



BL616/BL618

参考手册

Version: 0.98

Copyright @ 2024

www.bouffalolab.com

1 系统和存储器概述	28
1.1 系统架构	28
1.2 处理器	29
1.2.1 主要功能特点	29
1.2.2 扩展功能	30
1.2.3 实现标准	30
1.3 启动流程	31
1.3.1 一级引导程序	32
1.3.2 二级引导程序	32
1.3.3 应用程序启动阶段	32
1.3.3.1 硬件和系统运行环境初始化	33
1.3.3.2 运行主任务	33
1.3.4 从 UART/USB/SDIO 启动	33
1.4 地址映射	34
1.5 中断源	35
1.6 外设概述	37
2 复位和时钟	39
2.1 简介	39
2.2 复位管理	39
2.3 时钟源	40
3 GLB	42
3.1 简介	42
3.2 功能描述	42
3.2.1 时钟管理	42
3.2.2 复位管理	42
3.2.3 总线管理	42

3.2.4	内存管理	42
3.2.5	LDO 管理	43
4	GPIO	44
4.1	简介	44
4.2	主要特征	44
4.3	GPIO 功能列表	44
4.4	GPIO 输入设置	46
4.5	GPIO 输出设置	46
4.5.1	普通输出模式	46
4.5.2	Set/Clear 输出模式	46
4.5.3	可编程输出模式	47
4.5.4	可编程 Set/Clear 输出模式	50
4.6	I/O FIFO	50
4.7	I/O 中断	50
4.8	寄存器描述	51
4.8.1	dac_cfg0	53
4.8.2	dac_cfg1	54
4.8.3	dac_cfg2	55
4.8.4	dac_cfg3	55
4.8.5	gpio_cfg0	56
4.8.6	gpio_cfg1	58
4.8.7	gpio_cfg2	60
4.8.8	gpio_cfg3	62
4.8.9	gpio_cfg4	64
4.8.10	gpio_cfg5	66
4.8.11	gpio_cfg6	68
4.8.12	gpio_cfg7	70
4.8.13	gpio_cfg8	72
4.8.14	gpio_cfg9	74
4.8.15	gpio_cfg10	76
4.8.16	gpio_cfg11	78
4.8.17	gpio_cfg12	80
4.8.18	gpio_cfg13	82
4.8.19	gpio_cfg14	84
4.8.20	gpio_cfg15	86
4.8.21	gpio_cfg16	88
4.8.22	gpio_cfg17	90
4.8.23	gpio_cfg18	92
4.8.24	gpio_cfg19	94

4.8.25	gpio_cfg20	96
4.8.26	gpio_cfg21	98
4.8.27	gpio_cfg22	100
4.8.28	gpio_cfg23	102
4.8.29	gpio_cfg24	104
4.8.30	gpio_cfg25	106
4.8.31	gpio_cfg26	108
4.8.32	gpio_cfg27	110
4.8.33	gpio_cfg28	112
4.8.34	gpio_cfg29	114
4.8.35	gpio_cfg30	116
4.8.36	gpio_cfg31	118
4.8.37	gpio_cfg32	120
4.8.38	gpio_cfg33	122
4.8.39	gpio_cfg34	124
4.8.40	gpio_cfg128	126
4.8.41	gpio_cfg129	128
4.8.42	gpio_cfg136	128
4.8.43	gpio_cfg137	130
4.8.44	gpio_cfg138	130
4.8.45	gpio_cfg139	133
4.8.46	gpio_cfg141	137
4.8.47	gpio_cfg142	137
4.8.48	gpio_cfg143	138
4.8.49	gpio_cfg144	140
5	ADC	141
5.1	简介	141
5.2	主要特点	141
5.3	功能描述	142
5.3.1	ADC 引脚和内部信号	143
5.3.2	ADC 通道	143
5.3.3	ADC 时钟	144
5.3.4	ADC 转换模式	145
5.3.5	ADC 结果	145
5.3.6	ADC 中断	146
5.3.7	ADC FIFO	146
5.3.8	ADC 设置流程	147
5.3.9	VBAT 测量	147
5.3.10	TSEN 测量	148

5.4	寄存器描述	148
5.4.1	gpadc_config	149
5.4.2	gpadc_dma_rdata	150
5.4.3	gpadc_reg_cmd	150
5.4.4	gpadc_reg_config1	153
5.4.5	gpadc_reg_config2	156
5.4.6	gpadc_reg_scn_pos1	158
5.4.7	gpadc_reg_scn_pos2	159
5.4.8	gpadc_reg_scn_neg1	159
5.4.9	gpadc_reg_scn_neg2	160
5.4.10	gpadc_reg_status	161
5.4.11	gpadc_reg_isr	162
6	DAC	163
6.1	简介	163
6.2	主要特点	163
6.3	功能描述	163
6.3.1	DAC 通道使能	164
6.3.2	DAC 数据格式	165
6.3.3	DAC 输出电压	165
6.3.4	DAC 转换	166
6.3.4.1	独立模式时 CPU 方式	166
6.3.4.2	独立模式时 DMA 方式	166
6.3.4.3	联合模式，作为 AudioDAC 的模拟输出	167
6.4	寄存器描述	167
6.4.1	gpadc_config	167
6.4.2	gpadc_dma_rdata	169
6.4.3	gpdac_config	169
6.4.4	gpdac_dma_config	170
6.4.5	gpdac_dma_wdata	171
6.4.6	gpdac_tx_fifo_status	172
6.4.7	dac_cfg0	172
6.4.8	dac_cfg1	173
6.4.9	dac_cfg2	174
6.4.10	dac_cfg3	174
7	DMA	176
7.1	简介	176
7.2	主要特征	176
7.3	功能描述	177
7.3.1	工作原理	177

7.3.2	DMA 通道配置	178
7.3.3	外设支持	178
7.3.4	链表模式	179
7.3.5	DMA 中断	180
7.4	传输模式	180
7.4.1	内存到内存	180
7.4.2	内存到外设	181
7.4.3	外设到内存	181
7.5	寄存器描述	182
7.5.1	DMA_IntStatus	183
7.5.2	DMA_IntTCStatus	183
7.5.3	DMA_IntTCClear	184
7.5.4	DMA_IntErrorStatus	184
7.5.5	DMA_IntErrClr	185
7.5.6	DMA_RawIntTCStatus	185
7.5.7	DMA_RawIntErrorStatus	185
7.5.8	DMA_EnbldChns	186
7.5.9	DMA_SoftBReq	186
7.5.10	DMA_SoftSReq	187
7.5.11	DMA_SoftLBReq	187
7.5.12	DMA_SoftLSReq	187
7.5.13	DMA_Config	188
7.5.14	DMA_Sync	188
7.5.15	DMA_C0SrcAddr	189
7.5.16	DMA_C0DstAddr	189
7.5.17	DMA_C0LLI	189
7.5.18	DMA_C0Control	190
7.5.19	DMA_C0Config	191
7.5.20	DMA_C1SrcAddr	192
7.5.21	DMA_C1DstAddr	193
7.5.22	DMA_C1LLI	193
7.5.23	DMA_C1Control	193
7.5.24	DMA_C1Config	195
7.5.25	DMA_C2SrcAddr	196
7.5.26	DMA_C2DstAddr	196
7.5.27	DMA_C2LLI	196
7.5.28	DMA_C2Control	197
7.5.29	DMA_C2Config	198
7.5.30	DMA_C3SrcAddr	199

7.5.31	DMA_C3DstAddr	200
7.5.32	DMA_C3LLI	200
7.5.33	DMA_C3Control	200
7.5.34	DMA_C3Config	202
8	IR	204
8.1	简介	204
8.2	主要特征	204
8.3	功能描述	204
8.3.1	固定协议接收	204
8.3.2	脉冲宽度接收	206
8.3.3	IR 中断	206
8.4	寄存器描述	206
8.4.1	irrx_config	207
8.4.2	irrx_int_sts	208
8.4.3	irrx_pw_config	209
8.4.4	irrx_data_count	209
8.4.5	irrx_data_word0	209
8.4.6	irrx_data_word1	210
8.4.7	ir_fifo_config_0	210
8.4.8	ir_fifo_config_1	211
8.4.9	ir_fifo_rdata	211
9	SPI	212
9.1	简介	212
9.2	主要特征	212
9.3	功能描述	213
9.3.1	基本架构	213
9.3.2	时序控制	213
9.3.3	主设备连续传输模式	214
9.3.4	主从设备数据收发	214
9.3.5	接收忽略功能	215
9.3.6	滤波功能	215
9.3.7	可调整字节传输顺序	216
9.3.8	可配置每个字节的 MSB/LSB 优先传输	216
9.3.9	从模式超时机制	216
9.3.10	I/O 传输模式	217
9.3.11	DMA 传输模式	217
9.3.12	SPI 中断	217
9.4	寄存器描述	218
9.4.1	spi_config	218

9.4.2	spi_int_sts	220
9.4.3	spi_bus_busy	221
9.4.4	spi_prd_0	222
9.4.5	spi_prd_1	222
9.4.6	spi_rxd_ignr	222
9.4.7	spi_sto_value	223
9.4.8	spi_fifo_config_0	224
9.4.9	spi_fifo_config_1	224
9.4.10	spi_fifo_wdata	225
9.4.11	spi_fifo_rdata	226
9.4.12	backup_io_en	226
10	UART	227
10.1	简介	227
10.2	主要特征	227
10.3	功能描述	228
10.3.1	数据格式描述	228
10.3.2	基本架构	229
10.3.3	发送器	229
10.3.4	接收器	230
10.3.5	波特率设定	230
10.3.6	滤波	231
10.3.7	自动波特率检测	231
10.3.8	硬件流控	232
10.3.9	DMA 传输	233
10.3.10	LIN 总线支持	233
10.3.11	RS485 模式	235
10.4	UART 中断	235
10.4.1	TX/RX 传输完成中断	236
10.4.2	TX/RX FIFO 请求中断	236
10.4.3	RX 超时中断	237
10.4.4	RX 奇偶校验错误中断	237
10.4.5	TX/RX FIFO 溢出中断	237
10.4.6	RX BCR 中断	237
10.4.7	LIN 同步错误中断	237
10.4.8	通用/固定字符模式自动波特率检测中断	238
10.5	寄存器描述	238
10.5.1	utx_config	239
10.5.2	urx_config	240
10.5.3	uart_bit_prd	241

10.5.4	data_config	242
10.5.5	utx_ir_position	242
10.5.6	urx_ir_position	243
10.5.7	urx_rto_timer	243
10.5.8	uart_sw_mode	243
10.5.9	uart_int_sts	244
10.5.10	uart_int_mask	245
10.5.11	uart_int_clear	246
10.5.12	uart_int_en	247
10.5.13	uart_status	248
10.5.14	sts_urx_abr_prd	248
10.5.15	urx_abr_prd_b01	249
10.5.16	urx_abr_prd_b23	249
10.5.17	urx_abr_prd_b45	250
10.5.18	urx_abr_prd_b67	250
10.5.19	urx_abr_pw_tol	250
10.5.20	urx_bcr_int_cfg	251
10.5.21	utx_rs485_cfg	251
10.5.22	uart_fifo_config_0	252
10.5.23	uart_fifo_config_1	253
10.5.24	uart_fifo_wdata	253
10.5.25	uart_fifo_rdata	254
11	I2C	255
11.1	简介	255
11.2	主要特征	255
11.3	功能描述	255
11.3.1	基本架构	256
11.3.2	起始和停止条件	256
11.3.3	数据传输格式	257
11.3.4	仲裁	259
11.4	I2C 时钟设定	260
11.5	I2C 配置流程	261
11.5.1	配置项	261
11.5.2	读写标志位	261
11.5.3	从设备地址	261
11.5.4	从设备寄存器地址	261
11.5.5	从设备寄存器地址长度	261
11.5.6	数据	261
11.5.7	数据长度	262

11.5.8	使能信号	262
11.6	FIFO 管理	262
11.7	DMA 功能	263
11.7.1	DMA 发送流程	263
11.7.2	DMA 接收流程	264
11.8	中断	264
11.9	寄存器描述	265
11.9.1	i2c_config	265
11.9.2	i2c_int_sts	266
11.9.3	i2c_sub_addr	268
11.9.4	i2c_bus_busy	268
11.9.5	i2c_prd_start	269
11.9.6	i2c_prd_stop	269
11.9.7	i2c_prd_data	270
11.9.8	i2c_fifo_config_0	270
11.9.9	i2c_fifo_config_1	271
11.9.10	i2c_fifo_wdata	272
11.9.11	i2c_fifo_rdata	272
12	PWM	273
12.1	简介	273
12.2	主要特征	273
12.3	功能描述	273
12.3.1	时钟与分频器	273
12.3.2	有效电平	274
12.3.3	脉冲产生原理	274
12.3.4	刹车	275
12.3.5	死区	275
12.3.6	周期与占空比计算	276
12.3.7	PWM 中断	276
12.3.8	ADC 联动	277
12.4	寄存器描述	278
12.4.1	pwm_int_config	278
12.4.2	pwm_mc0_config0	279
12.4.3	pwm_mc0_config1	280
12.4.4	pwm_mc0_period	284
12.4.5	pwm_mc0_dead_time	285
12.4.6	pwm_mc0_ch0_thre	286
12.4.7	pwm_mc0_ch1_thre	286
12.4.8	pwm_mc0_ch2_thre	287

12.4.9	pwm_mc0_ch3_thre	287
12.4.10	pwm_mc0_int_sts	287
12.4.11	pwm_mc0_int_mask	288
12.4.12	pwm_mc0_int_clear	289
12.4.13	pwm_mc0_int_en	290
13	TIMER	291
13.1	简介	291
13.2	主要特征	292
13.3	功能描述	293
13.3.1	通用定时器工作原理	293
13.3.1.1	PreLoad 模式	294
13.3.1.2	FreeRun 模式	294
13.3.1.3	测量外部 GPIO 脉冲宽度	295
13.3.2	Watchdog 定时器工作原理	296
13.3.3	定时器 Alarm 设定	296
13.3.4	Watchdog Alarm 设定	296
13.4	寄存器描述	297
13.4.1	TCCR	299
13.4.2	TMR2_0	299
13.4.3	TMR2_1	299
13.4.4	TMR2_2	300
13.4.5	TMR3_0	300
13.4.6	TMR3_1	300
13.4.7	TMR3_2	301
13.4.8	TCR2	301
13.4.9	TCR3	301
13.4.10	TSR2	302
13.4.11	TSR3	302
13.4.12	TIER2	303
13.4.13	TIER3	303
13.4.14	TPLVR2	304
13.4.15	TPLVR3	304
13.4.16	TPLCR2	305
13.4.17	TPLCR3	305
13.4.18	WMER	306
13.4.19	WMR	306
13.4.20	WVR	307
13.4.21	WSR	307
13.4.22	TICR2	307

13.4.23	TICR3	308
13.4.24	WICR	309
13.4.25	TCER	309
13.4.26	TCMR	310
13.4.27	TILR2	310
13.4.28	TILR3	311
13.4.29	WCR	311
13.4.30	WFAR	312
13.4.31	WSAR	312
13.4.32	TCDR	313
13.4.33	GPIO	313
13.4.34	GPIO_LAT1	314
13.4.35	GPIO_LAT2	314
14	I2S	315
14.1	简介	315
14.2	主要特征	315
14.3	功能描述	316
14.3.1	数据格式描述	316
14.3.2	I2S 单通道模式	318
14.3.3	数据 MSB/LSB 调整	318
14.3.4	双声道数据合并与交换	318
14.3.5	静音模式	319
14.3.6	时钟源	319
14.3.7	I2S 中断	319
14.4	寄存器描述	319
14.4.1	i2s_config	320
14.4.2	i2s_int_sts	321
14.4.3	i2s_bclk_config	322
14.4.4	i2s_fifo_config_0	323
14.4.5	i2s_fifo_config_1	324
14.4.6	i2s_fifo_wdata	325
14.4.7	i2s_fifo_rdata	325
14.4.8	i2s_io_config	325
15	AudioDAC	327
15.1	简介	327
15.2	主要特征	327
15.3	时钟树	328
15.4	功能描述	328
15.4.1	AudioDAC 中断	328

15.4.2	FIFO 格式控制	329
15.4.3	FIFO 的启动与 DMA 搬运	330
15.4.4	可配置的 Mixer 声道混合器	330
15.4.5	音量控制	331
15.4.6	ZeroDetect	332
15.5	配置流程	332
15.6	寄存器描述	332
15.6.1	audac_0	333
15.6.2	audac_status	333
15.6.3	audac_s0	334
15.6.4	audac_s0_misc	336
15.6.5	audac_zd_0	337
15.6.6	audac_1	338
15.6.7	audac_fifo_ctrl	339
15.6.8	audac_fifo_status	341
15.6.9	audac_fifo_data	342
16	AudioADC	343
16.1	简介	343
16.2	主要特征	343
16.3	时钟树	344
16.4	功能描述	344
16.4.1	音源通道选择	345
16.4.2	AUADC 中断	345
16.4.3	FIFO 格式控制	345
16.4.4	测量模式	346
16.4.5	FIFO 的启动与 DMA 搬运	346
16.5	配置流程	346
16.6	寄存器描述	347
16.6.1	audpdm_top	347
16.6.2	audpdm_itf	348
16.6.3	pdm_adc_0	349
16.6.4	pdm_adc_1	349
16.6.5	pdm_dac_0	350
16.6.6	pdm_pdm_0	350
16.6.7	pdm_adc_s0	351
16.6.8	audadc_ana_cfg1	351
16.6.9	audadc_ana_cfg2	353
16.6.10	audadc_cmd	354
16.6.11	audadc_data	357

16.6.12	audadc_rx_fifo_ctrl	358
16.6.13	audadc_rx_fifo_status	361
16.6.14	audadc_rx_fifo_data	362
17	EMAC	363
17.1	简介	363
17.2	主要特征	363
17.3	功能描述	364
17.4	时钟	365
17.5	收发缓冲描述符 (BD, Buffer Descriptor)	366
17.6	PHY 交互	368
17.7	EMAC 工作流程	368
17.7.1	PHY 初始化	368
17.7.2	发送数据帧	369
17.7.3	接收数据帧	370
17.8	寄存器描述	370
17.8.1	MODE	371
17.8.2	INT_SOURCE	373
17.8.3	INT_MASK	374
17.8.4	IPGT	375
17.8.5	PACKETLEN	376
17.8.6	COLLCONFIG	376
17.8.7	TX_BD_NUM	377
17.8.8	MIIMODE	378
17.8.9	MIICOMMAND	378
17.8.10	MIIADDRESS	379
17.8.11	MIITX_DATA	380
17.8.12	MIIRX_DATA	380
17.8.13	MIISTATUS	381
17.8.14	MAC_ADDR0	381
17.8.15	MAC_ADDR1	382
17.8.16	HASH0_ADDR	382
17.8.17	HASH1_ADDR	382
17.8.18	TXCTRL	383
18	USB	384
18.1	简介	384
18.2	主要特征	384
18.2.1	通用特征	384
18.2.2	主机模式特征	384
18.2.3	从机模式特征	385

18.3	功能描述	385
18.3.1	USB 时钟	385
18.3.2	USB 设备	385
18.3.2.1	USB 设备端点	385
18.3.2.2	USB 设备 FIFO	385
18.3.2.3	USB 设备 VDMA	385
18.3.2.4	USB 设备中断	385
18.3.2.5	USB 设备枚举过程	385
18.3.3	USB 主机	385
19	CAM	386
19.1	简介	386
19.2	主要特征	386
19.3	功能描述	387
19.3.1	DVP(Digital Video Port) 信号与配置	387
19.3.2	YCbCr 格式	387
19.3.3	视频模式/照片模式	388
19.3.4	RGB888 转 RGB565/RGBA8888 输出	388
19.3.5	图像矩形裁剪	388
19.3.6	帧取舍功能	388
19.3.7	行帧同步信号完整性检测	389
19.3.8	缓存图像信息	389
19.3.9	支持多种中断信息 (可独立开关配置)	389
19.4	寄存器描述	390
19.4.1	cam_dvp2axi_configue	391
19.4.2	cam_dvp2axi_addr_start	393
19.4.3	cam_dvp2axi_mem_bcmt	393
19.4.4	cam_dvp_status_and_error	393
19.4.5	cam_dvp2axi_frame_bcmt	394
19.4.6	cam_dvp_frame_fifo_pop	395
19.4.7	cam_dvp2axi_frame_vld	395
19.4.8	cam_dvp2axi_frame_period	396
19.4.9	cam_dvp2axi_misc	396
19.4.10	cam_dvp2axi_hsync_crop	397
19.4.11	cam_dvp2axi_vsync_crop	397
19.4.12	cam_dvp2axi_fram_exm	398
19.4.13	cam_frame_start_addr0	398
19.4.14	cam_frame_start_addr1	398
19.4.15	cam_frame_start_addr2	399
19.4.16	cam_frame_start_addr3	399

19.4.17	cam_frame_id_sts01	399
19.4.18	cam_frame_id_sts23	400
19.4.19	cam_dvp_debug	400
20	MJPEG	401
20.1	简介	401
20.2	主要特征	401
20.3	功能描述	402
20.3.1	输入配置	402
20.3.2	量化系数表	402
20.3.3	软件模式和联动模式	402
20.3.4	swap 模式	403
20.3.5	kick 模式	403
20.3.6	jpg 功能	403
20.3.7	缓存图片信息	403
20.3.8	支持多种中断信息 (可独立开关配置)	403
20.4	寄存器描述	403
20.4.1	mjpeg_control_1	405
20.4.2	mjpeg_control_2	406
20.4.3	mjpeg_yy_frame_addr	407
20.4.4	mjpeg_uv_frame_addr	407
20.4.5	mjpeg_yuv_mem	408
20.4.6	jpeg_frame_addr	408
20.4.7	jpeg_store_memory	408
20.4.8	mjpeg_control_3	409
20.4.9	mjpeg_frame_fifo_pop	410
20.4.10	mjpeg_frame_size	411
20.4.11	mjpeg_header_byte	411
20.4.12	mjpeg_swap_mode	412
20.4.13	mjpeg_swap_bit_cnt	413
20.4.14	mjpeg_yuv_mem_sw	413
20.4.15	mjpeg_Y_frame_read_status_1	413
20.4.16	mjpeg_Y_frame_read_status_2	414
20.4.17	mjpeg_Y_frame_write_status	414
20.4.18	mjpeg_UV_frame_read_status_1	415
20.4.19	mjpeg_UV_frame_read_status_2	415
20.4.20	mjpeg_UV_frame_write_status	416
20.4.21	mjpeg_frame_w_hblk_status	416
20.4.22	mjpeg_start_addr0	417
20.4.23	mjpeg_bit_cnt0	417

20.4.24	mjpeg_start_addr1	418
20.4.25	mjpeg_bit_cnt1	418
20.4.26	mjpeg_start_addr2	418
20.4.27	mjpeg_bit_cnt2	419
20.4.28	mjpeg_start_addr3	419
20.4.29	mjpeg_bit_cnt3	419
20.4.30	mjpeg_q_enc	420
20.4.31	mjpeg_frame_id_10	420
20.4.32	mjpeg_frame_id_32	421
20.4.33	mjpeg_debug	421
21	DBI	423
21.1	简介	423
21.2	主要特征	423
21.3	功能描述	424
21.3.1	DBI Type B	425
21.3.1.1	写时序	425
21.3.1.2	读时序	427
21.3.1.3	输出 RGB565	427
21.3.1.4	输出 RGB666	428
21.3.1.5	输出 RGB888	429
21.3.2	DBI Type C 3-Line	430
21.3.2.1	写时序	430
21.3.2.2	读时序	431
21.3.2.3	输出 RGB565	432
21.3.2.4	输出 RGB666	432
21.3.2.5	输出 RGB888	433
21.3.3	DBI Type C 4-Line	433
21.3.3.1	写时序	433
21.3.3.2	读时序	434
21.3.3.3	输出 RGB565	434
21.3.3.4	输出 RGB666	435
21.3.3.5	输出 RGB888	436
21.3.4	QSPI	436
21.3.4.1	写时序	436
21.3.4.2	读时序	438
21.3.4.3	1 线命令、1 线地址、1 线数据模式	438
21.3.4.4	1 线命令、1 线地址、4 线数据模式	441
21.3.4.5	1 线命令、4 线地址、4 线数据模式	444
21.3.5	输入像素格式	445

21.3.6	CS 信号拉高释放条件配置	447
21.3.7	中断	447
21.3.8	DMA	447
21.4	寄存器描述	447
21.4.1	dbi_config	448
21.4.2	qspi_config	450
21.4.3	dbi_pix_cnt	451
21.4.4	dbi_prd	452
21.4.5	dbi_cmd	452
21.4.6	dbi_qspi_adr	453
21.4.7	dbi_rdata_0	453
21.4.8	dbi_rdata_1	453
21.4.9	dbi_int_sts	454
21.4.10	dbi_yuv_rgb_config_0	455
21.4.11	dbi_yuv_rgb_config_1	455
21.4.12	dbi_yuv_rgb_config_2	456
21.4.13	dbi_yuv_rgb_config_3	456
21.4.14	dbi_yuv_rgb_config_4	457
21.4.15	dbi_yuv_rgb_config_5	458
21.4.16	dbi_fifo_config_0	458
21.4.17	dbi_fifo_config_1	459
21.4.18	dbi_fifo_wdata	460
22	SDH	461
22.1	简介	461
22.2	主要特征	461
22.3	功能描述	461
22.3.1	SDH 总体结构	462
22.3.2	寄存器映射	463
22.3.3	多卡槽支持	464
22.3.4	支持 DMA	465
22.3.5	SD 命令生成	465
22.3.6	挂起和恢复机制	466
22.3.7	缓冲区控制	466
22.3.7.1	缓冲区指针的控制	467
22.3.7.2	确定缓冲块长度	467
22.3.7.3	分割大数据传输	467
22.3.8	中断控制寄存器之间的关系	467
22.3.9	硬件框图和定时部分	468
22.3.10	自动 CMD12	468

22.3.11 控制 SDCLK	469
22.3.12 高级 DMA	469
22.3.12.1 ADMA2 的框图	470
22.3.12.2 数据地址和数据长度要求	471
22.3.12.3 描述符表	472
22.3.12.4 ADMA2 状态	473
22.3.13 测试寄存器	474
22.3.14 块计数	474
22.3.15 采样时钟调谐	474
22.3.16 SD 主机标准寄存器	475
22.3.16.1 SD 主机控制寄存器映射	475
22.3.16.2 配置寄存器类型	476
22.3.16.3 寄存器初始值	477
22.3.16.4 寄存器的保留位	477
23 SDIO	478
23.1 简介	478
23.2 主要特征	478
23.3 功能描述	478
23.3.1 SDIO 信号定义	478
23.3.1.1 SDIO 卡类型	478
23.3.1.2 SDIO 卡模式	479
23.3.1.3 信号引脚	479
23.3.2 SDIO 卡初始化	479
23.3.2.1 I/O 卡初始化差异	479
23.3.2.2 IO_SEND_OP_COND 命令 (CMD5)	481
23.3.2.3 IO_SEND_OP_COND 响应 (R4)	481
23.3.2.4 组合卡的特殊初始化注意事项	483
23.3.3 与 SD 内存规范的差异	483
23.3.3.1 SDIO 命令列表	483
23.3.3.2 不支持的 SD 内存命令	486
23.3.3.3 改进的 R6 响应	487
23.3.3.4 SDIO 复位	488
23.3.3.5 总线宽度	488
23.3.3.6 卡检测电阻	488
23.3.3.7 数据传输块大小	488
23.3.3.8 数据传输中止	488
23.3.4 新的 I/O 读或写命令	489
23.3.4.1 IO_RW_DIRECT 命令 (CMD52)	489
23.3.4.2 IO_RW_DIRECT 响应 (R5)	490

23.3.4.3	IO_RW_EXTENDED 命令 (CMD53)	491
23.3.5	SDIO 卡内部操作	492
23.3.5.1	概述	492
23.3.5.2	寄存器访问时间	494
23.3.5.3	中断	494
23.3.5.4	挂起/恢复	494
23.3.5.5	读等待	494
23.3.5.6	数据传输中的 CMD52	495
23.3.5.7	固定内部映射	495
23.3.5.8	公共 I/O 区 (CIA)	495
23.3.5.9	卡通用控制寄存器 (CCCR)	495
23.3.5.10	功能基础寄存器 (FBR)	495
23.3.5.11	卡信息结构 (CIS)	496
23.3.5.12	多功能 SDIO 卡	496
23.3.5.13	使用 CMD53 设置块大小	496
23.3.5.14	总线状态图	496
23.3.6	嵌入式 I/O 代码存储区 (CSA)	498
23.3.6.1	CSA 访问	498
23.3.6.2	CSA 数据格式	498
23.3.7	SDIO 中断	498
23.3.7.1	SPI 和 SD 1 位模式中断	498
23.3.7.2	SD 4 位模式中断	499
23.3.7.3	中断清除时间	499
23.3.8	SDIO 挂起/恢复操作	499
23.3.9	SDIO 读取等待操作	499
24	LowPower	500
24.1	概述	500
24.2	主要特征	500
24.3	功能描述	501
24.3.1	电源域	501
24.3.2	电源模式	502
24.3.3	唤醒源	503
24.3.4	功耗模式	504
24.3.5	IO 唤醒	505
24.3.6	ACOMP 唤醒	508
24.3.7	IO 双边沿唤醒	508
24.3.8	IO 唤醒触发条件	509
24.4	POR & BOD 说明	510
24.4.1	UVLO 和 POR	510

24.4.2 BOD	510
25 SEC ENG	511
25.1 简介	511
25.2 主要特征	511
25.3 功能描述	511
25.3.1 AES 加速器	511
25.3.1.1 密钥	511
25.3.1.2 链接模式	512
25.3.1.3 明文、密文	512
25.3.1.4 初始化向量	512
25.3.1.5 加解密配置流程	513
25.3.2 SHA 加速器	513
25.3.2.1 SHA 模式	513
25.3.2.2 明文、密文	514
25.3.2.3 配置流程	514
25.3.3 TRNG	514
25.3.3.1 配置流程	514
26 版本信息	515

1.1	BL616 系统架构图	29
1.2	BL618 系统架构图	29
1.1	启动流程框图	31
2.1	时钟架构	41
4.1	普通 GPIO 输出波形	48
4.2	默认电平为高电平，逻辑 1 电平翻转输出波形	48
4.3	默认电平为低电平，逻辑 0 电平翻转输出波形	49
4.4	默认电平为高电平，逻辑 0 电平翻转，逻辑 1 电平翻转输出波形	49
5.1	ADC 基本框图	142
5.2	ADC 时钟	144
6.1	DAC 基本框图	164
7.1	DMA 框图	177
7.2	外设类型选择	178
7.3	LLI 框架	180
8.1	NEC 逻辑波形	205
8.2	NEC 协议波形	205
8.3	RC5 逻辑波形	205
8.4	RC5 协议波形	206
9.1	SPI 基本架构图	213
9.2	SPI 时序图	214
9.3	SPI Ignore 波形图	215
9.4	SPI 滤波波形图	216

10.1 UART 数据格式	228
10.2 UART 基本架构图	229
10.3 UART 采样波形图	230
10.4 UART 滤波波形图	231
10.5 UART 固定字符模式波形图	232
10.6 UART 硬件流控图	232
10.7 一个典型的 LIN 帧	234
10.8 LIN 的 Break 域	234
10.9 LIN 的 Sync 域	234
10.10 LIN 的 ID 域	235
11.1 I2C 基本架构图	256
11.2 I2C 起始和停止条件	257
11.3 主发送和从接收的数据格式	257
11.4 主接收和从发送的数据格式	257
11.5 主发送和从接收的时序	258
11.6 主接收和从发送的时序	258
11.7 主发送和从接收的数据格式 (10bit 从地址)	258
11.8 主接收和从发送的数据格式 (10bit 从地址)	259
11.9 同时传输数据波形示意图	260
11.1 I2C 时钟设定	260
12.1 PWM 输出波形图	275
12.2 PWM 插入死区输出波形图	276
13.1 定时器框图	291
13.2 Watchdog 定时器框图	292
13.1 定时器在 PreLoad 模式下工作时序	294
13.2 Watchdog 工作时序	296
13.3 Watchdog alarm 机制	297
14.1 I2S 协议 Normal 格式	316
14.2 I2S 协议 Left-Justified/Right-Justified 格式	317
14.3 PCM/TDM 模式	318
14.4 I2S 时钟	319
15.1 时钟示意图	328
15.1 模块示意图	328
15.2 Mixer 示意图	331
16.1 时钟示意图	344
16.1 模块框图	344
17.1 EMAC 框图	364

17.1 使用外置的参考时钟源	365
17.2 内部输出 50MHz 参考时钟源	366
19.1 Cam 框图	386
19.1 FIFO 框架	389
19.2 内存	390
20.1 MJPEG YUYV interleave order configuration	402
21.1 DBI 基本框图	425
21.2 写时序	426
21.3 读时序	427
21.4 RGB565 输出	428
21.5 RGB666 输出	429
21.6 RGB888 输出	430
21.7 写时序	431
21.8 读时序	431
21.9 RGB565 输出	432
21.10 RGB666 输出	432
21.11 RGB888 输出	433
21.12 写时序	433
21.13 读时序	434
21.14 RGB565 输出	435
21.15 RGB666 输出	435
21.16 RGB888 输出	436
21.17 写时序	437
21.18 读时序	438
21.19 RGB565 输出	439
21.20 RGB666 输出	440
21.21 RGB888 输出	441
21.22 RGB565 输出	442
21.23 RGB666 输出	443
21.24 RGB888 输出	444
21.25 RGB565 输出	444
21.26 RGB666 输出	445
21.27 RGB888 输出	445
21.28 输入像素 RGB 格式	446
22.1 SDH 硬件和驱动结构图	462
22.2 标准寄存器映射分类图	463
22.3 多卡槽控制器的寄存器映射	464
22.4 用于生成 SD 命令的寄存器	465
22.5 挂起和恢复机制	466

22.6 主机控制器框图	468
22.7 通过 SD 总线电源和 SD 时钟启用控制 SDCLK	469
22.8 ADMA2 的框图	470
22.9 32 位地址描述符表	472
22.10 ADMA2 的状态图	473
22.11 SD 主机控制寄存器映射	475
22.12 寄存器（和寄存器位字段）类型	476
23.1 2 个 4 位 SDIO 卡信号连接示意图	479
23.2 SDIO 对非 I/O 感知初始化的响应	480
23.3 IO_SEND_OP_COND 命令	481
23.4 SD 模式下的响应 R4	481
23.5 SPI 模式下的响应 R4	482
23.6 改进的 R1 响应	482
23.7 SD 模式命令列表	484
23.8 SPI 模式命令列表	485
23.9 不支持的 SD 内存命令	486
23.10 CMD3 的 R6 响应	487
23.11 SDIO R6 状态位	487
23.12 IO_RW_DIRECT 命令	489
23.13 SD 模式下的 IO_RW_DIRECT 响应	490
23.14 SPI 模式下的 IO_RW_DIRECT 响应	491
23.15 IO_RW_EXTENDED 命令	491
23.16 SDIO 内部映射	493
23.17 总线状态机状态图	497
24.1 低功耗模式	500
25.1 AES 运算模式图	512

1.1 启动模式	33
1.1 内存地址映射	34
1.2 地址映射	34
1.1 中断分配	35
1.1 外设列表	37
2.1 软件复位功能表	39
4.1 GPIO 功能列表	45
5.1 ADC 内部信号	143
5.2 ADC 外部引脚	143
5.3 ADC 转换结果含义	145
6.1 数据传输格式	165
6.2 内部参考电压输出电压	165
11.1 I2C 引脚	256
12.1 时钟源	274
12.2 死区长度计算公式	275
13.1 gpio_tmr_clk 功能配置	295
17.1 传输信号	368
24.1 电源模式	502
24.2 电源模式的功耗	503
24.3 唤醒源	503
24.4 IO wakeup	505

26.1 文档版本修改信息	515
---------------------	-----

1.1 系统架构

BL616/BL618 系列芯片采用 RISC-V 32-bit CPU，搭载 16KB D-Cache 和 32KB I-Cache，CPU 主频高达 320MHz，适用于物联网、嵌入式和人工智能等高性能应用领域。

主系统有以下部分组成：

- 总线接口：AXI 总线和 AHB 总线
 - CPU 通过 AXI 总线访问存储器
 - CPU 通过 AHB 总线访问外设
 - 具备低功耗，高性能等特性
- 存储器
 - 480KB SRAM 用于数据存储
 - 128KB ROM
 - 支持内嵌高速 PSRAM，扩展 RAM 容量（仅在 BL618 支持）
- 外设
 - 共有 30 个外设模块，其中包含 USB/SDH/SDU/EMAC（仅在 BL618 支持）/CAM（仅在 BL618 支持）/Display（仅在 BL618 支持）等先进外设

片内集成 Wi-Fi 6/BLE/Zigbee 无线子系统，可以实现多种无线连接和数据传输，提供多样化的连接与传输体验。

BL616/BL618 系统架构如下图所示：

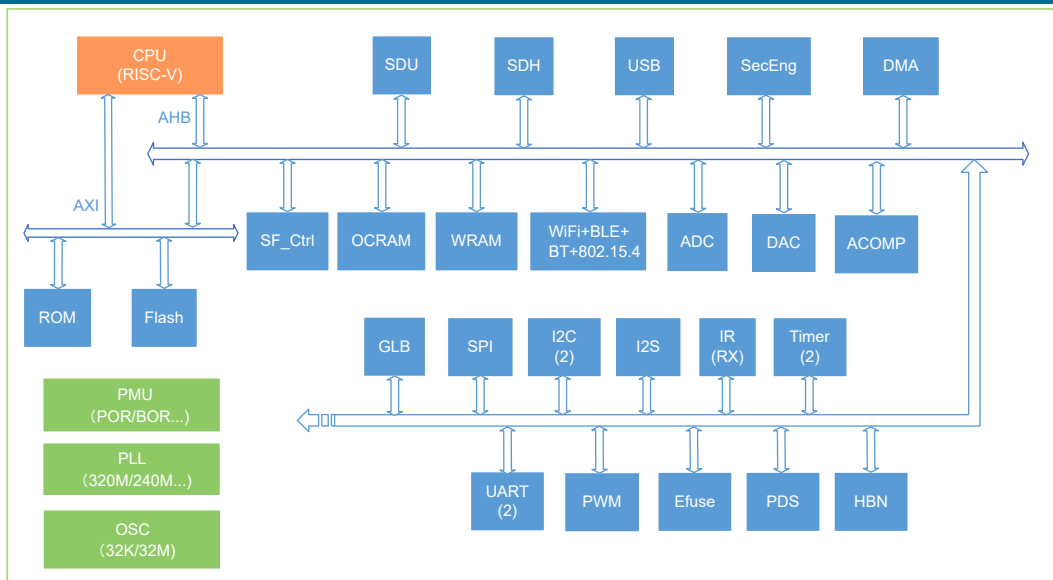


图 1.1: BL616 系统架构图

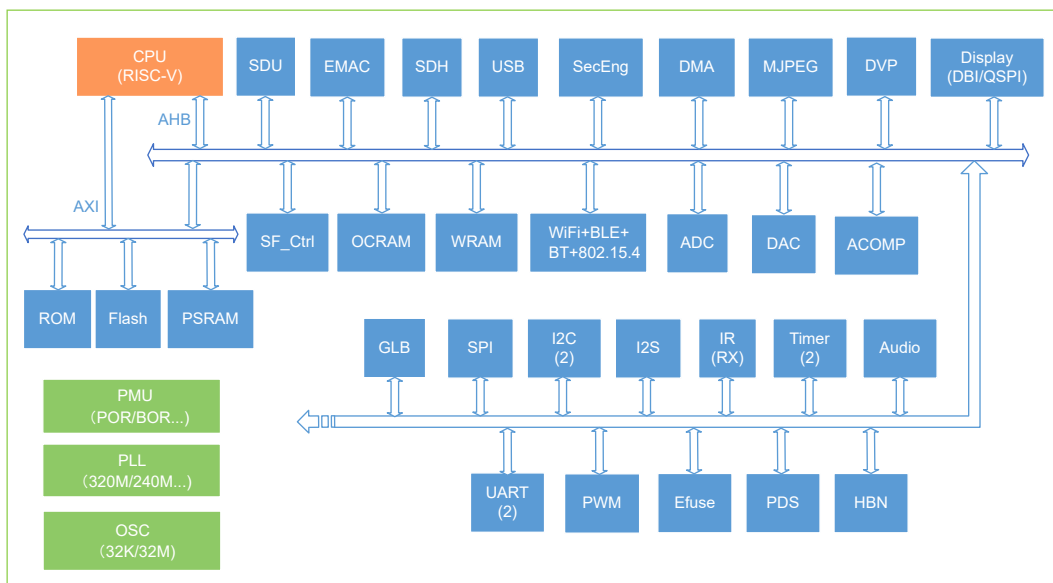


图 1.2: BL618 系统架构图

1.2 处理器

1.2.1 主要功能特点

BL616/BL618 内置一颗 32-bit RISC-V CPU，它采用 5 级流水线结构：取指、译码、执行、访存、写回，是一款高性能嵌入式微处理器。其主要特征如下：

- 32 位 RISC 处理器；
- 支持 RISC-V RV32IMAFCP 指令集；

- 支持 RISC-V 32/16 位混编指令集；
- 支持 RISC-V 机器模式和用户模式；
- 32 个 32 位整型通用寄存器，32 个 32 位/64 位浮点通用寄存器；
- 整型 5 级/浮点 7 级，单发射，顺序执行流水线；
- 支持 AXI4.0 主设备接口以及 AHB5.0 外设接口；
- 32KB I-Cache，两路组相联结构；
- 16KB D-Cache，两路组相联结构；
- 支持非对齐内存访问；
- 双周期硬件乘法器，基 4 硬件除法器；
- 支持 BHT（8K）和 BTB；
- 支持扩展增强指令集；
- 支持 MCU 特性扩展技术，包括中断处理加速技术、MCU 扩展特性；
- 兼容 RISC-V CLIC 中断标准，支持中断嵌套，外部中断源数量 96 个，有 4 个 Bits 可以用于配置中断优先级；
- 兼容 RISC-V PMP 内存保护标准，8 个区域可配置；
- 支持性能监测单元 (HPM)；
- 支持 RISC-V Debug 协议标准；

1.2.2 扩展功能

- 扩展实现了位操作指令，算术运算指令，内存访问增强指令
- 扩展实现了 CACHE 操作指令，同步指令等方便软件人员编程
- 扩展实现了中断投机压栈技术和矢量中断咬尾技术，加速中断响应，中断响应延时为 20 个处理器时钟周期
- 扩展实现了如 NMI，深浅睡眠低功耗模式，低功耗唤醒事件，锁定等 MCU 领域常见的应用需求
- 支持非对齐内存访问，并可软件开关，方便软件人员编程和调试

1.2.3 实现标准

该处理器兼容 RISC-V 标准，具体版本为：

- The RISC-V Instruction Set Manual, Volume I: RISC-V User-Level ISA, Version 2.2
- The RISC-V Instruction Set Manual, Volume II: RISC-V Privileged Architecture, Version 1.10
- RISC-V Core-Local Interrupt Controller (CLIC) Version 0.8

- RISC-V External Debug Support Version 0.13.2
- RISC-V “P” Extension Proposal Version 0.9

1.3 启动流程

本节将会介绍 BL616/618 从上电启动到运行 main 函数中间所经历的步骤（即芯片启动流程）。

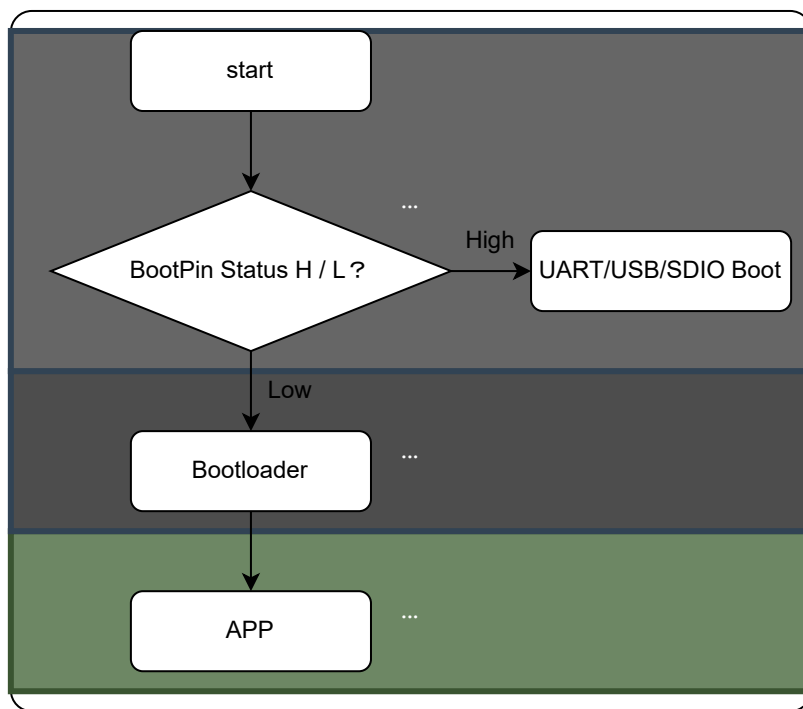


图 1.1: 启动流程框图

总体上芯片启动分为两类情况，启动流程可以分成如下 2~3 个步骤：

两种情况分别是：

- 从 Flash 启动
- 从 UART/USB/SDIO 启动

下面以从 flash 启动为例，详细阐述芯片的完整启动流程，整体上分为 3 个步骤：

1. 一级引导程序被固化在了 BL616/8 芯片内部的 ROM 中，它会从 flash 的 0x0 偏移地址处读取 Firmware header(下面简称 FW Header)，根据 Header 参数完成芯片相关配置和检查，配置完成后将会从 0x2000 地址启动二级引导程序。
2. 二级引导程序从 flash 中读取分区表，根据分区表启动最新的主程序固件。
3. 应用程序启动开始运行，根据用户需要设置芯片相关时钟、引脚等配置，最后启动 RTOS 的调度器。

1.3.1 一级引导程序

芯片复位或者重新上电后，CPU 会立即开始运行，执行一系列的初始化操作，将芯片恢复到一个初始状态。初始化复位代码固化于 BL616/8 芯片的掩膜 ROM 处，不可被更改。

一级引导程序在启动过程中，会根据 HBN_RSV0 寄存器和 GPIO2 的电平状态来确定 BL616/8 的启动模式，HBN_RSV0 寄存器中保存了芯片唤醒的标志，具体执行流程如下：

1. 从深度睡眠模式复位：检查 HBN_RSV0 寄存器，判断是否请求从深度睡眠模式唤醒，如果是，则从深度睡眠模式唤醒，并进入快速启动流程。校验 HBN_RSV0 寄存器中存在的 flag 合法后，那么程序就会使用 OCRAM 起始地址存储的启动参数，根据入口地址信息立即跳转到该地址运行。如果校验的 HBN_RSV0 寄存器 flag 不合法，或出现启动异常，那么就会像正常上电复位流程一样继续启动。
2. 上电复位，软件 PWRON 复位：检查 GPIO2 引脚电平，判断是否从 Flash 启动镜像。如果 GPIO2 引脚电平是 1 (High)，则进入 UART/USB/SDIO 启动模式，该模式主要于 Flash 烧写或者下载镜像到 RAM 执行 (无线透传场景)。如果 GPIO2 引脚电平是 0 (Low)，则进入 Flash 启动模式，进入该模式后，一般运行二级引导程序。
3. 安全启动：根据 efuse 中的配置，开启安全启动流程，它会检查二级引导程序的完整性和签名的合法性，并判断是否需要开启 flash 解密功能。

1.3.2 二级引导程序

从 flash 的 0x0 偏移地址（包括 FW header）处开始存放的镜像就是二级引导程序。二级引导程序的源码可以在释放的 SDK 中 boot2_isp 目录下找到，二级引导程序通过分区表可以方便的规划 flash 分区信息，并且二级引导程序支持用户应用程序的安全引导，加密启动，签名验证，以及空中升级（OTA）等功能。

一级引导程序启动后，它会从镜像的开头读取二级引导程序 FW Header，根据 FW Header 完成芯片相关配置和检查，例如检查是否加密启动等等；配置好之后并跳转到二级引导程序运行。

二级引导程序启动后，默认从 flash 的 0xE000 偏移地址处 (可配置) 读取分区表信息。默认情况下采用 A/B 双分区表设计，应用固件实体位置也会存在 A/B 两个独立分区，这样的设计便于空中升级（OTA）。二级引导程序会根据分区表中的 active_index 和 age 信息来确定启动哪一个分区。具体参考分区表使用说明。

对于选择将要启动的分区，二级引导程序将会从 flash 读取 FW Header，检查应用程序的完整性和签名的合法性，并判断是否需要开启 flash 解密功能。最后跳转到应用程序执行。

1.3.3 应用程序启动阶段

应用程序启动包含从应用程序开始执行运行到用户 APP main 函数之前的所有过程。主要分为两个阶段：

- 系统电源、时钟、基本 C 语言运行环境初始化。
- 运行主函数并调用 main。

1.3.3.1 硬件和系统运行环境初始化

应用程序入口会从 `start.S` 文件开始中的 `_start` 执行。这里的初始化功能会初始化基本的 C 运行环境 (CRT)，并对芯片的内部硬件进行了初始配置。

- 为应用程序重新配置 CPU 异常向量。
- 初始化 FPU 单元。
- 设置堆栈指针。
- 初始化内部存储器 (`data` 和 `bss`)。
- 将 CPU 中断状态寄存器恢复到默认状态，并使能中断。
- 使能高速缓存。
- 如果配置了内存保护，则初始化内存保护。

这些运行完成之后，就会调用用户 `main` 函数。

1.3.3.2 运行主任务

在初始化完成必要的运行环境后，就会跳转到用户程序。通常情况下用户程序中的第一段将会调用 `board.c` 中的 `board_init` 函数，完成用户的项目专用端口初始化。可能包括 UART,GPIO 等各类外设的端口初始化。

基本硬件端口初始化完成后，用户可以选择运行裸机还是运行 FreeRTOS 操作系统，这取决于用户。如果使用无线功能必须使能 FreeRTOS，那么用户就需要为其不同的功能创建一个任务去运行。如果仅是裸机运行，那么通常情况下用户的功能都在一个 While 循环中运行。

1.3.4 从 UART/USB/SDIO 启动

从 UART/USB/SDIO 启动，一般统称为从 `interface` 模式启动（或者烧写模式）。该模式是一级引导程序启动后的一个分支情况。此模式该模式主要于 Flash 烧写或者下载镜像到 RAM 执行 (无线透传场景)。

系统支持从 Flash/UART/USB/SDIO 启动，各个启动模式说明如下：

表 1.1: 启动模式

启动引脚	电平	描述
GPIO2	1	从 UART(GPIO21/22)/USB/SDIO 启动，该模式主要用于 Flash 烧写或者下载镜像到 RAM 执行 (无线透传场景)
	0	从 Flash 启动应用镜像

1.4 地址映射

表 1.1: 内存地址映射

模块	大小	开始地址	
		Cache	Non-cache
OCRAM	320KB	0x62FC0000	0x22FC0000
WRAM	160KB	0x63010000	0x23010000

OCRAM 和 WRAM 既可以通过 AHB 总线访问，也可以通过 AXI 访问，当 CPU 使用 0x62FC0000 地址访问 OCRAM 时，会经过内部 Cache 并通过 AXI 转 AHB 实现对 OCRAM 的访问，当 CPU 使用 0x22FC0000 地址访问 OCRAM 时，不会经过内部 Cache 并且直接通过 AHB 总线访问 OCRAM。

表 1.2: 地址映射

模块	目标	开始地址	大小	描述
FLASH	Flash	0xA0000000	128MB	应用程序地址空间
PSRAM	pSRAM	0xA8000000	128MB	pSRAM 存储器地址空间 (可选项，依赖芯片具体型号，仅在 BL618 支持)
RAM	HBN RAM	0x20010000	4KB	HBN RAM, 主要用于超低功耗模式下的数据保存
Peripheral	USB	0x20072000	4KB	USB High Speed OTG 控制寄存器
	EMAC	0x20070000	4KB	EMAC 控制寄存器 (仅在 BL618 支持)
	SDH	0x20060000	4KB	SDH 控制寄存器
	MJPEG	0x20059000	4KB	MJPEG 图像编码控制寄存器 (仅在 BL618 支持)
	CAM	0x20057000	4KB	CAM 摄像头接口控制寄存器 (仅在 BL618 支持)
	Efuse	0x20056000	4KB	Efuse 存储控制寄存器
	AUDIO DAC	0x20055000	4KB	Audio DAC 控制寄存器
	PSRAM_Ctrl	0x20052000	4KB	PSRAM 控制寄存器 (仅在 BL618 支持)
	HBN	0x2000F000	4KB	深度睡眠控制 (休眠) 寄存器
	PDS	0x2000E000	4KB	睡眠控制 (掉电睡眠) 寄存器
	SDU	0x2000D000	4KB	SDU 控制寄存器
	DMA	0x2000C000	4KB	DMA 控制寄存器
	SF_Ctrl	0x2000B000	4KB	Serial Flash 控制寄存器
	Audio ADC	0x2000A000	256B	Audio ADC 控制寄存器
	I2S	0x2000AB00	256B	I2S 控制寄存器
	I2C1	0x2000A900	256B	I2C1 控制寄存器
	Display	0x2000A800	256B	Display 控制寄存器 (仅在 BL618 支持)
	IRR	0x2000A600	256B	IR Receiver 控制寄存器
	TIMER	0x2000A500	256B	TIMER 控制寄存器

表 1.2: 地址映射 (continued)

模块	目标	开始地址	大小	描述
	PWM	0x2000A400	256B	PWM 控制寄存器
	I2C0	0x2000A300	256B	I2C0 控制寄存器
	SPI	0x2000A200	256B	SPI 控制寄存器
	UART1	0x2000A100	256B	UART1 控制寄存器
	UART0	0x2000A000	256B	UART0 控制寄存器
	TZ	0x20005000	4KB	TrustZone 控制寄存器
	SEC_ENG	0x20004000	4KB	安全引擎控制寄存器
	GPIP	0x20002000	1KB	通用 DAC/ADC/ACOMP 接口控制寄存器
	GLB	0x20000000	4KB	全局控制寄存器
ROM	ROM	0x90000000	128KB	Bootrom 区域地址空间

1.5 中断源

BL616/BL618 一共包含 64 个中断源，中断源与对应的中断号如下表所示：

表 1.1: 中断分配

中断源		中断号	描述
BMX	BUS Error	IRQ_NUM_BASE+0	BUS Error Responses Interrupt
	BUS Timeout	IRQ_NUM_BASE+1	BUS Responses Timeout Interrupt
Display	Display	IRQ_NUM_BASE+2	Display All Interrupt
SDU	SDU Software Reset	IRQ_NUM_BASE+3	SDU Reset Triggered by Host
Audio ADC	Audio ADC	IRQ_NUM_BASE+4	Audio ADC Interrupt
RF	RF Interrupt0	IRQ_NUM_BASE+5	RF Interrupt0
	RF Interrupt1	IRQ_NUM_BASE+6	RF Interrupt1
SDU	SDU Side Interrupt	IRQ_NUM_BASE+7	SDU Side All Interrupt
Wi-Fi	TBTT SLEEP	IRQ_NUM_BASE+8	Wi-Fi TBTT SLEEP Interrupt
SecEng	Group1	IRQ_NUM_BASE+9	Group1 SHA/AES/TRNG/PKA/GMAC Interrupt
	Group0	IRQ_NUM_BASE+10	Group0 SHA/AES/TRNG/PKA/GMAC Interrupt
	Group1 CDET	IRQ_NUM_BASE+11	Group1 CDET Interrupt
	Group0 CDET	IRQ_NUM_BASE+12	Group0 CDET Interrupt
SF Ctrl	Group1	IRQ_NUM_BASE+13	SF_Ctrl Group1 Interrupt
	Group0	IRQ_NUM_BASE+14	SF_Ctrl Group0 Interrupt
DMA	DMA0_ALL	IRQ_NUM_BASE+15	DMA0 ALL Interrupt

表 1.1: 中断分配 (continued)

中断源		中断号	描述
CAM_OUT0	CAM_OUT0	IRQ_NUM_BASE+16	CAM_OUT0 Interrupt
SDH	SDH All Interrupt	IRQ_NUM_BASE+17	SDH All Interrupt
CAM_OUT1	CAM_OUT1	IRQ_NUM_BASE+18	CAM_OUT1 Interrupt
Wi-Fi	TBTT WAKEUP	IRQ_NUM_BASE+19	Wi-Fi TBTT WAKEUP Interrupt
IR	IRRX	IRQ_NUM_BASE+20	IR RX Interrupt
USB	USB	IRQ_NUM_BASE+21	USB Interrupt
Audio DAC	Audio DAC	IRQ_NUM_BASE+22	Audio DAC Interrupt
MJPEG	Encoder	IRQ_NUM_BASE+23	MJPEG Encoder All Interrupt
EMAC	EMAC	IRQ_NUM_BASE+24	EMAC Interrupt
GPADC	GPADC_DMA	IRQ_NUM_BASE+25	GPADC_DMA Interrupt
Efuse	Efuse	IRQ_NUM_BASE+26	Efuse Interrupt
SPI	SPI	IRQ_NUM_BASE+27	SPI Interrupt
UART	UART0	IRQ_NUM_BASE+28	UART0 Interrupt
	UART1	IRQ_NUM_BASE+29	UART1 Interrupt
GPIO	GPIO_DMA	IRQ_NUM_BASE+31	GPIO DMA Interrupt
I2C0	I2C0	IRQ_NUM_BASE+32	I2C0 Interrupt
PWM	PWM	IRQ_NUM_BASE+33	PWM Interrupt
TIMER0	TIMER0_CH0	IRQ_NUM_BASE+36	Timer0 Channel 0 Interrupt
	TIMER0_CH1	IRQ_NUM_BASE+37	Timer0 Channel 1 Interrupt
	TIMER0_WDT	IRQ_NUM_BASE+38	Timer0 Watch Dog Interrupt
I2C1	I2C1	IRQ_NUM_BASE+39	I2C1 Interrupt
I2S	I2S	IRQ_NUM_BASE+40	I2S Interrupt
	Reserved	IRQ_NUM_BASE+41	Reserved
	Reserved	IRQ_NUM_BASE+42	Reserved
XTAL	Xtal Ready	IRQ_NUM_BASE+43	Xtal Ready Interrupt
GPIO	GPIO_INT0	IRQ_NUM_BASE+44	GPIO Interrupt
DM	DM	IRQ_NUM_BASE+45	DM Interrupt
BT	BT	IRQ_NUM_BASE+46	BT Interrupt
MAC154	ENH Ack	IRQ_NUM_BASE+47	MAC154 ENH Ack Interrupt
	Others	IRQ_NUM_BASE+48	MAC154 Other Interrupt
	AES	IRQ_NUM_BASE+49	MAC154 AES Interrupt
PDS	PDS	IRQ_NUM_BASE+50	PDS Interrupt
HBN	HBN OUT0	IRQ_NUM_BASE+51	HBN Out 0 Interrupt

表 1.1: 中断分配 (continued)

中断源		中断号	描述
	HBN OUT1	IRQ_NUM_BASE+52	HBN Out 1 Interrupt
BOR	BOR	IRQ_NUM_BASE+53	Brown Out Reset Interrupt
Wi-Fi	Wi-Fi	IRQ_NUM_BASE+54	Wi-Fi Interrupt
BZ Phy	BZ Phy	IRQ_NUM_BASE+55	BZ Phy Interrupt
BLE	BLE	IRQ_NUM_BASE+56	BLE Interrupt
WiFi	MAC TR Timer	IRQ_NUM_BASE+57	MAC TX&RX Timer Interrupt
	MAC TR MISC	IRQ_NUM_BASE+58	MAC TX&RX Misc Interrupt
	MAC RX Trigger	IRQ_NUM_BASE+59	MAC RX Trigger Interrupt
	MAC TX Trigger	IRQ_NUM_BASE+60	MAC TX Trigger Interrupt
	MAC General	IRQ_NUM_BASE+61	MAC General Interrupt
	MAC Prot	IRQ_NUM_BASE+62	MAC Prot Interrupt
	IPC	IRQ_NUM_BASE+63	MAC IPC Interrupt

注解：其中 IRQ_NUM_BASE 为 16，中断号 0-15 为 RISC-V 保留中断。

1.6 外设概述

表 1.1: 外设列表

外设	数量	备注
GPIO	19/35	QFN40 对应 19 GPIOs, QFN56 对应 35 GPIOs
UART	2	支持 RTS/CTS
SPI	1	支持 Master/Slave 模式
I2C	2	支持 Master 模式
I2S	1	支持 Left-Justified/Right-Justified/Normal I2S/DSP 等数据格式
PWM	4	支持输出极性可调、双门限值设定
Timer	2	支持 FreeRun 模式和 PreLoad 模式
DMA	4	支持 LLI 链表功能
IR	1	支持接收，协议包括 NEC 和 RC-5，另外支持以脉冲宽度计数方式接收数据
Audio DAC	1	支持音频播放
Audio ADC	1	支持录音
EMAC	1	支持 100Mbps

表 1.1: 外设列表 (continued)

外设	数量	备注
CAM	1	支持图像矩形裁剪
MJPEG	1	支持任意量化表
DBI	1	支持 Type B/Type C 3-wire/Type C 4-wire , 另外还集成了 QSPI 模式
SDH	1	支持高速 SD 卡
SDU	1	支持 CCCR(function0) 与 function1, 支持 SDU 软复位, function1 有 16 个 port 接收缓冲区
SEC_ENG	1	支持 AES/SHA/GMAC/TRNG
USB	1	USB2.0

2.1 简介

芯片内的时钟源：XTAL，PLL，RC，搭配分频等配置送至各模块。

2.2 复位管理

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 2.1: 软件复位功能表

BL616	CHIP_EN	Watch Dog/ PDS MISC/ Software Power On (swrst_cfg2[0])	Software Reset (swrst_cfg1)	System Reset (swrst_cfg2[2])/ PDS SOC	CPU Reset (swrst_cfg2[1])/ PDS NP
CPU	✓	✓			✓
BUS	✓	✓		✓	
GLB	✓	✓	swrst_s1[0]		
MIX	✓	✓	swrst_s1[1]		
GPIP	✓	✓	swrst_s1[2]	✓	
SEC_ENG	✓	✓	swrst_s1[4]	✓	
TZ	✓	✓		✓	
Efuse	✓	✓			
DMA	✓	✓	swrst_s1[12]	✓	
SDU/USB	✓	✓	swrst_s1[13]		
MM_MISC	✓	✓	swrst_s1_ext[1]	✓	
pSRAM_ctrl	✓	✓	swrst_s1_ext[2]	✓	
USB	✓	✓	swrst_s1_ext[3]	✓	
AUDIO DAC	✓	✓	swrst_s1_ext[5]	✓	
SDH	✓	✓	swrst_s1_ext[6]	✓	
EMAC	✓	✓	swrst_s1_ext[7]	✓	
PDS	✓		swrst_s1[14]		
HBN	✓				

表 2.1: 软件复位功能表 (continued)

BL616	CHIP_EN	Watch Dog/ PDS MISC/ Software Power On (swrst_cfg2[0])	Software Reset (swrst_cfg1)	System Reset (swrst_cfg2[2])/ PDS SOC	CPU Reset (swrst_cfg2[1])/ PDS NP
AON	✓				
UART0	✓	✓	swrst_s1a[0]	✓	
UART1	✓	✓	swrst_s1a[1]	✓	
SPI	✓	✓	swrst_s1a[2]	✓	
I2C0	✓	✓	swrst_s1a[3]	✓	
PWM	✓	✓	swrst_s1a[4]	✓	
TIMER	✓	✓	swrst_s1a[5]	✓	
IRR	✓	✓	swrst_s1a[6]	✓	
CKS	✓	✓	swrst_s1a[7]	✓	
DBI	✓	✓	swrst_s1a[8]	✓	
I2C1	✓	✓	swrst_s1a[9]	✓	
I2S	✓	✓	swrst_s1a[11]	✓	
AUDIO ADC	✓	✓	swrst_s1a[12]	✓	
Wi-Fi	✓	✓	swrst_s2[0]		
BLE	✓	✓	swrst_s3[0][2]		

2.3 时钟源

时钟源包含：

- XTAL：外部晶振时钟，视系统需求频率可选 24、26、32、38.4、40MHz
- XTAL32K：外部晶振时钟，频率 32768 Hz
- RC32K：RC 振荡器时钟，频率 32768 Hz，提供校准
- RC32M：RC 振荡器时钟，频率 32MHz，提供校准
- PLL：多个 PLL 模块，可以产生若干不同频率的时钟，以满足各个应用场景

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，动态分频进行时钟配置。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、UART、I2C、SPI、PWM、IR-remote、GPADC、GPDAC 等。

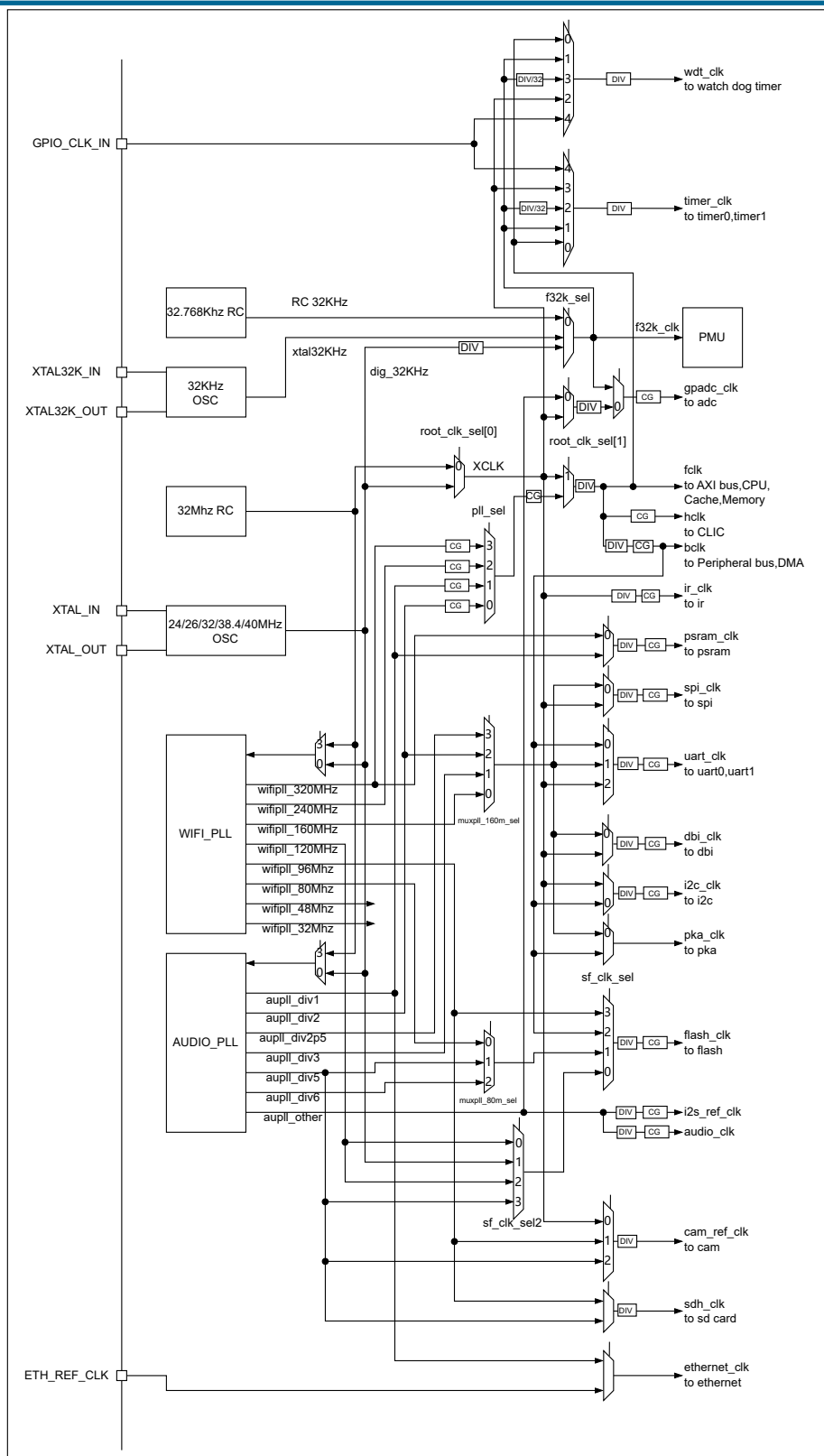


图 2.1: 时钟架构

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、power 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。详细设定可以参考系统时钟相关的章节。

3.2.2 复位管理

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。

详见复位管理。

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候是否产生中断，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

提供各块 memory 区域 size 管理：

- em 管理：WRAM 有一块空间可以作为 EM 使用（默认有 32KB 作为 EM），此功能可以分配指定 size 的空间作为 EM，其余空间仍旧为 WRAM。

提供各个内存模块在芯片系统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 LDO 管理

提供芯片内 LDO 管理：

- LDO 管理：可以对芯片内 LDO 输出电压进行一定调整，以降低功耗。

4.1 简介

GPIO(General Purpose I/O Ports) 是通用输入/输出端口，用户可将其与外部硬件设备连接达到控制外部硬件设备的目的。

4.2 主要特征

- 最多可达 35 个 I/O 引脚
- 每个 I/O 引脚可配置的功能多达 25 种
- 每个 I/O 引脚都可以配置为上拉、下拉或浮空模式
- 每个 I/O 引脚都可以配置为输入、输出或高阻态模式
- 每个 I/O 引脚的输出模式都有 4 种驱动能力可选择
- 每个 I/O 引脚的输入模式都可以设置是否开启施密特触发器
- 每个 I/O 引脚都支持 9 种外部中断模式
- FIFO 深度为 128 * 16-bit
- 可用 DMA 将数据从 RAM 搬至 I/O 引脚用于输出

4.3 GPIO 功能列表

GPIO_CFGxx（其中 xx 表示 GPIO 引脚号）寄存器的 reg_gpio_xx_func_sel 位用来设置 GPIO 的复用功能，复用功能编号如下表所示：

表 4.1: GPIO 功能列表

编号	功能
0	SDH
1	SPI0
2	FLASH
3	I2S0
4	PDM
5	I2C0
6	I2C1
7	UART0
8	EMAC
9	CAM
10	ANALOG
11	GPIO
12	SDIO
16	PWM0
17	JTAG
18	UART1
19	PWM1
20	SPI1
21	I2S1
22	DBI_B
23	DBI_C
24	QSPI
25	AUDAC
31	CLOCK_OUT

注解: 若 GPIO 设置为外设功能, 只需将 GPIO_CFGxx 寄存器的 reg_gpio_xx_func_sel 位设置为外设对应的编号即可。

4.4 GPIO 输入设置

通过设置 GPIO_CFGxx 寄存器，将通用的接口配置为输入模式，过程如下（其中 xx 表示 GPIO 引脚号）：

- 将 reg_gpio_xx_ie 设置为 1，使能 GPIO 输入模式
- 将 reg_gpio_xx_func_sel 设置为 11 即进入 SW-GPIO 模式
- 在 SW-GPIO 模式下，可通过 reg_gpio_xx_smt 设置是否使能施密特触发器，用于波形的整形
- 通过 reg_gpio_xx_pu 和 reg_gpio_xx_pd 设置是否使能内部上拉和下拉功能
- 如果需要中断则通过 reg_gpio_xx_int_mode_set 设定外部中断的类型，此时即可通过 reg_gpio_xx_i 读取 I/O 引脚的电平值

4.5 GPIO 输出设置

通过 GPIO_CFGxx 寄存器设置 GPIO 不同的输出模式，共支持以下四种模式。

4.5.1 普通输出模式

- 将 reg_gpio_xx_oe 设置为 1，使能 GPIO 输出模式
- 将 reg_gpio_xx_func_sel 设置为 11，进入 SW-GPIO 模式
- 将 reg_gpio_xx_mode 设置为 0，使能 I/O 的普通输出功能
- 可通过 reg_gpio_xx_pu 和 reg_gpio_xx_pd 设置是否使能内部上拉和下拉功能，此时即可通过 reg_gpio_xx_o 设置 I/O 引脚的电平值

4.5.2 Set/Clear 输出模式

- 将 reg_gpio_xx_oe 设置为 1，使能 GPIO 输出模式
- 将 reg_gpio_xx_func_sel 设置为 11，进入 SW-GPIO 模式
- 将 reg_gpio_xx_mode 设置为 1，使能 I/O 的 Set/Clear 输出功能
- 可通过 reg_gpio_xx_pu 和 reg_gpio_xx_pd 设置是否使能内部上拉和下拉功能

在 Set/Clear 输出模式下，可将 reg_gpio_xx_set 置 1，使对应 I/O 引脚设为高电平，将 reg_gpio_xx_clr 置 1 可将对应 I/O 引脚设为低电平，如果同时将 reg_gpio_xx_set 和 reg_gpio_xx_clr 置 1 则对应 I/O 引脚为高电平，对 reg_gpio_xx_set 和 reg_gpio_xx_clr 写 0 没有作用。

4.5.3 可编程输出模式

- 将 `<reg_gpio_xx_oe>` 设置为 1，使能 GPIO 输出模式
- 将 `<reg_gpio_xx_func_sel>` 设置为 11，进入 SW-GPIO 模式
- 将 `<reg_gpio_xx_mode>` 设置为 2，使能 I/O 的可编程输出功能
- 可通过 `<reg_gpio_xx_pu>` 和 `<reg_gpio_xx_pd>` 设置是否使能内部上拉和下拉功能。

在可编程输出模式下，当 GPIO_CFG142 寄存器的位 `<cr_gpio_tx_en>` 置 1 时，GPIO_CFG144 寄存器存储的数据，依照顺序逐个写入对应的 I/O 引脚，从而设置引脚的电平，缓冲区大小为 $128 * 16\text{bits}$ 。需要注意的是 GPIO_CFG144 寄存器储存的数据不是实际的引脚电平值，而是一种可编程的逻辑 0/1，逻辑 0/1 与实际电平之间的关系描述如下。

在此模式下输出到引脚的电平状态可以自由设置。以 XCLK 为时钟源，GPIO_CFG142 寄存器中 `<cr_code_total_time>` 设置的数值为一个周期：

逻辑 1 的电平状态：由 `<cr_code1_high_time>` 设置的一段高电平加上 `<cr_code_total_time>`-`<cr_code1_high_time>` 设置的一段低电平，当 `<cr_invert_code1_high>` 为 0 时，表示逻辑 1 先输出高电平再输出低电平，否则先输出低电平再输出高电平；

逻辑 0 的电平状态：由 `<cr_code0_high_time>` 设置一段高电平加上 `<cr_code_total_time>`-`<cr_code0_high_time>` 设置一段低电平，当 `<cr_invert_code0_high>` 为 0 时，表示逻辑 0 先输出高电平再输出低电平，否则先输出低电平再输出高电平。

注解：由于缓冲区的寄存器是 16 位宽，所以每 16 个引脚为一组，在一组中从低到高序号的引脚分别由缓冲区中对应的 bit 控制。此外，GPIO_CFG143 寄存器中的位 `<cr_gpio_dma_out_sel_latch>` 应该设置为 0。`<cr_gpio_dma_park_value>` 用于设置 I/O 默认电平，为 1 时默认电平是高电平，为 0 时默认电平是低电平。

当 `<cr_code_total_time> = 10`，`<cr_code0_high_time> = 1`，`<cr_code1_high_time> = 5`，`<cr_invert_code0_high> = 0`，`<cr_invert_code1_high> = 0`，`<cr_gpio_dma_park_value> = 0`，`<cr_gpio_dma_out_sel_latch> = 0` 时，对应的波形图如下所示：

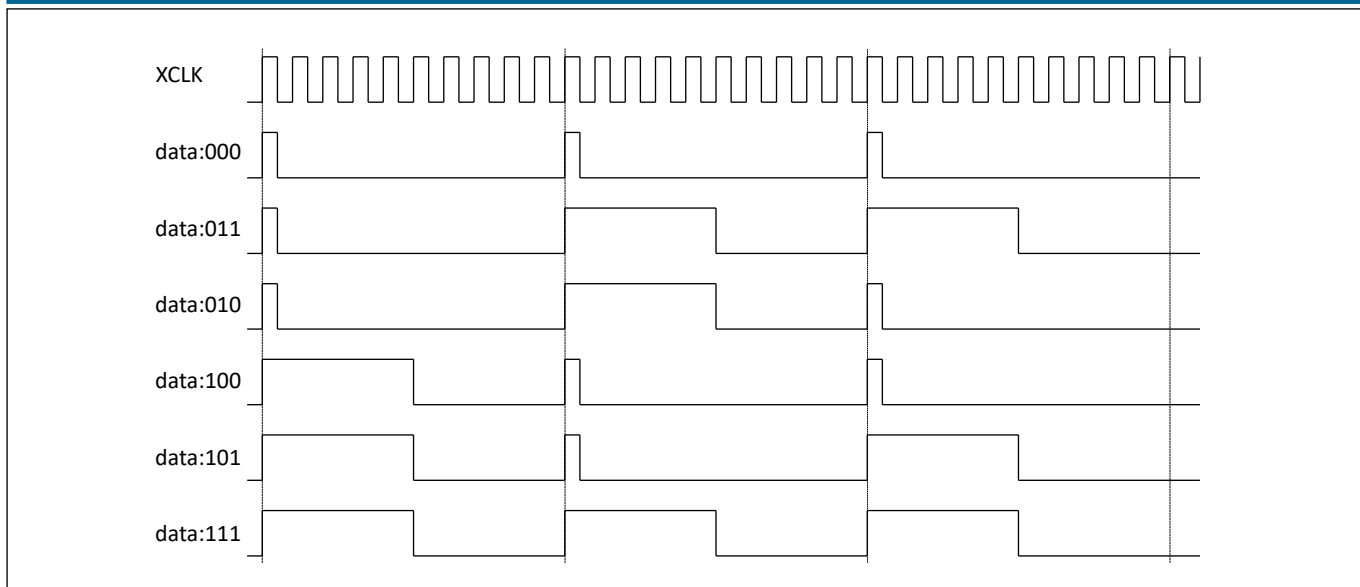


图 4.1: 普通 GPIO 输出波形

当 $\langle cr_code_total_time \rangle = 10$, $\langle cr_code0_high_time \rangle = 1$, $\langle cr_code1_high_time \rangle = 5$, $\langle cr_invert_code0_high \rangle = 0$, $\langle cr_invert_code1_high \rangle = 1$, $\langle cr_gpio_dma_park_value \rangle = 1$, $\langle cr_gpio_dma_out_sel_latch \rangle = 0$ 时, 对应的波形图如下所示:

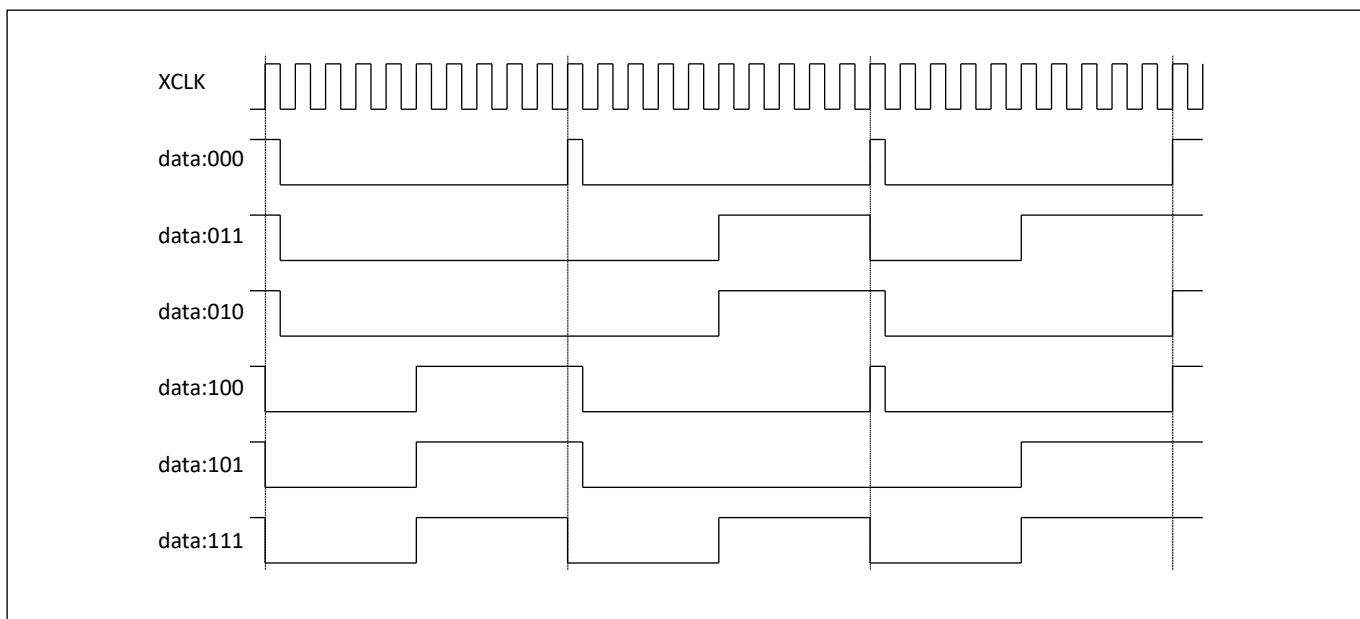
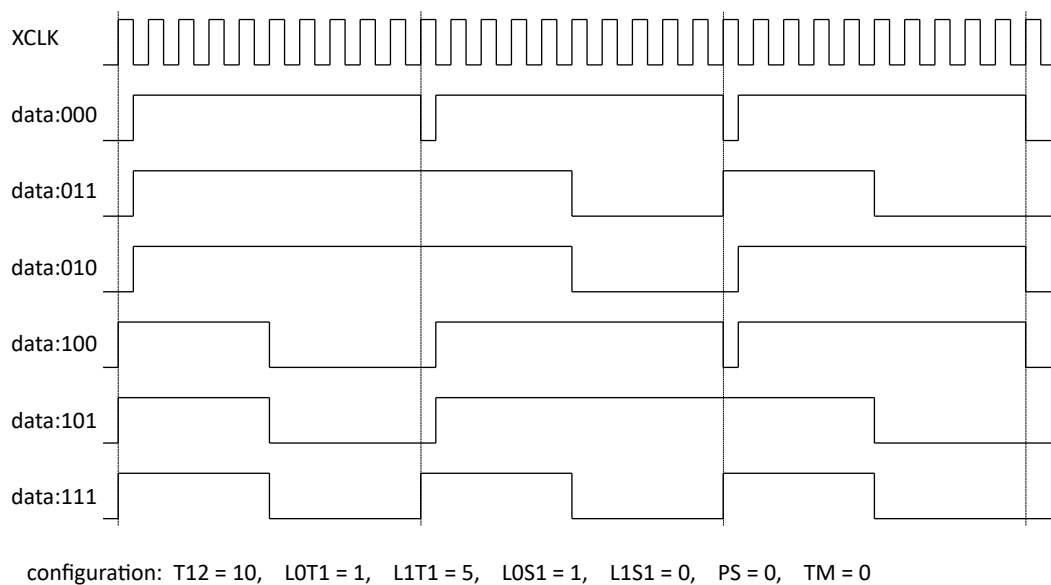


图 4.2: 默认电平为高电平, 逻辑 1 电平翻转输出波形

当 $\langle cr_code_total_time \rangle = 10$, $\langle cr_code0_high_time \rangle = 1$, $\langle cr_code1_high_time \rangle = 5$, $\langle cr_invert_code0_high \rangle = 1$, $\langle cr_invert_code1_high \rangle = 0$, $\langle cr_gpio_dma_park_value \rangle = 0$, $\langle cr_gpio_dma_out_sel_latch \rangle = 0$ 时, 对应的波形图如下所示:



Example 3 of GPIO output tx configuration

图 4.3: 默认电平为低电平，逻辑 0 电平翻转输出波形

当 $\langle cr_code_total_time \rangle = 10$, $\langle cr_code0_high_time \rangle = 1$, $\langle cr_code1_high_time \rangle = 5$, $\langle cr_invert_code0_high \rangle = 1$, $\langle cr_invert_code1_high \rangle = 1$, $\langle cr_gpio_dma_park_value \rangle = 1$, $\langle cr_gpio_dma_out_sel_latch \rangle = 0$ 时，对应的波形图如下所示：

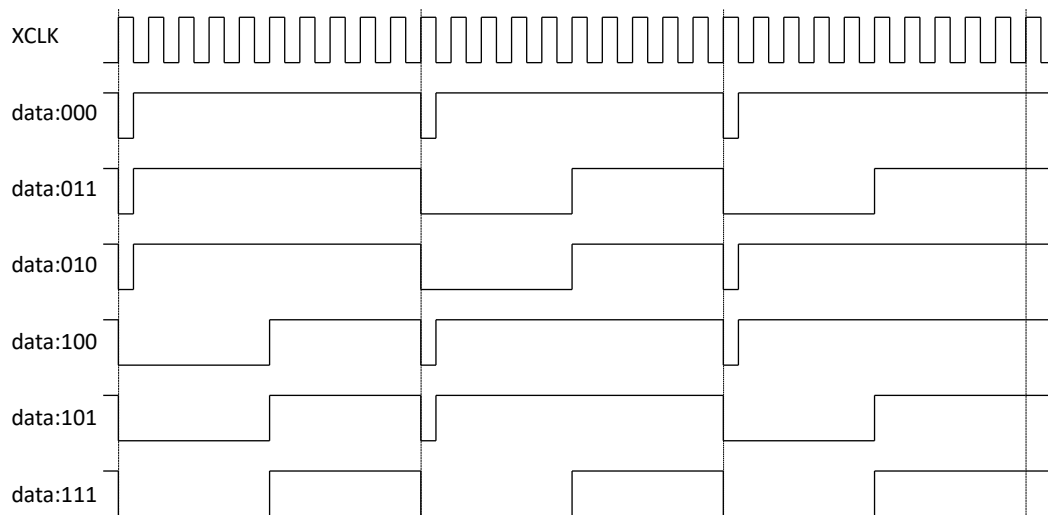


图 4.4: 默认电平为高电平，逻辑 0 电平翻转，逻辑 1 电平翻转输出波形

4.5.4 可编程 Set/Clear 输出模式

- 将 `<reg_gpio_xx_oe>` 设置为 1，使能 GPIO 输出模式
- 将 `<reg_gpio_xx_func_sel>` 设置为 11，进入 SW-GPIO 模式
- 并且将 `<reg_gpio_xx_mode>` 设置为 3，使能 I/O 的缓冲 Set/Clear 输出功能
- 可通过 `<reg_gpio_xx_pu>` 和 `<reg_gpio_xx_pd>` 设置是否使能内部上拉和下拉功能

此时当 GPIO_CFG142 寄存器的位 `<cr_gpio_tx_en>` 为 1 时，由 GPIO_CFG144 寄存器写入 FIFO 的数据将被按顺序逐个设置对应引脚的电平，缓冲区大小为 $128 * 16\text{bits}$ 。

在此模式下输出到引脚的电平状态可以自由设置。以 XCLK 为时钟源，GPIO_CFG142 寄存器中位 `<cr_code_total_time>` 设置的数值为一个周期：

每 8 个引脚为一组，GPIO_CFG144 寄存器低 8 位和高 8 位分别设置这 8 个引脚输出高/低电平。若低 8 位写 1，则对应的引脚输出高电平；若高 8 位写 1，则对应的引脚输出低电平，此时该寄存器中写 0 的位不生效，当高 8 位和低 8 位中对应位同时将同一个引脚写 1 时，该引脚电平为高电平。

此外，GPIO_CFG143 寄存器中的位 `<cr_gpio_dma_out_sel_latch>` 应该设置为 1。`<cr_gpio_dma_park_value>` 用于设置 I/O 默认电平，为 1 时默认电平是高电平，为 0 时默认电平是低电平。

4.6 I/O FIFO

I/O FIFO 的深度为 $128 * 16\text{-bit}$ ，GPIO_CFG143 寄存器中的位 `gpio_tx_fifo_cnt` 表示 FIFO 当前可用空间的大小，默认值是 128，每次向 GPIO_CFG144 寄存器写入一个数值后，`gpio_tx_fifo_cnt` 的值都会递减 1，如果减至 0 后继续向 GPIO_CFG144 寄存器写入数值，且 `cr_gpio_tx_fer_en` 为 1 即错误中断被使能，则会产生该中断。

当 GPIO_CFG142 寄存器的位 `cr_gpio_tx_en` 为 1 时，I/O FIFO 的数据被逐个发送到 I/O 引脚，此时 `gpio_tx_fifo_cnt` 的值会递增，当递增到大于 `cr_gpio_tx_fifo_th` 时，且 `cr_gpio_tx_fifo_en` 为 1 即 FIFO 中断被使能，则会产生该中断。

如果 CR_GPIO_CFG143 寄存器的位 `cr_gpio_dma_tx_en` 为 1，则使能 DMA 发送数据，此时如果 `cr_gpio_tx_fifo_th` 小于 `gpio_tx_fifo_cnt`，则 DMA 会将数据从设定好的 RAM 中搬运至缓冲区，此时中断标志 `r_gpio_tx_fifo_int` 自动被清除。

4.7 I/O 中断

I/O 支持多种外部中断，将 GPIO_CFGxx 寄存器的 `reg_gpio_xx_int_mask` 设置为 0，即可使能对应引脚的外部中断功能，`reg_gpio_xx_int_mode_set` 用于设置对应引脚的外部中断类型。支持的中断类型如下：

- 同步下降沿中断
 - 以 `f32k_clk` 时钟为基准，在每个时钟上升沿采样一次输入引脚电平，若出现一个高电平后紧跟两个低电平，则此时产生同步下降沿中断
- 同步上升沿中断
 - 以 `f32k_clk` 时钟为基准，在每个时钟上升沿采样一次输入引脚电平，若出现一个低电平后紧跟两个高电平，则

此时产生同步上升沿中断

- 同步低电平中断
 - 以 f32k_clk 时钟为基准，检测到低电平后，在第三个时钟上升沿处产生同步低电平中断
- 同步高电平中断
 - 以 f32k_clk 时钟为基准，检测到高电平后，在第三个时钟上升沿处产生同步高电平中断
- 同步双边沿中断
 - 以 f32k_clk 时钟为基准，若检测到高电平转换至低电平（低电平转换至高电平），会产生下降沿（上升沿）事件，事件产生后在第三个时钟上升沿处产生同步双边沿中断
- 异步下降沿中断
 - 当检测到高电平转换至低电平时，立即触发异步下降沿中断
- 异步上升沿中断
 - 当检测到低电平转换至高电平时，立即触发异步上升沿中断
- 异步低电平中断
 - 以 f32k_clk 时钟为基准，在每个时钟上升沿采样一次输入引脚电平，若出现连续 3 次都为低电平，则触发异步低电平中断
- 异步高电平中断
 - 以 f32k_clk 时钟为基准，在每个时钟上升沿采样一次输入引脚电平，若出现连续 3 次都为高电平，则触发异步高电平中断

在中断函数中可以通过 GPIO_CFGxx 寄存器 <gpio_xx_int_stat> 获取到产生中断的 GPIO 状态，同时可以通过 <reg_gpio_xx_int_clr> 清除对应的中断标志。

4.8 寄存器描述

名称	描述
dac_cfg0	DAC source control register
dac_cfg1	DAC CHA control register
dac_cfg2	DAC CHB control register
dac_cfg3	DAC converter data register
gpio_cfg0	GPIO0 config register
gpio_cfg1	GPIO1 config register

名称	描述
gpio_cfg2	GPIO2 config register
gpio_cfg3	GPIO3 config register
gpio_cfg4	GPIO4 config register
gpio_cfg5	GPIO5 config register
gpio_cfg6	GPIO6 config register
gpio_cfg7	GPIO7 config register
gpio_cfg8	GPIO8 config register
gpio_cfg9	GPIO9 config register
gpio_cfg10	GPIO10 config register
gpio_cfg11	GPIO11 config register
gpio_cfg12	GPIO12 config register
gpio_cfg13	GPIO13 config register
gpio_cfg14	GPIO14 config register
gpio_cfg15	GPIO15 config register
gpio_cfg16	GPIO16 config register
gpio_cfg17	GPIO17 config register
gpio_cfg18	GPIO18 config register
gpio_cfg19	GPIO19 config register
gpio_cfg20	GPIO20 config register
gpio_cfg21	GPIO21 config register
gpio_cfg22	GPIO22 config register
gpio_cfg23	GPIO23 config register
gpio_cfg24	GPIO24 config register
gpio_cfg25	GPIO25 config register
gpio_cfg26	GPIO26 config register
gpio_cfg27	GPIO27 config register
gpio_cfg28	GPIO28 config register
gpio_cfg29	GPIO29 config register
gpio_cfg30	GPIO30 config register

名称	描述
gpio_cfg31	GPIO31 config register
gpio_cfg32	GPIO32 config register
gpio_cfg33	GPIO33 config register
gpio_cfg34	GPIO34 config register
gpio_cfg128	GPIO input value register0
gpio_cfg129	GPIO input value register1
gpio_cfg136	GPIO output value register0
gpio_cfg137	GPIO output value register1
gpio_cfg138	GPIO set register0
gpio_cfg139	GPIO set register1
gpio_cfg141	GPIO clear register1
gpio_cfg142	GPIO TX config register0
gpio_cfg143	GPIO TX config register1
gpio_cfg144	GPIO TX fifo register

4.8.1 dac_cfg0

地址: 0x20000120

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
RSVD	gpdac_dat_chb_sel	gpdac_dat_cha_sel	gpdac_ana_clk_sel	gpdac_test_sel	gpdac_ref_sel	gpdac_test_en	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	gpdacb_rstn_ana	gpdaca_rstn_ana	

位	名称	权限	复位值	描述
31:15	RSVD			
14	gpdac_dat_chb_sel	r/w	1'h0	0:data from gpip, 1:data from audio pwm

位	名称	权限	复位值	描述
13	gpdac_dat_cha_sel	r/w	1'h0	0:data from gpip, 1:data from audio pwm
12	gpdac_ana_clk_sel	r/w	1'h0	0:clock from gpip, 1:clock from audio pwm
11:9	gpdac_test_sel	r/w	3'h0	select test point 0 7
8	gpdac_ref_sel	r/w	1'h0	Reference select 1'h0 Internal reference 1'h1 External reference
7	gpdac_test_en	r/w	1'h0	Test enable 1'h0 analog test disabled (ATEST is set in Hi-Z state) 1'h1 analog test point enabled to ATEST
6:2	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

4.8.2 dac_cfg1

地址：0x20000124

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	gpdac_ioa_en	gpdac_a_en

位	名称	权限	复位值	描述
31:20	RSVD			
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

4.8.3 dac_cfg2

地址：0x20000128

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	gpdac_b_en	gpdac_b_en

位	名称	权限	复位值	描述
31:20	RSVD			
19:18	gpdac_b_rng	r/w	2'h3	Output voltage range control with internal/external reference
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

4.8.4 dac_cfg3

地址：0x2000012c

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
RSVD	RSVD	RSVD	RSVD												

位	名称	权限	复位值	描述
31:28	RSVD			
27:16	gpdac_a_data	r/w	12'h0	Channel A Data input
15:12	RSVD			
11:0	gpdac_b_data	r/w	12'h0	Channel B Data input
-1:21	RSVD			
20	wifipll_en_rf_div3_hw	r	1'h1	
19:0	RSVD			

4.8.5 gpio_cfg0

地址：0x200008c4

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_0_mode															
RSVD															
reg_gpio_0_i															
RSVD															
reg_gpio_0_clr															
reg_gpio_0_set															
reg_gpio_0_o															
RSVD															
reg_gpio_0_int_mask															
gpio_0_int_stat															
reg_gpio_0_int_clr															
reg_gpio_0_int_mode_set															
reg_gpio_0_func_sel															
RSVD															
reg_gpio_0_oe															
reg_gpio_0_pd															
reg_gpio_0_pu															
reg_gpio_0_drv															
reg_gpio_0_smt															
reg_gpio_0_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_0_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Celar Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			

位	名称	权限	复位值	描述
28	reg_gpio_0_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_0_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_0_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_0_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_0_int_mask	r/w	1	mask interrupt (1)
21	gpio_0_int_stat	r	0	interrupt status
20	reg_gpio_0_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_0_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_0_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO)
7	RSVD			
6	reg_gpio_0_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_0_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_0_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up

位	名称	权限	复位值	描述
3:2	reg_gpio_0_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_0_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_0_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.6 gpio_cfg1

地址：0x200008c8

reg_gpio_1_mode RSVD reg_gpio_1_i RSVD reg_gpio_1_clr reg_gpio_1_set RSVD reg_gpio_1_int_mask gpio_1_int_stat reg_gpio_1_int_clr reg_gpio_1_int_mode_set																	
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_1_func_sel RSVD reg_gpio_1_oe reg_gpio_1_pd reg_gpio_1_pu reg_gpio_1_drv reg_gpio_1_smt reg_gpio_1_ie																	

位	名称	权限	复位值	描述
31:30	reg_gpio_1_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_1_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_1_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_1_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_1_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_1_int_mask	r/w	1	mask interrupt (1)
21	gpio_1_int_stat	r	0	interrupt status
20	reg_gpio_1_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_1_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_1_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_1_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_1_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_1_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_1_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_1_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_1_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.7 gpio_cfg2

地址：0x200008cc

reg_gpio_2_mode RSVD reg_gpio_2_i RSVD reg_gpio_2_clr reg_gpio_2_set reg_gpio_2_o RSVD reg_gpio_2_int_mask gpio_2_int_stat reg_gpio_2_int_clr reg_gpio_2_int_mode_set															
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_2_func_sel RSVD reg_gpio_2_oe reg_gpio_2_pd reg_gpio_2_pu reg_gpio_2_drv reg_gpio_2_smt reg_gpio_2_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_2_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_2_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_2_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_2_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_2_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_2_int_mask	r/w	1	mask interrupt (1)
21	gpio_2_int_stat	r	0	interrupt status
20	reg_gpio_2_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_2_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_2_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_2_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_2_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_2_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_2_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_2_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_2_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.8 gpio_cfg3

地址：0x200008d0

reg_gpio_3_mode RSVD reg_gpio_3_i RSVD reg_gpio_3_clr reg_gpio_3_set reg_gpio_3_o RSVD reg_gpio_3_int_mask gpio_3_int_stat reg_gpio_3_int_clr reg_gpio_3_int_mode_set															
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_3_func_sel RSVD reg_gpio_3_oe reg_gpio_3_pd reg_gpio_3_pu reg_gpio_3_drv reg_gpio_3_smt reg_gpio_3_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_3_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_3_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_3_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_3_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_3_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_3_int_mask	r/w	1	mask interrupt (1)
21	gpio_3_int_stat	r	0	interrupt status
20	reg_gpio_3_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_3_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_3_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_3_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_3_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_3_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_3_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_3_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_3_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.9 gpio_cfg4

地址：0x200008d4

reg_gpio_4_mode		RSVD		reg_gpio_4_i		RSVD		reg_gpio_4_clr		reg_gpio_4_set		RSVD		reg_gpio_4_int_mask		gpio_4_int_stat		reg_gpio_4_int_clr		reg_gpio_4_int_mode_set	
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_4_func_sel		RSVD		reg_gpio_4_oe		reg_gpio_4_pd		reg_gpio_4_pu		reg_gpio_4_drv		reg_gpio_4_smt		reg_gpio_4_ie	

