

JMT1801ED MCU

用户手册

江苏宏云技术有限公司

www.macrocloudtec.com

0512-58191012

1	产品概述	17
1.1	功能描述	17
1.2	主要特性	17
1.3	框图	19
1.4	应用范围	19
1.5	引脚配置	20
1.5.1	LQFP48L 引脚图	20
1.5.2	QFN40L 引脚图	21
1.5.3	QFN32L 引脚图	22
1.5.4	TSSOP24L 引脚图	23
1.5.5	TSSOP16L 引脚图	24
1.5.6	引脚说明	24
2	电源管理 (PMU)	26
2.1	概述	26
2.2	低功耗控制	27
2.2.1	降低系统时钟频率	28
2.2.2	外设时钟门控	28
2.2.3	关闭模拟模块	28
2.2.4	四种低功耗模式	28
2.2.5	OSC 开关说明	30
2.3	寄存器	31
2.3.1	低功耗控制寄存器(PMUCTR) [0xA1]	31
3	时钟复位管理 (CRM)	32
3.1	时钟管理	32
3.1.1	概述	32
3.1.2	框图	32
3.1.3	功能描述	32
3.1.4	寄存器描述	37
3.2	复位管理	45
3.2.1	概述	45
3.2.2	框图	45
3.2.3	功能描述	46

3.2.4	寄存器.....	49
4	JMT51 MCU 核.....	54
4.1	概述.....	54
4.2	指令集.....	54
5	JMT018 DSP 核.....	59
5.1	概述.....	59
5.1.1	框图.....	60
5.1.2	系统总线.....	60
5.1.3	运算单元 (ALU)	61
5.1.4	乘累加单元(MAC).....	61
5.1.5	除法单元 (DIV)	61
5.1.6	硬件循环 (HW loop)	61
5.1.7	数据传输.....	61
5.1.8	存储单元.....	62
5.2	指令集.....	62
5.2.1	数据传输指令.....	62
5.2.2	算术运算指令.....	64
5.2.3	移位运算指令.....	65
5.2.4	逻辑运算指令.....	66
5.2.5	乘法运算指令.....	66
5.2.6	并行运算指令.....	67
5.2.7	程序控制指令.....	70
5.3	数据运算器.....	72
5.3.1	通用寄存器.....	72
5.3.2	累加器 A.....	72
5.3.3	乘法.....	76
5.3.4	乘加运算.....	78
5.3.5	加减运算.....	80
5.3.6	移位运算.....	80
5.3.7	四舍五入方法.....	81
5.4	DSP 存储空间.....	81
5.4.1	数据存储空间.....	81
5.4.2	程序存储空间.....	81

5.5	DSP 工作配置.....	82
5.6	DSP 代码生成与下载.....	83
5.6.1	DSP 代码生成	83
5.6.2	DSP 代码下载	83
5.7	DSP 内部寄存器.....	83
5.7.1	DSP 通用寄存器 R0.....	84
5.7.2	DSP 通用寄存器 R1.....	84
5.7.3	DSP 通用寄存器 R2.....	84
5.7.4	DSP 通用寄存器 R3.....	84
5.7.5	DSP 通用寄存器 R4.....	84
5.7.6	DSP 通用寄存器 R5.....	84
5.7.7	DSP 通用寄存器 R6.....	85
5.7.8	DSP 通用寄存器 R7.....	85
5.7.9	X 数据空间指针寄存器 DP0.....	85
5.7.10	Y 数据空间指针寄存器 DP1.....	85
5.7.11	DSP 程序地址指针寄存器 PC.....	85
5.7.12	乘法及乘加结果寄存器 A.....	85
5.7.13	LINK 寄存器	85
5.7.14	状态寄存器 PSR.....	86
5.7.15	模式寄存器 PMR	86
5.7.16	硬件循环次数寄存器 LOOP	87
5.7.17	硬件循环指令个数寄存器 LIR	87
5.8	JMT51 访问 DSP 寄存器	87
5.8.1	DSP 配置寄存器(DSP_CFG) [0x8C].....	88
5.8.2	DSP 运行状态寄存器(DSP_STA) [0x89].....	89
5.8.3	PC 寄存器低字节(DSP_PCL) [0x8A]	89
5.8.4	PC 寄存器高字节(DSP_PCH) [0x8B].....	89
5.8.5	R0 寄存器低字节(DSP_R0L) [0x91]	89
5.8.6	R0 寄存器高字节(DSP_R0H) [0x92]	89
5.8.7	R1 寄存器低字节(DSP_R1L) [0x93]	90
5.8.8	R1 寄存器高字节(DSP_R1H) [0x94]	90
5.8.9	R2 寄存器低字节(DSP_R2L) [0x95]	90
5.8.10	R2 寄存器高字节(DSP_R2H) [0x96]	90
5.8.11	R3 寄存器低字节(DSP_R3L) [0x97]	90

5.8.12	R3 寄存器高字节(DSP_R3H) [0x99]	90
5.8.13	R4 寄存器低字节(DSP_R4L) [0x9A]	90
5.8.14	R4 寄存器高字节(DSP_R4H) [0x9B]	91
5.8.15	R5 寄存器低字节(DSP_R5L) [0x9C]	91
5.8.16	R5 寄存器高字节(DSP_R5H) [0x9D]	91
5.8.17	R6 寄存器低字节(DSP_R6L) [0x9E]	91
5.8.18	R6 寄存器高字节(DSP_R6H) [0x9F]	91
5.8.19	R7 寄存器低字节(DSP_R7L) [0xA1]	91
5.8.20	R7 寄存器高字节(DSP_R7H) [0xA2]	91
5.8.21	DP0 寄存器低字节(DSP_DP0L) [0xA3]	92
5.8.22	DP0 寄存器高字节(DSP_DP0H) [0xA4]	92
5.8.23	DP1 寄存器低字节(DSP_DP1L) [0xA5]	92
5.8.24	DP1 寄存器高字节(DSP_DP1H) [0xA6]	92
5.8.25	A 寄存器 7~0 比特(DSP_A0L) [0xA7]	92
5.8.26	A 寄存器 15~8 比特(DSP_A0H) [0xA9]	92
5.8.27	A 寄存器 23~16 比特(DSP_A1L) [0xAA]	92
5.8.28	A 寄存器 31~24 比特(DSP_A1H) [0xAB]	93
5.8.29	A 寄存器 39~32 比特(DSP_A2) [0xAC]	93
5.8.30	LNKR 寄存器低字节(DSP_LNKRL) [0xAD]	93
5.8.31	LNKR 寄存器高字节(DSP_LNKRH) [0xAE]	93
5.8.32	PSR 寄存器低字节(DSP_PSRL) [0xAF]	93
5.8.33	PSR 寄存器高字节(DSP_PSRH) [0xB1]	94
5.8.34	PMR 寄存器低字节(DSP_PMRL) [0xB2]	94
5.8.35	PMR 寄存器高字节(DSP_PMRH) [0xB3]	94
5.8.36	LOOP 寄存器低字节(DSP_LOOPL) [0xB4]	94
5.8.37	LOOP 寄存器高字节(DSP_LOOPH) [0xB5]	95
5.8.38	LIR 寄存器低字节(DSP_LIRL) [0xB6]	95
5.8.39	LIR 寄存器高字节(DSP_LIRH) [0xB7]	95
6	中断	96
6.1	概述	96
6.2	外部管脚中断 (EXINT)	97
6.3	中断优先级	97
6.4	中断处理过程	98

6.5	中断响应时间.....	98
6.6	中断寄存器.....	99
6.6.1	中断使能寄存器 0(IEN0) [0xA8].....	99
6.6.2	中断使能寄存器 1(IEN1) [0xB8].....	100
6.6.3	中断使能寄存器 2 (IEN2) [0xC0].....	100
6.6.4	中断优先级寄存器 0 (IP0) [0xB9].....	101
6.6.5	中断优先级寄存器 1(IP1) [0xBA].....	101
6.6.6	中断优先级寄存器 2(IP2) [0xBB].....	102
7	存储器.....	102
7.1	JMT51 存储器.....	103
7.1.1	程序存储器.....	103
7.1.2	外部数据存储器.....	104
7.1.3	内部数据存储器.....	104
7.1.4	特殊功能寄存器.....	106
7.2	JMT018 存储器.....	110
7.3	存储器访问冲突说明.....	111
7.4	JMT51 核寄存器.....	111
7.4.1	ACC 寄存器(A) [0xE0].....	111
7.4.2	B 寄存器(B) [0xF0].....	112
7.4.3	堆栈指针(SP) [0x81].....	112
7.4.4	程序状态字寄存器(PSW) [0xD0].....	112
7.4.5	数据指针低字节(DPL) [0x82].....	113
7.4.6	数据指针高字节(DPH) [0x83].....	113
7.4.7	MOVX 指令设置寄存器(MOVXCON) [0x86].....	114
7.5	PAGE 堆栈寄存器.....	114
7.5.1	SFR PAGE 寄存器(SFRPAGE) [0x84].....	115
7.5.2	PAGE 堆栈使能寄存器(SFRPGEN) [0x85].....	115
7.5.3	PAGE 堆栈寄存器 0(PGSTACK0) [0xF9].....	115
7.5.4	PAGE 堆栈寄存器 1(PGSTACK1) [0xFA].....	115
7.5.5	PAGE 堆栈寄存器 2(PGSTACK2) [0xFB].....	116
7.5.6	PAGE 堆栈寄存器 3(PGSTACK3) [0xFC].....	116
8	Flash 存储器.....	117
8.1	概述.....	117

8.2	FLASH 操作.....	117
8.2.1	FLASH 数据读取.....	117
8.2.2	FLASH 扇区擦除.....	118
8.2.3	FLASH 编程.....	118
8.3	FLASH 寄存器.....	119
8.3.1	FLASH 控制寄存器(FLSC) [0xF9].....	119
8.3.2	FLASH 关键字寄存器(FLSK) [0xFA].....	119
9	看门狗定时器 (WDT)	120
9.1	概述.....	120
9.2	功能描述.....	120
9.3	寄存器.....	120
9.3.1	看门狗分频系数寄存器(WDTCKDIV) [0xD9].....	120
9.3.2	看门狗比较值寄存器(WDTINT) [0xDA].....	121
9.3.3	看门狗中断标志寄存器(WDTINTF) [0xDB].....	121
9.3.4	看门狗喂狗寄存器(WDTLD) [0xDC].....	122
9.3.5	看门狗启停寄存器(WDTST) [0xDD].....	122
10	定时器/计数器 (TIMER)	123
10.1	概述.....	123
10.2	TIMER 管脚配置.....	123
10.3	Timer0 工作模式.....	123
10.3.1	模式 0(13 位定时器/计数器).....	123
10.3.2	模式 1(16 位定时器/计数器).....	124
10.3.3	模式 2(8 位自动重装定时器/计数器).....	124
10.3.4	模式 3(两个 8 比特定器).....	125
10.4	Timer1 工作模式.....	126
10.4.1	模式 0(13 位定时器/计数器).....	126
10.4.2	模式 1(16 位定时器/计数器).....	126
10.4.3	模式 2(8 位自动重装定时器/计数器).....	127
10.5	Timer2 工作模式.....	127
10.5.1	模式 0(13 位定时器/计数器).....	127
10.5.2	模式 1(16 位定时器/计数器).....	128
10.5.3	模式 2(16 位自动重装定时器/计数器).....	128
10.5.4	模式 3(1 个 8 比特定器/计数器).....	129

10.6	TIMER 寄存器.....	129
10.6.1	Timer0/1 控制寄存器(TCON)[0x88].....	130
10.6.2	Timer0/1 模式寄存器(TMOD)[0x89].....	130
10.6.3	Timer0 低 8 位(TL0)[0x8A].....	131
10.6.4	Timer1 低 8 位(TL1)[0x8B].....	132
10.6.5	Timer0 高 8 位(TH0)[0x8C].....	132
10.6.6	Timer1 高 8 位(TH1)[0x8D].....	132
10.6.7	Timer0/1 预分频控制器(TPSC)[0x8E].....	132
10.6.8	Timer2 控制寄存器(T2CON)[0xD1].....	132
10.6.9	Timer2 重载寄存器低 8 位(RL2)[0xD2].....	133
10.6.10	Timer2 重载寄存器高 8 位(RH2)[0xD3].....	133
10.6.11	Timer2 低 8 位(TL2)[0xD4].....	133
10.6.12	Timer2 高 8 位(TH2)[0xD5].....	134
10.6.13	Timer2 预分频控制器(T2PSC)[0xD6].....	134
11	高级定时器 (PWM)	135
11.1	概述.....	135
11.2	特性.....	135
11.3	框图.....	136
11.4	输入输出引脚配置.....	136
11.5	功能描述.....	137
11.5.1	计数器.....	137
11.5.2	输入捕获功能.....	149
11.5.3	从模式控制.....	150
11.5.4	输出比较功能.....	155
11.5.5	主模式控制.....	175
11.5.6	中断.....	177
11.6	寄存器描述.....	178
11.6.1	TIM 控制寄存器 0 (TIM_CONR0) [0xD1].....	180
11.6.2	TIM 控制寄存器 1 (TIM_CONR1) [0xD2].....	182
11.6.3	TIM 控制寄存器 2 (TIM_CONR2) [0xD3].....	183
11.6.4	TIM 控制寄存器 3 (TIM_CONR3) [0xD4].....	184
11.6.5	从模式控制寄存器 0 (TIM_TGICR0) [0xD5].....	186
11.6.6	从模式控制寄存器 1 (TIM_TGICR1) [0xD6].....	187
11.6.7	TIM 中断使能寄存器 0 (TIM_IER) [0xD7].....	188

11.6.8	TIM 通道 CC0 输入捕获/输出比较模式控制寄存器 (TIM_CC0MR) [0xD9]	189
11.6.9	TIM 通道 CC1 输入捕获/输出比较模式控制寄存器 (TIM_CC1MR) [0xDA]	193
11.6.10	TIM 通道 CC2 输入捕获/输出比较模式控制寄存器 (TIM_CC2MR) [0xDB]	197
11.6.11	TIM 通道 CC3 输入捕获/输出比较模式控制寄存器 (TIM_CC3MR) [0xDC]	202
11.6.12	通道使能控制寄存器 (TIM_CCENR) [0xDD]	206
11.6.13	通道极性控制寄存器 (TIM_CCPS) [0xDE]	207
11.6.14	TIM 死区时间寄存器 (TIM_DTG) [0xDF]	208
11.6.15	TIM 周期值自动装载寄存器的低 8 位 (TIM_ARRL) [0xE1]	209
11.6.16	TIM 周期值自动装载寄存器的高 8 位 (TIM_ARRH) [0xE2]	209
11.6.17	TIM 预分频寄存器的低 8 位 (TIM_PSCL) [0xE3]	209
11.6.18	TIM 预分频寄存器的高 8 位 (TIM_PSCH) [0xE4]	210
11.6.19	TIM 重复计数寄存器 (TIM_RCR) [0xE5]	210
11.6.20	通道 CC0 捕获/比较寄存器的低 8 位 (TIM_CC0RL) [0xE6]	211
11.6.21	通道 CC0 捕获/比较寄存器的高 8 位 (TIM_CC0RH) [0xE7]	211
11.6.22	通道 CC1 捕获/比较寄存器的低 8 位 (TIM_CC1RL) [0xE9]	212
11.6.23	通道 CC1 捕获/比较寄存器的高 8 位 (TIM_CC1RH) [0xEA]	212
11.6.24	通道 CC2 捕获/比较寄存器的低 8 位 (TIM_CC2RL) [0xEB]	213
11.6.25	通道 CC2 捕获/比较寄存器的高 8 位 (TIM_CC2RH) [0xEC]	213
11.6.26	通道 CC3 捕获/比较寄存器的低 8 位 (TIM_CC3RUL) [0xED]	214
11.6.27	通道 CC3 捕获/比较寄存器的高 8 位 (TIM_CC3RH) [0xEE]	215
11.6.28	TIM 刹车控制寄存器 (TIM_BRKC) [0xEF]	215
11.6.29	TIM 事件产生寄存器 (TIM_EGR) [0xF1]	218
11.6.30	相位偏移控制寄存器 (TIM_PHACON) [0XF2]	219
11.6.31	TIM 中断标志寄存器 0 (TIM_ISR0) [0xF3]	219
11.6.32	TIM 中断标志寄存器 1 (TIM_ISR1) [0xF4]	221
11.6.33	TIM 计数器的低 8 位 (TIM_CNTH) [0xF5]	221
11.6.34	TIM 计数器的高 8 位 (TIM_CNTH) [0xF6]	221
11.6.35	相位偏移长度寄存器的低 8 位 (TIM_PHASEL) [0XF7]	222
11.6.36	相位偏移长度寄存器的高 8 位 (TIM_PHASEL) [0XF9]	222
11.6.37	TIM 计数器使能寄存器 (TIM_CNTEN) [0xFA]	222
12	CORDIC 协处理器	223
12.1	概述	223

12.2	功能描述.....	224
12.2.1	输入数据格式.....	224
12.2.2	结果数据归一化.....	224
12.2.3	CORDIC 调度.....	225
12.2.4	CORDIC 中断.....	226
12.2.5	CORDIC 精度.....	226
12.3	CORDIC 寄存器.....	227
12.3.1	JMT51 MCU 调度寄存器.....	227
12.3.2	JMT018 DSP 指令调度寄存器.....	230
13	通用异步收发传输器 (UART)	233
13.1	概述.....	233
13.2	框图.....	233
13.3	UART 管脚配置.....	234
13.4	UART 工作模式.....	234
13.4.1	标准 UART 工作模式 0: 同步移位寄存器.....	234
13.4.2	标准 UART 工作模式 1: 8 位 UART, 波特率可变.....	235
13.4.3	标准 UART 工作模式 2: 9 位 UART, 波特率固定.....	236
13.4.4	标准 UART 工作模式 3: 9 位 UART, 波特率可变.....	236
13.4.5	38KHz 红外调制发送.....	237
13.4.6	IrDA(SIR)红外通信.....	238
13.5	UART 多机通信.....	239
13.6	UART 奇偶校验.....	240
13.7	UART 波特率计算.....	240
13.8	UART 寄存器.....	244
13.8.1	UART 控制寄存器(SCON) [0x98].....	245
13.8.2	UART 数据缓冲寄存器(SBUF) [0x99].....	246
13.8.3	UART 波特率重载寄存器低 8 位(SRELL) [0x9A].....	246
13.8.4	UART 波特率重载寄存器高 2 位(SRELH) [0x9B].....	246
13.8.5	UART 红外控制寄存器(SIRCON) [0x9C].....	246
14	I2C 接口.....	248
14.1	概述.....	248
14.2	框图.....	248
14.3	I2C 管脚配置.....	248

14.4	I2C 模块连接.....	248
14.5	I2C 数据格式.....	249
14.5.1	数据有效.....	249
14.5.2	读写控制比特.....	249
14.5.3	反馈 (ACK)	250
14.5.4	起始条件和结束条件.....	250
14.5.5	数据格式.....	250
14.5.6	时钟同步.....	251
14.5.7	仲裁.....	251
14.6	I2C 时钟产生.....	252
14.7	I2C 工作模式.....	253
14.7.1	从机模式.....	253
14.7.2	主机模式.....	255
14.8	I2C 中断.....	258
14.9	I2C 寄存器.....	259
14.9.1	I2C 控制寄存器(I2CCON) [0xE1].....	259
14.9.2	I2C 从机地址寄存器低 8 位(I2CSADDRL) [0xE2]	260
14.9.3	I2C 从机地址寄存器高 2 位(I2CSADDRH) [0xE3].....	260
14.9.4	I2C 本机地址寄存器低 8 位(I2COADDRL) [0xE4].....	261
14.9.5	I2C 本机地址寄存器高 2 位(I2COADDRH) [0xE5]	261
14.9.6	I2C 工作时钟分频控制寄存器(I2CDIV) [0xE6]	261
14.9.7	I2C 时钟 SCL 低电平时间配置寄存器(I2CDUTYL) [0xE7].....	262
14.9.8	I2C 时钟 SCL 高电平时间配置寄存器(I2CDUTYH) [0xE9]	262
14.9.9	I2C 数据 SDA 保持时间配置寄存器(I2CHOLD) [0xEA]	262
14.9.10	I2C 数据写缓冲寄存器(I2CWBUF) [0xEB].....	263
14.9.11	I2C 数据读缓冲寄存器(I2CRBUF) [0xEC].....	263
14.9.12	I2C 状态寄存器(I2CSTS) [0xED]	263
14.9.13	I2C 中断标志寄存器(I2CISC) [0xEE].....	264
14.9.14	I2C 中断使能寄存器(I2CIEN) [0xEF]	265
15	串行外设接口 (SPI)	266
15.1	概述.....	266
15.2	框图.....	266
15.3	SPI 管脚配置.....	266

15.4	SPI 主/从机互连.....	267
15.4.1	4 线互连模式.....	267
15.4.2	3 线互连模式.....	267
15.5	SPI 时序设置.....	268
15.6	数据发送和接收过程.....	269
15.6.1	主机模式.....	269
15.6.2	从机模式.....	271
15.7	SPI 中断.....	273
15.8	SPI 寄存器.....	273
15.8.1	SPI 控制寄存器(SPICON)[0xC9].....	273
15.8.2	SPI 中断使能寄存器(SPIIE)[0xCA].....	275
15.8.3	SPI 中断标志寄存器(SPIIF)[0xCB].....	275
15.8.4	SPI 波特率控制寄存器(SPIBR)[0xCC].....	276
15.8.5	SPI 数据缓冲寄存器(SPIBUF) [0xCD].....	276
15.8.6	SPI 主机采样控制寄存器(SPISPC)[0xCE].....	276
16	模拟/数字转换 (ADC) 和可编程增益放大器 (PGA)	278
16.1	概述.....	278
16.2	ADC 转换结果访问方式.....	278
16.3	ADC 输入端口配置.....	279
16.4	可编程增益放大器 (PGA) 说明.....	279
16.5	ADC 参考电压配置.....	282
16.6	ADC 转换功能.....	282
16.7	ADC 转换结果数据格式.....	282
16.8	ADC 转换时序.....	283
16.9	触发事件.....	283
16.9.1	软件触发事件.....	284
16.9.2	外部触发事件.....	284
16.10	ADC 操作流程.....	285
16.11	ADC 寄存器.....	285
16.11.1	JMT51 访问 ADC 寄存器.....	285
16.11.2	JMT018 访问 ADC 寄存器.....	290
17	模拟比较器 (CMP)	291

17.1	模拟比较器概述	291
17.2	低压检测	292
17.3	模拟比较器寄存器	293
17.3.1	模拟比较器使能寄存器(CMPEN) [0xB9]	294
17.3.2	模拟比较器滤波选择寄存器(CMPFT) [0xBA]	294
17.3.3	模拟比较器 VDD 分压寄存器(CMPVDD) [0xBB]	294
17.3.4	模拟比较器参考电压选择寄存器(CMPREF) [0xBC]	295
17.3.5	模拟比较器中断寄存器(CMPINT) [0xBD]	295
17.3.6	模拟比较器比较结果寄存器(CMPFLAG) [0xBE]	295
17.3.7	OC 控制寄存器(OCPRSEL) [0xC1]	296
17.3.8	OD 控制寄存器(ODPRSEL) [0xC2]	296
17.3.9	PWM 刹车控制寄存器(BKPRSEL) [0xC3]	296
18	通用输入/输出 (GPIO)	298
18.1	管脚设置	298
18.2	管脚复用	301
18.3	复用功能重映射	303
18.4	管脚数据寄存器读写	304
18.5	外设管脚复用配置	305
18.6	管脚滤波	309
18.7	EXINT 中断	310
18.8	PA 口唤醒	313
18.9	GPIO 寄存器	314
18.9.1	PA0 控制寄存器(PA0CTRL) [0xB9]	316
18.9.2	PA1 控制寄存器(PA1CTRL) [0xBA]	316
18.9.3	PA2 控制寄存器(PA2CTRL) [0xBB]	317
18.9.4	PA3 控制寄存器(PA3CTRL) [0xBC]	318
18.9.5	PA4 控制寄存器(PA4CTRL) [0xBD]	319
18.9.6	PA5 控制寄存器(PA5CTRL) [0xBE]	320
18.9.7	PA6 控制寄存器(PA6CTRL) [0xBF]	321
18.9.8	PA7 控制寄存器(PA7CTRL) [0xC1]	322
18.9.9	PB0 控制寄存器(PB0CTRL) [0x89]	323
18.9.10	PB1 控制寄存器(PB1CTRL) [0x8A]	323
18.9.11	PB2 控制寄存器(PB2CTRL) [0x8B]	324

18.9.12	PB3 控制寄存器(PB3CTRL) [0x8C].....	325
18.9.13	PB4 控制寄存器(PB4CTRL) [0x8D].....	326
18.9.14	PB5 控制寄存器(PB5CTRL) [0x8E].....	326
18.9.15	PB6 控制寄存器(PB6CTRL) [0x8F].....	327
18.9.16	PB7 控制寄存器(PB7CTRL) [0xA9].....	328
18.9.17	PC0 控制寄存器(PC0CTRL) [0x91].....	329
18.9.18	PC1 控制寄存器(PC1CTRL) [0x92].....	329
18.9.19	PC2 控制寄存器(PC2CTRL) [0x93].....	330
18.9.20	PC3 控制寄存器(PC3CTRL) [0x94].....	331
18.9.21	PC4 控制寄存器(PC4CTRL) [0x95].....	332
18.9.22	PC5 控制寄存器(PC5CTRL) [0x96].....	333
18.9.23	PC6 控制寄存器(PC6CTRL) [0x97].....	334
18.9.24	PC7 控制寄存器(PC7CTRL) [0xAA].....	335
18.9.25	PD0 控制寄存器(PD0CTRL) [0x99].....	336
18.9.26	PD1 控制寄存器(PD1CTRL) [0x9A].....	336
18.9.27	PD2 控制寄存器(PD2CTRL) [0x9B].....	337
18.9.28	PD3 控制寄存器(PD3CTRL) [0x9C].....	338
18.9.29	PD4 控制寄存器(PD4CTRL) [0x9D].....	339
18.9.30	PD5 控制寄存器(PD5CTRL) [0x9E].....	339
18.9.31	PD6 控制寄存器(PD6CTRL) [0x9F].....	340
18.9.32	PD7 控制寄存器(PD7CTRL) [0xAB].....	341
18.9.33	PE0 控制寄存器(PE0CTRL) [0xA1].....	342
18.9.34	PE1 控制寄存器(PE1CTRL) [0xA2].....	342
18.9.35	PE2 控制寄存器(PE2CTRL) [0xA3].....	343
18.9.36	PE3 控制寄存器(PE3CTRL) [0xA4].....	344
18.9.37	PE4 控制寄存器(PE4CTRL) [0xA5].....	345
18.9.38	PE5 控制寄存器(PE5CTRL) [0xA6].....	345
18.9.39	PE6 控制寄存器(PE6CTRL) [0xA7].....	346
18.9.40	PE7 控制寄存器(PE7CTRL) [0xAC].....	347
18.9.41	PAD 转换速率和驱动能力寄存器 0(PADSRDR0) [0xAD].....	348
18.9.42	PAD 转换速率和驱动能力寄存器 1(PADSRDR1) [0xAE].....	348
18.9.43	复用功能重映射寄存器 0(AFREMAP0) [0xB6].....	349
18.9.44	复用功能重映射寄存器 1(AFREMAP1) [0xC3].....	350
18.9.45	滤波时钟预分频寄存器(FPSC) [0xB7].....	351
18.9.46	复位管脚滤波选择寄存器(NRSTFQS) [0xC2].....	351

18.9.47	EXINT 上升沿中断使能寄存器(EXINTPE) [0xC4]	351
18.9.48	EXINT 下降沿中断使能寄存器(EXINTNE) [0xC5]	352
18.9.49	EXINT 中断使能寄存器(EXINTEN) [0xC6]	353
18.9.50	EXINT 中断控制寄存器(EXINTCON) [0xB5]	354
18.9.51	EXINT 中断管脚选择寄存器 0(EXINTSEL0) [0xB1]	355
18.9.52	EXINT 中断管脚选择寄存器 1(EXINTSEL1) [0xB2]	356
18.9.53	EXINT 中断管脚选择寄存器 2(EXINTSEL2) [0xB3]	356
18.9.54	EXINT 中断管脚选择寄存器 3(EXINTSEL3) [0xB4]	357
18.9.55	PA 数据寄存器(P0) [0x80]	358
18.9.56	PB 数据寄存器(P1) [0x90]	358
18.9.57	PC 数据寄存器(P2) [0xA0]	358
18.9.58	PD 数据寄存器(P3) [0xB0]	358
18.9.59	PE 数据寄存器(P4) [0xC8]	358
19	电气特性	359
19.1	极限参数	359
19.2	工作条件	359
19.2.1	芯片供电	359
19.2.2	功耗特性	359
19.2.3	I/O 特性	360
19.2.4	REGC 引脚特性	361
19.2.5	NRST 引脚特性	361
19.2.6	上电/掉电条件	361
19.2.7	时钟特性	362
19.2.8	通信接口	362
19.2.9	FLASH 特性	365
19.2.10	ADC 特性	366
19.2.11	PGA 特性	366
19.2.12	CMP 特性	367
19.2.13	电气敏感性	367
20	封装特性	369
20.1	LQFP48L	369
20.2	QFN40L	370
20.3	QFN32L	370

20.4	TSSOP24L	371
20.5	TSSOP16L	371

IRONCHIP

IRONCHIP

1 产品概述

1.1 功能描述

JMT1801ED 是一款内置 MCU 和 DSP 的双核 SOC 芯片。JMT1801ED 内置 MCU 为兼容 8051 的 JMT51 核，内置 DSP 为具有 132 条指令的 16 位定点 JMT018 核，MCU 和 DSP 的最高运行频率为 73.728MHz。JMT1801ED 集成了：32K 字节 Flash、1.5K 字节 SRAM、1 个高性能 CORDIC（坐标旋转矢量运算）单元、1 个高性能 16 位 PWM 模块（具有 3 路互补加 1 路独立 PWM）、1 个独立的 16 通道 12 位 ADC、1 个高增益的可编程放大器(PGA)、1 个模拟比较器 (CMP)、3 个 Timer、1 个 WDT、1 个 SPI、1 个 I2C、1 个 UART 通信接口、1 个蜂鸣器输出。

1.2 主要特性

JMT51 MCU 内核:

- 8 位 MCU，兼容 MCS51 指令集
- 1T 指令周期
- 内核运行最高频率 73.728MHz
- 支持 JTAG 调试

JMT018 DSP 内核:

- 16 位 DSP 内核，支持 132 条指令
- 内核运行最高频率 73.728MHz
- 程序空间:
 - 32K 字节 FLASH（和 JMT51 核共享）
- 数据空间:
 - 64x16bit SRAM（DM0）
 - 64x16bit SRAM（DM1）

时钟:

- 内置高精度 73.728MHz RC 振荡器，精度为 $\pm 1\%$ @5v, 25°C
- 内置 32KHz RC 振荡器，精度为 $\pm 10\%$ @5v, 25°C

外设特性:

- 上电复位、掉电复位、低压复位

- 3 个 Timer、1 个 WDT
- 40 个双向 I/O
 - 8 个具有唤醒功能的输入口
 - 所有 I/O 都可作为外部中断源
- 内置 32K 字节 Flash
 - 擦写次数：至少 20,000 次
 - 保持时间：至少 100 年
- 内置 1.5K 字节数据存储器(SRAM)
 - 128x8bit 内部数据存储器
 - 1024x8bit DM2
 - 64x16bit DM0(MCU 和 DSP 共享)
 - 64x16bit DM1(MCU 和 DSP 共享)
- 1 个 CORDIC 单元
 - 支持正弦、余弦计算
 - 支持反正切计算
 - 支持矢量模计算
 - 可作为 DSP 指令集使用
 - 可作为 JMT51 加速器使用

- 1 个 PWM 模块
 - 具有 3 路互补加 1 路独立 PWM
 - PWM 为 16 位计数器
 - 最高运行频率 73.728MHz
 - 支持内部、外部刹车(软硬件)
 - 支持比较器刹车
 - 支持死区可编程
 - 可以触发 ADC 采样
 - 支持正交解码
 - 支持霍尔传感器解码
 - 支持捕获模式
 - 支持各路 PWM 输出的相位偏移
 - 1 个独立 16 通道 12 位 ADC
 - 支持 16 通道输入
 - 支持外部管脚/PWM 触发
 - 采样时间延时可编程
 - 1 个 UART
 - 支持标准 UART 通信协议
 - 支持 IrDA (SIR 标准)
 - 支持输出 38kHz 红外调制
 - 1 个 SPI 接口总线
 - 支持 4 线全双工传输
 - 支持 3 线半双工传输
 - 8 位数据帧格式
 - 数据 MSB 在前和 LSB 在前可选择
 - 支持主机或从机模式
 - 1 个高倍数可编程增益放大器 (PGA)
 - 支持 1/2、1、2、4、8、16 放大倍数可配置
 - 支持同相、反相极性
 - 支持输入端四路可选
 - 1 个 I2C 接口总线
 - 支持标准模式: 100Kbps
 - 支持快速模式: 400Kbps
 - 支持 master/slave 工作模式
 - 支持单 master/多 master 操作
 - 支持 7/10 位设备地址寻址
 - 1 个蜂鸣器输出:
 - 蜂鸣器输出信号的频率和占空比可编程控制
 - 1 个模拟比较器
 - 支持低压检测功能
 - 比较器输入可选择: PA0/PGA 输出/VDDA
- 低功耗特性:**
- 芯片工作电压
 - V_{DD} : 2.75V~5.5V
 - 支持 5 种工作模式
 - Normal、Idle、Stop、Sleep 和 Deep Sleep
- 封装:**
- LQFP48L
 - QFN40L、QFN32L
 - TSSOP24L、TSSOP16L

1.3 框图

JMT1801ED 框图如 Figure 1 所示:

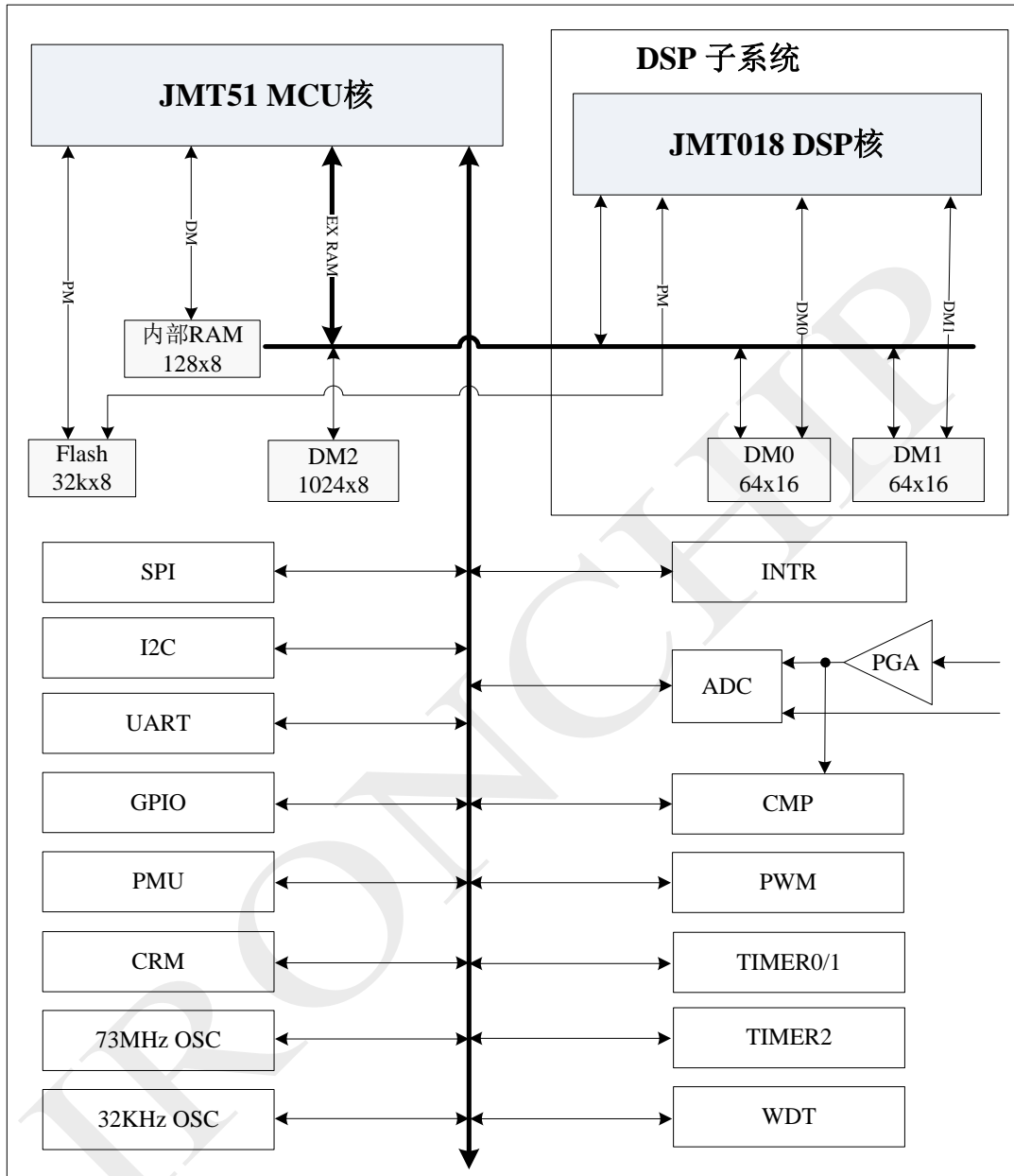


Figure 1 JMT1801ED 结构框图

1.4 应用范围

JMT1801ED 可应用于电机控制、太阳能逆变器、UPS 电源控制器等产品中。

JMT1801ED 支持有霍尔或者无霍尔直流无刷电机 (BLDC) 控制, 支持有霍尔或者无霍尔永磁同步电机 (PMSM) 控制。可使用在电动车、无人机、平衡车、跑步机和家用电器如变频空调、变频洗衣机、扫地机器人、空气净化器等产品中具有 PWM、SPWM、FOC-SVPWM 等控制模式的产品中。

JMT1801ED 支持 SPWM、SVPWM 逆变器。

JMT1801ED 支持 SPWM、SVPWM UPS。

JMT1801ED 作为永磁电机控制器的应用参见 Figure 2 所示，由 DSP 完成 FOC-SVPWM 算法，六路互补 PWM 驱动电机，ADC 完成电流、电压采样，ADC 配置有 4 路 PGA 进行放大，比较器（CMP）可以用作短路保护。

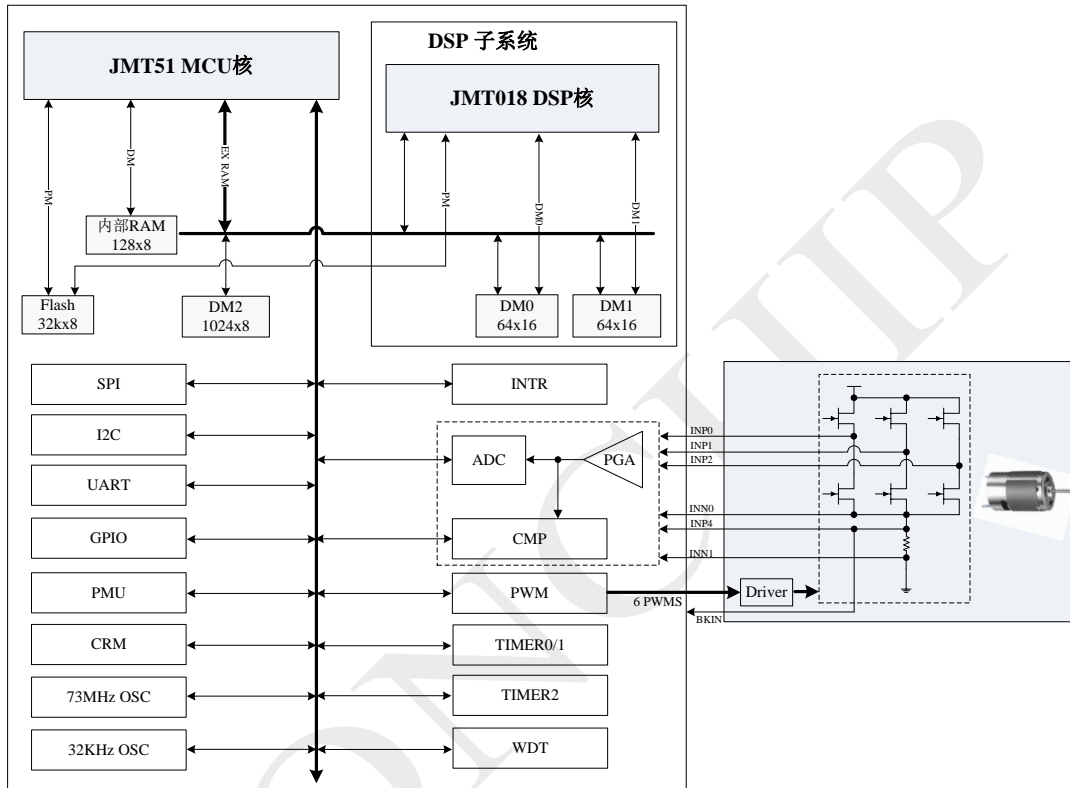


Figure 2 JMT1801ED 应用连接示例图

1.5 引脚配置

1.5.1 LQFP48L 引脚图

JMT1801ED 芯片 LQFP48L 封装如 Figure 3 所示：

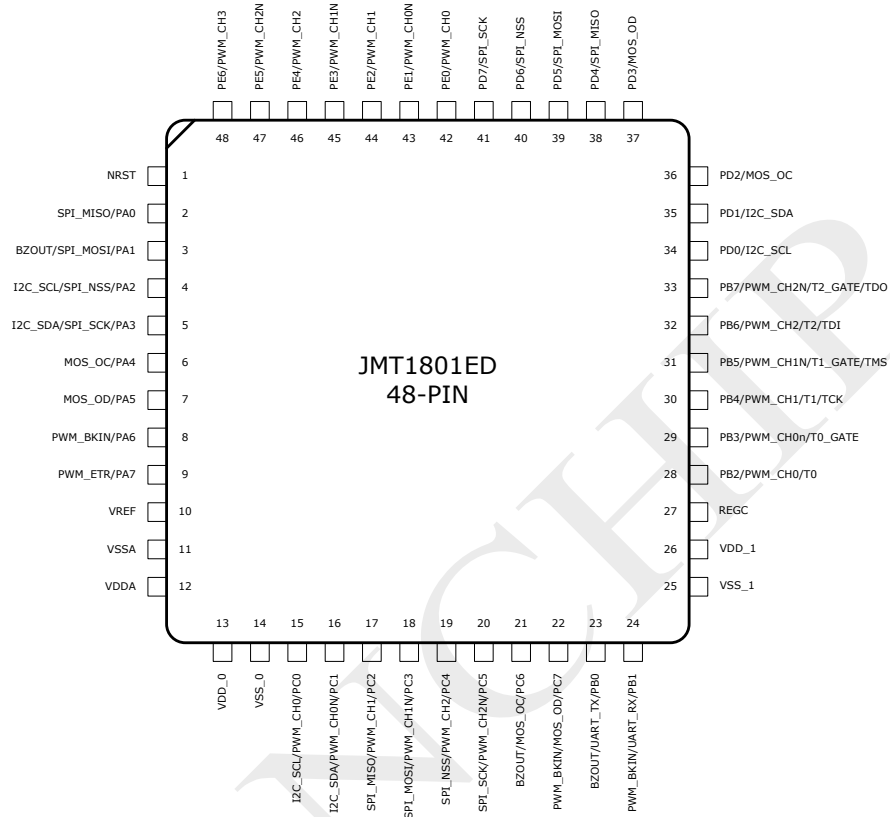


Figure 3 JMT1801ED 芯片 LQFP48L 封装

1.5.2 QFN40L 引脚图

JMT1801ED 芯片 QFN40L 封装如 Figure 4 所示:

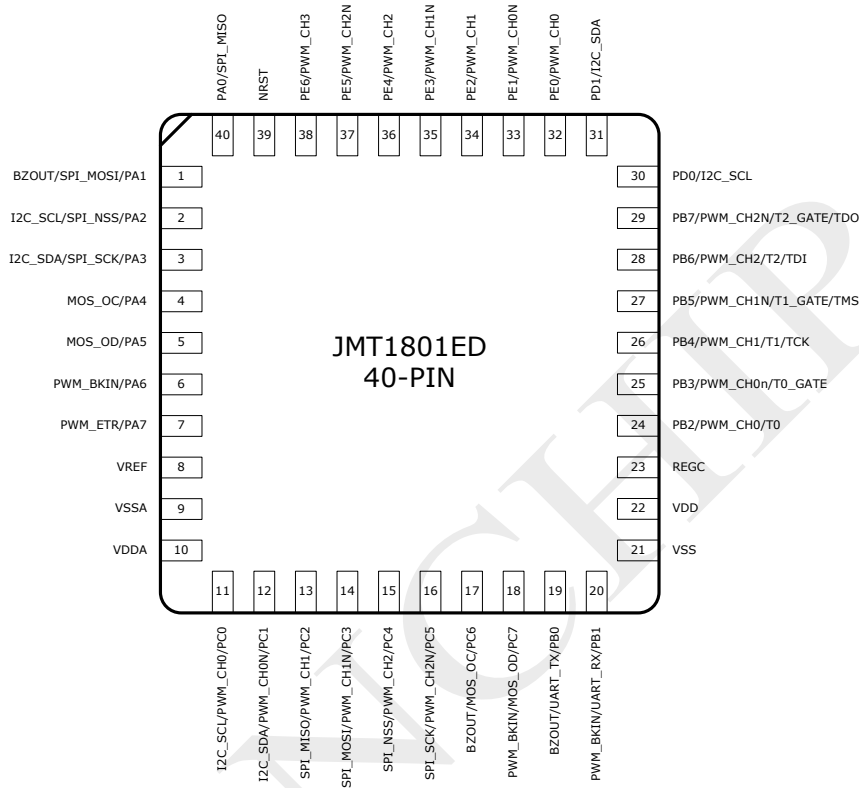


Figure 4 JMT1801ED 芯片 QFN40L 封装

1.5.3 QFN32L 引脚图

JMT1801ED 芯片 QFN32L 封装如 Figure 5 所示:

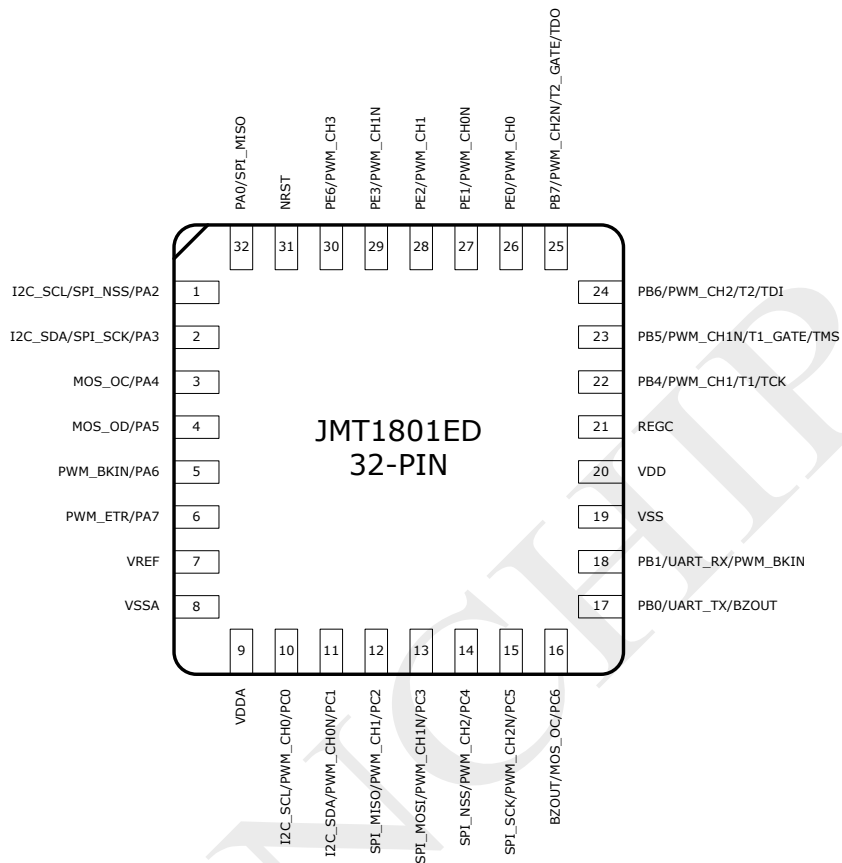


Figure 5 JMT1801ED 芯片 QFN32L 封装

1.5.4 TSSOP24L 引脚图

JMT1801ED 芯片 TSSOP24L 封装如 Figure 6 所示：

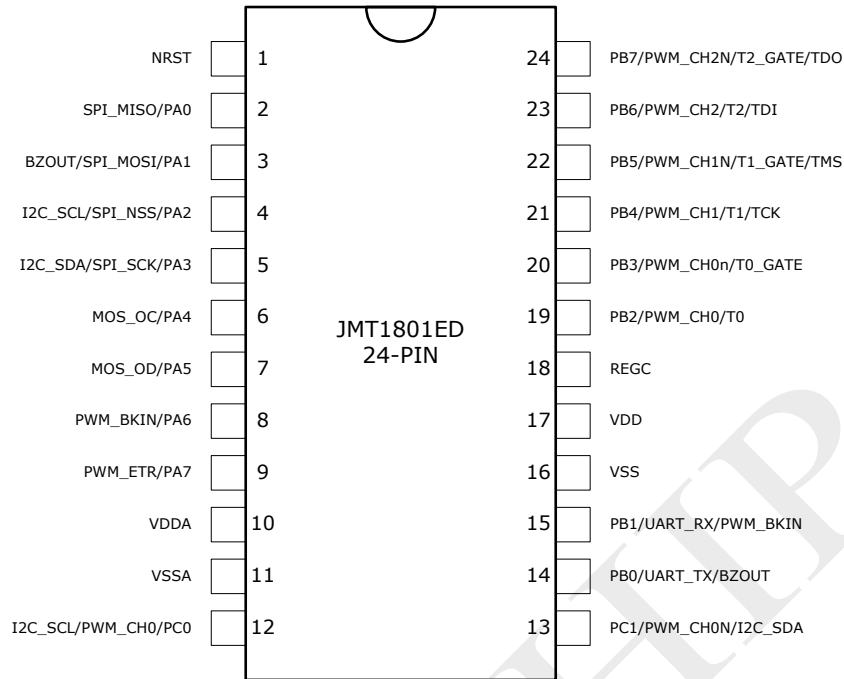


Figure 6 JMT1801ED 芯片 TSSOP24L 封装

1.5.5 TSSOP16L 引脚图

JMT1801ED 芯片 TSSOP16L 封装如 Figure 7 所示:

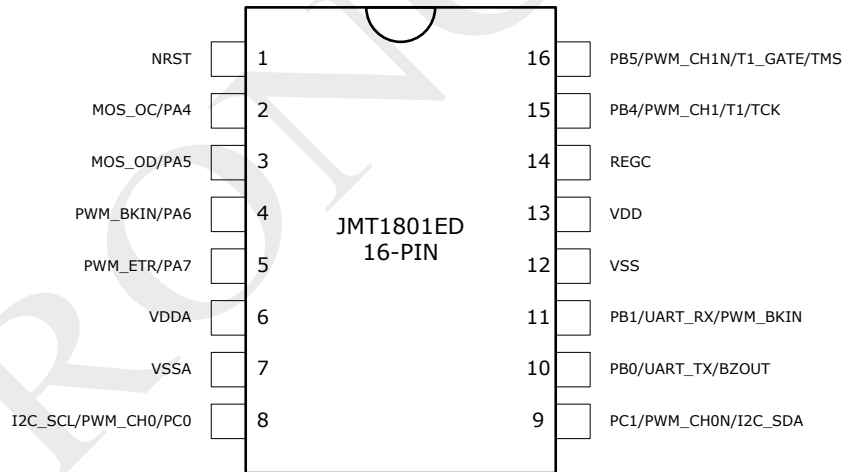


Figure 7 JMT1801ED 芯片 TSSOP16L 封装

1.5.6 引脚说明

JMT1801ED 芯片 I/O 分为两种类型，一种为数字、模拟复用 I/O，另一种为纯数字 I/O。PA0~PA7、PC0~PC7 为数字、模拟复用 I/O；PB0~PB7、PD0~PD7、PE0~PE7 为纯数字 I/O。管脚的功能复用见 Table 1:

Table 1 JMT1801ED 引脚说明

序号	序号	序号	序号	序号	引脚名	默认功能	复用功能	复用功能重映	JTAG
LQFP48L	QFN40L	QFN32L	TSSOP24L	TSSOP16L		能		射	引脚

序号 LQFP48L	序号 QFN40L	序号 QFN32L	序号 TSSOP24L	序号 TSSOP16L	引脚名	默认功 能	复用功能	复用功能重映 射	JTAG 引脚
1	39	31	1	1	NRST				
2	40	32	2		PA0	GPIO	SPI_MISO		
3	1		3		PA1	GPIO	SPI_MOSI	BZOUT	
4	2	1	4		PA2	GPIO	SPI_NSS	I2C_SCL	
5	3	2	5		PA3	GPIO	SPI_SCK	I2C_SDA	
6	4	3	6	2	PA4	GPIO	MOS_OC		
7	5	4	7	3	PA5	GPIO	MOS_OD		
8	6	5	8	4	PA6	GPIO	PWM_BKIN		
9	7	6	9	5	PA7	GPIO	PWM_ETR		
10	8	7	10	6	VREF				
11	9	8	11	7	VSSA				
12	10	9			VDDA				
13					VDD_0				
14					VSS_0				
15	11	10	12	8	PC0	GPIO	PWM_CH0	I2C_SCL	
16	12	11	13	9	PC1	GPIO	PWM_CH0N	I2C_SDA	
17	13	12			PC2	GPIO	PWM_CH1	SPI_MISO	
18	14	13			PC3	GPIO	PWM_CH1N	SPI_MOSI	
19	15	14			PC4	GPIO	PWM_CH2	SPI_NSS	
20	16	15			PC5	GPIO	PWM_CH2N	SPI_SCK	
21	17	16			PC6	GPIO	MOS_OC	BZOUT	
22	18				PC7	GPIO	MOS_OD	PWM_BKIN	
23	19	17	14	10	PB0	GPIO	UART_TX	BZOUT	
24	20	18	15	11	PB1	GPIO	UART_RX	PWM_BKIN	
25	21	19	16	12	VSS_1				
26	22	20	17	13	VDD_1				
27	23	21	18	14	REGC				
28	24		19		PB2	GPIO	PWM_CH0	T0	
29	25		20		PB3	GPIO	PWM_CH0N	T0_GATE	
30	26	22	21	15	PB4	GPIO	PWM_CH1	T1	TCK

序号 LQFP48L	序号 QFN40L	序号 QFN32L	序号 TSSOP24L	序号 TSSOP16L	引脚名	默认功 能	复用功能	复用功能重映 射	JTAG 引脚
31	27	23	22	16	PB5	GPIO	PWM_CH1N	T1_GATE	TMS
32	28	24	23		PB6	GPIO	PWM_CH2	T2	TDI
33	29	25	24		PB7	GPIO	PWM_CH2N	T2_GATE	TDO
34	30				PD0	GPIO	I2C_SCL		
35	31				PD1	GPIO	I2C_SDA		
36					PD2	GPIO	MOS_OC		
37					PD3	GPIO	MOS_OD		
38					PD4	GPIO	SPI_MISO		
39					PD5	GPIO	SPI_MOSI		
40					PD6	GPIO	SPI_NSS		
41					PD7	GPIO	SPI_SCK		
42	32	26			PE0	GPIO	PWM_CH0		
43	33	27			PE1	GPIO	PWM_CH0N		
44	34	28			PE2	GPIO	PWM_CH1		
45	35	29			PE3	GPIO	PWM_CH1N		
46	36				PE4	GPIO	PWM_CH2		
47	37				PE5	GPIO	PWM_CH2N		
48	38	30			PE6	GPIO	PWM_CH3		

2 电源管理 (PMU)

2.1 概述

JMT1801ED 的工作电压范围为 2.75V~5.5V，内置两个电压调节器：

- 主电压调节器 (MAIN LDO)
- 常开电压调节器 (AO LDO)

MAIN LDO 可以通过低功耗控制关闭，关闭后通过 AO LDO 为看门狗 (WDT) 以及一些控制电路供电。Figure 8¹为 JMT1801ED 供电示意图。

