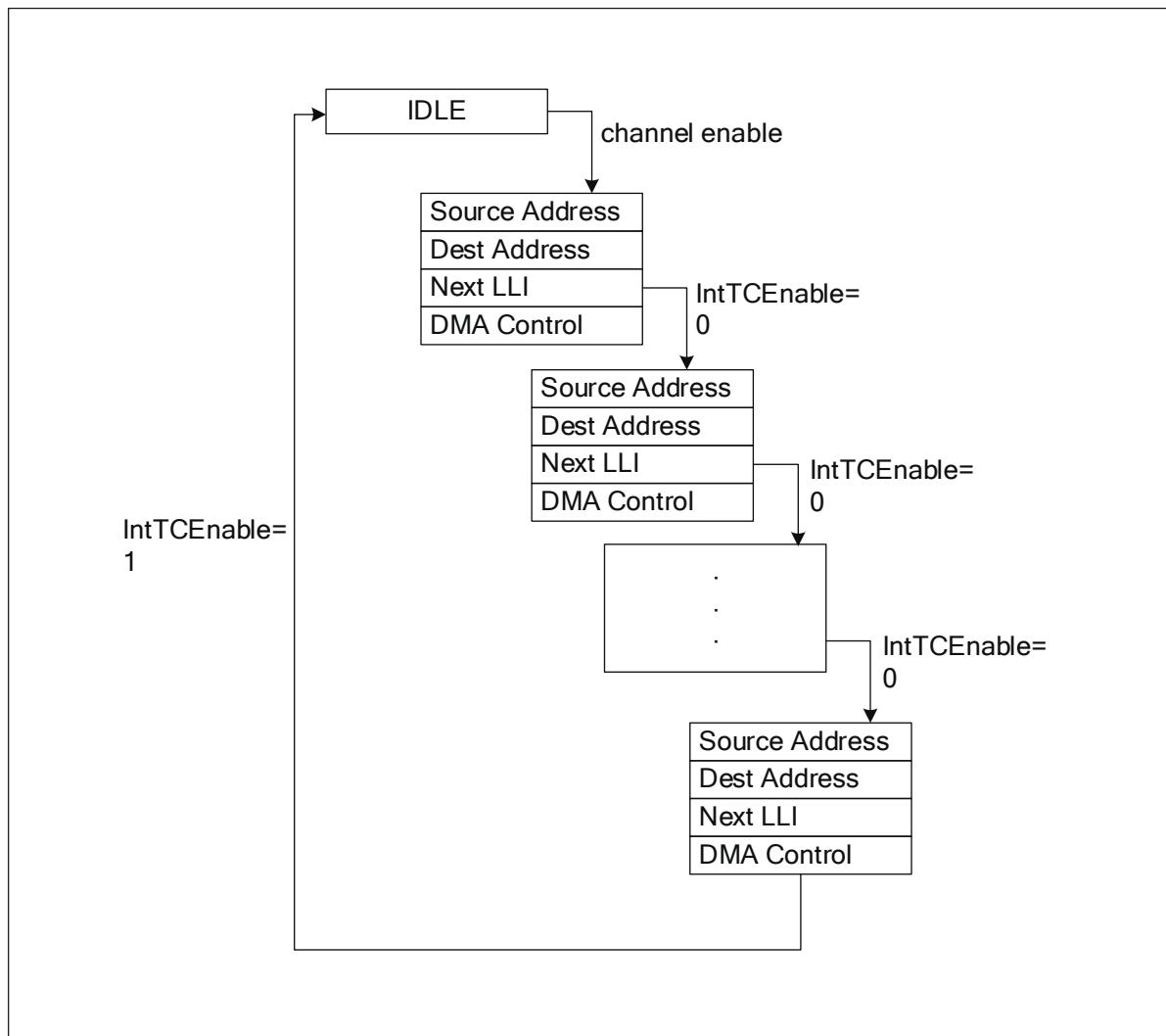


图 6.2: LLI 框架

### 6.3.5 DMA 中断

- DMA\_INT\_TCOMPLETED
  - 数据传输完成中断，当一次数据传输完毕后，会进入此中断
- DMA\_INT\_ERR
  - 数据传输出错中断，当数据传输过程中出现错误时，会进入此中断

## 6.4 传输模式

### 6.4.1 内存到内存

这个模式启动后，DMA 会根据设定好的搬移数量 (TransferSize)，将数据从来源地址搬到目标地址，传输完毕后 DMA 控制器会自动回到空闲状态，等待下一次的搬运。

具体配置流程如下：

1. 将寄存器 DMA\_C0SrcAddr 的值设置为来源的内存地址
2. 将寄存器 DMA\_C0DstAddr 的值设置为目标的内存地址
3. 选择传输模式，将寄存器 DMA\_C0Config[FLOWCTRL] 位的值设置为 0，即选择 memory-to-memory 模式
4. 设置 DMA\_C0Control 寄存器中对应的位的数值：DI、SI 位设置为 1，开启地址自动累加模式，DTW、STW 位分别设置目标和来源的传输宽度，DBS、SBS 位分别设置目标和来源的 burst 型态，
5. 选择合适的通道，使能 DMA，完成数据传输

### 6.4.2 内存到外设

在这种工作模式下，DMA 会根据设定好的搬移数量 (TransferSize)，把数据从来源端搬至内部缓存，当缓存空间不够时自动暂停，待有足够的缓存空间时继续，直到设定的搬移数量达到。另外一方面当目标外设请求触发会将目标配置 burst 到目标地址，直到达到设定搬移数量完成自动回到空闲状态，等待下一次启动。

具体配置流程如下：

1. 将寄存器 DMA\_C0SrcAddr 的值设置为来源的内存地址
2. 将寄存器 DMA\_C0DstAddr 的值设置为目标的外设地址
3. 选择传输模式，将寄存器 DMA\_C0Config[FLOWCTRL] 位的值设置为 1，即选择 Memory-to-peripheral 模式
4. 设置 DMA\_C0Control 寄存器中对应的位的数值：SI 位设置为 1，开启地址自动累加模式，DI 位设置为 0，禁用地址自动累加模式，DTW、STW 位分别设置来源和目标的传输宽度，DBS、SBS 位分别设置来源和目标的 burst 型态，
5. 选择合适的通道，使能 DMA，完成数据传输

### 6.4.3 外设到内存

在这种工作模式下，当来源外设请求触发时将来源配置 burst 到缓存，直到设定的搬移数量达到停止。另外一方面，当内部缓存足够一次目标 burst 数量时，DMA 会自动将缓存的内容搬到目标地址直到达到设定搬移数量完成自动回到空闲状态，等待下一次启动。

具体配置流程如下：

1. 将寄存器 DMA\_C0SrcAddr 的值设置为来源的外设地址
2. 将寄存器 DMA\_C0DstAddr 的值设置为目标的内存地址
3. 选择传输模式，将寄存器 DMA\_C0Config[FLOWCTRL] 位的值设置为 2，即选择 Peripheral-to-memory 模式
4. 设置 DMA\_C0Control 寄存器中对应的位的数值：DI、SI 位设置为 1，开启地址自动累加模式，DTW、STW 位分别设置来源和目标的传输宽度，DBS、SBS 位分别设置来源和目标的 burst 型态，
5. 选择合适的通道，使能 DMA，完成数据传输

### 6.4.4 外设到外设

在这种工作模式下，当来源外设请求触发时将来源配置 burst 到缓存，直到设定的搬移数量达到停止。另外一方面，当内部缓存足够一次目标 burst 数量时，DMA 会自动将缓存的内容搬到目标地址直到达到设定搬移数量完成自动回到空闲状态，等待下一次启动。

具体配置流程如下：

1. 将寄存器 DMA\_C0SrcAddr 的值设置为来源的外设地址
2. 将寄存器 DMA\_C0DstAddr 的值设置为目标的外设地址
3. 选择传输模式，将寄存器 DMA\_C0Config[FLOWCTRL] 位的值设置为 3，即选择 Peripheral-to-Peripheral 模式
4. 设置 DMA\_C0Control 寄存器中对应的位的数值：DI、SI 位设置为 0，禁用地址自动累加模式，STW、DTW 位分别设置来源和目标的传输宽度，SBS、DBS 位分别设置来源和目标的 burst 型态
5. 选择合适的通道，使能 DMA，完成数据传输

## 6.5 寄存器描述

名称	描述
DMA_IntStatus	Interrupt status
DMA_IntTCStatus	Interrupt terminal count request status
DMA_IntTCClear	Terminal count request clear
DMA_IntErrorStatus	Interrupt error status

名称	描述
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
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 contro
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

名称	描述
DMA_C3Config	Channel DMA configuration
DMA_C4SrcAddr	Channel DMA source address
DMA_C4DstAddr	Channel DMA Destination address
DMA_C4LLI	Channel DMA link list
DMA_C4Control	Channel DMA bus control
DMA_C4Config	Channel DMA configuration
DMA_C5SrcAddr	Channel DMA source address
DMA_C5DstAddr	Channel DMA Destination address
DMA_C5LLI	Channel DMA link list
DMA_C5Control	Channel DMA bus control
DMA_C5Config	Channel DMA configuration
DMA_C6SrcAddr	Channel DMA source address
DMA_C6DstAddr	Channel DMA Destination address
DMA_C6LLI	Channel DMA link list
DMA_C6Control	Channel DMA bus control
DMA_C6Config	Channel DMA configuration
DMA_C7SrcAddr	Channel DMA source address
DMA_C7DstAddr	Channel DMA Destination address
DMA_C7LLI	Channel DMA link list
DMA_C7Control	Channel DMA bus control
DMA_C7Config	Channel DMA configuration

### 6.5.1 DMA\_IntStatus

地址: 0x4000c000

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

IntStatus

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	IntStatus	r	0	Status of the DMA interrupts after masking

### 6.5.2 DMA\_IntTCStatus

地址: 0x4000c004

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

IntTCStatus

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	IntTCStatus	r	0	Interrupt terminal count request status

### 6.5.3 DMA\_IntTCClear

地址: 0x4000c008

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

RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD IntTCClear

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	IntTCClear	w	0	Terminal count request clear

#### 6.5.4 DMA\_IntErrorStatus

地址: 0x4000c00c

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

RSVD RSVD RSVD RSVD RSVD RSVD RSVD IntErrorStatus

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	IntErrorStatus	r	0	Interrupt error status

#### 6.5.5 DMA\_IntErrClr

地址: 0x4000c010

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

RSVD RSVD RSVD RSVD RSVD RSVD RSVD IntErrClr

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	IntErrClr	w	0	Interrupt error clear

### 6.5.6 DMA\_RawIntTCStatus

地址: 0x4000c014

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

RawIntTCStatus

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	RawIntTCStatus	r	0	Status of the terminal count interrupt prior to masking

### 6.5.7 DMA\_RawIntErrorStatus

地址: 0x4000c018

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

RawIntErrorStatus

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	RawIntErrorStatus	r	0	Status of the error interrupt prior to masking

### 6.5.8 DMA\_EnbldChns

地址: 0x4000c01c

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

EnabledChannels

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	EnabledChannels	r	0	Channel enable status

### 6.5.9 DMA\_SoftBReq

地址: 0x4000c020

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

SoftBReq

位	名称	权限	复位值	描述
31:0	SoftBReq	r/w	0	Software burst request

### 6.5.10 DMA\_SoftSReq

地址: 0x4000c024

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

SoftSReq

位	名称	权限	复位值	描述
31:0	SoftSReq	r/w	0	Software single request

### 6.5.11 DMA\_SoftLBReq

地址: 0x4000c028

SoftLBReq

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

SoftLBReq

位	名称	权限	复位值	描述
31:0	SoftLBReq	r/w	0	Software last burst request

### 6.5.12 DMA\_SoftLSReq

地址: 0x4000c02c

SoftLSReq

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

SoftLSReq

位	名称	权限	复位值	描述
31:0	SoftLSReq	r/w	0	Software last single request

### 6.5.13 DMA\_Config

地址: 0x4000c030

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

RSVD M E

位	名称	权限	复位值	描述
31:2	RSVD			
1	M	r/w	0	AHB Master endianness configuration: 0 = little-endian, 1 = big-endian
0	E	r/w	0	SMDMA Enable.

### 6.5.14 DMA\_Sync

地址: 0x4000c034

DMA\_Sync

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

DMA\_Sync

位	名称	权限	复位值	描述
31:0	DMA_Sync	r/w	0	DMA synchronization logic for DMA request signals: 0 = enable, 1 = disable

### 6.5.15 DMA\_C0SrcAddr

地址: 0x4000c100

SrcAddr

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

SrcAddr

位	名称	权限	复位值	描述
31:0	SrcAddr	r/w	0	DMA source address

### **6.5.16 DMA\_C0DstAddr**

地址: 0x4000c104

DstAddr

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

DstAddr

位	名称	权限	复位值	描述
31:0	DstAddr	r/w	0	DMA Destination address

### **6.5.17 DMA\_C0LLI**

地址: 0x4000c108

LLI

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

LLI

位	名称	权限	复位值	描述
31:0	LLI	r/w	0	First linked list item. Bits [1:0] must be 0.

### **6.5.18 DMA\_C0Control**

地址: 0x4000c10c

1	Prot	DI	SI	S_LargerD fix_cnt	DWidth	RSVD	SWidth	dst_add_mode	DBSize
31	30	29	28	27	26	25	24	23	22
15	14	13	12	11	10	9	8	7	6
									5
								4	4
								3	3
								2	2
								1	1
								0	0

TransferSize

DBSize

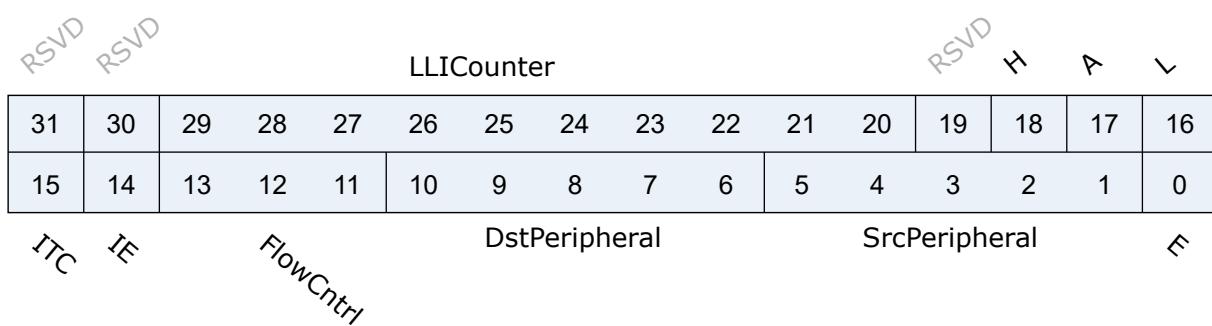
dst\_min\_mode

SBSIZE

位	名称	权限	复位值	描述
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	SLargerD	r/w	0	In Memory-to-memory mode, Set this bit high when Src data size is larger than Dst.
24:23	fix_cnt	r/w	2'd0	Only effect when dst_min_mode = 1 Destination transfer cnt = (total src byte cnt - (fix_-cnt«DWidth))«DWidth
22:21	DWidth	r/w	2'b10	Destination transfer width: 8/16/32
20	RSVD			
19:18	SWidth	r/w	2'b10	Source transfer width: 8/16/32
17	dst_add_mode	r/w	1'b0	Add mode : issue remain destination traffic
16:15	DBSize	r/w	2'b01	Destination burst size: 1/4/8/16
14	dst_min_mode	r/w	1'b0	Minus mode : Not issue all destination traffic
13:12	SBSIZE	r/w	2'b01	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSIZE*Swidth should <= 16B
11:0	TransferSize	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

地址: 0x4000c110



位	名称	权限	复位值	描述
31:30	RSVD			
29:20	LLICounter	r	0	LLI counter. Increased 1 each LLI run. Cleared 0 when config Control.
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.
13:11	FlowCntrl	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	DstPeripheral	r/w	0	Destination peripheral. [23:22] GPADC [21:18] I2S [17:14] PDM [13:10] SSP [ 9: 6] I2C [ 5: 0] UART
5:1	SrcPeripheral	r/w	0	Source peripheral.
0	E	r/w	0	Channel enable.

### 6.5.20 DMA\_C1SrcAddr

地址: 0x4000c200

SrcAddr

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

SrcAddr

位	名称	权限	复位值	描述
31:0	SrcAddr	r/w	0	

### 6.5.21 DMA\_C1DstAddr

地址: 0x4000c204

DstAddr

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

DstAddr

位	名称	权限	复位值	描述
31:0	DstAddr	r/w	0	

### 6.5.22 DMA\_C1LLI

地址: 0x4000c208

LLI

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

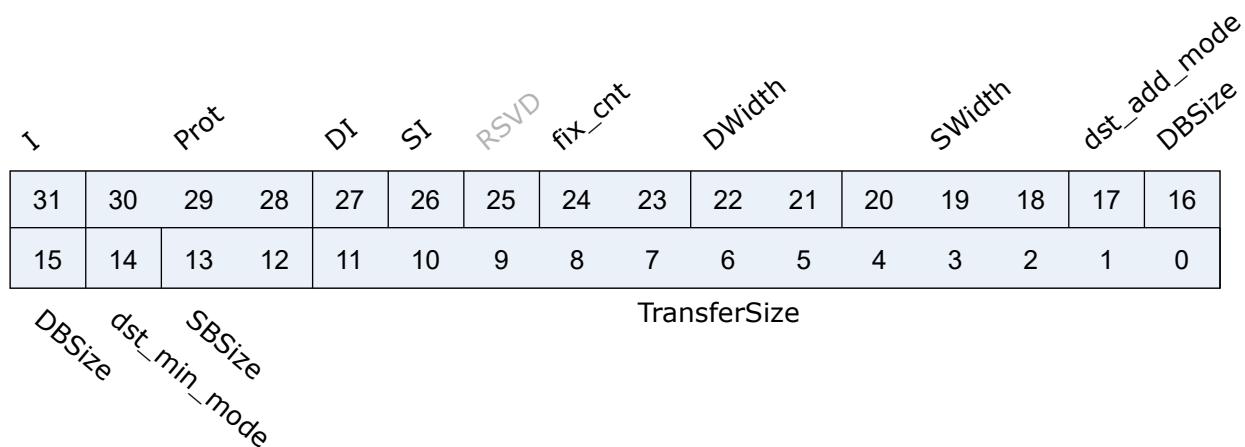
LLI

RSVD RSVD

位	名称	权限	复位值	描述
31:2	LLI	r/w	0	
1:0	RSVD			

### 6.5.23 DMA\_C1Control

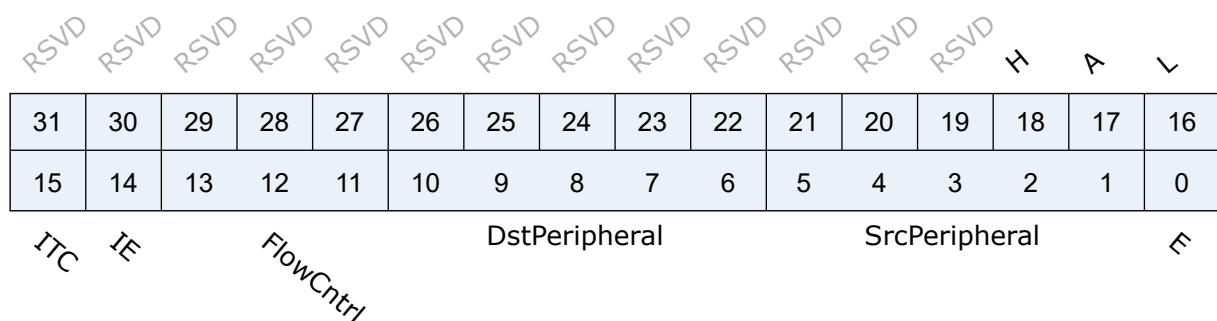
地址: 0x4000c20c



位	名称	权限	复位值	描述
31	I	r/w	0	
30:28	Prot	r/w	0	
27	DI	r/w	1	
26	SI	r/w	1	
25	RSVD			
24:23	fix_cnt	r/w	2'd0	
22:24	RSVD			
23:21	DWidth	r/w	3'b010	
20:18	SWidth	r/w	3'b010	
17	dst_add_mode	r/w	1'b0	
16:15	DBSize	r/w	3'b001	
14	dst_min_mode	r/w	1'b0	
13:12	SBSIZE	r/w	3'b001	
11:0	TransferSize	r/w	0	

### 6.5.24 DMA\_C1Config

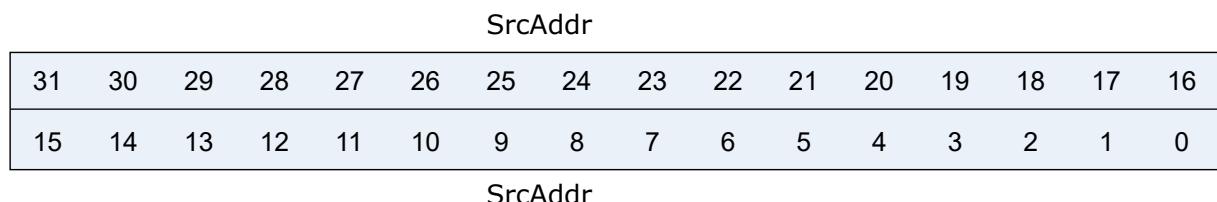
地址: 0x4000c210



位	名称	权限	复位值	描述
31:19	RSVD			
18	H	r/w	0	
17	A	r	0	
16	L	r/w	0	
15	ITC	r/w	0	
14	IE	r/w	0	
13:11	FlowCntrl	r/w	0	
10:6	DstPeripheral	r/w	0	
5:1	SrcPeripheral	r/w	0	
0	E	r/w	0	

### 6.5.25 DMA\_C2SrcAddr

地址: 0x4000c300



位	名称	权限	复位值	描述
31:0	SrcAddr	r/w	0	

### 6.5.26 DMA\_C2DstAddr

地址: 0x4000c304

DstAddr

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

DstAddr

位	名称	权限	复位值	描述
31:0	DstAddr	r/w	0	

### 6.5.27 DMA\_C2LLI

地址: 0x4000c308

LLI

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

LLI

RSVD RSVD

位	名称	权限	复位值	描述
31:2	LLI	r/w	0	
1:0	RSVD			

### 6.5.28 DMA\_C2Control

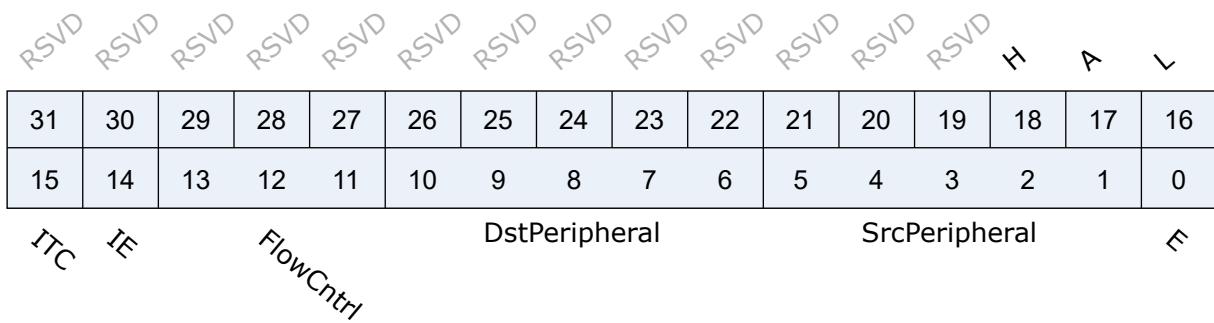
地址: 0x4000c30c

1	Prot	DI	SI	RSVD	fix_cnt	DWidth	SWidth	dst_add_mode	DBSize
31	30	29	28	27	26	25	24	23	22
15	14	13	12	11	10	9	8	7	6
									TransferSize
									DBSize dst_min_mode SBSIZE

位	名称	权限	复位值	描述
31	I	r/w	0	
30:28	Prot	r/w	0	
27	DI	r/w	1	
26	SI	r/w	1	
25	RSVD			
24:23	fix_cnt	r/w	2'd0	
22:24	RSVD			
23:21	DWidth	r/w	3'b010	
20:18	SWidth	r/w	3'b010	
17	dst_add_mode	r/w	1'b0	
16:15	DBSize	r/w	3'b001	
14	dst_min_mode	r/w	1'b0	
13:12	SBSIZE	r/w	3'b001	
11:0	TransferSize	r/w	0	

### 6.5.29 DMA\_C2Config

地址: 0x4000c310



位	名称	权限	复位值	描述
31:19	RSVD			
18	H	r/w	0	
17	A	r	0	
16	L	r/w	0	
15	ITC	r/w	0	

位	名称	权限	复位值	描述
14	IE	r/w	0	
13:11	FlowCntrl	r/w	0	
10:6	DstPeripheral	r/w	0	
5:1	SrcPeripheral	r/w	0	
0	E	r/w	0	

### 6.5.30 DMA\_C3SrcAddr

地址: 0x4000c400

SrcAddr

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

SrcAddr

位	名称	权限	复位值	描述
31:0	SrcAddr	r/w	0	

### 6.5.31 DMA\_C3DstAddr

地址: 0x4000c404

DstAddr

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

DstAddr

位	名称	权限	复位值	描述
31:0	DstAddr	r/w	0	

### 6.5.32 DMA\_C3LLI

地址: 0x4000c408

LLI																
位	名称			权限	复位值			描述								
31:2	LLI			r/w	0			RSVD								
1:0	RSVD															

### 6.5.33 DMA\_C3Control

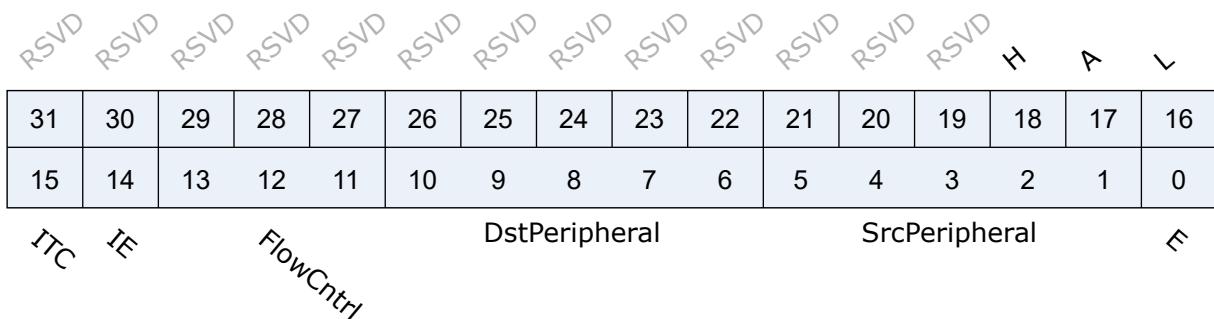
地址: 0x4000c40c

1	Prot				DI	SI	RSVD	fix_cnt	DWidth	SWidth	dst_add_mode	DBSize				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TransferSize																
DBSize	dst_min_mode															
位	名称			权限	复位值			描述								
31	I			r/w	0											
30:28	Prot			r/w	0											
27	DI			r/w	1											
26	SI			r/w	1											
25	RSVD															
24:23	fix_cnt			r/w	2'd0											
22:24	RSVD															

位	名称	权限	复位值	描述
23:21	DWidth	r/w	3'b010	
20:18	SWidth	r/w	3'b010	
17	dst_add_mode	r/w	1'b0	
16:15	DBSize	r/w	3'b001	
14	dst_min_mode	r/w	1'b0	
13:12	SBSIZE	r/w	3'b001	
11:0	TransferSize	r/w	0	

### 6.5.34 DMA\_C3Config

地址: 0x4000c410



位	名称	权限	复位值	描述
31:19	RSVD			
18	H	r/w	0	
17	A	r	0	
16	L	r/w	0	
15	ITC	r/w	0	
14	IE	r/w	0	
13:11	FlowCntrl	r/w	0	
10:6	DstPeripheral	r/w	0	
5:1	SrcPeripheral	r/w	0	
0	E	r/w	0	

### 6.5.35 DMA\_C4SrcAddr

地址: 0x4000c500

SrcAddr

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

SrcAddr

位	名称	权限	复位值	描述
31:0	SrcAddr	r/w	0	

### 6.5.36 DMA\_C4DstAddr

地址: 0x4000c504

DstAddr

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

DstAddr

位	名称	权限	复位值	描述
31:0	DstAddr	r/w	0	

### 6.5.37 DMA\_C4LLI

地址: 0x4000c508

LLI

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

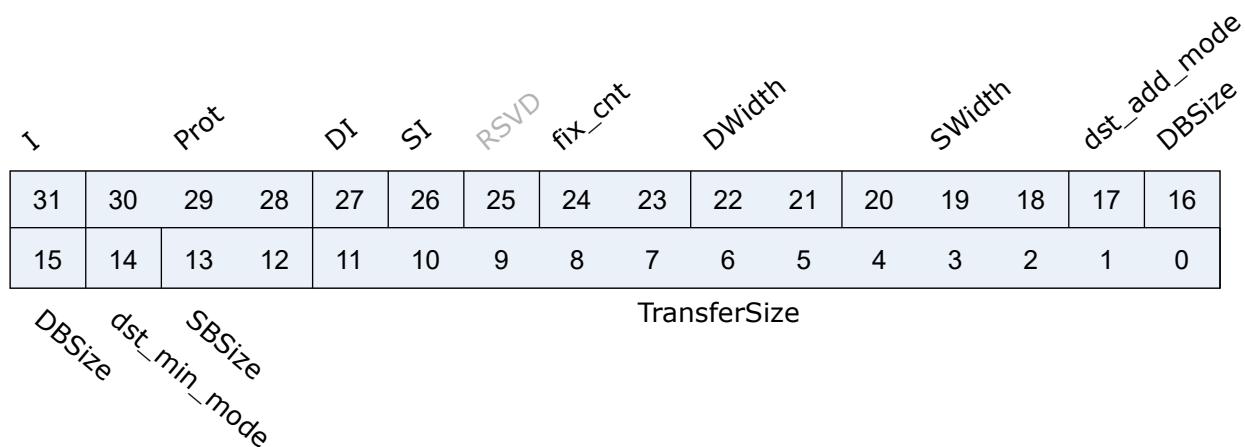
LLI

RSVD RSVD

位	名称	权限	复位值	描述
31:2	LLI	r/w	0	
1:0	RSVD			

### 6.5.38 DMA\_C4Control

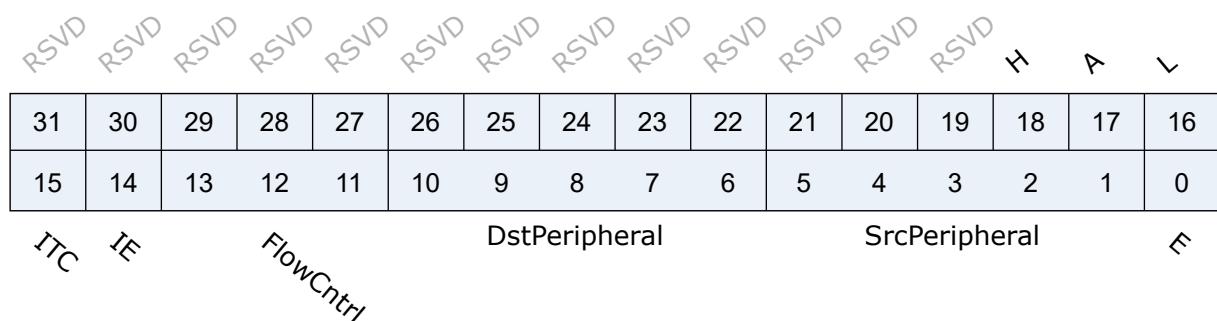
地址: 0x4000c50c



位	名称	权限	复位值	描述
31	I	r/w	0	
30:28	Prot	r/w	0	
27	DI	r/w	1	
26	SI	r/w	1	
25	RSVD			
24:23	fix_cnt	r/w	2'd0	
22:24	RSVD			
23:21	DWidth	r/w	3'b010	
20:18	SWidth	r/w	3'b010	
17	dst_add_mode	r/w	1'b0	
16:15	DBSize	r/w	3'b001	
14	dst_min_mode	r/w	1'b0	
13:12	SBSIZE	r/w	3'b001	
11:0	TransferSize	r/w	0	

### 6.5.39 DMA\_C4Config

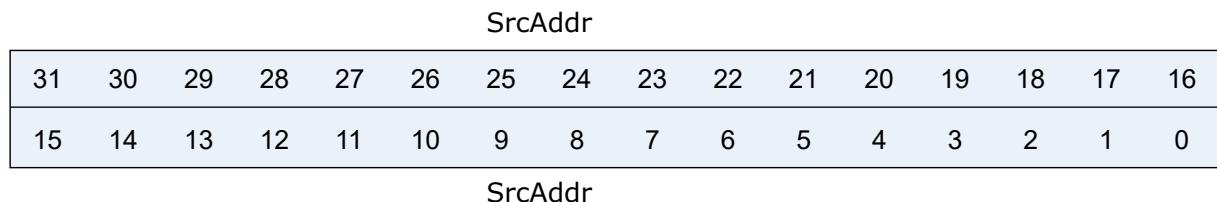
地址: 0x4000c510



位	名称	权限	复位值	描述
31:19	RSVD			
18	H	r/w	0	
17	A	r	0	
16	L	r/w	0	
15	ITC	r/w	0	
14	IE	r/w	0	
13:11	FlowCntrl	r/w	0	
10:6	DstPeripheral	r/w	0	
5:1	SrcPeripheral	r/w	0	
0	E	r/w	0	

### 6.5.40 DMA\_C5SrcAddr

地址: 0x4000c600



位	名称	权限	复位值	描述
31:0	SrcAddr	r/w	0	

### 6.5.41 DMA\_C5DstAddr

地址: 0x4000c604

DstAddr

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

DstAddr

位	名称	权限	复位值	描述
31:0	DstAddr	r/w	0	

### 6.5.42 DMA\_C5LLI

地址: 0x4000c608

LLI

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

LLI

RSVD RSVD

位	名称	权限	复位值	描述
31:2	LLI	r/w	0	
1:0	RSVD			

### 6.5.43 DMA\_C5Control

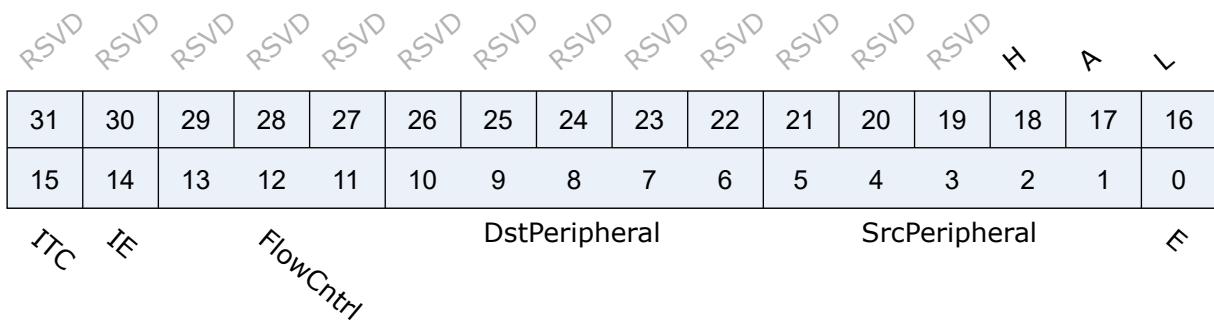
地址: 0x4000c60c

1	Prot	DI	SI	RSVD	fix_cnt	DWidth	SWidth	dst_add_mode	DBSize
31	30	29	28	27	26	25	24	23	22
15	14	13	12	11	10	9	8	7	6
									TransferSize
									DBSize dst_min_mode SBSIZE

位	名称	权限	复位值	描述
31	I	r/w	0	
30:28	Prot	r/w	0	
27	DI	r/w	1	
26	SI	r/w	1	
25	RSVD			
24:23	fix_cnt	r/w	2'd0	
22:24	RSVD			
23:21	DWidth	r/w	3'b010	
20:18	SWidth	r/w	3'b010	
17	dst_add_mode	r/w	1'b0	
16:15	DBSize	r/w	3'b001	
14	dst_min_mode	r/w	1'b0	
13:12	SBSIZE	r/w	3'b001	
11:0	TransferSize	r/w	0	

#### 6.5.44 DMA\_C5Config

地址: 0x4000c610



位	名称	权限	复位值	描述
31:19	RSVD			
18	H	r/w	0	
17	A	r	0	
16	L	r/w	0	
15	ITC	r/w	0	

位	名称	权限	复位值	描述
14	IE	r/w	0	
13:11	FlowCntrl	r/w	0	
10:6	DstPeripheral	r/w	0	
5:1	SrcPeripheral	r/w	0	
0	E	r/w	0	

### 6.5.45 DMA\_C6SrcAddr

地址: 0x4000c700

SrcAddr

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

SrcAddr

位	名称	权限	复位值	描述
31:0	SrcAddr	r/w	0	

### 6.5.46 DMA\_C6DstAddr

地址: 0x4000c704

DstAddr

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

DstAddr

位	名称	权限	复位值	描述
31:0	DstAddr	r/w	0	

### 6.5.47 DMA\_C6LLI

地址: 0x4000c708

LLI															
位	名称			权限	复位值		描述								
31:2	LLI			r/w	0										
1:0	RSVD														

### 6.5.48 DMA\_C6Control

地址: 0x4000c70c

1	Prot	DI	SI	RSVD	fix_cnt	DWidth	SWidth	dst_add_mode	DBSize
31	30 29 28	27	26	25	24 23	22 21	20 19 18	17	16
15	14	13 12	11	10	9 8 7	6 5	4 3 2	1	0
TransferSize									
DBSize dst_min_mode SBSIZE									

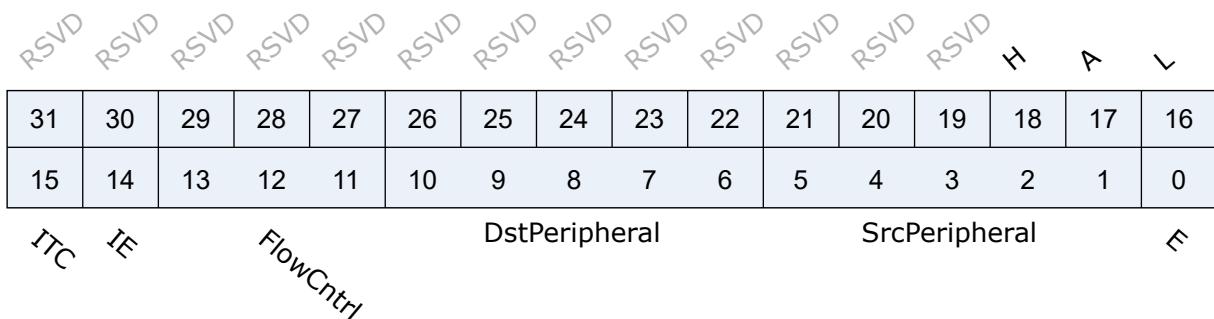
  

位	名称	权限	复位值	描述
31	I	r/w	0	
30:28	Prot	r/w	0	
27	DI	r/w	1	
26	SI	r/w	1	
25	RSVD			
24:23	fix_cnt	r/w	2'd0	
22:24	RSVD			

位	名称	权限	复位值	描述
23:21	DWidth	r/w	3'b010	
20:18	SWidth	r/w	3'b010	
17	dst_add_mode	r/w	1'b0	
16:15	DBSize	r/w	3'b001	
14	dst_min_mode	r/w	1'b0	
13:12	SBSIZE	r/w	3'b001	
11:0	TransferSize	r/w	0	

### 6.5.49 DMA\_C6Config

地址: 0x4000c710



位	名称	权限	复位值	描述
31:19	RSVD			
18	H	r/w	0	
17	A	r	0	
16	L	r/w	0	
15	ITC	r/w	0	
14	IE	r/w	0	
13:11	FlowCntrl	r/w	0	
10:6	DstPeripheral	r/w	0	
5:1	SrcPeripheral	r/w	0	
0	E	r/w	0	

### 6.5.50 DMA\_C7SrcAddr

地址: 0x4000c800

SrcAddr

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

SrcAddr

位	名称	权限	复位值	描述
31:0	SrcAddr	r/w	0	

### 6.5.51 DMA\_C7DstAddr

地址: 0x4000c804

DstAddr

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

DstAddr

位	名称	权限	复位值	描述
31:0	DstAddr	r/w	0	

### 6.5.52 DMA\_C7LLI

地址: 0x4000c808

LLI

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

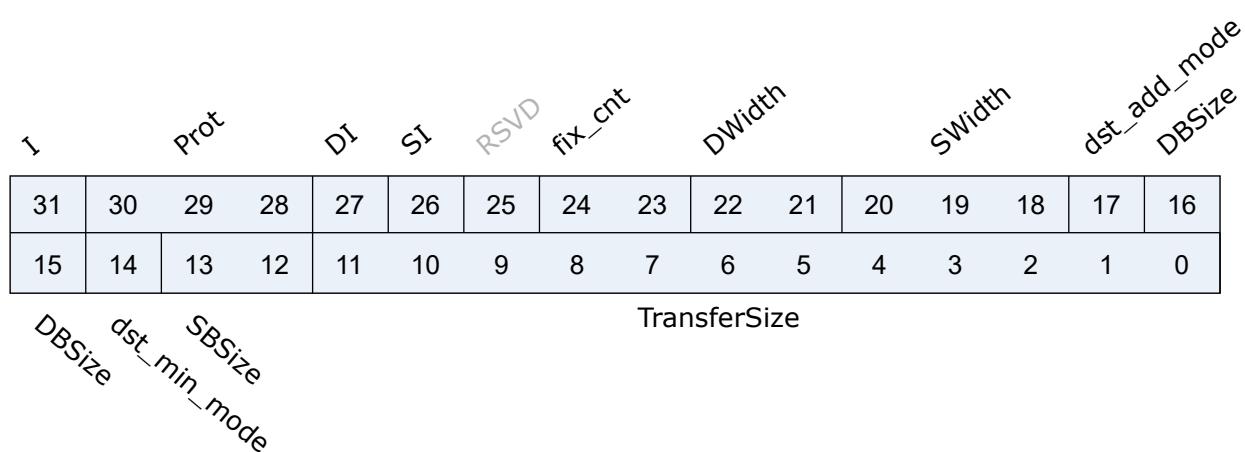
LLI

RSVD RSVD

位	名称	权限	复位值	描述
31:2	LLI	r/w	0	
1:0	RSVD			

### 6.5.53 DMA\_C7Control

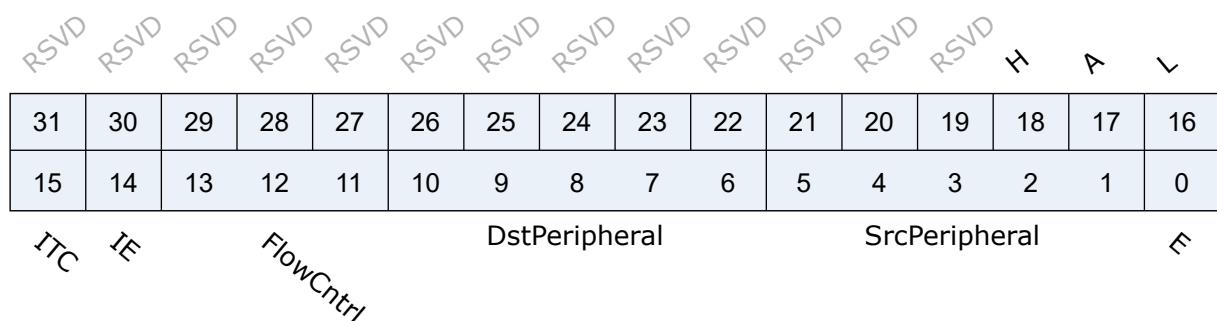
地址: 0x4000c80c



位	名称	权限	复位值	描述
31	I	r/w	0	
30:28	Prot	r/w	0	
27	DI	r/w	1	
26	SI	r/w	1	
25	RSVD			
24:23	fix_cnt	r/w	2'd0	
22:24	RSVD			
23:21	DWidth	r/w	3'b010	
20:18	SWidth	r/w	3'b010	
17	dst_add_mode	r/w	1'b0	
16:15	DBSize	r/w	3'b001	
14	dst_min_mode	r/w	1'b0	
13:12	SBSIZE	r/w	3'b001	
11:0	TransferSize	r/w	0	

### 6.5.54 DMA\_C7Config

地址: 0x4000c810



位	名称	权限	复位值	描述
31:19	RSVD			
18	H	r/w	0	
17	A	r	0	
16	L	r/w	0	
15	ITC	r/w	0	
14	IE	r/w	0	
13:11	FlowCntrl	r/w	0	
10:6	DstPeripheral	r/w	0	
5:1	SrcPeripheral	r/w	0	
0	E	r/w	0	

## 7.1 简介

L1 Cache Controller 是位于处理器外部, 用于管理 Flash/pSRAM 上代码或者数据缓冲, 提高 CPU 访问 Flash/pSRAM 速度的单元模块。架构如下图:

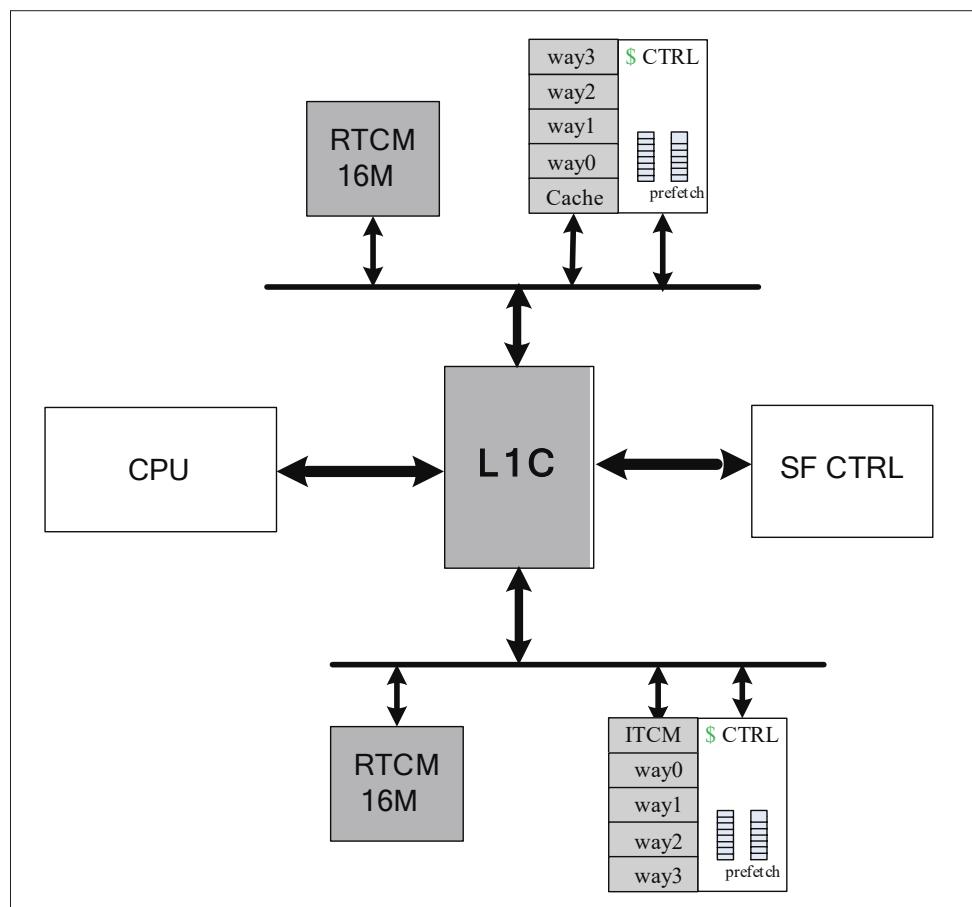


图 7.1: L1C 架构

L1C 是集成在处理器和 Flash/pSRAM 中间的高速单元, 因为处理器的速度非常快, 当处理器访问 Flash/pSRAM 时需

要等待的时间较长,等待的时间即代表浪费的时间,浪费的时间越少代表效能越高,L1C 缓存可以在处理器与 Flash/pSRAM 中间当作润滑的角色,让处理器效率提升。

## 7.2 主要特征

- 4 路相联映射缓存 (4-way Set-Associative mapping)
- 可调变的缓存空间大小
- 与 TCM 地址空间连续,可以方便的配置 L1C 空间为 TCM 空间
- 支持缓存效能统计

## 7.3 功能描述

### 7.3.1 TCM 与 Cache RAM 资源相互变换

为了增加内存使用效率,支持将 Cache 的 16K RAM 全部或者部分调整为 TCM 空间,这样方便用户根据实际情况,调整内存使用方式和效率。Cache 最大可设置为 16K,分为 4 条 way,每条 way 是 4K,调整的单位是 1 条 way,也就是 4K。ITCM 的默认大小是 16K。通过 WayDisable 的设定可以灵活的调整 Cache 和 ITCM 的实际空间大小。

表 7.1: WayDisable 的设定

WayDisable	Cache	ITCM
none	16K	0K
one way	12K	4K
two way	8K	8K
three way	4K	12K
four way	0K	16K

### 7.3.2 缓存 (Cache)

缓存每条 line buffer 单位为 32 字节,采用 4 路相联映射缓存,应用架构如下:

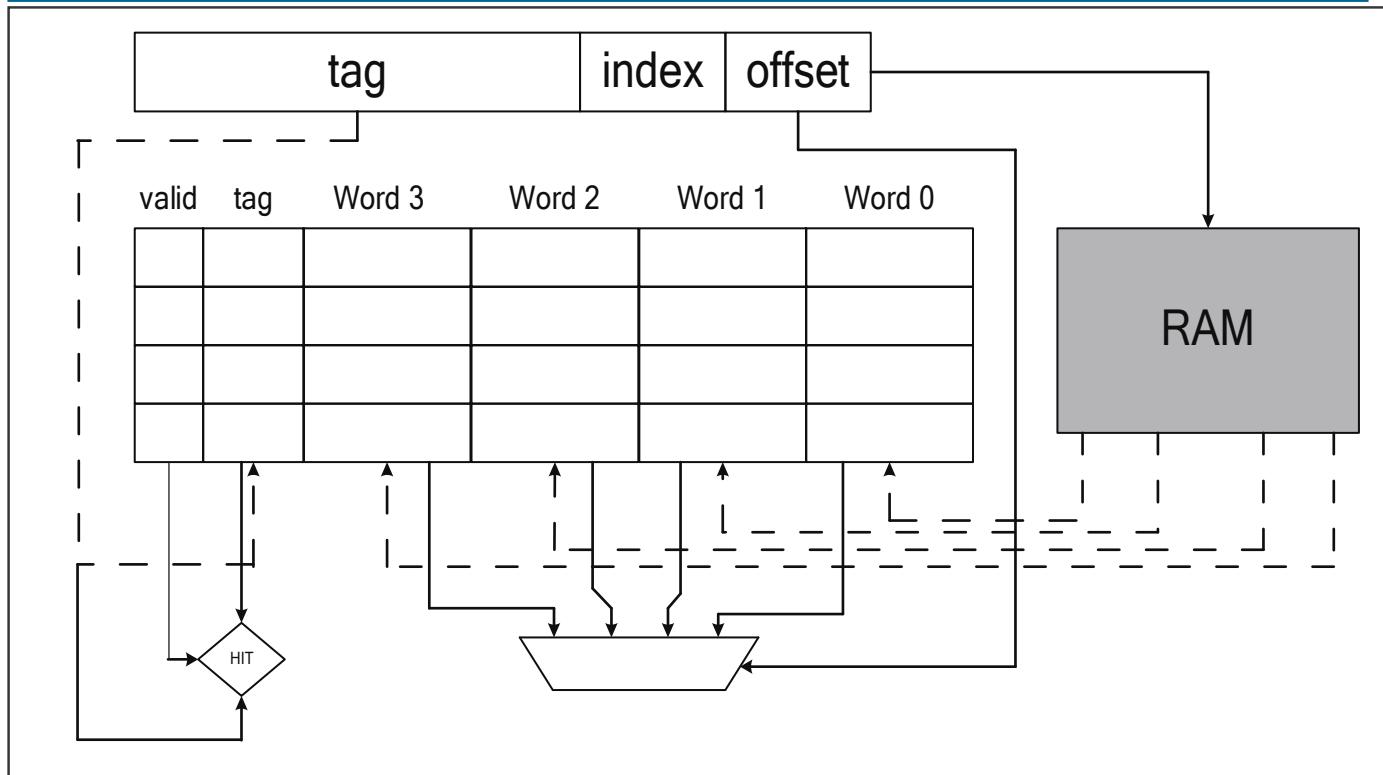


图 7.2: 缓存架构

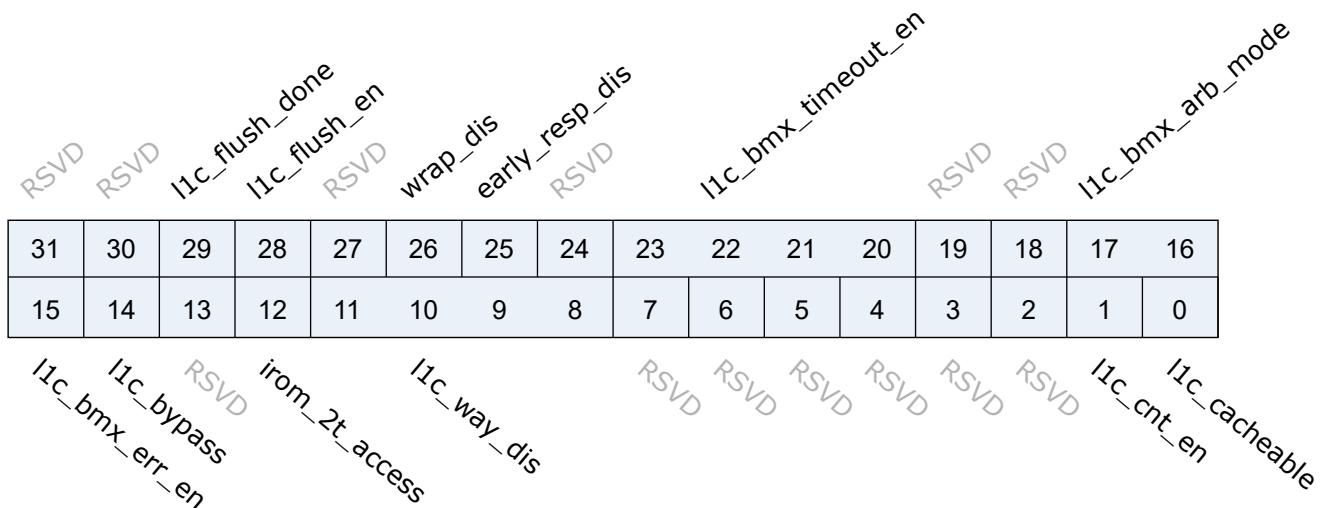
每组相联映射缓存包含两个部分，第一个是 **tag**，包含了有效值和地址映像关系，第二部分是数据储存。当处理器存取缓存时，缓存处理器会比对地址和 **tag** 的关系，当地址比对成功时代表可以直接从缓存取数据，反之缓存处理器则会通过 **AHB Master** 抓取相关的数据并同时放进缓存和响应处理器的数据。当大部分的数据都可以在 **tag** 中比对成功时，可以大幅度缩短处理器的等待时间，增加使用效率。

## 7.4 寄存器描述

名称	描述
l1c_config	L1C configuration
hit_cnt_lsb	Low 32-bit hit counter
hit_cnt_msb	High 32-bit hit counter
miss_cnt	Miss counter

### 7.4.1 I1c\_config

地址: 0x40009000



位	名称	权限	复位值	描述
31:30	RSVD			
29	I1c_flush_done	r	0	
28	I1c_flush_en	r/w	0	flush dirty line
27	RSVD			
26	wrap_dis	r/w	1	
25	early_resp_dis	r/w	1	
24	RSVD			
23:20	I1c_bmx_timeout_en	r/w	0	Bus timeout enable: detect slave no reaponse in 1024 cycles
19:18	RSVD			
17:16	I1c_bmx_arb_mode	r/w	0	[1:0] 0:fix, 2:round-robin, 3:random
15	I1c_err_en	r/w	0	Bus error response enable
14	I1c_bypass	r/w	0	bypass cache ; reset cache
13	RSVD			
12	irom_2t_access	r/w	0	Set 1 for ROM 2T access if CPU freq >72MHz
11:8	I1c_way_dis	r/w	4'b1111	Disable part of cache ways & used as ITCM
7:2	RSVD			
1	I1c_cnt_en	r/w	0	
0	I1c_cacheable	r/w	0	

#### 7.4.2 hit\_cnt\_lsb

地址: 0x40009004

hit\_cnt\_lsb

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

hit\_cnt\_lsb

位	名称	权限	复位值	描述
31:0	hit_cnt_lsb	r	0	

#### 7.4.3 hit\_cnt\_msb

地址: 0x40009008

hit\_cnt\_msb

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

hit\_cnt\_msb

位	名称	权限	复位值	描述
31:0	hit_cnt_msb	r	0	total hit count = hit_cnt_msb*232 + hit_cnt_lsb

#### 7.4.4 miss\_cnt

地址: 0x4000900c

miss\_cnt

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

miss\_cnt

位	名称	权限	复位值	描述
31:0	miss_cnt	r	0	

## 8.1 简介

红外遥控（Infrared remote，简称 IR）是一种无线、非接触式控制技术，具有抗干扰能力强、信息传输可靠、功耗低、成本低等优点。红外遥控的发射电路是采用红外发光二极管来发出经过调制的红外光波；接收电路由红外接收二极管、三极管或硅光电池组成，它们将红外发射器发射的红外光转换为相应的电信号，再送至后置放大器。

## 8.2 主要特征

- 支持以固定协议 NEC、RC-5 接收数据
- 支持以脉冲宽度计数方式接收任意格式数据
- 强大的红外波形编辑能力，可发出符合各种协议的波形
- 多达 15 个档位的功率设定，以适应不同的功耗要求
- 最多支持 64-bit 数据位
- 64 字节的接收 FIFO
- 载波频率与占空比可编程

## 8.3 功能描述

### 8.3.1 固定协议接收

IR 接收支持两种固定的协议，分别为 NEC 协议和 RC-5 协议。

- NEC 协议

NEC 协议的逻辑 1 与逻辑 0 波形如下图所示：

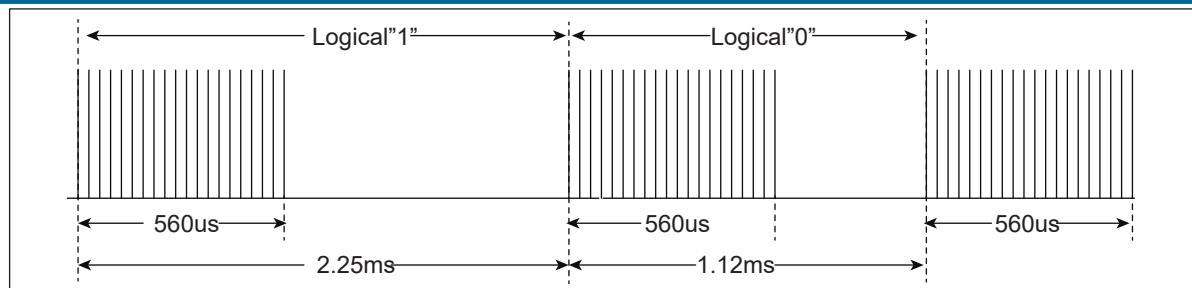


图 8.1: NEC 逻辑波形

逻辑 1 为 2.25ms，脉冲时间 560us；逻辑 0 位 1.12ms，脉冲时间 560us。NEC 协议的具体格式如下图所示：



图 8.2: NEC 协议波形

头脉冲是 9ms 的高电平脉冲和 4.5ms 的低电平，之后是 8-bit 的地址码及其反码，然后是 8-bit 的命令码及其反码，尾脉冲是 560us 高电平与 560us 低电平。

- RC-5 协议

RC-5 协议的逻辑 1 与逻辑 0 波形如下图所示：

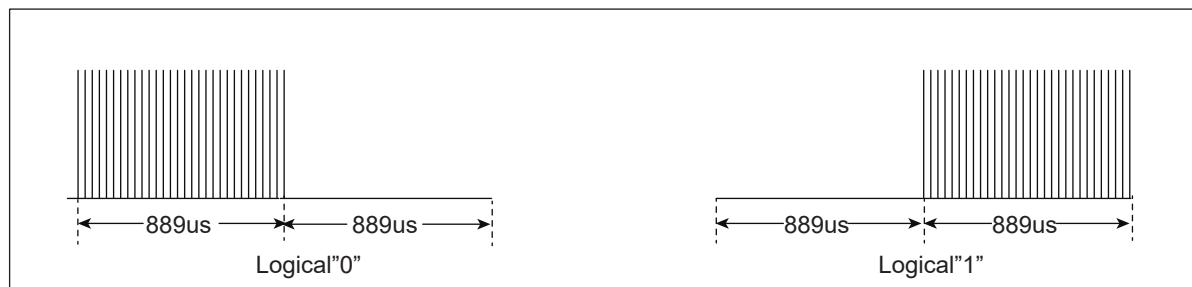


图 8.3: RC5 逻辑波形

逻辑 1 为 1.778ms，先是 889us 的低电平后是 889us 的高电平；逻辑 0 与逻辑 1 波形相反。RC-5 协议的具体格式如下图所示：

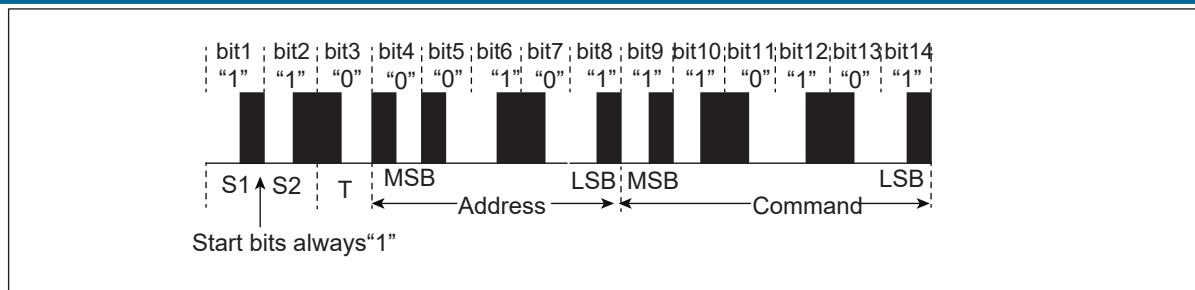


图 8.4: RC5 协议波形

前两位为开始位，固定为逻辑 1，第三位是翻转位，当一个键值发出然后再按下时该位会取反。之后 5 位是地址码与 6 位命令码。需要注意的是，常见的红外一体接收头为了提高接收灵敏度，接收到高电平后输出的是低电平，所以在使用 IR 接收功能时要将接收翻转功能打开。

### 8.3.2 脉冲宽度接收

对于 NEC、RC-5 协议以外的其他任意格式的数据，IR 会以其时钟去依次计数每个高低电平的持续时间，然后将数据存入到深度为 64 字节的接收 FIFO 之中。

### 8.3.3 普通发送模式

用户可根据具体协议分别对头脉冲、尾脉冲、逻辑 0 和逻辑 1 脉冲进行相应的配置。在设置时需要计算出所使用的协议内各种不同宽度脉冲的公共脉冲宽度单位，即最大公约数，填入寄存器 IRTX\_PULSE\_WIDTH 的低 12 位，各脉冲将其所对应的倍数填入寄存器 IRTX\_PW 中。IR 最大支持 64-bit 数据位，分为两个 32 位寄存器 IRTX\_DATA\_WORD0 和 IRTX\_DATA\_WORD1。

### 8.3.4 脉冲宽度发送

对于不适用于普通发送模式的协议，IR 提供了脉冲宽度发送的方式。先计算出所使用的协议内各种不同宽度脉冲的公共脉冲宽度单位，即最大公约数，填入寄存器 IRTX\_PULSE\_WIDTH 的低 12 位。然后将从第一个电平开始到最后一个电平为止的各个电平宽度所对应的倍数填入寄存器 IRTX\_SWM\_PW\_n(0<=n<=7) 中，每个电平宽度倍数占 4-bit。

### 8.3.5 载波调制

通过设置寄存器 IRTX\_PULSE\_WIDTH 的高 16 位可以产生不同频率和占空比的载波，该寄存器的 <TXMPH1W> 位设置的是载波相位 1 的宽度，<TXMPH0W> 位设置的是载波相位 0 的宽度。

### 8.3.6 IR 中断

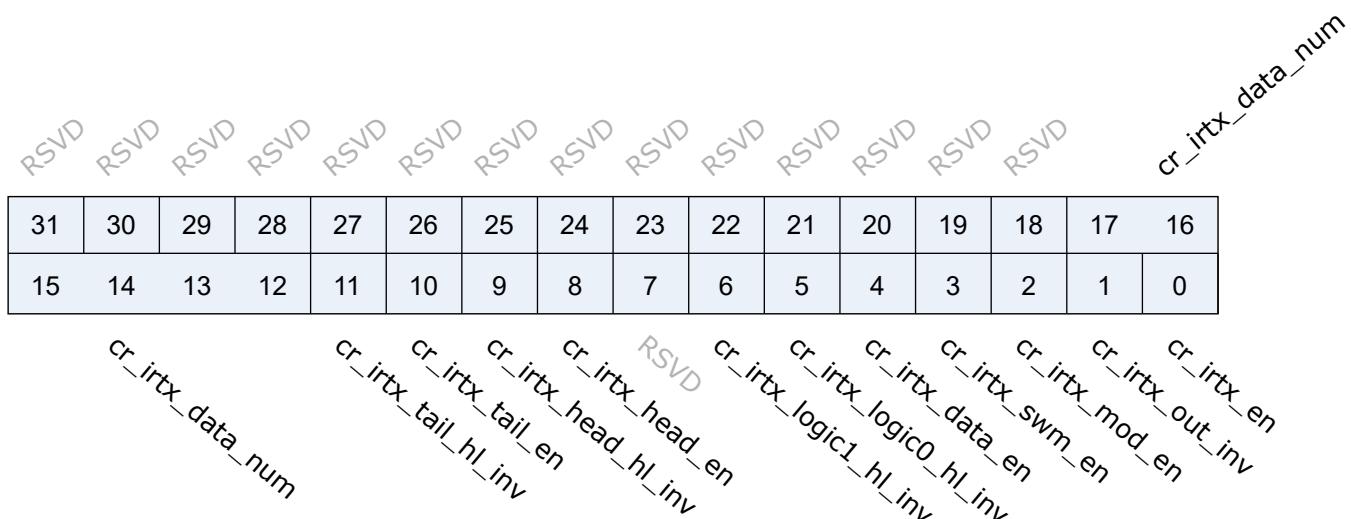
IR 有单独的发送和接收中断，当一次发送动作结束时，会产生发送中断。当接收到一段数据时，它会等待电平持续长度达到设定的结束阈值时产生接收中断。可以通过寄存器 IRTX\_INT\_STS 查询发送中断状态和清除中断，通过寄存器 IRRX\_INT\_STS 查询接收中断状态和清除中断。

## 8.4 寄存器描述

名称	描述
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
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

地址: 0x4000a600

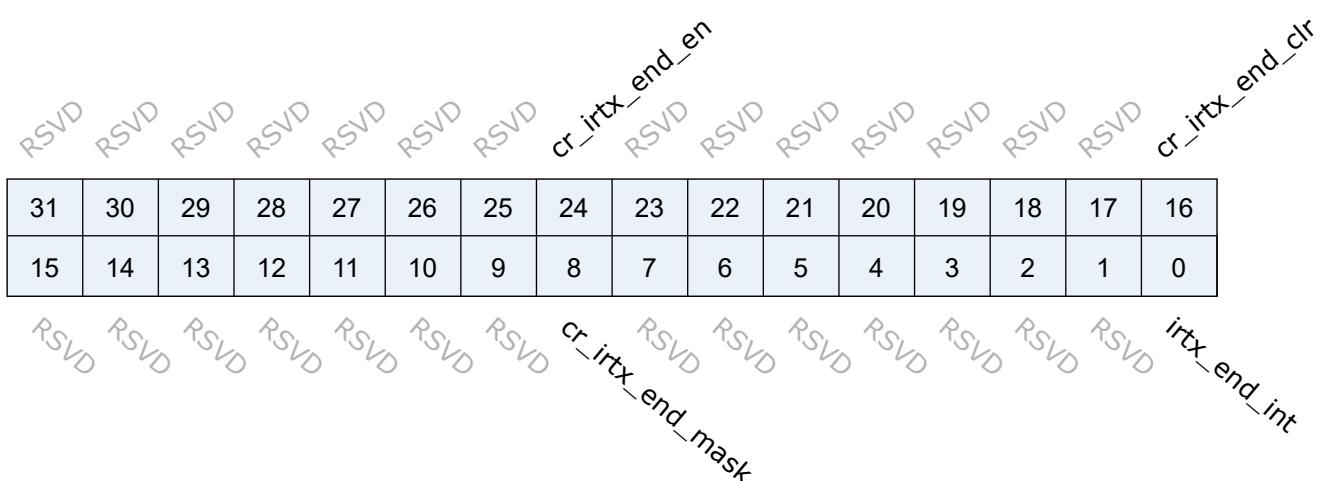


位	名称	权限	复位值	描述
31:18	RSVD			
17:12	cr_irtx_data_num	r/w	6'd31	Bit count of Data phase (unit: bit / PW for normal / SWM)
11	cr_irtx_tail_hl_inv	r/w	1'b0	Tail pulse H/L inverse signal (Don't care if SWM is enabled) 0: Phase 0 is High (Active), phase 1 is Low (Idle) (H -> L) 1: Phase 0 is Low (Idle), phase 1 is High (Active) (L -> H)
10	cr_irtx_tail_en	r/w	1'b1	Enable signal of tail pulse (Don't care if SWM is enabled)
9	cr_irtx_head_hl_inv	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	cr_irtx_head_en	r/w	1'b1	Enable signal of head pulse (Don't care if SWM is enabled)
7	RSVD			
6	cr_irtx_logic1_hl_inv	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	cr_irtx_logic0_hl_inv	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	cr_irtx_data_en	r/w	1'b1	Enable signal of data phase (Don't care if SWM is enabled)
3	cr_irtx_swm_en	r/w	1'b0	Enable signal of IRTX Software Mode (SWM)
2	cr_irtx_mod_en	r/w	1'b0	Enable signal of output modulation

位	名称	权限	复位值	描述
1	cr_irtx_out_inv	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	cr_irtx_en	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

地址: 0x4000a604



位	名称	权限	复位值	描述
31:25	RSVD			
24	cr_irtx_end_en	r/w	1'b1	Interrupt enable of irtx_end_int
23:17	RSVD			
16	cr_irtx_end_clr	w1c	1'b0	Interrupt clear of irtx_end_int
15:9	RSVD			
8	cr_irtx_end_mask	r/w	1'b1	Interrupt mask of irtx_end_int
7:1	RSVD			
0	irtx_end_int	r	1'b0	IRTX transfer end interrupt

### 8.4.3 irtx\_data\_word0

地址: 0x4000a608

cr\_irtx\_data\_word0

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

cr\_irtx\_data\_word0

位	名称	权限	复位值	描述
31:0	cr_irtx_data_word0	r/w	32'h0	TX data word 0 (Don't care if SWM is enabled)

### 8.4.4 irtx\_data\_word1

地址: 0x4000a60c

cr\_irtx\_data\_word1

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

cr\_irtx\_data\_word1

位	名称	权限	复位值	描述
31:0	cr_irtx_data_word1	r/w	32'h0	TX data word 1 (Don't care if SWM is enabled)

### 8.4.5 irtx\_pulse\_width

地址: 0x4000a610

cr\_irtx\_mod\_ph1\_w

cr\_irtx\_mod\_ph0\_w

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

RSVD RSVD RSVD RSVD

cr\_irtx\_pw\_unit

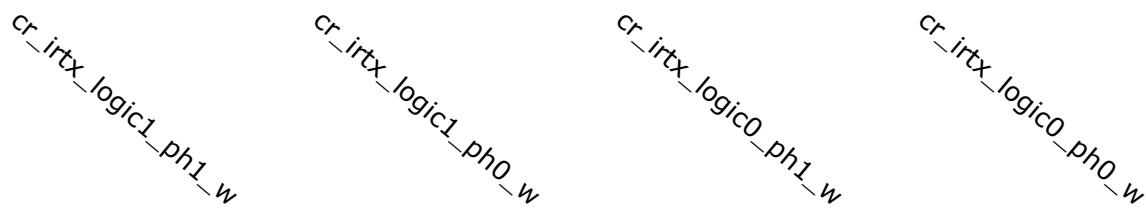
位	名称	权限	复位值	描述
31:24	cr_irtx_mod_ph1_w	r/w	8'd34	Modulation phase 1 width
23:16	cr_irtx_mod_ph0_w	r/w	8'd17	Modulation phase 0 width

位	名称	权限	复位值	描述
15:12	RSVD			
11:0	cr_irtx_pw_unit	r/w	12'd1124	Pulse width unit

#### 8.4.6 irtx\_pw

地址: 0x4000a614

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



位	名称	权限	复位值	描述
31:28	cr_irtx_tail_ph1_w	r/w	4'd0	Pulse width of tail pulse phase 1 (Don't care if SWM is enabled)
27:24	cr_irtx_tail_ph0_w	r/w	4'd0	Pulse width of tail pulse phase 0 (Don't care if SWM is enabled)
23:20	cr_irtx_head_ph1_w	r/w	4'd7	Pulse width of head pulse phase 1 (Don't care if SWM is enabled)
19:16	cr_irtx_head_ph0_w	r/w	4'd15	Pulse width of head pulse phase 0 (Don't care if SWM is enabled)
15:12	cr_irtx_logic1_ph1_w	r/w	4'd2	Pulse width of logic1 phase 1 (Don't care if SWM is enabled)
11:8	cr_irtx_logic1_ph0_w	r/w	4'd0	Pulse width of logic1 phase 0 (Don't care if SWM is enabled)
7:4	cr_irtx_logic0_ph1_w	r/w	4'd0	Pulse width of logic0 phase 1 (Don't care if SWM is enabled)
3:0	cr_irtx_logic0_ph0_w	r/w	4'd0	Pulse width of logic0 phase 0 (Don't care if SWM is enabled)

### **8.4.7 irtx\_swm\_pw\_0**

地址: 0x4000a640

cr\_irtx\_swm\_pw\_0

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

cr\_irtx\_swm\_pw\_0

位	名称	权限	复位值	描述
31:0	cr_irtx_swm_pw_0	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**

地址: 0x4000a644

cr\_irtx\_swm\_pw\_1

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

cr\_irtx\_swm\_pw\_1

位	名称	权限	复位值	描述
31:0	cr_irtx_swm_pw_1	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**

地址: 0x4000a648

cr\_irtx\_swm\_pw\_2

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

cr\_irtx\_swm\_pw\_2

位	名称	权限	复位值	描述
31:0	cr_irtx_swm_pw_2	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

地址: 0x4000a64c

cr\_irtx\_swm\_pw\_3

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

cr\_irtx\_swm\_pw\_3

位	名称	权限	复位值	描述
31:0	cr_irtx_swm_pw_3	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

地址: 0x4000a650

cr\_irtx\_swm\_pw\_4

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

cr\_irtx\_swm\_pw\_4

位	名称	权限	复位值	描述
31:0	cr_irtx_swm_pw_4	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

地址: 0x4000a654

cr\_irtx\_swm\_pw\_5

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

cr\_irtx\_swm\_pw\_5

位	名称	权限	复位值	描述
31:0	cr_irtx_swm_pw_5	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

地址: 0x4000a658

cr\_irtx\_swm\_pw\_6

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

cr\_irtx\_swm\_pw\_6

位	名称	权限	复位值	描述
31:0	cr_irtx_swm_pw_6	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

地址: 0x4000a65c

cr\_irtx\_swm\_pw\_7

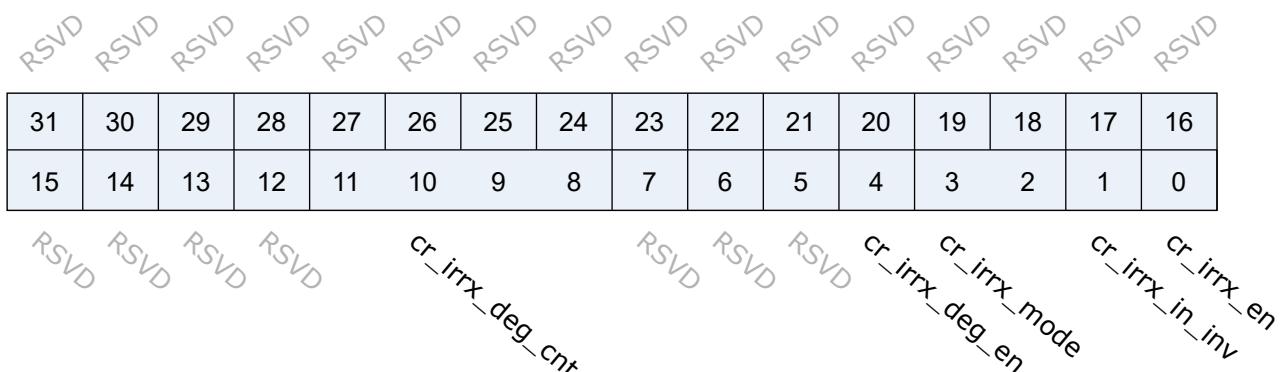
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

cr\_irtx\_swm\_pw\_7

位	名称	权限	复位值	描述
31:0	cr_irrx_swm_pw_7	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

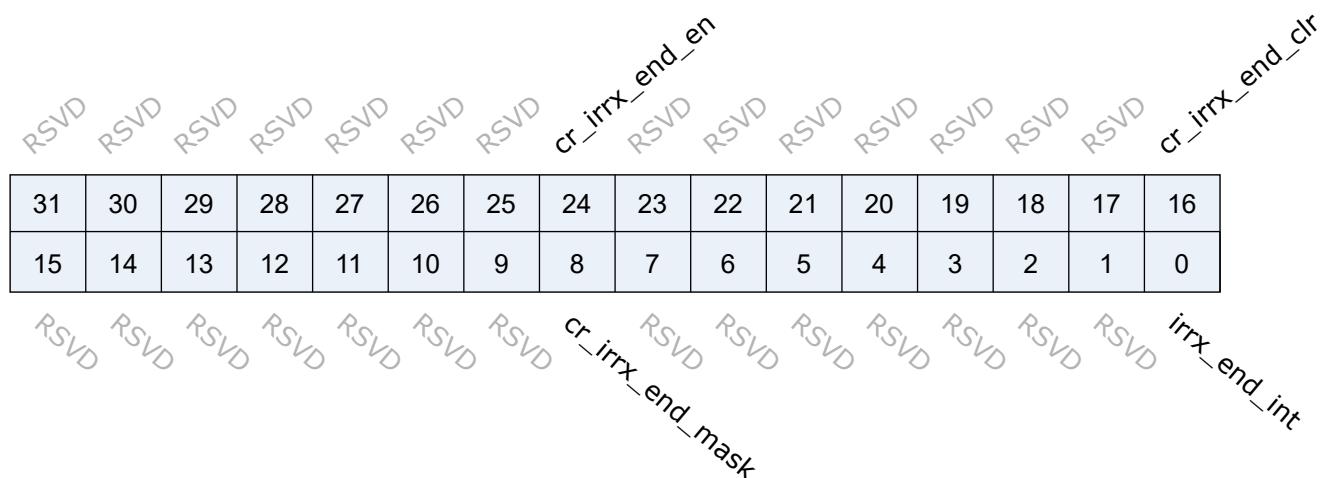
地址: 0x4000a680



位	名称	权限	复位值	描述
31:12	RSVD			
11:8	cr_irrx_deg_cnt	r/w	4'd0	De-glitch function cycle count
7:5	RSVD			
4	cr_irrx_deg_en	r/w	1'b0	Enable signal of IRRX input de-glitch function
3:2	cr_irrx_mode	r/w	2'd0	IRRX mode 0: NEC 1: RC5 2: SW pulse-width detection mode (SWM) 3: Reserved
1	cr_irrx_in_inv	r/w	1'b1	Input inverse signal
0	cr_irrx_en	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

地址: 0x4000a684



位	名称	权限	复位值	描述
31:25	RSVD			
24	cr_irrx_end_en	r/w	1'b1	Interrupt enable of irrx_end_int
23:17	RSVD			
16	cr_irrx_end_clr	w1c	1'b0	Interrupt clear of irrx_end_int
15:9	RSVD			
8	cr_irrx_end_mask	r/w	1'b1	Interrupt mask of irrx_end_int
7:1	RSVD			
0	irrx_end_int	r	1'b0	IRRX transfer end interrupt

### 8.4.17 irrx\_pw\_config

地址: 0x4000a688

cr\_irrx\_end\_th

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

cr\_irrx\_data\_th

位	名称	权限	复位值	描述
31:16	cr_irrx_end_th	r/w	16'd8999	Pulse width threshold to trigger END condition

位	名称	权限	复位值	描述
15:0	cr_irrx_data_th	r/w	16'd3399	Pulse width threshold for Logic0/1 detection (Don't care if SWM is enabled)

#### 8.4.18 irrx\_data\_count

地址: 0x4000a690

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

位	名称	权限	复位值	描述
31:7	RSVD			
6:0	sts_irrx_data_cnt	r	7'd0	RX data bit count (pulse-width count for SWM)

#### 8.4.19 irrx\_data\_word0

地址: 0x4000a694

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	sts_irrx_data_word0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

位	名称	权限	复位值	描述
31:0	sts_irrx_data_word0	r	32'h0	RX data word 0

### 8.4.20 irrx\_data\_word1

地址: 0x4000a698

sts\_irrx\_data\_word1

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

sts\_irrx\_data\_word1

位	名称	权限	复位值	描述
31:0	sts_irrx_data_word1	r	32'h0	RX data word 1

### 8.4.21 irrx\_swm\_fifo\_config\_0

地址: 0x4000a6c0

RSVD																
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	

rx\_fifo\_cnt      rx\_fifo\_overflow      rx\_fifo\_clr  
rx\_fifo\_underflow

位	名称	权限	复位值	描述
31:11	RSVD			
10:4	rx_fifo_cnt	r	7'd0	RX FIFO available count
3	rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
2	rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
1	RSVD			
0	rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO

### 8.4.22 irrx\_swm\_fifo\_rdata

地址: 0x4000a6c4

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

rx\_fifo\_rdata

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	rx_fifo_rdata	r	16'h0	IRRX Software Mode pulse width data

## 9.1 简介

串行外设接口（Serial Peripheral Interface Bus, SPI）是一种用于短程通信的同步串行通信接口规范，装置之间使用全双工模式通信，是一个主机和一个或多个从机的主从模式。需要至少 4 根线，事实上 3 根也可以（单向传输时），包括 SDI（数据输入）、SDO（数据输出）、SCLK（时钟）、CS（片选）。

## 9.2 主要特征

- 既可作为 SPI 主设备，也可作为 SPI 从设备
- 发送和接收通道各有深度为 4 帧的 FIFO
- 主从设备都支持 4 种时钟格式（CPOL, CPHA）
- 主从设备都支持 1/2/3/4 字节传输模式
- 灵活的时钟配置，最高可支持 40M 时钟
- 可配置 MSB/LSB 优先传输
- 接收过滤功能
- 从设备下的超时机制
- 支持 DMA 传输模式

## 9.3 功能描述

### 9.3.1 时钟控制

依照不同的时钟相位以及极性设定，SPI 时钟共有四种模式，可以通过寄存器 SPI\_CONFIG 的 bit4 (CPOL) 和 bit5 (CPHA) 进行设置。CPOL 用来决定 SCK 时钟信号空闲时的电平，CPOL=0 则空闲电平为低电平，CPOL=1 则空闲电平为高电平。CPHA 用来决定采样时刻，CPHA=0 则在每个周期的第一个时钟沿采样，CPHA=1 则在每个周期的第二个时钟沿采样。通过设置寄存器 SPI\_PRD\_0 和 SPI\_PRD\_1，还可以调整时钟的开始和结束电平持续时间、相位 0/1 的时间以及每帧数据之间的间隔。四种模式下的具体设置如下图所示：

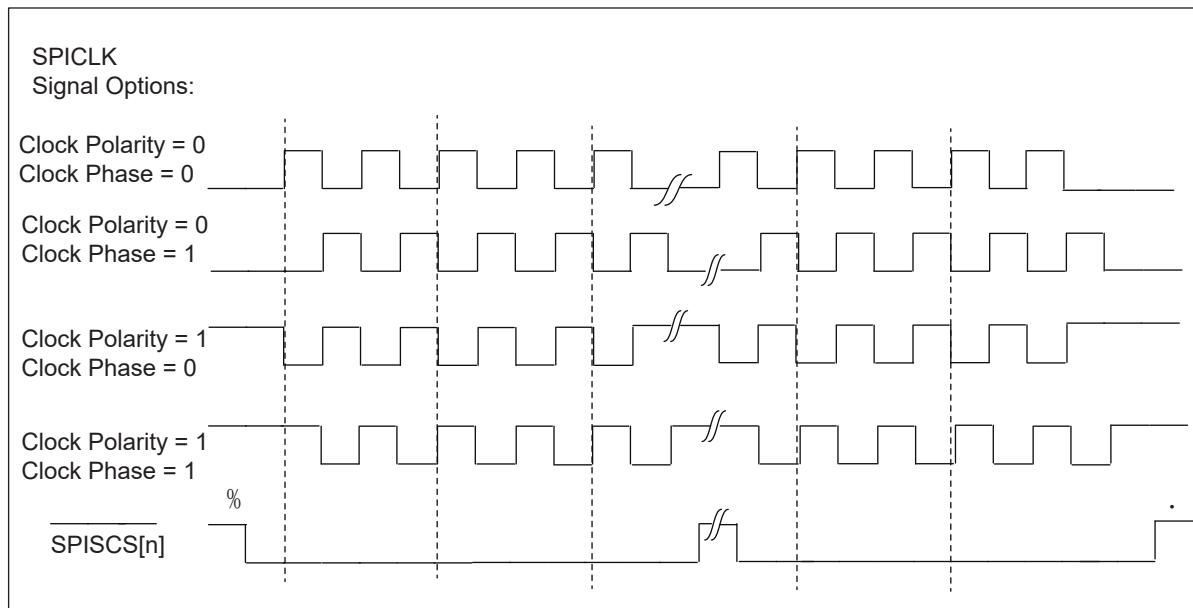


图 9.1: SPI 时序图

其中各数字含义如下：1 是起始条件的长度，2 是停止条件的长度，3 是相位 0 的长度，4 是相位 1 的长度，5 是每帧数据之间的间隔。

### 9.3.2 主设备持续传输模式

开启该模式后，在发送完当前数据而 FIFO 里还存在可用数据时，CS 信号不会被释放。

### 9.3.3 接收过滤功能

通过设置需要过滤掉的开始位和结束位，SPI 会将接收到数据中的对应数据段丢弃。如下图所示：

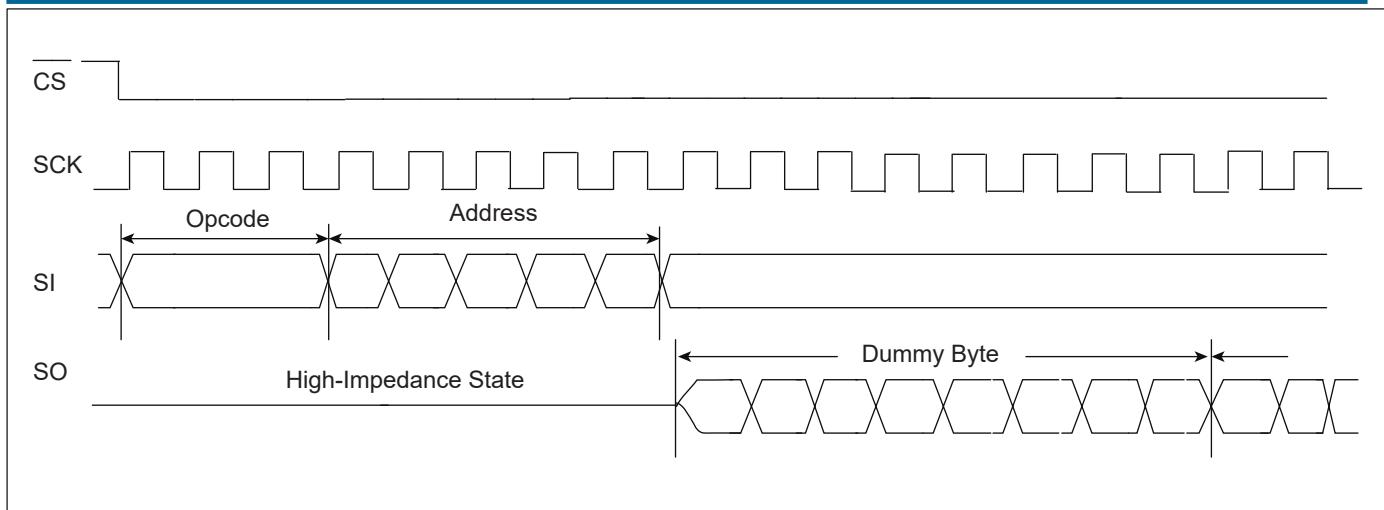


图 9.2: SPI Ignore 波形图

上图中过滤的开始位设为 0，结束位设为 7 则 Dummy Byte 会被收到，结束位设为 15 则 Dummy Byte 会被丢弃。

### 9.3.4 接收去差错功能

通过使能该功能和设置门限值，SPI 会将达不到门限值宽度的数据丢弃。

### 9.3.5 从模式超时机制

通过设定一个超时门限，当从模式下 SPI 超过该时间值未收到时钟信号时，会触发中断。

### 9.3.6 I/O 传输模式

芯片通信处理器可以响应来自 FIFO 的中断来执行 FIFO 填充和清空操作。每个 FIFO 都有一个可编程的 FIFO 触发阈值来触发中断。当 RX FIFO 中的数值超过 SPI 控制器 1 中的 RX FIFO 触发阈值时，将产生一个中断，向芯片通信处理器发送信号来清空 RX FIFO。当 TX FIFO 中的数值小于或等于 SPI 控制寄存器 1 中的 TX FIFO 触发阈值加 1 时，将产生中断，向芯片通信处理器发送信号来重新填充 TX FIFO。可以通过查询 SPI 状态寄存器来确定 FIFO 中的采样值以及 FIFO 的状态。软件负责确保正确的 RX FIFO 触发阈值和 TX FIFO 触发阈值，以防止接收 FIFO 超限和发送 FIFO 欠载。

### 9.3.7 DMA 传输模式

SPI 支持 DMA 传输模式。使用该模式需要分别设置 TX 和 RX FIFO 的阈值，当该模式启用后，UART 会对 TX/RX FIFO 进行检查，一旦 TX/RX 的 FIFO 可用计数值大于其设定的阈值，将会发起 DMA 请求，DMA 会按照设定将数据搬移至 TX FIFO 中或从 RX FIFO 中移出。

### 9.3.8 SPI 中断

SPI 有着丰富的中断控制，包括以下几种中断模式：

- SPI 传输结束中断
- TX FIFO 请求中断
- RX FIFO 请求中断
- 从模式传输超时中断
- 从模式 TX 过载中断
- TX/RX FIFO 溢出中断

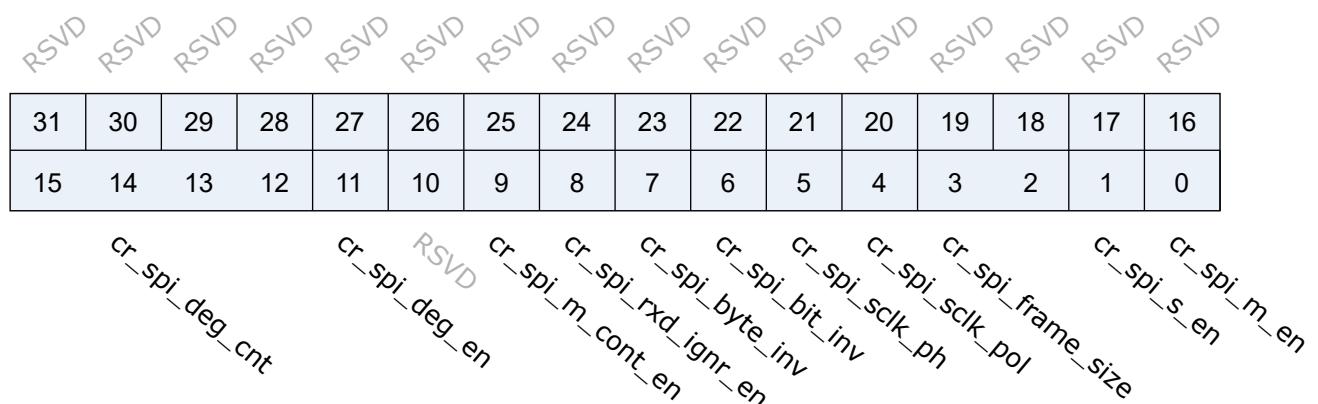
在主模式下，SPI 传输结束中断会在每帧数据传输结束时触发；在从模式下，SPI 传输结束中断会在 CS 信号被释放时触发。TX/RX FIFO 请求中断会在其 FIFO 可用计数值大于其设定的阈值时触发，当条件不满足时该中断标志会自动清除。从模式传输超时中断会在从模式下超过超时门限值未收到时钟信号时触发。如果 TX/RX FIFO 发生了上溢或者下溢，会触发 TX/RX FIFO 溢出中断，当 FIFO 清除位 TFC/RFC 被置 1 时，对应的 FIFO 会被清空，同时溢出中断标志会自动清除。可以通过寄存器 SPI\_INT\_STS 查询各中断状态和对相应的位写 1 清除中断。

## 9.4 寄存器描述

名称	描述
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 ignore 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
spi_fifo_rdata	SPI FIFO read data

### 9.4.1 spi\_config

地址: 0x4000a200

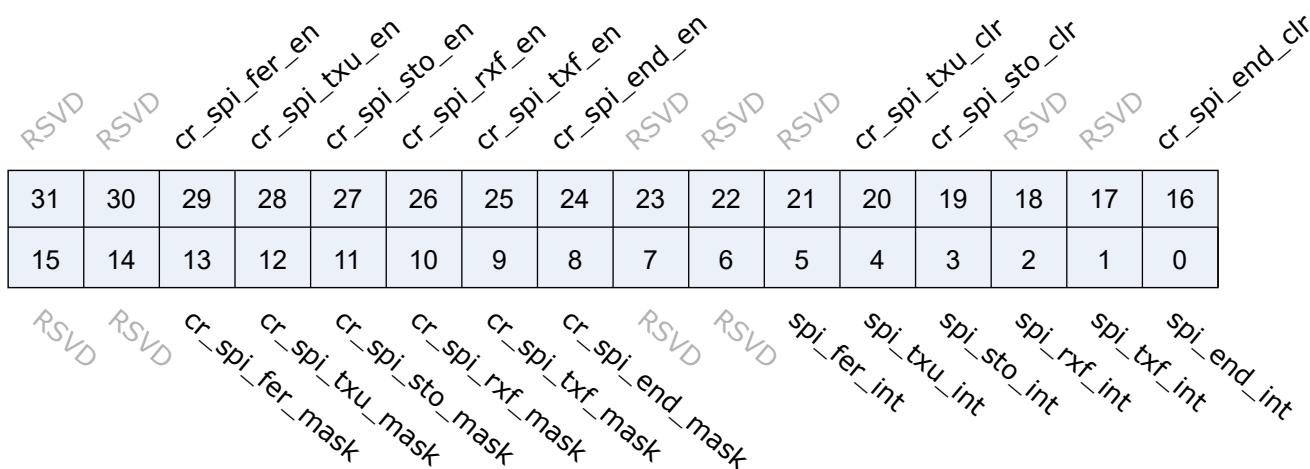


位	名称	权限	复位值	描述
31:16	RSVD			
15:12	cr_spi_deg_cnt	r/w	4'd0	De-glitch function cycle count
11	cr_spi_deg_en	r/w	1'b0	Enable signal of all input de-glitch function
10	RSVD			
9	cr_spi_m_cont_en	r/w	1'b0	Enable signal of master continuous transfer mode 1'b0: Disabled, SS_n will de-assert between each data frame 1'b1: Enabled, SS_n will stay asserted between each consecutive data frame if the next data is valid in the FIFO
8	cr_spi_rxd_ignr_en	r/w	1'b0	Enable signal of RX data ignore function
7	cr_spi_byte_inv	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	cr_spi_bit_inv	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	cr_spi_sclk_ph	r/w	1'b0	SCLK clock phase inverse signal
4	cr_spi_sclk_pol	r/w	1'b0	SCLK polarity 0: SCLK output LOW at IDLE state 1: SCLK output HIGH at IDLE state

位	名称	权限	复位值	描述
3:2	cr_spi_frame_size	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	cr_spi_s_en	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	cr_spi_m_en	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

地址: 0x4000a204

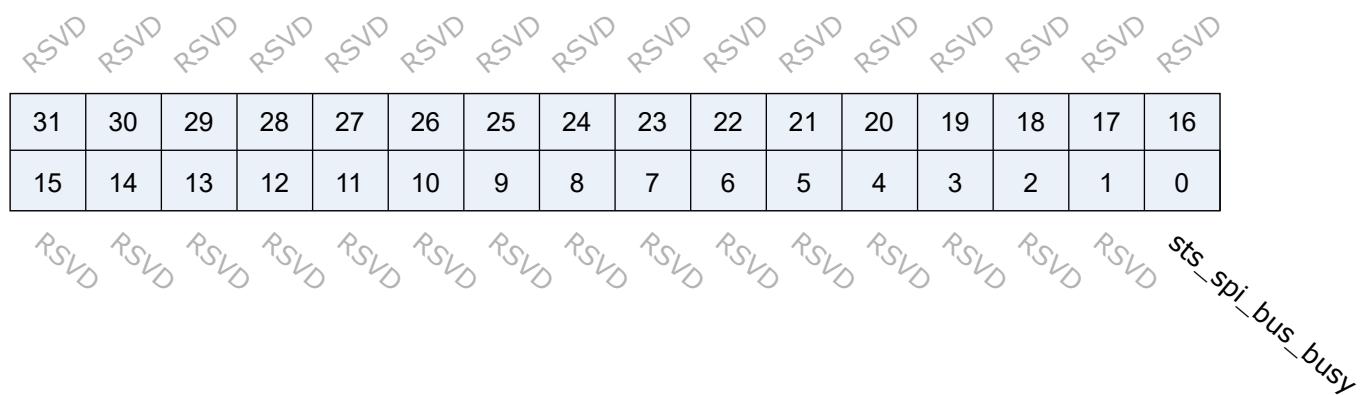


位	名称	权限	复位值	描述
31:30	RSVD			
29	cr_spi_fer_en	r/w	1'b1	Interrupt enable of spi_fer_int
28	cr_spi_txu_en	r/w	1'b1	Interrupt enable of spi_txu_int
27	cr_spi_sto_en	r/w	1'b1	Interrupt enable of spi_sto_int
26	cr_spi_rxf_en	r/w	1'b1	Interrupt enable of spi_rxf_int
25	cr_spi_txf_en	r/w	1'b1	Interrupt enable of spi_txf_int
24	cr_spi_end_en	r/w	1'b1	Interrupt enable of spi_end_int
23:21	RSVD			