位	名称	权限	复位值	描述
31:30	reg_gpio_4_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_4_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_4_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_4_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_4_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_4_int_mask	r/w	1	mask interrupt (1)
21	gpio_4_int_stat	r	0	interrupt status
20	reg_gpio_4_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_4_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_4_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_4_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_4_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_4_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_4_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_4_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_4_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.10 gpio_cfg5

地址：0x200008d8

reg_gpio_5_mode RSVD reg_gpio_5_i RSVD reg_gpio_5_clr reg_gpio_5_set reg_gpio_5_o RSVD reg_gpio_5_int_mask gpio_5_int_stat reg_gpio_5_int_clr reg_gpio_5_int_mode_set															
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_5_func_sel RSVD reg_gpio_5_oe reg_gpio_5_pd reg_gpio_5_pu reg_gpio_5_drv reg_gpio_5_smt reg_gpio_5_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_5_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_5_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_5_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_5_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_5_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_5_int_mask	r/w	1	mask interrupt (1)
21	gpio_5_int_stat	r	0	interrupt status
20	reg_gpio_5_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_5_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_5_func_sel	r/w	5'hB	GPIO Function Select (Default : SWGPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_5_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_5_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_5_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_5_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_5_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_5_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.11 gpio_cfg6

地址：0x200008dc

reg_gpio_6_mode RSVD reg_gpio_6_i RSVD reg_gpio_6_clr reg_gpio_6_set reg_gpio_6_o RSVD reg_gpio_6_int_mask gpio_6_int_stat reg_gpio_6_int_clr reg_gpio_6_int_mode_set															
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_6_func_sel RSVD reg_gpio_6_oe reg_gpio_6_pd reg_gpio_6_pu reg_gpio_6_drv reg_gpio_6_smt reg_gpio_6_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_6_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_6_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_6_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_6_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_6_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_6_int_mask	r/w	1	mask interrupt (1)
21	gpio_6_int_stat	r	0	interrupt status
20	reg_gpio_6_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_6_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_6_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_6_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_6_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_6_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_6_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_6_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_6_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.12 gpio_cfg7

地址: 0x200008e0

reg_gpio_7_mode RSVD reg_gpio_7_i RSVD reg_gpio_7_clr reg_gpio_7_set reg_gpio_7_o RSVD reg_gpio_7_int_mask gpio_7_int_stat reg_gpio_7_int_clr reg_gpio_7_int_mode_set															
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_7_func_sel RSVD reg_gpio_7_oe reg_gpio_7_pd reg_gpio_7_pu reg_gpio_7_drv reg_gpio_7_smt reg_gpio_7_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_7_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_7_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_7_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_7_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_7_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_7_int_mask	r/w	1	mask interrupt (1)
21	gpio_7_int_stat	r	0	interrupt status
20	reg_gpio_7_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_7_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_7_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_7_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_7_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_7_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_7_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_7_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_7_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.13 gpio_cfg8

地址：0x200008e4

reg_gpio_8_mode RSVD reg_gpio_8_i RSVD reg_gpio_8_clr reg_gpio_8_set reg_gpio_8_o RSVD reg_gpio_8_int_mask gpio_8_int_stat reg_gpio_8_int_clr reg_gpio_8_int_mode_set															
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_8_func_sel RSVD reg_gpio_8_oe reg_gpio_8_pd reg_gpio_8_pu reg_gpio_8_drv reg_gpio_8_smt reg_gpio_8_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_8_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_8_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_8_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_8_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_8_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_8_int_mask	r/w	1	mask interrupt (1)
21	gpio_8_int_stat	r	0	interrupt status
20	reg_gpio_8_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_8_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_8_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_8_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_8_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_8_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_8_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_8_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_8_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.14 gpio_cfg9

地址：0x200008e8

reg_gpio_9_mode RSVD reg_gpio_9_i RSVD reg_gpio_9_clr reg_gpio_9_set reg_gpio_9_o RSVD reg_gpio_9_int_mask gpio_9_int_stat reg_gpio_9_int_clr reg_gpio_9_int_mode_set															
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_9_func_sel RSVD reg_gpio_9_oe reg_gpio_9_pd reg_gpio_9_pu reg_gpio_9_drv reg_gpio_9_smt reg_gpio_9_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_9_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_9_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_9_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_9_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_9_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_9_int_mask	r/w	1	mask interrupt (1)
21	gpio_9_int_stat	r	0	interrupt status
20	reg_gpio_9_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_9_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_9_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_9_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_9_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_9_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_9_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_9_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_9_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.15 gpio_cfg10

地址: 0x200008ec

reg_gpio_10_mode															
RSVD															
reg_gpio_10_i															
RSVD															
reg_gpio_10_clr															
reg_gpio_10_set															
reg_gpio_10_o															
RSVD															
reg_gpio_10_int_mask															
gpio_10_int_stat															
reg_gpio_10_int_clr															
reg_gpio_10_int_mode_set															
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															
reg_gpio_10_func_sel															
RSVD															
reg_gpio_10_oe															
reg_gpio_10_pd															
reg_gpio_10_pu															
reg_gpio_10_drv															
reg_gpio_10_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_10_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_10_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_10_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_10_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_10_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_10_int_mask	r/w	1	mask interrupt (1)
21	gpio_10_int_stat	r	0	interrupt status
20	reg_gpio_10_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_10_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_10_func_sel	r/w	5'hF	GPIO Function Select (Default : CCI)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_10_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_10_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_10_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_10_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_10_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_10_ie	r/w	1	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.16 gpio_cfg11

地址：0x200008f0

reg_gpio_11_mode RSVD reg_gpio_11_i RSVD reg_gpio_11_clr reg_gpio_11_set reg_gpio_11_o RSVD reg_gpio_11_int_mask gpio_11_int_stat reg_gpio_11_int_clr reg_gpio_11_int_mode_set															
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_11_func_sel RSVD reg_gpio_11_oe reg_gpio_11_pd reg_gpio_11_pu reg_gpio_11_drv reg_gpio_11_smt reg_gpio_11_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_11_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_11_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_11_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_11_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_11_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_11_int_mask	r/w	1	mask interrupt (1)
21	gpio_11_int_stat	r	0	interrupt status
20	reg_gpio_11_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_11_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_11_func_sel	r/w	5'hF	GPIO Function Select (Default : CCI)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_11_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_11_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_11_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_11_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_11_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_11_ie	r/w	1	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.17 gpio_cfg12

地址: 0x200008f4

reg_gpio_12_mode RSVD reg_gpio_12_i RSVD reg_gpio_12_clr reg_gpio_12_set reg_gpio_12_o RSVD reg_gpio_12_int_mask gpio_12_int_stat reg_gpio_12_int_clr reg_gpio_12_int_mode_set															
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_12_func_sel RSVD reg_gpio_12_oe reg_gpio_12_pd reg_gpio_12_pu reg_gpio_12_drv reg_gpio_12_smt reg_gpio_12_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_12_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_12_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_12_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_12_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_12_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_12_int_mask	r/w	1	mask interrupt (1)
21	gpio_12_int_stat	r	0	interrupt status
20	reg_gpio_12_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_12_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_12_func_sel	r/w	5'hF	GPIO Function Select (Default : CCI)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_12_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_12_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_12_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_12_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_12_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_12_ie	r/w	1	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.18 gpio_cfg13

地址: 0x200008f8

reg_gpio_13_mode RSVD reg_gpio_13_i RSVD reg_gpio_13_clr reg_gpio_13_set reg_gpio_13_o RSVD reg_gpio_13_int_mask gpio_13_int_stat reg_gpio_13_int_clr reg_gpio_13_int_mode_set															
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_13_func_sel RSVD reg_gpio_13_oe reg_gpio_13_pd reg_gpio_13_pu reg_gpio_13_drv reg_gpio_13_smt reg_gpio_13_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_13_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_13_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_13_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_13_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_13_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_13_int_mask	r/w	1	mask interrupt (1)
21	gpio_13_int_stat	r	0	interrupt status
20	reg_gpio_13_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_13_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_13_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_13_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_13_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_13_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_13_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_13_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_13_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.19 gpio_cfg14

地址: 0x200008fc

reg_gpio_14_mode RSVD reg_gpio_14_i RSVD reg_gpio_14_clr reg_gpio_14_set reg_gpio_14_o RSVD reg_gpio_14_int_mask gpio_14_int_stat reg_gpio_14_int_clr reg_gpio_14_int_mode_set															
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_14_func_sel RSVD reg_gpio_14_oe reg_gpio_14_pd reg_gpio_14_pu reg_gpio_14_drv reg_gpio_14_smt reg_gpio_14_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_14_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_14_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_14_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_14_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_14_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_14_int_mask	r/w	1	mask interrupt (1)
21	gpio_14_int_stat	r	0	interrupt status
20	reg_gpio_14_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_14_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_14_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_14_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_14_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_14_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_14_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_14_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_14_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.20 gpio_cfg15

地址：0x20000900

reg_gpio_15_mode															
RSVD															
reg_gpio_15_i															
RSVD															
reg_gpio_15_clr															
reg_gpio_15_set															
reg_gpio_15_o															
RSVD															
reg_gpio_15_int_mask															
gpio_15_int_stat															
reg_gpio_15_int_clr															
reg_gpio_15_int_mode_set															
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															
reg_gpio_15_func_sel															
RSVD															
reg_gpio_15_oe															
reg_gpio_15_pd															
reg_gpio_15_pu															
reg_gpio_15_drv															
reg_gpio_15_smt															
reg_gpio_15_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_15_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_15_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_15_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_15_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_15_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_15_int_mask	r/w	1	mask interrupt (1)
21	gpio_15_int_stat	r	0	interrupt status
20	reg_gpio_15_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_15_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_15_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_15_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_15_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_15_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_15_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_15_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_15_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.21 gpio_cfg16

地址：0x20000904

reg_gpio_16_mode															
RSVD															
reg_gpio_16_i															
RSVD															
reg_gpio_16_clr															
reg_gpio_16_set															
reg_gpio_16_o															
RSVD															
reg_gpio_16_int_mask															
gpio_16_int_stat															
reg_gpio_16_int_clr															
reg_gpio_16_int_mode_set															
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															
reg_gpio_16_func_sel															
RSVD															
reg_gpio_16_oe															
reg_gpio_16_pd															
reg_gpio_16_pu															
reg_gpio_16_drv															
reg_gpio_16_ie															
reg_gpio_16_smt															

位	名称	权限	复位值	描述
31:30	reg_gpio_16_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_16_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_16_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_16_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_16_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_16_int_mask	r/w	1	mask interrupt (1)
21	gpio_16_int_stat	r	0	interrupt status
20	reg_gpio_16_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_16_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_16_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_16_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_16_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_16_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_16_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_16_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_16_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.22 gpio_cfg17

地址: 0x20000908

reg_gpio_17_mode RSVD reg_gpio_17_i RSVD reg_gpio_17_clr reg_gpio_17_set reg_gpio_17_o RSVD reg_gpio_17_int_mask gpio_17_int_stat reg_gpio_17_int_clr reg_gpio_17_int_mode_set															
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_17_func_sel RSVD reg_gpio_17_oe reg_gpio_17_pd reg_gpio_17_pu reg_gpio_17_drv reg_gpio_17_smt reg_gpio_17_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_17_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_17_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_17_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_17_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_17_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_17_int_mask	r/w	1	mask interrupt (1)
21	gpio_17_int_stat	r	0	interrupt status
20	reg_gpio_17_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_17_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_17_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_17_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_17_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_17_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_17_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_17_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_17_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.23 gpio_cfg18

地址：0x2000090c

reg_gpio_18_mode															
RSVD															
reg_gpio_18_i															
RSVD															
reg_gpio_18_clr															
reg_gpio_18_set															
reg_gpio_18_o															
RSVD															
reg_gpio_18_int_mask															
gpio_18_int_stat															
reg_gpio_18_int_clr															
reg_gpio_18_int_mode_set															
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															
reg_gpio_18_func_sel															
RSVD															
reg_gpio_18_oe															
reg_gpio_18_pd															
reg_gpio_18_pu															
reg_gpio_18_drv															
reg_gpio_18_ie															
reg_gpio_18_smt															

位	名称	权限	复位值	描述
31:30	reg_gpio_18_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_18_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_18_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_18_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_18_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_18_int_mask	r/w	1	mask interrupt (1)
21	gpio_18_int_stat	r	0	interrupt status
20	reg_gpio_18_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_18_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_18_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_18_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_18_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_18_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_18_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_18_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_18_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.24 gpio_cfg19

地址: 0x20000910

reg_gpio_19_mode RSVD reg_gpio_19_i RSVD reg_gpio_19_clr reg_gpio_19_set reg_gpio_19_o RSVD reg_gpio_19_int_mask gpio_19_int_stat reg_gpio_19_int_clr reg_gpio_19_int_mode_set															
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_19_func_sel RSVD reg_gpio_19_oe reg_gpio_19_pd reg_gpio_19_pu reg_gpio_19_drv reg_gpio_19_smt reg_gpio_19_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_19_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_19_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_19_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_19_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_19_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_19_int_mask	r/w	1	mask interrupt (1)
21	gpio_19_int_stat	r	0	interrupt status
20	reg_gpio_19_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_19_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_19_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_19_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_19_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_19_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_19_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_19_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_19_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.25 gpio_cfg20

地址：0x20000914

reg_gpio_20_mode RSVD reg_gpio_20_i RSVD reg_gpio_20_clr reg_gpio_20_set reg_gpio_20_o RSVD reg_gpio_20_int_mask gpio_20_int_stat reg_gpio_20_int_clr reg_gpio_20_int_mode_set															
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_20_func_sel RSVD reg_gpio_20_oe reg_gpio_20_pd reg_gpio_20_pu reg_gpio_20_drv reg_gpio_20_smt reg_gpio_20_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_20_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_20_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_20_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_20_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_20_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_20_int_mask	r/w	1	mask interrupt (1)
21	gpio_20_int_stat	r	0	interrupt status
20	reg_gpio_20_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_20_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_20_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_20_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_20_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_20_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_20_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_20_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_20_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.26 gpio_cfg21

地址：0x20000918

reg_gpio_21_mode RSVD reg_gpio_21_i RSVD reg_gpio_21_clr reg_gpio_21_set reg_gpio_21_o RSVD reg_gpio_21_int_mask gpio_21_int_stat reg_gpio_21_int_clr reg_gpio_21_int_mode_set															
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_21_func_sel RSVD reg_gpio_21_oe reg_gpio_21_pd reg_gpio_21_pu reg_gpio_21_drv reg_gpio_21_smt reg_gpio_21_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_21_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_21_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_21_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_21_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_21_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_21_int_mask	r/w	1	mask interrupt (1)
21	gpio_21_int_stat	r	0	interrupt status
20	reg_gpio_21_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_21_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_21_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_21_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_21_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_21_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_21_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_21_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_21_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.27 gpio_cfg22

地址: 0x2000091c

reg_gpio_22_mode															
RSVD															
reg_gpio_22_i															
RSVD															
reg_gpio_22_clr															
reg_gpio_22_set															
reg_gpio_22_o															
RSVD															
reg_gpio_22_int_mask															
gpio_22_int_stat															
reg_gpio_22_int_clr															
reg_gpio_22_int_mode_set															
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															
reg_gpio_22_func_sel															
RSVD															
reg_gpio_22_oe															
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
31:30	reg_gpio_22_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_22_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_22_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_22_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_22_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_22_int_mask	r/w	1	mask interrupt (1)
21	gpio_22_int_stat	r	0	interrupt status
20	reg_gpio_22_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_22_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_22_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_22_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_22_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_22_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_22_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_22_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_22_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.28 gpio_cfg23

地址：0x20000920

reg_gpio_23_mode															
RSVD															
reg_gpio_23_i															
RSVD															
reg_gpio_23_clr															
reg_gpio_23_set															
reg_gpio_23_o															
RSVD															
reg_gpio_23_int_mask															
gpio_23_int_stat															
reg_gpio_23_int_clr															
reg_gpio_23_int_mode_set															
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															
reg_gpio_23_func_sel															
RSVD															
reg_gpio_23_oe															
reg_gpio_23_pd															
reg_gpio_23_pu															
reg_gpio_23_drv															
reg_gpio_23_ie															
reg_gpio_23_smt															

位	名称	权限	复位值	描述
31:30	reg_gpio_23_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_23_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_23_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_23_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_23_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_23_int_mask	r/w	1	mask interrupt (1)
21	gpio_23_int_stat	r	0	interrupt status
20	reg_gpio_23_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_23_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_23_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_23_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_23_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_23_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_23_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_23_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_23_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.29 gpio_cfg24

地址：0x20000924

reg_gpio_24_mode															
RSVD															
reg_gpio_24_i															
RSVD															
reg_gpio_24_clr															
reg_gpio_24_set															
reg_gpio_24_o															
RSVD															
reg_gpio_24_int_mask															
gpio_24_int_stat															
reg_gpio_24_int_clr															
reg_gpio_24_int_mode_set															
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															
reg_gpio_24_func_sel															
RSVD															
reg_gpio_24_oe															
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
31:30	reg_gpio_24_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_24_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_24_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_24_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_24_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_24_int_mask	r/w	1	mask interrupt (1)
21	gpio_24_int_stat	r	0	interrupt status
20	reg_gpio_24_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_24_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_24_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_24_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_24_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_24_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_24_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_24_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_24_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.30 gpio_cfg25

地址：0x20000928

reg_gpio_25_mode															
RSVD															
reg_gpio_25_i															
RSVD															
reg_gpio_25_clr															
reg_gpio_25_set															
reg_gpio_25_o															
RSVD															
reg_gpio_25_int_mask															
gpio_25_int_stat															
reg_gpio_25_int_clr															
reg_gpio_25_int_mode_set															
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															
reg_gpio_25_func_sel															
RSVD															
reg_gpio_25_oe															
reg_gpio_25_pd															
reg_gpio_25_pu															
reg_gpio_25_drv															
reg_gpio_25_smt															
reg_gpio_25_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_25_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_25_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_25_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_25_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_25_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_25_int_mask	r/w	1	mask interrupt (1)
21	gpio_25_int_stat	r	0	interrupt status
20	reg_gpio_25_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_25_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_25_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_25_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_25_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_25_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_25_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_25_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_25_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.31 gpio_cfg26

地址: 0x2000092c

reg_gpio_26_mode RSVD reg_gpio_26_i RSVD reg_gpio_26_clr reg_gpio_26_set reg_gpio_26_o RSVD reg_gpio_26_int_mask gpio_26_int_stat reg_gpio_26_int_clr reg_gpio_26_int_mode_set															
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_26_func_sel RSVD reg_gpio_26_oe 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
31:30	reg_gpio_26_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_26_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_26_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_26_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_26_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_26_int_mask	r/w	1	mask interrupt (1)
21	gpio_26_int_stat	r	0	interrupt status
20	reg_gpio_26_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_26_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_26_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_26_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_26_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_26_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_26_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_26_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_26_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.32 gpio_cfg27

地址: 0x20000930

reg_gpio_27_mode															
RSVD															
reg_gpio_27_i															
RSVD															
reg_gpio_27_clr															
reg_gpio_27_set															
reg_gpio_27_o															
RSVD															
reg_gpio_27_int_mask															
gpio_27_int_stat															
reg_gpio_27_int_clr															
reg_gpio_27_int_mode_set															
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															
reg_gpio_27_func_sel															
RSVD															
reg_gpio_27_oe															
reg_gpio_27_pd															
reg_gpio_27_pu															
reg_gpio_27_drv															
reg_gpio_27_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_27_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_27_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_27_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_27_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_27_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_27_int_mask	r/w	1	mask interrupt (1)
21	gpio_27_int_stat	r	0	interrupt status
20	reg_gpio_27_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_27_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_27_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_27_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_27_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_27_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_27_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_27_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_27_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.33 gpio_cfg28

地址: 0x20000934

reg_gpio_28_mode																RSVD		reg_gpio_28_i				RSVD		reg_gpio_28_clr		reg_gpio_28_set		RSVD		reg_gpio_28_int_mask		gpio_28_int_stat		reg_gpio_28_int_clr		reg_gpio_28_int_mode_set	
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_28_func_sel								RSVD		reg_gpio_28_oe		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
31:30	reg_gpio_28_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_28_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_28_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_28_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_28_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_28_int_mask	r/w	1	mask interrupt (1)
21	gpio_28_int_stat	r	0	interrupt status
20	reg_gpio_28_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_28_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_28_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_28_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_28_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_28_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_28_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_28_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_28_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.34 gpio_cfg29

地址：0x20000938

reg_gpio_29_mode RSVD reg_gpio_29_i RSVD reg_gpio_29_clr reg_gpio_29_set reg_gpio_29_o RSVD reg_gpio_29_int_mask gpio_29_int_stat reg_gpio_29_int_clr reg_gpio_29_int_mode_set															
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 reg_gpio_29_oe reg_gpio_29_pd reg_gpio_29_pu reg_gpio_29_drv reg_gpio_29_smt reg_gpio_29_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_29_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_29_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_29_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_29_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_29_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_29_int_mask	r/w	1	mask interrupt (1)
21	gpio_29_int_stat	r	0	interrupt status
20	reg_gpio_29_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_29_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_29_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_29_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_29_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_29_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_29_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_29_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_29_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.35 gpio_cfg30

地址：0x2000093c

reg_gpio_30_mode RSVD reg_gpio_30_i RSVD reg_gpio_30_clr reg_gpio_30_set reg_gpio_30_o RSVD reg_gpio_30_int_mask gpio_30_int_stat reg_gpio_30_int_clr reg_gpio_30_int_mode_set															
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_30_func_sel RSVD reg_gpio_30_oe reg_gpio_30_pd reg_gpio_30_pu reg_gpio_30_drv reg_gpio_30_smt reg_gpio_30_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_30_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_30_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_30_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_30_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_30_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_30_int_mask	r/w	1	mask interrupt (1)
21	gpio_30_int_stat	r	0	interrupt status
20	reg_gpio_30_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_30_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_30_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_30_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_30_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_30_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_30_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_30_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_30_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.36 gpio_cfg31

地址：0x20000940

reg_gpio_31_mode															
RSVD															
reg_gpio_31_i															
RSVD															
reg_gpio_31_clr															
reg_gpio_31_set															
reg_gpio_31_o															
RSVD															
reg_gpio_31_int_mask															
gpio_31_int_stat															
reg_gpio_31_int_clr															
reg_gpio_31_int_mode_set															
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															
reg_gpio_31_func_sel															
RSVD															
reg_gpio_31_oe															
reg_gpio_31_pd															
reg_gpio_31_pu															
reg_gpio_31_drv															
reg_gpio_31_smt															
reg_gpio_31_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_31_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_31_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_31_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_31_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_31_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_31_int_mask	r/w	1	mask interrupt (1)
21	gpio_31_int_stat	r	0	interrupt status
20	reg_gpio_31_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_31_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_31_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_31_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_31_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_31_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_31_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_31_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_31_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.37 gpio_cfg32

地址：0x20000944

reg_gpio_32_mode RSVD reg_gpio_32_i RSVD reg_gpio_32_clr reg_gpio_32_set reg_gpio_32_o RSVD reg_gpio_32_int_mask gpio_32_int_stat reg_gpio_32_int_clr reg_gpio_32_int_mode_set															
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_32_func_sel RSVD reg_gpio_32_oe reg_gpio_32_pd reg_gpio_32_pu reg_gpio_32_drv reg_gpio_32_smt reg_gpio_32_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_32_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_32_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_32_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_32_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_32_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_32_int_mask	r/w	1	mask interrupt (1)
21	gpio_32_int_stat	r	0	interrupt status
20	reg_gpio_32_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_32_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_32_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_32_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_32_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_32_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_32_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_32_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_32_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.38 gpio_cfg33

地址：0x20000948

reg_gpio_33_mode RSVD reg_gpio_33_i RSVD reg_gpio_33_clr reg_gpio_33_set reg_gpio_33_o RSVD reg_gpio_33_int_mask gpio_33_int_stat reg_gpio_33_int_clr reg_gpio_33_int_mode_set															
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_33_func_sel RSVD reg_gpio_33_oe reg_gpio_33_pd reg_gpio_33_pu reg_gpio_33_drv reg_gpio_33_smt reg_gpio_33_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_33_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_33_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_33_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_33_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_33_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_33_int_mask	r/w	1	mask interrupt (1)
21	gpio_33_int_stat	r	0	interrupt status
20	reg_gpio_33_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_33_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_33_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_33_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_33_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_33_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_33_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_33_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_33_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.39 gpio_cfg34

地址：0x2000094c

reg_gpio_34_mode															
RSVD															
reg_gpio_34_i															
RSVD															
reg_gpio_34_clr															
reg_gpio_34_set															
reg_gpio_34_o															
RSVD															
reg_gpio_34_int_mask															
gpio_34_int_stat															
reg_gpio_34_int_clr															
reg_gpio_34_int_mode_set															
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															
reg_gpio_34_func_sel															
RSVD															
reg_gpio_34_oe															
reg_gpio_34_pd															
reg_gpio_34_pu															
reg_gpio_34_drv															
reg_gpio_34_smt															
reg_gpio_34_ie															

位	名称	权限	复位值	描述
31:30	reg_gpio_34_mode	r/w	0	When GPIO Function Selected to SWGPIO 00 (Output Value Mode): GPIO Output by reg_gpio_x_o Value 01 (Set/Clear Mode) :GPIO Output set by reg_gpio_x_set and clear by reg_gpio_x_clr 10 : SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Output value by gpio_dma_o 11: SWGPIO Source comes from GPIO DMA (GPIO DMA Mode), GPIO Outout value by gpio_dma_set/gpio_dma_clr
29	RSVD			
28	reg_gpio_34_i	r	0	GPIO input state
27	RSVD			
26	reg_gpio_34_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg_gpio_34_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg_gpio_34_o	r/w	0	When SWGPIO @ Output Value Mode 0 : set GPIO output Value to 0 1 : set GPIO output Value to 1
23	RSVD			
22	reg_gpio_34_int_mask	r/w	1	mask interrupt (1)
21	gpio_34_int_stat	r	0	interrupt status
20	reg_gpio_34_int_clr	r/w	0	clear interrupt
19:16	reg_gpio_34_int_mode_set	r/w	0	0000 : sync falling edge trigger 0001 : sync rising edge trigger 0010 : sync low level trigger 0011 : sync high level trigger 01xx : sync rising & falling edge trigger 1000 : async falling edge trigger 1001 : async rising edge trigger 1010 : async low level trigger 1011 : async high level trigger
15:13	RSVD			
12:8	reg_gpio_34_func_sel	r/w	5'hB	GPIO Function Select (Default : SW-GPIO)
7	RSVD			

位	名称	权限	复位值	描述
6	reg_gpio_34_oe	r/w	0	Register Controlled GPIO Output Enable (Used when GPIO Function select to Register Control GPIO) 0: disable GPIO output 1: enable GPIO output
5	reg_gpio_34_pd	r/w	0	GPIO Pull Down Control 0: not pull down 1: pull down
4	reg_gpio_34_pu	r/w	0	GPIO Pull Up Control 0: not pull up 1: pull up
3:2	reg_gpio_34_drv	r/w	0	GPIO Driving Control 0: driver level 0 1: driver level 1 2: driver level 2 3: driver level 3
1	reg_gpio_34_smt	r/w	1	GPIO SMT Control 0: disable schmitt trigger 1: enable schmitt trigger
0	reg_gpio_34_ie	r/w	0	GPIO Input Enable 0: disable GPIO input 1: enable GPIO input

4.8.40 gpio_cfg128

地址: 0x20000ac4

reg2_gpio_31_i	reg2_gpio_30_i	reg2_gpio_29_i	reg2_gpio_28_i	reg2_gpio_27_i	reg2_gpio_26_i	reg2_gpio_25_i	reg2_gpio_24_i	reg2_gpio_23_i	reg2_gpio_22_i	reg2_gpio_21_i	reg2_gpio_20_i	reg2_gpio_19_i	reg2_gpio_18_i	reg2_gpio_17_i	reg2_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
reg2_gpio_15_i	reg2_gpio_14_i	reg2_gpio_13_i	reg2_gpio_12_i	reg2_gpio_11_i	reg2_gpio_10_i	reg2_gpio_9_i	reg2_gpio_8_i	reg2_gpio_7_i	reg2_gpio_6_i	reg2_gpio_5_i	reg2_gpio_4_i	reg2_gpio_3_i	reg2_gpio_2_i	reg2_gpio_1_i	reg2_gpio_0_i

位	名称	权限	复位值	描述
31	reg2_gpio_31_i	r	0	Register Controlled GPIO Input value
30	reg2_gpio_30_i	r	0	Register Controlled GPIO Input value
29	reg2_gpio_29_i	r	0	Register Controlled GPIO Input value
28	reg2_gpio_28_i	r	0	Register Controlled GPIO Input value
27	reg2_gpio_27_i	r	0	Register Controlled GPIO Input value
26	reg2_gpio_26_i	r	0	Register Controlled GPIO Input value
25	reg2_gpio_25_i	r	0	Register Controlled GPIO Input value
24	reg2_gpio_24_i	r	0	Register Controlled GPIO Input value
23	reg2_gpio_23_i	r	0	Register Controlled GPIO Input value
22	reg2_gpio_22_i	r	0	Register Controlled GPIO Input value
21	reg2_gpio_21_i	r	0	Register Controlled GPIO Input value
20	reg2_gpio_20_i	r	0	Register Controlled GPIO Input value
19	reg2_gpio_19_i	r	0	Register Controlled GPIO Input value
18	reg2_gpio_18_i	r	0	Register Controlled GPIO Input value
17	reg2_gpio_17_i	r	0	Register Controlled GPIO Input value
16	reg2_gpio_16_i	r	0	Register Controlled GPIO Input value
15	reg2_gpio_15_i	r	0	Register Controlled GPIO Input value
14	reg2_gpio_14_i	r	0	Register Controlled GPIO Input value
13	reg2_gpio_13_i	r	0	Register Controlled GPIO Input value
12	reg2_gpio_12_i	r	0	Register Controlled GPIO Input value
11	reg2_gpio_11_i	r	0	Register Controlled GPIO Input value
10	reg2_gpio_10_i	r	0	Register Controlled GPIO Input value
9	reg2_gpio_9_i	r	0	Register Controlled GPIO Input value
8	reg2_gpio_8_i	r	0	Register Controlled GPIO Input value
7	reg2_gpio_7_i	r	0	Register Controlled GPIO Input value
6	reg2_gpio_6_i	r	0	Register Controlled GPIO Input value
5	reg2_gpio_5_i	r	0	Register Controlled GPIO Input value
4	reg2_gpio_4_i	r	0	Register Controlled GPIO Input value
3	reg2_gpio_3_i	r	0	Register Controlled GPIO Input value
2	reg2_gpio_2_i	r	0	Register Controlled GPIO Input value
1	reg2_gpio_1_i	r	0	Register Controlled GPIO Input value

位	名称	权限	复位值	描述
0	reg2_gpio_0_i	r	0	Register Controlled GPIO Input value

4.8.41 gpio_cfg129

地址: 0x20000ac8

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 RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD

RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD

reg2_gpio_32_i reg2_gpio_33_i reg2_gpio_34_i

位	名称	权限	复位值	描述
31:3	RSVD			
2	reg2_gpio_34_i	r	0	Register Controlled GPIO Input value
1	reg2_gpio_33_i	r	0	Register Controlled GPIO Input value
0	reg2_gpio_32_i	r	0	Register Controlled GPIO Input value

4.8.42 gpio_cfg136

地址: 0x20000ae4

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

reg2_gpio_31_o reg2_gpio_30_o reg2_gpio_29_o reg2_gpio_28_o reg2_gpio_27_o reg2_gpio_26_o reg2_gpio_25_o reg2_gpio_24_o reg2_gpio_23_o reg2_gpio_22_o reg2_gpio_21_o reg2_gpio_20_o reg2_gpio_19_o reg2_gpio_18_o reg2_gpio_17_o reg2_gpio_16_o

reg2_gpio_15_o reg2_gpio_14_o reg2_gpio_13_o reg2_gpio_12_o reg2_gpio_11_o reg2_gpio_10_o reg2_gpio_9_o reg2_gpio_8_o reg2_gpio_7_o reg2_gpio_6_o reg2_gpio_5_o reg2_gpio_4_o reg2_gpio_3_o reg2_gpio_2_o reg2_gpio_1_o reg2_gpio_0_o

位	名称	权限	复位值	描述
31	reg2_gpio_31_o	r/w	0	Register Controlled GPIO Output Value
30	reg2_gpio_30_o	r/w	0	Register Controlled GPIO Output Value
29	reg2_gpio_29_o	r/w	0	Register Controlled GPIO Output Value
28	reg2_gpio_28_o	r/w	0	Register Controlled GPIO Output Value
27	reg2_gpio_27_o	r/w	0	Register Controlled GPIO Output Value
26	reg2_gpio_26_o	r/w	0	Register Controlled GPIO Output Value
25	reg2_gpio_25_o	r/w	0	Register Controlled GPIO Output Value
24	reg2_gpio_24_o	r/w	0	Register Controlled GPIO Output Value
23	reg2_gpio_23_o	r/w	0	Register Controlled GPIO Output Value
22	reg2_gpio_22_o	r/w	0	Register Controlled GPIO Output Value
21	reg2_gpio_21_o	r/w	0	Register Controlled GPIO Output Value
20	reg2_gpio_20_o	r/w	0	Register Controlled GPIO Output Value
19	reg2_gpio_19_o	r/w	0	Register Controlled GPIO Output Value
18	reg2_gpio_18_o	r/w	0	Register Controlled GPIO Output Value
17	reg2_gpio_17_o	r/w	0	Register Controlled GPIO Output Value
16	reg2_gpio_16_o	r/w	0	Register Controlled GPIO Output Value
15	reg2_gpio_15_o	r/w	0	Register Controlled GPIO Output Value
14	reg2_gpio_14_o	r/w	0	Register Controlled GPIO Output Value
13	reg2_gpio_13_o	r/w	0	Register Controlled GPIO Output Value
12	reg2_gpio_12_o	r/w	0	Register Controlled GPIO Output Value
11	reg2_gpio_11_o	r/w	0	Register Controlled GPIO Output Value
10	reg2_gpio_10_o	r/w	0	Register Controlled GPIO Output Value
9	reg2_gpio_9_o	r/w	0	Register Controlled GPIO Output Value
8	reg2_gpio_8_o	r/w	0	Register Controlled GPIO Output Value
7	reg2_gpio_7_o	r/w	0	Register Controlled GPIO Output Value
6	reg2_gpio_6_o	r/w	0	Register Controlled GPIO Output Value
5	reg2_gpio_5_o	r/w	0	Register Controlled GPIO Output Value
4	reg2_gpio_4_o	r/w	0	Register Controlled GPIO Output Value
3	reg2_gpio_3_o	r/w	0	Register Controlled GPIO Output Value
2	reg2_gpio_2_o	r/w	0	Register Controlled GPIO Output Value
1	reg2_gpio_1_o	r/w	0	Register Controlled GPIO Output Value

位	名称	权限	复位值	描述
0	reg2_gpio_0_o	r/w	0	Register Controlled GPIO Output Value

4.8.43 gpio_cfg137

地址: 0x20000ae8

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 RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD

RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD

reg2_gpio_32_o reg2_gpio_33_o reg2_gpio_34_o

位	名称	权限	复位值	描述
31:3	RSVD			
2	reg2_gpio_34_o	r/w	0	Register Controlled GPIO Output Value
1	reg2_gpio_33_o	r/w	0	Register Controlled GPIO Output Value
0	reg2_gpio_32_o	r/w	0	Register Controlled GPIO Output Value

4.8.44 gpio_cfg138

地址: 0x20000aec

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

reg2_gpio_31_set reg2_gpio_30_set reg2_gpio_29_set reg2_gpio_28_set reg2_gpio_27_set reg2_gpio_26_set reg2_gpio_25_set reg2_gpio_24_set reg2_gpio_23_set reg2_gpio_22_set reg2_gpio_21_set reg2_gpio_20_set reg2_gpio_19_set reg2_gpio_18_set reg2_gpio_17_set reg2_gpio_16_set

reg2_gpio_15_set reg2_gpio_14_set reg2_gpio_13_set reg2_gpio_12_set reg2_gpio_11_set reg2_gpio_10_set reg2_gpio_9_set reg2_gpio_8_set reg2_gpio_7_set reg2_gpio_6_set reg2_gpio_5_set reg2_gpio_4_set reg2_gpio_3_set reg2_gpio_2_set reg2_gpio_1_set reg2_gpio_0_set

位	名称	权限	复位值	描述
31	reg2_gpio_31_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
30	reg2_gpio_30_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
29	reg2_gpio_29_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
28	reg2_gpio_28_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
27	reg2_gpio_27_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
26	reg2_gpio_26_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
25	reg2_gpio_25_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
24	reg2_gpio_24_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
23	reg2_gpio_23_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
22	reg2_gpio_22_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
21	reg2_gpio_21_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
20	reg2_gpio_20_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect

位	名称	权限	复位值	描述
19	reg2_gpio_19_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
18	reg2_gpio_18_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
17	reg2_gpio_17_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
16	reg2_gpio_16_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
15	reg2_gpio_15_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
14	reg2_gpio_14_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
13	reg2_gpio_13_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
12	reg2_gpio_12_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
11	reg2_gpio_11_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
10	reg2_gpio_10_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
9	reg2_gpio_9_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
8	reg2_gpio_8_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect

位	名称	权限	复位值	描述
7	reg2_gpio_7_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
6	reg2_gpio_6_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
5	reg2_gpio_5_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
4	reg2_gpio_4_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
3	reg2_gpio_3_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
2	reg2_gpio_2_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
1	reg2_gpio_1_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
0	reg2_gpio_0_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect

4.8.45 gpio_cfg139

地址: 0x20000af0

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	reg2_gpio_32_set	reg2_gpio_33_set
														reg2_gpio_34_set	

位	名称	权限	复位值	描述
31:3	RSVD			
2	reg2_gpio_34_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
1	reg2_gpio_33_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
0	reg2_gpio_32_set	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will set GPIO output value to 1,when set/clr at the same time, only set take effect
-1:32	RSVD			
31	reg2_gpio_31_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
30	reg2_gpio_30_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
29	reg2_gpio_29_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
28	reg2_gpio_28_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
27	reg2_gpio_27_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
26	reg2_gpio_26_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
25	reg2_gpio_25_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
24	reg2_gpio_24_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
23	reg2_gpio_23_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect

位	名称	权限	复位值	描述
22	reg2_gpio_22_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
21	reg2_gpio_21_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
20	reg2_gpio_20_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
19	reg2_gpio_19_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
18	reg2_gpio_18_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
17	reg2_gpio_17_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
16	reg2_gpio_16_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
15	reg2_gpio_15_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
14	reg2_gpio_14_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
13	reg2_gpio_13_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
12	reg2_gpio_12_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
11	reg2_gpio_11_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect

位	名称	权限	复位值	描述
10	reg2_gpio_10_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
9	reg2_gpio_9_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
8	reg2_gpio_8_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
7	reg2_gpio_7_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
6	reg2_gpio_6_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
5	reg2_gpio_5_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
4	reg2_gpio_4_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
3	reg2_gpio_3_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
2	reg2_gpio_2_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
1	reg2_gpio_1_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
0	reg2_gpio_0_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect

4.8.46 gpio_cfg141

地址：0x20000af8

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	reg2_gpio_34_clr	reg2_gpio_33_clr	reg2_gpio_32_clr

位	名称	权限	复位值	描述
31:3	RSVD			
2	reg2_gpio_34_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
1	reg2_gpio_33_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect
0	reg2_gpio_32_clr	w1p	0	When SWGPIO @ Set/Clear Mode Set this bit will clear GPIO output value to 0,when set/clr at the same time, only set take effect

4.8.47 gpio_cfg142

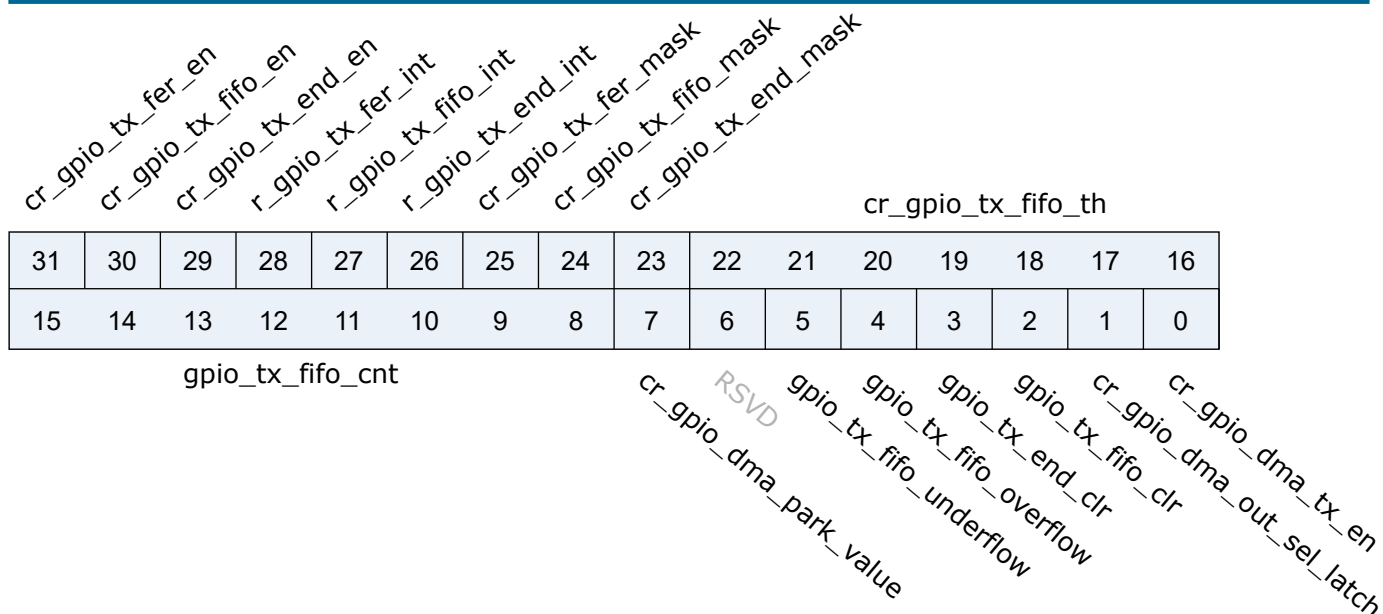
地址：0x20000afc

cr_code1_high_time								cr_code0_high_time							
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_code_total_time								RSVD	RSVD	RSVD	RSVD	cr_invert_code1_high	cr_invert_code0_high	cr_gpio_tx_en	

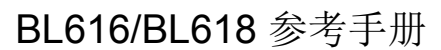
位	名称	权限	复位值	描述
31:24	cr_code1_high_time	r/w	8'd200	Used to generate Code 1 Duty Cycle Waveform (in units of xclk (XTAL or RC32M) clock cycle) waveform keep high during cr_code1_high_time waveform keep low during cr_code1_low_time (cr_code_total_time - cr_code1_high_time)
23:16	cr_code0_high_time	r/w	8'd200	Used to generate Code 0 Duty Cycle Waveform (in units of xclk (XTAL or RC32M) clock cycle) waveform keep high during cr_code0_high_time waveform keep low during cr_code0_low_time (cr_code_total_time - cr_code0_high_time)
15:7	cr_code_total_time	r/w	9'd400	Used to define Code0/Code1 total waveform time
6:3	RSVD			
2	cr_invert_code1_high	r/w	1'b0	code1 phase 0: first output high level and then output low level 1: first output low level and then output high level
1	cr_invert_code0_high	r/w	1'b0	code0 phase 0: first output high level and then output low level 1: first output low level and then output high level
0	cr_gpio_tx_en	r/w	1'b0	Enable GPIO DMA OUT/GPIO DMA Latch 0: disable GPIO TX function 1: enable GPIO TX function

4.8.48 gpio_cfg143

地址: 0x20000b00



位	名称	权限	复位值	描述
31	cr_gpio_tx_fer_en	r/w	1'b1	Interrupt enable of gpio_tx_fer_int (GPIO DMA FIFO Underflow or Overflow)
30	cr_gpio_tx_fifo_en	r/w	1'b1	Interrupt enable of gpio_tx_fifo_int
29	cr_gpio_tx_end_en	r/w	1'b1	Interrupt enable of gpio_tx_end_int
28	r_gpio_tx_fer_int	r	1'b0	GPIO TX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
27	r_gpio_tx_fifo_int	r	1'b0	GPIO TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed
26	r_gpio_tx_end_int	r	1'b0	GPIO TX END Interrupt (GPIO DMA FIFO Empty)
25	cr_gpio_tx_fer_mask	r/w	1'b1	Interrupt mask of gpio_tx_fer_int
24	cr_gpio_tx_fifo_mask	r/w	1'b1	Interrupt mask of gpio_tx_fifo_int
23	cr_gpio_tx_end_mask	r/w	1'b1	Interrupt mask of gpio_tx_end_int
22:16	cr_gpio_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	gpio_tx_fifo_cnt	r	8'd128	TX FIFO available count
7	cr_gpio_dma_park_value	r/w	1'b0	0: park at low level when TX FIFO empty 1: park at high level when TX FIFO empty
6	RSVD			
5	gpio_tx_fifo_underflow	r	1'b0	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	gpio_tx_fifo_overflow	r	1'b0	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	gpio_tx_end_clr	w1c	1'b0	Interrupt clear of gpio_tx_end_int



4.8.49 gpio_cfg144

RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD

gpio_tx_data_to_fifo

BL616/BL618 参考手册

5.1 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 可以工作在 4 种模式下，转换结果为 12/14/16bits 左对齐模式。ADC 拥有深度为 32 字节的 FIFO，支持多种中断，支持 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。

5.2 主要特点

- 高性能
 - 可以选择 12-bit, 14-bit, 16-bit 转换结果输出
 - 单通道连续转换模式最高采样率可达 2M
 - 其它转换模式最高采样率 500K
 - 支持 2.0V, 3.2V 可选参考电压
 - 支持 DMA 将转换结果搬运到内存
 - 支持单通道单次转换、单通道连续转换、多通道单次转换、多通道连续转换
 - 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

5.3 功能描述

ADC 模块基本框图如图所示。

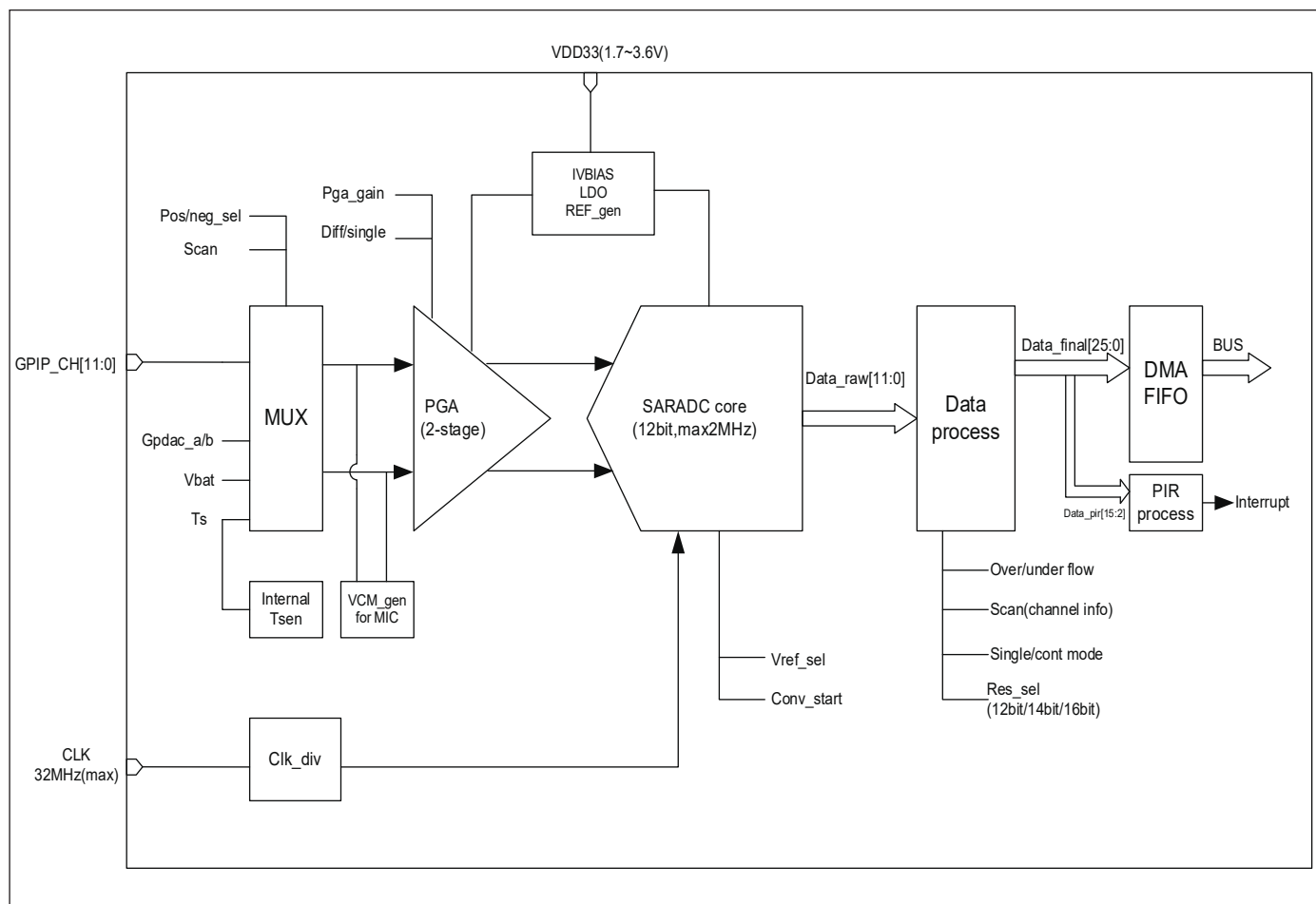


图 5.1: ADC 基本框图

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，包含外部模拟信号和内部模拟信号。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（直流或者交流）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，得到模拟信号到数字信号的转换。转换的结果为 12/14/16 bit，数据处理模块将转换的结果进一步处理，包括添加通道信息等。最后将得到的数据推送到 FIFO 中。

5.3.1 ADC 引脚和内部信号

表 5.1: 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
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 5.2: 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	模拟模块供电地
ADC_CHX	Input	模拟输入引脚，总共 12 路

5.3.2 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：

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

- ADC CH11
- DAC OUTA
- DAC OUTB
- VBAT/2
- TSEN
- VREF
- GND

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` 置为 1。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。

5.3.3 ADC 时钟

ADC 模块的工作时钟来源如下图所示。

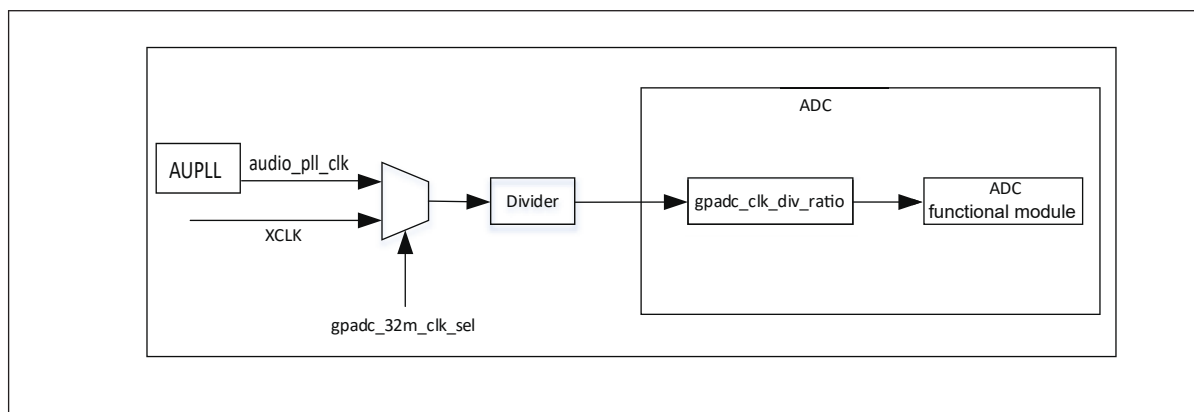


图 5.2: ADC 时钟

ADC 的时钟源可以选择来自 AUPLL 分频的 `audio_pll_clk`，XTAL 或者内部 RC32M，时钟源的选择在 GLB 模块中设定，同时 GLB 模块也提供了时钟分频。例如：ADC 的时钟源选择 XTAL，时钟分频是 1，到达 ADC 模块的时钟是 20M。在 ADC 模块内部，提供了一个时钟分频，如果选择 20 分频，则 ADC 模块内部的时钟则是 1 M。用户可以根据实际采样需求，自行调整 ADC 的时钟源和各个分频系数。`gpadc_32m_clk_div` 分频寄存器宽度为 6-bit，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 3-bit，其分频值定义如下：

- 3'b000: div=1
- 3'b001: div=4
- 3'b010: div=8
- 3'b011: div=12

- 3'b100: div=16
- 3'b101: div=20
- 3'b110: div=24
- 3'b111: div=32

5.3.4 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_scan_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_scan_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。

5.3.5 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 12-bit，无符号位，在差分模式下，最高位为符号位，剩下 11-bit 为转换的结果。

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

表 5.3: 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，其可以设置的值及采样次数如下 (以采样时钟 2M 为例，非单通道连续转换模式需降低时钟)：

- 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，最高位是符号位，即，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，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，所以用户一般是从 ADC FIFO 获取转换结果，ADC FIFO 的数据格式与 `gpadc_data_out` 寄存器中数据格式相同。

5.3.6 ADC 中断

- ADC 转换完成中断
 - 当 ADC 转换完成并将结果存入 FIFO 时，通过 `gpadc_rdy_mask` 设置中断开关，选择是否触发 ADC 转换完成中断。
- ADC 正（负）极采样超量程中断
 - 当 ADC 在正极采样超量程和负极采样超量程时，通过 `gpadc_pos_satur_mask`，`gpadc_neg_satur_mask` 设置中断开关，选择是否触发中断，当中断产生时，可通过 `gpadc_pos_satur` 和 `gpadc_neg_satur` 寄存器查询中断状态，通过设置 `gpadc_pos_satur_clr` 和 `gpadc_neg_satur_clr` 来清除中断。该功能可以用来判断输入电压是否异常。

5.3.7 ADC FIFO

ADC 模块拥有深度为 32 字节的 FIFO，数据宽度为 26-bit，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 满状态
- FIFO 非空状态
- FIFO Overrun 中断
- FIFO Underrun 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 用户可以通过三种模式获取数据：查询模式，中断模式，DMA 模式

查询模式

CPU 轮询 `gpadc_rdy` 位，当该控制位置位时，说明 FIFO 中存在有效数据，CPU 可以根据 `gpadc_fifo_data_count` 获知 FIFO 数据个数并从 FIFO 读出这些数据。

中断模式

CPU 设置 `gpadc_rdy_mask` 为 0，ADC 就会在 FIFO 有数据推入的时候产生中断，用户可在中断函数中，根据 `gpadc_fifo_data_count` 获知 FIFO 数据个数并从 FIFO 读出这些数据，然后设置 `gpadc_rdy_clr` 清除中断。

DMA 模式

用户设定 `gpadc_dma_en` 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 `gpadc_fifo_thl` 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。

5.3.8 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 `gpadc_clk_div_ratio`，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 `gpadc_pos_sel` 和 `gpadc_neg_sel` 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 `gpadc_scan_length`、`gpadc_reg_scn_posX` 和 `gpadc_reg_scn_negX`。

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 `gpadc_res_sel` 选择数据转换结果的精度，设置 `gpadc_global_en=1`，`gpadc_conv_start=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 `gpadc_conv_start` 设置为 0，再设置为 1，以便再次触发转换。

5.3.9 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 2.0V，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 `gpadc_vbat_en` 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。

5.3.10 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，即 **Temperature Sensor**，简称 **TSEN**。

TSEN 的测试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 ΔV 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 $\Delta(\text{ADC_out})=7.753T+X$ ，当知道了电压值，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，用户需要确定 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 $\Delta(\text{ADC_out})$ ，从而得到 X 。在用户使用的时候，只要根据公式 $T=[\Delta(\text{ADC_out})-X]/7.753$ ，就可以得到温度 T 。

在使用 TSEN 时，建议把 ADC 设置成 16-bit 模式，通过多次采样以减少误差，并且参考电压只能选择 2.0V，设置 `gpadc_ts_en` 为 1 以便启动 TSEN 功能。

如果选择内部二极管，`gpadc_tsext_sel=0`，如果选择外部二极管，`gpadc_tsext_sel=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。

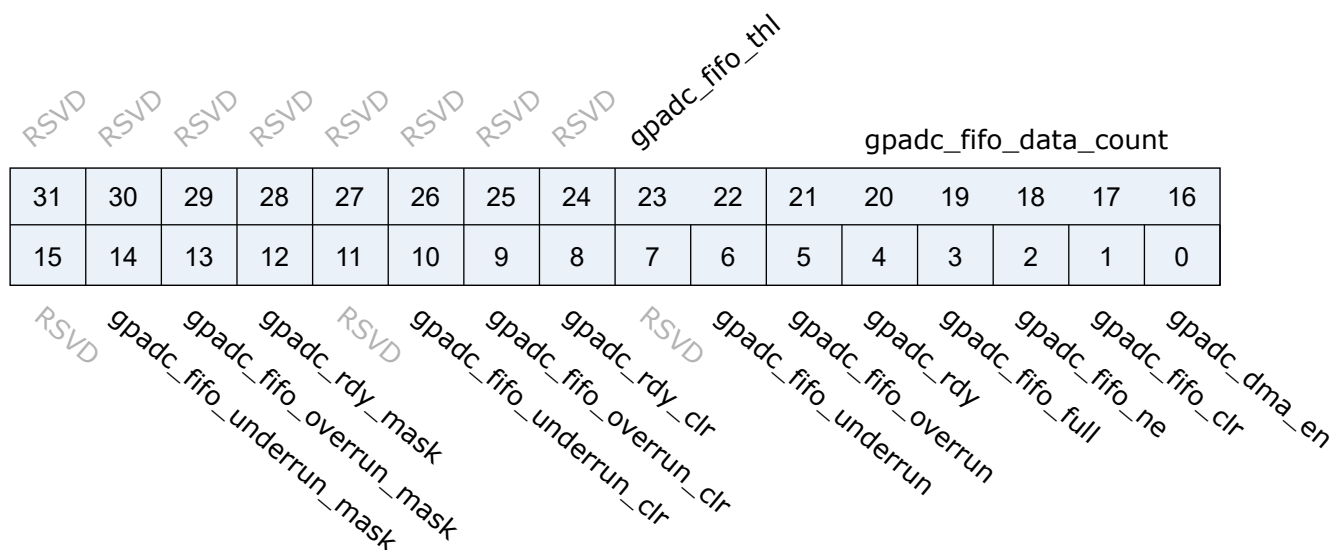
在上述设定完毕后，设置 `gpadc_tsvbe_low=0`，启动测量，得到测量结果 V_0 ，再设置 `gpadc_tsvbe_low=1`，启动测量，得到测量结果 V_1 ， $\Delta(\text{ADC_out})=V_1-V_0$ ，根据公式 $T=[\Delta(\text{ADC_out})-X]/7.753$ ，得到温度 T 。

5.4 寄存器描述

名称	描述
<code>gpadc_config</code>	gpadc fifo config
<code>gpadc_dma_rdata</code>	gpadc fifo
<code>gpadc_reg_cmd</code>	gpadc config0
<code>gpadc_reg_config1</code>	gpadc config1
<code>gpadc_reg_config2</code>	gpadc config2
<code>gpadc_reg_scn_pos1</code>	gpadc positive conervation sequence 1
<code>gpadc_reg_scn_pos2</code>	gpadc positive conervation sequence 2
<code>gpadc_reg_scn_neg1</code>	gpadc negative conervation sequence 1
<code>gpadc_reg_scn_neg2</code>	gpadc negative conervation sequence 2
<code>gpadc_reg_status</code>	gpadc ready status
<code>gpadc_reg_isr</code>	gpadc saturation interrupt config

5.4.1 gpadc_config

地址：0x20002000



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	RSVD			
14	gpadc_fifo_underrun_mask	r/w	1'b0	write 1 mask
13	gpadc_fifo_underrun_mask	r/w	1'b0	write 1 mask
12	gpadc_fifo_underrun_mask	r/w	1'b0	write 1 mask
11	RSVD			
10	gpadc_fifo_underrun_clr	r/w	1'b0	Write 1 to clear flag
9	gpadc_fifo_underrun_clr	r/w	1'b0	Write 1 to clear flag
8	gpadc_fifo_underrun_clr	r/w	1'b0	Write 1 to clear flag
7	RSVD			
6	gpadc_fifo_underrun	r	1'b0	FIFO underrun interrupt flag
5	gpadc_fifo_underrun	r	1'b0	FIFO underrun interrupt flag
4	gpadc_fifo_underrun	r	1'b0	Conversion data ready interrupt flag

位	名称	权限	复位值	描述
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: 0x20002004

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 gpadc_reg_cmd

地址: 0x2000f90c

gpadc_reg_cmd															
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_pos_sel: gpadc_pos_sel, gpadc_neg_sel, gpadc_neg_gnd, gpadc_micpga_en, gpadc_micbias_en
 gpadc_neg_sel: gpadc_neg_sel, gpadc_neg_gnd, gpadc_micpga_en, gpadc_micbias_en
 gpadc_global_en: gpadc_global_en, gpadc_conv_start, gpadc_soft_rst
 gpadc_conv_start: gpadc_conv_start, gpadc_soft_rst
 gpadc_soft_rst: gpadc_soft_rst

位	名称	权限	复位值	描述
31	gpadc_sen_test_en	r/w	1'b0	enable sensor dc test mux
30:28	gpadc_sen_sel	r/w	3'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
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	gpadc_rcal_en	r/w	1'b0	rcal enable 1'b0: rcal disable 1'b1: rcal enable
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

位	名称	权限	复位值	描述
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
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 dca 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 negative 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 dacc 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 conervation 1'h1: start conervation
0	gpadc_global_en	r/w	1'b0	1'h0: disable ADC 1'h1: enable ADC

5.4.4 gpadc_reg_config1

地址: 0x2000f910

RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
gpadc_dither_en															
gpadc_scan_en															
gpadc_scan_length															
gpadc_clk_div_ratio															
gpadc_clk_ana_inv															
gpadc_clk_ana_dly_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
gpadc_clk_ana_dly															
gpadc_pwm_trg_en															
gpadc_lowv_det_en															
gpadc_vcm_hyst_sel															
gpadc_vcm_sel_en															
RSVD															
RSVD															
RSVD															
gpadc_res_sel															
gpadc_cont_conv_en															
gpadc_cal_os_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	select scan mode length 4'b0000 : select gpadc_scan_pos_0 and gpadc_scan_neg_0 4'b0001 : select gpadc_scan_pos_1 and gpadc_scan_neg_1 4'b0010 : select gpadc_scan_pos_2 and gpadc_scan_neg_2 4'b0011 : select gpadc_scan_pos_3 and gpadc_scan_neg_3 4'b0100 : select gpadc_scan_pos_4 and gpadc_scan_neg_4 4'b0101 : select gpadc_scan_pos_5 and gpadc_scan_neg_5 4'b0110 : select gpadc_scan_pos_6 and gpadc_scan_neg_6 4'b0111 : select gpadc_scan_pos_7 and gpadc_scan_neg_7 4'b1000 : select gpadc_scan_pos_8 and gpadc_scan_neg_8 4'b1001 : select gpadc_scan_pos_9 and gpadc_scan_neg_9 4'b1010 : select gpadc_scan_pos_10 and gpadc_scan_neg_10 4'b1011 : select gpadc_scan_pos_11 and gpadc_scan_neg_11
20:18	gpadc_clk_div_ratio	r/w	3'h3	analog 32M clock division ratio 3'b000: div=1 3'b001: div=4 3'b010: div=8 3'b011: div=12 3'b100: div=16 3'b101: div=20 3'b110: div=24 3'b111: div=32
17	gpadc_clk_ana_inv	r/w	1'b0	analog clock 2M inverted
16	gpadc_clk_ana_dly_en	r/w	1'b0	analog clock 2M delay enable
15:12	gpadc_clk_ana_dly	r/w	4'd0	analog clock 2M delay cycle count
11	gpadc_pwm_trg_en	r/w	1'b0	Enable signal for PWM to trigger ADC sampling
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 hystersis select when vcm_sel_en is enabled

位	名称	权限	复位值	描述
8	gpadc_vcm_sel_en	r/w	1'b0	pga vcm selected when lowv_det_en is enable
7:5	RSVD			
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 mode16KS/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

5.4.5 gpadc_reg_config2

地址: 0x2000f914

gpadc_tsvbe_low				gpadc_dly_sel				gpadc_pga1_gain				gpadc_pga2_gain				gpadc_test_sel				gpadc_test_en				gpadc_bias_sel				gpadc_chop_mode																			
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_chop_mode				gpadc_pga_vcmi_en				gpadc_pga_en				gpadc_pga_os_cal				gpadc_pga_vcm				RSVD				gpadc_ts_en				gpadc_vbat_en				gpadc_vref_sel				gpadc_diff_mode				RSVD				RSVD			

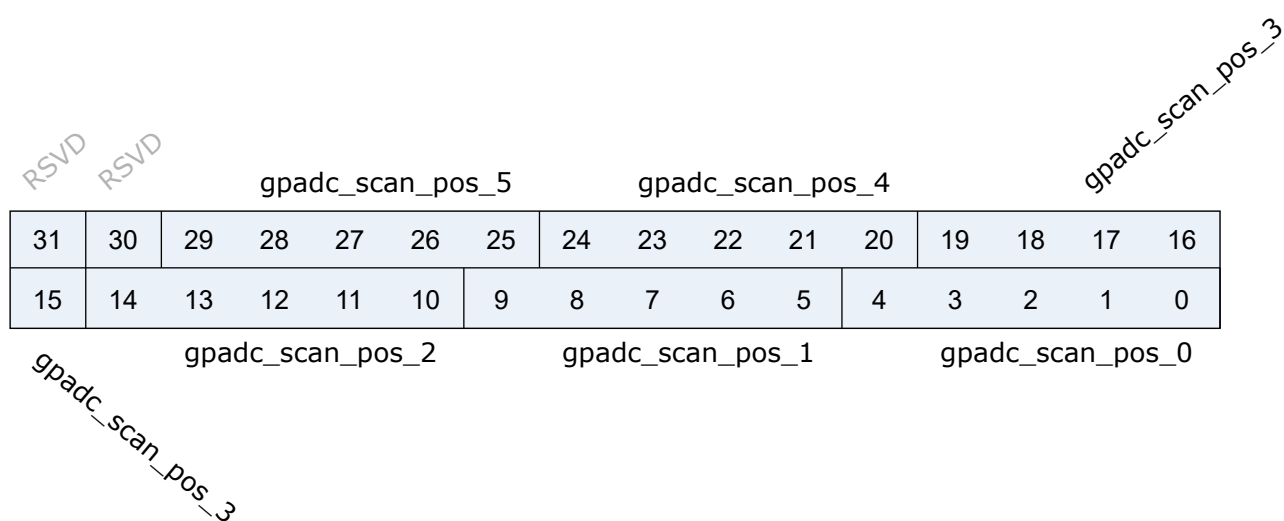
位	名称	权限	复位值	描述
31	gpadc_tsvbe_low	r/w	1'b0	tSEN diode current
30:28	gpadc_dly_sel	r/w	3'h0	adc conversion speed

位	名称	权限	复位值	描述
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
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

位	名称	权限	复位值	描述
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			

5.4.6 gpadc_reg_scn_pos1

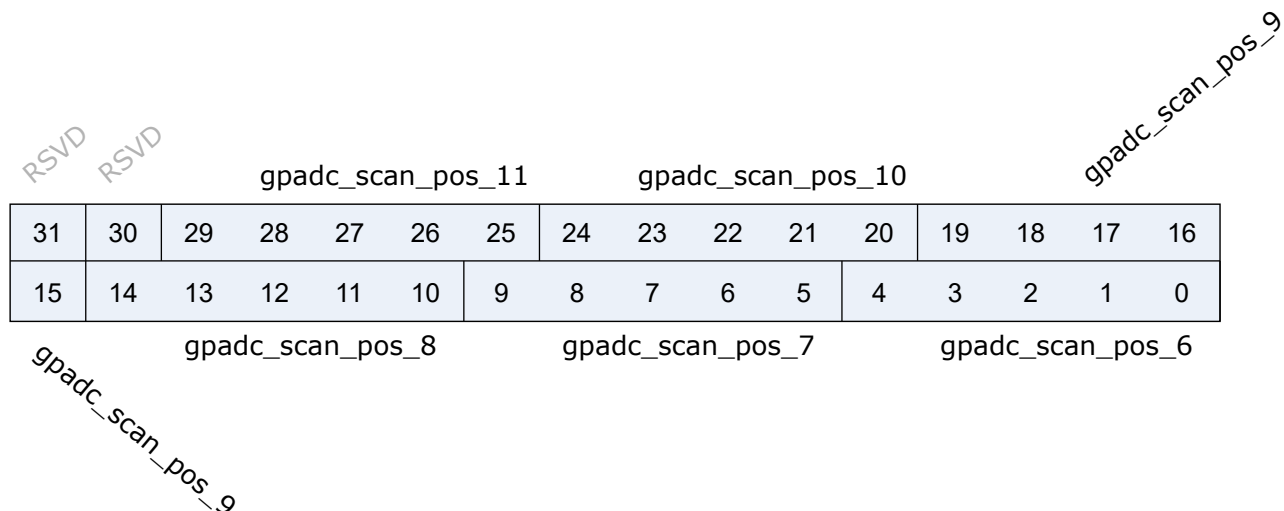
地址: 0x2000f918



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

5.4.7 gpadc_reg_scn_pos2

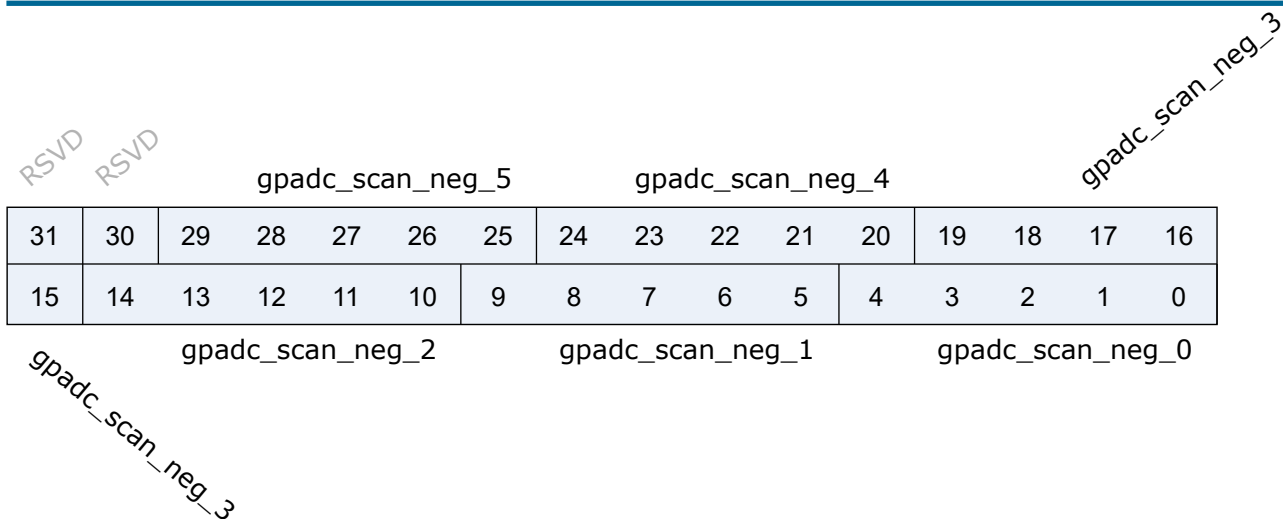
地址：0x2000f91c



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

5.4.8 gpadc_reg_scn_neg1

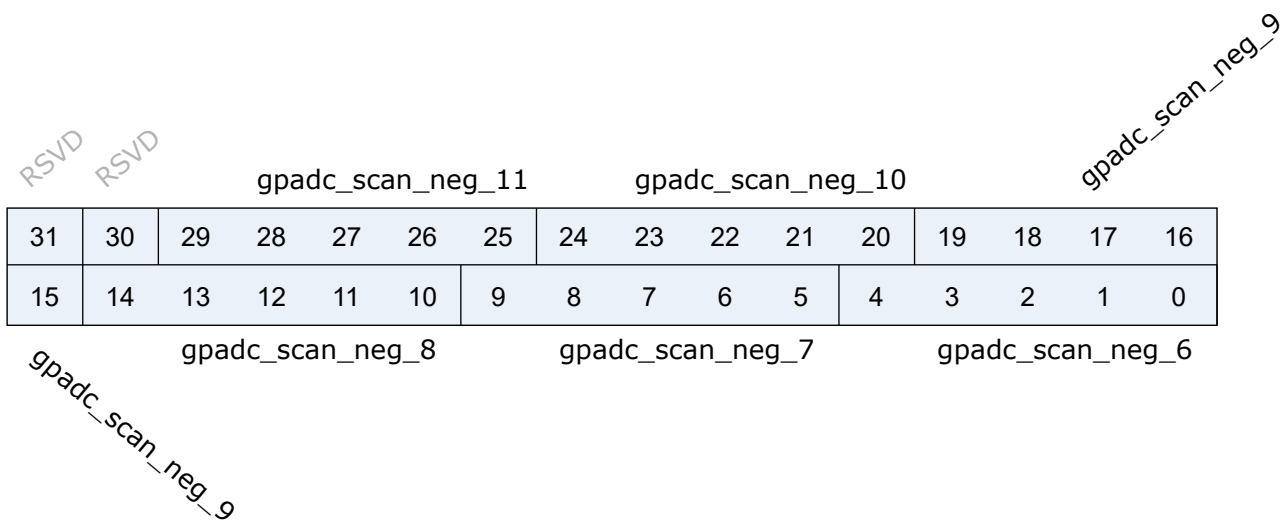
地址：0x2000f920



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

5.4.9 gpadc_reg_scn_neg2

地址: 0x2000f924



5.4.10 gpadc_reg_status

gpadc_reserved

[illegible]

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

5.4.11 gpadc_reg_isr

地址: 0x2000f92c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	gpadc_pos_satur_mask	gpadc_neg_satur_mask	RSVD	RSVD	gpadc_pos_satur_clr	gpadc_neg_satur_clr	RSVD	RSVD	gpadc_pos_satur	gpadc_neg_satur

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
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

6.1 简介

DAC 模块是 12 位电压输出数模转换器，可与 DMA 控制器配合使用。芯片内置的 DAC 模块有两个输出通道，每个通道各有一个独立的转换器，可以互不影响单独进行数模转换。另外此 DAC 的转化器还可以作为 AudioDAC 的模拟输出通道。可用于音频播放，变送器电压调制等应用。

6.2 主要特点

- DAC 调制精度为 12-bit
- DAC 的数模转化器可以作为 AudioDAC 模块的模拟输出通道
- DAC 的输入时钟可选为 32MHz，xclk 或者来自于 AudioDAC 模块
- 支持 DMA 功能，支持 10 种数据传输格式
- 支持 DAC 双通道同时转换
- DAC 的输出引脚固定为 ChannelA 为 GPIO3，ChannelB 为 GPIO2
- 支持内部和外部输入参考电压

6.3 功能描述

DAC 模块基本框图如图所示。

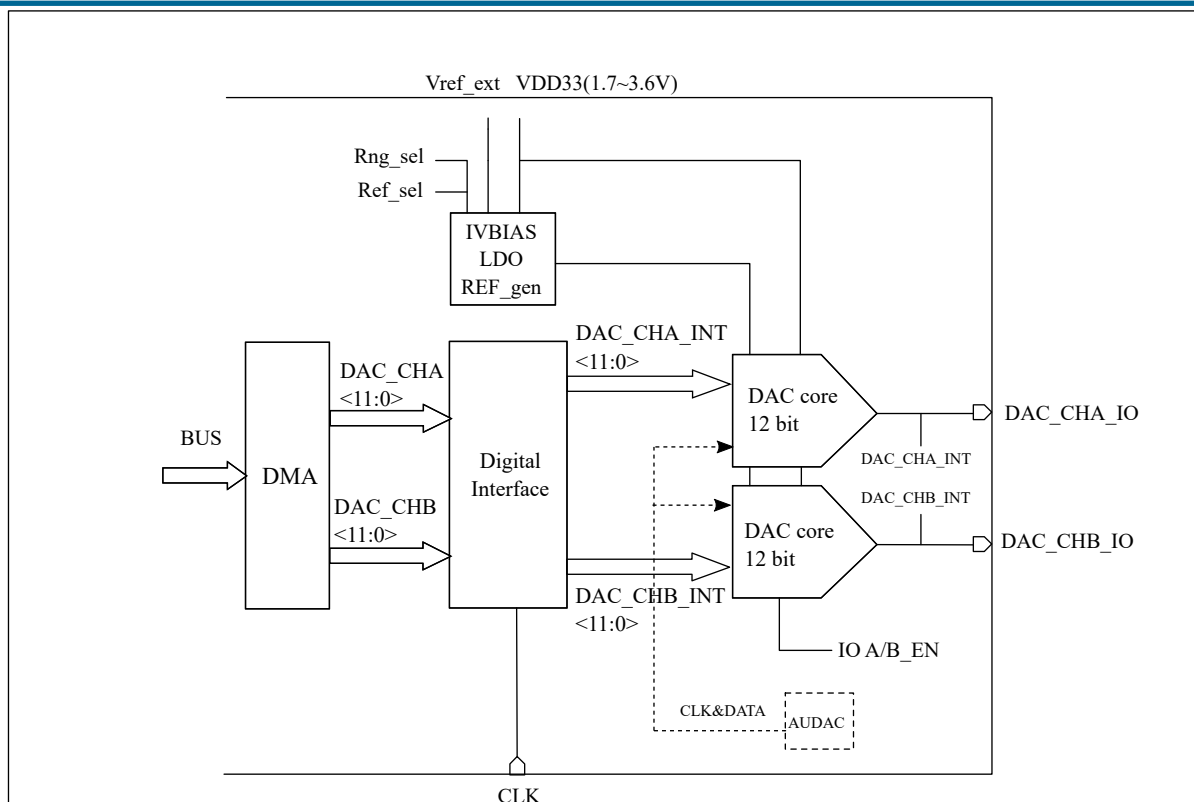


图 6.1: DAC 基本框图

DAC 模块包含两路 DAC 模数转化电路与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，Rng_Sel 来选择输出电压范围。DAC 数模转化器的时钟与数据可以来自于自身的数字控制电路（独立模式），也可以来自 AudioDAC 在模拟模式下的输出（联合模式）。

当 DAC 独立模式工作时，可以由 CPU 直接写入 DAC 调制寄存器（寄存器 dac_cfg3 中的 gpdac_a_data、gpdac_b_data），也可以由 DMA 搬运至 gpdac_dma_wdata 寄存器。

当 DAC 处于联合模式时，自身的 FIFO、数据格式、时钟配置等数字电路配置将会被旁路，不再生效，数模转化器的时钟与数据将会由 AudioDAC 模块接管。

6.3.1 DAC 通道使能

以独立模式下，使能 A 通道为例，配置流程为：

1. 将寄存器 gpdac_config 中对应的 gpdac_en 位置 1，使能 DAC。
2. 将寄存器 dac_cfg0 中对应的 gpdac_ana_clk_sel 与 gpdac_dat_cha_sel 位写 0，工作于独立模式。
3. 将寄存器 dac_cfg1 中对应的 gpdac_a_en 位置 1，使能 DAC A 通道转换
4. 将寄存器 dac_cfg1 中对应的 gpdac_ioa_en 位置 1，使能通道 A 转换结果到 GPIO 口

6.3.2 DAC 数据格式

独立模式下 DMA 或者 CPU 向 FIFO 写入待转化数据时，FIFO 可以支持 10 种不同的存储格式，通过设置寄存器 `gpdac_dma_config` 中对应的 `gpdac_dma_format` 位数值，可以配置 `gpdac_dma_wdata` 寄存器（宽度为 32-bit）中存储的数据传输格式，其对应关系如下：

表 6.1: 数据传输格式

gpdac_dma_format 位对应的数值	gpdac_dma_wdata (对应 A 和 B 通道的数据传输格式)
0	[11:0]: {A0}, {A1}, {A2} ...
1	[27:16][11:0]: {B0,A0}, {B1,A1}, {B2,A2} ...
2	[27:16][11:0]: {A1,A0}, {A3,A2}, {A5,A4} ...
4	[15:4]: {A0}, {A1}, {A2} ...
5	[31:20][15:4]: {B0,A0}, {B1,A1}, {B2,A2} ...
6	[31:20][15:4]: {A1,A0}, {A3,A2}, {A5,A4} ...
8	[31:24][23:16][15:8][7:0]: {A3,A2,A1,A0}, {A7,A6,A5,A4} ...
9	[31:24][23:16][15:8][7:0]: {B1,B0,A1,A0}, {B3,B2,A3,A2} ...
10	[31:24][23:16][15:8][7:0]: {B1,A1,B0,A0}, {B3,A3,B2,A2} ...
11	[15:8][7:0]: {B0,A0}, {B1,A1}, {B2,A2} ...

6.3.3 DAC 输出电压

用户可以通过设置 `dac_cfg0` 寄存器中对应的 `gpdac_ref_sel` 位数值来选择使用外部参考电压还是内部参考电压。

如果选择内部参考电压，配置以及输出电压如下表所示。如果选择外部参考电压，请将外部电压连入固定的 GPIO28。

表 6.2: 内部参考电压输出电压

gpdac_a_rng	gpdac_ref_sel	输出范围 (V)
00	0	0.2-1
01/10	0	0.225-1.425
11	0	0.2-1.8

6.3.4 DAC 转换

6.3.4.1 独立模式时 CPU 方式

独立模式下使用 CPU 搬运数据进行转化，以 A 通道和 B 通道同时转换为例，配置过程如下：

1. 设置 DAC 时钟：将寄存器 `dig_clk_cfg0` 中对应的 `dig_clk_src_sel` 位数值写 1，即选择 `xclk` 作为 DAC 时钟源
2. 将寄存器 `dac_cfg0` 中对应的 `gpdac_ana_clk_sel`、`gpdac_dat_chb_sel` 与 `gpdac_dat_cha_sel` 位写 0，数据与时钟来自于自身，工作于独立模式。
3. 设置时钟分频系数：用户根据需求设置寄存器 `dig_clk_cfg0` 对应的 `dig_512k_div` 位和寄存器 `gpdac_config` 对应的 `gpdac_mode` 位的数值
4. 初始化 A 和 B 通道的 GPIO 引脚
5. 初始化并使能 DAC A 通道和 B 通道
6. 将需要转换的数据写入寄存器 `dac_cfg3` 中对应的 `gpdac_a_data` 和 `gpdac_b_data` 位，完成数据转换

6.3.4.2 独立模式时 DMA 方式

每个 DAC 通道都具有 DMA 功能。以 A 通道使用 DMA 的方式进行数据转换为例，配置过程如下：

1. 设置 DAC 时钟：将寄存器 `dig_clk_cfg0` 中对应的 `dig_clk_src_sel` 位数值写 1，即选择 `xclk` 作为 DAC 时钟源
2. 将寄存器 `dac_cfg0` 中对应的 `gpdac_ana_clk_sel` 与 `gpdac_dat_cha_sel` 位写 0，数据与时钟来自于自身，工作于独立模式。
3. 设置时钟分频系数：用户根据需求设置寄存器 `dig_clk_cfg0` 对应的 `dig_512k_div` 位和寄存器 `gpdac_config` 对应的 `gpdac_mode` 位的数值
4. 初始化 A 通道的 GPIO 引脚为模拟复用
5. 初始化并使能 DAC A 通道
6. 初始化并使能 DMA 通道：设置 DMA 的传输数据宽度、来源地址、目标地址和数据传输长度等
7. 使能 DAC DMA 模式：将寄存器 `gpdac_dma_config` 中对应的 `gpdac_dma_tx_en` 位数值写 1
8. 将需要转换的数据写入寄存器 `gpdac_dma_wdata` 中，根据不同的数据格式作用于 A 或 B 通道，完成数据转换

6.3.4.3 联合模式，作为 AudioDAC 的模拟输出

DAC 数模转化器作为 AudioDAC 的模拟信号输出时，可以直接输出音频模拟信号播放音乐。以双通道差分模式为例，配置过程如下：

1. 将寄存器 `dac_cfg0` 中对应的 `gpdac_ana_clk_sel`、`gpdac_dat_chb_sel` 与 `gpdac_dat_cha_sel` 位写 1，数据与时钟来自于 AudioDAC，工作于联合模式。
2. 初始化并使能 A 和 B 通道，并配置对应引脚为模拟复用功能。
3. 初始化 AudioDAC 模块，并设置为 GPDAC 输出模式，详见 AudioDAC 章节。
4. 启动 AudioDAC，开始播放音频数据，此时 DAC 会同步开始转化。

6.4 寄存器描述

名称	描述
<code>gpdac_config</code>	dac data control register
<code>gpdac_dma_config</code>	dac dma control register
<code>gpdac_dma_wdata</code>	dac fifo data register
<code>gpdac_tx_fifo_status</code>	dac status register
<code>dac_cfg0</code>	DAC source control register
<code>dac_cfg1</code>	DAC CHA control register
<code>dac_cfg2</code>	DAC CHB control register
<code>dac_cfg3</code>	DAC converter data register

6.4.1 `gpadc_config`

地址：0x20002000

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	RSVD			
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	r/w	1'b0	Write 1 to clear flag
9	gpadc_fifo_overrun_clr	r/w	1'b0	Write 1 to clear flag
8	gpadc_rdy_clr	r/w	1'b0	Write 1 to clear flag
7	RSVD			
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

位	名称	权限	复位值	描述
0	gpadc_dma_en	r/w	1'b0	GPADC DMA enable

6.4.2 gpadc_dma_rdata

地址: 0x20002004

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

6.4.3 gpdac_config

地址: 0x20002040

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			

位	名称	权限	复位值	描述
23:20	gpdac_ch_b_sel	r/w	0	Channel B Source Select 0: Reg 1: DMA 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 3: Sin Gen
15:11	RSVD			
10:8	gpdac_mode	r/w	0	0:32k, 1:16k, 3:8k, 4:512k(for DMA only)
7:1	RSVD			
0	gpdac_en	r/w	0	GPDAC enable

6.4.4 gpdac_dma_config

地址: 0x20002044

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	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	gpdac_dma_format	RSVD	RSVD	gpdac_dma_inv_msb	gpdac_dma_tx_en		

位	名称	权限	复位值	描述
31:8	RSVD			

位	名称	权限	复位值	描述
7:4	gpdac_dma_format	r/w	0	DMA TX format (Data 12-bit) 0: ([11:0]) A0, A1, A2... 1: ([27:16][11:0]) B0,A0, B1,A1, B2,A2... 2: ([27:16][11:0]) A1,A0, A3,A2, A5,A4... 4: ([15:4]) A0, A1, A2... 5: ([31:20][15:4]) B0,A0, B1,A1, B2,A2... 6: ([31:20][15:4]) A1,A0, A3,A2, A5,A4... 8: ([31:24][23:16][15:8][7:0]) A3,A2,A1,A0, A7,A6,A5,A4... 9: ([31:24][23:16][15:8][7:0]) B1,B0,A1,A0, B3,B2,A3,A2... 10: ([31:24][23:16][15:8][7:0]) B1,A1,B0,A0, B3,A3,B2,A2... ... 11: ([15:8][7:0]) B0,A0, B1,A1, B2,A2...
3:2	RSVD			
1	gpdac_dma_inv_msb	r/w	0	GPDAC DMA Data Inverse MSB
0	gpdac_dma_tx_en	r/w	0	GPDAC DMA TX enable

6.4.5 gpdac_dma_wdata

地址: 0x20002048

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

6.4.6 gpdac_tx_fifo_status

地址: 0x2000204c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	TxFifoWrPtr			TxFifoRdPtr			tx_cs		tx_fifo_full	tx_fifo_empty

位	名称	权限	复位值	描述
31:2	RSVD			
1	tx_fifo_full	r	0	fifo full flag
0	tx_fifo_empty	r	0	fifo empty flag

6.4.7 dac_cfg0

地址: 0x20000120

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
RSVD	gpdac_dat_chb_sel	gpdac_dat_cha_sel	gpdac_ana_clk_sel	gpdac_test_sel	gpdac_ref_sel	gpdac_test_en	RSVD	RSVD	RSVD	RSVD	RSVD	gpdacb_rstn_ana	gpdaca_rstn_ana		

位	名称	权限	复位值	描述
31:15	RSVD			
14	gpdac_dat_chb_sel	r/w	1'h0	0:data from gpip, 1:data from audio pwm
13	gpdac_dat_cha_sel	r/w	1'h0	0:data from gpip, 1:data from audio pwm
12	gpdac_ana_clk_sel	r/w	1'h0	0:clock from gpip, 1:clock from audio pwm
11:9	gpdac_test_sel	r/w	3'h0	select test point 0 7

位	名称	权限	复位值	描述
8	gpdac_ref_sel	r/w	1'h0	Reference select 1'h0 Internal reference 1'h1 External reference
7	gpdac_test_en	r/w	1'h0	Test enable 1'h0 analog test disabled (ATEST is set in Hi-Z state) 1'h1 analog test point enabled to ATEST
6:2	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

6.4.8 dac_cfg1

地址：0x20000124

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
31:20	RSVD			
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

6.4.9 dac_cfg2

地址：0x20000128

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
31:20	RSVD			
19:18	gpdac_b_rng	r/w	2'h3	Output voltage range control with internal/external reference
17:2	RSVD			
1	gpdac_job_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

6.4.10 dac_cfg3

地址：0x2000012c

gpdac_a_data															
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_b_data															
31:28	RSVD														

位	名称	权限	复位值	描述
27:16	gpdac_a_data	r/w	12'h0	Channel A Data input
15:12	RSVD			
11:0	gpdac_b_data	r/w	12'h0	Channel B Data input

7.1 简介

DMA(Direct Memory Access) 是一种直接内存存取技术，可以独立地直接读写系统内存，而无需处理器介入处理。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。1 个独立的 DMA 控制器，有 4 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。支持 LLI 链接列表功能，可由软件配置传输数据大小、数据源地址和目标地址。

7.2 主要特征

- 1 个 DMA 控制器，包含 4 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
- 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

7.3 功能描述

7.3.1 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时，它会先向仲裁器发送 DMA 请求信号。DMA 向仲裁器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，设备会向仲裁器发送 DMA 结束信号，交还总线控制权。

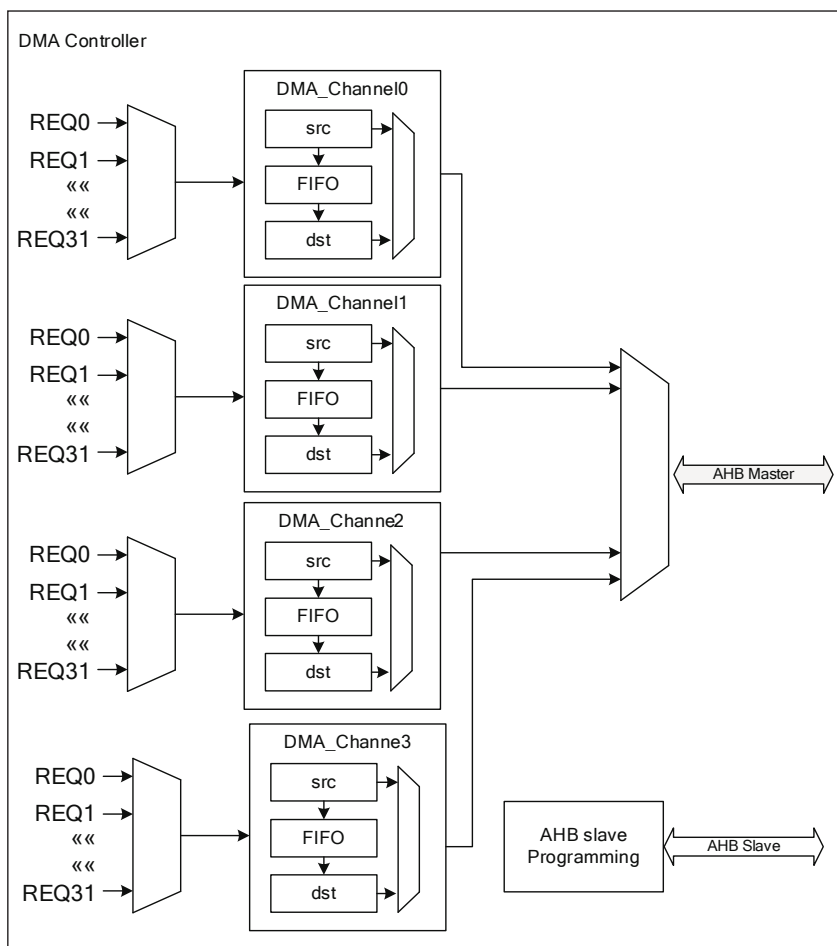


图 7.1: DMA 框图

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，作为数据搬移的端口。AHB Slave 接口作为配置 DMA 的接口，只支持 32-bit 存取。

7.3.2 DMA 通道配置

DMA 支持 4 路通道，各通道之间互不干涉，可以独立运行，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 寄存器中的 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 寄存器中的 SWidth(来源)、DWidth(目标) 位，宽度选项有单字节、双字节、四字节
5. Burst 次数，可通过配置 DMA_C0Control 寄存器中的 SBSIZE(来源)、DBSIZE(目标) 位来设置，配置选项有 INCR1、INCR4、INCR8、INCR16
6. 需要特别注意的是所配置的组合，burst 次数乘以数据宽度小于等于 16 字节
7. 设置数据传输长度的范围为：0-4095

7.3.3 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，外设类型与配置值的对应关系如下表所示：

Value	DMA0
0	UART0_RX
1	UART0_TX
2	UART1_RX
3	UART1_TX
6	I2C0_RX
7	I2C0_TX
9	GPIO_TX
10	SPI0_RX
11	SPI0_TX
13	AUDIO_TX
14	I2C1_RX
15	I2C1_TX
16	I2S_RX
17	I2S_TX
20	DBI_TX
21	AUADC_RX
22	GPADC_RX
23	GPDAC_TX

图 7.2: 外设类型选择

以下是部分外设配置示例：

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 中的 SrcPeripheral 位的值设置为 1，即将 Source peripheral 设置为 UART0_TX
2. 将寄存器 DMA_C0Config 中的 DstPeripheral 位的值设置为 0，即将 Destination peripheral 设置为 UART0_RX

I2C 使用 DMA 传输数据

配置如下：

1. 将寄存器 DMA_C0Config 中的 SrcPeripheral 位的值设置为 7，即将 Source peripheral 设置为 I2C0_TX
2. 将寄存器 DMA_C0Config 中的 DstPeripheral 位的值设置为 6，即将 Destination peripheral 设置为 I2C0_RX

SPI 使用 DMA 传输数据

配置如下：

1. 将寄存器 DMA_C0Config 中的 SrcPeripheral 位的值设置为 11，即将 Source peripheral 设置为 SPI0_TX
2. 将寄存器 DMA_C0Config 中的 DstPeripheral 位的值设置为 10，即将 Destination peripheral 设置为 SPI0_RX

ADC 使用 DMA 传输数据

配置如下：

1. 将寄存器 DMA_C0Config 中的 SrcPeripheral 位的值设置为 22，即将 Source peripheral 设置为 GPADC

7.3.4 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_COLL1 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。

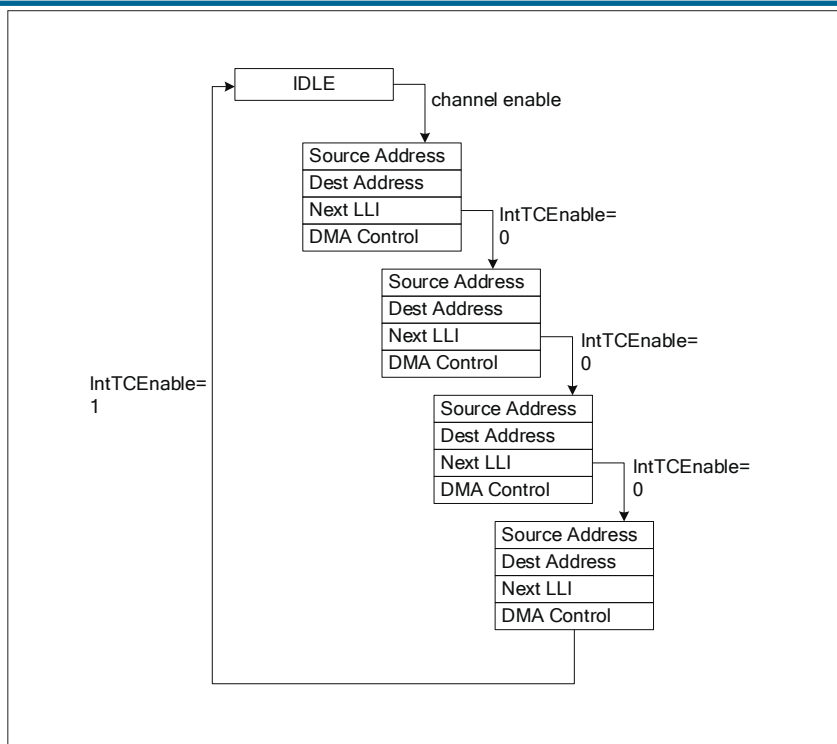


图 7.3: LLI 框架

7.3.5 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

7.4 传输模式

7.4.1 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 中的 FlowCntrl 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，DWidth、SWidth 位

分别设置来源和目标的传输宽度，DBSize、SBSIZE 位分别设置来源和目标的 burst 次数

5. 选择合适的通道，使能 DMA，完成数据传输

7.4.2 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，直到达到设定的搬移数量。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，直到达到设定的搬移数量，完成后自动回到空闲状态，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 中的 FlowCntrl 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，DI 位设置为 0，开启源地址自动累加模式，DWidth、SWidth 位分别设置来源和目标的传输宽度，DBSize、SBSIZE 位分别设置来源和目标的 burst 次数
5. 选择合适的通道，使能 DMA，完成数据传输

7.4.3 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的搬移数量，完成后自动回到空闲状态，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 的值设置为目标内存地址
3. 选择传输模式，将寄存器 DMA_C0Config 中的 FlowCntrl 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 位设置为 1，SI 位设置为 0，开启目的地址自动累加模式，DWidth、SWidth 位分别设置来源和目标的传输宽度，DBSize、SBSIZE 位分别设置来源和目标的 burst 次数
5. 选择合适的通道，使能 DMA，完成数据传输

7.5 寄存器描述

名称	描述
DMA_IntStatus	Interrupt status after masking
DMA_IntTCStatus	TC interrupt status
DMA_IntTCClear	TC interrupt clear
DMA_IntErrorStatus	Error interrupt status
DMA_IntErrClr	Error interrupt clear
DMA_RawIntTCStatus	TC interrupt status before masking
DMA_RawIntErrorStatus	Error interrupt status before 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	Enable and endian
DMA_Sync	Synchronization logic
DMA_C0SrcAddr	Channel 0 source address
DMA_C0DstAddr	Channel 0 destination address
DMA_C0LLI	Channel 0 linked list address
DMA_C0Control	Channel 0 control
DMA_C0Config	Channel 0 configure
DMA_C1SrcAddr	Channel 1 source address
DMA_C1DstAddr	Channel 1 destination address
DMA_C1LLI	Channel 1 linked list address
DMA_C1Control	Channel 1 control
DMA_C1Config	Channel 1 configure
DMA_C2SrcAddr	Channel 2 source address
DMA_C2DstAddr	Channel 2 destination address
DMA_C2LLI	Channel 2 linked list address

名称	描述
DMA_C2Control	Channel 2 control
DMA_C2Config	Channel 2 configure
DMA_C3SrcAddr	Channel 3 source address
DMA_C3DstAddr	Channel 3 destination address
DMA_C3LLI	Channel 3 linked list address
DMA_C3Control	Channel 3 control
DMA_C3Config	Channel 3 configure

7.5.1 DMA_IntStatus

地址: 0x2000c000

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

7.5.2 DMA_IntTCStatus

地址: 0x2000c004

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

7.5.3 DMA_IntTCClear

地址: 0x2000c008

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

IntTCClear

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

7.5.4 DMA_IntErrorStatus

地址: 0x2000c00c

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

IntErrorStatus

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

7.5.5 DMA_IntErrClr

地址：0x2000c010

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

IntErrClr

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

7.5.6 DMA_RawIntTCStatus

地址：0x2000c014

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

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

7.5.7 DMA_RawIntErrorStatus

地址：0x2000c018

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
RawIntErrorStatus															
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD

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

7.5.8 DMA_EnbldChns

地址: 0x2000c01c

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
EnabledChannels															
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD

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

7.5.9 DMA_SoftBReq

地址: 0x2000c020

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

7.5.10 DMA_SoftSReq

地址: 0x2000c024

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

7.5.11 DMA_SoftLBReq

地址: 0x2000c028

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

7.5.12 DMA_SoftLSReq

地址: 0x2000c02c

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

7.5.13 DMA_Config

地址: 0x2000c030

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	M	r/w	0	AHB Master endianness configuration: 0 = little-endian, 1 = big-endian
0	E	r/w	0	SMDMA Enable.