¹ Figure 8 中的 PAx 中的 x 表示 PA 口的标号，x=0, 1, 2, 3, 4, 5, 6, 7。下同。

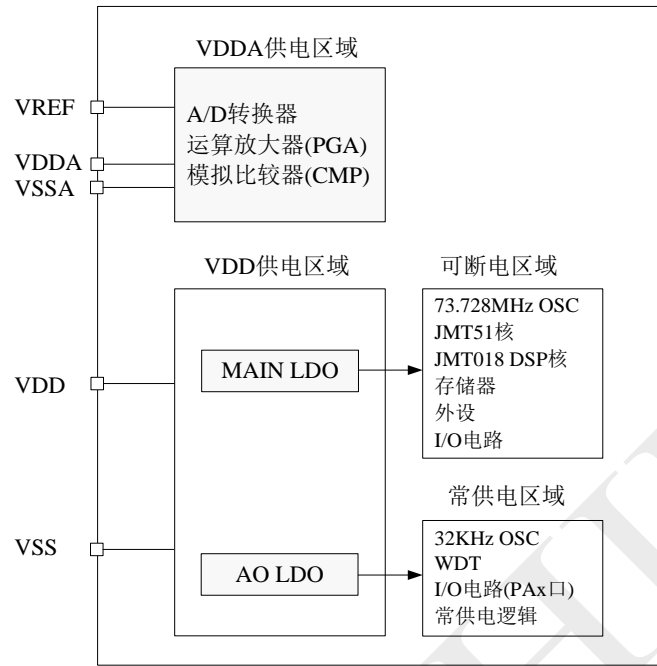


Figure 8 JMT1801ED 芯片供电示意图

Figure 8 中主要分成模拟供电区域（VDDA）和数字供电区域（VDD）。

- 模拟供电部分为 A/D 转换器，运算放大器及模拟比较器等模拟模块供电，VSSA 为模拟地，VREF 为芯片外部参考电压。
- 数字供电部分一方面通过 MAIN LDO 为可断电区域供电，主要包含 73.728MHz OSC、JMT51 核、JMT018 DSP 核、存储器、外设和 I/O 电路等；另一方面通过 AO LDO 为 32KHz OSC、WDT 和常供电逻辑供电，保证在 MAIN LDO 关闭状态下这些模块能够正常工作。

2.2 低功耗控制

在芯片上电后，微控制器处于正常运行模式（Normal）。根据用户需要，可以利用多种低功耗模式来节省功耗。用户可以根据最低功耗、唤醒时间和可用的唤醒源等条件，选定一个最佳的低功耗模式。

JMT1801ED 有四种低功耗模式：

- Idle 模式：JMT51 核时钟关闭，其他所有模块正常工作。
- Stop 模式：JMT51 核和外设时钟关闭；内部 73.728MHz OSC 关闭；MAIN LDO 及模拟模块正常供电，32KHz OSC 以及看门狗（WDT）正常工作。
- Sleep 模式：在 Stop 模式基础上进一步关闭 MAIN LDO 及模拟模块供电。
- Deep Sleep 模式：在 Sleep 模式基础上进一步关闭 32KHz OSC，此时 WDT 停止工作。
- 四种低功耗模式下对应功耗大小排序为： $P_{Idle} > P_{Stop} > P_{Sleep} > P_{Deep Sleep}$ 。

- 四种低功耗模式所需的唤醒时间长短排序为： $T_{Deep\ Sleep} = T_{Sleep} > T_{Stop} > T_{Idle}$ 。

此外，在 Normal 模式下，可以通过以下方式中的一种降低功耗：

- 降低系统时钟频率
- 关闭未被使用的外设时钟
- 关闭不工作的模拟模块，如 ADC，PGA，比较器等

各工作模式下工作时钟及供电开关如 Table 2 所示：

Table 2 各模式下时钟及供电开关场景表

工作模式	JMT51 时钟	JMT018 时钟	外设 时钟	73.728MHz OSC	32KHz OSC	MAIN LDO	AO LDO
Normal	开	开	开	开	开	开	开
Idle	关	开	开	开	开	开	开
Stop	关	关	关	关	开	开	开
Sleep	关	关	关	关	开	关	开
Deep Sleep	关	关	关	关	关	关	开

2.2.1 降低系统时钟频率

在正常运行模式下，可以通过配置系统时钟分频系数寄存器 `SYS_CLK_DIV` 来降低系统时钟频率，从而降低系统功耗；在进入 Idle 之前，也可以通过配置该寄存器降低外设的时钟。

2.2.2 外设时钟门控

JMT1801ED 提供了外设的时钟门控功能，用户可以根据自己使用的外设资源及应用场景灵活的打开关闭相应外设，这样可以有效地降低系统的动态功耗，外设门控的控制可参见 3.1.3.6.2 章节门控时钟寄存器的说明。

2.2.3 关闭模拟模块

JMT1801ED 模拟模块提供了低功耗模式管理，用户可根据具体使用情况对模拟模块进行开关，包括 ADC，运算放大器 PGA，模拟比较器 CMP。具体参见各自模块章节说明。

2.2.4 四种低功耗模式

PMU 控制全芯片进入或退出四种低功耗模式，即 Idle、Stop、Sleep 和 Deep Sleep 模式。通过配置寄存器 `PMUCTR` 进入四种低功耗模式。

寄存器 `PMUCTR` 管理四种不同的低功耗模式，若同时配置多种低功耗模式时，为了不引起混淆，将进入功耗最低的一种模式。例如同时配置 Idle 和 Sleep 模式时，则进入 Sleep

模式。同样唤醒时也要满足 Sleep 唤醒的条件才能够唤醒。

JMT1801ED 四种低功耗模式与正常运行模式的状态转换图如 Figure 9 所示²。

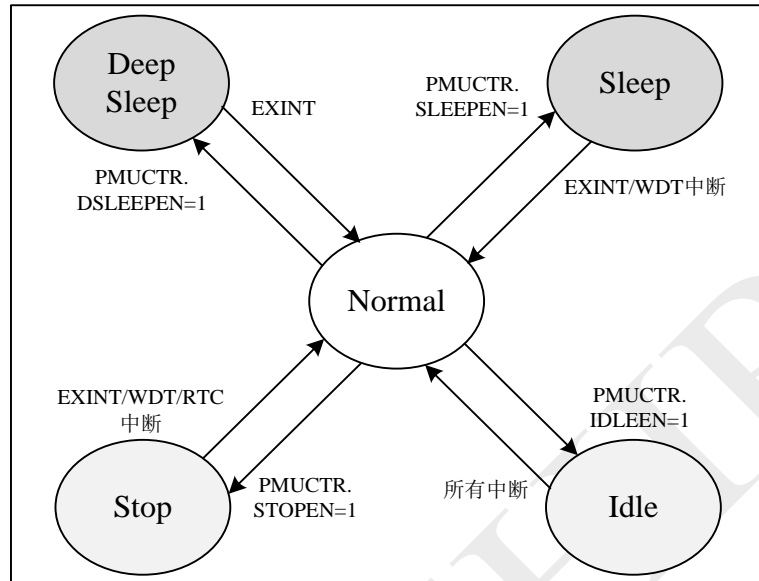


Figure 9 JMT1801ED 工作模式切换场景图

2.2.4.1 Idle 模式

2.2.4.1.1 进入 Idle 模式

通过配置寄存器 PMUCTR.IDLEEN 为 1，全芯片进入 Idle 模式。

在此模式下，JMT51 核时钟关闭，除 JMT51 核外所有模块正常工作。

2.2.4.1.2 退出 Idle 模式

退出 Idle 模式至 Normal 模式的唤醒源为所有中断。

该模式唤醒所需时间为 1 个系统时钟周期，退出 Idle 模式后切换至 Normal 模式，所有模块正常工作，JMT51 核继续取指向下执行应用程序。

2.2.4.2 Stop 模式

2.2.4.2.1 进入 Stop 模式

通过配置寄存器 PMUCTR.STOPEN 为 1，全芯片进入 Stop 模式。

在此模式下，JMT51 核和外设时钟关闭；内部 73.728MHz OSC 关闭，MAIN LDO 及模拟模块正常供电，32KHz OSC 以及看门狗（WDT）正常工作。

2.2.4.2.2 退出 Stop 模式

退出 Stop 模式至 Normal 模式的唤醒源有：

- 看门狗（WDT）中断唤醒或溢出复位
- 外部中断 EXINT（PA_x 中断）

² Figure 9 中的 EXINT 表示外部中断，对应管脚为 PA_x。

该模式唤醒所需时间为 20us, 退出 Stop 模式后切换至 Normal 模式, 所有模块正常工作, JMT51 核继续取指向下执行应用程序。

内部 73.728MHz 和 32KHz OSC 自动打开。系统时钟硬件自动切换至内部 73.728MHz 时钟源。

2.2.4.3 Sleep 模式

2.2.4.3.1 进入 Sleep 模式

通过配置寄存器 PMUCTR.SLEEPEN 为 1, 全芯片进入 Sleep 模式。

在此模式下, JMT51 核和外设时钟关闭; 内部 73.728MHz OSC 关闭, MAIN LDO 及模拟模块关闭供电, 内部 32KHz OSC、WDT 及 PAx 口由 AO LDO 供电正常工作。

2.2.4.3.2 退出 Sleep 模式

退出 Sleep 模式至 Normal 模式的唤醒源有:

- WDT 中断唤醒或溢出复位
- 外部中断 EXINT (PAx 中断)

该模式唤醒所需时间为 22us, 退出 Sleep 模式后全芯片将自动复位进入 Normal 模式, JMT51 核从 FLASH 的 0 地址开始执行应用程序。

退出 Sleep 模式后系统时钟硬件自动切换至内部 73.728MHz 时钟源。

2.2.4.4 Deep Sleep 模式

2.2.4.4.1 进入 Deep Sleep 模式

通过配置寄存器 PMUCTR.DSLEEPEN 为 1, 全芯片进入 Deep Sleep 模式。

在此模式下, JMT51 核和外设时钟关闭; 内部 73.728MHz OSC、32KHz OSC 关闭, MAIN LDO 及模拟模块关闭供电, WDT 停止工作。

2.2.4.4.2 退出 Deep Sleep 模式

退出 Deep Sleep 模式至 Normal 模式的唤醒源只有外部中断 EXINT (PAx 中断)。

该模式唤醒所需时间为 22us, 退出 Deep Sleep 模式后全芯片将自动复位进入 Normal 模式, JMT51 核从 FLASH 的 0 地址开始执行应用程序。

退出 Deep Sleep 模式后系统时钟硬件自动切换至内部 73.728MHz 时钟源。

2.2.5 OSC 开关说明

OSC 的打开和关闭可通过软件和硬件两种方式实现:

软件开关:

- JMT1801ED 有两个时钟, 其中 73.728MHz OSC 可以通过软件开关。
- 配置寄存器 OSCPD.OSC73MPD, 控制 73.728MHz OSC 的打开及关闭。配置为 0 时打开 (OSCPD.OSC73MPD 复位状态为 0), 配置为 1 时关闭。

- 对 73.728MHz OSC 正常软件开关时，要保证系统时钟使用的时钟源与关闭的时钟源无关，即系统时钟对应的时钟源为 32KHz OSC，这样才可以正常的通过配置寄存器实现软件开关操作。

硬件开关：

- 根据配置的低功耗模式寄存器 STOPEN、SLEEPEN 或 DSLEEPEN，硬件自动关闭相应的 OSC。由唤醒逻辑或复位，硬件自动打开 OSC，具体参见 2.2.4 四种低功耗模式中的说明。

2.3 寄存器

PMU 有 1 个寄存器，对应的 SFR PAGE 为 0，具体列表如 Table 3 所示。

Table 3 PMU 寄存器 (SFRPAGE=0x00)

地址	寄存器名	属性	复位值	功能描述
0xA1	PMUCTR	W	0x00	低功耗控制寄存器

属性：W/R—可读写，W—只写，R—只读（下同）

2.3.1 低功耗控制寄存器(PMUCTR) [0xA1]

Table 4 低功耗控制寄存器(PMUCTR)

参数名	比特位	属性	复位值	描述
IDLEEN	0	W	0	IDLE 模式使能： 1: 进入 IDLE 模式 0: 无效
STOPEN	1	W	0	STOP 模式使能： 1: 进入 STOP 模式 0: 无效
SLEEPEN	2	W	0	Sleep 模式使能： 1: 进入 Sleep 模式 0: 无效
DSLEEPEN	3	W	0	Deep Sleep 模式使能： 1: 进入 Deep Sleep 模式 0: 无效
Reserved	7-4	R	0	保留

3 时钟复位管理 (CRM)

3.1 时钟管理

3.1.1 概述

- 芯片包含两个振荡器：一个输出频率为 73.728MHz 的片内高频振荡器，一个输出频率为 32KHz 的片内低频振荡器。
- 时钟管理模块管理时钟选择、时钟分频和时钟门控，为各个模块提供所需的时钟。
- 看门狗 (WDT) 和 PMU 模块的时钟直接由片内低频振荡器产生。
- JMT51 核、外设时钟由片内高频振荡器产生的 73.728MHz 时钟、片内低频振荡器产生的 32KHz 时钟经过分频、选择和门控得到。

3.1.2 框图

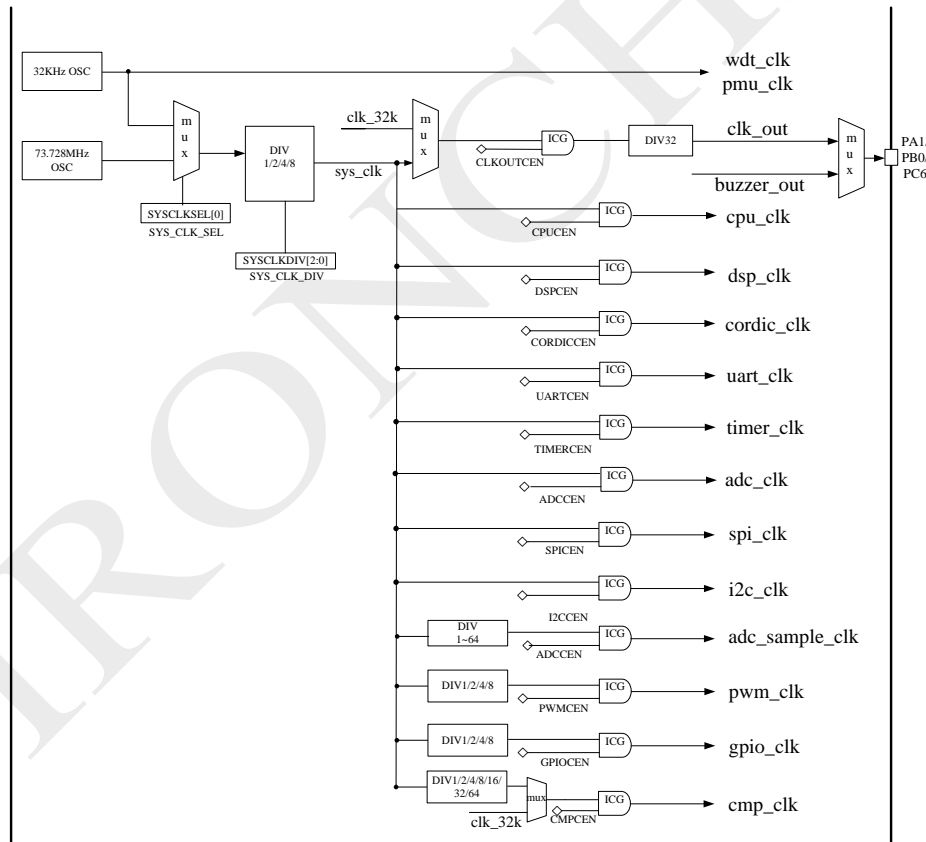


Figure 10 JMT1801ED 时钟树

3.1.3 功能描述

3.1.3.1 时钟源

JMT1801ED 时钟模块包含两个振荡器：输出频率为 32KHz 的片内低频振荡器、输出频率

率为 73.728MHz 的片内高频振荡器。

3.1.3.1.1 片内 73.728MHz RC 振荡器

片内高频 RC 振荡器提供 73.728MHz 的时钟，它可以直接作为系统时钟。它能够在不需要任何外部器件的条件下提供系统时钟。

制造工艺决定了不同芯片的 RC 振荡器频率不同，为了减小 73.728MHz 时钟频率的误差，在芯片出厂前需要对振荡器进行校准，通过将校准值写入寄存器 TRIMVAL73M[6:0]，可以调整片内高频振荡器时钟的频率。出厂时，片内高频振荡器的时钟频率误差被校准到 1% 以内（25℃）。

片内高频振荡器可以由软件控制启动和停止，配置寄存器 OSCPD 的 OSC73MPD 位为 1，关闭片内高频振荡器，配置寄存器 OSCPD 的 OSC73MPD 位为 0，打开片内高频振荡器。

3.1.3.1.2 片内 32KHz RC 振荡器

片内低频 RC 振荡器提供 32KHz 时钟，它可以直接作为 PMU、Watch Dog 的时钟，也可以作为系统时钟。它能够在不需要任何外部器件的条件下提供时钟。

片内低频振荡器只能由硬件控制启动和停止。芯片工作在 DEEP SLEEP 模式下，片内低频振荡器自动关闭，其他工作模式下，片内低频振荡器一直工作。

制造工艺决定了不同芯片的 RC 振荡器频率不同，通过配置寄存器 TRIMVAL32K[3:0] 可以调整片内低频振荡器的输出时钟频率，出厂时，片内低频振荡器的时钟频率误差被校准到 10% 以内（25℃）。

3.1.3.2 系统时钟

系统时钟的最高工作频率为 73.728MHz，复位后默认工作时钟频率为 9.19MHz。系统时钟 sys_clk 可以通过以下两个步骤得到：

首先，做时钟选择。时钟源 MUX 可通过配置 SYS_CLK_SEL 寄存器选择以下两个时钟源：

- 片内 32KHz 振荡器输出的时钟。
- 片内 73.728MHz 振荡器产生的时钟。

该时钟源 MUX 默认选择片内 73.728MHz 振荡器产生的时钟。

其次，做时钟分频，过程如下：

- 时钟源 MUX 输出连接至分频器。
- 分频器可通过配置寄存器 SYS_CLK_DIV 得到 1/2/4/8 分频。

分频后即得到 sys_clk。

3.1.3.3 时钟输出和蜂鸣器控制

时钟信号和蜂鸣器控制信号可由管脚 PA1、PB0 或 PC6 输出，当寄存器 BUZZER_CTRL1 的 BUZEN 位配置为 0 时，管脚 PA1、PB0 或 PC6 输出时钟信号，当 BUZZER_CTRL1 的

BUZEN 位配置为 1 时，管脚 PA1、PB0 和 PC6 输出蜂鸣器控制信号。

3.1.3.3.1 时钟输出

当寄存器 BUZZER_CTRL1 的 BUZEN 位配置为 0 时，JMT1801ED 允许输出时钟信号到外部 PA1、PB0 或 PC6 引脚，产生过程如下：

首先，做时钟选择，可通过配置寄存器 CRMCTRL 选择以下两个时钟：

- 片内 32KHz 振荡器输出的时钟。
- 系统时钟 sys_clk。

复位时，默认选择系统时钟 sys_clk 作为时钟输出。

其次，对选出的时钟做时钟门控，配置寄存器 CLK_EN0 的 CLKOUTCEN 位为 1，打开时钟输出，配置寄存器 CLK_EN0 的 CLKOUTCEN 位为 0，关闭时钟输出。

最后，对门控后的时钟做 32 分频。默认情况下，最终输出的时钟频率为 288KHz。

3.1.3.3.2 蜂鸣器控制

当 BUZZER_CTRL1 的 BUZEN 位配置为 1 时，管脚 PA1、PB0 和 PC6 输出蜂鸣器控制信号。

蜂鸣器控制信号在 32KHz 时钟的基础上通过分频得到，蜂鸣器控制信号的频率为：

$$f_{buzzer} = \frac{32KHz}{BUZZER_CTRL0[4:0] + BUZZER_CTRL1[4:0]}$$

占空比为：

$$duty = \frac{BUZZER_CTRL1[4:0]}{BUZZER_CTRL0[4:0] + BUZZER_CTRL1[4:0]}$$

例如，配置寄存器 BUZZER_CTRL0 为 0x04，配置寄存器 BUZZER_CTRL1 为 0x84，则蜂鸣器控制信号的频率为 4KHz，占空比为 50%；若配置寄存器 BUZZER_CTRL0 为 0x04，配置寄存器 BUZZER_CTRL1 为 0x8C，则蜂鸣器控制信号的频率为 2KHz，占空比为 75%，输出的波形图如 Figure 11 所示。

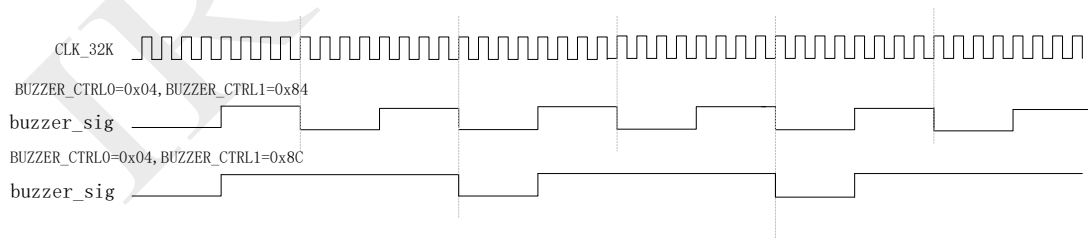


Figure 11 蜂鸣器输出波形图

3.1.3.4 JMT51 MCU 核时钟

JMT51 核时钟的最高工作频率为 73.728MHz，复位后默认工作时钟频率为 9.19MHz，JMT51 核时钟 cpu_clk 由系统时钟 sys_clk 经过门控得到，该时钟门控由硬件根据芯片的低功耗工作模式自动控制，具体参见 2.2 节。

3.1.3.5 JMT018 DSP 核时钟

JMT018 DSP 核时钟的最高工作频率为 73.728MHz，复位后默认工作时钟频率为 9.19MHz。DSP 核时钟 dsp_clk 由 sys_clk 经过门控得到，通过配置 CLK_EN1.DSPCEN 为 0，关闭 dsp_clk，将 CLK_EN1.DSPCEN 配置为 1，打开 dsp_clk。

3.1.3.6 外设时钟

3.1.3.6.1 WDT 和 PMU 模块时钟

此两个模块的时钟频率为 32KHz，由片内 32KHz 振荡器直接提供。

3.1.3.6.2 AD 采样时钟和 AD 控制模块时钟

AD 采样时钟产生过程如下：

首先，做时钟分频：

- 将 sys_clk 连接至分频器。
- 分频器可通过配置寄存器 ADC_CLK_DIV 得到 sys_clk 时钟的 1~64 分频。

其次，对分频得到的时钟做门控：

- 将寄存器 CLK_EN1 的 ADCCEN 位配置为 1，打开 AD 采样时钟，配置为 0，关闭 AD 采样时钟。

AD 控制模块 ADC 的时钟 adc_clk 由系统时钟 sys_clk 经过门控得到，通过配置 CLKEN1.ADCCEN 为 0，可关闭 adc_clk，将 CLKEN2.ADCCEN 配置为 1，打开 adc_clk。

3.1.3.6.3 CORDIC 时钟

CORDIC 模块时钟 cordic_clk 由 sys_clk 经过门控得到，通过配置 CLK_EN0.CORDICCEN 为 0，关闭 cordic_clk，将 CLKEN0.CORDICCEN 配置为 1，打开 cordic_clk。

3.1.3.6.4 UART 时钟

UART 时钟 uart_clk 由系统时钟 sys_clk 经过门控得到，通过配置 CLKEN0.UARTCEN 为 0，关闭 uart_clk，将 CLKEN0.UARTCEN 配置为 1，打开 uart_clk。

3.1.3.6.5 TIMER 时钟

TIMER 时钟 timer_clk 由 sys_clk 经过门控得到，通过配置 CLKEN0.TIMERCEN 为 0，关闭 timer_clk，将 CLKEN0.TIMERCEN 配置为 1，打开 timer_clk。

3.1.3.6.6 PWM 模块时钟

PWM 模块时钟产生过程如下：

首先，做时钟分频：

- 将 sys_clk 连接至分频器。
- PWM 分频器可通过配置寄存器 PWM_CLK_DIV 得到 sys_clk 时钟的 1/2/4/8 分频。

其次，做时钟门控：

- 将寄存器 CLK_EN1 的 PWMCEN 位配置为 0，关闭 PWM 模块时钟，配置为 1，打开 PWM 模块时钟。

3.1.3.6.7 GPIO 时钟

GPIO 模块的最高工作时钟为 73.728MHz，默认工作时钟为 9.19MHz，GPIO 模块的工作时钟 gpio_clk 产生过程如下：

首先，对 sys_clk 做时钟分频：

- 将时钟 sys_clk 链接至 GPIO 分频器。
- GPIO 分频器可通过配置寄存器 GPIO_CLK_DIV 得到 1/2/4/8 分频。

其次，对 GPIO 分频器输出的时钟做门控：

- 将 gpio 分频器的输出连接至 GPIO 门控单元。
- 配置 CLKEN0.GPIOCEN 为 0，关闭 gpio_clk，配置 CLKEN0.GPIOCEN 为 1，打开 gpio_clk。

3.1.3.6.8 SPI 时钟

SPI 模块的最高工作时钟为 73.728MHz，默认工作时钟为 9.19MHz，SPI 模块的工作时钟 spi_clk 产生过程如下：

对 SPI 分频器输出的时钟做门控：

- 将 SPI 分频器的输出连接至 SPI 门控单元。
- 配置 CLKEN1.SPICEN 为 0，关闭 spi_clk，配置 CLKEN1.SPICEN 为 1，打开 spi_clk。

3.1.3.6.9 I2C 时钟

I2C 模块的最高工作时钟为 73.728MHz，默认工作时钟为 9.19MHz，I2C 模块的工作时钟 i2c_clk 产生过程如下：

对 I2C 分频器输出的时钟做门控：

- 将 I2C 分频器的输出连接至 I2C 门控单元。
- 配置 CLKEN1.I2CCEN 为 0，关闭 i2c_clk，配置 CLKEN1.I2CCEN 为 1，打开 i2c_clk。

3.1.3.6.10 比较器滤波时钟

CMP 模块的最高工作时钟为 73.728MHz，默认工作时钟为 9.19MHz，CMP 模块的工作

时钟 `cmp_clk` 产生过程如下：

首先，对 `sys_clk` 做时钟分频：

- 将时钟 `sys_clk` 链接至 `CMP` 分频器。
- 分频器可通过配置寄存器 `CMP_CLK_CTRL` 得到 1/2/4/8/16/32/64 分频。
- 将寄存器 `CMP_CLK_CTRL` 的低 3 位配置为 111，可以选择 32KHz 作为 `CMP` 模块的时钟。

其次，对 `CMP` 分频器输出的时钟做门控：

- 将 `CMP` 分频器的输出连接至 `CMP` 门控单元。
- 配置 `CLKEN1.CMPCLKEN` 为 0，关闭 `cmp_clk`，配置 `CLKEN1.CMPCLKEN` 为 1，打开 `cmp_clk`。

3.1.4 寄存器描述

该模块共有 14 个寄存器，对应的 `SFR PAGE` 为 2，具体列表如 Table 5 所示：

Table 5 时钟模块寄存器（`SFRPAGE=0x02`）

地址	寄存器名	类型	复位值	功能描述
0xDA	<code>SYS_CLK_SEL</code>	W/R	0x00	系统时钟选择寄存器
0XD7	<code>SYS_CLK_DIV</code>	W/R	0x03	系统时钟分频系数选择寄存器
0xDB	<code>ADC_CLK_DIV</code>	W/R	0x09	AD 采样时钟分频系数选择寄存器
0xF2	<code>GPIO_CLK_DIV</code>	W/R	0x00	GPIO 时钟分频系数选择寄存器
0xF1	<code>CMP_CLK_CTRL</code>	W/R	0x00	模拟比较器滤波时钟控制寄存器
0xED	<code>PWM_CLK_DIV</code>	W/R	0x00	PWM0 时钟分频系数选择寄存器
0XE9	<code>CLK_EN0</code>	W/R	0x03	时钟门控寄存器 0
0xEA	<code>CLK_EN1</code>	W/R	0x3C	时钟门控寄存器 1
0XD6	<code>OSCPD</code>	W/R	0x00	振荡器时钟控制寄存器

地址	寄存器名	类型	复位值	功能描述
0xCF	TRMVAL73M	W/R	0x40	73.728MHz 时钟校准值
0XD1	TRMVAL32K	W/R	0x08	32KHz 时钟校准值
0XD9	BANDGAP_CTRL	W/R	0x10	BANDGAP 校准值
0XD3	ADCTRIM	W/R	0x08	ADC 内部参考电压校准值
0xDC	JTAGEN	W/R	0x00	JTAG 模式使能寄存器
0xDE	BUZZER_CTRL0	W/R	0x01	蜂鸣器控制寄存器 0
0xDF	BUZZER_CTRL1	W/R	0x01	蜂鸣器控制寄存器 1
0xD2	CRMCTRL	W/R	0x00	时钟输出控制寄存器

3.1.4.1 系统时钟选择寄存器 (SYS_CLK_SEL) [0xDA]

Table 6 系统时钟选择寄存器(SYS_CLK_SEL)

参数名	比特位	属性	复位值	描述
SYSCLKSEL	0	W/R	0	系统时钟选择寄存器： 0：选择 73MHz 时钟； 1：选择片内 32KHz 时钟。
RESERVED	7-2	R	0	保留。

3.1.4.2 系统时钟分频系数选择寄存器(SYS_CLK_DIV) [0XD7]

Table 7 系统时钟分频系数选择寄存器(SYS_CLK_DIV)

参数名	比特位	属性	复位值	描述
SYSCLKDIV	1-0	W/R	11	系统时钟分频系数选择寄存器： 00：1 分频； 01：2 分频； 10：4 分频； 11：8 分频。
RESERVED	7-2	R	0	保留。

3.1.4.3 AD 采样时钟分频系数选择寄存器 (ADC_CLK_DIV) [0xDB]

Table 8 AD 采样时钟分频系数选择寄存器(ADC_CLK_DIV)

参数名	比特位	属性	复位值	描述
ADCCLKSEL	5-0	W/R	001001	AD0 和 AD1 采样时钟分频系数选择： 0-63：对系统时钟 sys_clk 进行 1-64 分频，得到 AD0 和 AD1 采样时钟。
RESERVED	7-6	R	0	保留。

3.1.4.4 GPIO 时钟分频系数选择寄存器(GPIO_CLK_DIV) [0xF2]

Table 9 GPIO 时钟分频系数选择寄存器(GPIO_CLK_DIV)

参数名	比特位	属性	复位值	描述
GPIOCLKDIV	1-0	W/R	00	GPIO 模块时钟分频系数选择寄存器： 00：1 分频； 01：2 分频； 10：4 分频； 11：8 分频。
RESERVED	7-2	R	0	保留

3.1.4.5 模拟比较器滤波时钟控制寄存器(CMP_CLK_CTRL) [0xF1]

Table 10 模拟比较器滤波时钟控制寄存器(CMP_CLK_CTRL)

参数名	比特位	属性	复位值	描述
CMPCLKDIV	2-0	W/R	000	模拟比较器滤波时钟分频系数选择寄存器 000：sys_clk 1 分频； 001：sys_clk 2 分频； 010：sys_clk 4 分频； 011：sys_clk 8 分频； 100：sys_clk 16 分频； 101：sys_clk 32 分频。

参数名	比特位	属性	复位值	描述
				111: 选择 32KHz 时钟作为模拟比较器滤波时钟。
CMPCLKEN	3	W/R	0	模拟比较器滤波时钟门控寄存器: 0: 关闭模拟比较器滤波时钟; 1: 打开模拟比较器滤波时钟。
RESERVED	7-4	R	0	保留

3.1.4.6 PWM 时钟分频系数选择寄存器(PWM_CLK_DIV) [0xED]

Table 11 PWM 时钟分频系数选择寄存器(PWM_CLK_DIV)

参数名	比特位	属性	复位值	描述
PWMCLKDIV	1-0	W/R	00	PWM 模块时钟分频系数选择寄存器: 00: 1 分频; 01: 2 分频; 10: 4 分频; 11: 8 分频。
RESERVED	7-2	R	0	保留。

3.1.4.7 时钟门控寄存器 0 (CLK_EN0) [0xE9]

Table 12 时钟门控寄存器 0 (CLK_EN0)

参数名	比特位	属性	复位值	描述
CMPCLKEN	0	W/R	0	CMP 模块滤波时钟门控寄存器: 0: 关闭 CMP 模块滤波时钟; 1: 打开 CMP 模块滤波时钟。
UARTCEN	1	W/R	1	UART 模块时钟门控: 1: 打开 UART 模块时钟; 0: 关闭 UART 模块时钟。
TIMERCEN	2	W/R	1	TIMER 模块时钟门控:

参数名	比特位	属性	复位值	描述
				1: 打开 TIMER 模块时钟; 0: 关闭 TIMER 模块时钟。
GPIOCEN	3	W/R	1	GPIO 模块时钟门控: 1: 打开 GPIO 模块时钟; 0: 关闭 GPIO 模块时钟。
CORDICEN	4	W/R	0	CORDIC 模块时钟门控: 1: 打开 CORDIC 模块时钟 0: 关闭 CORDIC 模块时钟
DSPCEN	5	W/R	0	DSP 模块时钟门控: 1: 打开 DSP 模块时钟; 0: 关闭 DSP 模块时钟。
RESERVED	7-6	R	0	保留。

3.1.4.8 时钟门控寄存器 1 (CLK_EN1) [0xEA]

Table 13 时钟门控寄存器 1 (CLK_EN1)

参数名	比特位	属性	复位值	描述
I2CCEN	0	W/R	0	I2C 模块时钟门控: 1: 打开 I2C 模块时钟 0: 关闭 I2C 模块时钟
PWMCEN	1	W/R	0	PWM0 模块时钟门控: 1: 打开 PWM 模块时钟 0: 关闭 PWM 模块时钟
SPICEN	2	W/R	0	SPI 模块时钟门控: 1: 打开 SPI 模块时钟; 0: 关闭 SPI 模块时钟。
ADCCEN	3	W/R	0	ADC 模块时钟门控:

参数名	比特位	属性	复位值	描述
				1: 打开 ADC 模块时钟; 0: 关闭 ADC 模块时钟。
RESERVED	7-4	R	0	保留。

3.1.4.9 振荡器时钟控制寄存器 (OSCPD) [0xD6]

Table 14 振荡器时钟控制寄存器(OSCPD)

参数名	比特位	属性	复位值	描述
RESERVED	0	R	0	保留。
OSC73MPD	1	W/R	0	片内 73MHz 振荡器使能寄存器: 0: 片内 73MHz 振荡器正常工作; 1: 关闭片内 73MHz 振荡器。
RESERVED	7-2	R	0	保留

3.1.4.10 73.728MHz 时钟校准值 (TRMVAL73M) [0xCF]

Table 15 73.728MHz 时钟校准值(TRMVAL73M)

参数名	比特位	属性	复位值	描述
TRMVAL73M	6-0	W/R	1000000	73.728MHz 时钟校准时写入的校准值。
RESERVED	7	R	0	保留。

3.1.4.11 32KHz 时钟校准值 (TRMVAL32K) [0XD1]

Table 16 32KHz 时钟校准值 (TRMVAL32K)

参数名	比特位	属性	复位值	描述
TRMVAL32K	3-0	W/R	1000	32KHz 时钟校准时写入的校准值。
RESERVED	7-4	R	0	保留。

3.1.4.12 BANDGAP 输出值校准 (BANDGAP_CTRL) [0XD9]

Table 17 BANDGAP 输出值校准 (BANDGAP_CTRL)

参数名	比特位	属性	复位值	描述
BGTRIM	4-0	W/R	10000	BANDGAP 输出值校准。
RESERVED	7-5	R	0	保留。

3.1.4.13 ADC 参考电压校准值寄存器(ADCTRIM) [0XD3]

Table 18 ADC 参考电压校准值寄存器(ADCTRIM)

参数名	比特位	属性	复位值	描述
ADCTRIM	3-0	W/R	1000	ADC 参考电压校准值寄存器。
RESERVED	7-4	R	0	保留

3.1.4.14 JTAG 模式使能寄存器 (JTAGEN) [0xDC]

Table 19 JTAG 模式使能寄存器(JTAG_EN)

参数名	比特位	属性	复位值	描述
JTAGEN	0	W/R	0	JTAG 模式使能寄存器: 0: 关闭 JTAG 模式。 1: 使能 JTAG 模式。
RESERVED	7-1	R	0	保留

3.1.4.15 蜂鸣器控制寄存器 0 (BUZZER_CTRL0) [0xDE]

Table 20 蜂鸣器控制寄存器 0 (BUZZER_CTRL0)

参数名	比特位	属性	复位值	描述
LOW_DUTY	4-0	W/R	1	蜂鸣器控制信号中低电平持续的 32KHz 时钟数
RESERVED	7-5	R	0	保留

3.1.4.16 蜂鸣器控制寄存器 1 (BUZZER_CTRL1) [0xDF]

Table 21 蜂鸣器控制寄存器 1 (BUZZER_CTRL1)

参数名	比特位	属性	复位值	描述
HIGH_DUTY	4-0	W/R	1	蜂鸣器控制信号高电平持续的 32KHz 时钟数

参数名	比特位	属性	复位值	描述
RESERVED	6-5	R	0	保留
BUZZER_EN	7	W/R	0	蜂鸣器使能寄存器： 0：关闭蜂鸣器； 1：使能蜂鸣器。

3.1.4.17 输出时钟控制寄存器(CRMCTRL) [0XD2]

Table 22 输出时钟控制寄存器(CRMCTRL)

参数名	比特位	属性	复位值	描述
RESERVED	0	R	0	保留。
BORENCTRL	1	W/R	0	BOR 控制方式寄存器： 0：由软件配置 BORSWEN 寄存器，控制 BOR 模块的使能； 1：由硬件控制 BOR 模块的使能。芯片进入 DEEP SLEEP 模式时自动关闭 BOR，退出 DEEP SLEEP 模式时自动打开 BOR。
BORSWEN	2	W/R	1	BOR 模块使能寄存器： 0：BOR 模块正常工作； 1：关闭 BOR 模块。
LVRSTEN	3	W/R	0	LVR 复位信号使能控制： 0：LVR 复位信号无效； 1：LVR 复位信号有效。
CLKOUTCTRL	4	W/R	0	输出时钟选择寄存器： 0：选择系统时钟作为输出时钟； 1：选片内 32KHz 时钟作为输出时钟。
RESERVED	7-5	R	0	保留。

3.2 复位管理

3.2.1 概述

JMT1801ED 有 7 种复位方式：

- 上电复位（Power On Reset）
- 掉电复位（Brown Out Reset）
- 管脚硬复位（Hardware Reset）
- 低电压检测复位（LVR Reset）
- 看门狗复位（Watch Dog Reset）
- 全局软复位（Global Software Reset）
- 电源分区复位（Power Reset）
- 模块软复位（Module Software Reset）

JMT1801ED 中各个复位信号均为低有效。将各个复位源信号（低有效）送入 CRM 模块，在 CRM 模块产生各模块的复位信号。只要上电复位、掉电复位、管脚硬复位、低电压检测复位、看门狗复位和全局软复位中的任何一个复位源变为低电平，CRM 模块均会为各模块产生复位信号。退出 SLEEP 或 DEEP SLEEP 低功耗模式后，PMU 模块产生电源分区复位，对断电的模块进行复位。

3.2.2 框图

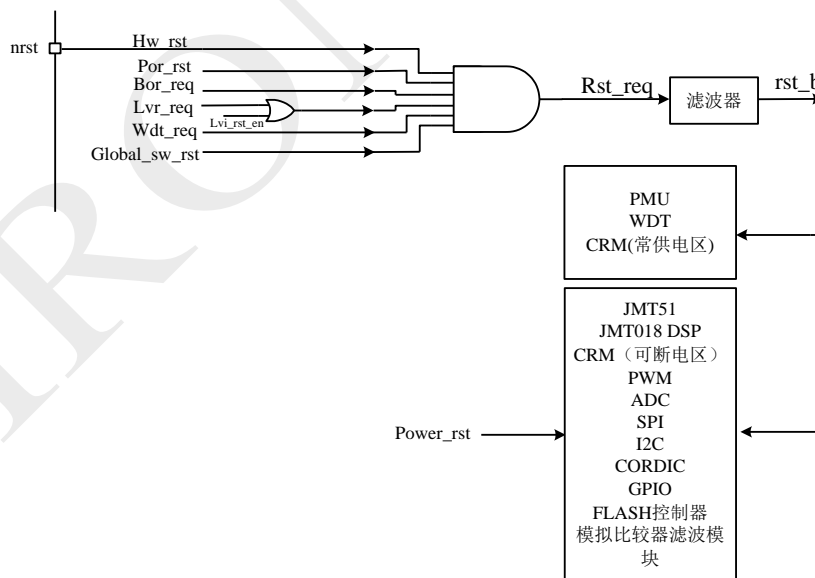


Figure 12 JMT1801ED 复位

3.2.3 功能描述

3.2.3.1 上电复位

上电复位 (Power On Reset): 上电后, 芯片内部产生上电复位信号送到 CRM 模块, 经过滤波后产生全芯片复位信号, 对全芯片所有模块进行复位, 复位结束后芯片进入 boot 流程。

3.2.3.2 掉电复位(BOR)

掉电复位电路监控主电源 VDD, 当 VDD 发生掉电, 电压小于限位电压 V_{BOR} 时 (参见 19.2.6 章节), 掉电复位信号有效。

可选择由硬件或软件控制掉电复位检测电路, 配置寄存器 CRMCTRL.BORENCTRL 为 1, 由硬件控制掉电复位检测电路的启停; 当 CRMCTRL.BORENCTRL 配置为 0 时, 由软件控制掉电复位检测电路的启停。默认为软件控制掉电复位检测电路的启停。

若掉电复位检测电路由软件控制, 当 CRMCTRL.BORSWEN 配置为 1, 打开掉电复位检测电路, 当 CRMCTRL.BORSWEN 配置为 0, 关闭掉电复位检测电路, 上电复位后, 掉电复位检测电路默认打开。

若掉电复位检测电路由硬件控制, 芯片工作在 DEEP SLEEP 模式下, 关闭掉电复位检测电路, 其他工作模式下, 掉电复位检测电路正常工作。

当掉电复位信号产生后, CRM 模块为各个模块产生复位信号, 对全芯片进行复位, 复位结束后, JMT51 核从 FLASH 的 0 地址开始执行应用程序。

3.2.3.3 管脚硬复位

硬复位信号由管脚 nrst 输入, 通过配置 NRSTFQS 寄存器的 NRSTFQS 位可以选择对输入的硬复位信号进行滤波, 硬复位输入到 CRM 模块, CRM 模块为各个模块产生复位信号, 对全芯片进行复位, 复位结束后芯片进入 boot 流程。

3.2.3.4 低电压检测复位

低电压检测电路对全芯片电压 VDD 进行监控, 当电压小于可编程的限位电压 V_{LVR} 时 (参见 19.2.6 章节), 低电压检测电路发出复位信号。

为了防止检测电路产生毛刺, 在检测电路中增加了滤波功能。

低电压检测电路产生复位请求信号后, 信号进入 CRM 模块, 在 CRM 模块为低电压复位信号设计了使能信号, 若寄存器 CRMCTRL 的 LVRRSTEN 位配置为 0, 复位请求信号被屏蔽, 不会对全芯片进行复位; 若寄存器 CRMCTRL 的 LVRRSTEN 位配置为 1, 则低电压检测电路发送过来的复位请求信号有效, CRM 模块为各个模块产生复位信号, 复位结束后, JMT51 核从 FLASH 的 0 地址开始执行应用程序。

3.2.3.5 看门狗(WDT)复位

上电复位后，WDT 默认处于工作状态，当 WDT 产生复位请求信号后，将复位请求信号送入 CRM 模块，CRM 模块为各个模块产生复位信号，对全芯片进行复位，复位结束后，JMT51 核从 FLASH 的 0 地址开始执行应用程序。

3.2.3.6 全局软复位

JMT1801ED 支持软件配置全局复位功能，将 GLOBALRST 寄存器配置为 0x01，对全芯片进行复位，硬件自动清除 GLOBALRST 寄存器，复位结束后，JMT51 核从 FLASH 的 0 地址开始执行应用程序。

3.2.3.7 电源分区复位

在 SLEEP 或 DEEP SLEEP 模式下，可断电区域的模块断电，重新唤醒后，芯片退出 SLEEP 或 DEEP SLEEP 模式，产生电源分区复位，对断电区域进行复位，JMT51 核从 FLASH 的 0 地址开始执行应用程序。

3.2.3.8 模块软复位

各个模块均具有了软件复位功能，当某模块的复位寄存器置 1 后，将会产生复位信号对该模块进行复位，硬件自动清除该复位寄存器。软件复位寄存器的详细描述参见 Table 27 和 Table 28。

3.2.3.9 复位信号对各模块的作用

Table 23 列出了复位信号对各模块的复位情况，表中 (1) 表示电源分区复位只对 CRM 模块的部分单元复位，其中复位的单元和寄存器有：

- GPIO 时钟分频寄存器 GPIO_CLK_DIV、时钟使能寄存器 CLK_EN1.GPIOCEN 和 GPIO 时钟产生单元。
- SPI 时钟使能寄存器 CLK_EN2.SPICEN 和 SPI 时钟产生单元。
- I2C 时钟使能寄存器 CLK_EN2.I2CCEN 和 SPI 时钟产生单元。
- 模拟比较器滤波时钟控制寄存器 CMP_CLK_CTRL 和模拟比较器滤波时钟产生单元。
- PWM 时钟分频寄存器 PWM_CLK_DIV、时钟使能寄存器 CLK_EN2.PWMCEN 和 PWM 时钟产生单元。
- DSP 核、TIMER、UART0、URAT1 和 CORDIC 的时钟使能寄存器及时钟门控单元。

Table 23 中 (2) 表示硬复位不对 nrst 管脚的滤波模块和滤波控制寄存器 NRSTFQS 复位，对 GPIO 的其他部分复位。

Table 23 复位信号对各模块的作用一览表

模块	POR_RST	HW_RST	BOR_RST	LVR_RST	WDT_RST	GALOBAL_RST	POWER_RST
PMU 模块	复位	复位	复位	复位	复位	复位	不复位
WDT 模块	复位	复位	复位	复位	复位	复位	不复位
CRM 模块	复位	复位	复位	复位	复位	复位	部分复位 (1)
GPIO 模块	复位	部分复 位 (2)	复位	复位	复位	复位	复位
PWM 模块	复位	复位	复位	复位	复位	复位	复位
JMT51 模块	复位	复位	复位	复位	复位	复位	复位
DSP 模 块	复位	复位	复位	复位	复位	复位	复位
CORDI C 模块	复位	复位	复位	复位	复位	复位	复位
SPI 模 块	复位	复位	复位	复位	复位	复位	复位
I2C 模 块	复位	复位	复位	复位	复位	复位	复位
ADC 模块	复位	复位	复位	复位	复位	复位	复位
FLASH 控制模 块	复位	复位	复位	复位	复位	复位	复位
模拟比 较器滤	复位	复位	复位	复位	复位	复位	复位

模块	POR_R ST	HW_R ST	BOR_R ST	LVR_R ST	WDT_R ST	GALOBAL_R ST	POWER_R ST
波模块							
RTC 模块	复位	不复位	不复位	不复位	不复位	不复位	不复位
是否进入 boot 流程	是	是	否	否	否	否	否

3.2.4 寄存器

该模块共有 5 个寄存器，对应的 SFR PAGE 为 2，具体列表如 Table 24 所示。

Table 24 复位模块寄存器(SFRPAGE = 0x02)

地址	寄存器名	属性	复位值	功能描述
0XD2	CRMCTRL	W/R	0x04	CRM 模块控制
0xCE	RSTFLAG	W/R	0x20	复位标志
0xCB	SWRST0	W	0x00	模块软复位 0
0xCC	SWRST1	W	0x00	模块软复位 1
0xCD	GLOBALRST	W/R	0x00	全局软复位

3.2.4.1 CRM 模块控制寄存器(CRMCTRL) [0XD2]

Table 25 CRM 模块控制寄存器(CRMCTRL)

参数名	比特位	属性	复位值	描述
RESERVED	0	R	0	保留。
BORENCTRL	1	W/R	0	BOR 控制方式寄存器： 0：由软件配置 BORSWEN 寄存器，控制 BOR 模块的使能； 1：由硬件控制 BOR 模块的使能。芯片进入 DEEP SLEEP 模式时自动关闭 BOR，退出 DEEP SLEEP 模式时自动打开 BOR。
BORSWEN	2	W/R	1	BOR 模块使能寄存器：

参数名	比特位	属性	复位值	描述
				0: BOR 模块正常工作; 1: 关闭 BOR 模块。
LVRSTEN	3	W/R	0	LVR 复位信号使能控制: 0: LVR 复位信号无效; 1: LVR 复位信号有效。
CLKOUTCTRL	4	W/R	0	输出时钟选择寄存器: 0: 选择系统时钟作为输出时钟; 1: 选片内 32KHz 时钟作为输出时钟。
RESERVED	7-5	R	0	保留。

3.2.4.2 复位标识寄存器(RSTFLAG) [0xCE]

Table 26 复位标识寄存器 (RSTFLAG)

参数名	比特位	属性	复位值	描述
WDTRSTFLAG	0	W/R	0	看门狗(WDT)复位标志位: 0: 没有 WDT 复位请求; 1: 有 WDT 复位请求; 软件写 0 清除该标志位。
BORRSTFLAG	1	W/R	0	掉电复位(BOR)标志位: 0: 没有 BOR 复位请求; 1: 有 BOR 复位请求; 软件写 0 清除该标志位。
LVRSTFLAG	2	W/R	0	低电压复位(LVR)标志位: 0: 没有 LVR 复位请求; 1: 有 LVR 复位请求; 软件写 0 清除该标志位。
HWRSTFLAG	3	W/R	0	硬复位标志位:

参数名	比特位	属性	复位值	描述
				0: 没有硬复位请求; 1: 有硬复位请求; 软件写 0 清除该标志。
GLBRSTFLAG	4	W/R	0	全局软复位标识位: 0: 没有全局软复位请求; 1: 有全局软复位请求; 软件写 0 清除该标识位。
PORRSTFLAG	5	W/R	1	上电复位(POR_RST)标识位: 0: 没有 POR 复位请求; 1: 有 POR 复位请求; 软件写 0 清除该标识位。
POWRSTFLAG	6	W/R	0	电源分区复位标识位: 0: 没有电源分区复位请求; 1: 有电源分区复位请求; 软件写 0 清除该标识位。
RESERVED	7	R	0	保留。

3.2.4.3 模块软复位寄存器 0 (SWRST0) [0xCB]

Table 27 模块软复位寄存器 0 (SWRST0)

参数名	比特位	属性	复位值	描述
WDTSWRST	0	W	0	WDT 模块软复位寄存器: 0: 无效; 1: 对 WDT 模块进行复位; 硬件自动将此位置 0 清除复位。

参数名	比特位	属性	复位值	描述
RESERVED	1	R	0	保留。
PWMSWRST	2	W	0	PWM 模块软复位寄存器： 0: 无效； 1: 对 PWM 模块进行复位； 硬件自动将此位置 0 清除复位。
RESERVED	4-3	R	0	保留。
SPISWRST	5	W	0	SPI 模块软复位寄存器： 0: 无效； 1: 对 SPI 模块进行复位； 硬件自动将此位置 0 清除复位。
I2CSWRST	6	W	0	I2C 模块软复位寄存器： 0: 无效； 1: 对 I2C 模块进行复位； 硬件自动将此位置 0 清除复位。
RESERVED	7	R	0	保留。

3.2.4.4 模块软复位寄存器 1 (SWRST1) [0xCC]

Table 28 模块软复位寄存器 1 (SWRST1)

参数名	比特位	属性	复位值	描述
GPIO SWRST	0	W	0	GPIO 模块软复位寄存器： 0: 无效； 1: 对 GPIO 模块进行复位； 硬件自动将此位置 0 清除复位。
ADC SWRST	1	W	0	ADC 模块软复位寄存器： 0: 无效； 1: 对 ADC 模块进行复位；

参数名	比特位	属性	复位值	描述
				硬件自动将此位置 0 清除复位。
RESERVED	3-2	R	0	保留。
DSPSWRST	4	W	0	DSP 模块软复位寄存器： 0: 无效； 1: 对 DSP 模块进行复位； 硬件自动将此位置 0 清除复位。
RESERVED	7-5	R	0	保留。

3.2.4.5 全局软复位寄存器(GLOBALRST) [0xCD]

Table 29 全局软复位寄存器 (GLOBALRST)

参数名	比特位	属性	复位值	描述
GLOBSWRST	0	W	0	全芯片软复位寄存器： 0: 无效； 1: 对全芯片进行复位； 硬件自动将此位置 0 清除复位。
RESERVED	7-1	R	0	保留

4 JMT51 MCU 核

4.1 概述

JMT51 MCU 内核为增强型 51 内核，与 MCS-51 指令集完全兼容，可以使用标准 803x/805x 的汇编器和编译器进行软件开发。JMT51 采用流水线结构，与标准的 8051 结构相比指令执行速度有很大的提高。标准 8051，除 MUL 和 DIV 以外的所有指令都需要 12 或 24 个系统时钟周期，对于 JMT51 内核，大部分指令的执行时间为 1 或 2 个系统时钟周期。JMT51 时钟最高频率为 73.728MHz。

JMT51 MCU 支持通过 JTAG 接口对 JMT51 核进行调试。具体使用方法可参考 JMT1801ED JTAG 使用说明文档。需要说明的是，在 FLASH 的 NVR0 中存放了 JTAG 的使能标志（地址 0x90079008）。上电之后，Boot 程序会比较该使能标志，如果不等于 0x5AA5，则将管脚 PB4、PB5、PB6、PB7 分别复用为 TCK、TMS、TDI、TDO。如果等于 0x5AA5，PB4、PB5、PB6、PB7 四个管脚则被复用为正常功能。因此在进行量产的时候，可通过修改 NVR0 中的 JTAG 使能标志，将管脚复用为正常功能。

JMT51 MCU 内核堆栈位于内部 128 字节的数据存储器，因此堆栈深度最大可以达到 128。通过堆栈指针 SP 指示当前堆栈位置，当子程序调用或中断响应时，PC 会被压入堆栈中。在子程序或中断服务程序结束后，通过 RET 或 RETI 指令，PC 会重新从堆栈中获取到原先的值。如果数据存储器被用作堆栈区域，将不能作为通用数据存储器使用。

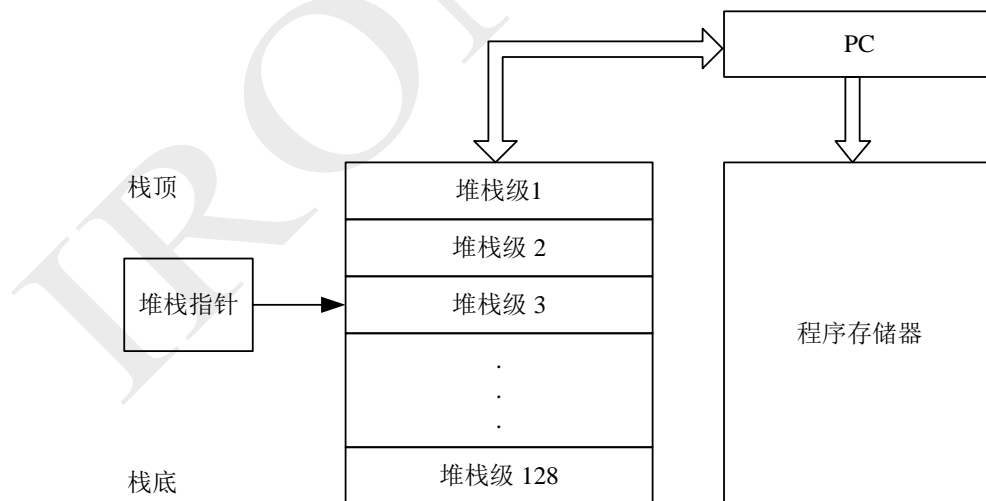


Figure 13 堆栈示意图

4.2 指令集

JMT51 MCU 核指令与标准 51 指令集完全兼容，可以使用标准 8051 的开发工具开发 JMT51 的软件。所有的 JMT51 指令在二进制码和功能上与同类的 MCS-51 产品完全等价，

包括操作码、寻址方式和对 PSW 标志的影响。

指令共分为五大类，算术运算类、逻辑运算类、数据传输类、程序控制类、布尔处理类：

- 算术运算类包括加法、带进位加法、带借位减法、自增、自减及乘法指令，运算都是按字节进行的。
- 逻辑运算类包含按位与、按位或、按位异或、左移、右移、对累加器清 0、取反等操作。
- 数据传输类包括立即数到直接地址、间接地址、外部数据存储器的传输，内部数据存储器到外部数据存储器之间的传输，内部数据存储器到程序存储器之间的传输，压栈/出栈。
- 程序控制类指令包括绝对跳转、相对跳转、子程序调用、有条件跳转、子程序返回。
- 布尔处理类指令包括位清 0、置 1、取反、与、或、传输等。

Table 30 指令集中符号说明

Rn	通用寄存器 R0~R7
Direct	8 比特直接地址，内部数据存储器低 128 字节和 SFR 区域地址 00~FFH
@Ri	间接寻址，Ri 中存放的是内部数据存储器的地址
#data	8 位立即数
#data 16	16 位立即数
addr 16	16 位地址，用于跳转或调用，指示整个 32K 字节程序空间
addr 11	11 位地址，用于跳转或调用，地址范围为 2K 字节
Rel	相对跳转 8 位有符号地址，地址范围-128~+127
Bit	位地址，内部数据存储器 128 位和 SFR 中可位寻址区域