位	名称	权限	复位值	描述
20	cr_spi_txu_clr	w1c	1'b0	Interrupt clear of spi_txu_int
19	cr_spi_sto_clr	w1c	1'b0	Interrupt clear of spi_sto_int
18:17	RSVD			
16	cr_spi_end_clr	w1c	1'b0	Interrupt clear of spi_end_int
15:14	RSVD			
13	cr_spi_fer_mask	r/w	1'b1	Interrupt mask of spi_fer_int
12	cr_spi_txu_mask	r/w	1'b1	Interrupt mask of spi_txu_int
11	cr_spi_sto_mask	r/w	1'b1	Interrupt mask of spi_sto_int
10	cr_spi_rxf_mask	r/w	1'b1	Interrupt mask of spi_rxf_int
9	cr_spi_txf_mask	r/w	1'b1	Interrupt mask of spi_txf_int
8	cr_spi_end_mask	r/w	1'b1	Interrupt mask of spi_end_int
7:6	RSVD			
5	spi_fer_int	r	1'b0	SPI TX/RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
4	spi_txu_int	r	1'b0	SPI slave mode TX underrun error flag, triggered when TxD is not ready during transfer in slave mode
3	spi_sto_int	r	1'b0	SPI slave mode transfer time-out interrupt, triggered when SPI bus is idle for a given value
2	spi_rxf_int	r	1'b0	SPI RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto-cleared when data is popped
1	spi_txf_int	r	1'b0	SPI TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed
0	spi_end_int	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

地址: 0x4000a208



位	名称	权限	复位值	描述
31:1	RSVD			
0	sts_spi_bus_busy	r	1'b0	Indicator of SPI bus busy

### 9.4.4 spi\_prd\_0

地址: 0x4000a210

cr_spi_prd_d_ph_1								cr_spi_prd_d_ph_0							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

位	名称	权限	复位值	描述
31:24	cr_spi_prd_d_ph_1	r/w	8'd15	Length of DATA phase 1 (please refer to "Timing" tab)
23:16	cr_spi_prd_d_ph_0	r/w	8'd15	Length of DATA phase 0 (please refer to "Timing" tab)
15:8	cr_spi_prd_p	r/w	8'd15	Length of STOP condition (please refer to "Timing" tab)
7:0	cr_spi_prd_s	r/w	8'd15	Length of START condition (please refer to "Timing" tab)

### 9.4.5 spi\_prd\_1

地址: 0x4000a214

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

cr\_spi\_prd\_i

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	cr_spi_prd_i	r/w	8'd15	Length of INTERVAL between frame (please refer to "Timing" tab)

### 9.4.6 spi\_rxd\_ignr

地址: 0x4000a218

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

cr\_spi\_rxd\_ignr\_s

cr\_spi\_rxd\_ignr\_p

位	名称	权限	复位值	描述
31:21	RSVD			
20:16	cr_spi_rxd_ignr_s	r/w	5'd0	Starting point of RX data ignore function
15:5	RSVD			
4:0	cr_spi_rxd_ignr_p	r/w	5'd0	Stopping point of RX data ignore function

#### 9.4.7 spi\_sto\_value

地址: 0x4000a21c

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

cr\_spi\_sto\_value

位	名称	权限	复位值	描述
31:12	RSVD			
11:0	cr_spi_sto_value	r/w	12'hFFF	Time-out value for spi_sto_int triggering

#### 9.4.8 spi\_fifo\_config\_0

地址: 0x4000a280

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

RSVD RSVD

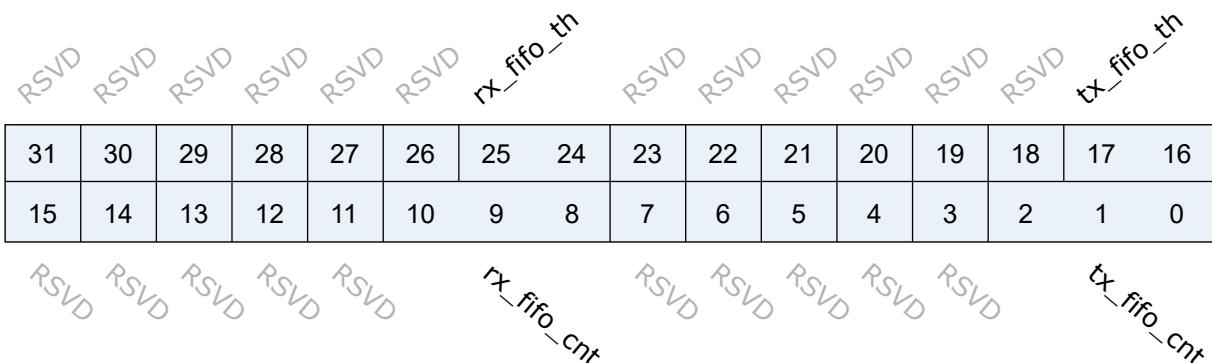
rx\_fifo\_underflow rx\_fifo\_overflow tx\_fifo\_underflow tx\_fifo\_overflow rx\_fifo\_clr tx\_fifo\_clr spi\_dma\_rx\_en spi\_dma\_tx\_en

位	名称	权限	复位值	描述
31:8	RSVD			
7	rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	spi_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface

位	名称	权限	复位值	描述
0	spi_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

#### 9.4.9 spi\_fifo\_config\_1

地址: 0x4000a284



位	名称	权限	复位值	描述
31:26	RSVD			
25:24	rx_fifo_th	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	tx_fifo_th	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	rx_fifo_cnt	r	3'd0	RX FIFO available count
7:3	RSVD			
2:0	tx_fifo_cnt	r	3'd4	TX FIFO available count

#### 9.4.10 spi\_fifo\_wdata

地址: 0x4000a288

spi\_fifo\_wdata

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

spi\_fifo\_wdata

位	名称	权限	复位值	描述
31:0	spi_fifo_wdata	w	x	

#### 9.4.11 spi\_fifo\_rdata

地址: 0x4000a28c

spi\_fifo\_rdata

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

spi\_fifo\_rdata

位	名称	权限	复位值	描述
31:0	spi_fifo_rdata	r	32'h0	

# 10

## UART

### 10.1 简介

通用异步收发传输器（Universal Asynchronous Receiver/Transmitter，通常称为 UART）是一种异步收发传输器，提供了与外部设备进行全双工数据交换的灵活方式。BL70x 共有 2 组 UART 口（UART0 和 UART1），通过配合 DMA 使用，可以实现高效的数据通信。

### 10.2 主要特征

- 全双工异步通信
- 数据位长度可选择 5/6/7/8 比特
- 停止位长度可选择 0.5/1/1.5/2 比特
- 支持奇/偶/无校验比特
- 可侦测错误的起始比特
- 支持 LIN 协议（收发 BREAK/SYNC）
- 丰富的中断控制
- 支持硬件流控（RTS/CTS）
- 便捷的波特率编程
- 可配置 MSB/LSB 优先传输
- 普通/固定字符的自动波特率检测
- 128 字节发送/接收 FIFO
- 支持 DMA 传输模式

## 10.3 功能描述

### 10.3.1 数据格式描述

正常的 UART 通信数据是由起始位、数据位、奇偶校验位、停止位组成的。BL70x 的 UART 支持可配置的数据位、奇偶校验位和停止位，这些都在寄存器 `utx_config` 和 `urx_config` 中设置。一帧数据的波形如下图所示：

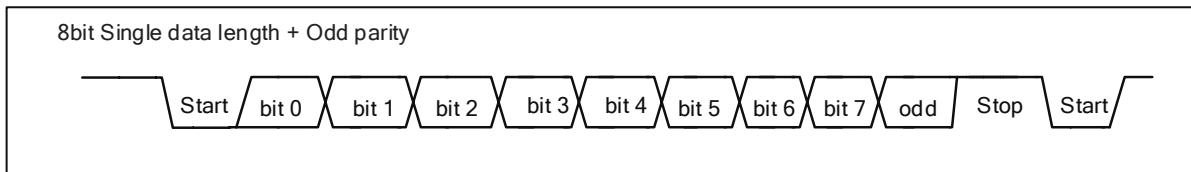


图 10.1: UART 数据格式

数据帧的起始位占用 1-bit，停止位可以通过配置寄存器 `utx_config` 中的 `cr_utx_bit_cnt_p` 实现 0.5/1/1.5/2 位宽。起始位为低电平，停止位为高电平。数据位宽可以通过寄存器 `utx_config` 中的 `cr_utx_bit_cnt_d` 配置为 5/6/7/8 位宽。当置位寄存器 `utx_config` 中的 `cr_utx_prt_en` 和寄存器 `urx_config` 中的 `cr_urx_prt_en` 时，数据帧会在数据之后添加一位奇偶校验位。寄存器 `utx_config` 中的 `cr_utx_prt_sel` 和寄存器 `urx_config` 中的 `cr_urx_prt_sel` 用于选择奇校验还是偶校验。当接收器检测到输入数据的校验位错误时会产生校验错误中断。但是接收的数据仍然会进入 FIFO。奇校验的计算方法：如果当前数据位 1 的个数是奇数个，奇校验位为 0；反之为 1。偶校验的计算方法：如果当前数据位 1 的个数是奇数个，偶校验位为 1；反之为 0。

### 10.3.2 基本架构图

### 10.3.3 时钟源

UART 有两个时钟源：96MHz PLL\_CLK 以及 FCLK。时钟中的分频器用于对时钟源进行分频，然后产生时钟信号来驱动 UART 模块。如下图所示：

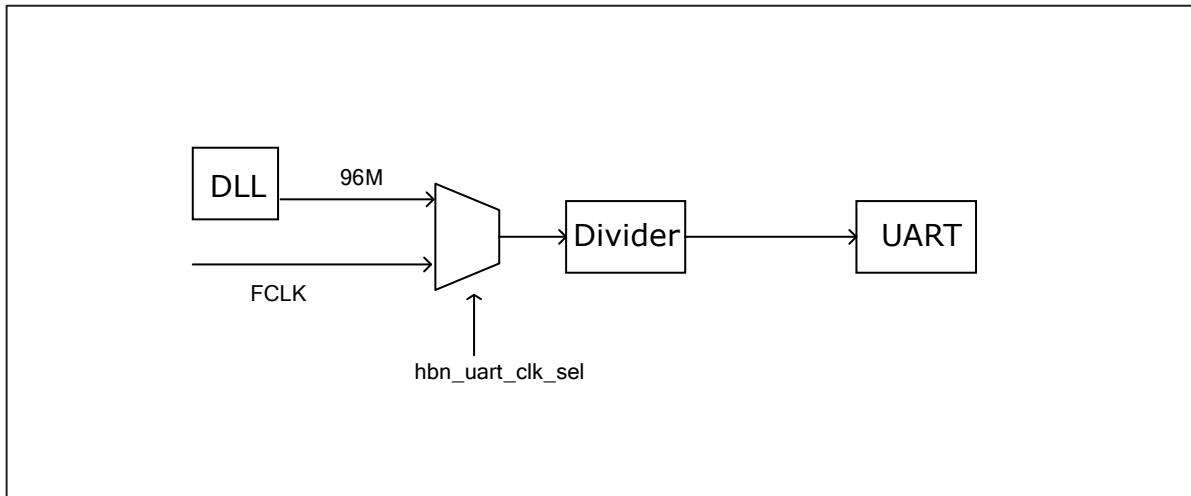


图 10.2: UART 时钟

### 10.3.4 波特率设定

$$\text{Baudrate} = \frac{\text{UART\_clk}}{\text{uart\_prd} + 1}$$

用户可单独设置 RX 与 TX 的波特率，以 TX 为例：uart\_prd 的数值为寄存器 UART\_BIT\_PRD 的低 16 位 cr\_utx\_bit\_prd 数值，由于 16 位位宽系数最大值为 65535，所以 UART 支持的最小波特率为：UART\_clk/65536。

在 UART 对数据进行采样之前，会先对数据进行滤波，将波形当中的毛刺滤掉。然后会在上述 16 位系数的中间值时刻进行采样，这样根据不同的波特率调整不同的采样时刻，以保持其采到的始终是中间值，大大提高了灵活性与精度。采样过程如下图所示：

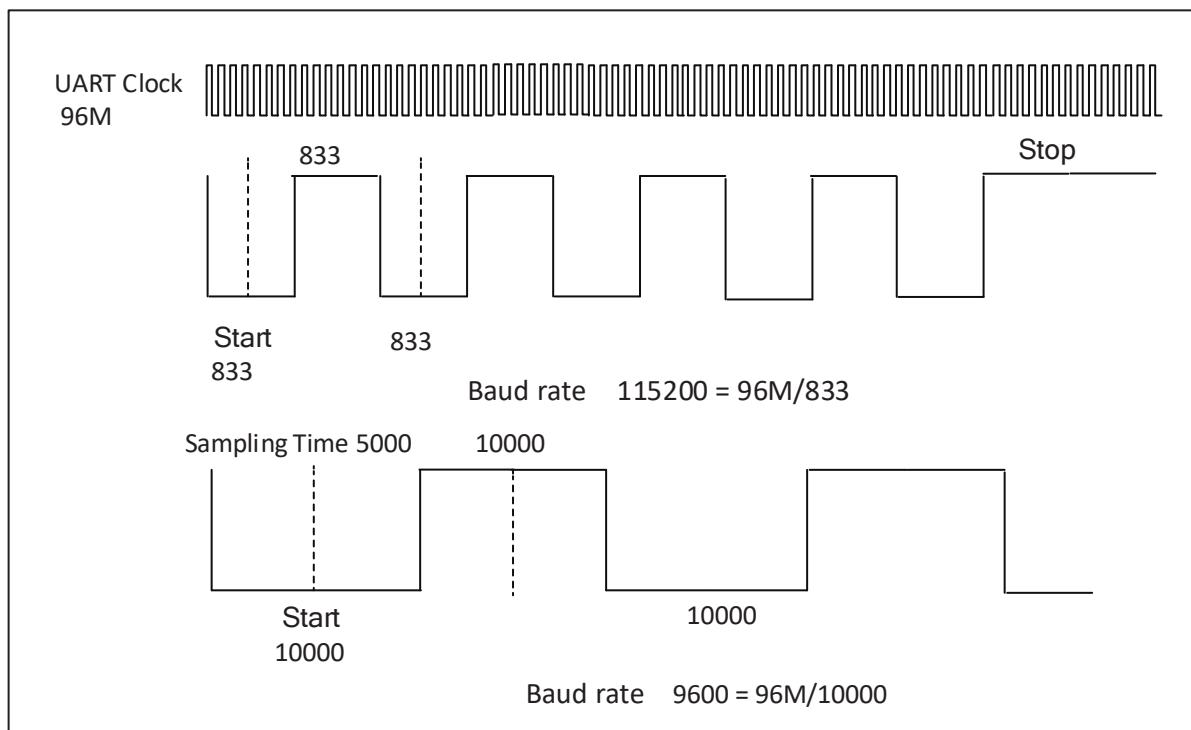


图 10.3: UART 采样波形图

### 10.3.5 发送器

发送器包含一个 128 字节的发送 FIFO，用来存放待发送的数据。软件可以通过 APB 总线写 TX FIFO，也可以通过 DMA 将数据搬入 TX FIFO。当发送使能位被设置时，FIFO 中存放的数据会从 TX 引脚输出。软件可以选择通过 DMA 或 APB 总线这两种方式将数据传入 TX FIFO。软件可以通过寄存器 uart\_fifo\_config\_1 中的 tx\_fifo\_cnt 查询 TX FIFO 剩余可用空间计数值来检查发送器的状态。

发送器的自由运行（FreeRun）模式如下：

- 如果没有开启自由运行（FreeRun）模式，则当发送字节达到指定长度时发送行为终止并产生中断，如果要继续发送则需重新关闭再使能发送使能位。
- 如果开启自由运行（FreeRun）模式，则当 TX FIFO 里存在数据时，发送器就会进行发送，不会因为发送字节达到

指定长度而终止。

### 10.3.6 接收器

接收器包含一个 128 字节的接收 FIFO，用来存放接收到的数据。软件可以通过寄存器 UART\_FIFO\_CONFIG\_1 的位 <RFICNT> 查询 RX FIFO 可用数据计数值来检查接收器的状态。寄存器 URX\_RTO\_TIMER 的低 8 位用于设定一个接收超时门限，当接收器超过该时间值未收到数据时，会触发中断。寄存器 URX\_CONFIG 的位 <DEGEN> 和 <DEGCNT> 用于使能去毛刺功能和设置门限值，其控制的是 UART 采样之前的滤波部分，UART 会将波形当中宽度低于门限值的毛刺滤掉，然后再将其送去采样。

### 10.3.7 自动波特率检测

UART 模块支持自动波特率检测，该检测分为两种，一种是通用模式，一种是固定字符(方波)模式。由寄存器 urx\_config 中的 cr\_urx\_abr\_en 控制，每次开启时，这两种检测模式都会启用。

通用模式

对于所接收到的任意字符数据，UART 模块会计数起始位宽当中的时钟数，此数字会接着被写入寄存器 STS\_URX\_ABR\_PRD 的低 16 位并用以计算波特率，因此当最先接收到的数据位为 1 时即可得到正确的波特率，如 LSB-FIRST 下的'0x01'。

固定字符模式

该模式下，UART 模块在计数起使位宽当中的时钟数后，会继续计数后续数据位的时钟数，并与起始位相比较，如果上下浮动在允许误差范围内，则通过检测，否则计数值会被丢弃。因此，只有在 LSB-FIRST 下接收到固定字符'0x55'/'0xD5' 或 MSB-FIRST 下的'0xAA'/'0xAB' 时，UART 模块才会将起使位宽当中的时钟数计数值写入寄存器 STS\_URX\_ABR\_PRD 的高 16 位。如下图所示：

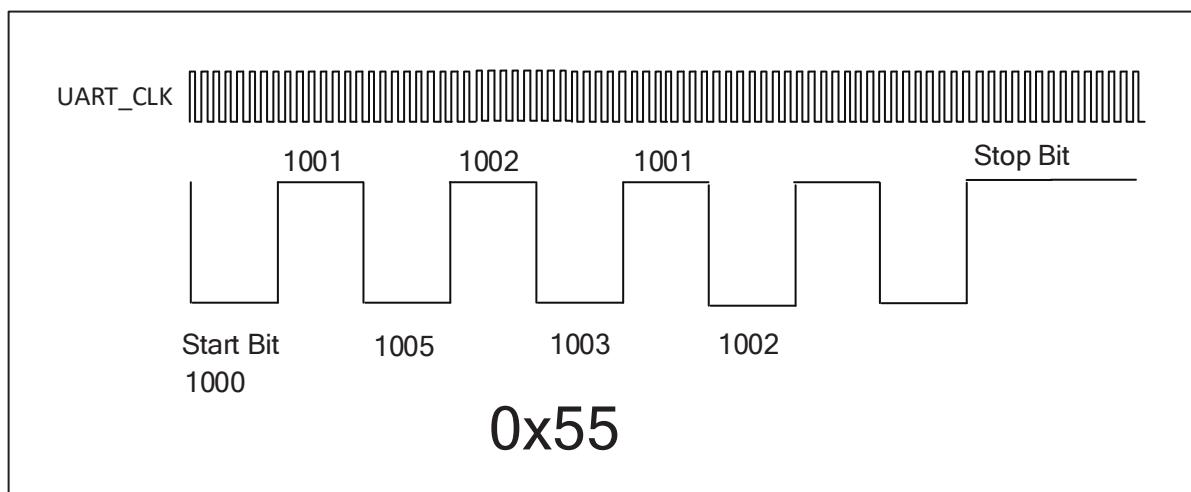


图 10.4: UART 固定字符模式波形图

如上图所示，假设设置的最大允许误差为 4，则对于接收到的某一波特率未知的数据，UART 用 UART\_CLK 去计数起始位的位宽为 1000，第二位的位宽为 1001，与前一位宽上下浮动不超过 4 个 UART\_CLK，则 UART 会继续计数第三位，第三位为 1005，与起始位相差超过 4，则检测不通过，数据丢弃。UART 会依次将数据位的前 6 位位宽与起始

位进行比较。

计算检测到的波特率的公式如下：

$$\text{Baudrate} = \frac{\text{UART\_clk}}{\text{Count} + 1}$$

### 10.3.8 硬件流控

UART 支持 CTS/RTS 方式的硬件流控，以防止 FIFO 里的数据由于来不及处理而丢失。硬件流控连接如下图所示：

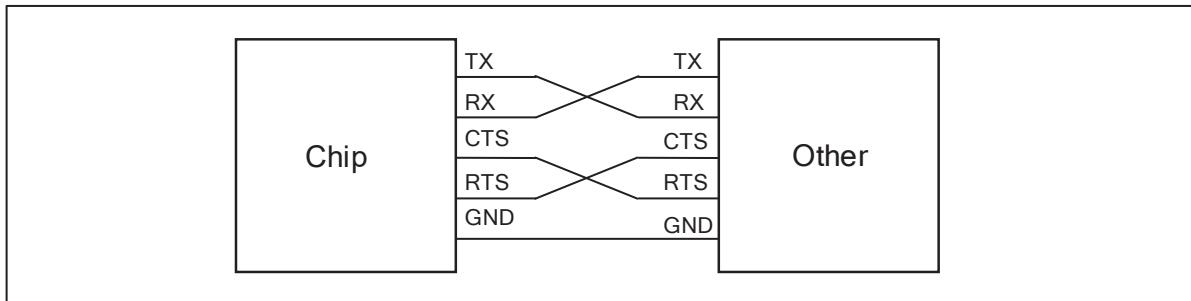


图 10.5: UART 硬件流控图

当使用硬件流控功能时，输出信号 RTS 为低电平表示请求对方发送数据，RTS 为高电平表示通知对方中止数据发送直到 RTS 恢复为低电平。发送器的硬件流控有两种方式。

- 硬件控制方式（寄存器 `uart_sw_mode` 中的 `cr_urx_rts_sw_mode` 等于 0）：接收未使能（寄存器 `urx_config` 中的 `cr_urx_en` 为 0）或 RX FIFO 剩余可用空间为一个字节时 RTS 拉高。
- 软件控制方式（寄存器 `uart_sw_mode` 中的 `cr_urx_rts_sw_mode` 等于 1）：软件可以通过配置寄存器 `urx_sw_mode` 中的 `cr_urx_rts_sw_val` 改变 RTS 的电平。

通过配置寄存器 `utx_config` 的位 `<cr_utx_cts_en>`，可以使能 TX CTS。当设备检测到输入信号 CTS 拉高时，TX 会停止发送数据，直到检测到 CTS 拉低时再继续发送。

### 10.3.9 LIN 传输模式

当发送器需使用 LIN 传输模式时，可以通过配置 `<cr_utx_lin_en>` 来送出间隔（BREAK）栏位以及同步（SYNC）栏位，间隔栏位的宽度由 `<cr_utx_bit_cnt_b>` 决定。而当接收器需使用 LIN 传输模式时，则可以通过配置 `<cr_urx_lin_en>` 来检测间隔栏位以及同步栏位，并在同步栏位格式错误时触发对应的中断 `<urx_lse_int>`。

### 10.3.10 DMA 传输模式

UART 支持 DMA 传输。使用 DMA 传输，需要通过寄存器 `uart_fifo_config_1` 中的 `tx_fifo_th` 和 `rx_fifo_th` 分别设置 TX 和 RX FIFO 的阈值。当该模式使能后，如果 `uart_fifo_config_1` 中的 `tx_fifo_cnt` 大于 `tx_fifo_th`，则会触发 DMA TX 请求，配置好 DMA 后，DMA 在收到该请求时，会按照设定从内存中将数据搬运到 TX FIFO；如果 `uart_fifo_config_1` 中的 `rx_fifo_cnt` 大于 `rx_fifo_th`，则会触发 DMA RX 请求，配置好 DMA 后，DMA 在收到该请求时，会按照设定将 RX FIFO 的数据搬运到内存。

### 10.3.11 UART 中断

UART 有着丰富的中断控制，包括以下几种中断模式：

- TX 传输结束中断
- RX 传输结束中断
- TX FIFO 请求中断
- RX FIFO 请求中断
- RX 超时中断
- RX 奇偶校验错误中断
- TX FIFO 溢出中断
- RX FIFO 溢出中断
- RX LIN 模式同步栏位 (SYNC Field) 错误中断

TX 和 RX 可以通过寄存器 `utx_config` 和 `urx_config` 的高 16 位分别设置一个传输长度值，当传输的字节数达到这个数值时，就会触发对应的 TX/RX 传输结束中断。TX/RX FIFO 请求中断会在其 FIFO 可用计数值大于寄存器 `uart_fifo_config_1` 中所设定的阈值时触发，当条件不满足时该中断标志会自动清除。**RX** 超时中断会在接收器超过超时门限值未收到数据时触发，而 **RX** 奇偶校验错误中断会发生在奇偶校验出错时。如果 TX/RX FIFO 发生了上溢或者下溢，会触发对应的溢出中断，当 FIFO 清除位 `TFICLR`/`RFICLR` 被置 1 时，对应的 FIFO 会被清空，同时溢出中断标志会自动清除。当启用 LIN 模式时，同步栏位 (SYNC Field) 依协议应为 0x55，因此当接收到的数据非 0x55 时，即触发同步栏位错误中断。可以通过寄存器 `uart_int_sts` 查询各中断状态，通过向寄存器 `uart_int_clear` 相应的位写 1 清除中断。

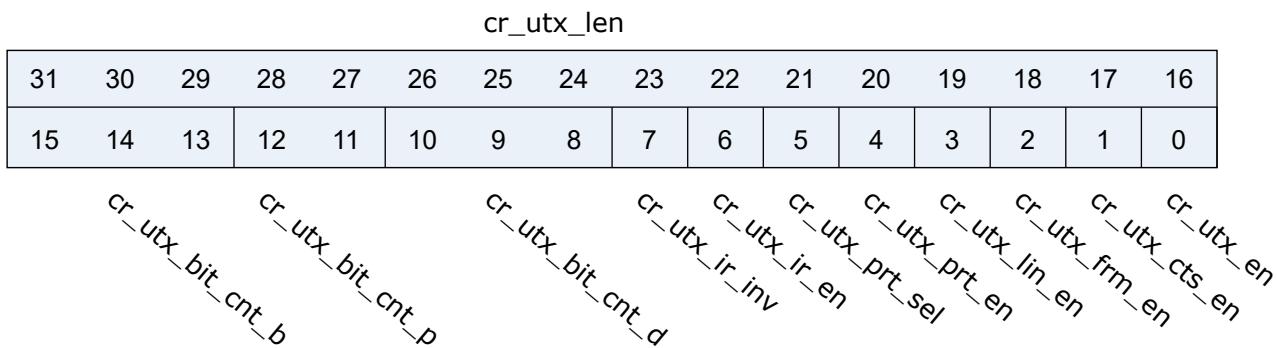
## 10.4 寄存器描述

名称	描述
<code>utx_config</code>	UART TX configuration register
<code>urx_config</code>	UART RX configuration register
<code>uart_bit_prd</code>	UART period control register
<code>data_config</code>	UART data configuration register
<code>utx_ir_position</code>	UART TX ir position control register
<code>urx_ir_position</code>	UART RX ir position control register
<code>urx_rto_timer</code>	RTO interrupt control register
<code>uart_sw_mode</code>	UART SW mode configuration register
<code>uart_int_sts</code>	UART interrupt status

名称	描述
uart_int_mask	UART interrupt mask
uart_int_clear	UART interrupt clear
uart_int_en	UART interrupt enable
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

地址: 0x4000a000

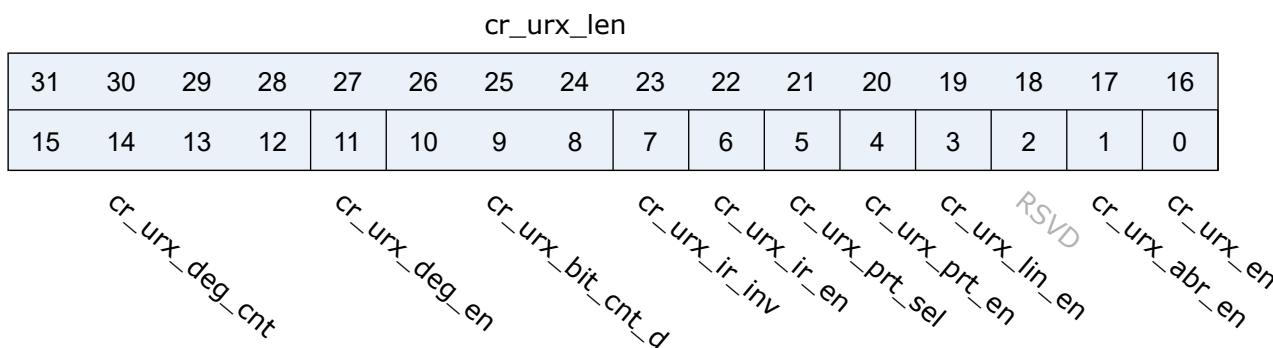


位	名称	权限	复位值	描述
31:16	cr_utx_len	r/w	16'd0	Length of UART TX data transfer (Unit: character/byte) (Don't-care if cr_utx_frm_en is enabled)
15:13	cr_utx_bit_cnt_b	r/w	3'd4	UART TX BREAK bit count (for LIN protocol) Note: Additional 8 bit times will be added since LIN Break field requires at least 13 bit times
12:11	cr_utx_bit_cnt_p	r/w	2'd1	UART TX STOP bit count (unit: 0.5 bit)
10:8	cr_utx_bit_cnt_d	r/w	3'd7	UART TX DATA bit count for each character
7	cr_utx_ir_inv	r/w	1'b0	Inverse signal of UART TX output in IR mode
6	cr_utx_ir_en	r/w	1'b0	Enable signal of UART TX IR mode

位	名称	权限	复位值	描述
5	cr_utx_prt_sel	r/w	1'b0	Select signal of UART TX parity bit 1: Odd parity 0: Even parity
4	cr_utx_prt_en	r/w	1'b0	Enable signal of UART TX parity bit
3	cr_utx_lin_en	r/w	1'b0	Enable signal of UART TX LIN mode (LIN header will be sent before sending data)
2	cr_utx_frm_en	r/w	1'b0	Enable signal of UART TX freerun mode (utx_end_int will be disabled)
1	cr_utx_cts_en	r/w	1'b0	Enable signal of UART TX CTS flow control function
0	cr_utx_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

地址: 0x4000a004



位	名称	权限	复位值	描述
31:16	cr_urx_len	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	cr_urx_deg_cnt	r/w	4'd0	De-glitch function cycle count
11	cr_urx_deg_en	r/w	1'b0	Enable signal of RXD input de-glitch function
10:8	cr_urx_bit_cnt_d	r/w	3'd7	UART RX DATA bit count for each character
7	cr_urx_ir_inv	r/w	1'b0	Inverse signal of UART RX input in IR mode
6	cr_urx_ir_en	r/w	1'b0	Enable signal of UART RX IR mode

位	名称	权限	复位值	描述
5	cr_urx_prt_sel	r/w	1'b0	Select signal of UART RX parity bit 1: Odd parity 0: Even parity
4	cr_urx_prt_en	r/w	1'b0	Enable signal of UART RX parity bit
3	cr_urx_lin_en	r/w	1'b0	Enable signal of UART RX LIN mode (LIN header will be required and checked before receiving data)
2	RSVD			
1	cr_urx_abr_en	r/w	1'b0	Enable signal of UART RX Auto Baud Rate detection function
0	cr_urx_en	r/w	1'b0	Enable signal of UART RX function

#### 10.4.3 uart\_bit\_prd

地址: 0x4000a008

cr\_urx\_bit\_prd

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

cr\_utx\_bit\_prd

位	名称	权限	复位值	描述
31:16	cr_urx_bit_prd	r/w	16'd255	Period of each UART RX bit, related to baud rate
15:0	cr_utx_bit_prd	r/w	16'd255	Period of each UART TX bit, related to baud rate

#### 10.4.4 data\_config

地址: 0x4000a00c

RSVD RSVD

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

RSVD cr\_uart\_bit\_inv

位	名称	权限	复位值	描述
31:1	RSVD			
0	cr_uart_bit_inv	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

地址: 0x4000a010

cr\_utx\_ir\_pos\_p

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

cr\_utx\_ir\_pos\_s

位	名称	权限	复位值	描述
31:16	cr_utx_ir_pos_p	r/w	16'd159	STOP position of UART TX IR pulse
15:0	cr_utx_ir_pos_s	r/w	16'd112	START position of UART TX IR pulse

#### 10.4.6 urx\_ir\_position

地址: 0x4000a014

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

cr\_urx\_ir\_pos\_s

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	cr_urx_ir_pos_s	r/w	16'd111	START position of UART RXD pulse recovered from IR signal

#### 10.4.7 urx\_rto\_timer

地址: 0x4000a018

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

cr\_urx\_rto\_value

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	cr_urx_rto_value	r/w	8'd15	Time-out value for triggering RTO interrupt (unit: bit time)

#### 10.4.8 uart\_sw\_mode

地址: 0x4000a01c

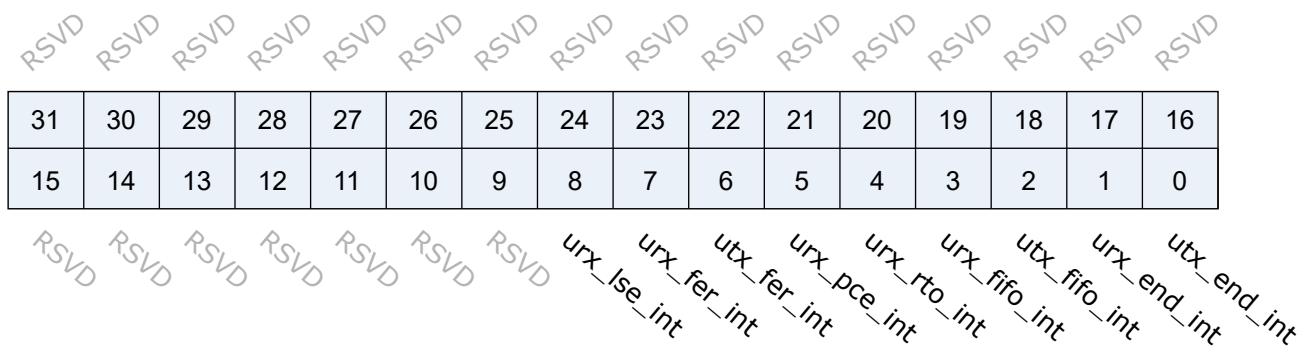
RSVD																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD																

cr\_urx\_rts\_sw\_val  
cr\_urx\_rts\_sw\_mode  
cr\_utx\_txd\_sw\_val  
cr\_utx\_txd\_sw\_mode

位	名称	权限	复位值	描述
31:4	RSVD			
3	cr_urx_rts_sw_val	r/w	1'b0	UART RX RTS output SW control value
2	cr_urx_rts_sw_mode	r/w	1'b0	UART RX RTS output SW control mode
1	cr_utx_txd_sw_val	r/w	1'b0	UART TX TXD output SW control value
0	cr_utx_txd_sw_mode	r/w	1'b0	UART TX TXD output SW control mode

### 10.4.9 uart\_int\_sts

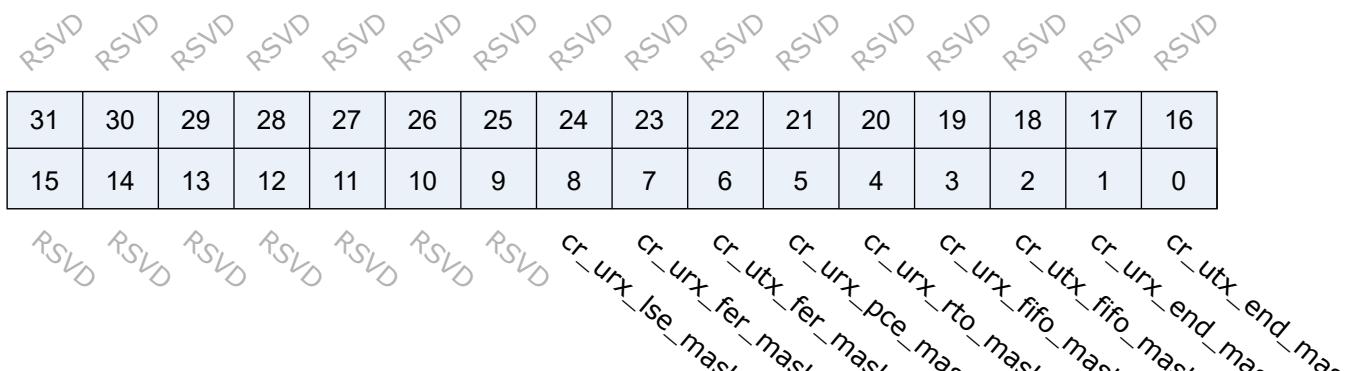
地址: 0x4000a020



位	名称	权限	复位值	描述
31:9	RSVD			
8	urx_lse_int	r	1'b0	UART RX LIN mode sync field error interrupt
7	urx_fer_int	r	1'b0	UART RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
6	utx_fer_int	r	1'b0	UART TX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
5	urx_pce_int	r	1'b0	UART RX parity check error interrupt
4	urx_rto_int	r	1'b0	UART RX Time-out interrupt
3	urx_fifo_int	r	1'b0	UART RX FIFO ready ( $rx\_fifo\_cnt > rx\_fifo\_th$ ) interrupt, auto-cleared when data is popped
2	utx_fifo_int	r	1'b0	UART TX FIFO ready ( $tx\_fifo\_cnt > tx\_fifo\_th$ ) interrupt, auto-cleared when data is pushed
1	urx_end_int	r	1'b0	UART RX transfer end interrupt (set according to cr_urx_len)
0	utx_end_int	r	1'b0	UART TX transfer end interrupt (set according to cr_utx_len)

#### 10.4.10 uart\_int\_mask

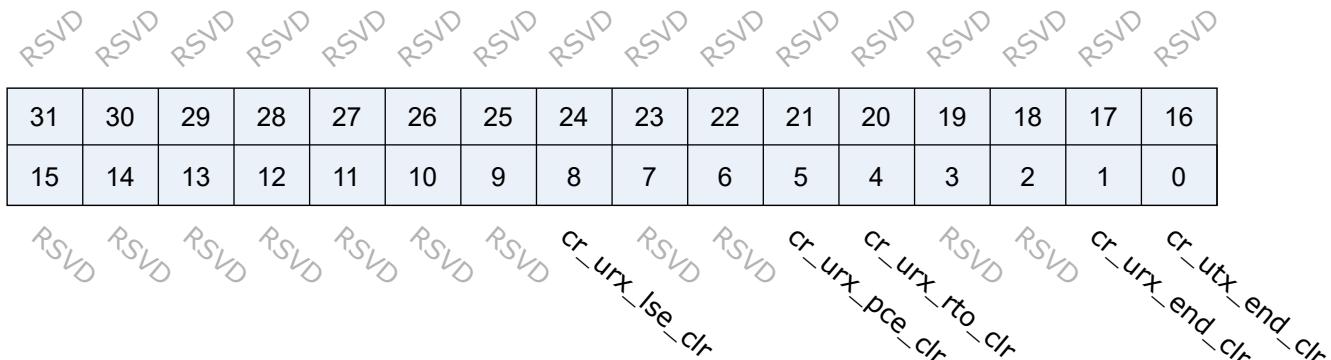
地址: 0x4000a024



位	名称	权限	复位值	描述
31:9	RSVD			
8	cr_urx_lse_mask	r/w	1'b1	Interrupt mask of urx_lse_int
7	cr_urx_fer_mask	r/w	1'b1	Interrupt mask of urx_fer_int
6	cr_utx_fer_mask	r/w	1'b1	Interrupt mask of utx_fer_int
5	cr_urx_pce_mask	r/w	1'b1	Interrupt mask of urx_pce_int
4	cr_urx_rto_mask	r/w	1'b1	Interrupt mask of urx_rto_int
3	cr_urx_fifo_mask	r/w	1'b1	Interrupt mask of urx_fifo_int
2	cr_utx_fifo_mask	r/w	1'b1	Interrupt mask of utx_fifo_int
1	cr_urx_end_mask	r/w	1'b1	Interrupt mask of urx_end_int
0	cr_utx_end_mask	r/w	1'b1	Interrupt mask of utx_end_int

#### 10.4.11 uart\_int\_clear

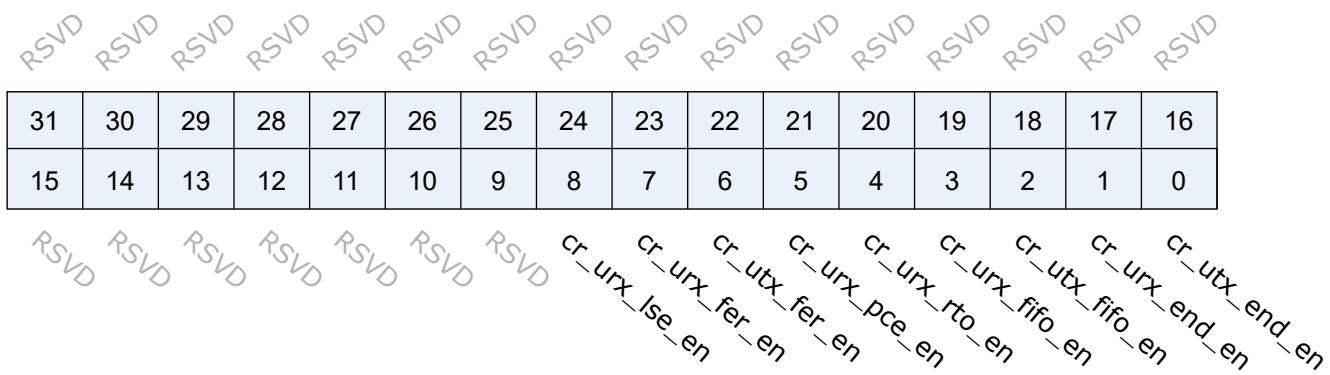
地址: 0x4000a028



位	名称	权限	复位值	描述
31:9	RSVD			
8	cr_urx_lse_clr	w1c	1'b0	Interrupt clear of urx_lse_int
7:6	RSVD			
5	cr_urx_pce_clr	w1c	1'b0	Interrupt clear of urx_pce_int
4	cr_urx_rto_clr	w1c	1'b0	Interrupt clear of urx_rto_int
3:2	RSVD			
1	cr_urx_end_clr	w1c	1'b0	Interrupt clear of urx_end_int
0	cr_utx_end_clr	w1c	1'b0	Interrupt clear of utx_end_int

#### 10.4.12 uart\_int\_en

地址: 0x4000a02c



位	名称	权限	复位值	描述
31:9	RSVD			
8	cr_urx_lse_en	r/w	1'b1	Interrupt enable of urx_lse_int
7	cr_urx_fer_en	r/w	1'b1	Interrupt enable of urx_fer_int
6	cr_utx_fer_en	r/w	1'b1	Interrupt enable of utx_fer_int
5	cr_urx_pce_en	r/w	1'b1	Interrupt enable of urx_pce_int
4	cr_urx_rto_en	r/w	1'b1	Interrupt enable of urx_rto_int
3	cr_urx_fifo_en	r/w	1'b1	Interrupt enable of urx_fifo_int
2	cr_utx_fifo_en	r/w	1'b1	Interrupt enable of utx_fifo_int
1	cr_urx_end_en	r/w	1'b1	Interrupt enable of urx_end_int
0	cr_utx_end_en	r/w	1'b1	Interrupt enable of utx_end_int

### 10.4.13 uart\_status

地址: 0x4000a030

位	名称	权限	复位值	描述
31:2	RSVD			
1	sts_urx_bus_busy	r	1'b0	Indicator of UART RX bus busy
0	sts_utx_bus_busy	r	1'b0	Indicator of UART TX bus busy

#### **10.4.14 sts\_urx\_abr\_prd**

地址: 0x4000a034

sts urx abr prd 0x55

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

sts urx abr prd start

位	名称	权限	复位值	描述
31:16	sts_urx_abr_prd_0x55	r	16'd0	Bit period of Auto Baud Rate detection using codeword 0x55
15:0	sts_urx_abr_prd_start	r	16'd0	Bit period of Auto Baud Rate detection using START bit

#### 10.4.15 uart\_fifo\_config\_0

地址: 0x4000a080

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

RSVD RSVD

rx\_fifo\_underflow rx\_fifo\_overflow tx\_fifo\_underflow tx\_fifo\_overflow tx\_fifo\_clr uart\_dma\_rx\_en

位	名称	权限	复位值	描述
31:8	RSVD			
7	rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	uart_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	uart_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

#### 10.4.16 uart\_fifo\_config\_1

地址: 0x4000a084

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

rx\_fifo\_cnt tx\_fifo\_cnt

位	名称	权限	复位值	描述
31	RSVD			

位	名称	权限	复位值	描述
30:24	rx_fifo_th	r/w	7'd0	RX FIFO threshold, dma_rx_req will not be asserted if tx_fifo_cnt is less than this value
23	RSVD			
22:16	tx_fifo_th	r/w	7'd0	TX FIFO threshold, dma_tx_req will not be asserted if tx_fifo_cnt is less than this value
15:8	rx_fifo_cnt	r	8'd0	RX FIFO available count
7:0	tx_fifo_cnt	r	8'd128	TX FIFO available count

#### 10.4.17 uart\_fifo\_wdata

地址: 0x4000a088

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

uart\_fifo\_wdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	uart_fifo_wdata	w	x	

#### 10.4.18 uart\_fifo\_rdata

地址: 0x4000a08c

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

uart\_fifo\_rdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	uart_fifo_rdata	r	8'h0	

# 11

I2C

## 11.1 简介

I2C (Inter-Integrated Circuit) 是一种串行通讯总线，使用多主从架构，用来连接低速外围装置。每个器件都有一个唯一的识别地址，并且都可以作为一个发送器或接收器。如果有两个或多个主机同时初始化，数据传输可以通过冲突检测和仲裁防止数据被破坏。BL702 包含一个 I2C 控制器主机，可灵活配置 `slaveAddr`、`subAddr` 以及传输数据，方便与从设备通信，提供 2 个 word 深度的 fifo，提供中断功能，可搭配 DMA 使用提高效率，可灵活调整时钟频率。

## 11.2 主要特征

- 支持主机模式
- 支持多主机模式和仲裁功能
- 时钟频率可灵活调整

## 11.3 功能描述

引脚列表：

表 11.1: I2C 引脚

名称	类型	描述
I2Cx_SCL	输入/输出	I2C 串行时钟信号
I2Cx_SDA	输入/输出	I2C 串行数据信号

### 11.3.1 起始和停止条件

所有传输都由起始条件 (START condition) 开始, 以停止条件 (STOP condition) 结束。起始条件和停止条件一般都由主机产生, 总线在起始条件后被认为处于总线忙的状态, 在停止条件后的某段时间内被认为处于空闲状态。

起始条件:SCL 为高电平时 SDA 产生一高至低的电平转换;

停止条件:SCL 为高电平时 SDA 产生一低至高的电平转换。

波形示意图如下:

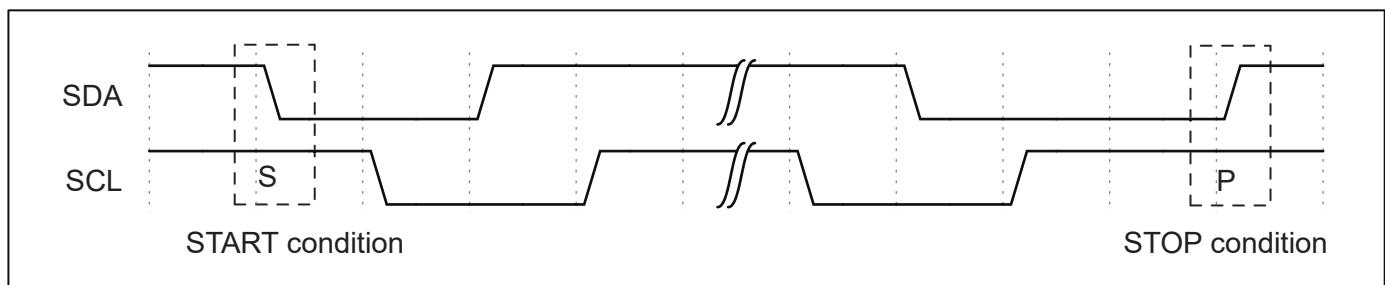


图 11.1: I2C 起始和停止条件

### 11.3.2 数据传输格式

传输的第一个 8 位为寻址字节, 包括 7 位从机地址和 1 位方向位。数据由主机发送或接收是由主机所送出的第 1 个字节的第 8 位控制, 若为 0 表示数据由主机发送; 为 1 则表示数据由主机接收, 紧接着从机发出应答位 (ACK), 在数据传输完成后, 主机发出停止信号, 波形图如下:

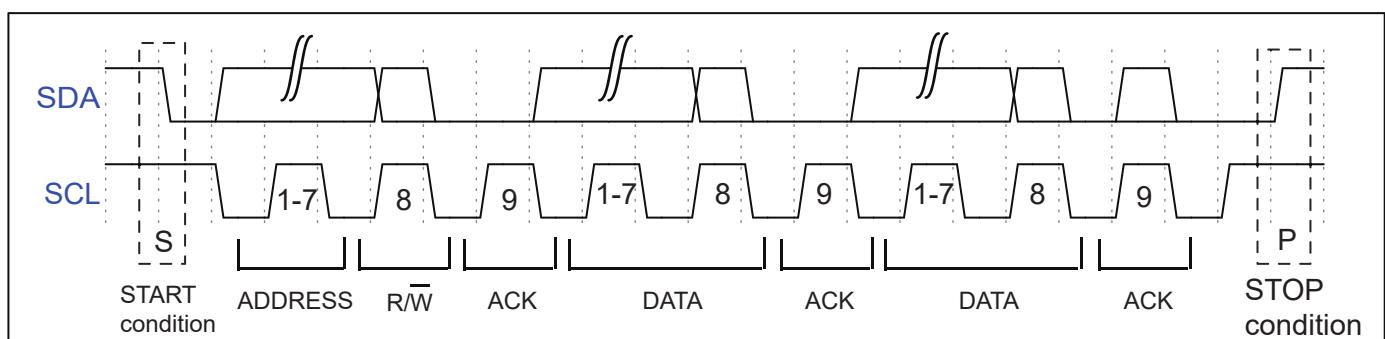


图 11.2: I2C 数据传输格式

主发送和从接收的时序

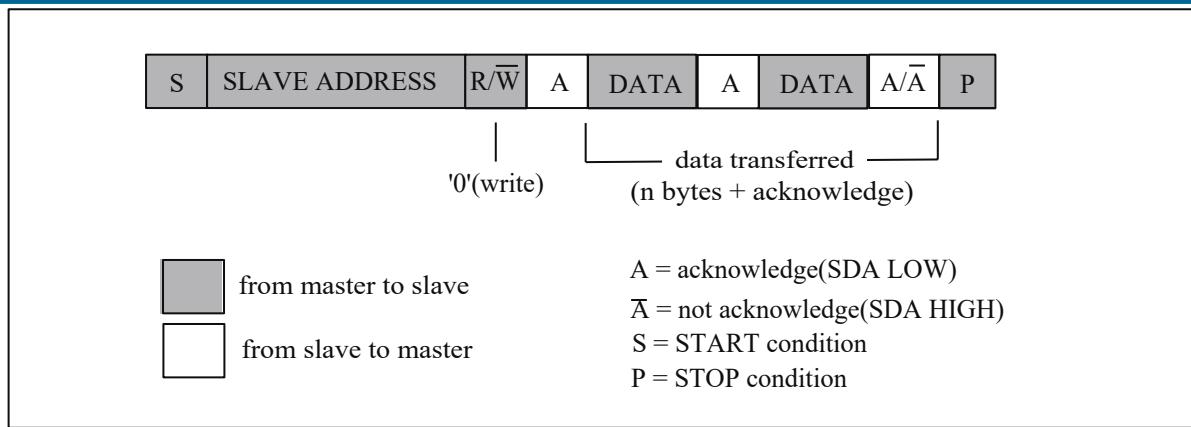


图 11.3: 主发送和从接收的时序

主接收和从发送的时序

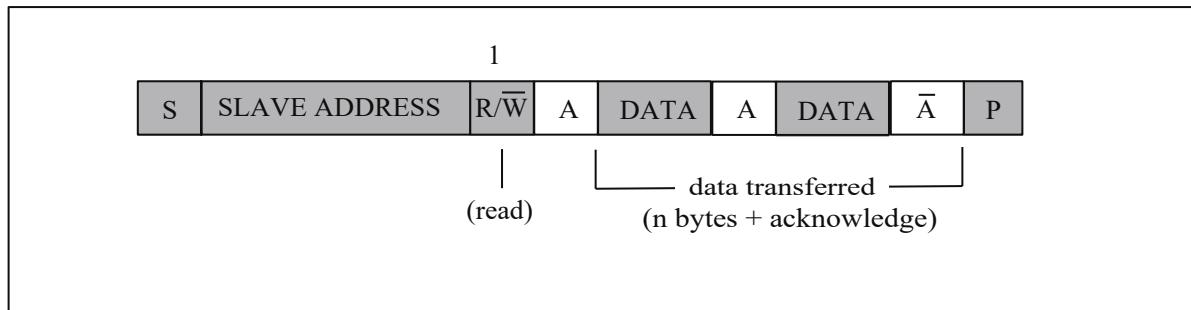


图 11.4: 主接收和从发送的时序

### 11.3.3 仲裁

当 I<sup>2</sup>C 总线存在多个主机时，可能会发生多个主机同时启动传输的情况，此时必须要依靠仲裁机制来决定哪个主机有权利完成接下来的数据传输，其余主机则须放弃对总线的控制，等到总线再次空出来后才能重新启动传输。

在传输过程中，所有主机都需要在 SCL 为高电平时检查 SDA 是否与自己所想送出的资料相符，当 SDA 电平与预期不同时，表示有别的主机也在同时进行传输，而发现 SDA 电平不同的主机则失去此次仲裁，由其他主机完成数据传输。

两主机同时传输数据并启动仲裁机制的波形示意图如下：

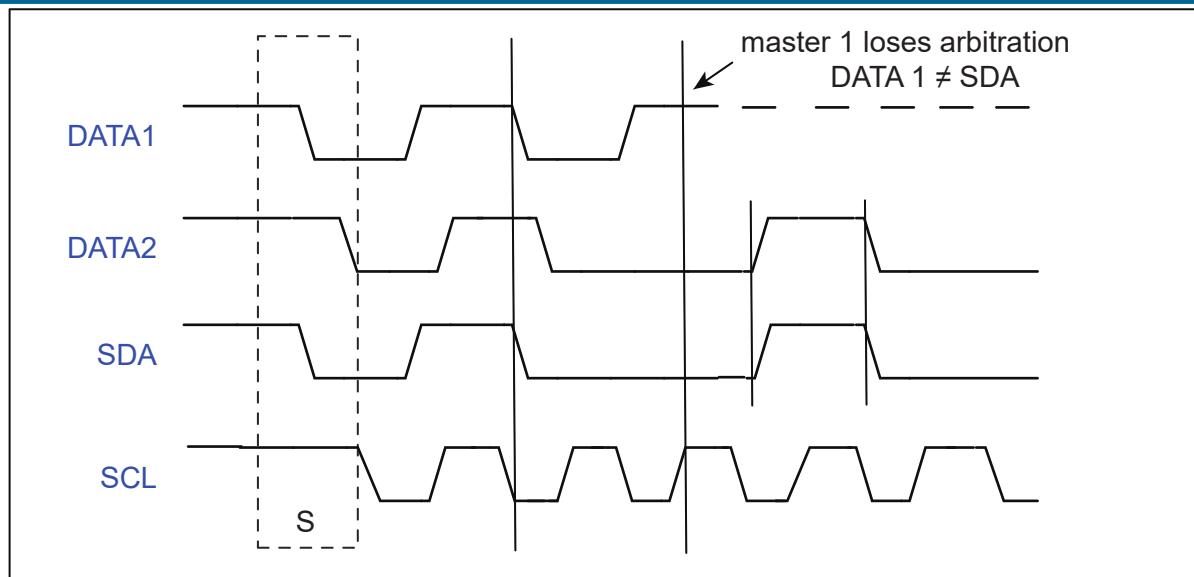


图 11.5: 同时传输数据波形示意图

## 11.4 I2C 时钟设定

I2C 的时钟是由 `bclk(bus clock)` 而来，可以在 `bclk` 时钟的基础上做分频处理。寄存器 `i2c_prd_data` 可以对数据段的时钟做分频处理。i2c 模块将数据发送分为 4 个阶段，每个阶段在寄存器中用单独一个字节来控制，每个阶段的采样个数是可以设置的，4 个采样数共同决定了 i2c clock 的分频系数。比如现在 `bclk` 是 32M，寄存器 `i2c_prd_data` 在不做配置默认情况下的值是 `0x0f0f0f0f`，那么 I2C 的时钟频率为  $32M/(15 + 1) * 4 = 500K$ 。同理，寄存器 `i2c_prd_start` 和 `i2c_prd_stop` 也会分别对起始位和停止位的时钟做分频处理。

## 11.5 I2C 配置流程

### 11.5.1 配置项

- 读写标志位
- 从设备地址
- 从设备寄存器地址
- 从设备寄存器地址长度
- 数据 (发送时，配置发送的数据；接收时，存储接收到的数据)
- 数据长度
- 使能信号

## 11.5.2 读写标志位

I2C 支持发送和接收两种工作状态，寄存器 cr\_i2c\_pkt\_dir 表示发送或者接收状态，设置为 0 时，表示发送状态，设置为 1 时，表示接收状态。

## 11.5.3 从设备地址

每个对接 I2C 的从设备，都会有唯一设备地址，通常该地址是 7 位长度，将从设备地址写入寄存器 cr\_i2c\_slv\_addr。I2C 在将从设备地址发送出去之前，会自动左移 1 位，并在最低位补上发送接收方向位。

## 11.5.4 从设备寄存器地址

从设备寄存器地址表示 I2C 需要对从设备某个寄存器做读写操作的寄存器地址。将从设备寄存器地址写入寄存器 i2c\_sub\_addr，同时需要将寄存器 cr\_i2c\_sub\_addr\_en 置 1。如果将寄存器 cr\_i2c\_sub\_addr\_en 置 0，那么 I2C 主机发送时会跳过从设备寄存器地址段。

## 11.5.5 从设备寄存器地址长度

将从设备寄存器地址长度减 1 再写入寄存器 cr\_i2c\_sub\_addr\_bc。

## 11.5.6 数据

数据部分表示需要发送到从设备的数据，或者需要从从设备接收的数据。当 I2C 发送数据时，需要将数据依次以 word 为单位写入 I2C FIFO，发送数据写 FIFO 的寄存器地址 i2c\_fifo\_wdata。当 I2C 接收数据时，需要依次以 word 为单位从 I2C FIFO 中将数据读出来，接收数据读 FIFO 的寄存器地址 i2c\_fifo\_rdata。

## 11.5.7 数据长度

将数据长度减 1 再写入寄存器 cr\_i2c\_pkt\_len。

## 11.5.8 使能信号

将以上几项配置完成后，再将使能信号寄存器 cr\_i2c\_m\_en 写 1，就自动启动 I2C 发送流程了。

当读写标志位配置为 0 时，I2C 发送数据，主机发送流程：

1. 起始位
2. (从设备地址左移 1 位 + 0) + ACK
3. 从设备寄存器地址 + ACK
4. 1 字节数据 + ACK
5. 1 字节数据 + ACK
6. 停止位

当读写标志位配置为 1 时, I2C 接收数据, 主机发送流程:

1. 起始位
2. (从设备地址左移 1 位 + 0) + ACK
3. 从设备寄存器地址 + ACK
4. 起始位
5. (从设备地址左移 1 位 + 1) + ACK
6. 1 字节数据 + ACK
7. 1 字节数据 + ACK
8. 停止位

## 11.6 FIFO 管理

I2C FIFO 深度为 2 个 word, I2C 发送和接收可分为 RX FIFO 和 TX FIFO。寄存器 i2c\_fifo\_config\_1 中的 rx\_fifo\_cnt 表示 RX FIFO 中有多少数据 (单位 word) 需要读取。寄存器 i2c\_fifo\_config\_1 中的 tx\_fifo\_cnt 表示 TX FIFO 中剩余多少空间 (单位 Word) 可供写入。

I2C FIFO 状态:

- RX FIFO underflow: 当 RX FIFO 中的数据被读取完毕或者为空, 继续从 RX FIFO 中读取数据, 则寄存器 i2c\_fifo\_config\_0 中的 rx\_fifo\_underflow 会被置 1;
- RX FIFO overflow: 当 I2C 接收数据直到 RX FIFO 的 2 个 word 被填满后, 在没有读取 RX FIFO 的情况下, I2C 再次接收到数据, 寄存器 i2c\_fifo\_config\_0 中的 rx\_fifo\_overflow 会被置 1;
- TX FIFO underflow: 当向 TX FIFO 中填入的数据大小不满足配置的 I2C 数据长度 (i2c\_config 中的 cr\_i2c\_pkt\_len), 并且已经没有新数据继续填入 TX FIFO 中时, 寄存器 i2c\_fifo\_config\_0 中的 tx\_fifo\_underflow 会被置 1;
- TX FIFO overflow: 当 TX FIFO 的 2 个 word 被填满后, 在 TX FIFO 中的数据没有发出去之前, 再次向 TX FIFO 中填入数据, 寄存器 i2c\_fifo\_config\_0 中的 tx\_fifo\_overflow 会被置 1。

## 11.7 搭配使用 DMA

I2C 可以使用 DMA 进行数据的发送和接收。将寄存器 i2c\_fifo\_config\_0 中的 i2c\_dma\_tx\_en 置 1, 则开启 DMA 发送模式, 为 I2C 分配好 DMA 通道后, DMA 会将数据从存储区搬运到 i2c\_fifo\_wdata 寄存器中。将寄存器 i2c\_fifo\_config\_0 中的 i2c\_dma\_rx\_en 置 1, 则开启 DMA 接收模式, 为 I2C 分配好 DMA 通道后, DMA 会将 i2c\_fifo\_rdata 寄存器中的数据搬运到存储区中。I2C 模块使用 DMA 功能时, 数据部分将由 DMA 自动完成搬运, 不需要 CPU 再将数据写入 I2C TX FIFO 或者从 I2C RX FIFO 中读出。

## 11.7.1 DMA 发送流程

1. 配置读写标志位 i2c\_config[cr\_i2c\_pkt\_dir] 为 0
2. 配置从设备地址 i2c\_config[cr\_i2c\_slv\_addr]
3. 如有有从设备寄存器地址，则配置从设备寄存器地址 i2c\_sub\_addr、从设备寄存器地址长度 i2c\_config[cr\_i2c\_sub\_addr\_bc]，配置从设备寄存器地址使能位 i2c\_config[cr\_i2c\_sub\_addr\_en] 为 1
4. 设置发送数据的长度 i2c\_config[cr\_i2c\_pkt\_len]
5. 使能 DMA 模式发送，设置 i2c\_fifo\_config\_0[i2c\_dma\_tx\_en] 为 1
6. 配置 DMA 传输的数据长度 DMA\_CxControl[TransferSize] (x=0~7)
7. 配置 DMA 源地址 DMA\_CxSrcAddr、数据宽度 DMA\_CxControl[SWidth]、burst 大小 DMA\_CxControl[SBSIZE]，将 DMA\_CxControl[SI] 置 1 开启地址自动累加模式
8. 配置 DMA 目的地地址 DMA\_CxDstAddr 为 i2c\_fifo\_wdata、数据宽度 DMA\_CxControl[DWidth] 为 32 和 burst 大小 DMA\_CxControl[DBSSize]，将 DMA\_CxControl[DI] 清零禁用地址自动累加模式
9. 使能 DMA
10. 配置 i2c\_config[cr\_i2c\_m\_en] 为 1 使能 I2C

## 11.7.2 DMA 接收流程

1. 配置读写标志位 i2c\_config[cr\_i2c\_pkt\_dir] 为 1
2. 配置从设备地址 i2c\_config[cr\_i2c\_slv\_addr]
3. 如有有从设备寄存器地址，则配置从设备寄存器地址 i2c\_sub\_addr、从设备寄存器地址长度 i2c\_config[cr\_i2c\_sub\_addr\_bc]，配置从设备寄存器地址使能位 i2c\_config[cr\_i2c\_sub\_addr\_en] 为 1
4. 设置接收数据的长度 i2c\_config[cr\_i2c\_pkt\_len]
5. 使能 DMA 模式接收，设置 i2c\_fifo\_config\_0[i2c\_dma\_rx\_en] 为 1
6. 配置 DMA 传输的数据长度 DMA\_CxControl[TS] (x=0~7)
7. 配置 DMA 源地址 DMA\_CxSrcAddr 为 i2c\_fifo\_rdata、数据宽度 DMA\_CxControl[SWidth] 为 32 和 burst 大小 DMA\_CxControl[SBSIZE]，将 DMA\_CxControl[SI] 清零禁用地址自动累加模式
8. 配置 DMA 目的地地址 DMA\_CxDstAddr、数据宽度 DMA\_CxControl[DWidth]、burst 大小 DMA\_CxControl[DBSSize]，将 DMA\_CxControl[DI] 置 1 开启地址自动累加模式
9. 使能 DMA
10. 配置 i2c\_config[cr\_i2c\_m\_en] 为 1 使能 I2C

## 11.8 中断

I2C 包括如下几种中断：

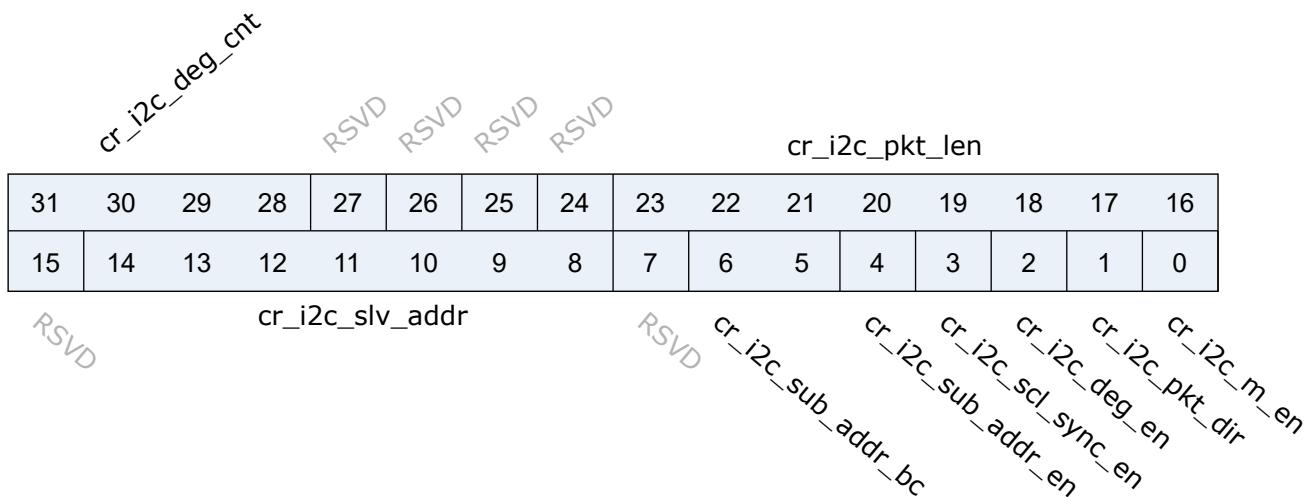
- I2C\_TRANS\_END\_INT: I2C 传输结束中断
- I2C\_TX\_FIFO\_READY\_INT: 当 I2C TX FIFO 有空闲空间可用于填充时，触发中断
- I2C\_RX\_FIFO\_READY\_INT: 当 I2C RX FIFO 接收到数据时，触发中断
- I2C\_NACK\_RECV\_INT: 当 I2C 模块检测到 NACK 状态，触发中断
- I2C\_ARB\_LOST\_INT: I2C 仲裁丢失中断
- I2C\_FIFO\_ERR\_INT: I2C FIFO ERROR 中断

## 11.9 寄存器描述

名称	描述
i2c_config	I2C configuration register
i2c_int_sts	I2C interrupt status
i2c_sub_addr	I2C sub-address configuration
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

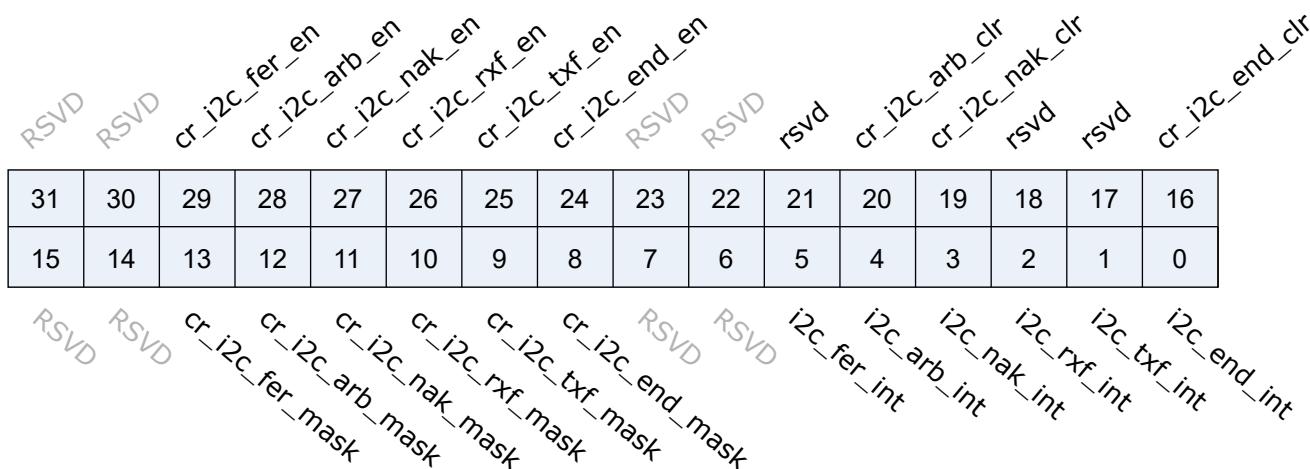
地址: 0x4000a300



位	名称	权限	复位值	描述
31:28	cr_i2c_deg_cnt	r/w	4'd0	De-glitch function cycle count
27:24	RSVD			
23:16	cr_i2c_pkt_len	r/w	8'd0	Packet length (unit: byte)
15	RSVD			
14:8	cr_i2c_slv_addr	r/w	7'd0	Slave address for I2C transaction (target address)
7	RSVD			
6:5	cr_i2c_sub_addr_bc	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	cr_i2c_sub_addr_en	r/w	1'b0	Enable signal of I2C sub-address field
3	cr_i2c_scl_sync_en	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)
2	cr_i2c_deg_en	r/w	1'b0	Enable signal of I2C input de-glitch function (for all input pins)
1	cr_i2c_pkt_dir	r/w	1'b1	Transfer direction of the packet 1'b0: Write; 1'b1: Read
0	cr_i2c_m_en	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

地址: 0x4000a304



位	名称	权限	复位值	描述
31:30	RSVD			
29	cr_i2c_fer_en	r/w	1'b1	Interrupt enable of i2c_fer_int
28	cr_i2c_arb_en	r/w	1'b1	Interrupt enable of i2c_arb_int
27	cr_i2c_nak_en	r/w	1'b1	Interrupt enable of i2c_nak_int
26	cr_i2c_rxf_en	r/w	1'b1	Interrupt enable of i2c_rxf_int
25	cr_i2c_txf_en	r/w	1'b1	Interrupt enable of i2c_txf_int
24	cr_i2c_end_en	r/w	1'b1	Interrupt enable of i2c_end_int
23:22	RSVD			
21	rsvd	rsvd	1'b0	
20	cr_i2c_arb_clr	w1c	1'b0	Interrupt clear of i2c_arb_int
19	cr_i2c_nak_clr	w1c	1'b0	Interrupt clear of i2c_nak_int
18	rsvd	rsvd	1'b0	
17	rsvd	rsvd	1'b0	
16	cr_i2c_end_clr	w1c	1'b0	Interrupt clear of i2c_end_int
15:14	RSVD			
13	cr_i2c_fer_mask	r/w	1'b1	Interrupt mask of i2c_fer_int
12	cr_i2c_arb_mask	r/w	1'b1	Interrupt mask of i2c_arb_int
11	cr_i2c_nak_mask	r/w	1'b1	Interrupt mask of i2c_nak_int
10	cr_i2c_rxf_mask	r/w	1'b1	Interrupt mask of i2c_rxf_int
9	cr_i2c_txf_mask	r/w	1'b1	Interrupt mask of i2c_txf_int

位	名称	权限	复位值	描述
8	cr_i2c_end_mask	r/w	1'b1	Interrupt mask of i2c_end_int
7:6	RSVD			
5	i2c_fer_int	r	1'b0	I2C TX/RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
4	i2c_arb_int	r	1'b0	I2C arbitration lost interrupt
3	i2c_nak_int	r	1'b0	I2C NACK-received interrupt
2	i2c_rxf_int	r	1'b0	I2C RX FIFO ready ( $rx\_fifo\_cnt > rx\_fifo\_th$ ) interrupt, auto-cleared when data is popped
1	i2c_txf_int	r	1'b0	I2C TX FIFO ready ( $tx\_fifo\_cnt > tx\_fifo\_th$ ) interrupt, auto-cleared when data is pushed
0	i2c_end_int	r	1'b0	I2C transfer end interrupt

### 11.9.3 i2c\_sub\_addr

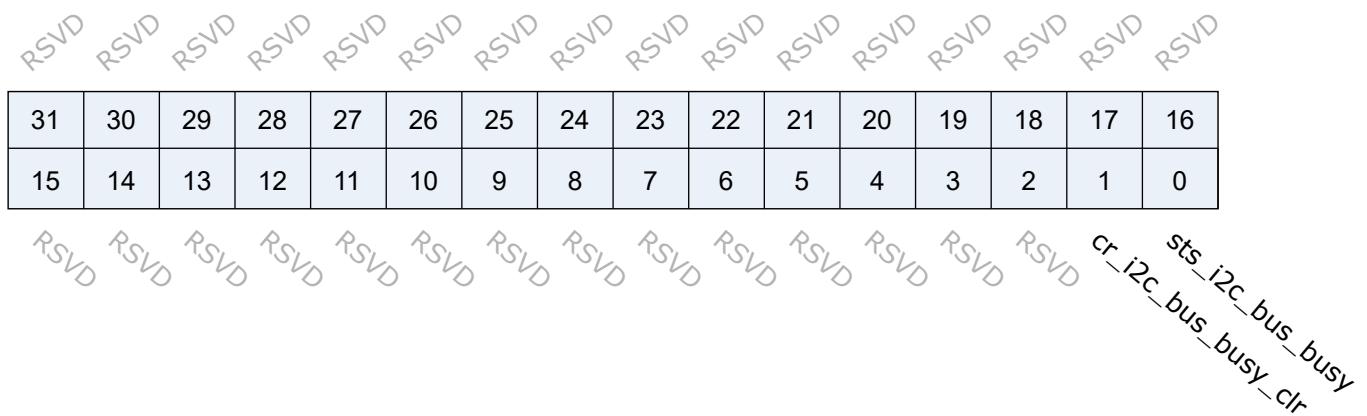
地址: 0x4000a308

cr_i2c_sub_addr_b3								cr_i2c_sub_addr_b2							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cr_i2c_sub_addr_b1								cr_i2c_sub_addr_b0							

位	名称	权限	复位值	描述
31:24	cr_i2c_sub_addr_b3	r/w	8'd0	I2C sub-address field - byte[3]
23:16	cr_i2c_sub_addr_b2	r/w	8'd0	I2C sub-address field - byte[2]
15:8	cr_i2c_sub_addr_b1	r/w	8'd0	I2C sub-address field - byte[1]
7:0	cr_i2c_sub_addr_b0	r/w	8'd0	I2C sub-address field - byte[0] (sub-address starts from this byte)

### 11.9.4 i2c\_bus\_busy

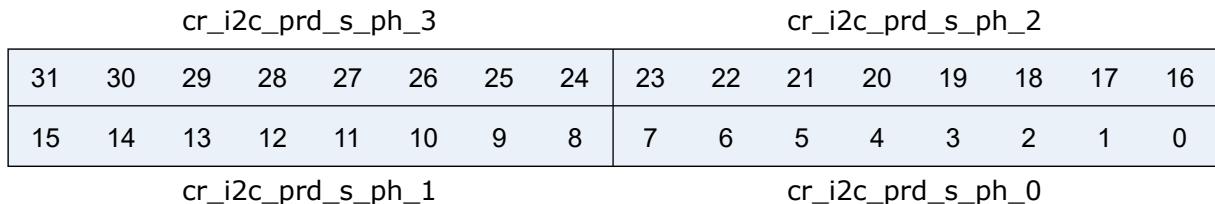
地址: 0x4000a30c



位	名称	权限	复位值	描述
31:2	RSVD			
1	cr_i2c_bus_busy_clr	w1c	1'b0	Clear signal of bus_busy status, not for normal usage (in case I2C bus hangs)
0	sts_i2c_bus_busy	r	1'b0	Indicator of I2C bus busy

### 11.9.5 i2c\_prd\_start

地址: 0x4000a310



位	名称	权限	复位值	描述
31:24	cr_i2c_prd_s_ph_3	r/w	8'd15	Length of START condition phase 3
23:16	cr_i2c_prd_s_ph_2	r/w	8'd15	Length of START condition phase 2
15:8	cr_i2c_prd_s_ph_1	r/w	8'd15	Length of START condition phase 1
7:0	cr_i2c_prd_s_ph_0	r/w	8'd15	Length of START condition phase 0

### 11.9.6 i2c\_prd\_stop

地址: 0x4000a314

cr_i2c_prd_p_ph_3								cr_i2c_prd_p_ph_2							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

位	名称	权限	复位值	描述
31:24	cr_i2c_prd_p_ph_3	r/w	8'd15	Length of STOP condition phase 3
23:16	cr_i2c_prd_p_ph_2	r/w	8'd15	Length of STOP condition phase 2
15:8	cr_i2c_prd_p_ph_1	r/w	8'd15	Length of STOP condition phase 1
7:0	cr_i2c_prd_p_ph_0	r/w	8'd15	Length of STOP condition phase 0

### 11.9.7 i2c\_prd\_data

地址: 0x4000a318

cr_i2c_prd_d_ph_3								cr_i2c_prd_d_ph_2							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

位	名称	权限	复位值	描述
31:24	cr_i2c_prd_d_ph_3	r/w	8'd15	Length of DATA phase 3
23:16	cr_i2c_prd_d_ph_2	r/w	8'd15	Length of DATA phase 2
15:8	cr_i2c_prd_d_ph_1	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	cr_i2c_prd_d_ph_0	r/w	8'd15	Length of DATA phase 0

### 11.9.8 i2c\_fifo\_config\_0

地址: 0x4000a380

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

RSVD RSVD

*rx\_fifo\_underflow rx\_fifo\_overflow tx\_fifo\_underflow tx\_fifo\_overflow tx\_fifo\_clr i2c\_dma\_rx\_en i2c\_dma\_tx\_en*

位	名称	权限	复位值	描述
31:8	RSVD			
7	rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	i2c_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	i2c_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

### 11.9.9 i2c\_fifo\_config\_1

地址: 0x4000a384

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

RSVD RSVD

*tx\_fifo\_th rx\_fifo\_th rx\_fifo\_cnt rx\_fifo\_th tx\_fifo\_th*

位	名称	权限	复位值	描述
31:25	RSVD			
24	rx_fifo_th	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	tx_fifo_th	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	rx_fifo_cnt	r	2'd0	RX FIFO available count
7:2	RSVD			
1:0	tx_fifo_cnt	r	2'd2	TX FIFO available count

### 11.9.10 i2c\_fifo\_wdata

地址: 0x4000a388

i2c\_fifo\_wdata

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

i2c\_fifo\_wdata

位	名称	权限	复位值	描述
31:0	i2c_fifo_wdata	w	x	

### 11.9.11 i2c\_fifo\_rdata

地址: 0x4000a38c

i2c\_fifo\_rdata

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

i2c\_fifo\_rdata

位	名称	权限	复位值	描述
31:0	i2c_fifo_rdata	r	32'h0	

# 12

## PWM

### 12.1 简介

脉冲宽度调制（Pulse width modulation，简称 PWM）是一种模拟控制方式，根据相应载荷的变化来调制晶体管基极或 MOS 管栅极的偏置，来实现晶体管或 MOS 管导通时间的改变，从而稳定开关电源的输出。这种方式能使电源的输出电压在工作条件变化时保持恒定，是利用微处理器的数字信号对模拟电路进行控制的一种非常有效的技术，广泛应用在从测量、通信、开关电源、电机控制等许多领域中。

### 12.2 主要特征

- 支持 5 通道 PWM 信号生成
- 三种时钟源可选择（总线时钟 `<bclk>`、晶振时钟 `<xtal>`、慢速时钟 `<f32k>`），搭配 16-bit 时钟分频器
- 双门限值设定，增加脉冲弹性

### 12.3 功能描述

#### 12.3.1 时钟与分频器

每个 PWM 计数器时钟来源都有三种选择，来源如下：

- `bclk` - 芯片的总线时钟
- `XTAL` - 外部晶振时钟
- `f32k` - 系统 RTC 时钟

每个计数器都有各自的 16-bit 分频器，PWM 计数器将以分频后的时钟作为计数周期单位，每经过一个计数周期进行计数器加一的动作。

### 12.3.2 脉冲产生原理

PWM 内部有计数器，当计数器介于可设定的两个门限值域中间时，PWM 的输出为 1，反之当计数器在设定的两个门限值之外时，PWM 输出为 0，如下图所示：

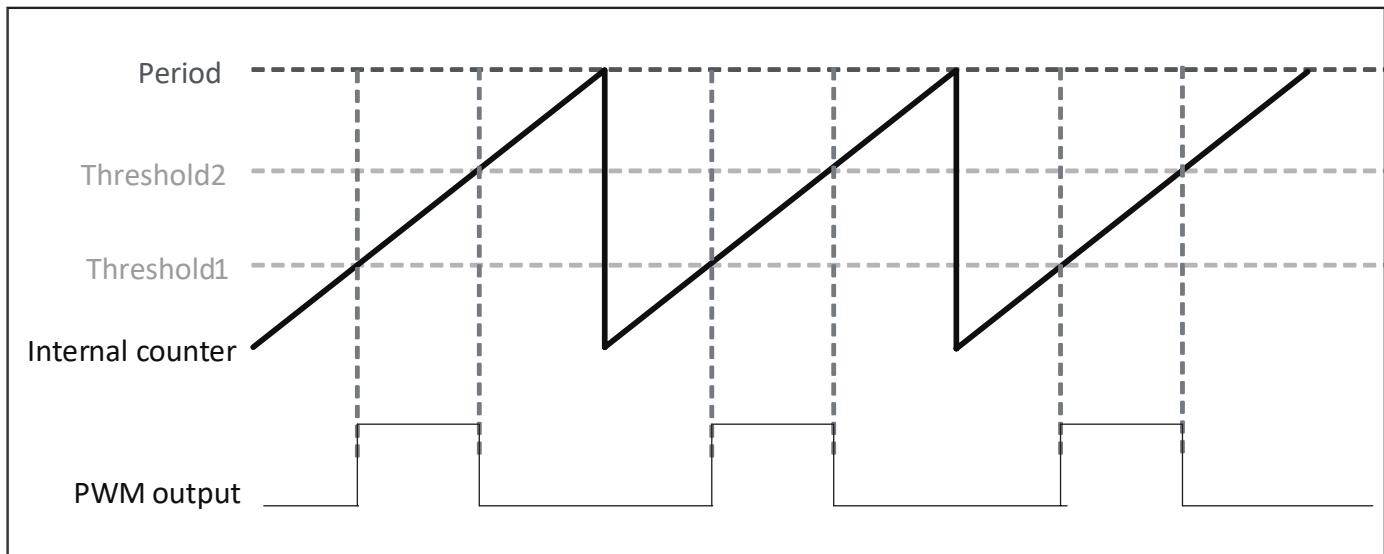


图 12.1: PWM 波形图

PWM 的周期由两部分决定，一个是时钟分频系数，一个是时钟持续周期。

时钟分频系数由寄存器 `PWMn_CLK_DIV[15:0]`(n 为 0~4) 进行设置，用于对 PWM 的源时钟进行分频。

时钟持续周期由寄存器 `PWMn_PERIOD[15:0]`(n 为 0~4) 进行设置，用于设置 PWM 的一个周期由多少个分频后的时钟周期组成。即  $\text{PWM 的周期} = \text{PWM 源时钟} / \text{PWMn_CLK_DIV[15:0]} / \text{PWMn_PERIOD[15:0]}$ 。

PWM 的占空比由时钟持续周期和两个阈值决定。第一个阈值由寄存器 `PWMn_THRE1[15:0]`(n 为 0~4) 进行设置，第二个阈值由寄存器 `PWMn_THRE2[15:0]`(n 为 0~4) 进行设置，PWM 的波形会在第一个阈值处拉高，在第二个阈值处拉低。即  $\text{PWM 的占空比} = (\text{PWMn_THRE2[15:0]} - \text{PWMn_THRE1[15:0]}) / \text{PWMn_PERIOD[15:0]}$ 。

例：若 PWM 时钟源选择为 bclk 即 72MHz，要产生 1kHz、占空比 20% 的 PWM 波，则设置如下：`PWMn_CLK_DIV[15:0]=2`

$$\text{PWMn_PERIOD[15:0]} = 72000000 / 2 / 1000 = 36000$$

$$\text{PWMn_THRE1[15:0]} = 0$$

$$\text{PWMn_THRE2[15:0]} = 0 + 36000 * 20\% = 7200$$

### 12.3.3 PWM 中断

对于每一个 PWM 通道，可以设置周期计数值，当 PWM 输出的周期数达到这个计数值时，将产生 PWM 中断。

表 12.1: 占空比参数

频率/MHz	支持的占空比 (n 为整数, 且 $2 \leq n \leq 65535^2$ )											
36	0%	50%	100%									
24	0%	33.33%	66.67%	100%								
18	0%	25%	50%	75%	100%							
14.4	0%	20%	40%	60%	80%	100%						
12	0%	16.67%	33.33%	50%	66.67%	83.33%	100%					
10.29	0%	14.29%	28.57%	42.86%	57.14%	71.43%	85.71%	100%				
9	0%	12.50%	25%	37.50%	50%	62.50%	75%	87.50%	100%			
8	0%	11.11%	22.22%	33.33%	44.44%	55.56%	66.67%	77.78%	88.89%	100%		
7.2	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	
•												
•												
•												
72/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 寄存器描述

名称	描述
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 second counter threshold configuration register
pwm0_period	PWM0 period setting register
pwm0_config	PWM0 configuration register
pwm0_interrupt	PWM0 interrupt register
pwm1_clkdiv	PWM1 clock division configuration register
pwm1_thre1	PWM1 first counter threshold configuration register
pwm1_thre2	PWM1 second counter threshold configuration register

名称	描述
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 second 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 second 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 second 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

地址: 0x4000a400

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

pwm\_int\_clear

pwm\_interrupt\_sts

位	名称	权限	复位值	描述
31:14	RSVD			
13:8	pwm_int_clear	w	6'd0	PWM channel interrupt clear
7:6	RSVD			
5:0	pwm_interrupt_sts	r	6'd0	PWM channel interrupt status

### 12.4.2 pwm0\_clkdiv

地址: 0x4000a420

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

pwm\_clk\_div

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_clk_div	r/w	16'b0	PWM clock division

### 12.4.3 pwm0\_thre1

地址: 0x4000a424

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

pwm\_thre1

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_thre1	r/w	16'b0	PWM first counter threshold, can't be larger than pwm_thre2

#### 12.4.4 pwm0\_thre2

地址: 0x4000a428

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

pwm\_thre2

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_thre2	r/w	16'd0	PWM sencond counter threshold, can't be smaller than pwm_thre1

#### 12.4.5 pwm0\_period

地址: 0x4000a42c

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

pwm\_period

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_period	r/w	16'd0	PWM period setting

### 12.4.6 pwm0\_config

地址: 0x4000a430

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

pwm\_sts\_top      pwm\_stop\_en      pwm\_sw\_mode      pwm\_stop\_mode      pwm\_out\_inv      reg\_clk\_sel  
 RSVD                RSVD               RSVD               RSVD               RSVD               RSVD               RSVD

位	名称	权限	复位值	描述
31:8	RSVD			
7	pwm_sts_top	r	1'b0	PWM stop status
6	pwm_stop_en	r/w	1'b0	PWM stop enable
5	pwm_sw_mode	r/w	1'b0	PWM SW Mode setting
4	pwm_sw_force_val	r/w	1'b0	PWM SW Mode force value
3	pwm_stop_mode	r/w	1'b1	PWM stop mode, 1'b1 - graceful ; 1'b0 - abrupt
2	pwm_out_inv	r/w	1'b0	PWM invert output mode
1:0	reg_clk_sel	r/w	2'd0	PWM clock source select, 2'b00-xclk ; 2'b01-bclk ; others-f32k_clk

### 12.4.7 pwm0\_interrupt

地址: 0x4000a434

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

位	名称	权限	复位值	描述
31:17	RSVD			
16	pwm_int_enable	r/w	1'b0	PWM interrupt enable
15:0	pwm_int_period_cnt	r/w	16'd0	PWM interrupt period counter threshold

### 12.4.8 pwm1\_clkdiv

地址: 0x4000a440

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

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_clk_div	r/w	16'b0	PWM clock division

### 12.4.9 pwm1\_thre1

地址: 0x4000a444

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

pwm\_thre1

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_thre1	r/w	16'b0	PWM first counter threshold, can't be larger than pwm_thre2

#### 12.4.10 pwm1\_thre2

地址: 0x4000a448

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

pwm\_thre2

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_thre2	r/w	16'd0	PWM sencond counter threshold, can't be smaller than pwm_thre1

#### 12.4.11 pwm1\_period

地址: 0x4000a44c

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

pwm\_period

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_period	r/w	16'd0	PWM period setting

### 12.4.12 pwm1\_config

地址: 0x4000a450

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

*pwm\_sts\_top*    *pwm\_stop\_en*    *pwm\_sw\_mode*    *pwm\_stop\_mode*    *pwm\_out\_inv*    *reg\_clk\_sel*  
*RSVD*    *RSVD*

位	名称	权限	复位值	描述
31:8	RSVD			
7	pwm_sts_top	r	1'b0	PWM stop status
6	pwm_stop_en	r/w	1'b0	PWM stop enable
5	pwm_sw_mode	r/w	1'b0	PWM SW Mode setting
4	pwm_sw_force_val	r/w	1'b0	PWM SW Mode force value
3	pwm_stop_mode	r/w	1'b1	PWM stop mode, 1'b1 - graceful ; 1'b0 - abrupt
2	pwm_out_inv	r/w	1'b0	PWM invert output mode
1:0	reg_clk_sel	r/w	2'd0	PWM clock source select, 2'b00-xclk ; 2'b01-bclk ; others-f32k_clk

### 12.4.13 pwm1\_interrupt

地址: 0x4000a454

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

位	名称	权限	复位值	描述
31:17	RSVD			
16	pwm_int_enable	r/w	1'b0	PWM interrupt enable
15:0	pwm_int_period_cnt	r/w	16'd0	PWM interrupt period counter threshold

### 12.4.14 pwm2\_clkdiv

地址: 0x4000a460

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

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_clk_div	r/w	16'b0	PWM clock division

### 12.4.15 pwm2\_thre1

地址: 0x4000a464

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

pwm\_thre1

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_thre1	r/w	16'b0	PWM first counter threshold, can't be larger than pwm_thre2

#### 12.4.16 pwm2\_thre2

地址: 0x4000a468

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

pwm\_thre2

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_thre2	r/w	16'd0	PWM sencond counter threshold, can't be smaller than pwm_thre1

#### 12.4.17 pwm2\_period

地址: 0x4000a46c

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

pwm\_period

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_period	r/w	16'd0	PWM period setting

## 12.4.18 pwm2\_config

地址: 0x4000a470

位	名称	权限	复位值	描述
31:8	RSVD			
7	pwm_sts_top	r	1'b0	PWM stop status
6	pwm_stop_en	r/w	1'b0	PWM stop enable
5	pwm_sw_mode	r/w	1'b0	PWM SW Mode setting
4	pwm_sw_force_val	r/w	1'b0	PWM SW Mode force value
3	pwm_stop_mode	r/w	1'b1	PWM stop mode, 1'b1 - graceful ; 1'b0 - abrupt
2	pwm_out_inv	r/w	1'b0	PWM invert output mode
1:0	reg_clk_sel	r/w	2'd0	PWM clock source select, 2'b00-xclk ; 2'b01-bclk ; others-f32k_clk

### 12.4.19 pwm2\_interrupt

地址: 0x4000a474

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

位	名称	权限	复位值	描述
31:17	RSVD			
16	pwm_int_enable	r/w	1'b0	PWM interrupt enable
15:0	pwm_int_period_cnt	r/w	16'd0	PWM interrupt period counter threshold

### 12.4.20 pwm3\_clkdiv

地址: 0x4000a480

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

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_clk_div	r/w	16'b0	PWM clock division

### 12.4.21 pwm3\_thre1

地址: 0x4000a484

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

pwm\_thre1

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_thre1	r/w	16'b0	PWM first counter threshold, can't be larger than pwm_thre2

### 12.4.22 pwm3\_thre2

地址: 0x4000a488

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

pwm\_thre2

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_thre2	r/w	16'd0	PWM sencond counter threshold, can't be smaller than pwm_thre1

### 12.4.23 pwm3\_period

地址: 0x4000a48c

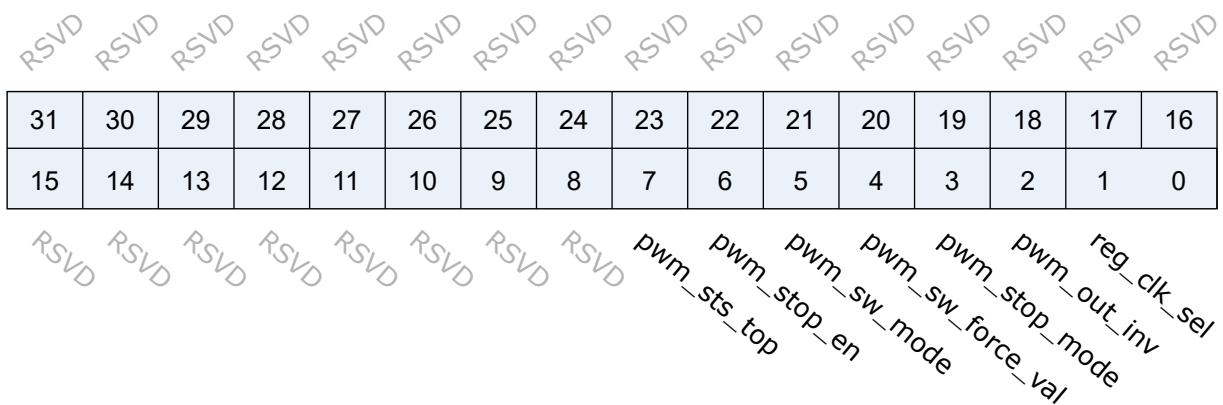
| RSVD |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 31   | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   |
| 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8    | 7    |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

pwm\_period

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_period	r/w	16'd0	PWM period setting

## 12.4.24 pwm3\_config

地址: 0x4000a490



位	名称	权限	复位值	描述
31:8	RSVD			
7	pwm_sts_top	r	1'b0	PWM stop status
6	pwm_stop_en	r/w	1'b0	PWM stop enable
5	pwm_sw_mode	r/w	1'b0	PWM SW Mode setting
4	pwm_sw_force_val	r/w	1'b0	PWM SW Mode force value
3	pwm_stop_mode	r/w	1'b1	PWM stop mode, 1'b1 - graceful ; 1'b0 - abrupt
2	pwm_out_inv	r/w	1'b0	PWM invert output mode
1:0	reg_clk_sel	r/w	2'd0	PWM clock source select, 2'b00-xclk ; 2'b01-bclk ; others-f32k_clk

### 12.4.25 pwm3\_interrupt

地址: 0x4000a494

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

位	名称	权限	复位值	描述
31:17	RSVD			
16	pwm_int_enable	r/w	1'b0	PWM interrupt enable
15:0	pwm_int_period_cnt	r/w	16'd0	PWM interrupt period counter threshold

### 12.4.26 pwm4\_clkdiv

地址: 0x4000a4a0

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

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_clk_div	r/w	16'b0	PWM clock division

### 12.4.27 pwm4\_thre1

地址: 0x4000a4a4

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

pwm\_thre1

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_thre1	r/w	16'b0	PWM first counter threshold, can't be larger than pwm_thre2

#### 12.4.28 pwm4\_thre2

地址: 0x4000a4a8

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

pwm\_thre2

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_thre2	r/w	16'd0	PWM sencond counter threshold, can't be smaller than pwm_thre1

#### 12.4.29 pwm4\_period

地址: 0x4000a4ac

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

pwm\_period

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	pwm_period	r/w	16'd0	PWM period setting

### 12.4.30 pwm4\_config

地址: 0x4000a4b0

位	名称	权限	复位值	描述
31:8	RSVD			
7	pwm_sts_top	r	1'b0	PWM stop status
6	pwm_stop_en	r/w	1'b0	PWM stop enable
5	pwm_sw_mode	r/w	1'b0	PWM SW Mode setting
4	pwm_sw_force_val	r/w	1'b0	PWM SW Mode force value
3	pwm_stop_mode	r/w	1'b1	PWM stop mode, 1'b1 - graceful ; 1'b0 - abrupt
2	pwm_out_inv	r/w	1'b0	PWM invert output mode
1:0	reg_clk_sel	r/w	2'd0	PWM clock source select, 2'b00-xclk ; 2'b01-bclk ; others-f32k_clk