7.5.14 DMA_Sync

地址: 0x2000c034

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

7.5.15 DMA_C0SrcAddr

地址: 0x2000c100

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

7.5.16 DMA_C0DstAddr

地址: 0x2000c104

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

7.5.17 DMA_C0LLI

地址: 0x2000c108

LLI

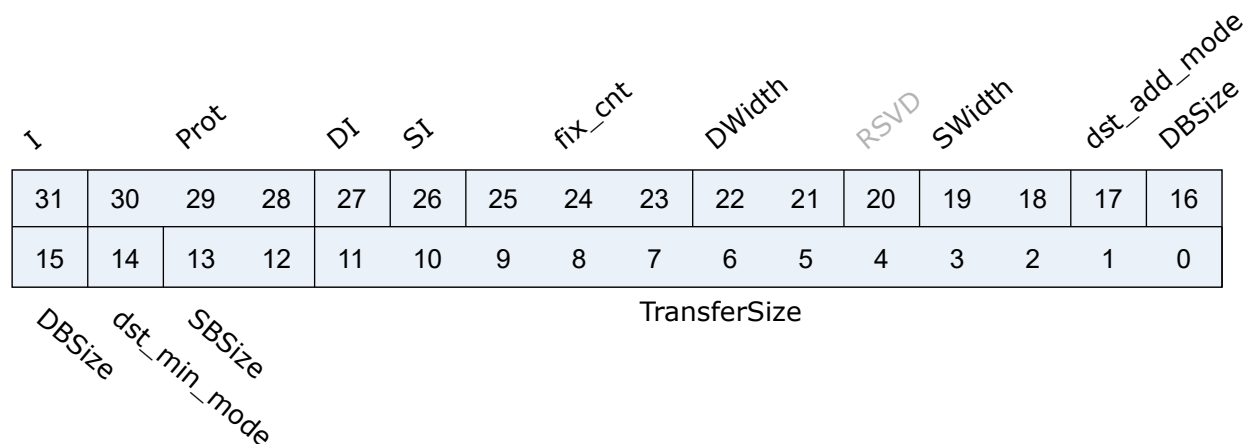
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

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.

7.5.18 DMA_C0Control

地址：0x2000c10c



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	No use for currently
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.
25:23	fix_cnt	r/w	3'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: 2'b00 : byte 2'b01 : half-word 2'b10 : word 2'b11 : double-word
20	RSVD			
19:18	SWidth	r/w	2'b10	Source transfer width 2'b00 : byte 2'b01 : half-word 2'b10 : word 2'b11 : double-word
17	dst_add_mode	r/w	1'b0	Add mode : issue remain destination traffic

位	名称	权限	复位值	描述
16:15	DBSize	r/w	2'b01	Destination burst size 2'b00 : INCR1 2'b01 : INCR4 2'b10 : INCR8 2'b11 : INCR16 Note : SBSIZE*Swidth should <= CH FIFO Size
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: 2'b00 : INCR1 2'b01 : INCR4 2'b10 : INCR8 2'b11 : INCR16 Note : SBSIZE*Swidth should <= CH FIFO Size
11:0	TransferSize	r/w	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

7.5.19 DMA_C0Config

地址: 0x2000c110

RSVD															
LLICounter															
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
ITC		IE		FlowCntrl		DstPeripheral				SrcPeripheral				F	

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.

位	名称	权限	复位值	描述
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.
5:1	SrcPeripheral	r/w	0	Source peripheral.
0	E	r/w	0	Channel enable.
-1:5	RSVD			
4	SrcRemnSgle	r/w	0	Source remain single issue mode
3	DstRemnSgle	r/w	0	Destination remain single issue mode
2:0	RSVD			

7.5.20 DMA_C1SrcAddr

地址: 0x2000c200

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

7.5.21 DMA_C1DstAddr

地址：0x2000c204

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

7.5.22 DMA_C1LLI

地址：0x2000c208

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.

7.5.23 DMA_C1Control

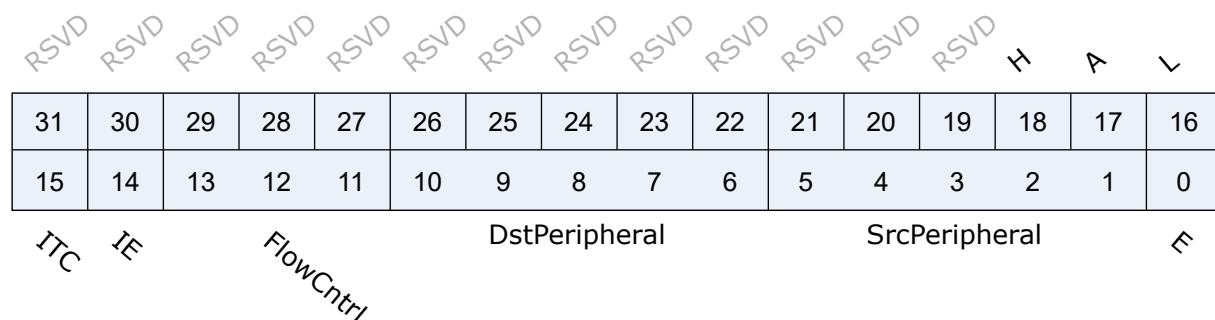
地址：0x2000c20c

1		Prot		DI	SI	fix_cnt			DWidth		RSVD	SWidth		dst_add_mode	
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	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	No use for currently
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.
25:23	fix_cnt	r/w	3'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: 2'b00 : byte 2'b01 : half-word 2'b10 : word 2'b11 : double-word
20	RSVD			
19:18	SWidth	r/w	2'b10	Source transfer width 2'b00 : byte 2'b01 : half-word 2'b10 : word 2'b11 : double-word
17	dst_add_mode	r/w	1'b0	Add mode : issue remain destination traffic
16:15	DBSize	r/w	2'b01	Destination burst size 2'b00 : INCR1 2'b01 : INCR4 2'b10 : INCR8 2'b11 : INCR16 Note : SBSIZE*Swidth should <= CH FIFO Size
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: 2'b00 : INCR1 2'b01 : INCR4 2'b10 : INCR8 2'b11 : INCR16 Note : SBSIZE*Swidth should <= CH FIFO Size
11:0	TransferSize	r/w	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

7.5.24 DMA_C1Config

地址：0x2000c210



位	名称	权限	复位值	描述
31:19	RSVD			
18	H	r/w	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	A	r	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	L	r/w	0	Lock.
15	ITC	r/w	0	Terminal count interrupt mask.
14	IE	r/w	0	Interrupt error mask.
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.
5:1	SrcPeripheral	r/w	0	Source peripheral.
0	E	r/w	0	Channel enable.
-1:5	RSVD			
4	SrcRemnSgle	r/w	0	Source remain single issue mode
3	DstRemnSgle	r/w	0	Destination remain single issue mode
2:0	RSVD			

7.5.25 DMA_C2SrcAddr

地址: 0x2000c300

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

7.5.26 DMA_C2DstAddr

地址: 0x2000c304

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

7.5.27 DMA_C2LLI

地址: 0x2000c308

LLI

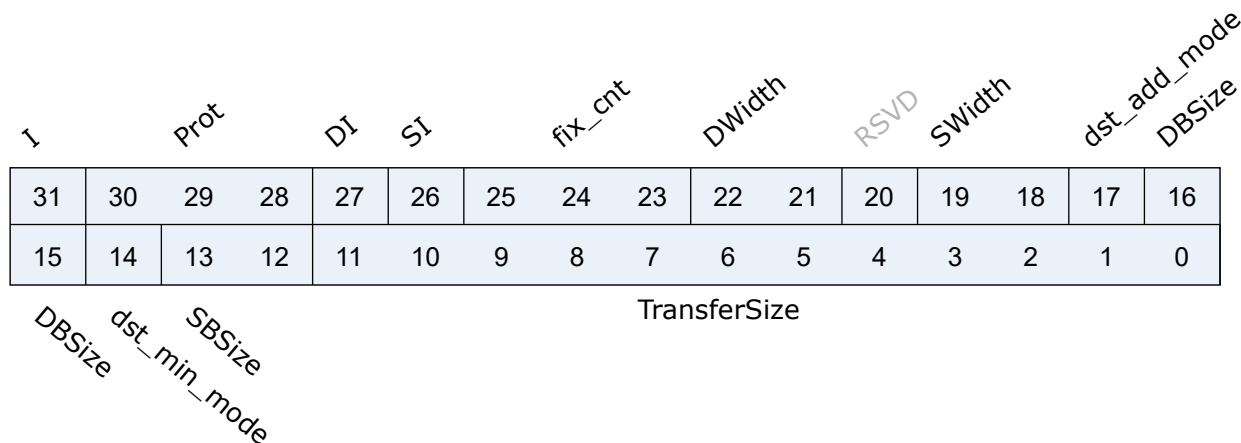
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

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.

7.5.28 DMA_C2Control

地址: 0x2000c30c



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	No use for currently
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.
25:23	fix_cnt	r/w	3'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: 2'b00 : byte 2'b01 : half-word 2'b10 : word 2'b11 : double-word
20	RSVD			

位	名称	权限	复位值	描述
19:18	SWidth	r/w	2'b10	Source transfer width 2'b00 : byte 2'b01 : half-word 2'b10 : word 2'b11 : double-word
17	dst_add_mode	r/w	1'b0	Add mode : issue remain destination traffic
16:15	DBSize	r/w	2'b01	Destination burst size 2'b00 : INCR1 2'b01 : INCR4 2'b10 : INCR8 2'b11 : INCR16 Note : SBSIZE*Swidth should <= CH FIFO Size
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: 2'b00 : INCR1 2'b01 : INCR4 2'b10 : INCR8 2'b11 : INCR16 Note : SBSIZE*Swidth should <= CH FIFO Size
11:0	TransferSize	r/w	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

7.5.29 DMA_C2Config

地址：0x2000c310

RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	H	A	L
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	
ITC	IE			FlowCntrl					DstPeripheral					SrcPeripheral		FF

位	名称	权限	复位值	描述
31:19	RSVD			

位	名称	权限	复位值	描述
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.
5:1	SrcPeripheral	r/w	0	Source peripheral.
0	E	r/w	0	Channel enable.
-1:5	RSVD			
4	SrcRemnSgle	r/w	0	Source remain single issue mode
3	DstRemnSgle	r/w	0	Destination remain single issue mode
2:0	RSVD			

7.5.30 DMA_C3SrcAddr

地址: 0x2000c400

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

7.5.31 DMA_C3DstAddr

地址: 0x2000c404

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

7.5.32 DMA_C3LLI

地址: 0x2000c408

LLI

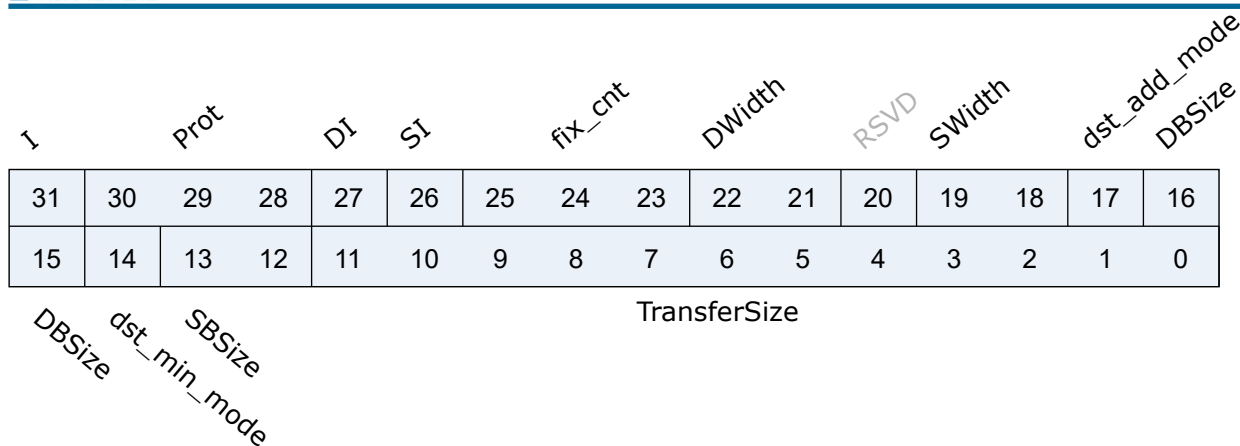
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

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.

7.5.33 DMA_C3Control

地址: 0x2000c40c



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	No use for currently
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.
25:23	fix_cnt	r/w	3'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: 2'b00 : byte 2'b01 : half-word 2'b10 : word 2'b11 : double-word
20	RSVD			
19:18	SWidth	r/w	2'b10	Source transfer width 2'b00 : byte 2'b01 : half-word 2'b10 : word 2'b11 : double-word
17	dst_add_mode	r/w	1'b0	Add mode : issue remain destination traffic

位	名称	权限	复位值	描述
16:15	DBSize	r/w	2'b01	Destination burst size 2'b00 : INCR1 2'b01 : INCR4 2'b10 : INCR8 2'b11 : INCR16 Note : SBSize*Swidth should <= CH FIFO Size
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: 2'b00 : INCR1 2'b01 : INCR4 2'b10 : INCR8 2'b11 : INCR16 Note : SBSize*Swidth should <= CH FIFO Size
11:0	TransferSize	r/w	0	Transfer size: 0 4095. Number of data transfers left to complete when the SMDMA is the flow controller.

7.5.34 DMA_C3Config

地址：0x2000c410

RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	H	A	L
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	
ITC	IE			FlowCntrl					DstPeripheral					SrcPeripheral		Lock

位	名称	权限	复位值	描述
31:19	RSVD			
18	H	r/w	0	Halt: 0 = enable DMA requests, 1 = ignore subsequent source DMA requests.
17	A	r	0	Active: 0 = no data in FIFO of the channel, 1 = FIFO of the channel has data.
16	L	r/w	0	Lock.
15	ITC	r/w	0	Terminal count interrupt mask.
14	IE	r/w	0	Interrupt error mask.

位	名称	权限	复位值	描述
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.
5:1	SrcPeripheral	r/w	0	Source peripheral.
0	E	r/w	0	Channel enable.
-1:5	RSVD			
4	SrcRemnSgle	r/w	0	Source remain single issue mode
3	DstRemnSgle	r/w	0	Destination remain single issue mode
2:0	RSVD			

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
- 接收最多支持 64-bit 数据位
- 64*2 字节的接收 FIFO
- 接收结束中断

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：

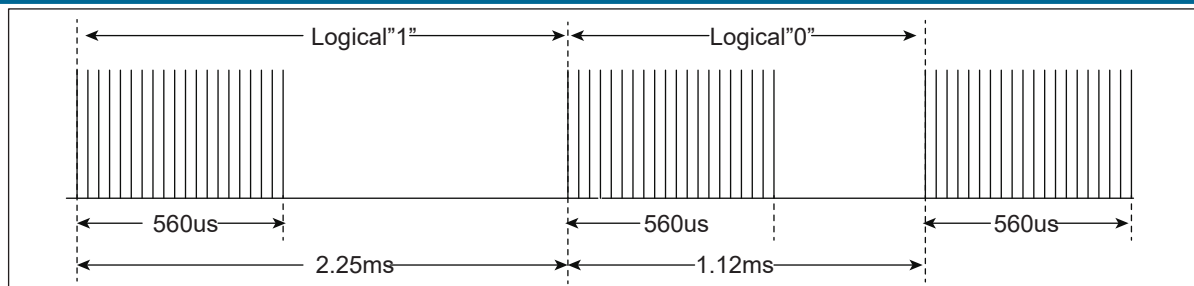


图 8.1: NEC 逻辑波形

逻辑 1 周期为 2.25ms，脉冲时间 560us；逻辑 0 周期为 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 的载波脉冲和 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 载波与 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：

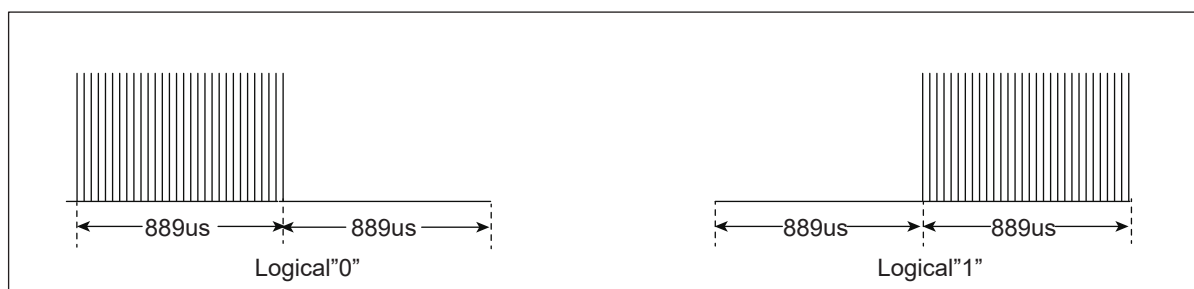


图 8.3: RC5 逻辑波形

逻辑 1 周期为 1.778ms，先是 889us 的低电平后是 889us 的载波；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：

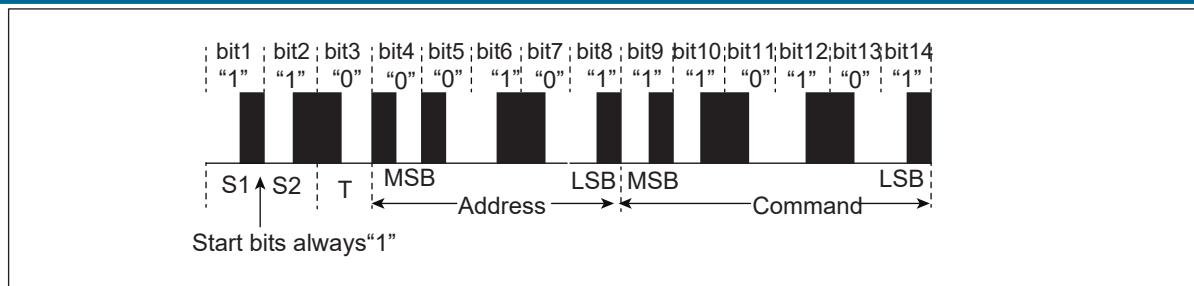


图 8.4: RC5 协议波形

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 会以其内部工作时钟去依次计数每个高或者低电平的持续时间，然后将数据存入到深度为 64，宽度为 2 字节的接收 FIFO 之中。

8.3.3 IR 中断

IR 有单独的接收结束中断，在接收过程中，它会用内部工作时钟去计数每段电平维持的时间，一旦这个计数达到设定的结束阈值，就会产生接收结束中断。可以通过寄存器 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
irrx_config	RX configure
irrx_int_sts	RX interrupt status
irrx_pw_config	RX pulse width threshold
irrx_data_count	RX data bit count
irrx_data_word0	RX low 32-bit data
irrx_data_word1	RX high 32-bit data
ir_fifo_config_0	FIFO status
ir_fifo_config_1	FIFO threshold and available count
ir_fifo_rdata	RX FIFO

8.4.1 irrx_config

地址: 0x2000a640

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	
RSVD	RSVD	RSVD	RSVD	cr_irrx_deg_cnt				RSVD	RSVD	RSVD	cr_irrx_deg_en		cr_irrx_mode		cr_irrx_in_inv	

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

8.4.2 irrx_int_sts

地址：0x2000a644

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
31:27	RSVD			
26	cr_irrx_fer_en	r/w	1'b1	Interrupt enable of irrx_fer_int
25	cr_irrx_frdy_en	r/w	1'b1	Interrupt enable of irrx_frdy_int
24	cr_irrx_end_en	r/w	1'b1	Interrupt enable of irrx_end_int
23:19	RSVD			
18	rsvd	rsvd	1'b0	
17	rsvd	rsvd	1'b0	
16	cr_irrx_end_clr	w1c	1'b0	Interrupt clear of irrx_end_int
15:11	RSVD			
10	cr_irrx_fer_mask	r/w	1'b1	Interrupt mask of irrx_fer_int
9	cr_irrx_frdy_mask	r/w	1'b1	Interrupt mask of irrx_frdy_int
8	cr_irrx_end_mask	r/w	1'b1	Interrupt mask of irrx_end_int
7:3	RSVD			
2	irrx_fer_int	r	1'b0	IRRX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
1	irrx_frdy_int	r	1'b0	IRRX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto-cleared when data is popped
0	irrx_end_int	r	1'b0	IRRX transfer end interrupt

8.4.3 irrx_pw_config

地址: 0x2000a648

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

8.4.4 irrx_data_count

地址: 0x2000a650

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)

8.4.5 irrx_data_word0

地址: 0x2000a654

sts_irrx_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

sts_irrx_data_word0

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

8.4.6 irrx_data_word1

地址: 0x2000a658

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

8.4.7 ir_fifo_config_0

地址: 0x2000a680

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 RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD

RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD rx_fifo_underflow rx_fifo_overflow RSVD RSVD rx_fifo_clr RSVD RSVD RSVD

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
5:4	RSVD			
3	rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO
2:0	RSVD			

8.4.8 ir_fifo_config_1

地址：0x2000a684

rx_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															

位	名称	权限	复位值	描述
31:30	RSVD			
29:24	rx_fifo_th	r/w	6'd0	RX FIFO threshold, irrx_frdy_int will not be asserted if rx_fifo_cnt is less than this value
23:15	RSVD			
14:8	rx_fifo_cnt	r	7'd0	RX FIFO available count
7:32	RSVD			
31:0	tx_fifo_wdata	w	32'h0	IRTX FIFO data Normal Mode: Each entry contains a 32-bit data word, LSB is sent first Software Mode: Each entry contains 4 pulse widths, [7:0] is the 1st pulse, [15:8] is the 2nd pulse, etc)

8.4.9 ir_fifo_rdata

地址：0x2000a68c

rx_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

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

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，设备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。SPI 使用 4 根线完成全双工的通信，这 4 根信号线分别是：CS（片选）、SCLK（时钟）、MOSI(主机输出从机输入)、MISO(主机输入从机输出)。

9.2 主要特征

- 既可作为 SPI 主设备，也可作为 SPI 从设备
- 主从设备都支持 4 种工作模式（CPOL，CPHA）
- 主从设备都支持 1/2/3/4 字节传输模式
- 发送和接收通道各有深度为 32 个字节的 FIFO
- 自适应的 FIFO 深度变化特性，适配高性能的应用场景
 - 当 Frame 为 32Bits 时，FIFO 的深度为 8
 - 当 Frame 为 24Bits 时，FIFO 的深度为 8
 - 当 Frame 为 16Bits 时，FIFO 的深度为 16
 - 当 Frame 为 8Bits 时，FIFO 的深度为 32
- 可调整每个 Frame 的字节传输顺序
- 可配置每个字节内的 MSB/LSB 优先传输
- 灵活的时钟配置，最高可支持 80M 时钟
- 接收忽略功能，可以忽略对每个 Frame 指定位置数据的接收
- 支持从设备模式下的超时机制

- 支持 DMA 传输模式

9.3 功能描述

9.3.1 基本架构

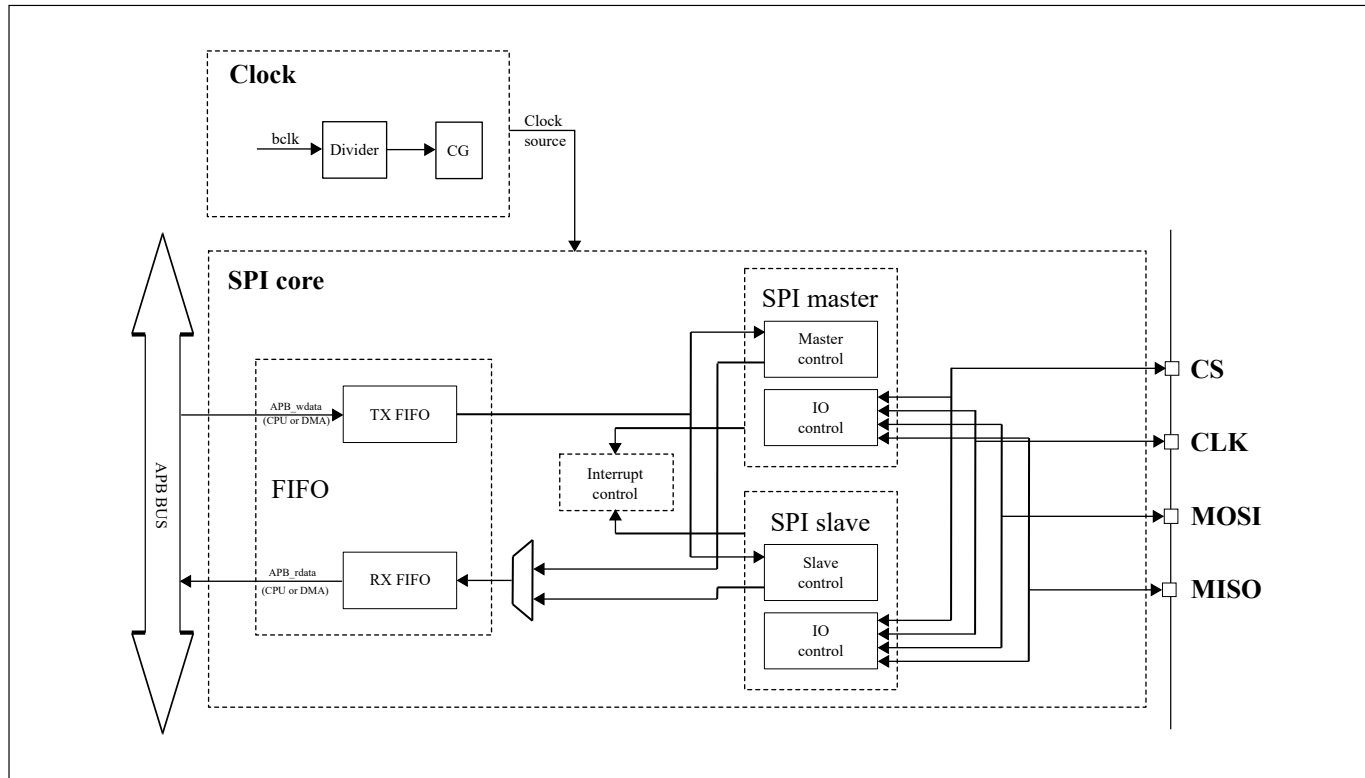


图 9.1: SPI 基本架构图

9.3.2 时序控制

依照不同的时钟极性和相位设定，SPI 时钟共有四种模式，可以通过寄存器 `spi_config` 的 `cr_spi_sclk_pol` (CPOL) 和 `cr_spi_sclk_ph` (CPHA) 进行设置。CPOL 用来决定 SCK 时钟信号空闲时的电平，CPOL=0(`cr_spi_sclk_pol=0`) 则空闲电平为低电平，CPOL=1(`cr_spi_sclk_pol=1`) 则空闲电平为高电平。CPHA 用来决定采样时刻，CPHA=0(`cr_spi_sclk_ph=1`) 则在每个周期的第一个时钟沿采样，CPHA=1(`cr_spi_sclk_ph=0`) 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、每个周期内相位 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：

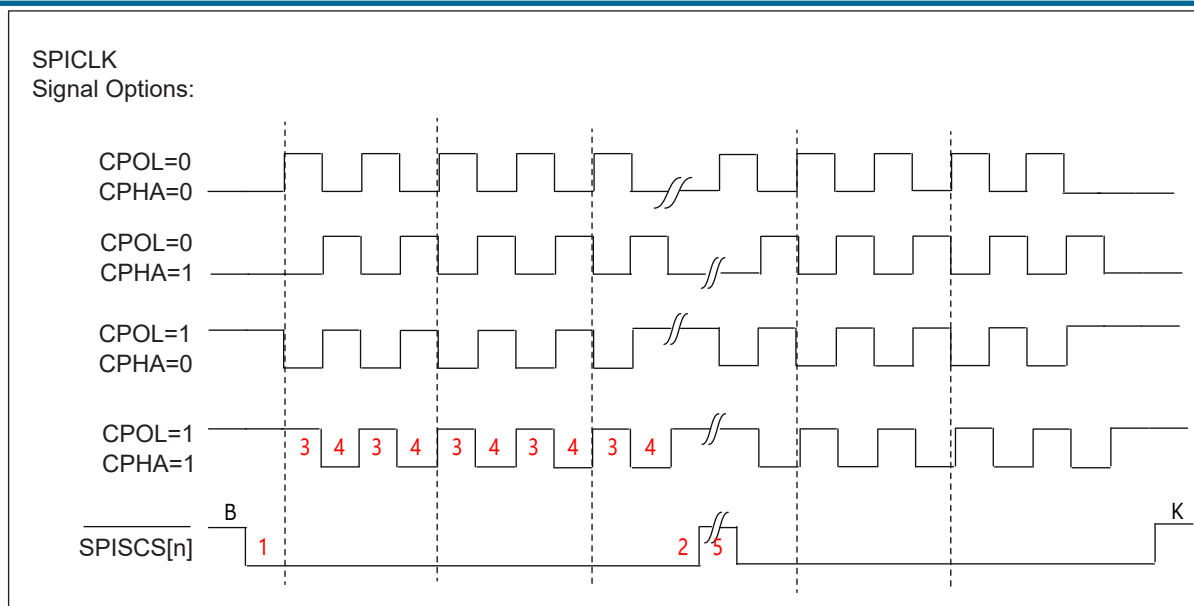


图 9.2: SPI 时序图

其中各数字含义如下：

- 1 是起始条件的长度，通过寄存器 `spi_prd_0` 中的 `cr_spi_prd_s` 进行配置。
- 2 是停止条件的长度，通过寄存器 `spi_prd_0` 中的 `cr_spi_prd_p` 进行配置。
- 3 是相位 0 的长度，通过寄存器 `spi_prd_0` 中的 `cr_spi_prd_d_ph_0` 进行配置。
- 4 是相位 1 的长度，通过寄存器 `spi_prd_0` 中的 `cr_spi_prd_d_ph_1` 进行配置。
- 5 是每帧数据之间的间隔，通过寄存器 `spi_prd_1` 中的 `cr_spi_prd_i` 进行配置。

9.3.3 主设备连续传输模式

开启该模式后，在发送完当前数据帧后 TX FIFO 里还存在待发送数据时，CS 信号不会被拉高。

9.3.4 主从设备数据收发

通过寄存器 `spi_config` 中的 `cr_spi_frame_size` 可以设置数据收发时的 `framesize` (8/16/24/32-bit)，主从设备应保持相同的 `framesize`。如果主设备和从设备约定以 32bits 的 `framesize` 进行通信，在某一帧数据中，主设备的 `clk` 由于异常不满足 32 个周期时，会出现下列现象：

- 主设备当前发送的这帧数据不会进入从设备的 RX FIFO 中，而是被丢弃，从设备当前发送的数据也不会进入主设备的 RX FIFO 中。
- 从设备会认为当前数据帧已经发送结束，等下次主设备 `clk` 正常，继续发送下一帧数据。

9.3.5 接收忽略功能

通过设置忽略的开始位和结束位，SPI 会将接收的每帧数据中的对应数据段丢弃。如下图所示：

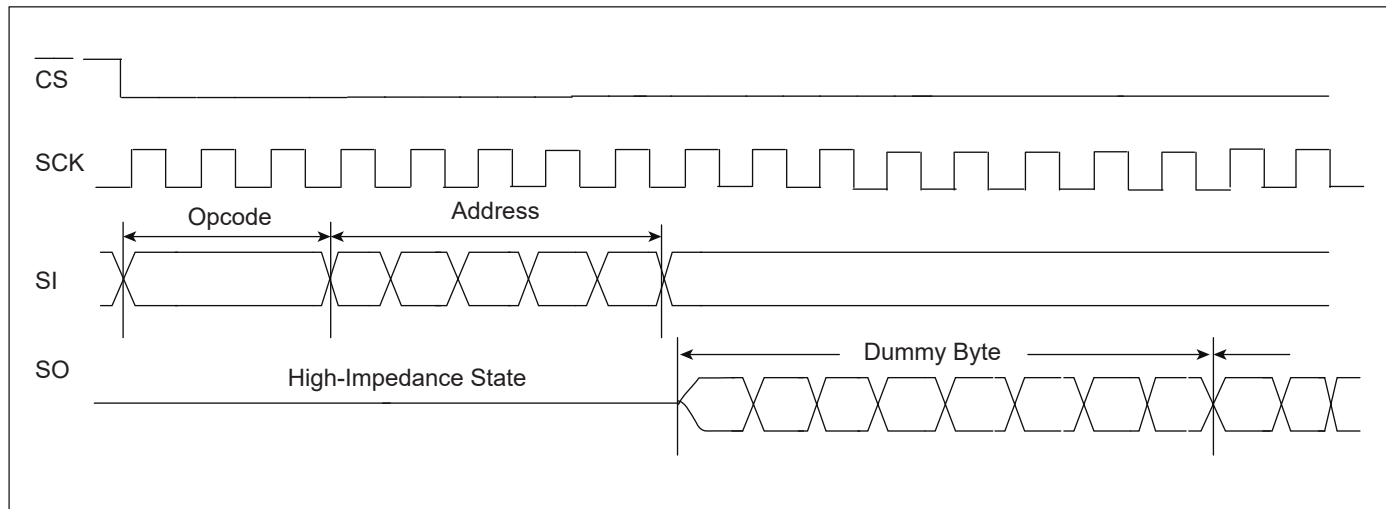


图 9.3: SPI Ignore 波形图

通过配置寄存器 `spi_config` 中的 `cr_spi_rxd_ignr_en` 开启忽略功能。通过配置寄存器 `spi_rxd_ignr` 中的 `cr_spi_rxd_ignr_s` 设置忽略功能的起始位。通过配置寄存器 `spi_rxd_ignr` 中的 `cr_spi_rxd_ignr_p` 设置忽略功能的结束位。

如上图所示，如果忽略的开始位设为 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。

9.3.6 滤波功能

通过使能该功能和设置门限值，SPI 会将小于或等于门限值宽度的数据过滤。假设 SPI top clock 为 160MHz，门限值设为 4，则宽度在 ($4/160\text{MHz} = 25\text{ns}$) 以下的数据都会被过滤掉。该功能由寄存器 `spi_config` 中的 `cr_spi_deg_en` 进行使能，通过配置 `cr_spi_deg_cnt` 可以设置门限值。滤波过程如下图所示，假设 `cr_spi_deg_cnt` 的值设置为 4，input 为初始数据，output 为滤波后的数据。

滤波逻辑过程：

- `tgl` 为 input 和 output 的异或结果。
- `deg_cnt` 从 0 开始计数，计数条件为 `tgl` 为高电平，并且 `reached` 为低电平。
- 若 `deg_cnt` 计数值达到 `cr_urx_deg_cnt` 设置的值时，`reached` 为高电平。
- 当 `reached` 为高电平时，将 input 输出到 output。
- 注释:`deg_cnt` 自加的条件: `tgl` 为高电平且 `reached` 为低电平，其余情况下 `deg_cnt` 会被清 0。

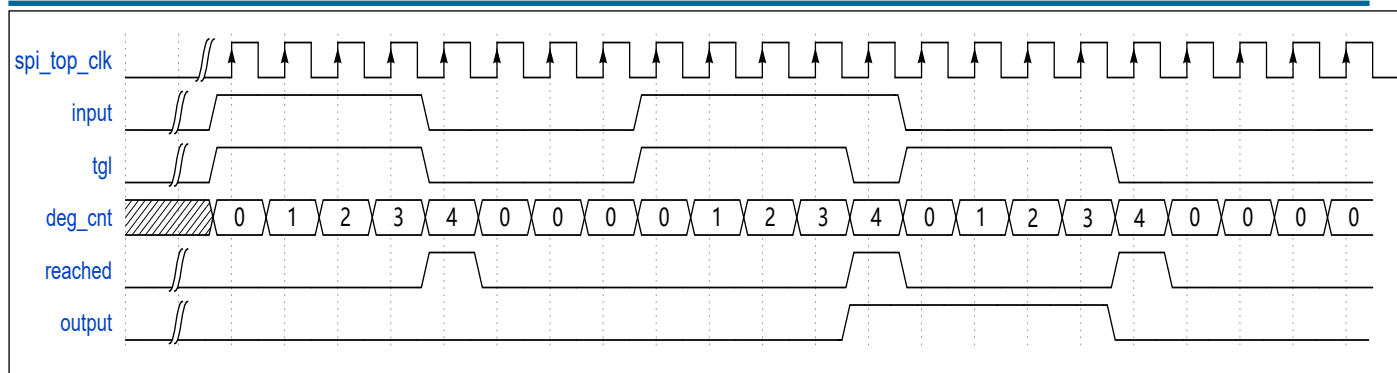


图 9.4: SPI 滤波波形图

9.3.7 可调整字节传输顺序

该功能仅限于调整每一帧数据内不同的 byte 间的优先传输顺序。通过配置寄存器 `spi_config` 中的 `cr_spi_byte_inv` 位进行设置。0 表示优先发送低字节，1 表示优先发送高字节。以 `frame size` 等于 24 bits 传输数据为例，数据格式为：`Data[23:0]=0x123456`。当设置优先发送低字节，传输的顺序为：0x56(第 1 个字节：低字节)；0x34(第 2 个字节：中间字节)；0x12(第 3 个字节：高字节)；当设置优先发送高字节，传输的顺序为：0x12(第 3 个字节：高字节)；0x34(第 2 个字节：中间字节)；0x56(第 1 个字节：低字节)；

字节传输顺序调整功能可以和 MSB/LSB 传输配置功能配合使用。

9.3.8 可配置每个字节的 MSB/LSB 优先传输

该功能仅限于设置每个 byte 中的 8 个 bits 间的优先传输顺序，通过配置寄存器 `spi_config` 中的 `cr_spi_bit_inv` 位进行设置。0 表示 MSB-First, 1 表示 LSB-First。同样以 `frame size` 等于 24 bits 传输数据为例，数据格式为：`Data[23:0]=0x123456`。

当设置为 MSB-First 传输时，传输的顺序为：01010110(二进制，第 1 个字节：0x56)；00110100(二进制，第 2 个字节：0x34)；00010010(二进制，第 3 个字节：0x12)；

当设置为 LSB-First 传输时，传输的顺序为：01101010(二进制，第 1 个字节：0x56)；00101100(二进制，第 2 个字节：0x34)；01001000(二进制，第 3 个字节：0x12)。

9.3.9 从模式超时机制

通过寄存器 `spi_sto_value` 可以设定超时门限值，当 SPI 处于从模式且检测到 CS 被拉低时，会开始计时，如果超过了该超时门限所对应的时间仍未收到时钟信号时，会触发超时中断。

9.3.10 I/O 传输模式

CPU 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。当寄存器 `spi_fifo_config_1` 中的 `rx_fifo_cnt` 大于 `rx_fifo_th` 触发阈值时，将产生 RX 请求中断，通知 CPU 读取 RX FIFO 中的数据。当寄存器 `spi_fifo_config_1` 中的 `tx_fifo_cnt` 大于 `tx_fifo_th` 时，将产生 TX 请求中断，通知 CPU 向 TX FIFO 填充数据。可以通过查询 FIFO 状态寄存器来确定 FIFO 中的采样值以及 FIFO 的状态。需要确保正确的 RX FIFO 触发阈值和 TX FIFO 触发阈值，以防止 FIFO overflow 或 underflow。

9.3.11 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，将寄存器 `spi_fifo_config_0` 中的 `spi_dma_tx_en` 置 1，则开启 DMA 发送模式。将寄存器 `spi_fifo_config_0` 中的 `spi_dma_rx_en` 置 1，则开启 DMA 接收模式。当该模式启用后，SPI 会对 TX/RX FIFO 进行检查，一旦寄存器 `spi_fifo_config_1` 中的 `tx_fifo_cnt/rx_fifo_cnt` 大于 `tx_fifo_th/rx_fifo_th`，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。

9.3.12 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
 - 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 信号被拉高时触发。
- TX FIFO 请求中断
 - TX FIFO 请求中断会在其 FIFO 可用计数值大于设定的阈值时触发，当条件不满足时该中断标志会自动清除。
- RX FIFO 请求中断
 - RX FIFO 请求中断会在其 FIFO 可用计数值大于设定的阈值时触发，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
 - 从模式传输超时中断会在从模式下检测到 CS 拉低之后，超过超时门限值对应的时间后仍未收到时钟信号时触发。
- 从模式 TX 过载中断
 - 从模式 TX 过载中断会在从模式下 TX 没有准备好数据传输而时钟信号却已经到来时触发。
- TX/RX FIFO 溢出中断
 - 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 清除寄存器 `spi_fifo_config_0` 中的 `tx_fifo_clr/rx_fifo_clr` 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	Master and slave configure
spi_int_sts	Interrupt configure and status
spi_bus_busy	Bus busy status
spi_prd_0	Period configure 0
spi_prd_1	Period configure 1
spi_rxd_ignr	RX ignore function
spi_sto_value	Timer-out value setting
spi_fifo_config_0	FIFO status and DMA mode
spi_fifo_config_1	FIFO threshold and available count
spi_fifo_wdata	TX FIFO
spi_fifo_rdata	RX FIFO
backup_io_en	IO backup

9.4.1 spi_config

地址：0x40019000

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
cr_spi_deg_cnt				cr_spi_deg_en				cr_spi_s_3pin_mode				cr_spi_m_cont_en			
								cr_spi_rxd_ignr_en				cr_spi_bit_inv			
								cr_spi_byte_inv				cr_spi_sclk_ph			
								cr_spi_sclk_inv				cr_spi_sclk_pol			
								cr_spi_frame_size				cr_spi_s_en			
												cr_spi_m_en			

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

位	名称	权限	复位值	描述
11	cr_spi_deg_en	r/w	1'b0	Enable signal of all input de-glitch function
10	cr_spi_s_3pin_mode	r/w	1'b0	SPI slave 3-pin mode 1'b0: 4-pin mode (SS_n is enabled) 1'b1: 3-pin mode (SS_n is disabled / don't care)
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(32-bit frame size) / byte[2] is send out first(24-bit frame size) / byte[1] is send out first(16-bit frame size)
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 0: Data is sampled on the second edge of SCLK 1: Data is sampled on the first edge of SCLK
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
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, FIFO space is 1*32 = 32 byte 2'd1: 16-bit, FIFO space is 2*16 = 32 byte 2'd2: 24-bit, FIFO space is 3*8 = 24 byte 2'd3: 32-bit, FIFO space is 4*8 = 32 byte
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：0x40019004

RSVD	RSVD	cr_spi_fer_en	cr_spi_txu_en	cr_spi_sto_en	cr_spi_rxf_en	cr_spi_txf_en	cr_spi_end_en	RSVD	RSVD	rsvd	cr_spi_txu_clr	cr_spi_sto_clr	rsvd	rsvd	cr_spi_end_clr
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	cr_spi_fer_mask	cr_spi_txu_mask	cr_spi_sto_mask	cr_spi_rxf_mask	cr_spi_txf_mask	cr_spi_end_mask	RSVD	RSVD	spi_fer_int	spi_txu_int	spi_sto_int	spi_rxf_int	spi_txf_int	spi_end_int

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_txe_int
24	cr_spi_end_en	r/w	1'b1	Interrupt enable of spi_end_int
23:22	RSVD			
21	rsvd	rsvd	1'b0	
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	rsvd	rsvd	1'b0	
17	rsvd	rsvd	1'b0	
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_txe_int

位	名称	权限	复位值	描述
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'b1	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：0x40019008

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	sts_spi_bus_busy

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 0: Idle 1: Busy

9.4.4 spi_prd_0

地址：0x40019010

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
cr_spi_prd_p								cr_spi_prd_s							

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 (unit: SPI source clock period)
23:16	cr_spi_prd_d_ph_0	r/w	8'd15	Length of DATA phase 0 (unit: SPI source clock period)
15:8	cr_spi_prd_p	r/w	8'd15	Length of STOP condition (unit: SPI source clock period)
7:0	cr_spi_prd_s	r/w	8'd15	Length of START condition (unit: SPI source clock period)

9.4.5 spi_prd_1

地址：0x40019014

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
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 (unit: SPI source clock period)

9.4.6 spi_rxd_ignr

地址：0x40019018

RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	cr_spi_rxd_ignr_s			
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	RSVD	RSVD	RSVD	RSVD	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 (unit: bit)
15:5	RSVD			
4:0	cr_spi_rxd_ignr_p	r/w	5'd0	Stopping point of RX data ignore function (unit: bit)

9.4.7 spi_sto_value

地址: 0x4001901c

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
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: 0x40019080

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	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	rx_fifo_underflow	rx_fifo_overflow	tx_fifo_underflow	tx_fifo_overflow	rx_fifo_clr	tx_fifo_clr	spi_dma_rx_en
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	rx_fifo_underflow	rx_fifo_overflow	tx_fifo_underflow	tx_fifo_overflow	rx_fifo_clr	tx_fifo_clr	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, RX FIFO will be empty when write 1 to this bit
2	tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO, TX FIFO will be empty when write 1 to this bit
1	spi_dma_rx_en	r/w	1'b0	Enable signal of dma_rx_req/ack interface
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: 0x40019084

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	rx_fifo_th	rx_fifo_th	rx_fifo_th	rx_fifo_th	rx_fifo_th	RSVD	RSVD	RSVD	tx_fifo_th	tx_fifo_th	tx_fifo_th	tx_fifo_th	tx_fifo_th
RSVD	RSVD	rx_fifo_cnt	rx_fifo_cnt	rx_fifo_cnt	rx_fifo_cnt	rx_fifo_cnt	rx_fifo_cnt	RSVD	RSVD	tx_fifo_cnt	tx_fifo_cnt	tx_fifo_cnt	tx_fifo_cnt	tx_fifo_cnt	tx_fifo_cnt

位	名称	权限	复位值	描述
31:29	RSVD			
28:24	rx_fifo_th	r/w	5'd0	RX FIFO threshold, dma_rx_req will not be asserted if rx_fifo_cnt is less than this value
23:21	RSVD			
20:16	tx_fifo_th	r/w	5'd0	TX FIFO threshold, dma_tx_req will not be asserted if tx_fifo_cnt is less than this value
15:14	RSVD			
13:8	rx_fifo_cnt	r	6'd0	RX FIFO available count, means byte count of data received in RX FIFO (unit: byte)
7:6	RSVD			
5:0	tx_fifo_cnt	r	6'd32	TX FIFO available count, means empty space remained in TX FIFO (unit: byte)

9.4.10 spi_fifo_wdata

地址: 0x40019088

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	TX FIFO write data port Note: Partial valid if cr_spi_frame_size is set to different value: 2'd0 (8-bit frame): Only [7:0] are valid and [31:8] are don't-care 2'd1 (16-bit frame): Only [15:0] are valid and [31:16] are don't-care 2'd2 (24-bit frame): Only [23:0] are valid and [31:24] are don't-care 2'd3 (32-bit frame): Entire [31:0] are valid

9.4.11 spi_fifo_rdata

地址：0x4001908c

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	RX FIFO read data port Note: Partial valid if cr_spi_frame_size is set to different value: 2'd0 (8-bit frame): Only [7:0] are valid and [31:8] are all 0s 2'd1 (16-bit frame): Only [15:0] are valid and [31:16] are all 0s 2'd2 (24-bit frame): Only [23:0] are valid and [31:24] are all 0s 2'd3 (32-bit frame): Entire [31:0] is valid

9.4.12 backup_io_en

地址：0x400190fc

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	backup_io_en	r/w	1'b0	Enable IO backup function

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。BL616/BL618 共有 2 组 UART，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
- 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
- 32 字节发送/接收 FIFO
- 支持 DMA 传输模式
- 支持 10Mbps 及其以下波特率
- 支持 LIN 总线协议

- 支持 RS485 模式
- 时钟源可以选择 160M/BCLK/XCLK
- 支持滤波功能

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。BL616/BL618 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：

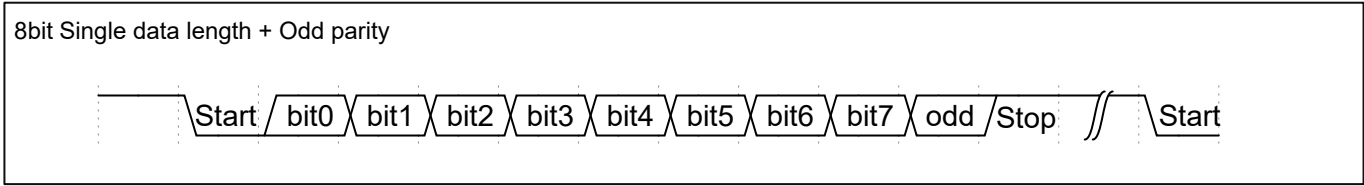


图 10.1: UART 数据格式

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

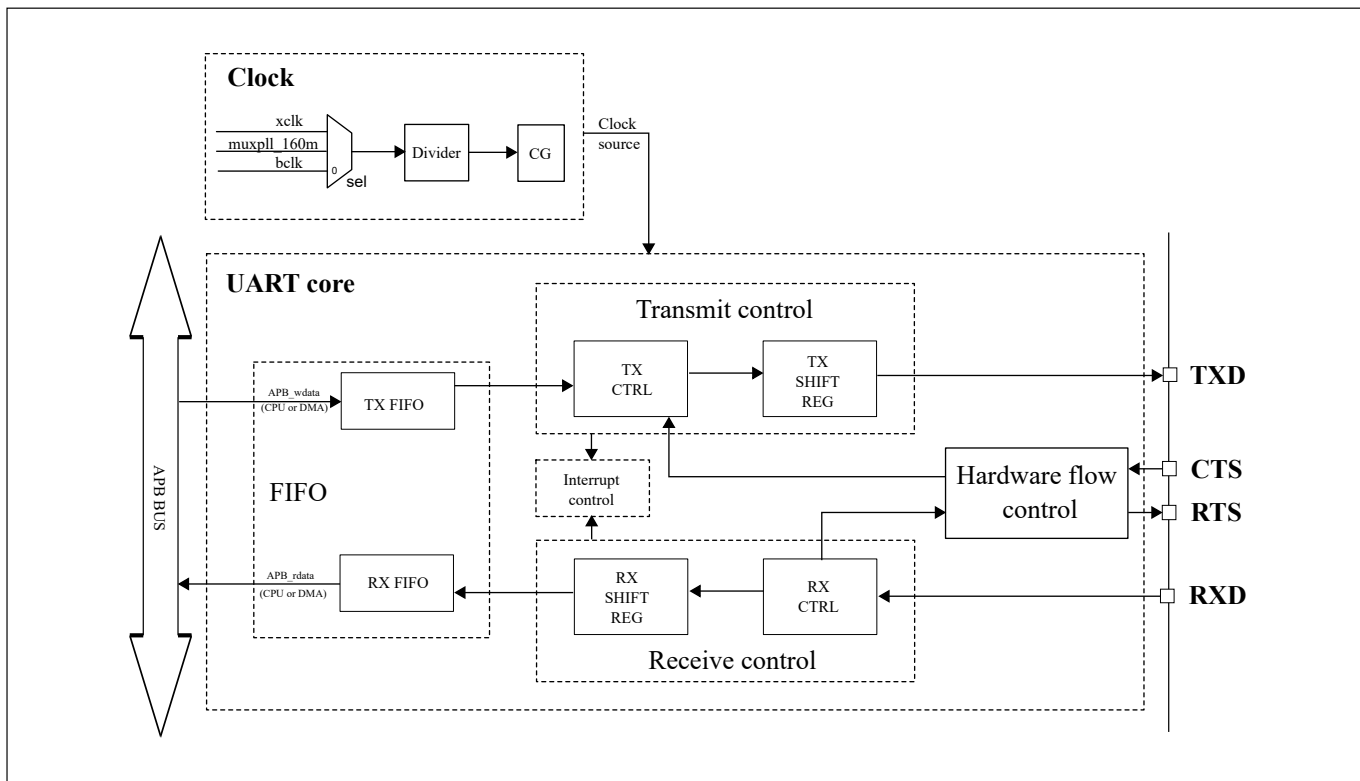


图 10.2: UART 基本架构图

UART 有 3 个时钟源：XCLK，160MHz CLK 以及 BCLK。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。

UART 控制器分为两个功能模块：发送器和接收器。

10.3.3 发送器

发送器包含一个 32 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。

10.3.4 接收器

接收器包含一个 32 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` 中的 `rx_fifo_cnt` 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。具体中断触发条件请参考 RX 超时中断小节。寄存器 `urx_config` 中的 `cr_urx_deg_en` 和 `cr_urx_deg_cnt` 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，然后在将其送去采样。

10.3.5 波特率设定

$$Baudrate = \frac{UART_clk}{uart_prd + 1}$$

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：

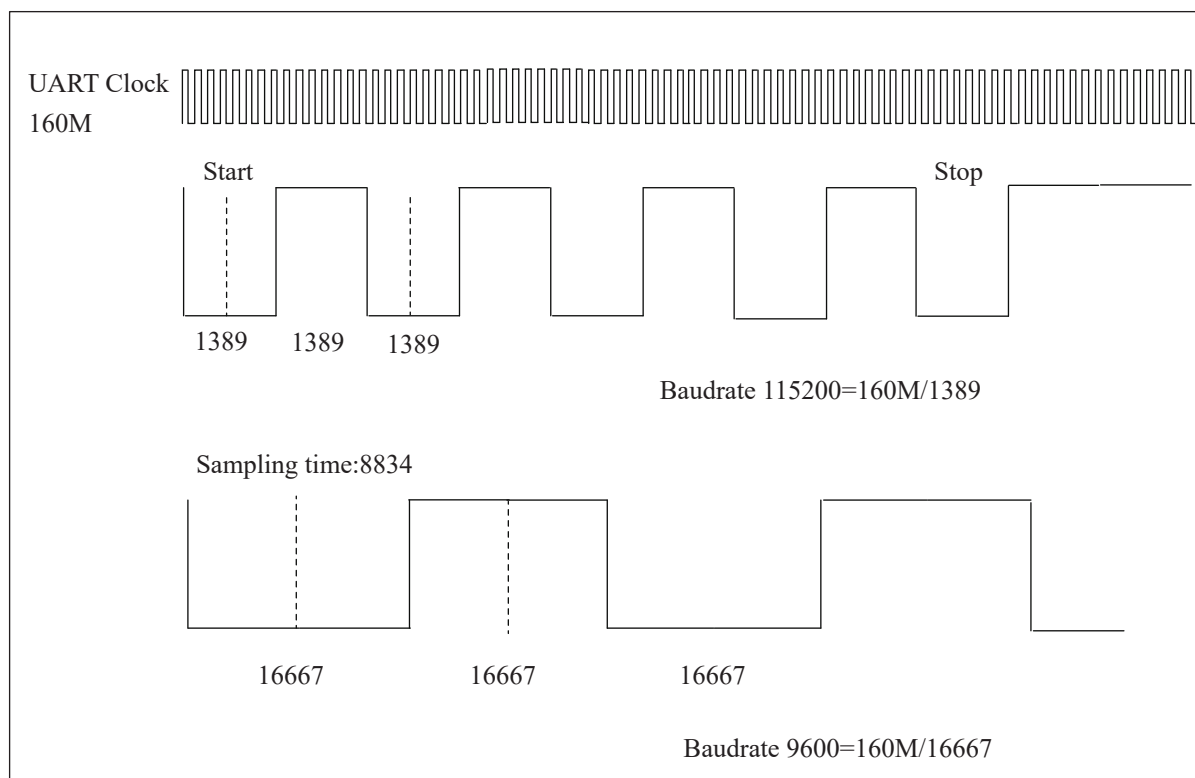


图 10.3: UART 采样波形图

10.3.6 滤波

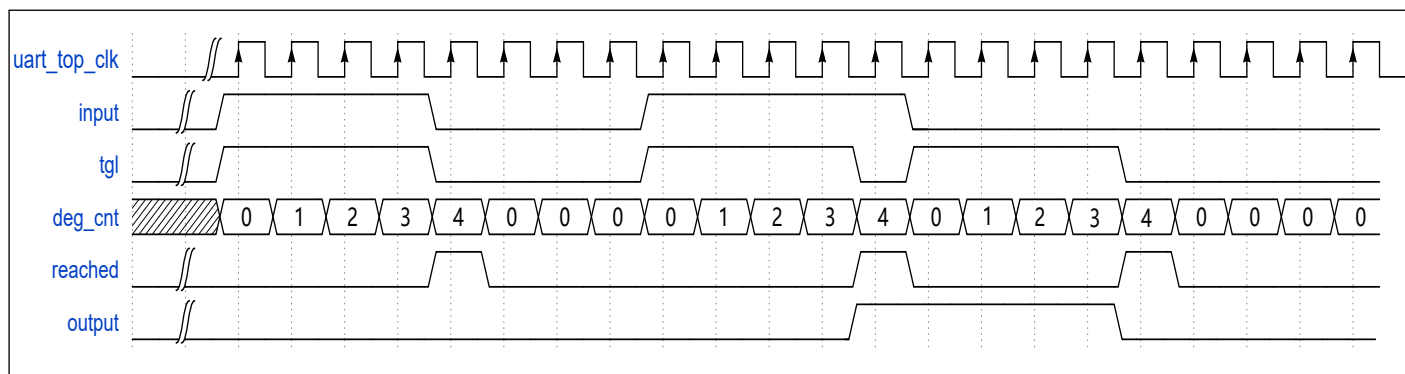


图 10.4: UART 滤波波形图

通过配置寄存器 `urx_config` 中的 `cr_urx_deg_en` 使能该功能，配置 `cr_urx_deg_cnt` 设置门限值，UART 会将达不到门限值宽度的数据过滤掉。

如上图所示，若想滤去数据宽度小于 4 的数据，需要将 `cr_urx_deg_cnt` 的值设置为 4。input 为初始数据，output 为滤波后的数据。

滤波逻辑过程：

- `tgl` 为 `input` 和 `output` 的异或结果。
- `deg_cnt` 从 0 开始计数，计数条件为 `tgl` 为高电平，并且 `reached` 为低电平。
- 若 `deg_cnt` 计数值达到 `cr_urx_deg_cnt` 设置的值时，`reached` 为高电平。
- 当 `reached` 为高电平时，将 `input` 输出到 `output`。
- 注释：`deg_cnt` 自加的条件：`tgl` 为高电平且 `reached` 为低电平，其余情况下 `deg_cnt` 会被清 0。

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

对于接收到的任意字符数据，UART 模块会以 UART 的时钟周期为单位去计数起使位的位宽，该计数值被写入寄存器 `STS_URX_ABR_PRD` 的低 16 位 `sts_urx_abr_prd_start` 中用以计算波特率，因此当起始位之后的第一个数据位为 1 时即可得到正确的波特率，如 LSB-FIRST 下的 '0x01'。

固定字符 (方波) 模式

该模式下，UART 模块在用时钟周期去计数起使位的位宽后，会继续计数后续数据位的位宽，并与起始位相比较，如果连续 7 个计数值之间的上下浮动在最大允许误差范围内，则通过检测，否则计数值会被丢弃。最大允许误差可以通过寄存器 `urx_abr_pw_tol` 中的 `cr_urx_abr_pw_tol` 位来设置，以 UART 的时钟周期为单位。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 模块才会将满足最大允许误差范围的连续 7 个计数值的第一个计数值写入寄存器 STS_URX_ABR_PRD 的高 16 位 sts_urx_abr_prd_0x55 中。

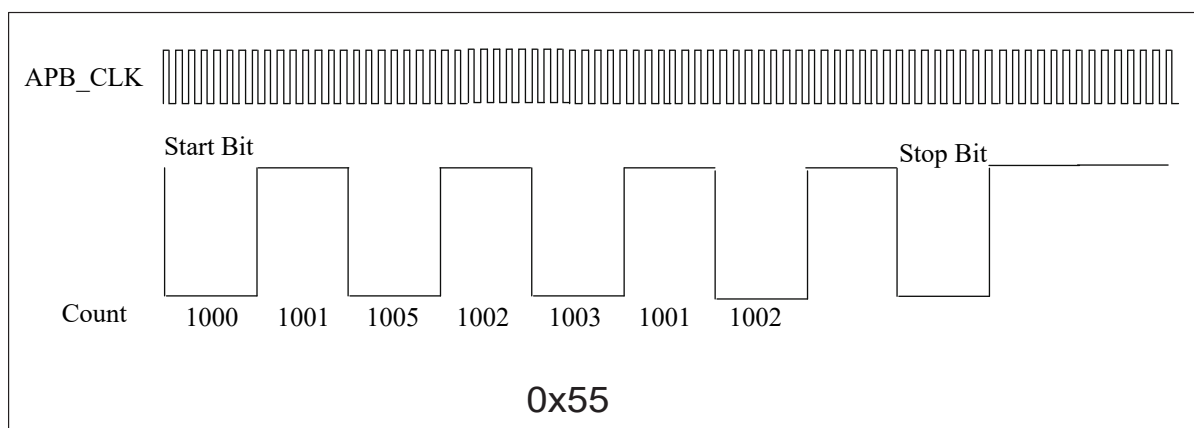


图 10.5: UART 固定字符模式波形图

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:

$$Baudrate = \frac{UART_clk}{Count + 1}$$

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:

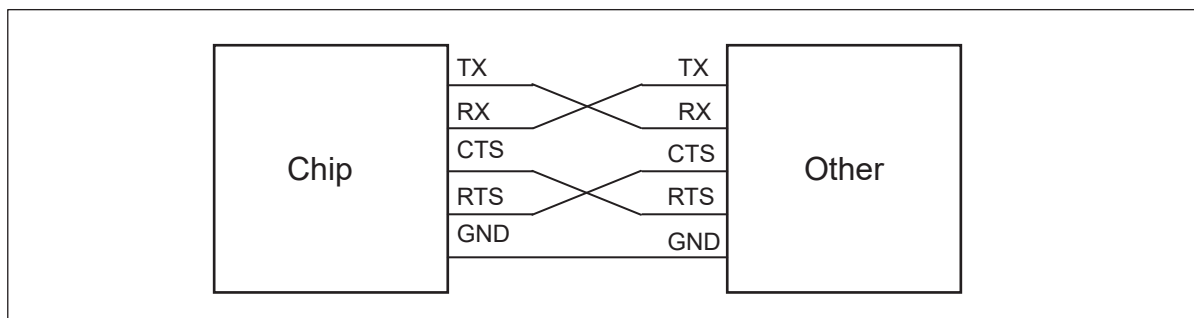


图 10.6: UART 硬件流控图

RTS (Require To Send, 发送请求) 为输出信号, 用于指示芯片准备好可接收对方数据, 低电平有效, 低电平说明芯片可以接收数据。

CTS (Clear To Send, 清除发送) 为输入信号, 用于判断芯片是否可以向对方发送数据, 低电平有效, 低电平说明芯片可以向对方发送数据。

当使用硬件流控功能时，芯片端输出信号 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。当芯片检测到输入信号 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。CTS 在通信过程中的任意时刻拉高或者拉低，不影响 TX 发送数据的连续性，对方收到的数据也是连续的。

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。

10.3.9 DMA 传输

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。

为了保证芯片 DMA TX Channel 搬运数据的正确性，Channel 配置中需要满足以下条件： $(transferWidth * burstSize) \leq (tx_fifo_th + 1)$ 。

为了保证芯片 DMA RX Channel 搬运数据的完整性，Channel 配置中需要满足以下条件： $(transferWidth * burstSize) = (rx_fifo_th + 1)$ 。

10.3.10 LIN 总线支持

本地互联网络（LIN）协议基于 Volvo 衍生公司 Volcano 通信技术公司（VCT）开发的 Volcano-Lite 技术。LIN 是 CAN 和 SAE J1850 协议的补充性协议，针对时间要求不高或不需要精确容错的应用（因为 LIN 没有 CAN 协议那样可靠）。LIN 的目标是易于使用，作为 CAN 协议的低成本替代品。LIN 在车辆中可以使用的场合包括车窗升降器、后视镜、雨刷和雨量传感器。

UART 模块支持 LIN 总线模式，LIN 总线采用主从模式，数据总是由主节点发起，主节点发送的帧（头）包含三个部分：同步间隔字段、同步字节字段和一个标识符字段。一个典型的 LIN 数据传输如图所示：

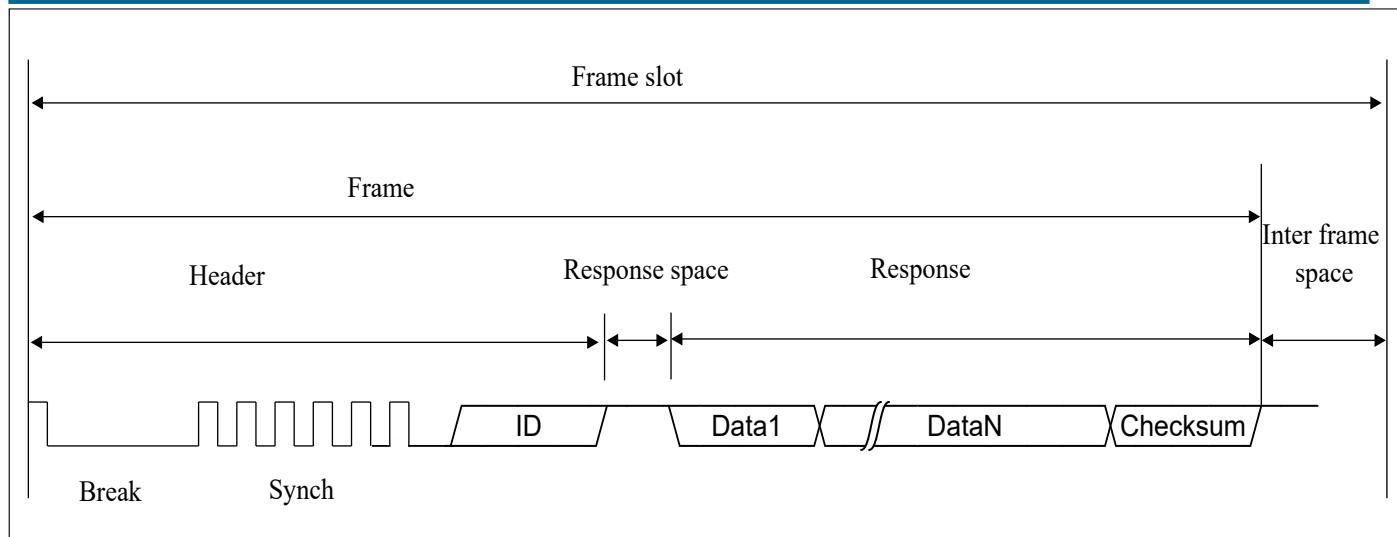


图 10.7: 一个典型的 LIN 帧

1. LIN 的 Break 域

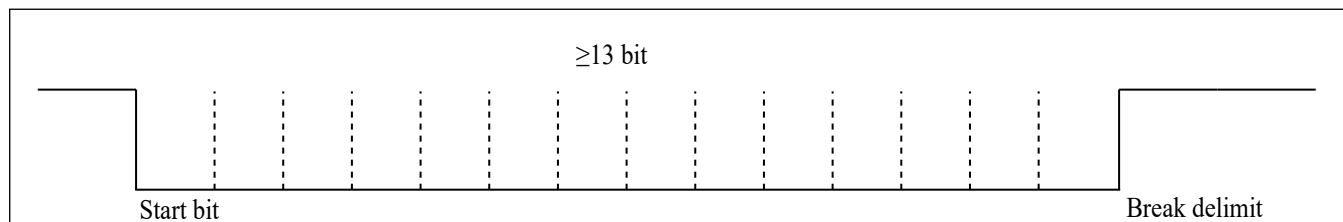


图 10.8: LIN 的 Break 域

同步间隔字段表示报文的开始，至少 13 个显性位（包括起始位）。同步间隔以一个“间隔分隔符”结束，该分隔符至少包含一个隐性位。LIN 帧中的 Break 长度可以通过 `utx_config` 中的 `cr_utx_bit_cnt_b` 来设定。

2. LIN 的 Sync 域

发送同步字节字段来确定两个下降沿之间的时间，从而确定主节点使用的传输速率。位模式是 0x55 (01010101，最大下降沿数量)。

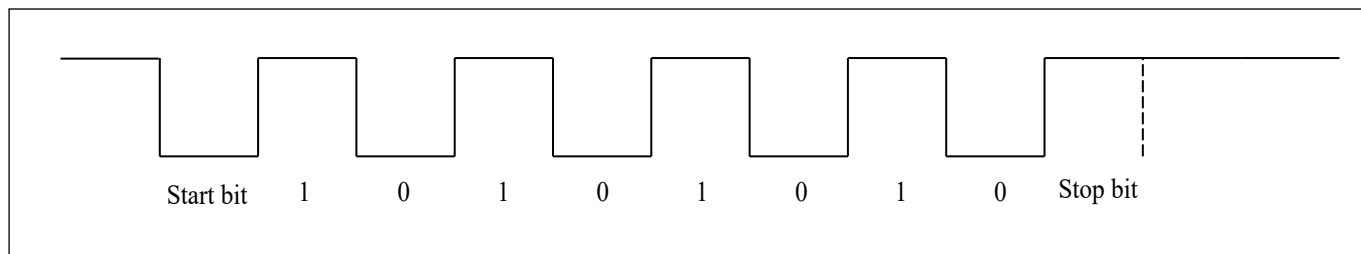


图 10.9: LIN 的 Sync 域

3. LIN 的 ID 域

标识符字段包含 6 位长的标识符和两个奇偶校验位。6 位标识符包含关于发送方和接收方的信息，以及响应中要求的

字节数。奇偶校验位如下进行计算：校验位 P0 是 ID0、ID1、ID2 和 ID4 之间进行逻辑“或”运算的结果。校验位 P1 是 ID1、ID3、ID4 和 ID5 之间逻辑“或”运算后再进行反转的结果。

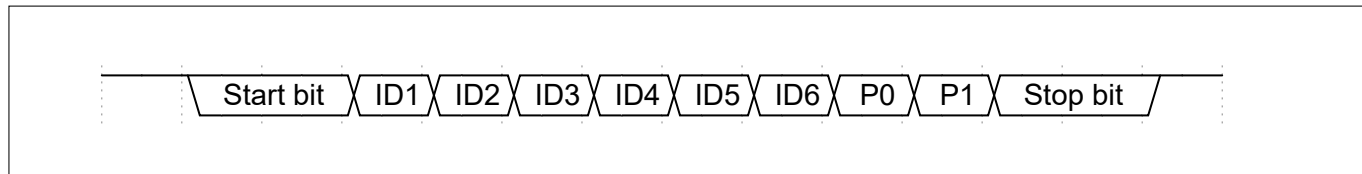


图 10.10: LIN 的 ID 域

从节点等待同步间隔字段，然后通过同步字节字段开始主从节点之间的同步。根据主节点发送的标识符，从节点将进行接收、发送或什么都不做。应该进行发送的从节点发送主节点请求的字节数，然后以一个检验和字段结束传输。

使用 LIN 模式需要设置寄存器 `utx_config` 中的 `cr_utx_lin_en`。通过配置 `cr_utx_bit_cnt_b`，应确保同步间隔段至少是由 13 位的显性电平组成。

10.3.11 RS485 模式

UART 模块支持 RS485 模式，通过设置寄存器 `UTX_RS485_CFG` 中的 `cr_utx_rs485_en` 可以让 UART 模块工作在 RS485 模式下，此时，可以通过外接一个 RS485 的收发器连接到 RS485 总线上，在该模式下，模块中的 RTS 引脚，会转为 RS485 收发器的 Dir 功能，当 UART 模块有数据需要发送时，会自动控制 RTS 引脚为高电平，让 RS485 收发器把数据发送到总线上，反之，当 UART 模块没有数据需要发送时，会自动控制 RTS 为低电平，让 RS485 收发器处在接收状态。

使用该模式需要设置寄存器 `UTX_RS485_CFG` 中的 `cr_utx_rs485_pol` 和 `cr_utx_rs485_en`。

10.4 UART 中断

UART 有着丰富的中断控制，包括以下几种中断模式：

- TX 传输完成中断
- RX 接收完成中断
- 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 BCR 中断
- LIN 同步错误中断

- 通用模式自动波特率检测中断
- 固定字符模式自动波特率检测中断

10.4.1 TX/RX 传输完成中断

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 传输完成中断，如果启用了 TX 自由运行（FreeRun）模式，则不会产生 TX 传输完成中断，TX 功能也不会停止。如果未启用 TX 自由运行（FreeRun）模式，则产生该中断的同时，TX 功能也会停止，用户如果需要使用 TX，必须重新启用 TX 功能。与 TX 不同，RX 功能在产生 RX 传输完成中断后仍可以正常使用。

对于 TX 而言，如果连续向 TX FIFO 中填入的数据大于设置的传输长度值，则 TX 引脚上只会传输设定长度值的数据，多余的数据会留存在 TX FIFO 中，重新启用 TX 功能后，TX FIFO 中剩余的数据将会发出。

例如：设置 TX 传输长度值为 64，启用 TX 功能，先向 TX FIFO 中填入 63 个字节，这 63 个字节会在引脚上传输，但是没有 TX 传输完成中断产生，再向 TX FIFO 中填入 1 字节，此时发送的长度达到 TX 设置的传输长度值，会产生发送完成中断，并且 TX 功能会停止。继续向 TX FIFO 中填入 1 字节，会发现引脚上没有数据传输，该字节还保留在 TX FIFO 中，将 TX 功能关闭后重新启用，该字节在引脚上发出。

对于 RX 而言，如果对方发送的数据长度超过了设置的传输长度，RX 在产生 RX 传输完成中断后依旧可以继续接收数据。

例如：设置 RX 传输长度值为 16，对方发送了 32 个字节数据，RX 会在收到 16 个字节数据时产生 RX 传输完成中断，并继续接收剩下的 16 个字节数据，全部保存在 RX FIFO 中。

10.4.2 TX/RX FIFO 请求中断

当 `uart_fifo_config_1` 中的 `tx_fifo_cnt` 大于 `tx_fifo_th` 时，产生 TX FIFO 请求中断，当条件不满足时该中断标志会自动清除。当 `uart_fifo_config_1` 中的 `rx_fifo_cnt` 大于 `rx_fifo_th` 时，产生 RX FIFO 请求中断，当条件不满足时该中断标志会自动清除。TX/RX 均可支持多次发送/接收，并非一次性达到 `tx_fifo_th/rx_fifo_th` 设置的值。

例如：

1. 设置寄存器 `uart_fifo_config_1` 中的 `tx_fifo_th/rx_fifo_th` 为 16。
2. 置位寄存器 `utx_config` 中的 `cr_utx_frm_en`，使能 free run mode。
3. 设置寄存器 `uart_int_mask` 中的 `cr_utx_frdy_mask/cr_urx_frdy_mask` 为 0，将 TX/RX 的 FIFO 中断打开。
4. 设置寄存器 `utx_config/urx_config` 中的 `cr_utx_en/cr_urx_en`，使能 TX/RX。
5. TX FIFO 中断：TX 会一直进入 FIFO 中断，当芯片发送 128 字节后，将 `cr_utx_frdy_mask` 置 1 屏蔽该中断。如果想再次进入 TX FIFO 中断，`cr_utx_frdy_mask` 置 0 即可。
6. RX FIFO 中断：对方先发送 15 字节数，无中断产生，此时获取 `rx_fifo_cnt` 值为 15，再次发送 1 个字节达到 `rx_fifo_th` 设置的值时产生中断。产生发送中断后，对方再次发送数据，芯片可以收到数据。

10.4.3 RX 超时中断

RX 超时中断产生条件是：在上一次接收数据之后，接收器会开始计时，当计时值超过超时门限值仍未收到下一个数据时触发中断。超时门限值以通信的 bit 为单位。

当对方给芯片发送数据时，达到设置的超时时长后，会产生一次超时中断。

10.4.4 RX 奇偶校验错误中断

RX 奇偶校验错误中断会发生在奇偶校验出错时。但是不影响 RX 正确接收并解析对方发来的数据。RX 在接收数据的时候，会取前 8bit 作为数据位，忽略奇偶校验位，因此数据一致性完整。

例如设置寄存器 `utx_config/urx_config` 中的 `cr_utx_prt_en/cr_urx_prt_en` 使能奇偶校验；设置寄存器 `utx_config/urx_config` 中的 `cr_utx_prt_sel/cr_urx_prt_sel` 选择奇偶校验种类；当对方使用奇校验/偶校验给芯片发送数据时，芯片的 RX 不开启奇偶校验，RX 仍能收到正确的数据。

10.4.5 TX/RX FIFO 溢出中断

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：寄存器 `UART_FIFO_CONFIG_0` 中的 `tx_fifo_clr` 和 `rx_fifo_clr` 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 `UART_INT_STS` 查询溢出的具体中断状态，通过向寄存器 `UART_INT_CLEAR` 相应的位写 1 清除中断。

10.4.6 RX BCR 中断

当 RX 接收到的数据字节数达到寄存器 `urx_bcr_int_cfg` 中的 `cr_urx_bcr_value` 设置的值时，会产生 Byte Count Reached (BCR) 中断。

RX BCR 中断与 RX END 中断都可以重复产生，两者的区别在于：RX END 中断内部的计数器在每次产生该中断时清零并重新开始计数，所以无法通过查询该计数器的状态得知当前总共收到了多少数据；而 RX BCR 中断内部的计数器在产生该中断时不会清零，而是继续累加，所以可以通过查询寄存器 `urx_bcr_int_cfg` 中的 `sts_urx_bcr_count` 位来获取当前接收的数据总量。如果想清除 RX BCR 中断内部的计数器，只需要向寄存器 `uart_int_clear` 中的 `cr_urx_bcr_clr` 位写 1。

10.4.7 LIN 同步错误中断

当 `utx_config` 中的 `cr_utx_lin_en` 使能后，进入 LIN 模式，在 LIN 模式下，如果接收数据时，没有检测到 LIN 总线的同步段，则产生 LIN 同步错误中断。

10.4.8 通用/固定字符模式自动波特率检测中断

在自动波特率模式下，根据配置对应的中断，在自动波特率检测通过时，可以产生通用模式自动波特率检测中断或者固定字符模式自动波特率检测中断。

10.5 寄存器描述

名称	描述
utx_config	TX configure
urx_config	RX configure
uart_bit_prd	Baud rate setting
data_config	LSB/MSB-first select
utx_ir_position	IR mode TX setting
urx_ir_position	IR mode RX setting
urx_rto_timer	Time-out value setting
uart_sw_mode	software mode
uart_int_sts	UART interrupt status
uart_int_mask	UART interrupt mask
uart_int_clear	UART interrupt clear
uart_int_en	UART interrupt enable
uart_status	Bus busy status
sts_urx_abr_prd	Auto baud rate detection
urx_abr_prd_b01	ABR detection bit 0/1
urx_abr_prd_b23	ABR detection bit 2/3
urx_abr_prd_b45	ABR detection bit 4/5
urx_abr_prd_b67	ABR detection bit 6/7
urx_abr_pw_tol	0x55 ABR max allowable error
urx_bcr_int_cfg	BCR interrupt counter
utx_rs485_cfg	RS-485 configure
uart_fifo_config_0	FIFO status and DMA mode
uart_fifo_config_1	FIFO threshold and available count
uart_fifo_wdata	TX FIFO

名称	描述
uart_fifo_rdata	RX FIFO

10.5.1 utx_config

地址: 0x40010000

cr_utx_len

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_cnt_b			cr_utx_bit_cnt_p			cr_utx_bit_cnt_d			cr_utx_ir_inv		cr_utx_ir_en		cr_utx_prt_sel		cr_utx_prt_en		cr_utx_lin_en		cr_utx_frm_en		cr_utx_cts_en		cr_utx_en	

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), TX end interrupt will be triggered when TX send cr_utx_len+1 bytes data (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 4: 4+1+8=13 bit 5: 5+1+8=14 bit ...
12:11	cr_utx_bit_cnt_p	r/w	2'd1	UART TX STOP bit count (unit: 0.5 bit) 0: 0.5 bit 1: 1 bit 2: 1.5 bit 3: 2 bit
10:8	cr_utx_bit_cnt_d	r/w	3'd7	UART TX DATA bit count for each character 4: 5 bit 5: 6 bit 6: 7 bit 7: 8 bit
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

10.5.2 urx_config

地址: 0x40010004

cr_urx_len

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_deg_cnt cr_urx_deg_en cr_urx_bit_cnt_d cr_urx_ir_inv cr_urx_ir_en cr_urx_prt_sel cr_urx_lin_en RSVD cr_urx_abr_en cr_urx_en

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, like cr_utx_bit_cnt_d
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

10.5.3 uart_bit_prd

地址: 0x40010008

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, RX baud rate = uart clock / (cr_urx_bit_prd + 1)
15:0	cr_utx_bit_prd	r/w	16'd255	Period of each UART TX bit, TX baud rate = uart clock / (cr_utx_bit_prd + 1)

10.5.4 data_config

地址：0x4001000c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	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

10.5.5 utx_ir_position

地址：0x40010010

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

10.5.6 urx_ir_position

地址: 0x40010014

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

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

10.5.7 urx_rto_timer

地址: 0x40010018

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

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)

10.5.8 uart_sw_mode

地址: 0x4001001c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	cr_urx_rts_sw_val	cr_urx_rts_sw_mode	cr_utx_txd_sw_val

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 0: Low level 1: High level
2	cr_urx_rts_sw_mode	r/w	1'b0	UART RX RTS output SW control mode enable
1	cr_utx_txd_sw_val	r/w	1'b0	UART TX TXD output SW control value 0: Low level 1: High level
0	cr_utx_txd_sw_mode	r/w	1'b0	UART TX TXD output SW control mode enable

10.5.9 uart_int_sts

地址：0x40010020

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
RSVD	RSVD	RSVD	RSVD	urx_ad5_int	urx_ads_int	urx_bcr_int	urx_lse_int	urx_fer_int	urx_fer_int	urx_pce_int	urx_rto_int	urx_frdy_int	urx_end_int	urx_end_int	urx_end_int

位	名称	权限	复位值	描述
31:12	RSVD			
11	urx_ad5_int	r	1'b0	UART RX ABR Detection finish interrupt using codeword 0x55

位	名称	权限	复位值	描述
10	urx_ads_int	r	1'b0	UART RX ABR Detection finish interrupt using START bit
9	urx_bcr_int	r	1'b0	UART RX byte count reached interrupt
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_frdy_int	r	1'b0	UART RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto-cleared when data is popped
2	utx_frdy_int	r	1'b1	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)

10.5.10 uart_int_mask

地址：0x40010024

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	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
RSVD	RSVD	RSVD	RSVD	cr_urx_ad5_mask	cr_urx_ads_mask	cr_urx_bcr_mask	cr_urx_lse_mask	cr_urx_fer_mask	cr_urx_pce_mask	cr_urx_rto_mask	cr_urx_frdy_mask	cr_urx_end_mask	cr_urx_end_mask	cr_urx_end_mask	cr_urx_end_mask

位	名称	权限	复位值	描述
31:12	RSVD			
11	cr_urx_ad5_mask	r/w	1'b1	Interrupt mask of urx_ad5_int

位	名称	权限	复位值	描述
10	cr_urx_ads_mask	r/w	1'b1	Interrupt mask of urx_ads_int
9	cr_urx_bcr_mask	r/w	1'b1	Interrupt mask of urx_bcr_int
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_frdy_mask	r/w	1'b1	Interrupt mask of urx_frdy_int
2	cr_utx_frdy_mask	r/w	1'b1	Interrupt mask of utx_frdy_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

10.5.11 uart_int_clear

地址：0x40010028

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
RSVD	RSVD	RSVD	RSVD	cr_urx_ad5_clr	cr_urx_ads_clr	cr_urx_bcr_clr	cr_urx_lse_clr	rsvd	rsvd	cr_urx_pce_clr	cr_urx_rto_clr	rsvd	rsvd	cr_urx_end_clr	cr_utx_end_clr

位	名称	权限	复位值	描述
31:12	RSVD			
11	cr_urx_ad5_clr	w1c	1'b0	Interrupt clear of urx_ad5_int
10	cr_urx_ads_clr	w1c	1'b0	Interrupt clear of urx_ads_int
9	cr_urx_bcr_clr	w1c	1'b0	Interrupt clear of urx_bcr_int
8	cr_urx_lse_clr	w1c	1'b0	Interrupt clear of urx_lse_int
7	rsvd	rsvd	1'b0	
6	rsvd	rsvd	1'b0	

位	名称	权限	复位值	描述
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	rsvd	rsvd	1'b0	
2	rsvd	rsvd	1'b0	
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

10.5.12 uart_int_en

地址：0x4001002c

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	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
RSVD	RSVD	RSVD	RSVD	cr_urx_ad5_en	cr_urx_ads_en	cr_urx_bcr_en	cr_urx_lse_en	cr_urx_fer_en	cr_utx_fer_en	cr_urx_pce_en	cr_urx_rto_en	cr_urx_frdy_en	cr_utx_frdy_en	cr_urx_end_en	cr_utx_end_en

位	名称	权限	复位值	描述
31:12	RSVD			
11	cr_urx_ad5_en	r/w	1'b1	Interrupt enable of urx_ad5_int
10	cr_urx_ads_en	r/w	1'b1	Interrupt enable of urx_ads_int
9	cr_urx_bcr_en	r/w	1'b1	Interrupt enable of urx_bcr_int
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_frdy_en	r/w	1'b1	Interrupt enable of urx_frdy_int
2	cr_utx_frdy_en	r/w	1'b1	Interrupt enable of utx_frdy_int
1	cr_urx_end_en	r/w	1'b1	Interrupt enable of urx_end_int

位	名称	权限	复位值	描述
0	cr_utx_end_en	r/w	1'b1	Interrupt enable of utx_end_int

10.5.13 uart_status

地址: 0x40010030

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	sts_urx_bus_busy	sts_utx_bus_busy

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: Idle 1: Busy
0	sts_utx_bus_busy	r	1'b0	Indicator of UART TX bus busy 0: Idle 1: Busy

10.5.14 sts_urx_abr_prd

地址: 0x40010034

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, baud rate = uart clock / (sts_urx_abr_prd_0x55 + 1)
15:0	sts_urx_abr_prd_start	r	16'd0	Bit period of Auto Baud Rate detection using START bit, baud rate = uart clock / (sts_urx_abr_prd_start + 1)

10.5.15 urx_abr_prd_b01

地址: 0x40010038

sts_urx_abr_prd_bit1

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_bit0

位	名称	权限	复位值	描述
31:16	sts_urx_abr_prd_bit1	r	16'd0	Bit period of Auto Baud Rate detection - bit[1]
15:0	sts_urx_abr_prd_bit0	r	16'd0	Bit period of Auto Baud Rate detection - bit[0]

10.5.16 urx_abr_prd_b23

地址: 0x4001003c

sts_urx_abr_prd_bit3

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_bit2

位	名称	权限	复位值	描述
31:16	sts_urx_abr_prd_bit3	r	16'd0	Bit period of Auto Baud Rate detection - bit[3]
15:0	sts_urx_abr_prd_bit2	r	16'd0	Bit period of Auto Baud Rate detection - bit[2]

10.5.17 urx_abr_prd_b45

地址: 0x40010040

sts_urx_abr_prd_bit5

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_bit4

位	名称	权限	复位值	描述
31:16	sts_urx_abr_prd_bit5	r	16'd0	Bit period of Auto Baud Rate detection - bit[5]
15:0	sts_urx_abr_prd_bit4	r	16'd0	Bit period of Auto Baud Rate detection - bit[4]

10.5.18 urx_abr_prd_b67

地址: 0x40010044

sts_urx_abr_prd_bit7

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_bit6

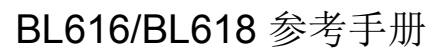
位	名称	权限	复位值	描述
31:16	sts_urx_abr_prd_bit7	r	16'd0	Bit period of Auto Baud Rate detection - bit[7]
15:0	sts_urx_abr_prd_bit6	r	16'd0	Bit period of Auto Baud Rate detection - bit[6]

10.5.19 urx_abr_pw_tol

地址: 0x40010048

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD

cr_urx_abr_pw_tol



10.5.20 urx_bcr_int_cfg

sts_urx_bcr_count

cr_urx_bcr_value

10.5.21 utx_rs485_cfg

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

BL616/BL618 参考手册

位	名称	权限	复位值	描述
1	cr_utx_rs485_pol	r/w	1'b1	DE pin polarity in RS-485 transceiver mode 1'b0: DE is active-low 1'b1: DE is active-high
0	cr_utx_rs485_en	r/w	1'b0	Enable signal for RS-485 transceiver mode 1'b0: Disabled, normal UART 1'b1: Enabled, IO is connected to RS-485 transceiver and RTS_O becomes DE function

10.5.22 uart_fifo_config_0

地址: 0x40010080

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	rx_fifo_underflow	rx_fifo_overflow	tx_fifo_underflow	tx_fifo_overflow	rx_fifo_clr	tx_fifo_clr	uart_dma_rx_en	uart_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, RX FIFO will be empty when write 1 to this bit
2	tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO, TX FIFO will be empty when write 1 to this bit
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

10.5.23 uart_fifo_config_1

地址：0x40010084

RSVD			RSVD			RSVD			rx_fifo_th								RSVD			RSVD			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																		
RSVD			RSVD			rx_fifo_cnt								RSVD			RSVD			tx_fifo_cnt													

位	名称	权限	复位值	描述
31:29	RSVD			
28:24	rx_fifo_th	r/w	5'd0	RX FIFO threshold, dma_rx_req and rx_fifo_int will not be asserted if rx_fifo_cnt is less than this value
23:21	RSVD			
20:16	tx_fifo_th	r/w	5'd0	TX FIFO threshold, dma_tx_req and tx_fifo_int will not be asserted if tx_fifo_cnt is less than this value
15:14	RSVD			
13:8	rx_fifo_cnt	r	6'd0	RX FIFO available count, means byte count of data received in RX FIFO
7:6	RSVD			
5:0	tx_fifo_cnt	r	6'd32	TX FIFO available count, means empty space remained in TX FIFO

10.5.24 uart_fifo_wdata

地址：0x40010088

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		
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	uart_fifo_wdata								RSVD	RSVD

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	uart_fifo_wdata	w	x	TX FIFO, size is 32*1=32-byte

10.5.25 uart_fifo_rdata

地址：0x4001008c

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
uart_fifo_rdata															

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	uart_fifo_rdata	r	8'h0	RX FIFO, size is 32*1=32-byte

11.1 简介

I2C (Inter-Integrated Circuit) 是一种串行通讯总线，使用多主从架构，用来连接低速外围设备。每个器件都有一个唯一的地址识别，并且都可以作为发送器或接收器。如果有多个主机同时操作一个从机，数据传输可以通过冲突检测和仲裁机制防止数据被破坏。BL616/BL618 包含 2 组 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`，支持中断控制，可搭配 DMA 使用提高效率，起始、结束和数据发送阶段的电平持续时间可分段灵活控制。

11.2 主要特征

- 支持主机模式
- 支持多主机模式和仲裁功能
- 起始、结束和数据发送阶段的电平持续时间可分段灵活控制
- 支持 7 位地址格式和 10 位地址格式
- 支持 DMA 传输模式
- 支持多种中断机制

11.3 功能描述

11.3.1 基本架构

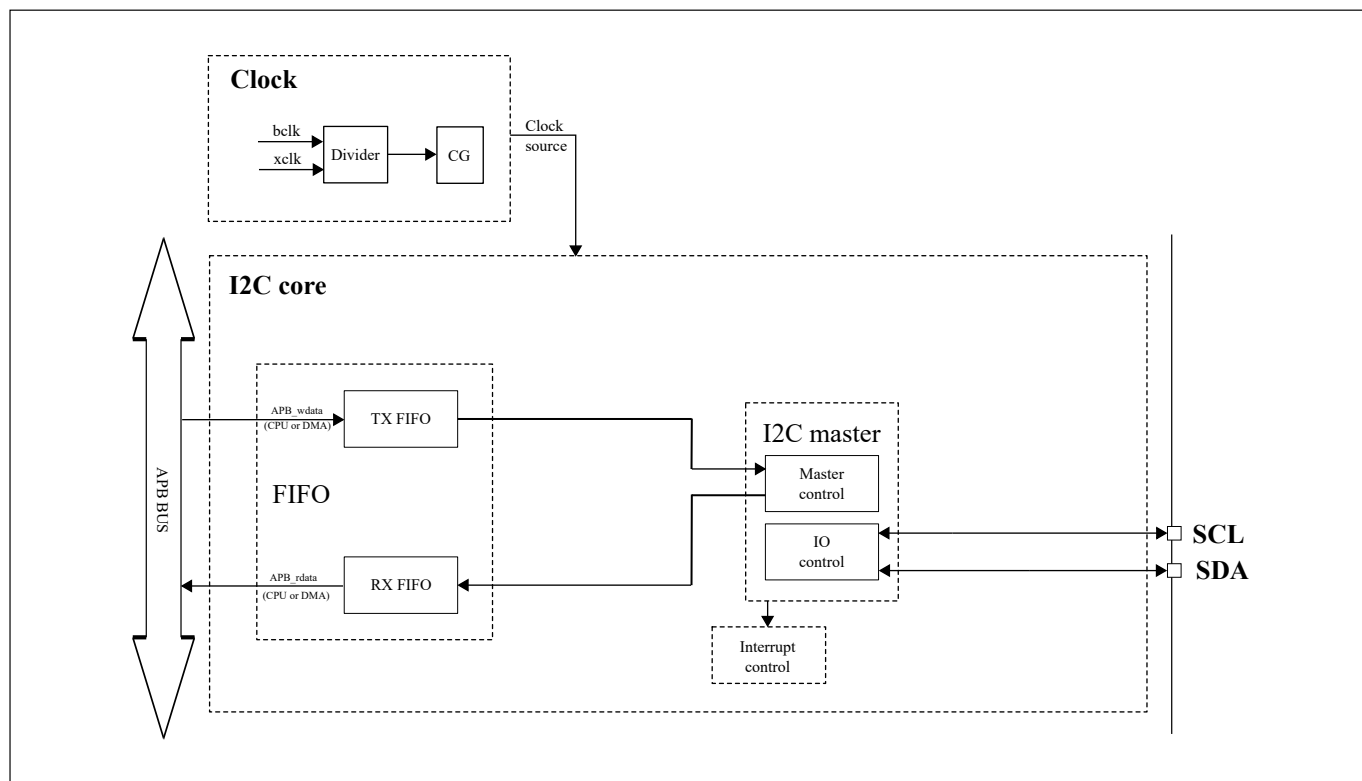


图 11.1: I2C 基本架构图

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

11.3.2 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，总线在起始条件后处于总线忙的状态，在停止条件后至下一次起始条件前处于空闲状态。

起始条件: SCL 为高电平时 SDA 产生由高至低的电平转换;

停止条件: SCL 为高电平时 SDA 产生由低至高的电平转换。

波形示意图如下:

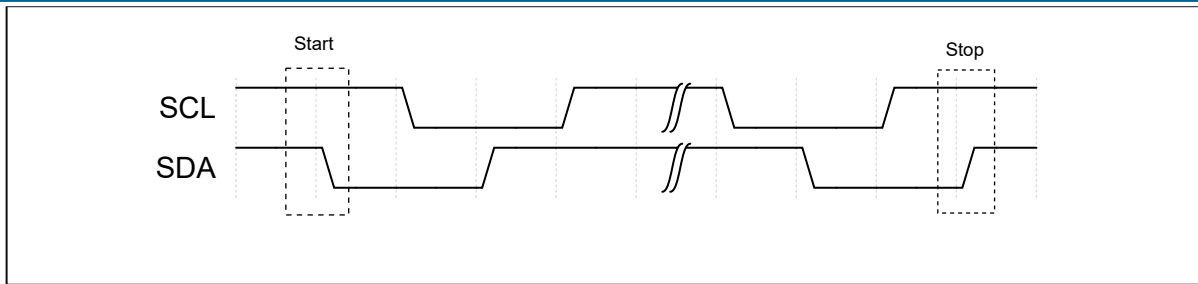


图 11.2: I2C 起始和停止条件

11.3.3 数据传输格式

1. 7 位寻址模式

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第 1 个字节的第 8 位（即方向位）控制的，为 0 表示数据由主机发送；为 1 表示数据由主机接收。在方向位之后是应答位 (ACK)，由从机发出应答（将信号拉低），主机在收到应答后，开始传输指定长度的数据。数据传输完成后，主机将发出停止信号，数据传输格式如下：