Table 31 算术类指令

助记符	描述	指令码	字节
ADD A,Rn	寄存器与 A 相加	28~2F	1
ADD A,direct	直接地址中的数与 A 相加	25	2
ADD A,@Ri	间接地址中的数与 A 相加	26~27	1
ADD A,#data	立即数与 A 相加	24	2
ADDC A,Rn	寄存器与 A 带进位相加	38~3F	1
ADDC A,direct	直接地址中的数与 A 带进位相加	35	2
ADDC A,@Ri	间接地址中的数与 A 带进位相加	36~37	1
ADDC A,#data	立即数与 A 带进位相加	34	2
SUBB A,Rn	A 带借位减寄存器	98~9F	1
SUBB A,direct	A 带借位减直接地址中的数	95	2

助记符	描述	指令码	字节
SUBB A,@Ri	A 带借位减间接地址中的数	96~97	1
SUBB A,#data	A 带借位减立即数	94	2
INC A	A 自增 1	04	1
INC Rn	寄存器中的数自增 1	08~0F	1
INC direct	直接地址中的数自增 1	05	2
INC @Ri	间接地址中的数自增 1	06~07	1
INC DPTR	数据指针自增 1	A3	1
DEC A	A 自减 1	14	1
DEC Rn	寄存器中的数自减 1	18~1F	1
DEC direct	直接地址中的数自减 1	15	2
DEC @Ri	间接地址中的数自减 1	16~17	1
MUL AB	A 与 B 相乘	A4	1
DIV	A 除与 B	84	1
DAA	A 中数十进制调整	D4	1

Table 32 逻辑运算类指令

助记符	描述	指令码	字节
ANL A,Rn	A 与寄存器按位与	58~5F	1
ANL A,direct	A 与直接地址中的数按位与	55	2
ANL A,@Ri	A 与间接地址中的数按位与	56~57	1
ANL A,#data	A 与立即数按位与	54	2
ANL direct,A	直接地址与 A 按位与	52	2
ANL direct,#data	直接地址与立即数按位与	53	3
ORL A,Rn	A 与寄存器按位或	48~4F	1
ORL A,direct	A 与直接地址中的数按位或	45	2
ORL A,@Ri	A 与间接地址中的数按位或	46~47	1
ORL A,#data	A 与立即数按位或	44	2
ORL direct,A	直接地址与 A 按位或	42	2
ORL direct,#data	直接地址与立即数按位或	43	3
XRL A,Rn	A 与寄存器按位异或	68~6F	1
XRL A,direct	A 与直接地址中的数按位异或	65	2
XRL A,@Ri	A 与间接地址中的数按位异或	66~67	1

助记符	描述	指令码	字节
XRL A,#data	A 与立即数按位异或	64	2
XRL direct,A	直接地址与 A 按位异或	62	2
XRL direct,#data	直接地址与立即数按位异或	63	3
CLR A	A 清 0	E4	1
CPL A	A 按位取反	F4	1
RL A	A 左移 1 位	23	1
RLC A	A 带进位左移 1 位	33	1
RR A	A 右移 1 位	03	1
RRC A	A 带进位右移 1 位	13	1
SWAP A	A 中高/低半字节数据交换	C4	1

Table 33 数据传输类指令

助记符	描述	指令码	字节
MOV A,Rn	寄存器数到 A	E8~EF	1
MOV A,direct	直接地址中的数到 A	E5	2
MOV A,@Ri	间接地址中的数到 A	E6~E7	1
MOV A,#data	立即数数到 A	E4	2
MOV Rn,A	A 到寄存器	F8~FF	1
MOV Rn,direct	直接地址到寄存器	A8~AF	2
MOV Rn,#data	立即数到寄存器	78~7F	2
MOV direct,A	A 到直接地址	F5	2
MOV direct,Rn	寄存器到直接地址	88~8F	2
MOV direct1,direct2	直接地址 2 到直接地址 1	85	3
MOV direct,@Ri	间接地址中的数到直接地址	86~87	2
MOV direct,#data	立即数到直接地址	75	3
MOV @Ri,A	A 到间接地址	F6~F7	1
MOV @Ri,direct	直接地址中的数到间接地址	A6~A7	2
MOV @Ri,#data	立即数到间接地址	76~77	2
MOV DPTR,#data16	16 位立即数到 DPTR	90	3
MOVC A,@A+DPTR	A+DPTR 程序地址中的数到 A	93	1
MOVC A,@A+PC	A+PC 程序地址中的数到 A	83	1
MOVX A,@Ri	外部间接地址空间(8 位)数到 A	E2~E3	1

助记符	描述	指令码	字节
MOVX A,@DPTR	外部间接地址空间(16位)数到 A	E0	1
MOVX @Ri,A	A 到外部间接地址空间(8位)	F2~F3	1
MOVX @DPTR,A	A 到外部间接地址空间(16位)	F0	1
PUSH direct	直接地址中的数压栈	C0	2
POP direct	堆栈中数出栈到直接地址	D0	2
XCH A,Rn	A 与寄存器数交换	C8~CF	1
XCH A,direct	A 与直接地址中的数交换	C5	2
XCH A,@Ri	A 与间接地址中的数交换	C6~C7	1
XCHD A,@Ri	A 与间接地址中的数的低半字节数交换	C6~C7	1

Table 34 程序控制类指令

助记符	描述	指令码	字节
ACALL addr11	绝对子程序调用, 地址范围 2K 字节	xxx10001	2
LCALL addr16	长调用, 地址范围 64K 字节	12	3
RET	子程序返回	22	1
RETI	中断子程序返回	32	1
AJMP addr11	绝对跳转, 地址范围 2K 字节	xxx00001	2
LJMP addr16	长跳转, 地址范围 64K 字节	02	3
SJMP rel	相对跳转, 地址范围-128~+127	80	2
JMP @A+DPTR	跳转到 A+DPTR 地址	73	1
JZ rel	如果 A=0, 跳转到相对地址	60	2
JNZ rel	如果 A! =0, 跳转到相对地址	70	2
JC rel	如果 C=1, 跳转到相对地址	40	2
JNC rel	如果 C=0, 跳转到相对地址	50	2
JB bit,rel	如果位地址数=1, 跳转到相对地址	20	3
JNB bit,rel	如果位地址数! =1, 跳转到相对地址	30	3
JBC bit, el	如果位地址数=1, 清除后, 跳转到相对地址	10	3
CJNE A,direct,rel	如果 A 与直接地址中数不等, 跳转到相对地址	B5	3
CJNE A,#data,rel	如果 A 与立即数不等, 跳转到相对地址	B4	3
CJNE Rn,#data,rel	如果寄存器数与立即数不等, 跳转到相对地址	B8~BF	3

助记符	描述	指令码	字节
CJNE @Ri,#data,rel	如果间接地址的数与立即数不等, 跳转到相对地址	B6~B7	3
DJNZ Rn,rel	寄存器数自减 1, 不等于 0, 跳转到相对地址	D8~DF	2
DJNZ direct,rel	直接地址中的数自减 1, 不等于 0, 跳转到相对地址	D5	3
NOP	空操作	00	1

Table 35 布尔类指令

助记符	描述	指令码	字节
CLR C	C 置 0	C3	1
CLR bit	位地址的数置 0	C2	2
SETB C	C 置 1	D3	1
SETB bit	位地址的数置 1	D2	2
CPL C	C 取反	B3	1
CPL bit	位地址的数取反	B2	2
ANL C,bit	位地址的数与 C 相与	82	2
ANL C,/bit	位地址的数取反与 C 相与	B0	2
ORL C,bit	位地址的数与 C 相或	72	2
ORL C,/bit	位地址的数取反与 C 相或	A0	2
MOV C,bit	位地址的数赋值到 C	A2	2
MOV bit,C	C 赋值到位地址	92	2

5 JMT018 DSP 核

5.1 概述

JMT018 DSP 核由三个并行计算单元(ALU, MAC, DIV)、一个独立地址发生器、一个程序控制器和寄存器文件组成。

DSP 指令为固定 16 位长度指令, 全部为单字指令。大部分指令为单周期指令, 每个时钟周期执行一条指令, 跳转及 MAC 指令为多周期指令。

5.1.1 框图

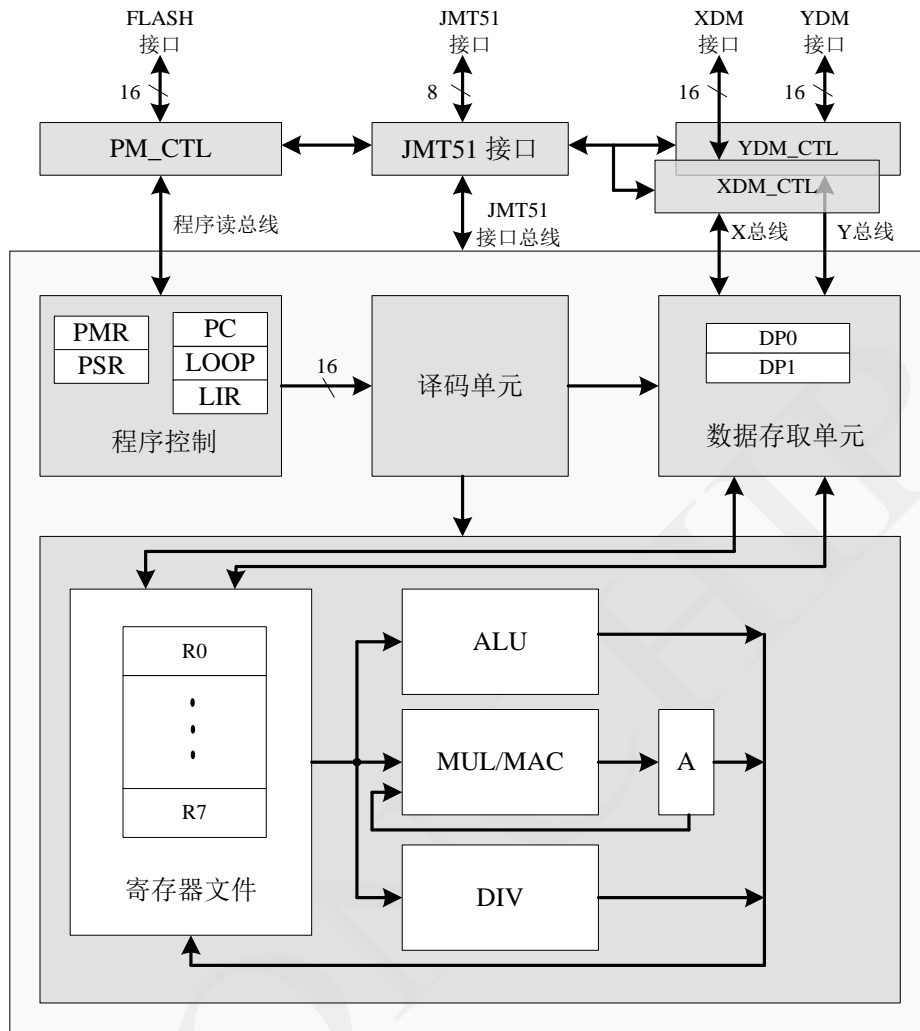


Figure 14 JMT018 DSP 核框图

5.1.2 系统总线

JMT018 DSP 核包含程序读总线, X 数据空间总线, Y 数据空间总线和 JMT51 接口总线。

程序读总线, 用于访问 DSP 的程序空间 (FLASH), 由程序控制单元产生。一个周期完成一次读。

X 数据空间总线, 用于读写 X 数据空间存储器 (DM0) 和数据寄存器 (CORDIC/ADC), 由 DP0 数据指针寄存器产生读写地址, 读写数据与通用寄存器 R0~R7 交互。一个周期完成一次读或者写。

Y 数据空间总线, 用于读写 Y 数据空间存储器 (DM1), 由 DP1 数据指针寄存器产生读写地址, 读写数据与通用寄存器 R0~R7 交互。一个周期完成一次读或者写。

X 数据空间和 Y 数据空间可以并行读写。

5.1.3 运算单元 (ALU)

运算单元包含加减运算、逻辑运算、数据移位、数据比较、求最大值、求最小值、限幅等运算。

操作数来源为通用寄存器 R0~R7，运行结果写入通用寄存器 R0~R7。

加减运算影响进位或借位标志位 (C)、零结果标志位 (Z) 及溢出标志位 (OV)；比较运算影响比较结果标志位 (CF)；移位运算影响进位或借位标志位 (C) 和零结果标志位 (Z)，可以进行算术及逻辑移位，移位的位数由寄存器或立即数指定。

加减运算指令支持并行执行，在读数据同时，进行加减等运算。

5.1.4 乘累加单元(MAC)

乘累加单元主要完成乘法、乘加运算、乘减运算等运算。

乘法运算支持：

- ◆ 16bit x 16bit 有符号整数；
- ◆ 16bit x 16bit 无符号整数；
- ◆ 16bit x 16bit 有符号小数；
- ◆ 16bit x 16bit 有符号小数带四舍五入。

乘加运算支持：

- ◆ 40bit + 16bit x 16bit 有符号整数；
- ◆ 40bit + 16bit x 16bit 有符号小数；
- ◆ 40bit + 16bit x 16bit 有符号小数带四舍五入。

乘减运算支持：

- ◆ 40bit - 16bit x 16bit 有符号整数；
- ◆ 40bit - 16bit x 16bit 有符号小数；
- ◆ 40bit - 16bit x 16bit 有符号小数带四舍五入。

5.1.5 除法单元 (DIV)

除法单元完成有符号或无符号整数除法运算，被除数为 32 位，有两个寄存器组成被除数，除数为 16 位，运算结果由余数和商组成。运算的商写入被除数的位置，余数写入除数位置。

5.1.6 硬件循环 (HW loop)

支持硬件循环指令，通过 LOOP 寄存器设置循环次数，LIR 寄存器设置循环的指令个数，当执行到 doloop 指令时，会循环执行由寄存器 LOOP 和 LIR 所描述的指令。

5.1.7 数据传输

JMT018 支持的数据传输类型有：

- ◆ 立即数到通用寄存器；
- ◆ 立即数到数据指针寄存器 (DP0/DP1)；
- ◆ 存储器 (DM0/DM1) 和数据寄存器 (CORDIC/ADC) 到通用寄存器；
- ◆ 通用寄存器到存储器或数据寄存器；
- ◆ 通用寄存器到通用寄存器；
- ◆ 通用寄存器到数据指针寄存器 (DP0/DP1)；
- ◆ 通用寄存器到状态寄存器 (PSR)；
- ◆ 通用寄存器到模式寄存器 (PMR)；
- ◆ 通用寄存器到硬件循环寄存器 (LOOP、LIR)；
- ◆ 通用寄存器到 A 寄存器；
- ◆ A 寄存器到通用寄存器。

5.1.8 存储单元

JMT018 可访问的存储单元包括程序存储器 (FLASH), X 数据存储器 (DM0) 和数据寄存器 (CORDIC/ADC), Y 数据存储器 (DM1), 具体参见章节 7.2 存储器描述。

XDM 数据读写、YDM 数据读写、MAC/ALU 运算, 三者可以并行操作。

5.2 指令集

JMT018 指令码采用固定长度 16 位编码, 根据指令的功能, 可以分为数据传输类、算术运算类、移位运算类、逻辑运算类、乘法运算类、并行运算类和程序控制类等指令类型。

指令中的 Rn、Rm、Rd 为 16 位通用寄存器 R0~R7; DP0、DP1 为 16 位数据指针寄存器; A 为 40 位累加器; A2 为 40 位累加器 A 的高 8 位; A1 为 40 位累加器 A 的高字 16 位; A0 为 40 位累加器 A 的低字 16 位; #imd 为立即数; label 为程序段名 (以字母开头, 以冒号结尾的字符串) 指示的 16bit 无符号数。

5.2.1 数据传输指令

数据传输指令如下表格所述。

Table 36 数据传输指令

指令	周期	标志位				描述
		C	OV	Z	CF	
MOV DP0,#imd(12bit)	1					#imd(12bit)->DP0
MOV DP1,#imd(12bit)	1					#imd(12bit)->DP1
MOVL Rd,#imd(8bit)	1					#imd(8bit)->Rdl
MOVH Rd,#imd(8bit)	1					#imd(8bit)->Rdh
MOV Rd,Rn	1					Rn->Rd

指令	周期	标志位				描述
		C	OV	Z	CF	
MOV Rd,A2	1					A2(符号扩展)->Rd
MOV Rd,A1	1					A1->Rd
MOV Rd,A0	1					A0->Rd
MOVR Rd,A	1					A(四舍五入)->Rd
MOVS Rd,A	1					A(饱和)->Rd
MOV Rd,DP0	1					DP0->Rd
MOV Rd,DP1	1					DP1->Rd
MOV Rd,PSR	1					PSR->Rd
MOV Rd,PMR	1					PMR->Rd
MOV Rd,LNKR	1					LNKR -> Rd
MOV Rd,XDM	2					存储器 XDM[DP0]->Rd
MOV Rd,XDM+	2					存储器 XDM[DP0]->Rd; 地址指针 DP0 自增 1
MOV Rd,XDM-	2					存储器 XDM[DP0]->Rd, 地址指针 DP0 自减 1
MOV Rd,YDM	2					存储器 YDM[DP1]->Rd
MOV Rd,YDM+	2					存储器 YDM[DP1]->Rd, 地址指针 DP1 自增 1
MOV Rd,YDM-	2					存储器 YDM[DP1]->Rd, 地址指针 DP1 自减 1
MOV A2,Rn	1					Rn (低 8 位) ->A2
MOV A1,Rn	1					Rn->A1
MOV A0,Rn	1					Rn->A0
MOV DP0,Rn	1					Rn->DP0
MOV DP1,Rn	1					Rn->DP1
MOV LOOP,Rn	1					Rn->LOOP
MOV LIR,Rn	1					Rn->LIR
MOV PSR,Rn	1	是	是	是	是	Rn->PSR
MOV PMR,Rn	1					Rn->PMR
MOV LNKR,Rn	1					Rn -> LNKR

指令	周期	标志位				描述
		C	OV	Z	CF	
MOV XDM,Rn	1					Rn->存储器 XDM[DP0]
MOV XDM+,Rn	1					Rn->存储器 XDM[DP0], 地址指针 DP0 自增 1
MOV XDM-,Rn	1					Rn->存储器 XDM[DP0], 地址指针 DP0 自减 1
MOV YDM, Rn	1					Rn->存储器 YDM[DP1]
MOV YDM+,Rn	1					Rn->存储器 YDM[DP1], 地址指针 DP1 自增 1
MOV YDM-,Rn	1					Rn->存储器 YDM[DP1], 地址指针 DP1 自减 1
MOV Rd,#imd(16bit)	2					#imd(低 8bit)->Rdl #imd(高 8bit)->Rdh
MOV Rd,label	2					label(低 8bit)->Rdl label(高 8bit)->Rdh

5.2.2 算术运算指令

算术运算指令如下表格所述。

Table 37 算术运算指令

指令	周期	标志位				描述
		C	OV	Z	CF	
MAX Rd,Rn,Rm	1					选择寄存器中较大的数据（有符号数）
MIN Rd,Rn,Rm	1					选择寄存器中较小的数据（有符号数）
LIM Rd,Rn,Rm	1					对寄存器数据门限控制（有符号数）
DIV Rd,Rn,Rm	17		是			无符号/有符号整数除
ADD Rd,Rn	1	是	是	是		加法 Rd+Rn->Rd
ADDC Rd,Rn	1	是	是	是		带进位加 Rd+Rn+C->Rd
SUB Rd,Rn	1	是	是	是		减法 Rd-Rn->Rd
SUBB Rd,Rn	1	是	是	是		带进位减 Rd-Rn-C->Rd
CLB Rd,Rn	1					求寄存器 Rn 中前导 0/1 个数
CLB Rd,A	1					求寄存器 A 中前导 0/1 个数
LT Rn,Rm	1				是	有符号数比较小于

指令	周期	标志位				描述
		C	OV	Z	CF	
LTU Rn,Rm	1				是	无符号数比较小于
LE Rn,Rm	1				是	有符号数比较小于等于
LEU Rn,Rm	1				是	无符号数比较小于等于
GT Rn,Rm	1				是	有符号数比较大于
GTU Rn,Rm	1				是	无符号数比较大于
GE Rn,Rm	1				是	有符号数比较大于等于
GEU Rn,Rm	1				是	无符号数比较大于等于
EQ Rn,Rm	1				是	比较等于
NE Rn,Rm	1				是	比较不等于
INC Rd	1			是		寄存器 Rd 自增 1
INC DP0	1					寄存器 DP0 自增 1
INC DP1	1					寄存器 DP1 自增 1
DEC Rd	1			是		寄存器 Rd 自减 1
DEC DP0	1					寄存器 DP0 自减 1
DEC DP1	1					寄存器 DP1 自减 1
NEG Rd	1		是			寄存器 Rd 数据取负
ABS Rd	1		是			取寄存器 Rd 中数据的绝对值
CLR Rd	1					寄存器 Rd 清 0
ROUND A	1					寄存器 A 四舍五入运算
SAT A	1					寄存器 A 饱和运算
CLR A	1					寄存器 A 清 0
REV Rd,#imd	1					寄存器 Rd 中数据根据立即数(4 位)反转
BITCLR PSR,#imd	1	是	是	是	是	寄存器 PSR 按比特清 0
BITSET PSR,#imd	1	是	是	是	是	寄存器 PSR 按比特置 1

5.2.3 移位运算指令

移位运算指令如下表格所述。

Table 38 移位运算指令

指令	周期	标志位				描述
		C	OV	Z	CF	
ASHR Rd,Rn	1	是		是		算术右移 Rn 位

指令	周期	标志位				描述
		C	OV	Z	CF	
ASHL Rd,Rn	1	是		是		算术左移 Rn 位
LSHR Rd,Rn	1	是		是		逻辑右移 Rn 位
LSHR Rd,Rn	1	是		是		逻辑左移 Rn 位
ASHR Rd,#imd	1	是		是		算术右移#imd 位
ASHL Rd,#imd	1	是		是		算术左移#imd 位
LSHR Rd,#imd	1	是		是		逻辑右移#imd 位
LSHR Rd,#imd	1	是		是		逻辑左移#imd 位

5.2.4 逻辑运算指令

逻辑运算指令包括按位与、按位或、按位异或、按位取反，如下表格所述。

Table 39 逻辑运算指令

指令	周期	标志位				描述
		C	OV	Z	CF	
AND Rd,Rn	1			是		按位与 Rn&Rd -> Rd
OR Rd,Rn	1			是		按位或 Rn Rd -> Rd
XOR Rd,Rn	1			是		按位异或 Rn^Rd -> Rd
CPL Rd	1					按位取反~Rd -> Rd

5.2.5 乘法运算指令

乘法运算指令如下表格所述。

Table 40 乘法运算指令

指令	周期	标志位				描述
		C	OV	Z	CF	
IMULU Rn,Rm	2			是		无符号整数乘 Rn*Rm->A
IMUL Rn,Rm	2			是		有符号整数乘 Rn*Rm->A
MUL Rn,Rm	2			是		有符号小数乘 Rn*Rm->A
MULR Rn,Rm	2			是		带四舍五入的有符号小数乘 Rn*Rm->A
IMAC Rn,Rm	2		是	是		有符号整数乘加 A+Rn*Rm->A
IMACN Rn,Rm	2		是	是		有符号整数乘减 A-Rn*Rm->A
MAC Rn,Rm	2		是	是		有符号小数乘加
MACN Rn,Rm	2		是	是		有符号小数乘减

指令	周期	标志位				描述
		C	OV	Z	CF	
MACR Rn,Rm	2		是	是		带四舍五入的有符号小数乘加
MACNR Rn,Rm	2		是	是		带四舍五入的有符号小数乘减

5.2.6 并行运算指令

并行运算指令如下表格所述。

Table 41 并行运算指令

指令	周期	标志位				描述
		C	OV	Z	CF	
ADD Rd,Rn:XDM,Rm:YDM	2	是	是	是		加法并行： Rn+Rm->Rd; XDM[DP0]->Rn; YDM[DP1]->Rm
ADDC Rd,Rn:XDM,Rm:YDM	2	是	是	是		带进位加并行： Rd+Rn+C->Rd; XDM[DP0]->Rn; YDM[DP1]->Rm
SUB Rd,Rn:XDM,Rm:YDM	2	是	是	是		减法并行： Rn-Rm->Rd; XDM[DP0]->Rn; YDM[DP1]->Rm
SUBB Rd,Rn:XDM,Rm:YDM	2	是	是	是		带进位减并行： Rd-Rn-C->Rd; XDM[DP0]->Rn; YDM[DP1]->Rm
IMULU Rn:XDM+,Rm:YDM+	2			是		无符号整数乘并行： Rn*Rm->A; XDM[DP0]->Rn, DP0 自增 1; YDM[DP1]->Rm, DP1 自增 1
IMULU Rn:XDM-,Rm:YDM-	2			是		无符号整数乘并行： Rn*Rm->A; XDM[DP0]->Rn, DP0 自减 1;

指令	周期	标志位				描述
		C	OV	Z	CF	
						YDM[DP1]->Rm, DP1 自减 1
IMUL Rn:XDM+,Rm:YDM+	2			是		有符号整数乘并行: Rn*Rm->A; XDM[DP0]->Rn, DP0 自增 1; YDM[DP1]->Rm, DP1 自增 1
IMUL Rn:XDM-,Rm:YDM-	2			是		有符号整数乘并行: Rn*Rm->A; XDM[DP0]->Rn, DP0 自减 1; YDM[DP1]->Rm, DP1 自减 1
MUL Rn:XDM+,Rm:YDM+	2			是		有符号小数乘并行: Rn*Rm->A; XDM[DP0]->Rn, DP0 自增 1; YDM[DP1]->Rm, DP1 自增 1
MUL Rn:XDM-,Rm:YDM-	2			是		有符号小数乘并行: Rn*Rm->A; XDM[DP0]->Rn, DP0 自减 1; YDM[DP1]->Rm, DP1 自减 1
MULR Rn:XDM+,Rm:YDM+	2			是		带四舍五入的有符号小数乘并行: (Rn*Rm) 四舍五入->A; XDM[DP0]->Rn, DP0 自增 1; YDM[DP1]->Rm, DP1 自增 1
MULR Rn:XDM-,Rm:YDM-	2			是		带四舍五入的有符号小数乘并行: (Rn*Rm) 四舍五入->A; XDM[DP0]->Rn, DP0 自减 1; YDM[DP1]->Rm, DP1 自减 1
IMAC Rn:XDM+,Rm:YDM+	2		是	是		有符号整数乘加并行: A+Rn*Rm->A; XDM[DP0]->Rn, DP0 自增 1; YDM[DP1]->Rm, DP1 自增 1

指令	周期	标志位				描述
		C	OV	Z	CF	
IMAC Rn:XDM-,Rm:YDM-	2		是	是		有符号整数乘加并行： $A+Rn*Rm \rightarrow A$; XDM[DP0] \rightarrow Rn, DP0 自减 1; YDM[DP1] \rightarrow Rm, DP1 自减 1
IMACN Rn:XDM+,Rm:YDM+	2		是	是		有符号整数乘减并行： $A-Rn*Rm \rightarrow A$; XDM[DP0] \rightarrow Rn, DP0 自增 1; YDM[DP1] \rightarrow Rm, DP1 自增 1
IMACN Rn:XDM-,Rm:YDM-	2		是	是		有符号整数乘减并行： $A-Rn*Rm \rightarrow A$; XDM[DP0] \rightarrow Rn, DP0 自减 1; YDM[DP1] \rightarrow Rm, DP1 自减 1
MAC Rn:XDM+,Rm:YDM+	2		是	是		有符号小数乘加并行： $A+Rn*Rm \rightarrow A$; XDM[DP0] \rightarrow Rn, DP0 自增 1; YDM[DP1] \rightarrow Rm, DP1 自增 1
MAC Rn:XDM-,Rm:YDM-	2		是	是		有符号小数乘加并行： $A+Rn*Rm \rightarrow A$; XDM[DP0] \rightarrow Rn, DP0 自减 1; YDM[DP1] \rightarrow Rm, DP1 自减 1
MACN Rn:XDM+,Rm:YDM+	2		是	是		有符号小数乘减并行： $A-Rn*Rm \rightarrow A$; XDM[DP0] \rightarrow Rn, DP0 自增 1; YDM[DP1] \rightarrow Rm, DP1 自增 1
MACN Rn:XDM-,Rm:YDM-	2		是	是		有符号小数乘减并行： $A-Rn*Rm \rightarrow A$; XDM[DP0] \rightarrow Rn, DP0 自减 1; YDM[DP1] \rightarrow Rm, DP1 自减 1
MACR Rn:XDM+,Rm:YDM+	2		是	是		带四舍五入的有符号小数乘加并行： $(A+Rn*Rm)$ 四舍五入 $\rightarrow A$;

指令	周期	标志位				描述
		C	OV	Z	CF	
						XDM[DP0]->Rn, DP0 自增 1; YDM[DP1]->Rm, DP1 自增 1
MACR Rn:XDM-,Rm:YDM-	2		是	是		带四舍五入的有符号小数乘加并行: (A+Rn*Rm) 四舍五入->A; XDM[DP0]->Rn, DP0 自减 1; YDM[DP1]->Rm, DP1 自减 1
MACNR Rn:XDM+,Rm:YDM+	2		是	是		带四舍五入的有符号小数乘减并行: (A-Rn*Rm) 四舍五入->A; XDM[DP0]->Rn, DP0 自增 1; YDM[DP1]->Rm, DP1 自增 1
MACNR Rn:XDM-,Rm:YDM-	2		是	是		带四舍五入的有符号小数乘减并行: (A-Rn*Rm) 四舍五入->A; XDM[DP0]->Rn, DP0 自减 1; YDM[DP1]->Rm, DP1 自减 1

5.2.7 程序控制指令

程序控制指令包括无条件跳转、有条件跳转、硬件循环、停止指令，如下表格所述。

Table 42 程序控制指令

指令	周期	标志位				描述
		C	OV	Z	CF	
JMP Rn	2					绝对跳转, Rn->PC
JMP Rn,label	4					label(低 8bit)->Rnl label(高 8bit)->Rnh 绝对跳转, Rn->PC
JMP #imd	2					绝对跳转, #imd(10bit)->PC
JMPR Rn	2					相对跳转, Rn+PC->PC
JMPR Rn,label	4					label-2(低 8bit)->Rnl label-2(高 8bit)->Rnh

指令	周期	标志位				描述
		C	OV	Z	CF	
						相对跳转, Rn+PC->PC
JC Rn	2					条件绝对跳转, Rn->PC
JC Rn,label	4					label(低 8bit)->Rnl label(高 8bit)->Rnh 条件绝对跳转, Rn->PC
JC #imd	2					条件绝对跳转, #imd(10bit)->PC
JCR Rn	2					条件相对跳转, Rn+PC->PC
JCR Rn,label	4					label-2(低 8bit)->Rnl label-2(高 8bit)->Rnh 条件相对跳转, Rn+PC->PC
CAL Rn	2					子程序调用, Rn->PC
CAL Rn,label	4					label(低 8bit)->Rnl label(高 8bit)->Rnh 子程序调用, Rn->PC
CAL #imd	2					子程序调用, #imd(10bit)->PC
RET	2					子函数返回, LNKR->PC
DOLOOP	2					硬件循环
LOOPBK	1					强制退出当前 DOLOOP 循环
STCDC	19					Cordic 指令
STOP	1					停止指令
NOP	1					空指令

5.3 数据运算器

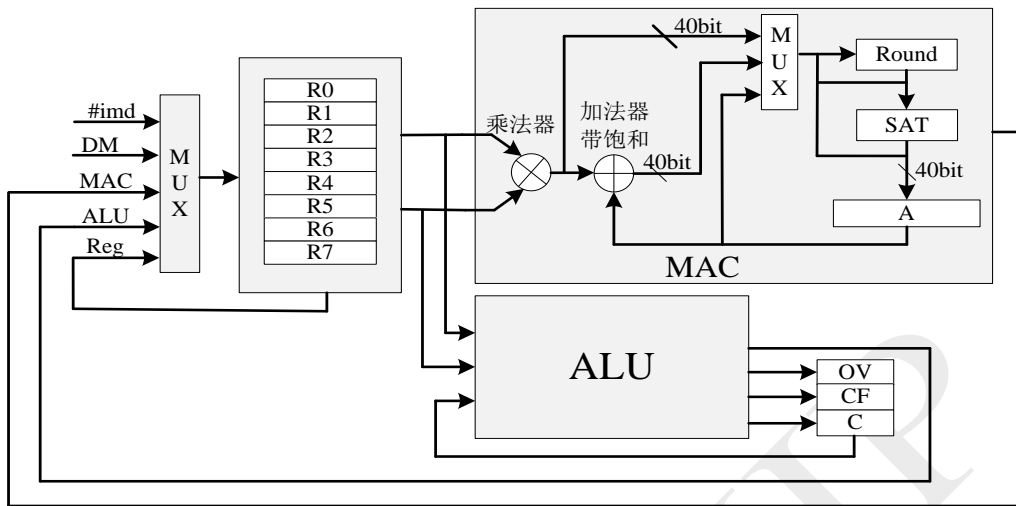


Figure 15 数据运算器框图

5.3.1 通用寄存器

共包含 8 个 16 位的通用寄存器 R0~R7。通用寄存器可以直接与存储器进行数据交互，同时寄存器之间也可数据交互，其他寄存器同样也可以与通用寄存器数据交互。

通用寄存器可以作为算术运算加减、逻辑运算与、或、异或及其他操作的源操作数和目的的操作数。

乘法器使用通用寄存器作为源操作数，目的操作数使用累加器 A。

除法器的被除数为 32 位，使用通用寄存器中的两个寄存器组成 32 位数据，除数为 16 位，除法运算完成后，商存入被除数，余数存入除数。

5.3.2 累加器 A

累加器 A 为 40 位，作为乘法运算的目的操作数，又作为乘加的源操作数和目的寄存器。累加器 A 由 3 部分组成：8 位扩展寄存器 A2，16 位高字寄存器 A1，16 位低字寄存器 A0。

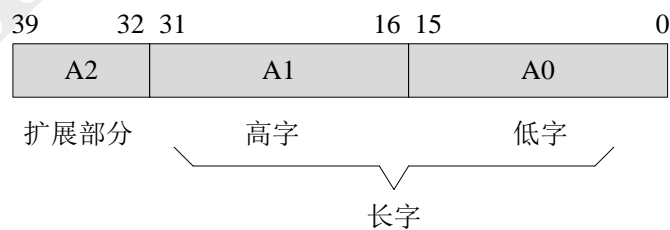


Figure 16 累加器 A 组成

A 作为整体表示整数时，最高位 39 位为符号位，其他为有效数字。表示的范围为 $[-2^{39}, 2^{39}-1]$ 。在整数乘法时，乘积结果存入 A1A0，扩展部分 A2 自动符号扩展。

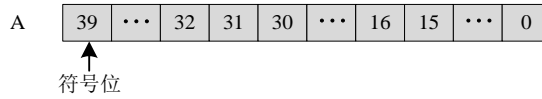


Figure 17 累加器 A 作为整数时数据结构

A 作为整体表示小数时，最高位 39 位为符号位，小数点位置在 31 位后面，表示小数范围[-256.0,255.999999995]。在小数乘法时，乘积结果存入 A1A0，扩展部分 A2 自动符号扩展。

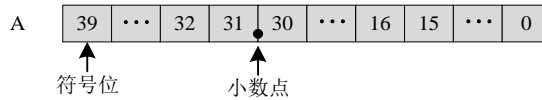


Figure 18 累加器 A 作为小数时数据结构

当 A 作为整体时，可以将 A 赋值给通用寄存器，有两种方式，一种是饱和赋值，一种四舍五入赋值。

MOVS Rd,A 指令，饱和赋值。将累加器 A 作为整个寄存器判断，如果 $A < -32768$ 时， $Rd=0x8000$ ；如果 $A > 32767$ 时， $Rd=0x7fff$ ；其他值时， $Rd=A$ 。赋值后，A 寄存器值保持不变。

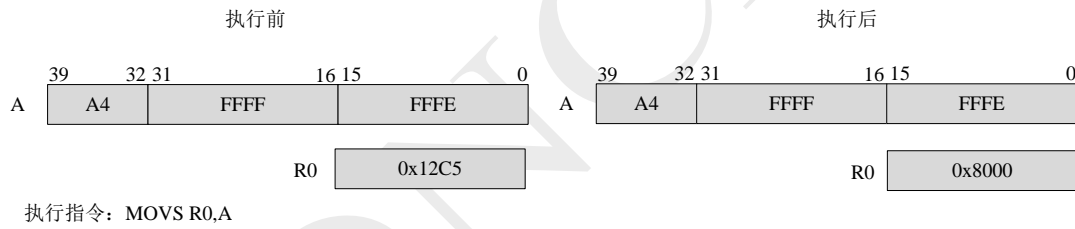


Figure 19 MOVS Rd,A 指令(例 a)

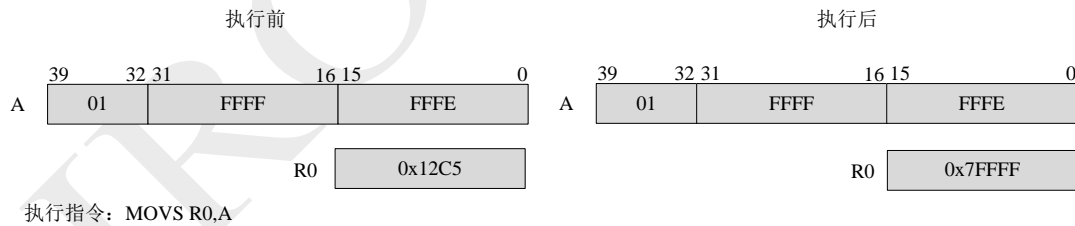


Figure 20 MOVS Rd,A 指令(例 b)

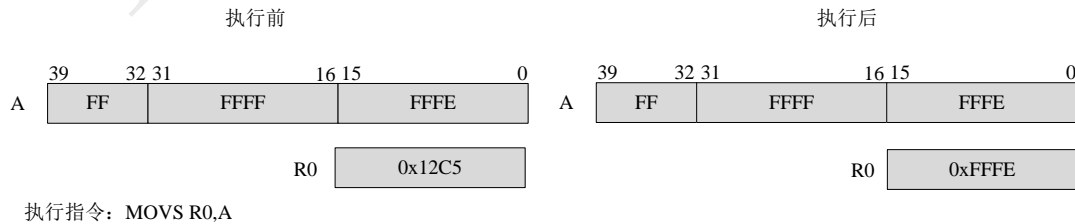


Figure 21 MOVS Rd,A 指令(例 c)

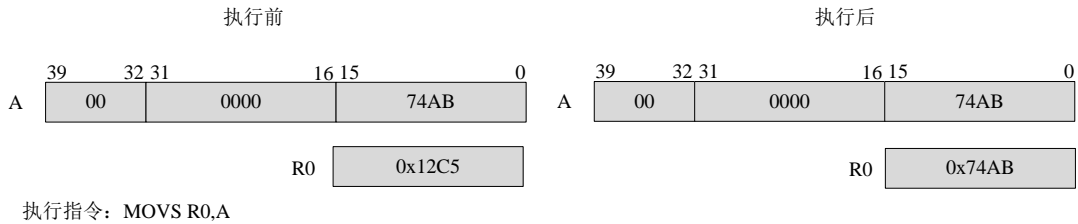


Figure 22 MOVs Rd,A 指令(例 d)

MOVR Rd,A 指令, 四舍五入赋值, 将 A 寄存器中 40 位进行四舍五入为 24 位, 然后将 24 位数据饱和限制到 16 位, 赋值给寄存器 Rd, A 寄存器中值保持不变。

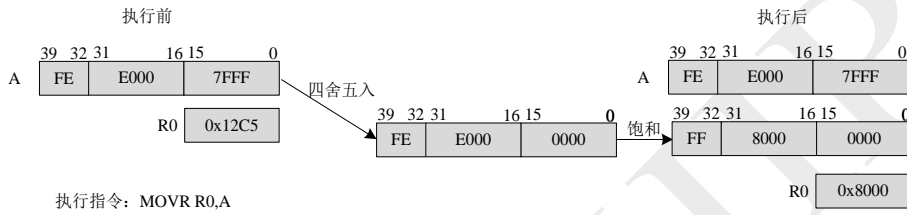


Figure 23 MOVR Rd,A 指令(例 a)

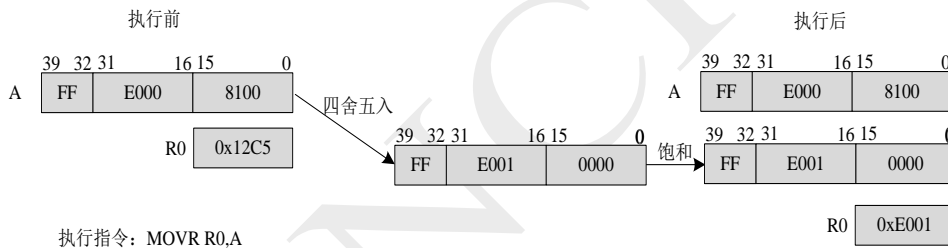


Figure 24 MOVR Rd,A 指令(例 b)

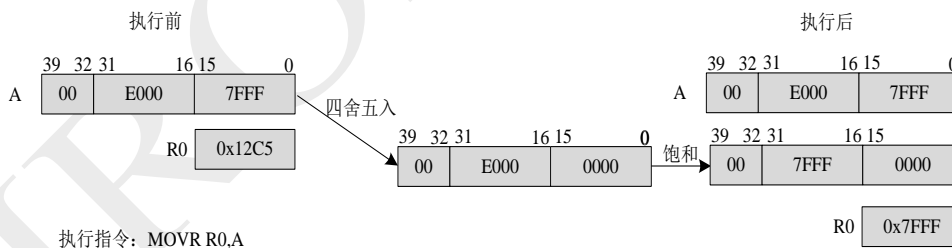


Figure 25 MOVR Rd,A 指令(例 c)

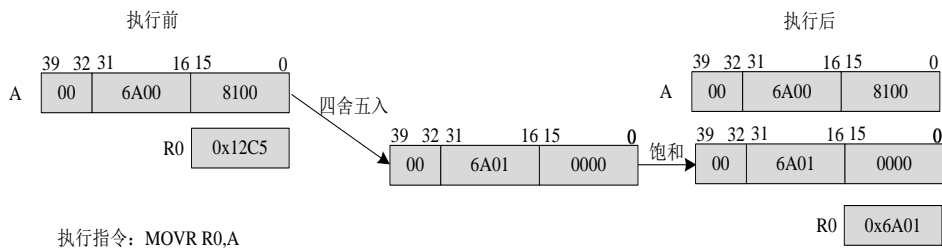


Figure 26 MOVR Rd,A 指令(例 d)

有两条指令专门针对累加器 A 进行操作, 四舍五入和饱和操作。

Round A 指令对 40 位 A 寄存器进行四舍五入操作，结果存入 A2A1(24 位)，A0 自动清 0。

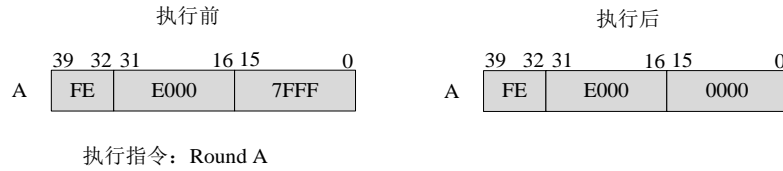


Figure 27 Round A 指令(例 a)

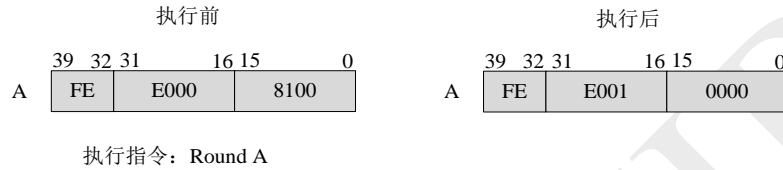


Figure 28 Round A 指令(例 b)

SAT A 指令对 40 位 A 寄存器进行饱和运算到 32 位，结果存入 A1A0，A2 为符号扩展。

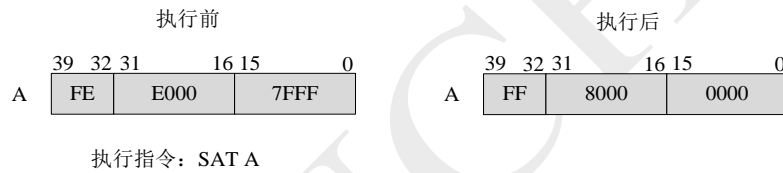


Figure 29 SAT A 指令(例 a)

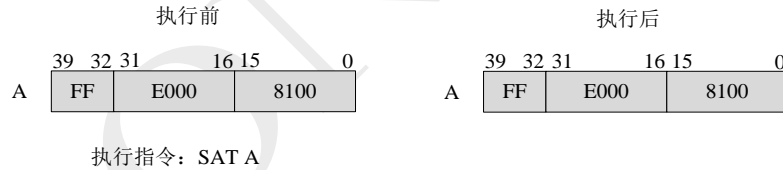


Figure 30 SAT A 指令(例 b)

累加器 A 的每个部分都可以单独访问，A1、A0 可分别作为 16 位数据赋值给通用寄存器，当读 A2 时，将 A2 符号扩展为 16 位数据赋值给通用寄存器，写 A2 时，取 16 位数据的低 8 位写入 A2。

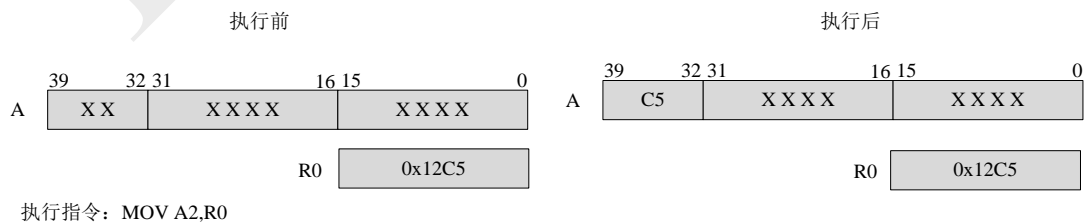


Figure 31 MOV A2,Rn 指令

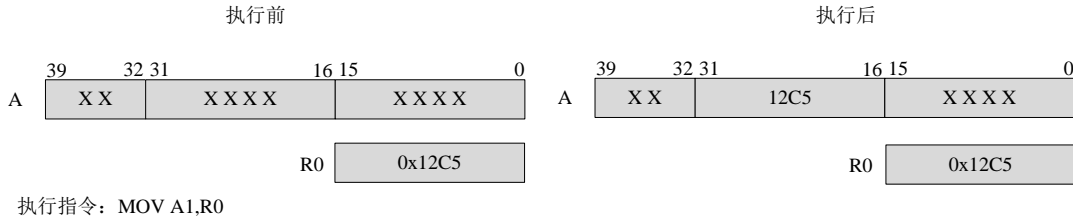


Figure 32 MOV A1,Rn 指令

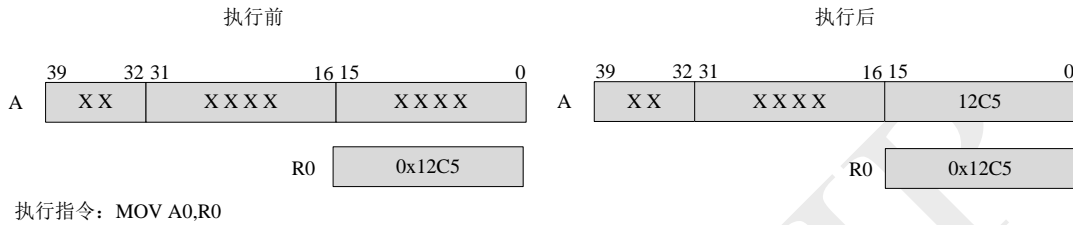


Figure 33 MOV A0,Rn 指令

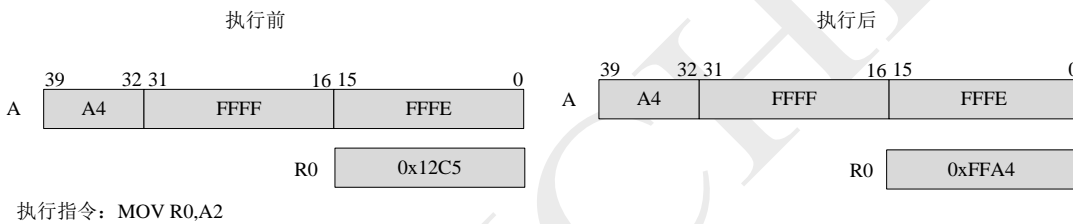


Figure 34 MOV Rd,A2 指令

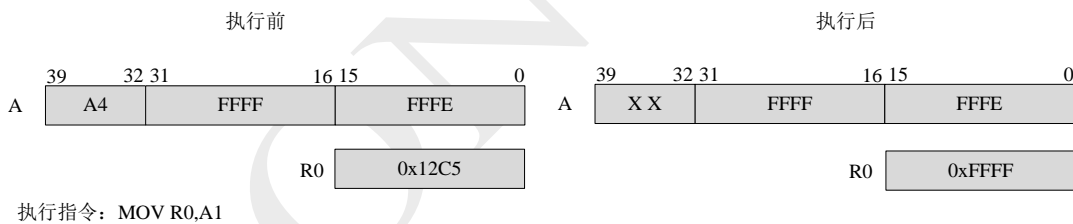


Figure 35 MOV Rd,A1 指令

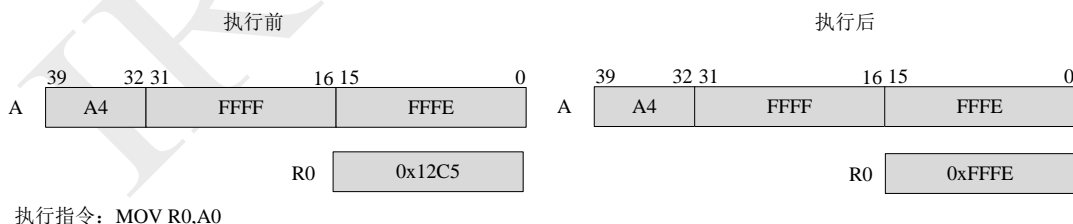


Figure 36 MOV Rd,A0 指令

乘法或乘减运算时，若运算结果超过 40 位，自动饱和到 40 位数据，同时溢出标志位 OV 置 1。

5.3.3 乘法

无符号整数乘法 (IMULU)，无符号整数 16bit x 16bit 结果存入累加器 A 的 A1A0 中，A2 自动清 0。

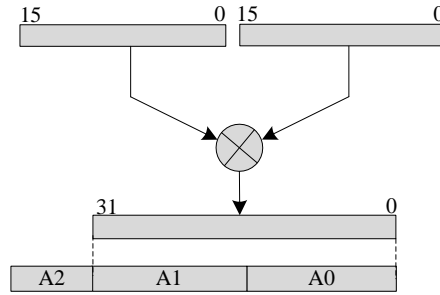


Figure 37 无符号整数相乘

有符号整数乘法 (IMUL)，有符号整数 16bit x 16bit 结果为 31 位有符号数，高位符号扩展组成 32 位数据，存入累加器 A 的 A1A0 中，A2 自动符号扩展。

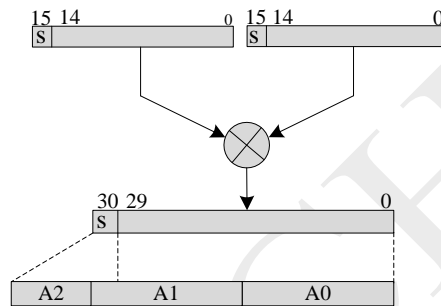


Figure 38 有符号整数相乘

有符号小数乘法 (MUL)，有符号小数 16bit x 16bit 结果为 31 位有符号数，低位补 0 组成 32 位数据，存入累加器 A 的 A1A0 中，A2 符号扩展。

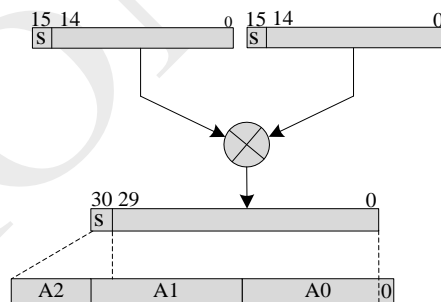


Figure 39 有符号小数相乘

有符号小数乘法带四舍五入 (MULR)，有符号小数 16bit x 16bit 结果四舍五入后存入累加器 A 的 A1 中，A2 符号扩展，A0 自动清 0。

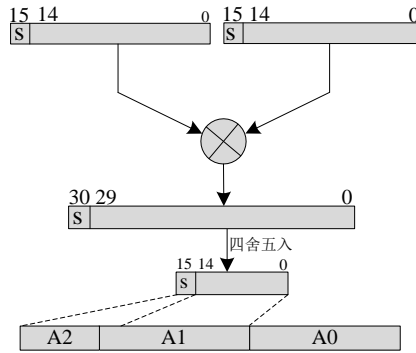


Figure 40 有符号小数相乘带四舍五入

乘法运算对标志位影响参见章节 5.2.5。

5.3.4 乘加运算

有符号整数乘加 (IMAC)，有符号整数 16bit x 16bit 结果先扩展至 40 位，再与累加器 A 相加，结果存入累加器 A 中。

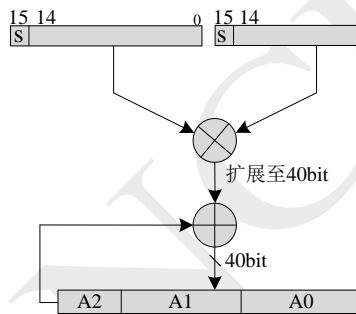


Figure 41 有符号整数乘加

有符号整数乘减 (IMACN)，有符号整数 16bit x 16bit 结果扩展至 40 位，取负后，再与累加器 A 相加，结果存入累加器 A 中。

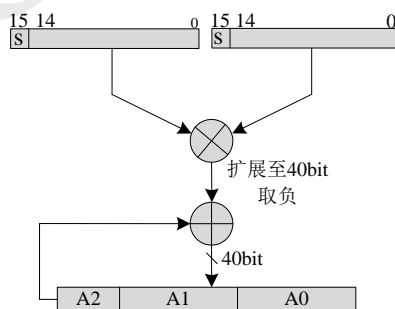


Figure 42 有符号整数乘加

有符号小数乘加 (MAC)，有符号小数 16bit x 16bit 结果扩展至 40 位小数，小数点在 31 位后面，再与累加器 A 相加，结果存入累加器 A 中。

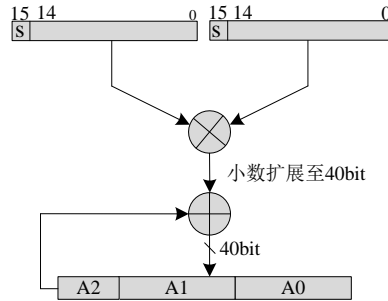


Figure 43 有符号小数乘加

有符号小数乘减 (MACN), 有符号小数 16bit x 16bit 结果扩展至 40 位小数, 小数点在 31 位后面, 取负后, 再与累加器 A 相加, 结果存入累加器 A 中。

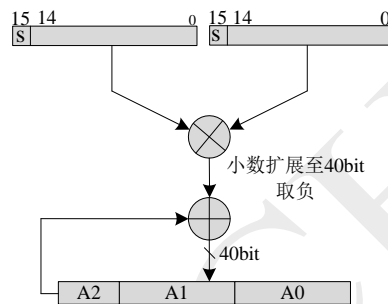


Figure 44 有符号小数乘减

有符号小数乘加带四舍五入(MACR), 有符号小数 16bit x 16bit 结果四舍五入到 16 位, 再扩展至 40 位小数, 小数点在 31 位后, 然后与累加器 A 相加, 结果四舍五入后存入累加器 A2A1 中, A0 自动清 0。

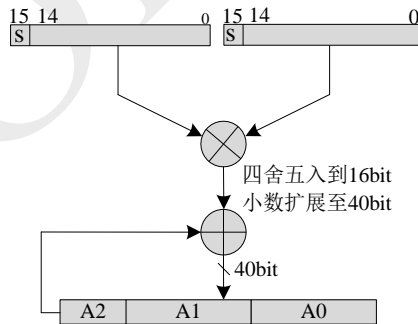


Figure 45 有符号小数乘加带四舍五入

有符号小数乘减带四舍五入 (MACNR), 有符号小数 16bit x 16bit 结果四舍五入到 16 位, 再扩展至 40 位小数, 小数点在 31 位后, 取负后, 与累加器 A 相加, 结果四舍五入后存入累加器 A2A1 中, A0 自动清 0。