### 12.4.31 pwm4\_interrupt

地址: 0x4000a4b4

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

位	名称	权限	复位值	描述
31:17	RSVD			
16	pwm_int_enable	r/w	1'b0	PWM interrupt enable
15:0	pwm_int_period_cnt	r/w	16'd0	PWM interrupt period counter threshold

## 13.1 简介

芯片内置 2 组 32-bit 计数器，每个计数器可独立配置其参数与时钟频率。

芯片内有一组看门狗计数器，不可预知的软件或硬件行为有可能导致应用程序工作失常，看门狗定时器可以帮助系统从中恢复，如果当前阶段超过预定时间，但没有喂狗或关闭看门狗定时器，可依设定触发中断或系统复位。

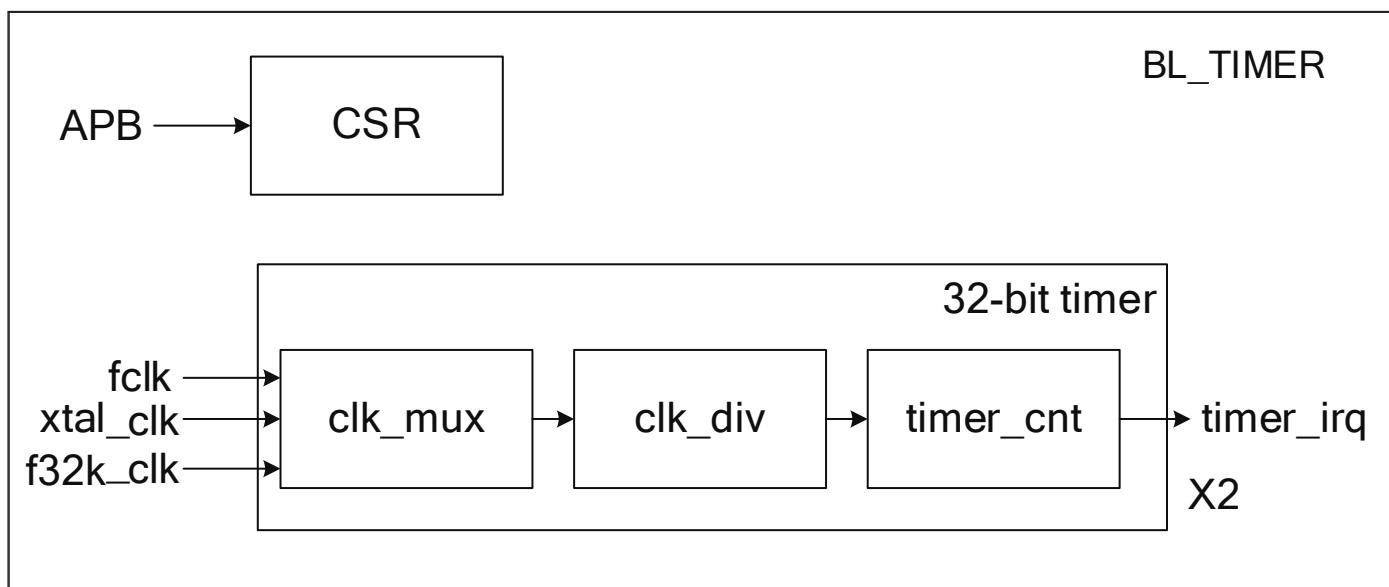


图 13.1: 定时器框图

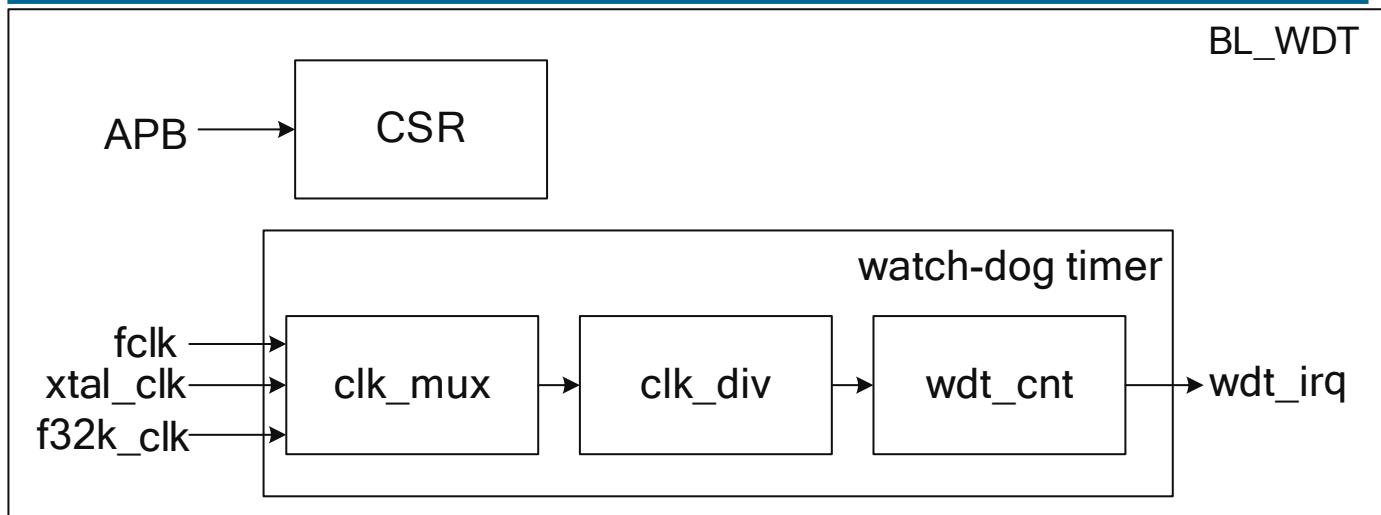


图 13.2: 看门狗定时器框图

## 13.2 主要特征

- 多种时钟来源选择
- 8-bit 时钟分频器，分频系数为 1-256
- 两组 32-bit 定时器
- 每个定时器包含三组报警值设定，可独立设定每组报警值溢出时报警
- 支持 FreeRun 模式和 PreLoad 模式
- 16-bit 看门狗定时器
- 支持写入密码保护，防止误设定造成系统异常
- 支持中断或复位两种看门狗溢出方式

## 13.3 功能描述

### 13.3.1 8-bit 分频器

看门狗定时器时钟有 3 种选择：

- Fclk--系统主时钟
- 32K--32K 时钟
- Xtal--外部晶振

每个定时器时钟来源都有四种选择，来源如下：

- Fclk--系统主时钟

- 32K--32K 时钟
- 1K--1K 时钟（32K 的分频）
- Xtal--外部晶振

每个计数器有各自的 8-bit 分频器，可通过 APB 将选择到的时钟进行 1-256 的分频，具体来说设定为 0 时表示不分频，设定为 1 时进行 2 分频以此类推，最大分频系数为 256，计数器将以分频后的时钟作为计数周期单位，每经过一个计数周期进行上数一的动作。

### 13.3.2 通用定时器工作原理

每个通用定时器都包含三组比较器，一个计数器以及一个预加载寄存器，当设定好时钟源，启动定时器后，计数器开始向上累加计数，当计数器的值与比较器相等的时候，比较标志置位同时可以产生比较中断。

计数器的初始值取决于定时的模式，在 FreeRun 模式下，计数器的初始值是 0，然后累加计数，当达到计数最大值后，然后从 0 再次开始计数。

在 PreLoad 模式下，计数器的初始值是 PreLoad 寄存器的值，然后向上累加计数，当满足 PreLoad 条件时，计数器的值被置为 PreLoad 寄存器的值，然后计数器再次开始向上累加计数，在定时器的计数过程中，一旦计数器的值与三个比较器中的某比较值一致，该比较器的比较标志就会置位，并可以产生相应的比较中断。

若预加载寄存器的值为 10，比较器 0 的值为 13，比较器 1 的值为 16，比较器 2 的值为 19，则定时器在 PreLoad 的模式下工作时序如下图：

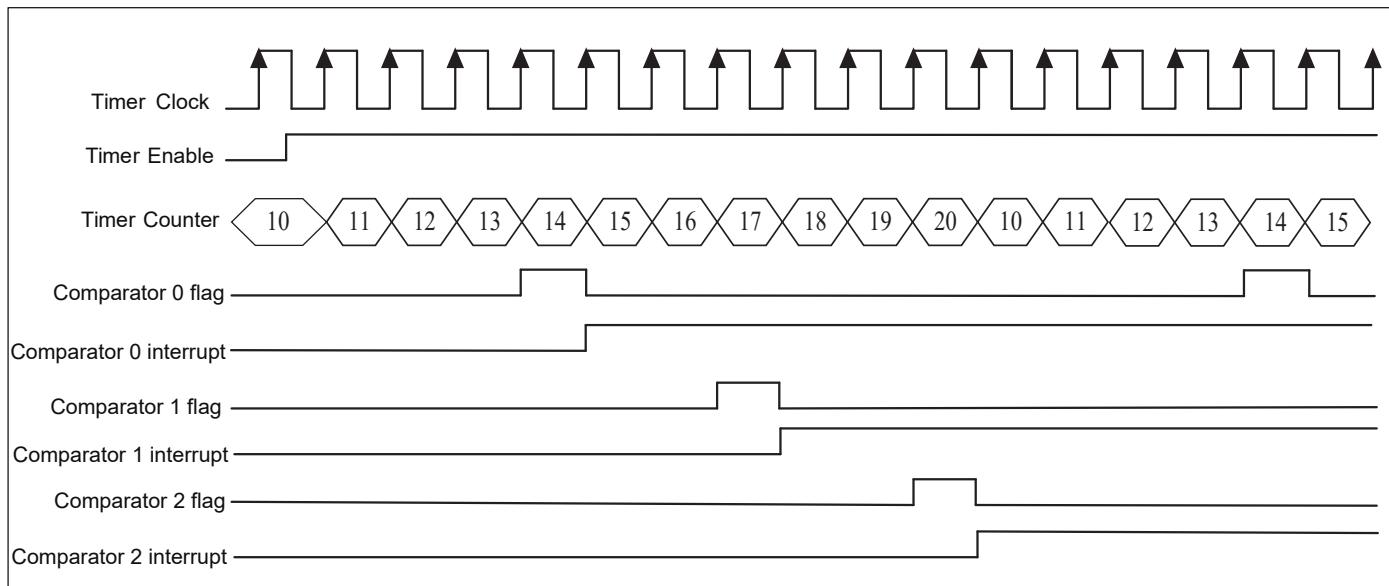


图 13.3: 定时器在 PreLoad 模式下工作时序

在 FreeRun 模式下，定时器工作时序与 PreLoad 基本相同，只是计数器会从 0 开始累计到最大值，期间产生的比较标志和比较中断的机制与 PreLoad 模式相同。

### 13.3.3 看门狗定时器工作原理

Watchdog 定时器包含一个计数器和一个比较器，计数器从 0 开始累加计数，如果计数器被复位（喂狗），则从 0 再次开始向上计数，当计数器的值与比较器相等的时候，可以产生一个比较中断信号或者系统复位信号，用户可以根据需要选择使用其中一个。看门狗计数器会在每个计数周期单位上加 1，软件可以在任何时间点通过 APB 将看门狗计数器归零。

若比较器的值为 6, Watchdog 的工作时序如下图所示

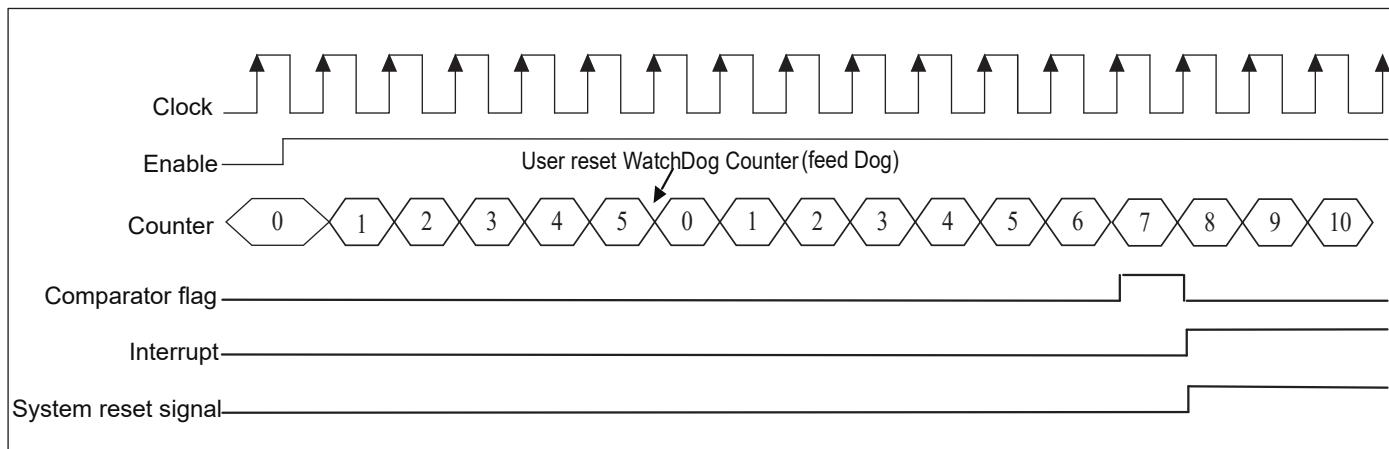


图 13.4: Watchdog 工作时序

### 13.3.4 报警设定

每一组计数器有三个比较值提供软件设定，并可设定每一组比较值是否触发报警中断，当计数器与比较值吻合且设定会报警时，计数器会通过中断通知处理器。软件可以通过 APB 读取目前是否发生报警和是哪个比较值触发报警中断，当清理报警中断时亦会同步清理报警状态。

### 13.3.5 看门狗报警

每个计数器可设定一组比较值，当软件因为系统错误，来不及将看门狗计数器归零，导致看门狗计数器超过比较值时，便会触发看门狗报警，报警方式有两种，第一种是通过中断通知软件进行必要的处置，第二种是进入系统看门狗复位，看门狗复位被触发时，会通知系统复位控制器，并做好系统复位前准备，当一切就绪后进入系统看门狗复位，值得注意的是，软件可通过 APB 读取 WSR 寄存器得知是否曾经发生看门狗系统复位。

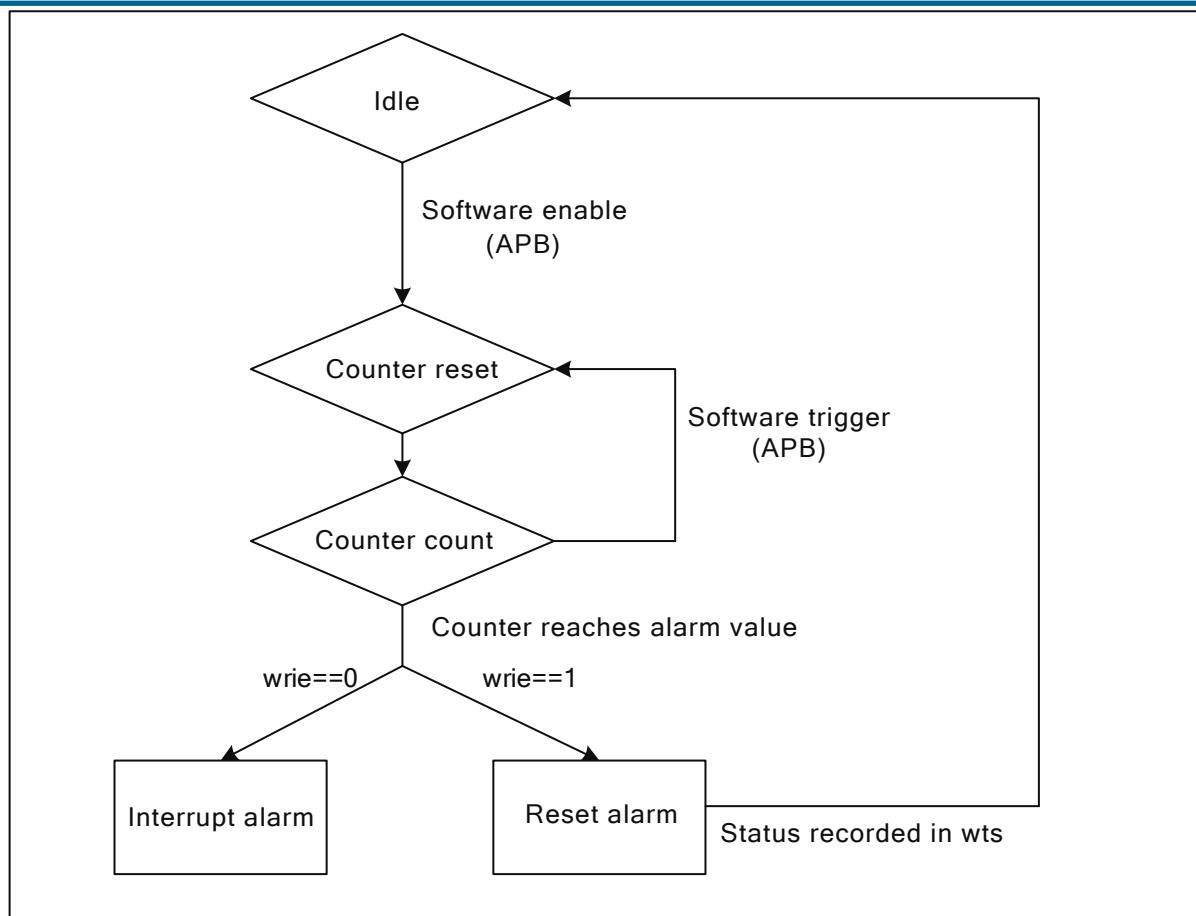


图 13.5: 看门狗报警机制

## 13.4 寄存器描述

名称	描述
TCCR	Timer clock source configuration register
TMR2_0	Timer2 match register 0
TMR2_1	Timer2 match register 1
TMR2_2	Timer2 match register 2
TMR2_0	Timer3 match register 0
TMR2_1	Timer3 match register 1
TMR2_2	Timer3 match register 2
TCR2	Timer2 counter register
TCR3	Timer3 counter register
TMSR2	Timer2 match register status

名称	描述
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

地址: 0x4000a500

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

位	名称	权限	复位值	描述
31:10	RSVD			
9:8	cs_wdt	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	cs_2	r/w	2'd0	
4	RSVD			
3:2	cs_1	r/w	2'd0	
1:0	RSVD			

### 13.4.2 TMR2\_0

地址: 0x4000a510

tmr\_3\_0

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

tmr\_3\_0

位	名称	权限	复位值	描述
31:0	tmr_2_0	r/w	32'hffffffff	Timer2 match register 0

### 13.4.3 TMR2\_1

地址: 0x4000a514

tmr\_3\_1

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

tmr\_3\_1

位	名称	权限	复位值	描述
31:0	tmr_2_1	r/w	32'hffffffff	Timer2 match register 1

### 13.4.4 TMR2\_2

地址: 0x4000a518

tmr\_3\_2

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

tmr\_3\_2

位	名称	权限	复位值	描述
31:0	tmr_2_2	r/w	32'hffffffff	Timer2 match register 2

### 13.4.5 TMR2\_0

地址: 0x4000a51c

tmr\_3\_0

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

tmr\_3\_0

位	名称	权限	复位值	描述
31:0	tmr_3_0	r/w	32'hffffffff	Timer3 match register 0

### 13.4.6 TMR2\_1

地址: 0x4000a520

tmr\_3\_1

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

tmr\_3\_1

位	名称	权限	复位值	描述
31:0	tmr_3_1	r/w	32'hffffffff	Timer3 match register 1

### 13.4.7 TMR2\_2

地址: 0x4000a524

tmr\_3\_2

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

tmr\_3\_2

位	名称	权限	复位值	描述
31:0	tmr_3_2	r/w	32'hffffffff	Timer3 match register 2

### 13.4.8 TCR2

地址: 0x4000a52c

tcr2\_counter

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

tcr2\_counter

位	名称	权限	复位值	描述
31:0	tcr2_counter	r	32'h0	Timer2 counter register

### 13.4.9 TCR3

地址: 0x4000a530

tcr3\_counter

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

tcr3\_counter

位	名称	权限	复位值	描述
31:0	tcr3_counter	r	32'h0	Timer3 counter register

### 13.4.10 TMSR2

地址: 0x4000a538

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

RSVD tmsr2\_2 tmsr2\_1 tmsr2\_0

位	名称	权限	复位值	描述
31:3	RSVD			
2	tmsr2_2	r	1'b0	Timer2 match register 2 status/Clear interrupt would also clear this bit
1	tmsr2_1	r	1'b0	Timer2 match register 1 status/Clear interrupt would also clear this bit
0	tmsr2_0	r	1'b0	Timer2 match register 0 status/Clear interrupt would also clear this bit

### 13.4.11 TMSR3

地址: 0x4000a53c

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

*RSVD RSVD RSVD*  
*tmsr3\_2 tmsr3\_1 tmsr3\_0*

位	名称	权限	复位值	描述
31:3	RSVD			
2	tmsr3_2	r	1'b0	Timer3 match register 2 status/Clear interrupt would also clear this bit
1	tmsr3_1	r	1'b0	Timer3 match register 1 status/Clear interrupt would also clear this bit
0	tmsr3_0	r	1'b0	Timer3 match register 0 status/Clear interrupt would also clear this bit

### 13.4.12 TIER2

地址: 0x4000a544

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

*RSVD RSVD RSVD*  
*tier2\_2 tier2\_1 tier2\_0*

位	名称	权限	复位值	描述
31:3	RSVD			
2	tier2_2	r/w	1'b0	Timer2 match register 2 interrupt enable register
1	tier2_1	r/w	1'b0	Timer2 match register 1 interrupt enable register
0	tier2_0	r/w	1'b0	Timer2 match register 0 interrupt enable register

### 13.4.13 TIER3

地址: 0x4000a548

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

tier3\_2 tier3\_1 tier3\_0

位	名称	权限	复位值	描述
31:3	RSVD			
2	tier3_2	r/w	1'b0	Timer3 match register 2 interrupt enable register
1	tier3_1	r/w	1'b0	Timer3 match register 1 interrupt enable register
0	tier3_0	r/w	1'b0	Timer3 match register 0 interrupt enable register

### 13.4.14 TPLVR2

地址: 0x4000a550

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

tplvr2

位	名称	权限	复位值	描述
31:0	tplvr2	r/w	32'h0	Timer2 pre-load value register

### 13.4.15 TPLVR3

地址: 0x4000a554

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

tplvr3



位	名称	权限	复位值	描述
31:0	tplvr3	r/w	32'h0	Timer3 pre-load value register

### 13.4.16 TPLCR2

地址: 0x4000a55c

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

位	名称	权限	复位值	描述
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

地址: 0x4000a560

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

位	名称	权限	复位值	描述
31:2	RSVD			

位	名称	权限	复位值	描述
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

地址: 0x4000a564

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

位	名称	权限	复位值	描述
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

地址: 0x4000a568

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

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	wmr	r/w	16'hffff	WDT counter match value register

### 13.4.20 WVR

地址: 0x4000a56c

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

wvr

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	wvr	r	16'h0	WDT counter value register

### 13.4.21 WSR

地址: 0x4000a570

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

RSVD wts

位	名称	权限	复位值	描述
31:1	RSVD			

位	名称	权限	复位值	描述
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 cleared (Read)1'b1 - Watchdog timer caused reset

### 13.4.22 TICR2

地址: 0x4000a578

RSVD	RSVD																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	tclr2_2	tclr2_1	tclr2_0

位	名称	权限	复位值	描述
31:3	RSVD			
2	tclr2_2	w	1'b0	Timer2 Interrupt clear for match comparator 2
1	tclr2_1	w	1'b0	Timer2 Interrupt clear for match comparator 1
0	tclr2_0	w	1'b0	Timer2 Interrupt clear for match comparator 0

### 13.4.23 TICR3

地址: 0x4000a57c

RSVD	RSVD																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	tclr3_2	tclr3_1	tclr3_0

位	名称	权限	复位值	描述
31:3	RSVD			
2	tclr3_2	w	1'b0	Timer3 Interrupt clear for match comparator 2
1	tclr3_1	w	1'b0	Timer3 Interrupt clear for match comparator 1
0	tclr3_0	w	1'b0	Timer3 Interrupt clear for match comparator 0

13.4.24 WICR

地址: 0x4000a580

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

位	名称	权限	复位值	描述
31:1	RSVD			
0	wiclr	w	1'b0	WDT Interrupt clear register

## 13.4.25 TCER

地址: 0x4000a584

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

位	名称	权限	复位值	描述
31:3	RSVD			
2	timer3_en	r/w	1'b0	Timer3 count enable

位	名称	权限	复位值	描述
1	timer2_en	r/w	1'b0	Timer2 count enable
0	RSVD			

## 13.4.26 TCMR

地址: 0x4000a588

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

位	名称	权限	复位值	描述
31:3	RSVD			
2	timer3_mode	r/w	1'b0	Timer1/2/3 count mode register 1'b0 - pre-load mode 1'b1 - free run mode
1	timer2_mode	r/w	1'b0	
0	RSVD			

## 13.4.27 TILR2

地址: 0x4000a590

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

位	名称	权限	复位值	描述
31:3	RSVD			
2	tilr2_2	r/w	1'b0	Timer2 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt
1	tilr2_1	r/w	1'b0	
0	tilr2_0	r/w	1'b0	

### 13.4.28 TILR3

地址: 0x4000a594

RSVD	RSVD	RSVD																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	tilr3_2	tilr3_1	tilr3_0

位	名称	权限	复位值	描述
31:3	RSVD			
2	tilr3_2	r/w	1'b0	Timer3 match 0/1/2 interrupt mode register 1'b0 - level interrupt 1'b1 - pulse interrupt
1	tilr3_1	r/w	1'b0	
0	tilr3_0	r/w	1'b0	

### 13.4.29 WCR

地址: 0x4000a598

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

位	名称	权限	复位值	描述
31:1	RSVD			
0	wcr	w	1'b0	WDT timer count reset register

### 13.4.30 WFAR

地址: 0x4000a59c

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

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	wfar	w	16'b0	WDT access key1 - 16'hBABA

### 13.4.31 WSAR

地址: 0x4000a5a0

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

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	wsar	w	16'b0	WDT access key2 - 16'hEB10

### 13.4.32 TCVWR2

地址: 0x4000a5a8

tcvwr2

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

tcvwr2

位	名称	权限	复位值	描述
31:0	tcvwr2	r	32'h0	Timer2 capture value of counter

### 13.4.33 TCVWR3

地址: 0x4000a5ac

tcvwr3

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

tcvwr3

位	名称	权限	复位值	描述
31:0	tcvwr3	r	32'h0	Timer3 capture value of counter

### 13.4.34 TCVSYN2

地址: 0x4000a5b4

tcvsyn2

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

tcvsyn2

位	名称	权限	复位值	描述
31:0	tcvsyn2	r	32'h0	Timer2 synchronous value of counter

### 13.4.35 TCVSYN3

地址: 0x4000a5b8

tcvsyn3

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

tcvsyn3

位	名称	权限	复位值	描述
31:0	tcvsyn3	r	32'h0	Timer3 synchronous value of counter

### 13.4.36 TCDR

地址: 0x4000a5bc

wcdr

tcdr3

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

tcdr2

RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD

位	名称	权限	复位值	描述
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			

## 14.1 简介

正交解码器（quadrature decoder），用于将双路旋转编码器产生的两组相位相差 90 度的脉冲解码为对应转速和旋转方向。

## 14.2 主要特征

- 三组 QDEC 可供使用
- QDEC 的时钟源可以为 32K (f32k\_clk) 或 32M (xclk)，正常工作时建议选择 32M 作为时钟源，当进入睡眠模式并希望可以由 QDEC 唤醒时建议使用 32k
- 支持 5 位分频值，可分频 1~32
- 16 位脉冲计数范围 (-32768~32767 pulse/sample)
- 12 种可配置的 sample 周期 (32us~131ms per sample at 1MHz)
- 16 位可设置的 report 周期 (0~65535 sample/report)
- 内置一个可随采样进行闪烁的 LED 功能 (LED on/off 0~511 us/sample)
- 中断可配 (sample 中断、report 中断、error 中断、overflow 中断)
- 可配置为 PDS 的唤醒源 (clock source 需要配置为 32k)

## 14.3 功能描述

QDEC 预期的工作频率为 1MHz，检测的速度越快所需的工作频率越高。每次采样会将编码器输出的 A/B 两相脉冲解码为高低电平，对比前次采样结果可获得当前编码器旋转方向与脉冲计数变化情况（顺时针转 +1，逆时针转 -1，不转不变，出错报错并计数），经过 report 所设定的采样次数后，即可获得此段时间内编码器的旋转方向和脉冲计数，据此求解出此段 report 时间内转速方向均值。每次采样的周期可配，在工作频率为 1MHz 时，最低 32us 一次采样，最高 131ms 一次采样。可配置中断为单次采样结束触发（sample 中断）、多次采样结束触发（report 中断），以灵活测量转速。可配置的 LED 闪烁功能，闪烁频率 =LED 周期/采样周期，每次闪烁中的 on/off 由 LED 极性决定。

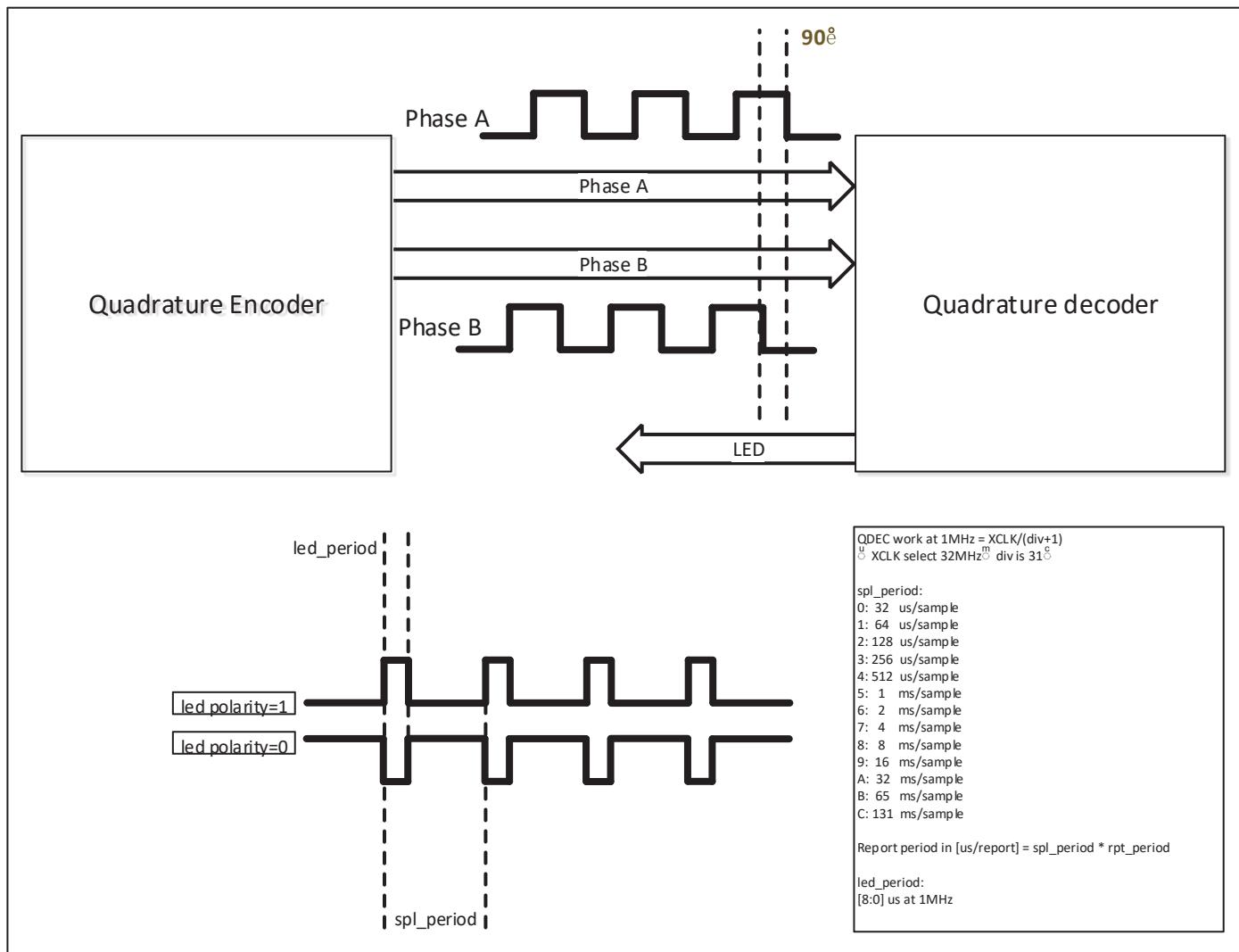


图 14.1: QDEC 功能框图

## 14.4 寄存器描述

名称	描述
qdec0_ctrl0	QDEC0 control0
qdec0_ctrl1	QDEC0 control1
qdec0_value	QDEC0 value
qdec0_int_en	QDEC0 interrupt enable
qdec0_int_sts	QDEC0 interrupt status
qdec0_int_clr	QDEC0 interrupt clear
qdec1_ctrl0	QDEC1 control0
qdec1_ctrl1	QDEC1 control1
qdec1_value	QDEC1 value
qdec1_int_en	QDEC1 interrupt enable
qdec1_int_sts	QDEC1 interrupt status
qdec1_int_clr	QDEC1 interrupt clear
qdec2_ctrl0	QDEC2 control0
qdec2_ctrl1	QDEC2 control1
qdec2_value	QDEC2 value
qdec2_int_en	QDEC2 interrupt enable
qdec2_int_sts	QDEC2 interrupt status
qdec2_int_clr	QDEC2 interrupt clear

### 14.4.1 qdec0\_ctrl0

地址: 0x4000a800

RSVD  
 RSVD  
 RSVD  
 RSVD

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

rpt\_period      spl\_period      deg\_cnt      deg\_en      led\_pol      led\_en      qdec\_en

位	名称	权限	复位值	描述
31:28	RSVD			
27:12	rpt_period	r/w	16'd10	"RPT" report period in [samples/report]. Specifies the number of samples to be accumulated in the ACC1 register before the RPT_RDY and DBL_RDY events can be generated "RPT_US" report period in [us/report] = SP * RP
11:8	spl_period	r/w	4'h2	"SPL" sample period in [us/sample]. The SAMPLE register will be updated for every new sample 0: 32 us 1: 64 2: 128 3: 256 4: 512 5: 1 ms 6: 2 7: 4 8: 8 9: 16 A: 32 B: 65 C: 131
7:4	deg_cnt	r/w	0	
3	deg_en	r/w	0	deglitch
2	led_pol	r/w	1	led polarity
1	led_en	r/w	0	
0	qdec_en	r/w	0	

#### 14.4.2 qdec0\_ctrl1

地址: 0x4000a804

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

RSVD	input_swap	rpt_mode	spl_mode	acc_mode													
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------------	----------	----------	----------

位	名称	权限	复位值	描述
31:25	RSVD			
24:16	led_period	r/w	0	Period in us the LED is switched on prior to sampling
15:4	RSVD			
3	input_swap	r/w	0	input a/b swap
2	rpt_mode	r/w	0	rpt option 0: Count time only if sample change 1: Continue time
1	spl_mode	r/w	0	spl option 0: Stop sample if rpt_rdy 1: Continue sample
0	acc_mode	r/w	1	acc option 0: Stop accumulate if overflow 1: Continue accumulate

#### 14.4.3 qdec0\_value

地址: 0x4000a808

RSVD	RSVD	spl_val	RSVD	acc1_val	acc2_val																			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									

位	名称	权限	复位值	描述
31:30	RSVD			

位	名称	权限	复位值	描述
29:28	spl_val	r	0	Sample value. Direction of last change 00: no change 01: clockwise 11: counter-clockwise 10: Error
27:20	RSVD			
19:16	acc2_val	r	0	Double error accumulation (0 15)
15:0	acc1_val	r	0	Sample accumulation (-1024 1023)

#### 14.4.4 qdec0\_int\_en

地址: 0x4000a810

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

RSVD RSVD

dbl\_rdy\_en spl\_rdy\_en rpt\_rdy\_en  
 overflow\_en

位	名称	权限	复位值	描述
31:4	RSVD			
3	overflow_en	r/w	0	
2	dbl_rdy_en	r/w	0	
1	spl_rdy_en	r/w	0	
0	rpt_rdy_en	r/w	1	

#### 14.4.5 qdec0\_int\_sts

地址: 0x4000a814

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

RSVD RSVD

overflow\_sts dbl\_rdy\_sts spl\_rdy\_sts rpt\_rdy\_sts

位	名称	权限	复位值	描述
31:4	RSVD			
3	overflow_sts	r	0	ACC1 or ACC2 overflow
2	dbl_rdy_sts	r	0	ACC2 double error
1	spl_rdy_sts	r	0	Event being generated for every new sample value written to the SAMPLE register
0	rpt_rdy_sts	r	0	Non-null report ready

#### 14.4.6 qdec0\_int\_clr

地址: 0x4000a818

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

RSVD RSVD

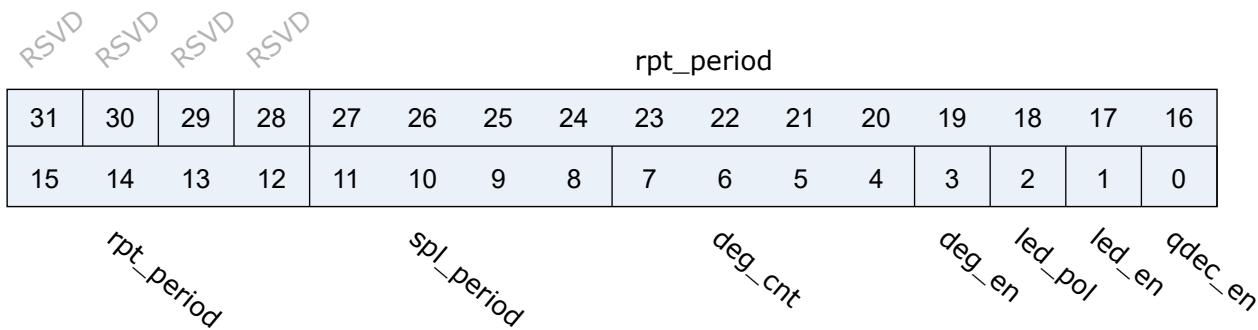
overflow\_clr dbl\_rdy\_clr spl\_rdy\_clr rpt\_rdy\_clr

位	名称	权限	复位值	描述
31:4	RSVD			
3	overflow_clr	w1c	0	
2	dbl_rdy_clr	w1c	0	
1	spl_rdy_clr	w1c	0	

位	名称	权限	复位值	描述
0	rpt_rdy_clr	w1c	0	

#### 14.4.7 qdec1\_ctrl0

地址: 0x4000a840



位	名称	权限	复位值	描述
31:28	RSVD			
27:12	rpt_period	r/w	16'd10	"RPT" report period in [samples/report]. Specifies the number of samples to be accumulated in the ACC1 register before the RPT_RDY and DBL_RDY events can be generated "RPT_US" report period in [us/report] = SP * RP
11:8	spl_period	r/w	4'h2	"SPL" sample period in [us/sample]. The SAMPLE register will be updated for every new sample 0: 32 us 1: 64 2: 128 3: 256 4: 512 5: 1 ms 6: 2 7: 4 8: 8 9: 16 A: 32 B: 65 C: 131
7:4	deg_cnt	r/w	0	
3	deg_en	r/w	0	deglitch

位	名称	权限	复位值	描述
2	led_pol	r/w	1	led polarity
1	led_en	r/w	0	
0	qdec_en	r/w	0	

#### 14.4.8 qdec1\_ctrl1

地址: 0x4000a844

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

位	名称	权限	复位值	描述
31:25	RSVD			
24:16	led_period	r/w	0	Period in us the LED is switched on prior to sampling
15:4	RSVD			
3	input_swap	r/w	0	input a/b swap
2	rpt_mode	r/w	0	rpt option 0: Count time only if sample change 1: Continue time
1	spl_mode	r/w	0	spl option 0: Stop sample if rpt_rdy 1: Continue sample
0	acc_mode	r/w	1	acc option 0: Stop accumulate if overflow 1: Continue accumulate

#### 14.4.9 qdec1\_value

地址: 0x4000a848

RSVD	RSVD	spl_val	RSVD	acc2_val											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

acc1\_val

位	名称	权限	复位值	描述
31:30	RSVD			
29:28	spl_val	r	0	Sample value. Direction of last change 00: no change 01: clockwise 11: counter-clockwise 10: Error
27:20	RSVD			
19:16	acc2_val	r	0	Double error accumulation (0 15)
15:0	acc1_val	r	0	Sample accumulation (-1024 1023)

#### 14.4.10 qdec1\_int\_en

地址: 0x4000a850

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

位	名称	权限	复位值	描述
31:4	RSVD			
3	overflow_en	r/w	0	
2	dbl_rdy_en	r/w	0	

位	名称	权限	复位值	描述
1	spl_rdy_en	r/w	0	
0	rpt_rdy_en	r/w	1	

#### 14.4.11 qdec1\_int\_sts

地址: 0x4000a854

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

RSVD  
 overflow\_sts dbl\_rdy\_sts spl\_rdy\_sts rpt\_rdy\_sts

位	名称	权限	复位值	描述
31:4	RSVD			
3	overflow_sts	r	0	ACC1 or ACC2 overflow
2	dbl_rdy_sts	r	0	ACC2 double error
1	spl_rdy_sts	r	0	Event being generated for every new sample value written to the SAMPLE register
0	rpt_rdy_sts	r	0	Non-null report ready

#### 14.4.12 qdec1\_int\_clr

地址: 0x4000a858

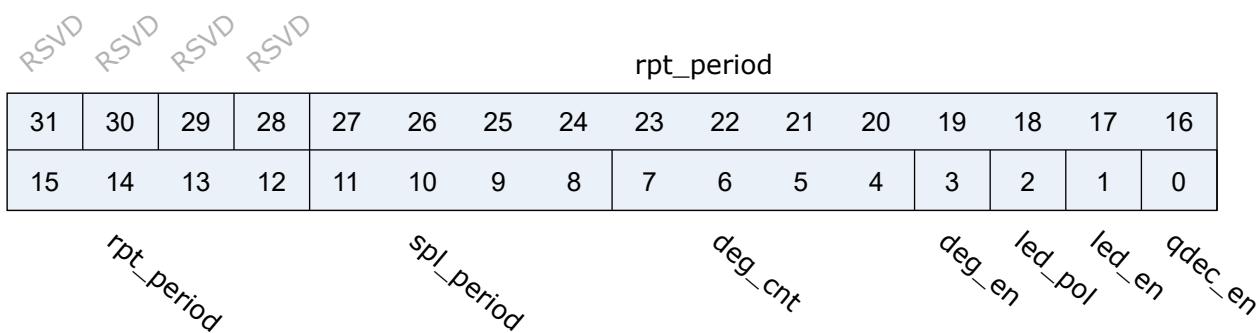
RSVD																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

RSVD  
 overflow\_clr dbl\_rdy\_clr spl\_rdy\_clr rpt\_rdy\_clr

位	名称	权限	复位值	描述
31:4	RSVD			
3	overflow_clr	w1c	0	
2	dbl_rdy_clr	w1c	0	
1	spl_rdy_clr	w1c	0	
0	rpt_rdy_clr	w1c	0	

#### 14.4.13 qdec2\_ctrl0

地址: 0x4000a880

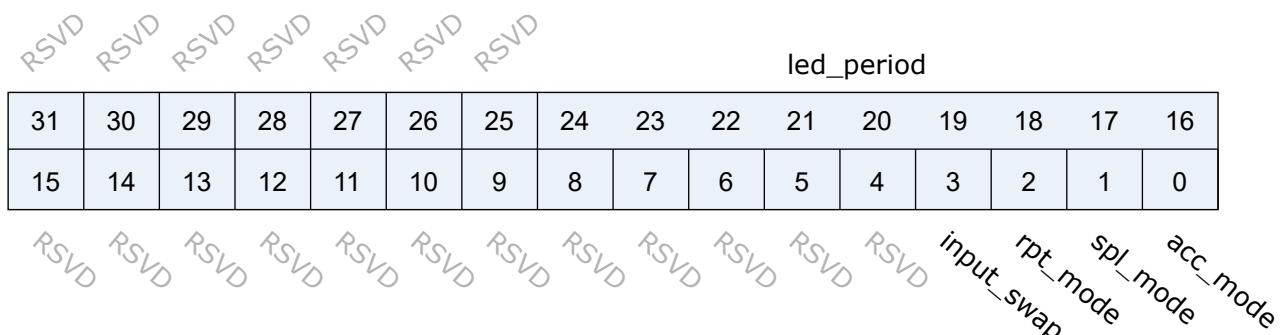


位	名称	权限	复位值	描述
31:28	RSVD			
27:12	rpt_period	r/w	16'd10	"RPT" report period in [samples/report]. Specifies the number of samples to be accumulated in the ACC1 register before the RPT_RDY and DBL_RDY events can be generated "RPT_US" report period in [us/report] = SP * RP

位	名称	权限	复位值	描述
11:8	spl_period	r/w	4'h2	"SPL" sample period in [us/sample]. The SAMPLE register will be updated for every new sample 0: 32 us 1: 64 2: 128 3: 256 4: 512 5: 1 ms 6: 2 7: 4 8: 8 9: 16 A: 32 B: 65 C: 131
7:4	deg_cnt	r/w	0	
3	deg_en	r/w	0	deglitch
2	led_pol	r/w	1	led polarity
1	led_en	r/w	0	
0	qdec_en	r/w	0	

#### 14.4.14 qdec2\_ctrl1

地址: 0x4000a884

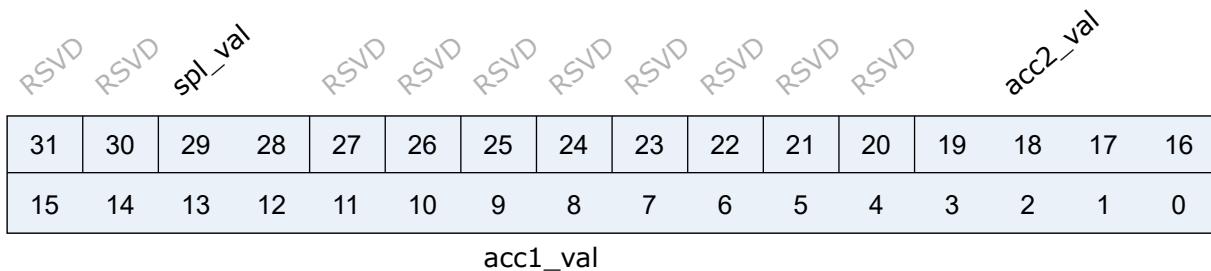


位	名称	权限	复位值	描述
31:25	RSVD			
24:16	led_period	r/w	0	Period in us the LED is switched on prior to sampling
15:4	RSVD			

位	名称	权限	复位值	描述
3	input_swap	r/w	0	input a/b swap
2	rpt_mode	r/w	0	rpt option 0: Count time only if sample change 1: Continue time
1	spl_mode	r/w	0	spl option 0: Stop sample if rpt_rdy 1: Continue sample
0	acc_mode	r/w	1	acc option 0: Stop accumulate if overflow 1: Continue accumulate

#### 14.4.15 qdec2\_value

地址: 0x4000a888



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

acc1\_val

位	名称	权限	复位值	描述
31:30	RSVD			
29:28	spl_val	r	0	Sample value. Direction of last change 00: no change 01: clockwise 11: counter-clockwise 10: Error
27:20	RSVD			
19:16	acc2_val	r	0	Double error accumulation (0 15)
15:0	acc1_val	r	0	Sample accumulation (-1024 1023)

#### 14.4.16 qdec2\_int\_en

地址: 0x4000a890

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

RSVD RSVD

位	名称	权限	复位值	描述
31:4	RSVD			
3	overflow_en	r/w	0	
2	dbl_rdy_en	r/w	0	
1	spl_rdy_en	r/w	0	
0	rpt_rdy_en	r/w	1	

#### 14.4.17 qdec2\_int\_sts

地址: 0x4000a894

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

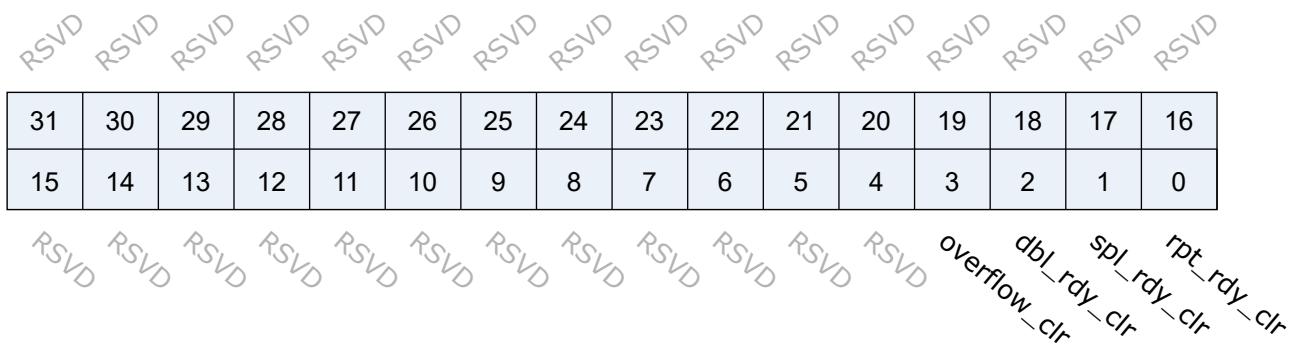
RSVD RSVD

位	名称	权限	复位值	描述
31:4	RSVD			
3	overflow_sts	r	0	ACC1 or ACC2 overflow
2	dbl_rdy_sts	r	0	ACC2 double error
1	spl_rdy_sts	r	0	Event being generated for every new sample value written to the SAMPLE register