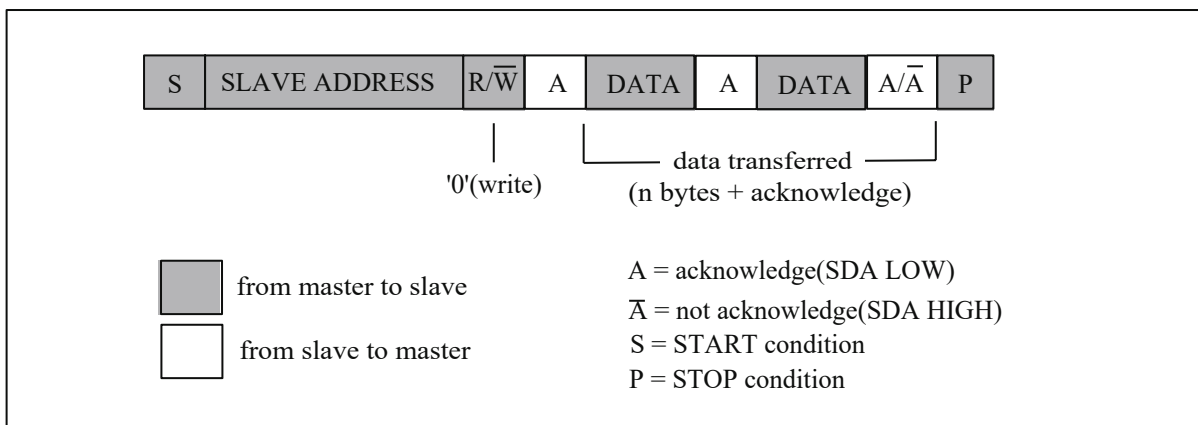


图 11.3: 主发送和从接收的数据格式

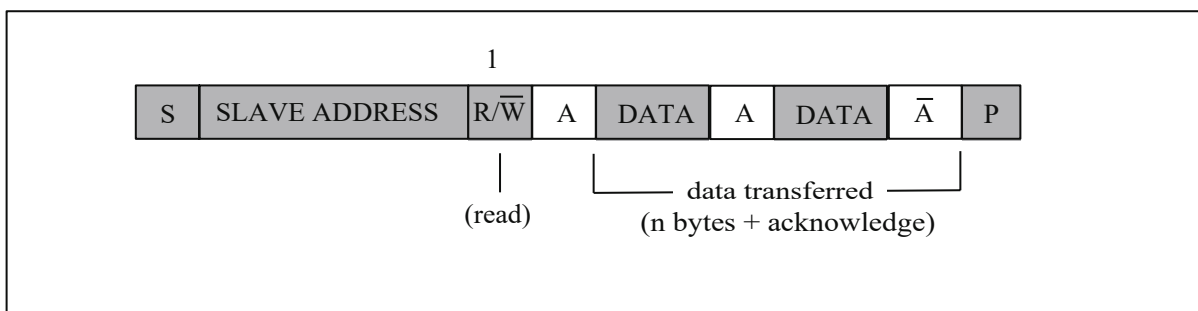


图 11.4: 主接收和从发送的数据格式

主发送和从接收的时序

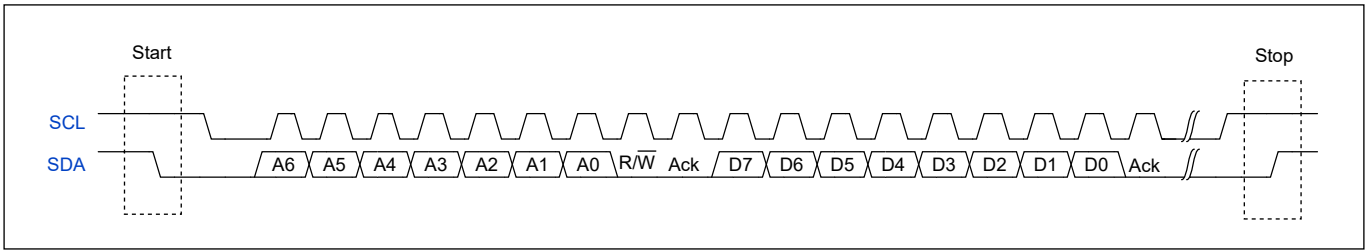


图 11.5: 主发送和从接收的时序

主接收和从发送的时序

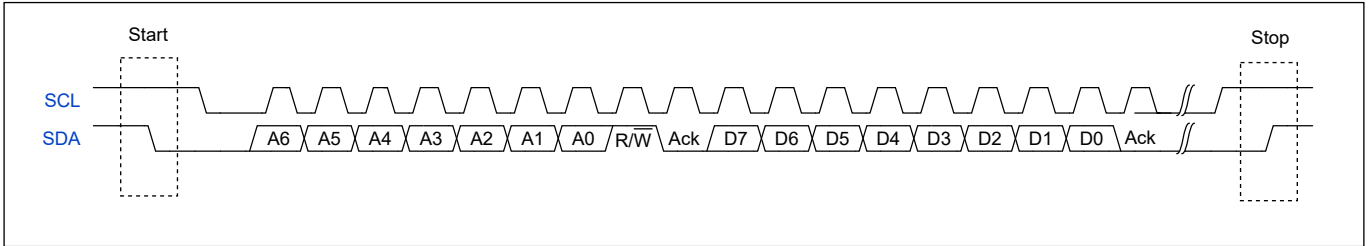


图 11.6: 主接收和从发送的时序

2. 10 位寻址模式

使用 10 位寻址模式时，需要将寄存器 `i2c_config` 中的 `cr_i2c_10b_addr_en` 置 1。

10bit 的从机地址由开始条件 (S) 或重复开始条件 (Sr) 后的两个字节组成。第一个字节的前 7 位是 1111 0XX，XX 是 10 bit 地址的最高两位。第一个字节的第 8 bit 是读写位，决定传输方向。第二个字节是 10 bit 地址中剩下的低 8 位。数据传输格式如下：

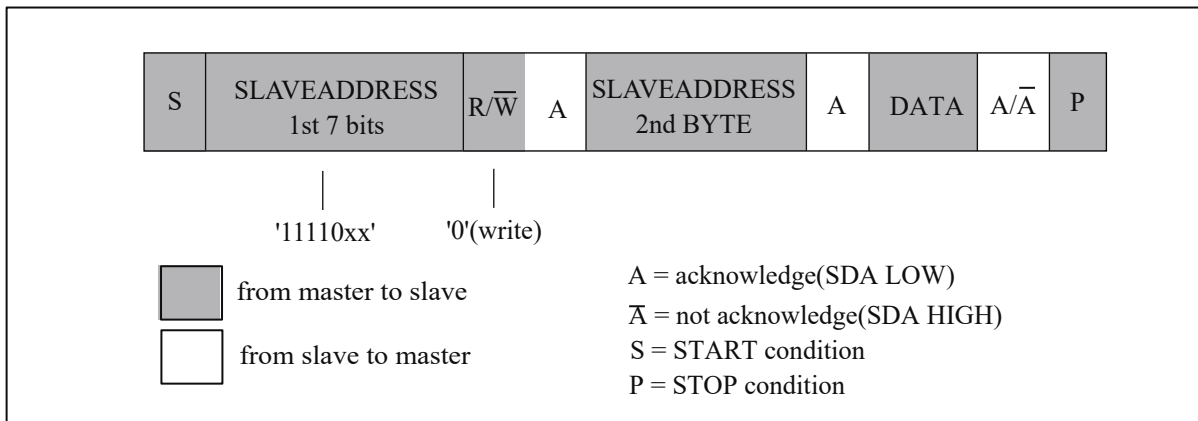


图 11.7: 主发送和从接收的数据格式 (10bit 从地址)

当从机接收到 10 bit 地址时，会从第一个字节 (1111 0XX) 开始匹配，并检查第 8bit (读写位) 是否为 0 (写)。如果第一个字节中 XX 的值与从机 10 bit 地址的最高两位相同，则第一个字节匹配通过，从机将给出应答 A。如果有多个从机设备连接到总线上，则可能有多个设备匹配并产生应答 A。接下来所有从机开始匹配第二个字节 (XXXX XXXX)，其中只会有一个从机的 10 bit 地址低八位与第二个字节完全相同，该从机将给出应答 A。被主机寻址匹配的从机将保持被

寻址的状态，直到接收到终止条件或者是重复开始条件。

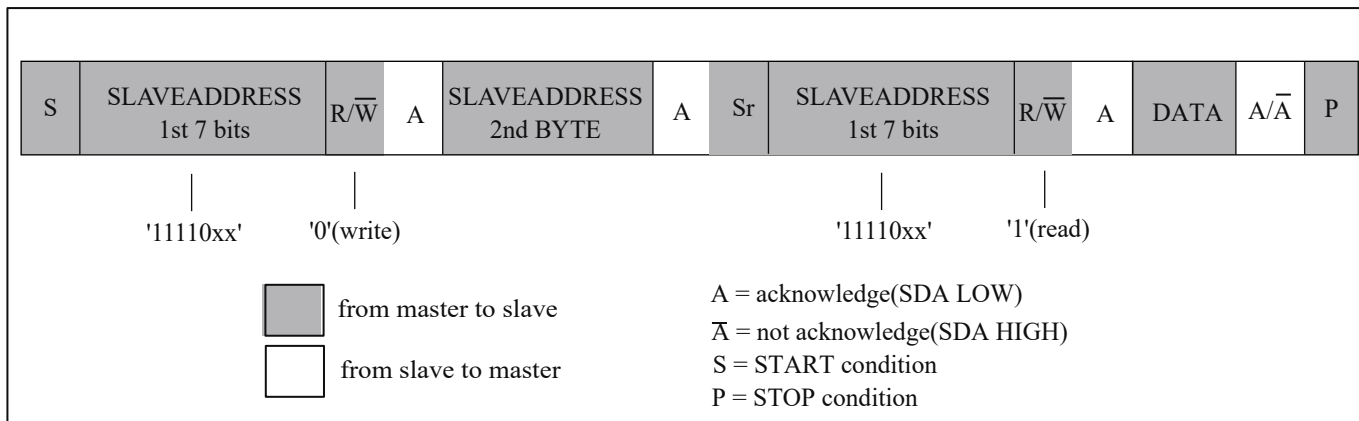


图 11.8: 主接收和从发送的数据格式 (10bit 从地址)

在第二个应答 **A** 之前，处理过程与上面的主-发送器寻址从-接收器一致。在重复开始条件 (**Sr**) 之后，匹配的从机将保持被寻址的状态。从机会检查 **Sr** 之后的第一个字节的前 7bit 是否为 1111 0XX，并检查第 8bit 是否为 1 (读)。如果匹配，从机就认定它作为发送器寻址成功并产生应答 **A**。从-发送器会保持被寻址的状态，直到接收到终止条件 (**P**) 或者重复开始条件 (**Sr**)。接收到重复开始条件 (**Sr**) 后，所有的从机会与 11110XX 匹配并测试第八位 (读写位)。然而它们中除了之前已经匹配的从机，都不会被寻址到。因为对于 10bit 地址的设备，读写位是 1，不满足第一个字节读写位为 0 的匹配条件；或者对于 7bit 地址的设备，1111 0XX 的从机地址不匹配。

11.3.4 仲裁

当 I2C 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权，等到总线再次空闲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 的电平状态是否与自己想发送的数据电平状态相符，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对总线的控制权，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：

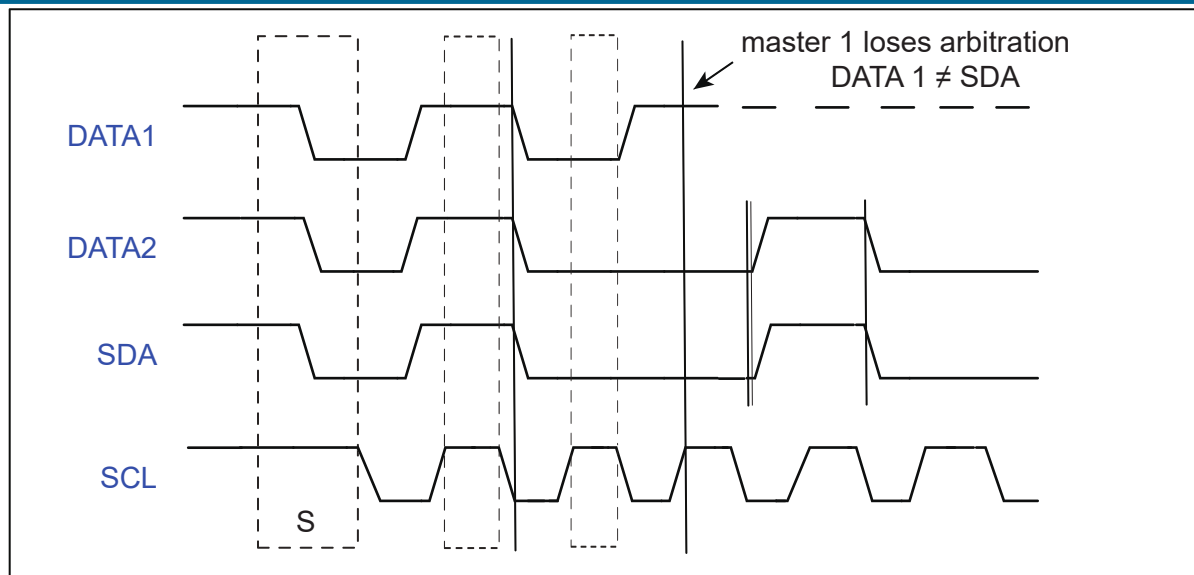


图 11.9: 同时传输数据波形示意图

11.4 I2C 时钟设定

I2C 的时钟可由 `bclk`(bus clock) 和 `xclk` 而来，可以在其基础上做分频处理。开始条件的持续时间、每一位数据的持续时间以及结束条件的持续时间分别由寄存器 `i2c_prd_start`、`i2c_prd_data` 和 `i2c_prd_stop` 进行设置。其中每一种持续时间又可以细分为 4 个阶段，每个阶段的采样个数在寄存器中用单独一个字节来控制（实际值为寄存器值加 1），数据部分的 4 个阶段设置值共同决定了 `i2c clock` 的分频系数。如下图所示，假设 I2C 时钟源选为 80M 的 `bclk`，寄存器 `i2c_prd_data` 设置为 0x09070b09，即图中的第二个 0 为 $0x09+1=0x0a$ ，第二个 1 为 $0x07+1=0x08$ ，第二个 2 为 $0x0b+1=0x0c$ ，第二个 3 为 $0x09+1=0x0a$ 。则 I2C 的时钟频率为 $80\text{MHz}/(0x0a+0x08+0x0c+0x0a) = 2\text{MHz}$ 。同理，第一个 0、1、2 和 3 由寄存器 `i2c_prd_start` 设置，决定了开始条件的持续时间；第三个 0、1、2 和 3 由寄存器 `i2c_prd_stop` 设置，决定了结束条件的持续时间。

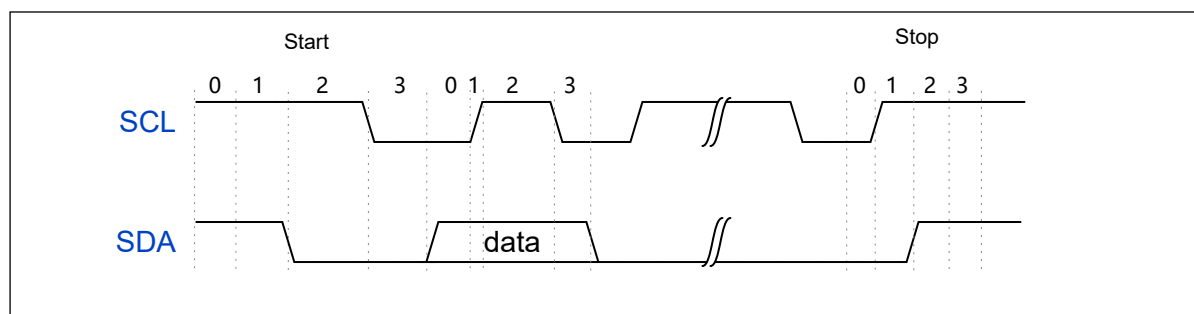


图 11.1: I2C 时钟设定

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 `i2c_config` 中的 `cr_i2c_pkt_dir` 用于控制发送或接收, 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, 该地址通常是 7 bit 的, 将该 7 bit 值写入寄存器 `i2c_config` 中的 `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

通过设置从设备寄存器地址, 可以对从设备的寄存器进行读写操作。使用时需要将寄存器地址写入寄存器 `i2c_sub_addr`, 同时将寄存器 `i2c_config` 中的 `cr_i2c_sub_addr_en` 置 1。如果将寄存器 `i2c_config` 中的 `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

可以通过寄存器 `i2c_config` 中的 `cr_i2c_sub_addr_bc` 设置从设备寄存器地址长度 (从设备寄存器地址长度为写入寄存器的值 + 1), 详细配置参考寄存器描述。

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到的数据。当 I2C 发送数据时, 需要将数据依次以 word 为单位写入寄存器 `i2c_fifo_wdata` 中。当 I2C 接收数据时, 需要依次以 word 为单位从寄存器 `i2c_fifo_rdata` 中将数据读出来。

11.5.7 数据长度

可以通过寄存器 `i2c_config` 中的 `cr_i2c_pkt_len` 设置发送数据长度（发送数据长度为写入寄存器的值 + 1），最大发送长度为 256 字节。

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 `i2c_config` 中的 `cr_i2c_m_en` 置 1，就自动启动 I2C 发送流程。

当读写标志位配置为 0 时，I2C 发送数据，以发送 2 字节为例，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，以接收 2 字节为例，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 分为 RX FIFO 和 TX FIFO，大小各为 2 个 word。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 DMA 功能

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. 配置读写标志位为 0
2. 配置从设备地址
3. 配置从设备寄存器地址
4. 配置从设备寄存器地址长度
5. 数据长度
6. 使能信号寄存器置 1
7. 配置 DMA transfer size
8. 配置 DMA 源地址 transfer width
9. 配置 DMA 目的地址 transfer width(需要注意 I2C 使用 DMA 功能时, 目的地址 transfer width 需要设置为 32bits, 以 word 对齐使用)
10. 配置 DMA 源地址为存储发送数据的内存地址
11. 配置 DMA 目的地址为 I2C TX FIFO 地址, 即 `i2c_fifo_wdata`
12. 使能 DMA

11.7.2 DMA 接收流程

1. 配置读写标志位为 1
2. 配置从设备地址
3. 配置从设备寄存器地址
4. 配置从设备寄存器地址长度
5. 数据长度
6. 使能信号寄存器置 1
7. 配置 DMA transfer size
8. 配置 DMA 源地址 transfer width(需要注意 I2C 使用 DMA 功能时, 源地址 transfer width 需要设置为 32bits, 以 word 对齐使用)
9. 配置 DMA 目的地址 transfer width
10. 配置 DMA 源地址为 I2C RX FIFO 地址, 即 i2c_fifo_rdata
11. 配置 DMA 目的地址为存储接收数据的内存地址
12. 使能 DMA

11.8 中断

I2C 包括以下几种中断:

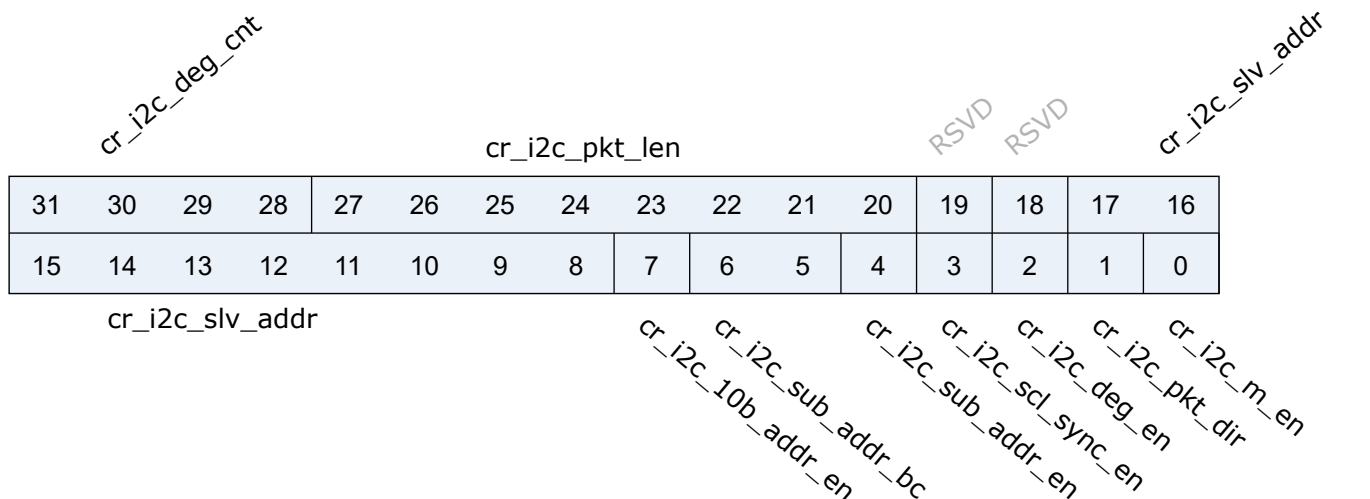
- I2C_TRANS_END_INT * I2C 传输结束中断, 当 I2C 完成一次传输时产生该中断
- I2C_TX_FIFO_READY_INT * 当 i2c_fifo_config_1 中的 tx_fifo_cnt 大于 tx_fifo_th 时, 产生 TX FIFO 请求中断, 当条件不满足时该中断标志会自动清除
- I2C_RX_FIFO_READY_INT * 当 i2c_fifo_config_1 中的 rx_fifo_cnt 大于 rx_fifo_th 时, 产生 RX FIFO 请求中断, 当条件不满足时该中断标志会自动清除
- I2C_NACK_RECV_INT * 当 I2C 模块检测到 NACK 状态时, 产生 NACK 中断
- I2C_ARB_LOST_INT * I2C 仲裁丢失中断
- I2C_FIFO_ERR_INT * 当 TX/RX FIFO 发生 overflow 或 underflow 时, 产生 FIFO ERROR 中断

11.9 寄存器描述

名称	描述
i2c_config	Master configure
i2c_int_sts	Interrupt configure and status
i2c_sub_addr	Sub-address setting
i2c_bus_busy	Bus busy status
i2c_prd_start	Start period setting
i2c_prd_stop	Stop period setting
i2c_prd_data	Data period setting
i2c_fifo_config_0	FIFO status and DMA mode
i2c_fifo_config_1	FIFO threshold and available count
i2c_fifo_wdata	TX FIFO
i2c_fifo_rdata	RX FIFO

11.9.1 i2c_config

地址：0x40018000



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

位	名称	权限	复位值	描述
27:20	cr_i2c_pkt_len	r/w	8'd0	Packet length (unit: byte)
19:18	RSVD			
17:8	cr_i2c_slv_addr	r/w	10'h0	Slave address for I2C transaction (target address)
7	cr_i2c_10b_addr_en	r/w	1'b0	Slave address 10-bit mode enable
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: 0x40018004

RSVD	RSVD	cr_i2c_fer_en	cr_i2c_arb_en	cr_i2c_nak_en	cr_i2c_rxf_en	cr_i2c_txf_en	cr_i2c_end_en	RSVD	RSVD	rsvd	cr_i2c_arb_clr	cr_i2c_nak_clr	rsvd	rsvd	cr_i2c_end_clr
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	cr_i2c_fer_mask	cr_i2c_arb_mask	cr_i2c_nak_mask	cr_i2c_rxf_mask	cr_i2c_txf_mask	cr_i2c_end_mask	RSVD	RSVD	i2c_fer_int	i2c_arb_int	i2c_nak_int	i2c_rxf_int	i2c_txf_int	i2c_end_int

位	名称	权限	复位值	描述
31:30	RSVD			

位	名称	权限	复位值	描述
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
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'b1	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：0x40018008

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：0x4001800c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	cr_i2c_bus_busy_clr	sts_i2c_bus_busy

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 0: Idle 1: Busy

11.9.5 i2c_prd_start

地址：0x40018010

cr_i2c_prd_s_ph_3								cr_i2c_prd_s_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
cr_i2c_prd_s_ph_1								cr_i2c_prd_s_ph_0							

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 (unit: I2C source clock period)
23:16	cr_i2c_prd_s_ph_2	r/w	8'd15	Length of START condition phase 2 (unit: I2C source clock period)
15:8	cr_i2c_prd_s_ph_1	r/w	8'd15	Length of START condition phase 1 (unit: I2C source clock period)
7:0	cr_i2c_prd_s_ph_0	r/w	8'd15	Length of START condition phase 0 (unit: I2C source clock period)

11.9.6 i2c_prd_stop

地址：0x40018014

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
cr_i2c_prd_p_ph_1								cr_i2c_prd_p_ph_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 (unit: I2C source clock period)
23:16	cr_i2c_prd_p_ph_2	r/w	8'd15	Length of STOP condition phase 2 (unit: I2C source clock period)
15:8	cr_i2c_prd_p_ph_1	r/w	8'd15	Length of STOP condition phase 1 (unit: I2C source clock period)
7:0	cr_i2c_prd_p_ph_0	r/w	8'd15	Length of STOP condition phase 0 (unit: I2C source clock period)

11.9.7 i2c_prd_data

地址：0x40018018

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
cr_i2c_prd_d_ph_1								cr_i2c_prd_d_ph_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 (unit: I2C source clock period)
23:16	cr_i2c_prd_d_ph_2	r/w	8'd15	Length of DATA phase 2 (unit: I2C source clock period)
15:8	cr_i2c_prd_d_ph_1	r/w	8'd15	Length of DATA phase 1 (unit: I2C source clock period) 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 (unit: I2C source clock period)

11.9.8 i2c_fifo_config_0

地址：0x40018080

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	rx_fifo_underflow	rx_fifo_overflow	tx_fifo_underflow	tx_fifo_overflow	rx_fifo_clr	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

位	名称	权限	复位值	描述
3	rx_fifo_clr	w1c	1'b0	Clear signal of RX FIFO, RX FIFO will be empty when write 1 to this bit
2	tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO, TX FIFO will be empty when write 1 to this bit
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: 0x40018084

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 rx_fifo_th RSVD RSVD RSVD RSVD RSVD RSVD RSVD tx_fifo_th

RSVD RSVD RSVD RSVD RSVD RSVD rx_fifo_cnt RSVD RSVD RSVD RSVD RSVD RSVD tx_fifo_cnt

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 rx_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, means count of data received in RX FIFO
7:2	RSVD			
1:0	tx_fifo_cnt	r	2'd2	TX FIFO available count, means empty space remained in TX FIFO

11.9.10 i2c_fifo_wdata

地址: 0x40018088

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	TX FIFO, size is 4*2 = 8-byte

11.9.11 i2c_fifo_rdata

地址: 0x4001808c

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	RX FIFO, size is 4*2 = 8-byte

12.1 简介

脉冲宽度调制（Pulse width modulation，简称 PWM）是一种利用微处理器的数字信号对模拟电路进行有效控制的技术，广泛应用于测量、通信和功率控制与变换的许多领域中。

12.2 主要特征

- 支持 1 组 PWM 信号生成，包含 4 通道 PWM 信号输出，每通道可以设置为 2 路互补 PWM
- 三种时钟源可选择（bclk、xclk、f32k），搭配 16-bit 时钟分频器，最大 65535 分频
- 每通道 PWM 都有双门限值设定，可以设定不同的占空比和相位，增加脉冲弹性
- 每通道 PWM 都有独立的死区时间设定
- 每路 PWM 输出引脚都可以设定不同的有效电平
- 每路 PWM 都有独立的连接开关用来选择是否与内部计数器相连，并可设定不连接时的默认输出电平
- 刹车信号可以将 PWM 输出电平置于预先设定的状态
- 多达 9 种可用于触发 ADC 转换的触发源
- 支持 10 种中断

12.3 功能描述

12.3.1 时钟与分频器

PWM 计数器时钟来源有三种选择，可通过 pwm_mc0_config0 寄存器中的 reg_clk_sel 域设定，时钟源配置如下表所示：

表 12.1: 时钟源

reg_clk_sel	PWM 时钟源
0	xclk
1	bclk
others	f32k

计数器有 16-bit 分频器，可通过 APB 将选择到的时钟进行分频，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数一的动作。

12.3.2 有效电平

不同的应用场景所需要的有效电平状态不一样，一部分为低电平有效，一部分为高电平有效，每路 PWM 输出的有效电平可独立设定。

当 pwm_mc0_config1 寄存器中的位 <pwm_chy_ppl> 设置为 1 时表示 pwm 的通道 y (y=0,1,2,3) 的正向通道高电平有效，即 PWM 模块输出逻辑 1 时其对应的引脚输出高电平，输出逻辑 0 时其对应的引脚输出低电平。如果 <pwm_chy_ppl> 设置为 0 则表示低电平有效，即 PWM 模块输出逻辑 1 时其对应的引脚输出低电平，输出逻辑 0 时其对应的引脚输出高电平。互补通道的设置位为 <pwm_chy_npl>，其设置规则与正向通道相同。

12.3.3 脉冲产生原理

当 pwm_mc0_config1 寄存器中的位 <pwm_chy_pen> 设置为 0 时，表示 pwm 的通道 y (y=0,1,2,3) 的正向通道输出为一个确定的逻辑状态，此时状态由位 <pwm_chy_psi> 确定，当 <pwm_chy_psi> 为 0 时对应的正向通道引脚输出逻辑 0，当 <pwm_chy_psi> 为 1 时对应的正向通道引脚输出逻辑 1，实际输出电平需要和 <pwm_chy_ppl> 共同确定。互补通道的设置位为 <pwm_chy_nen>，其设置规则与正向通道相同。

当 pwm_mc0_config1 寄存器中的位 <pwm_chy_pen> (<pwm_chy_nen>) 设置为 1 时表示 pwm 的通道 y (y=0,1,2,3) 的正向（互补）通道输出由内部计数器与两个阈值的比较结果确定，当计数器的值在两个阈值中间时，正向通道输出逻辑 1，互补通道输出逻辑 0，当计数器的值在两个阈值之外时，正向通道输出逻辑 0，互补通道输出逻辑 1。具体的波形如下图所示：

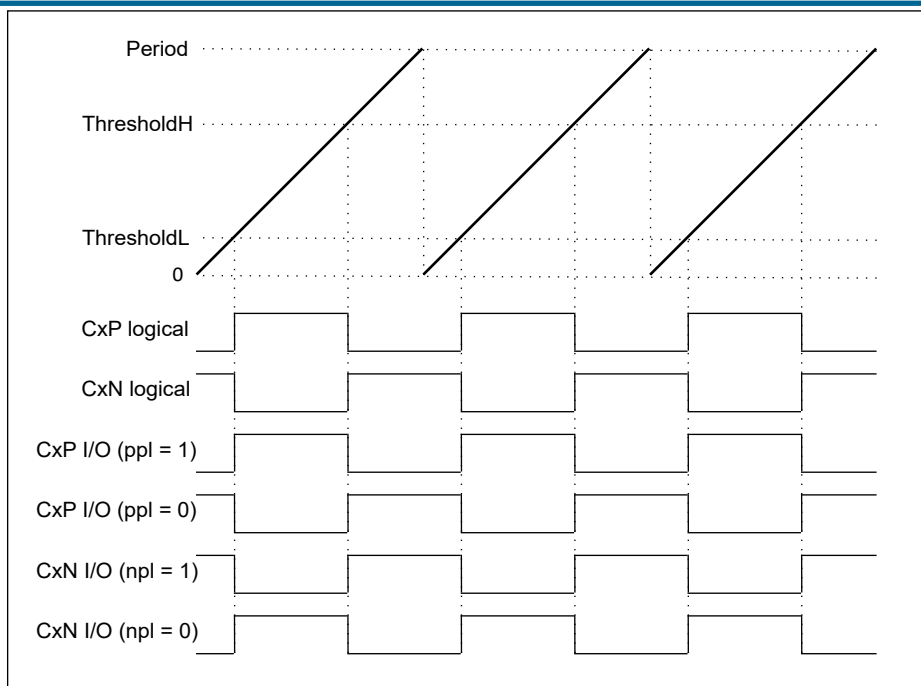


图 12.1: PWM 输出波形图

12.3.4 刹车

当 `pwm_mc0_config0` 的位 `<pwm_sw_break_en>` 为 1 时即产生刹车信号，此时 PWM 输出的电平都会被修改为预设定的状态。预设定状态由 `pwm_mc0_config1` 寄存器中的位 `<pwm_chy_pbs>`、`<pwm_chy_nbs>` ($y=0,1,2,3$) 设定，当该位设置为 1 时，刹车信号产生后该路 PWM 输出逻辑 1，当该位设置为 0 时，刹车信号产生后该路 PWM 输出逻辑 0。`<pwm_sw_break_en>` 为 0 时即退出刹车状态，PWM 恢复之前的运行方式。刹车信号不会触发中断。

12.3.5 死区

PWM 每个通道都可以独立设置不同的死区时间，当死区设定值不为 0 时，PWM 的正向通道在与 `ThresholdL` 匹配时会延迟改变电平的状态，在与 `ThresholdH` 匹配时立即改变电平状态。PWM 的互补通道在与 `ThresholdL` 匹配时会立即改变电平状态，在与 `ThresholdH` 匹配时会延迟改变电平的状态。死区长度由 `pwm_mc0_dead_time` 寄存器设置，死区长度的计算如下表所示：

表 12.2: 死区长度计算公式

<code>pwm_chy_dtg</code>	死区长度
0xxxxxxx	<code>pwm_chy_dtg[7:0]</code>
10xxxxxx	$(64 + \text{pwm_chy_dtg}[5:0]) * 2$
110xxxxx	$(32 + \text{pwm_chy_dtg}[4:0]) * 8$
111xxxxx	$(32 + \text{pwm_chy_dtg}[4:0]) * 16$

对 PWM 插入死区后的输出波形如下图所示：

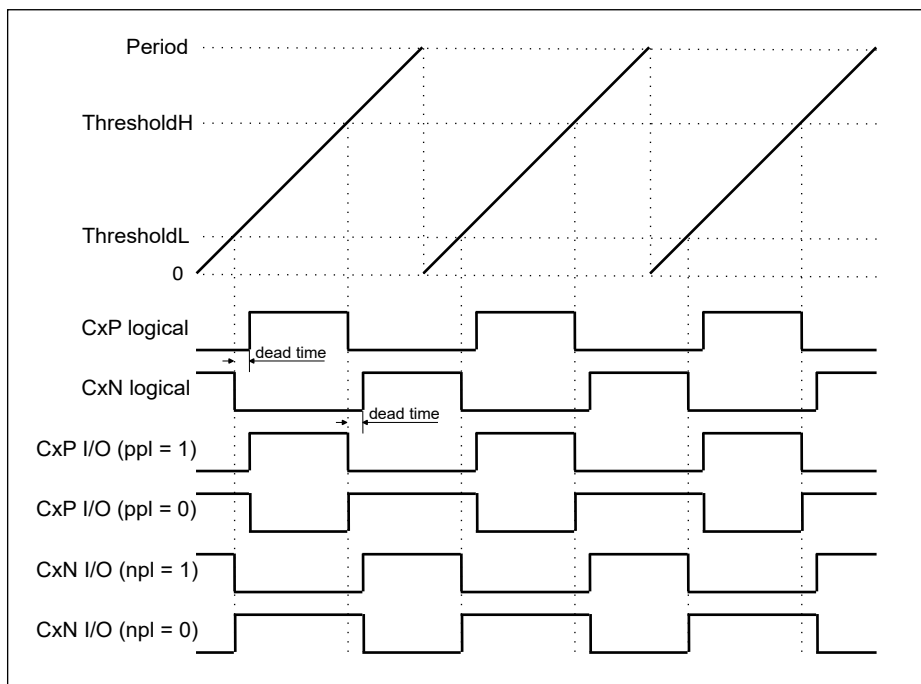


图 12.2: PWM 插入死区输出波形图

12.3.6 周期与占空比计算

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 `pwm_mc0_config0` 中的域 `pwm_clk_div` 进行设置，用于对 PWM 的源时钟进行分频。该域的值即为分频系数，需要注意的是该域为 0 或 1 时都表示 1 分频，最大为 65535 表示 65535 分频。

时钟持续周期由寄存器 `pwm_mc0_period` 中的域 `pwm_period` 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。

PWM 周期 = PWM 源时钟 / `pwm_clk_div` / `pwm_period`

12.3.7 PWM 中断

PWM 共可以产生 10 种中断，分别描述如下：

- 通道 0 阈值 ThresholdL 中断
 - 当 PWM 计数值等于 `pwm_mc0_ch0_thre` 寄存器中的域 `pwm_ch0_threL` 的值时产生该中断。
- 通道 0 阈值 ThresholdH 中断
 - 当 PWM 计数值等于 `pwm_mc0_ch0_thre` 寄存器中的域 `pwm_ch0_threH` 的值时产生该中断。
- 通道 1 阈值 ThresholdL 中断
 - 当 PWM 计数值等于 `pwm_mc0_ch1_thre` 寄存器中的域 `pwm_ch1_threL` 的值时产生该中断。

- 通道 1 阈值 ThresholdH 中断
 - 当 PWM 计数值等于 `pwm_mc0_ch1_thre` 寄存器中的域 `pwm_ch1_threH` 的值时产生该中断。
- 通道 2 阈值 ThresholdL 中断
 - 当 PWM 计数值等于 `pwm_mc0_ch2_thre` 寄存器中的域 `pwm_ch2_threL` 的值时产生该中断。
- 通道 2 阈值 ThresholdH 中断
 - 当 PWM 计数值等于 `pwm_mc0_ch2_thre` 寄存器中的域 `pwm_ch2_threH` 的值时产生该中断。
- 通道 3 阈值 ThresholdL 中断
 - 当 PWM 计数值等于 `pwm_mc0_ch3_thre` 寄存器中的域 `pwm_ch3_threL` 的值时产生该中断。
- 通道 3 阈值 ThresholdH 中断
 - 当 PWM 计数值等于 `pwm_mc0_ch3_thre` 寄存器中的域 `pwm_ch3_threH` 的值时产生该中断。
- 周期结束中断
 - 当 PWM 计数值等于寄存器 `pwm_mc0_period` 中的域 `pwm_period` 的值时产生该中断，表示一个 PWM 周期结束。
- 周期重复中断
 - 当 PWM 的周期数达到寄存器 `pwm_mc0_period` 中的域 `pwm_int_period_cnt` 设定的值时产生该中断。在步进电机的应用场景中可以使用该中断，设定 `pwm_int_period_cnt` 即可让步进电机在旋转特定角度后产生中断。

12.3.8 ADC 联动

当计数器与门限值发生匹配或者计数值达到周期数时都可以产生内部触发 ADC 启动转换的信号。具体的触发源由 `pwm_mcx_config0` 寄存器的位 `<pwm_adc_trg_src>` 设置。这种特性在定时采样的应用场景中较为常用，以下举例说明。

应用场景：在 BLDC 的应用中，有这样一种应用需求，在用 PWM 控制电机的转速的同时，还需要检测流过线圈的电流。在一个 PWM 周期内，PWM 控制功率器件导通后，经过一段特定的时间，电流达到稳定，此时需要采样电流值，这意味着触发 ADC 转换的时刻点与 PWM 之间有严格的相位差。比如 PWM 的通道 0 用于驱动电机的其中一相，需要产生 10KHz 占空比为 20% 的方波，并且需要在方波高电平的中间时刻点执行 ADC 采样，则该 PWM 周期为 100us，当时钟源为 1MHz 时，周期计数值为 100，可以将通道 1 的 ThresholdL 和 ThresholdH 分别设置为 0 和 20，此时计数器在 0~20 之间时，PWM 输出高电平，否则输出低电平。将通道 2 的 ThresholdL 设置为 10，且将 `pwm_mc0_config0` 寄存器中的 `<pwm_adc_trg_src>` 设置为 4，即 `pwm_ch2l_int` 可以触发 ADC 转换，则计数器数到 10 时（通道 1 产生的高电平中间时刻）会启动 ADC 开始采样转换，这样可以保证每次采样都满足精确的时间要求，且不需要 CPU 干预，从而提高了性能。

12.4 寄存器描述

名称	描述
pwm_int_config	PWM interrupt config register
pwm_mc0_config0	PWM config0 register
pwm_mc0_config1	PWM config1 register
pwm_mc0_period	PWM period register
pwm_mc0_dead_time	PWM dead time register
pwm_mc0_ch0_thre	PWM channel0 threshold register
pwm_mc0_ch1_thre	PWM channel1 threshold register
pwm_mc0_ch2_thre	PWM channel2 threshold register
pwm_mc0_ch3_thre	PWM channel3 threshold register
pwm_mc0_int_sts	PWM interrupt status register
pwm_mc0_int_mask	PWM interrupt mask register
pwm_mc0_int_clear	PWM interrupt clear register
pwm_mc0_int_en	PWM interrupt enable register

12.4.1 pwm_int_config

地址：0x2000a400

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	pwm0_int_clr	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	pwm0_int_sts

位	名称	权限	复位值	描述
31:9	RSVD			
8	pwm0_int_clr	w1c	1'b0	PWM 0 interrupt clear (clear pwm_mc0_int_sts[10:0] all)

位	名称	权限	复位值	描述
7:1	RSVD			
0	pwm0_int_sts	r	1'b0	PWM 0 interrupt status (Check pwm_mc0_int_sts for detailed interrupt status)

12.4.2 pwm_mc0_config0

地址: 0x2000a440

reg_clk_sel		pwm_sts_stop		pwm_stop_mode		pwm_stop_en		RSVD		RSVD		pwm_sw_break_en		pwm_adc_trg_src		pwm_stop_on_rept		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								
pwm_clk_div																							

位	名称	权限	复位值	描述
31:30	reg_clk_sel	r/w	2'd0	PWM clock source select 2'd0: xclk 2'd1: bclk others: f32k_clk
29	pwm_sts_stop	r	1'b0	PWM stop status 0: PWM run status 1: PWM stop status
28	pwm_stop_mode	r/w	1'b1	PWM stop mode, 0: abrupt 1: graceful
27	pwm_stop_en	r/w	1'b0	PWM stop enable 0: start PWM 1: stop PWM
26:25	RSVD			
24	pwm_sw_break_en	r/w	1'b0	PWM break enable 0: Disabled, normal operation 1: Enabled, PWM output will be determined by CxPBS/CxNBS

位	名称	权限	复位值	描述
23:20	pwm_adc_trg_src	r/w	4'hF	Select signal of ADC triggering source 4'd0: pwm_ch0l_int (Channel 0 ThresholdL reached) 4'd1: pwm_ch0h_int (Channel 0 ThresholdH reached) 4'd2: pwm_ch1l_int (Channel 1 ThresholdL reached) 4'd3: pwm_ch1h_int (Channel 1 ThresholdH reached) 4'd4: pwm_ch2l_int (Channel 2 ThresholdL reached) 4'd5: pwm_ch2h_int (Channel 2 ThresholdH reached) 4'd6: pwm_ch3l_int (Channel 3 ThresholdL reached) 4'd7: pwm_ch3h_int (Channel 3 ThresholdH reached) 4'd8: pwm_prde_int (Period End reached) Others: Disabled
19	pwm_stop_on_rept	r/w	1'b0	PWM stopped when rept_int is asserted 0: Disabled, PWM keeps running when rept_int is asserted 1: Enabled, PWM stops when rept_int is asserted. Clear rept_int to restore operation
18:16	RSVD			
15:0	pwm_clk_div	r/w	16'b0	PWM clock division 0: 1 divisor 1: 1 divisor 2: 2 divisor 65535: 65535 divisor

12.4.3 pwm_mc0_config1

地址: 0x2000a444

pwm_ch3_nbs	pwm_ch3_pbs	pwm_ch2_nbs	pwm_ch2_pbs	pwm_ch1_nbs	pwm_ch1_pbs	pwm_ch0_nbs	pwm_ch0_pbs	pwm_ch3_npl	pwm_ch3_ppl	pwm_ch2_npl	pwm_ch2_ppl	pwm_ch1_npl	pwm_ch1_ppl	pwm_ch0_npl	pwm_ch0_ppl
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_ch3_nsi	pwm_ch3_nen	pwm_ch3_psi	pwm_ch3_pen	pwm_ch2_nsi	pwm_ch2_nen	pwm_ch2_psi	pwm_ch2_pen	pwm_ch1_nsi	pwm_ch1_nen	pwm_ch1_psi	pwm_ch1_pen	pwm_ch0_nsi	pwm_ch0_nen	pwm_ch0_psi	pwm_ch0_pen

位	名称	权限	复位值	描述
31	pwm_ch3_nbs	r/w	1'b0	PWM channel 3 negative break state 0: PWM channel 3 negative output logic 0 in break state 1: PWM channel 3 negative output logic 1 in break state
30	pwm_ch3_pbs	r/w	1'b0	PWM channel 3 positive break state 0: PWM channel 3 positive output logic 0 in break state 1: PWM channel 3 positive output logic 1 in break state
29	pwm_ch2_nbs	r/w	1'b0	PWM channel 2 negative break state 0: PWM channel 2 negative output logic 0 in break state 1: PWM channel 2 negative output logic 1 in break state
28	pwm_ch2_pbs	r/w	1'b0	PWM channel 2 positive break state 0: PWM channel 2 positive output logic 0 in break state 1: PWM channel 2 positive output logic 1 in break state
27	pwm_ch1_nbs	r/w	1'b0	PWM channel 1 negative break state 0: PWM channel 1 negative output logic 0 in break state 1: PWM channel 1 negative output logic 1 in break state
26	pwm_ch1_pbs	r/w	1'b0	PWM channel 1 positive break state 0: PWM channel 1 positive output logic 0 in break state 1: PWM channel 1 positive output logic 1 in break state
25	pwm_ch0_nbs	r/w	1'b0	PWM channel 0 negative break state 0: PWM channel 0 negative output logic 0 in break state 1: PWM channel 0 negative output logic 1 in break state
24	pwm_ch0_pbs	r/w	1'b0	PWM channel 0 positive break state 0: PWM channel 0 positive output logic 0 in break state 1: PWM channel 0 positive output logic 1 in break state
23	pwm_ch3_npl	r/w	1'b1	PWM channel 3 negative polarity 0: PWM channel 3 negative output high level in logic 0 state, output low level in logic 1 state 1: PWM channel 3 negative output low level in logic 0 state, output high level in logic 1 state
22	pwm_ch3_ppl	r/w	1'b1	PWM channel 3 positive polarity 0: PWM channel 3 positive output high level in logic 0 state, output low level in logic 1 state 1: PWM channel 3 positive output low level in logic 0 state, output high level in logic 1 state
21	pwm_ch2_npl	r/w	1'b1	PWM channel 2 negative polarity 0: PWM channel 2 negative output high level in logic 0 state, output low level in logic 1 state 1: PWM channel 2 negative output low level in logic 0 state, output high level in logic 1 state

位	名称	权限	复位值	描述
20	pwm_ch2_ppl	r/w	1'b1	PWM channel 2 positive polarity 0: PWM channel 2 positive output high level in logic 0 state, output low level in logic 1 state 1: PWM channel 2 positive output low level in logic 0 state, output high level in logic 1 state
19	pwm_ch1_npl	r/w	1'b1	PWM channel 1 negative polarity 0: PWM channel 1 negative output high level in logic 0 state, output low level in logic 1 state 1: PWM channel 1 negative output low level in logic 0 state, output high level in logic 1 state
18	pwm_ch1_ppl	r/w	1'b1	PWM channel 1 positive polarity 0: PWM channel 1 positive output high level in logic 0 state, output low level in logic 1 state 1: PWM channel 1 positive output low level in logic 0 state, output high level in logic 1 state
17	pwm_ch0_npl	r/w	1'b1	PWM channel 0 negative polarity 0: PWM channel 0 negative output high level in logic 0 state, output low level in logic 1 state 1: PWM channel 0 negative output low level in logic 0 state, output high level in logic 1 state
16	pwm_ch0_ppl	r/w	1'b1	PWM channel 0 positive polarity 0: PWM channel 0 positive output high level in logic 0 state, output low level in logic 1 state 1: PWM channel 0 positive output low level in logic 0 state, output high level in logic 1 state
15	pwm_ch3_nsi	r/w	1'b1	PWM channel 3 negative set idle state 0: PWM channel 3 negative logic 0 state 1: PWM channel 3 negative logic 1 state
14	pwm_ch3_nen	r/w	1'b0	PWM channel 3 negative enable pwm out 0: disable, PWM output is determined by pwm_ch3_nsi and pwm_ch3_npl 1: enable, PWM output is determined by counter and pwm_mc0_ch3_thre
13	pwm_ch3_psi	r/w	1'b0	PWM channel 3 positive set idle state 0: PWM channel 3 positive output logic 0 in idle state 1: PWM channel 3 positive output logic 1 in idle state

位	名称	权限	复位值	描述
12	pwm_ch3_pen	r/w	1'b0	PWM channel 3 positive enable pwm out 0: disable, PWM output is determined by pwm_ch3_psi and pwm_ch3_ppl 1: enable, PWM output is determined by counter and pwm_mc0_ch3_thre
11	pwm_ch2_nsi	r/w	1'b1	PWM channel 2 negative set idle state 0: PWM channel 2 negative logic 0 state 1: PWM channel 2 negative logic 1 state
10	pwm_ch2_nen	r/w	1'b0	PWM channel 2 negative enable pwm out 0: disable, PWM output is determined by pwm_ch2_nsi and pwm_ch2_npl 1: enable, PWM output is determined by counter and pwm_mc0_ch2_thre
9	pwm_ch2_psi	r/w	1'b0	PWM channel 2 positive set idle state 0: PWM channel 2 positive output logic 0 in idle state 1: PWM channel 2 positive output logic 1 in idle state
8	pwm_ch2_pen	r/w	1'b0	PWM channel 2 positive enable pwm out 0: disable, PWM output is determined by pwm_ch2_psi and pwm_ch2_ppl 1: enable, PWM output is determined by counter and pwm_mc0_ch2_thre
7	pwm_ch1_nsi	r/w	1'b1	PWM channel 1 negative set idle state 0: PWM channel 1 negative logic 0 state 1: PWM channel 1 negative logic 1 state
6	pwm_ch1_nen	r/w	1'b0	PWM channel 1 negative enable pwm out 0: disable, PWM output is determined by pwm_ch1_nsi and pwm_ch1_npl 1: enable, PWM output is determined by counter and pwm_mc0_ch1_thre
5	pwm_ch1_psi	r/w	1'b0	PWM channel 1 positive set idle state 0: PWM channel 1 positive output logic 0 in idle state 1: PWM channel 1 positive output logic 1 in idle state
4	pwm_ch1_pen	r/w	1'b0	PWM channel 1 positive enable pwm out 0: disable, PWM output is determined by pwm_ch1_psi and pwm_ch1_ppl 1: enable, PWM output is determined by counter and pwm_mc0_ch1_thre

位	名称	权限	复位值	描述
3	pwm_ch0_nsi	r/w	1'b1	PWM channel 0 negative set idle state 0: PWM channel 0 negative logic 0 state 1: PWM channel 0 negative logic 1 state
2	pwm_ch0_nen	r/w	1'b0	PWM channel 0 negative enable pwm out 0: disable, PWM output is determined by pwm_ch0_nsi and pwm_ch0_npl 1: enable, PWM output is determined by counter and pwm_mc0_ch0_thre
1	pwm_ch0_psi	r/w	1'b0	PWM channel 0 positive set idle state 0: PWM channel 0 positive output logic 0 in idle state 1: PWM channel 0 positive output logic 1 in idle state
0	pwm_ch0_pen	r/w	1'b0	PWM channel 0 positive enable pwm out 0: disable, PWM output is determined by pwm_ch0_psi and pwm_ch0_ppl 1: enable, PWM output is determined by counter and pwm_mc0_ch0_thre

12.4.4 pwm_mc0_period

地址: 0x2000a448

pwm_int_period_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

pwm_period

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

12.4.5 pwm_mc0_dead_time

地址：0x2000a44c

pwm_ch3_dtg								pwm_ch2_dtg							
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_ch1_dtg								pwm_ch0_dtg							

位	名称	权限	复位值	描述
31:24	pwm_ch3_dtg	r/w	8'h0	PWM Channel 3 dead time generator (DTG) DTG[7:5]=0xx => DT = DTG[7:0]*1 (unit: divided PWM clock) DTG[7:5]=10x => DT = (64+DTG[5:0])*2 (unit: divided PWM clock) DTG[7:5]=110 => DT = (32+DTG[4:0])*8 (unit: divided PWM clock) DTG[7:5]=111 => DT = (32+DTG[4:0])*16 (unit: divided PWM clock)
23:16	pwm_ch2_dtg	r/w	8'h0	PWM Channel 2 dead time generator (DTG) DTG[7:5]=0xx => DT = DTG[7:0]*1 (unit: divided PWM clock) DTG[7:5]=10x => DT = (64+DTG[5:0])*2 (unit: divided PWM clock) DTG[7:5]=110 => DT = (32+DTG[4:0])*8 (unit: divided PWM clock) DTG[7:5]=111 => DT = (32+DTG[4:0])*16 (unit: divided PWM clock)
15:8	pwm_ch1_dtg	r/w	8'h0	PWM Channel 1 dead time generator (DTG) DTG[7:5]=0xx => DT = DTG[7:0]*1 (unit: divided PWM clock) DTG[7:5]=10x => DT = (64+DTG[5:0])*2 (unit: divided PWM clock) DTG[7:5]=110 => DT = (32+DTG[4:0])*8 (unit: divided PWM clock) DTG[7:5]=111 => DT = (32+DTG[4:0])*16 (unit: divided PWM clock)

位	名称	权限	复位值	描述
7:0	pwm_ch0_dtg	r/w	8'h0	PWM Channel 0 dead time generator (DTG) DTG[7:5]=0xx => DT = DTG[7:0]*1 (unit: divided PWM clock) DTG[7:5]=10x => DT = (64+DTG[5:0])*2 (unit: divided PWM clock) DTG[7:5]=110 => DT = (32+DTG[4:0])*8 (unit: divided PWM clock) DTG[7:5]=111 => DT = (32+DTG[4:0])*16 (unit: divided PWM clock)

12.4.6 pwm_mc0_ch0_thre

地址: 0x2000a450

pwm_ch0_threH

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_ch0_threL

位	名称	权限	复位值	描述
31:16	pwm_ch0_threH	r/w	16'd0	PWM HIGH counter threshold, can't be smaller that threL
15:0	pwm_ch0_threL	r/w	16'd0	PWM LOW counter threshold, can't be larger that threH

12.4.7 pwm_mc0_ch1_thre

地址: 0x2000a454

pwm_ch1_threH

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_ch1_threL

位	名称	权限	复位值	描述
31:16	pwm_ch1_threH	r/w	16'd0	PWM HIGH counter threshold, can't be smaller that threL
15:0	pwm_ch1_threL	r/w	16'd0	PWM LOW counter threshold, can't be larger that threH

12.4.8 pwm_mc0_ch2_thre

地址：0x2000a458

pwm_ch2_threH

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_ch2_threL

位	名称	权限	复位值	描述
31:16	pwm_ch2_threH	r/w	16'd0	PWM HIGH counter threshold, can't be smaller that threL
15:0	pwm_ch2_threL	r/w	16'd0	PWM LOW counter threshold, can't be larger that threH

12.4.9 pwm_mc0_ch3_thre

地址：0x2000a45c

pwm_ch3_threH

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_ch3_threL

位	名称	权限	复位值	描述
31:16	pwm_ch3_threH	r/w	16'd0	PWM HIGH counter threshold, can't be smaller that threL
15:0	pwm_ch3_threL	r/w	16'd0	PWM LOW counter threshold, can't be larger that threH

12.4.10 pwm_mc0_int_sts

地址：0x2000a460

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
RSVD	RSVD	RSVD	RSVD	RSVD	pwm_rept_int	RSVD	pwm_prde_int	pwm_ch3h_int	pwm_ch3l_int	pwm_ch2h_int	pwm_ch2l_int	pwm_ch1h_int	pwm_ch1l_int	pwm_ch0h_int	pwm_ch0l_int

位	名称	权限	复位值	描述
31:11	RSVD			
10	pwm_rept_int	r	1'b0	PWM repeat count interrupt status
9	RSVD			
8	pwm_prde_int	r	1'b0	PWM period end interrupt status
7	pwm_ch3h_int	r	1'b0	PWM Channel 3 ThresholdH interrupt status
6	pwm_ch3l_int	r	1'b0	PWM Channel 3 ThresholdL interrupt status
5	pwm_ch2h_int	r	1'b0	PWM Channel 2 ThresholdH interrupt status
4	pwm_ch2l_int	r	1'b0	PWM Channel 2 ThresholdL interrupt status
3	pwm_ch1h_int	r	1'b0	PWM Channel 1 ThresholdH interrupt status
2	pwm_ch1l_int	r	1'b0	PWM Channel 1 ThresholdL interrupt status
1	pwm_ch0h_int	r	1'b0	PWM Channel 0 ThresholdH interrupt status
0	pwm_ch0l_int	r	1'b0	PWM Channel 0 ThresholdL interrupt status

12.4.11 pwm_mc0_int_mask

地址：0x2000a464

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	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
RSVD	RSVD	RSVD	RSVD	RSVD	cr_pwm_rept_mask	RSVD	cr_pwm_prde_mask	cr_pwm_ch3h_mask	cr_pwm_ch3l_mask	cr_pwm_ch2h_mask	cr_pwm_ch2l_mask	cr_pwm_ch1h_mask	cr_pwm_ch1l_mask	cr_pwm_ch0h_mask	cr_pwm_ch0l_mask

位	名称	权限	复位值	描述
31:11	RSVD			
10	cr_pwm_rept_mask	r/w	1'b1	Interrupt mask of pwm_rept_int
9	RSVD			
8	cr_pwm_prde_mask	r/w	1'b1	Interrupt mask of pwm_prde_int
7	cr_pwm_ch3h_mask	r/w	1'b1	Interrupt mask of pwm_ch3h_int
6	cr_pwm_ch3l_mask	r/w	1'b1	Interrupt mask of pwm_ch3l_int

位	名称	权限	复位值	描述
5	cr_pwm_ch2h_mask	r/w	1'b1	Interrupt mask of pwm_ch2h_int
4	cr_pwm_ch2l_mask	r/w	1'b1	Interrupt mask of pwm_ch2l_int
3	cr_pwm_ch1h_mask	r/w	1'b1	Interrupt mask of pwm_ch1h_int
2	cr_pwm_ch1l_mask	r/w	1'b1	Interrupt mask of pwm_ch1l_int
1	cr_pwm_ch0h_mask	r/w	1'b1	Interrupt mask of pwm_ch0h_int
0	cr_pwm_ch0l_mask	r/w	1'b1	Interrupt mask of pwm_ch0l_int

12.4.12 pwm_mc0_int_clear

地址: 0x2000a468

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	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	cr_pwm_rept_clr	cr_pwm_prde_clr	cr_pwm_ch3h_clr	cr_pwm_ch3l_clr	cr_pwm_ch2h_clr	cr_pwm_ch2l_clr	cr_pwm_ch1h_clr	cr_pwm_ch1l_clr	cr_pwm_ch0h_clr	cr_pwm_ch0l_clr

位	名称	权限	复位值	描述
31:11	RSVD			
10	cr_pwm_rept_clr	w1c	1'b0	Interrupt clear of pwm_rept_int
9	RSVD			
8	cr_pwm_prde_clr	w1c	1'b0	Interrupt clear of pwm_prde_int
7	cr_pwm_ch3h_clr	w1c	1'b0	Interrupt clear of pwm_ch3h_int
6	cr_pwm_ch3l_clr	w1c	1'b0	Interrupt clear of pwm_ch3l_int
5	cr_pwm_ch2h_clr	w1c	1'b0	Interrupt clear of pwm_ch2h_int
4	cr_pwm_ch2l_clr	w1c	1'b0	Interrupt clear of pwm_ch2l_int
3	cr_pwm_ch1h_clr	w1c	1'b0	Interrupt clear of pwm_ch1h_int
2	cr_pwm_ch1l_clr	w1c	1'b0	Interrupt clear of pwm_ch1l_int
1	cr_pwm_ch0h_clr	w1c	1'b0	Interrupt clear of pwm_ch0h_int
0	cr_pwm_ch0l_clr	w1c	1'b0	Interrupt clear of pwm_ch0l_int

12.4.13 pwm_mc0_int_en

地址：0x2000a46c

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
RSVD	RSVD	RSVD	RSVD	RSVD	cr_pwm_rept_en	RSVD	cr_pwm_prde_en	cr_pwm_ch3h_en	cr_pwm_ch3l_en	cr_pwm_ch2h_en	cr_pwm_ch2l_en	cr_pwm_ch1h_en	cr_pwm_ch1l_en	cr_pwm_ch0h_en	cr_pwm_ch0l_en

位	名称	权限	复位值	描述
31:11	RSVD			
10	cr_pwm_rept_en	r/w	1'b1	Interrupt enable of pwm_rept_int
9	RSVD			
8	cr_pwm_prde_en	r/w	1'b1	Interrupt enable of pwm_prde_int
7	cr_pwm_ch3h_en	r/w	1'b1	Interrupt enable of pwm_ch3h_int
6	cr_pwm_ch3l_en	r/w	1'b1	Interrupt enable of pwm_ch3l_int
5	cr_pwm_ch2h_en	r/w	1'b1	Interrupt enable of pwm_ch2h_int
4	cr_pwm_ch2l_en	r/w	1'b1	Interrupt enable of pwm_ch2l_int
3	cr_pwm_ch1h_en	r/w	1'b1	Interrupt enable of pwm_ch1h_int
2	cr_pwm_ch1l_en	r/w	1'b1	Interrupt enable of pwm_ch1l_int
1	cr_pwm_ch0h_en	r/w	1'b1	Interrupt enable of pwm_ch0h_int
0	cr_pwm_ch0l_en	r/w	1'b1	Interrupt enable of pwm_ch0l_int

13.1 简介

芯片内置两个 32-bit 定时器，定时器可独立控制并配置其参数与时钟频率。

芯片内有一个 Watchdog 定时器，不可预知的软件或硬件行为可能导致应用程序工作失常，Watchdog 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，仍没有喂狗或关闭 Watchdog 定时器，会触发中断或系统复位。

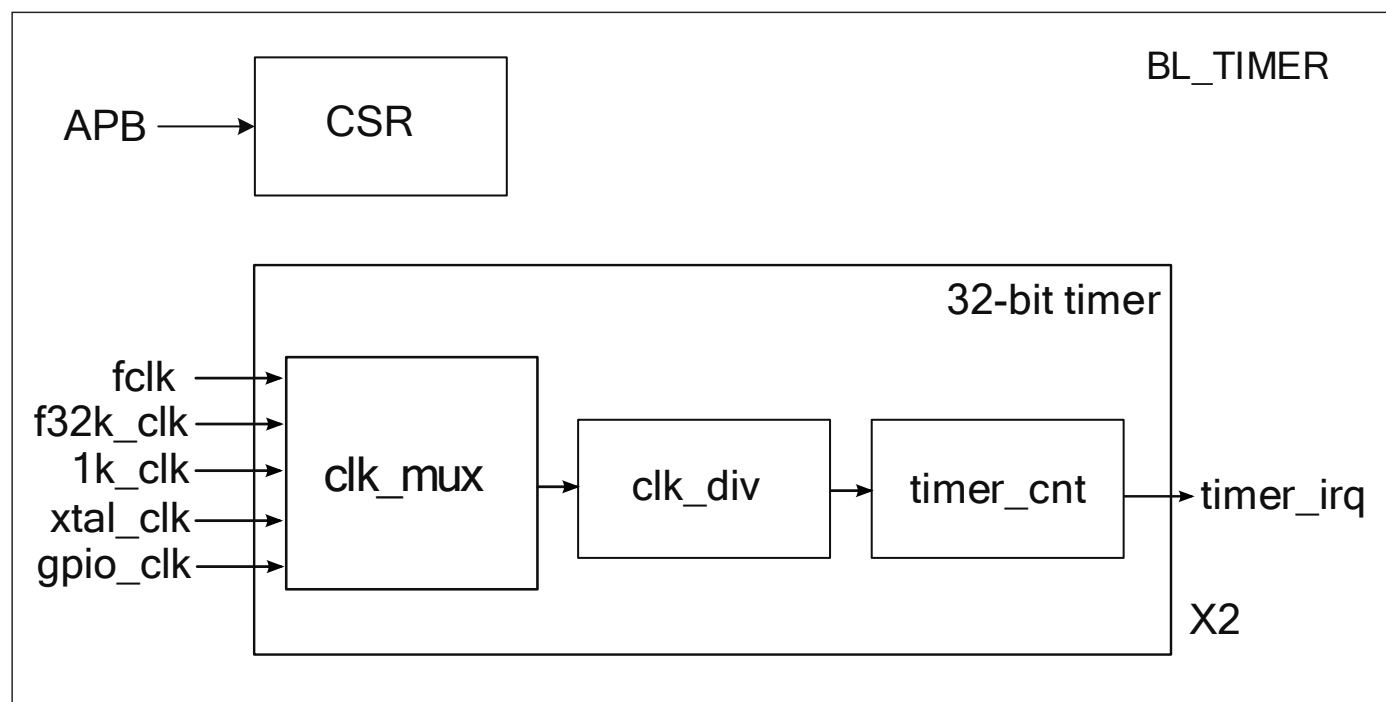


图 13.1: 定时器框图

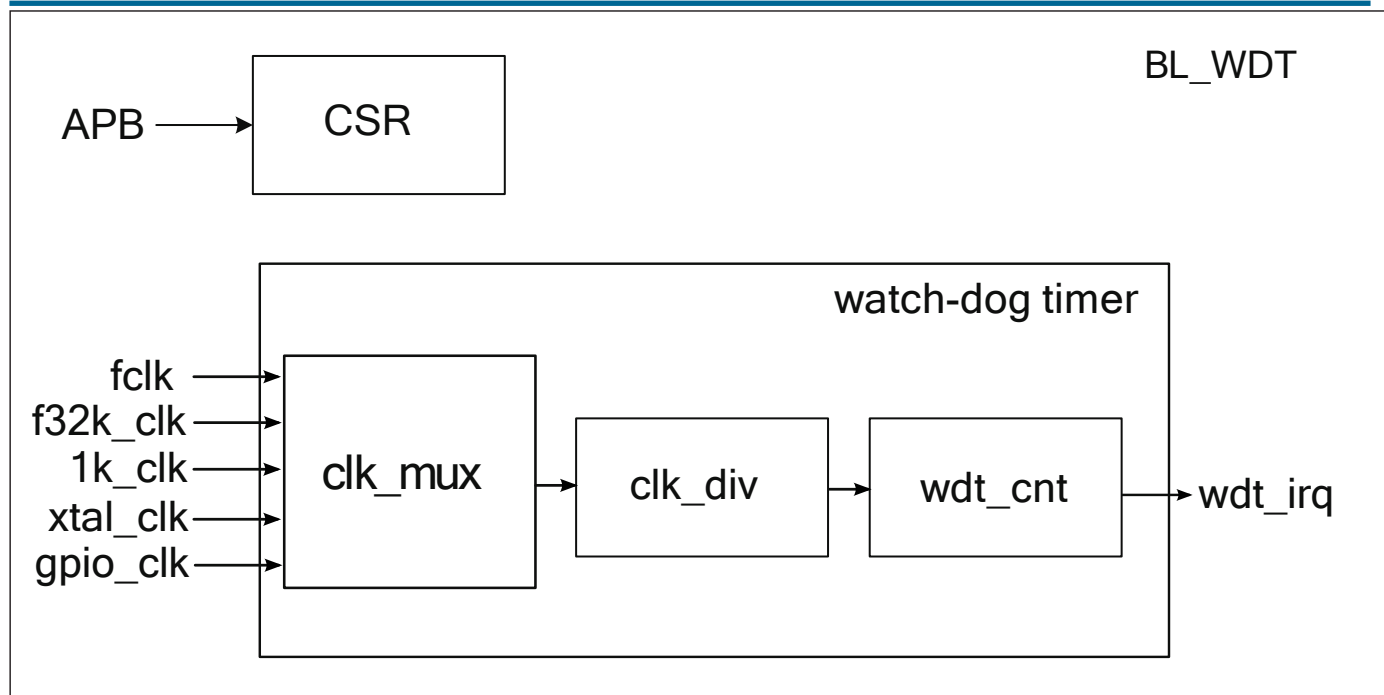


图 13.2: Watchdog 定时器框图

13.2 主要特征

- 32-bit 定时器
 - 多种时钟来源，最高可支持 80M 时钟
 - 8-bit 时钟分频器，分频系数为 1-256
 - 两个 32-bit 定时器
 - 定时器包含三组触发阈值设定，达到阈值时触发中断
 - 支持 FreeRun 模式和 Preload 模式
 - 支持测量外部 GPIO 的脉冲宽度
- Watchdog 定时器
 - 一个 16-bit Watchdog 定时器
 - 支持写保护，防止误设定造成系统异常
 - 支持中断或复位两种方式

13.3 功能描述

5 种 Watchdog 定时器时钟，可通过寄存器 TCCR 中的 cs_wdt 选择：

- FCLK--总线时钟
- 32K--32K 时钟
- 1K--1K 时钟（32K 的分频）
- XTAL--外部晶振
- GPIO--外部 GPIO

5 种 timer 定时器时钟，可通过寄存器 TCCR 中的 cs_2 和 cs_3 选择：

- FCLK--总线时钟
- 32K--32K 时钟
- 1K--1K 时钟（32K 的分频）
- XTAL--外部晶振
- GPIO--外部 GPIO

定时器有各自的 8-bit 分频器，通过寄存器 TCDR 中的 tcd_r2、tcd_r3、wcd_r 配置，可对其选择的时钟源进行 1-256 的分频。设定为 0 时表示不分频，设定为 1 时表示 2 分频以此类推，最大分频系数为 256，定时器以分频后的时钟作为计数周期单位。

13.3.1 通用定时器工作原理

通用定时器包含三组比较器，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，计数器从初始值开始向上累加计数，当计数器的值与比较器相等的时候，比较寄存器置位的同时产生比较中断。

寄存器 TCCR2 中的 tclr2_0 设置 timer0 比较器 0 的值，寄存器 TCCR2 中的 tclr2_1 设置 timer0 比较器 1 的值，寄存器 TCCR2 中的 tclr2_2 设置 timer0 比较器 2 的值。寄存器 TPLVR2 中的 tplvr2 设置 timer0 预加载值。

寄存器 TCCR3 中的 tclr3_0 设置 timer1 比较器 0 的值，寄存器 TCCR3 中的 tclr3_1 设置 timer1 比较器 1 的值，寄存器 TCCR3 中的 tclr3_2 设置 timer1 比较器 2 的值。寄存器 TPLVR3 中的 tplvr3 设置 timer1 预加载值。

定时器支持两种计数模式：PreLoad 模式和 FreeRun 模式，通过寄存器 TCMR 配置。寄存器 TCMR 中的 timer2_mode 设置 timer0 的计数模式，寄存器 TCMR 中的 timer3_mode 设置 timer1 的计数模式。

13.3.1.1 PreLoad 模式

PreLoad 模式下计数器的初始值为 TPLVR 寄存器（预加载寄存器）的值，从这个初始值开始向上累加计数。寄存器 TPLVR2 设置 timer0 的 Preload 预加载值，寄存器 TPLVR3 配置 timer1 的 Preload 预加载值。寄存器 TPLCR2 中的 tplcr2 配置 timer0 的 PreLoad 触发条件，寄存器 TPLCR3 中的 tplcr3 配置 timer1 的 PreLoad 触发条件。当满足 PreLoad 触发条件时，计数器的值会被重新置为 PreLoad 寄存器的值，然后计数器再次开始向上累加计数。

在定时器的计数器计数过程中，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：

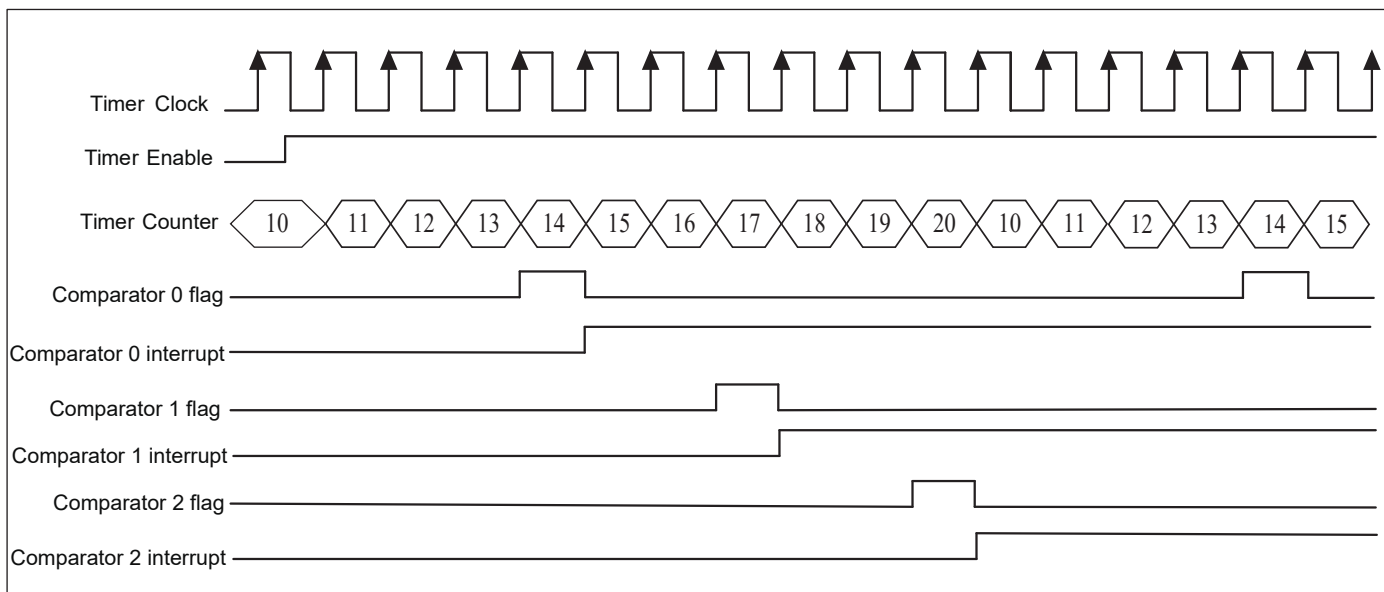


图 13.1: 定时器在 PreLoad 模式下工作时序

13.3.1.2 FreeRun 模式

FreeRun 模式即计数器累加模式，在 FreeRun 模式下计数器的初始值为 0，在启动定时器后从 0 开始累加计数，当达到计数最大值后，再次从 0 开始计数。

在定时器的计数器计数过程中，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。

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。

13.3.1.3 测量外部 GPIO 脉冲宽度

定时器支持使用内部时钟源计算外部 GPIO 的脉冲宽度。

配置的步骤如下：

1. 将外部 GPIO 配置成 gpio_tmr_clk 功能。通过配置 GLB 模块中寄存器 dig_clk_cfg2 实现。

寄存器 dig_clk_cfg2[13:12] 位为 gpio_tmr_clk_sel: 选择 GPIO clock 模式；寄存器 dig_clk_cfg2[11:8] 中的某一位配置为 0，表示 clock input 模式。这两个寄存器需要配套使用，配置方法如下，根据使用的 GPIO 以此类推。

表 13.1: gpio_tmr_clk 功能配置

GPIO	dig_clk_cfg2[13:12]	dig_clk_cfg2[11:8]
GPIO0	gpio_tmr_clk = 0	chip_clk_out_0_en = 0
GPIO1	gpio_tmr_clk = 1	chip_clk_out_1_en = 0
GPIO2	gpio_tmr_clk = 2	chip_clk_out_2_en = 0
GPIO3	gpio_tmr_clk = 3	chip_clk_out_3_en = 0
GPIO4	gpio_tmr_clk = 0	chip_clk_out_0_en = 0

2. 寄存器 GPIO 中的 timer2_gpio_inv / timer3_gpio_inv 位配置需要测量外部脉宽的高电平还是低电平。如果该位为 0，表示高电平；该位为 1 表示低电平。
3. 配置寄存器 GPIO 中的 timer2_gpio_en 使能 GPIO 测量功能
4. 配置定时器时钟源和运行模式，并使能 timer
5. 当寄存器 GPIO 中的 gpio_lat_ok 置 1 后，获取寄存器 GPIO_LAT2 和寄存器 GPIO_LAT1 的值并计算宽度

外部 GPIO 的脉冲宽度的计算方式：(GPIO_LAT2 - GPIO_LAT1) * timer 内部时钟源的 1 个周期的宽度；

例如：timer 的内部时钟源为 80M，外部 GPIO 的频率为 2M，占空比为 1: 1，计算外部 GPIO 低电平和高电平的宽度。

- 将 timer2_gpio_inv 位写 1，表示计算外部 gpio 低电平的宽度。按照上述配置完成后，得到寄存器 GPIO_LAT2 和寄存器 GPIO_LAT1 的差为 20，则外部 gpio 的低电平宽度为：20 * (1 / 80000000) = 1 / 4000000。
- 将 timer2_gpio_inv 位写 0，表示计算外部 gpio 高电平的宽度。按照上述配置完成后，如果寄存器 GPIO_LAT2 和寄存器 GPIO_LAT1 的差为 20，则外部 gpio 的低电平宽度为：20 * (1 / 80000000) = 1 / 4000000。

13.3.2 Watchdog 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。Watchdog 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 Watchdog 计数器归零。

Watchdog 定时器通过寄存器 WMR 中的 wmr 设置比较值，若比较值为 6，Watchdog 的工作时序如下图所示：

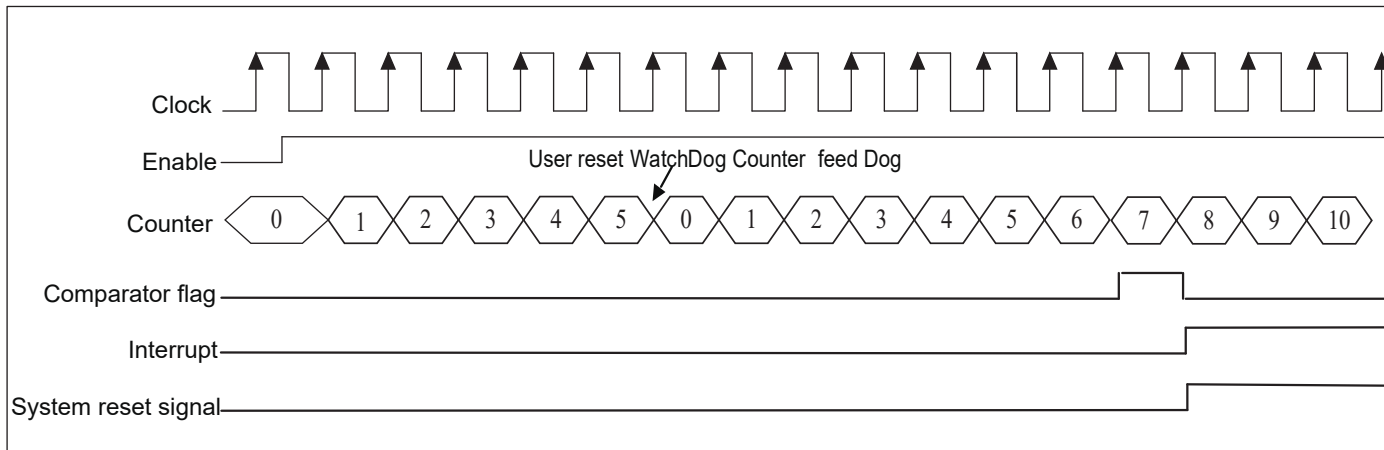


图 13.2: Watchdog 工作时序

13.3.3 定时器 Alarm 设定

计数器有三个比较值提供软件设定，并可设定比较值是否触发 alarm 中断，当计数器达到比较值且设定会 alarm 时，计数器通过中断通知处理器。软件可以通过 APB 读取目前是否发生 alarm 和是哪个比较值触发 alarm 中断，当清理 alarm 中断时亦会同步清理 alarm 状态。

13.3.4 Watchdog Alarm 设定

Watchdog 可设定一组比较值，当软件因为系统错误，来不及将 Watchdog 计数器归零，导致 Watchdog 计数器超过比较值时，便会触发 Watchdog alarm，alarm 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 Watchdog 复位。Watchdog 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 Watchdog 复位，且软件可通过 APB 读取 WSR 寄存器得知是否曾经发生过 Watchdog 系统复位。

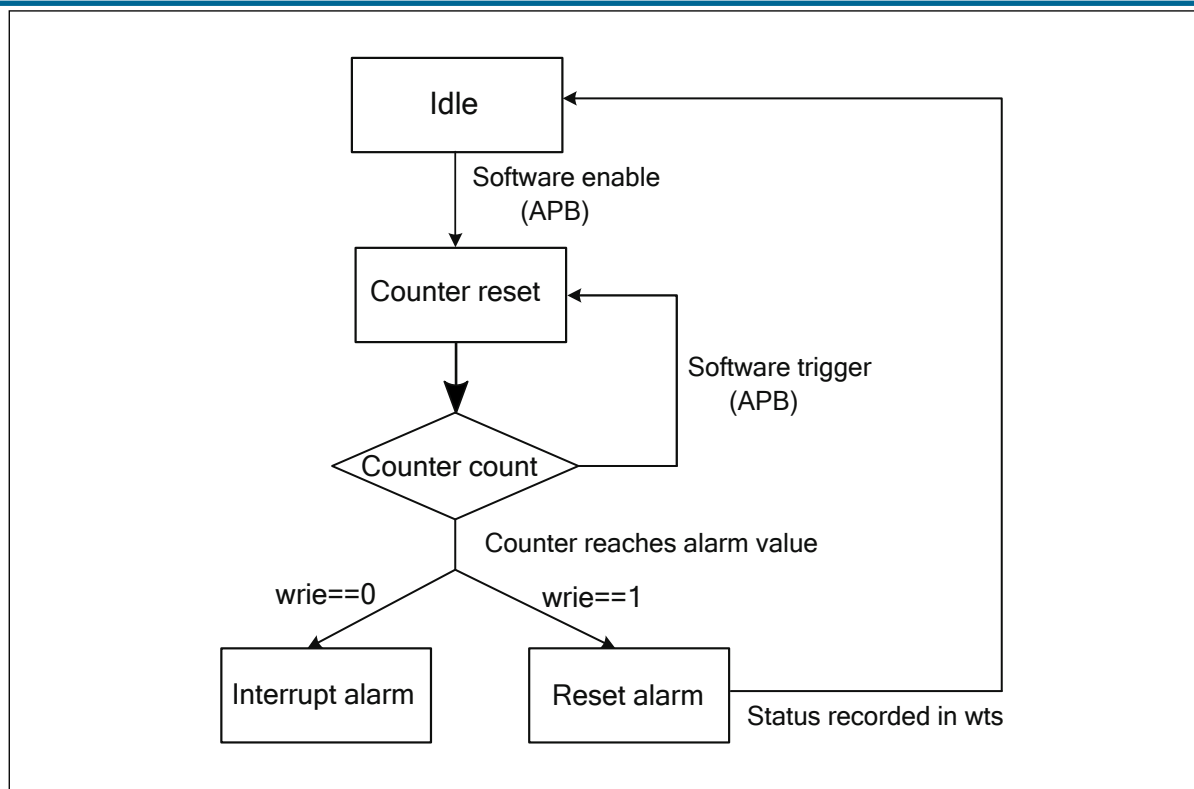


图 13.3: Watchdog alarm 机制

13.4 寄存器描述

名称	描述
TCCR	Timer Clock Source
TMR2_0	Timer2 Match Value 0
TMR2_1	Timer2 Match Value 1
TMR2_2	Timer2 Match Value 2
TMR3_0	Timer3 Match Value 0
TMR3_1	Timer3 Match Value 1
TMR3_2	Timer3 Match Value 2
TCR2	Timer2 Counter Value
TCR3	Timer3 Counter Value
TSR2	Timer2 Match Status
TSR3	Timer3 Match Status
TIER2	Timer2 Match Interrupt Enable

名称	描述
TIER3	Timer3 Match Interrupt Enable
TPLVR2	Timer2 Pre-Load Value
TPLVR3	Timer3 Pre-Load Value
TPLCR2	Timer2 Pre-Load Control
TPLCR3	Timer3 Pre-Load Control
WMER	Watch-dog reset/interrupt Mode
WMR	Watch-dog Match Value
WVR	Watch-dog Counter Value
WSR	Watch-dog Reset Status
TICR2	Timer2 Interrupt Clear
TICR3	Timer3 Interrupt Clear
WICR	WDT Interrupt Clear
TCER	Timer Counter Enable/Clear
TCMR	Timer Counter Mode
TILR2	Timer2 Match Interrupt Mode
TILR3	Timer3 Match Interrupt Mode
WCR	WDT Counter Reset
WFAR	WDT Access Key1
WSAR	WDT Access Key2
TCDR	Timer Division
GPIO	GPIO Mode
GPIO_LAT1	GPIO Latch Value1
GPIO_LAT2	GPIO Latch Value2

13.4.1 TCCR

地址: 0x2000a500

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 RSVD RSVD RSVD RSVD RSVD RSVD RSVD RSVD

RSVD RSVD RSVD RSVD cs_wdt cs_3 cs_2

位	名称	权限	复位值	描述
31:12	RSVD			
11:8	cs_wdt	r/w	4'd1	WDT 0:fclk / 1:f32k / 2:1k / 3:32M / 4:GPIO / 5:No clock
7:4	cs_3	r/w	4'd5	Timer3 0:fclk / 1:f32k / 2:1k / 3:32M / 4:GPIO / 5:No clock
3:0	cs_2	r/w	4'd5	Timer2 0:fclk / 1:f32k / 2:1k / 3:32M / 4:GPIO / 5:No clock

13.4.2 TMR2_0

地址: 0x2000a510

tmr2_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

tmr2_0

位	名称	权限	复位值	描述
31:0	tmr2_0	r/w	32'hffffff	Timer2 Match Value 0

13.4.3 TMR2_1

地址: 0x2000a514

tmr2_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

tmr2_1

位	名称	权限	复位值	描述
31:0	tmr2_1	r/w	32'hffffff	Timer2 Match Value 1

13.4.4 TMR2_2

地址: 0x2000a518

tmr2_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

tmr2_2

位	名称	权限	复位值	描述
31:0	tmr2_2	r/w	32'hffffff	Timer2 Match Value 2

13.4.5 TMR3_0

地址: 0x2000a51c

tmr3_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

tmr3_0

位	名称	权限	复位值	描述
31:0	tmr3_0	r/w	32'hffffff	Timer3 Match Value 0

13.4.6 TMR3_1

地址: 0x2000a520

tmr3_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

tmr3_1

位	名称	权限	复位值	描述
31:0	tmr3_1	r/w	32'hffffff	Timer3 Match Value 1

13.4.7 TMR3_2

地址: 0x2000a524

tmr3_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

tmr3_2

位	名称	权限	复位值	描述
31:0	tmr3_2	r/w	32'hffffff	Timer3 Match Value 2

13.4.8 TCR2

地址: 0x2000a52c

tcr2_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

tcr2_cnt

位	名称	权限	复位值	描述
31:0	tcr2_cnt	r	0	Timer2 Counter Value

13.4.9 TCR3

地址: 0x2000a530

tcr3_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

tcr3_cnt

位	名称	权限	复位值	描述
31:0	tcr3_cnt	r	0	Timer3 Counter Value

13.4.10 TSR2

地址: 0x2000a538

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	tsr2_2	tsr2_1
														tsr2_0	

位	名称	权限	复位值	描述
31:3	RSVD			
2	tsr2_2	r	0	Timer2 match value 2 status/Clear interrupt would also clear this bit
1	tsr2_1	r	0	Timer2 match value 1 status/Clear interrupt would also clear this bit
0	tsr2_0	r	0	Timer2 match value 0 status/Clear interrupt would also clear this bit

13.4.11 TSR3

地址: 0x2000a53c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	tsr3_2	tsr3_1
														tsr3_0	

位	名称	权限	复位值	描述
31:3	RSVD			

位	名称	权限	复位值	描述
2	tsr3_2	r	0	Timer3 match value 2 status/Clear interrupt would also clear this bit
1	tsr3_1	r	0	Timer3 match value 1 status/Clear interrupt would also clear this bit
0	tsr3_0	r	0	Timer3 match value 0 status/Clear interrupt would also clear this bit

13.4.12 TIER2

地址: 0x2000a544

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	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	0	Timer2 match value 2 interrupt enable
1	tier2_1	r/w	0	Timer2 match value 1 interrupt enable
0	tier2_0	r/w	0	Timer2 match value 0 interrupt enable

13.4.13 TIER3

地址: 0x2000a548

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	tier3_2	tier3_1
														tier3_0	

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

13.4.14 TPLVR2

地址: 0x2000a550

tplvr2

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	0	Timer2 Pre-Load Value

13.4.15 TPLVR3

地址: 0x2000a554

tplvr3

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	0	Timer3 Pre-Load Value

13.4.16 TPLCR2

地址：0x2000a55c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	tplcr2

位	名称	权限	复位值	描述
31:2	RSVD			
1:0	tplcr2	r/w	0	Timer2 pre-load control 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：0x2000a560

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	tplcr3

位	名称	权限	复位值	描述
31:2	RSVD			
1:0	tplcr3	r/w	0	Timer3 pre-load control 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：0x2000a564

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	wrie	r/w	0	WDT reset/interrupt mode 1'b0 - WDT expiration to generate interrupt 1'b1 - WDT expiration to generate reset source
0	we	r/w	0	WDT enable register

13.4.19 WMR

地址：0x2000a568

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
31:17	RSVD			
16	wdt_align	r/w	0	WDT compare value update align interrupt
15:0	wmr	r/w	16'hffff	WDT counter match value

13.4.20 WVR

地址：0x2000a56c

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

wdt_cnt

位	名称	权限	复位值	描述
31:16	RSVD			
15:0	wdt_cnt	r	0	WDT counter value

13.4.21 WSR

地址：0x2000a570

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

ws

位	名称	权限	复位值	描述
31:1	RSVD			
0	ws	w	0	WDT reset status Write 0 to clear the WDT reset status Read 1 indicates reset was caused by the WDT

13.4.22 TCR2

地址：0x2000a578

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	tclr2_2	tclr2_1
															tclr2_0

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

13.4.23 TICR3

地址: 0x2000a57c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	tclr3_2	tclr3_1
															tclr3_0

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

13.4.24 WICR

地址：0x2000a580

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	wiclr

位	名称	权限	复位值	描述
31:1	RSVD			
0	wiclr	w	0	WDT Interrupt Clear

13.4.25 TCER

地址：0x2000a584

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	tcr3_cnt_clr	tcr2_cnt_clr	RSVD	RSVD	timer3_en	timer2_en

位	名称	权限	复位值	描述
31:7	RSVD			
6	tcr3_cnt_clr	r/w	0	Timer3 count clear
5	tcr2_cnt_clr	r/w	0	Timer2 count clear
4:3	RSVD			
2	timer3_en	r/w	0	Timer3 count enable
1	timer2_en	r/w	0	Timer2 count enable
0	RSVD			

13.4.26 TCMR

地址：0x2000a588

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
31:7	RSVD			
6	timer3_align	r/w	0	Timer3 compare value update align interrupt
5	timer2_align	r/w	0	Timer2 compare value update align interrupt
4:3	RSVD			
2	timer3_mode	r/w	0	0:pre-load mode 1:free run mode
1	timer2_mode	r/w	0	0:pre-load mode 1:free run mode
0	RSVD			

13.4.27 TILR2

地址：0x2000a590

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	0	0:level 1:edge
1	tilr2_1	r/w	0	0:level 1:edge

位	名称	权限	复位值	描述
0	tilr2_0	r/w	0	0:level 1:edge

13.4.28 TILR3

地址: 0x2000a594

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	tilr3_2	r/w	0	0:level 1:edge
1	tilr3_1	r/w	0	0:level 1:edge
0	tilr3_0	r/w	0	0:level 1:edge

13.4.29 WCR

地址: 0x2000a598

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	0	WDT Counter Reset

13.4.30 WFAR

地址：0x2000a59c

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	0	WDT access key1 - 16'hBABA

13.4.31 WSAR

地址：0x2000a5a0

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	0	WDT access key2 - 16'hEB10
-1:32	RSVD			
31:0	tcr2_cnt_lat	r	0	Timer2 Counter Latch Value
-1:32	RSVD			
31:0	tcr3_cnt_lat	r	0	Timer3 Counter Latch Value
-1:32	RSVD			
31:0	tcr2_cnt_sync	r	0	Timer2 Counter Sync Value (continue readable)
-1:32	RSVD			
31:0	tcr3_cnt_sync	r	0	Timer3 Counter Sync Value (continue readable)

13.4.32 TCDR

地址：0x2000a5bc

wcdr								tcd3							
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
tcd2								RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD

位	名称	权限	复位值	描述
31:24	wcdr	r/w	0	WDT clock division value register
23:16	tcd3	r/w	0	Timer3 clock division value register
15:8	tcd2	r/w	0	Timer2 clock division value register
7:0	RSVD			

13.4.33 GPIO

地址：0x2000a5c0

gpio_lat_ok															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															

位	名称	权限	复位值	描述
4:2	RSVD			
1	timer2_gpio_en	r/w	0	Timer2 gpio measure enable
0	RSVD			

13.4.34 GPIO_LAT1

地址: 0x2000a5c4

gpio_lat1

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_lat1

位	名称	权限	复位值	描述
31:0	gpio_lat1	r	0	Pos-Edge Latch Timer2

13.4.35 GPIO_LAT2

地址: 0x2000a5c8

gpio_lat2

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_lat2

位	名称	权限	复位值	描述
31:0	gpio_lat2	r	0	Neg-Edge Latch Timer2

14.1 简介

I2S 全称 Inter-IC Sound, Integrated Interchip Sound, 或简写 IIS, 是飞利浦在 1986 年定义 (1996 年修订) 的数字音频传输标准, 一般用于数字音频数据在系统的内部器件之间传输。I2S 是简单的数字接口协议, 没有地址或设备选择机制, 支持全双工对等传输。在 I2S 总线中, 提供时钟 (BCLK 和 FS) 的设备为主设备。多数场景下, I2S 总线上分别有一个主机设备和一个从机设备单向或双向传输数据。但在一些特殊场景下, 也可以由一个主机仅提供时钟, 控制两个从机之间的数据传输, 可以精确控制数据流。

芯片集成的 I2S 模块另外兼容 PCM/TDM 格式, 相比于 I2S 只能传输双声道音频数据, PCM/TDM 接口通过时分复用 (Time Division Multiplexing) 方式, 将传输的声道数拓展到更多。PCM/TDM 与 I2S 的主要区别在于帧时钟的位置、长度和帧长度, 可以认为 I2S 是 PCM/TDM 接口的特例。但 PCM/TDM 没有完全统一的应用标准, 不同厂商在应用时可能略有差异。

14.2 主要特征

- 支持主模式与从模式
- 兼容的标准 I2S 协议
 - Normal I2S 格式 (飞利浦格式)
 - Left-Justified 格式
 - Right-Justified 格式
 - 通道 Slot 与有效数据宽度分别支持 8/16/24/32 比特, 有效数据宽度不能大于通道宽度
- 支持 PCM/TDM 格式, 帧时钟位置与帧长度可灵活调整, 最高支持 6 声道, 兼容多数常见的模式
 - 支持 2 通道、3 通道、4 通道和 6 通道模式, 通道 Slot 支持 8/16/24/32 比特
 - PCM/TDM Mode A, 数据在 FS 有效后, BCLK 的第 2 个上升沿有效
 - PCM/TDM Mode B, 数据在 FS 有效后, BCLK 的第 1 个上升沿有效

- Long Frame Sync, 长帧同步模式, FS 脉冲宽度等于 1 个 Slot 的长度
- Short Frame Sync, 短帧同步模式, FS 脉冲宽度等于 1 个 BCLK 周期长度
- 支持 8/11.025/16/22.05/32/44.1/48/96/192 KHz 采样率, 可通过配置时钟源与分频系数达到更精细的控制
- 支持播放单声道音频复制为双声道模式
- 支持低于 8/16bits 双声道录音数据合并, 提高 FIFO 使用效率
- 支持动态静音切换功能
- 支持数据 MSB/LSB 切换
- 支持直接将时钟源输出, 可以作为 MCLK, 也可以配合 GLB 的 CLK_OUT 功能输出 MCLK
- 支持输出信号极性翻转
- 支持 DMA 传输模式
- 数据发送/接收 FIFO 的宽度为 32 位, 深度 16

14.3 功能描述

14.3.1 数据格式描述

I2S 模块的信号时序格式高度灵活可配, 包括 FS 信号的模式、FS 信号到有效数据的偏移、Slot 宽度与有效数据宽度、可以兼容绝大多数的常见格式。I2S 协议为双声道模式, FS 信号长度与一个 Slot 相等, 有效数据长度小于或等于 Slot 长度。其中 Normal 格式下有效数据左对齐, 并且会相对 FS 信号偏移一位。因此, 设置 i2s_config 寄存器的 cr_fs_ch_cnt 段为双声道, 设置 cr_fs_1t_mode 段 FS 信号长度等于 Slot, 设置 cr_i2s_mode 段为数据左对齐, 设置 cr_ofs_en 与 cr_ofs_cnt 段为一位数据偏移, 即可得到 I2S 协议 Normal 格式, 如下图。

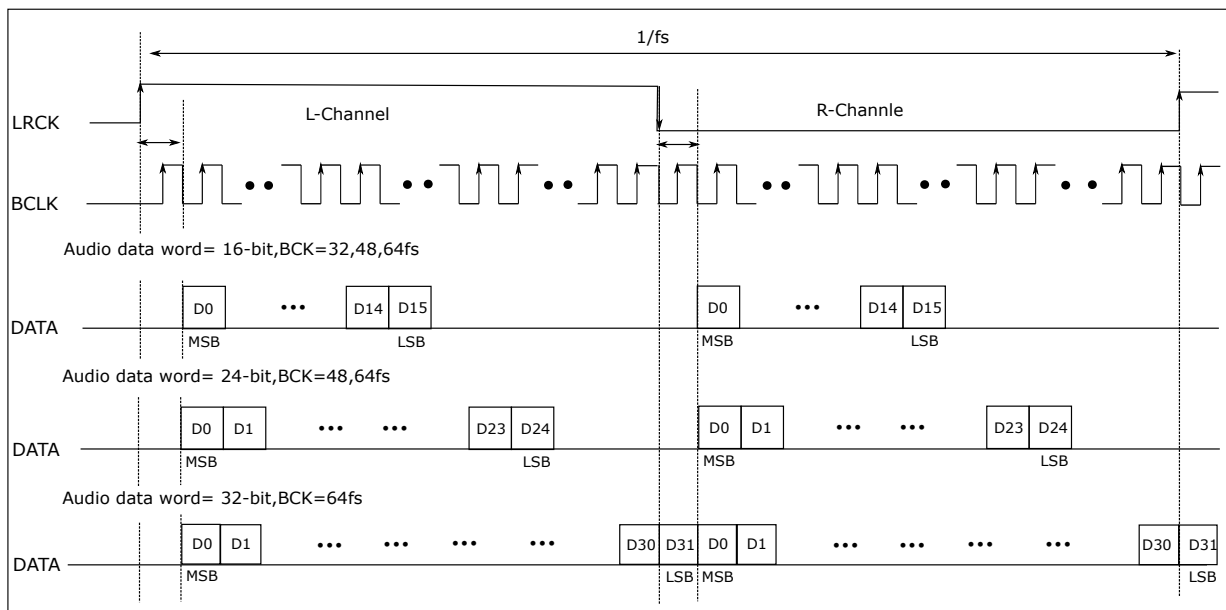


图 14.1: I2S 协议 Normal 格式

I2S 协议的左对齐格式与 Normal 格式相似，但没有数据偏移。右对齐格式的有效数据会在 Slot 内向右对齐排布。因此，在 Normal 格式的基础上，设置 `i2s_config` 寄存器的 `cr_ofs_en` 段关闭数据偏移就可得到 Left-Justified 格式，再将 `cr_i2s_mode` 段设置为数据右对齐就可得到 Right-Justified 格式，如下图。