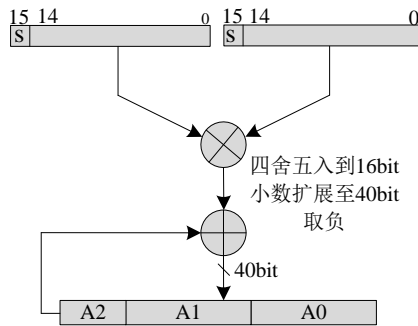


Figure 46 有符号小数乘减带四舍五入

乘加运算的结果会影响溢出标志位 OV 和零结果标志位 Z。当运算结果产生溢出时，饱和至最大或最小值，同时溢出标志位置 1；当乘加运算结果为 0 时，零结果标志位 Z 置 1。

5.3.5 加减运算

加减运算的源、目的操作数都来自通用寄存器，加法运算产生进位后，进位标志 C 置 1，发生溢出时，溢出标志位 OV 置 1，运算结果为 0 时，零结果标志位置 1。

支持有符号整数、无符号整数、有符号小数、无符号小数加减。

有符号运算时，进位或借位标志位作为符号位，判断运行结果是否溢出；无符号运算，将进位或借位标志位与结果寄存器共同判断运算结果是否溢出。

5.3.6 移位运算

移位运算的源、目的操作数都来自通用寄存器，分为算术移位和逻辑移位。算术左移将最高位移至进位标志 C，最低位补 0；算术右移最低位移入进位标志 C，最高位与原先最高位一致。

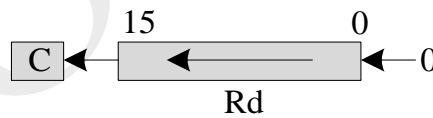


Figure 47 算术左移

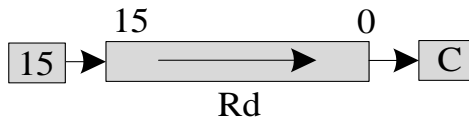


Figure 48 算术右移

逻辑左移与算术左移一致，逻辑右移最高位补 0，最低位移入进位或借位标志位 C。

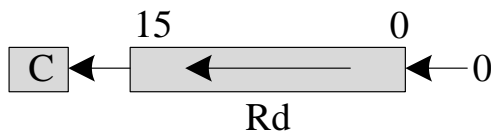


Figure 49 逻辑左移

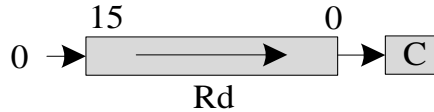


Figure 50 逻辑右移

5.3.7 四舍五入方法

四舍五入只针对运算结果写入累加器 A 及从累加器 A 读出数据时，将累加器 A 的低字节 A0 舍去，结果保留成 16 位。四舍五入支持两种方法，一种收敛四舍五入，一种 2 的补码四舍五入，通过 DSP 的“MOV PMR,Rn”指令设置。

收敛四舍五入：

- 1) $A0 < 0x8000$ 时，A0 清 0，A2A1 保持不变；
- 2) $A0 > 0x8000$ 时，A0 清 0，A2A1 加 1；
- 3) $A0 = 0x8000$ 时，且 $A1[0] = 0$ ，与情况 1 一致；
- 4) $A0 = 0x8000$ 时，且 $A1[0] = 1$ ，与情况 2 一致。

2 的补码四舍五入：

- 1) $A0 < 0x8000$ 时，A0 清 0，A2A1 保持不变；
- 2) $A0 \geq 0x8000$ 时，A0 清 0，A2A1 加 1。

5.4 DSP 存储空间

5.4.1 数据存储空间

JMT018 DSP 数据存储空间包括两部分：16 位 X 总线数据存储空间和 16 位 Y 总线数据存储空间，具体参见章节 7.2 和 7.3。。

5.4.2 程序存储空间

DSP 程序空间为 FLASH 程序存储器，DSP 的寻址地址为 $0x0000 \sim 0x3fff$ 时，对应的 MCU 程序存储空间地址为 $0x0000 \sim 0x7fff$ 。由于 MCU 起始的地址有固定用途，建议 DSP 起始地址设置为 512 或更大，即 DSP_PC 寄存器配置为 512 或更大。

16 位 DSP 程序空间与 8 位 MCU 访问空间的对应关系如 Figure 51 所示。

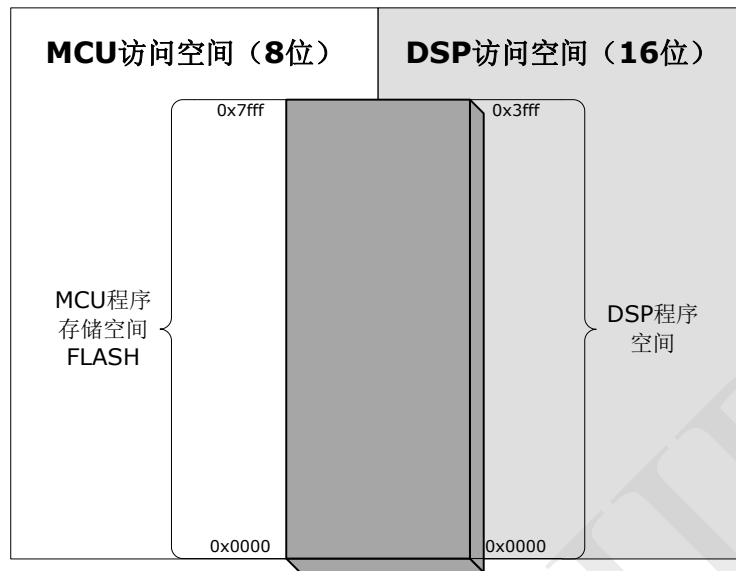


Figure 51 DSP 程序存储空间

DSP 程序在 FLASH 中运行时，最高频率为 36.864MHz，具体见 Table 43。

Table 43 JMT018 工作特性

DSP 程序空间	sys_clk 频率 (MHz)	JMT51 工作频率 (MHz)	JMT018 工作频率 (MHz)
DSP 工作于 FLASH 中	73.728	73.728	36.864
	36.864	36.864	36.864
	18.432	18.432	18.432
	9.216	9.216	9.216

5.5 DSP 工作配置

DSP 的运算能力强，当 JMT51 MCU 需要 DSP 加速性能的时候，可以启动 DSP 完成所需要的算法加速。

MCU 通过 DSP 核的 JMT51 接口，可访问 DSP 的内部寄存器，具体地址参见章节 5.8。MCU 通过访问这些寄存器，实现配置 DSP 的 PC 值、启动及读取 DSP 运行状态。

MCU 配置 DSP 启动的过程如下：

- (1) MCU 加载 DSP 程序：将 DSP 运行程序放入 FLASH 中；
- (2) MCU 配置 DSP 的 PC 值（寄存器 DSP_PCH 和 DSP_PCL），确定 DSP 运行程序的首地址；
- (3) MCU 配置寄存器 DSP_CFG.START 为 1，启动 DSP 运行。

DSP 执行到 STOP 指令后，停止运行，同时停止状态位 DSP_STA.DSP_INT 自动置 1，

并给 MCU 发送中断，MCU 根据相应程序处理中断。

DSP 每次停止后，DSP 的 PC 值保持为代码的最后一个地址加 1，如需再次启动，分为两种场景：

- 只配置寄存器 DSP_CFG.START 为 1，DSP 从上一 STOP 指令后的指令开始运行；
- 重新配置目标程序的 PC 值（寄存器 DSP_PCH 和 DSP_PCL），确定新的目标程序的首地址；配置寄存器 DSP_CFG.START 为 1，启动 DSP 运行。

5.6 DSP 代码生成与下载

5.6.1 DSP 代码生成

通过 JMT018 的汇编器将 DSP 汇编语言转换成可执行的 16 位指令代码。

5.6.2 DSP 代码下载

在 JMT51 代码中，将生成的可执行 16 位代码定义为常数数组，和 JMT51 的其他代码一起编译，生成 JMT1801ED 的下载程序。

通过芯片的 BOOT 程序将 JMT1801ED 的程序下载至 FLASH 中。

5.7 DSP 内部寄存器

DSP 内部寄存器包括通用寄存器及特殊寄存器，这些寄存器的访问方式为通过 DSP 指令访问，如 Table 44 所示。

Table 44 DSP 内部寄存器

寄存器名	属性	复位值	功能描述
R0	W/R	0x0000	DSP 通用寄存器 R0
R1	W/R	0x0000	DSP 通用寄存器 R1
R2	W/R	0x0000	DSP 通用寄存器 R2
R3	W/R	0x0000	DSP 通用寄存器 R3
R4	W/R	0x0000	DSP 通用寄存器 R4
R5	W/R	0x0000	DSP 通用寄存器 R5
R6	W/R	0x0000	DSP 通用寄存器 R6
R7	W/R	0x0000	DSP 通用寄存器 R7
DP0	W/R	0x0000	X 数据空间指针寄存器
DP1	W/R	0x0000	Y 数据空间指针寄存器
PC	W/R	0x0000	DSP 程序地址指针寄存器
A	W/R	0x0	乘法及乘加结果寄存器
LNKR	W/R	0x0000	LINK 寄存器

寄存器名	属性	复位值	功能描述
PSR	W/R	0x0000	状态寄存器
PMR	W/R	0x0004	模式寄存器
LOOP	W/R	0x0001	硬件循环次数寄存器
LIR	W/R	0x0001	硬件循环指令个数寄存器

5.7.1 DSP 通用寄存器 R0

Table 45 DSP 通用寄存器 R0

参数名	比特位	属性	复位值	描述
R0	15-0	W/R	0	DSP 通用寄存器 R0

5.7.2 DSP 通用寄存器 R1

Table 46 DSP 通用寄存器 R1

参数名	比特位	属性	复位值	描述
R1	15-0	W/R	0	DSP 通用寄存器 R1

5.7.3 DSP 通用寄存器 R2

Table 47 DSP 通用寄存器 R2

参数名	比特位	属性	复位值	描述
R2	15-0	W/R	0	DSP 通用寄存器 R2

5.7.4 DSP 通用寄存器 R3

Table 48 DSP 通用寄存器 R3

参数名	比特位	属性	复位值	描述
R3	15-0	W/R	0	DSP 通用寄存器 R3

5.7.5 DSP 通用寄存器 R4

Table 49 DSP 通用寄存器 R4

参数名	比特位	属性	复位值	描述
R4	15-0	W/R	0	DSP 通用寄存器 R4

5.7.6 DSP 通用寄存器 R5

Table 50 DSP 通用寄存器 R5

参数名	比特位	属性	复位值	描述
R5	15-0	W/R	0	DSP 通用寄存器 R5

5.7.7 DSP 通用寄存器 R6

Table 51 DSP 通用寄存器 R6

参数名	比特位	属性	复位值	描述
R6	15-0	W/R	0	DSP 通用寄存器 R6

5.7.8 DSP 通用寄存器 R7

Table 52 DSP 通用寄存器 R7

参数名	比特位	属性	复位值	描述
R7	15-0	W/R	0	DSP 通用寄存器 R7

5.7.9 X 数据空间指针寄存器 DP0

Table 53 X 数据空间指针寄存器 DP0

参数名	比特位	属性	复位值	描述
DP0	15-0	W/R	0	X 数据空间指针寄存器 DP0

5.7.10 Y 数据空间指针寄存器 DP1

Table 54 Y 数据空间指针寄存器 DP1

参数名	比特位	属性	复位值	描述
DP1	15-0	W/R	0	Y 数据空间指针寄存器 DP1

5.7.11 DSP 程序地址指针寄存器 PC

Table 55 DSP 程序地址指针寄存器 PC

参数名	比特位	属性	复位值	描述
PC	15-0	W/R	0	DSP 程序地址指针寄存器 PC

5.7.12 乘法及乘加结果寄存器 A

Table 56 乘法及乘加结果寄存器 A

参数名	比特位	属性	复位值	描述
A	39-0	W/R	0	乘法及乘加结果寄存器 A

5.7.13 LINK 寄存器

Table 57 LINK 寄存器

参数名	比特位	属性	复位值	描述
LNKR	15-0	W/R	0	调用子程序时, 硬件自动将当前 PC 值保存在 LNK R 寄存器, 然后跳转至子程序中运行; 子程序返回时, 硬件自动将 LNK R 寄

参数名	比特位	属性	复位值	描述
				寄存器中保存的值赋值给 PC。 如果存在嵌套的子程序调用，需要软件将 LNKR 寄存器进行压栈出栈操作。

5.7.14 状态寄存器 PSR

Table 58 状态寄存器 PSR

参数名	比特位	属性	复位值	描述
C	0	W/R	0	进位或借位标志位： 0: 无借位或进位产生 1: 有借位或进位产生
OV	1	W/R	0	运算结果溢出标志位： 0: 无溢出 1: 有溢出
Z	2	W/R	0	零结果标志位： 0: 运算结果不为 0 1: 运算结果等于 0
Reserved	3	R	0	保留
CF	4	W/R	0	比较结果标志位： 0: 比较条件不成立 1: 比较条件成立
Reserved	15-5	R	0	保留

5.7.15 模式寄存器 PMR

Table 59 模式寄存器 PMR

参数名	比特位	属性	复位值	描述
R	0	W/R	0	四舍五入方法： 0: 收敛四舍五入 1: 2 次补码四舍五入
S	1	W/R	0	除法运算是否有符号： 0: 无符号 1: 有符号
SAT	2	W/R	1	Round A 指令是否自动饱和： 0: Round A 指令不进行自动饱和

参数名	比特位	属性	复位值	描述
				1: Round A 指令进行自动饱和
Reserved	15-3	R	0	保留

5.7.16 硬件循环次数寄存器 LOOP

Table 60 硬件循环次数寄存器 LOOP

参数名	比特位	属性	复位值	描述
LOOP	15-0	W/R	0x0001	硬件循环次数寄存器 LOOP，取值范围为 [1,65536]

5.7.17 硬件循环指令个数寄存器 LIR

Table 61 硬件循环指令个数寄存器 LIR

参数名	比特位	属性	复位值	描述
LIR	15-0	W/R	0x0001	硬件循环指令个数寄存器 LIR，取值范围 [1,16384]

5.8 JMT51 访问 DSP 寄存器

由于 DSP 的内部寄存器映射在 MCU 的特殊功能寄存器 (SFR)，通过 MCU 可以读写 DSP 寄存器。Table 62 中所列为 JMT51 访问 DSP 寄存器的地址，对应的 SFRPAGE 为 1。

Table 62 JMT51 访问 DSP 寄存器 (SFRPAGE=0x01)

地址	寄存器名	属性	复位值	功能描述
0x8C	DSP_CFG	W	0x00	DSP 配置寄存器
0x89	DSP_STA	W/R	0x00	DSP 运行状态寄存器
0x8A	DSP_PCL	W/R	0x00	PC 寄存器低字节
0x8B	DSP_PCH	W/R	0x00	PC 寄存器高字节
0x91	DSP_R0L	W/R	0x00	R0 寄存器低字节
0x92	DSP_R0H	W/R	0x00	R0 寄存器高字节
0x93	DSP_R1L	W/R	0x00	R1 寄存器低字节
0x94	DSP_R1H	W/R	0x00	R1 寄存器高字节
0x95	DSP_R2L	W/R	0x00	R2 寄存器低字节
0x96	DSP_R2H	W/R	0x00	R2 寄存器高字节
0x97	DSP_R3L	W/R	0x00	R3 寄存器低字节
0x99	DSP_R3H	W/R	0x00	R3 寄存器高字节
0x9A	DSP_R4L	W/R	0x00	R4 寄存器低字节

地址	寄存器名	属性	复位值	功能描述
0x9B	DSP_R4H	W/R	0x00	R4 寄存器高字节
0x9C	DSP_R5L	W/R	0x00	R5 寄存器低字节
0x9D	DSP_R5H	W/R	0x00	R5 寄存器高字节
0x9E	DSP_R6L	W/R	0x00	R6 寄存器低字节
0x9F	DSP_R6H	W/R	0x00	R6 寄存器高字节
0xA1	DSP_R7L	W/R	0x00	R7 寄存器低字节
0xA2	DSP_R7H	W/R	0x00	R7 寄存器高字节
0xA3	DSP_DP0L	W/R	0x00	DP0 寄存器低字节
0xA4	DSP_DP0H	W/R	0x00	DP0 寄存器高字节
0xA5	DSP_DP1L	W/R	0x00	DP1 寄存器低字节
0xA6	DSP_DP1H	W/R	0x00	DP1 寄存器高字节
0xA7	DSP_A0L	R	0x00	A 寄存器 7~0 比特
0xA9	DSP_A0H	R	0x00	A 寄存器 15~8 比特
0xAA	DSP_A1L	R	0x00	A 寄存器 23~16 比特
0xAB	DSP_A1H	R	0x00	A 寄存器 31~24 比特
0xAC	DSP_A2	R	0x00	A 寄存器 39~32 比特
0xAD	DSP_LNKR L	R	0x00	LNKR 寄存器低字节
0xAE	DSP_LNKR H	R	0x00	LNKR 寄存器高字节
0xAF	DSP_PSR L	R	0x00	PSR 寄存器低字节
0xB1	DSP_PSR H	R	0x00	PSR 寄存器高字节
0xB2	DSP_PMR L	R	0x04	PMR 寄存器低字节
0xB3	DSP_PMR H	R	0x00	PMR 寄存器高字节
0xB4	DSP_LOOP L	R	0x01	LOOP 寄存器低字节
0xB5	DSP_LOOP H	R	0x00	LOOP 寄存器高字节
0xB6	DSP_LIR L	R	0x01	LIR 寄存器低字节
0xB7	DSP_LIR H	R	0x00	LIR 寄存器高字节

5.8.1 DSP 配置寄存器(DSP_CFG) [0x8C]

Table 63 DSP 配置寄存器(DSP_CFG)

参数名	比特位	属性	复位值	描述
START	0	W	0	DSP 启动控制位： 0: 无效

参数名	比特位	属性	复位值	描述
				1: 启动 DSP 运行
Reserved	7-1	R	0	保留

5.8.2 DSP 运行状态寄存器(DSP_STA) [0x89]

Table 64 DSP 运行状态寄存器(DSP_STA)

参数名	比特位	属性	复位值	描述
STARUN	0	R	0	DSP 运行状态位: 0: DSP 停止状态 1: DSP 运行状态
DSP_INT	1	W/R	0	DSP 中断标志位: <ul style="list-style-type: none"> 当有 DSP 运行结束中断申请时, 该位置为 1 软件配置该位为 1 可产生软中断, 在中断处理程序中需要软件写 0 来清除
Reserved	7-2	R	0	保留

5.8.3 PC 寄存器低字节(DSP_PCL) [0x8A]

Table 65 PC 寄存器低字节(DSP_PCL)

参数名	比特位	属性	复位值	描述
DSP_PCL	7-0	W/R	0	PC 寄存器低字节

5.8.4 PC 寄存器高字节(DSP_PCH) [0x8B]

Table 66 PC 寄存器高字节(DSP_PCH)

参数名	比特位	属性	复位值	描述
DSP_PCH	7-0	W/R	0	PC 寄存器高字节

5.8.5 R0 寄存器低字节(DSP_R0L) [0x91]

Table 67 R0 寄存器低字节(DSP_R0L)

参数名	比特位	属性	复位值	描述
DSP_R0L	7-0	W/R	0	R0 寄存器低字节

5.8.6 R0 寄存器高字节(DSP_R0H) [0x92]

Table 68 R0 寄存器高字节(DSP_R0H)

参数名	比特位	属性	复位值	描述
DSP_R0H	7-0	W/R	0	R0 寄存器高字节

5.8.7 R1 寄存器低字节(DSP_R1L) [0x93]

Table 69 R1 寄存器低字节(DSP_R1L)

参数名	比特位	属性	复位值	描述
DSP_R1L	7-0	W/R	0	R1 寄存器低字节

5.8.8 R1 寄存器高字节(DSP_R1H) [0x94]

Table 70 R1 寄存器高字节(DSP_R1H)

参数名	比特位	属性	复位值	描述
DSP_R1H	7-0	W/R	0	R1 寄存器高字节

5.8.9 R2 寄存器低字节(DSP_R2L) [0x95]

Table 71 R2 寄存器低字节(DSP_R2L)

参数名	比特位	属性	复位值	描述
DSP_R2L	7-0	W/R	0	R2 寄存器低字节

5.8.10 R2 寄存器高字节(DSP_R2H) [0x96]

Table 72 R2 寄存器高字节(DSP_R2H)

参数名	比特位	属性	复位值	描述
DSP_R2H	7-0	W/R	0	R2 寄存器高字节

5.8.11 R3 寄存器低字节(DSP_R3L) [0x97]

Table 73 R3 寄存器低字节(DSP_R3L)

参数名	比特位	属性	复位值	描述
DSP_R3L	7-0	W/R	0	R3 寄存器低字节

5.8.12 R3 寄存器高字节(DSP_R3H) [0x99]

Table 74 R3 寄存器高字节(DSP_R3H)

参数名	比特位	属性	复位值	描述
DSP_R3H	7-0	W/R	0	R3 寄存器高字节

5.8.13 R4 寄存器低字节(DSP_R4L) [0x9A]

Table 75 R4 寄存器低字节(DSP_R4L)

参数名	比特位	属性	复位值	描述
DSP_R4L	7-0	W/R	0	R4 寄存器低字节

5.8.14 R4 寄存器高字节(DSP_R4H) [0x9B]

Table 76 R4 寄存器高字节(DSP_R4H)

参数名	比特位	属性	复位值	描述
DSP_R4H	7-0	W/R	0	R4 寄存器高字节

5.8.15 R5 寄存器低字节(DSP_R5L) [0x9C]

Table 77 R5 寄存器低字节(DSP_R5L)

参数名	比特位	属性	复位值	描述
DSP_R5L	7-0	W/R	0	R5 寄存器低字节

5.8.16 R5 寄存器高字节(DSP_R5H) [0x9D]

Table 78 R5 寄存器高字节(DSP_R5H)

参数名	比特位	属性	复位值	描述
DSP_R5H	7-0	W/R	0	R5 寄存器高字节

5.8.17 R6 寄存器低字节(DSP_R6L) [0x9E]

Table 79 R6 寄存器低字节(DSP_R6L)

参数名	比特位	属性	复位值	描述
DSP_R6L	7-0	W/R	0	R6 寄存器低字节

5.8.18 R6 寄存器高字节(DSP_R6H) [0x9F]

Table 80 R6 寄存器高字节(DSP_R6H)

参数名	比特位	属性	复位值	描述
DSP_R6H	7-0	W/R	0	R6 寄存器高字节

5.8.19 R7 寄存器低字节(DSP_R7L) [0xA1]

Table 81 R7 寄存器低字节(DSP_R7L)

参数名	比特位	属性	复位值	描述
DSP_R7L	7-0	W/R	0	R7 寄存器低字节

5.8.20 R7 寄存器高字节(DSP_R7H) [0xA2]

Table 82 R7 寄存器高字节(DSP_R7H)

参数名	比特位	属性	复位值	描述
DSP_R7H	7-0	W/R	0	R7 寄存器高字节

5.8.21 DP0 寄存器低字节(DSP_DP0L) [0xA3]

Table 83 DP0 寄存器低字节(DSP_DP0L)

参数名	比特位	属性	复位值	描述
DSP_DP0L	7-0	W/R	0	DP0 寄存器低字节

5.8.22 DP0 寄存器高字节(DSP_DP0H) [0xA4]

Table 84 DP0 寄存器高字节(DSP_DP0H)

参数名	比特位	属性	复位值	描述
DSP_DP0H	7-0	W/R	0	DP0 寄存器高字节

5.8.23 DP1 寄存器低字节(DSP_DP1L) [0xA5]

Table 85 DP1 寄存器低字节(DSP_DP1L)

参数名	比特位	属性	复位值	描述
DSP_DP1L	7-0	W/R	0	DP1 寄存器低字节

5.8.24 DP1 寄存器高字节(DSP_DP1H) [0xA6]

Table 86 DP1 寄存器高字节(DSP_DP1H)

参数名	比特位	属性	复位值	描述
DSP_DP1H	7-0	W/R	0	DP1 寄存器高字节

5.8.25 A 寄存器 7~0 比特(DSP_A0L) [0xA7]

Table 87 A 寄存器 7~0 比特(DSP_A0L)

参数名	比特位	属性	复位值	描述
DSP_A0L	7-0	R	0	A 寄存器 7~0 比特

5.8.26 A 寄存器 15~8 比特(DSP_A0H) [0xA9]

Table 88 A 寄存器 15~8 比特(DSP_A0H)

参数名	比特位	属性	复位值	描述
DSP_A0H	7-0	R	0	A 寄存器 15~8 比特

5.8.27 A 寄存器 23~16 比特(DSP_A1L) [0xAA]

Table 89 A 寄存器 23~16 比特(DSP_A1L)

参数名	比特位	属性	复位值	描述
DSP_A1L	7-0	R	0	A 寄存器 23~16 比特

5.8.28 A 寄存器 31~24 比特(DSP_A1H) [0xAB]

Table 90 A 寄存器 31~24 比特(DSP_A1H)

参数名	比特位	属性	复位值	描述
DSP_A1H	7-0	R	0	A 寄存器 31~24 比特

5.8.29 A 寄存器 39~32 比特(DSP_A2) [0xAC]

Table 91 A 寄存器 39~32 比特(DSP_A2)

参数名	比特位	属性	复位值	描述
DSP_A2	7-0	R	0	A 寄存器 39~32 比特

5.8.30 LNKR 寄存器低字节(DSP_LNKRL) [0xAD]

Table 92 LNKR 寄存器低字节(DSP_LNKRL)

参数名	比特位	属性	复位值	描述
DSP_LNKRL	7-0	R	0	LNKR 寄存器低字节

5.8.31 LNKR 寄存器高字节(DSP_LNKRH) [0xAE]

Table 93 LNKR 寄存器高字节(DSP_LNKRH)

参数名	比特位	属性	复位值	描述
DSP_LNKRH	7-0	R	0	LNKR 寄存器高字节

5.8.32 PSR 寄存器低字节(DSP_PSRL) [0xAF]

Table 94 PSR 寄存器低字节(DSP_PSRL)

参数名	比特位	属性	复位值	描述
C	0	R	0	进位或借位标志位： 0: 无借位或进位产生 1: 有借位或进位产生
OV	1	R	0	运算结果溢出标志位： 0: 无溢出 1: 有溢出
Z	2	R	0	零结果标志位： 0: 运算结果不为 0

参数名	比特位	属性	复位值	描述
				1: 运算结果等于 0
Reserved	3	R	0	保留
CF	4	R	0	比较结果标志位: 0: 比较条件不成立 1: 比较条件成立
Reserved	7-5	R	0	保留

5.8.33 PSR 寄存器高字节(DSP_PSRH) [0xB1]

Table 95 PSR 寄存器高字节(DSP_PSRH)

参数名	比特位	属性	复位值	描述
DSP_PSRH	7-0	R	0	PSR 寄存器高字节

5.8.34 PMR 寄存器低字节(DSP_PMRL) [0xB2]

Table 96 PMR 寄存器低字节(DSP_PMRL)

参数名	比特位	属性	复位值	描述
R	0	R	0	四舍五入方法: 0: 收敛四舍五入 1: 2 次补码四舍五入
S	1	R	0	除法运算是否有符号: 0: 无符号 1: 有符号
SAT	2	R	1	Round A 指令是否自动饱和: 0: Round A 指令不进行自动饱和 1: Round A 指令进行自动饱和
Reserved	7-3	R	0	保留

5.8.35 PMR 寄存器高字节(DSP_PMRH) [0xB3]

Table 97 PMR 寄存器高字节(DSP_PMRH)

参数名	比特位	属性	复位值	描述
DSP_PMRH	7-0	R	0	PMR 寄存器高字节

5.8.36 LOOP 寄存器低字节(DSP_LOOPL) [0xB4]

Table 98 LOOP 寄存器低字节(DSP_LOOPL)

参数名	比特位	属性	复位值	描述
DSP_LOOPL	7-0	R	0x01	LOOP 寄存器低字节

5.8.37 LOOP 寄存器高字节(DSP_LOOPH) [0xB5]

Table 99 LOOP 寄存器高字节(DSP_LOOPH)

参数名	比特位	属性	复位值	描述
DSP_LOOPH	7-0	R	0	LOOP 寄存器高字节

5.8.38 LIR 寄存器低字节(DSP_LIRL) [0xB6]

Table 100 LIR 寄存器低字节(DSP_LIRL)

参数名	比特位	属性	复位值	描述
DSP_LIRL	7-0	R	0x01	LIR 寄存器低字节

5.8.39 LIR 寄存器高字节(DSP_LIRH) [0xB7]

Table 101 LIR 寄存器高字节(DSP_LIRH)

参数名	比特位	属性	复位值	描述
DSP_LIRH	7-0	R	0	LIR 寄存器高字节

6 中断

6.1 概述

JMT51 包含一个扩展的中断系统，支持 13 个中断源，每个中断源有 4 个优先级。每个中断源在特殊功能寄存器（SFR）中有相应的中断标志。当一个中断源满足有效的中断条件时，相应的中断标志被置 1。

中断使能采用多级控制：

- 第一级控制为总控制，通过配置 IEN0.7 中断使能位 EA，对所有中断进行开启或屏蔽。
- 第二级控制为 13 个中断源分别控制，通过配置寄存器的 IEN0、IEN1 及 IEN2 相应比特位开启或屏蔽相应中断。
- 第三级控制为连接到第二级 13 个中断源的外设中断的控制，具体寄存器参见各外设模块寄存器说明。

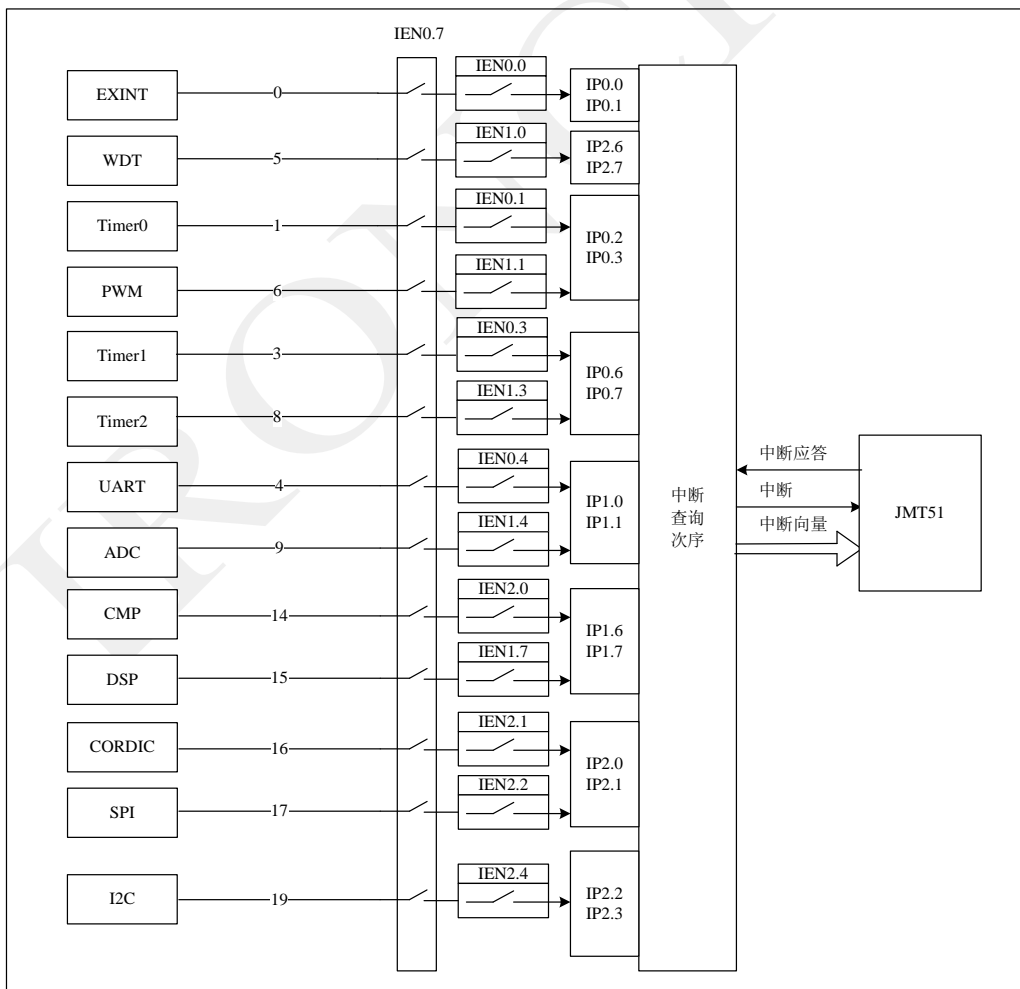


Figure 52 中断各级使能框图

Timer 中断标志位在 JMT51 进入中断服务程序后自动清除，其他中断标志硬件不会自动清除，必须在中断返回之前软件配置寄存器清除，具体见 Table 102 中“断标志位清除说明”。

如果一个中断标志位在 JMT51 执行完中断返回 (RETI) 指令后仍然保持有效状态，则会立即产生一个新的中断请求，JMT51 执行完下一条指令后再次进入该中断。

中断分布如 Table 102 所示。

Table 102 中断分布情况

中断号	中断向量	中断源	默认中断优先级	中断标志位清除说明
0	0x03	EXINT 中断	最高 0	软件清除
1	0x0B	Timer 0 中断	2	硬件自动清除
3	0x1B	Timer 1 中断	4	硬件自动清除
4	0x23	UART 中断	6	软件清除
5	0x2B	WDT 中断	1	软件清除
6	0x33	PWM 中断	3	软件清除
8	0x43	Timer2 中断	5	硬件自动清除
9	0x4B	ADC 中断	7	软件清除
14	0x73	CMP 中断	8	软件清除
15	0x7B	DSP 中断	9	软件清除
16	0x83	CORDIC 中断	10	软件清除
17	0x8B	SPI 中断	11	软件清除
19	0x9B	I2C 中断	12	软件清除

6.2 外部管脚中断 (EXINT)

JMT51 共有 1 个外部管脚中断：EXINT，由多个 GPIO 中断合并而成，每个 GPIO 中断又可独立配置为上升沿触发、下降沿触发或双沿触发。有关外部管脚中断 EXINT 的详细说明参见 18.7 章节描述。

6.3 中断优先级

中断优先级描述如下：

- 每个中断可以通过配置中断优先级寄存器 IP0、IP1、IP2 选择中断优先级。
- 同优先级的中断源同时申请中断，硬件会根据 Table 102 中各中断源的默认中断优先级顺序进行响应。
- 不同优先级的中断源同时申请中断时，JMT51 将响应较高优先级的中断申请。
- JMT51 在处理中断服务程序时，可被更高优先级的中断打断，但不能被同级或低优先级的中断打断。

- JMT51 在处理最高级中断服务程序时，不可被其他任何中断打断。

Table 103 中断优先级分组

组	中断源		描述
0	EXINT 中断		通过 IP0.0、IP0.1 设置优先级
1	Timer 0 中断	PWM 中断	通过 IP0.2、IP0.3 设置优先级
2	保留		通过 IP0.4、IP0.5 设置优先级
3	Timer 1 中断	Timer2 中断	通过 IP0.6、IP0.7 设置优先级
4	UART 中断	ADC 中断	通过 IP1.0、IP1.1 设置优先级
5	保留		通过 IP1.2、IP1.3 设置优先级
6	保留		通过 IP1.4、IP1.5 设置优先级
7	CMP 中断	DSP 中断	通过 IP1.6、IP1.7 设置优先级
8	SPI 中断	CORDIC 中断	通过 IP2.0、IP2.1 设置优先级
9	I2C 中断		通过 IP2.2、IP2.3 设置优先级
10	保留		通过 IP2.4、IP2.5 设置优先级
11	WDT 中断		通过 IP2.6、IP2.7 设置优先级

6.4 中断处理过程

每个系统时钟周期 JMT51 都将去检测是否存在中断申请, JMT51 捕获到中断标志位后, 将利用长转移指令 (LCALL) 调用该中断的中断服务程序, 下列情况下中断将会延迟处理:

- 同级或更高级的优先级中断在运行中。
- JMT51 正在执行的指令是 RETI 指令。在 RETI 之后之后, 至少再执行一条其他指令之后才会响应中断。
- JMT51 正在访问中断相关寄存器的指令, 在读写中断相关寄存器之后, 至少再执行一条其他指令之后才会响应中断。

若检测到某一中断, LCALL 指令会把程序计数器 (PC 值) 中的内容进行压栈, 将相应中断源的向量地址存入程序计数器 (PC 值), 中断服务程序从该中断向量指定地址开始运行, 直到 RETI 指令, RETI 指令通知 JMT51 中断服务程序结束, 将堆栈数据重载入程序计数器 (PC 值)。

6.5 中断响应时间

中断响应时间取决于中断发生时 JMT51 状态。

中断系统在每个 JMT51 时钟周期采样中断标志位, 最快响应时间为 5 个 JMT51 时钟周期: 1 个周期用于检测中断, 4 个周期完成对中断服务程序长调用 LCALL。如果中断标志有效时 JMT51 正在执行 RETI 指令, 则需要再执行 1 条指令才能进入中断服务程序。如果 RETI

后面跟 1 个多周期指令，则响应时间会更长。

6.6 中断寄存器

中断模块共有 6 个寄存器，对应的 SFRPAGE 为所有 PAGE（公共寄存器），也即在所有 PAGE 配置中断寄存器均生效。具体列表如 Table 104 所示。

Table 104 中断模块寄存器（所有 PAGE）

地址	寄存器名	类型	复位值	功能描述
0xA8	IEN0	W/R	0x00	中断使能寄存器 0
0xB8	IEN1	W/R	0x00	中断使能寄存器 1
0xC0	IEN2	W/R	0x00	中断使能寄存器 2
0xB9	IP0	W/R	0x00	中断优先级寄存器 0
0xBA	IP1	W/R	0x00	中断优先级寄存器 1
0xBB	IP2	W/R	0xC0	中断优先级寄存器 2

6.6.1 中断使能寄存器 0(IEN0) [0xA8]

Table 105 中断使能寄存器 0(IEN0)

参数名	比特位	属性	复位值	描述
EXINT	0	W/R	0	EXINT 中断允许位： 0：禁止 EXINT 中断 1：允许 EXINT 中断
ET0	1	W/R	0	Timer0 中断允许位： 0：禁止 Timer0 中断 1：允许 Timer0 中断
Reserved	2	R	0	保留
ET1	3	W/R	0	Timer1 中断允许位： 0：禁止 Timer1 中断 1：允许 Timer1 中断
ES	4	W/R	0	UART 发送/接收中断允许位： 0：禁止 UART 发送/接收中断 1：允许 UART 发送/接收中断
Reserved	6-5	R	0	保留
EA	7	W/R	0	总中断允许位： 0：禁止所有中断 1：允许所有中断

6.6.2 中断使能寄存器 1(IEN1) [0xB8]

Table 106 中断使能寄存器 1(IEN1)

参数名	比特位	属性	复位值	描述
WDTINT	0	W/R	0	WDT 中断允许位： 0：禁止 WDT 中断 1：允许 WDT 中断
PWMINT	1	W/R	0	PWM 中断允许位： 0：禁止 PWM 中断 1：允许 PWM 中断
Reserved	2	R	0	保留
ET2	3	W/R	0	Timer2 中断允许位： 0：禁止 Timer2 中断 1：允许 Timer2 中断
ADCINT	4	W/R	0	ADC 中断允许位： 0：禁止 ADC 中断 1：允许 ADC 中断
Reserved	6-5	R	0	保留
DSPINT	7	W/R	0	DSP 中断允许位： 0：禁止 DSP 中断 1：允许 DSP 中断

6.6.3 中断使能寄存器 2 (IEN2) [0xC0]

Table 107 中断使能寄存器 2(IEN2)

参数名	比特位	属性	复位值	描述
CMPINT	0	W/R	0	CMP 中断允许位： 0：禁止 CMP 中断 1：允许 CMP 中断
CORDICINT	1	W/R	0	CORDIC 中断允许位： 0：禁止 CORDIC 中断 1：允许 CORDIC 中断
SPIINT	2	W/R	0	SPI 中断允许位： 0：禁止 SPI 中断 1：允许 SPI 中断

参数名	比特位	属性	复位值	描述
Reserved	3	R	0	保留
I2CINT	4	W/R	0	I2C 中断允许位： 0: 禁止 I2C 中断 1: 允许 I2C 中断
Reserved	7-5	R	0	保留

6.6.4 中断优先级寄存器 0 (IP0) [0xB9]

Table 108 中断优先级寄存器 0(IP0)

参数名	比特位	属性	复位值	描述
IP0.1~IP0.0	1-0	W/R	0	EXINT 中断优先级控制： 00: 第 0 级（最低） 01: 第 1 级 10: 第 2 级 11: 第 3 级（最高）
IP0.3~IP0.2	3-2	W/R	0	Timer0 中断和 PWM 中断优先级控制： 00: 第 0 级（最低） 01: 第 1 级 10: 第 2 级 11: 第 3 级（最高）
Reserved	5-4	W/R	0	保留
IP0.7~IP0.6	7-6	W/R	0	Timer1 中断和 Timer2 中断优先级控制： 00: 第 0 级（最低） 01: 第 1 级 10: 第 2 级 11: 第 3 级（最高）

6.6.5 中断优先级寄存器 1(IP1) [0xBA]

Table 109 中断优先级寄存器 1(IP1)

参数名	比特位	属性	复位值	描述
IP1.1~IP1.0	1-0	W/R	0	UART 中断和 ADC 中断优先级控制： 00: 第 0 级（最低） 01: 第 1 级 10: 第 2 级

参数名	比特位	属性	复位值	描述
				11: 第 3 级 (最高)
Reserved	5-2	W/R	0	保留
IP1.7~IP1.6	7-6	W/R	0	CMP 中断和 DSP 中断优先级控制: 00: 第 0 级 (最低) 01: 第 1 级 10: 第 2 级 11: 第 3 级 (最高)

6.6.6 中断优先级寄存器 2(IP2) [0xBB]

Table 110 中断优先级寄存器 2(IP2)

参数名	比特位	属性	复位值	描述
IP2.1~IP2.0	1-0	W/R	0	CORDIC 中断和 SPI 中断优先级控制: 00: 第 0 级 (最低) 01: 第 1 级 10: 第 2 级 11: 第 3 级 (最高)
IP2.3~IP2.2	3-2	W/R	0	I2C 中断优先级控制: 00: 第 0 级 (最低) 01: 第 1 级 10: 第 2 级 11: 第 3 级 (最高)
Reserved	5-4	W/R	0	保留
IP2.7~IP2.6	7-6	W/R	11	WDT 中断优先级控制: 00: 第 0 级 (最低) 01: 第 1 级 10: 第 2 级 11: 第 3 级 (最高)

7 存储器

JMT1801ED 具有两个内核: JMT51 MCU、JMT018 DSP。对存储器的访问按照访问方式也分为两大类:

- JMT51 访问的存储器;

- JMT018 DSP 访问的存储器。

7.1 JMT51 存储器

JMT51 和标准的 8051 类似，存储器由三部分组成：程序存储器、外部数据存储器以及内部数据存储器。其中，程序存储器和外部数据存储器共享相同的地址空间，通过使用不同的指令类型进行访问。

JMT51 的存储器组织如 Figure 53 所示。

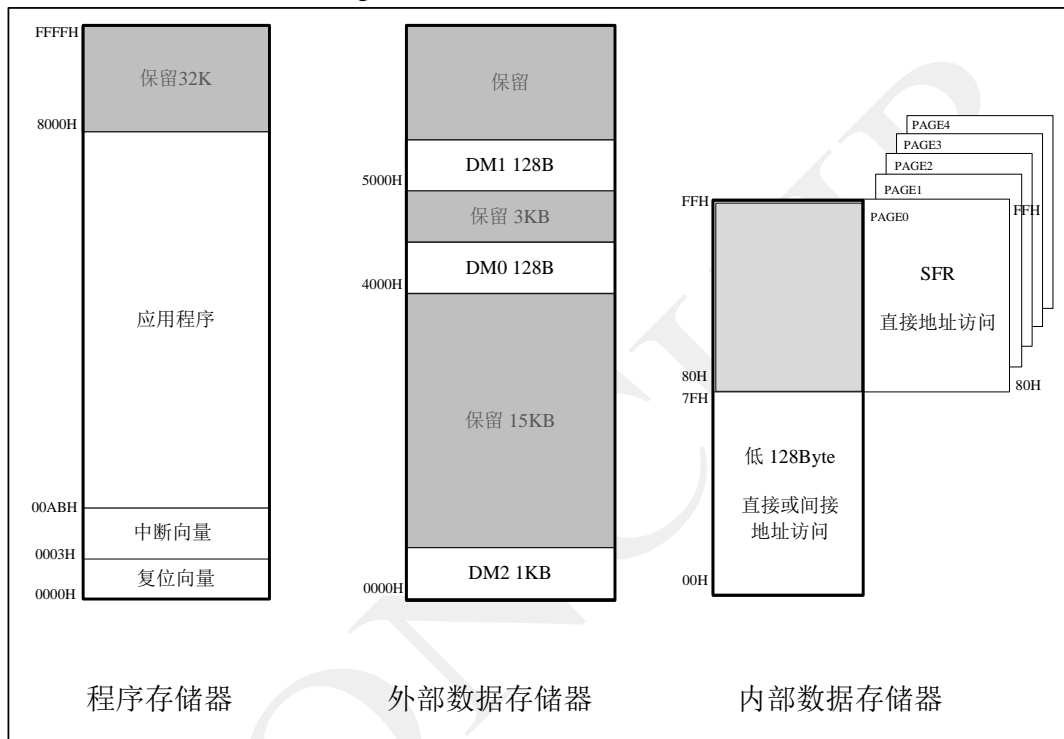


Figure 53 JMT51 存储器分布图

7.1.1 程序存储器

JMT51 有 32K 字节的程序存储空间，该程序存储器为可编程的 FLASH 存储器，地址范围为 0x0000~0x7fff，如 Figure 53 中程序存储器所示。

JMT51 内部含有 BOOT ROM，用于用户应用程序的烧录，该 BOOT ROM 对用户不可见。

FLASH 存储器一般用于存储 JMT51 运行的应用程序，该应用程序为通过 BOOT ROM 下载至 FLASH 中的。上电之后，JMT51 自动从 FLASH 的 0x0000 地址开始运行所下载的应用程序。

FLASH 存储器也可用于在线存储用户的非易失性数据，具体参见章节 0 FLASH 存储器描述。

7.1.2 外部数据存储器

JMT51 外部数据存储器由三部分组成，如 Figure 53 中外部数据存储器所示：

- DM2: DM2 外部存储器大小为 1K 字节，地址空间为：0x0000~0x03ff；
- DM0: DM0 外部存储器大小为 128 字节，地址空间为：0x4000~0x407f；
- DM1: DM1 外部存储器大小为 128 字节，地址空间为：0x5000~0x507f。

配置寄存器 MOVXCON.PMW 位为 0，JMT51 即可使用 MOVX 指令对外部数据存储器进行读写操作，该存储器的读写有两种寻址方式：DPTR 地址、间接地址。

7.1.3 内部数据存储器

JMT51 内部数据存储器大小为 128 字节，地址空间为 0x00~0x7f，如 Figure 54 所示。

128 字节内部数据存储器可用于通用寄存器和临时存储器，可通过直接或间接寻址方式进行访问。

地址从 0x00 到 0x1f 为 4 个通用寄存器区，每个区有 8 个字节的寄存器。

从地址 0x20 到 0x2f 的 16 个字节，既可以字节寻址，又可作为 128 个位地址利用直接位寻址方式进行访问。

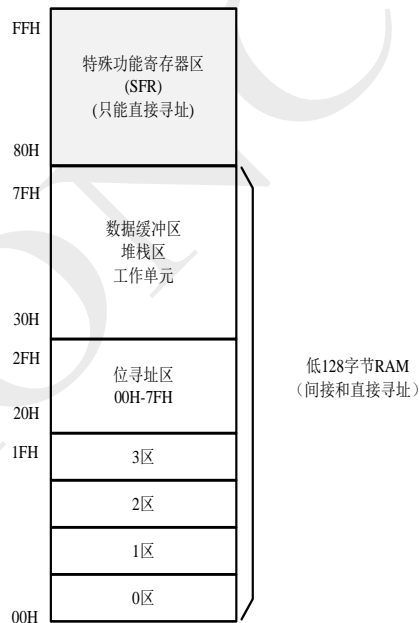


Figure 54 内部数据存储器分区

7.1.3.1 通用寄存器区

通用寄存器区(0x00~0x1f)，共 32 个单元分为四组，每组包含 8 个 8 位寄存器，均以 R0~R7 命名。通过状态寄存器 PSW 的 RS0 和 RS1 位决定使用哪一组寄存器，某一时刻只能选择其中的一组寄存器，其余的可以作为正常的数据存储器使用。间接寻址方式使用 R0 和 R1 作为间接寄存器。JMT51 复位后，默认选中第 0 组寄存器。

Table 111 通用寄存器地址

组	RS1	RS0	R0	R1	R2	R3	R4	R5	R6	R7
0	0	0	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
1	0	1	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
2	1	0	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
3	1	1	0x20	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F

7.1.3.2 位寻址区

位寻址区(0x20~0x2F)，内部数据存储器的 0x20~0x2F 单元为位寻址区，即可作为一般单元用字节寻址，也可对它们的位进行寻址。位寻址共有 16 个字节，128 位，位地址为 0x00~0x7F。位地址分配如 Table 112 所示。

特殊功能寄存器（SFR）中，直接地址可被 8 整除的寄存器，也可以进行位寻址。

JMT51 能直接访问位地址对应的 BIT，执行如置 1、清 0、求反、转移、传送、逻辑等操作。

Table 112 位寻址地址

单元地址	MSB 位地址 LSB							
0x2F	0x7F	0x7E	0x7D	0x7C	0x7B	0x7A	0x79	0x78
0x2E	0x77	0x76	0x75	0x74	0x73	0x72	0x71	0x70
⋮	---							
0x2B	0x0F	0x0E	0x0D	0x0C	0x0B	0x0A	0x09	0x08
0x20	0x07	0x06	0x05	0x04	0x03	0x02	0x01	0x00

7.1.3.3 堆栈

程序的堆栈可以位于 128 字节数据存储器中的任何位置，JMT51 堆栈深度参见 4.1 描述。

堆栈区域用堆栈指针 SP 指定，SP 指向最后使用的位置。下一个压入堆栈的数据将被存放在 SP+1 位置，然后 SP 加 1。复位以后堆栈指针被初始化为 0x07，因此第一个被压入堆栈的数据将被存放在地址 0x08，0x08 地址的内容是通用寄存器组 1 的第一个寄存器 R0。如果使用不止一个通用寄存器组，SP 值应被初始化为 128 字节内部数据存储器中不用于数据存储的位置。

一般用户要设置 SP 初值，堆栈是向上生成的。

7.1.3.4 一般存储区

片内 RAM 128 字节中，通用寄存器占有 32 个字节，位寻址区占有 16 个字节，剩下的 80 个字节为可供用户使用的一般寄存器区，对这部分区域的使用不作任何规定和限制，堆栈一般开辟在这个区域。

7.1.4 特殊功能寄存器

从 0x80 到 0xFF 的直接寻址存储空间为特殊功能寄存器 (SFR)，通过配置这些寄存器，可以设置外设及 JMT51 的功能及特性。

任何时刻用直接寻址访问从 0x80 到 0xff 的存储器空间将访问特殊功能寄存器。地址以 0x0 或 0x8 结尾的 SFR（例如 P0、TCON、SCON 等）既可以字节寻址也可以位寻址，其他所有 SFR 只能字节寻址。

SFR 空间中未使用的地址保留以后使用，访问这些会产生不确定结果，应避免访问。

7.1.4.1 特殊功能寄存器 (SFR) 分页

特殊功能寄存器 (SFR) 分页，使得从 0x80 到 0xff 的 SFR 地址空间可以映射更多的寄存器。JMT1801ED 芯片 SFR 分为 3 页 (PAGE)：PAGE0/PAGE1/PAGE2。通过配置寄存器 SFRPAGE 来选择所属的页。

读写特殊功能寄存器地址空间的寄存器步骤如下：

- 配置寄存器 SFRPAGE，选择合适的 PAGE 号；
- 通过直接寻址方式，使用 MOV 指令读/写特殊功能寄存器 (SFR)。

7.1.4.2 中断和 SFR 分页

若 JMT51 检测某一个中断，寄存器 SFRPAGE 将自动的切换至该中断模块对应的 PAGE 号，该功能极大的降低了软件在中断服务程序中对 PAGE 进行切换的负担。中断服务程序响应完成之后，退出中断，寄存器 SFRPAGE 又将自动的切换至进入中断服务程序之前的 PAGE 号。该功能通过一个 5 级 SFR PAGE 堆栈完成，该堆栈不占用 JMT51 内部数据存储器。

5 级 SFR PAGE 堆栈，最上面一级的堆栈为当前使用的 PAGE，也即寄存器 SFRPAGE，往下依次为 PGSTACK0、PGSTACK1、PGSTACK2、PGSTACK3，如 Figure 55 所示：

- 当检测到某一中断，寄存器 SFRPAGE 将自动压栈至 PGSTACK0，PGSTACK0 将自动压栈至 PGSTACK1，PGSTACK1 将自动压栈至 PGSTACK2，PGSTACK2 将自动压栈至 PGSTACK3，然后硬件自动将该中断所属模块对应的 PAGE 号加载至寄存器 SFRPAGE。
- 当退出某一中断时，之前存储至寄存器 PGSTACK0 的 PAGE 号将自动加载至寄存器 SFRPAGE，PGSTACK1 将自动出栈至 PGSTACK0，PGSTACK2 将自动出栈至 PGSTACK1，PGSTACK3 将自动出栈至 PGSTACK2，PGSTACK3 将自动赋值 0x00。
- 在中断服务程序中，如果需要修改当前 PAGE 号，用户可配置寄存器 SFRPAGE 进行修改。配置寄存器 SFRPAGE 不会发生压栈/出栈的动作，仅会更改当前使用的 PAGE 号。
- 在中断服务程序中，如果需要，用户可通过配置栈寄存器 PGSTACK0、PGSTACK1、PGSTACK2 和 PGSTACK3，修改中断返回之后的 PAGE。配置栈寄存器不会发生

压栈/出栈的动作，仅当发生中断时，才会发生压栈/出栈的动作。

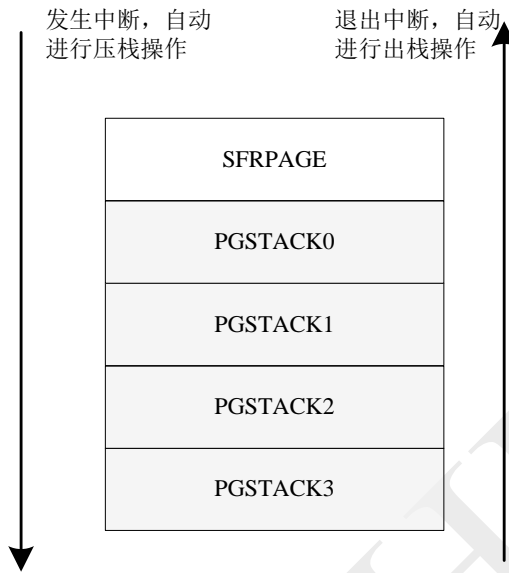


Figure 55 SFR PAGE 堆栈

通过配置寄存器 SFRPGEN，可以对硬件基于发生中断自动压栈，退出中断自动出栈的动作进行开关，默认为使能该功能。

Table 113 列出了 JMT1801ED 各模块对应的 PAGE 号：

Table 113 各模块对应的 PAGE 号

PAGE0	PAGE1	PAGE2
Timer0/1	PWM	GPIO
Timer2	CORDIC	PAGE 堆栈
UART	DSP	CRM
SPI		
I2C		
PMU		
WDT		
CMP		
ADC		
FLASH		

Table 115、Table 116 和 Table 117 分别列出了 PAGE0、PAGE1 和 PAGE2 地址对应的寄存器名，从中可以看出，部分寄存器在所有 PAGE 中均可以访问，称之为公共寄存器，公共寄存器所包括的寄存器见 Table 114 所示。

Table 114 特殊功能寄存器（公共寄存器）

公共寄存器								
F8H	IP2							
F0H	B							
E8H	IP1							
E0H	ACC							
D8H	IP0							
D0H	PSW							
C8H	P4							
C0H	IEN2							
B8H	IEN1							
B0H	P3							
A8H	IEN0							
A0H	P2							
98H								
90H	P1							
88H								
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

Table 115 特殊功能寄存器(PAGE0)

PAGE0								
F8H	IP2	FLSC	FLSK	FLSM				
F0H	B							
E8H	IP1	I2CDUTYH	I2CHOLD	I2CWBUF	I2CRBUF	I2CSTS	I2CISC	I2CIEN
E0H	ACC	I2CCON	I2CSADDRL	I2CSADDRH	I2COADDRL	I2COADDRH	I2CDIV	I2CDUTYL
D8H	IP0	WDTCKDIV	WDTINT	WDTINTF	WDTLD	WDTST		
D0H	PSW	T2CON	RL2	RH2	TL2	TH2	T2PSC	
C8H	P4	SPICON	SPIIE	SPIF	SPIBR	SPIBUF	SPISPC	
C0H	IEN2	OCPRSEL	ODPRSEL	BKPRSEL				
B8H	IEN1	CMPEN	CMPFT	CMPVDD	CMPREF	CMPINT	CMPFLAG	
B0H	P3							
A8H	IEN0							
A0H	P2	PMUCTR						