位	名称	权限	复位值	描述
0	rpt_rdy_sts	r	0	Non-null report ready

#### 14.4.18 qdec2\_int\_clr

地址: 0x4000a898



位	名称	权限	复位值	描述
31:4	RSVD			
3	overflow_clr	w1c	0	
2	dbl_rdy_clr	w1c	0	
1	spl_rdy_clr	w1c	0	
0	rpt_rdy_clr	w1c	0	

## 15.1 简介

KYS(Key Scan) 模块用于对矩阵键盘进行扫描以获取键值，它会周期性的对键盘连接引脚进行扫描，一旦发现有按键按下会立即产生一个中断。

## 15.2 主要特征

- 可配置的行列数，最大可支持 8 行 \*20 列的矩阵键盘
- 最多存储四个键值
- 支持按键中断

## 15.3 功能描述

### 15.3.1 可配置的行列数

通过寄存器 KS\_CTRL 的位 <ROW\_NUM> 和 <COL\_NUM> 可以分别配置矩阵键盘的行列数，配置值为实际值减一。行数最大支持 8 行，列数最大支持 20 列。

### 15.3.2 GPIO 选择

由于按键扫描固定从功能为 ROW\_0 和 COL\_0 的 GPIO 引脚开始扫描，所以行引脚需要从功能为 ROW\_0 的 GPIO 开始往下依次选择，即从 GPIO0/GPIO8/GPIO16/GPIO24 中的任意一个开始，列引脚需要从功能为 COL\_0 的 GPIO 开始往下依次选择，即从 GPIO0/GPIO20 中的任意一个开始。

### 15.3.3 键值

键值会存储在寄存器 KEYCODE\_VALUE 中, 每 8 位为一个键值, 第一个键值会存放在最低 8 位。当寄存器 KEYCODE\_CLR 中对应序号的位被置一时, 对应的键值和中断标志位会被清除。键值对应的行号 = 键值% 总行数, 键值对应的列号 = 键值/总行数。

### 15.3.4 中断

当检测到有按键按下时, 中断标志位会置一, 同时产生一个中断。

## 15.4 寄存器描述

名称	描述
ks_ctrl	Keyscan control
ks_int_en	Keyscan interrupt enable
ks_int_sts	Keyscan interrupt status
keycode_clr	Keycode clear
keycode_value	Keycode value

### 15.4.1 ks\_ctrl

地址: 0x4000a900

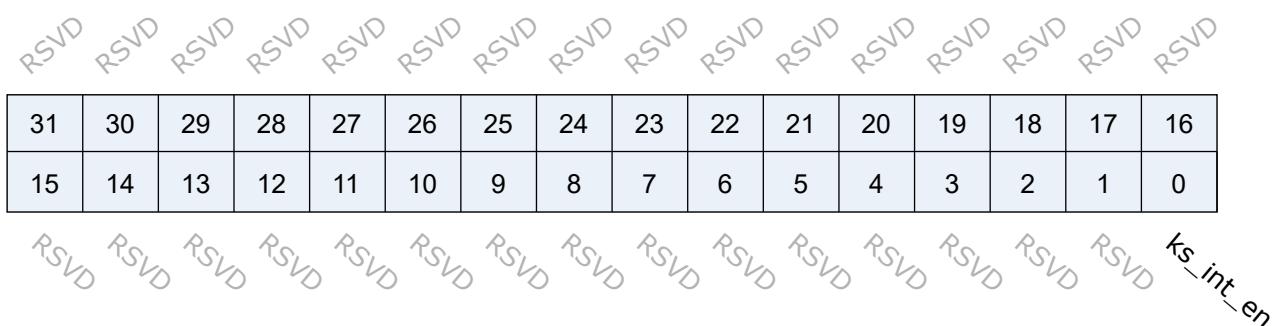
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	row_num									
															col_num
RSVD	rc_ext	deg_cnt	deg_en	ghost_en	RSVD	RSVD	RSVD	RSVD	ks_en						

位	名称	权限	复位值	描述
31:25	RSVD			
24:20	col_num	r/w	5'd19	col_num + 1
19	RSVD			

位	名称	权限	复位值	描述
18:16	row_num	r/w	3'd7	row_num + 1
15:10	RSVD			
9:8	rc_ext	r/w	2'd3	idle duration between column scans
7:4	deg_cnt	r/w	0	
3	deg_en	r/w	0	deglitch
2	ghost_en	r/w	0	ghost key event detection
1	RSVD			
0	ks_en	r/w	0	

### 15.4.2 ks\_int\_en

地址: 0x4000a910



位	名称	权限	复位值	描述
31:1	RSVD			
0	ks_int_en	r/w	1	

### 15.4.3 ks\_int\_sts

地址: 0x4000a914

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

*keycode\_valid*

位	名称	权限	复位值	描述
31:4	RSVD			
3:0	keycode_valid	r	0	

### 15.4.4 keycode\_clr

地址: 0x4000a918

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

*keycode\_clr*

位	名称	权限	复位值	描述
31:4	RSVD			
3:0	keycode_clr	w1c	0	

### 15.4.5 keycode\_value

地址: 0x4000a91c

keycode3								keycode2							
31 30 29 28 27 26 25 24								23 22 21 20 19 18 17 16							
15 14 13 12 11 10 9 8								7 6 5 4 3 2 1 0							

keycode1

keycode0

位	名称	权限	复位值	描述
31:24	keycode3	r	8'hff	
23:16	keycode2	r	8'hff	
15:8	keycode1	r	8'hff	
7:0	keycode0	r	8'hff	Col = keycode / (row_num+1) Row = keycode

## 16.1 简介

I2S 是传输数字音频数据的一种接口标准，采用序列的方式传输 2 组（左右声道）数据。

I2S 将时钟信号与数据信号分开传输，使得接收端不用从数据信号还原时钟，进而降低接收端的设计难度。

## 16.2 主要特征

- 支持主模式以及从模式
- 支持 Left-justified/Right-justified/DSP 等数据格式
- 支持 8/16/24/32 比特数据宽度
- 除单声道/双声道模式之外，同时支持四声道模式
- 支持动态静音切换功能
- 数据发送 FIFO 的宽度为 32 位，深度 16
- 数据接收 FIFO 的宽度为 32 位，深度 16

## 16.3 功能描述

引脚列表：

表 16.1: I2S 引脚

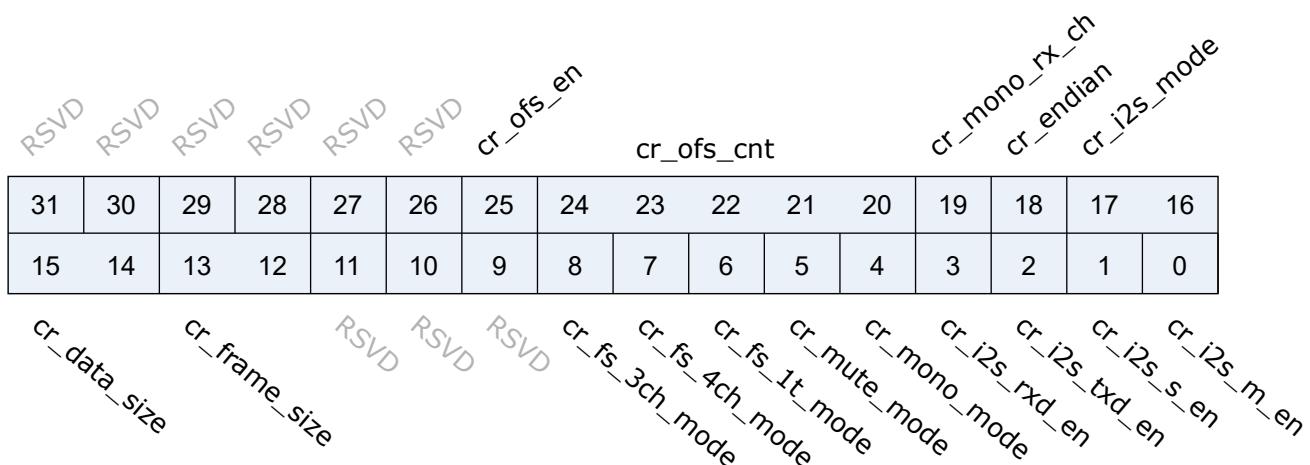
名称	类型	描述
I2Sx_DI	输入	串行数据输入
I2Sx_DO	输出	串行数据输出
I2Sx_BCLK	输入/输出	同步传输时钟，作为主机时为输出，作为从机时为输入
I2Sx_FS	输入/输出	数据起始/结束表示信号，作为主机时为输出，作为从机时为输入

## 16.4 寄存器描述

名称	描述
i2s_config	I2S configuration
i2s_int_sts	I2S interrupt status
i2s_bclk_config	I2S clock configuration
i2s_fifo_config_0	I2S FIFO configuration0
i2s_fifo_config_1	I2S DMA FIFO configuration
i2s_fifo_wdata	I2S FIFO write data
i2s_fifo_rdata	I2S FIFO read data
i2s_io_config	I2S IO configuration

### 16.4.1 i2s\_config

地址: 0x4000aa00



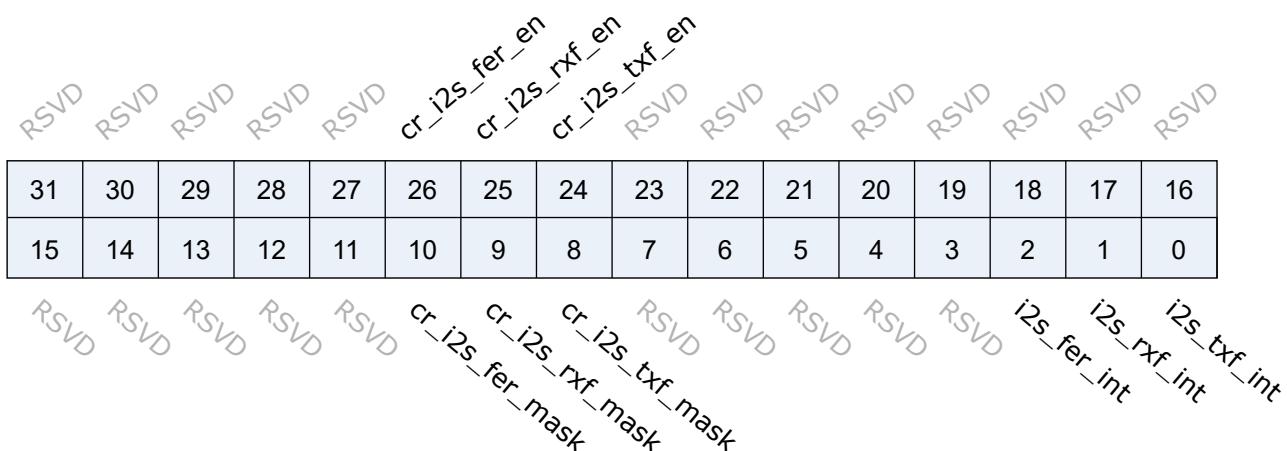
位	名称	权限	复位值	描述
31:26	RSVD			

位	名称	权限	复位值	描述
25	cr_ofs_en	r/w	1'b0	Offset enable 1'b0: Disabled, 1'b1: Enabled
24:20	cr_ofs_cnt	r/w	5'd0	Offset cycle count (unit: cycle of I2S BCLK) 5'd0: 1 cycle 5'd1: 2 cycles ...
19	cr_mono_rx_ch	r/w	1'b0	RX mono mode channel select signal 1'b0: L-channel 1'b1: R-channel
18	cr_endian	r/w	1'b0	Data endian (bit reverse) 1'b0: MSB goes out first, 1'b1: LSB goes out first
17:16	cr_i2s_mode	r/w	2'd0	2'd0: Left-Justified, 2'd1: Right-Justified, 2'd2: DSP, 2'd3: Reserved
15:14	cr_data_size	r/w	2'd1	Data bit width of each channel 2'd0: 8, 2'd1: 16, 2'd2: 24, 2'd3: 32 (bits)
13:12	cr_frame_size	r/w	2'd1	Frame size of each channel 2'd0: 8, 2'd1: 16, 2'd2: 24, 2'd3: 32 (cycles)
11:9	RSVD			
8	cr_fs_3ch_mode	r/w	1'b0	1'b0: FS 2-channel mode, 1'b1: FS 3-channel mode (DSP mode only) Note: cr_fs_3ch_mode & cr_fs_4ch_mode should NOT be enabled at the same time Note: cr_mono_mode & cr_fifo_lr_merge will be invalid in 3-channel mode Note: When 3-channel mode is enabled, frame_size must equal data_size
7	cr_fs_4ch_mode	r/w	1'b0	1'b0: FS 2-channel mode, 1'b1: FS 4-channel mode (DSP mode only) Note: cr_fs_3ch_mode & cr_fs_4ch_mode should NOT be enabled at the same time Note: When 4-channel mode is enabled, frame_size must equal data_size
6	cr_fs_1t_mode	r/w	1'b0	1'b0: FS high/low is even, 1'b1: FS only asserts for 1 cycle
5	cr_mute_mode	r/w	1'b0	1'b0: Normal mode, 1'b1: Mute mode
4	cr_mono_mode	r/w	1'b0	1'b0: Stereo mode, 1'b1: Mono mode Note: csr_mono_mode & csr_fifo_lr_merge should NOT be enabled at the same time
3	cr_i2s_rxd_en	r/w	1'b0	Enable signal of I2S RXD signal

位	名称	权限	复位值	描述
2	cr_i2s_txd_en	r/w	1'b0	Enable signal of I2S TXD signal
1	cr_i2s_s_en	r/w	1'b0	Enable signal of I2S Slave function, cannot enable both csr_i2s_m_en & csr_i2s_s_en
0	cr_i2s_m_en	r/w	1'b0	Enable signal of I2S Master function, cannot enable both csr_i2s_m_en & csr_i2s_s_en

### 16.4.2 i2s\_int\_sts

地址: 0x4000aa04



位	名称	权限	复位值	描述
31:27	RSVD			
26	cr_i2s_fer_en	r/w	1'b1	Interrupt enable of i2s_fer_int
25	cr_i2s_rxf_en	r/w	1'b1	Interrupt enable of i2s_rxf_int
24	cr_i2s_txf_en	r/w	1'b1	Interrupt enable of i2s_txf_int
23:11	RSVD			
10	cr_i2s_fer_mask	r/w	1'b1	Interrupt mask of i2s_fer_int
9	cr_i2s_rxf_mask	r/w	1'b1	Interrupt mask of i2s_rxf_int
8	cr_i2s_txf_mask	r/w	1'b1	Interrupt mask of i2s_txf_int
7:3	RSVD			
2	i2s_fer_int	r	1'b0	I2S TX/RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
1	i2s_rxf_int	r	1'b0	I2S RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto-cleared when data is popped

位	名称	权限	复位值	描述
0	i2s_txf_int	r	1'b0	I2S TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed

### 16.4.3 i2s\_bclk\_config

地址: 0x4000aa10

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

位	名称	权限	复位值	描述
31:28	RSVD			
27:16	cr_bclk_div_h	r/w	12'd1	I2S BCLK active high period (unit: cycle of i2s_clk)
15:12	RSVD			
11:0	cr_bclk_div_l	r/w	12'd1	I2S BCLK active low period (unit: cycle of i2s_clk)

### 16.4.4 i2s\_fifo\_config\_0

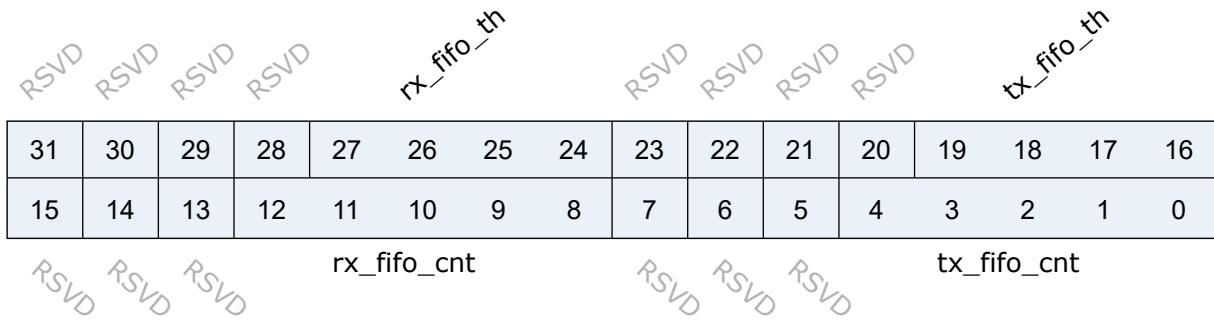
地址: 0x4000aa80

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD															
cr_fifo_24b_lj cr_fifo_lr_exchg rx_fifo_lr_merge rx_fifo_overflow rx_fifo_underflow tx_fifo_overflow tx_fifo_underflow tx_fifo_clr tx_fifo_clr i2s_dma_rx_en i2s_dma_tx_en															

位	名称	权限	复位值	描述
31:11	RSVD			
10	cr_fifo_24b_lj	r/w	1'b0	FIFO 24-bit data left-justified mode 1'b0: Right-justified, 8'h0, data[23:0] 1'b1: Left-justified, data[23:0], 8'h0 Note: Valid only when cr_data_size = 2'd2 (24-bit)
9	cr_fifo_lr_exchg	r/w	1'b0	The position of L/R channel data within each entry is exchanged if this bit is enabled Can only be enabled if data size is 8 or 16 bits and csr_fifo_lr_merge is enabled
8	cr_fifo_lr_merge	r/w	1'b0	Each FIFO entry contains both L/R channel data if this bit is enabled Can only be enabled if data size is 8 or 16 bits Note: cr_fifo_lr_merge & cr_mono_mode should NOT be enabled at the same time Note: cr_fifo_lr_merge & cr_fifo_l_shift should NOT be enabled at the same time
7	rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	i2s_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	i2s_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

#### 16.4.5 i2s\_fifo\_config\_1

地址: 0x4000aa84



位	名称	权限	复位值	描述
31:28	RSVD			
27:24	rx_fifo_th	r/w	4'd0	RX FIFO threshold, dma_rx_req will not be asserted if tx_fifo_cnt is less than this value
23:20	RSVD			
19:16	tx_fifo_th	r/w	4'd0	TX FIFO threshold, dma_tx_req will not be asserted if tx_fifo_cnt is less than this value
15:13	RSVD			
12:8	rx_fifo_cnt	r	5'd0	RX FIFO available count
7:5	RSVD			
4:0	tx_fifo_cnt	r	5'd16	TX FIFO available count

#### 16.4.6 i2s\_fifo\_wdata

地址: 0x4000aa88

i2s\_fifo\_wdata

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

i2s\_fifo\_wdata

位	名称	权限	复位值	描述
31:0	i2s_fifo_wdata	w	x	

#### 16.4.7 i2s\_fifo\_rdata

地址: 0x4000aa8c

i2s\_fifo\_rdata

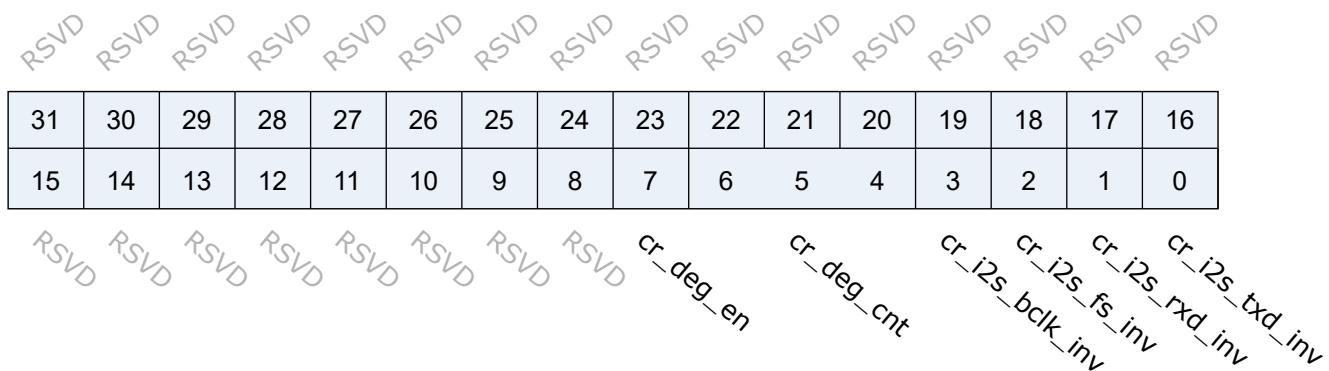
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

i2s\_fifo\_rdata

位	名称	权限	复位值	描述
31:0	i2s_fifo_rdata	r	32'h0	

### 16.4.8 i2s\_io\_config

地址: 0x4000aafc



位	名称	权限	复位值	描述
31:8	RSVD			
7	cr_deg_en	r/w	1'b0	Deglitch enable (for all th input pins) 1'b0: Disabled, 1'b1: Enabled
6:4	cr_deg_cnt	r/w	3'd0	Deglitch cycle count (unit: cycle of I2S kernel clock) 3'd0: 1 cycle 3'd1: 2 cycles ...
3	cr_i2s_bclk_inv	r/w	1'b0	Inverse BCLK signal 0: No inverse, 1: Inverse
2	cr_i2s_fs_inv	r/w	1'b0	Inverse FS signal 0: No inverse, 1: Inverse
1	cr_i2s_rxd_inv	r/w	1'b0	Inverse RXD signal 0: No inverse, 1: Inverse
0	cr_i2s_txd_inv	r/w	1'b0	Inverse TXD signal 0: No inverse, 1: Inverse

## 17.1 简介

EMAC 模块是一个兼容 IEEE 802.3 的 10/100Mbps 以太网 MAC(Ethernet Media Access Controller)。其包含状态及控制寄存器组，收发模块，收发缓冲描述符组，主机接口，MDIO，物理层芯片 (PHY) 接口。

状态及控制寄存器组包含了 EMAC 的状态位及控制位，是与用户程序的接口，负责控制数据收发，并汇报状态。

收发模块负责根据收发描述符内的控制字，从指定内存位置取得数据帧，添加前导，CRC，并扩充短的帧后通过 PHY 发出；或是从 PHY 接收数据，并根据收发缓冲描述符，将数据放入指定内存。收发完成后配置相关的事件标志。如果使能了事件中断，将通过中断请求到主机进行处理。

MDIO 及 MII/RMII 接口负责与 PHY 进行通信，包括读写 PHY 的寄存器，以及数据包的收发。

## 17.2 主要特征

- 兼容 IEEE 802.3 定义的 MAC 层功能
- 支持 IEEE 802.3 定义的 MII、RMII 接口的 PHY
- 通过 MDIO 与 PHY 交互
- 支持 10Mbps 与 100Mbps 以太网
- 支持半双工与全双工
- 在全双工模式下，支持自动流控及生成控制帧
- 在半双工模式下，支持碰撞检测及重传
- 支持 CRC 的生成及校验
- 数据帧前导生成及移除
- 发送时，自动扩展短的数据帧
- 检测过长或过短的数据帧 (长度限制)

- 可传输长数据帧 (> 标准以太帧长度)
- 自动丢弃重发次数超限或帧间隙过小的包
- 广播包过滤
- 用于保存多达 128 个 BD(Buffer Descriptor) 的内部 RAM
- 发送/接收的各种事件标志
- 在事件发生时产生对应中断

### 17.3 功能描述

EMAC 模块的组成如下图。

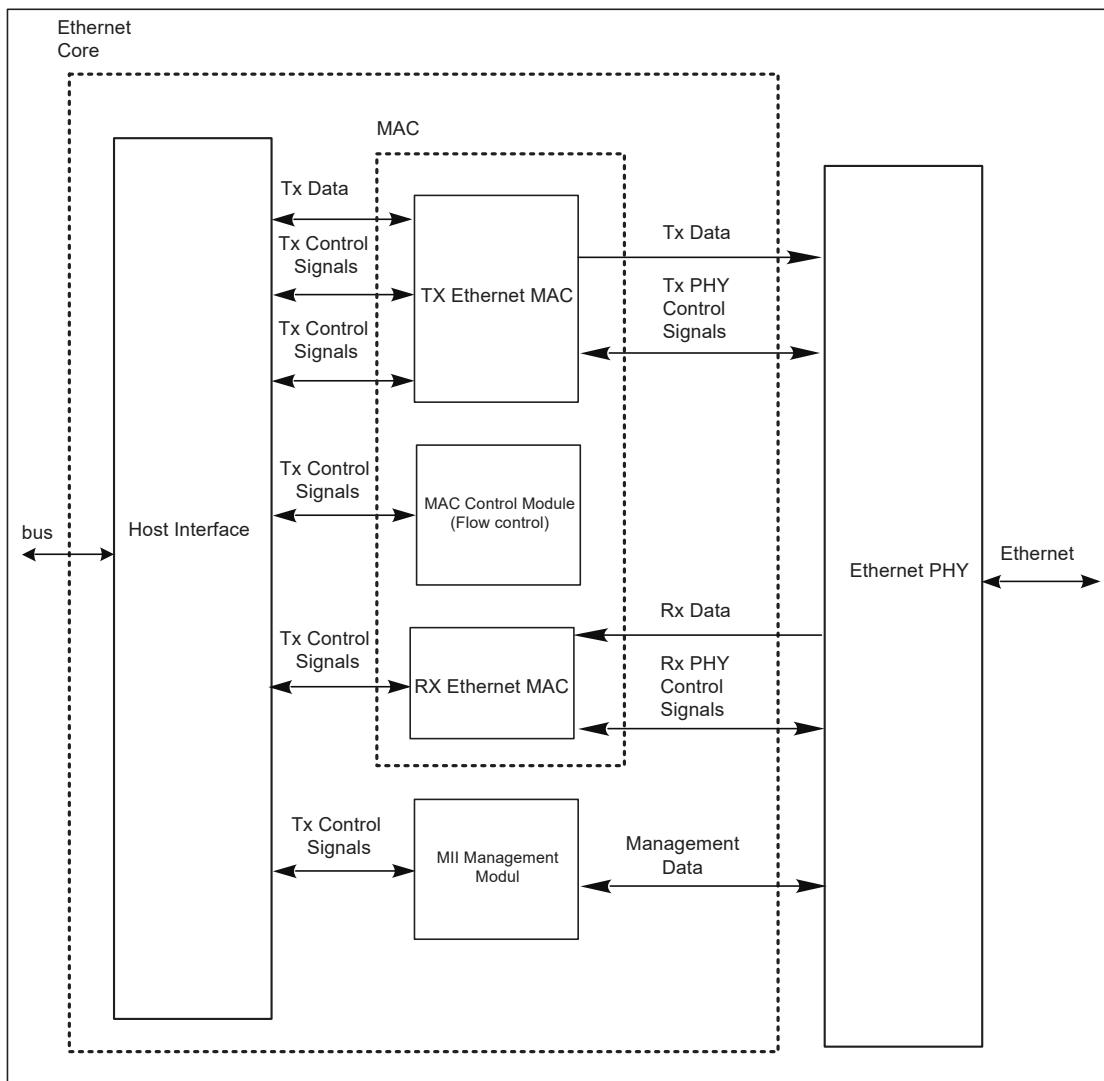


图 17.1: EMAC 框图

模块的控制寄存器通过 MDIO，可以读写 PHY 的寄存器从而实现配置、选择模式(半/全双工)、发起协商等操作。接收

模块过滤并检查收到的数据帧：是否有合法的前导，FCS，长度等，并根据描述符，将数据帧存放到指定缓冲地址。发送模块根据数据缓冲描述符，从内存中取得数据，添加前导，FCS，pad 等，然后根据 CSMA/CD 协议，将数据发出。如果检测到 CRS，将会延迟重试。收发缓冲描述符组连接到外部的 RAM，此 RAM 用于保存发送和接收的以太网数据帧。每个描述符包含相应的控制状态字以及对应的缓冲内存地址。描述符一共有 128 组，可以灵活分配，用于发送或者接收。

## 17.4 时钟

EMAC 模块需要一路时钟用于同步收发 (100Mbps 时，25MHz(MII) 或 50MHz(RMII); 10Mbps 时，2.5MHz)。此时钟必须在 EMAC 与 PHY 之间同步。

## 17.5 收发缓冲描述符 (BD, Buffer Descriptor)

收发缓冲描述符，用于提供 EAMC 与数据帧缓存地址信息之间的关联，对收发数据帧进行控制，以及提供收发状态提示。每个描述符由两个连续的 word(32bit) 构成，低地址的 word 提供了本 buffer 包含的数据帧的长度，控制及状态位；高地址的 word 是内存指针。具体的 BD 描述可以参考寄存器描述章节。需要注意的是：对于 BD，需要按 word 写入。EMAC 模块支持 128 个 BD，由发送/接收逻辑共享，可自由组合。但发送 BD 总是占据前面的连续区域(个数由 MAC\_TX\_BD\_NUM 寄存器中的 TXBDNUM 域来指定)。EMAC 按照 BD 的顺序，循环处理发送/接收 BD，直到遇到标记为 WR 的 BD 就回绕到发送/接收各自的首个 BD。

## 17.6 PHY 交互

PHY 交互寄存器组提供了 PHY 交互需要的命令及数据通信方式。EMAC 通过 MDIO 控制 PHY 的工作模式，并保证两者匹配(速率，全/半双工)。数据包通过 MII/RMII 接口在 EMAC 与 PHY 之间交互，可以通过 EMAC 的模式寄存器 (EMAC\_MODE) 中的 RMII\_EN 选择。当此 bit 为 1，则选择 RMII 模式，否则就是 MII 模式。MII 及 RMII 模式均支持 IEEE 802.3u 标准中指定的 10Mbps 与 100Mbps 的传输速率。MII 及 RMII 的传输信号描述与下表。

表 17.1: 传输信号

名称	MII	RMII
EXTCK_EREFCK	ETXCK: 发送时钟信号	EREFCK: 参考时钟
ECRS	ECRS: 载波探测	-
ECOL	ECOL: 碰撞检测	-
ERXDV	ERXDV: 数据 valid	ECRSDV: 载波检测/数据 valid
ERX0-ERX3	ERX0-ERX3: 4-bit 接收数据	ERX0-ERX1: 2-bit 接收数据
ERXER	ERXER: 接收错误指示	ERXER: 接收错误指示
ERXCK	ERXCK: 接收时钟信号	-
ETXEN	ETXEN: 发送使能	ETXEN: 发送使能
ETX0-ETX3	ETX0-ETX3: 4-bit 发送数据	ETX0-ETX1: 2-bit 发送数据
ETXER	ETXER: 发送错误指示	-
EMDC	MDIO Clock	MDIO Clock

表 17.1: 传输信号 (continued)

名称	MII	RMII
EMDIO	MDIO Data Input Output	MDIO Data Input Output

RMII 接口引脚较少，使用 2-bit 数据线用于收发，在 100Mbps 速率时，需要提供 50MHz 的参考时钟。

## 17.7 编程流程

### 17.7.1 PHY 初始化

- 根据 PHY 类型，设置 EMAC\_MODE 寄存器中的 RMII\_EN 位来选择合适的连接方式
- 设置 EMAC 的 MAC 地址到 EMAC\_MAC\_ADDR0 与 EMAC\_MAC\_ADDR1 中
- 通过编程 EMAC\_MIIMODE 寄存器中的域 CLKDIV，为 MDIO 部分设置合适的时钟
- 设置对应 PHY 的地址到寄存器 EMAC\_MIIADDRESS 的域 FIAD 中
- 根据 PHY 的手册，通过 EMAC\_MIICOMMAND 与 EMAC\_MIITX\_DATA 寄存器发送命令
- 读取 PHY 的数据会保存在 EMAC\_MIIRX\_DATA 寄存器中
- 通过 EMAC\_MIISTATUS 寄存器可以查询与 PHY 命令交互的状态

基础的交互完成后，应当使 PHY 进入自动协商状态。协商完成之后，根据协商结果编程模式到 EMAC\_MODE 寄存器中的 FULLD 位。

### 17.7.2 发送数据帧

- 配置 EMAC\_MODE 寄存器中数据帧格式、间隔等位域
- 通过配置 EMAC\_TX\_BD\_NUM 寄存器中的 TXBDNUM 域来指定发送所使用的 BD 的个数，那么剩余的就是 RX 的 BD
- 在内存中准备好需要发送的数据帧
- 将数据帧的地址填写到对应发送 BD 的数据指针域 (word 1) 中
- 清空对应发送 BD 的控制与状态域 (word 0) 中的状态标记，并设置控制域 (CRC 使能，PAD 使能，中断使能等)
- 写入数据帧长度，并设置好 RD 域，告知 EMAC 此 BD 数据需要发送；如需要，设置上 IRQ 位，以使能中断
- 特别的，如果是最后一个发送的 BD，需要设置上 WR 位，EMAC 会在处理完这个 BD 之后“回绕”到第一个发送 BD 进行处理
- 如果有多个 BD 需要发送，则重复设置 BD 的步骤以填充所有的发送 BD

- 如果需要使能发送中断，还需要配置 **EMAC\_INT\_MASK** 寄存器中的 TX 相关位
- 配置 **EMAC\_MODE** 寄存器中的 **TXEN** 位，以使能发送
- 如果使能了中断，在发送的中断中，可用通过 **EMAC\_TX\_BD\_NUM** 寄存器中的 **TXBDNUM** 域获取当前的 BD
- 根据当前 BD 的状态字进行相应的处理
- 数据已被发送出去的 BD，其控制域中的 RD 位会被硬件清零，且不会被再次发送；需要填充新数据后，置位 RD，此 BD 即可再次用于发送

### 17.7.3 接收数据帧

- 配置 **EMAC\_MODE** 寄存器中数据帧格式、间隔等位域
- 通过配置 **EMAC\_TX\_BD\_NUM** 寄存器中的 **TXBDNUM** 域来指定发送所使用的 BD 的个数，那么剩余的就是 RX 的 BD
- 在内存中准备好接收数据的区域
- 将数据帧的地址填写到对应接收 BD 的数据指针域 (**word 1**) 中
- 清空对应发送 BD 的控制与状态域 (**word 0**) 中的状态标记，并设置控制域 (中断使能等)
- 写入可接收的数据帧长度，并设置好 E 位域，告知 EMAC 此 BD 空闲，可以用于数据接收；如需要，设置上 **IRQ** 位，以使能中断
- 特别的，如果是最后一个有效接收 BD，需要设置上 **WR** 位，EMAC 会在处理完这个 BD 之后“回绕”到第一个接收 BD 进行处理
- 如果有多个 BD 可供接收数据，则重复设置 BD 的步骤以填充所有的 BD
- 如果需要使能接收中断，还需要配置 **EMAC\_INT\_MASK** 寄存器中的 RX 相关位
- 配置 **EMAC\_MODE** 寄存器中的 **RXEN** 位，以使能接收
- 如果使能了中断，在接收的中断中，可用通过 **EMAC\_TX\_BD\_NUM** 寄存器中的 **RXBDNUM** 域获取当前的 BD
- 根据当前 BD 的状态字进行相应的处理
- 接收完成的 BD，其控制域中的 E 位会被硬件清零，且不会被再次用于接收；需要取走数据，置位 E，此 BD 即可再次用于接收

## 17.8 寄存器描述

名称	描述
MODE	EMAC configuration
INT_SOURCE	EMAC transmit control
INT_MASK	EMAC interrupt mask
IPGT	Inter packet gap
PACKETLEN	Frame length
COLLCONFIG	Collision configuration
TX_BD_NUM	TX buffer descriptors number
MIIMODE	Management Data configuration
MIICOMMAND	Trigger command
MIIADDRESS	Register address
MIITX_DATA	Control data to be written to PHY
MIIRX_DATA	Received data from PHY
MIISTATUS	MIIM I/F status
MAC_ADDR0	Ethernet MAC address0
MAC_ADDR1	Ethernet MAC address1
HASH0_ADDR	Lower 32-bit of HASH register
HASH1_ADDR	Upper 32-bit of HASH register
TXCTRL	TX control

### 17.8.1 MODE

地址: 0x4000d000

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

PAD    HUGEN    CRCEN    RSVD    RSVD    FULLD    RSVD    RSVD    IFG    PRO    RSVD    BRO    NOPRE    TXEN    RXEN

位	名称	权限	复位值	描述
31:18	RSVD			
17	RMII_EN	r/w	1'b0	RMII mode enable 0: MII PHY I/F is used 1: RMII PHY I/F is used
16	RECSMALL	r/w	1'b0	Receive small frame enable 0: Frames smaller than MINFL are ignored. 1: Frames smaller than MINFL are accepted.
15	PAD	r/w	1'b1	Padding enable 0: Do not add pads to frames shorter than MINFL. 1: Add pads to short frames, until the length equals MINFL.
14	HUGEN	r/w	1'b0	Huge frames enable 0: The maximum frame length is MAXFL. All additional bytes are dropped. 1: Frame size is not limited by MAXFL and can be up to 64K bytes.
13	CRCEN	r/w	1'b1	CRC Enable 0: TX MAC does not append CRC field. 1: TX MAC will append CRC field to every frame.
12:11	RSVD			
10	FULLD	r/w	1'b0	Full duplex 0: Half duplex mode. 1: Full duplex mode.
9:7	RSVD			
6	IFG	r/w	1'b0	Inter frame gap check 0: IFG is verified before each frame be received. 1: All frames are received regardless to IFG requirement.
5	PRO	r/w	1'b0	Promiscuous mode enable 0: The destination address is checked before receiving. 1: All frames received regardless of the address.

位	名称	权限	复位值	描述
4	RSVD			
3	BRO	r/w	1'b1	<p>Broadcast address enable</p> <p>0: Reject all frames containing the broadcast address unless the PRO bit is asserted.</p> <p>1: Receive all frames containing broadcast address.</p>
2	NOPRE	r/w	1'b0	<p>No preamble mode</p> <p>0: 7-byte preamble will be sent.</p> <p>1: No preamble will be sent.</p>
1	TXEN	r/w	1'b0	<p>Transmit enable</p> <p>0: Transmitter is disabled.</p> <p>1: Transmitter is enabled.</p> <p>If TX_BD_NUM equals 0x0 (zero buffer descriptors are used), then the transmitter is disabled regardless of TXEN.</p>
0	RXEN	r/w	1'b0	<p>Receiver enable</p> <p>0: Receiver is disabled.</p> <p>1: Receiver is enabled.</p> <p>If TX_BD_NUM equals 0x80 (all buffer descriptors are used for TX), then the receiver is disabled regardless of RXEN.</p>

## 17.8.2 INT\_SOURCE

地址: 0x4000d004

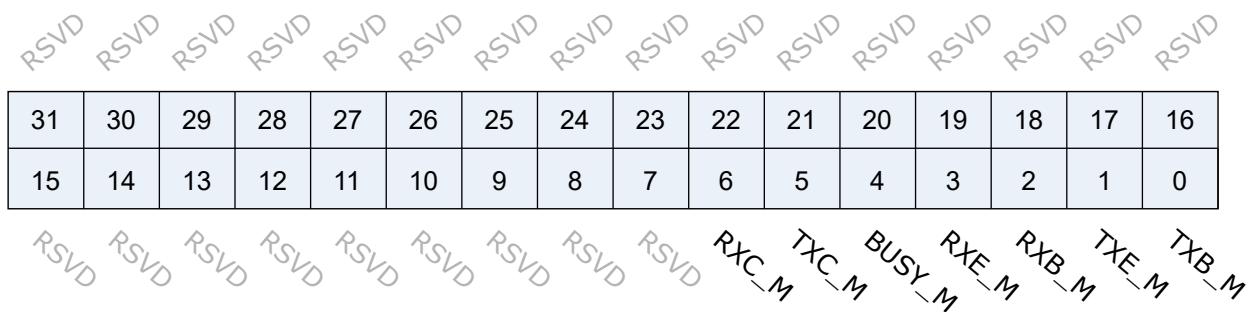
RSVD																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RSVD		RSVD		RSVD		RSVD		RSVD		RXC		TXC		BUSY		RXE	
										RXB		TXF		TXB			

位	名称	权限	复位值	描述
31:7	RSVD			
6	RXC	r/w	1'b0	<p>Receive control frame</p> <p>This bit indicates that the control frame was received. It is cleared by writing 1 to it.</p> <p>Bit RXFLOW in the CTRLMODE register must be set to 1 in order to get the RXC bit set.</p>

位	名称	权限	复位值	描述
5	TXC	r/w	1'b0	<p>Transmit control frame</p> <p>This bit indicates that a control frame was transmitted. It is cleared by writing 1 to it.</p> <p>Bit TXFLOW in the CTRLMODE register must be set to 1 in order to get the TXC bit set.</p>
4	BUSY	r/w	1'b0	<p>Busy</p> <p>This bit indicates that RX packet is being received and there is no empty buffer descriptor to use. It is cleared by writing 1 to it.</p> <p>This bit appears regardless to the IRQ bits in the Receive Buffer Descriptor.</p>
3	RXE	r/w	1'b0	<p>Receive error</p> <p>This bit indicates that an error occurred while receiving data (overrun, receiver error, dribble nibble, too long, &gt;64K, CRC error, bus error or late collision. It is cleared by writing 1 to it.</p> <p>This bit appears only when IRQ bit is set in the Receive Buffer Descriptor.</p>
2	RXB	r/w	1'b0	<p>Receive frame</p> <p>This bit indicates that a frame was received. It is cleared by writing 1 to it.</p> <p>This bit appears only when IRQ bit is set in the Receive Buffer Descriptor.</p>
1	TXE	r/w	1'b0	<p>Transmit error</p> <p>This bit indicates that a buffer was not transmitted due to a transmit error (underrun, retransmission limit, late collision, bus error or defer timeout). It is cleared by writing 1 to it.</p> <p>This bit appears only when IRQ bit is set in the Transmit Buffer Descriptor.</p>
0	TXB	r/w	1'b0	<p>Transmit buffer</p> <p>This bit indicates that a buffer has been transmitted. It is cleared by writing 1 to it.</p> <p>This bit appears only when IRQ bit is set in the Transmit Buffer Descriptor.</p>

### 17.8.3 INT\_MASK

地址: 0x4000d008



位	名称	权限	复位值	描述
31:7	RSVD			
6	RXC_M	r/w	1'b1	Receive control frame mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
5	TXC_M	r/w	1'b1	Transmit control frame mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
4	BUSY_M	r/w	1'b1	Busy mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
3	RXE_M	r/w	1'b1	Receive error mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
2	RXB_M	r/w	1'b1	Receive frame mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
1	TXE_M	r/w	1'b1	Transmit error mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
0	TXB_M	r/w	1'b1	Transmit buffer mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked

### 17.8.4 IPGT

地址: 0x4000d00c

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

IPGT

位	名称	权限	复位值	描述
31:7	RSVD			
6:0	IPGT	r/w	7'h18	Inter packet gap The recommended value is 0x18 (24 clock cycles), which equals 9.6 us for 10 Mbps and 0.96 us for 100 Mbps mode

### 17.8.5 PACKETLEN

地址: 0x4000d018

MINFL

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

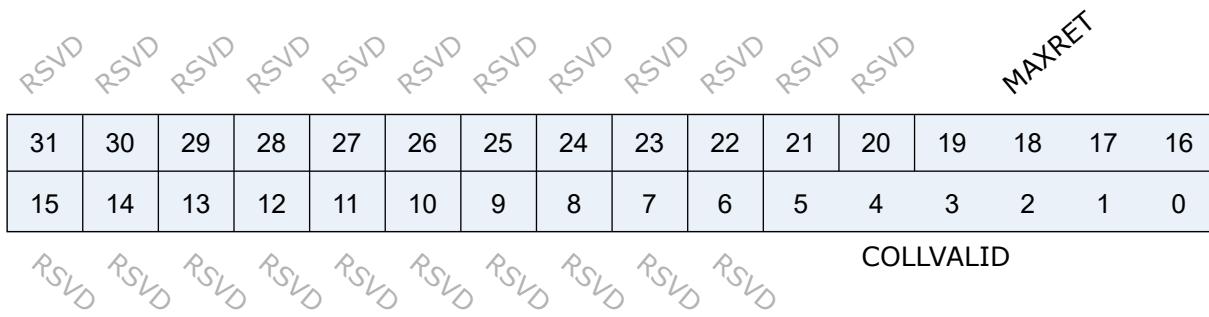
MAXFL

位	名称	权限	复位值	描述
31:16	MINFL	r/w	16'h40	Minimum frame length The minimum Ethernet packet is 64 bytes long (0x40). To receive small packets, assert the RECSMALL bit or change the MINFL value. To transmit small packets, assert the PAD bit or change the MINFL value.

位	名称	权限	复位值	描述
15:0	MAXFL	r/w	16'h600	<p>Maximum frame length</p> <p>The maximum Ethernet packet is 1518 bytes long. To support this and to have some additional space for tags, a default maximum packet length equals to 1536 bytes (0x600).</p> <p>For bigger packets, you can assert the HUGEN bit or increase the value of MAXFL field.</p>

### 17.8.6 COLLCONFIG

地址: 0x4000d01c



位	名称	权限	复位值	描述
31:20	RSVD			
19:16	MAXRET	r/w	4'hF	<p>Maximum retry</p> <p>This field specifies the maximum number of consequential retransmission attempts after the collision is detected.</p> <p>When the maximum number has been reached, the TX MAC reports an error and stops transmitting the current packet.</p> <p>According to the Ethernet standard, the MAXRET default value is set to 0xf (15).</p>
15:6	RSVD			
5:0	COLLVALID	r/w	6'h3F	<p>Collision valid</p> <p>This field specifies a collision time window. A collision that occurs later than the time window is reported as a "Late Collisions" and transmission of the current packet is aborted.</p>

### 17.8.7 TX\_BD\_NUM

地址: 0x4000d020

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

位	名称	权限	复位值	描述
31	RSVD			
30:24	RXBDPTR	r	7'h0	RX buffer descriptors (BD) pointer, pointing at the RXBD currently being used
23	RSVD			
22:16	TXBDPTR	r	7'h0	TX buffer descriptors (BD) pointer, pointing at the TXBD currently being used
15:8	RSVD			
7:0	TXBDNUM	r/w	8'h40	TX buffer descriptors (BD) number Number of TX BD. TX and RX share 128 (0x80) descriptors, so the number of RX BD equals 0x80 - TXBDNUM. The maximum number of TXBDNUM is 0x80. Values greater than 0x80 cannot be written into this register.

### 17.8.8 MIIMODE

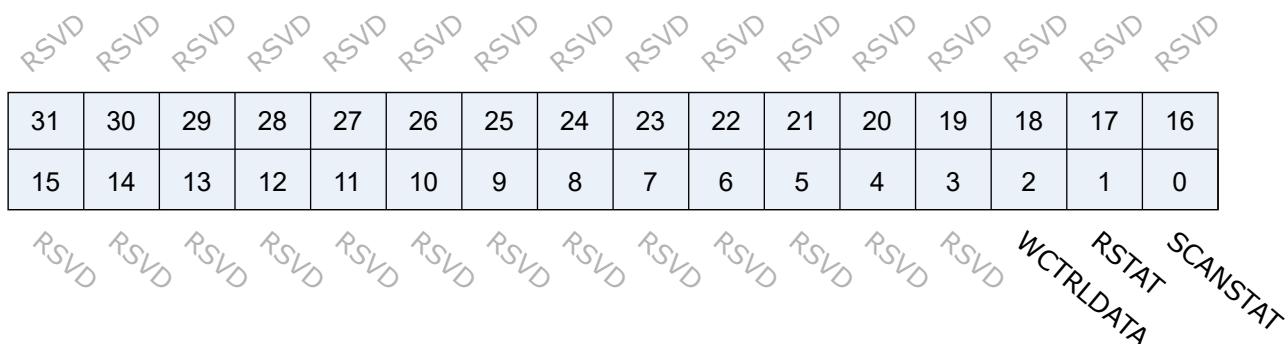
地址: 0x4000d028

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

位	名称	权限	复位值	描述
31:9	RSVD			
8	MIINOPRE	r/w	1'b0	No preamble for Management Data (MD) 0: 32-bit preamble will be sent. 1: No preamble will be sent.
7:0	CLKDIV	r/w	8'h64	Clock divider for Management Data Clock (MDC) The source clock is bus clock and can be divided by any even number.