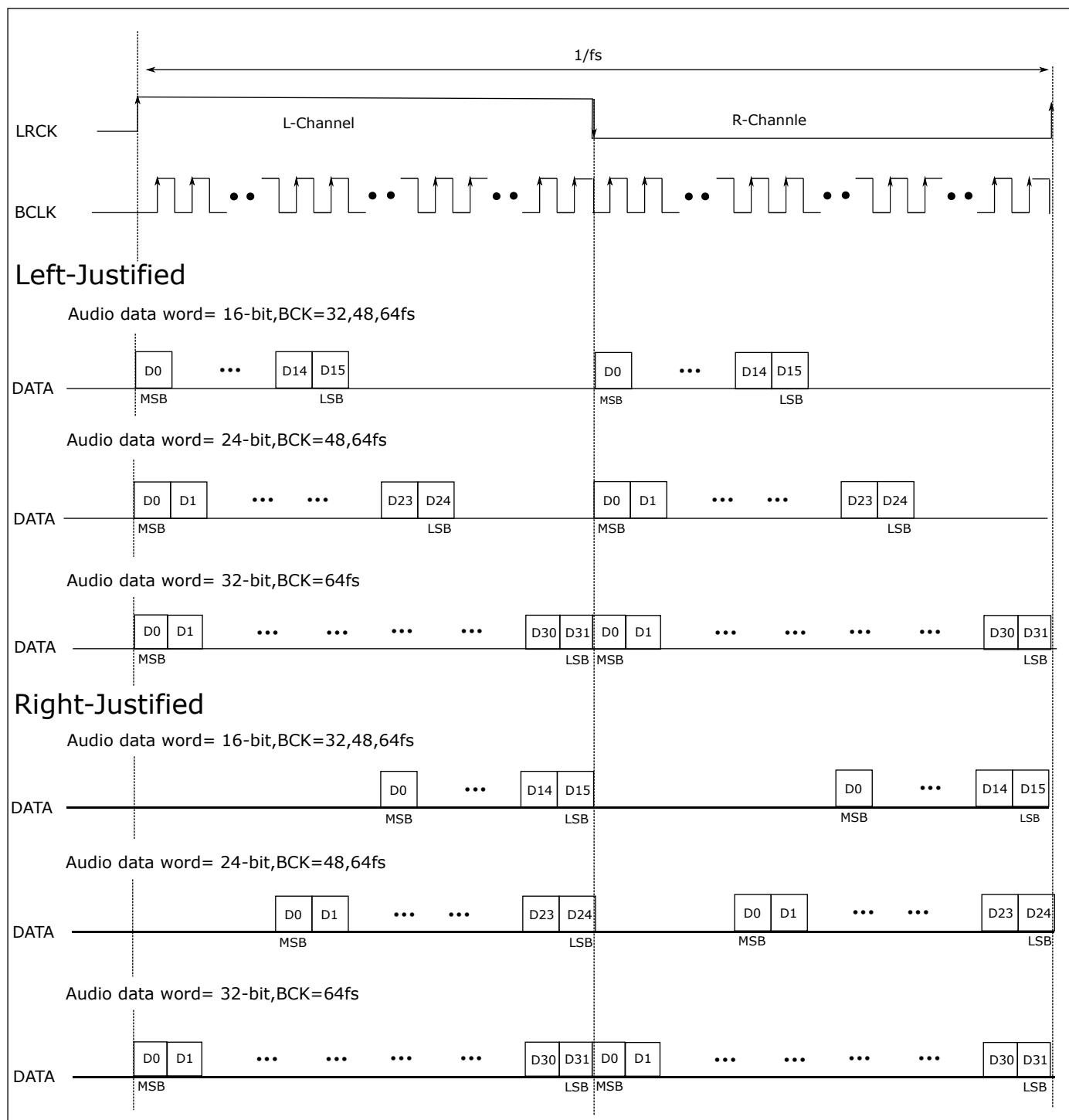


图 14.2: I2S 协议 Left-Justified/Right-Justified 格式

PCM/TDM 模式的配置更加灵活，设置 `i2s_config` 寄存器的 `cr_i2s_mode` 段为 2 即为 PCM/TDM 模式，设置 `cr_fs_ch_cnt` 段选择通道数，设置 `cr_fs_1t_mode` 段选择长帧同步 (FS 信号长度等于 Slot) 或短帧同步 (FS 信号长度为一个

BCLK 周期) 模式, 设置 `cr_ofs_en` 段与 `cr_ofs_cnt` 段选择数据偏移, 一般定义 Mode A 偏移 1 位, Mode B 无偏移, 但是不同厂商的定义可能有所差别。下图为 Mode B 在 Long/Short Frame Sync 时的时序, 将数据向右偏移一位即为 Mode A 的时序。

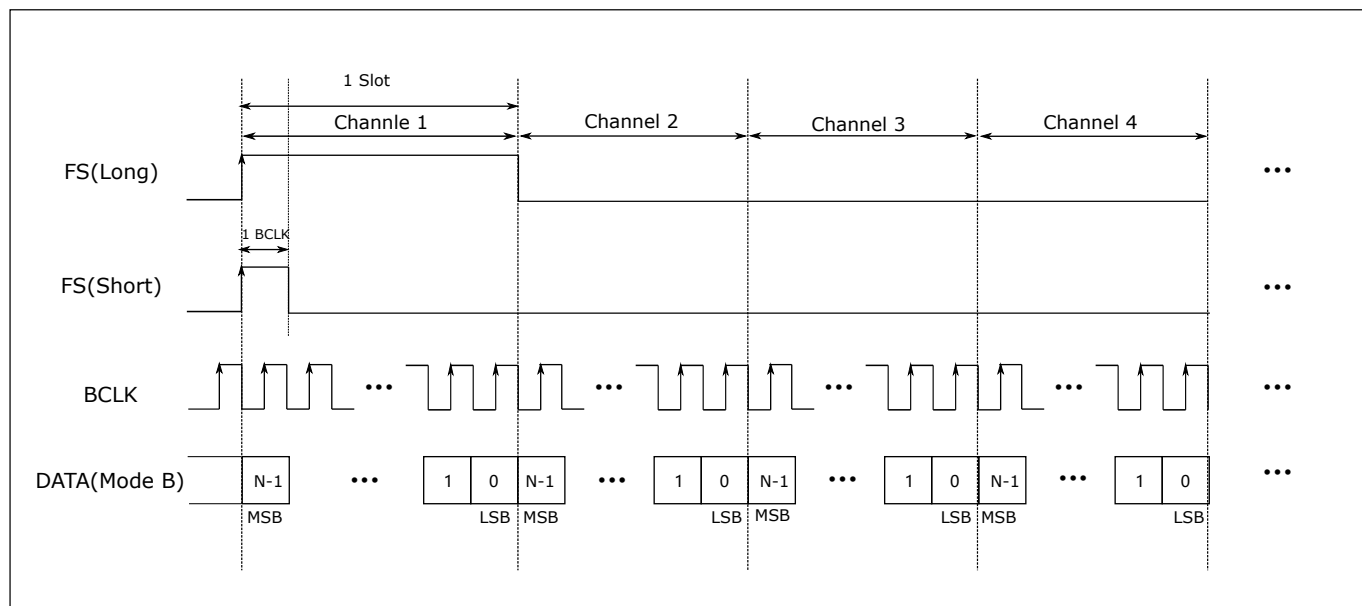


图 14.3: PCM/TDM 模式

14.3.2 I2S 单通道模式

通常 I2S 使用双声道传输, 但在使用单声道设备时会有资源浪费。配置 `i2s_config` 寄存器的 `cr_mono_mode` 段为 Mono 模式后, 就会使用单声道模式, 需要注意的是发送与接收都会使用单声道模式。此时会将 TX FIFO 中的数据同时放置到左右两个声道中, 并可以通过 `i2s_config` 寄存器的 `cr_mono_rx_ch` 段来选择保留左声道还是右声道接收到的数据。

14.3.3 数据 MSB/LSB 调整

通常 I2S 使用 MSB 格式传输数据, 但部分厂商设备的某些模式可能会使用 LSB 格式传输, 可通过 `i2s_config` 寄存器的 `cr_endian` 位调整格式。

14.3.4 双声道数据合并与交换

在使用 I2S 协议的 8/16bit 有效数据时, 为了增加 FIFO 的利用效率, 可以通过 `i2s_fifo_config_0` 寄存器的 `cr_fifo_lr_merge` 段合并双通道数据, 此时 FIFO 中每个数据同时包含左声道与右声道数据, 在 16bit 时 [0:15] 为左声道, [16:31] 为右声道, 在 8bit 时 [0: 7] 为左声道, [8:15] 为右声道。此时在 DMA 模式下的数据宽度也需要作出对应的调整。在合并模式下, 可以通过 `i2s_fifo_config_0` 寄存器的 `cr_fifo_lr_exchg` 位控制交换左右声道输出。

14.3.5 静音模式

有些场景下，需要暂停播放，但是要保持播放进度持续变化，静音模式可以满足这种需求。通过 `i2s_config` 寄存器的 `cr_mute_mode` 进入静音模式，在静音期间 TX FIFO 中数据发送速度不变，但是发出的数据会被强制保持为各声道静音前的最后一个数据。

14.3.6 时钟源

I2S 的时钟源由 audio PLL 提供, 时钟中的分频器用于对时钟源进行分频，然后输出时钟信号来驱动 I2S 模块。如下图所示：

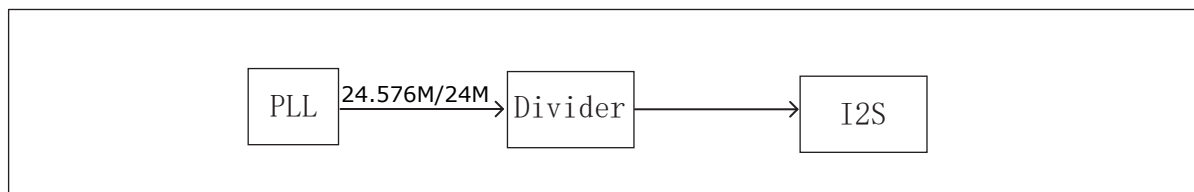


图 14.4: I2S 时钟

14.3.7 I2S 中断

I2S 有着丰富的中断控制，包括以下几种中断模式：

- TX FIFO 请求中断
 - 当 `I2S_FIFO_CONFIG_1` 中 `TX_FIFO_CNT` 大于 `TX_FIFO_TH` 时, 产生 TX FIFO 请求中断。当条件不满足时该中断标志会自动清除
- RX FIFO 请求中断
 - 当 `I2S_FIFO_CONFIG_1` 中 `RX_FIFO_CNT` 大于 `RX_FIFO_TH` 时, 产生 RX FIFO 请求中断。当条件不满足时该中断标志会自动清除
- overrun/underrun error 中断
 - 如果 TX/RX FIFO 发生了上溢或者下溢，会触发 error 中断，当异常消失后，标志位会自动清空

I2S 的所有中断的使能位以及中断标志位都在 `I2S_INT_STS` 寄存器

14.4 寄存器描述

名称	描述
<code>i2s_config</code>	I2S control
<code>i2s_int_sts</code>	Interrupt control
<code>i2s_bclk_config</code>	Frequency division control

名称	描述
i2s_fifo_config_0	FIFO control 0
i2s_fifo_config_1	FIFO control 1
i2s_fifo_wdata	TX FIFO
i2s_fifo_rdata	RX FIFO
i2s_io_config	IO control

14.4.1 i2s_config

地址：0x2000ab00

RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	cr_ofs_en	cr_ofs_cnt	cr_mono_rx_ch	cr_endian	cr_i2s_mode				
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_data_size	cr_frame_size	RSVD	RSVD	RSVD	cr_fs_ch_cnt	cr_fs_1t_mode	cr_mute_mode	cr_mono_mode	cr_i2s_rxd_en	cr_i2s_txd_en	cr_i2s_s_en	cr_i2s_m_en			

位	名称	权限	复位值	描述
31:26	RSVD			
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

位	名称	权限	复位值	描述
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			
8:7	cr_fs_ch_cnt	r/w	2'd0	Channel count of each frame 2'd0: FS 2-channel mode 2'd1: FS 3-channel mode (DSP mode only) 2'd2: FS 4-channel mode (DSP mode only) 2'd3: FS 6-channel mode (DSP mode only) Note: cr_mono_mode & cr_fifo_lr_merge will be invalid in 3-channel mode Note: frame_size must equal data_size in 3/4/6-channel mode
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
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

14.4.2 i2s_int_sts

地址: 0x2000ab04

RSVD	RSVD	RSVD	RSVD	RSVD	cr_i2s_fer_en	cr_i2s_rxf_en	cr_i2s_txf_en	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
RSVD	RSVD	RSVD	RSVD	RSVD	cr_i2s_fer_mask	cr_i2s_rxf_mask	cr_i2s_txf_mask	RSVD	RSVD	RSVD	RSVD	RSVD	i2s_fer_int	i2s_rxf_int	i2s_txf_int

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
0	i2s_txf_int	r	1'b1	I2S TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed

14.4.3 i2s_bclk_config

地址: 0x2000ab10

RSVD	RSVD	RSVD	RSVD	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
RSVD	RSVD	RSVD	RSVD	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)

14.4.4 i2s_fifo_config_0

地址：0x2000ab80

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
RSVD	RSVD	RSVD	RSVD	RSVD	cr_fifo_24b_lj	cr_fifo_lr_exchg	cr_fifo_lr_merge	rx_fifo_underflow	rx_fifo_overflow	tx_fifo_underflow	tx_fifo_overflow	rx_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

位	名称	权限	复位值	描述
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

14.4.5 i2s_fifo_config_1

地址: 0x2000ab84

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
31:28	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

14.4.6 i2s_fifo_wdata

地址: 0x2000ab88

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	TX FIFO write data port

14.4.7 i2s_fifo_rdata

地址: 0x2000ab8c

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	RX FIFO read data port

14.4.8 i2s_io_config

地址: 0x2000abfc

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	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	cr_deg_en	cr_deg_cnt	cr_i2s_bclk_inv	cr_i2s_fs_inv	cr_i2s_rxd_inv	cr_i2s_txd_inv	

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

15.1 简介

芯片内置一个单声道音频解调模块，可选 GPDAC 模拟输出模式与 PWM 输出模式，输出音频差分模拟信号实现音频播放。

15.2 主要特征

- 集成 1 路 PWM 调制输出，可输出一对占空比的值差分的 PWM 调制信号，经过外置 RC 滤波电路后，形成差分模拟信号
 - 采样率：8k~48k
 - 信噪比 (A-W)：95dB 增益
 - 谐波失真 + 噪声：-70dB @ 0dB 增益
- 可以将 GPDAC 模块作为音频模拟信号输出，将处理后的音频数据，直接由双通道的 GPDAC 模块转化为模拟差分信号输出
- 独立的数字音量控制，并支持斜率可调的渐变音量调节与渐变静音
- 支持 Mixer 混合器，支持双声道音源数据输入，并可以选择其中一个单声道输出或混合成一个声道后输出
- 32 位宽度 FIFO，深度为 32，支持 32bit/24bit/20bit/16bit 四种数据格式，支持单声道数据源与双声道混合数据源
- 支持 DMA 传输模式

- TX FIFO underrun 中断
 - 当 TX FIFO 中并没有数据，但是用户却通过 TX_FIFO_CTRL 中的 TX_CH_EN 使能了 TX FIFO 状态机，则会进入 TX FIFO underrun 中断。
- TX FIFO overrun 中断
 - 当 TX FIFO 已满，但继续向 FIFO 中填入数据时，会导致 TX FIFO 溢出，会产生 TX FIFO overrun 中断。
- 音量调节完成中断
 - Audio DAC 音量控制支持渐入/渐出的功能，当音量调节到设定值后，会触发音量调节完成中断，以通知渐入/渐出调节完成。

15.4.2 FIFO 格式控制

AUPWM_TX_FIFO_CTRL 可以控制音频数据储存在 FIFO 的格式，和设置音频数据源的通道数。FIFO 会根据数据格式与通道数设置，截取其中有效的 16bit 数据，并作为下一级 Mixer 混合器的对应通道的输入。

Audio DAC 支持如下四种数据存储格式，由 FIFO_CTRL[25:24] 来决定。

- Mode 0:
32bit 模式，DATA[15:0] = {FIFO[31:16]}，有效数据的最高位在 31 bits
- Mode 1:
24bit 模式，DATA[15:0] = {FIFO[23:8]}，有效数据的最高位在 23 bits
- Mode 2:
20bit 模式，DATA[15:0] = {FIFO[19:4]}，有效数据的最高位在 19 bits
- Mode 3:
16bit 模式，DATA[15:0] = {FIFO[15:0]}，有效数据的最高位在 15 bits

多数情况下，音频源数据是 16bit 的宽度时，选择 Mode3 即可。Audio DAC 的最大分辨率也为 16bits。其他模式存在的意义在于，如果待播放的音频数据大小为 32bit，其中有效数据宽度为 32/24/20 bits 时，用户就无需在软件中将数据转化为 16bit 模式，由 FIFO 自动转化，提高效率。

Audio DAC 的数据源通道数由 audac_fifo_ctrl 寄存器的 tx_ch_en 段控制，对应的效果如下：

- 0: 此时左右通道都关闭，FIFO 里的数据不会被传输到 Mixer 混合器，Audio DAC 停止播放
- 1: 单声道模式，此时 FIFO 里的数据全部被作为左声道数据，会被逐个输入到 Mixer 混合器的左声道中
- 2: 单声道模式，此时 FIFO 里的数据全部被作为右声道数据，会被逐个输入到 Mixer 混合器的右声道中
- 3: 双声道模式，此时 FIFO 里的奇数个数据被作为左声道数据，偶数个数据被作为右声道数据，分别输入到 Mixer 混合器的左右声道中

因为 Audio DAC 只支持一路声道输出，所以在多数情况下应当选择单声道音源数据。但在声源数据已经是双声道无法改变时，可以借助 Mixer 混合器，选择其中一路或者混合播放，详见后文中对 Mixer 混合器的介绍。

15.4.3 FIFO 的启动与 DMA 搬运

Audio DAC 的 TX FIFO 数据可以通过 DMA 进行搬运，开启 DMA 模式需要将 `audac_0` 寄存器的 `dac_itf_en` 位 DMA 接口使能置 1，和 `audac_fifo_ctrl` 寄存器的 `tx_drq_en` 位 DMA 请求使能位置 1。

通过配置 `audac_fifo_ctrl` 寄存器的 `tx_drq_cnt` 段来选择触发 DMA 请求的 FIFO 数据量阈值，有 4 种可选，分别为 8/16/32，和与 `tx_trg_level` 段配置的中断阈值相同。

用户可以通过 `audac_fifo_status` 寄存器的 `txa_cnt` 实时获得目前空闲的 FIFO 数量的数量。

当 FIFO 内空闲数量的值大于设定阈值，则会触发一次 DMA 搬运。

注意，如果 TX FIFO 里剩余的空闲数小于 DMA 一次搬入的数量，则触发 `overrun` 错误，如果在 Audio ADC 工作时 DMA 没有及时搬入数据，也会触发 `underrun` 错误，因此要注意 FIFO 阈值与 DMA burst 的配置。

15.4.4 可配置的 Mixer 声道混合器

Audio DAC 只支持播放一路音频播放通道，如果源数据是双声道的交叉混合数据，则可以使用 Mixer 混合器，选择其中一路音频播放，或者将两路音频混合后播放。由寄存器 `audac_1` 寄存器的 `dac_mix_sel` 段控制，支持以下 4 种模式：

- 仅选择左声道播放 (奇数次的数据)
- 仅选择右声道播放 (偶数次的数据)
- 将左右声道值相加后播放
- 取左右声道的平均值播放

注意，Mixer 混声器的声道配置，必须与 FIFO 的声道配置相互匹配，否则 Audio DAC 会停止工作。具体过程见下图所示。

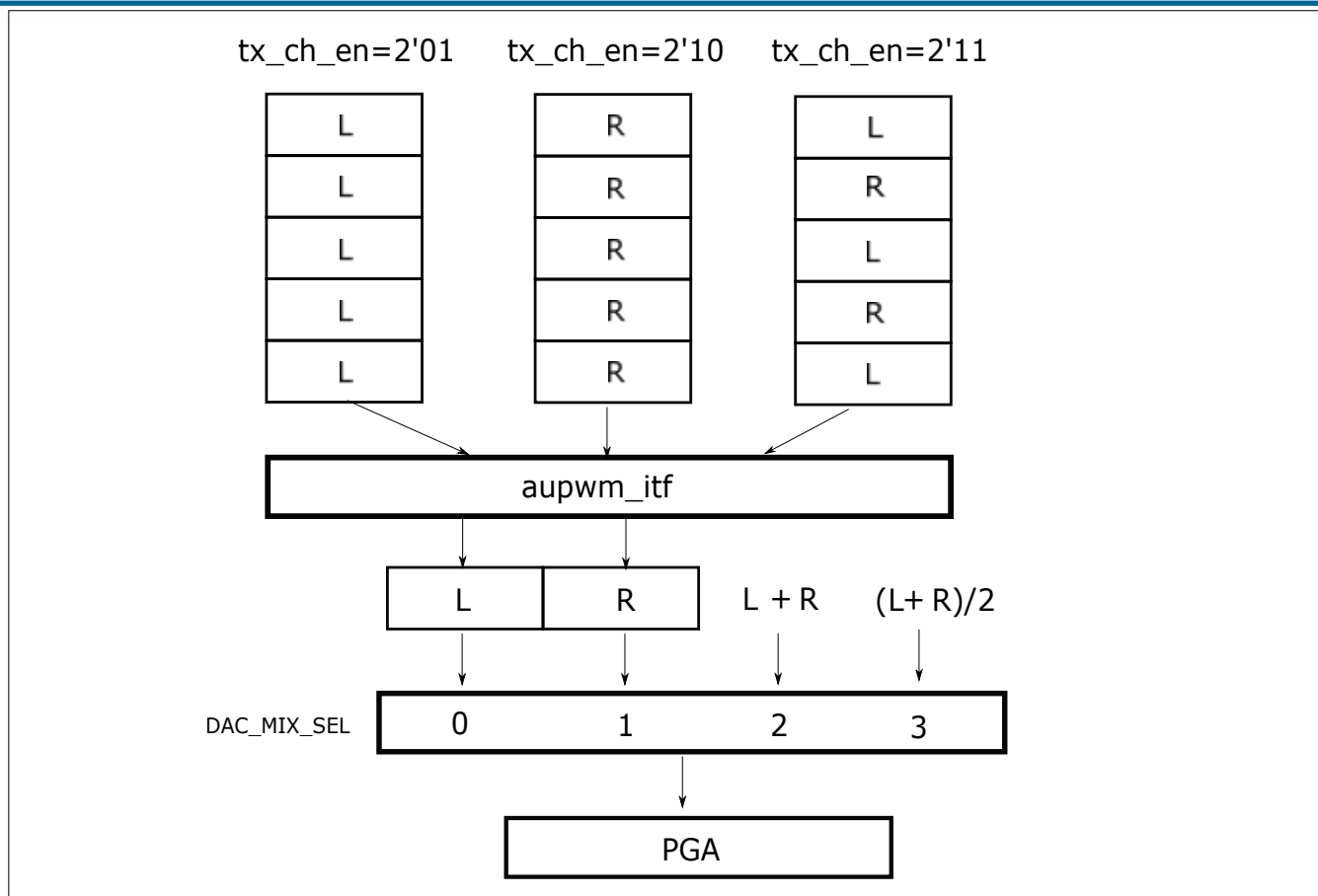


图 15.2: Mixer 示意图

当配置播放源为立体声的时候，此时 DMA 顺序搬来的数据会依次填入左右声道中，这要求数据源也是 L-R-L-R 交叉排序方式存储。此时可以通过 Mixer 选择器选择需要哪一路声道或者将两个声道求和或取平均进入下一级调制电路。

注意 tx_ch_en 和 Mixer 选择。如果通过 tc_ch_en 将单声道数据放入左声道，但是 mixer 却是选择右声道，会造成错误无法播放。

15.4.5 音量控制

用户可以通过 audac_s0 的 dac_s0_volume 寄存器配置音量，增益范围为-95.5dB-18dB，步进单位为 0.5dB。当 dac_s0_volume_update 寄存器为 1 时，音量才会被更新生效。

用户可以通过 dac_s0_mute_softmode，dac_s0_ctrl_mode 选择音量调节模式，有 3 种调节模式：

- 0: 直接改变，新的音量生效后，会被立即从原来的音量改变到新的音量配置。
- 1: 过零点渐变模式: 新的音量生效后，会从原来的音量逐渐线性变化到新的音量，并且只会在音频数据为 0 时变化音量争议。
- 2: 渐变模式，新的音量生效后，会从原来的音量逐渐线性变化到新的音量。

过零点渐变的斜率由 dac_s0_ctrl_zcd_rate 控制，渐变模式的斜率由 dac_s0_ctrl_rmp_rate 控制，可配置为每 2 到

2048 个采样点变化 0.5dB，采样点计算方式为 $2^{(n+1)}$ 次幂。当渐变中，音量试图改变但是长时间没有遇到零点数据时，超过 `dac_s0_ctrl_zcd_timeout` 设定的值时，会强行更新一个步进量 0.5dB。

另外还支持在不改变音量时，直接设置静音，同时静音也支持渐变模式，静音的渐变斜率与静音解除的渐变斜率分别为 `dac_s0_mute_rmpdn_rate` 与 `dac_s0_mute_rmpup_rate` 控制

15.4.6 ZeroDetect

AudioDAC 提供 ZeroDetect 功能，当打开此功能 (`audac_zd_0` 寄存器的 `zd_en` 位置 1) 时，如果 Mixer 混声器输出的数据一直为零，并且数量超过阈值 (`audac_zd_0` 寄存器的 `zd_time` 段)，则会关闭 DSM 调制器，保持输出为 0。这样做的目的是减少播零数据与静音时带来的底噪。

15.5 配置流程

1. 根据待播的音频采样率，选择对应采样率。
2. 根据实际需求，配置成 PWM 输出模式或 GPDAC 输出模式。GPDAC 模式时需要额外配置 DAC 模块，详见 DAC 说明文档。
3. 根据待播的音频声道数，配置 FIFO 与 Mixer 声道混合器。
4. 配置 DMA 将数据搬运到 AudioDAC TX FIFO。
5. 通过 `audac_fifo_ctrl` 的 `tx_ch_en` 使能通道，开始播放。
6. 在播放的过程中调整音量（可选）。

15.6 寄存器描述

名称	描述
<code>audac_0</code>	Clock control register
<code>audac_status</code>	Status register
<code>audac_s0</code>	Volume control register 1
<code>audac_s0_misc</code>	Volume control register 2
<code>audac_zd_0</code>	zero detect control register
<code>audac_1</code>	
<code>audac_fifo_ctrl</code>	fifo control register
<code>audac_fifo_status</code>	fifo status register
<code>audac_fifo_data</code>	fifo data register



15.6.1 audac_0

地址: 0x20055000

au_pwm_mode				ckg_ena											
				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
RSVD				RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	dac_itf_en	
				dac_0_en											

位	名称	权限	复位值	描述
31:28	au_pwm_mode	r/w	4'd0	pwm output mode,rang 0 6: 0:8KHz, 1:16KHz, 2:32KHz, 3:24KHz, 4:48KHz, 5:22.05KHz, 6:44.1KHz gpdac output mode,rang 9 14: 9:16KHz, 10:32KHz, 11:24KHz, 12:48KHz, 13:22.05KHz, 14:44.1KHz,
27	ckg_ena	r/w	1	enabl eclock gen
26:2	RSVD			
1	dac_itf_en	r/w	0	enable dac to audio dma interface
0	dac_0_en	r/w	0	enable dac ch0

15.6.2 audac status

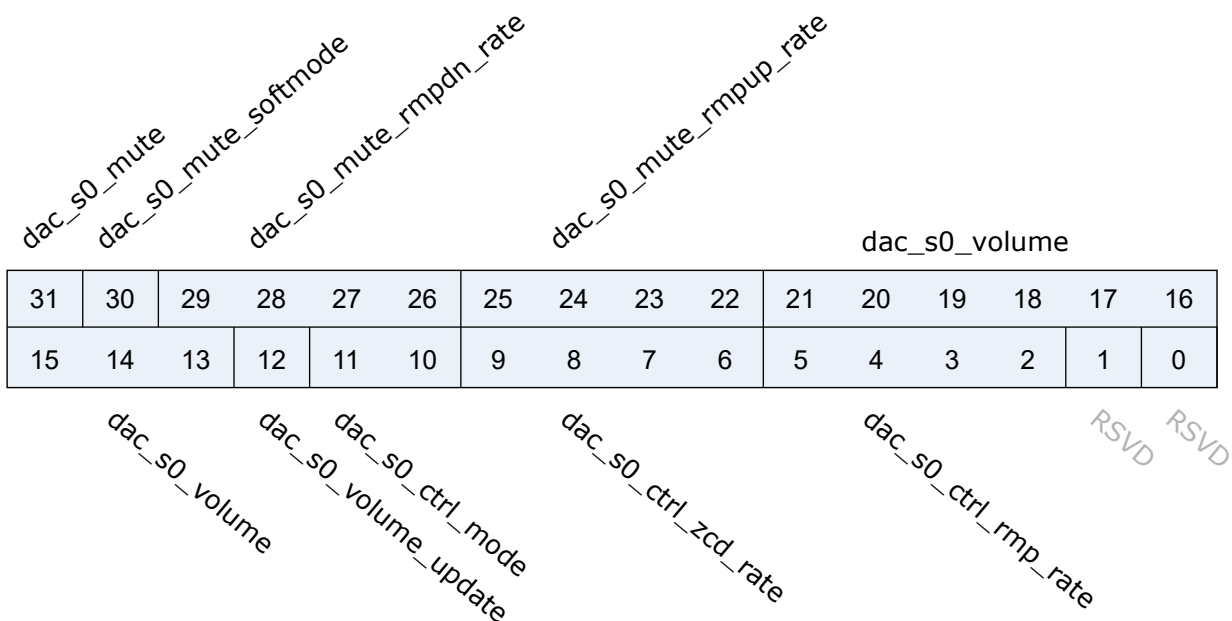
地址: 0x20055004

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
31:25	RSVD			
24	audio_int_all	r	0	mute signal to analog
23	zd_amute	r	0	zero detect signal to analog
22:18	RSVD			
17	dac_s0_int_clr	r/w	0	clear and close interrupt
16	dac_s0_int	r	0	mute done interrupt status
15:14	RSVD			
13	dac_h0_mute_done	r	1	dvga mute done
12	dac_h0_busy	r	0	dvga busy
11:0	RSVD			

15.6.3 audac_s0

地址: 0x20055008



位	名称	权限	复位值	描述
31	dac_s0_mute	r/w	0	dac dpga ch0 sw volume control 1:mute
30	dac_s0_mute_softmode	r/w	1	0:mute directly, 1:mute with ramp down

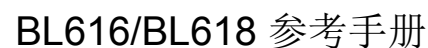
位	名称	权限	复位值	描述
29:26	dac_s0_mute_rmpdn_rate	r/w	4'd6	mute ramp down rate: 0:2 fs sample 1:4 fs sample 2:8 fs sample 3:16 fs sample 4:32 fs sample 5:64 fs sample 6:128 fs sample 7:256 fs sample 8:512 fs sample 9:1024 fs sample 8:2048 fs sample
25:22	dac_s0_mute_rmpup_rate	r/w	4'd0	mute ramp up rate 0:2 fs sample 1:4 fs sample 2:8 fs sample 3:16 fs sample 4:32 fs sample 5:64 fs sample 6:128 fs sample 7:256 fs sample 8:512 fs sample 9:1024 fs sample 8:2048 fs sample
21:13	dac_s0_volume	r/w	9'd0	volume s9.1, -95.5dB +18dB in 0.5dB step
12	dac_s0_volume_update	r/w	0	enable volume update
11:10	dac_s0_ctrl_mode	r/w	2'd2	0:direct force volume, 1:update volume at zero crossing, 2:update volume with ramp

位	名称	权限	复位值	描述
9:6	dac_s0_ctrl_zcd_rate	r/w	4'd2	zero crossing rate 0:2 fs sample 1:4 fs sample 2:8 fs sample 3:16 fs sample 4:32 fs sample 5:64 fs sample 6:128 fs sample 7:256 fs sample 8:512 fs sample 9:1024 fs sample 8:2048 fs sample
5:2	dac_s0_ctrl_rmp_rate	r/w	4'd6	ramp rate 0:2 fs sample 1:4 fs sample 2:8 fs sample 3:16 fs sample 4:32 fs sample 5:64 fs sample 6:128 fs sample 7:256 fs sample 8:512 fs sample 9:1024 fs sample 8:2048 fs sample
1:0	RSVD			

15.6.4 audac_s0_misc

地址: 0x2005500c

dac_s0_ctrl_zcd_timeout															
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	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD



15.6.5 audac_zd_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
31:17	RSVD			
16	zd_en	r/w	0	enable zero detect
15	RSVD			
14:0	zd_time	r/w	15'd512	number of zeros

15.6.6 audac_1

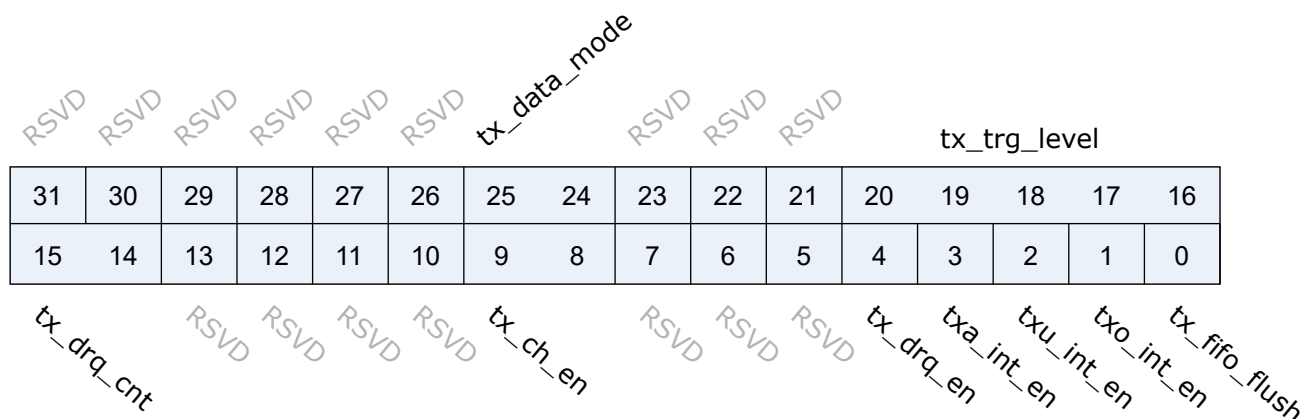
地址：0x20055014

RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	dac_dsm_dither_prbs_mode
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	
	dac_dsm_dither_en	dac_dsm_dither_amp	dac_dsm_scaling_en	RSVD	dac_dsm_scaling_mode	dac_dsm_order	dac_dsm_out_fmt	RSVD	RSVD	dac_mix_sel						
	dac_dsm_dither_prbs_mode															

位	名称	权限	复位值	描述
31:17	RSVD			
16:15	dac_dsm_dither_prbs_mode	r/w	0	dac dsm dither lfsr mode:0:LFSR32, 1:LFSR24, 2:LFSR16, 3:LFSR12
14	dac_dsm_dither_en	r/w	1	enable dac dsm dither
13:11	dac_dsm_dither_amp	r/w	0	dac dsm dither amplitue
10	dac_dsm_scaling_en	r/w	1	enable dac dsm scaling
9	RSVD			
8:7	dac_dsm_scaling_mode	r/w	0	dac dsm scaling value; u4.4
6:5	dac_dsm_order	r/w	0	0: 2-order, 1: 3-order
4	dac_dsm_out_fmt	r/w	0	offset binary 1:2's complement
3:2	RSVD			
1:0	dac_mix_sel	r/w	0	L channel, 1:R channel, 2: L+R, 3: (L+R)/2

15.6.7 audac_fifo_ctrl

地址：0x2005508c

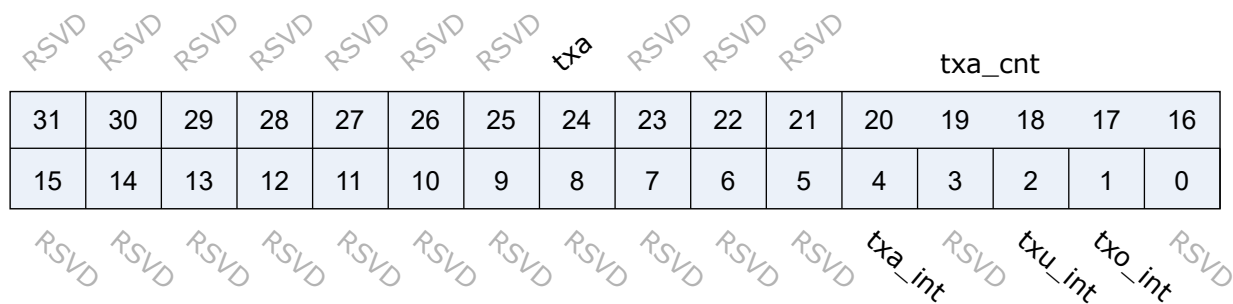


位	名称	权限	复位值	描述
31:26	RSVD			
25:24	tx_data_mode	r/w	2'b0	TX_FIFO_DATIN_MODE. TX FIFO DATA Input Mode (Mode 0, 1, 2, 3) Mode 0: Valid data's MSB is at [31] of TX_FIFO register Mode 1: Valid data's MSB is at [23] of TX_FIFO register Mode 2: Valid data's MSB is at [19] of TX_FIFO register Mode 3: Valid data's MSB is at [15] of TX_FIFO register For 16-bits transmitted audio sample: Mode 0: FIFO_I[15:0] = TXDATA[31:16] Mode 1: FIFO_I[15:0] = TXDATA[23:8] Mode 2: FIFO_I[15:0] = TXDATA[19:4] Mode 3: FIFO_I[15:0] = TXDATA[15:0]
23:21	RSVD			
20:16	tx_trg_level	r/w	5'd7	TX_FIFO_TRG_LEVEL. TX FIFO Trigger Level (TXTL[4:0]) Interrupt and DMA request trigger level for TX FIFO room available condition IRQ/DRQ Generated when WLEVEL > TXTL[4:0] Notes: WLEVEL represents the number of room available in the TX FIFO

位	名称	权限	复位值	描述
15:14	tx_drq_cnt	r/w	2'b0	DAC_DRQ_CLR_CNT. When TX FIFO available room less than or equal N, DRQ Request will be de-asserted. N is defined here: 00: IRQ/DRQ de-asserted when WLEVEL <= TXTL[4:0] 01: IRQ/DRQ de-asserted when WLEVEL < 2 10: IRQ/DRQ de-asserted when WLEVEL < 4 11: IRQ/DRQ de-asserted when WLEVEL < 8 WLEVEL represents the number of room available in the TX FIFO
13:10	RSVD			
9:8	tx_ch_en	r/w	2'b0	TX_FIFO_DATOUT_DST. TX FIFO Data Output Destination Select. 0: Disable 1: Enable Bit9: DAC2 data Bit8: DAC1 data When some of the above bits set to '1', these data are always arranged in order from low-bit to high-bit.(bit8->bit9)
7:5	RSVD			
4	tx_drq_en	r/w	1'b0	DAC_DRQ_EN. DAC FIFO Room Available DRQ Enable. 0: Disable 1: Enable
3	txa_int_en	r/w	1'b0	DAC_IRQ_EN. DAC FIFO Room Available IRQ Enable. 0: Disable 1: Enable
2	txu_int_en	r/w	1'b0	DAC_UNDERRUN_IRQ_EN. DAC FIFO Under Run IRQ Enable 0: Disable 1: Enable
1	txo_int_en	r/w	1'b0	DAC_OVERRUN_IRQ_EN. DAC FIFO Over Run IRQ Enable 0: Disable 1: Enable
0	tx_fifo_flush	w1c	1'b0	DAC_FIFO_FLUSH. DAC FIFO Flush. Write '1' to flush TX FIFO, self clear to '0'.

15.6.8 audac_fifo_status

地址：0x20055090



位	名称	权限	复位值	描述
31:25	RSVD			
24	txa	r	1'b1	TXA. TX FIFO Room Available 0: No room for new sample in TX FIFO 1: More than one room for new sample in TX FIFO (>= 1 word)
23:21	RSVD			
20:16	txa_cnt	r	5'd16	TXA_CNT. TX FIFO Available Room Word Counter
15:5	RSVD			
4	txa_int	r	1'b0	TXA_INT. TX FIFO Room Available Pending Interrupt 0: No Pending IRQ 1: Room Available Pending IRQ Automatic clear if interrupt condition fails.
3	RSVD			
2	txu_int	r	1'b0	TXU_INT. TX FIFO Underrun Pending Interrupt 0: No Pending IRQ 1: FIFO Underrun Pending IRQ Write '1' to clear this interrupt
1	txo_int	r	1'b0	TXO_INT. TX FIFO Overrun Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt
0	RSVD			

15.6.9 audac_fifo_data

地址: 0x20055094

tx_data

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

tx_data

位	名称	权限	复位值	描述
31:0	tx_data	w	32'h0	TX_DATA. Transmitting left, right channel sample data should be written this register one by one. The left channel sample data is first and then the right channel sample.

16.1 简介

芯片内置一个 AudioADC 模块，可实现单路录音功能，集成一路 16bit 高精度 ADC 用于模拟音频信号采集，此外支持 PDM 数字接口。

16.2 主要特征

- 集成 1 个 Sigma-Delta 高性能 ADC, 可支持差分或单端的模拟 mic 信号输入, 复用 GPIO 模拟输入
 - 采样率: 8k~96k
 - 信噪比 (A-W): 90dB @ 6dB 增益
 - 谐波失真 + 噪声: -80dB @ 6dB 增益
 - 模拟前置增益: 6~42 dB, 以 3dB 步进
 - ADC 的输入阻抗等模拟参数可配置
- 额外支持 PDM 数字接口, 可用于数字 mic 录音, 复用 GPIO 输入
- 除音频录制模式外, 额外支持直流测量模式, 将 AudioADC 作为高精度 ADC 使用, 支持差分与单端模式, 分辨率可达到 16bit
- 可调节的高通滤波器和独立的数字音量控制
- 32 位宽度 FIFO 深度为 32, 数据支持 32bit/24bit/20bit/16bit 四种存储格式
- 支持 DMA 传输模式

16.3 时钟树

用户需要先将 Audio PLL 的输出时钟配置到 491.52M 或 451.58M，分别对应 48K 系列与 44.1K 系列的采样率，选择 Audio PLL 输出的 5 分频作为 AudioDAC 时钟源，AudioADC 模块内，会根据工作模式与采样率设置，自动配置每个子模块的分频系数，无需手动设置分频系数。此寄存器数值与分频比如下所示

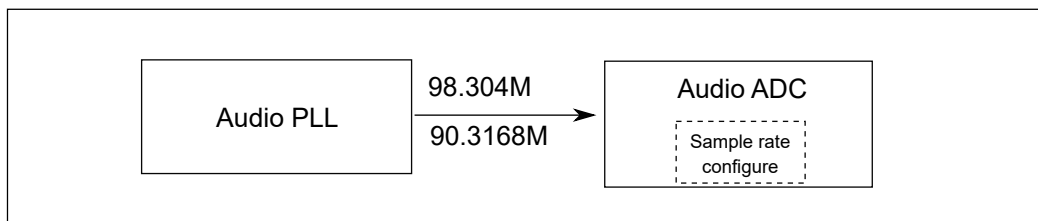


图 16.1: 时钟示意图

16.4 功能描述

AudioADC 模块基本框图如图所示。

BL616 ADUIO & ADDA

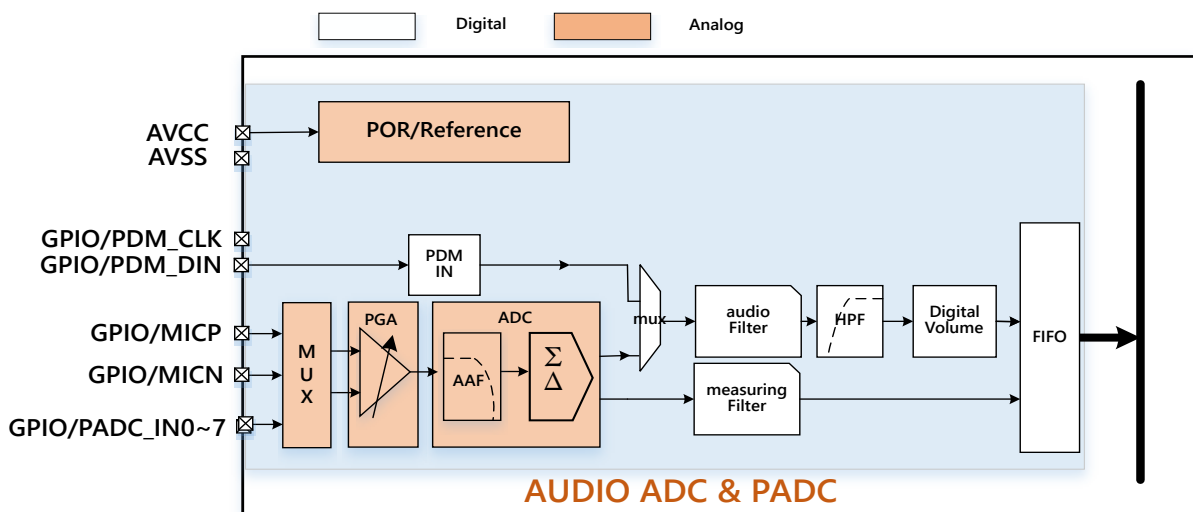


图 16.1: 模块框图

AUADC 模块输入既支持音频模拟信号，也支持 PDM 数字信号。当选择为 PDM 数字信号的时候，输入数据在经过 PDM 电路处理后，进入音频处理通道中。当选择为数字信号时，模拟信号会先经过 PGA 放大，再结果 ADC 量化处理后，进入音频处理通道中。音量处理通道会对初始数据使用 SINC 函数低通滤波和重采样，然后进行高通滤波与音量增益控制的处理，最后将数据推入 FIFO 中。

16.4.1 音源通道选择

AudioADC 模块支持模拟 mic 输入与 PDM 数字 mic 接口，其中 PDM 时序规范中，在时钟上升沿与下降沿的数据分别为左声道与右声道。当使用模拟 mic 输入时，需要配置内置 ADC 并选择好要在所集成的 ADC 的配置完成后，通过 `pdm_dac_0` 寄存器的 `adc_0_src` 位选择音源为 ADC 即可。当使用 PDM 数字 mic 时，需要通过 `pdm_dac_0` 中的 `pdm_0_en` 启动 PDM，再通过 `pdm_pdm_0` 中的 `adc_0_pdm_sel` 选择左声道或右声道。

16.4.2 AUADC 中断

AUADC 模块有着丰富的中断控制，包括以下几种中断模式：

- RX FIFO 阈值请求中断
- RX FIFO underrun 中断
- RX FIFO overrun 中断

当 FIFO 中的有效的数据量大于 `audadc_rx_fifo_ctrl` 中的 `rx_trg_level` 时，产生 RX FIFO 请求中断。当条件不满足时该中断标志会自动清除。FIFO 中有效数据量，可以在 `audadc_rx_fifo_status` 寄存器的 `rx_cnt` 段查询。

当 RX FIFO 中并没有数据，但 CPU 或 DMA 读取了 FIFO，则会产生 RX FIFO underrun 中断。

启动 AudioADC 后，如果 FIFO 中的数据没有及时读出，数据累积超过最大深度，会导致 RX FIFO 溢出，从而产生 RX FIFO overrun 中断。

16.4.3 FIFO 格式控制

`audadc_rx_fifo_ctrl` 寄存器中的 `rx_data_mode` 可以控制音频数据储存在 FIFO 中的格式。

FIFO 控制器支持如下四种数据存储格式：

- Mode 0:
 $DATA[31:0] = \{FIFO[15:0], 16'h0\}$
- Mode 1:
 $DATA[31:0] = \{8\{FIFO[15]\}, FIFO[15:0], 8'h0\}$
- Mode 2:
 $DATA[31:0] = \{12\{FIFO[15]\}, FIFO[15:0], 4'h0\}$
- Mode 3:
 $DATA[31:0] = \{16\{FIFO[15]\}, FIFO[15:0]\}$

最高有效位的分布

- Mode 0:
有效数据的最高位在 31 bits

- Mode 1:

有效数据的最高位在 23 bits

- Mode 2:

有效数据的最高位在 19 bits

- Mode 3:

有效数据的最高位在 15 bits

如果对储存格式没有特殊要求，一般来说选择 Mode3，因为 ADC 的分辨率为 16bit，使用 16bit 宽度去储存音频时内存的占用最低。其他格式会将有效的 16bit 数据，左移到 32bits 中的不同位置，低位使用 0 去补齐，高位使用符号位进行补齐，用以满足一些特殊的存储要求。

16.4.4 测量模式

AudioADC 中的高性能 ADC，除了可以音频模拟信号进行采样外，还支持作为高精度 ADC 使用，并且自带了增益可调节的 PGA 放大器，可用于单端与差分的微弱信号采集。测量模式需要在 `audpdm_top` 寄存器的 `adc_rate` 段中选择测量模式，在 `audadc_cmd` 寄存器中，`audadc_meas_filter_en` 位置 1 使能测量模式，在 `audadc_pga_mode` 段与 `audadc_pga_gain` 段中选择测量的模拟通道，在 `audadc_pga_mode` 段中配置 ADC 模式为直流差分或者直流单端模式，此时音频处理通道将会被旁路。测量模式下的 FIFO 格式等其他配置，与音频模式相同。

16.4.5 FIFO 的启动与 DMA 搬运

AudioADC 的 FIFO 数据可以通过 DMA 进行搬运。

用户可以通过 `PDM_RX_FIFO_STATUS` 寄存器实时获得目前 FIFO 有效数据的数量。

通过配置 `audadc_rx_fifo_ctrl` 中的 `rx_drq_cnt` 选择触发 DMA request 的 FIFO 阈值，有 4 种选择，8、16、32，或者与 `rx_trg_level` 配置的 FIFO 中断阈值相同。

当 FIFO count 的值大于设定阈值，并且 `AUDADC_RX_FIFO_CTRL[4]` DMA 模式被使能后，则会向 DMA 发起请求。

注意，启动 AudioADC 后，如果 FIFO 中的数据没有及时读出，当 FIFO 溢出后会触发 `overrun` 错误，同时会造成数据丢失，需要注意配置顺序。

16.5 配置流程

1. 选择录制音频采样率
2. 根据录制的信号源是否是 PDM 数字信号或者是模拟信号配置 `pdm_dac_0` 的 `adc_0_src` 寄存器
3. 如果是 pdm 格式，通过 `pdm_pdm_0` 的 `adc_0_pdm_sel` 选择 pdm 的声道
4. 配置 DMA 将 Audio 的 RX FIFO 数据实时搬运到指定区域
5. 通过 `audadc_rx_fifo_ctrl` 的 `rx_ch_en` 打开状态机开始录音

6. 在录制的过程中调整音量（可选）

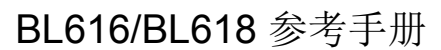
16.6 寄存器描述

名称	描述
audpdm_top	Clock control register
audpdm_itf	Interface control register 1
pdm_adc_0	Filter control register 1
pdm_adc_1	Filter control register 2
pdm_dac_0	Interface control register 2
pdm_pdm_0	pdm control register
pdm_adc_s0	volume control register
audadc_ana_cfg1	ADC analog control register 1
audadc_ana_cfg2	ADC analog control register 2
audadc_cmd	ADC control register
audadc_data	measuring mode control register
audadc_rx_fifo_ctrl	fifo control register
audadc_rx_fifo_status	fifo status register
audadc_rx_fifo_data	fifo data register

16.6.1 audpdm_top

地址：0x2000ac00

adc_rate				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												pdm_itf_inv_sel			
RSVD												adc_itf_inv_sel			
RSVD												audio_ckg_en			



16.6.2 audpdm_itf

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

@2024 Bouffalo Lab

16.6.3 pdm_adc_0

地址：0x2000ac08

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	adc_0_fir_mode

位	名称	权限	复位值	描述
31:1	RSVD			
0	adc_0_fir_mode	r/w	1'd0	adc fir mode

16.6.4 pdm_adc_1

地址：0x2000ac0c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	adc_0_k2_en	adc_0_k2	adc_0_k1_en	adc_0_k1						

位	名称	权限	复位值	描述
31:10	RSVD			
9	adc_0_k2_en	r/w	1'd0	adc ch0 hpf parameter k2 enable
8:5	adc_0_k2	r/w	4'd13	adc ch0 hpf parameter k2
4	adc_0_k1_en	r/w	1'd1	adc ch0 hpf parameter k1 enable
3:0	adc_0_k1	r/w	4'd8	adc ch0 hpf parameter k1

地址: 0x2000ac10

位	名称	权限	复位值	描述
31:13	RSVD			
12	adc_0_src	r/w	1'd0	0:adc mode, 1:pdm mode
11:10	RSVD			
9:6	adc_pdm_l	r/w	4'b1010	pdm low value
5:4	RSVD			
3:0	adc_pdm_h	r/w	4'b0110	pdm high value

地址: 0x2000ac1c

位	名称	权限	复位值	描述
31:6	RSVD			
5:3	adc_0_pdm_sel	r/w	3'd0	adc ch0 source select: 0:pdm_l, 1:pdm_r
2:1	RSVD			

位	名称	权限	复位值	描述
0	pdm_0_en	r/w	1'd0	enable pdm

16.6.7 pdm_adc_s0

地址: 0x2000ac38

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

adc_s0_volume

位	名称	权限	复位值	描述
31:9	RSVD			
8:0	adc_s0_volume	r/w	9'd0	volume s9.1, -95.5dB +18dB in 0.5dB step

16.6.8 audadc_ana_cfg1

地址: 0x2000ac60

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

audadc_sel_edge
audadc_ckb_en
audadc_pga_lp_en
audadc_ictrl_pga_mic
audadc_ictrl_pga_aaf
audadc_pga_nois_ctrl
audadc_pga_rhpa_sel
audadc_pga_chop_cfg
audadc_pga_chop_en
audadc_pga_chop_freq
audadc_pga_chop_cksel

位	名称	权限	复位值	描述
31:30	RSVD			
29	audadc_sel_edge	r/w	1'h0	ADC output data clock edge 0 = falling edge sent, rising edge recieve 1 = rising edge sent, falling edge recieve
28	audadc_ckb_en	r/w	1'h0	AUDADC clock phase control 0 = 0° 1 = 180°
27:25	RSVD			
24	audadc_pga_lp_en	r/w	1'h0	PGA lowpower funciton reversed, not realized in circuit
23:22	RSVD			
21:20	audadc_ictrl_pga_mic	r/w	2'h1	PGA_OPMIC bias current control 00 = 4uA 01 = 5uA 10 = 6uA 11 = 7uA
19:18	RSVD			
17:16	audadc_ictrl_pga_aaf	r/w	2'h1	PGA_OPAAF bias current control 00 = 4uA 01 = 5uA 10 = 6uA 11 = 7uA
15:14	RSVD			
13:12	audadc_pga_nois_ctrl	r/w	2'h0	PGA noise control when configured to single-ended not used
11:10	RSVD			
9:8	audadc_pga_rhpas_sel	r/w	2'h0	PGA high pass filter R control when configured to AC-coupled mode 00 = 480kΩ 01 = 320kΩ 10 = 160kΩ 11 = 4kΩ, fast startup
7	RSVD			
6:5	audadc_pga_chop_cfg	r/w	2'h3	control chopper for opmic&opaaf 00 = opmic off & opaaf off 01 = opmic off & opaaf on 10 = opmic on & opaaf off 11 = opmic on & opaaf on
4	audadc_pga_chop_en	r/w	1'h1	PGA chopper control 0 = disable 1 = enable

位	名称	权限	复位值	描述
3:1	audadc_pga_chop_freq	r/w	3'h4	PGA chopper frequency control @Fs=2048k 000 = 8k 001 = 16k 010 = 32k 011 = 64k 100 = 128k 101 = 256k 110 = 512k 111 = 1024k
0	audadc_pga_chop_cksel	r/w	1'h0	PGA chopper clock source selection 0 = adc clock 1 = synchronized clock from SDM not used

16.6.9 audadc_ana_cfg2

地址：0x2000ac64

RSVD																RSVD																audadc_reserved																RSVD																RSVD																RSVD																audadc_sdm_lp_en																RSVD																RSVD																audadc_ictrl_adc																RSVD																audadc_nctrl_adc1																																																																															
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																audadc_nctrl_adc2																RSVD																RSVD																RSVD																audadc_dem_en																RSVD																RSVD																audadc QUAN gain																audadc_dither_ena																audadc_dither_sel																audadc_dither_order																																																															

位	名称	权限	复位值	描述
31:30	RSVD			
29:28	audadc_reserved	r/w	2'h0	AUDADC reserved register
27:25	RSVD			
24	audadc_sdm_lp_en	r/w	1'h0	SDM lowpower funciton 0 = disable 1 = enable, 0.6 of disable
23:22	RSVD			
21:20	audadc_ictrl_adc	r/w	2'h1	SDM bias current control 00 = 4uA 01 = 5uA 10 = 6uA 11 = 7uA

位	名称	权限	复位值	描述
19	RSVD			
18:16	audadc_nctrl_adc1	r/w	3'h3	op number control for first integrator in SDM 000 = 1(12uA) 001 = 2(24uA) 010 = 3(36uA) 011 = 4(48uA) 100 = 5(60uA) 101 = 5(60uA) 110 = 5(60uA) 111 = 5(60uA)
15:14	RSVD			
13:12	audadc_nctrl_adc2	r/w	2'h1	op number control for second integrator in SDM 00 = 1(12uA) 01 = 2(24uA) 10 = 3(36uA) 11 = 3(36uA)
11:9	RSVD			
8	audadc_dem_en	r/w	1'h1	dem function control 0 = disable 1 = enable
7:6	RSVD			
5:4	audadc_quan_gain	r/w	2'h1	quantizer gain control for SDM 00 = Vref/14 01 = Vref/12 10 = Vref/10 11 = Vref/8
3	audadc_dither_ena	r/w	1'h1	dither control 0 = disable 1 = enable
2:1	audadc_dither_sel	r/w	2'h2	dither level control for SDM 00 = 0 01 = LSB*1/15 10 = LSB*2/15 11 = LSB*3/15
0	audadc_dither_order	r/w	1'h0	dither order control for SDM 0 = 0 order 1 = 1 order

16.6.10 audadc_cmd

地址: 0x2000ac68

RSVD																audadc_pga_pu		audadc_sdm_pu		audadc_conv		RSVD		RSVD		audadc_channel_en		RSVD		audadc_channel_selp		RSVD		audadc_channel_seln	
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		audadc_pga_mode				audadc_pga_gain				RSVD		audadc_audio_osr_sel		audadc_meas_filter_en		audadc_meas_filter_type		audadc_meas_odr_sel															

位	名称	权限	复位值	描述
31	RSVD			
30	audadc_pga_pu	r/w	1'h0	PGA related circuit enable 0 = disable 1 = enable
29	audadc_sdm_pu	r/w	1'h0	SDM related circuit enable 0 = disable 1 = enable
28	audadc_conv	r/w	1'h0	SDM conversion start singal 0 = remain resetting status 1 = start conversion both analog intetrator and measuring digital decimation filter will be reset when audadc_conv configured to low. Measuring digital decimation filter need to reset to same initial condition because it's feedback configuration for FIR. Audio filter dont need this resetting.
27:26	RSVD			
25:24	audadc_channel_en	r/w	2'h0	channel mux switch enable or disable MSB controls Positive channel, LSB controls Negative channel 0 = disable, look into each channel will see high impedance 1 = enable, one of eight channel will be choose
23	RSVD			

位	名称	权限	复位值	描述
22:20	audadc_channel_selp	r/w	3'h0	Positive channel selection, connected to PGA positive terminal 000 = AIN0 001 = AIN1 010 = AIN2 011 = AIN3 100 = AIN4 101 = AIN5 110 = AIN6 111 = AIN7
19	RSVD			
18:16	audadc_channel_seln	r/w	3'h0	Negative channel selection, connected to PGA negative terminal 000 = AIN0 001 = AIN1 010 = AIN2 011 = AIN3 100 = AIN4 101 = AIN5 110 = AIN6 111 = AIN7
15:14	RSVD			
13:12	audadc_pga_mode	r/w	2'h0	PGA mode configuration 00: AC-Coupled & differential-ended, Audio application 01: AC-Coupled & single-ended, Audio application 10: DC-Coupled & differential-ended, Measuring application 11: DC-Coupled & single-ended, Measuring application(may not used)
11:8	audadc_pga_gain	r/w	4'h0	PGA Gain control 0000 = 6dB 0001 = 6dB 0010 = 6dB 0011 = 9dB 0100 = 12dB 0101 = 15dB 0110 = 18dB 0111 = 21dB 1000 = 24dB 1001 = 27dB 1010 = 30dB 1011 = 33dB 1100 = 36dB 1101 = 39dB 1110 = 42dB 1111 = 42dB
7	audadc_audio_filter_en	r/w	1'h0	audio mode enable, audio filter is on when set to high 0 = disable 1 = enable
6	audadc_audio_osr_sel	r/w	1'h0	audio osr configuration 0 = 128 1 = 64
5	audadc_meas_filter_en	r/w	1'h0	measuring mode enable, measuring filter is on when set to high 0 = disable 1 = enable
4	audadc_meas_filter_type	r/w	1'h0	digital dicimation filter selection when in measuring mode 0 = SINC3 1 = Low-Latency

位	名称	权限	复位值	描述
3:0	audadc_meas_odr_sel	r/w	4'h3	audadc output data rate selection when configured to measuring mode 0000 = 2.5SPS 0001 = 5SPS 0010 = 10SPS 0011 = 20SPS 0100 = 25SPS 0101 = 50SPS 0110 = 100SPS 0111 = 200SPS 1000 = 400SPS 1001 = 800SPS 1010 = 1000SPS 1011 = 2000SPS 1100 = 4000SPS 1101 = 4000SPS 1110 = 4000SPS 1111 = 4000SPS

16.6.11 audadc_data

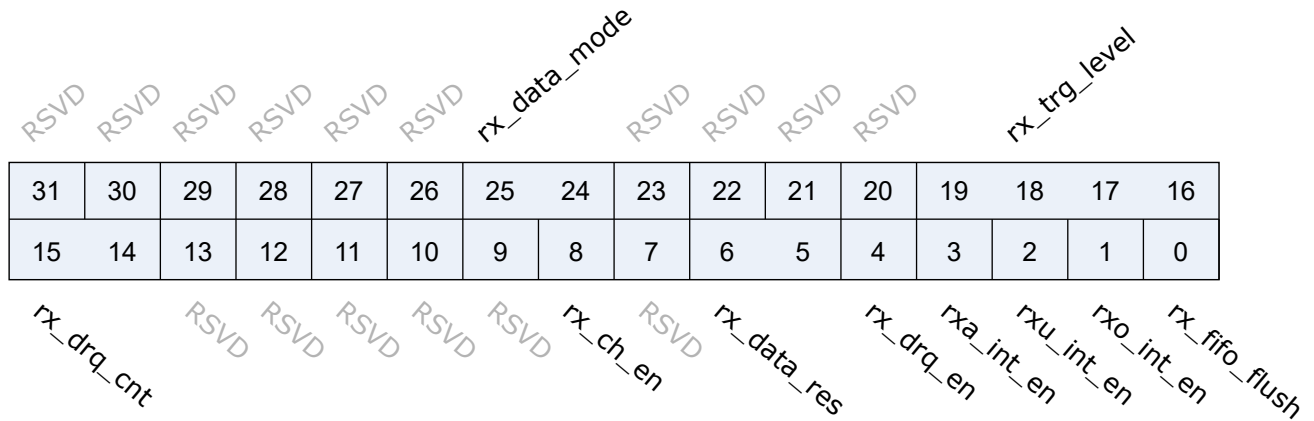
地址: 0x2000ac6c

audadc_valid_4s_en audadc_valid_4s_val audadc_soft_rst RSVD RSVD RSVD RSVD audadc_data_rdy audadc_raw_data															
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
audadc_raw_data															

位	名称	权限	复位值	描述
31:30	RSVD			
29	audadc_soft_rst	r/w	1'h0	don't care
28:25	RSVD			
24	audadc_data_rdy	r	1'h0	audadc data ready indicator when measuring mode selected, auto reset to 0 after read 0 = not ready 1 = ready
23:0	audadc_raw_data	r	16'h0	audadc output 16bit data, 2's

16.6.12 audadc_rx_fifo_ctrl

地址: 0x2000ac80



位	名称	权限	复位值	描述
31:26	RSVD			

位	名称	权限	复位值	描述
25:24	rx_data_mode	r/w	2'b0	RX_FIFO_DATOUT_MODE. RX FIFO DATA Output Mode (Mode 0, 1, 2, 3) Mode 0: Valid data's MSB is at [31] of RX_FIFO register Mode 1: Valid data's MSB is at [23] of RX_FIFO register Mode 2: Valid data's MSB is at [19] of RX_FIFO register Mode 3: Valid data's MSB is at [15] of RX_FIFO register Note: Expanding '0' at LSB of RX FIFO register (data invalid region) Expanding sign bit at MSB of RX FIFO register (data invalid region) For 24-bit received audio sample resolution: Mode 0: RXDATA[31:0] = FIFO_O[23:0], 8' h0 Mode 1: RXDATA[31:0] = 8FIFO_O[23], FIFO_O[23:0] Mode 2: RXDATA[31:0] = 12FIFO_O[23], FIFO_O[23:4] Mode 3: RXDATA[31:0] = 16FIFO_O[23], FIFO_O[23:8] For 20-bit received audio sample resolution: Mode 0: RXDATA[31:0] = FIFO_O[23:4], 12' h0 Mode 1: RXDATA[31:0] = 8FIFO_O[23], FIFO_O[23:4], 4' h0 Mode 2: RXDATA[31:0] = 12FIFO_O[23], FIFO_O[23:4] Mode 3: RXDATA[31:0] = 16FIFO_O[23], FIFO_O[23:8] For 16-bit received audio sample resolution: Mode 0: RXDATA[31:0] = FIFO_O[23:8], 16' h0 Mode 1: RXDATA[31:0] = 8FIFO_O[23], FIFO_O[23:8], 8' h0 Mode 2: RXDATA[31:0] = 12FIFO_O[23], FIFO_O[23:8], 4'h0 Mode 3: RXDATA[31:0] = 16FIFO_O[23], FIFO_O[23:8]
23:20	RSVD			
19:16	rx_trg_level	r/w	4'd3	RX_FIFO_TRG_LEVEL. RX FIFO Trigger Level (RXTL[3:0]) Interrupt and DMA request trigger level for RX FIFO Data Available condition IRQ/DRQ Generated when WLEVEL > RXTL[3:0] Notes: WLEVEL represents the number of valid samples in the RX FIFO

位	名称	权限	复位值	描述
15:14	rx_drq_cnt	r/w	2'b0	RX_DRQ_CLR_CNT. When RX FIFO available data less than or equal N, DRQ Request will be de-asserted. N is defined here: 00: IRQ/DRQ de-asserted when WLEVEL <= RXTL[3:0] 01: IRQ/DRQ de-asserted when WLEVEL < 1 10: IRQ/DRQ de-asserted when WLEVEL < 2 11: IRQ/DRQ de-asserted when WLEVEL < 4 WLEVEL represents the number of valid samples in the RX FIFO
13:9	RSVD			
8	rx_ch_en	r/w	1'b0	RX_FIFO_DATIN_SRC. RX FIFO Data Input Source Select. 0: Disable 1: Enable Bit8: ADC1 data
7	RSVD			
6:5	rx_data_res	r/w	2'd0	RX_SAMPLE_BITS. Receiving Audio Sample Resolution 0: 16 bits 1: 20 bits 2: 24 bits 3: Reserved
4	rx_drq_en	r/w	1'b0	ADC_DRQ_EN. ADC FIFO Data Available DRQ Enable. 0: Disable 1: Enable
3	rx_int_en	r/w	1'b0	ADC_IRQ_EN. ADC FIFO Data Available IRQ Enable. 0: Disable 1: Enable
2	rxu_int_en	r/w	1'b0	ADC_UNDERRUN_IRQ_EN. ADC FIFO Under Run IRQ Enable 0: Disable 1: Enable
1	rxo_int_en	r/w	1'b0	ADC_OVERRUN_IRQ_EN. ADC FIFO Over Run IRQ Enable 0: Disable 1: Enable

位	名称	权限	复位值	描述
0	rx_fifo_flush	w1c	1'b0	ADC_FIFO_FLUSH. ADC FIFO Flush. Write '1' to flush TX FIFO, self clear to '0' .

16.6.13 audadc_rx_fifo_status

地址: 0x2000ac84

RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	rsx	RSVD	RSVD	RSVD	RSVD	rsx_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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	rsx_int	RSVD	rsx_int	rsx_int

位	名称	权限	复位值	描述
31:25	RSVD			
24	rx_a	r	1'b0	RXA. RX FIFO Available 0: No available data in RX FIFO 1: More than one sample in RX FIFO (>= 1 word)
23:20	RSVD			
19:16	rx_a_cnt	r	4'h0	RXA_CNT. RX FIFO Available Sample Word Counter
15:5	RSVD			
4	rx_a_int	r	1'b0	RXA_INT. RX FIFO Data Available Pending Interrupt 0: No Pending IRQ 1: Data Available Pending IRQ Automatic clear if interrupt condition fails.
3	RSVD			
2	rx_u_int	r	1'b0	RXU_INT. RX FIFO Underrun Pending Interrupt 0: No Pending IRQ 1: FIFO Underrun Pending IRQ Write '1' to clear this interrupt

位	名称	权限	复位值	描述
1	rxo_int	r	1'b0	RXO_INT. RX FIFO Overrun Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt
0	RSVD			

16.6.14 audadc_rx_fifo_data

地址: 0x2000ac88

rx_data

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_data

位	名称	权限	复位值	描述
31:0	rx_data	r	32'h0	RX_DATA. RX Sample Host can get one sample by reading this register. The left channel sample data is first and then the right channel sample.

17.1 简介

EMAC(Ethernet Media Access Controller) 模块是一个支持 100Mbps 的以太网 MAC, 兼容 IEEE 802.3 以太网技术标准。EMAC 模块中包括状态及控制寄存器组, 数据收发模块, 数据收发缓冲描述符组, 主机接口, MDIO(Management Data Input/Output) 接口和物理层芯片 (PHY) 接口。

状态及控制寄存器组包含了 EMAC 的状态位及控制位, 该寄存器组是与用户程序的接口, 可以控制数据收发, 并检查 EMAC 状态。

数据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并添加前导或 CRC 数据, 经扩充后将该短帧发送给 PHY 并通过 PHY 发出; 或者是从 PHY 接收数据, 并根据收发缓冲描述符, 将数据放入指定内存。收发完成后设置相关的事件标志。如果使能了事件中断, 将产生中断信号请求主机进行处理。

MDIO 模块通过 RMII 接口与 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 RMII 接口的 PHY
- 支持通过 MDIO 接口与 PHY 交互
- 支持 100Mbps 以太网
- 支持半双工与全双工模式
- 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
- 支持数据帧前导生成及移除

- 发送时，支持自动扩展短的数据帧
- 支持检测过长或过短的数据帧 (长度限制)
- 支持传输长数据帧 (大于标准以太帧长度)
- 支持自动丢弃重发次数超限或帧间隙过小的数据包
- 支持广播包过滤
- 最大支持保存 128 个 BD(Buffer Descriptor)
- 在发送时，支持将一个数据包分拆配置到多个连续的 BD
- 支持发送/接收的各种事件标志，并支持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。

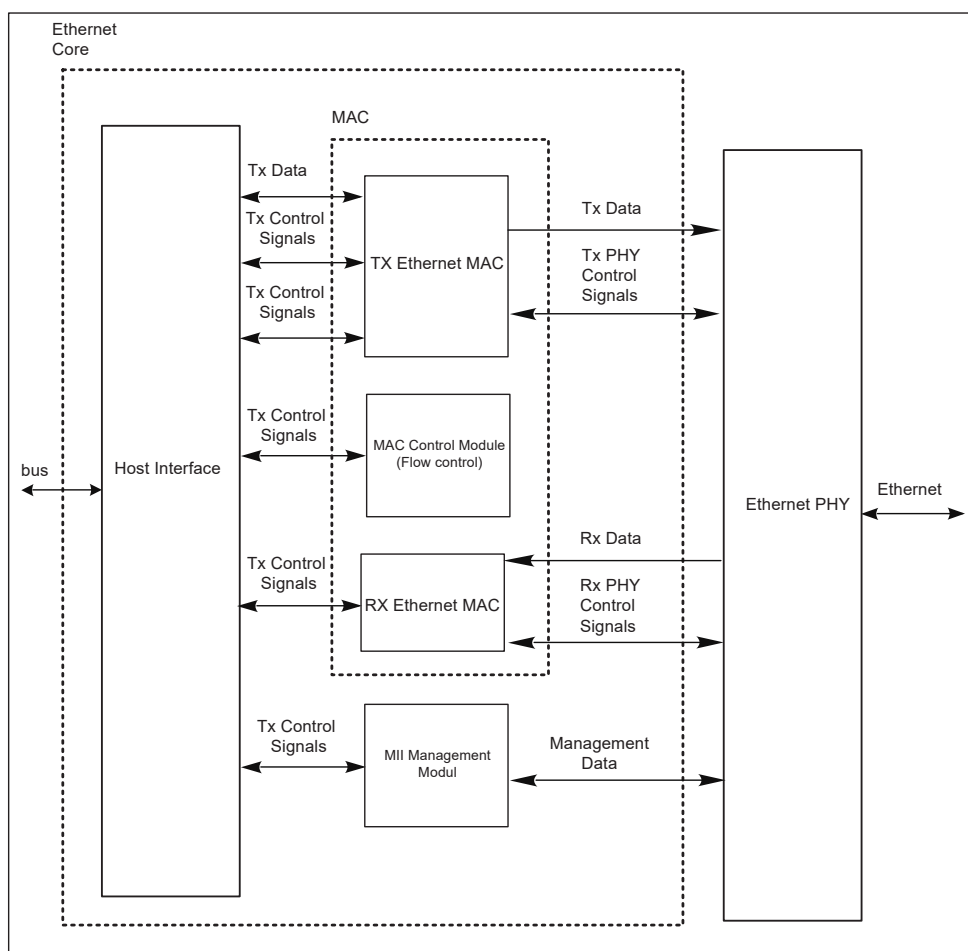


图 17.1: EMAC 框图

模块的控制寄存器通过 MDIO 接口，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。

作。接收模块过滤并检查收到的数据帧：是否有合法的前导，帧校验序列 (FCS) 检查，长度检查等，并根据缓冲描述符 (BD)，将数据存放到指定内存地址。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，帧校验序列 (FCS)，和 PAD 以及其他需要填充的信息，然后根据 CSMA/CD 协议，将数据发出。如果检测到 CRS(Carrier Sense) 载波侦测信号，将会延迟重试。收发缓冲描述符组从系统 RAM 中分配，此 RAM 空间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。描述符最大一共有 128 组，可用于发送描述符或者接收描述符，可以灵活分配。

17.4 时钟

EMAC 模块需要一路时钟用于同步收发 (100Mbps 时需要 25MHz(MII) 或 50MHz(RMII) 时钟，BL618 目前仅支持 RMII 接口)。此时钟必须在 EMAC 与 PHY 之间同步。BL618 支持输出 50MHz 参考时钟源，也可以使用外置的参考时钟源输入。

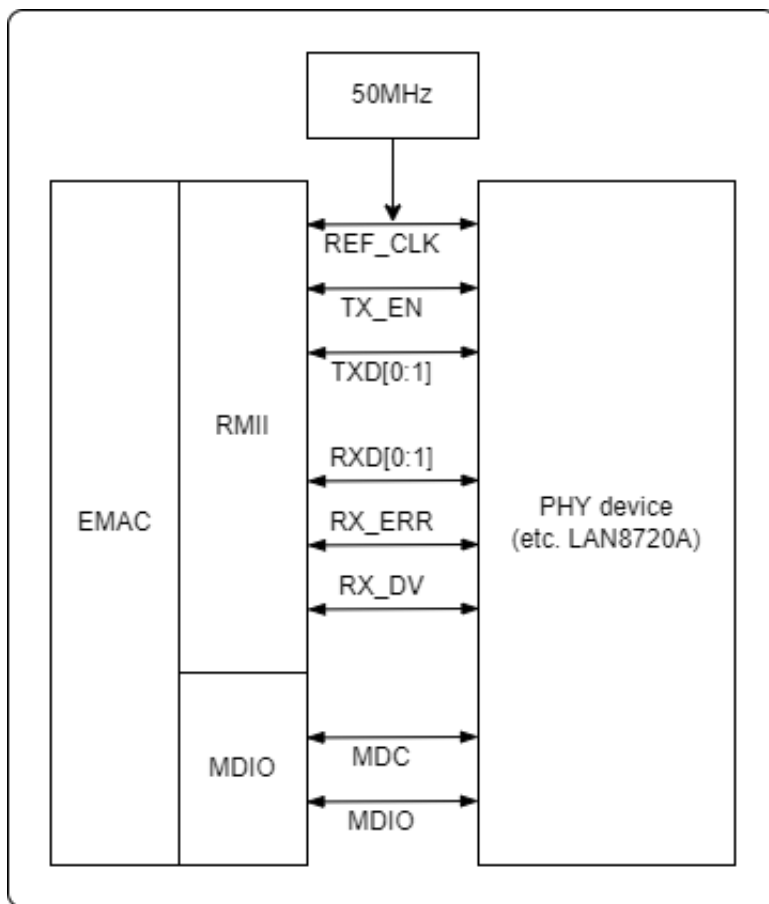


图 17.1: 使用外置的参考时钟源

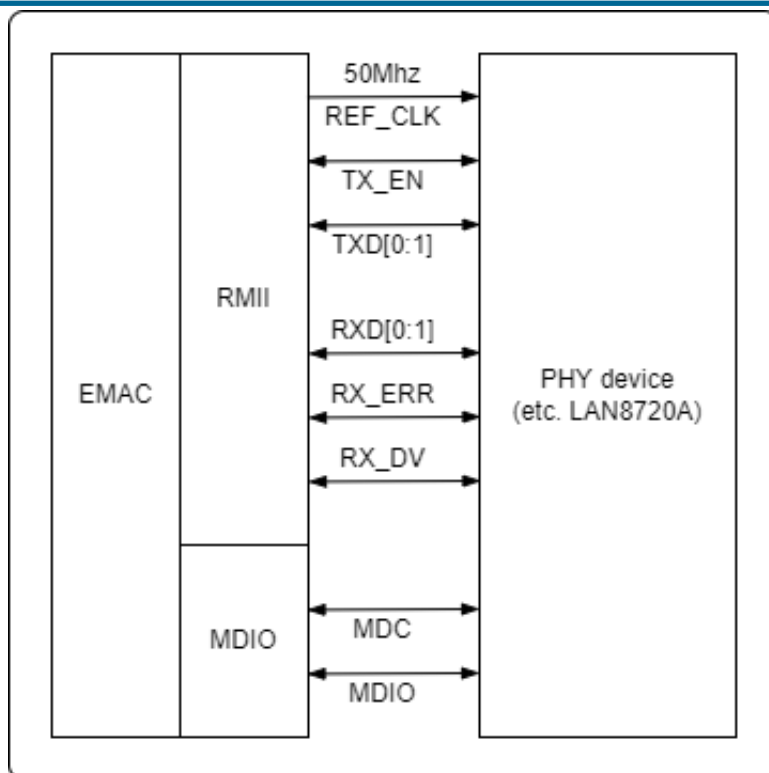


图 17.2: 内部输出 50MHz 参考时钟源

17.5 收发缓冲描述符 (BD, Buffer Descriptor)

收发缓冲描述符，用于提供 EMAC 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 word0 提供了本 buffer 包含的数据帧的长度，控制及状态位；高地址的 word1 是对应的数据帧缓存的 32 位内存地址 (指针)。

发送 TXBD 中的 word0 具体描述：

[31:16]: 发送数据包长度 (LEN)。

[15]: 发送 BD 准备好标志 (RD)。软件写 1 通知 EMAC 此 BD 包含需要发送的数据，硬件写 0 说明此 BD 数据已发送完成或有错误发生。

[14]: 中断使能标志位 (IRQ)。置位则此 BD 可以触发 TXE 或 TXB 中断。设置为 0 时，传输完成后不产生中断。

[13]: 回环标记 (WR)。置位标志着此 BD 是最后一个发送 BD，硬件将从起始的第一个 BD 开始重新循环发送。

[12]: 填充标志 (PAD)。若置位，则在 EMAC 中设置了填充使能，那么发送 BD 将会自动填充过短的数据包。

[11]: 校验标志 (CRC)。若置位，则 EMAC 会自动计算发送数据包的 CRC 值，并附在数据包中。

[10]: 帧结束标志 (EoF)。此位为 0 则表示一帧数据还未传完会继续取下一个 BD，若此位置位为 1 则此标志着这帧数据传输结束。如果一帧数据占用多个 BD，那么就会在一帧数据传完之后才会置位。

[8]: underfun 标志 (UR)。置位则说明 BD 的发送过程中发生了 FIFO underrun 的错误。

[7:4]: 重发次数计数器 (RTRY)。记录了重发次数。

[3]: 重发次数超限标志 (RL)。置位则表明重发次数超过了 COLLCONFIG 寄存器中配置的最大重发次数 MAXRET。

[2]: Late Collision 标志 (LC)。置位则表明在发送此 BD 时发生了 Late Collision，传输被中止了。

[1]: Defer Indication 标志 (DF)。置位则表明此包被延迟发送。

[0]: 载波检测失效 (CS)。如果在发送过程中检测不到载波，则置位。

接收 RXBD 中的 word0 具体描述:

[31:16]: 发送数据包长度 (LEN)。

[15]: 接收 BD 空标志 (RD)。置位说明此 BD 为空 (没保存收到的数据)，清零说明此 BD 有接收到的数据，或接收过程中有错误发生。

[14]: 中断使能标志位 (IRQ)。置位则此 BD 可以触发 RXE 或 RXB 中断。若设置为 0 时，接收完成后不产生中断。

[13]: 回环标记 (WR)。置位标志着此 BD 是最后一个接收 BD，硬件将从接收起始的第一个 BD 开始重新循环接收。

[8]: 控制帧标志 (CF)。置位则说明此 BD 接收到了一个 Control Frame。清零表示收到的是正常的帧。

[7]: miss 标志 (M)。若在混杂模式收到了数据包，但被内部地址逻辑标记为 miss，则 EMAC 设置此标志位。否则清零了此标志位，表示是由于地址匹配正确而接收的数据包。

[6]: overrun 标志 (OR)。置位则说明接收过程中发生了 FIFO overrun 错误。

[5]: 接收错误标志 (RE)。置位则说明接收过程中收到了 PHY 发出的 RX ERR 信号。

[4]: Dribble Nibble 标志 (DN)。置位则说明接收到了奇数个 nibbles。

[3]: 数据包过长标志 (TL)。置位则说明接收的数据包太长，超过了 PACKETLEN 寄存器的设置。

[2]: 数据包过短标志 (SF)。置位则表明接收到的数据包长度小于允许的最小值。

[1]: CRC 错误标志 (CRC)。置位则表明接收到的数据包 CRC 校验失败。

[0]: Late Collision 标志 (LC)。置位则表明在接收数据到此 BD 时发生了 Late Collision。

需要注意的是：对于 BD，需要按 word 写入。

EMAC 模块最大支持 128 个 BD，由于发送/接收逻辑共享所有可用的 BD，因此 TXBD 和 RXBD 个数可自由组合，但是总个数不能超过 128 个。需要注意的是发送 TXBD 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与 PHY 交互需要的命令及数据通信的方式。EMAC 通过 MDIO 接口控制 PHY 的工作模式，并保证两者的工作模式匹配 (速率，全/半双工等)。

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 位选择：当此 bit 为 1，则选择 RMII 模式，否则就是 MII 模式，但是由于 BL618 的引脚限制，暂不支持 MII 模式。

目前的 RMII 模式仅支持 IEEE 802.3u 标准中指定的 100Mbps 的传输速率。

RMII 的传输信号描述见下表。

表 17.1: 传输信号

名称	MI	RMII
EXTCK_EREFC	ETXCK: 发送时钟信号	EREFC: 参考时钟
ECRS	ECRS: 载波探测	-
ECOL	ECOL: 碰撞检测	-
ERXD	ERXD: 数据 valid	ECRSD: 载波检测/数据 valid
ERX0-ERX3	ERX0-ERX1: 2-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-ETX1: 2-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
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 EMAC 工作流程

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 寄存器中
- 通过设置 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 寄存器发送命令控制 PHY 芯片

- 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，根据协商结果设置相应的模式到 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 128-TXBDNUM 的就是 RXBDNUM 的最大可用值，但是由于内存空间的限制一般情况下 RXBDNUM 也需要设定一个符合应用场景的合理值。
- 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 (word1) 中
- 清空对应发送 BD 的控制与状态域 (word0) 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
- 如果一个数据包只包含在一个 BD 中，那么需要设置其 EOF 位为 1
- 如果一个数据包分在多个 BD 里进行发送，那么只需要将其占用的最后一个 BD 标记为数据包结束 (设置 EOF 位)
- 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 的状态位进行相应的处理
- 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 128-TXBDNUM 的就是 RXBDNUM 的最大可用值，但是由于内存空间的限制一般情况下 RXBDNUM 也需要设定一个符合应用场景的合理值。
- 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 (word1) 中
- 清空对应发送 BD 的控制与状态域 (wrod0) 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 Empty 位域，告知 EMAC 此 BD 空闲，可以用于数据接收；如需要还可设置 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 Empty 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 Empty 位，此 BD 即可再次用于接收

17.8 寄存器描述

名称	描述
MODE	ethernet mac mode setting register
INT_SOURCE	interrupt control register
INT_MASK	interrupt mask register
IPGT	inter packet gap register
PACKETLEN	packet length control register
COLLCONFIG	collision and retry config register
TX_BD_NUM	transmit buffer number register
MIIMODE	MII mode config register

名称	描述
MIICOMMAND	MII command config register
MIIADDRESS	MII address config register
MIITX_DATA	MII transmit data register
MIIRX_DATA	MII read data register
MIISTATUS	MII status register
MAC_ADDR0	ethernet mac address0 register
MAC_ADDR1	Ethernet mac address1 register
HASH0_ADDR	hash0 address register
HASH1_ADDR	hash1 address register
TXCTRL	Transmit control register

17.8.1 MODE

地址：0x20070000

RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	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	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.

位	名称	权限	复位值	描述
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.
4	RSVD			
3	BRO	r/w	1'b1	Broadcast address enable 0: Reject all frames containing the broadcast address unless the PRO bit is asserted. 1: Receive all frames containing broadcast address.
2	NOPRE	r/w	1'b0	No preamble mode 0: 7-byte preamble will be sent. 1: No preamble will be sent.
1	TXEN	r/w	1'b0	Transmit enable 0: Transmitter is disabled. 1: Transmitter is enabled. If TX_BD_NUM equals 0x0 (zero buffer descriptors are used), then the transmitter is disabled regardless of TXEN.

位	名称	权限	复位值	描述
0	RXEN	r/w	1'b0	Receiver enable 0: Receiver is disabled. 1: Receiver is enabled. If TX_BD_NUM equals 0x80 (all buffer descriptors are used for TX), then the receiver is disabled regardless of RXEN.

17.8.2 INT_SOURCE

地址：0x20070004

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RXC	TXC	BUSY	RXE	RXB	TXE	TXB

位	名称	权限	复位值	描述
31:7	RSVD			
6	RXC	r/w	1'b0	Receive control frame This bit indicates that the control frame was received. It is cleared by writing 1 to it. Bit RXFLOW in the CTRLMODE register must be set to 1 in order to get the RXC bit set.
5	TXC	r/w	1'b0	Transmit control frame This bit indicates that a control frame was transmitted. It is cleared by writing 1 to it. Bit TXFLOW in the CTRLMODE register must be set to 1 in order to get the TXC bit set.
4	BUSY	r/w	1'b0	Busy 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. This bit appears regardless to the IRQ bits in the Receive Buffer Descriptor.

位	名称	权限	复位值	描述
3	RXE	r/w	1'b0	Receive error This bit indicates that an error occurred while receiving data (overrun, receiver error, dribble nibble, too long, >64K, CRC error, bus error or late collision. It is cleared by writing 1 to it. This bit appears only when IRQ bit is set in the Receive Buffer Descriptor.
2	RXB	r/w	1'b0	Receive frame This bit indicates that a frame was received. It is cleared by writing 1 to it. This bit appears only when IRQ bit is set in the Receive Buffer Descriptor.
1	TXE	r/w	1'b0	Transmit error 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. This bit appears only when IRQ bit is set in the Transmit Buffer Descriptor.
0	TXB	r/w	1'b0	Transmit buffer This bit indicates that a buffer has been transmitted. It is cleared by writing 1 to it. This bit appears only when IRQ bit is set in the Transmit Buffer Descriptor.

17.8.3 INT_MASK

地址：0x20070008

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RXC_M	TXC_M	BUSY_M	RXE_M	RXB_M	TXE_M
															TXB_M

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：0x2007000c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	IPGT					

位	名称	权限	复位值	描述
31:7	RSVD			

位	名称	权限	复位值	描述
6:0	IPGT	r/w	7'h18	Inter packet gap The recommended value is 0x18 (24 clock cycles), which equals 0.96 us for 100 Mbps mode

17.8.5 PACKETLEN

地址: 0x20070018

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.
15:0	MAXFL	r/w	16'h600	Maximum frame length 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). For bigger packets, you can assert the HUGEN bit or increase the value of MAXFL field.

17.8.6 COLLCONFIG

地址: 0x2007001c

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

COLLVALID

位	名称	权限	复位值	描述
31:20	RSVD			
19:16	MAXRET	r/w	4'hF	Maximum retry This field specifies the maximum number of consequential retransmission attempts after the collision is detected. When the maximum number has been reached, the TX MAC reports an error and stops transmitting the current packet. According to the Ethernet standard, the MAXRET default value is set to 0xf (15).
15:6	RSVD			
5:0	COLLVALID	r/w	6'h3F	Collision valid 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.