PAGE0								
98H	SCON	SBUF	SRELL	SRELH	SIRCON			
90H	P1	ADCCON0	ADCCON1	ADCPGAC	ADCETC	ADCPLY	ADCPL	ADCPLH
88H	TCON	TMOD	TL0	TL1	TH0	TH1	TPSC	
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

Table 116 特殊功能寄存器(PAGE1)

PAGE1								
F8H	IP2	TIM_PHASEH	TIM_CNTEN					
F0H	B	TIM_EGR	TIM_PHACON	TIM_ISR1	TIM_ISR2	TIM_CNTH	TIM_CNTH	TIM_PHASEL
E8H	IP1	TIM_CC1RL	TIM_CC1RH	TIM_CC2RL	TIM_CC2RH	TIM_CC3RL	TIM_CC3RH	TIM_BRKC
E0H	ACC	TIM_ARRL	TIM_ARRH	TIM_PSCL	TIM_PSCH	TIM_RCR	TIM_CC0RL	TIM_CC0RH
D8H	IP0	TIM_CC0MR	TIM_CC1MR	TIM_CC2MR	TIM_CC3MR	TIM_CCENR	TIM_CCPS	TIM_DTC
D0H	PSW	TIM_CONR0	TIM_CONR1	TIM_CONR2	TIM_CONR3	TIM_TGICR0	TIM_TGICR1	TIM_IER
C8H	P4	CORDIC_START	CORDIC_MODE	CORDIC_STATUS	CORDIC_INT			
C0H	IEN2	XN_REG_LOW	XN_REG_HIGH	YN_REG_LOW	YN_REG_HIGH	ZN_REG_LOW	ZN_REG_HIGH	
B8H	IEN1	X0_REG_LOW	X0_REG_HIGH	Y0_REG_LOW	Y0_REG_HIGH	Z0_REG_LOW	Z0_REG_HIGH	
B0H	P3	DSP_PSRH	DSP_PMRL	DSP_PMRH	DSP_LOOPL	DSP_LOOPH	DSP_LIRL	DSP_LIRH
A8H	IEN0	DSP_A0H	DSP_A1L	DSP_A1H	DSP_A2	DSP_LNKRL	DSP_LNKRH	DSP_PSRL
A0H	P2	DSP_R7L	DSP_R7H	DSP_DP0L	DSP_DP0H	DSP_DP1L	DSP_DP1H	DSP_A0L
98H		DSP_R3H	DSP_R4L	DSP_R4H	DSP_R5L	DSP_R5H	DSP_R6L	DSP_R6H
90H	P1	DSP_R0L	DSP_R0H	DSP_R1L	DSP_R1H	DSP_R2L	DSP_R2H	DSP_R3L
88H		DSP_STA	DSP_PCL	DSP_PCH	DSP_CFG			
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

Table 117 特殊功能寄存器(PAGE2)

PAGE2								
F8H	IP2	PGSTACK0	PGSTACK1	PGSTACK2	PGSTACK3			
F0H	B	CMP_CLK_DIV	GPIO_CLK_DIV					
E8H	IP1	CLK_EN0	CLK_EN1			PWM_CLK_DIV		
E0H	ACC							
D8H	IP0	BANDGAP_CTRL	SYS_CLK_SEL	ADC_CLK_DIV	JTAGEN	TRIMKEY	BUZZER_CTRL0	BUZZER_CTRL0
D0H	PSW	TRMVAL32K	CRMCTRL	ADCTRIM	TRIM_CNT_72ML	TRIM_CNT_72MH	OSCPD	SYS_CLK_DIV
C8H	P4	LDO_CTRL	INSTRST	SWRST0	SWRST1	GLOBALRST	RSTFLAG	TRMVAL72M

PAGE2								
C0H	IEN2	PA7CTRL	NRSTFQS	AFREMAP1	EXINTPE	EXINTNE	EXINTEN	ADCOEN
B8H	IEN1	PA0CTRL	PA1CTRL	PA2CTRL	PA3CTRL	PA4CTRL	PA5CTRL	PA6CTRL
B0H	P3	EXINTSEL0	EXINTSEL1	EXINTSEL2	EXINTSEL3	EXINTCON	AFREMAP0	FPSC
A8H	IEN0	PB7CTRL	PC7CTRL	PD7CTRL	PE7CTRL	PADSRDR0	PADSRDR1	JTAGEN
A0H	P2	PE0CTRL	PE1CTRL	PE2CTRL	PE3CTRL	PE4CTRL	PE5CTRL	PE6CTRL
98H		PD0CTRL	PD1CTRL	PD2CTRL	PD3CTRL	PD4CTRL	PD5CTRL	PD6CTRL
90H	P1	PC0CTRL	PC1CTRL	PC2CTRL	PC3CTRL	PC4CTRL	PC5CTRL	PC6CTRL
88H		PB0CTRL	PB1CTRL	PB2CTRL	PB3CTRL	PB4CTRL	PB5CTRL	PB6CTRL
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

7.2 JMT018 存储器

JMT018 DSP 存储器由三部分组成：16 位程序读总线的程序存储器、16 位 X 总线数据存储器 and 16 位 Y 总线数据存储器，如 Figure 56 所示。

JMT018 DSP 程序存储器共享 JMT51 存储器：FLASH，通过 16 位程序读总线进行访问。JMT018 DSP 通过程序总线对 FLASH 的寻址地址为 0x0000~0x3fff。

JMT018 DSP X 总线数据存储器共享 JMT51 外部数据存储器 DM0，通过 JMT018 DSP 的 16 位 X 数据空间总线进行访问。JMT018 DSP 通过 X 数据总线寻址地址为 0x0000~0x003f。

JMT018 DSP Y 总线数据存储器共享 JMT51 外部数据存储器 DM1，通过 JMT018 DSP 的 16 位 Y 数据空间总线进行访问。JMT018 DSP 通过 Y 数据总线寻址地址为 0x0000~0x003f。

通过 JMT018 DSP 的 16 位 X 数据空间总线，还可以访问：（1）CORDIC 协处理器的寄存器；（2）ADC 的采样结果寄存器。具体地址见章节 12.3.2 和章节 16.11.2。

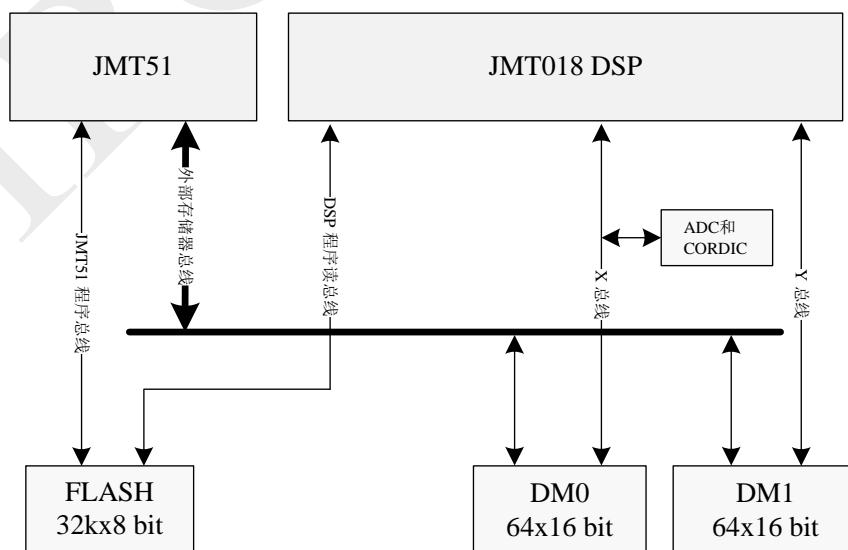


Figure 56 JMT018 DSP 存储器分布图

7.3 存储器访问冲突说明

JMT018 DSP 的所有存储器均为共享 JMT51 的存储器，包括 JMT51 的程序存储器（FLASH）和外部数据存储器（DM0/DM1），使用有如下限制：

- 当 JMT51 和 JMT018 DSP 同时访问 FLASH 空间时，硬件设置了保护机制，软件无需关注访问是否冲突。若 JMT51 和 JMT018 DSP 同时访问 FLASH 空间，FLASH 空间控制权优先提供给 JMT018 DSP，待 JMT018 DSP 访问完成之后，访问权将自动交给 JMT51，JMT51 将自动继续访问 FLASH。基于 DSP 优先级更高，若同时访问 FLASH 空间，JMT51 将停止工作直至 DSP 不再访问 FLASH。
- 当 JMT51 和 JMT018 DSP 同时访问相同数据存储器（DM0/DM1）时，JMT018 DSP 优先级更高，只有等待 DSP 访问完成之后，JMT51 才能够进行访问。JMT51 需要确保 JMT018 DSP 没有访问该块存储器时，JMT51 才能访问，否则 JMT51 访问数据出错。只有当 JMT018 DSP 不工作时，JMT51 才能访问数据存储器（DM0/DM1），JMT51 可通过读取 DSP 的工作状态寄存器，获取 DSP 是否正在工作。
- 若 JMT018 DSP 正在工作，建议芯片不进入 IDLE 工作模式，待 JMT018 DSP 停止工作后，可以根据情况配置寄存器，使芯片进入各种低功耗模式。

7.4 JMT51 核寄存器

JMT51 核有 7 个寄存器，对应的 SFRPAGE 为所有 PAGE（公共寄存器），也即在所有 PAGE 配置 JMT51 核寄存器均生效。具体说明如下。

Table 118 JMT51 核寄存器（所有 PAGE）

地址	寄存器名	属性	复位值	功能描述
0xE0	ACC	W/R	0x00	累加器 ACC
0xF0	B	W/R	0x00	B 寄存器
0x81	SP	W/R	0x07	堆栈指针(SP)
0xD0	PSW	W/R	0x00	程序状态字(PSW)寄存器
0x82	DPL	W/R	0x00	数据指针低字节(DPL)
0x83	DPH	W/R	0x00	数据指针高字节(DPH)
0x86	MOVXCON	W/R	0x01	MOVX 指令设置寄存器(MOVXCON)

7.4.1 ACC 寄存器(A) [0xE0]

Table 119 ACC 寄存器(A)

参数名	比特位	属性	复位值	描述
ACC	7-0	W/R	0	ACC 可以直接寻址，也可以位寻址，位地址范围 0xe0~0xe7

7.4.2 B 寄存器(B) [0xF0]

Table 120 B 寄存器(B)

参数名	比特位	属性	复位值	描述
B	7-0	W/R	0	B 可以直接寻址，也可以位寻址，位地址范围 0xf0~0xf7； 乘法运算中存放乘数，运算结束后，存放计算结果高 8 位； 除法运算中存放除数，运算结束后，存放余数。

7.4.3 堆栈指针(SP) [0x81]

Table 121 堆栈指针寄存器(SP)

参数名	比特位	属性	复位值	描述
SP	7-0	W/R	0111	在执行 PUSH、各种子程序调用、中断响应指令时，SP 先加 1，再将数据压栈； 执行 POP、RET、RETI 指令时，数据退出堆栈后 SP 再减 1。 堆栈栈顶可以是 128 字节内部数据存储器任意地址。

7.4.4 程序状态字寄存器(PSW) [0xD0]

程序状态字寄存器包含了程序状态信息。

PSW 可以直接寻址，可以位寻址，位地址范围 (0xd0~0xd7)。

Table 122 程序状态字寄存器(PSW)

参数名	比特位	属性	复位值	描述
P	0	W/R	0	奇偶标志位： 0: 累加器 A 中 1 的位数为偶数 1: 累加器 A 中 1 的位数为奇数
F1	1	W/R	0	用户标志位： 受软件控制的通用标志位，用户可以自己定义。
OV	2	W/R	0	溢出标志位： 0: 没有溢出发生 1: 下列情况下溢出： ● ADD、ADDC 或 SUBB 指令引起

参数名	比特位	属性	复位值	描述
				符号位变化溢出 <ul style="list-style-type: none"> ● MUL 指令结果大于 255 ● DIV 指令除数为 0
RS1~RS0	4-3	W/R	0	R0~R7 寄存器区选择: RS1 RS0 0 0 : 组 0 0 1 : 组 1 1 0 : 组 2 1 1 : 组 3
F0	5	W/R	0	用户标志位: 受软件控制的通用标志位, 用户可以自己定义。
AC	6	W/R	0	辅助进位标志位: 0: 算术运算中, 没有低字节向高字节进位或借位 1: 算术运算中, 有低字节向高字节进位或借位
C	7	W/R	0	进位标志位: 0: 算术运算中, 没有进位或借位发生 1: 算术运算中, 有进位或借位发生

7.4.5 数据指针低字节(DPL) [0x82]

Table 123 数据指针低字节(DPL)

参数名	比特位	属性	复位值	描述
DPL	7-0	W/R	0	数据指针低字节, 为 16 位数据指针 DPTR 的低字节。 DPTR 用于访问间接寻址的外部数据存储器或 FLASH 存储器。

7.4.6 数据指针高字节(DPH) [0x83]

Table 124 数据指针高字节(DPH)

参数名	比特位	属性	复位值	描述
DPH	7-0	W/R	0	数据指针高字节, 为 16 位数据指针 DPTR 的

参数名	比特位	属性	复位值	描述
				高字节。 DPTR 用于访问间接寻址的外部数据存储器或 FLASH 存储器。

7.4.7 MOVX 指令设置寄存器(MOVXCON) [0x86]

Table 125 MOVX 指令设置寄存器(MOVXCON)

参数名	比特位	属性	复位值	描述
XMCON	2-0	W/R	001	MOVX 存储器访问等待周期： 000: 读 1 个周期，写 1 个周期 001: 读 2 个周期，写 1 个周期 010: 读 3 个周期，写 2 个周期 011: 读 4 个周期，写 3 个周期 100: 读 5 个周期，写 4 个周期 101: 读 6 个周期，写 5 个周期 110: 读 7 个周期，写 6 个周期 111: 读 8 个周期，写 7 个周期
Reversed	6-3	R	0	保留
PMW	7	W/R	0	MOVX 指令访问空间选择： 0: MOVX 指令访问空间为外部数据存储器 1: MOVX 指令访问空间为 FLASH 存储器

其他特殊功能寄存器（SFR）详见各个模块的说明。

7.5 PAGE 堆栈寄存器

PAGE 堆栈寄存器分为两部分：

- SFRPAGE 和 SFRPGEN。这两个寄存器对应的 SFRPAGE 为所有 PAGE（公共寄存器），也即在所有 PAGE 配置这两个寄存器均生效。
- 堆栈寄存器 PGSTACK0、PGSTACK1、PGSTACK2 和 PGSTACK3。对应的 SFRPAGE 为 2。

Table 126 PAGE 堆栈寄存器

地址	寄存器名	属性	复位值	功能描述	PAGE 号
0x84	SFRPAGE	W/R	0x00	SFR PAGE 寄存器，指示当前特殊功能寄存器所属的 PAGE 号	所有 PAGE
0x85	SFRPGEN	W/R	0x01	PAGE 堆栈使能	所有 PAGE

地址	寄存器名	属性	复位值	功能描述	PAGE 号
0xF9	PGSTACK0	W/R	0x00	PAGE 堆栈寄存器 0	PAGE2
0xFA	PGSTACK1	W/R	0x00	PAGE 堆栈寄存器 1	PAGE2
0xFB	PGSTACK2	W/R	0x00	PAGE 堆栈寄存器 2	PAGE2
0xFC	PGSTACK3	W/R	0x00	PAGE 堆栈寄存器 3	PAGE2

7.5.1 SFR PAGE 寄存器(SFRPAGE) [0x84]

Table 127 SFR PAGE 寄存器(SFRPAGE)

参数名	比特位	属性	复位值	描述
SFRPAGE	2-0	W/R	0	SFR PAGE 寄存器, 指示当前特殊功能寄存器所属的 PAGE 号: 000: PAGE0 001: PAGE1 010: PAGE2 其它: PAGE0
Reserved	7-3	R	0	保留

7.5.2 PAGE 堆栈使能寄存器(SFRPGEN) [0x85]

Table 128 PAGE 堆栈使能寄存器(SFRPGEN)

参数名	比特位	属性	复位值	描述
SFRPGEN	0	W/R	1	PAGE 堆栈使能: 0: 不使能 1: 使能
Reserved	7-1	R	0	保留

7.5.3 PAGE 堆栈寄存器 0(PGSTACK0) [0xF9]

Table 129 PAGE 堆栈寄存器 0(PGSTACK0)

参数名	比特位	属性	复位值	描述
PGSTACK0	2-0	W/R	0	PAGE 堆栈寄存器 0
Reserved	7-3	R	0	保留

7.5.4 PAGE 堆栈寄存器 1(PGSTACK1) [0xFA]

Table 130 PAGE 堆栈寄存器 1(PGSTACK1)

参数名	比特位	属性	复位值	描述
PGSTACK1	2-0	W/R	0	PAGE 堆栈寄存器 1

参数名	比特位	属性	复位值	描述
Reserved	7-3	R	0	保留

7.5.5 PAGE 堆栈寄存器 2(PGSTACK2) [0xFB]

Table 131 PAGE 堆栈寄存器 2(PGSTACK2)

参数名	比特位	属性	复位值	描述
PGSTACK2	2-0	W/R	0	PAGE 堆栈寄存器 2
Reserved	7-3	R	0	保留

7.5.6 PAGE 堆栈寄存器 3(PGSTACK3) [0xFC]

Table 132 PAGE 堆栈寄存器 3(PGSTACK3)

参数名	比特位	属性	复位值	描述
PGSTACK3	2-0	W/R	0	PAGE 堆栈寄存器 3
Reserved	7-3	R	0	保留

8 Flash 存储器

8.1 概述

JMT1801ED 采用 FLASH 作为程序存储空间，FLASH 主程序存储空间（MAIN）大小为 32K Bytes，并包括 2 个非易失性扇区（NVR0 和 NVR1，每个 512 Bytes），其中 NVR0 部分空间用于存储芯片校准值，NVR1 用户可自由使用。

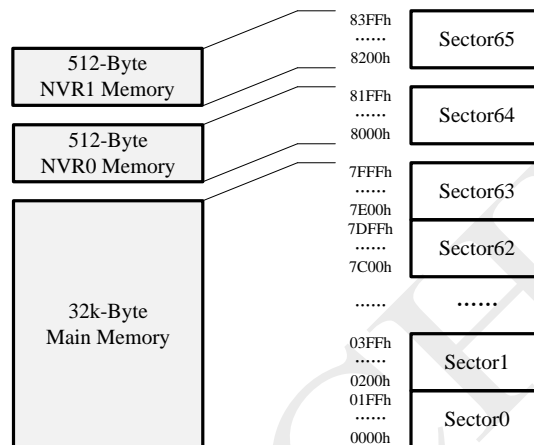


Figure 57 FLASH Memory

FLASH 特性如 Table 133 所示。

Table 133 FLASH 特性

特性参数	大小
扇区（Sector）大小	512Bytes
数据保持时间	最少 100 年
擦除次数	最少 20,000 次

8.2 FLASH 操作

FLASH 在编程之前要先进行擦除，支持对 FLASH 的以下操作：

- 支持 MAIN，NVR0 和 NVR1 读取
- 支持 MAIN，NVR0 和 NVR1 扇区（SECTOR）擦除
- 支持 MAIN，NVR0 和 NVR1 编程

8.2.1 FLASH 数据读取

读取 Flash 中的数据，在汇编程序中可以使用以下两种方式：

➢ “MOVX A,@DPTR”方式如下：

- 1) 寄存器 MOVXCON.PMW 置 1，选择 MOVX 指令对 FLASH 的操作
- 2) 寄存器 FLSC.OM 设置为“000”（默认），选择对 FLASH 读取操作

- 3) 将要读取的 FLASH 地址写入 DPTR, 使用“MOVX A,@DPTR”指令, 将数据存入累加器 A
- “MOVX A,@DPTR”方式如下:
- 1) FLSC.OM 设置为“000” (默认), 选择对 FLASH 读取操作
 - 2) 将要读取的 FLASH 地址写入 DPTR, 清零累加器 A, 使用“MOVX A,@DPTR”指令, 将数据存入累加器 A

读取 Flash 中的数据, C 语言实现例程如下:

- “MOVX A,@DPTR”方式:
- ```
MOVXCON |= 0x80;
buffer = *((unsigned char xdata *) address);
MOVXCON &= 0x7F;
```
- “MOVX A,@DPTR”方式:
- ```
buffer = *((unsigned char code *) address);
```

8.2.2 FLASH 扇区擦除

操作流程如下:

- 1) 配置寄存器 WDTST 为 0, 关闭 WDT
 - 2) 寄存器 MOVXCON.PMW 置 1, 选择“MOVX @DPTR,A”对 FLASH 进行操作
 - 3) 向 FLSK 先写入 0xA5, 再写入 0xF1, 解除对 FLASH 操作模式的锁定
 - 4) 寄存器 FLSC.OM 设置为“011”, 选择对 FLASH 扇区擦除操作
 - 5) 将要擦除的 FLASH 扇区首地址写入 DPTR, 向累加器 A 写入任意数据, 使用“MOVX @DPTR,A”指令, 启动对 FLASH 该扇区的擦除操作
 - 6) FLASH 擦除过程中, MCU 处于等待状态, 擦除完成后, MCU 继续取指工作
- FLASH 擦除时间较长, 擦除前需要关闭 WDT, 防止在擦除过程中产生 WDT 复位。

8.2.3 FLASH 编程

操作流程如下:

- 1) 配置寄存器 WDTST 为 0, 关闭 WDT
- 2) 寄存器 MOVXCON.PMW 置 1, 选择“MOVX @DPTR,A”对 FLASH 进行操作
- 3) 向 FLSK 先写入 0xA5, 再写入 0xF1, 解除对 FLASH 操作模式的锁定
- 4) 寄存器 FLSC.OM 设置为“001”, 选择对 FLASH 的编程操作
- 5) 将要编程的 FLASH 地址写入 DPTR, 向累加器 A 写入需要编程的数据, 使用“MOVX @DPTR,A”指令, 启动对 FLASH 的按字节编程操作
- 6) FLASH 编程过程中, MCU 处于等待状态, 编程完成后, MCU 继续取指工作
- 7) 重复操作 3) 到 6), 直到完成所有字节的编程

FLASH 编程时间较长，编程前需要关闭 WDT，防止在编程过程中产生 WDT 复位。

8.3 FLASH 寄存器

该模块共有 2 个寄存器，对应的 SFRPAGE 为 0，具体列表如 Table 134 所示。

Table 134 FLASH 寄存器 (SFRPAGE=0x00)

地址	寄存器名	类型	复位值	功能描述
0xF9	FLSC	W/R	0x00	FLASH 控制寄存器
0xFA	FLSK	W/R	0x00	FLASH 关键字寄存器

8.3.1 FLASH 控制寄存器(FLSC) [0xF9]

Table 135 FLASH 控制寄存器(FLSC)

参数名	比特位	属性	复位值	描述
OM	2-0	W/R	0	FLASH 操作模式控制位： 000：读操作（默认） 001：编程 011：扇区擦除 其他：保留
Reserved	7-3	R	0	保留

8.3.2 FLASH 关键字寄存器(FLSK) [0xFA]

Table 136 FLASH 关键字寄存器(FLSK)

参数名	比特位	属性	复位值	描述
FLSK	7-0	W/R	0	只有向 FLSK 中顺序写入 0xA5 和 0xF1 后，FLSC.OM 比特才能更新；OM 写入 0 时，没有此限制； FLSK.0 回读值指示 0xA5 是否写入，若继续写入 0xF1，该位回读值清零； FLSK.1 回读值指示 0xA5 和 0xF1 已经连续写入，当对 FLSC.OM 写入更新后，FLSK.0 和 FLSK.1 自动清零。

9 看门狗定时器 (WDT)

9.1 概述

看门狗定时器 (WDT) 可用于检测 and 解决由软件错误引起的故障。

看门狗定时器 (WDT) 主要功能如下:

- 16 位计数器;
- 定时时钟可由 32KHz OSC 及其分频得到 (1-8 分频可配置);
- 可产生中断和复位;

9.2 功能描述

芯片正常工作时, 每隔一段时间需要配置 WDTLD 寄存器为 1 对 WDT 进行喂狗操作, 使得 WDT 计数器重新初始化并开始新一轮计数。若不及时喂狗, 计数溢出后则会产生复位信号进行全芯片复位。

WDT 有两种工作模式:

- 先中断后复位模式: 配置 WDTCKDIV.WDTINTS 为 0 时 WDT 工作于先中断后复位模式。该模式下当 WDT 从 0xFFFF 递减至 16 位比较值 WDTINT 时, 产生中断, 软件需要根据该中断配置 WDTLD 为 1 进行喂狗操作, 若不及时进行喂狗操作, 当计数器递减至 0 时, 产生全芯片复位。
- 复位模式: 配置 WDTCKDIV.WDTINTS 为 1 时 WDT 工作于复位模式。该模式下 WDT 从 0xFFFF 递减至 16 位比较值 WDTINT 时, 产生全芯片复位。

9.3 寄存器

WDT 模块共有 5 个寄存器, 对应的 SFRPAGE 为 0, 具体说明如 Table 137 所示。

Table 137 WDT 寄存器 (SFRPAGE=0x00)

地址	寄存器名	属性	复位值	功能描述
0xD9	WDTCKDIV	W/R	0x00	看门狗分频系数寄存器
0xDA	WDTINT	W/R	0x00	看门狗比较值寄存器
0xDB	WDTINTF	W/R	0x00	看门狗中断标志寄存器
0xDC	WDTLD	W/R	0x00	看门狗喂狗寄存器
0xDD	WDTST	W/R	0x01	看门狗启停寄存器

9.3.1 看门狗分频系数寄存器(WDTCKDIV) [0xD9]

Table 138 看门狗分频系数寄存器(WDTCKDIV)

参数名	比特位	属性	复位值	描述
WDTINTS	0	W/R	0	计数达到 WDTINT 时产生复位或中断： 0：产生中断 1：产生复位 其中 WDTINT=[WDTINTH:0xFF]
CKDIV	3-1	W/R	0	工作时钟分频系数： 000：1 分频 001：2 分频 010：3 分频 011：4 分频 100：5 分频 101：6 分频 110：7 分频 111：8 分频
Reserved	7-4	R	0	保留

9.3.2 看门狗比较值寄存器(WDTINT) [0xDA]

Table 139 看门狗比较值寄存器(WDTINT)

参数名	比特位	属性	复位值	描述
WDTINTH	7-0	W/R	0	比较值寄存器高 8 位，低 8 位固定为 0xFF，可配置最大值为 0xFE，即当该寄存器配置为以下值时对应的 16 位比较值 WDTINT 如下所示： 0xFE：真实比较值为 0xFEFF 0xFD：真实比较值为 0xFDFE 0x01：真实比较值为 0x01FF 0x00：真实比较值为 0x00FF 该寄存器配置后需要配置 WDTLD 为 1 才会加载生效

9.3.3 看门狗中断标志寄存器(WDTINTF) [0xDB]

Table 140 看门狗中断标志寄存器(WDTINTF)

参数名	比特位	属性	复位值	描述
WDTINTFLG	0	W/R	0	看门狗中断标志位： <ul style="list-style-type: none"> 当看门狗有中断申请时，该位置为 1 软件配置该位为 1 可产生软中断，在中断处理程序中需要软件写 0 来清除
Reserved	7-1	R	0	保留

9.3.4 看门狗喂狗寄存器(WDTLD) [0xDC]

Table 141 看门狗喂狗寄存器(WDTLD)

参数名	比特位	属性	复位值	描述
WDTLD	0	W/R	0	WDT 喂狗信号： 1：进行喂狗操作，重加载比较值，重加载时钟分频计数值，看门狗初始计数从 0xFFFF 重新开始递减计数，硬件自清 0：不进行喂狗
Reserved	7-1	R	0	保留

9.3.5 看门狗启停寄存器(WDTST) [0xDD]

Table 142 看门狗启停寄存器(WDTST)

参数名	比特位	属性	复位值	描述
WDTST	1-0	W/R	01	WDT 启动关闭信号： 01/10：启动看门狗 其他：关闭看门狗
Reserved	7-2	R	0	保留

10 定时器/计数器 (TIMER)

10.1 概述

JMT1801ED 芯片有 3 个 16 位定时器/计数器: Timer0, Timer1 和 Timer2, 都有两种模式:

- 定时器模式: 以 timer_clk 时钟周期为基准进行定时。该模式下, 通过配置 TPSC 或 T2PSC 寄存器, 可对每个 Timer 的时钟进行预分频 (1/8/64/256 分频)。
- 计数器模式: 对外部输入管脚 T0、T1 或 T2 的下降沿进行计数。

Timer0 有 4 种工作模式:

- 模式 0: 13 位定时器/计数器
- 模式 1: 16 位定时器/计数器
- 模式 2: 8 位自动重装定时器/计数器
- 模式 3: 两个 8 位定时器/计数器

Timer1 有 3 种工作模式:

- 模式 0: 13 位定时器/计数器
- 模式 1: 16 位定时器/计数器
- 模式 2: 8 位自动重装定时器/计数器

Timer2 有 4 种工作模式:

- 模式 0: 13 位定时器/计数器
- 模式 1: 16 位定时器/计数器
- 模式 2: 16 位自动重装定时器/计数器
- 模式 3: 1 个 8 位定时器/计数器

10.2 TIMER 管脚配置

TIMER 的管脚配置可参见 Table 1 引脚说明表格, 管脚配置说明见 GPIO 寄存器说明。

10.3 Timer0 工作模式

配置寄存器 TMOD 中的 M1(TM0D.1)、M0(TM0D.0), 设置 Timer0 的工作模式。

10.3.1 模式 0(13 位定时器/计数器)

配置寄存器 TMOD 的标志位 TMOD[1:0]³=00, 使得 Timer0 工作在模式 0。配置寄存器 TMOD[2] (C/T 位) 可选择为定时器模式或计数器模式。

Timer0 被分为两个 8 位寄存器, 低字节 TL0 和高字节 TH0, TL0 的低 5 位和 TH0 组成

³ TMOD[1:0]代表寄存器 TMOD 的低两位

一个 13 位的计数器，当 Timer0 溢出时寄存器位 TF0 置位，同时产生 Timer0 溢出中断。程序进入中断后，该位被自动清零。

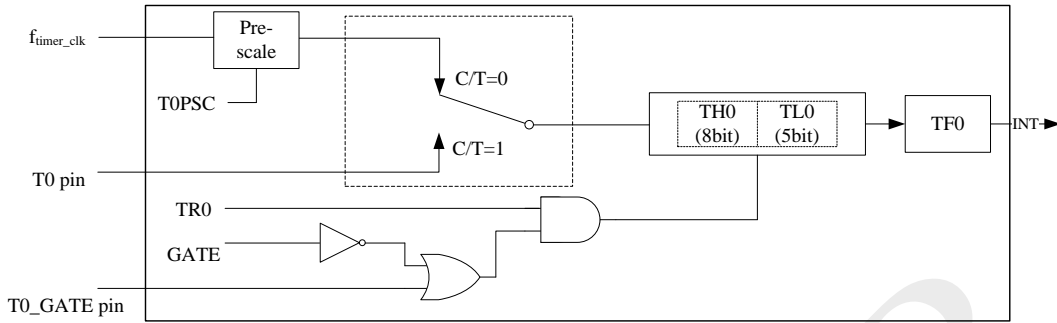


Figure 58 Timer0 模式 0 实现框图

当 C/T 为 0，Timer0 用作定时器时， $X=\{TH0[7:0],TL0[4:0]\}$ ，定时时间计算公式为：

$$T_d = (2^{13} - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.3.2 模式 1(16 位定时器/计数器)

配置寄存器 TMOD 的标志位 $TMOD[1:0]=01$ ，使得 Timer0 工作在模式 1。配置寄存器 TMOD[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 1 为 TL0 和 TH0 组成一个 16 位的计数器，当 Timer0 溢出时寄存器位 TF0 置位，同时产生 Timer0 溢出中断。程序进入中断后，该位被自动清零。

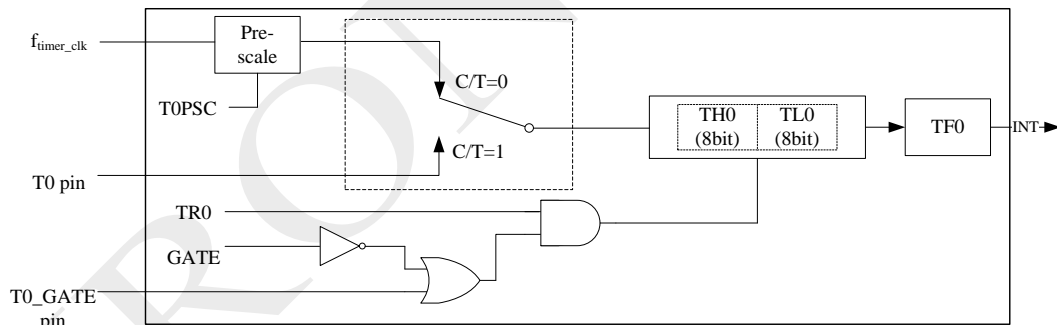


Figure 59 timer0 模式 1 实现框图

当 C/T 为 0，Timer0 用作定时器时， $X=\{TH0[7:0],TL0[7:0]\}$ ，定时时间计算公式为：

$$T_d = (2^{16} - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.3.3 模式 2(8 位自动重装定时器/计数器)

配置寄存器 TMOD 的标志位 $TMOD[1:0]=10$ ，使得 Timer0 工作在模式 2。配置寄存器 TMOD[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 2 中只有低字节寄存器 TL0 计数。当 Timer0 溢出时，TH0 中的值会自动装载到 TL0 中，TF0 置位，同时产生 Timer0 溢出中断。程序进入中断后，该位被自动清零。

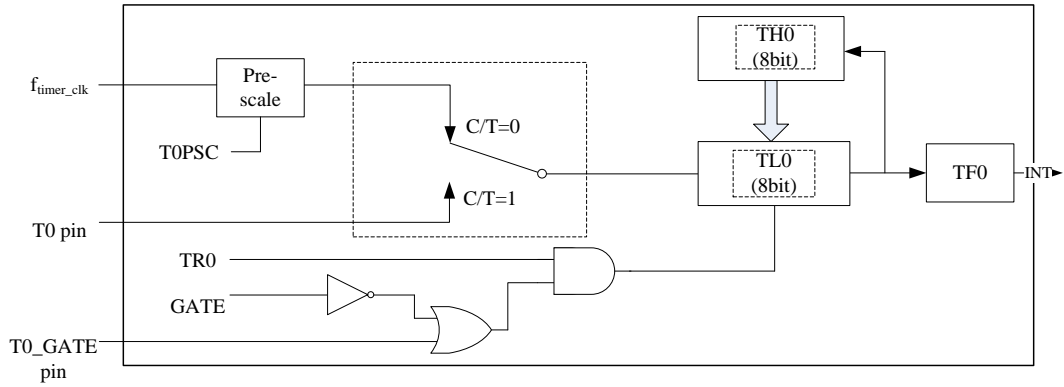


Figure 60 timer0 模式 2 实现框图

当 C/T 为 0, Timer0 用作定时器时, $X=TH0[7:0]$, 定时时间计算公式为:

$$T_d = (2^8 - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.3.4 模式 3(两个 8 比特定器)

配置寄存器 TMOD 的标志位 $TMOD[1:0]=11$, 使得 Timer0 工作在模式 3。配置寄存器 TMOD[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 3 中, 定时器/计数器的低字节受 TR0 控制, 高字节受 TR1 控制。低字节可作定时器/计数器, 高字节只能作定时器。

当 Timer0 低字节溢出时, TF0 置位, 同时产生 Timer0 溢出中断, 程序进入中断后, 该位被自动清零。

当 Timer0 的高字节溢出时, TF1 置位, 同时产生 Timer1 溢出中断, 程序进入中断后, 该位被自动清零。

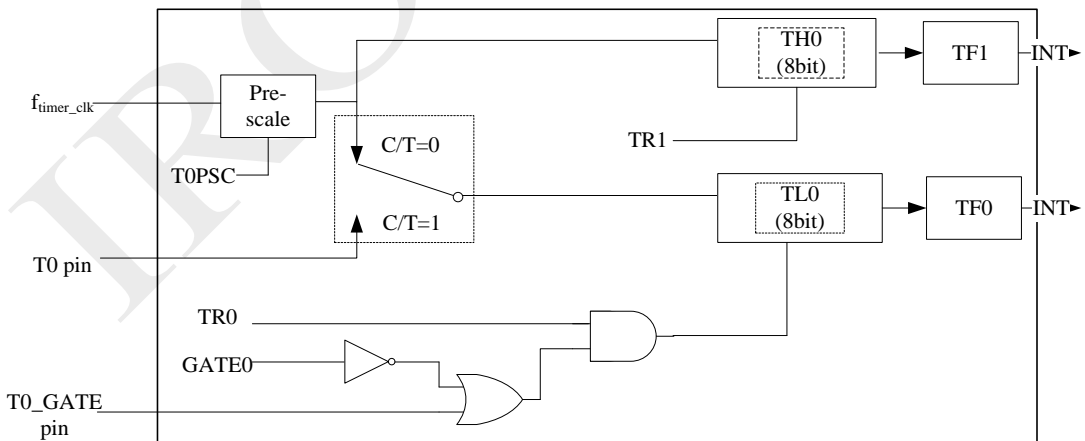


Figure 61 timer0 模式 3 实现框图

当 C/T 为 0, Timer0 用作定时器时, $X=TH0[7:0]$ 或 $TL0[7:0]$, 定时时间计算公式为:

$$T_d = (2^8 - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.4 Timer1 工作模式

配置寄存器 TMOD 中的 M1(TMOD.5)、M0(TMOD.4)，设置 Timer1 的工作模式。

10.4.1 模式 0(13 位定时器/计数器)

配置寄存器 TMOD 的标志位 TMOD[5:4]=00，使得 Timer1 工作在模式 0。配置寄存器 TMOD[6] (C/T 位) 可选择为定时器模式或计数器模式。

Timer1 被分为两个 8 位寄存器，低字节 TL1 和高字节 TH1，TL1 的低 5 位和 TH1 组成一个 13 位的计数器，当 Timer1 溢出时寄存器位 TF1 置位，同时产生 Timer1 溢出中断。程序进入中断后，该位被自动清零。

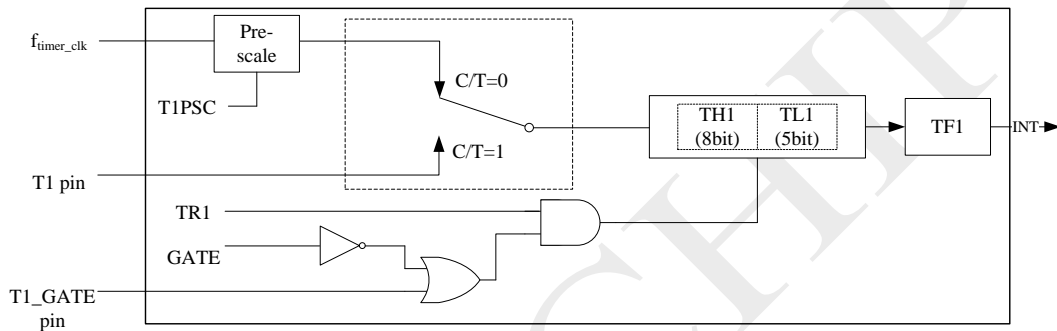


Figure 62 Timer1 模式 0 实现框图

当 C/T 为 0，Timer1 用作定时器， $X=\{TH1[7:0],TL1[4:0]\}$ ，定时时间计算公式为：

$$T_d = (2^{13} - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.4.2 模式 1(16 位定时器/计数器)

配置寄存器 TMOD 的标志位 TMOD[5:4]=01，使得 Timer1 工作在模式 1。配置寄存器 TMOD[6] (C/T 位) 可选择为定时器模式或计数器模式。

模式 1 为 TL1 和 TH1 组成一个 16 位的定时器/计数器，当 Timer1 溢出时寄存器位 TF1 置位，同时产生 Timer1 溢出中断。程序进入中断后，该位被自动清零。

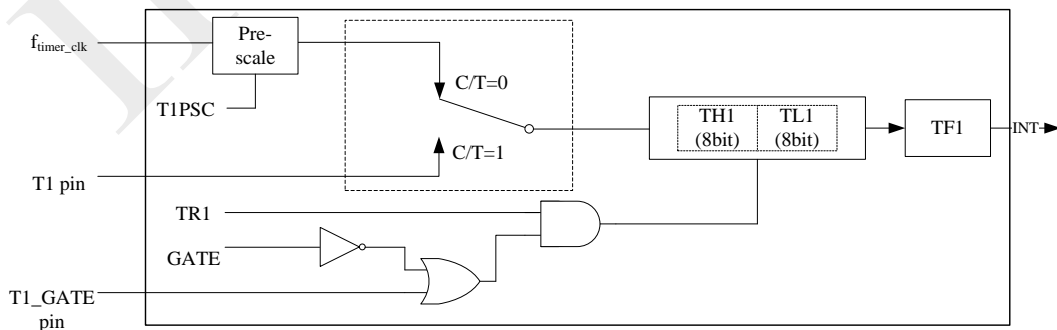


Figure 63 Timer1 模式 1 实现框图

当 C/T 为 0，Timer1 用作定时器时， $X=\{TH1[7:0],TL1[7:0]\}$ ，定时时间计算公式为：

$$T_d = (2^{16} - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.4.3 模式 2(8 位自动重装定时器/计数器)

配置寄存器 TMOD 的标志位 TMOD[5:4]=10，使得 Timer1 工作在模式 2。配置寄存器 TMOD[6]（C/T 位）可选择为定时器模式或计数器模式。

模式 2 中只有低字节寄存器 TL1 进行计数。当 Timer1 溢出时，TH1 中的值会自动装载到 TL1 中，TF1 置位，同时产生 Timer1 溢出中断。程序进入中断后，该位被自动清零。

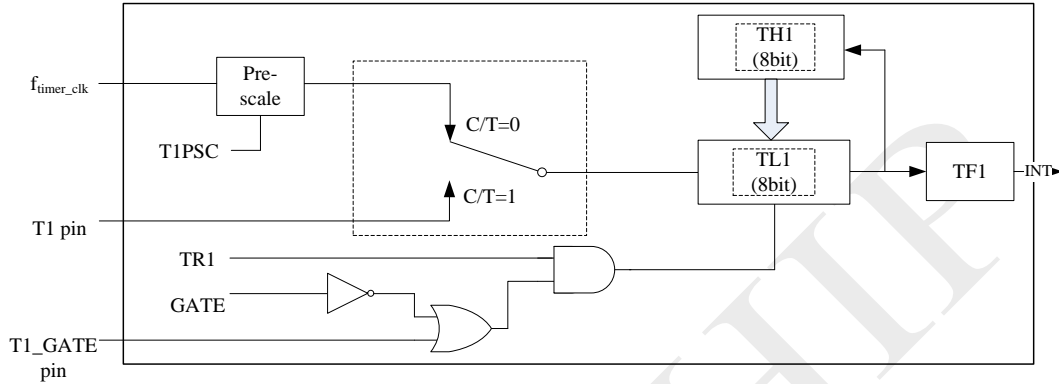


Figure 64 Timer1 模式 2 实现框图

当 C/T 为 0，Timer1 用作定时器时， $X=TH1[7:0]$ ，定时时间计算公式为：

$$T_d = (2^8 - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.5 Timer2 工作模式

配置寄存器 T2CON 中的 M1(T2CON.1)、M0(T2CON.0)，设置 Timer2 的工作模式。

10.5.1 模式 0(13 位定时器/计数器)

配置寄存器 T2CON 的标志位 T2CON[1:0]=00，使得 Timer2 工作在模式 0。配置寄存器 T2CON[2]（C/T 位）可选择为定时器模式或计数器模式。

Timer2 被分为两个 8 位寄存器，低字节 TL2 和高字节 TH2，TL2 的低 5 位和 TH2 组成一个 13 位的计数器，当 Timer2 溢出时寄存器位 TF2 置位，同时产生 Timer2 溢出中断。程序进入中断后，该位被自动清零。

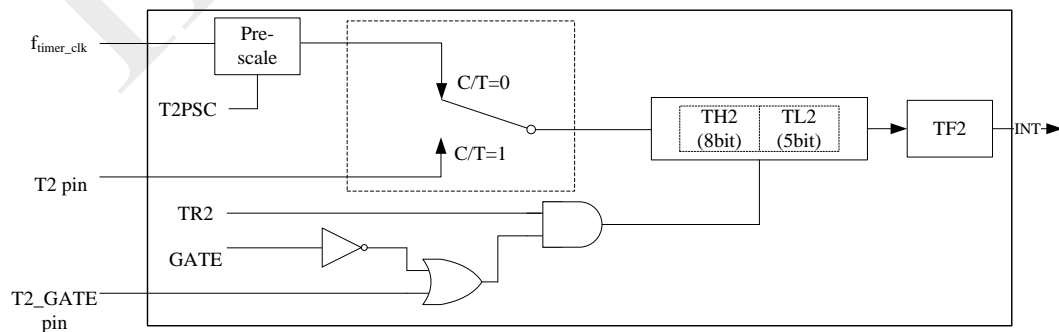


Figure 65 Timer2 模式 0 实现框图

当 C/T 为 0，Timer2 用作定时器， $X=\{TH2[7:0],TL2[4:0]\}$ ，定时时间计算公式为：

$$T_d = (2^{13} - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.5.2 模式 1(16 位定时器/计数器)

配置寄存器 T2CON 的标志位 T2CON[1:0]=01, 使得 Timer2 工作在模式 1。配置寄存器 T2CON[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 1 为 TL2 和 TH2 组成一个 16 位的计数器, 当 Timer2 溢出时寄存器位 TF2 置位, 同时产生 Timer2 溢出中断。程序进入中断后, 该位被自动清零。

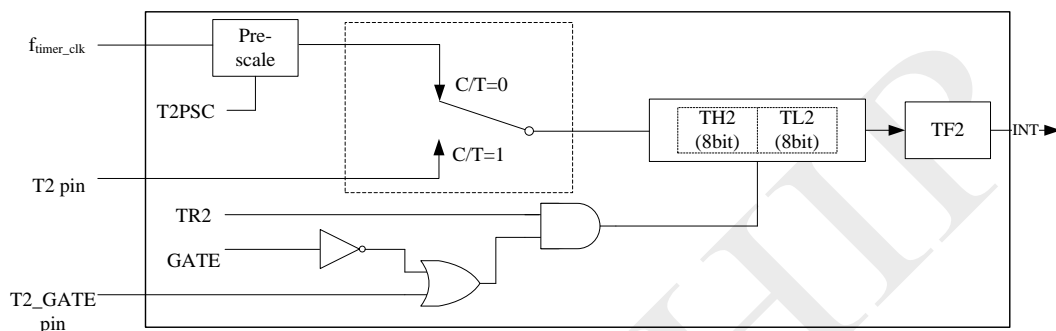


Figure 66 timer2 模式 1 实现框图

当 C/T 为 0, Timer2 用作定时器时, $X=\{TH2[7:0],TL2[7:0]\}$, 定时时间计算公式为:

$$T_d = (2^{16} - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.5.3 模式 2(16 位自动重装定时器/计数器)

配置寄存器 T2CON 的标志位 T2CON[1:0]=10, 使得 Timer2 工作在模式 2。配置寄存器 T2CON[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 2 中 TL2 和 TH2 组成一个 16 位的计数器, 当 Timer2 溢出时, RH2 中的值会自动装载到 TH2 中, RL2 中的值会自动装载到 TL2 中, TF2 置位, 同时产生 Timer2 溢出中断。程序进入中断后, 该位被自动清零。

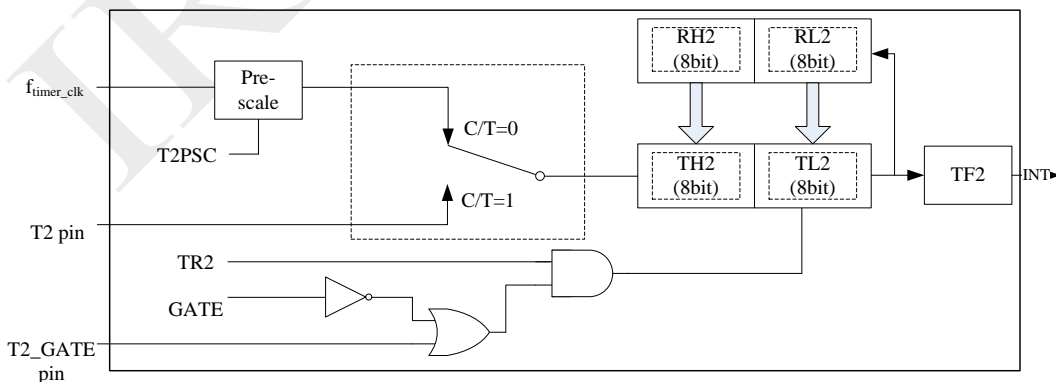


Figure 67 timer2 模式 2 实现框图

当 C/T 为 0, Timer2 用作定时器时, $X=\{RH2[7:0],RL2[7:0]\}$, 定时时间计算公式为:

$$T_d = (2^{16} - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.5.4 模式 3(1 个 8 比特定定时器/计数器)

配置寄存器 T2CON 的标志位 T2CON [1:0]=11, 使得 Timer2 工作在模式 3。配置寄存器 T2CON[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 3 中, 低字节可作定时器/计数器。

当 Timer2 低字节溢出时, TF2 置位, 同时产生 Timer2 溢出中断, 程序进入中断后, 该位被自动清零。

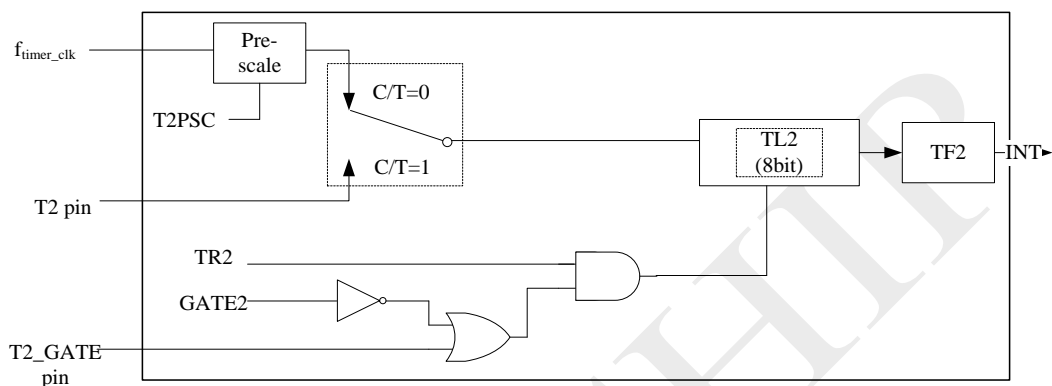


Figure 68 timer2 模式 3 实现框图

当 C/T 为 0, Timer2 用作定时器时, $X=TL2[7:0]$, 定时时间计算公式为:

$$T_d = (2^8 - X) \times T_{timer_clk} \times \text{预分频系数}$$

10.6 TIMER 寄存器

TIMER 模块有 13 个寄存器, 对应的 SFRPAGE 为 0, 具体说明如 Table 143 所示。

Table 143 TIMER 寄存器 (SFRPAGE=0x00)

地址	寄存器名	类型	复位值	功能描述
0x88	TCON	W/R	0x00	Timer0/1 控制寄存器
0x89	TMOD	W/R	0x00	Timer0/1 模式寄存器
0x8A	TL0	W/R	0x00	Timer0 低 8 位
0x8B	TL1	W/R	0x00	Timer1 低 8 位
0x8C	TH0	W/R	0x00	Timer0 高 8 位
0x8D	TH1	W/R	0x00	Timer1 高 8 位
0x8E	TPSC	W/R	0x00	Timer0/1 预分频控制器
0xD1	T2CON	W/R	0x00	Timer2 控制寄存器
0xD2	RL2	W/R	0x00	Timer2 重载寄存器低 8 位
0xD3	RH2	W/R	0x00	Timer2 重载寄存器高 8 位
0xD4	TL2	W/R	0x00	Timer2 低 8 位

地址	寄存器名	类型	复位值	功能描述
0xD5	TH2	W/R	0x00	Timer2 高 8 位
0xD6	T2PSC	W/R	0x00	Timer2 预分频控制器

10.6.1 Timer0/1 控制寄存器(TCON)[0x88]

Table 144 Timer0/1 控制寄存器(TCON)

参数名	比特位	属性	复位值	描述
UBSEL	0	W/R	0	UART 波特率产生选择位： 0: 使用 Timer 1 产生波特率 1: 使用 SRELL 和 SRELH 寄存器计数产生波特率
Reserved	1	R	0	保留
SMOD	2	W/R	0	UART 双倍波特率使能控制位： 0: 正常模式 1: 使用双倍波特率
Reserved	3	R	0	保留
TR0	4	W/R	0	Timer0 运行控制位： 0: 停止 1: 启动
TF0	5	W/R	0	Timer0 溢出中断标志： 0: 未溢出 1: 溢出
TR1	6	W/R	0	Timer1 运行控制位： 0: 停止 1: 启动
TF1	7	W/R	0	Timer1 溢出中断标志： 0: 未溢出 1: 溢出

10.6.2 Timer0/1 模式寄存器(TMOD)[0x89]

Table 145 Timer0/1 模式寄存器(TMOD)

参数名	比特位	属性	复位值	描述
M0	0	W/R	0	Timer0 模式, 见 Table 146 Timer0/1 工作模

参数名	比特位	属性	复位值	描述
M1	1	W/R	0	式选择描述
C/T	2	W/R	0	Timer0 用作定时器/计数器选择位： 0: 定时器 1: 计数器
GATE	3	W/R	0	Timer0 门控控制位： 0: T0_GATE 不起作用 1: T0_GATE 可起作用
M0	4	W/R	0	Timer1 模式，见 Table 146 Timer0/1 工作模式选择描述
M1	5	W/R	0	
C/T	6	W/R	0	Timer1 用作定时器/计数器选择位： 0: 定时器 1: 计数器
GATE	7	W/R	0	Timer1 门控控制位： 0: T1_GATE 不起作用 1: T1_GATE 可起作用

Table 146 Timer0/1 工作模式选择

M1	M0	模式	说明
0	0	0	13 位定时器/计数器，TL0 (TL1) 中的低 5 位和 TH0(TH1) 中的 8 位。
0	1	1	16 位定时器/计数器。
1	0	2	8 位定时器/计数器，具有自动装载功能，在 TL0(TL1) 溢出时装载 TH0(TH1) 的值。
1	1	3	Timer0 设置为该模式时，分成两个 8 位的独立计数器：TL0 使用控制位 TR0，溢出时将 TF0 置位；TH0 使用控制位 TR1，溢出时将 TF1 置位，该模式下 Timer1 不能工作。 Timer1 设置为该模式时停止工作。

10.6.3 Timer0 低 8 位(TL0)[0x8A]

Table 147 Timer0 低 8 位(TL0)

参数名	比特位	属性	复位值	描述
TL0	7-0	W/R	0	Timer0 低 8 位

10.6.4 Timer1 低 8 位(TL1)[0x8B]

Table 148 Timer1 低 8 位(TL1)

参数名	比特位	属性	复位值	描述
TL1	7-0	W/R	0	Timer1 低 8 位

10.6.5 Timer0 高 8 位(TH0)[0x8C]

Table 149 Timer0 高 8 位(TH0)

参数名	比特位	属性	复位值	描述
TH0	7-0	W/R	0	Timer0 高 8 位

10.6.6 Timer1 高 8 位(TH1)[0x8D]

Table 150 Timer1 高 8 位(TH1)

参数名	比特位	属性	复位值	描述
TH1	7-0	W/R	0	Timer1 高 8 位

10.6.7 Timer0/1 预分频控制器(TPSC)[0x8E]

Table 151 Timer0/1 预分频控制器(TPSC)

参数名	比特位	属性	复位值	描述
TOPSC	1-0	W/R	0	Timer0 时钟预分频系数： 00: 1 分频 01: 8 分频 10: 64 分频 11: 256 分频
T1PSC	3-2	W/R	0	Timer1 时钟预分频系数： 00: 1 分频 01: 8 分频 10: 64 分频 11: 256 分频
Reserved	7-4	R	0	保留

10.6.8 Timer2 控制寄存器(T2CON)[0xD1]

Table 152 Timer2 控制寄存器(T2CON)

参数名	比特位	属性	复位值	描述
M0	0	W/R	0	Timer2 模式, 见 Table 153 Timer2 工作模式选择
M1	1	W/R	0	

参数名	比特位	属性	复位值	描述
C/T	2	W/R	0	Timer2 用作定时器/计数器选择位： 0: 定时器 1: 计数器
GATE	3	W/R	0	Timer2 门控控制位： 0: T2_GATE 不起作用 1: T2_GATE 可起作用
TR2	4	W/R	0	Timer2 运行控制位： 0: 停止 1: 启动
TF2	5	W/R	0	Timer2 溢出中断标志： 0: 未溢出 1: 溢出
Reserved	7-6	R	0	保留