### 17.8.9 MIICOMMAND

地址: 0x4000d02c



位	名称	权限	复位值	描述
31:3	RSVD			
2	WCTRLDATA	r/w	1'b0	Write control data, setting this bit to 1 will trigger the command (auto cleared) Note: [2]/[1]/[0] cannot be asserted at the same time, execute one command at a time
1	RSTAT	r/w	1'b0	Read status, setting this bit to 1 will trigger the command (auto cleared) Note: [2]/[1]/[0] cannot be asserted at the same time, execute one command at a time
0	SCANSTAT	r/w	1'b0	Scan status, setting this bit to 1 will trigger the command (auto cleared) Note: [2]/[1]/[0] cannot be asserted at the same time, execute one command at a time

## 17.8.10 MIIADDRESS

地址: 0x4000d030

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

RGAD

FIAD

位	名称	权限	复位值	描述
31:13	RSVD			
12:8	RGAD	r/w	5'h0	Register Address
7:5	RSVD			
4:0	FIAD	r/w	5'h0	PHY Address

## 17.8.11 MIITX\_DATA

地址: 0x4000d034

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

CTRLDATA

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	CTRLDATA	r/w	16'h0	Control Data to be written to PHY

## 17.8.12 MIIRX\_DATA

地址: 0x4000d038

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

PRSD

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	PRSD	r	16'h0	Received Data from PHY

### 17.8.13 MIISTATUS

地址: 0x4000d03c

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

*RSVD RSVD RSVD*

位	名称	权限	复位值	描述
31:2	RSVD			
1	MIIM_BUSY	r	1'b0	MIIM I/F busy signal 0: The MIIM I/F is ready. 1: The MIIM I/F is busy.
0	MIIM_LINKFAIL	r	1'b0	MIIM I/F link fail signal

### 17.8.14 MAC\_ADDR0

地址: 0x4000d040

MAC_B2								MAC_B3							
31	30	29	28	27	26	25	24	MAC_B3							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MAC\_B4                                    MAC\_B5

位	名称	权限	复位值	描述
31:24	MAC_B2	r/w	8'd0	Ethernet MAC address byte 2
23:16	MAC_B3	r/w	8'd0	Ethernet MAC address byte 3
15:8	MAC_B4	r/w	8'd0	Ethernet MAC address byte 4
7:0	MAC_B5	r/w	8'd0	Ethernet MAC address byte 5

### 17.8.15 MAC\_ADDR1

地址: 0x4000d044

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

MAC\_B0

MAC\_B1

位	名称	权限	复位值	描述
31:16	RSVD			
15:8	MAC_B0	r/w	8'd0	Ethernet MAC address byte 0
7:0	MAC_B1	r/w	8'd0	Ethernet MAC address byte 1

### 17.8.16 HASH0\_ADDR

地址: 0x4000d048

HASH0

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

HASH0

位	名称	权限	复位值	描述
31:0	HASH0	r/w	32'h0	Lower 32-bit of HASH register

### 17.8.17 HASH1\_ADDR

地址: 0x4000d04c

HASH1

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

HASH1

位	名称	权限	复位值	描述
31:0	HASH1	r/w	32'h0	Upper 32-bit of HASH register

### 17.8.18 TXCTRL

地址: 0x4000d050

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

TXPAUSETV

位	名称	权限	复位值	描述
31:17	RSVD			
16	TXPAUSERQ	r/w	1'b0	TX Pause Request Writing 1 to this bit starts sending control frame and is automatically cleared to zero.
15:0	TXPAUSETV	r/w	16'h0	TX Pause Timer Value The value that is sent in the pause control frame.

## 18.1 简介

USB(Universal Serial Bus) 通用串行总线，完全支持 USB1.1 全速设备，对 USB2.0 部分支持。

## 18.2 主要特征

- 支持 USB full speed device-mode
- 支持 8 个双向端点：EP0 可配置为控制/批量/中断/同步端点，EP1-EP7 可配置为批量/中断/同步端点
- 每个端点均包含 TX、RX 两个方向的 FIFO，FIFO 深度 64 字节，并且支持 DMA
- 支持内部 transceiver
- 支持 suspend/resume
- 支持 LPM
- 支持多种 USB 中断配置

## 18.3 功能描述

### 18.3.1 USB 使用步骤

1. 配置内部 transceiver，寄存器在 0x40000228 和 0x4000022C
2. 配置 usb\_config 及各个端点的 epx\_config
3. 配置 USB 中断相关寄存器
4. 配置 USB DMA 相关（可选）
5. 配置 GPIO 为 USB 功能（内部 transceiver---function 为 10）
6. 置一 0x40000228[20]usb\_enum 以触发主机端枚举流程

### 18.3.2 部分寄存器配置及功能描述

- **swrdy:** 只读，仅当此位为 0 时，才可以向 `cr_usb_ep0_sw_rdy` 写入 1
- **crsr:** 写 1 自动清零，当软件允许下一包回复 ACK 时向此域写 1，则仅下一包会回复 ACK，对于 OUT/IN 事务数据会收入 FIFO/从 FIFO 发出（FIFO 放行一次）
- **e0snko:** 需要置一，代表 OUT 事务会默认回复 NAK，数据不会进入 FIFO（FIFO 不放行）
- **e0snki:** 需要置一，代表 IN 事务会默认回复 NAK，数据不会从 FIFO 发出（FIFO 不放行）
- **e0ss:** 写 1 自动清零，当软件允许下一包回复 STALL 时向此域写 1，则仅下一包会回复 STALL
- **epxdit:** “令牌包 => 触发 `xxx_cmd_int` => 数据包 => 触发 `xxx_done_int` => 握手包”
- **epxcxit:** “令牌包 => 触发 `xxx_cmd_int` => 数据包 => 触发 `xxx_done_int` => 握手包”
- **ep0odit:** “令牌包 => 触发 `xxx_cmd_int` => 数据包 => 触发 `xxx_done_int` => 握手包”
- **ep0ocit:** “令牌包 => 触发 `xxx_cmd_int` => 数据包 => 触发 `xxx_done_int` => 握手包”
- **ep0idit:** “令牌包 => 触发 `xxx_cmd_int` => 数据包 => 触发 `xxx_done_int` => 握手包”
- **ep0icit:** “令牌包 => 触发 `xxx_cmd_int` => 数据包 => 触发 `xxx_done_int` => 握手包”
- **ep0sdit:** “令牌包 => 触发 `xxx_cmd_int` => 数据包 => 触发 `xxx_done_int` => 握手包”
- **ep0scit:** “令牌包 => 触发 `xxx_cmd_int` => 数据包 => 触发 `xxx_done_int` => 握手包”
- **exrs:** 只读，仅当此位为 0 时，才可以向 `cr_epx_rdy` 写入 1
- **exr:** 写 1 自动清零，当软件允许下一包回复 ACK 时向此域写 1，则仅下一包会回复 ACK（FIFO 放行一次）
- **exn:** 需要置一，代表事务会默认回复 NAK，IN/OUT 取决于当前端点的传输方向配置（FIFO 不放行）
- **exs:** 当软件希望挂起此端点时置一，置一后此端点永远只回复 STALL

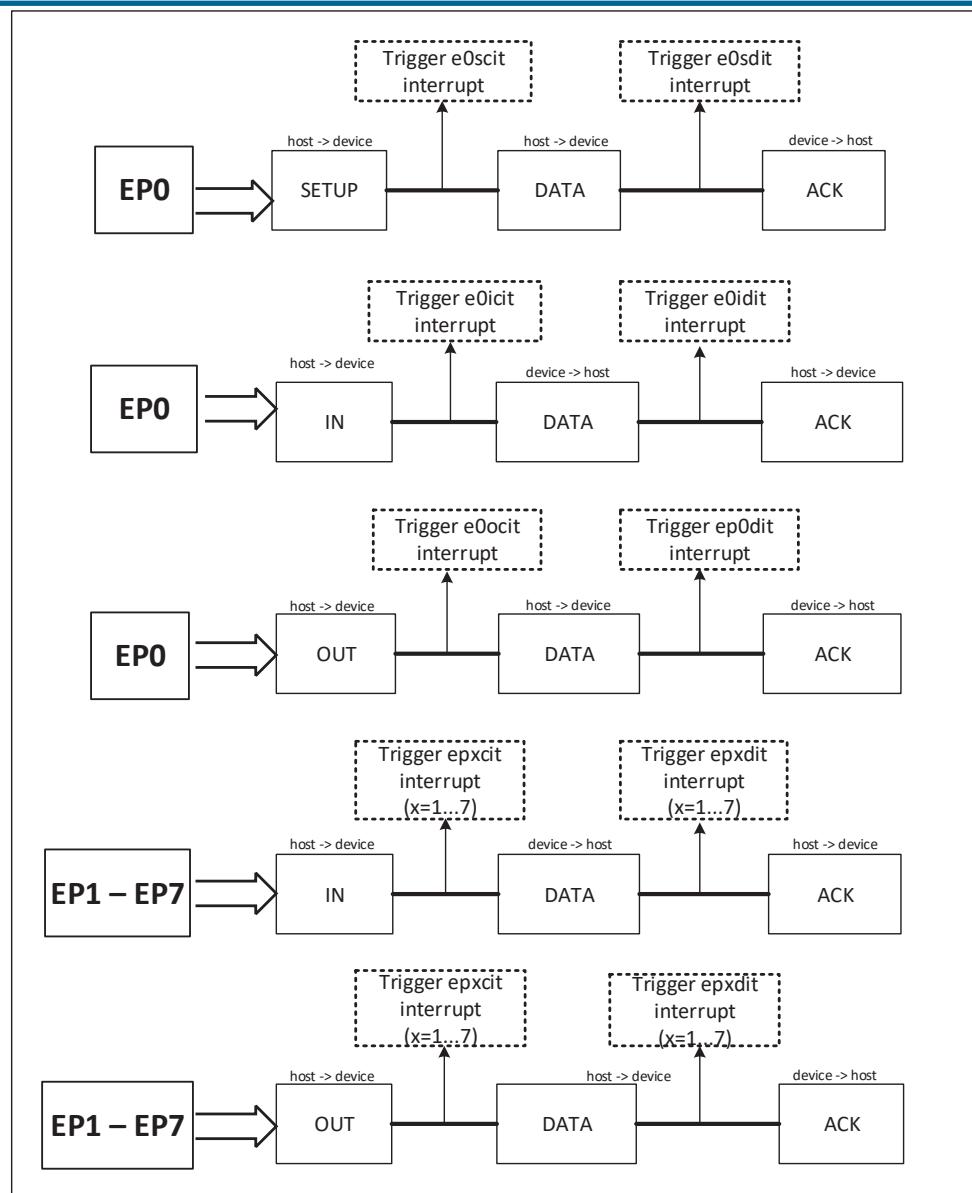


图 18.1: USB 中断触发方式

### 18.3.3 USB 枚举阶段中断处理流程

- 首先是 10ms 以上的 reset，会触发 reset 中断。
- 当 reset 结束时，会触发 reset end 中断。
- 枚举过程的 SETUP 事务、IN 事务、OUT 事务会分别触发 e0sdit、e0icit、ep0dit。
- 枚举结束后，主机与特定端点 EPx 之间的 IN 事务、OUT 事务会分别触发 epxxit、epxdit。

### 18.3.4 各传输事务的寄存器操作流程

- 控制传输——**SETUP** 事务数据接收:
  - 进入中断
    - 判断 e0sdit 中断标志位
    - e0rfr 内存放的就是 setup 事务所传来的数据, 根据 e0rfc 读取 e0rfr 即可获得
    - 置一 crsr 以放行后续事务
    - 清除中断标志位
    - 退出中断
- 控制传输——**IN** 事务数据发送:
  - 进入中断
    - 判断 e0idit 中断标志位
    - 等待 swrdy 为 0 时才可以向 e0tfw 内写入数据
    - 置一 crsr 以放行后续事务
    - 清除中断标志位
    - 退出中断
- 控制传输——**OUT** 事务数据接收:
  - 进入中断
    - 判断 ep0dit 中断标志位
    - e0rfr 内存放的就是 OUT 事务所传来的数据, 根据 e0rfc 读取 e0rfr 即可获得
    - 置一 crsr 以放行后续事务
    - 清除中断标志位
    - 退出中断
- **EPx(x=1...7)**——**IN** 事务数据发送:
  - 进入中断
    - 判断 epxcit 中断标志位
    - 等待 exrs 为 0 时才可以向 extfw 内写入数据 (仅当只发送 1 个字节时, exs 需要修改为 1)
    - 置一 exr 以放行后续事务
    - 清除中断标志位
    - 退出中断
- **EPx(x=1...7)**——**OUT** 事务数据接收:
  - 进入中断
    - 判断 epxdit 中断标志位
    - exrfr 内存放的就是 OUT 事务所传来的数据, 根据 exrfc 读取 exrfr 即可获得

- 置一 exrs 以放行后续事务
- 清除中断标志位
- 退出中断

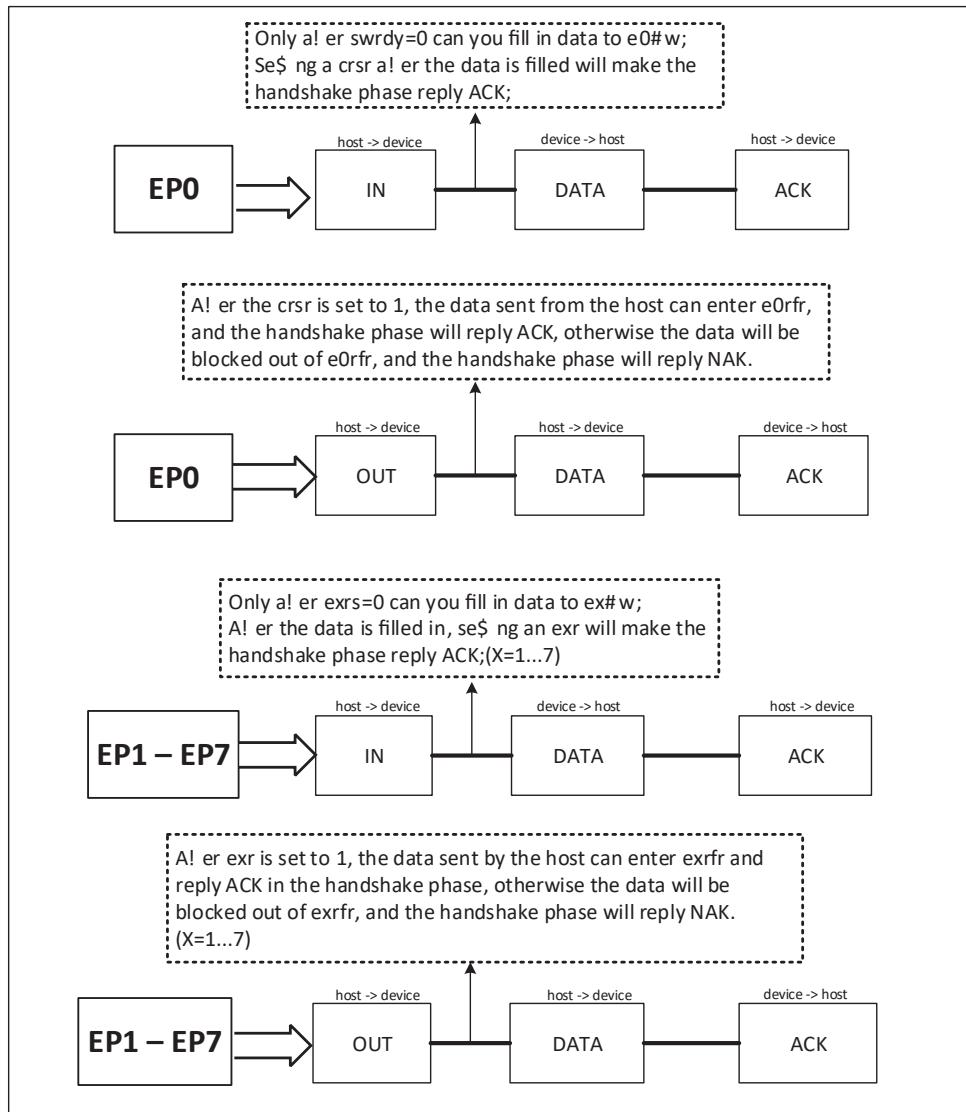


图 18.2: USB 通信方式

内部 transceiver 寄存器推荐配置:

表 18.1: 寄存器配置 1

usb_xcvr	value
usb_rcv	read only
usb_vip	read only
usb_vim	read only

**表 18.1: 寄存器配置 1(continued)**

usb_xcvr	value
usb_bd	read only
pu_usb	0->1
usb_sus	0
usb_spd	1
usb_enum	0->1
usb_data_convert	0
usb_oeb	read only
usb_oeb_reg	1
usb_oeb_sel	0
usb_rout_pmos	3
usb_rout_nmos	3
pu_usb_ldo	0
usb_ldo_vfb	3

**表 18.2: 寄存器配置 2**

usb_xcvr_config	value
usb_slewrate_p_rise	4
usb_slewrate_p_fall	3
usb_slewrate_m_rise	4
usb_slewrate_m_fall	3
usb_res_pullup_tune	2
reg_usb_use_ctrl	0
usb_str_drv	1
reg_usb_use_xcvr	1
usb_bd_vth	7
usb_v_hys_p	1
usb_v_hys_m	1

注意：当准备开启内部 transceiver 时需要将 pu\_usb 和 usb\_enum 置一。

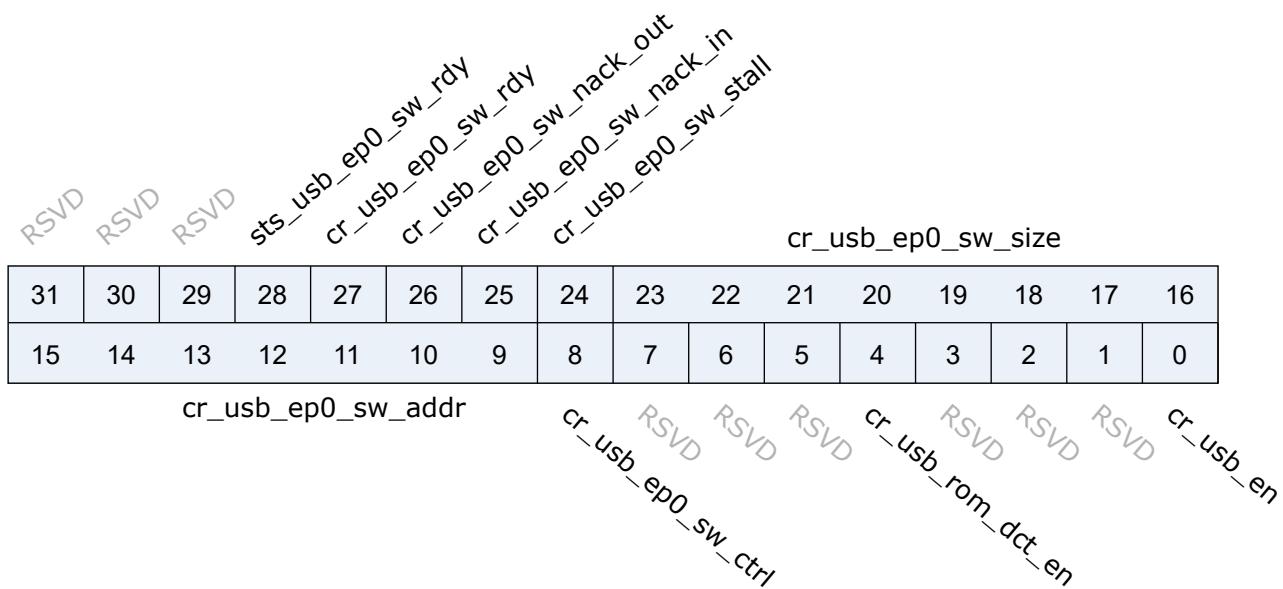
## 18.4 寄存器描述

名称	描述
usb_config	USB configuration
usb_lpm_config	USB lpm configuration
usb_resume_config	USB resume configuration
usb_frame_no	USB frame number
usb_error	USB error
usb_int_en	USB interrupt enable
usb_int_sts	USB interrupt status
usb_int_mask	USB interrupt mask
usb_int_clear	USB interrupt clear
ep1_config	EP1 configuration
ep2_config	EP2 configuration
ep3_config	EP3 configuration
ep4_config	EP4 configuration
ep5_config	EP5 configuration
ep6_config	EP6 configuration
ep7_config	EP7 configuration
ep0_fifo_config	EP0 fifo configuration
ep0_fifo_status	EP0 fifo status
ep0_tx_fifo_wdata	EP0 tx fifo write data
ep0_rx_fifo_rdata	EP0 rx fifo write data
ep1_fifo_config	EP1 fifo configuration
ep1_fifo_status	EP1 fifo status
ep1_tx_fifo_wdata	EP1 tx fifo write data
ep1_rx_fifo_rdata	EP1 rx fifo write data
ep2_fifo_config	EP2 fifo configuration
ep2_fifo_status	EP2 fifo status
ep2_tx_fifo_wdata	EP2 tx fifo write data

名称	描述
ep2_rx_fifo_rdata	EP2 rx fifo write data
ep3_fifo_config	EP3 fifo configuration
ep3_fifo_status	EP3 fifo status
ep3_tx_fifo_wdata	EP3 tx fifo write data
ep3_rx_fifo_rdata	EP3 rx fifo write data
ep4_fifo_config	EP4 fifo configuration
ep4_fifo_status	EP4 fifo status
ep4_tx_fifo_wdata	EP4 tx fifo write data
ep4_rx_fifo_rdata	EP4 rx fifo write data
ep5_fifo_config	EP5 fifo configuration
ep5_fifo_status	EP5 fifo status
ep5_tx_fifo_wdata	EP5 tx fifo write data
ep5_rx_fifo_rdata	EP5 rx fifo read data
ep6_fifo_config	EP6 fifo configuration
ep6_fifo_status	EP6 fifo status
ep6_tx_fifo_wdata	EP6 tx fifo write data
ep6_rx_fifo_rdata	EP6 rx fifo read data
ep7_fifo_config	EP7 fifo configuration
ep7_fifo_status	EP7 fifo status
ep7_tx_fifo_wdata	EP7 tx fifo write data
ep7_rx_fifo_rdata	EP7 rx fifo read data
xcvr_if_config	

### 18.4.1 usb\_config

地址: 0x4000d800

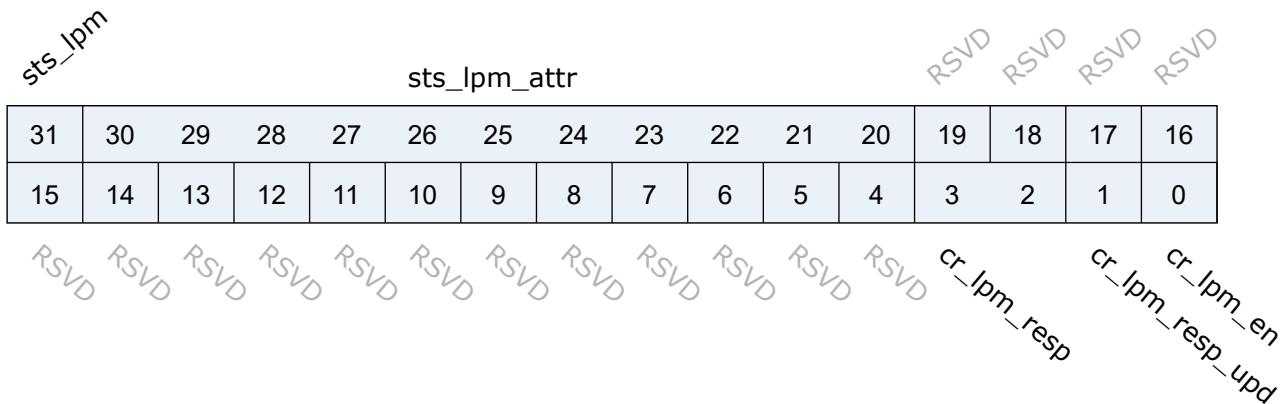


位	名称	权限	复位值	描述
31:29	RSVD			
28	sts_usb_ep0_sw_rdy	r	1'b0	EP0 transaction ready status bit. Asserted with sw_rdy, and de-asserted when ACK is sent/received.
27	cr_usb_ep0_sw_rdy	w1c	1'b0	EP0 transaction ready. When NACK is enabled, asserting this bit will allow one packet to be transferred even if NACK is asserted
26	cr_usb_ep0_sw_nack_out	r/w	1'b0	EP0 OUT/SETUP transaction nack response (SW control mode) Note: Should NOT enable both ep0_sw_nack_out and ep0_sw_stall at the same time
25	cr_usb_ep0_sw_nack_in	r/w	1'b1	EP0 IN transaction nack response (SW control mode) Note: Should NOT enable both ep0_sw_nack_in and ep0_sw_stall at the same time
24	cr_usb_ep0_sw_stall	w1c	1'b0	EP0 stall response (SW control mode) Note: Should NOT enable both ep0_sw_nack_in/out and ep0_sw_stall at the same time
23:16	cr_usb_ep0_sw_size	r/w	8'd0	EP0 transfer size (SW control mode)
15:9	cr_usb_ep0_sw_addr	r/w	7'd0	EP0 address (SW control mode)
8	cr_usb_ep0_sw_ctrl	r/w	1'b0	EP0 software control enable 1'b1: EP0 IN/OUT transaction is fully controlled by SW 1'b0: EP0 IN/OUT transaction is controlled by HW

位	名称	权限	复位值	描述
7:5	RSVD			
4	cr_usb_rom_dct_en	r/w	1'b1	Enable signal of ROM-based descriptors (don't care if ep0_sw_ctrl is asserted) 1'b1: USB descriptors stored in ROM will be used 1'b0: SW should prepare the descriptors requested by HOST
3:1	RSVD			
0	cr_usb_en	r/w	1'b0	Enable signal of USB function

#### 18.4.2 usb\_lpm\_config

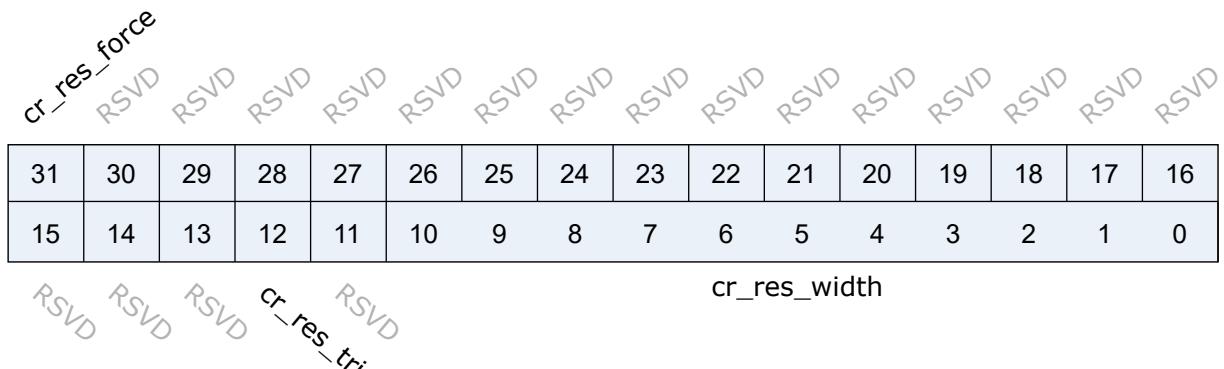
地址: 0x4000d804



位	名称	权限	复位值	描述
31	sts_lpm	r	1'b0	LPM status bit
30:20	sts_lpm_attr	r	11'h0	LPM attributes received in LPM packet
19:4	RSVD			
3:2	cr_lpm_resp	r/w	2'd2	Response when LPM packet is received 2'd3: NYET 2'd2: STALL 2'd1: NACK 2'd0: ACK
1	cr_lpm_resp_upd	w1c	1'b0	Response update signal (for async concern) Assert this bit when cr_lpm_resp is updated
0	cr_lpm_en	w1c	1'b0	LPM enable signal

### 18.4.3 usb\_resume\_config

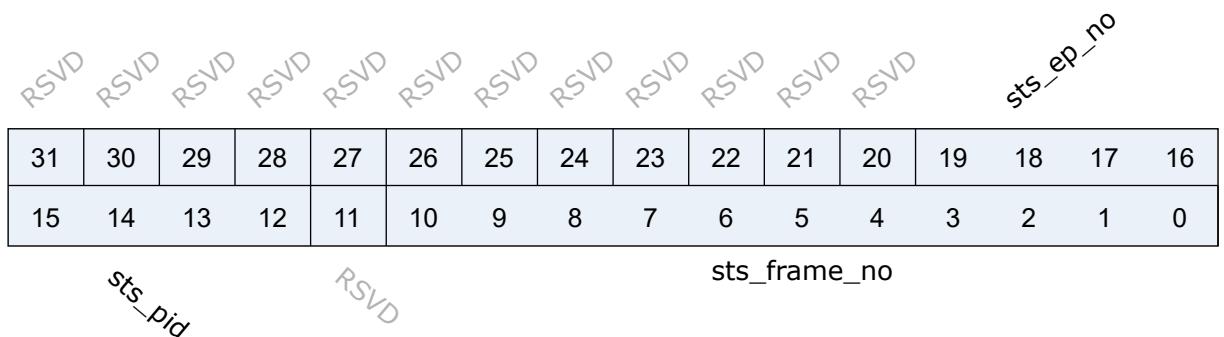
地址: 0x4000d808



位	名称	权限	复位值	描述
31	cr_res_force	r/w	1'b0	Force to output K-state
30:13	RSVD			
12	cr_res_trig	w1c	1'b0	Resume K-state trigger
11	RSVD			
10:0	cr_res_width	r/w	11'd26	Resume K-state width (unit: 2.67us)

### 18.4.4 usb\_frame\_no

地址: 0x4000d818



位	名称	权限	复位值	描述
31:20	RSVD			
19:16	sts_ep_no	r	4'd0	Endpoint number of the current transaction
15:12	sts_pid	r	4'd0	PID value of the current transaction
11	RSVD			

位	名称	权限	复位值	描述
10:0	sts_frame_no	r	11'h0	Current frame number

### 18.4.5 usb\_error

地址: 0x4000d81c

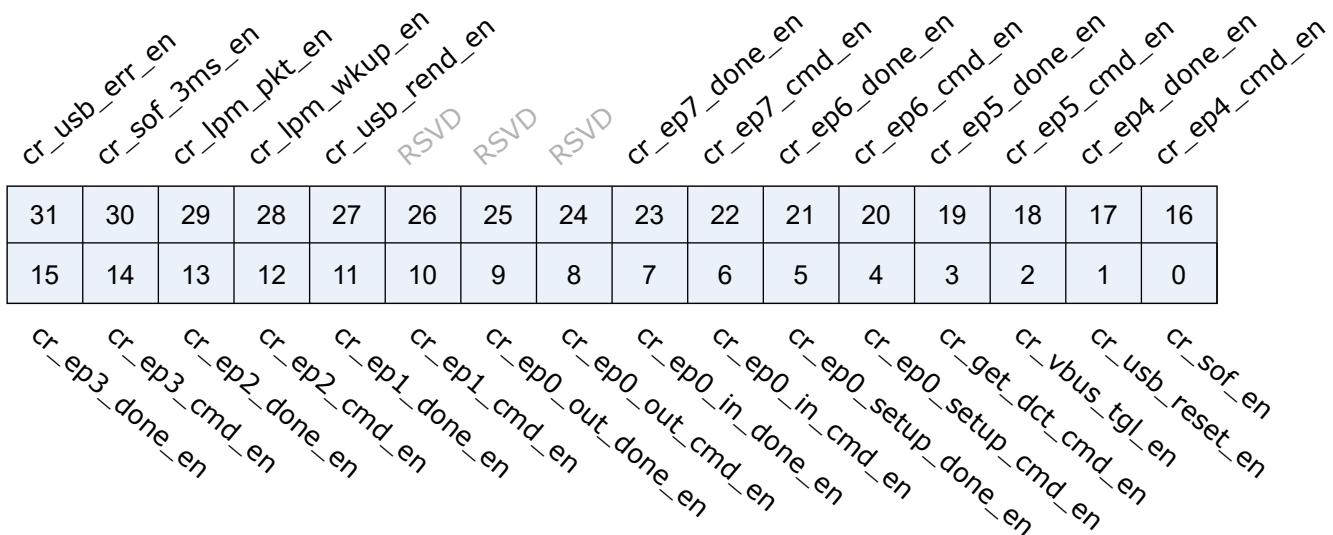
RSVD																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

RSVD RSVD

位	名称	权限	复位值	描述
31:7	RSVD			
6	crc16_err	r	1'b0	Data CRC error occurs, cleared by cr_usb_err_clr
5	crc5_err	r	1'b0	Token CRC error occurs, cleared by cr_usb_err_clr
4	pid_cks_err	r	1'b0	PID check sum error occurs, cleared by cr_usb_err_clr
3	pid_seq_err	r	1'b0	PID sequence error occurs, cleared by cr_usb_err_clr
2	ivld_ep_err	r	1'b0	Invalid endpoint error occurs, cleared by cr_usb_err_clr
1	xfer_to_err	r	1'b0	Transfer time-out error occurs, cleared by cr_usb_err_clr
0	utmi_rx_err	r	1'b0	UTMI I/F RX error occurs, cleared by cr_usb_err_clr

### 18.4.6 usb\_int\_en

地址: 0x4000d820

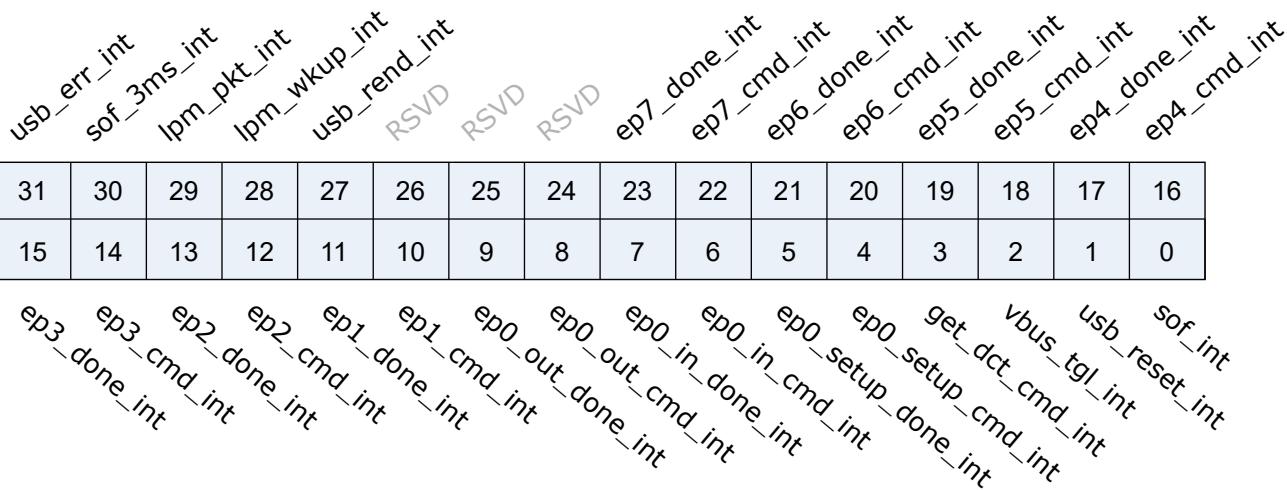


位	名称	权限	复位值	描述
31	cr_usb_err_en	r/w	1'b1	Interrupt enable of usb_err_int
30	cr_sof_3ms_en	r/w	1'b0	Interrupt enable of sof_3ms_int
29	cr_lpm_pkt_en	r/w	1'b0	Interrupt enable of lpm_pkt_int
28	cr_lpm_wkup_en	r/w	1'b0	Interrupt enable of lpm_wkup_int
27	cr_usb_rend_en	r/w	1'b0	Interrupt enable of usb_rend_int
26:24	RSVD			
23	cr_ep7_done_en	r/w	1'b1	Interrupt enable of ep7_done_int
22	cr_ep7_cmd_en	r/w	1'b1	Interrupt enable of ep7_cmd_int
21	cr_ep6_done_en	r/w	1'b1	Interrupt enable of ep6_done_int
20	cr_ep6_cmd_en	r/w	1'b1	Interrupt enable of ep6_cmd_int
19	cr_ep5_done_en	r/w	1'b1	Interrupt enable of ep5_done_int
18	cr_ep5_cmd_en	r/w	1'b1	Interrupt enable of ep5_cmd_int
17	cr_ep4_done_en	r/w	1'b1	Interrupt enable of ep4_done_int
16	cr_ep4_cmd_en	r/w	1'b1	Interrupt enable of ep4_cmd_int
15	cr_ep3_done_en	r/w	1'b1	Interrupt enable of ep3_done_int
14	cr_ep3_cmd_en	r/w	1'b1	Interrupt enable of ep3_cmd_int
13	cr_ep2_done_en	r/w	1'b1	Interrupt enable of ep2_done_int
12	cr_ep2_cmd_en	r/w	1'b1	Interrupt enable of ep2_cmd_int
11	cr_ep1_done_en	r/w	1'b1	Interrupt enable of ep1_done_int

位	名称	权限	复位值	描述
10	cr_ep1_cmd_en	r/w	1'b1	Interrupt enable of ep1_cmd_int
9	cr_ep0_out_done_en	r/w	1'b1	Interrupt enable of ep0_out_done_int
8	cr_ep0_out_cmd_en	r/w	1'b1	Interrupt enable of ep0_out_cmd_int
7	cr_ep0_in_done_en	r/w	1'b1	Interrupt enable of ep0_in_done_int
6	cr_ep0_in_cmd_en	r/w	1'b1	Interrupt enable of ep0_in_cmd_int
5	cr_ep0_setup_done_en	r/w	1'b1	Interrupt enable of ep0_setup_done_int
4	cr_ep0_setup_cmd_en	r/w	1'b1	Interrupt enable of ep0_setup_cmd_int
3	cr_get_dct_cmd_en	r/w	1'b1	Interrupt enable of get_dct_cmd_int
2	cr_vbus_tgl_en	r/w	1'b1	Interrupt enable of vbus_tgl_int
1	cr_usb_reset_en	r/w	1'b1	Interrupt enable of usb_reset_int
0	cr_sof_en	r/w	1'b1	Interrupt enable of sof_int

#### 18.4.7 usb\_int\_sts

地址: 0x4000d824

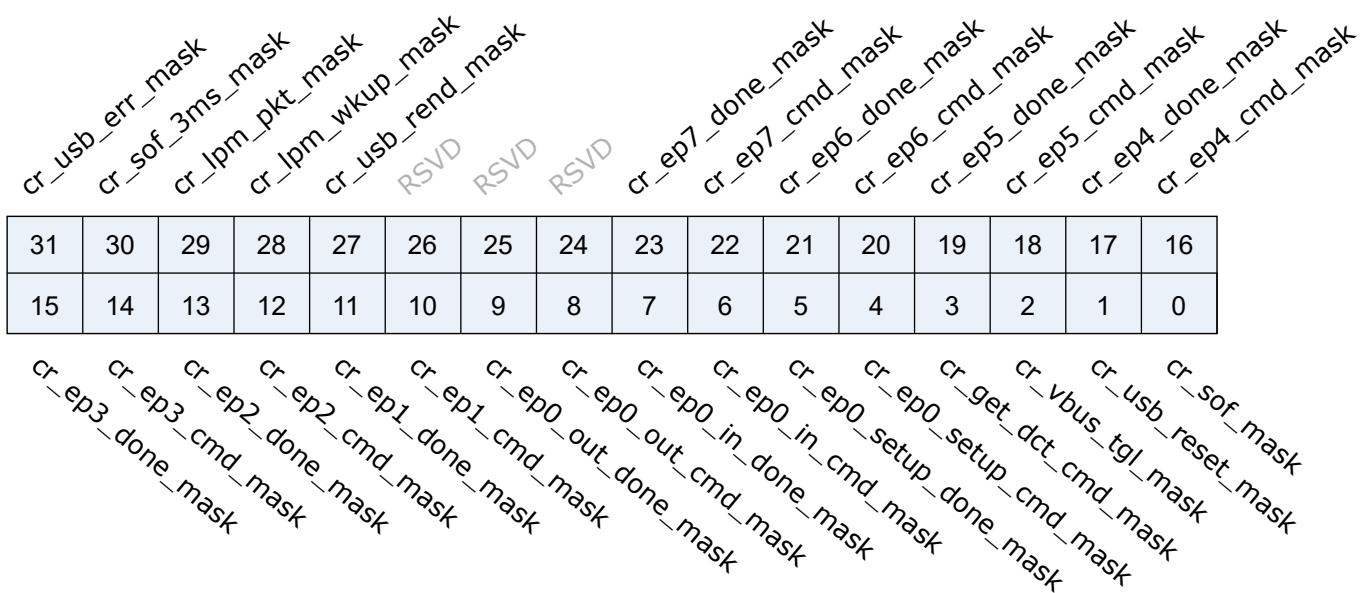


位	名称	权限	复位值	描述
31	usb_err_int	r	1'b0	USB error occurs, check usb_error for detailed error type
30	sof_3ms_int	r	1'b0	SOF is absent for 3 ms
29	lpm_pkt_int	r	1'b0	LPM packet is received
28	lpm_wkup_int	r	1'b0	LPM resume (wakeup) signal is received
27	usb_rend_int	r	1'b0	USB reset de-assert is triggered

位	名称	权限	复位值	描述
26:24	RSVD			
23	ep7_done_int	r	1'b0	EP7 IN or OUT command is finished
22	ep7_cmd_int	r	1'b0	EP7 IN or OUT command is received
21	ep6_done_int	r	1'b0	EP6 IN or OUT command is finished
20	ep6_cmd_int	r	1'b0	EP6 IN or OUT command is received
19	ep5_done_int	r	1'b0	EP5 IN or OUT command is finished
18	ep5_cmd_int	r	1'b0	EP5 IN or OUT command is received
17	ep4_done_int	r	1'b0	EP4 IN or OUT command is finished
16	ep4_cmd_int	r	1'b0	EP4 IN or OUT command is received
15	ep3_done_int	r	1'b0	EP3 IN or OUT command is finished
14	ep3_cmd_int	r	1'b0	EP3 IN or OUT command is received
13	ep2_done_int	r	1'b0	EP2 IN or OUT command is finished
12	ep2_cmd_int	r	1'b0	EP2 IN or OUT command is received
11	ep1_done_int	r	1'b0	EP1 IN or OUT command is finished
10	ep1_cmd_int	r	1'b0	EP1 IN or OUT command is received
9	ep0_out_done_int	r	1'b0	EP0 OUT command is finished
8	ep0_out_cmd_int	r	1'b0	EP0 OUT command is received
7	ep0_in_done_int	r	1'b0	EP0 IN command is finished
6	ep0_in_cmd_int	r	1'b0	EP0 IN command is received
5	ep0_setup_done_int	r	1'b0	EP0 SETUP command is finished
4	ep0_setup_cmd_int	r	1'b0	EP0 SETUP command is received
3	get_dct_cmd_int	r	1'b0	GET_DESCRIPTOR command is received
2	vbus_tgl_int	r	1'b0	VBUS detection is toggled, check 0x1FC[31] for vbus_detect status
1	usb_reset_int	r	1'b0	USB reset is triggered
0	sof_int	r	1'b0	SOF is received

### 18.4.8 usb\_int\_mask

地址: 0x4000d828

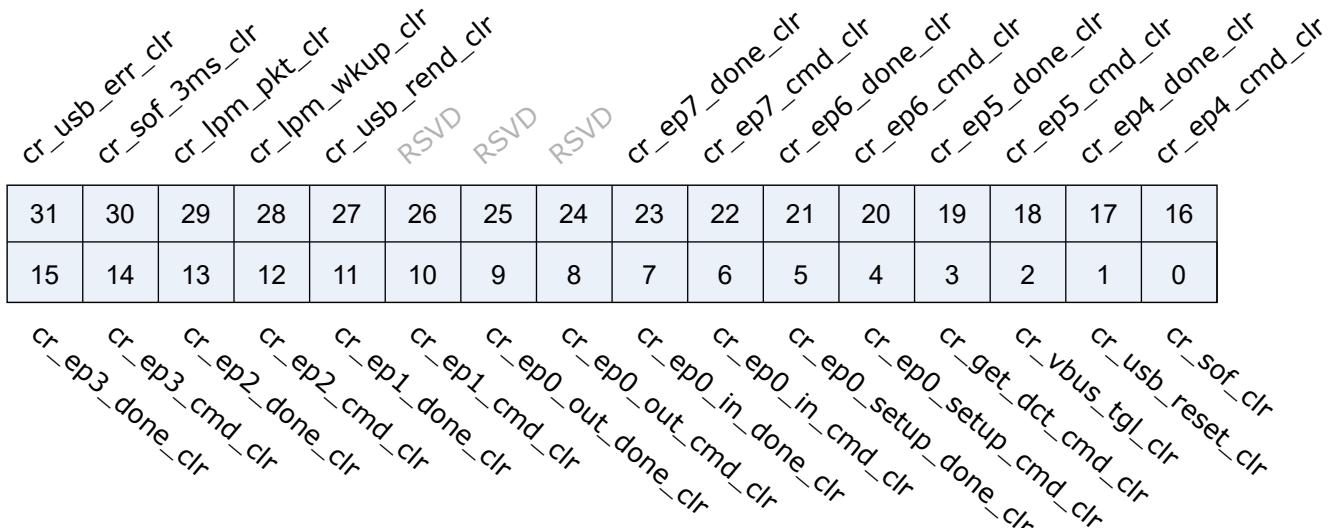


位	名称	权限	复位值	描述
31	cr_usb_err_mask	r/w	1'b1	Interrupt mask of usb_err_int
30	cr_sof_3ms_mask	r/w	1'b1	Interrupt mask of sof_3ms_int
29	cr_lpm_pkt_mask	r/w	1'b1	Interrupt mask of lpm_pkt_int
28	cr_lpm_wkup_mask	r/w	1'b1	Interrupt mask of lpm_wkup_int
27	cr_usb_rend_mask	r/w	1'b1	Interrupt mask of usb_rend_int
26:24	RSVD			
23	cr_ep7_done_mask	r/w	1'b1	Interrupt mask of ep7_done_int
22	cr_ep7_cmd_mask	r/w	1'b1	Interrupt mask of ep7_cmd_int
21	cr_ep6_done_mask	r/w	1'b1	Interrupt mask of ep6_done_int
20	cr_ep6_cmd_mask	r/w	1'b1	Interrupt mask of ep6_cmd_int
19	cr_ep5_done_mask	r/w	1'b1	Interrupt mask of ep5_done_int
18	cr_ep5_cmd_mask	r/w	1'b1	Interrupt mask of ep5_cmd_int
17	cr_ep4_done_mask	r/w	1'b1	Interrupt mask of ep4_done_int
16	cr_ep4_cmd_mask	r/w	1'b1	Interrupt mask of ep4_cmd_int
15	cr_ep3_done_mask	r/w	1'b1	Interrupt mask of ep3_done_int
14	cr_ep3_cmd_mask	r/w	1'b1	Interrupt mask of ep3_cmd_int
13	cr_ep2_done_mask	r/w	1'b1	Interrupt mask of ep2_done_int
12	cr_ep2_cmd_mask	r/w	1'b1	Interrupt mask of ep2_cmd_int

位	名称	权限	复位值	描述
11	cr_ep1_done_mask	r/w	1'b1	Interrupt mask of ep1_done_int
10	cr_ep1_cmd_mask	r/w	1'b1	Interrupt mask of ep1_cmd_int
9	cr_ep0_out_done_mask	r/w	1'b1	Interrupt mask of ep0_out_done_int
8	cr_ep0_out_cmd_mask	r/w	1'b1	Interrupt mask of ep0_out_cmd_int
7	cr_ep0_in_done_mask	r/w	1'b1	Interrupt mask of ep0_in_done_int
6	cr_ep0_in_cmd_mask	r/w	1'b1	Interrupt mask of ep0_in_cmd_int
5	cr_ep0_setup_done_-mask	r/w	1'b1	Interrupt mask of ep0_setup_done_int
4	cr_ep0_setup_cmd_-mask	r/w	1'b1	Interrupt mask of ep0_setup_cmd_int
3	cr_get_dct_cmd_mask	r/w	1'b1	Interrupt mask of get_dct_cmd_int
2	cr_vbus_tgl_mask	r/w	1'b1	Interrupt mask of vbus_tgl_int
1	cr_usb_reset_mask	r/w	1'b1	Interrupt mask of usb_reset_int
0	cr_sof_mask	r/w	1'b1	Interrupt mask of sof_int

#### 18.4.9 usb\_int\_clear

地址: 0x4000d82c

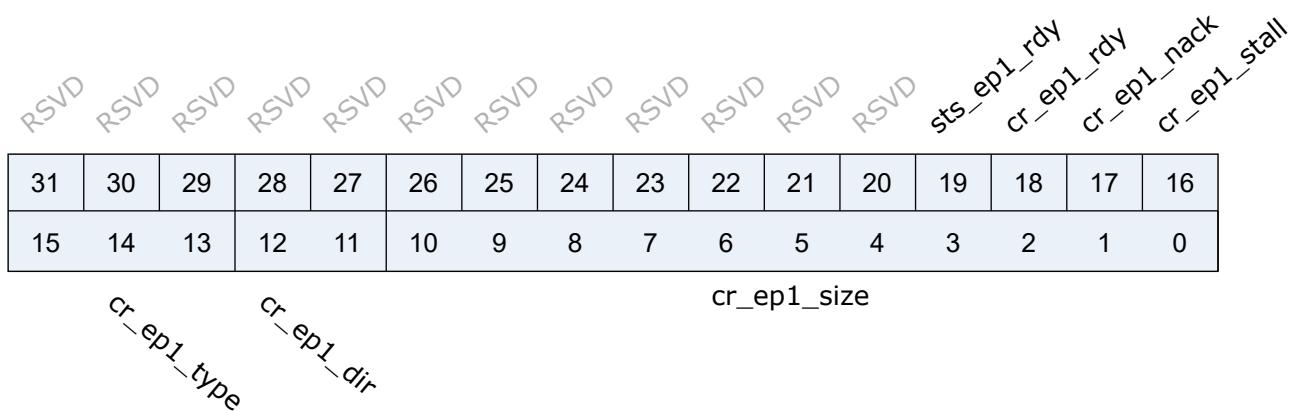


位	名称	权限	复位值	描述
31	cr_usb_err_clr	w1c	1'b0	Interrupt clear of usb_err_int
30	cr_sof_3ms_clr	w1c	1'b0	Interrupt clear of sof_3ms_int