17.8.7 TX_BD_NUM

地址: 0x20070020

RSVD								RSVD							
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
RSVD								TXBDNUM							
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD							

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			

位	名称	权限	复位值	描述
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：0x20070028

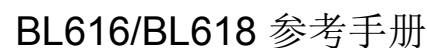
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															
RSVD															

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：0x2007002c

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										WCTRLDATA					
RSVD										RSTAT		SCANSTAT			



17.8.10 MIIADDRESS

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

BL616/BL618 参考手册

17.8.11 MIITX_DATA

地址: 0x20070034

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

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: 0x20070038

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

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：0x2007003c

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	MIIM_BUSY	MIIM_LINKFAIL

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：0x20070040

MAC_B2								MAC_B3							
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_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：0x20070044

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
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：0x20070048

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：0x2007004c

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: 0x20070050

RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	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,是一个外部总线标准,用于规范电脑与外部设备的连接和通讯。BL616/BL618 支持 USB2.0 (HighSpeed + FullSpeed),可作为主机控制器 (Host) 或者外围控制器 (Device)。作为主机控制器时,它包含一个支持所有速度事务的 USB 主机控制器。在没有软件干预的情况下,主机控制器可以自行处理基于事务的数据结构以减轻 CPU 的负载并自动在 USB 总线上发送和接收数据。当作为外围控制器时,除端点 0 外的每个端点都支持 USB 规范的传输类型,以适配各种应用类型。

18.2 主要特征

主要特征可以分为两类:通用特征、主机模式、从机模式

18.2.1 通用特征

- 兼容 USB2.0 标准
- 内嵌高速 PHY
- 可通过软件配置成主机角色或者从机角色

18.2.2 主机模式特征

- 兼容 EHCI 1.0 标准 (不支持 FSTN 和 SITD)
- 支持枚举高速、全速、低速设备

18.2.3 从机模式特征

- 支持高速、全速设备
- 1 个双向控制端点 0
- 4 个双向端点（支持 IN/ OUT），可配置为批量传输、中断传输、同步传输
- 内置专用 DMA (DMA, VDMA)，不支持直接读取 FIFO
- 共享 4 个 512 字节 fifo，可灵活配置成单缓冲、双缓冲、三缓冲模式
- 支持 suspend，resume 和 remote wake-up 功能
- 支持软断开功能

18.3 功能描述

USB 模块系统框图如下：

18.3.1 USB 时钟

18.3.2 USB 设备

18.3.2.1 USB 设备端点

18.3.2.2 USB 设备 FIFO

18.3.2.3 USB 设备 VDMA

18.3.2.4 USB 设备中断

18.3.2.5 USB 设备枚举过程

18.3.3 USB 主机

19.1 简介

CAM(Camera) 模块负责将并行接口 (DVP) 转换成 AHB 接口，把图像传感器生成的像素数据写入系统内存中，作为后续影像传输或压缩使用。CAM 模块拥有灵活的输出格式配置，可以满足多种多样的图像处理需求。

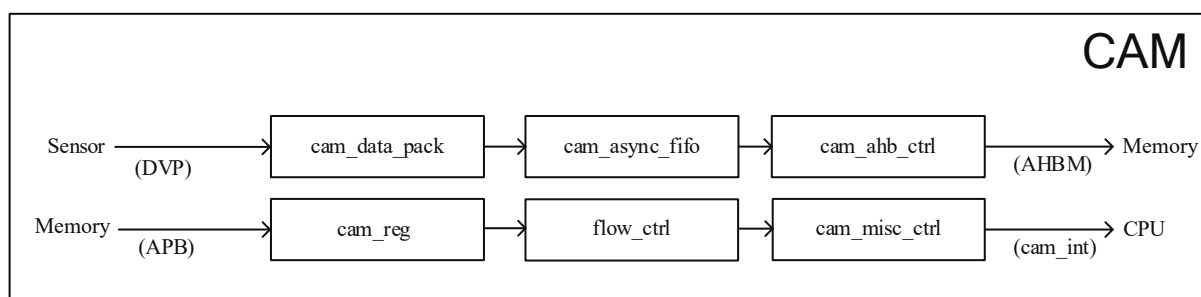


图 19.1: Cam 框图

19.2 主要特征

- 并行接口 8-bitDVP 信号，高速数据传输 (80M)，可配置 DVP 信号有效水平与逻辑组合
- 支持 8-bit/16-bit/24-bit 输入像素位宽，输入格式支持 YUV422、YUV400、RGB565、RGB888
- 输出格式支持 YUV422、YUV420、YUV400、RGB565、RGB888、RGB888 转 RGB565、RGB888 转 ARGB8888
- 支持视频模式和照片模式
- 可配置丢弃模式，包括：
 - 丢弃奇数位数据
 - 丢弃偶数位数据
 - 丢弃奇数行的奇数位数据
 - 丢弃奇数行的偶数位数据

- 可配置的图像传感器行帧同步信号选择和极性选择
- 支持图像矩形裁剪
- 以 1~32 为周期的帧取舍功能
- 支持行帧同步信号的完整性检测
- AHB 总线通信接口
- 512 字节缓存 FIFO 以应对总线偶而忙碌的状态
- 连续缓存多达 4 组图像信息
- 多种应用中断，有利于弹性使用与出错提示

19.3 功能描述

19.3.1 DVP(Digital Video Port) 信号与配置

DVP (Digital Video Port) 是并行接口，主要有时钟，帧同步，行同步与 8-bit 数据管脚，时钟的极限限制在 80MHz，所以一般用于 5MP 以下分辨率 Sensor。芯片内可独立配置帧同步与行同步的有效电平，并在有效数据上提供四种模式：

- A. 帧同步与行同步同时有效 (“&” 逻辑)
- B. 帧同步或行同步择一有效 (“|” 逻辑)
- C. 帧同步有效
- D. 行同步有效

19.3.2 YCbCr 格式

亮度信号被称为 Y，色度信号是由两个互相独立的信号组成。视颜色系统和格式不同，两种色度信号经常被称为 U、V 或 Pb、Pr 或 Cb、Cr。这是由不同的编码格式产生的，但实际上它们的概念基本相同。

由于人类视网膜上识别亮度的视网膜杆细胞要多于识别色度的视网膜锥细胞，人眼对亮度的敏感程度要高于对色度的敏感程度。所以可以将色度信息丢弃一部分而不被人眼所察觉。

色度信号分辨率最高的格式是 4:4:4，即每 4 点 Y 采样对应了 4 点 Cb 和 4 点 Cr 采样。而 4:2:2 是每 4 点 Y 采样对应了 2 点 Cb 和 2 点 Cr 采样，在这种格式中，色度信号的扫描线数量和亮度信号一样多，但是每条扫描线上的色度采样点却只有亮度信号的一半。与上面提到的格式不同，4:2:0 并不是每 4 点 Y 采样对应 2 点 Cb 和 0 点 Cr 采样，而是每 4 点 Y 采样对应 1 点 Cb 和 1 点 Cr 采样。4:0:0 是丢弃所有的色度信息，即灰度图。

19.3.3 视频模式/照片模式

照片模式下当软件给的固定大小的存储器被写满时，CAM 会停止，需要软件进行 POP 操作将空间空出后才会继续写入。

视频模式下会在软件给的固定大小的存储器上不停地复写，也就是将内存当作 ring buffer 的概念，无需软件进行 POP 操作，使用上要确保图像被实时取出或是跟着 MJPEG 模块做连动。

19.3.4 RGB888 转 RGB565/RGBA8888 输出

对于输入格式为 RGB888 的图像数据，可以选择转为 RGB565 或 RGBA8888 写入到内存中。如果转为 RGB565 格式，则 R、G、B 的排列顺序可以通过寄存器 MISC 的位 FORMAT_565 进行控制，不同的值对应的排列顺序如下所示：

- 0: byte2[7:3], byte1[7:2], byte0[7:3]
- 1: byte1[7:3], byte2[7:2], byte0[7:3]
- 2: byte2[7:3], byte0[7:2], byte1[7:3]
- 3: byte0[7:3], byte2[7:2], byte1[7:3]
- 4: byte1[7:3], byte0[7:2], byte2[7:3]
- 5: byte0[7:3], byte1[7:2], byte2[7:3]

如果转为 RGBA8888 格式，则 A 的值与寄存器 MISC 的位 ALPHA 中填入的值一致。

19.3.5 图像矩形裁剪

通过寄存器 HSYNC_CONTROL 和 VSYNC_CONTROL 的高 16 位和低 16 位分别设置行同步信号和帧同步信号裁剪的起始和结束位置，就可以将指定位置和大小矩形窗口内的图像裁剪下来，超出矩形部分的数据会被丢弃。其中行同步信号起始和结束设置的是像素序号，帧同步信号起始和结束设置的是行号，裁剪后的图像包含起始点而不包含结束点。

19.3.6 帧取舍功能

可以通过寄存器 FRAME_PERIOD 设置一个帧周期 n ， n 的取值范围是 0~31，对应的实际值是 $n+1$ ，然后通过寄存器 FRAME_VLD 设置在一个帧周期中保留哪几帧图像。需要保留的在对应的 bit 位置写 1，需要舍弃的在对应的 bit 位置写 0。比如 n 设为 5，FRAME_VLD 的值设为 0x13，则每 6 帧图像中，第 1、2、5 帧图像会写入内存中，而第 3、4、6 帧图像会被舍弃，以 6 帧为周期进行循环。

19.3.7 行帧同步信号完整性检测

通过寄存器 `FRAME_SIZE_CONTROL` 的低 16 位和高 16 位可以分别设置行同步信号比较值和帧同步信号比较值，可以对信号的完整性进行检测。其中行同步信号设置的是每行的总像素数，帧同步信号设置的是总行数。当一帧图像的行或帧同步信号计数值与比较值不相等时，会有对应的中断产生。

19.3.8 缓存图像信息

模块内部包含 4 组 FIFO 记录图像地址和图像 ID。每当此模块完整写入一帧到内存，便会将此帧图像的起始地址和图像 ID 纪录于此 FIFO 中，但要注意的是当发生内存剩余不足，或是 4 组 FIFO 满存的状况时，模块会自动丢掉接下来图像的讯息，在图像信息取出的部分，可通过 `pop` 操作将最旧的图像信息空出，此时 FIFO 会自动推进，保证 FIFO 内部图像信息的时序，如下图：

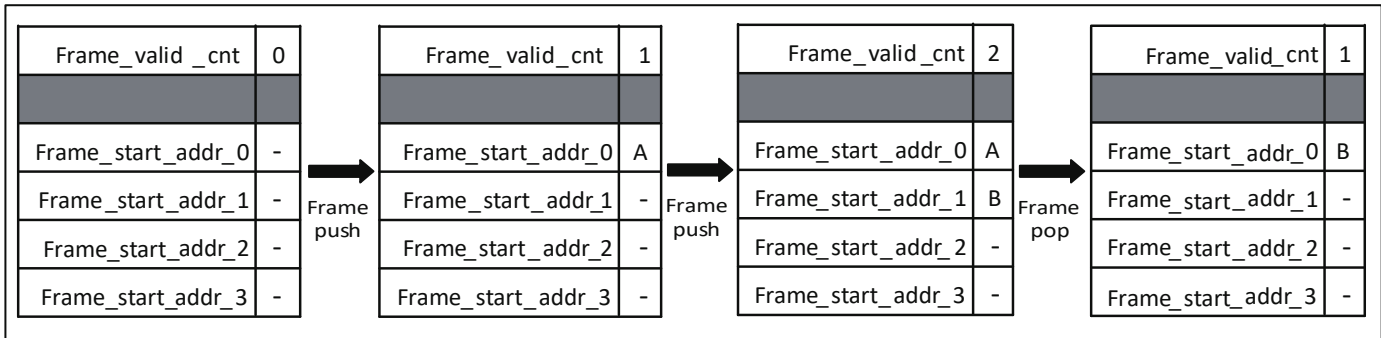


图 19.1: FIFO 框架

19.3.9 支持多种中断信息 (可独立开关配置)

- **Normal 中断** * 可设定一个计数值 `n`，每当写入 `n` 张图像后就会触发一次中断
- **Memory 中断** * 当内存剩余空间不足一帧大小，已经使用的内存被复写时，触发中断，如下图所示：

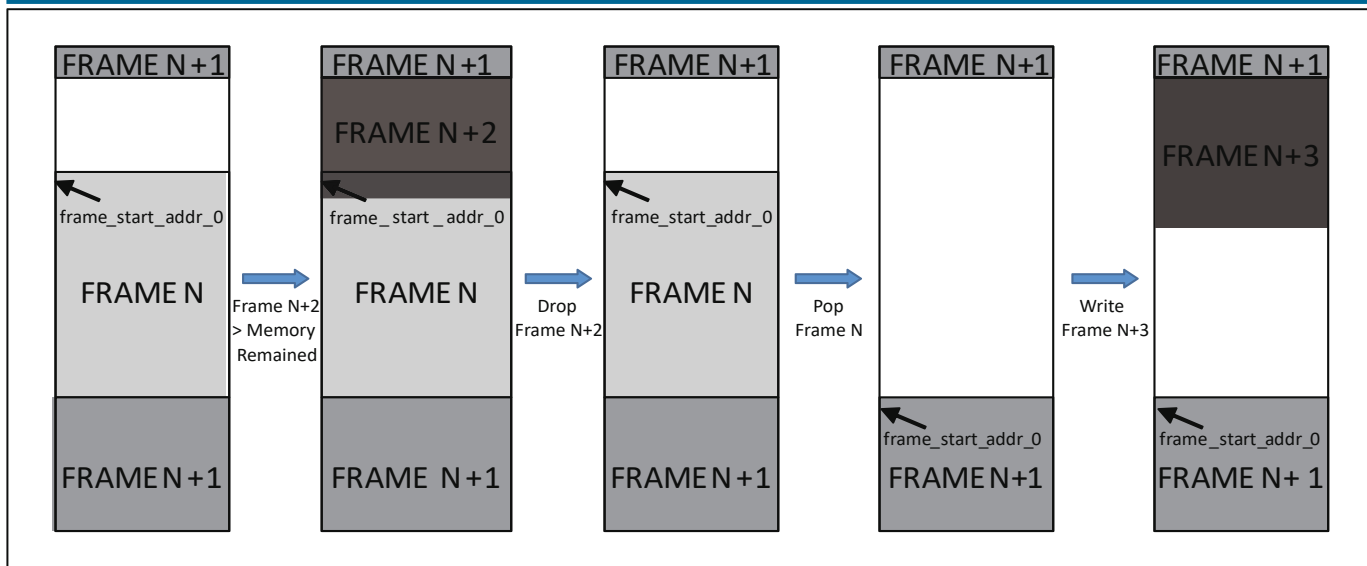


图 19.2: 内存

- **Frame 中断** * 当未处理的图像超过 4 组，无法再存储更多的图像信息时，触发中断
- **FIFO 中断** * 当总线来不及写入内存，导致 FIFO 溢出时，触发中断
- **Hsync 中断** * 当一帧图像中某行的像素点数与设置值不相等（行同步信号完整性检测不通过）时，触发中断
- **Vsync 中断** * 当一帧图像的总行数与设置值不相等（帧同步信号完整性检测不通过）时，发出中断

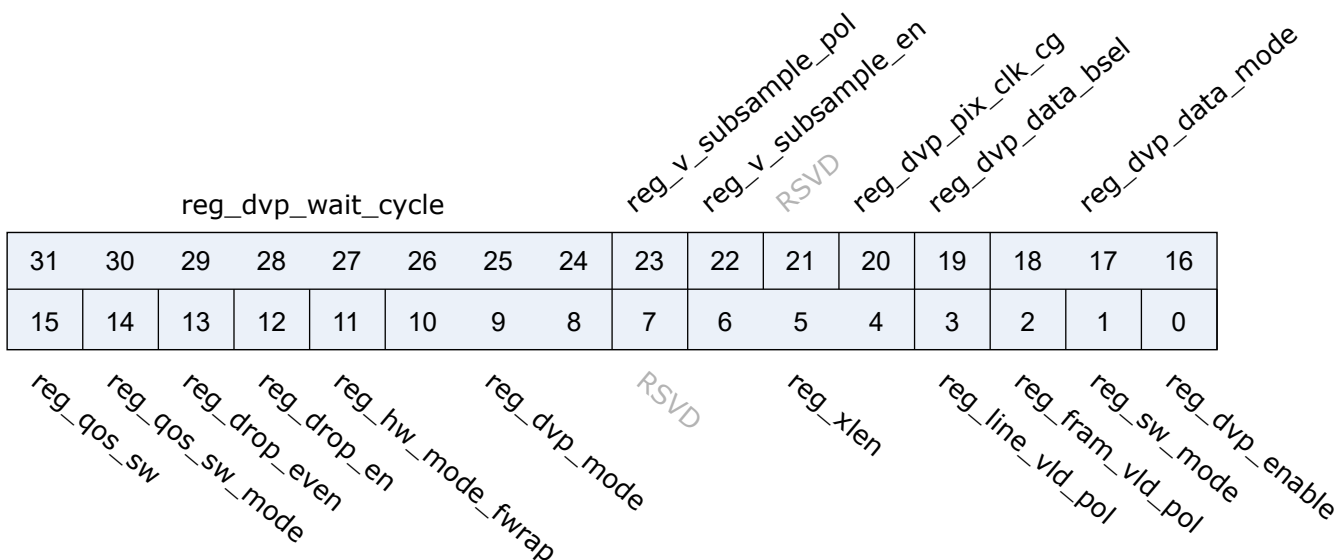
19.4 寄存器描述

名称	描述
cam_dvp2axi_configure	Camera configure
cam_dvp2axi_addr_start	Start of write address
cam_dvp2axi_mem_bcmt	Burst count of memory
cam_dvp_status_and_error	Interrupt and bus status
cam_dvp2axi_frame_bcmt	Byte count of frame
cam_dvp_frame_fifo_pop	Interrupt clear and pop
cam_dvp2axi_frame_vld	Frame valid in period
cam_dvp2axi_frame_period	Frame period
cam_dvp2axi_misc	RGB565 and RGBA8888 setting
cam_dvp2axi_hsync_crop	Hsync crop
cam_dvp2axi_vsync_crop	Vsync crop

名称	描述
cam_dvp2axi_fram_exm	Total valid count
cam_frame_start_addr0	Start address of frame0
cam_frame_start_addr1	Start address of frame1
cam_frame_start_addr2	Start address of frame2
cam_frame_start_addr3	Start address of frame3
cam_frame_id_sts01	ID of frame 0/1
cam_frame_id_sts23	ID of frame 2/3
cam_dvp_debug	ID latch timing

19.4.1 cam_dvp2axi_configue

地址：0x20057000



位	名称	权限	复位值	描述
31:24	reg_dvp_wait_cycle	r/w	8'h40	Cycles in FSM Wait mode
23	reg_v_subsample_pol	r/w	1'b0	DVP2BUS vertical sub-sampling polarity 1'b0: Odd lines are masked 1'b1: Even lines are masked
22	reg_v_subsample_en	r/w	1'b0	DVP2BUS vertical sub-sampling enable
21	RSVD			

位	名称	权限	复位值	描述
20	reg_dvp_pix_clk_cg	r/w	1'b0	DVP pix clk gate
19	reg_dvp_data_bsel	r/w	1'b0	Byte select signal for DVP 8-bit mode, don't care if reg_dvp_data_8bit is disabled 1'b0: Select the lower byte of pix_data 1'b1: Select the upper byte of pix_data
18:16	reg_dvp_data_mode	r/w	3'b0	DVP 8-bit mode enable 3'd0: DVP pix_data is 16-bit wide 3'd1: DVP pix_data is 24-bit mode 3'd2: DVP pix_data is 24-comp-16-bit mode 3'd3: DVP pix_data is 24-exp-32-bit mode 3'd4: DVP pix_data is 8-bit wide Others - Reserved
15	reg_qos_sw	r/w	1'b0	AXI Qos software mode value
14	reg_qos_sw_mode	r/w	1'b0	AXI QoS software mode enable
13	reg_drop_even	r/w	1'b0	Only effect when reg_drop_en=1 : 1'b1 : Drop all even bytes 1'b0 : Drop all odd bytes
12	reg_drop_en	r/w	1'b0	Drop mode Enable
11	reg_hw_mode_fwrap	r/w	1'b1	DVP2BUS HW mode with frame start address wrap to reg_addr_start
10:8	reg_dvp_mode	r/w	3'd0	Image sensor mode selection: 3'd0-Vsync&Hsync 3'd1-Vsync Hsync 3'd2-Vsync 3'd3-Hsync
7	RSVD			
6:4	reg_xlen	r/w	3'd3	burst length setting 3'd0 - Single / 3'd1 - INCR4 / 3'd2 - INCR8 3'd3 - INCR16 / 3'd5 - INCR32 / 3'd6 - INCR64
3	reg_line_vld_pol	r/w	1'b1	Image sensor line valid polarity, 1'b0 - Active low, 1'b1 - Active high
2	reg_fram_vld_pol	r/w	1'b1	Image sensor frame valid polarity, 1'b0 - Active low, 1'b1 - Active high
1	reg_sw_mode	r/w	1'b0	DVP2BUS SW manual mode (don't care if reg_swap_mode is enabled)
0	reg_dvp_enable	r/w	1'b0	module enable

19.4.2 cam_dvp2axi_addr_start

地址: 0x20057004

reg_addr_start

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_addr_start

位	名称	权限	复位值	描述
31:0	reg_addr_start	r/w	32'h80000000	AXI start address

19.4.3 cam_dvp2axi_mem_bcnc

地址: 0x20057008

reg_mem_burst_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

reg_mem_burst_cnt

位	名称	权限	复位值	描述
31:0	reg_mem_burst_cnt	r/w	32'hC000	AXI burst cnt before wrap to "reg_addr_start"

19.4.4 cam_dvp_status_and_error

地址: 0x2005700c

RSVD	RSVD	st_dvp_idle	axi_idle	st_bus_fish	st_bus_wait	st_bus_func	st_bus_idle	RSVD	sts_vcnc_int	sts_hcnt_int	frame_valid_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
sts_fifo_int	sts_frame_int	sts_mem_int	sts_normal_int	reg_int_fifo_en	reg_int_frame_en	reg_int_mem_en	reg_int_normal_en	reg_int_vcnc_en	reg_int_hcnt_en	RSVD	reg_frame_cnt_trgr_int				

位	名称	权限	复位值	描述
31:30	RSVD			
29	st_dvp_idle	r	1'b1	DVP2BUS asynchronous fifo idle status
28	axi_idle	r	1'b1	DVP2BUS AHB idle status
27	st_bus_fish	r	1'b0	DVP in flush state
26	st_bus_wait	r	1'b0	DVP in wait state
25	st_bus_func	r	1'b0	DVP in functional state
24	st_bus_idle	r	1'b1	DVP in idle state
23	RSVD			
22	sts_vcmt_int	r	1'b0	Vsync valid line count non-match interrupt status
21	sts_hcnt_int	r	1'b0	Hsync valid pixel count non-match interrupt status
20:16	frame_valid_cnt	r	5'd0	Frame counts in memory before read out in SW mode
15	sts_fifo_int	r	1'b0	FIFO OverWrite interrupt status
14	sts_frame_int	r	1'b0	Frame OverWrite interrupt status
13	sts_mem_int	r	1'b0	Memory OverWrite interrupt status
12	sts_normal_int	r	1'b0	Normal Write interrupt status
11	reg_int_fifo_en	r/w	1'b1	FIFO OverWrite interrupt enable
10	reg_int_frame_en	r/w	1'b0	Frame OverWrite interrupt enable
9	reg_int_mem_en	r/w	1'b0	Memory OverWrite interrupt enable
8	reg_int_normal_en	r/w	1'b0	Normal Write interrupt enable
7	reg_int_vcmt_en	r/w	1'b0	Vsync valid line count match interrupt enable
6	reg_int_hcnt_en	r/w	1'b0	Hsync valid pixel count match interrupt enable
5	RSVD			
4:0	reg_frame_cnt_trgr_int	r/w	5'd0	Frame to issue interrupt at SW Mode

19.4.5 cam_dvp2axi_frame_bcmt

地址: 0x20057010

reg_frame_byte_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

reg_frame_byte_cnt

位	名称	权限	复位值	描述
31:0	reg_frame_byte_cnt	r/w	32'h7e90	Single Frame byte cnt(Need pre-calculation)

19.4.6 cam_dvp_frame_fifo_pop

地址: 0x20057014

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	reg_int_vcmt_clr	reg_int_hcnt_clr	reg_int_fifo_clr	reg_int_frame_clr	reg_int_mem_clr	reg_int_normal_clr	RSVD	RSVD	RSVD	rfifo_pop

位	名称	权限	复位值	描述
31:10	RSVD			
9	reg_int_vcmt_clr	w1p	1'd0	Interrupt clear
8	reg_int_hcnt_clr	w1p	1'd0	Interrupt clear
7	reg_int_fifo_clr	w1p	1'd0	Interrupt clear
6	reg_int_frame_clr	w1p	1'd0	Interrupt clear
5	reg_int_mem_clr	w1p	1'd0	Interrupt clear
4	reg_int_normal_clr	w1p	1'd0	Interrupt clear
3:1	RSVD			
0	rfifo_pop	w1p	1'b0	Write this bit will trigger fifo pop

19.4.7 cam_dvp2axi_frame_vld

地址: 0x20057018

reg_frame_n_vld

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_frame_n_vld

位	名称	权限	复位值	描述
31:0	reg_frame_n_vld	r/w	32'hfff_ffff	Bitwise frame valid in period

19.4.8 cam_dvp2axi_frame_period

地址: 0x2005701c

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_frame_period

位	名称	权限	复位值	描述
31:5	RSVD			
4:0	reg_frame_period	r/w	5'h0	Frame period cnt. (EX. Set this register 0, the period is 1)

19.4.9 cam_dvp2axi_misc

地址: 0x20057020

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_alpha

reg_format_565

位	名称	权限	复位值	描述
31:11	RSVD			

位	名称	权限	复位值	描述
10:8	reg_format_565	r/w	3'd0	Only work when reg_dvp_data_mode=2 (24-comp-16-bit mode) 3'd0: B2(5)B1(6)B0(5) 3'd1: B1(5)B2(6)B0(5) 3'd2: B2(5)B0(6)B1(5) 3'd3: B0(5)B2(6)B1(5) 3'd4: B1(5)B0(6)B2(5) 3'd5: B0(5)B1(6)B2(5)
7:0	reg_alpha	r/w	8'h0	Only work when "reg_dvp_data_mode==2'd3(DVP pix_data is 24-exp-32-bit mode)" The value of [31:24]

19.4.10 cam_dvp2axi_hsync_crop

地址: 0x20057030

reg_hsync_act_start

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_hsync_act_end

位	名称	权限	复位值	描述
31:16	reg_hsync_act_start	r/w	16'h0	Valid hsync start cnt
15:0	reg_hsync_act_end	r/w	16'hFFFF	Valid hsync end cnt

19.4.11 cam_dvp2axi_vsync_crop

地址: 0x20057034

reg_vsync_act_start

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_vsync_act_end

位	名称	权限	复位值	描述
31:16	reg_vsync_act_start	r/w	16'h0	Valid vsync start cnt
15:0	reg_vsync_act_end	r/w	16'hFFFF	Valid vsync end cnt

19.4.12 cam_dvp2axi_fram_exm

地址: 0x20057038

reg_total_vcmt

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_total_hcnt

位	名称	权限	复位值	描述
31:16	reg_total_vcmt	r/w	16'h0	Total valid line count in a frame
15:0	reg_total_hcnt	r/w	16'h0	Total valid pix count in a line

19.4.13 cam_frame_start_addr0

地址: 0x20057040

frame_start_addr_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

frame_start_addr_0

位	名称	权限	复位值	描述
31:0	frame_start_addr_0	r	32'd0	DVP2BUS PIC 0 Start address

19.4.14 cam_frame_start_addr1

地址: 0x20057048

frame_start_addr_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

frame_start_addr_1

位	名称	权限	复位值	描述
31:0	frame_start_addr_1	r	32'd0	DVP2BUS PIC 1 Start address

19.4.15 cam_frame_start_addr2

地址: 0x20057050

frame_start_addr_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

frame_start_addr_2

位	名称	权限	复位值	描述
31:0	frame_start_addr_2	r	32'd0	DVP2BUS PIC 2 Start address

19.4.16 cam_frame_start_addr3

地址: 0x20057058

frame_start_addr_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

frame_start_addr_3

位	名称	权限	复位值	描述
31:0	frame_start_addr_3	r	32'd0	DVP2BUS PIC 3 Start address

19.4.17 cam_frame_id_sts01

地址: 0x20057060

frame_id_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

frame_id_0

位	名称	权限	复位值	描述
31:16	frame_id_1	r	16'd0	DVP2BUS PIC 1 ID
15:0	frame_id_0	r	16'd0	DVP2BUS PIC 0 ID

19.4.18 cam_frame_id_sts23

地址: 0x20057064

frame_id_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

frame_id_2

位	名称	权限	复位值	描述
31:16	frame_id_3	r	16'd0	DVP2BUS PIC 3 ID
15:0	frame_id_2	r	16'd0	DVP2BUS PIC 2 ID

19.4.19 cam_dvp_debug

地址: 0x200570f0

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_id_latch_line

位	名称	权限	复位值	描述
31:12	RSVD			
11:8	reg_id_latch_line	r/w	4'd5	ID latch timing (line count)
7:32	RSVD			
31:0	RESERVED	rsvd	32'hf0f0f0f0	RESERVED

20.1 简介

MJPEG(Motion Joint Photographic Experts Group) 是一种视频编码格式，可精确到帧编辑和多层图像处理，把运动的视频序列作为连续的静止图像来处理，这种压缩方式单独完整地压缩每一帧。通过对 YCbCr 格式的原始数据进行压缩，可以大幅降低一帧图像所占用的内存空间。

20.2 主要特征

- 可配置的输入格式，包括：
- YCbCr4:2:2 平面或打包模式
- YCbCr4:2:0 平面模式
- YCbCr4:0:0
- 可配置的输入数据 Y、Cb、Cr 排列顺序
- 量化系数表可自由配置
- 支持软件模式和联动模式
- 支持 swap 模式
- 支持 kick 模式
- 可预留 jpg 头部空间和自动添加 jpg 尾
- 连续缓存多达 4 组图片信息
- 多种应用中断，有利于弹性使用与出错提示

20.3 功能描述

20.3.1 输入配置

通过寄存器 MJPEG_CONTROL_1 的位 <REG_YUV_MODE> 可以选择输入数据的 YCbCr 格式，包括 YCbCr4:2:2 平面或打包模式，YCbCr4:2:0 平面模式和 YCbCr4:0:0 灰度图。当选择 YCbCr4:2:2 打包模式时，通过寄存器 MJPEG_HEADER_BYTE 的高 8 位可以配置 Y、Cb、Cr 的排列顺序。在选择其他模式时，可以通过寄存器 MJPEG_CONTROL_1 的位 <REG_ORDER_U_EVEN> 设定 Cb 和 Cr 的顺序。详细的配置如下图所示。

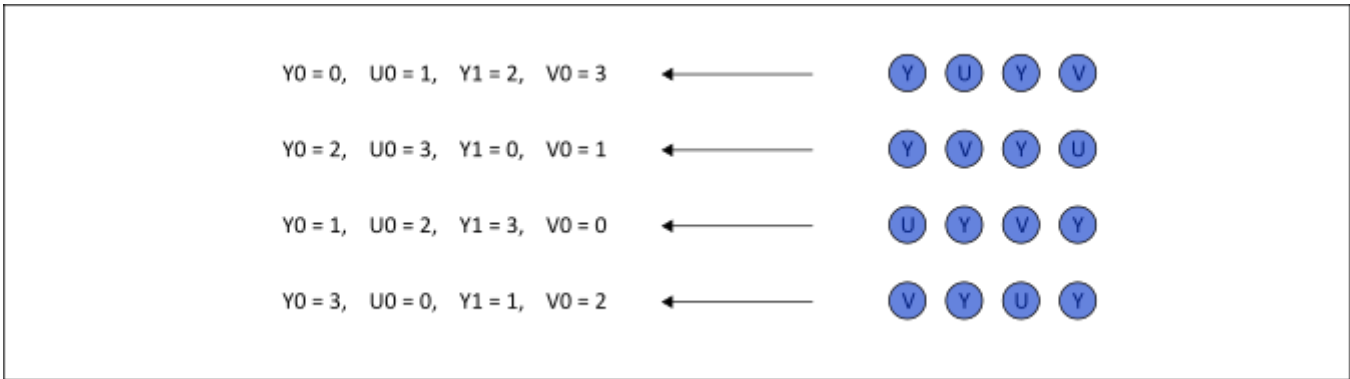


图 20.1: MJPEG YUYV interleave order configuration

20.3.2 量化系数表

量化系数表可由用户自由配置，<reg_q_0_00> 到 <reg_q_0_3F> 表示灰度信息 Y 分量的量化表，<reg_q_1_00> 到 <reg_q_1_3F> 表示色度信息 Cb 和 Cr 的量化表。量化表按照从左上角向下排列的顺序，第一列结束后紧跟第二列，即 reg_q_0_00 代表 Y 分量第一行第一列的量化数值，reg_q_0_01 代表 Y 分量第二行第一列的量化数值，reg_q_0_07 代表 Y 分量第八行第一列的量化数值，reg_q_0_08 代表 Y 分量第一行第二列的量化数值，依次类推。

20.3.3 软件模式和联动模式

将 MJPEG_CONTROL_2 寄存器的位 <REG_MJPEG_SW_MODE> 置 1 时即使能软件模式，在此模式下 MJPEG 会从内存中读取指定帧数的原始图片数据进行压缩，此模式下需要预先将图片数据准备好。

将 MJPEG_CONTROL_2 寄存器的位 <REG_MJPEG_SW_MODE> 清 0 时即使能联动模式，在此模式下 MJPEG 会将 CAM 模块的输出作为其输入进行处理，MJPEG 是以 8*8 的数据块为一个单元进行处理的，当 CAM 向内存中写完 8 行数据后，MJPEG 就会开始工作，MJPEG 处理完的内存空间会释放给 CAM 重新使用，这样 CAM 无需一整张图片的内存空间即可完成与 MJPEG 的联动。

20.3.4 swap 模式

使用 swap 模式时, MJPEG 存储空间会被平均分成两块, 当 MJPEG 写完其中一块时会产生一个中断通知软件将数据读走, 而它会将数据写入另一块中。来回交替使用, 从而用不足一帧图片的存储空间进行数据处理。

20.3.5 kick 模式

将 MJPEG_CONTROL_2 寄存器的位 <REG_SW_KICK_MODE> 置 1 时即使能 kick 模式, 在此模式下每次向 MJPEG_CONTROL_2 寄存器的位 <REG_SW_KICK> 写 1 即可启动一次压缩, 压缩行数由 MJPEG_YUV_MEM_SW 寄存器的位 <REG_SW_KICK_HBLK> 确定。需要注意的是 kick 模式只能在软件模式下使用, 不能在联动模式下使用。

20.3.6 jpg 功能

jpg 功能会自动在每帧数据的开头留出一定字节数的空间并填充指定数据, 该空间的大小由寄存器 MJPEG_HEADER_BYTE 的低 12 位进行设置。在偏移 0x800 处有 768 字节的内存用于存放需要填充在每帧开头的数据, 用户只需将 jpg 头部信息写入其中, 则生成的每帧数据开头都会包含这段信息。另外如果使能自动填充 jpg 尾的话还会在每帧数据的结尾补上两个字节数据, 这两个字节的值为 0xFFD9。

20.3.7 缓存图片信息

模块内部包含 4 组 FIFO 记录图片地址、图片大小和图片 ID。每当此模块完整写入一帧到内存, 便会将此帧图片的起始地址、图片大小和图片 ID 纪录于此 FIFO 中, 但要注意的是当发生内存剩余不足, 或是 4 组 FIFO 满存的状况时, 模块会自动丢掉接下来图片的讯息, 在图片信息取出的部分, 可通过 APB 接口做 pop 的动作, 将最旧的图片信息空出, 此时 FIFO 会自动推进, 保证 FIFO 内部图片信息的时序。

20.3.8 支持多种中断信息 (可独立开关配置)

- A、Normal 中断-可设定固定写入几张图片后发出中断
- B、Camera 中断-MJPEG 处理的速度跟不上 CAM 模块写入的速度导致 CAM 溢出时, 发出中断
- C、Memory 中断-当内存被复写时, 发出中断
- D、Frame 中断-当未处理图片超过 4 组时, 发出中断
- E、Swap 中断 - 当 swap 模式下一个内存块被写满时, 发出中断

20.4 寄存器描述

名称	描述
mjpeg_control_1	MJPEG configure
mjpeg_control_2	MJPEG software mode
mjpeg_yy_frame_addr	Memory start address of Y

名称	描述
mjpeg_uv_frame_addr	Memory start address of UV
mjpeg_yuv_mem	Number of Y/UV block line
jpeg_frame_addr	Memory start address of JPEG
jpeg_store_memory	Burst count of memory
mjpeg_control_3	Interrupt and bus status
mjpeg_frame_fifo_pop	Interrupt clear and pop
mjpeg_frame_size	Total block count
mjpeg_header_byte	YUV order and auto fill
mjpeg_swap_mode	Swap mode
mjpeg_swap_bit_cnt	Remain bit count
mjpeg_yuv_mem_sw	Number of kick block line
mjpeg_Y_frame_read_status_1	Y read status1
mjpeg_Y_frame_read_status_2	Y read status2
mjpeg_Y_frame_write_status	Y write status
mjpeg_UV_frame_read_status_1	UV read status1
mjpeg_UV_frame_read_status_2	UV read status2
mjpeg_UV_frame_write_status	UV write status
mjpeg_frame_w_hblk_status	Vertical block write status
mjpeg_start_addr0	Start address of frame0
mjpeg_bit_cnt0	Bit count of frame0
mjpeg_start_addr1	Start address of frame1
mjpeg_bit_cnt1	Bit count of frame1
mjpeg_start_addr2	Start address of frame2
mjpeg_bit_cnt2	Bit count of frame2
mjpeg_start_addr3	Start address of frame3
mjpeg_bit_cnt3	Bit count of frame3
mjpeg_q_enc	Quantize SRAM setting

名称	描述
mjpeg_frame_id_10	ID of frame 0/1
mjpeg_frame_id_32	ID of frame 2/3
mjpeg_debug	ID latch timing

20.4.1 mjpeg_control_1

地址：0x30021000

reg_mjpeg_hw_frame															
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_yuv_mode reg_w_xlen reg_read_fwrap reg_reflect_dmy reg_last_hf_hblk_dmy reg_hw_mode_swen reg_order_u_even reg_mjpeg_bit_order reg_mjpeg_enable															

位	名称	权限	复位值	描述
31:30	RSVD			
29:24	reg_mjpeg_hw_frame	r/w	6'd0	Non-SW mode frame cnt 0 - Non auto stop
23:14	RSVD			
13:12	reg_yuv_mode	r/w	2'd0	YUV format setting 2'd0 : YUV420 Planar 2'd1 : YUV400 grey scale 2'd2 : YUV422 Planar 2'd3 : INTV422
11	RSVD			
10:8	reg_w_xlen	r/w	3'd3	AXI burst length setting 3'd0: Single 3'd1: INCR4 3'd2: INCR8 3'd3: INCR16 Others: RSVD

位	名称	权限	复位值	描述
7	reg_read_fwrap	r/w	1'b1	Only effect in non SW mode This bit determine whether YUV frame read start @ start address
6	reg_reflect_dmy	r/w	1'b0	UV dummy with relect
5	reg_last_hf_hblk_dmy	r/w	1'b0	MJPEG last half vertical block with dummy data 8'h80
4	reg_last_hf_wblk_dmy	r/w	1'b0	MJPEG last half horizational block with dummy data 8'h80
3	reg_hw_mode_swen	r/w	1'b0	Only word In hardware mode. SW enable immediately to skip waiting first frame done
2	reg_order_u_even	r/w	1'b1	1'b0: U is odd byte of UV frame / V is even byte of UV frame 1'b1: U is even byte of UV frame / V is odd byte of UV frame
1	reg_mjpeg_bit_order	r/w	1'b1	MJPEG bitstream byte order adjustment
0	reg_mjpeg_enable	r/w	1'b0	MJPEG enable

20.4.2 mjpeg_control_2

地址: 0x30021004

reg_mjpeg_wait_cycle

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_uv_dvp2axi_sel

reg_yy_dvp2axi_sel

reg_mjpeg_sw_run

reg_mjpeg_sw_mode

reg_sw_kick_mode

reg_sw_frame

RSVD

位	名称	权限	复位值	描述
31:16	reg_mjpeg_wait_cycle	r/w	16'h100	Cycle count in wait state
15:13	reg_uv_dvp2axi_sel	r/w	3'd1	DVP2AXI selection for UV frame(Don't care if using SW mode)
12:10	reg_yy_dvp2axi_sel	r/w	3'd0	DVP2AXI selection for Y frame(Don't care if using SW mode)
9	reg_mjpeg_sw_run	r/w	1'b0	MJPEG SW mode run (should enable reg_mjpeg_sw_mode before reg_mjpeg_sw_run)

位	名称	权限	复位值	描述
8	reg_mjpeg_sw_mode	r/w	1'b0	MJPEG SW mode enable (should enable reg_mjpeg_sw_mode before reg_mjpeg_sw_run)
7	reg_sw_kick_mode	r/w	1'b0	MJPEG SW mode hblk kick mode 1'b0: SW frame mode 1'b1: SW kick mode
6	reg_sw_kick	w1p	1'b0	MJPEG SW mode hblk kick
5	RSVD			
4:0	reg_sw_frame	r/w	5'd0	MJPEG SW mode frame count

20.4.3 mjpeg_yy_frame_addr

地址: 0x30021008

reg_yy_addr_start

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_yy_addr_start

位	名称	权限	复位值	描述
31:0	reg_yy_addr_start	r/w	32'h80000000	Memory start address of Y frame

20.4.4 mjpeg_uv_frame_addr

地址: 0x3002100c

reg_uv_addr_start

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_uv_addr_start

位	名称	权限	复位值	描述
31:0	reg_uv_addr_start	r/w	32'h80000000	Memory start address of UV frame

20.4.5 mjpeg_yuv_mem

地址: 0x30021010

reg_uv_mem_hblk															
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_yy_mem_hblk															

位	名称	权限	复位值	描述
31:29	RSVD			
28:16	reg_uv_mem_hblk	r/w	13'h2	Memory to store block line for UV frame
15:13	RSVD			
12:0	reg_yy_mem_hblk	r/w	13'h2	Memory to store block line for Y frame

20.4.6 jpeg_frame_addr

地址: 0x30021014

reg_w_addr_start															
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_w_addr_start															

位	名称	权限	复位值	描述
31:0	reg_w_addr_start	r/w	32'h80400000	Memory start address of JPEG frame

20.4.7 jpeg_store_memory

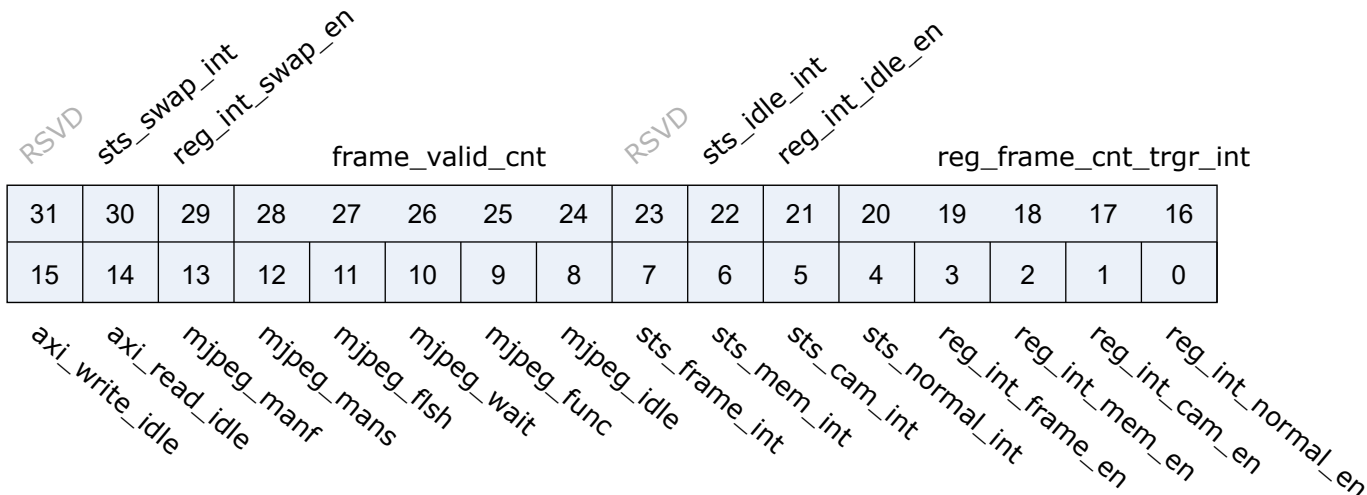
地址: 0x30021018

reg_w_burst_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
reg_w_burst_cnt															

位	名称	权限	复位值	描述
31:0	reg_w_burst_cnt	r/w	32'h4000	Memory to store jpeg image burst count

20.4.8 mjpeg_control_3

地址: 0x3002101c



位	名称	权限	复位值	描述
31	RSVD			
30	sts_swap_int	r	1'b0	Swap memory block interrupt status
29	reg_int_swap_en	r/w	1'b0	Swap memory block interrupt enable
28:24	frame_valid_cnt	r	5'd0	Frame count valid in status
23	RSVD			
22	sts_idle_int	r	1'b0	Normal Write interrupt status
21	reg_int_idle_en	r/w	1'b0	Back idle interrupt enable
20:16	reg_frame_cnt_trgr_int	r/w	5'd0	Frame threshold to issue interrupt
15	axi_write_idle	r	1'b0	AXI write bus idle state
14	axi_read_idle	r	1'b0	AXI read bus idle state
13	mjpeg_manf	r	1'b0	MJPEG in SW run state
12	mjpeg_mans	r	1'b0	MJPEG in SW set state
11	mjpeg_flush	r	1'b0	MJPEG in flush state
10	mjpeg_wait	r	1'b0	MJPEG in wait state
9	mjpeg_func	r	1'b0	MJPEG in HW function state

位	名称	权限	复位值	描述
8	mjpeg_idle	r	1'b1	MJPEG in idle state
7	sts_frame_int	r	1'b0	Frame OverWrite interrupt status
6	sts_mem_int	r	1'b0	Memory OverWrite interrupt status
5	sts_cam_int	r	1'b0	CAM OverWrite interrupt status (fatal)
4	sts_normal_int	r	1'b0	Normal Write interrupt status
3	reg_int_frame_en	r/w	1'b0	Frame OverWrite interrupt enable
2	reg_int_mem_en	r/w	1'b0	JPEG frame memory OverWrite interrupt enable
1	reg_int_cam_en	r/w	1'b1	YUV frame memory OverWrite interrupt enable (fatal)
0	reg_int_normal_en	r/w	1'b1	Normal Write interrupt enable

20.4.9 mjpeg_frame_fifo_pop

地址：0x30021020

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	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
RSVD	RSVD	reg_int_swap_clr	reg_int_idle_clr	reg_int_frame_clr	reg_int_mem_clr	reg_int_cam_clr	reg_int_normal_clr	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	reg_w_swap_clr	rfifo_pop

位	名称	权限	复位值	描述
31:14	RSVD			
13	reg_int_swap_clr	w1p	1'b0	Write this bit with 1'b1 to trigger swap interrupt clear
12	reg_int_idle_clr	w1p	1'b0	Write this bit with 1'b1 to trigger back idle interrupt clear
11	reg_int_frame_clr	w1p	1'b0	Write this bit with 1'b1 to trigger frame overwrite interrupt clear
10	reg_int_mem_clr	w1p	1'b0	Write this bit with 1'b1 to trigger memory overwrite interrupt clear
9	reg_int_cam_clr	w1p	1'b0	Write this bit with 1'b1 to trigger YUV overwrite interrupt clear
8	reg_int_normal_clr	w1p	1'b0	Write this bit with 1'b1 to trigger normal interrupt clear

位	名称	权限	复位值	描述
7:2	RSVD			
1	reg_w_swap_clr	w1p	1'b0	Free current read memory block
0	rfifo_pop	w1p	1'b0	Write this bit with 1'b1 to trigger JPEG frame fifo pop

20.4.10 mjpeg_frame_size

地址: 0x30021024

reg_frame_hblk															
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_frame_wblk															
----------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

位	名称	权限	复位值	描述
31:28	RSVD			
27:16	reg_frame_hblk	r/w	12'd20	Frame total vertical block count
15:12	RSVD			
11:0	reg_frame_wblk	r/w	12'd15	Frame total horizontal block count

20.4.11 mjpeg_header_byte

地址: 0x30021028

reg_head_byte															
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	reg_v0_order	r/w	2'd3	V order in Interleave mode
29:28	reg_y1_order	r/w	2'd2	Y1 order in Interleave mode
27:26	reg_u0_order	r/w	2'd1	U order in Interleave mode
25:24	reg_y0_order	r/w	2'd0	Y0 order in Interleave mode
23:17	RSVD			
16	reg_tail_exp	r/w	1'b0	Auto fill tail 0xFF and 0xD9
15:12	RSVD			
11:0	reg_head_byte	r/w	12'h0	Preserve head memory space for each frame

20.4.12 mjpeg_swap_mode

地址: 0x30021030

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	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
RSVD	RSVD	RSVD	sts_swap_fend	sts_swap_fstart	sts_read_swap_idx	sts_swap1_full	sts_swap0_full	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	reg_w_swap_mode

位	名称	权限	复位值	描述
31:13	RSVD			
12	sts_swap_fend	r	1'b0	Current read memory block is frame end
11	sts_swap_fstart	r	1'b0	Current read memory block is frame start
10	sts_read_swap_idx	r	1'b0	Current read memory block index 0: memory block0 1: memory block1
9	sts_swap1_full	r	1'b0	Memory block1 is full
8	sts_swap0_full	r	1'b0	Memory block0 is full
7:1	RSVD			
0	reg_w_swap_mode	r/w	1'b0	Enable write swap mode

20.4.13 mjpeg_swap_bit_cnt

地址: 0x30021034

frame_swap_end_bit_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

frame_swap_end_bit_cnt

位	名称	权限	复位值	描述
31:0	frame_swap_end_bit_cnt	r	32'd0	In swap mode frame remain bit count in last block Only valid when "sts_swap_fend==1"

20.4.14 mjpeg_yuv_mem_sw

地址: 0x30021038

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_sw_kick_hblk

位	名称	权限	复位值	描述
31:13	RSVD			
12:0	reg_sw_kick_hblk	r/w	13'h2	Memory to store block line for frame on SW kick

20.4.15 mjpeg_Y_frame_read_status_1

地址: 0x30021040

yy_frm_hblk_r

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

yy_mem_hblk_r

位	名称	权限	复位值	描述
31:29	RSVD			
28:16	yy_frm_hblk_r	r	13'd0	Y frame veritical block read status
15:13	RSVD			
12:0	yy_mem_hblk_r	r	13'd0	Y memory veritical block read status

20.4.16 mjpeg_Y_frame_read_status_2

地址：0x30021044

yy_frm_cnt_r

yy_mem_rnd_r

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

yy_wblk_r

位	名称	权限	复位值	描述
31:24	yy_frm_cnt_r	r	8'd0	Y frame read count
23:16	yy_mem_rnd_r	r	8'd0	Y memory read round
15:13	RSVD			
12:0	yy_wblk_r	r	13'd0	Y frame horizontal block read status

20.4.17 mjpeg_Y_frame_write_status

地址：0x30021048

yy_frm_cnt_w								yy_mem_rnd_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

yy_mem_hblk_w

位	名称	权限	复位值	描述
31:24	yy_frm_cnt_w	r	8'd0	Y frame write count
23:16	yy_mem_rnd_w	r	8'd0	Y memory write round
15:13	RSVD			
12:0	yy_mem_hblk_w	r	13'd0	Y memory veritical block write status

20.4.18 mjpeg_UV_frame_read_status_1

地址: 0x3002104c

RSVD RSVD RSVD

uv_frm_hblk_r															
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

uv_mem_hblk_r

位	名称	权限	复位值	描述
31:29	RSVD			
28:16	uv_frm_hblk_r	r	13'd0	UV frame veritical block read status
15:13	RSVD			
12:0	uv_mem_hblk_r	r	13'd0	UV memory veritical block read status

20.4.19 mjpeg_UV_frame_read_status_2

地址: 0x30021050

uv_frm_cnt_r								uv_mem_rnd_r							
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

uv_wblk_r

位	名称	权限	复位值	描述
31:24	uv_frm_cnt_r	r	8'd0	UV frame read count
23:16	uv_mem_rnd_r	r	8'd0	UV memory read round
15:13	RSVD			
12:0	uv_wblk_r	r	13'd0	UV frame horizontal block read status

20.4.20 mjpeg_UV_frame_write_status

地址: 0x30021054

uv_frm_cnt_w								uv_mem_rnd_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

uv_mem_hblk_w

位	名称	权限	复位值	描述
31:24	uv_frm_cnt_w	r	8'd0	UV frame write count
23:16	uv_mem_rnd_w	r	8'd0	UV memory write round
15:13	RSVD			
12:0	uv_mem_hblk_w	r	13'd0	UV memory vertical block write status

20.4.21 mjpeg_frame_w_hblk_status

地址: 0x30021058

uv_frm_hblk_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

yy_frm_hblk_w

位	名称	权限	复位值	描述
31:29	RSVD			
28:16	uv_frm_hblk_w	r	13'd0	UV frame vertical block write status
15:13	RSVD			
12:0	yy_frm_hblk_w	r	13'd0	YY frame vertical block write status

20.4.22 mjpeg_start_addr0

地址：0x30021080

frame_start_addr_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

frame_start_addr_0

位	名称	权限	复位值	描述
31:0	frame_start_addr_0	r	32'd0	MJPEG FRAME0 Start address

20.4.23 mjpeg_bit_cnt0

地址：0x30021084

frame_bit_cnt_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

frame_bit_cnt_0

位	名称	权限	复位值	描述
31:0	frame_bit_cnt_0	r	32'd0	MJPEG FRAME0 total bit count

20.4.24 mjpeg_start_addr1

地址: 0x30021088

frame_start_addr_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

frame_start_addr_1

位	名称	权限	复位值	描述
31:0	frame_start_addr_1	r	32'd0	MJPEG FRAME1 Start address

20.4.25 mjpeg_bit_cnt1

地址: 0x3002108c

frame_bit_cnt_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

frame_bit_cnt_1

位	名称	权限	复位值	描述
31:0	frame_bit_cnt_1	r	32'd0	MJPEG FRAME1 total bit count

20.4.26 mjpeg_start_addr2

地址: 0x30021090

frame_start_addr_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

frame_start_addr_2

位	名称	权限	复位值	描述
31:0	frame_start_addr_2	r	32'd0	MJPEG FRAME2 Start address

20.4.27 mjpeg_bit_cnt2

地址: 0x30021094

frame_bit_cnt_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

frame_bit_cnt_2

位	名称	权限	复位值	描述
31:0	frame_bit_cnt_2	r	32'd0	MJPEG FRAME2 total bit count

20.4.28 mjpeg_start_addr3

地址: 0x30021098

frame_start_addr_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

frame_start_addr_3

位	名称	权限	复位值	描述
31:0	frame_start_addr_3	r	32'd0	MJPEG FRAME3 Start address

20.4.29 mjpeg_bit_cnt3

地址: 0x3002109c

frame_bit_cnt_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

frame_bit_cnt_3

位	名称	权限	复位值	描述
31:0	frame_bit_cnt_3	r	32'd0	MJPEG FRAME3 total bit count

20.4.30 mjpeg_q_enc

地址: 0x30021100

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
31:26	RSVD			
25	sts_q_sram_enc	r	1'd0	Current Quantize SRAM selection for Encode
24	reg_q_sram_sw	w1p	1'd0	Quantize SRAM Switch
23:4	RSVD			
3	frame_q_sram_3	r	1'd0	MJPEG FRAME3 Quantize SRAM selection
2	frame_q_sram_2	r	1'd0	MJPEG FRAME2 Quantize SRAM selection
1	frame_q_sram_1	r	1'd0	MJPEG FRAME1 Quantize SRAM selection
0	frame_q_sram_0	r	1'd0	MJPEG FRAME0 Quantize SRAM selection

20.4.31 mjpeg_frame_id_10

地址: 0x30021110

frame_id_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
frame_id_0															

位	名称	权限	复位值	描述
31:16	frame_id_1	r	16'd0	JPEG PIC 1 ID
15:0	frame_id_0	r	16'd0	JPEG PIC 0 ID

20.4.32 mjpeg_frame_id_32

地址: 0x30021114

frame_id_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

frame_id_2

位	名称	权限	复位值	描述
31:16	frame_id_3	r	16'd0	JPEG PIC 3 ID
15:0	frame_id_2	r	16'd0	JPEG PIC 2 ID

20.4.33 mjpeg_debug

地址: 0x300211f0

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_id_latch_hblk reg_mjpeg_dbg_sel reg_mjpeg_dbg_en

位	名称	权限	复位值	描述
31:12	RSVD			
11:8	reg_id_latch_hblk	r/w	4'd1	ID latch timing (hblk count)
7:4	reg_mjpeg_dbg_sel	r/w	4'd0	MJPEG debug flag selection
3:1	RSVD			

位	名称	权限	复位值	描述
0	reg_mjpeg_dbg_en	r/w	1'b0	MJPEG debug flag enable

21.1 简介

DBI 是显示总线接口 (Display Bus Interface) 的简称, 也被称为 MCU 接口, 是 MIPI 联盟定义的用于与显示屏通信的接口协议。DBI 协议的数据线和控制线是复用的, 驱动的显示模块需要带有 GRAM。DBI 又可以细分为 MIPI DBI Type A、MIPI DBI Type B 和 MIPI DBI Type C 三种不同的模式, 不同模式下的硬件接口以及数据采样都有所不同。如 MIPI DBI Type A 是下降沿采样 (基于 Motorola 6800 总线), 而 MIPI DBI Type B 是上升沿采样 (基于 Intel 8080 总线)。该模块另外还集成了 QSPI(Quad Serial Peripheral Interface) 显示接口, QSPI 有四根数据线, 是对 SPI 接口的扩展, 极大提高了传输效率。

21.2 主要特征

- 符合 MIPI® 联盟标准
- 支持 Type B 和 Type C 模式:
 - Type B 为 8 位数据接口
 - Type C 支持 3-Line 和 4-Line
- 额外支持 QSPI 模式, QSPI 模式下特征如下:
 - 一次传输包含一字节命令阶段、一到三字节可调的地址阶段、可选的数据阶段
 - 命令、地址和数据读写分别都支持 1 线/4 线模式, 可以任意组合
- 除 QSPI 模式外, 一次传输都支持可选的命令与数据阶段, QSPI 接口有另外的地址阶段
- 数据阶段可设置为以字节为单位的 normal 模式和以像素为单位的 pixel 模式:
 - 数据阶段 normal 模式用于传输配置数据, 单次最多可写入 256 个字节, 最多可读出 8 个字节
 - 数据阶段 pixel 模式用于传输像素数据, 数据量以像素为单位, 单次最多可以写 2 的 24 次方个像素, 不可读
- 输入像素格式支持 RGB 和 YUV444, 其中 RGB 支持以下八种内存排列方式:

- RGBA8888
- BGRA8888
- ARGB8888
- ABGR8888
- RGB888
- BGR888
- RGB565
- BGR565
- YUV444 支持以下六种内存排列方式：
 - YUVN8888
 - VUYN8888
 - NYUV8888
 - NVUY8888
 - YUV888
 - VUY888
- YUV444 格式到 RGB565/666/888 格式的硬件转码
- 输出像素格式支持 RGB565、RGB666 和 RGB888
- CS 信号拉高释放条件可配置
- 多种中断控制
- 支持 DMA 传输模式

21.3 功能描述

DBI 模块基本框图如图所示。

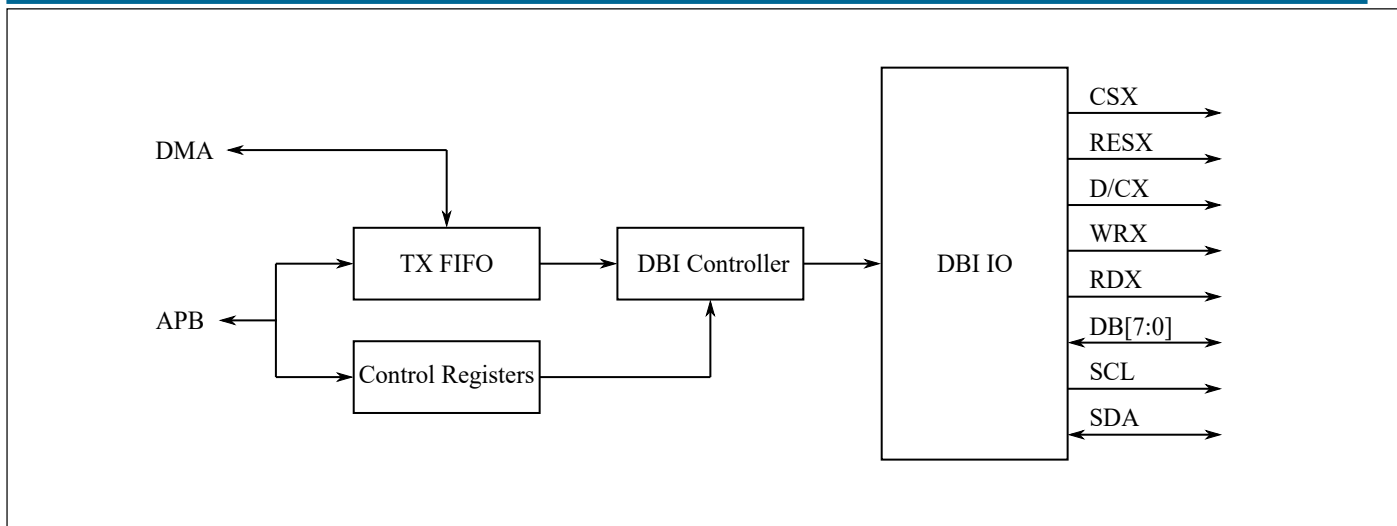


图 21.1: DBI 基本框图

21.3.1 DBI Type B

21.3.1.1 写时序

DBI Type B 模式拥有 8 位并行数据线，MCU 向显示模块写入数据的时序如下图所示：

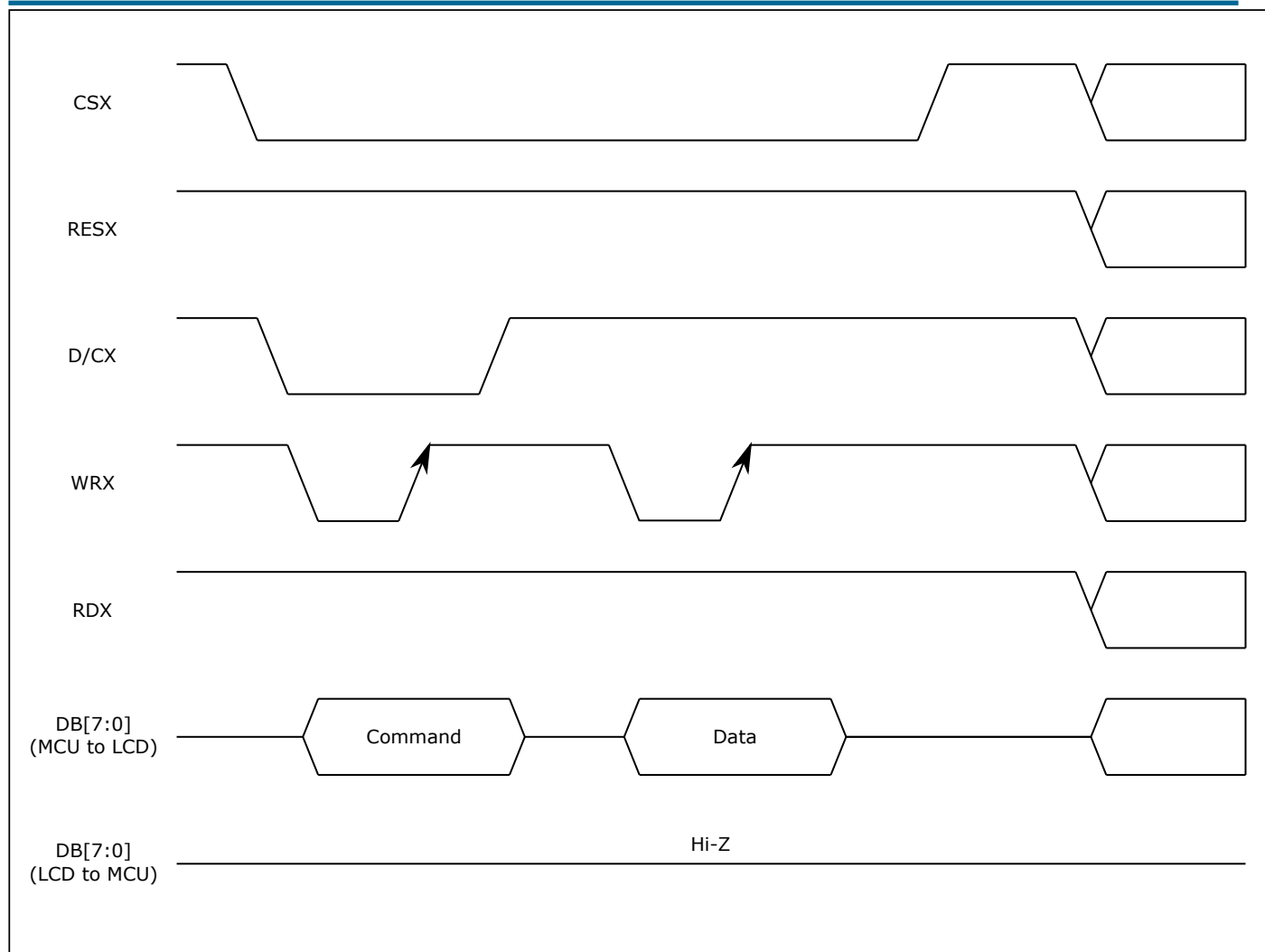


图 21.2: 写时序

其中:

- **CSX:** 片选信号。当该信号为低时显示模块被选中，为高时显示模块将忽略其他所有的接口信号；
- **RESX:** 外部复位信号。当该信号为低时显示模块被复位；
- **D/CX:** 数据/命令选择信号。当该信号为 0 时，DB[7:0] 位为命令；当该信号为 1 时，DB[7:0] 位为 RAM 数据或命令参数；
- **WRX:** 并行数据写入选通信号。该信号在写入周期内从高到低驱动，然后拉回到高。显示模块在该信号的上升沿读取 MCU 传输的信息。该信号是一种非同步信号，可在不使用时终止；
- **RDX:** 并行数据读取选通信号。该信号从高到低被驱动，然后在读取周期中被拉回到高。MCU 在该信号的上升沿读取显示模块传输的信息。该信号是一种非同步信号，可在不使用时终止；
- **DB[7:0]:** 8 位数据信号。用于传输命令、命令参数或数据。

21.3.1.2 读时序

MCU 从显示模块读取数据的时序如下图所示：

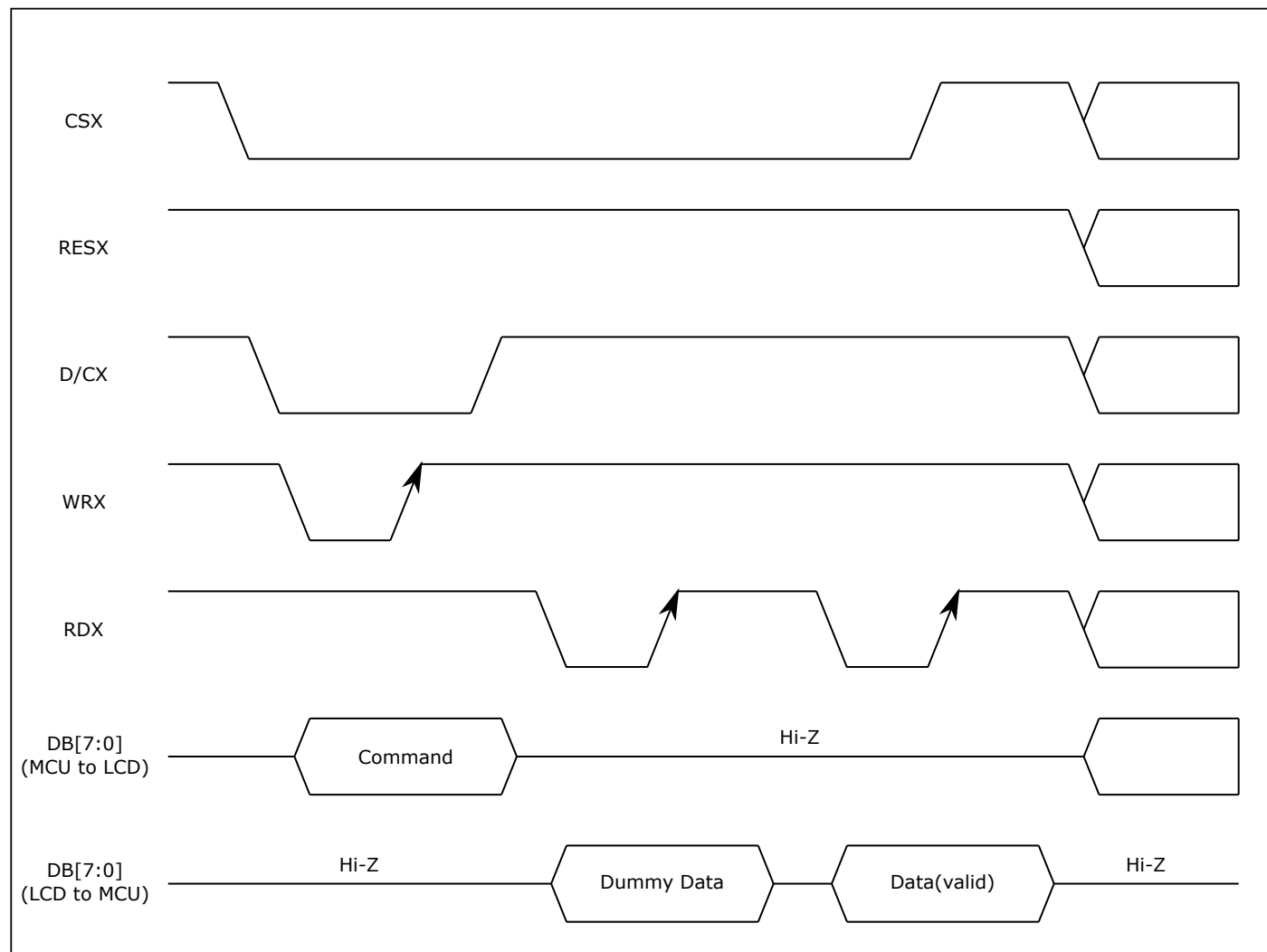


图 21.3: 读时序

21.3.1.3 输出 RGB565

RGB565 格式数据输出如下图所示：

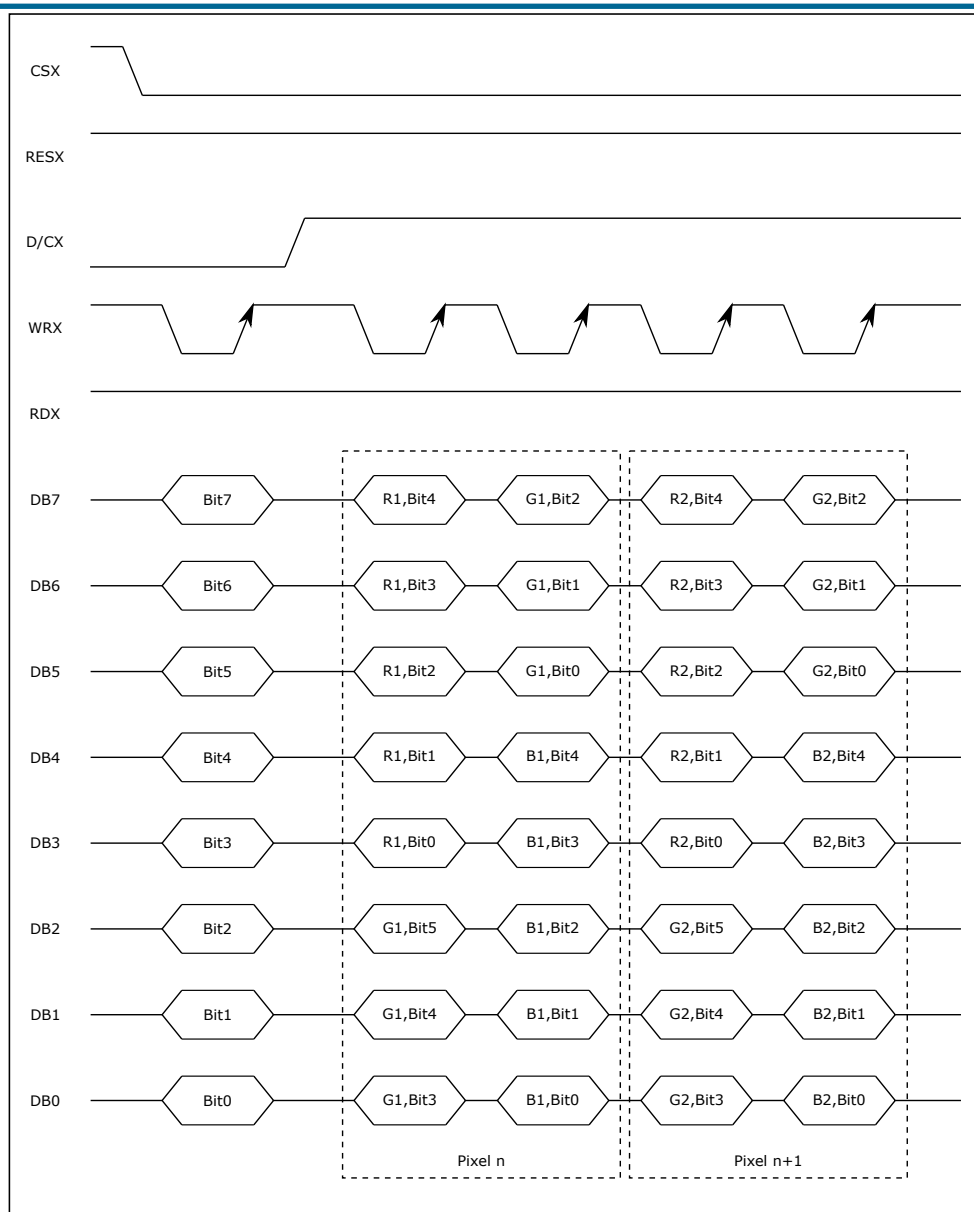


图 21.4: RGB565 输出

21.3.1.4 输出 RGB666

RGB666 格式数据输出如下图所示:

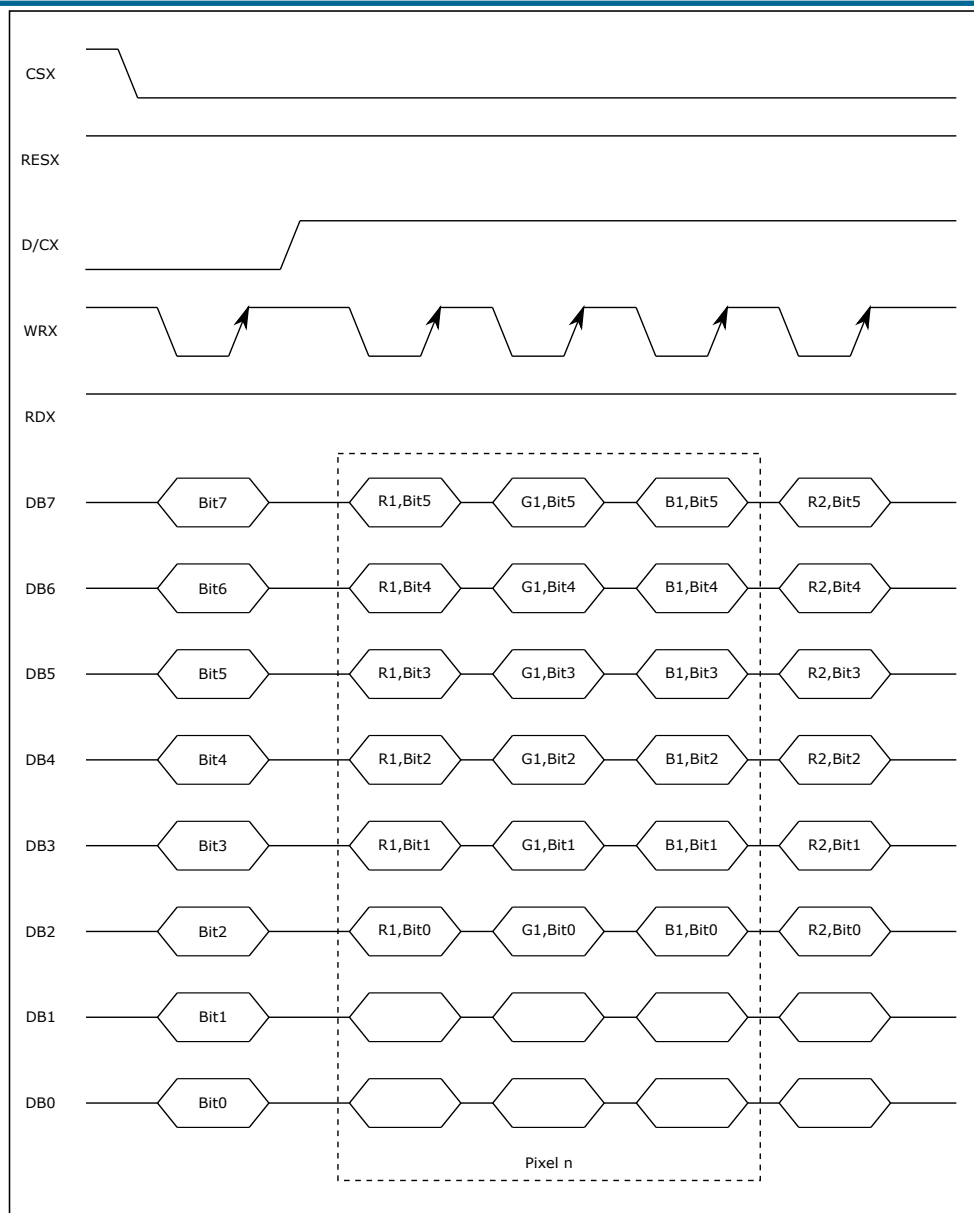


图 21.5: RGB666 输出

21.3.1.5 输出 RGB888

RGB888 格式数据输出如下图所示:

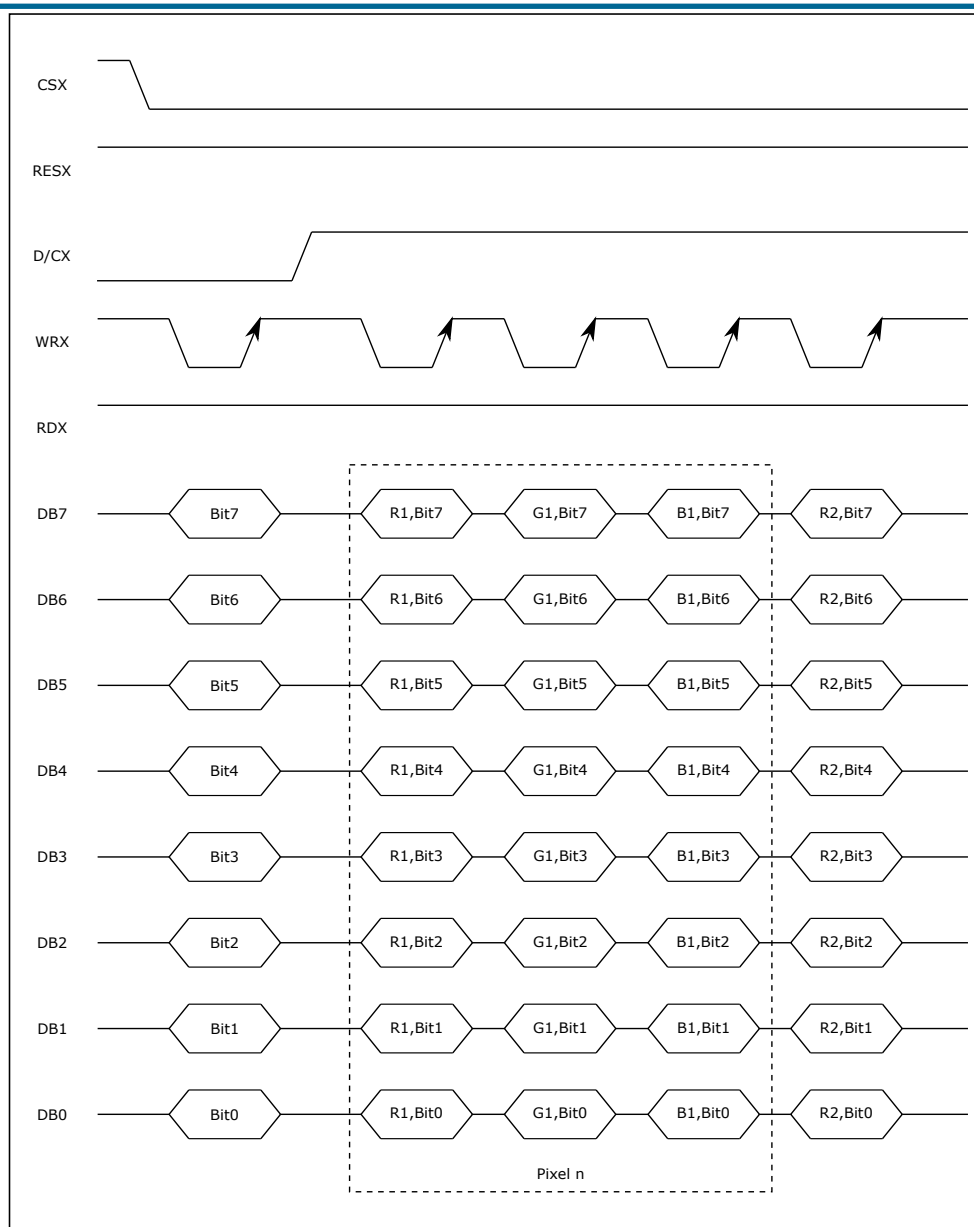


图 21.6: RGB888 输出

21.3.2 DBI Type C 3-Line

21.3.2.1 写时序

DBI Type C 模式是 3 线 9 位串行接口，MCU 向显示模块写入数据的时序如下图所示：

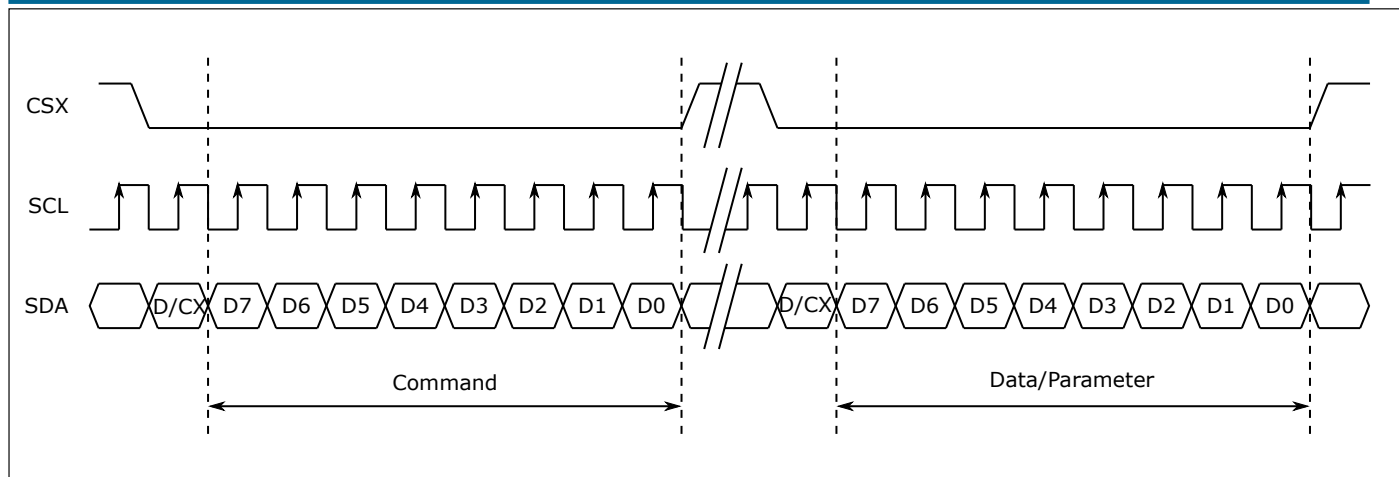


图 21.7: 写时序

其中:

- **CSX**: 片选信号。当该信号为低时显示模块被选中，为高时显示模块将忽略其他所有的接口信号；
- **SCL**: 串行时钟输入信号。用于为数据传输提供时钟信号。
- **SDA**: 串行数据输入/输出信号。在写操作中，串行数据包包含一个 **D/CX**(数据/命令) 选择位和一个传输字节。如果 **D/CX** 位为低，则传输字节为命令；如果 **D/CX** 位为高，则传输字节为显示数据或命令参数。

21.3.2.2 读时序

MCU 从显示模块读取数据的时序如下图所示:

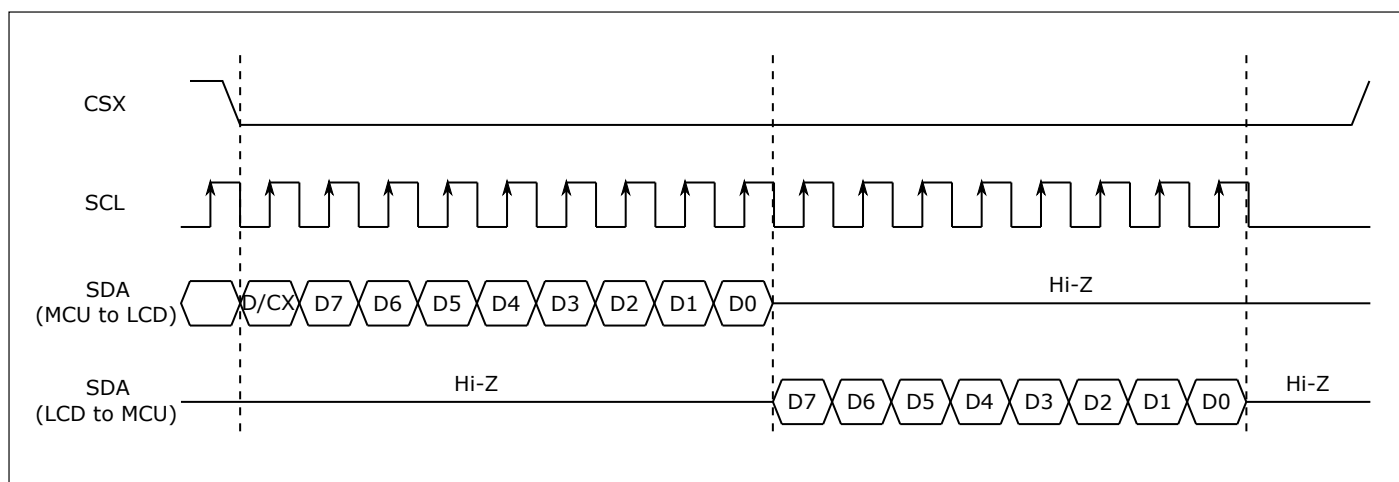


图 21.8: 读时序

21.3.2.3 输出 RGB565

RGB565 格式数据输出如下图所示：

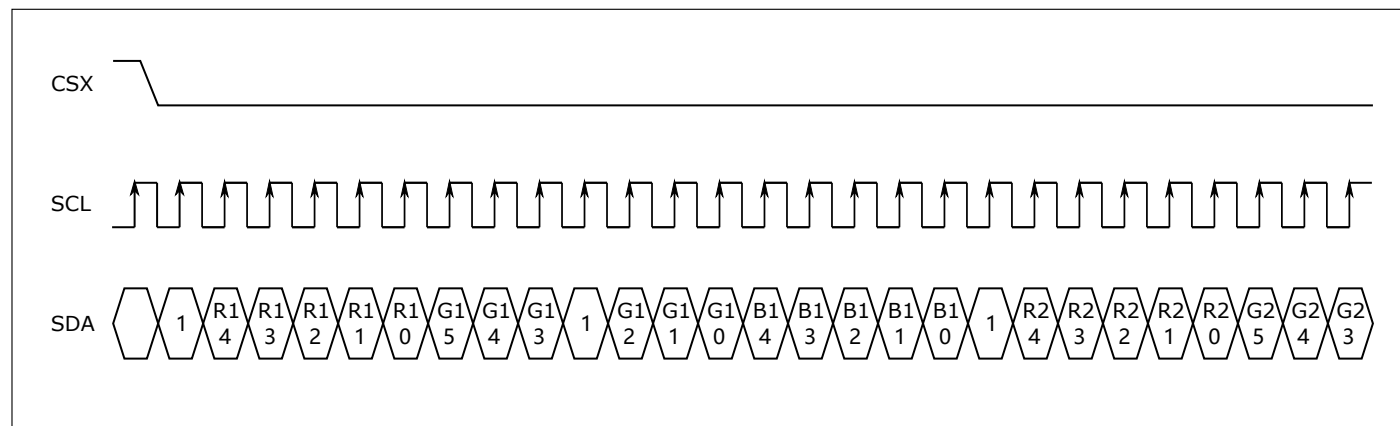


图 21.9: RGB565 输出

21.3.2.4 输出 RGB666

RGB666 格式数据输出如下图所示：

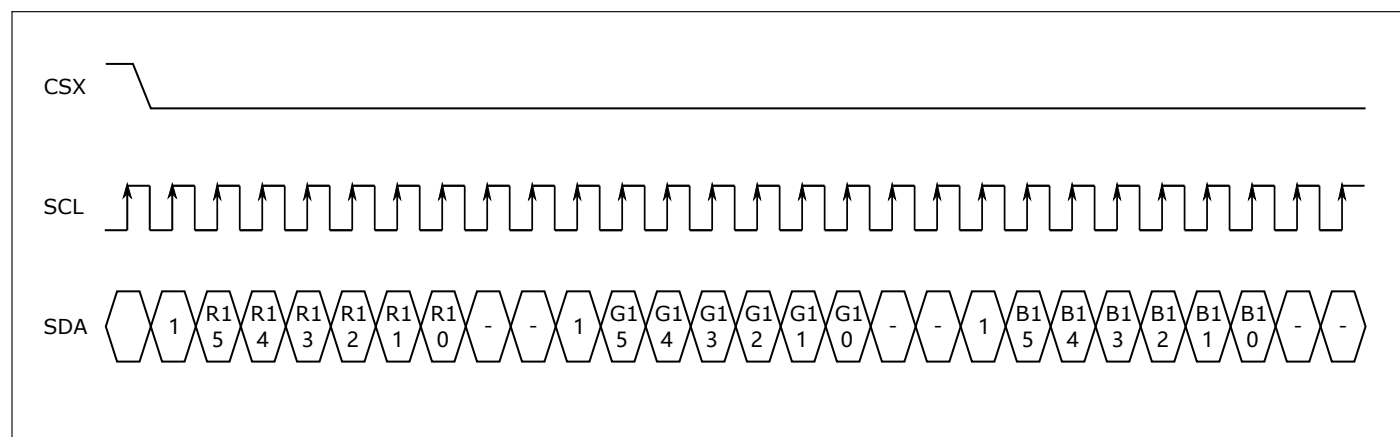


图 21.10: RGB666 输出

21.3.2.5 输出 RGB888

RGB888 格式数据输出如下图所示：

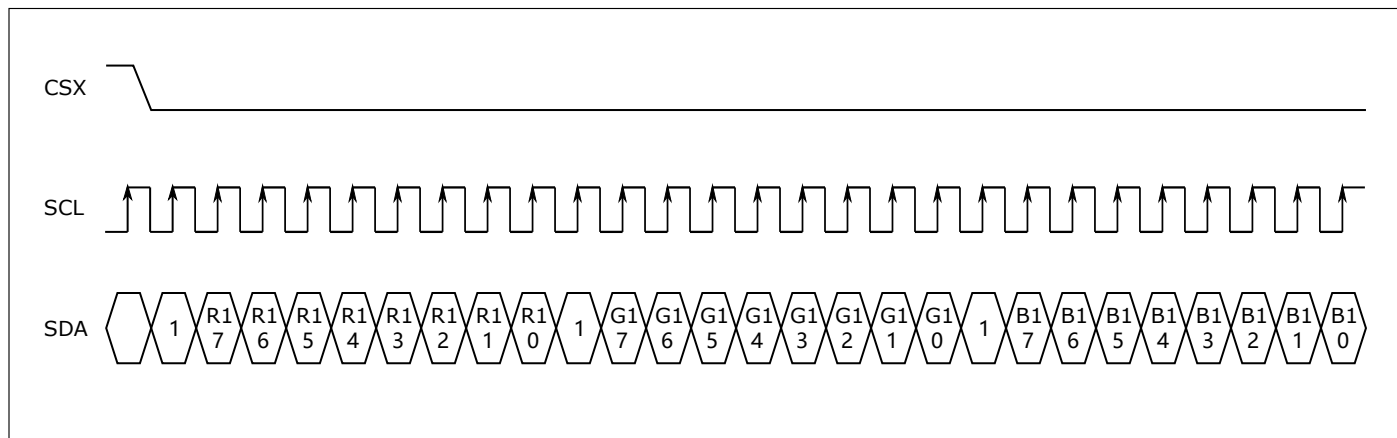


图 21.11: RGB888 输出

21.3.3 DBI Type C 4-Line

21.3.3.1 写时序

DBI Type C 模式是 4 线 8 位串行接口，MCU 向显示模块写入数据的时序如下图所示：

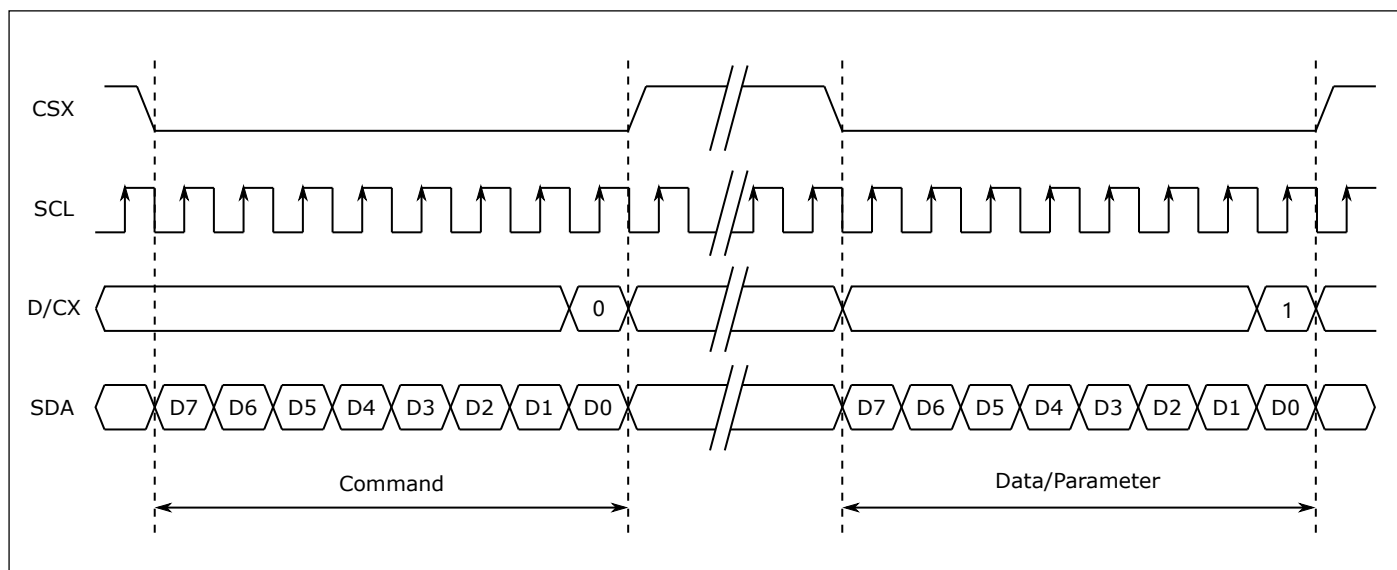


图 21.12: 写时序

其中：

- **CSX**: 片选信号。当该信号为低时显示模块被选中，为高时显示模块将忽略其他所有的接口信号；
- **D/CX**: 数据/命令选择信号。如果该信号为低，则 **SDA** 传输的信息为命令；如果该信号为高，则 **SDA** 传输的信息

为显示数据或命令参数。

- **SCL**: 串行时钟输入信号。用于为数据传输提供时钟信号。
- **SDA**: 串行数据输入/输出信号。用于传输命令、命令参数或数据。

21.3.3.2 读时序

MCU 从显示模块读取数据的时序如下图所示：

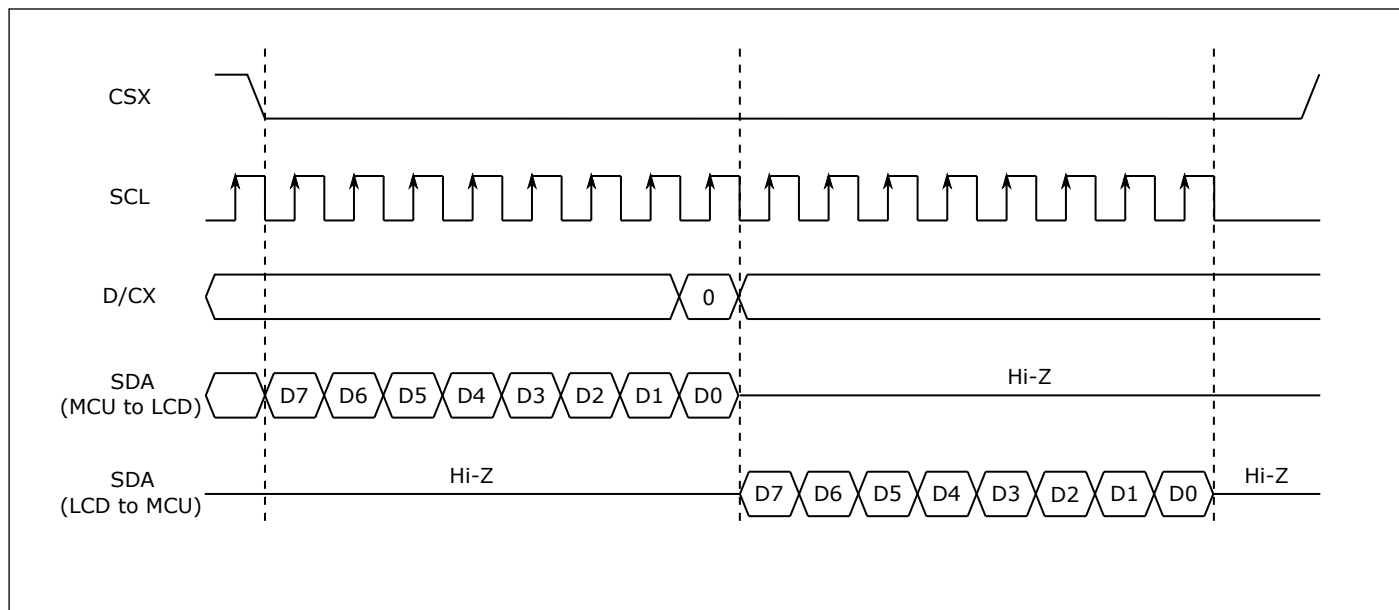


图 21.13: 读时序

21.3.3.3 输出 RGB565

RGB565 格式数据输出如下图所示：

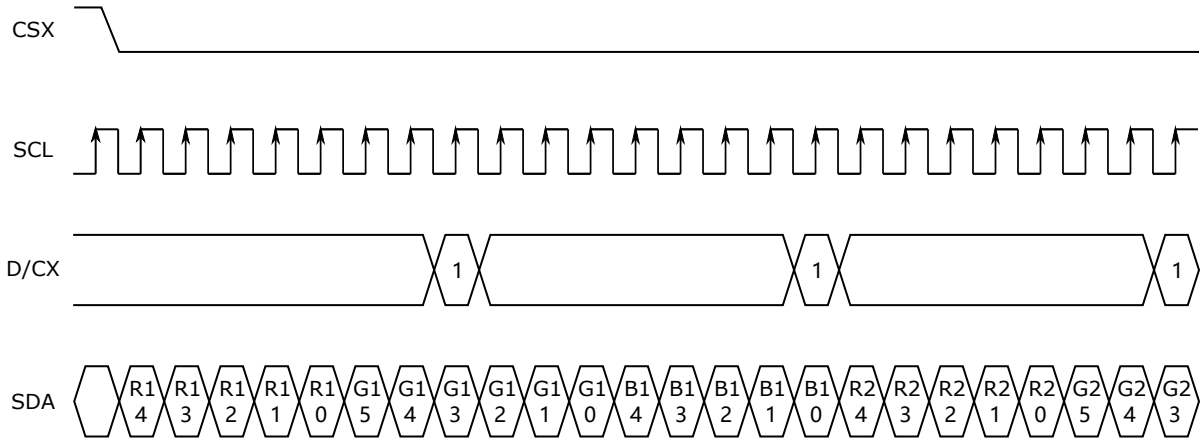


图 21.14: RGB565 输出

21.3.3.4 输出 RGB666

RGB666 格式数据输出如下图所示：

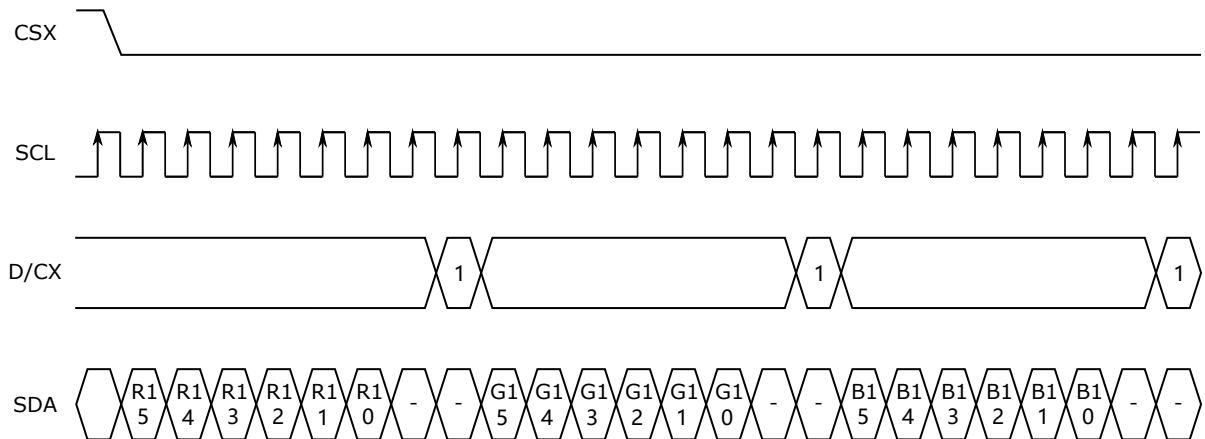


图 21.15: RGB666 输出

21.3.3.5 输出 RGB888

RGB888 格式数据输出如下图所示：

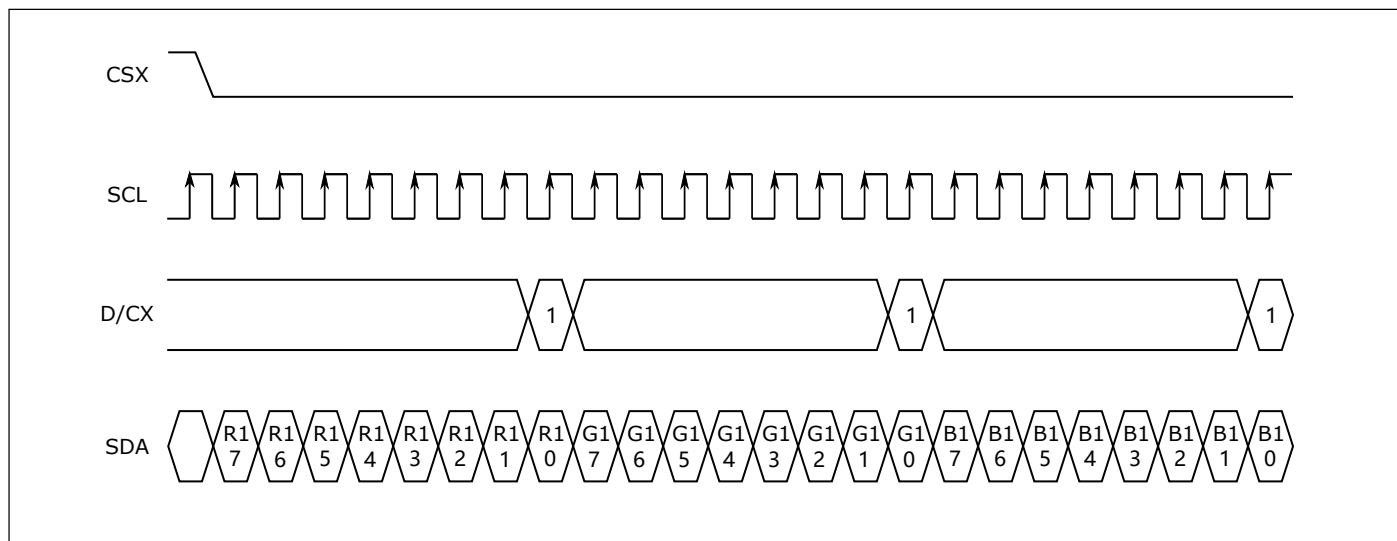


图 21.16: RGB888 输出

21.3.4 QSPI

21.3.4.1 写时序

以 1 线命令、1 线 3 字节长度地址、1 线数据模式为例，MCU 向显示模块写入数据的时序如下图所示：

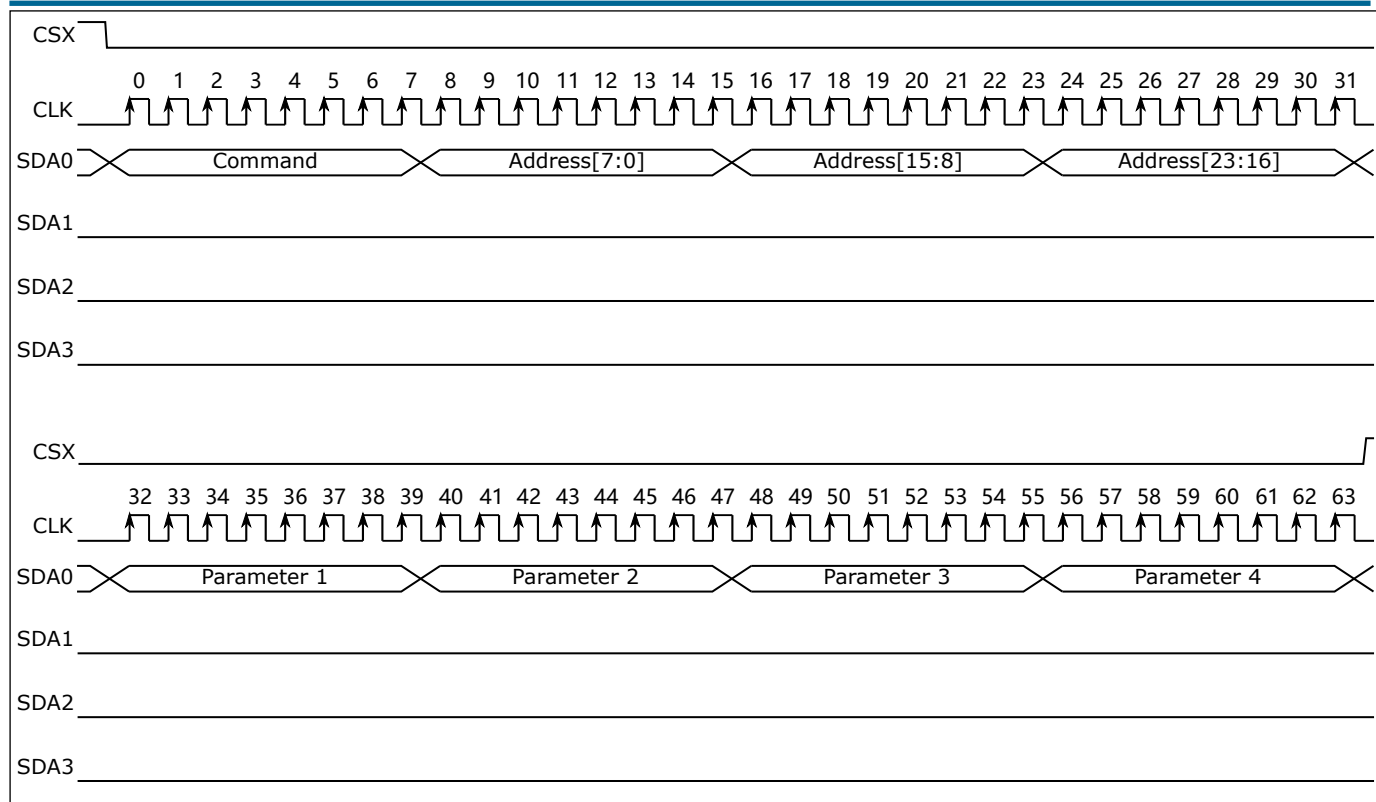


图 21.17: 写时序

其中：

- **CSX:** 片选信号。当该信号为低时显示模块被选中，为高时显示模块将忽略其他所有的接口信号；
- **CLK:** 时钟输入信号。用于为数据传输提供时钟信号。
- **SDA0:** 数据线 0，用于传输命令、地址或数据。
- **SDA1:** 数据线 1，用于传输命令、地址或数据。
- **SDA2:** 数据线 2，用于传输命令、地址或数据。
- **SDA3:** 数据线 3，用于传输命令、地址或数据。