Table 153 Timer2 工作模式选择

M1	M0	模式	说明
0	0	0	13 位定时器/计数器，TL2 中的低 5 位和 TH2 中的 8 位。
0	1	1	16 位定时器/计数器。
1	0	2	16 位定时器/计数器，具有自动装载功能，在 Timer2 溢出时，{TH2,TL2}自动装载{RH2,RL2}的值。
1	1	3	8 位定时器/计数器。

10.6.9 Timer2 重载寄存器低 8 位(RL2)[0xD2]

Table 154 Timer2 重载寄存器低 8 位(RL2)

参数名	比特位	属性	复位值	描述
RL2	7-0	W/R	0	Timer2 重载寄存器低 8 位

10.6.10 Timer2 重载寄存器高 8 位(RH2)[0xD3]

Table 155 Timer2 重载寄存器高 8 位(RH2)

参数名	比特位	属性	复位值	描述
RH2	7-0	W/R	0	Timer2 重载寄存器高 8 位

10.6.11 Timer2 低 8 位(TL2)[0xD4]

Table 156 Timer2 低 8 位(TL2)

参数名	比特位	属性	复位值	描述
TL2	7-0	W/R	0	Timer2 低 8 位

10.6.12 Timer2 高 8 位(TH2)[0xD5]

Table 157 Timer2 高 8 位(TH2)

参数名	比特位	属性	复位值	描述
TH2	7-0	W/R	0	Timer2 高 8 位

10.6.13 Timer2 预分频控制器(T2PSC)[0xD6]

Table 158 Timer2 预分频控制器(T2PSC)

参数名	比特位	属性	复位值	描述
T2PSC	1-0	W/R	0	Timer2 时钟预分频系数： 00: 1 分频 01: 8 分频 10: 64 分频 11: 256 分频
Reserved	7-2	R	0	保留

11 高级定时器 (PWM)

11.1 概述

JMT1801ED 高级定时器主要包含一个 16 位的向上/向下自动重载计数器、八个 16 位的比较/捕获寄存器、一个可编程的时钟预分频器和多个控制寄存器。

本定时器主要用途有：

- 产生多种输出波形（输出比较波形、PWM 波形、嵌入死区的互补 PWM 波形等）
- 测量输入信号的脉冲宽度（输入捕获功能）
- 作为通用定时器使用
- 作为霍尔传感器接口
- 作为正交增量编码器接口

本定时器可以产生 3 对互补的 PWM 输出加 1 路独立的 PWM 输出。

通过控制定时器的时钟分频（两种方式实现：控制 CRM 模块中 PWM 时钟分频器和控制定时器自身的时钟分频器），本定时器可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

11.2 特性

本定时器的特性：

- 16 向上、向下和向上/向下自动装载计数器。
- 16 位可编程分频器，计数器时钟频率的分频系数为 1~65536 之间的任意值。
- 4 个独立通道，每个通道支持以下功能：
 - 输入捕获功能
 - 输出比较功能
 - PWM 波形生成（边沿或中央对齐模式）
 - 单脉冲模式输出
 - 死区时间可编程的互补输出
- 在互补模式下可以输出 3 对互补的 PWM 波形和 1 路独立的 PWM 波形。
- 可以在指定数目的计数器周期之后更新定时器中的预装载寄存器。
- 相位偏移功能：可以选择 CC0~CC3 中的任一通道，使其输出波形比其他通道的输出延迟一定的时间，延迟时间可由软件编程控制。
- 刹车功能：外部刹车信号可以将定时器输出信号置于复位状态或者一个已知状态，也可以通过软件配置刹车事件实现刹车功能。
- 如下事件发生时产生中断

- 更新：计数器向上溢出、向下溢出，计数器初始化（通过软件或内部/外部触发）
- 触发事件（计数器启动，停止，初始化或由内部/外部触发计数）产生触发中断
- 输入捕获，当捕获数据时产生捕获中断
- 输出比较，当计数器与输出比较寄存器匹配时产生比较中断
- 刹车信号输入
- 软件配置 COM 事件，更新通道输出控制寄存器。

- 正交增量编码器接口功能
- 霍尔传感器接口功能

11.3 框图

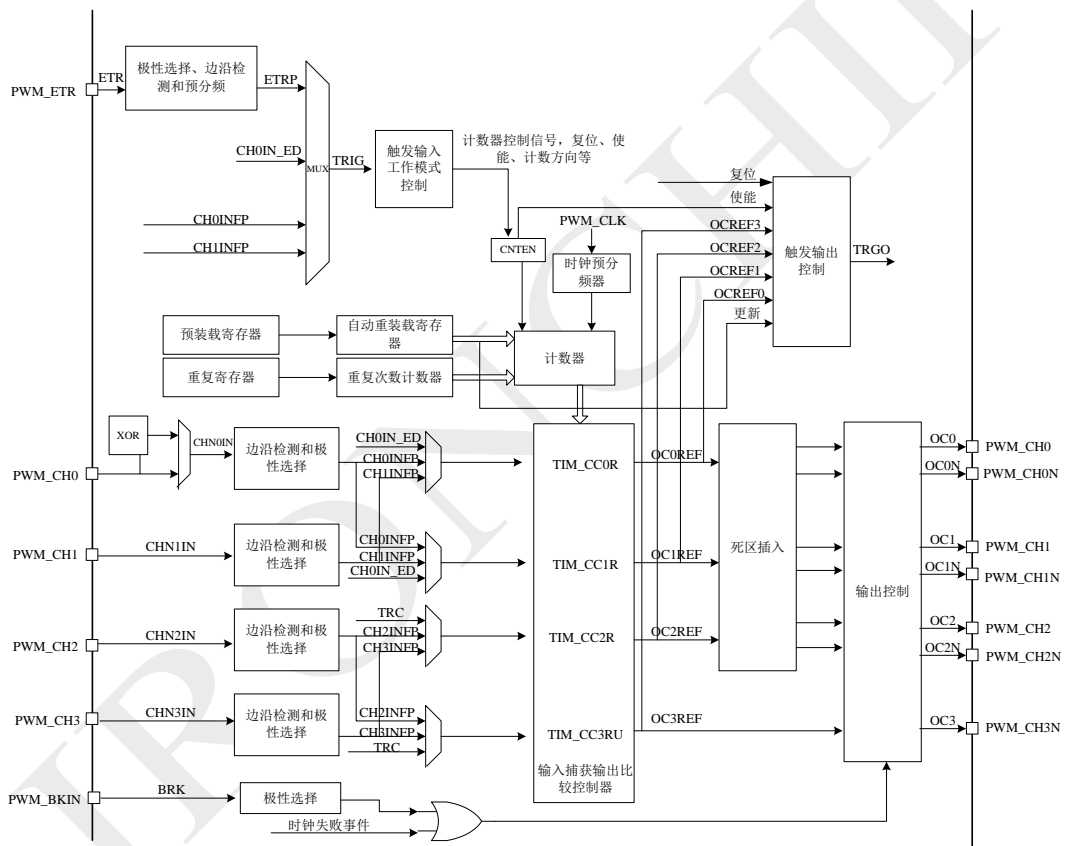


Figure 69 JMT1801ED 高级定时器（TIM/PWM）框图

11.4 输入输出引脚配置

有些引脚可以复用为高级定时器的输入输出功能，详细描述见 18.5 节。

当高级定时器配置为输入捕获功能时（寄存器 TIM_CCxMR⁴的 CCxMS 位配置为 01/10/11），外部信号通过 PWM_CHx 或 PWM_ETR 通道输入，刹车信号通过 PWM_BKIN 通道输入；当高级定时器配置为输出比较功能时（寄存器 TIM_CCxMR 的 CCxMS 位配置为

⁴ CCxMR 中的 x 比较通道标号，x=0, 1, 2, 3。

00), 高级定时器的输出信号通过 PWM_CHx 和 PWM_CHxN 输出。

当高级定时器输出互补 PWM 波形时, PWM_CHx 和 PWM_CHxN 是通道 CCx 的一对互补 PWM 输出。

11.5 功能描述

高级定时器的结构主要分为五部分: 计数器部分、从模式控制器、主模式控制器、四个通道的输入捕获控制器和四个通道的输出比较控制器。下面分别介绍高级定时器各个部分的功能和各种工作模式的控制方法。

11.5.1 计数器

11.5.1.1 计数器的计数周期

高级定时器 TIM 包含一个 16 位的计数器, 计数器的周期值保存在自动重装载寄存器中, 自动重装载寄存器是预先装载的 (TIM_ARR 是其预装载寄存器), 根据 TIM_CONR2 寄存器中的自动装载使能位 (ARPLE) 的设置, 预装载寄存器的值被立即或在更新事件时传送到影子寄存器 (软件不可见)。写或读自动重装载寄存器将访问预装载寄存器 (TIM_ARR)。

11.5.1.2 计数器的使能控制

计数器使能受到寄存器 TIM_CNTEN 的 CNTEN 位和从模式控制器控制, 具体控制方式如下:

- 当寄存器 TIM_TGICR0 的 SMS 位配置为 000 时, 高级定时器关闭从模式, 此时计数器使能只受 TIM_CNTEN 的 CNTEN 位控制, 当 CNTEN 配置为 1 时, 计数器计数, 当 CNTEN 配置为 0 时, 计数器停止计数。
- 当寄存器 TIM_TGICR0 的 SMS 位配置为 110 时, 高级定时器工作在触发模式下, 此时将寄存器 TIM_CONR2 的 CNTEN 位配置为 0, 计数器在触发输入的有效沿启动 (但不复位)。
- 当寄存器 TIM_TGICR0 的 SMS 位配置为其他值时, 计数器使能同时受 TIM_CNTEN 的 CNTEN 位和触发信号的控制, 此时需要将寄存器 CNTEN 位配置为 1, 计数器在输入触发信号的控制下工作。

寄存器 TIM_CNTEN 的 CNTEN 位置位后意味着计数器即将开始计数, 高级定时器也即将进入工作状态, 此时其他寄存器应该完成配置, 所以建议 TIM_CNTEN 的 CNTEN 位最后配置。

11.5.1.3 计数器预分频控制器

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。通过配置寄存器 TIM_PSCL 和 TIM_PSCH 可以改变分频系数, 此寄存器带有缓存器, 所以可以在工作时配置, 新的预分频器的参数在下一次更新事件 (UEV) 到来时被采用。Figure 70 为预分频器

的参数由 1 分频变为 4 分频时，计数器的时序图。

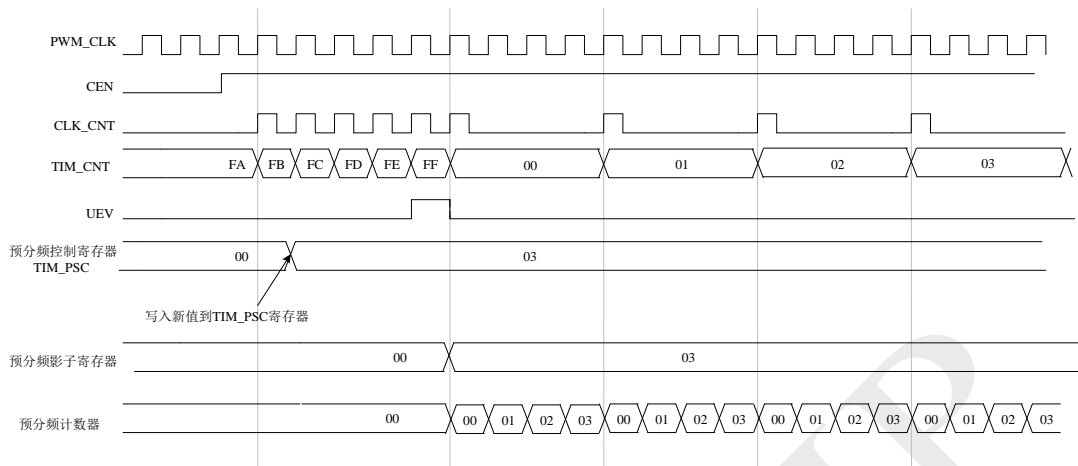


Figure 70 预分频器的参数由 1 分频变为 4 分频时，计数器的时序图

11.5.1.4 计数器计数模式控制

计数器有三种计数模式：向上计数模式、向下计数模式和中央对齐计数模式（向上向下交替计数），配置 TIM_CONR2 的 CNTMC 位可以设置计数器的计数模式：

当 CNTMC 配置为 00 时，计数器工作在边沿计数模式下，此时可以通过配置 TIM_CONR2 寄存器的 CNTDIR 位设置计数器的计数方向。

当 CNTMC 配置为 01/10/11 时，计数器工作在中央对齐模式下，此时不可以配置 TIM_CONR2 寄存器的 CNTDIR 位，计数器的方向由硬件自动设置。

当寄存器 TIM_TGICR0 的 SMS 位配置为 001/010/011 时，即高级定时器工作在增量编码器接口模式下，TIM_CONR2 的 CNTDIR 位也是不可写的，此时计数器的计数方向由硬件根据触发信号确定。

11.5.1.4.1 向上计数模式

当寄存器 TIM_CONR2 的 CNTMC 位配置为 00 且 CNTDIR 位配置为 0 时，计数器向上计数，计数器的值从 0 开始累加，一直累加到计数器的重载值（TIM_ARR 中的值）。一旦计数器达到了重载值，寄存器将从 0 开始重新计数。若设置了重复周期计数的值(TIM_RCR)，在计数器向上溢出次数达到设置的重复周期次数时才产生更新事件；否则每当计数器溢出时会产生更新事件。在此种计数模式下，设置 TIM_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，计数器初始化为 0；

更新事件产生的规则：

- 设置寄存器 TIM_CONR3 中的 UPOUDIS 位为 1，可以禁止上溢更新事件的产生，若更新事件被禁止，预分频寄存器的数值不变，同时计数器按预分频的频率进行计数。禁止更新事件产生的目的为：避免在向预装载寄存器中写入新值时更新影子寄存器。

- 设置 TIM_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，可以产生一个更新事件。
- 如果寄存器 TIM_CONR2 中的 URC 位被设置为 1，设置 UG 位为 1 将产生一个更新事件 UEV，但硬件不更新中断标志位 SOFTUIF（即不产生软更新中断）。这是为了避免在捕获模式下初始化计数器时产生更新中断。

当发生更新事件时，硬件做如下处理：

- 上溢更新标志位 UOVERIF 自动置 1，若上溢更新中断使能（TIM0_CONR3 寄存器的 UPOIE 位为 1）有效，则产生上溢更新中断。
- 若通过软件配置 UG 位为 1 产生更新事件，硬件根据 URC 位的配置更新中断标志位 SOFTUIF，若软更新中断使能（TIM0_CONR3 寄存器的 SOFTUPIE 位）有效，则产生软件更新中断。
- 自动重载寄存器被重新置入预装载寄存器的值。
- 重复周期计数器被重新加载为 TIM_RCR 寄存器的内容。
- 预分频器的影子寄存器被置入预装载寄存器（TIM_PSCL 和 TIM_PSCH 寄存器）的值。

Figure 71 和 Figure 72 为向上计数模式下计数器的时序，计数器时钟的预分频系数为 1（2 分频）。

Figure 71 中自动装载寄存器的预装载功能使能，新配置的计数器周期值在更新事件时传递到影子寄存器中，下一周期生效。Figure 72 的自动装载寄存器的预装载功能关闭，新配置的计数器周期值立即传送到影子寄存器中，在本周期即生效。

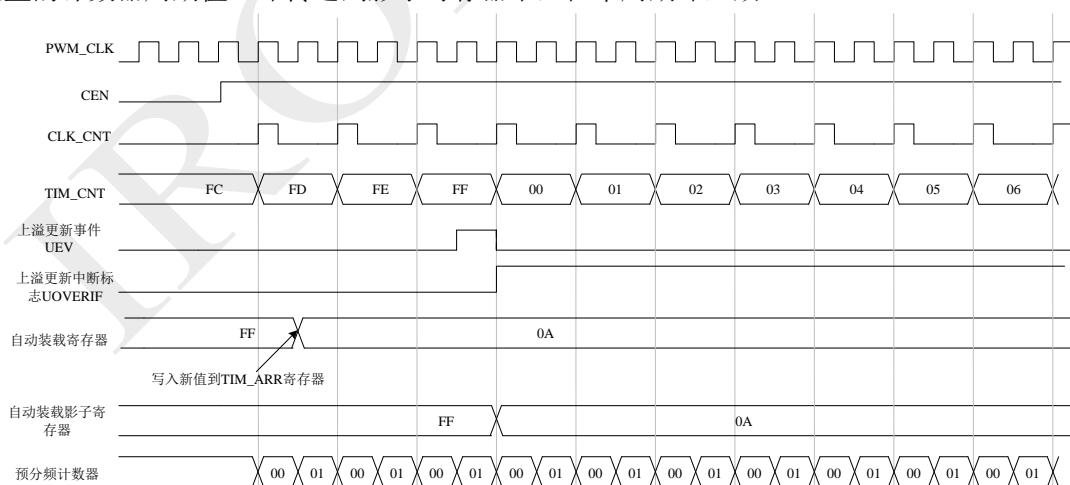


Figure 71 向上计数模式下，自动装载寄存器的预装载功能使能时计数器的时序图

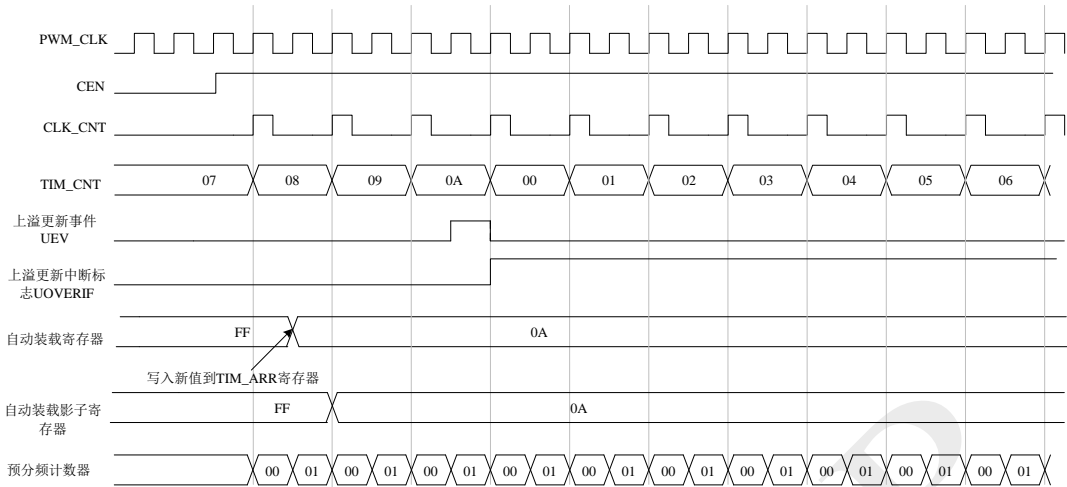


Figure 72 向上计数模式下，当自动装载寄存器的预装载功能不使能时计数器的时序图

11.5.1.4.2 向下计数模式

当寄存器 TIM_CONR2 的 CNTMC 位配置为 00 且 CNTDIR 位配置为 1 时，计数器向下计数，计数器的值从自动装入的值 TIM_ARR 开始递减到 0，然后从自动装入的值重新开始计数并且产生一个计数器向下溢出事件。若设置了重复周期计数的值，在计数器向下溢出次数达到设置的重复周期次数时才产生更新事件；否则每当计数器向下溢出时会产生更新事件。在此种计数模式下，当设置 TIM_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，计数器初始化为周期值 TIM_ARR。

更新事件产生的规则：

- 设置寄存器 TIM_CONR3 中的 DOWNOUDIS 位为 1，可以禁止下溢更新事件的产生。若更新事件被禁止，预分频寄存器的数值不变，同时计数器按预分频的频率进行计数。禁止更新事件产生的目的为：避免在向预装载寄存器中写入新值时更新影子寄存器。
- 设置 TIM_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，可以产生一个更新事件。
- 如果 TIM_CONR3 中的 URC 位被设置为 1，设置 UG 位为 1 将产生一个更新事件 UEV，但硬件不更新中断标志 SOFTUIF（即软更新中断标志）。这是为了避免在捕获模式下初始化计数器时产生更新中断。

当发生更新事件时，硬件做如下处理：

- 下溢更新标志位 DOVERRIDE 自动置 1，若下溢更新中断使能（TIM0_CONR3 寄存器的 DOWNOIE 位）有效，则产生下溢更新中断。
- 若通过软件配置 UG 位为 1 产生更新事件，硬件根据 URC 位的配置更新中断标志位 SOFTUIF，若软更新中断使能（TIM0_CONR3 寄存器的 SOFTUPIE 位）有效，则产生软件更新中断。

- 自动重载寄存器被重新置入预装载寄存器的值。
- 重复周期计数器被重新加载为 TIM_RCR 寄存器的内容。
- 预分频器的影子寄存器被置入预装载寄存器 (TIM_PSCL 和 TIM_PSCH 寄存器) 的值。

值得注意的是，在向下计数模式下，自动重载寄存器的值在计数器重载入之前更新，因此下一个周期将是预期的值。Figure 73 为向下计数模式下计数器的时序图。

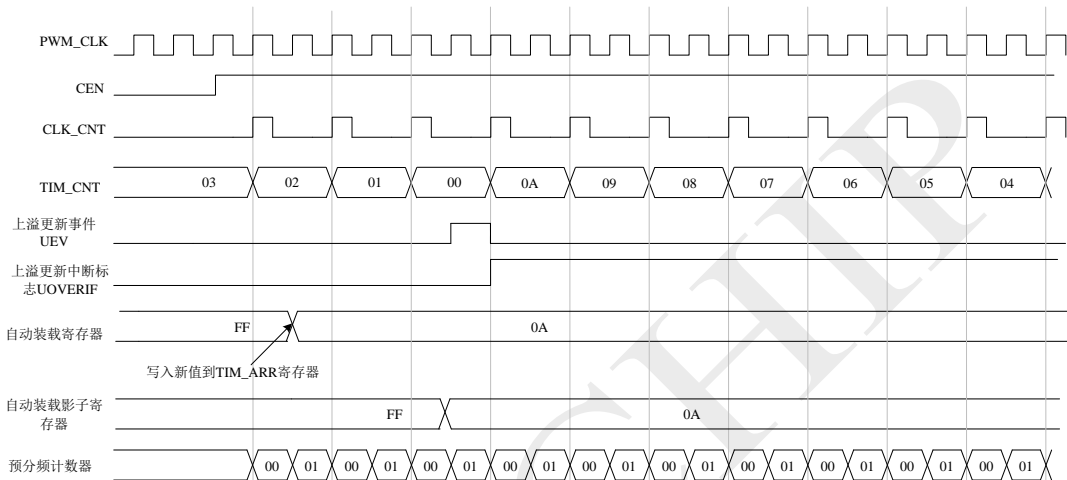


Figure 73 向上计数时计数器的时序图

11.5.1.4.3 中央对齐计数模式

寄存器 TIM_CONR2 的 CNTMC 位配置为 00/01/11 时，计数器的计数模式为中央对齐模式，中央对齐模式下，寄存器 TIM_CONR2 的 CNTDIR 位是不可写的，它由硬件更新并指示当前的计数方向。

在这种模式下，计数器从 0 开始计数到自动加载值 (TIM_ARR 寄存器) 减 1，产生一个计数器上溢事件，然后向下计数到 1 并且产生一个计数器下溢事件，然后再从 0 开始计数。在此种计数模式下，当设置 TIM_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，计数器初始化为 0；

每一次计数上溢和每一次计数下溢都可以产生更新事件。

更新事件产生的规则：

- 设置寄存器 TIM_CONR3 中的 UPOUDIS 位为 1，可以禁止上溢更新事件的产生，设置寄存器 TIM_CONR3 中的 DOWNOUDIS 位为 1，可以禁止下溢更新事件的产生。若更新事件被禁止，预分频寄存器的数值不变，同时计数器按预分频的频率进行计数。禁止更新事件产生的目的为：避免在向预装载寄存器中写入新值时更新影子寄存器。
- 设置 TIM_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时可以产生一个软件更新事件。

- 如果 TIM_CONR3 中的 URC 位被设置为 1，设置 UG 位为 1 将产生一个更新事件 UEV，但硬件不更新 SOFTUIF 标志（即不产生软更新中断）。这是为了避免在捕获模式下初始化计数器时产生更新。

当发生更新事件时，硬件做如下处理：

- 若更新事件是上溢更新事件，上溢更新标志位 UOVERIF 自动置位，若上溢更新中断使能（TIM_CONR3 寄存器的 UPOIE 位）有效，则产生上溢更新中断。若更新事件是下溢更新事件，下溢更新标志位 DOVERIF 自动置位，若设置了下溢更新中断使能（TIM_CONR3 寄存器的 DOWNOIE 位）有效，则产生下溢更新中断。
- 若配置 UG 位为 1 产生更新事件时，硬件根据 URC 位更新中断标志位 SOFTUIF，若软更新中断使能（TIM_CONR3 寄存器的 SOFTUPIE 位）有效，则产生软件更新中断。
- 自动重载寄存器被重新置入预装载寄存器的值。
- 重复周期计数器被重新加载为 TIM_RCR 寄存器的值。
- 预分频器的影子寄存器被置入预装载寄存器的值（TIM_PSCL 和 TIM_PSCH 寄存器的内容）。

值得注意的是，如果因为计数器下溢而产生更新，自动重载寄存器的值在计数器重载入之前更新，因此一下个周期将是预期的值。Figure 74 和 Figure 75 为中央计数模式下计数器的时序。Figure 74 的自动装载寄存器的预装载功能使能，新配置的计数器周期值在更新事件时传递到影子寄存器中，下一周期生效。Figure 75 的自动装载寄存器的预装载功能关闭，新配置的计数器周期值立即传送到影子寄存器中，在本周期即生效。

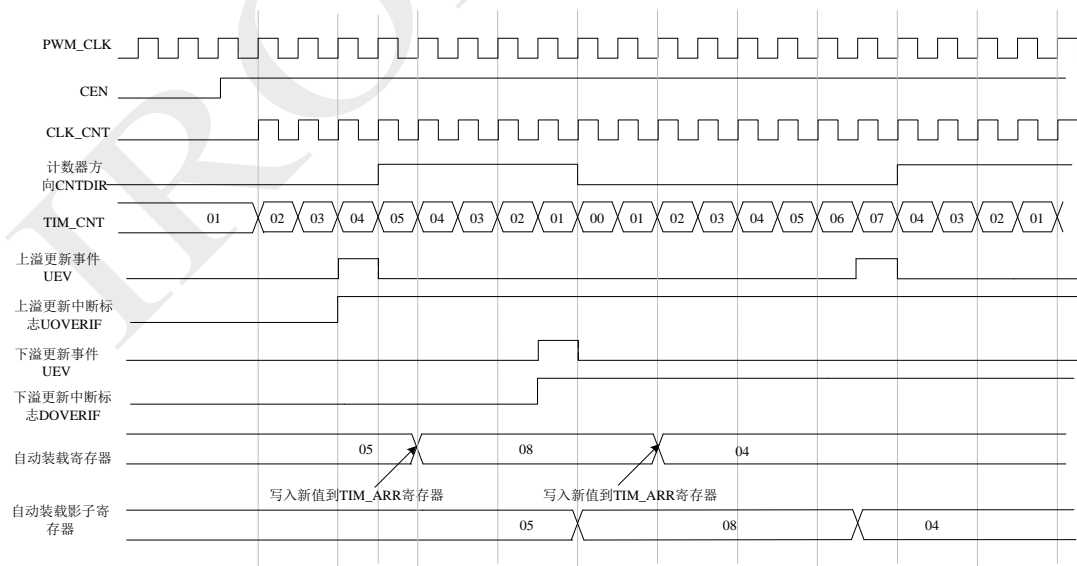


Figure 74 中央计数模式下，自动装载寄存器的预装载功能使能时计数器的时序图

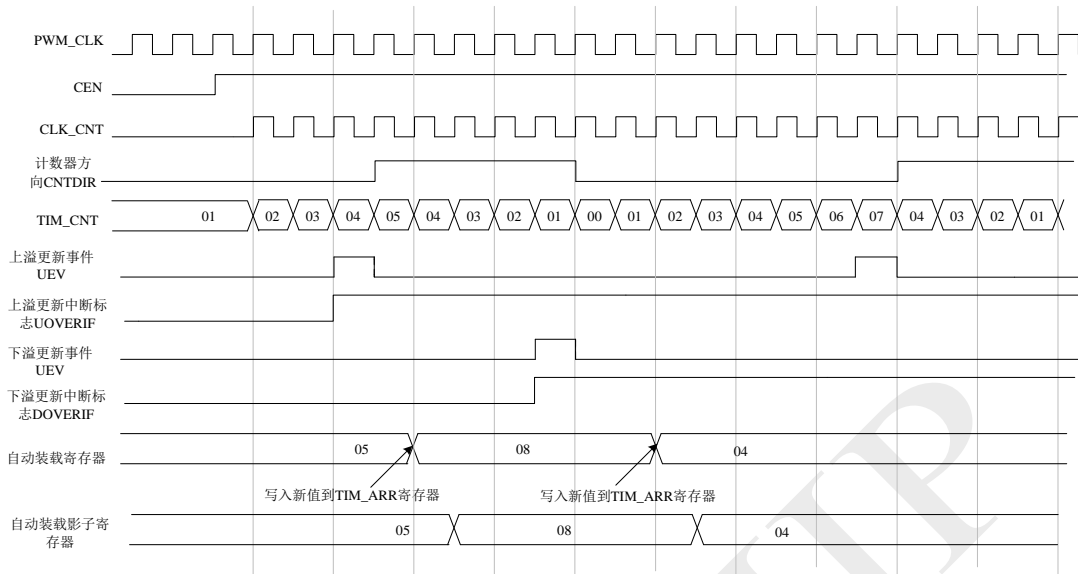


Figure 75 中央计数模式下，自动装载寄存器的预装载功能不使能时计数器的时序图

11.5.1.5 计数器的重复周期计数控制器

借助重复周期计数功能，可以在计数器连续上溢/下溢 N 次后，才会产生计数器更新事件，更新各寄存器的内容。

周期计数器递减的条件有：

- 向上计数模式下的每次计数器上溢；
- 向下计数模式下的每次计数器下溢；
- 中央对齐模式下的每次计数器上溢/下溢。

周期计数器是自动加载的向下计数器，重复次数由寄存器 TIM_RCR 的值定义。当更新事件由软件产生或者通过硬件的从模式控制器产生时，无论周期计数器的值是多少，立即发生更新事件，并且寄存器 TIM_RCR 的内容被重新载入到周期计数器。

Figure 76 为向上计数模式下周期计数控制器对更新事件的控制，图中箭头表示产生更新事件，当寄存器 TIM_CRC 配置为 0 时，每当计数器溢出时产生更新事件，当寄存器 TIM_CRC 配置为 1 时，计数器连续 2 次溢出时产生更新事件，当 TIM_CRC 配置为 N 时，计数器连续 N+1 次溢出时产生更新事件。若软件配置更新事件时，立即发生更新事件。

Figure 77 为向下计数模式下周期计数控制器对更新事件的控制。

Figure 78 为中央对齐计数模式下周期计数控制器对更新事件的控制。

Figure 79 为中央对齐模式下，当重复周期次数为 3 时，更新事件产生的时序图。

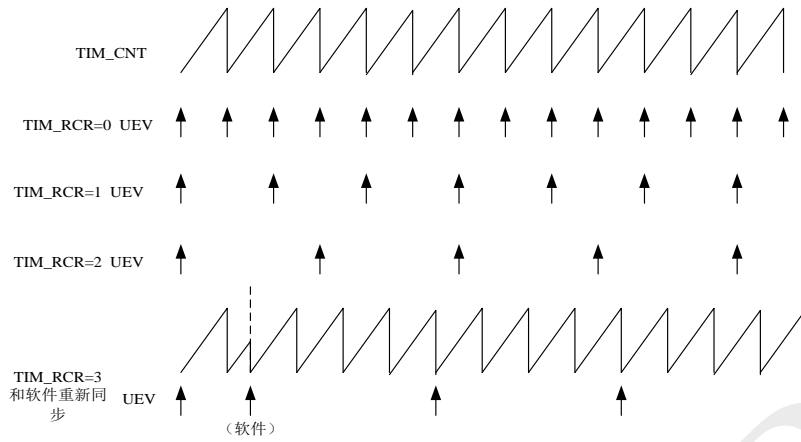


Figure 76 向上计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

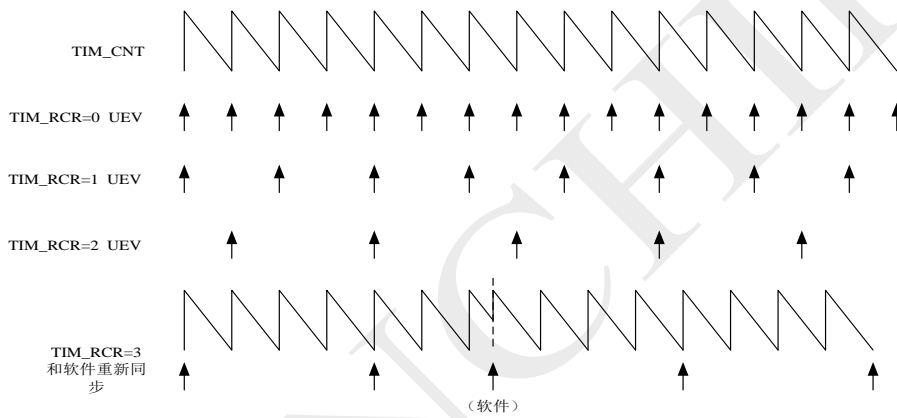


Figure 77 向下计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

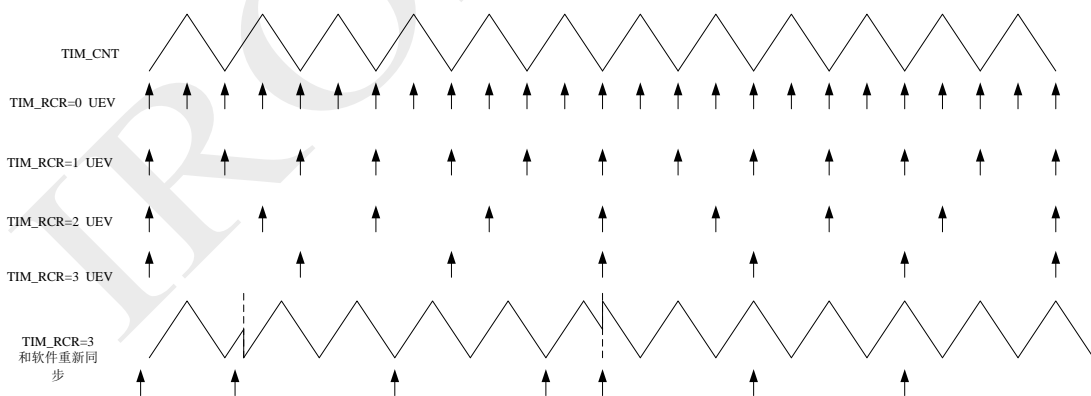


Figure 78 中央对齐计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

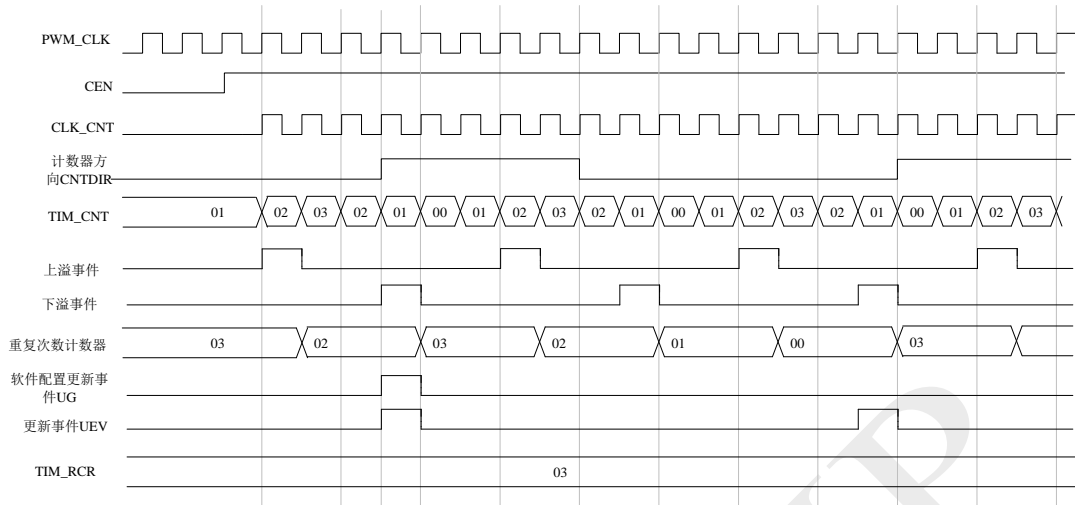


Figure 79 中央对齐模式下，更新事件的产生时序图

11.5.1.6 计数器时钟选择

计数器的时钟可由下列时钟源提供：

- 内部时钟。
- 外部输入脚 PWM_CHx (x=1、2) 提供的输入信号（外部时钟源模式 1）。
- 外部触发输入 PWM_ETR 提供的输入信号（外部时钟模式 2）。

11.5.1.6.1 内部时钟

内部时钟由 CRM 模块提供，最高频率为 73.728MHz，可以通过配置 CRM 模块中的 PWM_CLK_DIV 寄存器对时钟进行 1/2/4/8 分频，预分频器可以对输入的 pwm_clk 再进行 1~65536 分频，可以通过配置寄存器 TIM_PSCL 和 TIM_PSCH 实现。

将 TIM_CONR2 寄存器的 SMS 位配置为 000，则从模式控制器被关闭，预分频器的驱动时钟由内部时钟 pwm_clk 提供。计数器的使能、方向和软更新事件由软件配置确定。当 TIM_CNTEN 寄存器的 CNTEN 位配置为 1 时，计数器开始计数。

11.5.1.6.2 外部时钟源模式 1

当寄存器 TIM_TGICR0 的 SMS 位配置为 111 时，外部时钟源模式 1 被选中，计数器可在选定的输入脚 PWM_CHx 输入信号的每个上升沿或下降沿计数，通过配置 TIM_TGICR0 寄存器的 TRGS 位可以选择输入管脚 PWM_CHx 的信号。Figure 80 为选择输入脚 PWM_CH0 的信号作为外部时钟的例子，具体配置步骤如下：

- 外部信号输入后，通过配置寄存器 TIM_CC0MR 中的 CC0MS 位可以从 CC0 的输入信号 CH0IN 和通道 CC1 的输入信号 CH1IN 中选择一个作为通道 CC0 的输入信号，本例中选择 CH0INF 作为 CC0 的输入信号，本例中选择 CH0IN。
- 将信号 CH0IN 送入边沿检测器做边沿检测，得到上升沿信号 CH0INF_R、下降沿信号 CH0INF_D 和边沿信号 CH0INF_ED（包括上升沿信号和下降沿信号）。

- 通过配置寄存器 TIM_CCPS 寄存器的 CC0P 位可以选择输入信号的有效沿，当 CC0P 配置为 0 时，选择信号的上升沿 CH0INF_R 作为有效边沿；当 CC0P 配置为 1 时，选择信号的下降沿 CH0INF_D 作为有效边沿，得到边沿选择后的输入信号 CH0INFP，本例中选择上升沿为有效边沿。
- 配置寄存器 TIM_TGICR0 的 TRGS 位为 101，选择 CH0INFP 作为从模式控制器的输入触发信号。
- 配置寄存器 TIM_TGICR0 的 SMS 位为 111，使从模式控制器工作在外部时钟模式 1 下。
- 配置寄存器 TIM_CNTEN 的 CNTEN 位为 1，启动计数器。

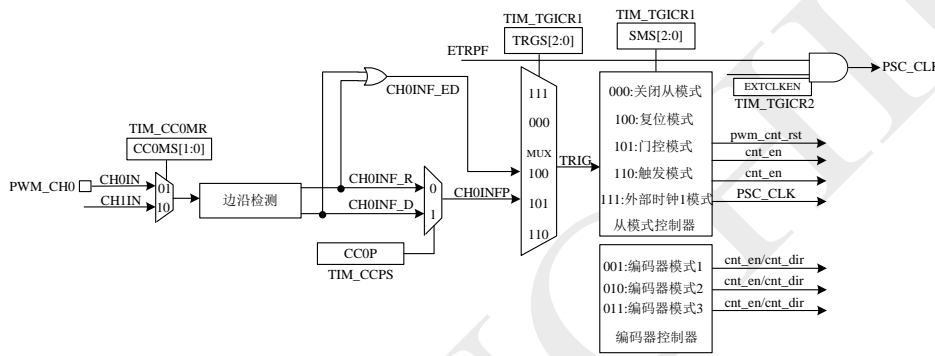


Figure 80 输入脚 PWM_CH0 的信号作为外部时钟的连接例子

Figure 81 为输入信号 CH0IN 的上升沿触发计数器计数的时序图，当输入信号 CH0IN 出现一次上升沿时，计数器记一次。输入信号 CH0IN 的上升沿和实际的计数器时钟之间的延时取决于 CH0IN 输入通道的重同步电路。

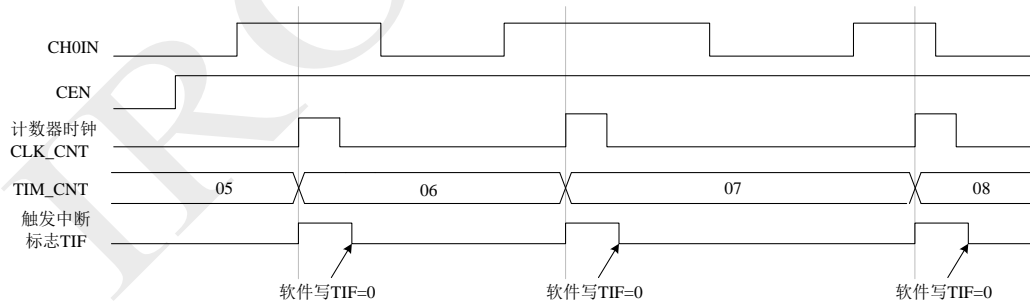


Figure 81 输入脚 PWM_CH0 的信号作为外部时钟的时序图

11.5.1.6.3 外部时钟源模式 2

将寄存器 TIM_TGICR1 的 EXTCLKEN 位配置为 1，即采用此种模式，计数器可以由外部管脚 PWM_ETR 的每个上升沿或下降沿驱动计数。Figure 82 为外部触发信号 PWM_ETR 作为外部时钟源的例子，具体配置为：

- 配置寄存器 TIM_TGICR1 的 EXTP 位，选择输入信号的极性。若 EXTP 位配置为 0，

不对输入的信号反相，输入信号的上升沿或高电平有效；若 EXTP 位配置为 1，则对输入的信号反相，输入信号的下降沿或低电平有效，极性选择后得到信号 PWM_ETRP。

- 配置寄存器 TIM_TGICR1 的 EXTDIV 位，对 PWM_ETRP 进行分频，得到信号 PWM_ETRPDIV。
- 配置寄存器 TIM_TGICR1 的 EXTCLKEN 位为 1，将计数器时钟配置为外部时钟源模式 2。

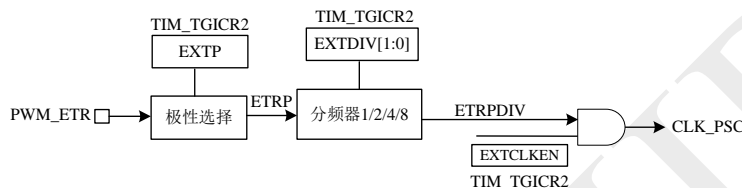


Figure 82 外部触发信号作为外部时钟源

值得注意的是，将计数器的时钟配置为外部时钟模式 2 所达到的效果与将时钟配置为外部时钟模式 1 同时把外部输入信号 PWM_ETR 作为信号源效果是一样的，都是在 PWM_ETR 的有效边沿触发计数器计数。

外部时钟模式 2 可以与各种从模式配合使用，例如，当时钟配置为外部时钟模式 2 时，从模式可以配置为门控模式。外部时钟模式 1 则不可以和从模式配合使用。Figure 83 为外部时钟模式 2 下计数器的时序图，外部信号 PWM_ETR 出现一次上升沿，则计数器计数一次。PWM_ETR 信号的上升沿和实际的时钟上升沿的延时取决于 PWM_ETR 输入通道的重同步电路。

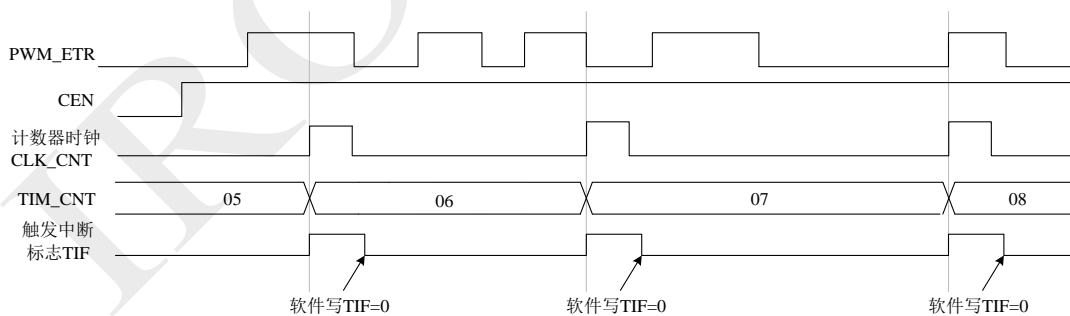


Figure 83 外部时钟模式 2 下计数器的时序图

11.5.1.7 寄存器更新及安全级别设置

11.5.1.7.1 16 位寄存器的读写

本定时器中有些寄存器是 16 位寄存器，而芯片的数据总线是 8 位，为了便于寄存器读写，将 16 位寄存器拆分为两个 8 位寄存器，即高字节寄存器 MS 和低字节寄存器 LS，例如

通道 CCx 的捕获比较寄存器 TIM_CCxR 分为高字节寄存器 TIM_CCxRH 和低字节寄存器 TIM_CCxRL。在本定时器中读写 16 位寄存器应遵循以下规则：先读写高字节寄存器 MS 再读写低字节寄存器 LS。

11.5.1.7.2 寄存器的预装载模式和立即更新模式

定时器中有些寄存器有两种更新模式：预装载模式和立即更新模式。

立即更新模式指：当向寄存器写入新值时，新值立即生效。

预装载模式指：具有预装载功能的寄存器有两个寄存器--预装载寄存器和影子寄存器，TIM 使用的是影子寄存器值，而软件只能读写预装载寄存器。

若开启了预装载功能，当软件配置寄存器时，只能将新值写入预装载寄存器，当发生寄存器的更新事件时，硬件才将预装载寄存器中的值装载到影子寄存器，此时新配置的值才会生效。

TIM 中所有具有预装载功能的寄存器有：TIM_ARR、TIM_PSC（无立即更新模式）、TIM_CCxR、TIM_CCENR、TIM_CCPS 和寄存器 TIM_CCxMR 的 OCxMS 位，根据预装载模式的使能信号和更新信号的不同，可以将这些寄存器分为三类：

- 由寄存器 TIM_CONR2 的 ARPLE 位控制预装载模式是否使能，由更新事件 UEV 更新寄存器，这些寄存器有：TIM_ARR、TIM_PSC（无立即更新模式）。
- 由 TIM_CCxMR 寄存器的 OCxPEN 位控制预装载模式是否使能，由更新事件 UEV 更新寄存器，这些寄存器有：TIM_CCxRL、TIM_CCxRH。
- 由 TIM_CONR0 寄存器的 CCPE 位控制预装载模式是否使能，在预装载模式下由 COM 事件或从模式的触发输入更新（由 TIM_CONR0 寄存器的 CCUS 位决定）寄存器，这些寄存器有：TIM_CCENR、TIM_CCPS 和 TIM_CCxMR 寄存器的 OCxMS 位。

具有预装载功能寄存器的预装载功能可以关闭（TIM_PSC 除外），当预装载功能关闭时，具有预装载功能的寄存器使用立即更新模式。

11.5.1.7.3 寄存器的安全级别控制

为了防止寄存器被误配，高级定时器为寄存器设置了配置保护，可以通过配置寄存器 TIM_BRKC 的 LOCK 位设置对不同寄存器的保护，复位后 LOCK 位为 00，关闭对寄存器的保护，所有寄存器均可配置，LOCK 位只能配置一次非 0 值，一旦配置为非 0 值后不可以再更改直到高级定时器模块被复位。

寄存器 TIM_BRKC.LOCK 对寄存器的保护分为三个级别：

- 当寄存器 TIM_BRKC.LOCK 配置为 01 时，将寄存器的安全级别配置为锁定级别 1，不能写入 TIM_DTG 寄存器、TIM_BRKC 的 BRKE、BRKP、AOE 位、TIM_CONR2 寄存器。
- 当寄存器 TIM_BRKC.LOCK 配置为 10 时，将寄存器的安全级别配置为锁定级别 2，

不能写入锁定级别 1 中的所有寄存器，也不能写入 TIM_CCPS 寄存器，不能写入 TIM_BRKC 的 ROSS 和 IOSS 位。

- 当寄存器 TIM_BRKC.LOCK 配置为 11 时，将寄存器的安全级别配置为锁定级别 3，不能写入锁定级别 1 和锁定级别 2 中的所有寄存器，不能写入 TIM_CCxMR 的 OCxMS 和 OCxPEN 位。

11.5.2 输入捕获功能

高级定时器包含四个通道，可以分别对四个通道独立控制，每个通道可以配置为输入捕获模式或输出比较模式，下面以通道 CC0 为例介绍当通道配置为输入捕获功能时的使用方法。

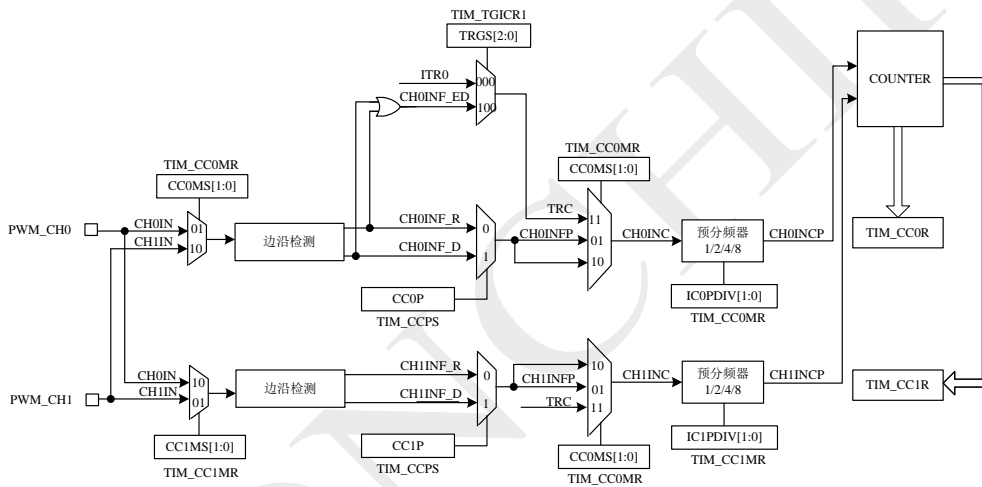


Figure 84 输入捕获通道

如 Figure 84 所示，当寄存器 TIM_CC0MR 的 CC0MS 位配置为 01/10/11 时，通道 CC0 配置为输入捕获功能，外部输入信号的有效边沿可以捕获计数器的值，并将捕获的值存入通道 CC0 的输入捕获寄存器 TIM_CC0R 中，具体控制方法如下：

- 配置 GPIO 模块中的寄存器，将相应的管脚配置为高级定时器的信号输入输出功能。
- 通过配置寄存器 TIM_CC0MR 中的 CC0MS 位可以选 CH0IN 或通道 CC1 的输入信号 CH1IN 作为通道 CC0 的输入信号。
- 将滤波后的信号送入边沿检测器做边沿检测，得到上升沿信号 CH0INF_R、下降沿信号 CH0INF_D 和边沿信号 CH0INF_ED（包括上升沿信号和下降沿信号）。
- 将边沿信号 CH0INF_ED 送入从模式控制器，通过配置寄存器 TIM_TGICR0 的 TRGI 位可以从 CH0INF_ED 和 ITR0 中选择一个信号作为触发捕获信号 TRC。通过配置寄存器 TIM_CCPS 的 CC0P 位可以选择输入信号的有效沿，当 CC0P 配置为 0 时，选择信号的上升沿 CH0INF_R 作为有效边沿；当 CC0P 配置为 1 时，选择信号的下降沿 CH0INF_D 作为有效边沿，得到边沿选择后的输入信号 CH0INF_P。

- 通过配置寄存器 TIM_CC0MR 中的 CC0MS 位可以从 TRC 信号和 CH0INFP 中选择一个作为通道 CC0 原始的捕获信号 CH0INC。
- 配置寄存器 TIM_CC0MR 的 ICPDIV 位,对原始的捕获信号 CH0INC 进行预分频,得到输入捕获信号 CH0INCP。
- 将输入捕获信号 CH0INCP 送入计数器部分进行数据捕获,捕获后将数据存入输入捕获寄存器 TIM_CC0RU。

以上为 CC0 通道的输入捕获模式的控制步骤,CC1、CC2 和 CC3 通道的控制步骤相同。在软件访问 16 位寄存器时,先访问高 8 位部分再访问低 8 位部分,按照此规则,在软件读取 TIM_CC0R 中的捕获值时,先读取寄存器 TIM_CC0RH 再读取 TIM_CC0RL, Figure 85 为软件读取捕获寄存器 TIM_CC0R 的过程。

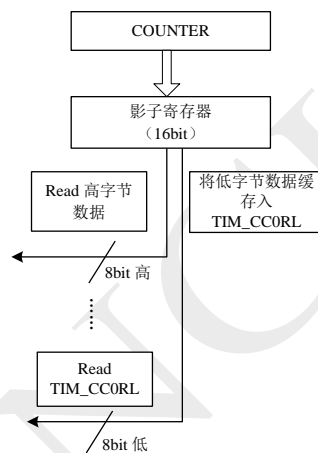


Figure 85 向软件读取捕获寄存器

为了防止在读取捕获值的高 8 位部分时再次发生捕获,软件一旦开始读取高 8 位部分数据时,硬件将低 8 位数据从影子寄存器缓存入预装载寄存器 TIM_CC0RL。

当产生捕获信号 CH0INCP 时,寄存器 TIM_ISR0 中捕获中断标志位 CC0IF 置位,若捕获/比较中断使能打开,则产生捕获中断。读取捕获值后捕获中断标志位 CC0IF 自动清除,也可以通过软件配 0 将捕获中断标志位 CC0IF 清除。

若捕获的数据没有及时读取,此时又有捕获信号来到,则寄存器 TIM_ISR1 中的过捕获标志位 CC0OIF 置位,新捕获的数据将原来的数据覆盖。需要软件配置 CC0OIF 位为 0 清除过捕获标志 CC0OIF。

11.5.3 从模式控制

11.5.3.1 从模式控制器的信号源

从模式是指高级定时器受外部触发信号驱动的工作模式, Figure 86 所示为从模式控制器,外部触发信号的来源有:

- 外部信号输入通道 PWM_ETR，触发信号产生过程为：
 - 配置寄存器 TIM_TGICR1 的 EXTP 位，选择输入信号的极性，若 EXTP 位配置为 0，不对输入的信号反相，输入信号的上升沿或高电平有效；若 EXTP 位配置为 1，则对输入的信号反相，输入信号的下降沿或低电平有效，极性选择后得到信号 EXTRP。
 - 配置寄存器 TIM_TGICR1 的 EXTDIV 位，对 EXTRP 进行分频，得到信号 PWM_ETRPDIV。
 - PWM_ETRPDIV 信号有两个用途：i) 作为外部触发信号，进入从模式控制器驱动高级定时器工作。ii) 当寄存器 TIM_TGICR1 的 EXTCLKEN 位配置为 1 时，外部时钟模式 2 打开，PWM_ETRPF 也可作为计数器预分频器的时钟 PSC_CLK。
- 通道 CC0 的双边沿信号 CH0IN_ED，产生过程见 11.5.2 部分说明。
- 通道 CC0 的单边沿信号 CH0INFP，产生过程见 11.5.2 部分说明，此信号一方面送入从模式控制器驱动高级定时器工作，另一方面作为编码器控制器的输入，在编码器接口模式下控制计数器计数。
- 通道 CC1 的单边沿信号 CH1INFP，产生过程见 11.5.2 部分说明，此信号一方面送入从模式控制器驱动高级定时器工作，另一方面作为编码器控制器的输入，在编码器接口模式下控制计数器工作。