位	名称	权限	复位值	描述
29	cr_lpm_pkt_clr	w1c	1'b0	Interrupt clear of lpm_pkt_int
28	cr_lpm_wkup_clr	w1c	1'b0	Interrupt clear of lpm_wkup_int
27	cr_usb_rend_clr	w1c	1'b0	Interrupt clear of usb_rend_int
26:24	RSVD			
23	cr_ep7_done_clr	w1c	1'b0	Interrupt clear of ep7_done_int
22	cr_ep7_cmd_clr	w1c	1'b0	Interrupt clear of ep7_cmd_int
21	cr_ep6_done_clr	w1c	1'b0	Interrupt clear of ep6_done_int
20	cr_ep6_cmd_clr	w1c	1'b0	Interrupt clear of ep6_cmd_int
19	cr_ep5_done_clr	w1c	1'b0	Interrupt clear of ep5_done_int
18	cr_ep5_cmd_clr	w1c	1'b0	Interrupt clear of ep5_cmd_int
17	cr_ep4_done_clr	w1c	1'b0	Interrupt clear of ep4_done_int
16	cr_ep4_cmd_clr	w1c	1'b0	Interrupt clear of ep4_cmd_int
15	cr_ep3_done_clr	w1c	1'b0	Interrupt clear of ep3_done_int
14	cr_ep3_cmd_clr	w1c	1'b0	Interrupt clear of ep3_cmd_int
13	cr_ep2_done_clr	w1c	1'b0	Interrupt clear of ep2_done_int
12	cr_ep2_cmd_clr	w1c	1'b0	Interrupt clear of ep2_cmd_int
11	cr_ep1_done_clr	w1c	1'b0	Interrupt clear of ep1_done_int
10	cr_ep1_cmd_clr	w1c	1'b0	Interrupt clear of ep1_cmd_int
9	cr_ep0_out_done_clr	w1c	1'b0	Interrupt clear of ep0_out_done_int
8	cr_ep0_out_cmd_clr	w1c	1'b0	Interrupt clear of ep0_out_cmd_int
7	cr_ep0_in_done_clr	w1c	1'b0	Interrupt clear of ep0_in_done_int
6	cr_ep0_in_cmd_clr	w1c	1'b0	Interrupt clear of ep0_in_cmd_int
5	cr_ep0_setup_done_clr	w1c	1'b0	Interrupt clear of ep0_setup_done_int
4	cr_ep0_setup_cmd_clr	w1c	1'b0	Interrupt clear of ep0_setup_cmd_int
3	cr_get_dct_cmd_clr	w1c	1'b0	Interrupt clear of get_dct_cmd_int
2	cr_vbus_tgl_clr	w1c	1'b0	Interrupt clear of vbus_tgl_int
1	cr_usb_reset_clr	w1c	1'b0	Interrupt clear of usb_reset_int
0	cr_sof_clr	w1c	1'b0	Interrupt clear of sof_int

### 18.4.10 ep1\_config

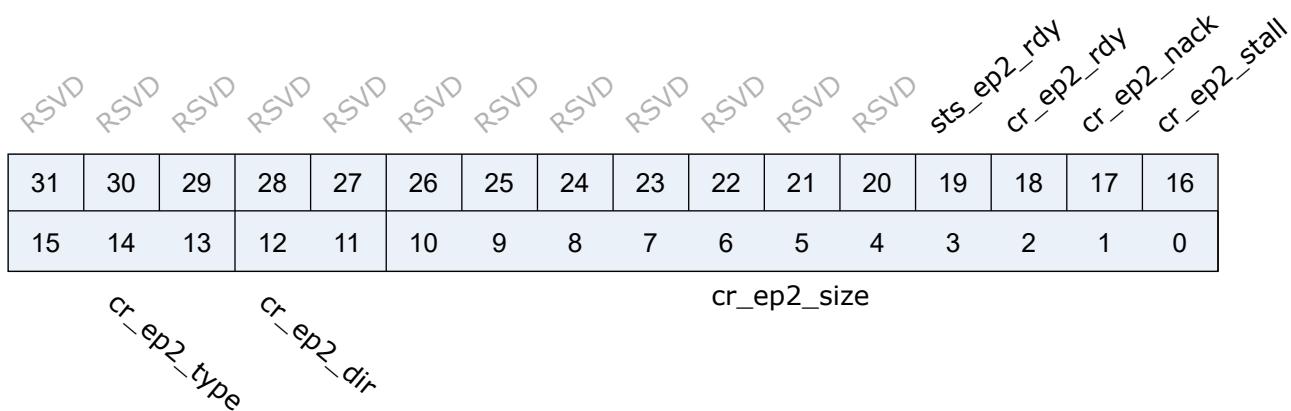
地址: 0x4000d840



位	名称	权限	复位值	描述
31:20	RSVD			
19	sts_ep1_rdy	r	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and deasserted when ACK is sent/received.
18	cr_ep1_rdy	w1c	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	cr_ep1_nack	r/w	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time
16	cr_ep1_stall	r/w	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time
15:13	cr_ep1_type	r/w	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	cr_ep1_dir	r/w	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	cr_ep1_size	r/w	11'd64	Endpoint max packet size

### 18.4.11 ep2\_config

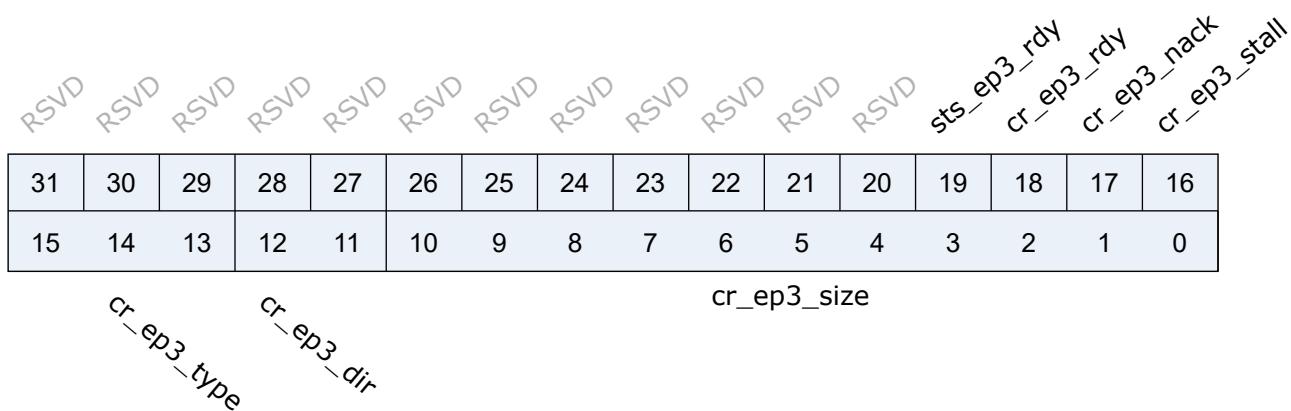
地址: 0x4000d844



位	名称	权限	复位值	描述
31:20	RSVD			
19	sts_ep2_rdy	r	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and deasserted when ACK is sent/received.
18	cr_ep2_rdy	w1c	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	cr_ep2_nack	r/w	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time
16	cr_ep2_stall	r/w	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time
15:13	cr_ep2_type	r/w	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	cr_ep2_dir	r/w	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	cr_ep2_size	r/w	11'd64	Endpoint max packet size

### 18.4.12 ep3\_config

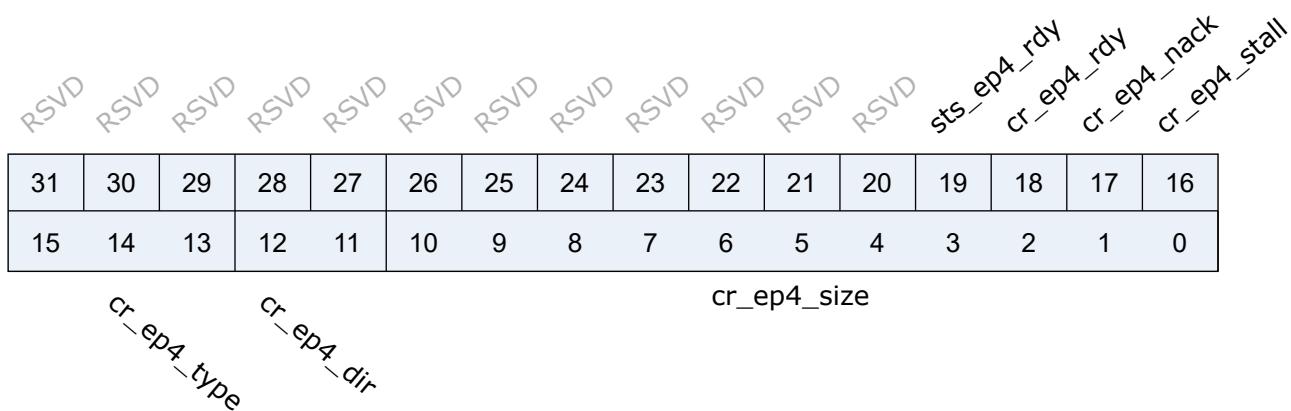
地址: 0x4000d848



位	名称	权限	复位值	描述
31:20	RSVD			
19	sts_ep3_rdy	r	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and deasserted when ACK is sent/received.
18	cr_ep3_rdy	w1c	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	cr_ep3_nack	r/w	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time
16	cr_ep3_stall	r/w	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time
15:13	cr_ep3_type	r/w	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	cr_ep3_dir	r/w	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	cr_ep3_size	r/w	11'd64	Endpoint max packet size

### 18.4.13 ep4\_config

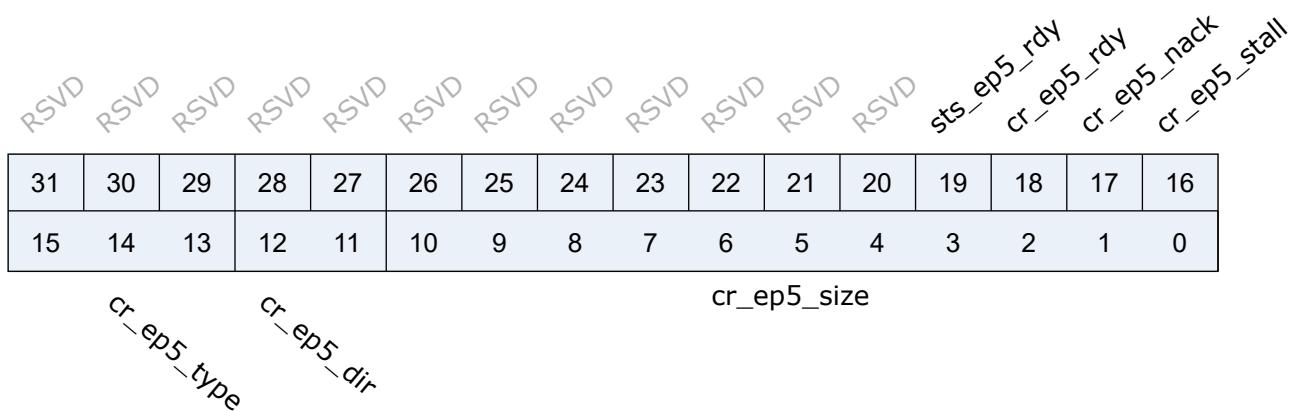
地址: 0x4000d84c



位	名称	权限	复位值	描述
31:20	RSVD			
19	sts_ep4_rdy	r	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and deasserted when ACK is sent/received.
18	cr_ep4_rdy	w1c	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	cr_ep4_nack	r/w	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time
16	cr_ep4_stall	r/w	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time
15:13	cr_ep4_type	r/w	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	cr_ep4_dir	r/w	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	cr_ep4_size	r/w	11'd64	Endpoint max packet size

### 18.4.14 ep5\_config

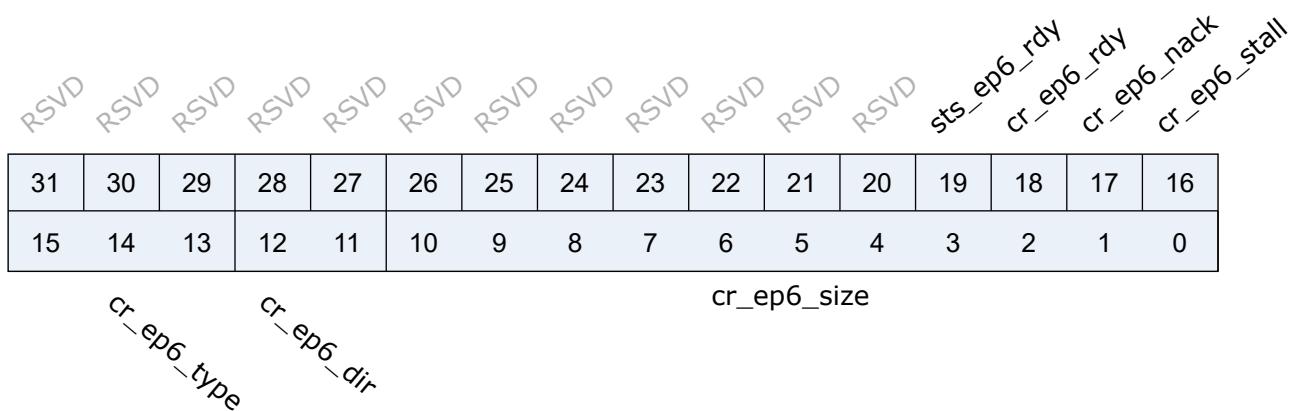
地址: 0x4000d850



位	名称	权限	复位值	描述
31:20	RSVD			
19	sts_ep5_rdy	r	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and deasserted when ACK is sent/received.
18	cr_ep5_rdy	w1c	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	cr_ep5_nack	r/w	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time
16	cr_ep5_stall	r/w	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time
15:13	cr_ep5_type	r/w	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	cr_ep5_dir	r/w	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	cr_ep5_size	r/w	11'd64	Endpoint max packet size

### 18.4.15 ep6\_config

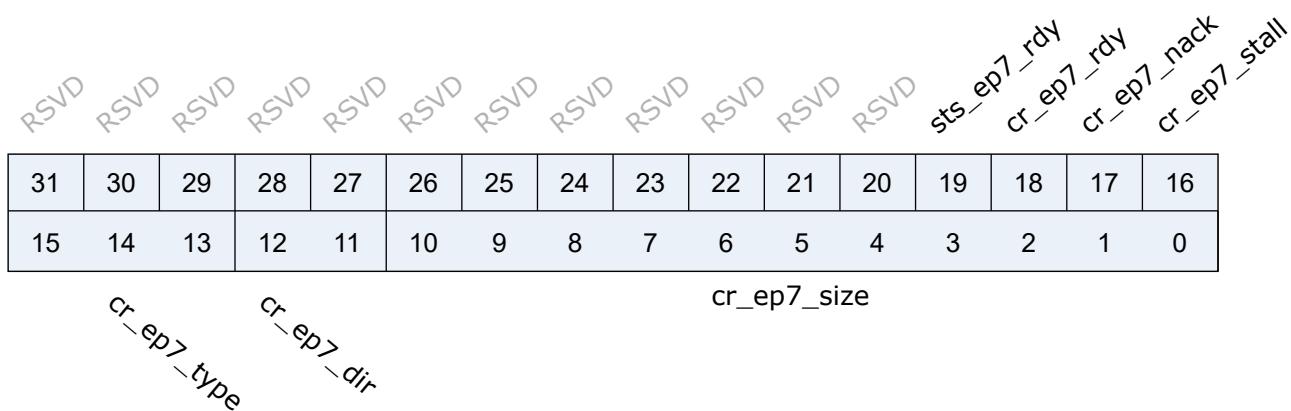
地址: 0x4000d854



位	名称	权限	复位值	描述
31:20	RSVD			
19	sts_ep6_rdy	r	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and de-asserted when ACK is sent/received.
18	cr_ep6_rdy	w1c	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	cr_ep6_nack	r/w	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time
16	cr_ep6_stall	r/w	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time
15:13	cr_ep6_type	r/w	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	cr_ep6_dir	r/w	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	cr_ep6_size	r/w	11'd64	Endpoint max packet size

### 18.4.16 ep7\_config

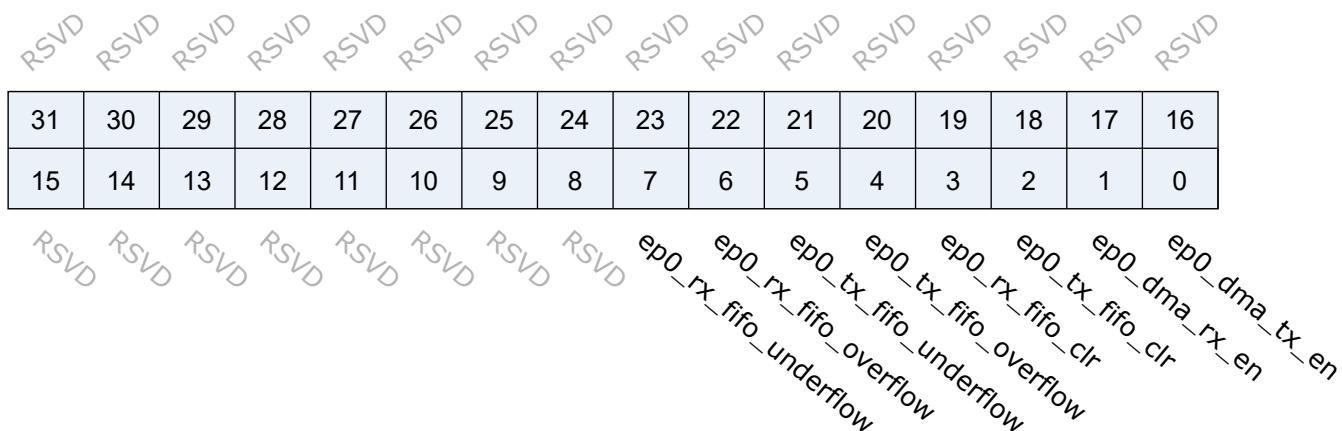
地址: 0x4000d858



位	名称	权限	复位值	描述
31:20	RSVD			
19	sts_ep7_rdy	r	1'b0	Endpoint ready status bit. Asserted with ep_rdy, and de-asserted when ACK is sent/received.
18	cr_ep7_rdy	w1c	1'b0	Endpoint ready. When Endpoint NACK is enabled, asserting this bit will allow one packet to be transferred
17	cr_ep7_nack	r/w	1'b1	Endpoint NACK response enable, should not be enabled with STALL at the same time
16	cr_ep7_stall	r/w	1'b0	Endpoint STALL response enable, should not be enabled with NACK at the same time
15:13	cr_ep7_type	r/w	3'b100	Endpoint type 3'b101: CTRL 3'b010: ISO 3'b100: BULK 3'b000: INT Others: Reserved
12:11	cr_ep7_dir	r/w	2'b01	Endpoint direction 2'b00: Disabled 2'b01: IN 2'b10: OUT 2'b11: Reserved
10:0	cr_ep7_size	r/w	11'd64	Endpoint max packet size

### 18.4.17 ep0\_fifo\_config

地址: 0x4000d900



位	名称	权限	复位值	描述
31:8	RSVD			
7	ep0_rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	ep0_rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	ep0_tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	ep0_tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	ep0_rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	ep0_tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	ep0_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	ep0_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

### 18.4.18 ep0\_fifo\_status

地址: 0x4000d904

ep0_rx_fifo_full															
ep0_rx_fifo_empty															
RSVD															
ep0_rx_fifo_cnt															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ep0_tx_fifo_full															
ep0_tx_fifo_empty															
RSVD															
ep0_tx_fifo_cnt															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

位	名称	权限	复位值	描述
31	ep0_rx_fifo_full	r	1'b0	RX FIFO full flag
30	ep0_rx_fifo_empty	r	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	ep0_rx_fifo_cnt	r	7'd0	RX FIFO available count
15	ep0_tx_fifo_full	r	1'b0	TX FIFO full flag
14	ep0_tx_fifo_empty	r	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	ep0_tx_fifo_cnt	r	7'd64	TX FIFO available count

#### 18.4.19 ep0\_tx\_fifo\_wdata

地址: 0x40000d908

RSVD															
ep0_tx_fifo_wdata															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep0_tx_fifo_wdata	w	x	

### 18.4.20 ep0\_rx\_fifo\_rdata

地址: 0x4000d90c

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

ep0\_rx\_fifo\_rdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep0_rx_fifo_rdata	r	8'h0	

### 18.4.21 ep1\_fifo\_config

地址: 0x4000d910

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

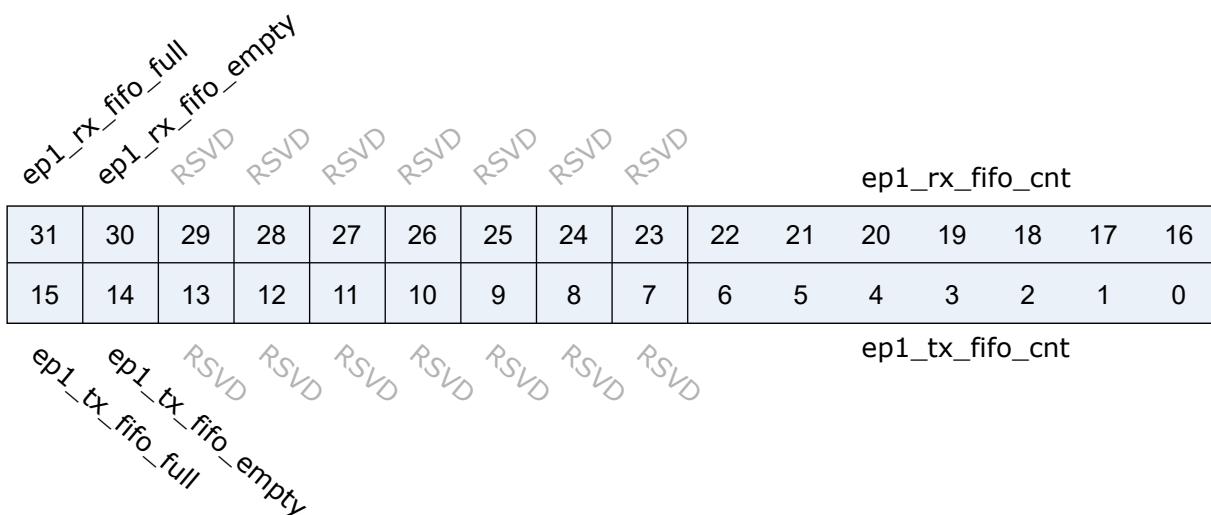
ep1\_rx\_fifo\_underflow    ep1\_rx\_fifo\_overflow    ep1\_tx\_fifo\_underflow    ep1\_tx\_fifo\_overflow    ep1\_rx\_fifo\_clr    ep1\_tx\_fifo\_clr    ep1\_dma\_rx\_en    ep1\_dma\_tx\_en

位	名称	权限	复位值	描述
31:8	RSVD			
7	ep1_rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	ep1_rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	ep1_tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr

位	名称	权限	复位值	描述
4	ep1_tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	ep1_rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	ep1_tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	ep1_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	ep1_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

#### 18.4.22 ep1\_fifo\_status

地址: 0x4000d914



位	名称	权限	复位值	描述
31	ep1_rx_fifo_full	r	1'b0	RX FIFO full flag
30	ep1_rx_fifo_empty	r	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	ep1_rx_fifo_cnt	r	7'd0	RX FIFO available count
15	ep1_tx_fifo_full	r	1'b0	TX FIFO full flag
14	ep1_tx_fifo_empty	r	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	ep1_tx_fifo_cnt	r	7'd64	TX FIFO available count

### 18.4.23 ep1\_tx\_fifo\_wdata

地址: 0x4000d918

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

ep1\_tx\_fifo\_wdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep1_tx_fifo_wdata	w	x	

### 18.4.24 ep1\_rx\_fifo\_rdata

地址: 0x4000d91c

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

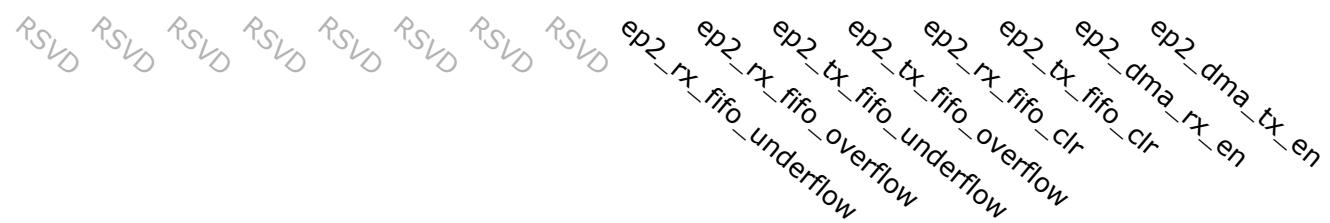
ep1\_rx\_fifo\_rdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep1_rx_fifo_rdata	r	8'h0	

### 18.4.25 ep2\_fifo\_config

地址: 0x4000d920

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



位	名称	权限	复位值	描述
31:8	RSVD			
7	ep2_rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	ep2_rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	ep2_tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	ep2_tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	ep2_rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	ep2_tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	ep2_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	ep2_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

#### 18.4.26 ep2\_fifo\_status

地址: 0x4000d924

ep2_rx_fifo_full	ep2_rx_fifo_empty	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	ep2_rx_fifo_cnt							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



位	名称	权限	复位值	描述
31	ep2_rx_fifo_full	r	1'b0	RX FIFO full flag
30	ep2_rx_fifo_empty	r	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	ep2_rx_fifo_cnt	r	7'd0	RX FIFO available count
15	ep2_tx_fifo_full	r	1'b0	TX FIFO full flag
14	ep2_tx_fifo_empty	r	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	ep2_tx_fifo_cnt	r	7'd64	TX FIFO available count

#### 18.4.27 ep2\_tx\_fifo\_wdata

地址: 0x4000d928

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

ep2\_tx\_fifo\_wdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep2_tx_fifo_wdata	w	x	

#### 18.4.28 ep2\_rx\_fifo\_rdata

地址: 0x4000d92c

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

ep2\_rx\_fifo\_rdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep2_rx_fifo_rdata	r	8'h0	

#### 18.4.29 ep3\_fifo\_config

地址: 0x4000d930

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

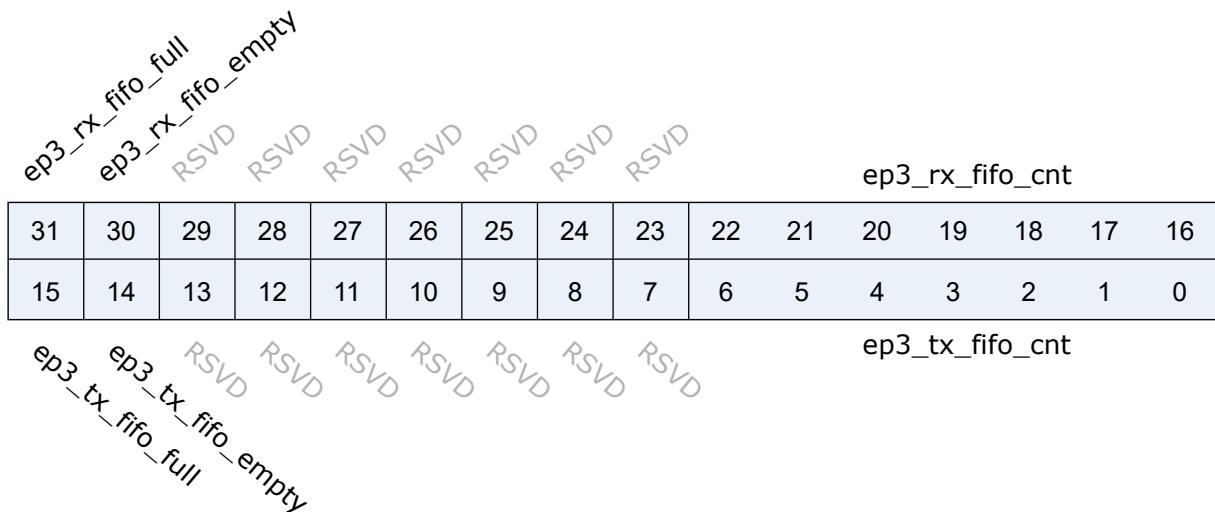
RSVD RSVD

ep3\_rx\_fifo\_underflow ep3\_rx\_fifo\_overflow ep3\_tx\_fifo\_underflow ep3\_tx\_fifo\_overflow ep3\_rx\_fifo\_clr ep3\_tx\_fifo\_clr ep3\_dma\_rx\_en ep3\_dma\_tx\_en

位	名称	权限	复位值	描述
31:8	RSVD			
7	ep3_rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	ep3_rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	ep3_tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	ep3_tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	ep3_rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	ep3_tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	ep3_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	ep3_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

### 18.4.30 ep3\_fifo\_status

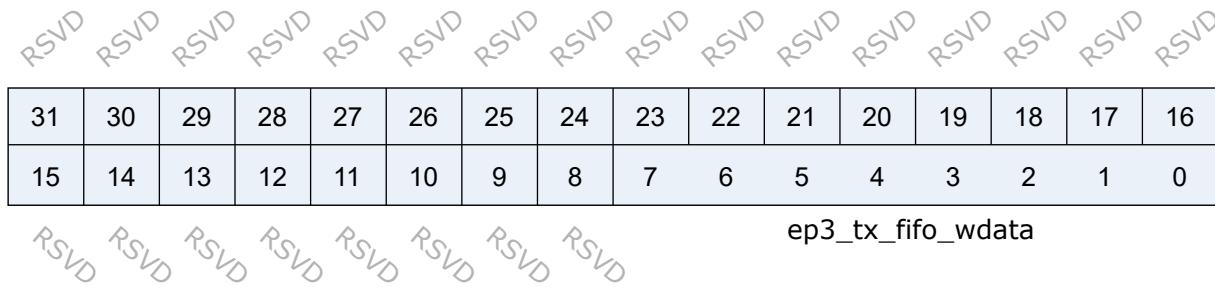
地址: 0x4000d934



位	名称	权限	复位值	描述
31	ep3_rx_fifo_full	r	1'b0	RX FIFO full flag
30	ep3_rx_fifo_empty	r	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	ep3_rx_fifo_cnt	r	7'd0	RX FIFO available count
15	ep3_tx_fifo_full	r	1'b0	TX FIFO full flag
14	ep3_tx_fifo_empty	r	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	ep3_tx_fifo_cnt	r	7'd64	TX FIFO available count

### 18.4.31 ep3\_tx\_fifo\_wdata

地址: 0x4000d938



位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep3_tx_fifo_wdata	w	x	

#### 18.4.32 ep3\_rx\_fifo\_rdata

地址: 0x4000d93c

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

ep3\_rx\_fifo\_rdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep3_rx_fifo_rdata	r	8'h0	

#### 18.4.33 ep4\_fifo\_config

地址: 0x4000d940

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

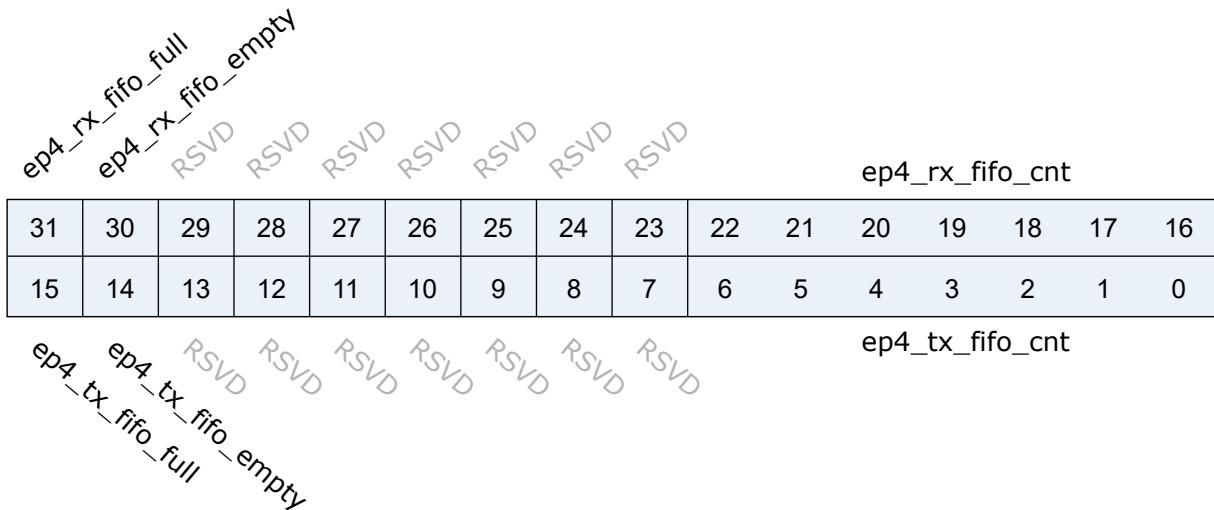
ep4\_rx\_fifo\_underflow  
ep4\_tx\_fifo\_overflow  
ep4\_tx\_fifo\_clr  
ep4\_rx\_fifo\_clr  
ep4\_dma\_tx\_en  
ep4\_dma\_rx\_en

位	名称	权限	复位值	描述
31:8	RSVD			
7	ep4_rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr

位	名称	权限	复位值	描述
6	ep4_rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	ep4_tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	ep4_tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	ep4_rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	ep4_tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	ep4_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	ep4_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

#### 18.4.34 ep4\_fifo\_status

地址: 0x4000d944



位	名称	权限	复位值	描述
31	ep4_rx_fifo_full	r	1'b0	RX FIFO full flag
30	ep4_rx_fifo_empty	r	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	ep4_rx_fifo_cnt	r	7'd0	RX FIFO available count
15	ep4_tx_fifo_full	r	1'b0	TX FIFO full flag
14	ep4_tx_fifo_empty	r	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	ep4_tx_fifo_cnt	r	7'd64	TX FIFO available count

### 18.4.35 ep4\_tx\_fifo\_wdata

地址: 0x4000d948

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

ep4\_tx\_fifo\_wdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep4_tx_fifo_wdata	w	x	

### 18.4.36 ep4\_rx\_fifo\_rdata

地址: 0x4000d94c

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

ep4\_rx\_fifo\_rdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep4_rx_fifo_rdata	r	8'h0	

### 18.4.37 ep5\_fifo\_config

地址: 0x4000d950

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

RSVD RSVD RSVD RSVD RSVD RSVD RSVD ep5\_rx\_fifo\_overflow ep5\_tx\_fifo\_overflow ep5\_rx\_fifo\_clr ep5\_tx\_fifo\_clr ep5\_dma\_rx\_en ep5\_dma\_tx\_en  
 RSVD RSVD RSVD RSVD RSVD RSVD ep5\_rx\_fifo\_underflow ep5\_tx\_fifo\_underflow ep5\_rx\_fifo\_clr ep5\_tx\_fifo\_clr ep5\_dma\_rx\_en ep5\_dma\_tx\_en

位	名称	权限	复位值	描述
31:8	RSVD			
7	ep5_rx_fifo_overflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	ep5_rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	ep5_tx_fifo_overflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	ep5_tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	ep5_rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	ep5_tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	ep5_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	ep5_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

#### 18.4.38 ep5\_fifo\_status

地址: 0x4000d954

ep5_rx_fifo_full	ep5_rx_fifo_empty	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	ep5_rx_fifo_cnt
31	30	29	28	27	26	25	24	23
15	14	13	12	11	10	9	8	7

ep5\_tx\_fifo\_full RSVD RSVD RSVD RSVD RSVD RSVD RSVD ep5\_tx\_fifo\_empty  
 ep5\_tx\_fifo\_full RSVD RSVD RSVD RSVD RSVD RSVD RSVD ep5\_tx\_fifo\_empty

位	名称	权限	复位值	描述
31	ep5_rx_fifo_full	r	1'b0	RX FIFO full flag
30	ep5_rx_fifo_empty	r	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	ep5_rx_fifo_cnt	r	7'd0	RX FIFO available count
15	ep5_tx_fifo_full	r	1'b0	TX FIFO full flag
14	ep5_tx_fifo_empty	r	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	ep5_tx_fifo_cnt	r	7'd64	TX FIFO available count

#### 18.4.39 ep5\_tx\_fifo\_wdata

地址: 0x4000d958

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

ep5\_tx\_fifo\_wdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep5_tx_fifo_wdata	w	x	

#### 18.4.40 ep5\_rx\_fifo\_rdata

地址: 0x4000d95c

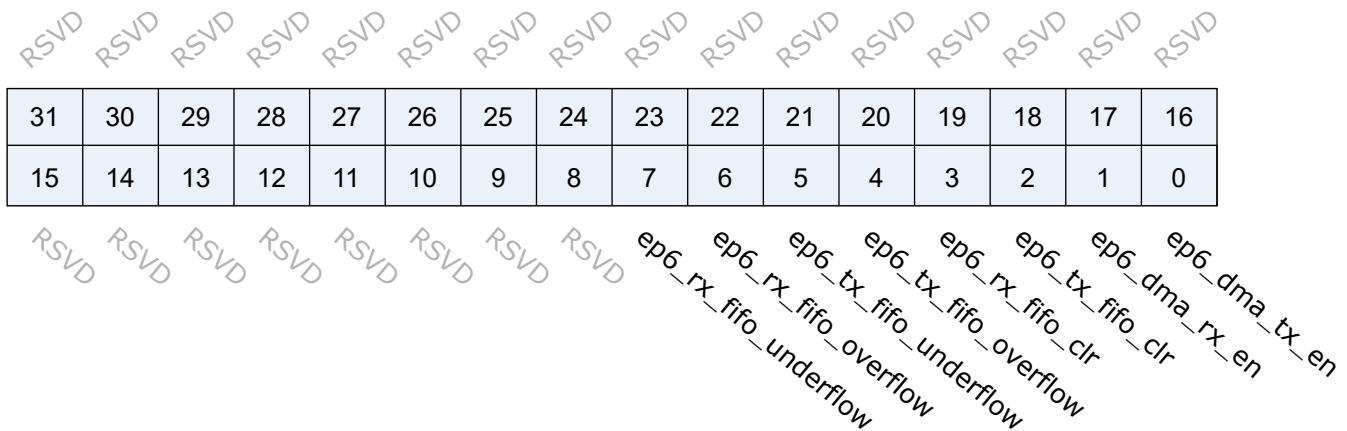
RSVD																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

ep5\_rx\_fifo\_rdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep5_rx_fifo_rdata	r	8'h0	

#### 18.4.41 ep6\_fifo\_config

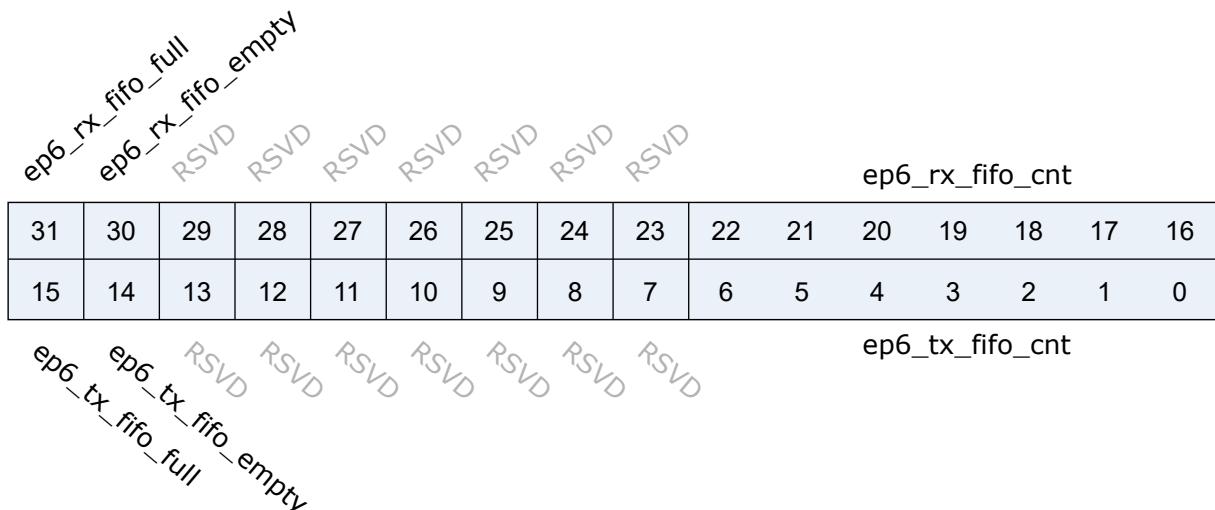
地址: 0x4000d960



位	名称	权限	复位值	描述
31:8	RSVD			
7	ep6_rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	ep6_rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	ep6_tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	ep6_tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	ep6_rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	ep6_tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	ep6_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	ep6_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

#### 18.4.42 ep6\_fifo\_status

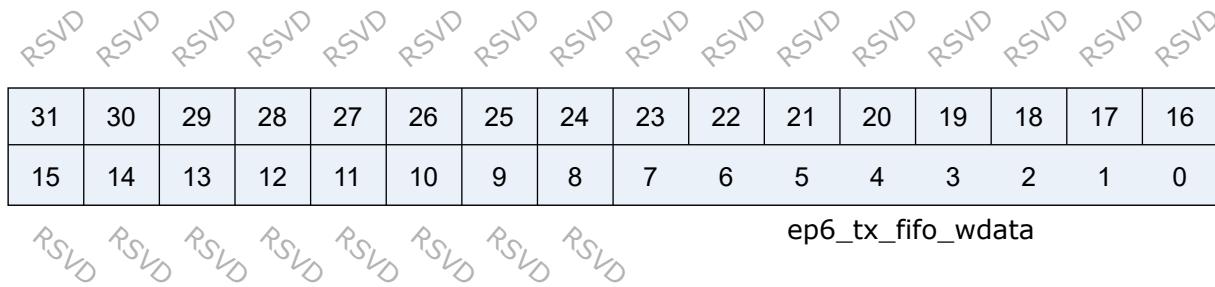
地址: 0x4000d964



位	名称	权限	复位值	描述
31	ep6_rx_fifo_full	r	1'b0	RX FIFO full flag
30	ep6_rx_fifo_empty	r	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	ep6_rx_fifo_cnt	r	7'd0	RX FIFO available count
15	ep6_tx_fifo_full	r	1'b0	TX FIFO full flag
14	ep6_tx_fifo_empty	r	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	ep6_tx_fifo_cnt	r	7'd64	TX FIFO available count

#### 18.4.43 ep6\_tx\_fifo\_wdata

地址: 0x4000d968



位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep6_tx_fifo_wdata	w	x	

#### 18.4.44 ep6\_rx\_fifo\_rdata

地址: 0x4000d96c

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

ep6\_rx\_fifo\_rdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep6_rx_fifo_rdata	r	8'h0	

#### 18.4.45 ep7\_fifo\_config

地址: 0x4000d970

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

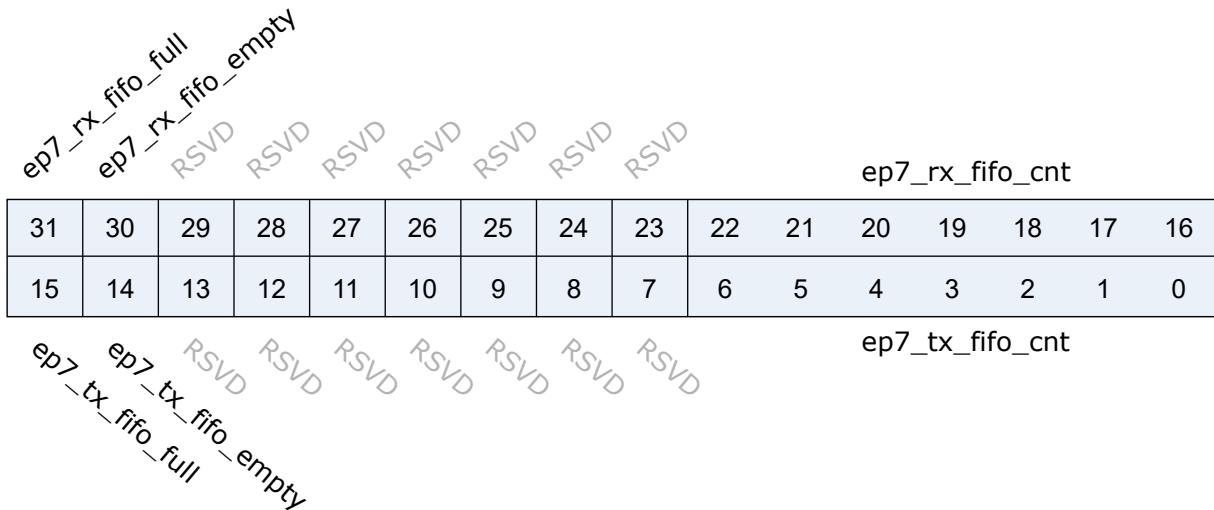
ep7\_rx\_fifo\_underflow      ep7\_rx\_fifo\_overflow      ep7\_tx\_fifo\_overflow      ep7\_tx\_fifo\_clr      ep7\_dma\_rx\_en      ep7\_dma\_tx\_en

位	名称	权限	复位值	描述
31:8	RSVD			
7	ep7_rx_fifo_underflow	r	1'b0	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr

位	名称	权限	复位值	描述
6	ep7_rx_fifo_overflow	r	1'b0	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	ep7_tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	ep7_tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	ep7_rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2	ep7_tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	ep7_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
0	ep7_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

#### 18.4.46 ep7\_fifo\_status

地址: 0x4000d974



位	名称	权限	复位值	描述
31	ep7_rx_fifo_full	r	1'b0	RX FIFO full flag
30	ep7_rx_fifo_empty	r	1'b1	RX FIFO empty flag
29:23	RSVD			
22:16	ep7_rx_fifo_cnt	r	7'd0	RX FIFO available count
15	ep7_tx_fifo_full	r	1'b0	TX FIFO full flag
14	ep7_tx_fifo_empty	r	1'b1	TX FIFO empty flag
13:7	RSVD			
6:0	ep7_tx_fifo_cnt	r	7'd64	TX FIFO available count

#### **18.4.47 ep7\_tx\_fifo\_wdata**

地址: 0x4000d978

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

ep7\_tx\_fifo\_wdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep7_tx_fifo_wdata	w	x	

#### **18.4.48 ep7\_rx\_fifo\_rdata**

地址: 0x4000d97c

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

ep7\_rx\_fifo\_rdata

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	ep7_rx_fifo_rdata	r	8'h0	

#### **18.4.49 xcvr\_if\_config**

地址: 0x4000d9fc

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

*RSVD*    *RSVD*    *RSVD*    *RSVD*    *cr\_xcvr\_om\_rx\_dn*    *cr\_xcvr\_om\_rx\_dp*    *cr\_xcvr\_om\_rx\_sel*    *cr\_xcvr\_force\_rx\_dn*    *cr\_xcvr\_force\_rx\_dp*    *cr\_xcvr\_force\_tx\_en*    *cr\_xcvr\_force\_tx\_dp*    *cr\_xcvr\_force\_tx\_en*    *cr\_xcvr\_force\_tx\_dp*    *cr\_xcvr\_force\_tx\_en*    *cr\_xcvr\_force\_tx\_dp*

位	名称	权限	复位值	描述
31	sts_vbus_det	r	1'b0	Transceiver VBUS detection status
30:12	RSVD			
11	cr_xcvr_om_rx_dn	r/w	1'b0	Transceiver RX signals output mode CR value
10	cr_xcvr_om_rx_dp	r/w	1'b1	Transceiver RX signals output mode CR value
9	cr_xcvr_om_rx_d	r/w	1'b1	Transceiver RX signals output mode CR value
8	cr_xcvr_om_rx_sel	r/w	1'b0	Select signal of transceiver RX signals in output mode (tx_-oe# = 0) 1'b0: rx_d/dp/dn is directly from transceiver 1'b1: rx_d/dp/dn is controlled by CR (cr_xcvr_om_rx_-d/dp/dn)
7	cr_xcvr_force_rx_dn	r/w	1'b0	Transceiver RX signals force mode value
6	cr_xcvr_force_rx_dp	r/w	1'b1	Transceiver RX signals force mode value
5	cr_xcvr_force_rx_d	r/w	1'b1	Transceiver RX signals force mode value
4	cr_xcvr_force_rx_en	r/w	1'b0	Enable signal of transceiver RX signals force mode 1'b0: rx_d/dp/dn is from Transceiver 1'b1: tx_d/dp/dn is controlled by CR (cr_xcvr_force_rx_-d/dp/dn)
3	cr_xcvr_force_tx_dn	r/w	1'b0	Transceiver TX signals force mode value
2	cr_xcvr_force_tx_dp	r/w	1'b1	Transceiver TX signals force mode value
1	cr_xcvr_force_tx_oe	r/w	1'b0	Transceiver TX signals force mode value
0	cr_xcvr_force_tx_en	r/w	1'b0	Enable signal of transceiver TX signals force mode 1'b0: tx_oe/dp/dn is controlled by HW 1'b1: tx_oe/dp/dn is controlled by CR (cr_xcvr_force_tx_-oe/dp/dn)



# 19

## 版本信息

表 19.1: 文档版本修改信息

日期	版本	修改内容
2020/9/9	1.0	初版
2021/3/8	1.1	修改 DAC 对应的 GPIO 口
2021/9/30	1.2	更新 KeyScan 寄存器描述, 低功耗唤醒源介绍