21.3.4.2 读时序

以 1 线命令、1 线 3 字节长度地址、1 线数据模式为例，MCU 从显示模块读取数据的时序如下图所示：

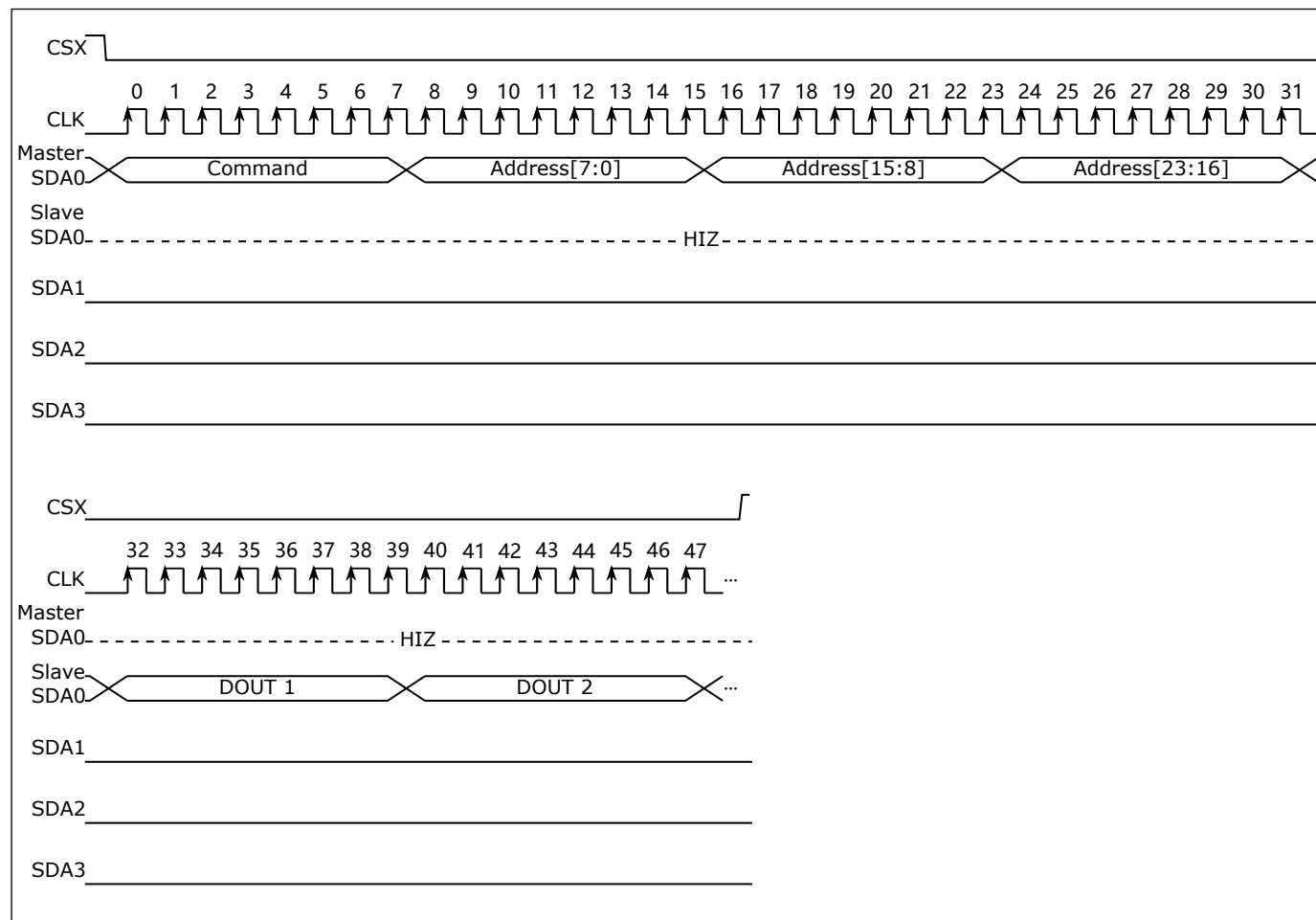


图 21.18: 读时序

21.3.4.3 1 线命令、1 线地址、1 线数据模式

以命令为 0x02，地址为 3 字节 0x002C00/0x003C00 为例，1 线命令、1 线地址、1 线数据模式下 RGB565 格式数据输出如下图所示：

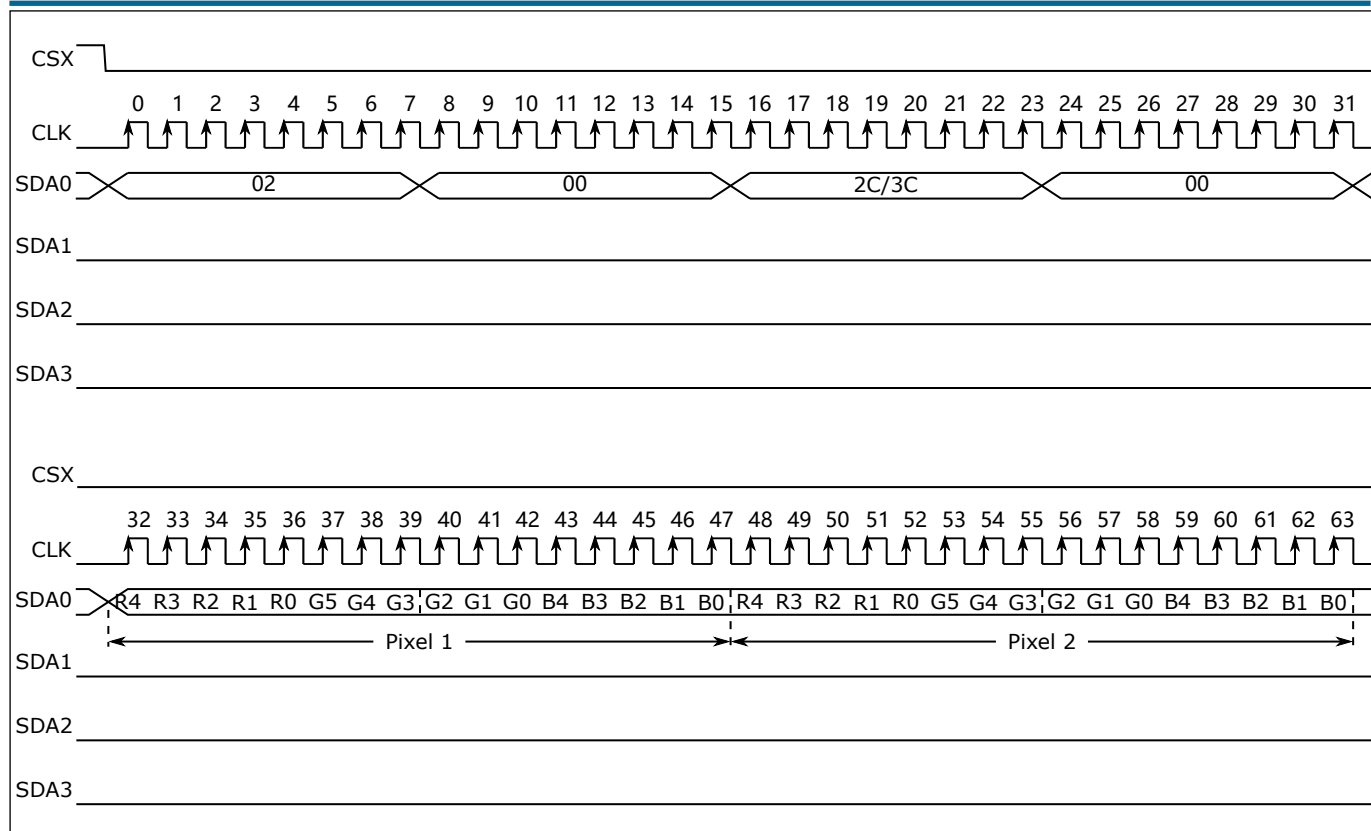


图 21.19: RGB565 输出

RGB666 格式数据输出如下图所示:

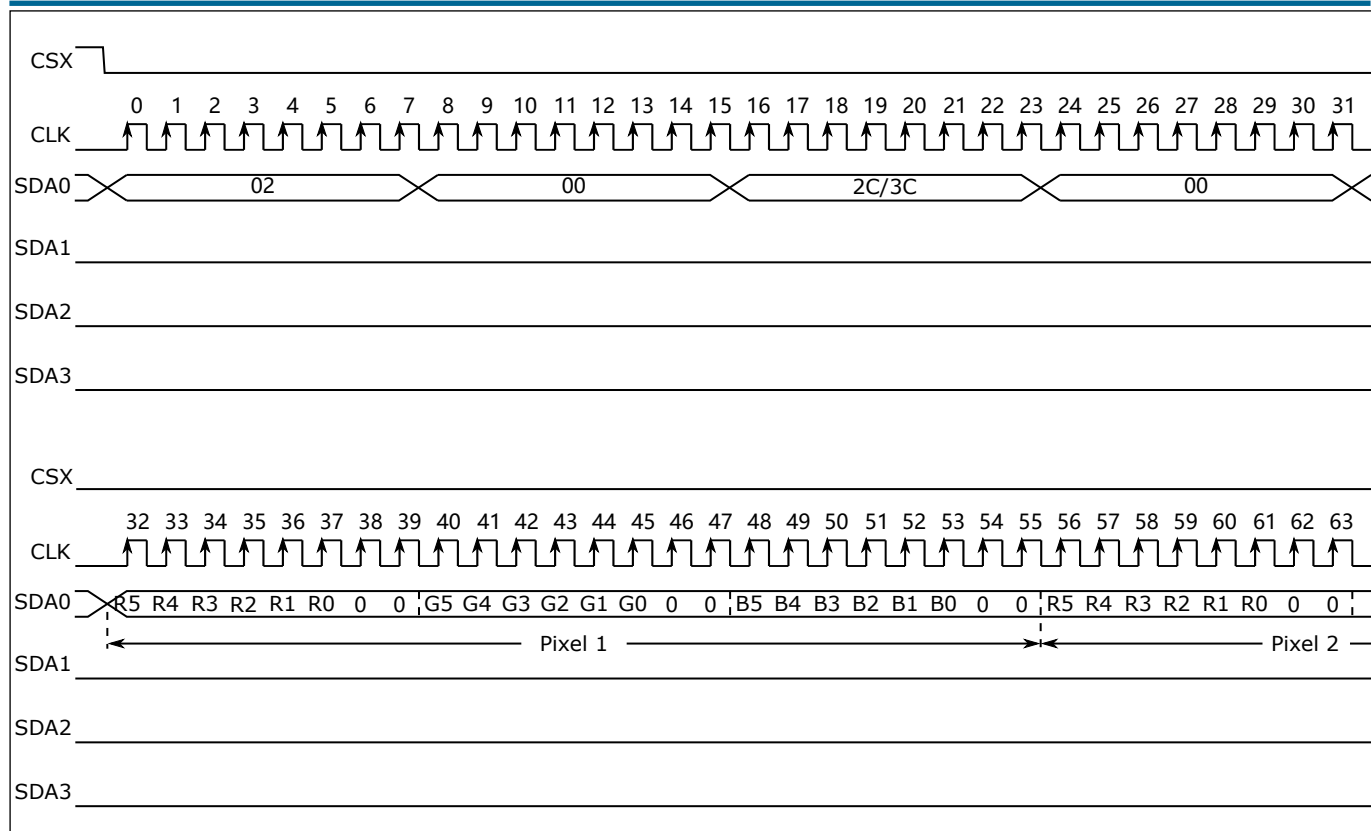


图 21.20: RGB666 输出

RGB888 格式数据输出如下图所示:

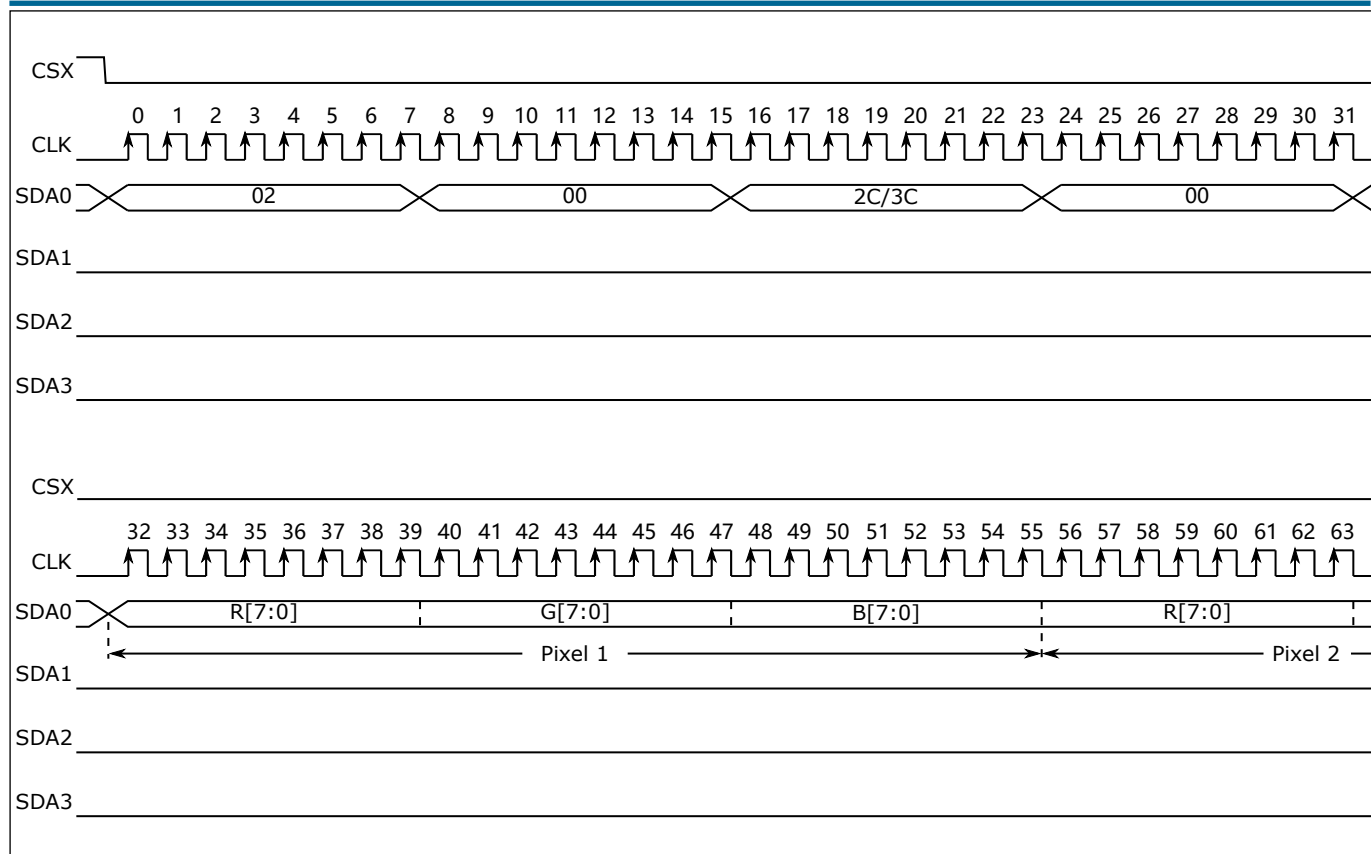


图 21.21: RGB888 输出

21.3.4.4 1 线命令、1 线地址、4 线数据模式

以命令为 0x32，地址为 3 字节 0x002C00/0x003C00 为例，1 线命令、1 线地址、4 线数据模式下 RGB565 格式数据输出如下图所示：

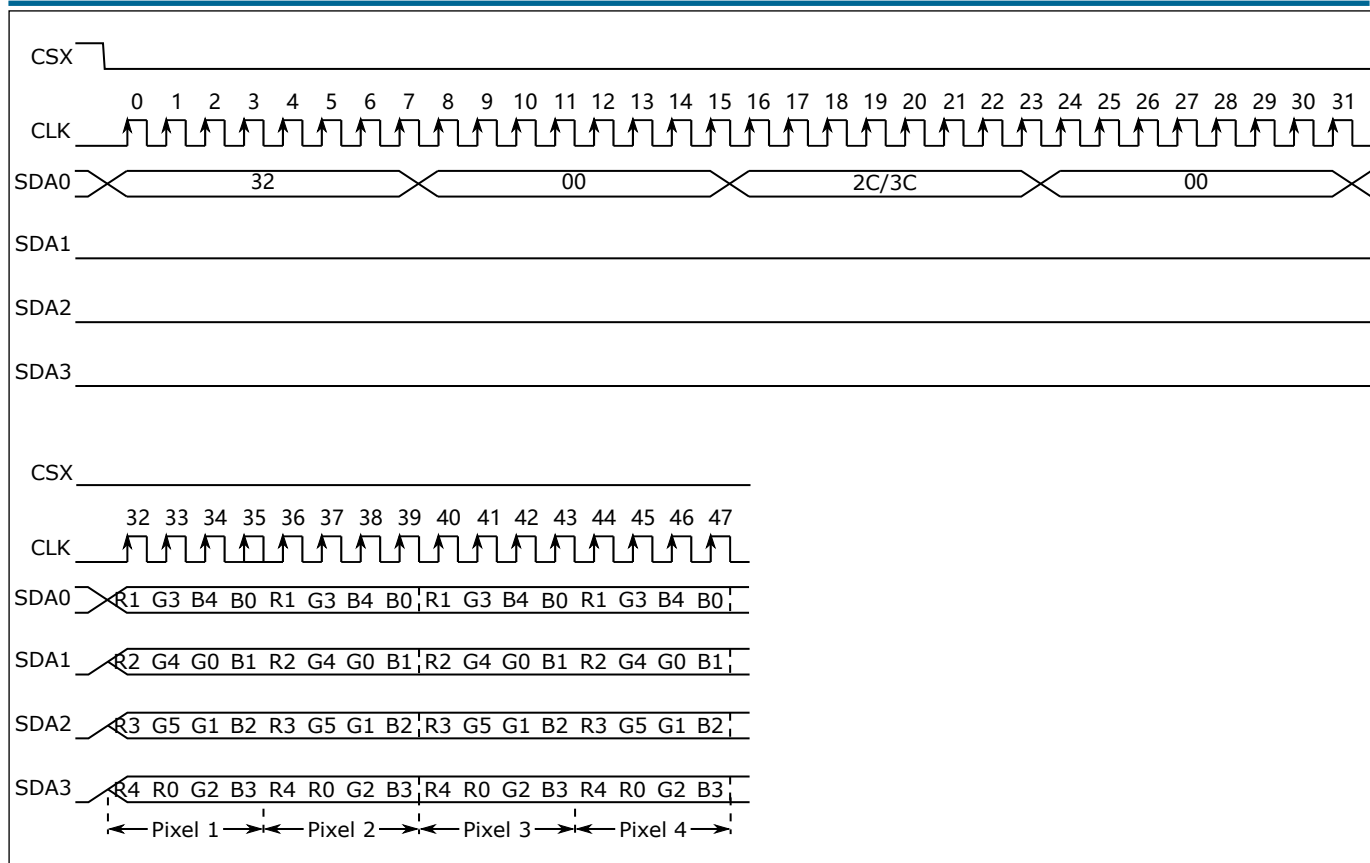


图 21.22: RGB565 输出

RGB666 格式数据输出如下图所示:

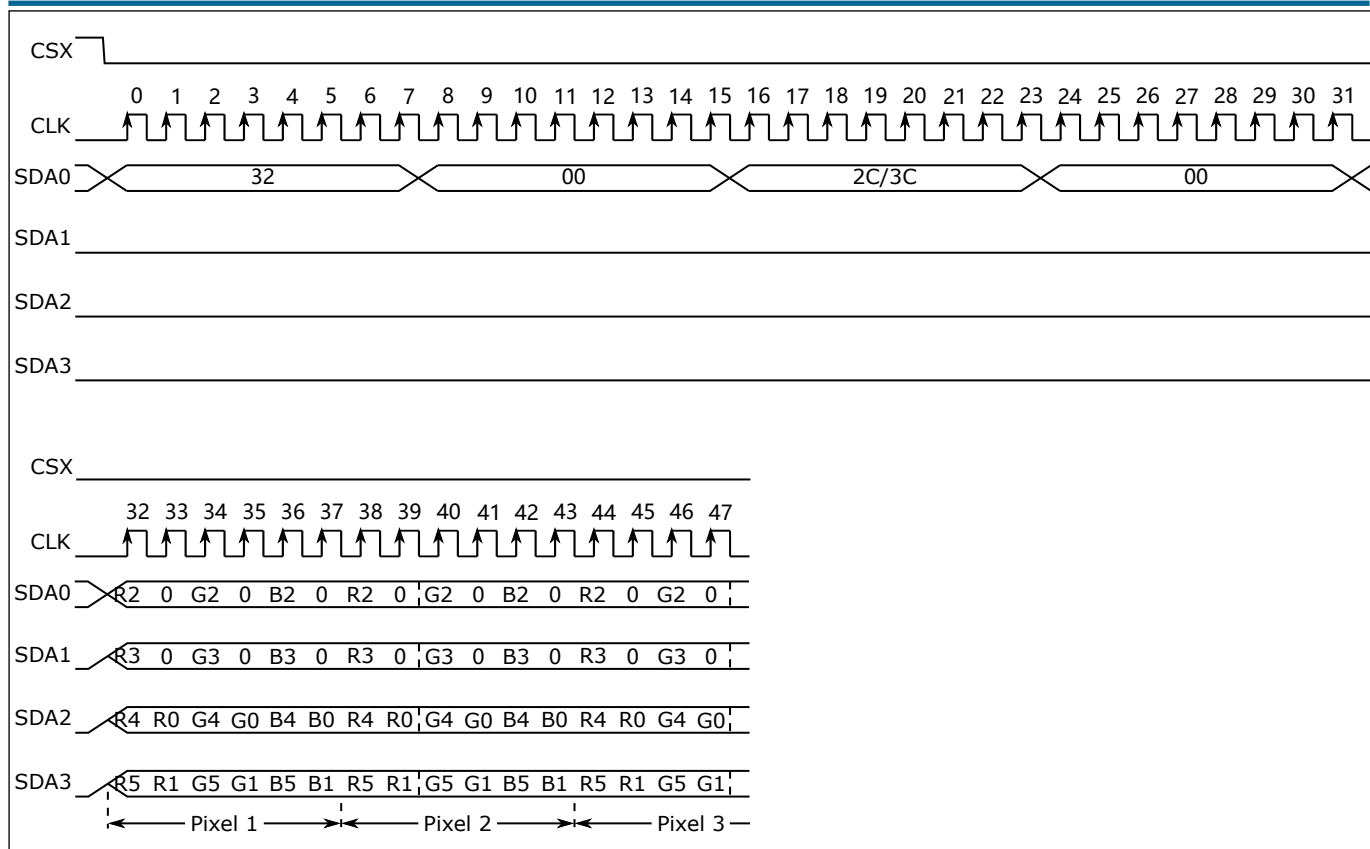


图 21.23: RGB666 输出

RGB888 格式数据输出如下图所示:

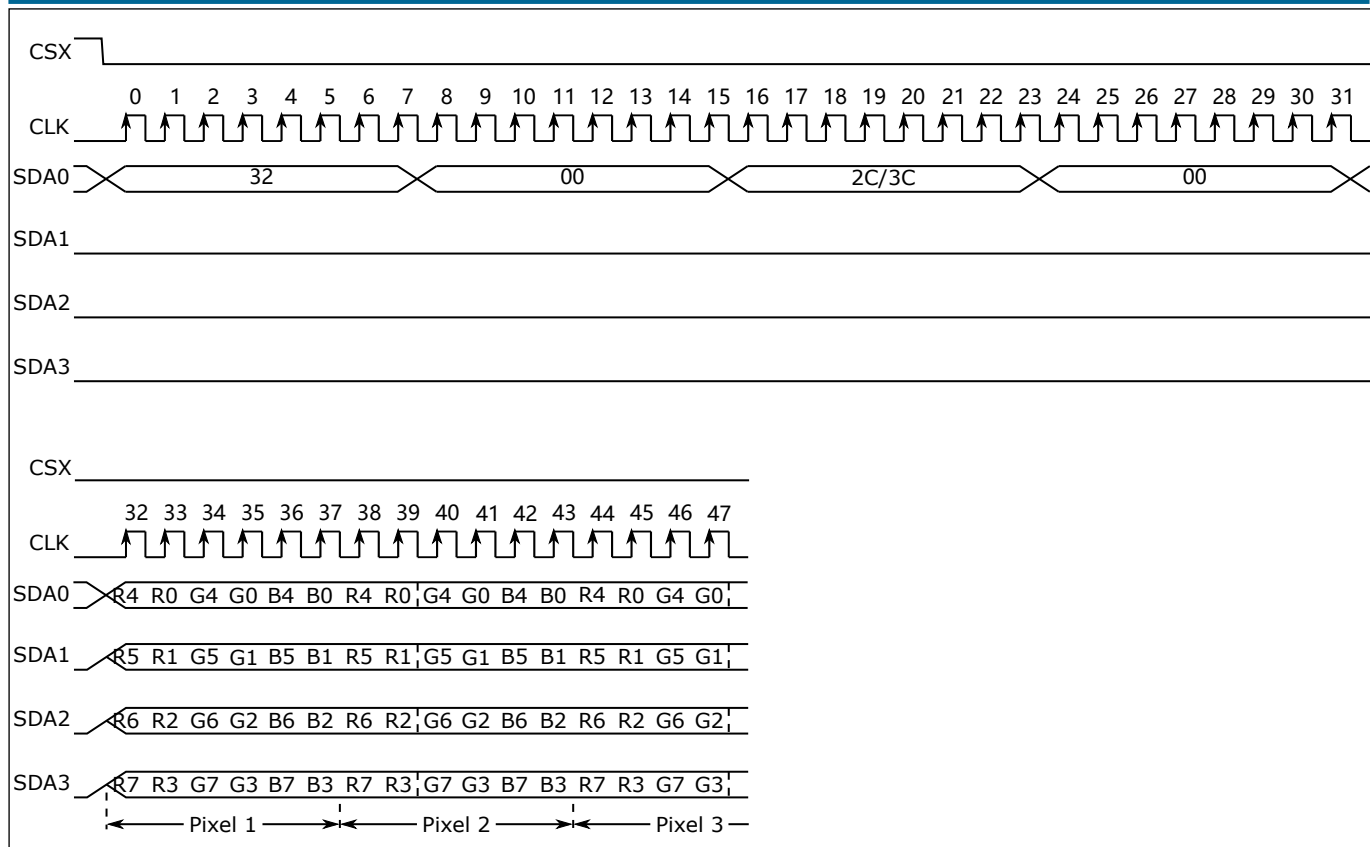


图 21.24: RGB888 输出

21.3.4.5 1 线命令、4 线地址、4 线数据模式

以命令为 0x12，地址为 3 字节 0x002C00/0x003C00 为例，1 线命令、4 线地址、4 线数据模式下 RGB565 格式数据输出如下图所示：

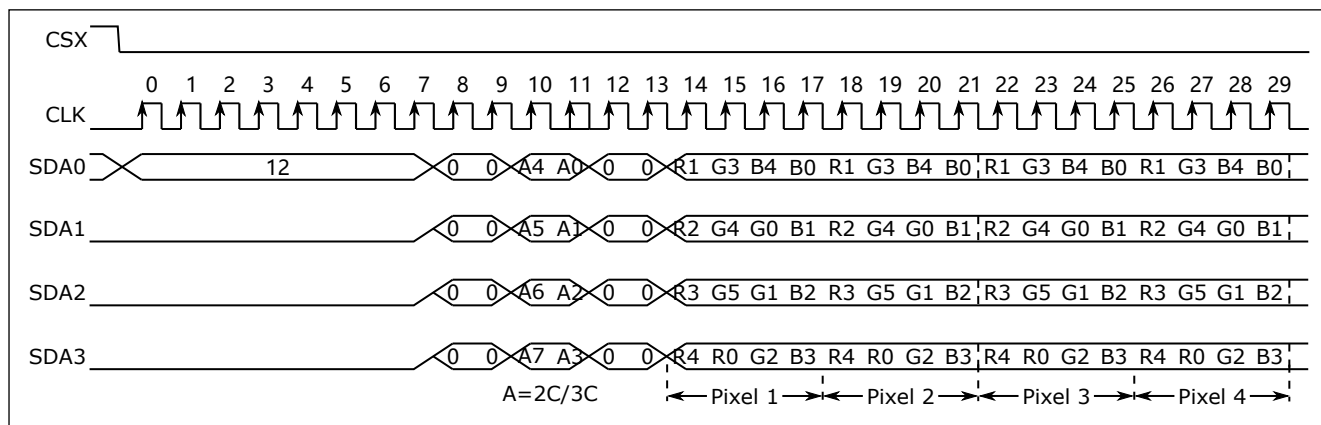


图 21.25: RGB565 输出

RGB666 格式数据输出如下图所示：

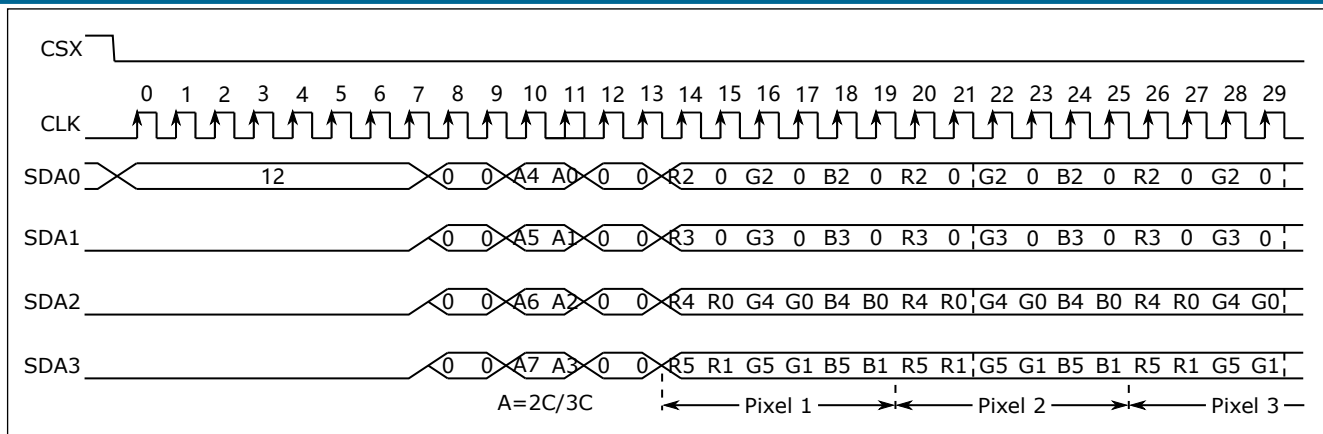


图 21.26: RGB666 输出

RGB888 格式数据输出如下图所示：

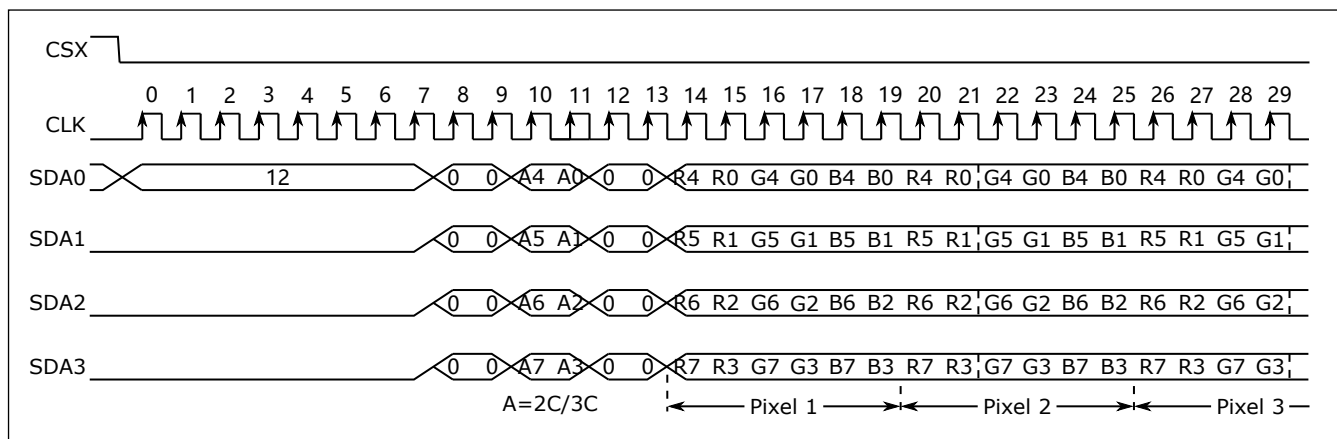


图 21.27: RGB888 输出

21.3.5 输入像素格式

DBI 模块支持 8 种不同的输入像素 RGB 格式和 6 种 YUV444 格式，各种 RGB 格式在内存中的排布如下图所示：

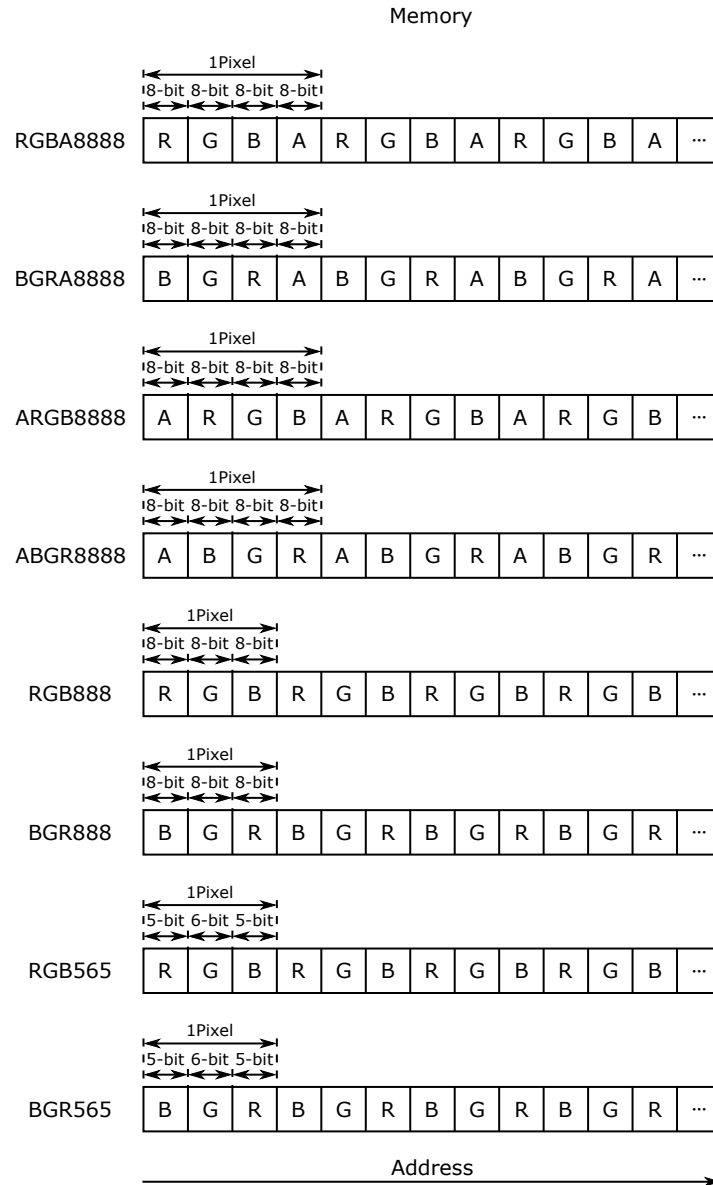


图 21.28: 输入像素 RGB 格式

对于 YUV444 格式的内存排布，只需用 Y 替换 R，用 U 替换 G，用 V 替换 B 即可。

21.3.6 CS 信号拉高释放条件配置

在 Type B 和 Type C 模式下，可以通过寄存器 CONFIG 的位 CONT_EN 配置 CS 信号的拉高释放条件。如果不使能该功能，CS 信号会在每个像素传输完成后释放一次，即每两个像素之间 CS 信号都会拉高。如果使能该功能，在一次传输过程中，CS 信号都不会释放，只有当该次传输全部完成后 CS 信号才会拉高释放。

在 QSPI 模式下，可以通过寄存器 CONFIG 的位 CS_STRETCH 配置 CS 信号的拉高释放条件。如果不使能该功能，在一次传输过程中，当 FIFO 为空（即 FIFO 中的数据全部发出且没有新的数据填入）时，CS 信号会被拉高释放。如果使能该功能，在一次传输过程中，当 FIFO 为空（即 FIFO 中的数据全部发出且没有新的数据填入）时，CS 信号会维持在低电平，等待新的数据填入。

21.3.7 中断

DBI 具有以下几种中断：

- 传输结束中断
- TX FIFO 请求中断
- FIFO 溢出错误中断

通过设置一个像素计数值，传输结束中断会在传输的像素数据达到计数值时产生，Type B 和 Type C 都会产生该中断；

当 DBI_FIFO_CONFIG_1 中 TFICNT 大于 TFITH 时，产生 TX FIFO 请求中断，当条件不满足时该中断标志会自动清除；

FIFO 溢出错误中断会在 TX 发生 Overflow 或者 Underflow 时产生。

21.3.8 DMA

DBI TX 支持 DMA 传输模式，使用该模式需要通过寄存器 DBI_FIFO_CONFIG_1 的位 <TFITH> 设置 TX FIFO 的阈值。当该模式启用后，如果 TFICNT 大于 TFITH，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。

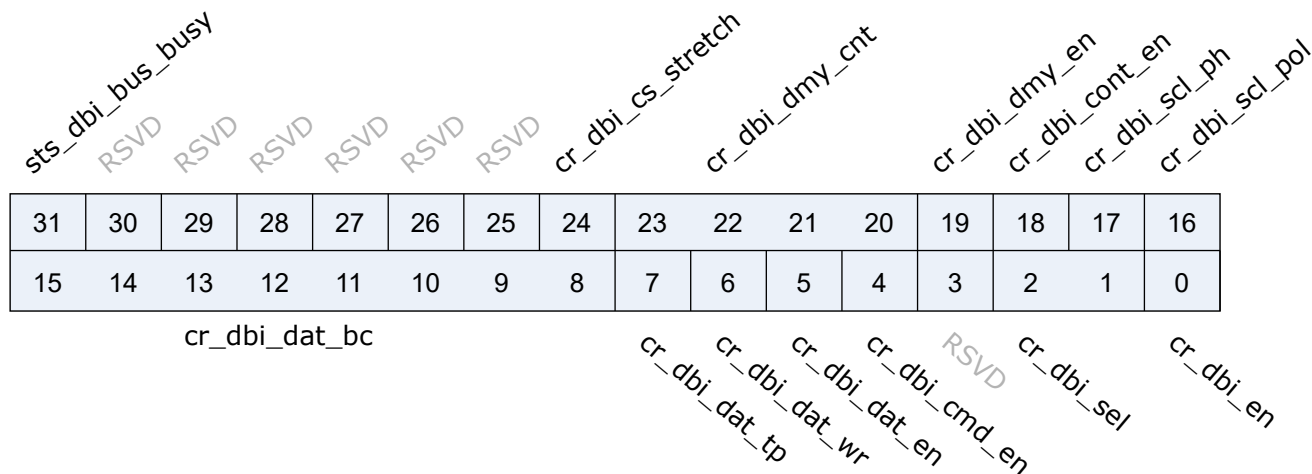
21.4 寄存器描述

名称	描述
dbi_config	DBI configure
qspi_config	QSPI configure
dbi_pix_cnt	Pixel format and count
dbi_prd	DBI period
dbi_cmd	DBI command
dbi_qspi_adr	QSPI address

名称	描述
dbi_rdata_0	Read data 0
dbi_rdata_1	Read data 1
dbi_int_sts	Interrupt status and setting
dbi_yuv_rgb_config_0	YUV2RGB configure0
dbi_yuv_rgb_config_1	YUV2RGB configure1
dbi_yuv_rgb_config_2	YUV2RGB configure2
dbi_yuv_rgb_config_3	YUV2RGB configure3
dbi_yuv_rgb_config_4	YUV2RGB configure4
dbi_yuv_rgb_config_5	YUV2RGB configure5
dbi_fifo_config_0	FIFO format and DMA mode
dbi_fifo_config_1	FIFO threshold and available count
dbi_fifo_wdata	TX FIFO

21.4.1 dbi_config

地址：0x2000a800



位	名称	权限	复位值	描述
31	sts_dbi_bus_busy	r	1'b0	Indicator of dbi bus busy
30:25	RSVD			

位	名称	权限	复位值	描述
24	cr_dbi_cs_stretch	r/w	1'b0	Enable signal of CS-low stretch mode 1'b0: Disabled, if FIFO is empty during a pixel data transfer, CS will de-assert before FIFO is filled again and new transfer starts 1'b1: Enabled, if FIFO is empty during a pixel data transfer, CS will stay asserted while waiting for FIFO to be filled again
23:20	cr_dbi_dmy_cnt	r/w	4'd0	Dummy cycle count, unit: period defined by dbi_prd_d Effective only in Type C (Fixed to 1 dbi_prd_d period in Type B)
19	cr_dbi_dmy_en	r/w	1'b0	Enable signal of dummy cycle(s) 1'b0: Disabled, no dummy cycle(s) between command phase and data phase 1'b1: Enabled, dummy cycle(s) will be inserted between command phase and data phase Note: Don't-care if QSPI mode is selected (no dummy cycle)
18	cr_dbi_cont_en	r/w	1'b1	Enable signal of pixel data continuous transfer mode 1'b0: Disabled, CS_n will de-assert between each pixel 1'b1: Enabled, CS_n will stay asserted between each consecutive pixel Note: Don't-care if QSPI mode is selected (always in continuous mode)
17	cr_dbi_scl_ph	r/w	1'b0	SCL clock phase inverse signal
16	cr_dbi_scl_pol	r/w	1'b1	SCL clock polarity 0: SCL output LOW at IDLE state 1: SCL output HIGH at IDLE state
15:8	cr_dbi_dat_bc	r/w	8'd0	Data byte count of normal data (pixel data count is determined by cr_dbi_pix_cnt) Note: Normal read data can only be up to 8-byte (8'd7). No limit for normal write data (can be up to 256-byte using FIFO)
7	cr_dbi_dat_tp	r/w	1'b0	Data type select 1'b0: Normal data (parameter) 1'b1: Pixel data Note: Read command supports normal data only
6	cr_dbi_dat_wr	r/w	1'b1	Data phase Read/Write select 1'b0: Read data 1'b1: Write data

位	名称	权限	复位值	描述
5	cr_dbi_dat_en	r/w	1'b1	Data enable signal 1'b0: Data phase disabled 1'b1: Data phase enabled
4	cr_dbi_cmd_en	r/w	1'b1	Command enable signal 1'b0: No command phase 1'b1: Command will be sent Note: Don't-care if QSPI mode is selected (Command always enabled)
3	RSVD			
2:1	cr_dbi_sel	r/w	2'd0	DBI mode select 2'd0: DBI Type B 2'd1: DBI Type C, 4-wire mode 2'd2: DBI Type C, 3-wire mode 2'd3: QSPI mode
0	cr_dbi_en	r/w	1'b0	Enable signal of DBI function Asserting this bit will trigger the transaction, and should be de-asserted after finish

21.4.2 qspi_config

地址: 0x2000a804

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	cr_qspi_adr_bc	RSVD	cr_qspi_dat_4b	cr_qspi_adr_4b	cr_qspi_cmd_4b

位	名称	权限	复位值	描述
31:6	RSVD			
5:4	cr_qspi_adr_bc	r/w	2'd2	QSPI Address byte count
3	RSVD			

位	名称	权限	复位值	描述
2	cr_qspi_dat_4b	r/w	1'b1	QSPI Data 4-bit (quad) mode 1'b0: Data sent/received in 1-bit mode 1'b1: Data sent/received in 4-bit mode
1	cr_qspi_adr_4b	r/w	1'b1	QSPI Address (display command) 4-bit (quad) mode 1'b0: Address sent in 1-bit mode 1'b1: Address sent in 4-bit mode
0	cr_qspi_cmd_4b	r/w	1'b0	QSPI Command 4-bit (quad) mode 1'b0: Command sent in 1-bit mode 1'b1: Command sent in 4-bit mode

21.4.3 dbi_pix_cnt

地址: 0x2000a808

cr_dbi_pix_format															
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_dbi_pix_cnt															

位	名称	权限	复位值	描述
31	cr_dbi_pix_format	r/w	1'b0	Pixel format 1'b0: RGB565 1'b1: RGB888/RGB666
30:24	RSVD			
23:0	cr_dbi_pix_cnt	r/w	24'h0	Pixel count Note: Should be a multiple of 2 if RGB565 is selected and a multiple of 4 if RGB888/RGB666 mode is selected

21.4.4 dbi_prd

地址：0x2000a80c

cr_dbi_prd_d_ph_1								cr_dbi_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
cr_dbi_prd_i								cr_dbi_prd_s							

位	名称	权限	复位值	描述
31:24	cr_dbi_prd_d_ph_1	r/w	8'd15	Length of DATA phase 1 (please refer to "Timing" tab)
23:16	cr_dbi_prd_d_ph_0	r/w	8'd15	Length of DATA phase 0 (please refer to "Timing" tab)
15:8	cr_dbi_prd_i	r/w	8'd15	Length of INTERVAL between pixel data (please refer to "Timing" tab)
7:0	cr_dbi_prd_s	r/w	8'd15	Length of START/STOP condition (please refer to "Timing" tab)

21.4.5 dbi_cmd

地址：0x2000a810

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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	cr_dbi_cmd						

位	名称	权限	复位值	描述
31:8	RSVD			
7:0	cr_dbi_cmd	r/w	8'h2C	DBI Command

21.4.6 dbi_qspi_adr

地址: 0x2000a814

cr_qspi_adr

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_qspi_adr

位	名称	权限	复位值	描述
31:0	cr_qspi_adr	r/w	32'h00002000	QSPI Address (display command) Note: LSB is sent first

21.4.7 dbi_rdata_0

地址: 0x2000a818

sts_dbi_rdata_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

sts_dbi_rdata_0

位	名称	权限	复位值	描述
31:0	sts_dbi_rdata_0	r	32'h0	Data read from display IC using read command

21.4.8 dbi_rdata_1

地址: 0x2000a81c

sts_dbi_rdata_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

sts_dbi_rdata_1

位	名称	权限	复位值	描述
31:0	sts_dbi_rdata_1	r	32'h0	Data read from display IC using read command

21.4.9 dbi_int_sts

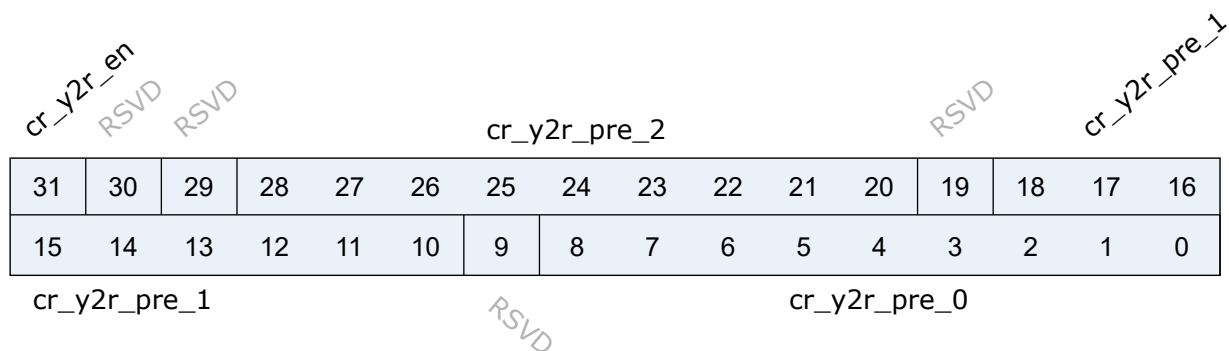
地址: 0x2000a830

RSVD	RSVD	RSVD	RSVD	RSVD	cr_dbi_fer_en	cr_dbi_txf_en	cr_dbi_end_en	RSVD	RSVD	RSVD	RSVD	RSVD	rsvd	rsvd	cr_dbi_end_clr
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	cr_dbi_fer_mask	cr_dbi_txf_mask	cr_dbi_end_mask	RSVD	RSVD	RSVD	RSVD	RSVD	dbi_fer_int	dbi_txf_int	dbi_end_int

位	名称	权限	复位值	描述
31:27	RSVD			
26	cr_dbi_fer_en	r/w	1'b1	Interrupt enable of dbi_fer_int
25	cr_dbi_txf_en	r/w	1'b1	Interrupt enable of dbi_txe_int
24	cr_dbi_end_en	r/w	1'b1	Interrupt enable of dbi_end_int
23:19	RSVD			
18	rsvd	rsvd	1'b0	
17	rsvd	rsvd	1'b0	
16	cr_dbi_end_clr	w1c	1'b0	Interrupt clear of dbi_end_int
15:11	RSVD			
10	cr_dbi_fer_mask	r/w	1'b1	Interrupt mask of dbi_fer_int
9	cr_dbi_txf_mask	r/w	1'b1	Interrupt mask of dbi_txe_int
8	cr_dbi_end_mask	r/w	1'b1	Interrupt mask of dbi_end_int
7:3	RSVD			
2	dbi_fer_int	r	1'b0	TX/RX FIFO error interrupt, auto-cleared when FIFO overflow/underflow error flag is cleared
1	dbi_txf_int	r	1'b1	TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto-cleared when data is pushed
0	dbi_end_int	r	1'b0	Transfer end interrupt, shared by both Type B and C mode

21.4.10 dbi_yuv_rgb_config_0

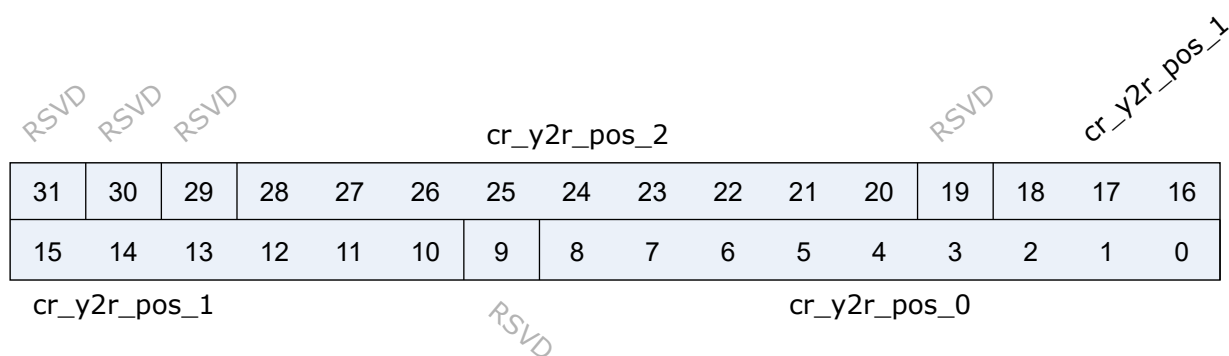
地址: 0x2000a860



位	名称	权限	复位值	描述
31	cr_y2r_en	r/w	1'b1	Display module YUV2RGB enable signal
30:29	RSVD			
28:20	cr_y2r_pre_2	r/w	9'd0	Display module YUV2RGB "pre_offset_2" parameter
19	RSVD			
18:10	cr_y2r_pre_1	r/w	9'd0	Display module YUV2RGB "pre_offset_1" parameter
9	RSVD			
8:0	cr_y2r_pre_0	r/w	9'd0	Display module YUV2RGB "pre_offset_0" parameter

21.4.11 dbi_yuv_rgb_config_1

地址: 0x2000a864



位	名称	权限	复位值	描述
31:29	RSVD			
28:20	cr_y2r_pos_2	r/w	9'd0	Display module YUV2RGB "post_offset_2" parameter

位	名称	权限	复位值	描述
19	RSVD			
18:10	cr_y2r_pos_1	r/w	9'd0	Display module YUV2RGB "post_offset_1" parameter
9	RSVD			
8:0	cr_y2r_pos_0	r/w	9'd0	Display module YUV2RGB "post_offset_0" parameter

21.4.12 dbi_yuv_rgb_config_2

地址: 0x2000a868

cr_y2r_mtx_02_l								cr_y2r_mtx_01							
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_y2r_mtx_01

cr_y2r_mtx_00

位	名称	权限	复位值	描述
31:24	cr_y2r_mtx_02_l	r/w	8'h0	Display module YUV2RGB "matrix_02" parameter lower bits
23:12	cr_y2r_mtx_01	r/w	12'h0	Display module YUV2RGB "matrix_01" parameter
11:0	cr_y2r_mtx_00	r/w	12'h0	Display module YUV2RGB "matrix_00" parameter

21.4.13 dbi_yuv_rgb_config_3

地址: 0x2000a86c

cr_y2r_mtx_12_l

cr_y2r_mtx_11

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_y2r_mtx_10

cr_y2r_mtx_02_u

位	名称	权限	复位值	描述
31:28	cr_y2r_mtx_12_l	r/w	4'h0	Display module YUV2RGB "matrix_12" parameter lower bits
27:16	cr_y2r_mtx_11	r/w	12'h0	Display module YUV2RGB "matrix_11" parameter
15:4	cr_y2r_mtx_10	r/w	12'h0	Display module YUV2RGB "matrix_10" parameter
3:0	cr_y2r_mtx_02_u	r/w	4'h0	Display module YUV2RGB "matrix_02" parameter upper bits

21.4.14 dbi_yuv_rgb_config_4

地址：0x2000a870

cr_y2r_mtx_21

cr_y2r_mtx_20

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_y2r_mtx_20

cr_y2r_mtx_12_u

位	名称	权限	复位值	描述
31:20	cr_y2r_mtx_21	r/w	12'h0	Display module YUV2RGB "matrix_21" parameter
19:8	cr_y2r_mtx_20	r/w	12'h0	Display module YUV2RGB "matrix_20" parameter
7:0	cr_y2r_mtx_12_u	r/w	8'h0	Display module YUV2RGB "matrix_12" parameter upper bits

地址: 0x2000a874

位	名称	权限	复位值	描述
31:12	RSVD			
11:0	cr_y2r_mtx_22	r/w	12'h0	Display module YUV2RGB "matrix_22" parameter

地址: 0x2000a880

fifo_format			fifo_yuv_mode			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			
RSVD			RSVD			RSVD	RSVD	RSVD	RSVD	RSVD	tx_fifo_underflow		tx_fifo_overflow		RSVD	tx_fifo_clr	RSVD	dbi_dma_tx_en

位	名称	权限	复位值	描述
31:29	fifo_format	r/w	3'd0	FIFO pixel data format (see Tab 'FIFO Format' for details) 3'd0: 8'h0, B[7:0], G[7:0], R[7:0] 3'd1: 8'h0, R[7:0], G[7:0], B[7:0] 3'd2: B[7:0], G[7:0], R[7:0], 8'h0 3'd3: R[7:0], G[7:0], B[7:0], 8'h0 3'd4: R_n[7:0], B[7:0], G[7:0], R[7:0] 3'd5: B_n[7:0], R[7:0], G[7:0], B[7:0] 3'd6: 2B[7:3], G[7:2], R[7:3] 3'd7: 2R[7:3], G[7:2], B[7:3] Note: FIFO data format does not affect normal data, which is fixed to LSB sent first Byte3[7:0], Byte2[7:0], Byte1[7:0], Byte0[7:0]
28	fifo_yuv_mode	r/w	1'b0	FIFO data YUV mode R <-> Y G <-> U/Cb B <-> V/Cr
27:6	RSVD			
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	RSVD			
2	tx_fifo_clr	w1c	1'b0	Clear signal of TX FIFO
1	RSVD			
0	dbi_dma_tx_en	r/w	1'b0	Enable signal of dma_tx_req/ack interface

21.4.17 dbi_fifo_config_1

地址：0x2000a884

RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	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
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	tx_fifo_cnt

位	名称	权限	复位值	描述
31:19	RSVD			
18:16	tx_fifo_th	r/w	3'd0	TX FIFO threshold, dma_tx_req will not be asserted if tx_fifo_cnt is less than this value
15:4	RSVD			
3:0	tx_fifo_cnt	r	4'd8	TX FIFO available count

21.4.18 dbi_fifo_wdata

地址: 0x2000a888

dbi_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

dbi_fifo_wdata

位	名称	权限	复位值	描述
31:0	dbi_fifo_wdata	w	x	Pixel data with 8 types of format (determined by fifo_format)

22.1 简介

SD 主机控制器（SD Host Controller，即 SDH）用于控制与 SDIO 或 SD 卡的通信，所有访问都通过标准的控制寄存器进行设置。

22.2 主要特征

- 符合 SD 协会定义的 SD 主机控制器规范
- 适用于 SDIO/SD 存储卡
- 支持 SD 主机控制器标准寄存器设置
- 支持针对高速数据传输的 DMA 操作
- 支持中断
- 支持数据块传输

22.3 功能描述

22.3.1 SDH 总体结构

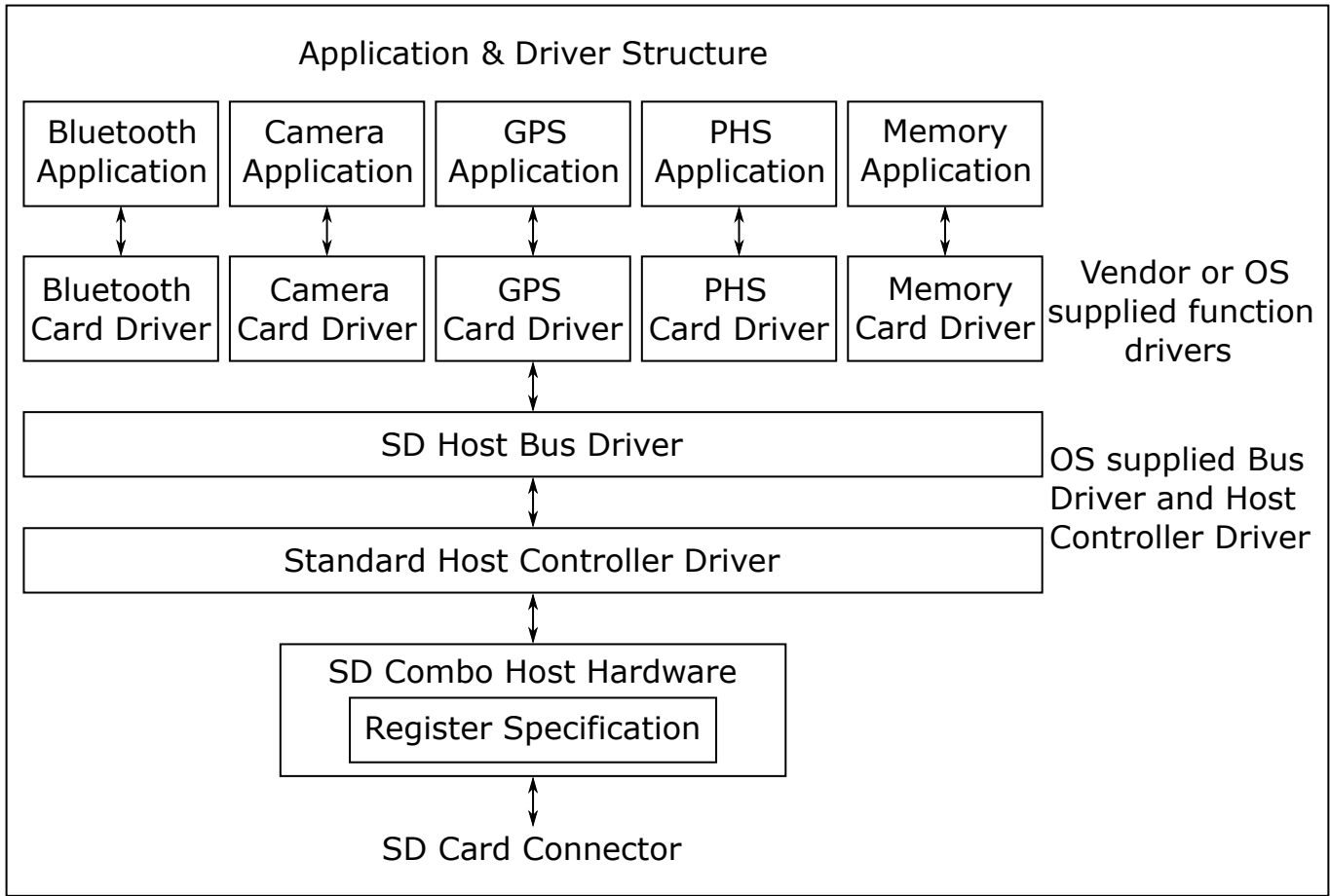


图 22.1: SDH 硬件和驱动结构图

22.3.2 寄存器映射

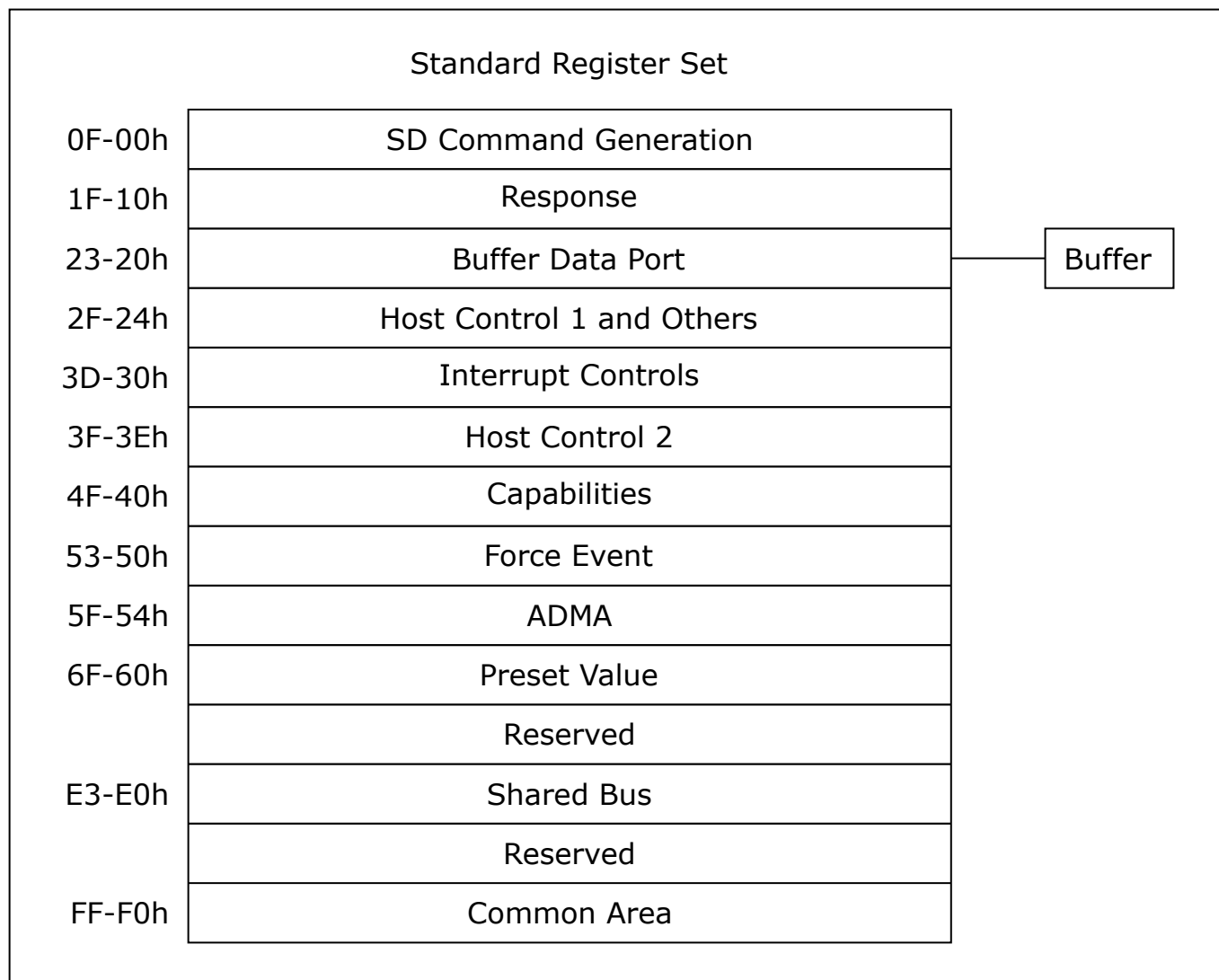


图 22.2: 标准寄存器映射分类图

其中:

- SD Command Generation: 用于生成 SD 命令的参数;
- Response: 来自卡的响应值;
- Buffer Data Port: 内部缓冲区的数据访问端口;
- Host Control 1 and Others: 当前状态、SD 总线控制、主机复位等;
- Interrupt Controls: 中断状态和启用;
- Host Control 2: 供应商特定的主机控制器支持信息;
- Capabilities: 主机控制寄存器的扩展;

- Force Event: 通过软件生成事件的测试寄存器；
- ADMA: 高级 DMA 寄存器；
- Preset Value: 时钟频率选择和驱动强度选择预设；
- Shared Bus: 共享总线系统的设备控制；
- Common Area: 公共信息区。

22.3.3 多卡槽支持

为每个卡槽定义一个标准寄存器集，如果主机控制器有两个卡槽，则需要两个寄存器集。每个卡槽都是独立控制的。这使得支持总线接口电压、总线定时和 SD 时钟频率的组合成为可能。

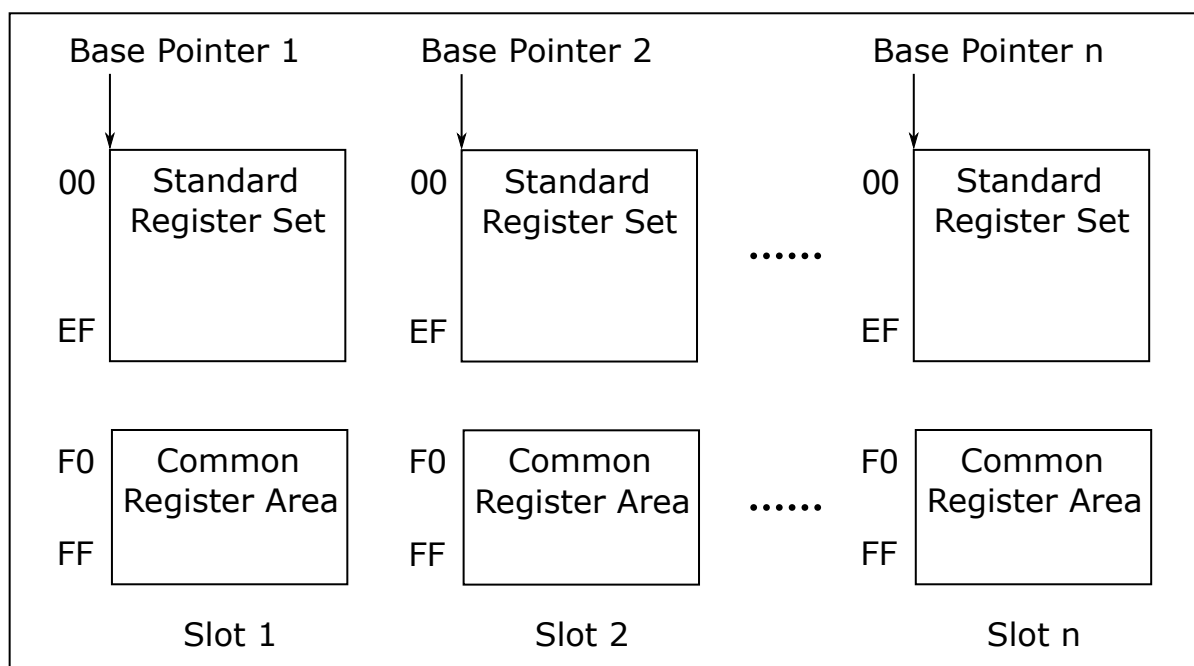


图 22.3: 多卡槽控制器的寄存器映射

上图显示了多卡槽主机控制器的寄存器映射。主机驱动程序应使用 PCI 配置寄存器或供应商特定方法确定卡槽数量和指向每个卡槽标准寄存器集的基指针。从 0F0h 到 0FFh 的偏移量保留给公共寄存器区域，该区域定义卡槽控制和公共状态的信息。公共寄存器区域可从任何卡槽的寄存器集访问。这允许软件独立控制每个卡槽，因为它可以从每个寄存器集中访问卡槽中断状态寄存器和主机控制器版本寄存器。

22.3.4 支持 DMA

主机控制器为主机驱动程序提供“编程 I/O”方法，以便使用缓冲区数据端口寄存器传输数据。可选地，主机控制器实现者可以支持使用 DMA 的数据传输。在使用 DMA 之前，主机驱动程序应确认主机控制器和系统总线均支持 DMA（PCI 总线可支持 DMA）。DMA 应支持单块和多块传输。在 DMA 传输执行期间，主机控制器寄存器应保持可访问性，以发出非 DAT 行命令。无论采用何种系统总线数据传输方法，DMA 传输的结果应相同。主机驱动程序可以通过块间隙控制寄存器中的控制位停止和重新启动 DMA 操作。通过设置在块间隙请求时停止，DMA 操作可以在块间隙处停止。通过设置继续请求，可以重新启动 DMA 操作。如果发生错误，DMA 操作应停止。要中止 DMA 传输，主机驱动程序应通过软件重置寄存器中 DAT 行的软件重置来重置主机控制器，并在执行多块读/写命令时发出 CMD12。

22.3.5 SD 命令生成

	SDMA Command	ADMA Command	CPU Data Transfer	Non-DAT Transfer
SDMA System Address /Argument 2	Yes /No	No /Auto CMD23	No /Auto CMD23	No /No
Block Size	Yes	Yes	Yes	No (Protected)
Block Count	Yes	Yes	Yes	No (Protected)
Argument 1	Yes	Yes	Yes	Yes
Transfer Mode	Yes	Yes	Yes	No (Protected)
Command	Yes	Yes	Yes	Yes

图 22.4: 用于生成 SD 命令的寄存器

上表显示了三种类型的事务所需的寄存器设置（寄存器集中从 000h 到 00Fh 的偏移量）：DMA 生成的传输（SDMA 或 ADMA）、CPU 数据传输（使用“编程 I/O”）和非 DAT 传输。启动事务时，主机驱动程序应按从 000h 到 00Fh 的顺序对这些寄存器进行编程。起始寄存器偏移量可根据事务类型计算。最后写入的偏移量应始终为 00Fh，因为写入命令寄存器的高位字节将触发 SD 命令的发出。在一个数据事务期间，主机驱动程序不应读取 SDMA 系统地址、块大小和块计数寄存器，除非由于值正在更改且不稳定而停止或挂起传输。为了防止在发出命令时使用数据传输破坏寄存器，块大小、块计数和传输模式寄存器应由主机控制器进行写保护，同时当前状态寄存器中将命令禁止（DAT）设置为 1（此信号无法保护 SDMA 系统地址）。当命令禁止（CMD）设置为 1 时，主机驱动程序不得写入参数 1 和命令寄存器。

22.3.6 挂起和恢复机制

通过检查功能寄存器中的挂起/恢复支持，可以确定对挂起/恢复的支持。当 SD 卡接受挂起请求时，主机驱动程序在发出其他 SD 命令之前将信息保存在前 14 字节寄存器（即偏移量 000h-00dh）中。恢复时，主机驱动程序将恢复这些寄存器，然后发出 **Resume** 命令以继续挂起的操作。SDIO 卡在响应恢复命令时设置 **DF**（恢复数据标志）。（由于挂起和恢复命令是 **CMD52** 操作，因此响应数据实际上是 **CCCCR** 中的功能选择寄存器。）如果 **DF** 设置为 0，则表示 SDIO 卡在挂起时无法继续数据传输。该位可用于控制数据传输和中断周期。如果 **Resume Data**（恢复数据）标志设置为 0，则表示正在恢复的事务处于 4 位模式，则不再传输更多数据，并启动中断周期。如果 **DF** 设置为 1，则数据传输将继续。需要注意的是，要使用挂起和恢复功能，SDIO 卡必须支持挂起和恢复命令以及读取等待控制。

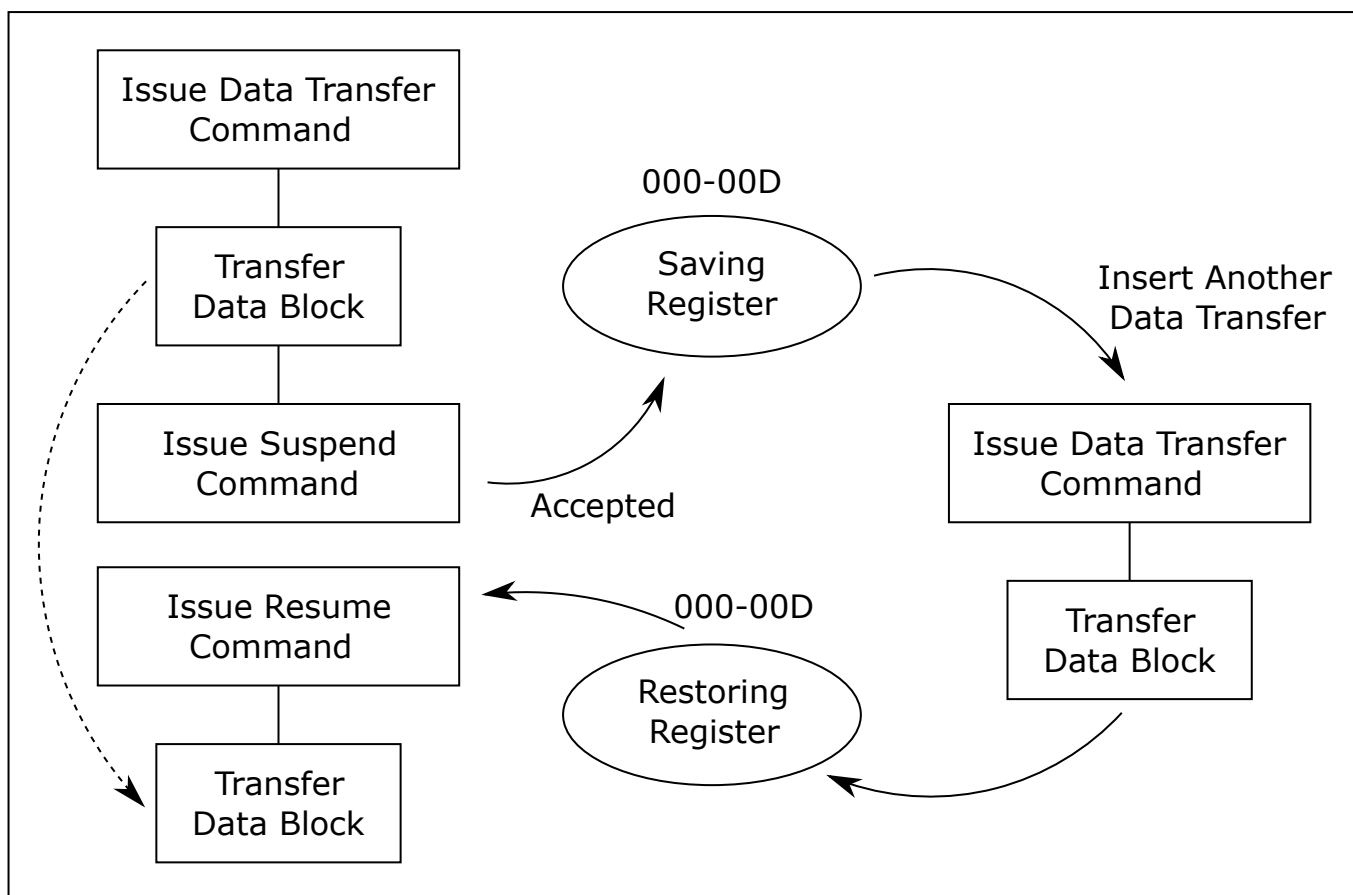


图 22.5: 挂起和恢复机制

22.3.7 缓冲区控制

主机控制器具有用于数据传输的数据缓冲区。主机驱动程序通过 32 位缓冲区数据端口寄存器访问内部缓冲区。下面是一些访问缓冲区的规则。

22.3.7.1 缓冲区指针的控制

在内部，主机控制器维护一个指针来控制数据缓冲区。主机驱动程序无法直接访问指针，每次访问缓冲区数据端口寄存器时，指针都会根据写入缓冲区的数据量递增。为了适应各种系统总线，无论系统总线宽度如何，都应实现该指针（可支持 8 位、16 位、32 位或 64 位系统总线宽度）。

22.3.7.2 确定缓冲块长度

为了能够在突发时传输数据块，主机控制器和 SD 卡缓冲区大小之间的关系非常重要。主机驱动器应为主机控制器和卡使用相同的数据块长度。如果控制器和卡缓冲区大小不同，主机驱动程序应使用较小的值。最大主机控制器缓冲区大小由功能寄存器中的最大块长度字段定义。

22.3.7.3 分割大数据传输

SDIO 命令 **CMD53** 定义根据以下公式限制数据传输的最大数据大小：最大数据大小 = 块大小 x 块计数例如，块大小由缓冲区大小指定，块计数的最大值为 **512**（9 位计数），如 **CMD53** 的命令参数中所指定。在最坏的情况下，如果卡只有一个 1 字节的缓冲区，则最多可以使用 **CMD53** 传输 **512** 字节的数据（块大小 = 1，块计数 = 512）。如果卡不支持多块模式，在这种情况下只能传输一个字节。如果应用程序或卡驱动程序希望传输更大尺寸的数据，主机驱动程序应将大数据划分为多个 **CMD53** 块。

22.3.8 中断控制寄存器之间的关系

主机控制器实现许多中断源。中断源可以作为中断或系统唤醒信号启用。如果正常中断状态启用或错误中断状态启用寄存器中的中断源对应位为 1，且中断变为激活状态，则其激活状态被锁定，并可供正常中断状态寄存器或错误中断状态寄存器中的主机驱动程序使用。当中断状态启用被清除时，中断状态应被清除。如果在正常中断信号启用寄存器或错误中断信号启用寄存器中也设置了相应位，则在中断状态寄存器中设置位的中断源应断言系统中断信号。一旦发出信号，大多数中断将通过向中断状态寄存器中的相关位写入 1 来清除。但是，卡中断应由卡驱动程序清除。如果生成卡中断，主机驱动程序可以清除卡中断状态启用，以在卡驱动程序处理卡中断时禁用卡中断。清除所有中断源后，主机驱动程序再次将其设置为启用另一个卡中断。禁用卡中断状态启用可避免在处理中断服务期间产生多个中断。唤醒控制寄存器允许对卡中断、卡插入或卡移除状态更改进行配置，以生成系统唤醒信号。这些中断的启用或屏蔽独立于正常中断信号启用寄存器。可从正常中断状态寄存器读取唤醒事件的类型。中断信号和唤醒信号为逻辑或，应从插槽中断状态寄存器中读取。

22.3.9 硬件框图和定时部分

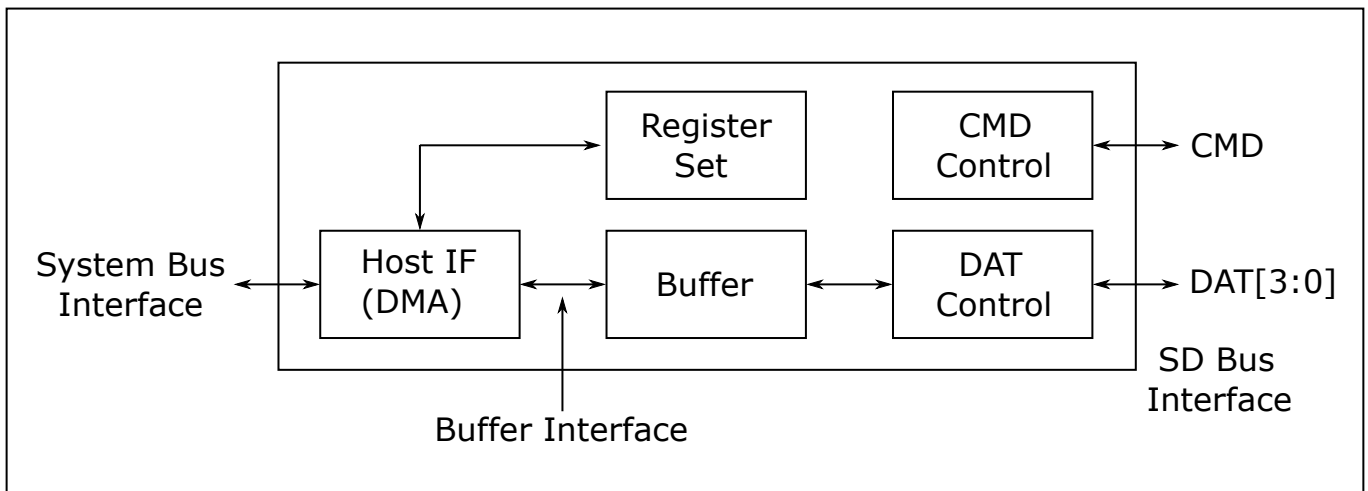


图 22.6: 主机控制器框图

主机控制器有两个总线接口，系统总线接口和 SD 总线接口。主机控制器假定这些接口是异步的（即，在不同的时钟频率下工作）。主机驱动程序处于系统总线时间（因为它是由主机控制器 CPU 在其系统时钟上执行的软件）。SD 卡在 SD 总线时间上（也就是说，其操作由 SDCLK 同步）。主机控制器应同步信号，以便在这些接口之间进行通信。数据块应在缓冲模块上同步。所有状态寄存器应通过系统时钟同步，并在输出至系统接口期间保持同步。触发 SD 总线事务的控制寄存器应由 SDCLK 同步。因此，在两个接口之间传播信号时将存在定时延迟。这意味着主机驱动程序无法实时控制 SD 总线，需要依靠主机控制器根据寄存器设置控制 SD 总线。缓冲区接口启用内部读写缓冲区。传输完成中断状态指示 DMA 和非 DMA 传输的读/写传输完成。但是，读和写之间的计时是不同的。读取传输应在所有有效数据传输到主机系统并准备好供主机驱动程序访问后完成。写入传输应在所有有效数据传输到 SD 卡且忙碌状态结束后完成。

22.3.10 自动 CMD12

SD 内存的多个块传输需要 CMD12 停止事务。最后一次数据块传输完成时，主机控制器自动发出 CMD12。主机控制器的此功能称为自动 CMD12。发出多块传输命令时，主机驱动程序应在传输模式寄存器中设置 Auto CMD12 Enable（自动 CMD12 启用）。应通过主机控制器中的硬件完成与最后一个数据块的自动 CMD12 定时同步。在多次块传输期间，可以发出不使用 DAT 行的命令。这些命令使用 CMD_wo_DAT 符号表示。为了防止 DAT 线路命令和 CMD_wo_DAT 命令发生冲突，主机控制器应仲裁在 SD 总线上发出每个命令的时间。因此，在主机驱动程序写入命令寄存器后，可能不会立即发出命令。该命令可以在自动 CMD12 之前或之后发出，具体取决于时间。为了能够区分 DAT line 和 CMD_wo_DAT 命令的响应，可以从响应寄存器的前四个字节（在标准寄存器集中的偏移量 01Ch 处）确定自动 CMD12 响应。如果检测到与自动 CMD12 相关的错误，主机控制器应发出自动 CMD 错误中断。主机驱动程序可以通过读取自动 CMD 错误状态寄存器来检查自动 CMD12 错误状态（命令索引/结束位/CRC/超时错误）。如果未执行自动 CMD12，则主机驱动程序需要从 CMD_wo_DAT 错误中恢复，并发出 CMD12 以停止多块传输。如果未执行 CMD_wo_DAT，则主机驱动程序可以在从自动 CMD12 错误恢复后再次发出该命令。在 UHS 模式 SDR104 中，主机驱动程序应使用自动 CMD23 停止多块读/写操作，而不是使用自动 CMD12。在其他总线速度模式下，如果卡支持 CMD23，则主机驱动程序应使用自动 CMD23，而不是使用 CMD12。

22.3.11 控制 SDCLK

下表显示了 SDCLK 如何由电源控制寄存器中的 SD 总线电源和时钟控制寄存器中的 SD 时钟启用控制：

SD Bus Power	SD Clock Enable	State of SDCLK
Change 0 to 1	0	Drive Low
	1	Start Clock With Specified Period Of High
Change 1 to 0	0	Drive Low
	1	Drive Low Immediately
0	Don't Care	Drive Low
1	Change 0 to 1	Start Clock With Specified Period Of High
	Change 1 to 0	Maintains Period Of High And Then Stops Clock And Drive Low

图 22.7: 通过 SD 总线电源和 SD 时钟启用控制 SDCLK

SDCLK 的时钟周期由时钟控制寄存器中的 SDCLK 频率选择和功能寄存器中的 SD 时钟的基本时钟频率指定。由于 SD 卡可以同时使用两个时钟边缘，因此 SD 时钟的占空比应为平均 50%（分散在 45-55% 范围内），高电平周期应为时钟周期的一半。SDCLK 的振荡从驱动规定的高电压周期开始。当 SDCLK 被 SD 时钟启用停止时，主机控制器应在高驱动周期后停止 SDCLK，以维持时钟占空比。当 SDCLK 被 SD 总线电源停止时，主机控制器应立即停止 SDCLK（驱动器低），并且应清除 SD 时钟启用。

22.3.12 高级 DMA

SD 主机控制器标准规范 2.00 版定义了 ADMA（高级 DMA）传输算法。SD 主机控制器标准规范 1.00 版中定义的 DMA 算法称为 SDMA（单操作 DMA）。SDMA 的缺点是，在每个页面边界产生的 DMA 中断会干扰 CPU 重新编程新的系统地址。这种 SDMA 算法通过在每个页面边界处中断而形成性能瓶颈。ADMA 采用分散-聚集 DMA 算法，因此具有更高的数据传输速度。在执行 ADMA 之前，主机驱动程序可以将系统内存和 SD 卡之间的数据传输列表编程到描述符表中。它使 ADMA 能够在不中断主机驱动程序的情况下运行。此外，ADMA 不仅可以支持 32 位系统内存寻址，还可以支持 64 位系统内存寻址。32 位系统内存寻址使用 64 位地址寄存器的低 32 位字段。ADMA 有两种类型：ADMA1 和 ADMA2。ADMA1 仅支持系统内存中 4KB 对齐数据的数据传输。ADMA2 改进了该限制，使任何位置和大小数据都可以在系统内存中传输。描述符表的格式在它们之间是不同的。本文档中使用术语“ADMA”时，指 ADMA2。

22.3.12.1 ADMA2 的框图

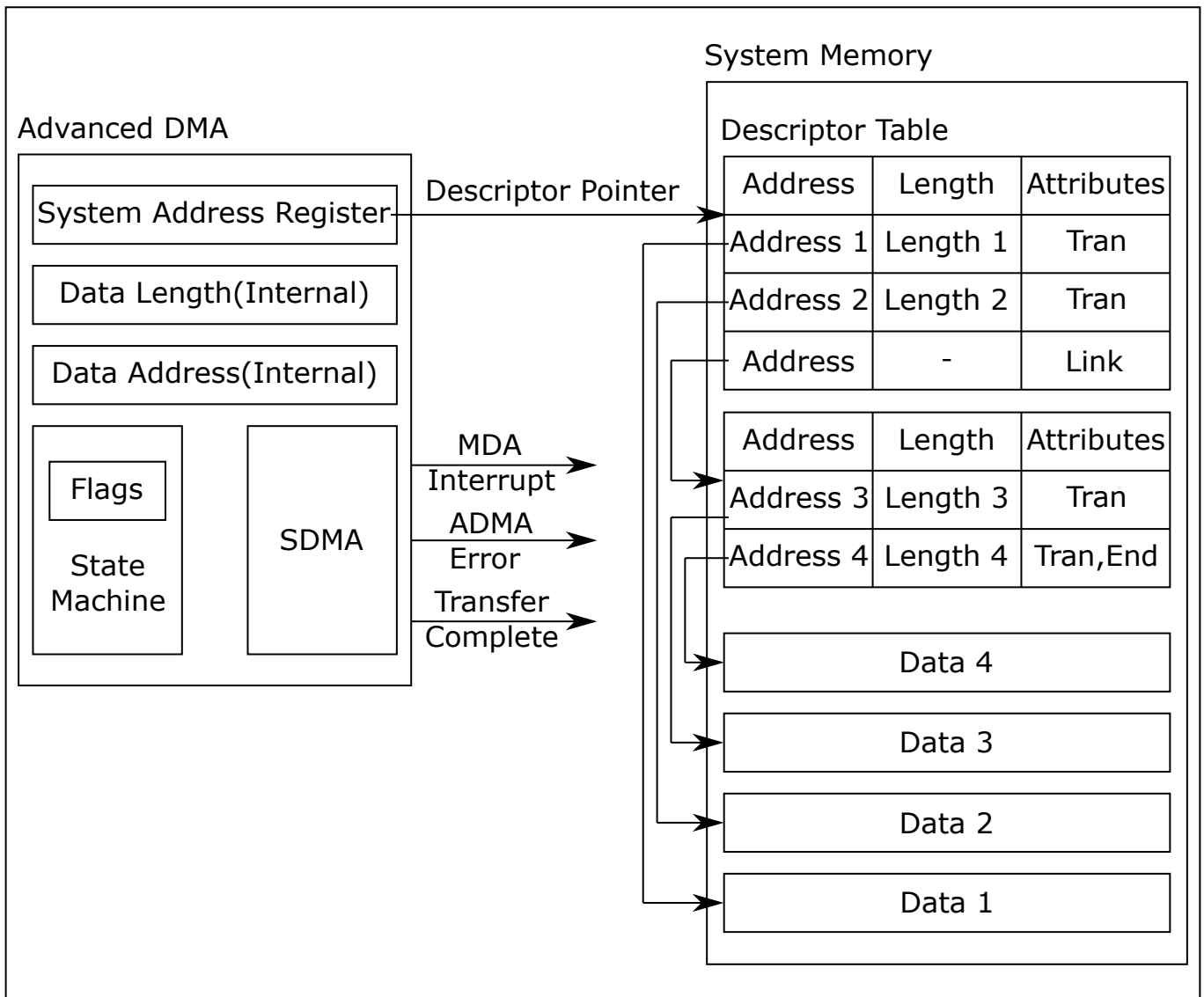


图 22.8: ADMA2 的框图

描述符表由主机驱动程序在系统内存中创建。32 位地址描述符表用于 32 位寻址的系统，64 位地址描述符表用于 64 位寻址的系统。每个描述符行（一个可执行单元）由地址、长度和属性字段组成。该属性指定描述符行的操作。ADMA2 包括 SDMA、状态机和寄存器电路。ADMA2 不使用 32 位 SDMA 系统地址寄存器（偏移量 0），但使用 64 位高级 DMA 系统地址寄存器（偏移量 058h）作为描述符指针。写入命令寄存器会触发关闭 ADMA2 传输。ADMA2 获取一个描述符行并执行它。重复此过程，直到找到描述符的末尾（属性中的 end=1）。

22.3.12.2 数据地址和数据长度要求

对描述符进行编程有 3 个要求:

- 地址的最小单位为 4 字节;
- 每个描述符行的最大数据长度小于 64KB;
- 总长度 = 长度 1 + 长度 2 + 长度 3 + ... + 长度 n = 块大小的倍数。

若描述符的总长度不是块大小的倍数, 则可能不会终止 ADMA2 传输。在这种情况下, 应通过数据超时中止传输。块计数寄存器限制最大 65535 块传输。如果 ADMA2 操作小于或等于 65535 块传输, 则可以使用块计数寄存器。在这种情况下, 描述符表的总长度应等于乘以块大小和块计数。如果 ADMA2 操作超过 65535 块传输, 则应通过在传输模式寄存器中将 0 设置为块计数启用来禁用块计数寄存器。在这种情况下, 数据传输的长度不是由块计数指定的, 而是由描述符表指定的。因此, 检测 SD 总线上最后一个块的定时可能不同, 并且它影响当前状态寄存器中的读传输活动、写传输活动和 DAT 线活动的控制。在读取操作的情况下, 可能会读取多个块。如果读取操作是针对最后一块内存区域, 则主机驱动程序应忽略超出范围的错误。

22.3.12.3 描述符表

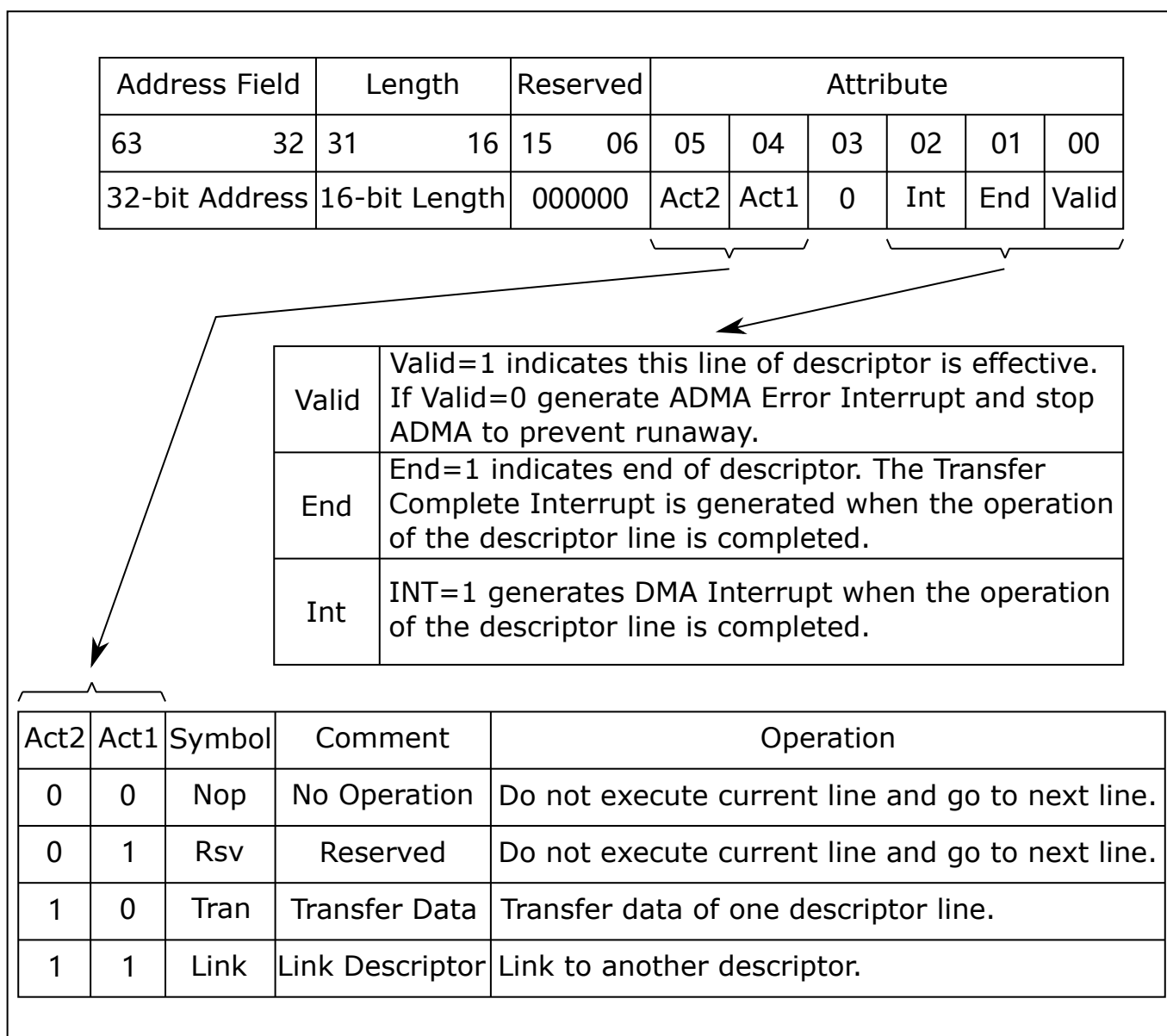


图 22.9: 32 位地址描述符表

上图显示了 32 位地址描述符表的定义。一个描述符行消耗 64 位（8 字节）内存空间。属性用于控制描述符。指定了 3 个动作符号：“Nop”操作跳过当前描述符行并获取下一行；“Tran”操作传输由地址和长度字段指定的数据；“Link”操作用于连接分开的两个描述符。链接的地址字段指向下一个描述符表。Act2=0 和 Act1=1 的组合被保留并定义为与 Nop 相同的操作。控制器的未来版本可能会使用此字段并重新定义新操作。32 位地址存储在 64 位地址寄存器的低 32 位中。对于 32 位地址描述符表，地址字段应设置在 32 位边界上（低 2 位始终设置为 0）。

22.3.12.4 ADMA2 状态

下图是 ADMA2 的状态图，定义了 4 个状态，获取描述符状态、更改地址状态、传输数据状态和停止 ADMA 状态：

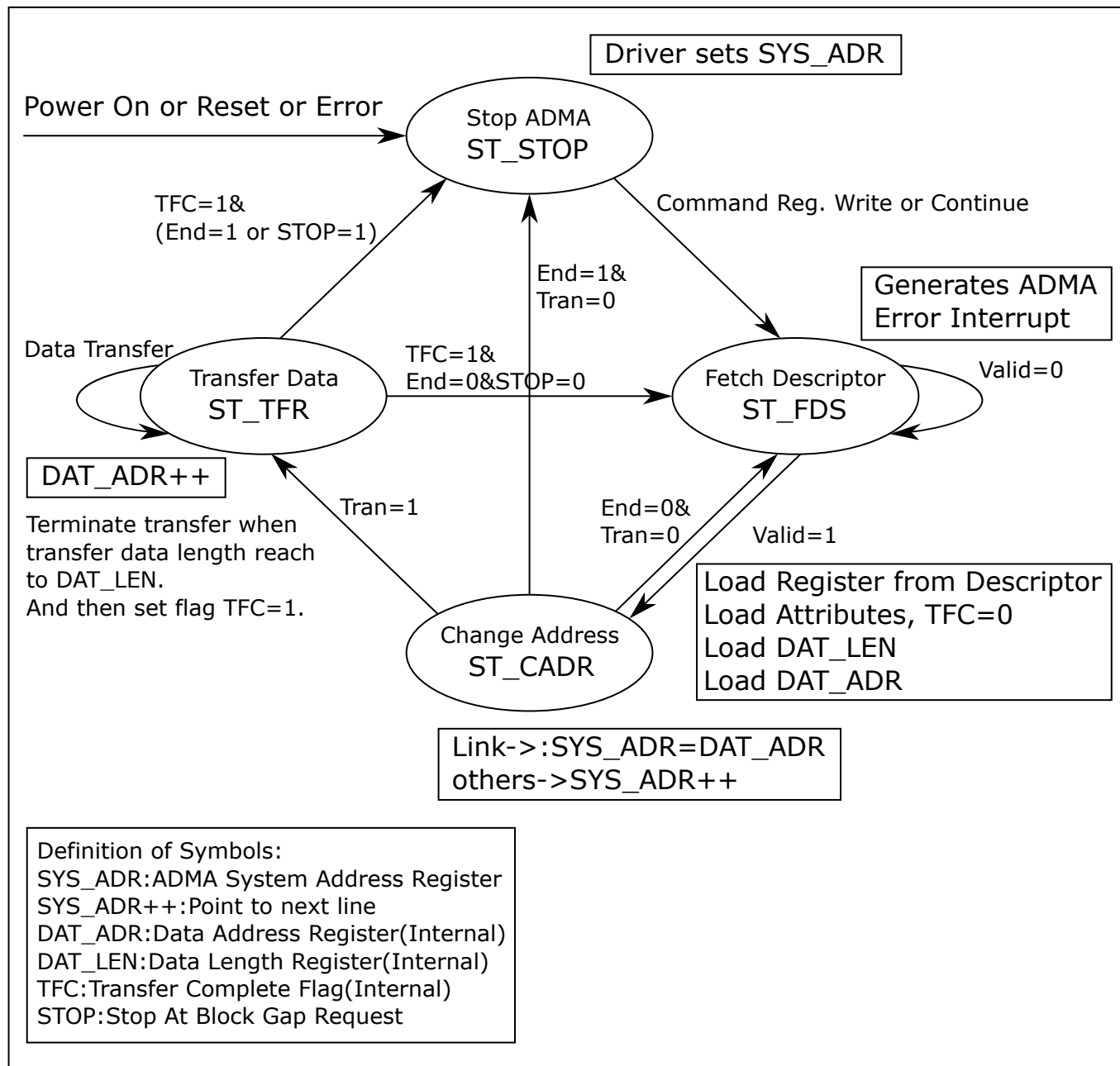


图 22.10: ADMA2 的状态图

其中：

- ST_FDS (获取描述符):** ADMA2 获取描述符并在内部寄存器中设置参数，接下来转到 ST_CADR 状态；
- ST_CADR (更改地址):** 链接操作将另一个描述符地址加载到 ADMA 系统地址寄存器。在其他操作中，ADMA 系统地址寄存器递增到下一个描述符行。如果 End=0，则转到 ST_TFR 状态。即使出现一些错误，ADMA2 也不得在此状态下停止；

- **ST_TFR** (传输数据): 在系统内存和 SD 卡之间执行一条描述符行的数据传输。如果数据传输继续 (End=0), 则转到 **ST_FDS** 状态。如果数据传输完成, 则转到 **ST_STOP** 状态;
- **ST_STOP** (停止 DMA): 在以下情况下, **ADMA2** 将保持此状态, 一是上电复位或软件复位后, 一是所有描述符数据传输都已完成。如果通过写入命令寄存器启动新的 **ADMA2** 操作, 转至 **ST_FDS** 状态。

ADMA2 不支持挂起/恢复功能, 但可以使用停止和继续。当在 **ADMA2** 操作期间设置块间隙控制寄存器中的块间隙停止请求时, 当 **ADMA2** 在块间隙停止时, 将生成块间隙事件中断。主机控制器应使用读取等待或停止 SD 时钟停止 **ADMA2** 读取操作。停止 **ADMA2** 时, 无法发出任何 SD 命令。**ADMA2** 传输期间发生错误可能会停止 **ADMA2** 操作并生成 **ADMA** 错误中断。**ADMA** 错误状态寄存器中的 **ADMA** 错误状态字段保存 **ADMA2** 已停止的状态。主机驱动程序可以通过以下方法识别错误描述符位置: 如果 **ADMA** 在 **ST_FDS** 状态下停止, 则 **ADMA** 系统地址寄存器将指向错误描述符行。如果 **ADMA** 在 **ST_TFR** 或 **ST_STOP** 状态下停止, **ADMA** 系统地址寄存器将指向错误描述符行的下一个位置。因此, **ADMA2** 不得在 **ST_CADR** 状态停止。

22.3.13 测试寄存器

测试寄存器是为测试目的而定义的。当很难有意生成某些中断时, 可以使用此功能手动生成这些中断以进行驱动程序调试。为此, 定义了用于控制错误中断状态和自动 **CMD** 错误状态的强制事件寄存器。有意控制插卡和拔卡也很困难, 主机控制 1 寄存器中的卡检测信号选择和卡检测测试级别使能够人为控制当前状态寄存器所插入的卡, 并在正常中断状态寄存器中生成卡插入和卡移除中断。

22.3.14 块计数

块计数命令 (**CMD23**) 提供了停止多块操作的无计时方法。在 **CMD23** 的参数中设置块计数, 以指定其后的 **CMD18** 或 **CMD25** 的传输长度。自动 **CMD23** 是在发送 **CMD18** 或 **CMD25** 之前自动发出 **CMD23** 的功能。此功能的目的是通过删除 **CMD23** 的中断服务, 避免在内存访问期间性能下降。偏移量 008h 参数 1 寄存器用于 **CMD18** 或 **CMD25**。然后为 **CMD23** 的参数 2 寄存器分配偏移量 000h。主机控制器不使用参数 2 寄存器计算数据传输长度。主机端数据传输操作的数据长度有两种情况: 非 **ADMA** 和 **ADMA**。**ADMA** 描述符的总长度是指 **ADMA** 描述符每行的 16 位数据长度之和。应为 **ADMA** 禁用块计数启用。需要注意的是, 主机控制器的总数据传输长度应等于卡的传输长度。

22.3.15 采样时钟调谐

在 **UHS-I** 模式下, SD 总线可以在高时钟频率模式下运行, 然后 **CMD** 和 **DAT[3:0]** 线路上的卡的数据窗口变小。数据窗口的位置将根据卡和主机系统的实施情况而变化。因此, 当通过执行调谐程序并调整采样时钟来支持 **SDR104** 或 **SDR50** (如果在能力寄存器中将 **SDR50** 的使用调谐设置为 1) 时, 主机控制器应支持调谐电路。主机控制 2 寄存器中的执行调谐和采样时钟选择用于控制调谐电路。

22.3.16 SD 主机标准寄存器

22.3.16.1 SD 主机控制寄存器映射

下表总结了标准 SD 主机控制器寄存器集。主机驱动程序需要确定由主机系统特定方法设置的寄存器的基址。寄存器集的大小为 256 字节。对于多个卡槽控制器，每个卡槽分配一个寄存器集，但偏移量 0F0h-0FFh 处的寄存器分配为公共区域，这些寄存器包含每个卡槽寄存器集的相同值。

Offset	15-08 bit	07-00 bit	Offset	15-08 bit	07-00 bit
002h	SDMA System Address(High),Argument 2(High)		000h	SDMA System Address(Low),Argument 2(Low)	
006h	Block Count		004h	Block Size	
00Ah	Argument 1(High)		008h	Argument 1(Low)	
00Eh	Command		00Ch	Transfer Mode	
012h	Response1		010h	Response0	
016h	Response3		014h	Response2	
01Ah	Response5		018h	Response4	
01Eh	Response7		01Ch	Response6	
022h	Buffer Data Port1		020h	Buffer Data Port0	
026h	Present State		024h	Present State	
02Ah	Wakeup Control	Block Gap Control	028h	Power Control	Host Control 1
02Eh	Software Reset	Timeout Control	02Ch	Clock Control	
032h	Error Interrupt Status		030h	Normal Interrupt Status	
036h	Error Interrupt Status Enable		034h	Normal Interrupt Status Enable	
03Ah	Error Interrupt Signal Enable		038h	Normal Interrupt Signal Enable	
03Eh	Host Control 2		03Ch	Auto CMD Error Status	
042h	Capabilities		040h	Capabilities	
046h	Capabilities		044h	Capabilities	
04Ah	Maximum Current Capabilities		048h	Maximum Current Capabilities	
04Eh	Maximum Current Capabilities(Reserved)		04Ch	Maximum Current Capabilities(Reserved)	
052h	Force Event for Error Interrupt Status		050h	Force Event for Auto CMD Error Status	
056h	---		054h	---	ADMA Error Status
05Ah	ADMA System Address [31:16]		058h	ADMA System Address [15:00]	
05Eh	ADMA System Address [63:48]		05Ch	ADMA System Address [47:32]	
062h	Preset Value		060h	Preset Value	
066h	Preset Value		064h	Preset Value	
06Ah	Preset Value		068h	Preset Value	
06Eh	Preset Value		06Ch	Preset Value	
---	---		---	---	
0E2h	Shared Bus Control(High)		0E0h	Shared Bus Control(Low)	
---	---		---	---	
0F2h	---		0F0h	---	
---	---		---	---	
0FEh	Host Controller Version		0FCh	Slot Interrupt Status	

图 22.11: SD 主机控制寄存器映射

22.3.16.2 配置寄存器类型

配置寄存器字段被分配下表描述的一个属性：

Register Attribute	Description
RO	Read-only register: Register bits are read-only and cannot be altered by software or any reset operation. Writes to these bits are ignored.
ROC	Read-only status: These bits are initialized to zero at reset. Writes to these bits are ignored.
RW	Read-Write register: Register bits are read-write and may be either set or cleared by software to the desired state.
RW1C	Read-only status, Write-1-to-clear status: Register bits indicate status when read, a set bit indicating a status event may be cleared by writing a 1. Writing a 0 to RW1C bits has no effect.
RWAC	Read-Write, automatic clear register: The Host Driver requests a Host Controller operation by setting the bit. The Host Controllers shall clear the bit automatically when the operation of complete. Writing a 0 to RWAC bits has no effect.
HwInit	Hardware Initialized: Register bits are initialized by firmware or hardware mechanisms such as pin strapping or serial EEPROM. Bits are read-only after initialization, and writes to these bits are ignored.
Rsvd	Reserved. These bits are initialized to zero, and writes to them are ignored.
WO	Write-only register. It is not physically implemented register. Rather, it is an address at which registers can be written.