其中，TIM_TRGO 和 CH0IN_ED 信号还可以作为捕获信号。

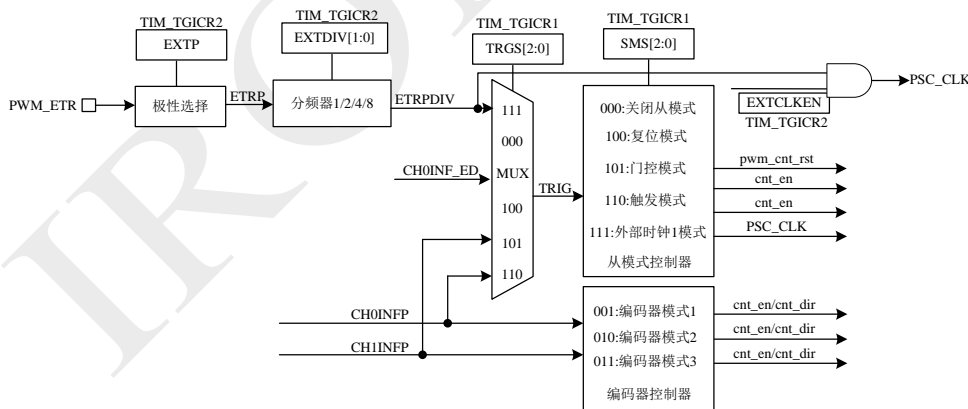


Figure 86 从模式控制器示意图

11.5.3.2 从模式控制

配置寄存器 TIM_TGICR0 的 TRGS 位，从 ETRPF、CH0INFP、CH0INFP 和 CH1INFP 中选择一个作为触发信号送入从模式控制器，在不同的控制模式下驱动高级定时器工作。对高级定时器的控制模式有：

- 关闭从模式：将寄存器 TIM_TGICR0 的 SMS 位配置为 000，关闭从模式，此时预

分频器的驱动时钟由内部时钟 `pwm_clk` 提供。计数器的使能、方向和软更新事件由软件配置确定。当 `TIM_CNTEN` 寄存器的 `CNTEN` 位配置为 1 时，计数器开始计数。

- **复位模式：**将寄存器 `TIM_TGICR0` 的 `SMS` 位配置为 100 时，从模式控制器工作在复位模式下，选中的触发信号 `TRGI` 的有效沿重新初始化计数器，并且产生一个更新事件来更新寄存器。若 `TIM_CONR2` 的 `URC` 位配置为 0，则寄存器 `TIM_ISR0` 的中断标志 `SOFTUIF` 置位，若软更新中断使能打开，则产生一个软更新中断，此时，相当于配置了寄存器 `TIM_EGR` 的 `UG` 事件；若 `TIM_CONR2` 的 `URC` 位配置为 1，则在触发信号 `TRGI` 的有效沿仅初始化计数器，并且产生一个更新事件来更新寄存器，而不产生中断和中断标志。
- **门控模式：**将寄存器 `TIM_TGICR0` 的 `SMS` 位配置为 101 时，从模式控制器工作在门控模式下，高级定时器的计数器使能受触发信号 `TRGI` 的控制，当 `TRGI` 为有效电平时，计数器开始计数，当 `TRGI` 为无效电平时，计数器停止计数（但不复位）。此时 `TIM_CNTEN` 的 `CNTEN` 位仍需要配置为 1，否则，即使 `TRGI` 为有效电平，计数器也停止计数。
- **触发模式：**将寄存器 `TIM_TGICR0` 的 `SMS` 位配置为 110 时，从模式控制器工作在触发模式下，选中的触发信号 `TRGI` 的有效沿启动计数器计数（但不复位）。此时 `TIM_CNTEN` 的 `CNTEN` 位不需要配置为 1，触发信号 `TRGI` 的有效沿会将 `CNTEN` 位置 1。计数器启动后配置 `TIM_CNTEN` 的 `CNTEN` 位为 0，可以停止计数器计数。
- **外部时钟源模式 1：**将寄存器 `TIM_TGICR0` 的 `SMS` 位配置为 111 时，选择触发信号 `TRGI` 的有效沿驱动计数器计数。此时 `TIM_CNTEN` 的 `CNTEN` 位需要配置为 1。
- **外部时钟源模式 2：**配置寄存器 `TIM_TGICR1` 的 `EXTCLKEN` 位为 1，`PWM_ETR` 的有效沿驱动计数器计数。此时 `TIM_CNTEN` 的 `CNTEN` 位需要配置为 1。

有三点需要注意：

第一，在 `CH0IN_ED` 被选为触发信号 `TRGI` 时，不要使用门控模式，因为 `CH0IN_ED` 是一个脉冲信号，而门控模式时要检查触发输入的电平。

第二，从模式控制器配置为外部时钟源模式 1 且选 `PWM_ETR` 作为触发信号 `TRGI` 与直接将定时器配置为外部时钟源模式 2 效果是一样的，都是使用 `PWM_ETR` 的有效沿驱动计数器计数。

第三，外部时钟源模式 2 和各种从模式可以配合使用，当定时器配置为外部时钟源模式 2 时，从模式控制器也可以配置为各种从模式，此时从模式控制器的输入触发信号 `TRGI` 不要选择 `PWM_ETR`。

11.5.3.3 编码器接口模式

使用通道 CC0 和 CC1 作为正交增量编码器两路信号的接口，在通道 CC0 和 CC1 对信号做必要的处理后得到信号 CH0INFP 和 CH1INFP，再将 CH0INFP 和 CH1INFP 直接送入编码器控制器，在编码器控制器中根据两路信号驱动计数器计数。

编码器接口模式的配置步骤为：

- 根据实际需要配置寄存器 TIM_CCPS 的 CC0P 位，对输入的编码器信号进行极性选择，得到信号 CH0INFP。
- 根据实际需要配置寄存器 TIM_CCPS 的 CC1P 位，对输入的编码器信号进行极性选择，得到信号 CH1INFP。
- 配置寄存器 TIM_TGICR0 的 SMS 位：若将 SMS 位配置为 001，定时器工作在编码器模式 1 下，计数器以 CH0INFP 作为相对信号由 CH1INFP 的边沿驱动计数；若将 SMS 位配置为 010，定时器工作在编码器模式 2 下，计数器以 CH1INFP 作为相对信号由 CH0INFP 的边沿驱动计数；若将 SMS 位配置为 011，定时器工作在编码器模式 3 下，计数器以 CH1INFP 和 CH0INFP 作为相对信号由 CH0INFP 和 CH1INFP 的边沿驱动计数。计数器的计数方向也在编码器控制器中根据一定的规则产生。
- 配置寄存器 TIM_CCENR 的 CC0EN 和 CC1EN 位为 1，使能通道 CC0 和通道 CC1。
- 将定时器配置为编码器模式后，需要将 TIM_CTENR 寄存器的 CTNEN 位置 1 来启动计数器。

假设计数器已经使能，则计数器由 CH0INFP 和 CH1INFP 的有效跳变沿驱动。编码器控制器根据两个输入信号的跳变顺序，产生计数器的计数脉冲和计数方向，使计数器向上或向下计数，无论计数器是由 CH0INFP 驱动计数还是由 CH1INFP 驱动计数或者由 CH0INFP 和 CH1INFP 的边沿同时驱动，任一输入端跳变都会重新计算计数器的方向。计数器的计数方向产生规则如 Table 159 所示，表中相对信号的含义为：若计数器由 CH0INFP 的有效跳变沿驱动计数时，相对信号为 CH1INFP；若计数器由 CH1INFP 的有效跳变沿驱动计数时，相对信号为 CH0INFP；若计数器由 CH0INFP 和 CH1INFP 的有效跳变沿同时驱动计数时，互为相对信号。

Table 159 增量编码器接口模式下计数器计数规则

有效边沿	相对信号的 电平	CH0INFP 信号		CH1INFP 信号	
		上升沿	下降沿	上升沿	下降沿
仅在 CH0INFP 边 沿计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 CH1INFP 边	高	不计数	不计数	向上计数	向下计数

有效边沿	相对信号的 电平	CH0INFP 信号		CH1INFP 信号	
		上升沿	下降沿	上升沿	下降沿
沿计数	低	不计数	不计数	向下计数	向上计数
在 CH0INFP 和 CH1INFP 边沿计 数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

编码器接口模式相当于使用了一个带有方向选择的外部时钟，这意味着计数器只有在 0 到 TIM_ARR 寄存器的自动装载值之间连续计数（根据 Table 159 增量编码器接口模式下计数器计数规则中的方向，或者是 0 到 TIM_ARR 计数，或者是 TIM_ARR 到 0 计数），所以在开始计数之前必须配置 TIM_ARR。

在这种模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置，计数器方向与相连的传感器旋转的方向对应。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。Figure 87 为在增量编码器模式 3 下对输入信号 CH0INFP 和 CH1INFP 的编码。

例子：在增量编码器模式下，计数器在输入信号 CH0INFP 和 CH1INFP 的驱动下计数器，计数过程如 Figure 87 所示。

- 配置 GPIO 模块寄存器，将相应的引脚复用为高级定时器的输入管脚 PWM_CH0 和 PWM_CH1，并且将增量编码器的输出信号连接至 PWM_CH0 和 PWM_CH1，这样增量编码器的信号由定时器的通道 CC0 和 CC1 输入。
- 配置寄存器 TIM_CCENR 的 CC0EN 和 CC1EN 位为 1，使能通道 CC0 和 CC1 输入。
- 配置寄存器 TIM_CCPS 的 CC0P 和 CC1P 位为 0，通道 CC0 和 CC1 输入信号不反相。
- 配置寄存器 TIM_TGICR0 的 SMS 位配置为 011，将定时器从模式配置为编码器模式 3。
- 配置寄存器 TIM_CNTEN 的 CNTEN 位为 1，使能计数器。

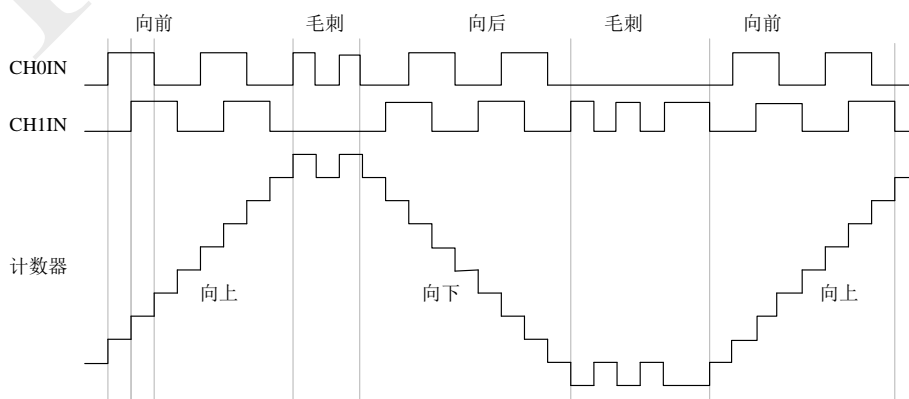


Figure 87 高级定时器作为增量编码接口的例子

11.5.3.4 PWM 输入模式

PWM 输入模式是输入捕获模式的一个特例,可以测量一个 PWM 信号的占空比和周期,假设 PWM 信号由通道 CC0 输入,具体配置如下:

- 将 TIM_ARR 配置为最大值。
- 配置寄存器 TIM_CCPS 的 CC0P 位为 0, CC0 通道使用信号的上升沿捕获数据。
- 配置寄存器 TIM_CC0MR 的 CC0MS 位为 01, 选择 CH0INF 的上升沿捕获数据。
- 不对捕获信号分频, 配置寄存器 TIM_CC0MR 的 IC0PDIV 位为 00。
- 配置寄存器 TIM_CCPS 的 CC1PS 位为 1, 配置寄存器 TIM_CC0MR 的 CC0MS 位为 10, 选择通道 CC0 信号 CH0INF 的下降沿作为通道 CC1 的捕获信号。
- 配置寄存器 TIM_TGICR0 的 TRGS 位为 101, 选择 CH0INFP 作为从模式的触发信号 TRGI。
- 配置寄存器 TIM_TGICR0 的 SMS 位为 100, 将从模式配置为复位模式, CH0INFP 的上升沿将计数器清零。
- 配置寄存器 TIM_CCENR 的 CC0EN 位和 CC1EN 位为 1, 使能通道 CC0 和通道 CC1 的捕获使能。
- 配置寄存器 TIM_EGR 的 UG 位和 COMG 位为 1, 初始化计数器并更新相应寄存器。
- 配置寄存器 TIM_CNTEN 的 CNTEN 位为 1, 启动计数器。

因为只有 CH0INFP 和 CH1INFP 连接到了从模式控制器,所以 PWM 输入模式只能使用 CC0 和 CC1 通道。PWM 输入模式捕获时序如 Figure 88 所示。

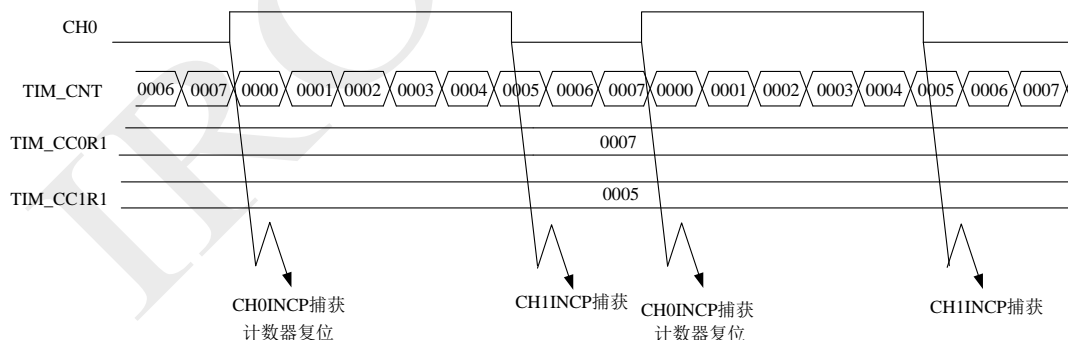


Figure 88 PWM 输入模式时序图

11.5.4 输出比较功能

将寄存器 TIM_CCxMR 的 CCxMS 位配置为 00, 通道 CCx 工作在输出比较模式下。在输出比较模式下,四个通道是相互独立的,可以独立配置各自的工作模式。通道 CC0、CC1 和 CC2 配置为比较输出时可以输出两路互补的 PWM 波形,输出控制过程如 Figure 89 所示,通道 CC3 只能输出一路 PWM 波形,输出控制过程如 Figure 90 所示。

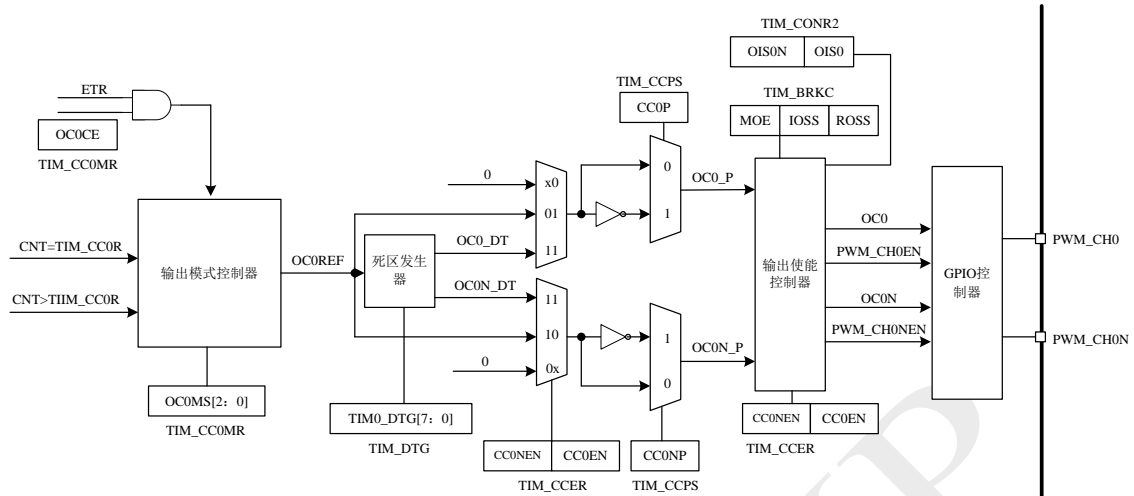


Figure 89 通道 CC0 输出控制

以通道 CC0 为例说明通道 CC0、CC1、CC2 的输出控制过程如下：

- 配置寄存器 TIM_CC0MR 的 CC0MS 位为 00，将 CC0 通道配置为输出功能。
- 配置寄存器 TIM_CC0MR 的 OCOMS 位域，选择输出模式，可选的输出模式有：冻结模式、匹配时输出有效电平、匹配时输出无效电平、强制输出有效电平、强制输出无效电平、翻转、PWM 模式 1、PWM 模式 2，各种模式的详细描述见 11.5.4.1 节，输出基准参考信号 OC0REF，OC0REF 高电平为有效电平、低电平为无效电平。当寄存器 TIM_CC0MR 的 OC0CE 位配为 1 时，外部信号 PWM_ETR 的有效电平清除 OC0REF。
- 若输出模式配置为 PWM 模式 1 或 PWM 模式 2 且主路输出和互补输出的使能均有效（CC0ENN=1，CC0NEN=1），则根据寄存器 TIM_DTG 的配置插入死区，输出两路互补信号 OC0_DT 和 OC0N_DT。若两路使能不同时有效，则不插入死区，直接在使能打开的一路输出 OC0REF，使能关闭的一路输出低电平。
- 对输出信号进行极性控制，配置寄存器 TIM_CCPS 的 CC0P 位为 0 时，主路输出高电平为有效电平，当 CC0P 位为 1 时，主路输出低电平为有效电平，极性控制后得到输出信号 OC0_P；配置寄存器 TIM_CCPS 的 CC0NP 位为 0 时，互补路输出高电平为有效电平，当 CC0NP 位为 1 时，互补路输出低电平为有效电平，极性控制后得到输出信号 OC0N_P。
- 根据寄存器 TIM_BRKC 中 MOE 位、IOSS、ROSS 位、寄存器 TIM_CCER 中的 CC0EN、CC0NEN、寄存器 TIM_CONR1 中的 OIS0 和 OIS0N 位的配置控制主路输出和互补路的输出，并控制主路和互补路的输出使能，详细控制方法见 11.5.4.4 节所述。最终从高级定时器中输出信号 OC0 和 OC0N 以及两路的输出使能信号 PWM_CH0EN 和 PWM_CH0NEN 至 GPIO 模块，在 GPIO 模块中选择 PWM 输出功能将 OC0 和

OC0N 输出到相应的管脚。

- 若计数器与输出比较寄存器 TIM_CC0R 比较成功，则寄存器 TIM_ISR0 的中断标志位 CC0IF 置位。若设置了通道 CC0 的输入输出中断使能，则产生通道 CC0 的输出比较中断。

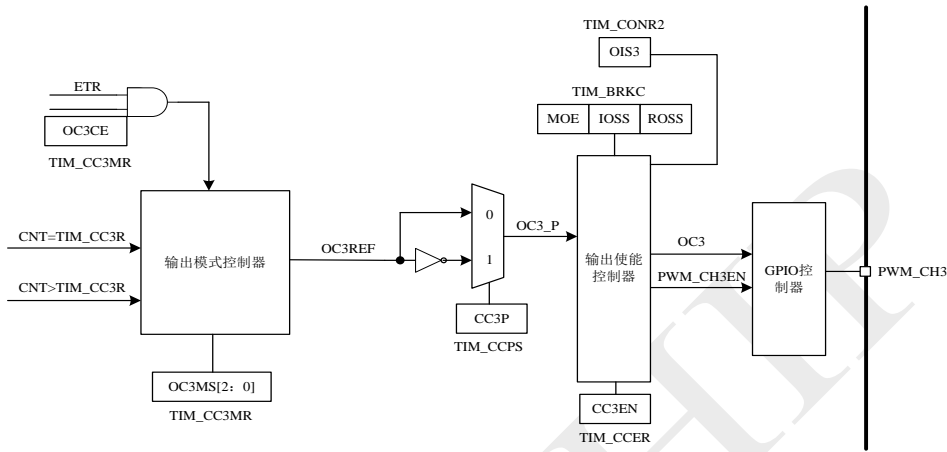


Figure 90 通道 CC3 输出控制

通道 CC3 的输出与通道 CC0-CC2 不同，通道 CC3 只有主路输出 OC3，没有互补路输出，输出控制如 Figure 90 所示，通道 CC3 的输出控制过程为：

- 配置寄存器 TIM_CC3MR 的 CC3MS 位为 00，将 CC3 通道配置为输出功能。
- 配置寄存器 TIM_CC3MR 的 OC3MS 位，选择输出模式，可选的输出模式有：冻结模式、匹配时输出有效电平、匹配时输出无效电平、强制为有效电平、强制为无效电平、翻转、PWM 模式 1、PWM 模式 2，各种模式的详细描述见 11.5.4.1 节，输出基准参考信号 OC3REF，OC3REF 高电平为有效电平、低电平为无效电平。当寄存器 TIM_CC3MR 的 OC3CE 位配置为 1 时，外部信号 PWM_ETR 的有效电平清除 OC3REF。
- 对输出信号进行极性控制，配置寄存器 TIM_CCPS 的 CC3P 位为 0 时，主路输出高电平为有效电平，当 CC3P 位为 1 时，主路输出低电平为有效电平，极性控制后得到输出信号 OC3_P。
- 根据寄存器 TIM_BRKC 中 MOE 位、IOSS、ROSS 位、寄存器 TIM_CCER 中的 CC3EN、寄存器 TIM_CONR1 中的 OIS3 位的配置控制主路输出，并控制主路输出使能，详细控制方法见 11.5.4.4 节所述。最终从高级定时器中输出信号 OC3 以及输出使能信号 PWM_CH3EN，将 OC3 和 PWM_CH3EN 送入 GPIO 控制模块，在 GPIO 模块做相应的控制后输出到相应的管脚。
- 若计数器与输出比较寄存器 TIM_CC3R 比较成功，则寄存器 TIM_ISR0 的中断标志位 CC3IF 置位。若设置了通道 CC3 的输入输出中断使能，则产生通道 CC3 的输出比较中断。

出比较中断。

11.5.4.1 输出模式控制

11.5.4.1.1 强置输出模式

在输出模式下（寄存器 TIM_CCxMR 的 CCxMS 位配置为 00），输出比较信号（OCxREF 和相应的 OCx/OCxN）能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器的比较结果。

将寄存器 TIM_CCxMR 的 OCxMS 位配置为 101，可将输出比较信号（OCxREF/OCx）强置为有效状态，这样 OCxREF 被强置为高电平（OCxREF 始终为高电平有效），同时 OCx 得到与寄存器 TIM_CCPS 的 CCxP 位相反的信号。例如，CCxP 位配置为 1（将 OCx 配置为低电平有效），则 OCx 被强置为低电平（与 CCxP 值相反）。

将寄存器 TIM_CCxMR 寄存器的 OCxMS 位配置为 100，可将输出比较信号（OCxREF/OCx）强置为无效状态，这样 OCxREF 被强置为低电平（OCxREF 始终为高电平有效），同时 OCx 得到与寄存器 TIM_CCPS 的 CCxP 位相同的信号，例如，CCxP 配置为 1（将 OCx 配置为低电平有效），则 OCx 被强置为高电平（与 OCxP 的值相同）。

在强置输出模式下，比较寄存器 TIM_CCxR 和计数器之间的比较仍在进行，相应的标志也会被修改。因此仍然会产生相应的中断标志和中断请求。

Figure 91 为当高级定时器配置为强置输出模式时的时序图。

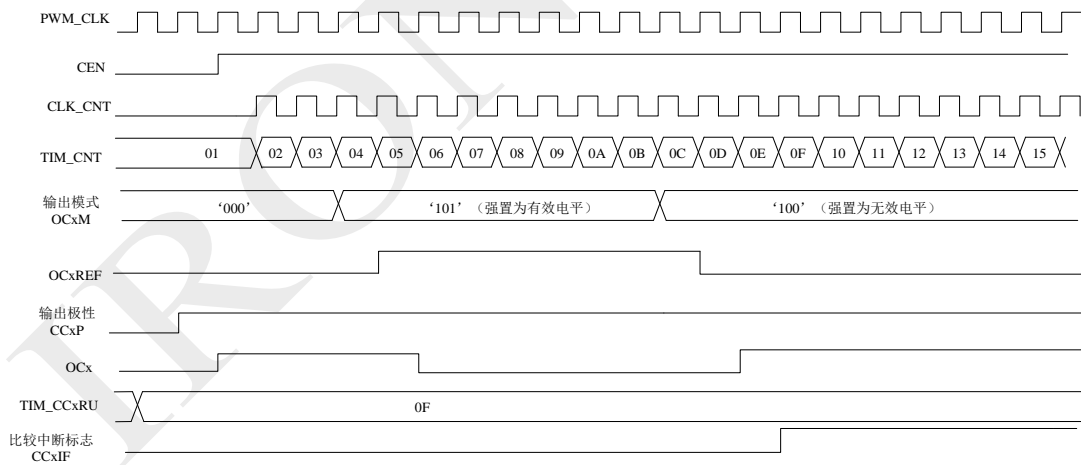


Figure 91 高级定时器强置输出模式时序图

11.5.4.1.2 输出比较模式

当计数器与比较寄存器的内容相同时，输出比较功能做如下操作：

- 定义基准参考波形 OCxREF 和定时器输出 OCx 的变化，当计数器与比较寄存器匹配时（TIM_CNT = TIM_CCxR）：
 - 若寄存器 TIM_CCxMR 的 OCxMS 位配置为 000，则 OCxREF 保持不变，OCx 保持不变。

- 若寄存器 TIM_CCxMR 的 OCxMS 位配置为 001, 则 OCxREF 被设置为高电平, 输出 OCx 被设置为有效电平, 即若 CCxP=0, 则 OCx 输出高电平, 若 CCxP=1, 则 OCx 输出低电平。
- 若寄存器 TIM_CCxMR 的 OCxMS 位配置为 010, 则 OCxREF 被设置为低电平, 输出 OCx 被设置为无效电平, 即若 CCxP=0, 则 OCx 输出低电平, 若 CCxP=1, 则 OCx 输出高电平。
- 若寄存器 TIM_CCxMR 的 OCxMS 位配置为 011, 则 OCxREF 波形翻转, 输出 OCx 的电平翻转。
- 设置中断状态寄存器 TIM_ISR0 中的中断标志位 CCxIF。
- 若设置了相应的中断使能, 则产生一个比较中断。

在输出比较模式下, 更新事件 UEV 对 OCxREF 输出没有影响。

输出比较模式的配置步骤为:

- 选择计数器时钟和预分频器 (内部时钟, 外部时钟, 预分频器)。
- 配置寄存器 TIM_CONR2 的 ARPLE 位和 TIM_CCxMR 的 OCxPEN 位, 确定 TIM_ARR 寄存器、TIM_CCxR 寄存器的预装载功能是否打开。
- 配置自动装载寄存器 TIM_ARR、捕获比较寄存器 TIM_CCxR, 都是遵循先配置高 8 位再配置低 8 位的规则。
- 配置寄存器 TIM_CCxMR 的 OCxMS 位, 选择输出模式, 配置 TIM_CCPS 寄存器, 选择输出极性。
- 配置寄存器 TIM_CCENR, 使能相应的输出通道。
- 若要产生一个输出比较中断请求, 设置寄存器 TIM_IER 的 CCxIE 位。
- 配置寄存器 TIM_EGR 的 UG 位和 COMG 位为 1, 更新具有预装载功能的寄存器。
- 设置 TIM_CNTEN 寄存器的 CNTEN 位, 启动计数器。

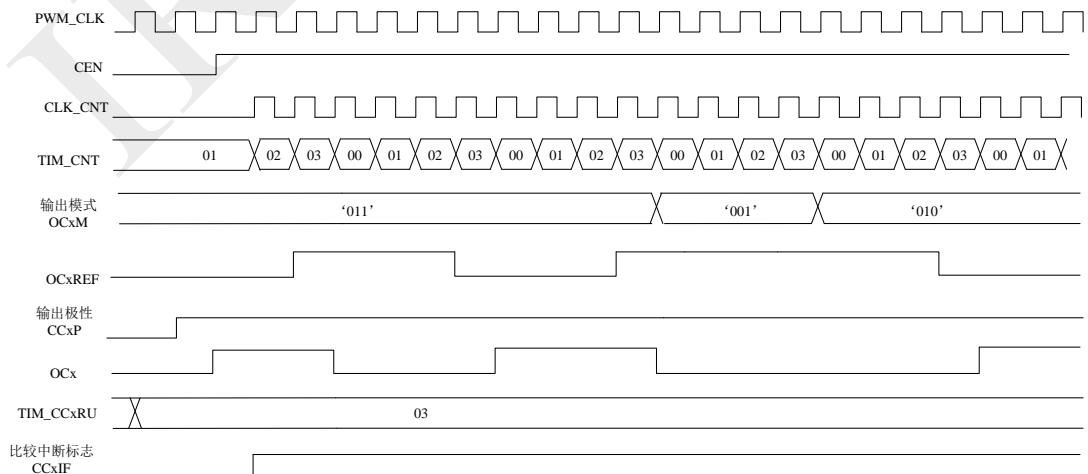


Figure 92 高级定时器比较输出模式

11.5.4.1.3 PWM 输出模式

脉冲宽度调制模式 (PWM) 可以产生一个由 TIM_ARR 寄存器确定周期、由 TIM_CCxR 寄存器确定占空比的信号。

配置 TIM_CCxMR 寄存器中的 OCxMS 位为“110”，定时器输出 PWM 模式 1 的波形；配置 TIM_CCxMR 寄存器中的 OCxMS 位为“111”，定时器输出 PWM 模式 2 的波形。

TIM_CCMRx 寄存器 OCxPEN 位控制比较寄存器 TIM_CCxR 的预装载使能。TIM_CONR2 的 ARPLE 位控制自动重装载寄存器 TIM_ARR 的预装载使能。若使能了预装载功能，仅当发生一个更新事件时，预装载寄存器才能被传送到影子寄存器，因此在计数器计数之前，必须通过设置 TIM_EGR 寄存器的 UG 位来更新具有预装载功能寄存器。

输出波形的极性可以通过软件在 TIM_CCPS 中设置，可以设置高电平为有效电平或低电平为有效电平。高级定时器在管脚上的输出 PWM_CHx 或 PWM_CHxN 通过 CCxEN、CCxNEN、MOE、IOSS、ROSS、OISx 和 OISxN 位的组合控制，详见 11.5.4.4 节的描述。

PWM 模式分为 PWM 模式 1 和 PWM 模式 2，PWM 模式 1 波形产生的规则为：当 $TIM_CNT < TIM_CCxR$ 时，PWM 基准参考信号 OCxREF 为有效电平（高电平），否则为无效电平（低电平）；PWM 模式 2 波形产生的规则为：当 $TIM_CNT < TIM_CCxR$ 时，PWM 基准参考信号 OCxREF 为无效电平（低电平），否则为有效电平（高电平）；

下面以 PWM 模式 1 为例说明在不同的计数器计数模式下 PWM 输出控制。

- 向上计数模式下的 PWM 模式 1

当 TIM_CONR2 寄存器中的 CNTMC 位配置为 00 且 CNTDIR 位配置为 0 时，计数器向上计数。此时，PWM 模式 1 的输出为：当 $TIM_CNT < TIM_CCxR$ 时，PWM 基准参考信号 OCxREF 为有效电平（高电平），否则为无效电平（低电平）。如果 TIM_CCxR 中的比较值大于自动重装载值，则 OCxREF 保持为有效电平（高电平）。如果比较值为 0，则 OCxREF 保持为无效电平（低电平）。Figure 93 为 TIM_ARR=8 且计数器向上计数时 PWM 模式 1 的输出波形。

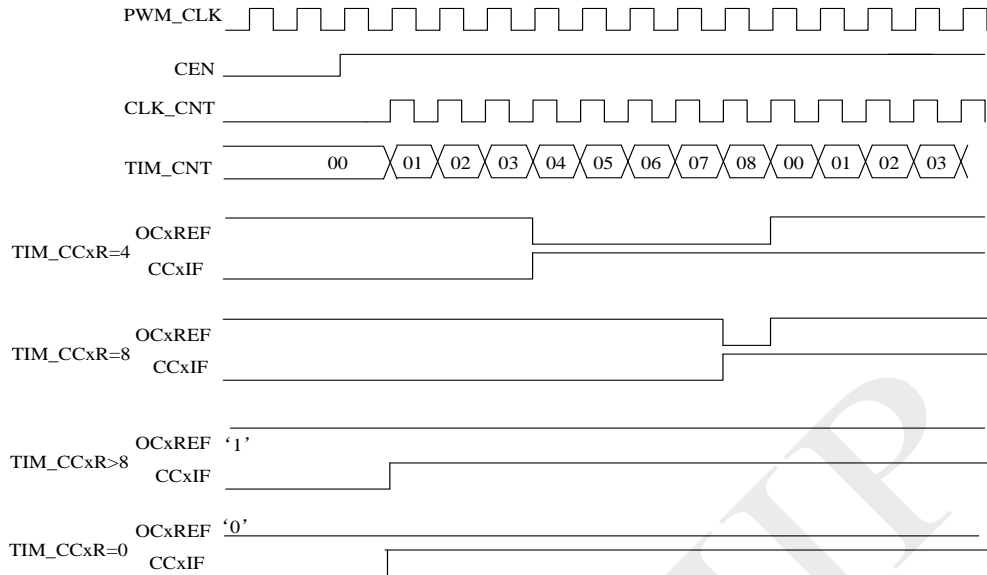


Figure 93 向上计数模式下，PWM 模式 1 输出

● 向下计数模式下的 PWM 模式 1

当 TIM_CONR2 寄存器中的 CNTMC 位配置为 00 且 CNTDIR 位配置为 1 时，计数器向上计数。此时，PWM 模式 1 的输出为：当 TIM_CNT > TIM0_CCxR 时，PWM 基准参考信号 OCxREF 为无效电平（低电平），否则为有效电平（高电平）。如果 TIM_CCxR 中的比较值大于自动重装载值，则 OCxREF 保持为有效电平（高电平）。该模式下不能产生 0% 的 PWM 波形。Figure 94 为 TIM_ARR=8 时向上计数器时 PWM 模式 1 的输出波形。

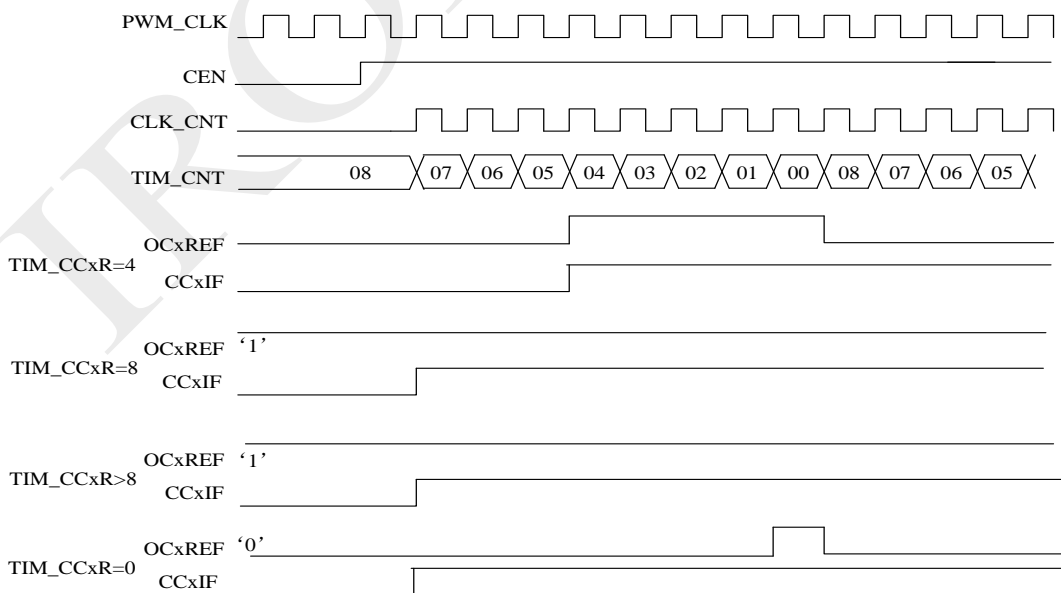


Figure 94 向下计数模式下，PWM 模式 1 输出

● 中央对齐计数模式下的 PWM 模式 1

当 TIM_CONR2 寄存器中的 CNTMC 位配置为 01/10/11 时，计数器工作在中央对齐模式下，TIM_CONR2 寄存器中的计数方向 CNTDIR 不能用软件配置。

在此种计数模式下，PWM 模式 1 输出波形规则为：在向上计数过程中，当 $TIM_CNT < TIM_CCxR$ 时，PWM 基准参考信号 OCxREF 为有效电平（高电平），否则为无效电平（低电平）；在向下计数过程中，当 $TIM_CNT > TIM_CCxR$ 时，PWM 基准参考信号 OCxREF 为无效电平（低电平），否则为有效电平（高电平）。当 CNTMC 位配置为 01 时，比较标志只在计数器向上计数时被置位；当 CNTMC 位配置为 10 时，比较标志只在计数器向下计数时被置位；当 CNTMC 位配置为 11 时，比较标志在计数器向上和向下计数时都被置位。Figure 95 为中央对齐的 PWM 模式 1 输出波形，图中箭头表示将比较中断标志置位。

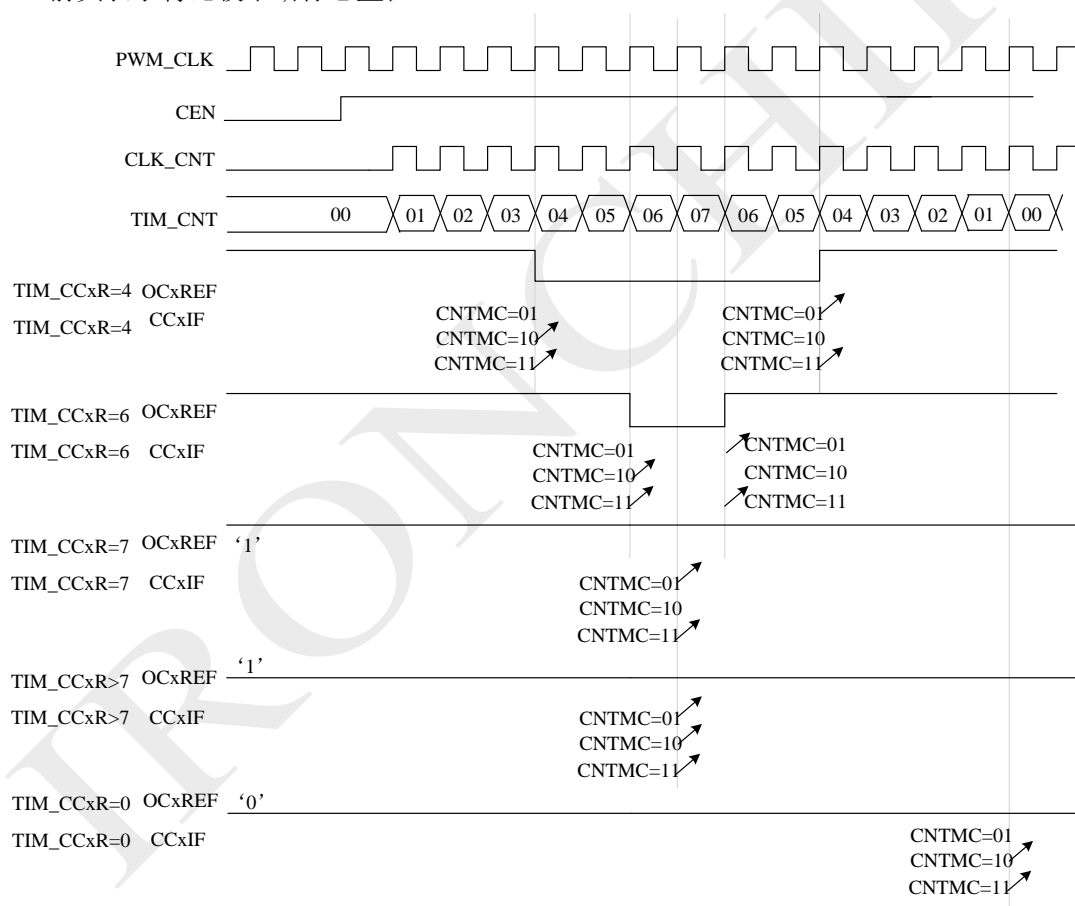


Figure 95 中央对齐计数模式下，PWM 模式 1 输出

使用中央对齐模式的注意事项：

- 在中央对齐计数模式下，若配置了计数器的复位功能（通过配置 TIM_EGR 寄存器的 UG 位或通过从模式控制器产生复位信号），计数器开始向上计数，若没有配置计数器的复位功能，计数器按照以前保留的计数方向开始计数。此外，CNTDIR 和 CNTMC 位不能同时被软件修改，在边沿计数模式和中央对齐计数模式间转换时需

按照以下步骤配置计数模式控制寄存器：

- 从边沿计数模式向中央对齐计数模式转换

首先，配置寄存器 TIM_CONR2 的 CNTDIR 位，若希望中央计数模式下首先向下计数，则配置 CNTDIR 位为 1，若希望中央计数模式下首先向上计数，则配置 CNTDIR 位为 0，若希望中央计数模式下首先向下计数，将 CNTDIR 位配置为 1 后不可再配置 TIM_EGR 寄存器的 UG 位。

然后，配置寄存器 TIM_CONR2 的 CNTMC 位为 01/10/11。

- 从中央对齐计数模式向边沿计数模式转换

首先，配置寄存器 TIM_CONR2 的 CNTMC 位为 00，将计数器切换至边沿计数模式。

然后，配置寄存器 TIM_CONR2 的 CNTDIR 位，确定计数方向。

- 不建议当运行在中央对齐模式时改写计数器，因为会产生不可预知的结果：
 - 如果写入计数器的值大于自动重加载的值，则方向不会被更新，若计数器正在向上计数，计数器将继续向上计数。
 - 如果将 0 或者 TIM_ARR 的值写入计数器，方向被更新，但是不产生更新事件。
- 使用中央对齐模式推荐的方法为：在启动计数器之前产生一个软件更新（设置 TIM_EGR 寄存器中的 UG 位），不要在计数过程中修改计数器的值。

11.5.4.2 互补输出和死区插入

高级定时器能够输出两路互补信号 OCx_DT 和 OCxN_DT（相反的两路信号），也能控制输出在一段时间内无效，然后再接通，这段时间通常称为死区，本定时器的死区时间是可编程的，在具体应用中应该根据连接到输出的器件和他们的特性（电平转换的延时、电源开关的延时等）来调整死区时间。

互补输出 OCx_DT 和 OCxN_DT 在参考信号 OCxREF 基础上产生，定时器通过以下方式控制互补输出 OCx_DT 和 OCxN_DT：

- 配置 TIM_CCPS 寄存器的 CCxP 和 CCxNP 位，可以为每一路 PWM 输出独立地选择极性；
- 通过配置 TIM_CCxENR 寄存器的 CCxEN 和 CCxNEN 位，控制互补输出的使能。
- 当定时器工作在不同的工作状态时（如刹车状态），互补输出还受到 TIM_BRKC 寄存器的 MOE 位、IOSS、ROSS 位和 TIM_CONR1 寄存器的 OISx、OISxN 位的控制，详见 11.5.4.4 章节。

同时设置 CCxE、CCxNE 位，则 PWM 输出将插入死区。每一个通道都有一个死区发生器，定时器根据参考信号 OCxREF 产生两路带死区的互补输出 OCx_DT 和 OCxN_DT，若两路互补输出都配置为高电平有效：

- 输出信号 OCx_DT 与参考信号 OCxREF 相同，只是他的上升沿相对于参考信号的上升沿有一个延迟。
- 输出信号 OCxN_DT 与参考信号 OCxREF 相反，只是他的上升沿相对于参考信号的下降沿有一个延迟。

如果死区时间大于当前有效的输出宽度，则不会产生相应的脉冲。

Figure 96—Figure 98 显示了带死区的互补输出与当前参考信号 OCxREF 之间的关系。

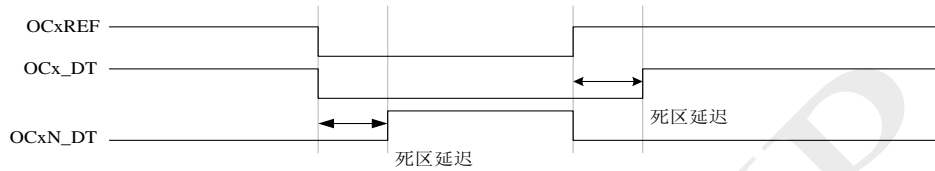


Figure 96 正常死区插入

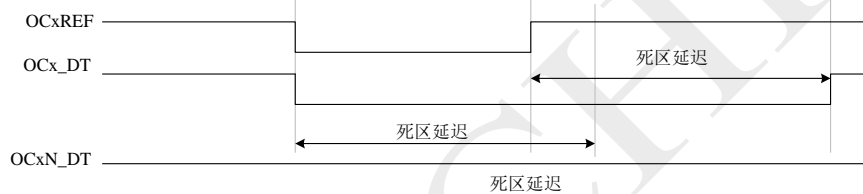


Figure 97 死区时间大于 OCxREF 无效电平宽度时的死区插入



Figure 98 死区时间大于 OCxREF 有效电平宽度时的死区插入

值得注意的是：只要同时设置 CCxE 和 CCxNE 位为 1，无论在 PWM 输出模式下还是在非 PWM 输出模式下(强置输出模式和输出比较模式), OCx 和 OCxN 均输出互补的波形，因此，在非 PWM 输出模式下，若不需要输出互补的波形，只需使能主路输出即可（即只设置 CCxE）。

11.5.4.3 相位偏移功能

在本高级定时器中，通过配置相应的寄存器可以使某一通道的 PWM 输出比其他通道的 PWM 输出具有一定的相位偏移。

将寄存器 TIM_PHACON 的 PHASEEN 位配置为 1，打开定时器的相位偏移功能，将 PHASEEN 位配置为 0，关闭定时的相位偏移功能。

通过配置 TIM_PHACON 的 PHASECH 位选择具有相位偏移功能（向后延迟）的输出通道，例如将 PHASECH 位配置为 00，则通道 CC0 的 PWM 输出 OCx 和 OCxN 比其他通道的 PWM 输出有一定的相位偏移。

配置 TIM_PHASEL 和 TIM_PHASEH 决定相位偏移的长度，当配置的相位偏移长度大于等于计数器的周期值时，输出信号不偏移，当配置的相位偏移长度为 0 时，输出信号也不偏移。

Figure 99 为通道 CC1 的 PWM 输出具有相位偏移的时序图。

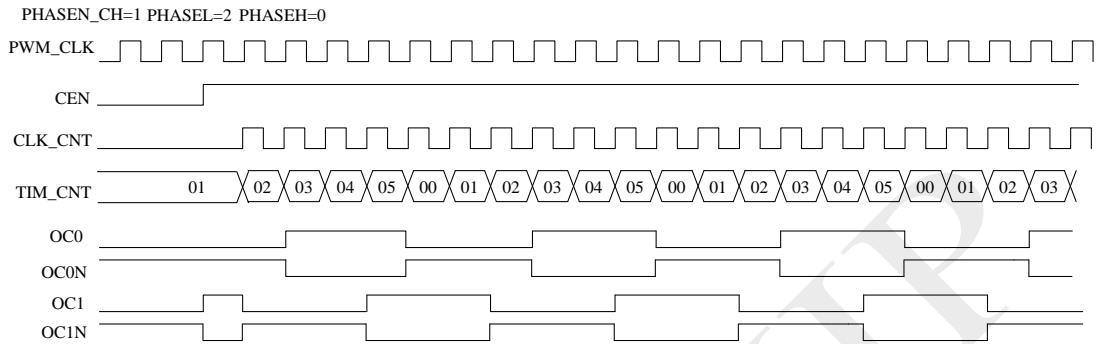


Figure 99 具有相位偏移的 PWM 输出时序图

11.5.4.4 输出使能控制

高级定时器最终向引脚输出每一通道的两路输出使能信号 PWM_CHxEN、PWM_CHxNEN 和每一通道互补输出 OCx、OCxN。

若输出使能信号 PWM_CHxEN 为 1，则相应的引脚输出 OCx，若 PWM_CHxEN 为 0，则相应的引脚不输出 OCx，而是输出高阻；输出使能信号 PWM_CHxNEN 对引脚的控制同理。

输出使能信号 PWM_CHxEN 和 PWM_CHxNEN 受到 TIM_BRKC 寄存器的 MOE、IOSS、ROSS 位和 TIM_CCENR 寄存器的 CCxEN、CCxNEN 位控制，具体控制方式如 Table 160 和 Table 161 所示。

高级定时器的管脚输出 PWM_CHx 和 PWM_CHxN 受到 TIM0_BRKC 寄存器的 MOE 位、IOSS 位、ROSS 位、TIM_CCENR 寄存器的 CCxEN、CCxNEN 位和计数器使能 CEN 的控制，当 CEN=0 时，计数器不工作，此时 PWM_CHx 和 PWM_CHxN 的输出值如 Table 160 所示。当 CEN=1 时，计数器正常计数，高级定时器处于正常工作状态，此时 PWM_CHx 和 PWM_CHxN 的输出值如 Table 161 所示。

寄存器 TIM_BRKC 的 MOE 位为 1，表示定时器处于运行模式，MOE 为 0 表示定时器处于空闲模式。

由 Table 160 和 Table 161 可以看出，当 MOE 位为 1 时，管脚的 PWM 输出使能（PWM_CHxEN/PWM_CHxNEN）受寄存器 TIM_BRKC 的 ROSS 位和 TIM_CCENR 的 CCxEN、CCxNEN 位控制，ROSS 位表示运行模式下（MOE=1）关闭状态（TIM_CCENR 的 CCxEN 位或 CCxNEN 位为 0）选择。当 ROSS 位为 0 时，表示运行模式下管脚的输出使能 PWM_CHxEN/PWM_CHxNEN 与 CCxEN/CCxNEN 的值一致，当 CCxE/CCxNE 配置为 0

时，相应管脚的输出使能 PWM_CHxEN/PWM_CHxNEN 为 0，即管脚的输出使能关闭，此时管脚输出高阻。若 CCxE/CCxNE 配置为 1，则相应管脚的输出使能 PWM_CHxEN/PWM_CHxNEN 为 1，管脚的输出使能有效，管脚输出 PWM 模块的输出值 OCx/OCxN。当 ROSS 位为 1 时，无论 CCxEN/CCxNEN 位配置为何值，管脚的输出使能 PWM_CHxEN/PWM_CHxNEN 均有效，管脚输出 PWM 模块的输出值。

当 MOE 为 0 时，管脚上的 PWM 输出和管脚的输出使能受寄存器 TIM_BRKC 的 IOSS 位和 TIM_CCENR 的 CCxEN、CCxNEN 位控制，IOSS 位表示空闲模式下 (MOE=0) 关闭状态 (TIM_CCENR 的 CCxEN 位或 CCxNEN 位为 0) 选择。当 IOSS 配置为 0 时，无论 CCxEN/CCxNEN 位配置为何值，管脚的输出使能 PWM_CHxEN/PWM_CHxNEN 均无效，PWM 输出管脚输出高阻。当 IOSS 配置为 1 时，管脚的输出使能 PWM_CHxEN/PWM_CHxNEN 均有效，管脚输出 PWM 模块的输出值 OCx/OCxN。

Table 160 互补模式下，计数器不使能时 PWM 各个通道的输出及输出使能控制

CEN=0						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWM_CHx 输出状态	PWM_CHxN 输出状态
1	X	0	0	0	输出禁止 (引脚输出高阻) PWM_CHxEN =0; PWM_CHx=高阻	输出禁止 (引脚输出高阻) PWM_CHxNEN =0; PWM_CHxN=高阻;
		0	0	1	输出禁止 (引脚输出高阻) PWM_CHxEN =0 PWM_CHx=高阻	PWM_CHxNEN =1; PWM_CHxN=OCxN =CCxNP;
		0	1	0	PWM_CHxEN =1; PWM_CHx=OCx =CCxP;	输出禁止 (引脚输出高阻) PWM_CHxNEN =0; PWM_CHxN=高阻;
		0	1	1	PWM_CHxEN =1 PWM_CHx=OCx =死区+CCxP;	PWM_CHxNEN =1; PWM_CHxN=OCxN =死区+!CCxNP;
		1	0	0	关闭状态 (输出使能有效, 输出无效电	关闭状态 (输出使能有效, 输出无效电平)

CEN=0									
控制位					输出状态				
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWM_CHx 输出状态	PWM_CHxN 输出状态			
					平) PWM_CHxEN =1 PWM_CHx=OCx = CCxP;	PWM_CHxNEN=1; PWM_CHxN=OCxN = CCxNP;			
					1	0	1	关闭状态(输出使能有效, 输出无效电平) PWM_CHxEN =1 PWM_CHx=OCx = CCxP;	PWM_CHxNEN =1; PWM_CHxN=OCxN =CCxNP;
					1	1	0	PWM_CHxEN =1 PWM_CHx=OCx =CCxP;	关闭状态(输出使能有效, 输出无效电平) PWM_CHxNEN =1; PWM_CHxN=OCxN = CCxNP;
					1	1	1	PWM_CHxEN =1 PWM_CHx=OCx =CCxP;	PWM_CHxNEN =1; PWM_CHxN=OCxN =死区+!CCxNP;
0	0	X	0	0	输出禁止(引脚输出高阻)				
	0		0	1	PWM_CHxEN = 0; PWM_CHxNEN = 0;				
	0		1	0	PWM_CHx =高阻;				
	0		1	1	PWM_CHx N=高阻;				
	1		0	0	关闭状态(输出使能有效, 输出无效电平) PWM_CHxEN = 1, PWM_CHxNEN = 1;				
	1		0	1	● 当 CCxP=CCxNP 时: 若 OISx=OISx, PWM_CHx=OCx =CCxP, PWM_CHxN=OCxN =CCxNP;				
	1		1	0	若 OISxOISxN, PWM_CHx=OCx =OISx, PWM_CHxN=OCxN =OISxN。				
	1		1	1					

CEN=0						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWM_CHx 输出状态	PWM_CHxN 输出状态
					● 当 $CCxP \neq CCxNP$ 时： 若 $OISx = OISxN$ ，则 $PWM_CHx = OCx = OIS$ ， $PWM_CHxN = OCxN = OISxN$ ； 若 $OIS \neq OISxN$ ，则 $PWM_CHx = OCx = CCxP$ ， $PWM_CHxN = OCxN = CCxNP$ 。	

Table 161 互补模式下，计数器使能时 PWM 各个通道的输出及输出使能控制

CEN=1						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxE N 位	CCxNE N 位	PWM_CHx 输出状态	PWM_CHxN 输出状态
1	X	0	0	0	输出禁止（引脚输出高阻） $PWM_CHxEN = 0$ ； $PWM_CHx =$ 高阻；	输出禁止（引脚输出高阻） $PWM_CHxNEN = 0$ ； $PWM_CHxN =$ 高阻；
		0	0	1	输出禁止（引脚输出高阻） $PWM_CHxEN = 0$ ； $PWM_CHx =$ 高阻；	$PWM_CHxNEN = 1$ ； $PWM_CHxN = OCxN = OCxREF \text{ xor } CCxNP$
		0	1	0	$PWM_CHxEN = 1$ ； $PWM_CHx = OCx = OCxREF \text{ xor } CCxP$	输出禁止（引脚输出高阻） $PWM_CHxNEN = 0$ ； $PWM_CHxN =$ 高阻
		0	1	1	$PWM_CHxEN = 1$ ； $PWM_CHx = OCx = OCxREF + \text{死区} + \text{极性}$	$PWM_CHxNEN = 1$ ； $PWM_CHxN = OCxN = OCxREF + \text{死区} + \text{极性}$
		1	0	0	关闭状态（输出使能有效，输出无效电平） $PWM_CHxEN = 1$ ；	关闭状态（输出使能有效，输出无效电平） $PWM_CHxNEN = 1$ ；

CEN=1						
控制位					输出状态	
MO E 位	IOSS 位	ROS S 位	CCxE N 位	CCxNE N 位	PWM_CHx 输出状态	PWM_CHxN 输出状态
					PWM_CHx=OCx = CCxP	PWM_CHxN=OCxN = CCxNP
		1	0	1	关闭状态（输出使能有效，输出无效电平） PWM_CHxEN =1; PWM_CHx=OCx = CCxP	PWM_CHxNEN =1; PWM_CHxN=OCxN = OCxREF xor CCxNP
		1	1	0	PWM_CHxEN =1; PWM_CHx =OCx =OCxREF xor CCxP	关闭状态（输出使能有效，输出无效电平） PWM_CHxNEN =1; PWM_CHxN=OCxN = CCxNP
		1	1	1	PWM_CHxEN =1; PWM_CHx =OCx =OCxREF +死区+ 极性	PWM_CHxNEN =1; PWM_CHxN =OCxN =OCxREF +死区+ 极性
0	0	X	0	0	输出禁止（引脚输出高阻）	
	0		0	1	PWM_CHxEN = 0; PWM_CHxNEN = 0;	
	0		1	0	PWM_CHx =高阻;	
	0		1	1	PWM_CHx N=高阻;	
	1	1	0	0	关闭状态（输出使能有效，输出无效电平）	
	1		0	1	PWM_CHxEN = 1, PWM_CHxNEN = 1;	
	1		1	0	● 当 CCxP=CCxNP 时:	
	1		1	1	若 OISx=OISxN, PWM_CHx=OCx =CCxP, PWM_CHxN=OCxN =CCxNP; 若 OISxOISxN, PWM_CHx=OCx =OISx, PWM_CHxN=OCxN =OISxN。 ● 当 CCxPCCxNP 时: 若 OISx=OISxN, 则 PWM_CHx=OCx =OISx,	