图 22.12: 寄存器（和寄存器位字段）类型

22.3.16.3 寄存器初始值

主机控制器在上电复位时将所有寄存器设置为其初始值，所有其他寄存器的默认值应为设置为零的所有位。能力寄存器和最大当前能力寄存器的值取决于主机控制器，主机控制器版本寄存器的值也取决于主机控制器。

22.3.16.4 寄存器的保留位

“保留”表示该位可定义供将来使用，当前设置为 0。这些位应写为 0。

23.1 简介

安全数字输入输出（Secure Digital Input and Output, SDIO）是在 SD 存储卡接口基础上发展起来的一种外设接口。SDIO 卡兼容 SD 存储卡，目的是提供高速数据 I/O 与低功耗的移动电子设备。

23.2 主要特征

- 针对便携式和固定式应用
- 最小或不需要修改 SD 物理总线
- 内存驱动软件的最小更改
- 可用于特殊应用的扩展物理形状因子
- 即插即用
- 多功能支持，包括多个 I/O、组合 I/O 和内存
- 一张卡最多支持 7 个 I/O 功能和 1 个内存
- 允许卡中断主机

23.3 功能描述

23.3.1 SDIO 信号定义

23.3.1.1 SDIO 卡类型

SDIO 卡分为两种类型，全速 SDIO 卡和低速 SDIO 卡。在 0~25MHz 的全时钟范围内，全速卡支持 SPI、1 位 SD 和 4 位 SD 传输模式。全速 SDIO 卡的数据传输速率超过 100Mb/秒（10MB/秒）。低速 SDIO 卡只需要 SPI 和 1 位 SD 传输模式，支持 0~400KHz 的全时钟范围。低速卡的预期用途是用最少的硬件支持低速 I/O 能力，低速卡支持调制解调器、条码扫描器、GPS 接收器等功能。

23.3.1.2 SDIO 卡模式

有 3 种为 SD 存储卡定义的信号模式，也适用于 SDIO 卡：**SPI 模式**：在这种模式下，引脚 8 被用作中断引脚，所有其他引脚和信令协议都与 SD 物理规范相同。**1 位 SD 数据传输模式**：在这种模式下，数据只在 DAT[0] 引脚上传输。引脚 8 被用作中断引脚，所有其他引脚和信令协议都与 SD 内存规范相同。**4 位 SD 数据传输模式**：在这种模式下，数据在所有 4 个数据引脚 DAT[3:0] 上传输。此模式下中断引脚不可独占使用，因为它被用作数据传输线。因此，如果需要中断功能，则需要特殊的定时来提供中断。**4 位 SD 模式**提供最快的数据传输可能，最高可达 100Mb/秒。

23.3.1.3 信号引脚

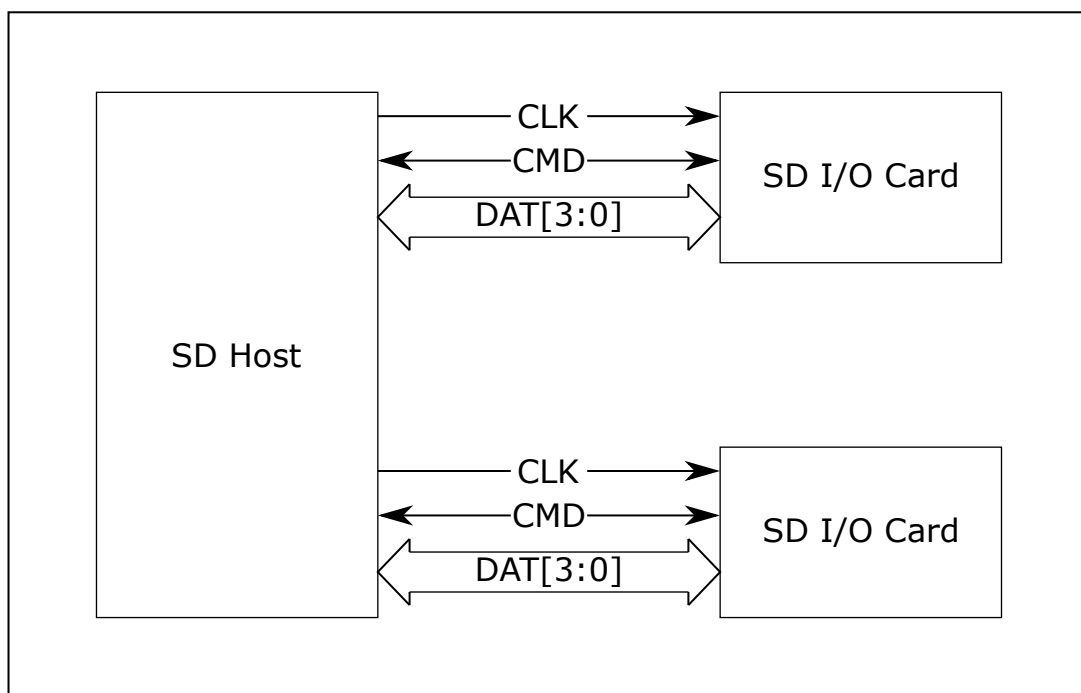


图 23.1: 2 个 4 位 SDIO 卡信号连接示意图

23.3.2 SDIO 卡初始化

23.3.2.1 I/O 卡初始化差异

SDIO 规范的一个要求是，当插入 SDIO 卡时，不应导致非 I/O 感知主机失败。为了防止在非 I/O 感知的主机中操作 I/O 功能，需要更改 SD 卡识别模式流程图。添加了一个新的命令（IO_SEND_OP_COND, CMD5）来替换 ACMD41，通过 I/O 感知主机进行 SDIO 初始化。复位或上电后，卡上的所有 I/O 功能都被禁用，当 CS 为低时，除 CMD5 或 CMD0 外，卡的 I/O 部分不会执行任何操作。如果有 SD 内存安装在卡上，该内存应正常响应所有正常的强制内存命令。仅 I/O 卡不得响应 ACMD41，因此最初显示为 MMC 卡。仅 I/O 卡也不得响应用于初始化 MMC 卡的 CMD1，并显示为非响应卡。然后主机放弃并禁用该卡。因此，非感知主机不接收来自仅 I/O 卡的响应，并强制其进入非活动状态。下图显示了具有非 I/O 感知主机的 I/O 卡的操作，实线为实际路径，而虚线未执行：

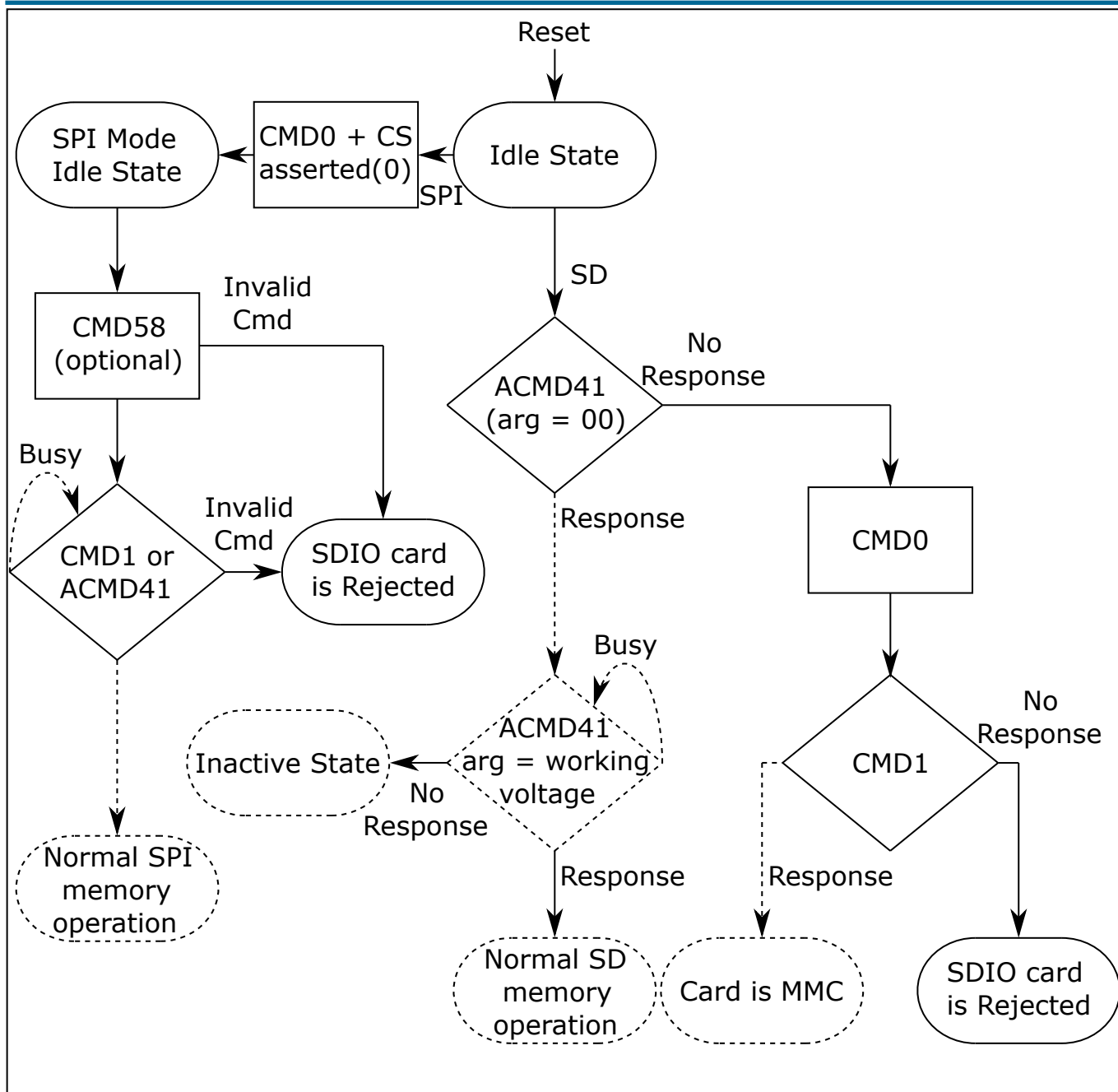


图 23.2: SDIO 对非 I/O 感知初始化的响应

23.3.2.2 IO_SEND_OP_COND 命令（CMD5）

IO_SEND_OP_COND 命令（CMD5）对 SDIO 卡的功能类似于对 SD 存储卡的 ACMD41 的操作，用于查询 I/O 卡所需要的电压范围。CMD5 的正常响应是 SD 或 SPI 格式的 R4。CMD5 的格式如下图所示：

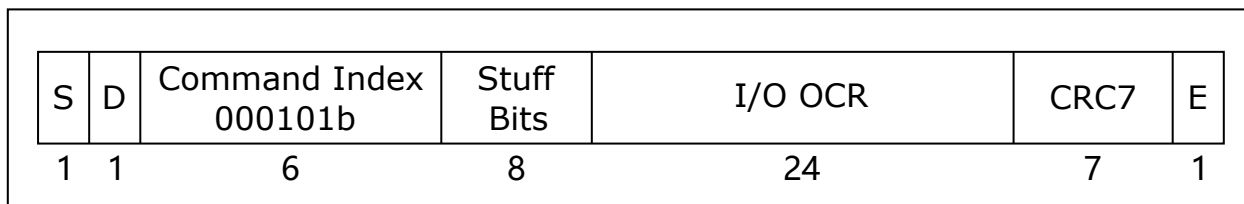


图 23.3: IO_SEND_OP_COND 命令

其中：

- S: 起始位。总是 0；
- D: 方向。总是 1，表示从主机到卡的传输；
- Command Index: 使用值 000101b 来标识 CMD5 命令；
- Stuff Bits: 不使用，应设置为 0；
- I/O OCR: 操作条件寄存器。VDD 支持的最小值和最大值；
- CRC7: 7 位 CRC 数据；
- E: 结束位。总是 1。

23.3.2.3 IO_SEND_OP_COND 响应（R4）

接收到 CMD5 的 SDIO 卡应响应一个唯一的 SDIO 响应 R4。R4 在 SD 和 SPI 模式下的格式分别如下图所示：

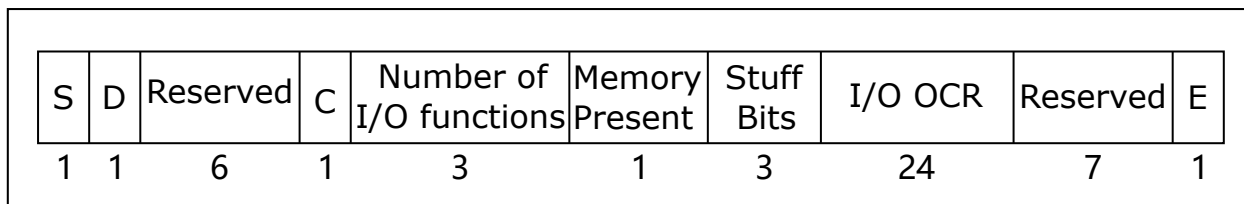


图 23.4: SD 模式下的响应 R4

Modified R1	C	Number of I/O functions	Memory Present	Stuff Bits	I/O OCR
8	1	3	1	3	24

图 23.5: SPI 模式下的响应 R4

其中:

- S: 起始位。总是 0;
- D: 方向。总是 0, 表示从卡到主机的传输;
- Reserved: 为将来使用而保留的位。这些位应设置为 1;
- C: 如果卡在初始化完后准备好了操作, 则设置为 1;
- Stuff Bits: 不使用, 应设置为 0;
- I/O OCR: 操作条件寄存器。VDD 支持的最小值和最大值;
- E: 结束位。总是 1。
- Memory Present: 如果卡还包含 SD 内存, 则设置为 1。如果卡仅为 I/O, 则设置为 0;
- Number of I/O Functions: 指示此卡支持的 I/O 功能总数, 范围是 0~7。需要注意的是, 此计数不包括功能 0 的所有 I/O 卡上的公共区域。I/O 功能应从功能 1 开始按顺序执行;
- Modified R1: SD 物理规范中描述的 SPI R1 响应字节针对 I/O 的改进如下;

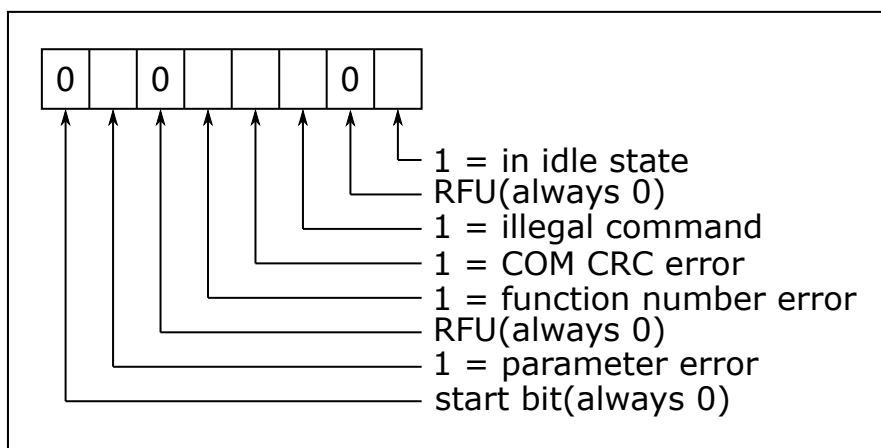


图 23.6: 改进的 R1 响应

23.3.2.4 组合卡的特殊初始化注意事项

主机在初始化组合卡时必须注意一些特殊情况（SDIO 加上同一卡上的 SD 内存）。这是因为组合卡的实现实际上可以在同一个包中使用两个单独的控制器（内存和 I/O），并共享相同的总线线路。主机必须检测并正确配置组合卡的两个部分（控制器），以防止 SDIO 和 SD 内存控制器之间发生冲突。这些问题是由于两个控制器对复位（硬或软）的响应不同而引起的。另一个问题是内存控制器中存在的 RCA（相对卡地址）的值。上述注意事项仅适用于 SD 1 位和 SD 4 位模式。在 SPI 模式下，卡选择/取消选择使用硬件 CS 线而不是 RCA 完成。

23.3.3 与 SD 内存规范的差异

23.3.3.1 SDIO 命令列表

使用 SD 总线接口时 SD 内存和 SDIO 卡接受的命令列表如下图所示：

Supported Commands	Abbreviation	SDMEM System	SDIO System	Comments
CMD0	GO_IDLE_STATE	Mandatory	Mandatory	Used to change from SD to SPI mode
CMD2	ALL_SEND_CID	Mandatory		CID not supported by SDIO
CMD3	SEND_RELATIVE_ADDR	Mandatory	Mandatory	
CMD4	SET_DSR	Optional		DSR not supported by SDIO
CMD5	IO_SEND_OP_COND		Mandatory	
CMD6	SWITCH_FUNC	Mandatory	Mandatory	
CMD7	SELECT/DESELECT_CARD	Mandatory	Mandatory	
CMD9	SEND_CSD	Mandatory		CSD not supported by SDIO
CMD10	SEND_CID	Mandatory		CID not supported by SDIO
CMD12	STOP_TRANSMISSION	Mandatory		
CMD13	SEND_STATUS	Mandatory		Card Status includes only SDMEM information
CMD15	GO_INACTIVE_STATE	Mandatory	Mandatory	
CMD16	SET_BLOCKLEN	Mandatory		
CMD17	READ_SINGLE_BLOCK	Mandatory		
CMD18	READ_MULTIPLE_BLOCK	Mandatory		
CMD24	WRITE_BLOCK	Mandatory		
CMD25	WRITE_MULTIPLE_BLOCK	Mandatory		
CMD27	PROGRAM_CSD	Mandatory		CSD not supported by SDIO
CMD28	SET_WRITE_PROT	Optional		
CMD29	CLR_WRITE_PROT	Optional		
CMD30	SEND_WRITE_PROT	Optional		
CMD32	ERASE_WR_BLK_START	Mandatory		
CMD33	ERASE_WR_BLK_END	Mandatory		
CMD38	ERASE	Mandatory		
CMD42	LOCK_UNLOCK	Optional		
CMD52	IO_RW_DIRECT		Mandatory	
CMD53	IO_RW_EXTENDED		Mandatory	Block mode is optional
CMD55	APP_CMD	Mandatory		
CMD56	GEN_CMD	Mandatory		
ACMD6	SET_BUS_WIDTH	Mandatory		
ACMD13	SD_STATUS	Mandatory		
ACMD22	SEND_NUM_WR_BLOCKS	Mandatory		
ACMD23	SET_WR_BLK_ERASE_COUNT	Mandatory		
ACMD41	SD_APP_OP_COND	Mandatory		
ACMD42	SET_CLR_CARD_DETECT	Mandatory		
ACMD51	SEND_SCR	Mandatory		SCR not supported by SDIO

图 23.7: SD 模式命令列表

使用 SPI 总线接口时 SD 内存和 SDIO 卡接受的命令列表如下图所示：

Supported Commands	Abbreviation	SDMEM System	SDIO System	Comments
CMD0	GO_IDLE_STATE	Mandatory	Mandatory	Used to change from SD to SPI mode
CMD1	SEND_OP_COND	Mandatory		
CMD5	IO_SEND_OP_COND		Mandatory	
CMD6	SWITCH_FUNC	Mandatory	Mandatory	
CMD9	SEND_CSD	Mandatory		CSD not supported by SDIO
CMD10	SEND_CID	Mandatory		CID not supported by SDIO
CMD12	STOP_TRANSMISSION	Mandatory		
CMD13	SEND_STATUS	Mandatory		Card Status includes only SDMEM information
CMD16	SET_BLOCKLEN	Mandatory		
CMD17	READ_SINGLE_BLOCK	Mandatory		
CMD18	READ_MULTIPLE_BLOCK	Mandatory		
CMD24	WRITE_BLOCK	Mandatory		
CMD25	WRITE_MULTIPLE_BLOCK	Mandatory		
CMD27	PROGRAM_CSD	Mandatory		CSD not supported by SDIO
CMD28	SET_WRITE_PROT	Optional		
CMD29	CLR_WRITE_PROT	Optional		
CMD30	SEND_WRITE_PROT	Optional		
CMD32	ERASE_WR_BLK_START	Mandatory		
CMD33	ERASE_WR_BLK_END	Mandatory		
CMD38	ERASE	Mandatory		
CMD42	LOCK_UNLOCK	Optional		
CMD52	IO_RW_DIRECT		Mandatory	
CMD53	IO_RW_EXTENDED		Mandatory	Block mode is optional
CMD55	APP_CMD	Mandatory		
CMD56	GEN_CMD	Mandatory		
CMD58	READ_OCR	Mandatory		
CMD59	CRC_ON_OFF	Mandatory	Mandatory	
ACMD13	SD_STATUS	Mandatory		
ACMD22	SEND_NUM_WR_BLOCKS	Mandatory		
ACMD23	SET_WR_BLK_ERASE_COUNT	Mandatory		
ACMD41	SD_APP_OP_COND	Mandatory		
ACMD42	SET_CLR_CARD_DETECT	Mandatory		
ACMD51	SEND_SCR	Mandatory		SCR includes only SDMEM information

图 23.8: SPI 模式命令列表

23.3.3.2 不支持的 SD 内存命令

SDIO 专用卡或组合卡的 I/O 部分不支持 SD 存储卡所需的几个命令。其中一些命令在 SDIO 卡中没有用处，例如 **Erase** 命令，因此在 SDIO 中不受支持。此外，有几个用于 SD 存储卡的命令在与卡的 SDIO 部分一起使用时具有不同的命令。下表列出了这些 SD 内存命令和等效的 SDIO 命令：

SD Memory Command	SDIO Command	Comments
CMD0	CMD52 (write to I/O reset in CCCR)	The reset command (CMD0) is only used for memory or the memory portion of Combo cards. In order to reset an I/O only card or the I/O portion of a combo card, use CMD52 to write a 1 to the RES bit in the CCCR (bit 3 of register 6). Note that in the SD mode, CMD0 is only used to indicate entry into SPI mode and shall be supported. An I/O only card or the I/O portion of a combo card is not reset with CMD0.
CMD12	CMD52 (write to I/O abort)	In order to CMD12. In write to the See 4.8 for abort the block transfer of data, SD memory use order to abort an I/O transaction, use CMD52 to abort register in the CCCR (bits 2:0 of register 6).
CMD16	CMD52 (write to I/O Block Length)	CMD16 sets the block length for SD memory. In order to set the block length for each I/O function, use CMD52 to write the block length in the FBR.
CMD2	NONE	The CID register does not exist in an SDIO only card.
CMD4	NONE	The DSR register does not exist in an SDIO only card.
CMD9	NONE	The CSD register does not exist in an SDIO only card.
CMD10	NONE	The CID register does not exist in an SDIO only card.
CMD13	NONE	An SDIO only card or the I/O portion of a combo card does not support the same SEND_STATUS (CMD13) protocol the SD memory uses.
ACMD6	CMD52 (write to Bus_Width [1:0] in CCCR)	SET_BUS_VWIDTH is handled by a write to the CCCR.
ACMD13	NONE	The SD Status register does not exist in an SDIO only card.
ACMD41	CMD5	SDIO cards and hosts use the IO SEND OP COND Command (CMD5).
ACMD42	CMD52	In the SD mode, the pull-up resistor on DAT[3] is controlled by writing to the CD Disable bit in the CCCR. For Combo Cards, this resistor is enabled unless both the memory and the I/O control registers are set to disable the resistor.
ACMD51	NONE	The SCR register does not exist in an SDIO only card.
CMD17 CMD18 CMD24 CMD25	CMD53	I/O block operations use CMD53, rather than memory block read/write commands.

图 23.9: 不支持的 SD 内存命令

23.3.3.3 改进的 R6 响应

存储卡对 CMD3 的正常响应为 R6，如下表所示：

Bit position	47	46	[45:40]	[39:8]Argument field		[7:1]	0
Width(bits)	1	1	6	16	16	7	1
Value	'0'	'0'	X	X	X	X	'1'
Description	Start bit	Direction bit	Command index ('000011')	New published RCA [31:16] of the card	[15:0] Card status	CRC7	end bit

图 23.10: CMD3 的 R6 响应

当 CMD3 被发送到仅 I/O 卡时，卡状态位（8~23）会发生更改。在这种情况下，16 位响应应为下表所示的仅 SDIO 值：

Bits	Identifier	Type	Value	Description	Clear Condition
15	COM_CRC_ERROR	E R	'0' = no error '1' = error	CRC check failed on previous command	B
14	ILLEGAL_COMMAND	E R	'0' = no error '1' = error	Command is illegal for card status	B
13	ERROR	E R X	'0' = no error '1' = error	A general or unknown error occurs during operation	C
12:0	Not defined. Only the SDIO card should read 0. The host should ignore these bits				

图 23.11: SDIO R6 状态位

23.3.3.4 SDIO 复位

为了复位 SDIO 卡或组合卡的 SDIO 部分中的所有功能，定义了一种不同于用于 SD 内存的方法。复位命令（CMD0）只用于内存或组合卡的内存部分。要复位仅 I/O 卡或组合卡的 I/O 部分，请使用 CMD52 将 1 写入 CCCR 中的 RES 位（寄存器 6 的位 3）。要注意的是，在 SD 模式下，CMD0 仅用于指示进入 SPI 模式，CMD0 不会复位仅 I/O 卡或组合卡的 I/O 部分。

23.3.3.5 总线宽度

对于 SD 存储卡，SD 模式的总线宽度使用 ACMD6 设置。SDIO 卡使用 CMD52 写入 CCCR 来选择总线宽度。对于组合卡，两种选择方法都存在。这种情况下主机应在开始任何数据传输之前，通过使用具有相同宽度设置的 ACMD6 和写入 CCCR 的 CMD52，在两个位置设置总线宽度。

23.3.3.6 卡检测电阻

SD 内存和 I/O 卡使用 DAT[3] 上的上拉电阻来检测卡插入。SD 存储器和 SDIO 之间启用/禁用此电阻的步骤不同。SD 内存使用 ACMD42 控制该电阻，而 SDIO 使用 CMD52 写入 CCCR。如果是组合卡，两个控制方式都存在，并应由主机管理。对于组合卡，仅当内存和 I/O 控制寄存器都启用电阻时，电阻才启用。也就是说，通电后，主机应使用 ACMD42 对内存控制器或对 SDIO 控制器的 CCCR 写入禁用电阻，因为电阻启用是两个启用中的逻辑与。通电后，两个位置默认为电阻启用。请注意，在 I/O 复位后，I/O 电阻启用不会改变。

23.3.3.7 数据传输块大小

SDIO 卡可以多字节（1 到 512 字节）或可选块格式传输数据，而 SD 存储卡固定在块传输模式下。SD 物理规范将数据传输的块大小限制为 2 的次方（即 512、1024、2048），除非使用部分读写。SDIO 规范允许从 1 字节到 2048 字节的任何块大小，以适应 I/O 功能的各种自然块大小。值得注意的是，SDIO 卡功能可在 CIS 中定义小于上述最大值的最大块大小或字节数。

23.3.3.8 数据传输中止

与 SD 存储卡通信的主机使用 CMD12 中止从/向该卡读取或写入数据的传输。对于 SDIO 卡，CMD12 中止由对 CCCR 中 ASx 位的写入代替。通常，中止用于停止无限块传输（块计数 = 0）。如果要传输的块的数量是确定的，建议主机发出具有正确块计数的块命令，而不是使用无限计数并在正确的时间中止数据。

23.3.4 新的 I/O 读或写命令

增加了两条额外的数据传输指令以支持 I/O。IO_RW_DIRECT 是一个类似于 MMC “快速 I/O” 命令的直接 I/O 命令，IO_RW_EXTENDED 允许使用字节或块地址进行快速访问。这两个命令都属于类 9（I/O 命令）。

23.3.4.1 IO_RW_DIRECT 命令（CMD52）

IO_RW_DIRECT 是在任何 I/O 功能，包括公共 I/O 区（CIA）中访问 128K 寄存器空间内的单个寄存器的最简单方法。此命令读取或写入 1 个字节仅使用 1 个命令/响应对。通常用于初始化寄存器或监视 I/O 功能的状态值。此命令是读取或写入单个 I/O 寄存器的最快方法，因为它只需要单个命令/响应对。该命令结构如下图所示：

S	D	Command Index 110100b	R/W flag	Function Number	RAW flag	Stuff	Register Address	Stuff	Write Data or Stuff Bits	CRC7	E
1	1	6	1	3	1	1	17	1	8	7	1

图 23.12: IO_RW_DIRECT 命令

其中：

- S: 起始位。总是 0；
- D: 方向。总是 1，表示从主机到卡的传输；
- Command Index: 使用值 110100b 来标识 IO_RW_DIRECT 命令；
- R/W Flag: 该位确定 I/O 操作的方向。如果该位为 0，则该命令应在功能号指定的地址和主机的寄存器地址从 SDIO 卡读取数据。数据字节在响应 R5 中返回。如果该位设置为 1，则命令应将写入数据字段中的字节写入由功能号和寄存器地址寻址的 I/O 位置。如果原始标志为 0，则应读取寄存器中写入的数据，并在响应中返回该值；
- RAW Flag: 写入后读取标志。如果该位设置为 1 且 R/W 标志设置为 1，则命令应在写入后读取寄存器的值。这对于允许写入控制寄存器和读取同一地址的状态非常有用。如果清除该位，R5 响应中返回的值应与命令中的写入数据相同。如果设置了该位，R5 响应的数据字段应包含写入操作后从寻址寄存器读取的值；
- Function Number: 希望读取或写入的 I/O 卡中的功能编号。功能 0 选择公共 I/O 区域（CIA）；
- Register Address: 这是所选功能中要读取或写入的数据字节的地址。有 17 位地址可用，因此寄存器位于该功能的前 128K（131072）地址内；
- Write Data/Stuff Bits: 对于直接写入命令（R/W=1），这是写入所选地址的字节。对于直接读取（R/W=0），不使用此字段，应将其设置为 0；
- CRC7: 7 位 CRC 数据；
- E: 结束位。总是 1。

23.3.4.2 IO_RW_DIRECT 响应 (R5)

SDIO 卡对 CMD52 的响应应采用两种格式之一。如果卡和主机之间的通信处于 1 位或 4 位 SD 模式，则响应应为 48 位响应 (R5)。如果操作是读取命令，则正在读取的数据将作为 8 位值返回。此外，还返回 15 位状态信息。如下图所示：

S	D	Command Index 110100b	Stuff	Response Flags 7-----Bit-----0	Read or Write Data	CRC7	E
1	1	6	16	8	8	7	1

图 23.13: SD 模式下的 IO_RW_DIRECT 响应

其中：

- S: 起始位。总是 0；
- D: 方向。总是 0，表示从卡到主机的传输；
- Command Index: 使用值 110100b 来标识 IO_RW_DIRECT 命令；
- Stuff Bits: 不使用，应设置为 0；
- Response Flags: 指示 SDIO 卡状态的 8 位标志数据；
- Read or Write Data: 对于设置了原始标志 (RAW=1) 的 I/O 写入 (R/W=1)，该字段应包含写入命令中包含的数据后从寻址寄存器读取的值。注意，在这种情况下，根据硬件的设计，读回数据可能与写入寄存器的数据不同。对于原始位为 0 的 I/O 写入，SDIO 功能不应进行写后读取操作，且该字段中的数据应与写入命令中的数据字节相同。对于 I/O 读取 (R/W=0)，从该 I/O 位置读取的实际值将在该字段中返回。
- CRC7: 7 位 CRC 数据；
- E: 结束位。总是 1。

如果通信使用 SPI 模式，响应应为 16 位 R5 响应。如果操作是读取命令，则正在读取的数据将作为 8 位值返回。此外，在 SPI R1 响应字节中返回 8 位状态信息。如下图所示：

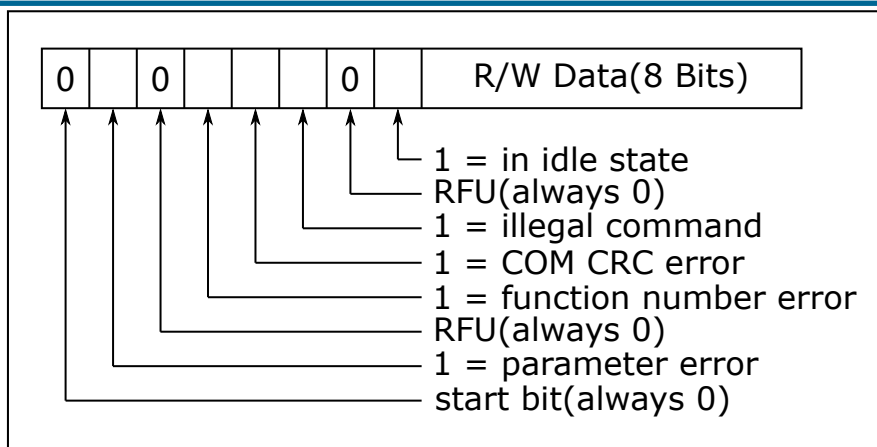


图 23.14: SPI 模式下的 IO_RW_DIRECT 响应

注：读/写（R/W）数据与 SD R5 响应所述的读/写数据相同。SPI 模式中的参数错误状态对应于超出范围和 SD 模式响应中的错误。对于 CMD53，还应使用数据错误标记指示超出范围和错误。

23.3.4.3 IO_RW_EXTENDED 命令（CMD53）

为了用一个命令读写多个 I/O 寄存器，定义了一个新命令 IO_RW_EXTENDED。此命令包含在命令类 9（I/O 命令）中。此命令允许使用单个命令读取或写入大量 I/O 寄存器。因为这是一个数据传输命令，所以它提供了可能的最高传输速率。IO_RW_EXTENDED 命令如下图所示：

S	D	Command Index	R/W flag	Function Number	Block Mode	OP Code	Register Address	Byte/Block Count	CRC7	E
1	1	110101b	1	3	1	1	17	9	7	1

图 23.15: IO_RW_EXTENDED 命令

其中：

- S: 起始位。总是 0；
- D: 方向。总是 1，表示从主机到卡的传输；
- Command Index: 使用值 110101b 来标识 IO_RW_EXTENDED 命令；
- R/W Flag: 该位确定 I/O 操作的方向。如果该位为 0，则该命令将从 SDIO 卡中以功能号和寄存器地址指定的地址向主机读取数据。读取的数据应在 DAT[x] 行上返回。如果该位设置为 1，则命令应将 DAT[x] 行中的字节写入由功能号和寄存器地址寻址的 I/O 位置；
- Function Number: 希望读取或写入的 I/O 卡中的功能编号。功能 0 选择公共 I/O 区域（CIA）；
- Block Mode: （可选）如果该位设置为 1，则表示读取或写入操作应在块基础上执行，而不是在正常字节基础上执

行。如果设置了该位，字节/块计数值应包含要读取/写入的块数。通过将块大小写入 FBR 中的 I/O 块大小寄存器来设置功能 1-7 的块大小。功能 0 的块大小通过写入 CCCR 中的 FN0 块大小寄存器来设置。块 I/O 模式的卡和主机支持是可选的。主机可以通过读取 CCCR 中的卡支持 MBIO 位（SMB）来确定卡是否支持块 I/O。块模式 =1 时使用的块大小和块模式 =0 时使用的每个命令的最大字节数可以从元组 TPLFE_MAX_BLK_SIZE 中的 CIS 中按功能读取；

- **OP code:** 定义读/写操作。0 表示对固定地址的多字节读/写，1 表示对递增地址的多字节读/写；
- **Register Address:** 要读取或写入的 I/O 寄存器的起始地址。范围为 0~0x1FFF；
- **Byte/Block Count:** 如果命令以字节（块模式 =0）操作，则此字段包含要读取或写入的字节数。0x000 的值将导致 512 字节被读取或写入；
- **CRC7:** 7 位 CRC 数据；
- **E:** 结束位。总是 1。

23.3.5 SDIO 卡内部操作

I/O 访问与内存的不同之处在于，寄存器可以单独或直接写入和读取，而无需 FAT 文件结构或块的概念（尽管支持块访问）。这些寄存器允许访问 I/O 数据、控制 I/O 功能、报告状态或向主机传输 I/O 数据。SD 内存依赖于固定块长度的概念，命令读取/写入这些固定大小块的倍数。I/O 可能有固定的块长度，也可能没有固定的块长度，读取大小可能不同于写入大小。因此，I/O 操作可能基于长度（字节计数）或块大小。

23.3.5.1 概述

每个 SDIO 卡可能有 1 到 7 个功能，以及内置的一个存储功能。功能是一个自包含的 I/O 设备。I/O 功能可能相同，也可能完全不同。所有 I/O 功能都组织为寄存器集合。每个 I/O 功能最多可能有 131072 (2^{17}) 个寄存器。这些寄存器及其各个位可以是只读（RO）、只写（WO）或读/写（R/W）。这些寄存器在卡中可以是 8、16 或 32 位宽。所有寻址都基于字节访问。这些寄存器可以一次写入和/或读取一个，乘法到同一地址或乘法到递增地址。单个 R/W 访问通常用于初始化 I/O 功能或读取单个状态或数据值。对固定地址的多次读取用于从卡中的数据 FIFO 寄存器读取或写入数据。读到递增地址用于向卡内的 RAM 区域读写数据集合。下图显示了 SDIO 卡的 CIA 和可选 CSA 空间的映射：

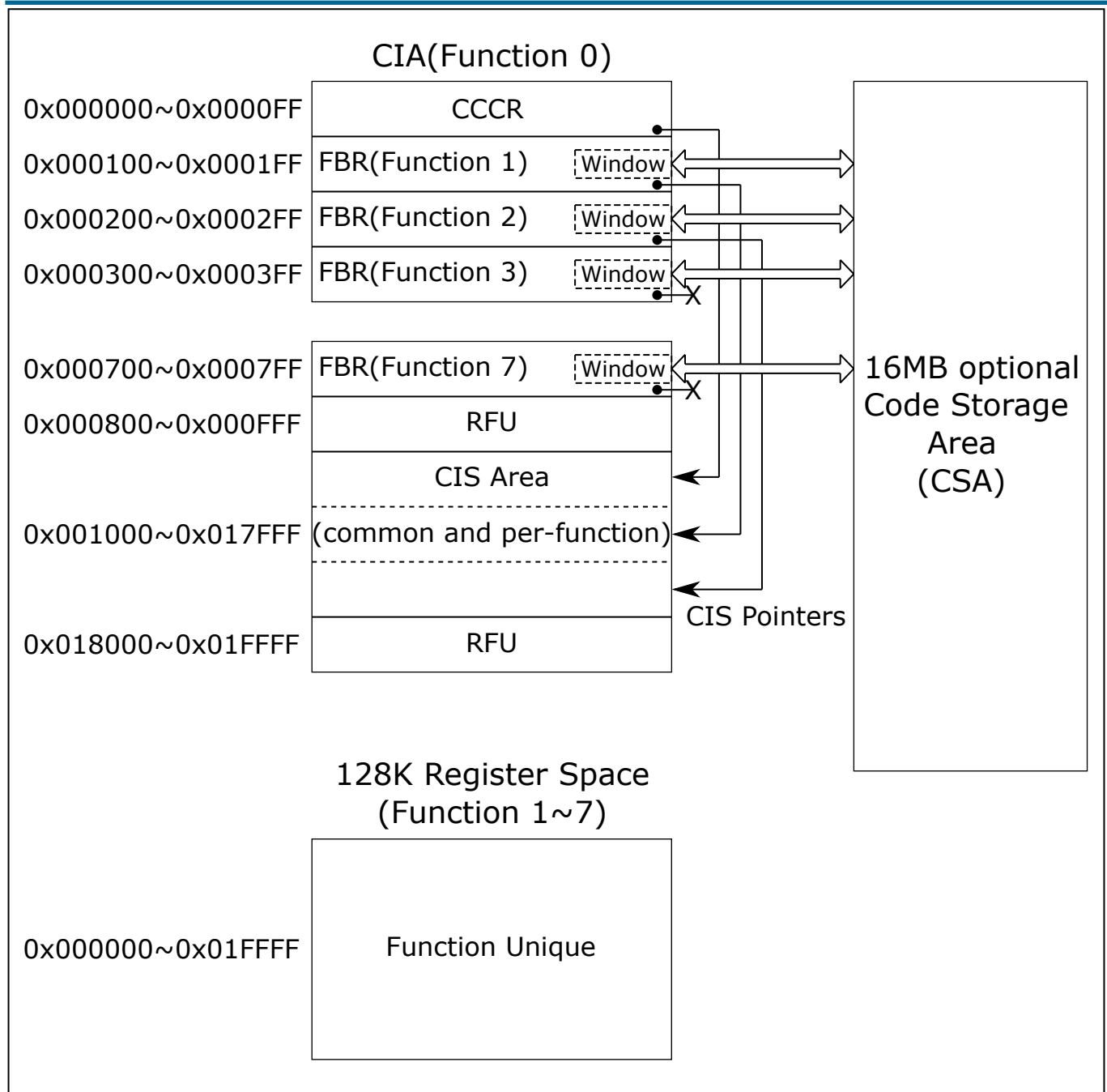


图 23.16: SDIO 内部映射

23.3.5.2 寄存器访问时间

仅 SDIO 卡和组合卡 SDIO 部分中的所有寄存器应在 1 秒内完成读写数据传输。此超时值与请求数据在 DAT[x] 线上传输到主机或从主机传输的时间有关，而不是与命令和响应之间的时间有关。该等待时间由卡向主机发出信号，使用 busy 进行写入或延迟开始位进行读取操作。主机可以使用 1 秒作为无响应位置的超时值。

23.3.5.3 中断

所有 SDIO 主机都应支持硬件中断。如果主机不支持中断，则可能难以使用期望对中断条件做出快速响应的 SDIO 卡。SDIO 或组合卡中的每个功能都可以根据需要实现中断。SDIO 功能上使用的中断类型通常称为“电平敏感”。电平敏感是指任何功能可在任何时候发出中断信号，但一旦该功能发出中断信号，在主机移除中断原因或命令这样做之前，不得释放（停止信号）中断。由于只有一条中断线，它可能被多个中断源共享。该功能应继续发出中断信号，直到主机响应并清除中断。由于多个中断可能同时处于活动状态，因此主机有责任确定中断源并根据需要进行处理。这是通过使用两位（中断启用和中断挂起）在 SDIO 功能上完成的。每个可能产生中断的功能都有一个中断启用位。此外，SDIO 卡有一个主中断启用，用于控制所有功能。如果功能的启用和卡的主启用都已设置，则仅向 SD 总线发送中断信号。第二个中断位称为中断挂起。该只读位告诉主机哪些功能可能发出中断信号。每个可以生成中断的功能都有一个中断挂起位。这些位位于 CCCR 区域中。

23.3.5.4 挂起/恢复

在多功能 SDIO 或组合卡中，有多个设备（I/O 和内存）共享对 SD 总线的访问。为了允许在多个设备之间共享对主机的访问，SDIO 和组合卡可以实现挂起/恢复的可选概念。如果卡支持挂起/恢复，主机可能会暂时停止对一个功能或内存的数据传输操作（挂起），以便释放总线，以便向不同的功能或内存进行更高优先级的传输。完成此高优先级传输后，原始传输将在停止的位置重新开始（恢复）。支持挂起/恢复是每个卡的可选功能。如果执行挂起/恢复，则组合卡的内存（如果有）和除 0（CIA）之外的所有 I/O 功能应支持挂起/恢复。主机可以挂起多个事务，并按所需的任何顺序恢复它们。I/O 功能 0 不支持挂起/恢复。支持挂起/恢复的任何卡还应支持读取等待和直接命令（SRW 和 SDC=1）。注意，挂起/恢复仅针对 SD 1 和 4 位模式定义，它不适用于 SPI 传输。

23.3.5.5 读等待

根据 SD 物理规范构建的主机设备应控制 SDCLK，以便在主机无法接受更多数据时，停止执行多重读取命令的卡的读取数据块输出。在主机停止 SDCLK 期间，无法发出 CMD52。此限制会导致一个问题，即按照 SD 物理规范构建的主机设备无法在多个读取周期内执行 I/O 命令。为了消除此限制，SDIO 规范添加了读取等待控件，以使主机能够在多个读取周期中发出 CMD52。读取等待使用 DAT[2] 线允许主机向卡发送信号，以暂时停止卡发送读取数据。此功能对于 SDIO 或组合卡是可选的。但是，如果 SDIO 或组合支持读取等待，则所有功能和任何内存都应支持读取等待。支持挂起/恢复的任何卡也应支持读取等待。注意，读取等待仅针对 SD 1 和 4 位模式定义，它不适用于 SPI 传输。

23.3.5.6 数据传输中的 CMD52

如果卡支持直接命令，则可以在数据传输期间接受 CMD52。对于 SD 和 SPI 模式，如果在数据传输过程中发生错误，SDIO 卡应接受 CMD52，以允许 I/O 中止和复位，而不管 SDC 值的该位值如何。

23.3.5.7 固定内部映射

SDIO 卡具有固定的内部寄存器空间和功能唯一区域。固定区域包含有关卡的信息以及固定位置的某些强制和可选寄存器。固定位置允许任何主机获取有关卡的信息，并以常见方式执行简单操作，如启用。功能唯一区域是每个功能区域，由标准 SDIO 功能的应用规范或非标准功能的供应商定义。

23.3.5.8 公共 I/O 区 (CIA)

公共 I/O 区 (CIA) 应在所有 SDIO 卡上实施。主机通过对功能 0 的 I/O 读写访问 CIA。CIA 内的寄存器用于启用/禁用 I/O 功能的操作、控制中断的产生以及可选地加载软件以支持 I/O 功能。CIA 中的寄存器还提供有关功能能力和要求的信息。CIA 支持三种不同的寄存器结构。他们是：

- 卡通用控制寄存器 (CCCR)
- 功能基础寄存器 (FBR)
- 卡信息结构 (CIS)

23.3.5.9 卡通用控制寄存器 (CCCR)

卡通用控制寄存器允许快速主机检查和控制 I/O 卡在每个卡（主卡）和每个功能的基础上的启用和中断。CCCR 中的位混合为读/写和只读。如果 SDIO 卡上未提供 7 种可能功能中的任何一种，则与未使用功能相对应的位应全部为只读，并读取为 0。所有保留供将来使用的位 (RFU) 应为只读，并返回值 0。通电或复位后，所有可写位均设置为 0。当 I/O 功能被禁用时，即使在初始化之后也可以访问 CCCR。这允许主机在初始化后启用功能。

23.3.5.10 功能基础寄存器 (FBR)

除 CCCR 外，每个受支持的 I/O 功能都有一个 256 字节的区域，用于让主机快速确定每个功能的能力和要求，为每个功能启用电源选择，并启用软件加载。该区域的地址从 0x00n00 到 0x00nFF，其中 n 是功能号 (0x1 到 0x7)。

23.3.5.11 卡信息结构 (CIS)

卡信息结构提供有关卡和单个功能的更完整信息。CIS 是用于读取卡中所有 I/O 功能信息的公共区域。该设计基于 PCMCIA 标准化的 PC Card16 设计。支持 I/O 的所有卡应具有通用 CIS 和每个功能的 CIS。通过读取 0x1000~0x17FFF 区域来访问 CIS，该区域作为公共 CIS 和每个功能的存储区域服务于卡。公共区域和每个功能都有一个指针，指向该内存空间中 CIS 的开始。

23.3.5.12 多功能 SDIO 卡

多功能 SDIO 卡为卡上的每个功能配备一套单独的配置寄存器。多功能 SDIO 卡使用卡上所有功能共用的 CIS 和卡上每个功能专用的单独功能专用 CIS 的组合。通用 CIS 描述了卡上所有功能的通用功能。每个功能特定 CIS 描述 SDIO 卡上特定功能的细节特征。功能从 1 开始按顺序编号。CMD5 响应表示功能总数，其中包括“虚拟”功能。主机应根据 CMD5 响应遍历 CIS 条目。R5 响应的错误状态标志为“E R X”类型，可指示前一个命令中的错误。由于主机软件需要一种方法来确定检测到错误的功能，因此多功能 SDIO 卡只能在向同一功能发出的后续命令中返回 R5 错误状态标志。

23.3.5.13 使用 CMD53 设置块大小

主机通过写入 FBR 中的 16 位功能 I/O 块大小寄存器，为函数的多个块传输设置块大小。主机不得使用块模式设置为 1 的 CMD53 写入此寄存器。如果卡在执行 CMD53 之前检测到无效的块大小，且块模式设置为 1，则卡应在当前响应中指示超出范围错误，并且不应执行数据传输。这也将停止中断周期。

23.3.5.14 总线状态图

下图是 SDIO 卡的总线状态图，它显示了总线状态及其与 SDIO 命令和挂起/恢复的关系：

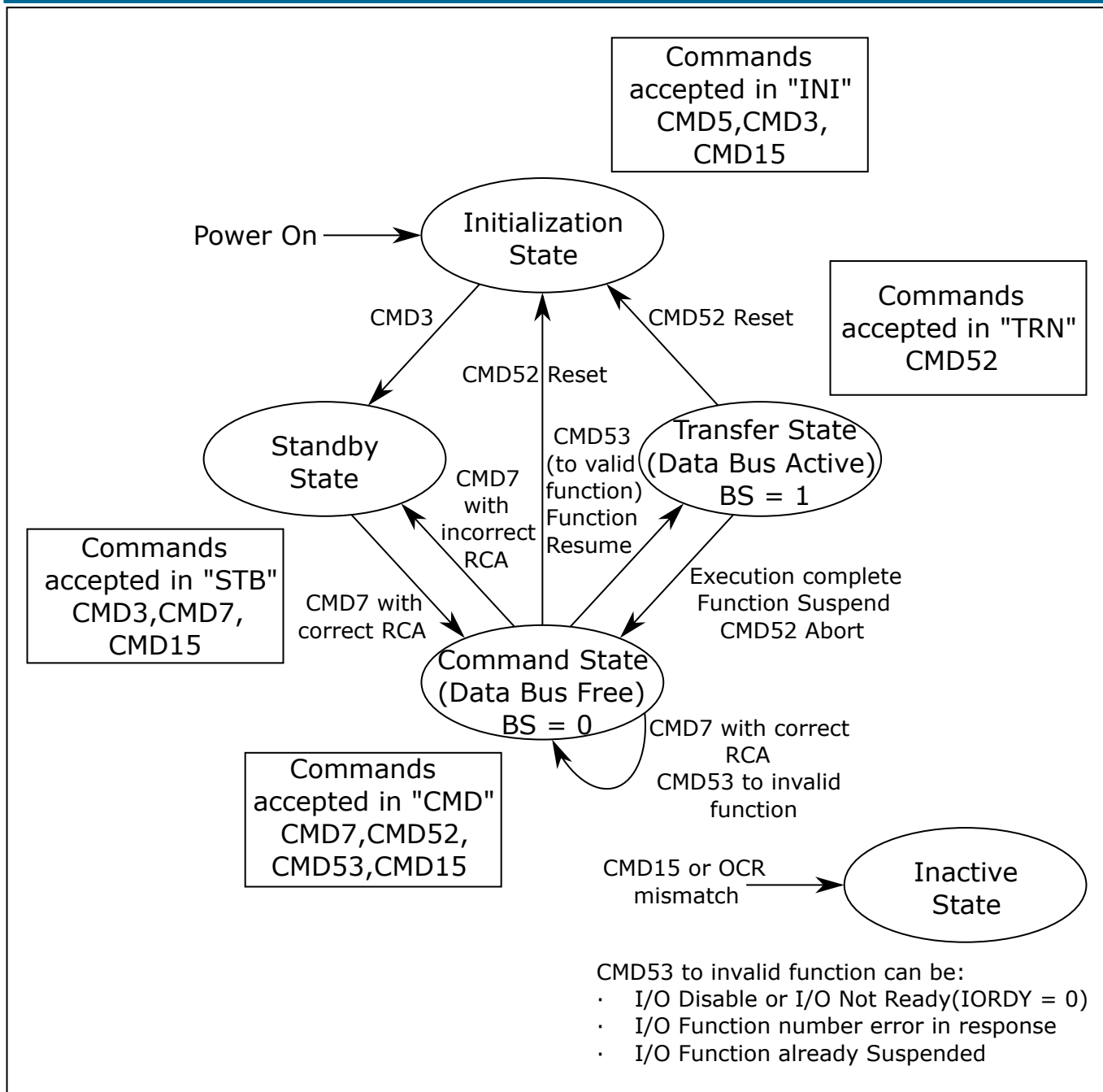


图 23.17: 总线状态机状态图

23.3.6 嵌入式 I/O 代码存储区 (CSA)

为了支持 SDIO 卡的“即插即用”概念，卡中包含的每个功能可能需要包含一块内存，用于存储驱动程序和/或应用程序。此外，由于同一 SDIO 卡可在多个不同的主机平台上使用，因此每个功能可能需要多个不同版本的代码。一种选择是将这些程序存储在组合卡的标准 SD 内存部分。或者，用于加载代码的标准访问装置包含在可选代码存储区域 (CSA) 中。CSA 是一个单独的 16MB 内存区域，可使用 FBR 寄存器中包含的 CSA 地址指针和 CSA 窗口寄存器进行访问。

23.3.6.1 CSA 访问

为了让主机访问功能的 CSA，它首先应确定该功能是否支持 CSA。主机读取地址 0x00n00 处的 FBR 寄存器，其中 n 是功能号 (0x1 到 0x7)。如果位 6=1，则函数支持 CSA，主机通过写入位 7=1 启用访问。下一步是主机加载 24 位地址以开始读取或写入。这是通过将 24 位 (A23-0) 写入寄存器 0x00n0C 到 0x00n0E 来实现的，其中 n 是功能号 (0x1 到 0x7)。写入起始地址后，可以通过访问寄存器 0x00n0F (CSA 数据窗口寄存器) 读取或写入数据。如果需要读取或写入超过 1 个字节，则可以使用 0 (固定地址) 的操作码执行扩展 I/O 命令 (字节或块)。每次访问窗口寄存器时，地址指针应自动递增，以便访问 CSA 内的顺序地址。一旦操作完成，下一个操作的地址应保存在 24 位地址寄存器中，以便主机读取。

23.3.6.2 CSA 数据格式

存储在 CSA 中的数据应使用 FAT12/FAT16 格式进行结构化。使用 CSA 存储不同主机类型的程序或数据需要 SDIO 卡制造商以主机可以识别的文件格式加载程序和数据。这方面的一个例子是使用保存在特定子目录中的特定文件名，该文件名由特定主机操作系统识别和执行。这些格式是特定的，有时是不同主机实现和操作系统的专有格式。

23.3.7 SDIO 中断

为了允许 SDIO 卡中断主机，在 SD 接口的引脚上添加了中断功能。在 4 位 SD 模式下操作时用作 DAT[1] 的引脚 8 用于向主机发送卡中断信号。对于卡中的每个卡或功能，中断的使用是可选的。SDIO 中断是“电平敏感”的，也就是说，中断线应保持激活 (低)，直到主机识别并对其进行操作，或者由于中断周期结束而取消断言。一旦主机为中断提供服务，它将通过一些功能独特的 I/O 操作被清除。所有主机应在数据线 DAT[3:0] 上提供上拉电阻。

23.3.7.1 SPI 和 SD 1 位模式中断

在 SPI 和 1 位 SD 模式下，引脚 8 专用于中断功能。因此，在 SPI 和 SD 1 位模式中，中断没有时间限制。SPI 或 1 位 SD 模式下的卡通过断言引脚 8 为低，随时向主机发送中断信号。主机使用电平敏感输入检测此挂起中断。主机负责清除中断。如果 SDIO 卡在 SPI 模式下运行，如果未选择该卡，则可能不会断言该卡的中断。(CS=0)。只有当卡在选择时 (CCCR 中的 SCS1 位 =1) 能够中断，并且该功能已打开 (ECSI 位 =1) 时，才会出现此要求的例外情况。在这种情况下，无论 CS 线路的状态如何，卡都可以断言中断。

23.3.7.2 SD 4 位模式中断

由于引脚 8 在 4 位 SD 模式下的 IRQ 和 DAT[1] 使用之间共享，因此中断只能由卡发送，并在特定时间内由主机识别。引脚 8 上的低电平被识别为中断的时间被定义为中断周期。在中断期间，SDIO 主机只能将引脚 8（DAT[1]/IRQ）上的电平采样到中断检测器中。在所有其他时间，主机中断控制器应忽略引脚 8 上的电平。注意，中断周期适用于内存和 I/O 操作。对于单块和多块数据传输的操作，中断周期的定义是不同的。

23.3.7.3 中断清除时间

由于 SDIO 卡使用电平敏感中断，主机应通过对某个功能唯一区域的 I/O 读取或写入来清除未决中断。在某些主机实现中，向卡发送 CMD52 由主机适配器硬件处理，而主机 CPU 可以执行其他操作。如果中断清除的时间不受控制，此情况可能允许已处理的中断重新中断主机。为防止出现这种情况，任何实现中断的 SDIO 卡在写入清除中断的功能唯一区后，应遵循从 DAT[1] 行中删除中断所需的时间。中断的清除可由功能唯一方法中的 I/O 写入或功能唯一 I/O 读取引起。使用 I/O 读取清除中断的一个例子是，数据寄存器的读取可自动清除数据就绪中断。

23.3.8 SDIO 挂起/恢复操作

用于在 SD 总线上执行挂起/恢复操作的步骤为：

- 主机确定当前使用 DAT[3:0] 行的功能。
- 主机请求较低优先级或较慢的事务挂起。
- 主机检查事务挂起是否完成。
- 主机开始更高优先级的事务。
- 主机等待高优先级事务的完成。
- 主机恢复挂起的事务。

如果当前事务可以接受挂起，并且卡在读取等待期间收到挂起命令，则应接受挂起请求。

23.3.9 SDIO 读取等待操作

可选的读取等待（RW）操作仅针对 SD 1 位和 4 位模式定义。读取等待操作允许主机向正在执行多次读取（CMD53）操作的卡发送信号，以临时暂停数据传输，同时允许主机向 SDIO 卡内的任何功能发送命令。为了确定卡是否支持读取等待协议，主机应测试 CCCR 卡容量字节中的 SRW 容量位。读取等待的计时基于中断周期。如果卡不支持读等待协议，则主机必须在读多命令的中间停顿（不中止）数据的唯一手段是控制 SDCLK。读等待支持对于支持挂起/恢复的卡是强制性的。

24.1 概述

低功耗是物联网应用的一项重要指标。芯片的处理器包含 3 种功耗模式，包含工作模式、空闲省电模式和休眠模式，可以根据当前应用场景选择合适的功耗模式，降低芯片功耗延长电池寿命。

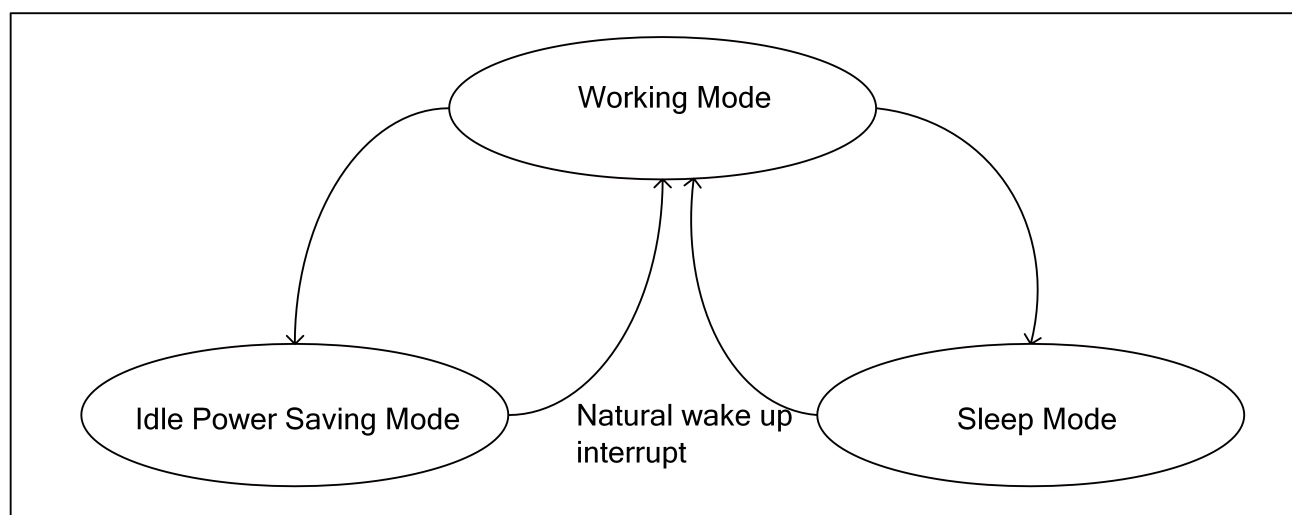


图 24.1: 低功耗模式

24.2 主要特征

- 时钟控制: GLB 中对各外设的时钟控制，小范围的省电，响应速度较快
- 睡眠控制 (PDS): 包含 PDS1/2/3/7/15 这 5 个等级，大范围省电，响应速度中等
- 深度睡眠控制 (HBN): 包含 HBN0/1/2/3 这 4 个等级，全局省电，响应时间长

24.3 功能描述

24.3.1 电源域

BL616/BL618 芯片中有 8 个电源域，每个电源域的主要功能如下所示：

- PD_AON
 - HBN 状态机，可控制 PD_AON_HBNRTC/PD_AON_HBNCORE/PD_CORE 的电源/隔离单元/复位/时钟
 - AON_PIN (GPIO16/17/18/19) 引脚唤醒功能
 - 预留 4 个可读可写的寄存器，可在 PDS/HBN0/HBN1/HBN2 模式下保存数据
 - BOR (Brown Out Reset) 设置功能
 - LDO11_AON/RT/SOC 输出电压选择单元
 - Root、F32K、Uart 时钟源选择
 - HBN_OUT0_PIR (Acomp0/Acomp1/bor/pir) 唤醒屏蔽和使能寄存器、中断状态寄存器
- PD_AON_HBNRTC
 - 选择 RTC Counter 时钟源的控制单元
 - RTC 可以用于唤醒，也可用于 LED 闪烁
 - RC32K 和 Xtal32X 控制功能
- PD_AON_HBNCORE
 - 部分电源控制寄存器
 - 保留 HBN_RAM，可用于保存程序和数据，进入 PDS/HBN0 后数据不会消失
 - 在 HBN 模式下，具有控制 AON_PIN 和保持其他 IO 的功能
 - PIR 数字控制，PIR 是热释电红外传感器，HBN 区域内的一个外设，可以用作 HBN 唤醒源
 - 保留 LDO11SOC、LDO15RF、DCDC0、DCDC1、DCDC2 控制寄存器
 - 保留 XTAL、TSEN、Acomp0/1、GPADC 控制寄存器
 - 预留 4 个可读可写的寄存器，可在 PDS/HBN0 模式下保存数据
- PD_CORE
 - PDS 状态机控制 PD_CORE_MISC/PD_USB/PD_CPU/PD_WB 的电源、隔离单元、复位、时钟和 Memory
 - 保留 PDS_RAM, 进入 PDS 后数据不会消失
 - WIFI/BLE 计时器控制

- 160KB WRAM Retention/Sleep
- 控制 PDS 中断和唤醒功能
- 保留 GPIO0~15/20~36 的控制
- 保留 RC32M 的控制
- PDS_Timer 定时器
- PD_CORE_MISC
 - PD_CORE_MISC 是 PD_CORE_MISC_DIG 和 PD_CORE_MISC_ANA 的统称
 - 外设（包括 I2C/SDIO/UART/FLASH/SPI/ADC/DAC/GPIO/PWM 等）
 - 芯片 GLB 寄存器
- PD_USB
 - USB 控制器
- PD_CPU
 - NP_CPU 及缓存单元
 - ROM、TCM
- PD_WB
 - WIFI PHY/MAC
 - BLE PHY/MAC
 - RF Controller

24.3.2 电源模式

BL616/BL618 共有 10 个电源模式，8 个电源域。每种电源模式对用的电源域的开关状态如下表所示：

表 24.1: 电源模式

NO.	Scenario	Power Domain							
		PD_AON	PD_AON_- HBNRTC	PD_- AON_- HBNCORE	PD_- CORE	PD_CORE_- MISC	PD_USB	PD_CPU	PD_WB
1	Normal	ON	ON	ON	ON	ON	ON	ON	ON
2	PDS1	ON	ON	ON	ON	ON	ON	ON	OFF
3	PDS2	ON	ON	ON	ON	ON	ON	OFF	ON
4	PDS3	ON	ON	ON	ON	ON	ON	OFF	OFF
5	PDS7	ON	ON	ON	ON	ON	OFF	OFF	OFF

表 24.1: 电源模式 (continued)

NO.	Scenario	Power Domain							
		PD_AON	PD_AON_- HBNRTC	PD_- AON_- HBNCORE	PD_- CORE	PD_CORE_- MISC	PD_USB	PD_CPU	PD_WB
6	PDS15	ON	ON	ON	ON	OFF	OFF	OFF	OFF
7	HBN0	ON	ON	ON	OFF	OFF	OFF	OFF	OFF
8	HBN1	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
9	HBN2	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
10	HBN3	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

10 种电源模式的功耗分别如下：

表 24.2: 电源模式的功耗

NO.	Scenario	current/uA
1	Normal	30000
2	PDS1	930
3	PDS2	1400
4	PDS3	550
5	PDS7	540
6	PDS15	70
7	HBN0	3.1
8	HBN1	2.7
9	HBN2	2.1
10	HBN3	

24.3.3 唤醒源

芯片内部支持多种唤醒源，可从不同电源模式中唤醒。

不同电源模式的唤醒源如下表所示：

表 24.3: 唤醒源

电源模式	唤醒源
PDS1	AON_PIN/BOR/RTC/Pir/Acomp0/Acomp1/PDS_Timer/GPIO/IRRX/DM/USB

表 24.3: 唤醒源 (continued)

电源模式	唤醒源
PDS2	AON_PIN/BOR/RTC/Pir/Acomp0/Acomp1/PDS_Timer/GPIO/IRRX/DM/USB/WIFI
PDS3	AON_PIN/BOR/RTC/Pir/Acomp0/Acomp1/PDS_Timer/GPIO/IRRX/DM/USB
PDS7	AON_PIN/BOR/RTC/Pir/Acomp0/Acomp1/PDS_Timer/GPIO/IRRX/DM
PDS15	AON_PIN/BOR/RTC/Pir/Acomp0/Acomp1/PDS_Timer/PDS_GPIO
HBN0	AON_PIN/BOR/RTC/Pir/Acomp0/Acomp1
HBN1	AON_PIN/RTC
HBN2	AON_PIN
HBN3	重新给 VDDIO2 供电

24.3.4 功耗模式

工作模式

芯片提供处理器与外设独立的时钟控制，在 GLB 和时钟的章节介绍对各模块的时钟控制，软件可以根据当前应用场景，对于不需要使用的处理器或外设进行时钟控制。时钟控制的响应是实时的，在此工作模式下，不需要担心响应时间。

掉电睡眠模式 (PDS)

掉电模式相较于工作模式功耗较低。进入 PDS 模式后，将 RTC(Real Time Clock) 之外的时钟进行管控，会切换为内部低速时钟，并将外部晶振与 PLL 关闭达到更加省电的状态，因此进入与离开此低功耗模式会有时间延迟。当进入掉电睡眠模式时，OCRAM 区域的数据可以自动进入 retention 状态而保留下来，当唤醒后可以自行退出 retention 状态。

1. 进入空闲省电模式

软件可通过 PDS 配置让此模块进入掉电模式，等待处理，进入等待中断模式 (WFI) 后，PDS 模块会触发时钟控制模块进入 gate clock 操作，并通知模拟电路关闭 PLL 以及外部晶振

2. 离开空闲省电模式

离开空闲省电模式的方式有两种，第一是空闲中间有特定的中断或事件打断空闲状态，第二是软件设定 PDS_TIMER 的空闲时间达到，两者均会触发 PDS 模块进入或离开掉电模式。注意：因为打开晶振需要约 1ms 的时间，PDS 提供软件提前打开晶振的方式，这个做法可以加速 PDS 醒来，当 PDS 模块准备醒来时，此模块会通过中断通知处理器离开等待中断模式 (WFI)。

休眠模式 (HBN)

休眠模式在保留 AON(Always On) 电源的状态下，将大部分的芯片逻辑进行断电 (Vcore)，直到收到外部事件才会将内部电路唤醒的。在休眠模式下可以达到极致的省电状态，但相对于前两者需要的响应时间也最长，适合长时间不需要工作的状态下，可以进入此状态，延长电池寿命。休眠时期会将大部分的电路断电，对应的寄存器值和内存的数据也会消失。因此 HBN 内部留有 4KB HBN_RAM，这个内存在休眠状态时不会断电，软件有需要保存的资料或状态可以

在进入休眠前拷贝到这个内存。从休眠恢复时，可以直接从 RAM 中存取数据，通常可以用作状态的纪录或是数据快速恢复。

24.3.5 IO 唤醒

PDS1~15、HBN0~2 都支持 IO 唤醒，但是具体实现方式不同。

表 24.4: IO wakeup

电源模式	IO 唤醒实现方法
PDS1~7	使用 GLB_IO 唤醒
PDS15	使用 PDS_IO 唤醒 (高电平/下降沿/睡眠之前为低电平的上升沿) 和 AON_IO 唤醒 (低电平/高电平/下降沿/上升沿/上升或者下降沿)
HBN0~2	使用 AON_IO 唤醒

GLB_IO 唤醒 PDS1~7 模式

所有 IO 都支持唤醒 PDS1~7 模式，PDS1~7 模式启动 IO 唤醒代码如下：

Listing 24.1: glb io wakeup pds1~7

```
1 BL_Err_Type glbio_wakeup(uint32_t gpio_id, uint32_t trigger_type)
2 {
3     gpio_set_mode(gpio_id, trigger_type);
4     gpio_irq_enable(gpio_id, ENABLE);
5     gpio_attach_irq(gpio_id, pm_gpio_callback);
6
7     pm_pds_wakeup_src_en(PDS_WAKEUP_BY_GLB_GPIO_IRQ_EN_POS);
8 }
```

PDS_IO 唤醒 PDS15 模式

PDS_IO 是除了 AON_IO (GPIO16~19) 的所有 IO, 都支持唤醒 PDS15 模式。

1. PDS_IO 支持的唤醒模式

- 同步下降沿
- 同步高电平
- 异步下降沿
- 异步高电平

2. PDS_IO 唤醒模式的分组

PDS_IO 的唤醒模式可分为 GPIO0~7、GPIO8~15、GPIO20~27、GPIO28~34 四组。每组 GPIO 共享相同的唤醒模

式，例如，GPIO0 设置为高电平唤醒，GPIO7 也只能配置为高电平唤醒，而 GPIO20 可以配置为下降沿唤醒。

注意：如果在 PDS15 状态下使能 PDS_IO 唤醒，则必须使能该 GPIO 的内部 PU/PD，否则电流将增加到 1000+ua。

3. PDS_IO PUPD 配置的分组

PDS_IO 的 PUPDIE 配置可分为 GPIO0~15、GPIO20~34 两组。每组 GPIO 共享相同的 PUPDIE 配置，例如，GPIO0 设置为 PU，GPIO8 也只能配置为 PU，而 GPIO20 可以配置为 PD。

注意：由于 GPIO0~15 共享一组 PUPDIE 配置，可能会限制 GPIO0~7、GPIO8~15 的触发方式配置。例如，GPIO0~15 设为 PU，那么 GPIO0~7、GPIO8~15 的唤醒模式都不能选择高电平唤醒。同理，由于 GPIO20~34 共享一组 PUPDIE 配置，可能会限制 GPIO20~27、GPIO28~34 的触发方式配置。

注意：使用 GPIO0~15 作为唤醒引脚时，需要注意 GPIO2 是 BOOT 脚，通常接了下拉电阻，GPIO0~15 设为 PU 时可能由于 GPIO2 下拉导致漏电；而 GPIO4 是 FLASH_CS 脚，通常接了上拉电阻，GPIO0~15 设为 PD 时可能由于 GPIO4 上拉导致漏电。

GPIO0~15 作为唤醒引脚时，如果 GPIO2、GPIO4 外部同时分别有下拉电阻和上拉电阻，建议将 GPIO0~15 设为 PU，同时把 BOOT 脚的下拉电阻换成 1M 欧姆的，减少漏电流。

注意：使用 GPIO20~34 作为唤醒引脚时，需要注意 GPIO21、22 可能用作 UART 脚，串口空闲状态为高电平，所以建议设置内部上拉；如果 GPIO21、22 没有接串口，那么配置上拉或者下拉都可以。

例如，PDS15 模式启动 IO3 下降沿唤醒代码如下：

```
PDS_Set_GPIO_Pad_IntClr(PDS_GPIO_INT_SET_1_GPIO0_GPIO7);
PDS_Set_GPIO_Pad_Pn_Pu_Pd_Ie(PDS_GPIO_GROUP_SET_GPIO0_GPIO15, 1, 0, 1);
PDS_Set_GPIO_Pad_IntMode(PDS_GPIO_INT_SET_1_GPIO0_GPIO7,
                          (PDS_GPIO_INT_TRIG_Type)trigger_type);
PDS_Set_GPIO_Pad_IntMask((GLB_GPIO_Type)GLB_GPIO_PIN_3, UNMASK);

pm_pds_wakeup_src_en(PDS_WAKEUP_BY_PDS_GPIO_IRQ_EN_POS);
```

由于低功耗时芯片内部的 PU/PD 是按照分组设置的，在实际使用中，很难保证这一组内的 GPIO 全部是上拉或者下拉，因此，建议通过 PCB 上分别给 GPIO 加上拉或者下拉电阻的方式来避免漏电，不建议使用芯片内部的 PU/PD 实现上拉或者下拉。

对于 BL616 QFN40 封装，GPIO4~9/23~26/31~34 没有封出来，内部 pin 是打线连接到 GND 状态，如果此时使用芯

片内部的 PU 实现上拉，那么这些引脚就会有漏电，为避免出现这种情况，SDK 的驱动接口中，已经禁止用户设置芯片的上拉。

AON_IO 唤醒 PDS15 模式

PDS15 模式支持 AON_IO (GPIO16~19) 唤醒，AON_IO 共享相同的触发方式，但是 PUPDIE 配置是独立的。例如如果 GPIO16 设置为下降沿触发方式唤醒，则 GPIO17~19 不能配置为其他触发方式。如果 GPIO16 设置为 PU，GPIO17~19 可以不配置 PU。

```
static BL_Err_Type aonio_wakeup_pds15(uint32_t pin, uint32_t trigger_type)
{
    uint32_t maskVal = 0xF;
    HBN_AON_PAD_CFG_Type aonPadCfg;

    aonPadCfg.ctrlEn = 1;
    aonPadCfg.ie = 1;
    aonPadCfg.oe = 0;
    aonPadCfg.pullUp = 0;
    aonPadCfg.pullDown = 0;
    HBN_Aon_Pad_Cfg(DISABLE, (pin - 16), &aonPadCfg);
    maskVal = 0xF & ~(1 << (pin - 16));
    HBN_Aon_Pad_WakeUpCfg(DISABLE, trigger_type, maskVal, 0, 7);

    pm_hbn_out0_irq_register();

    pm_pds_wakeup_src_en(PDS_WAKEUP_BY_HBN_IRQ_OUT_EN_POS);

    pm_pds_mode_enter(PM_PDS_LEVEL_15, 0);
}
```

AON_IO 唤醒 HBN0~2 模式

HBN0~2 模式支持 AON_IO (GPIO16~19) 唤醒，且 AON_IO 共享相同的触发方式，但是 PUPDIE 配置是独立的。例如，如果 GPIO16 设置为下降沿触发方式唤醒，则 GPIO17~19 不能配置其他触发方式。如果 GPIO16 设置为 PU，GPIO17~19 可以不配置 PU。

```
static BL_Err_Type aonio_wakeup_hbn(uint32_t pin, uint32_t trigger_type)
{
    uint32_t maskVal = 0xF;
    HBN_AON_PAD_CFG_Type aonPadCfg;

    aonPadCfg.ctrlEn = 1;
    aonPadCfg.ie = 1;
    aonPadCfg.oe = 0;
```

(continues on next page)

```
aonPadCfg.pullUp = 0;
aonPadCfg.pullDown = 0;
HBN_Aon_Pad_Cfg(DISABLE, (pin - 16), &aonPadCfg);
maskVal = 0xF & ~(1 << (pin - 16));
HBN_Aon_Pad_WakeUpCfg(DISABLE, trigger_type, maskVal, 0, 7);

pm_hbn_mode_enter(PM_HBN_LEVEL_0, 0);
}
```

24.3.6 ACOMP 唤醒

芯片支持 ACOMP 唤醒，ACOMP 有两个比较模块，每个 ACOMP 模块均有正极输入通道和负极输入通道两个输入口，每个输入口都可以选择：

- ADC channel0-7
- DAC 的输出 A/B
- 系统 1.25V 的参考电压 (低功耗模式该电压不存在)
- VDD33 的分压 (分压系数可调整)

分压系数可以选择以下中的一个：

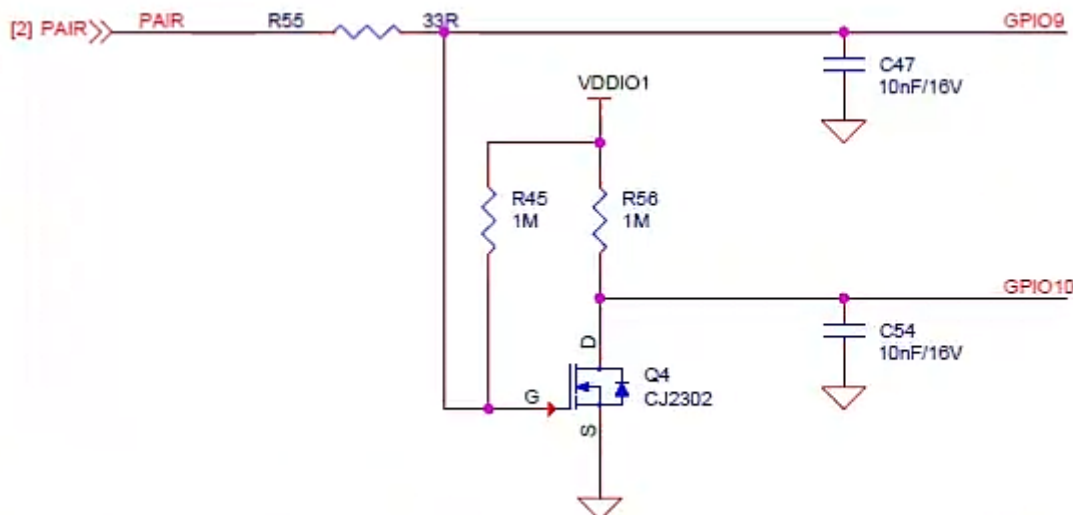
- 0.25
- 0.5
- 0.75
- 1

在常规的应用中，一般将正极输入通道设置为 ADC channel0-7 中的一个，也就是使用 GPIO2、3、10、12、13、14、19、20 中一个，负极选择 VDD33 的分压，分压系数可根据实际应用场景选择，在正极输入电压由小于负极电压变为大于负极电压时产生一个上升沿中断唤醒源，在正极输入电压由大于负极电压变为小于负极电压时产生一个下降沿中断唤醒源。当使能一路 ACOMP 时电流增加约 7uA 左右，同时使能两路 ACOMP 电流增加约 10uA 左右。

24.3.7 IO 双边沿唤醒

AON_IO 和 ACOMP 支持双边沿唤醒。

PDS_IO 实现双边沿唤醒需要使用两个引脚，都设置成下降沿唤醒，配合以下电路可以实现双边沿唤醒：



但是在双边沿唤醒的场景中，引脚的输入既有可能是高电平，又有可能是低电平，所以不能使能内部上下拉；否则一旦引脚电平和内部上下拉配置相反，会产生漏电，漏电电流 $= 3.3V / \text{电阻}$ 。由于内部不能使能上下拉，则同组的其他引脚也没有内部的上下拉，则 pcd 上需要配合外部的上下拉来保证引脚不是悬空的，如果用户能保证芯片睡眠期间外部的输入不是悬空的，就可以不用上下拉。

例如，GPIO15 和 GPIO20 接同一输入源，GPIO15 和 GPIO20 设为下降沿唤醒，那么 GPIO0~15、GPIO20~34 这两组引脚都不能使能内部上下拉，GPIO0~14、GPIO21~34 引脚外部需要接上下拉电阻。

错误：GPIO15 和 GPIO20 接同一输入源，GPIO15 设为下降沿唤醒，GPIO20 设为高电平唤醒；当电平由低到高时，芯片从 PDS 模式唤醒，但是 PDS_GPIO_STAT 寄存器的值是 0x00014000，表示同时出现了高电平唤醒和下降沿唤醒，可是示波器并没有抓到下降沿。

24.3.8 IO 唤醒触发条件

io 唤醒对电平时序的要求如下：

	时间 / us
pds io	100
aon io	100
acompl	50

例如，GPIO0 作为 pds_io 唤醒引脚，触发模式是下降沿，那么电平的时序必须满足下降沿出现后，低电平保持大于等于 100us 的要求。

GPIO18 作为 aon_io 唤醒引脚，触发模式是高电平，那么电平的时序必须保持高电平时间大于等于 100us。

GPIO13 作为 `acompl` 唤醒，触发模式是上升沿唤醒，那么电平的时序必须满足上升沿出现后，高电平保持大于等于 50us 的要求。

24.4 POR & BOD 说明

BL616 内置了低压锁定 (Under Voltage Lock out) 电路, POR (Power on Reset) 电路和低压检测 (Brown out Detector) 电路。

24.4.1 UVLO 和 POR

当电源电压 VDD 低于 1.8V 时, UVLO 电路无法正常工作, 芯片状态不确定, 电流也不确定。当电源电压 VDD 在 1.8V ~ 2.0V 之间时, UVLO 电路正常工作, 保持为高电平, 芯片会把内部所有电源断电, 芯片的总电流很小。当电源电压 VDD 渐渐升高到 2.0V 时, UVLO 输出低电平, POR 开始为高电平; 2ms 过后, POR 由高变低, 系统释放 Reset。

注解: UVLO 和 POR 电路上电就默认工作, 不需要软件配置。

24.4.2 BOD

BOD 有 8 个档位可选, 分别是 2.4V、2.35V、2.3V、2.25V、2.2V、2.15V、2.1V、2.05V。BOD 有 50mV 迟滞, 以 2.4V 挡位为例, 电源电压 VDD 低于 2.4V 时, BOD 信号会保持为高电平; VDD 高于 2.45V 时, BOD 信号输出低电平。

注解: 当电源电压 vdd 低于 2.0V 时, UVLO 会把 BOD 电路关断。

注解: BOD 电路上电默认关闭, 需要软件使能才能工作。

BOD 分为中断和复位两种使用方式。中断方式就是在 BOD 变成高电平时触发一次中断, 当 BOD 恢复低电平时不会触发中断。复位方式是通过把 BOD 和 POR 设为关联实现的。BOD 信号是高电平, POR 信号也是高电平, 系统被复位, 包括 BOD 电路; 复位后, BOD 默认不工作, BOD 信号、POR 信号由高变低, 芯片被复位, 直到再次使能 BOD 电路, 并把 BOD 和 POR 设为关联, 系统会再次复位。

BOD 复位会复位寄存器, 但是 ram 数据会被保持, 包络 hbnram、wram、sram。

25.1 简介

SEC ENG 内置了多种运算模块，包括 AES、SHA、CRC、GMAC、PKA、TRNG。

25.2 主要特征

- AES
 - 支持 128 位、192 位和 256 位密钥长度
 - 支持多种链接模式的加密和解密（ECB/CBC/CTR/XTS）
 - 独有的 AES LINK 功能
- SHA
 - 支持 SHA1/SHA224/SHA256/SHA384/SHA512
 - 独有的 SHA LINK 功能
- 支持 MD5, CRC-16, CRC-32

25.3 功能描述

25.3.1 AES 加速器

25.3.1.1 密钥

通过配置寄存器 `se_aes_0_ctrl` 中的 `se_aes_0_mode` 和 `se_aes_0_dec_en` 可以选择加密模式和解密模式时需要的密钥长度。

amode	adec en	Operation
0	0	AES-128 encryption
1	0	AES-256 encryption
2	0	AES-192 encryption
0	1	AES-128 decryption
1	1	AES-256 decryption
2	1	AES-192 decryption

图 25.1: AES 运算模式图

通过配置寄存器 `se_aes_0_ctrl` 中的 `aes_0_hw_key_en` 选择是否开启硬件密钥，如果使用的是软件密钥，则还需要配置寄存器 `se_aes_0_key_0~se_aes_0_key_7` 用于存放密钥，每个寄存器存放 4 个字节的密钥。

25.3.1.2 链接模式

通过配置寄存器 `se_aes_0_ctrl` 中的 `se_aes_0_block_mode` 可以选择不同的链接模式，目前支持 ECB、CBC、CTR、XTS 模式。

25.3.1.3 明文、密文

- 明文或者密文需要是 16 的倍数
- 寄存器 `se_aes_0_msa` 存放加密时输入的明文地址或者解密时输入的密文地址
- 寄存器 `se_aes_0_mda` 存放加密时输出的密文地址或者解密时输出的明文地址
- 寄存器 `se_aes_0_ctrl` 中的 `se_aes_0_msg_len` 用来设定密文或者明文长度（以 16 字节为单位）

25.3.1.4 初始化向量

寄存器 `se_aes_0_iv_0~se_aes_0_iv_3` 存放初始化向量 IV，通过配置 `se_aes_0_ctrl` 中的 `aes_0_iv_sel` 选择是否使用新的 iv，第一次配置 iv 时需要清 0，后续一直使用此 iv 或者自动更新 iv 则需要置 1。

25.3.1.5 加解密配置流程

- 配置寄存器 `se_aes_0_ctrl` 中的 `se_aes_0_en` 使能 AES
- 配置寄存器 `se_aes_0_endian`, 其中包括 `se_aes_0_dout_endian`、`se_aes_0_din_endian`、`se_aes_0_key_endian`、`se_aes_0_iv_endian`、`se_aes_0_twk_endian`, 若值为 0 表示 little-endian, 值为 1 表示 big-endian
- 配置寄存器 `se_aes_0_ctrl` 中的 `se_aes_0_block_mode` 选择链接模式
- 配置寄存器 `se_aes_0_ctrl` 中的 `se_aes_0_mode` 选择密钥长度
- 使用软件密钥则配置寄存器 `se_aes_0_key_0~se_aes_0_key_7` 存放密钥, 使用硬件密钥则置位寄存器 `se_aes_0_ctrl` 中的 `aes_0_hw_key_en`
- 配置寄存器 `se_aes_0_iv_0~se_aes_0_iv_3` 设置 IV, MSB 时填写顺序为 `se_aes_0_iv_0~se_aes_0_iv_3`, LSB 时填写顺序为 `se_aes_0_iv_3~se_aes_0_iv_0`
- 配置寄存器 `se_aes_0_ctrl` 中的 `se_aes_0_dec_en` 选择加密或者解密模式
- 配置寄存器 `se_aes_0_msa` 设置待处理数据的源地址
- 配置寄存器 `se_aes_0_mda` 设置处理结果存放的目的地址
- 配置寄存器 `se_aes_0_ctrl` 中的 `se_aes_0_msg_len` 设置待处理数据的长度, 以 16 字节为单位
- 配置寄存器 `se_aes_0_ctrl` 中的 `se_aes_0_trig_1t` 触发 AES 运行

结果会输出到 `se_aes_0_mda` 指定的目标地址。

25.3.2 SHA 加速器

25.3.2.1 SHA 模式

- 寄存器 `se_sha_0_ctrl` 中的 `se_sha_0_mode` 用于选择 SHA 的具体模式: 0:SHA-256 1:SHA-224 2:SHA-1 3:SHA-14:SHA-512 5:SHA-384 6:SHA-512/224 7:SHA-512/256
- 寄存器 `se_sha_0_ctrl` 中的 `se_sha_0_mode_ext` 由于选择扩展模式, 如果不使用扩展模式应设为 0: 0:SHA 1:MD5 2:CRC-16 3:CRC-32

25.3.2.2 明文、密文

- 寄存器 `se_sha_0_msa` 存放明文地址。
- 寄存器 `se_sha_0_hash_l_0~se_sha_0_hash_l_7` 存放密文。

25.3.2.3 配置流程

- 配置寄存器 `se_sha_0_ctrl` 中的 `se_sha_0_mode` 选择 SHA 的具体模式
- 配置寄存器 `se_sha_0_ctrl` 中的 `se_sha_0_en` 使能 SHA
- 配置寄存器 `se_sha_0_ctrl` 中的 `se_sha_0_hash_sel`, 0 表示开始新的 HASH 计算, 1 表示沿用上一次的结果进行 HASH 计算
- 配置寄存器 `se_sha_0_msa` 设置输入数据的地址
- 配置寄存器 `se_sha_0_ctrl` 中的 `se_sha_0_msg_len` 待处理数据的长度, SHA-1、SHA-224、SHA-256 以 512-bit 为单位, SHA-512、SHA-384、SHA-512/224、SHA-512/256 以 1024-bit 为单位
- 配置寄存器 `se_sha_0_ctrl` 中的 `se_sha_0_trig_1t` 触发 SHA 运行
- 结果输出存储在 `se_sha_0_hash_l_0~se_sha_0_hash_l_7` 中 (默认 MSB), MSB:`se_sha_0_hash_l_0~se_sha_0_hash_l_7`, LSB:`se_sha_0_hash_l_7~se_sha_0_hash_l_0`

25.3.3 TRNG

SEC ENG 内置一个 TRNG(真随机数发生器), 其生成的随机数可作为加密等操作的基础。

- 真随机数: 真正的随机数是使用物理现象产生的: 比如掷钱币、骰子、转轮、使用电子元件的噪音、核裂变等等, 这样的随机数发生器叫做物理性随机数发生器, 它们的缺点是技术要求比较高。
- 伪随机数: 真正意义上的随机数 (或者随机事件) 在某次产生过程中是按照实验过程中表现的分布概率随机产生的, 其结果是不可预测的, 是不可见的。而计算机中的随机函数是按照一定算法模拟产生的, 其结果是确定的, 是可见的。我们可以认为这个可预见的结果其出现的概率是 100%。所以用计算机随机函数所产生的“随机数”并不随机, 是伪随机数。

25.3.3.1 配置流程

- 配置寄存器 `se_trng_0_ctrl_0` 中的 `se_trng_0_en` 使能 TRNG
- 配置寄存器 `se_trng_0_ctrl_0` 中的 `se_trng_0_trig_1t` 触发 TRNG 运行
- 结果输出存储在 `se_trng_0_dout_0~se_trng_0_dout_7` 中

表 26.1: 文档版本修改信息

日期	版本	修改内容
2021/9/22	0.9	初版
2022/8/22	0.91	添加寄存器描述
2022/11/28	0.92	增加 Non-cache 和 cache 对应的内存映射地址
2022/12/22	0.93	修改 Reset table, 增加 ACOMP 描述
2023/3/30	0.94	更新 I2C 时序框图, 修正笔误
2023/4/28	0.95	更新时钟树框图
2023/5/23	0.96	Reset table 中增加 CHIP_EN 复位功能
2024/3/5	0.97	修改 EMAC 特性
2025/1/15	0.98	增加 BOOT 流程介绍