CEN=1						
控制位					输出状态	
MO E 位	IOSS 位	ROS S 位	CCxE N 位	CCxNE N 位	PWM_CHx 输出状态	PWM_CHxN 输出状态
					PWM_CHxN=OCxN =OISxN; 若 OISxOISxN, 则 PWM_CHx=OCx =CCxP, PWM_CHxN=OCxN =CCxNP。	

由 Table 160 和 Table 161 可知,管脚上的 PWM 输出使能和 PWM 输出受 MOE、ROSS、IOSS、CCxE、CCxNEN、CCxP、CCxNP 和 CEN 控制,这些寄存器初始值都为 0,所以初始状态下 PWM 管脚输出高阻,为了保证 PWM 平稳启动,建议按照如下顺序配置寄存器:

- 1) 首先配置极性寄存器 TIM_CCxPS,若需要控制空闲模式(MOE=0)下的输出电平,可以配置寄存器 TIM_CONR1。
- 2) 配置 TIM_CONR2 的 ARPLE 位和 TIM_CONR0 的 CCPE 位,控制相应寄存器的预装载功能。
- 3) 配置各个通道输入捕获/输出比较模式控制寄存器 TIM_CCxMR。
- 4) 配置周期寄存器 TIM_ARR、比较/捕获寄存器 TIM_CCxR、通道使能寄存器 TIM_CCxEN、死区控制等各种寄存器。
- 5) 配置软更新事件,将新配置的参数装载到影子寄存器中(配置 TIM_EGR 的 UG 位为 1, COMG 位为 1, 装载各个预装载寄存器)。
- 6) 配置 TIM_BKRC 寄存器,对 MOE 位、ROSS 位、和 IOSS 位进行配置(建议配置 ROSS 和 IOSS 位为 1)。
- 7) 配置 TIM_CNTEN 寄存器的 CNTEN 位,启动 PWM 计数器。

11.5.4.5 刹车功能

当使用刹车功能时,依据相应的控制位(TIM_BRKC 寄存器的 MOE、IOSS 和 ROSS, TIM_CONR1 寄存器的 OISx 和 OISxN 位),输出使能和输出信号都会被修改。但无论何时,均保证输出信号 PWM_CHx 和 PWM_CHxN 不会同时处于有效电平。

11.5.4.5.1 刹车源

刹车源包括三种:

- 引脚刹车事件。
- 软刹车事件。通过配置刹车事件寄存器(配置寄存器 TIM_EGR 的 BG 位为 1)对 PWM 输出进行刹车。
- 比较器刹车,比较器的输出信号可以作为 PWM 的刹车信号,详见 0 节。

系统复位后,刹车电路被禁止,MOE 位为低,定时器工作在空闲模式下。设置 TIM_BRKC 寄存器的 BRKEN 位可以使能刹车功能(软刹车不受 BRKEN 位控制),刹车输入信号的极性可以通过配置寄存器 TIM_BRKC 中的 BKP 位选择。

11.5.4.5.2 刹车过程

当刹车事件发生时,硬件做如下处理:

- 非互补模式下:

MOE 位被异步地立即清零,每一个通道输出由寄存器 TIM_CONR2 的 OIS_x 位或 OIS_{xN} 位设定电平,如果 IOSS=0,则定时器管脚输出使能无效,否则管脚输出使能始终有效。这个特性在系统时钟关闭时依然有效。

- 互补模式下:

- 输出首先被置于无效状态,即使没有时钟,此功能也有效。
- 如果定时器的时钟存在,死区生成器会重新生效,在死区之后根据 OIS_x、OIS_{xN} 位以及极性指示的电平驱动输出端口,详见 Table 160 和 Table 161 的 MOE=0 部分。即使在这种情况下,PWM_CH_x 和 PWM_CH_{xN} 也不会被同时驱动到有效电平。因为重新同步 MOE,死区时间比通常情况长一些(大约 2 个定时器时钟周期)。
- 如果 IOSS=0,定时器管脚输出使能无效,管脚输出高阻,否则管脚输出使能有效。

- 如果设置了寄存器 TIM_IER 中的 BIE 位,则产生一个中断。

- 如果寄存器 TIM_BRKC 中的 AOE 位为 1,在下一个更新事件时 MOE 位被自动置 1;否则,MOE 始终保持为低直到被软件再次置 1;

当刹车输入有效时,软件无法配置 MOE,同时,状态标志不能被清除。

11.5.4.5.3 刹车后输出信号控制

因为当外部刹车信号来到时,MOE 必须立即变为低电平,即 MOE 的下降沿可以是异步的,所以在实际信号(作用在定时器的输出端)和寄存器的控制位之间设置了一个同步电路,当向 MOE 位写入数据时,硬件会将软件写入的值作为异步信号对待,对其做同步后再写入相应的寄存器位。因此,如果当 MOE 为 0 时写 MOE 为 1,则读出它之前必须先插入一个延时(空指令)才能读到正确的值,这是因为写入的是异步信号而读的是同步信号。

寄存器 TIM_BRKC 的 MOE 位控制 PWM 在管脚上的输出,当 MOE 为 1 时,PWM 处于运行模式下,若 TIM_BRKC 寄存器的 ROSS 位设置为 1,则无论 CC_xE 和 CC_xNE 配置为使能与否,PWM_CH_x 和 PWM_CH_{xN} 管脚的输出使能都为高;若 TIM_BRKC 寄存器的 ROSS 位设置为 0,则当 CC_xE 配置为有效时,PWM_CH_x 管脚的输出使能有效,否则,PWM_CH_x 管脚的输出使能为无效,PWM_CH_x 管脚输出高阻,PWM_CH_{xN} 的输出管脚控制同理。

当 MOE 为 0 位，PWM 输出处于空闲模式（刹车后的状态），若 TIM_BRKC 寄存器的 IOSS 位设置为 1，则无论 CCxE 和 CCxNE 配置为使能与否，PWM_CHx 和 PWM_CHxN 管脚的输出使能都有效；若 TIM_BRKC 寄存器的 IOSS 位设置为 0，则 PWM_CHx 和 PWM_CHxN 管脚的输出使能设置为无效，PWM_CHx 和 PWM_CHxN 管脚输出高阻。当 CNTEN 为 0 时，计数器不计数，定时器处于初始化状态，PWM_CHx 和 PWM_CHxN 管脚上的输出如 Table 160 所示，当 CNTEN 为 1 时，定时器处于工作状态，PWM_CHx 和 PWM_CHxN 管脚上的输出如 Table 161 所示。

Figure 100 显示了响应刹车的输出实例。

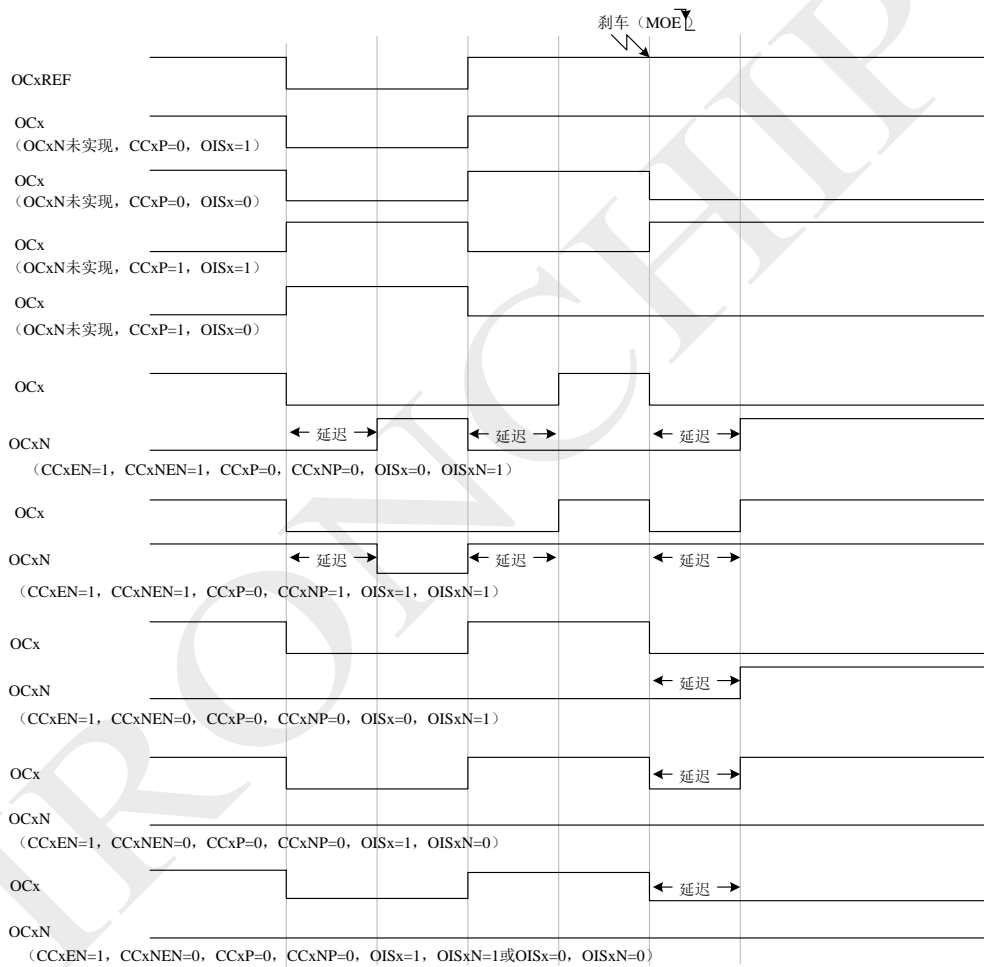


Figure 100 高级定时器刹车输出

11.5.4.6 外部事件 (PWM_ETR) 清除 OCxREF 信号

对于一个给定的通道,设置寄存器 TIM_CCxMR 中的 OCxCE 位为 1,能够用 PWM_ETR 通道输入信号的有效电平把 OCxREF 信号清为低电平, PWM_ETR 信号无效后 OCxREF 信号将保持为无效电平直到发生下一次更新事件,该功能只能用于输出比较和 PWM 模式,而不能用于强置模式。PWM_ETR 通道配置如下:

- 关闭 PWM_ETR 通道的预分频器。
- 禁止外部时钟模式 2。
- 配置外部触发信号的极性(EXTP)。

Figure 101 显示了当 PWM_ETR 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的变化。在这个例子中，定时器被设置为 PWM 模式。

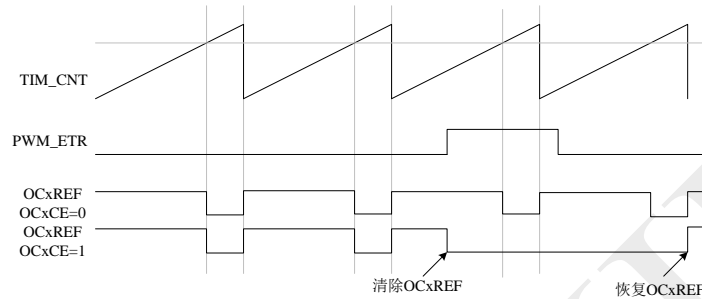


Figure 101 外部信号 PWM_ETR 清除 OCxREF

11.5.4.7 霍尔传感器接口功能

高级定时器 TIM 可以作为“接口定时器”来连接霍尔传感器。配置 TIM_CONR0 寄存器的 CH0INSEL 位为 1，选择定时器的三个输入脚（PWM_CH0、PWM_CH1、PWM_CH2）异或后的信号连接至 CC0 输入通道，高级定时器捕获这个信号。

配置高级定时器的从模式控制器为复位模式，选择 CH0INF_ED 作为从模式控制器的触发输入，每当 3 个霍尔信号中任意一个发生翻转时，计数器重新从 0 开始计数，这样产生一个由霍尔输入端的任何变化而触发的时间基准。

高级定时器的通道 CC0 配置为捕获模式，捕获信号为 CH0INF_ED，捕获值反映了两个输入变化间的时间延迟，给出了马达速度的信息。

高级定时器可以将 CC1-CC3 中的某一通道配置为输出模式，在输出模式下产生一个比较输出波形，将此波形通过 TRGO 送出。

11.5.4.8 六步 PWM

通过高级定时器 TIM 的 COM 事件产生六步 PWM。COM 事件可由两种方式产生：一种通过霍尔信号产生，另外一种通过无霍尔应用中的反电动势产生。

霍尔信号产生 COM 事件的方式如下：将三路霍尔信号连接到芯片的输入引脚，打开三个引脚的中断使能，当霍尔信号发生跳变时，管脚发出中断，在中断中设置 TIM0_EGR 寄存器的 COM 位进行换相。

每个六步 PWM 相位的输出控制有以下两种方法：

- 通过不同的 OCxMS、CCxE 和 CCxNE 的设置可以得到六步 PWM 每个相位的输出，这些控制位具有预装载功能，更新事件为 COM 事件。

- 通过设置 CCxE、CCxNE、CCxP 和 CCxNP 也可以得到六步 PWM 每个相位的输出，这些控制位具有预装载功能，更新事件为 COM 事件。

在发生 COM 事件时，这些预装载位被传送到影子寄存器，这样就可以预先设置好相应相位的输出控制寄存器，并在同一个时刻更新所有通道的配置。

例：将三路霍尔信号连接到 GPIO 的 PA0，PA1 和 PA2 管脚，打开 PA0、PA1 和 PA2 的中断使能，在 GPIO 中断中软件触发 TIM 的 COM 事件来改变 TIM 的输出配置。

- 配置 TIM 的周期值寄存器 TIM_ARR、输出模式控制寄存器 TIM_CCxMR、极性控制寄存器 TIM_CCPS、通道使能控制寄存器 TIM_CCENR、比较值寄存 TIM_CCxR 或 TIM0_CCxR 等。
- 配置寄存器 TIM_CONR0 的 CCPE 位为 1，使能比较输出控制寄存器的预装载功能。
- 将三路霍尔信号连接到 PA0，PA1 和 PA2 管脚，配置 PA0，PA1 和 PA2 为 GPIO 管脚模式。
- 配置寄存器 EXINTEN 为 0x07，配置寄存器 EXINTPE 为 0x07，配置寄存器 EXINTNE 为 0x07，将 PA0、PA1 和 PA2 的上升沿下降沿中断使能打开。
- 在中断处理函数中通过配置寄存器 TIM_CCENR、TIM_CCPS 实现对六步 PWM 每个相位的控制，在触发中断处理函数中配置 TIM_EGR 的 COM 位，产生 COM 事件，更新 TIM 的输出控制寄存器。

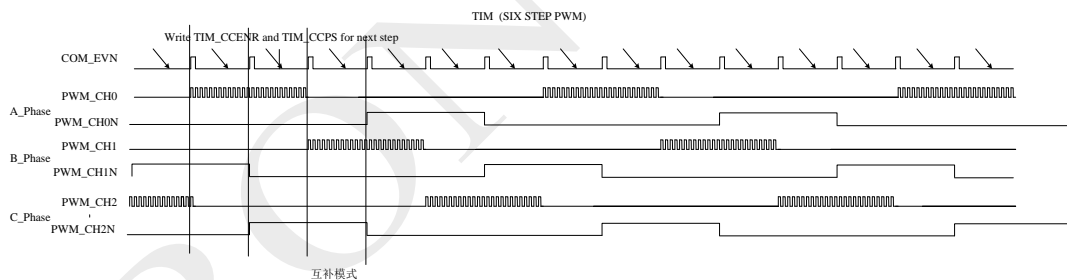


Figure 102 TIM 产生六步 PWM 的示意图

11.5.4.9 单脉冲模式

将寄存器 TIM_CONR2 的 OPM 位置 1，则定时器工作在单脉冲模式下，在此种模式下，计数器计数到周期值，定时器输出一个单脉冲，然后停止输出。

这种模式允许计数器响应一个激励，并在一个程序可控的延时后产生一个脉宽可程序控制的脉冲。

将寄存器 TIM_CNTEN 的 CNTEN 位置 1 或通过触发信号触发，都可以使能计数器，定时器在输出比较模式或 PWM 模式下产生波形，若发生更新事件或软件清除计数器使能，则计数器停止计数，假如软件清除计数器使能导致计数器停止计数，则计数器的值保持不变，假如计数器在发生更新事件时停止，则计数器处于初始化状态。

在计数器启动时，只有比较值与计数器的初始值不同时，才能产生一个脉冲，计数器启

动前，配置的比较值（TIM_CCxR）需满足以下条件：

- 向上计数模式：TIM_CNT < TIM_CCxR < TIM_ARR
- 向下计数模式：TIM_CNT > TIM_CCxR

例 1：从 PWM0_CHx 输入脚上检测到一个上升沿开始，延时 t1（DELAY）后，在 OC1 上产生一个长度为 t2（PULSE）的正脉冲。

- 配置寄存器 TIM0_CC1MR 的 CC1MS 位为 10，选择 CC0 通道的输入信号作为 CC1 通道的触发信号。
- 配置寄存器 TIM0_CCPS 的 CC1P 位为 0，将输入信号的上升沿作为有效边沿。
- 配置寄存器 TIM0_TGICR0 的 TRGS 位为 110，选择 CH0INF_ED 作为从模式控制器的触发输入信号。
- 配置寄存器 TIM0_CC1R，TIM0_CC1R = t1（DELAY）。
- 配置寄存器 TIM0_ARR，TIM0_ARR = t1（DELAY）+ t2（PULSE）。
- 假设在比较匹配时需要产生由 0 到 1 的波形，计数器计到周期值时产生由 1 到 0 的波形，将寄存器 TIM0_CC1MR 的 OC1MS 位配置为 111（即 PWM 模式 2）；假设在比较匹配时需要产生由 1 到 0 的波形，计数器计到周期值时产生由 0 到 1 的波形，将寄存器 TIM0_CC1MR 的 OC1MS 位配置为 110（即 PWM 模式 1）。
- 将寄存器 TIM0_CONR2 的 OPM 位配置为 1，使能单脉冲模式。
- 配置寄存器 TIM0_TGICR0 的 SMS 位为 110，将从模式控制器配置为触发模式。

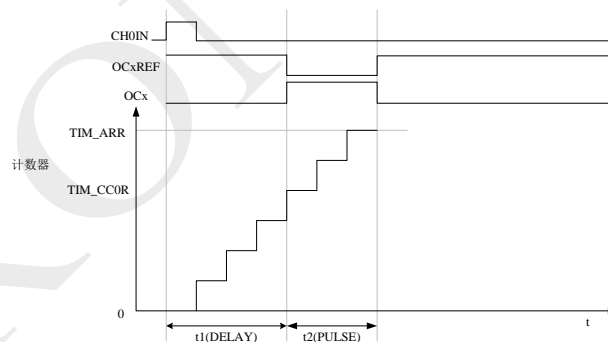


Figure 103 单脉冲模式的例子

11.5.5 主模式控制

定时器工作在主模式下，可产生触发信号 TRGO 去控制 AD 采样模块。Figure 104 为主模式控制器示意图。

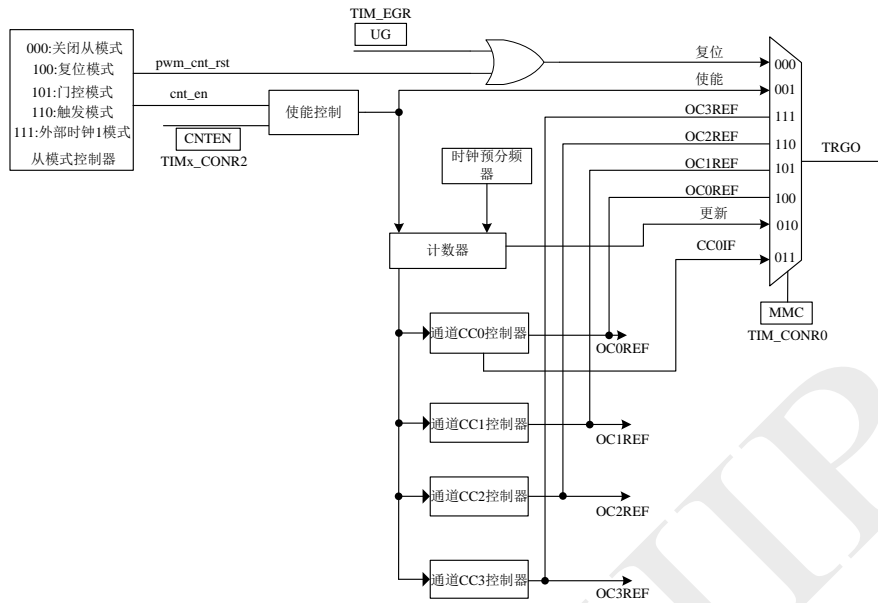


Figure 104 主模式控制示意图

主模式控制器的主要功能是选择定时器的相应信号作为触发输出 TRGO，控制过程如下：

- 配置寄存器 TIM_CONR0 的 MMC 位为 000，选择计数器的复位信号作为触发输出，复位信号包括由从模式控制器产生的复位信号和 TIM_EGR 寄存器的 UG 位。当 UG 为配置为 1 时，产生一个初始化信号去初始化寄存器，同时将此信号选为触发输出。
- 配置寄存器 TIM_CONR0 的 MMC 位为 001，选择计数器的使能信号作为触发输出 TIM_TRGO。
- 配置寄存器 TIM_CONR0 的 MMC 位域为 010，选择通道 CC0 的比较标志 CC0IF 作为触发输出 TIM_TRGO。若通道 CC0 配置为输入捕获功能，当发生一次捕获时，CC0IF 置位；若通道 CC0 配置为输出比较功能，当计数器的值和 TIM_CC0R 匹配时，CC0IF 置位。在这两种情况下都可以将 CC0IF 信号作为触发输出 TIM_TRGO。
- 配置寄存器 TIM_CONR0 的 MMC 位为 011，选择更新事件 UEV 作为触发输出。
- 配置寄存器 TIM_CONR0 的 MMC 位为 100，选择通道 CC0 的基准参考信号 OC0REF 作为触发输出。
- 配置寄存器 TIM_CONR0 的 MMC 位为 101，选择通道 CC1 的基准参考信号 OC1REF 作为触发输出。
- 配置寄存器 TIM_CONR0 的 MMC 位为 110，选择通道 CC2 的基准参考信号 OC2REF 作为触发输出。
- 配置寄存器 TIM_CONR0 的 MMC 位域为 111，选择通道 CC3 的基准参考信号 OC3REF 作为触发输出。

定时器发出的触发 AD 采样的 CC 事件有：

- 定时器 TIM 的触发输出 TIM_TRGO；
- 定时器 TIM 的通道 CC0 的主路输出 TIM_OC0；
- 定时器 TIM 的通道 CC1 的主路输出 TIM_OC1；
- 定时器 TIM 的通道 CC2 的主路输出 TIM_OC2；
- 定时器 TIM 的通道 CC3 的主路输出 TIM_OC3；

11.5.6 中断

11.5.6.1 中断源

高级定时器共有 17 个中断源，每个中断产生的条件如下：

1) 刹车中断。若已打开刹车中断使能，刹车中断产生的条件有：

- 刹车使能有效时外部输入刹车信号。
- 软件配置寄存器 TIM_EGR 的 BG 位为 1，产生软刹车事件。
- 产生比较器刹车信号。

2) 触发中断。若已打开触发中断使能，触发中断产生的条件有：

- 软件配置寄存器 TIM_EGR 的 TG 位为 1，产生触发事件。
- 在从模式控制器的触发输入端 TRGI 检测到有效边沿。

3) COM 中断。若已打开 COM 中断使能，COM 中断产生的条件有：

- 软件配置寄存器 TIM_EGR 的 COM 位为 1，产生 COM 事件。

4) 通道 CCx 的捕获/比较中断。若已打开通道 CCx 的捕获/比较中断使能，通道 CCx 的捕获/比较中断产生的条件有：

- 若通道 CCx 配置为输入捕获功能（寄存器 TIM_CCxMR 的 CCxMS 位配置为 01/10/11），软件配置寄存器 TIM_EGR 的 CCxG 位为 1，产生捕获事件。
- 若通道 CCx 配置为输入捕获功能（寄存器 TIM_CCxMR 的 CCxMS 位配置为 01/10/11），检测到通道 CCx 的捕获信号 CHxINCP 的有效边沿。
- 若通道 CCx 配置为输出比较功能（寄存器 TIM_CCxMR 的 CCxMS 位配置为 00），软件配置寄存器 TIM_EGR 的 CCxG 位为 1，产生比较事件。
- 若通道 CCx 配置为输出比较功能（寄存器 TIM_CCxMR 的 CCxMS 位配置为 00），计数器与比较值匹配时。

5) 计数器上溢更新中断。若已打开上溢更新中断使能，上溢更新中断产生的条件有：

- 若计数器的上溢事件没有被屏蔽（配置寄存器 TIM_CONR3 的 UPOUDIS 位为 0），当计数器上溢时产生计数器上溢更新中断。

6) 计数器下溢更新中断。若已打开下溢更新中断使能，下溢更新中断产生的条件

有：

- 若计数器的下溢事件没有被屏蔽（配置寄存器 TIM_CONR3 的 DOWNOUDIS 位为 0），当计数器下溢时产生计数器下溢更新中断。
- 7) 复位更新中断。若已打开通复位更新中断使能，复位更新中断产生的条件有：
 - 设置 TIM_EGR 寄存器的 UG 位为 1。
 - 从模式控制器产生的复位信号。

各个中断源产生后，在 TIM 中进行中断合并得到中断 PWM_INT，将 PWM_INT 送入 JMT51，在 JMT51 中经过一级门控后得到 JMT51 中的 TIM 中断，Figure 105 为中断产生过程。

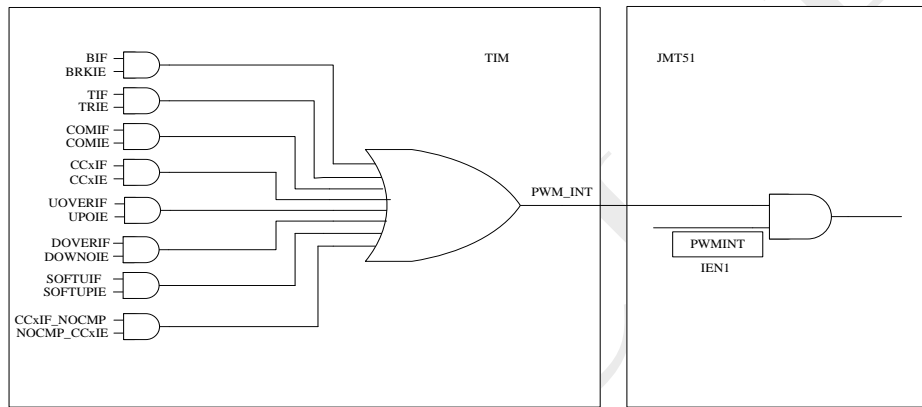


Figure 105 JMT1801ED 高级定时器中断

11.5.6.2 中断配置及处理

TIM 中断配置及处理过程如下：

- 配置中断使能寄存器 TIM_IER 和 TIM_CONR3 的中断使能位，使能相应的中断源。
- 配置寄存器 IEN1 的 PWMINT 位，使能高级定时器中断。
- 当产生 TIM 中断后，软件需要回读中断标志寄存器 TIM_ISR0、TIM_ISR1，来判断高级定时器的中断源，根据不同中断源，向中断标志寄存器相应位写 0 清除中断，并做相应处理。

11.6 寄存器描述

PWM 模块有 37 个寄存器，对应的 SFRPAGE 为 1，具体说明如 Table 162 所示。

Table 162 PWM 寄存器（SFRPAGE=0x01）

地址	寄存器名	属性	复位值	功能描述
0XD1	TIM_CONR0	W/R	0x00	TIM 控制寄存器 0
0XD2	TIM_CONR1	W/R	0x00	TIM 控制寄存器 1
0XD3	TIM_CONR2	W/R	0x00	TIM 控制寄存器 2

地址	寄存器名	属性	复位值	功能描述
0XD4	TIM_CONR3	W/R	0x20	TIM 控制寄存器 3
0XD5	TIM_TGICR0	W/R	0x00	从模式控制寄存器 0
0XD6	TIM_TGICR1	W/R	0x00	从模式控制寄存器 1
0XD7	TIM_IER	W/R	0x00	TIM 中断使能寄存器 0
0XD9	TIM_CC0MR	W/R	0x00	TIM 通道 CC0 输入捕获/输出比较模式控制寄存器
0XDA	TIM_CC1MR	W/R	0x00	TIM 通道 CC1 输入捕获/输出比较模式控制寄存器
0XDB	TIM_CC2MR	W/R	0x00	TIM 通道 CC2 输入捕获/输出比较模式控制寄存器
0XDC	TIM_CC3MR	W/R	0x00	TIM 通道 CC3 输入捕获/输出比较模式控制寄存器
0XDD	TIM_CCENR	W/R	0x00	通道使能控制寄存器
0XDE	TIM_CCPS	W/R	0x00	通道极性控制寄存器
0XDF	TIM_DTG	W/R	0x00	TIM 死区时间寄存器
0XE1	TIM_ARRL	W/R	0x00	TIM 周期值自动装载寄存器的低 8 位
0XE2	TIM_ARRH	W/R	0x00	TIM 周期值自动装载寄存器的高 8 位
0XE3	TIM_PSCL	W/R	0x00	TIM 预分频寄存器的低 8 位
0XE4	TIM_PSCH	W/R	0x00	TIM 预分频寄存器的高 8 位
0XE5	TIM_RCR	W/R	0x00	TIM 重复计数寄存器
0XE6	TIM_CC0RL	W/R	0x00	通道 CC0 捕获/比较寄存器的低 8 位
0XE7	TIM_CC0RH	W/R	0x00	通道 CC0 捕获/比较寄存器的高 8 位
0XE9	TIM_CC1RL	W/R	0x00	通道 CC1 捕获/比较寄存器的低 8 位
0XEA	TIM_CC1RH	W/R	0x00	通道 CC1 捕获/比较寄存器的高 8 位
0XEB	TIM_CC2RL	W/R	0x00	通道 CC2 捕获/比较寄存器的低 8 位
0XEC	TIM_CC2RH	W/R	0x00	通道 CC2 捕获/比较寄存器的高 8 位
0XED	TIM_CC3RL	W/R	0x00	通道 CC3 捕获/比较寄存器的低 8 位
0XEE	TIM_CC3RH	W/R	0x00	通道 CC3 捕获/比较寄存器的高 8 位
0XEF	TIM_BRKC	W/R	0x00	TIM 刹车控制寄存器

地址	寄存器名	属性	复位值	功能描述
0XF1	TIM_EGR	W	0x00	TIM 事件产生寄存器
0XF2	TIM_PHACON	W/R	0X00	相位偏移控制寄存器
0XF3	TIM_ISR0	W/R	0x00	TIM 中断标志寄存器 0
0XF4	TIM_ISR1	W/R	0x00	TIM 中断标志寄存器 1
0XF5	TIM_CNTL	W/R	0x00	TIM 计数器的低 8 位
0XF6	TIM_CNTH	W/R	0x00	TIM 计数器的高 8 位
0XF7	TIM_PHASEL	W/R	0x00	相位偏移长度的低 8 位
0XF9	TIM_PHASEH	W/R	0x00	相位偏移长度的高 8 位
0XFA	TIM_CNTEN	W/R	0x00	TIM 计数器使能寄存器

11.6.1 TIM 控制寄存器 0 (TIM_CONR0) [0xD1]

Table 163 TIM 控制寄存器 0 (TIM_CONR0)

参数名	比特位	属性	复位值	描述
CLKDIV	1-0	W/R	0	<p>时钟分频系数，定义了定时器时钟（pwm_clk）与死区及滤波时钟之间的分频系数。时钟 dts_clk 用于死区时间发生器以及数字滤波器。</p> <p>00: $T_{dts_clk} = T_{pwm_clk}$</p> <p>01: $T_{dts_clk} = 2 \times T_{pwm_clk}$</p> <p>10: $T_{dts_clk} = 4 \times T_{pwm_clk}$</p> <p>11: 保留。</p>
CHOINSEL	2	W/R	0	<p>通道 CC0 输入信号选择寄存器：</p> <p>0: PWM_CH0 管脚上的信号连接到通道 CC0</p> <p>1: PWM_CH0、PWM_CH1 和 PWM_CH2 管脚上的信号异或后连接到通道 CC0</p>
MMC	5-3	W/R	0	<p>主模式下输出信号选择寄存器，当定时器用作主定时器时，此寄存器用于选择送到从定时器的同步信号（TRGO）：</p> <p>000: 复位信号</p> <ul style="list-style-type: none"> ● 当 TIM_EGR 寄存器的 UG 置位

参数名	比特位	属性	复位值	描述
				<p>时，或由从模式控制器产生复位时，复位信号作为触发输出（TRGO）。</p> <p>001: 使能信号</p> <ul style="list-style-type: none"> ● 计数器使能信号被用于作为触发输出（TRGO）。 <p>若需要在同一时间启动多个定时器，或者要实现在一定的时间窗口内使能从定时器，那么这种模式将非常有用。</p> <p>当 CNTEN 控制位被置位，或当从模式下输入有效信号时，计数器使能信号将会被置位。</p> <p>当计数器使能信号受控于触发输入时，TRGO 比触发输入信号有一个延迟，除非选择了主/从模式，详述参看寄存器 TIM_TGICR0 的 MSM 位。</p> <p>010: 更新事件</p> <ul style="list-style-type: none"> ● 更新事件（上溢更新事件、下溢更新事件和软件更新事件）被选为触发输出（TRGO）。 <p>011: 通道 CC0 的捕获/比较脉冲</p> <ul style="list-style-type: none"> ● 一旦发生一次捕获或一次比较成功，送出一个正脉冲。 <p>100: 通道 CC0 的基准参考信号 OC0REF</p> <p>101: 通道 CC1 的基准参考信号 OC1REF</p> <p>110: 通道 CC2 的基准参考信号 OC2REF</p> <p>111: 通道 CC3 的基准参考信号 OC3REF</p>
CCUS	6	W/R	0	<p>捕获/比较控制位更新条件：</p> <p>0：如果捕获/比较控制位是预装载的（CCPE =1），只能通过设置 COM 位更新。</p> <p>1：如果捕获/比较控制位是预装载的（CCPE =1），可以通过设置 COM 位或 TRGI</p>

参数名	比特位	属性	复位值	描述
				的有效沿更新。
CCPE	7	W/R	0	捕获/比较控制位预装载使能控制寄存器： 0: 捕获/比较控制位 CCxE, CCxNE, CCxP, CCxNP 和 OCxMS 不是预装载的。 1: 捕获/比较控制位 CCxE, CCxNE, CCxP, CCxNP 和 OCxMS 位是预装载的，设置该位后，这些寄存器只在设置了 COM 位或 TRGI 的有效沿更新。

11.6.2 TIM 控制寄存器 1 (TIM_CONR1) [0xD2]

Table 164 TIM 控制寄存器 1 (TIM_CONR1)

参数名	比特位	属性	复位值	描述
OIS0	0	W/R	0	通道 CC0 输出的空闲状态 (OC0 输出状态): 0: 当 MOE=0 时 (空闲状态), 死区后 OC0=0。 1: 当 MOE=0 时, 死区后 OC0=1。 注: 若设置了 LOCK (TIM_BRKC 寄存器) 级别 1、2 或 3, 该位不能被修改。
OIS0N	1	W/R	0	通道 CC0 输出的空闲状态 (OC0N 输出状态): 0: 当 MOE=0 时, 死区后 OC0N=0; 1: 当 MOE=0 时, 死区后 OC0N=1; 注: 若设置了 LOCK (TIM_BKR 寄存器) 级别 1、2 或 3, 该位不能被修改。
OIS1	2	W/R	0	通道 CC1 输出的空闲状态 (OC1 输出状态), 参见 OIS0。
OIS1N	3	W/R	0	通道 CC1 输出的空闲状态 (OC1N 输出状态), 参见 OIS0N。
OIS2	4	W/R	0	通道 CC2 输出的空闲状态 (OC2 输出状态), 参见 OIS0。

参数名	比特位	属性	复位值	描述
OIS2N	5	W/R	0	通道 CC2 输出的空闲状态 (OC2N 输出状态), 参见 OIS0N。
OIS3	6	W/R	0	通道 CC3 输出的空闲状态 (OC3 输出状态), 参见 OIS0。
RESERVED	7	R	0	保留。

11.6.3 TIM 控制寄存器 2 (TIM_CONR2) [0xD3]

Table 165 TIM 控制寄存器 2 (TIM_CONR2)

参数名	比特位	属性	复位值	描述
ARPLE	0	W/R	0	计数器周期值预装载使能位: 0: 关闭 TIM_ARR 寄存器预装载功能。 1: 打开 TIM_ARR 寄存器预装载功能。
CNTMC	2-1	W/R	0	计数器模式控制: 00: 边沿对齐模式, 计数器依据方向位 CNTDIR 的配置向上或向下计数。 01: 中央对齐模式 1, 计数器交替向上向下计数, 只在计数器向下计数时更新输出通道的输出比较中断标志位。 10: 中央对齐模式 2, 计数器交替向上向下计数, 只在计数器向上计数时更新输出通道的输出比较中断标志位。 11: 中央对齐模式 3, 计数器交替向上向下计数, 在计数器向上计数和向下计数时更新输出通道的输出比较中断标志位。 注: 当计数器正在计数时, 不允许从边沿对齐模式转换到中央对齐模式。
CNTDIR	3	W/R	0	计数器计数方向: 0: 计数器向上计数。 1: 计数器向下计数。 注: 当计数器配置为中央对齐模式或编码器模式时, 该位只读。
OPM	4	W/R	0	单脉冲模式配置寄存器: 0: 不是单脉冲模式, 在发生更新事件时,

参数名	比特位	属性	复位值	描述
				计数器不停止。 1: 配置为单脉冲模式, 在发生下一次更新事件时, 计数器停止。
URC	5	W/R	0	更新源选择寄存器, 软件通过该位选择更新事件的源: 0: 下述任一事件产生一个更新事件, 如果更新中断使能, 则同时产生一个更新中断: --计数器上溢/下溢 --设置 TIM_EGR 寄存器的 UG 位 --从模式控制器产生的复位信号 1: 下述任一事件产生一个更新事件, 如果更新中断使能, 只有计数器上溢/下溢时产生一个更新中断: --计数器上溢/下溢 --设置 TIM_EGR 寄存器的 UG 位 --从模式控制器产生的复位信号。
RESERVED	7-6	W/R	0	保留。

11.6.4 TIM 控制寄存器 3 (TIM_CONR3) [0xD4]

Table 166 TIM 模式选择寄存器 3(TIM_CONR3)

参数名	比特位	属性	复位值	描述
UPOUDIS	0	W/R	0	上溢无效寄存器: 0: 当计数器上溢时产生更新事件。 1: 当计数器上溢时不产生更新事件。
DOWNOUDIS	1	W/R	0	下溢无效寄存器: 0: 当计数器下溢时产生更新事件。 1: 当计数器下溢时不产生更新事件。
UPOIE	2	W/R	0	上溢中断使能寄存器: 0: 当计数器上溢时不允许产生更新中断。 1: 当计数器上溢时允许产生更新

参数名	比特位	属性	复位值	描述
				中断。
DOWMOIE	3	W/R	0	下溢中断使能寄存器： 0：当计数器下溢时不允许产生更新中断。 1：当计数器下溢时允许产生更新中断。
SOFTUPIE	4	W/R	0	复位更新中断使能寄存器： 0：不允许产生复位更新中断。 1：当产生以下更新事件时允许产生更新中断： <ul style="list-style-type: none"> ● 设置 TIM_EGR 寄存器的 UG 位为 1。 ● 从模式控制器产生的复位信号。
LOCK	6-5	W/R	0	寄存器锁定设置，该位为防止软件错误配置而提供写保护。 00：锁定关闭，寄存器无写保护。 01：锁定级别 1，不能写入 TIM_DTG 寄存器、TIM_BRKC 的 BRKE、BRKP、AOE 位、TIM_CONR2 寄存器。 10：锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 TIM_CCPS 寄存器，不能写入 TIM_BRKC 的 ROSS 和 IOSS 位。 11：锁定级别 3，不能写入锁定级别 2 中的所有寄存器，不能写入 TIM_CCxMR 的 OCxMS 和 OCxPEN 位。 注：只有 LOCK 值为 0 时才可以配置为其他非 0 值，其内容冻结直到高级定时器被复位。

参数名	比特位	属性	复位值	描述
RESERVED	7	R	0	保留

11.6.5 从模式控制寄存器 0 (TIM_TGICR0) [0xD5]

Table 167 从模式控制寄存器 0(TIM_TGICR0)

参数名	比特位	属性	复位值	描述
RESERVED	0	R	0	保留
TRGS	3-1	W/R	0	从模式控制器触发信号选择寄存器： 000：通道 CC0 的边沿检测信号 CH0INFP_ED； 001：通道 CC0 滤波后的定时器输入 CH0INFP； 010：通道 CC1 滤波后的定时器输入 CH1INFP； 100：外部触发输入 PWM_ETR。 其他：保留
SMS	6-4	W/R	0	从模式选择寄存器： 000：关闭从模式 <ul style="list-style-type: none"> ● 通过配置计数器能寄存器 (TIM_CNTEN.CNTEN) 启动计数器。 001：编码器模式 1 <ul style="list-style-type: none"> ● 根据 CH0INFP0 的电平，计数器在 CH1INFP1 的边沿向上/向下计数。 010：编码器模式 2 <ul style="list-style-type: none"> ● 根据 CH1INFP1 的电平，计数器在 CH0INFP0 的边沿向上/向下计数。 011：编码器模式 3 <ul style="list-style-type: none"> ● 根据 CH1INFP1 和 CH0INFP0 的电平，计数器在 CH0INFP0 和 CH1INFP1 的边沿向上/向下计数。

参数名	比特位	属性	复位值	描述
				100: 复位模式 <ul style="list-style-type: none"> ● 选中的触发输入的有效沿重新初始化计数器,并且产生一个更新事件。 101: 门控模式 <ul style="list-style-type: none"> ● 当触发输入为有效电平时,计数器计数。一旦触发输入变为无效电平,计数器停止计数(但不复位)。 110: 触发模式 <ul style="list-style-type: none"> ● 计数器在触发输入的有效沿启动(但不复位),仅控制计数器启动。 111: 外部时钟模式 1 <ul style="list-style-type: none"> ● 选中的触发输入的有效沿驱动计数器计数。 注: 如果 CH1INFP1_ED 被选为触发输入时,不要使用门控模式,这是因为 CH1INFP1_ED 是一个脉冲信号,而门控模式是要检查触发输入的电平。
RESERVED	7	R	0	保留

11.6.6 从模式控制寄存器 1 (TIM_TGICR1) [0xD6]

Table 168 从模式控制寄存器 1 (TIM_TGICR1)

参数名	比特位	属性	复位值	描述
EXTP	0	W/R	0	外部触发信号极性选择: 0: PWM_ETR 不反相,高电平或上升沿有效。 1: PWM_ETR 被反相,低电平或下降沿有效。
EXTCLKEN	1	W/R	0	外部时钟模式 2 使能: 0: 禁止外部时钟模式 2 1: 启动外部时钟模式 2,计数器由

参数名	比特位	属性	复位值	描述
				<p>PWM_ETRF 信号的有效沿驱动。</p> <p>注 1: 设置 EXTCLKEN 位与选择外部时钟模式 1 并将 TRGI 连到 PWM_ETRF 具有相同功效。</p> <p>注 2: 下述从模式可以与外部时钟模式 2 同时使用: 复位模式、门控模式和触发模式; 但是这时 TRGI 不能连到 PWM_ETRF。</p>
EXTDIV	3-2	W/R	0	<p>外部触发信号 PWM_ETR 预分频:</p> <p>00: 1 分频</p> <p>01: 2 分频</p> <p>10: 4 分频</p> <p>11: 8 分频</p>
RESERVED	7-4	R	0	保留

11.6.7 TIM 中断使能寄存器 (TIM_IER) [0xD7]

Table 169 TIM 中断使能寄存器 (TIM_IER)

参数名	比特位	属性	复位值	描述
BRKIE	0	W/R	0	<p>刹车中断使能:</p> <p>0: 禁止刹车中断</p> <p>1: 允许刹车中断</p>
TRIE	1	W/R	0	<p>触发中断使能:</p> <p>0: 禁止触发中断</p> <p>1: 允许触发中断</p>
COMIE	2	W/R	0	<p>COM 中断使能:</p> <p>0: 禁止 COM 中断</p> <p>1: 允许 COM 中断</p>
CC0IE	3	W/R	0	<p>互补模式下通道 CC0 捕获/比较中断使能:</p> <p>0: 禁止捕获/比较中断</p> <p>1: 允许捕获/比较中断</p>
CC1IE	4	W/R	0	<p>互补模式下通道 CC1 捕获/比较中断使能:</p> <p>0: 禁止捕获/比较中断</p> <p>1: 允许捕获/比较中断</p>

参数名	比特位	属性	复位值	描述
CC2IE	5	W/R	0	互补模式下通道 CC2 捕获/比较中断使能： 0：禁止捕获/比较中断 1：允许捕获/比较中断
CC3IE	6	W/R	0	互补模式下通道 CC3 捕获/比较中断使能： 0：禁止捕获/比较中断 1：允许捕获/比较中断
RESERVED	7	R	0	保留

11.6.8 TIM 通道 CC0 输入捕获/输出比较模式控制寄存器 (TIM_CC0MR) [0xD9]

Table 170 TIM 通道 CC0 输入捕获/输出比较模式控制寄存器(输出比较模式)(TIM_CC0MR)

参数名	比特位	属性	复位值	描述
CC0MS	1-0	W/R	00	捕获/比较功能选择： 00：CC0 通道被配置为输出比较功能； 01：CC0 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自通道 CC0 的输入信号 CH0IN。 10：CC0 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自通道 CC1 的输入信号 CH1IN。 11：CC0 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自 TRC。 注：CC0MS 仅在通道关闭（寄存器 TIMx_CCENR 的 CC0EN=0）时才是可写的，因此在使用通道 CC0 时，应该先配置 CC0MS，再配置 CC0EN 为 1。
OC0FE	2	W/R	0	输出比较快速使能控制寄存器，该位用于加快 CC0 输出对触发输入事件的响应： 0：根据计数器与 CC0RU 的值，CC0 正常输出。当触发输入有一个有效沿时，激活 CC0 输出的最小延时为 5 个时钟周期。 1：输入信号有效沿的作用就像发生了一次比较匹配，此时 CC0 的输出被设置为

参数名	比特位	属性	复位值	描述
				<p>计数器与比较值匹配时的电平,而与比较结果无关。触发输入的有效沿和 CC0 输出间的延时被缩短为 3 个时钟周期。</p> <p>OC0FE 只在通道配置为 PWM 模式 1 或 PWM 模式 2 时起作用。</p>
OCOPEN	3	W/R	0	<p>通道 CC0 的输出比较寄存器的预装载使能:</p> <p>0: 禁止寄存器 TIM_CC0R 的预装载功能,可随时将数据写入寄存器 TIM_CC0R,且新值立即生效。</p> <p>1: 开启寄存器 TIM_CC0R 的预装载功能,读写操作仅对预装载寄存器操作, TIM_CC0R 的预装载值在更新事件产生时被载入影子寄存器。</p> <p>注 1: 一旦 LOCK 级别设为 3, 并且通道 CC0 被配置为输出比较模式时, 该位不能被修改。</p> <p>注 2: 若未使能输出比较寄存器的预装载功能, 当改变输出比较寄存器的值时, PWM 模式下可能会出现不规则的输出。单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OCOMS	6-4	W/R		<p>输出模式选择寄存器, 该位定义了输出参考信号 OC0REF 的动作, 而 OC0REF 决定了 OC0、OC0N 的输出, 从而决定输出管脚上的 PWM 输出波形。OC0REF 为高电平有效, 而 OC0 和 OC0N 的有效电平取决于 CC0P 和 CC0NP 位的配置。</p> <p>000: 冻结</p> <ul style="list-style-type: none"> ● 输出比较寄存器 TIM_CC0RU 与计数器间的比较结果对 OC0REF 不起作用。

参数名	比特位	属性	复位值	描述
				<p>001: 匹配时输出有效电平</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM_CC0R 与计数器的值相同时, OC0REF 输出有效电平 (即高电平)。 <p>010: 匹配时输出无效电平</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM_CC0R 与计数器的值相同时, OC0REF 输出无效电平 (即低电平)。 <p>011: 翻转</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM_CC0R 与计数器的值相同时, OC0REF 信号翻转。 <p>100: 强制输出无效电平</p> <ul style="list-style-type: none"> 当配置此种输出模式时, 立即将 OC0REF 设置为无效电平。 <p>101: 强制输出有效电平</p> <ul style="list-style-type: none"> 当配置此种输出模式时, 立即将 OC0REF 设置为有效电平。 <p>110: PWM 模式 1</p> <ul style="list-style-type: none"> 边沿计数模式下: 向上计数时, 若 $TIM_CNT < TIM_CC0R$, OC0REF 为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0); 向下计数时, 若 $TIM_CNT > TIM_CC0R$, OC0REF 为无效电平 (OC0REF=0), 否则为有效电平 (OC0REF=1)。 中央对齐计数模式下: 向上计数时, 若 $TIM_CNT < TIM_CC0R$, OC0REF 为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0); 向下计数时, 若 $TIM_CNT > TIM_CC0R$, OC0REF 为无效电平 (OC0REF=0),

参数名	比特位	属性	复位值	描述
				<p>否则为有效电平 (OC0REF=1)。</p> <p>111: PWM 模式 2</p> <ul style="list-style-type: none"> 边沿计数模式下: 向上计数时, 若 $TIM_CNT < TIM_CC0R$, OC0REF 为无效电平 (OC0REF=0), 否则为有效电平 (OC0REF=1); 向下计数时, 若 $TIM_CNT > TIM_CC0R$, OC0REF 为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0)。 中央对齐计数模式下: 向上计数时, 若 $TIM_CNT < TIM_CC0R$, OC0REF 为无效电平 (OC0REF=0), 否则为有效电平 (OC0REF=1); 向下计数时, 若 $TIM_CNT > TIM_CC0R$, OC0REF 为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0)。
OC0CE	7	W/R	0	<p>比较输出 OC0REF 清除控制:</p> <p>0: OC0REF 不受 PWM_ETRF 输入的影响。</p> <p>1: PWM_ETRF 输入有效电平时, OC0REF 输出无效电平。</p>

PWM 通道 CC0 输入捕获/输出比较模式控制寄存器 (输入捕获模式) (TIM_CC0MR)

参数名	比特位	属性	复位值	描述
CC0MS	1-0	W/R	00	<p>捕获/比较功能选择寄存器:</p> <p>00: CC0 通道被配置为输出比较功能;</p> <p>01: CC0 通道被配置为输入捕获功能, 输入捕获信号 CH0INC 来自通道 CC0 的输入信号 CH0IN。</p> <p>10: CC0 通道被配置为输入捕获功能, 输入捕获信号 CH0INC 来自通道 CC1 的输入信号 CH1IN。</p> <p>11: CC0 通道被配置为输入捕获功能, 输</p>

参数名	比特位	属性	复位值	描述
				入捕获信号 CH0INC 来自 TRC。 注：CC0MS 仅在通道关闭（寄存器 TIMx_CCENR 的 CC0EN=0）时才是可写的，因此在使用通道 CC0 时，应该先配置 CC0MS，再配置 CC0EN 为 1。
IC0PDIV	3-2	W/R	0	通道 CC0 输入捕获信号预分频控制寄存器，这两位定义了 CC0 输入信号的预分频系数。 00：不分频。 01：每 2 个事件触发一次捕获。 10：每 4 个事件触发一次捕获。 11：每 8 个事件触发一次捕获。
RESERVED	7-4	R	0	保留

11.6.9 TIM 通道 CC1 输入捕获/输出比较模式控制寄存器 (TIM_CC1MR) [0xDA]

Table 171 TIMx 通道 CC1 输入捕获/输出比较模式控制寄存器（输出比较模式）
(TIMx_CC1MR)

参数名	比特位	属性	复位值	描述
CC1MS	1-0	W/R	00	捕获/比较功能选择寄存器： 00：CC1 通道被配置为输出比较功能； 01：CC1 通道被配置为输入捕获功能，输入捕获信号 CH1INC 来自通道 CC1 的输入信号 CH1IN。 10：CC1 通道被配置为输入捕获功能，输入捕获信号 CH1INC 来自通道 CC0 的输入信号 CH0IN。 11：CC1 通道被配置为输入捕获功能，输入捕获信号 CH1INC 来自 TRC。 注：CC1MS 仅在通道关闭（寄存器 TIMx_CCENR 的 CC1E=0）时才是可写的，因此在使用通道 CC1 时，应该先配置 CC1MS，再配置 CC1E 为 1。

参数名	比特位	属性	复位值	描述
OC1FE	2	W/R	0	<p>输出比较快速使能控制寄存器, 该位用于加快 CC1 输出对触发输入事件的响应:</p> <p>0: 根据计数器与 CC1RU 的值, CC1 正常输出。当触发输入有一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。</p> <p>1: 输入信号有效沿的作用就像发生了一次比较匹配, 此时 CC1 的输出被设置为计数器与比较值匹配时的电平, 而与比较结果无关。触发输入的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。</p> <p>OC1FE 只在通道配置为 PWM 模式 1 或 PWM 模式 2 时起作用。</p>
OC1PEN	3	W/R	0	<p>输出比较寄存器的预装载使能控制寄存器:</p> <p>0: 禁止寄存器 TIM_CC1R 的预装载功能, 可随时将数据写入寄存器 TIMx_CC1R, 且新值立即生效。</p> <p>1: 开启寄存器 TIMx_CC1R 的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CC1R 的预装载值在更新事件产生时被载入影子寄存器中。</p> <p>注 1: 一旦 LOCK 级别设为 3, 并且通道被配置为输出比较模式时, 该位不能被修改。</p> <p>注 2: 若未使能输出比较寄存器的预装载功能, 当改变输出比较寄存器的值时, PWM 模式下可能会出现不规则的输出。单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC1MS	6-4	W/R		<p>输出模式选择寄存器, 该位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1、OC1N 的输出, 从而决定输出</p>

参数名	比特位	属性	复位值	描述
				<p>管脚上的 PWM 输出波形。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平取决于 CC1P 和 CC1NP 位的配置。</p> <p>000: 冻结。</p> <ul style="list-style-type: none"> ● 输出比较寄存器 TIM_CC1R 与计数器间的比较结果对 OC1REF 不起作用。 <p>001: 匹配时输出有效电平。</p> <ul style="list-style-type: none"> ● 当输出比较寄存器 TIM_CC1R 与计数器的值相同时，设置 OC1REF 为有效电平（即高电平）。 <p>010: 匹配时输出无效电平。</p> <ul style="list-style-type: none"> ● 当输出比较寄存器 TIM_CC1R 与计数器的值相同时，设置 OC1REF 为无效电平（即低电平）。 <p>011: 翻转。</p> <ul style="list-style-type: none"> ● 当输出比较寄存器 TIM_CC1R 与计数器的值相同时，OC1REF 信号翻转。 <p>100: 强制输出无效电平。</p> <ul style="list-style-type: none"> ● 当配置此种输出模式时，立即将 OC1REF 设置为无效电平。 <p>101: 强制输出有效电平。</p> <ul style="list-style-type: none"> ● 当配置此种输出模式时，立即将 OC1REF 设置为有效电平。 <p>110: PWM 模式 1</p> <ul style="list-style-type: none"> ● 边沿计数模式下：向上计数时，若 $TIM_CNT < TIM_CC1R$，OC1REF 为有效电平（OC1REF=1），否则为

参数名	比特位	属性	复位值	描述
				<p>无效电平 (OC1REF=0); 向下计数时, 若 $TIM_CNT > TIM_CC1R$, OC1REF 为无效电平 (OC1REF=0), 否则为有效电平 (OC1REF=1)。</p> <ul style="list-style-type: none"> 中央对齐计数模式下: 向上计数时, 若 $TIM_CNT < TIM_CC1R$, OC1REF 为有效电平 (OC1REF=1), 否则为无效电平 (OC1REF=0); 向下计数时, 若 $TIM_CNT > TIM_CC1R$, OC1REF 为无效电平 (OC1REF=0), 否则为有效电平 (OC1REF=1)。 <p>111: PWM 模式 2</p> <ul style="list-style-type: none"> 边沿计数模式下: 向上计数时, 若 $TIM_CNT < TIM_CC1R$, OC1REF 为无效电平 (OC1REF=0), 否则为有效电平 (OC1REF=1); 向下计数时, 若 $TIM_CNT > TIM_CC1R$, OC1REF 为有效电平 (OC1REF=1), 否则为无效电平 (OC1REF=0)。 中央对齐计数模式下: 向上计数时, 若 $TIM_CNT < TIM_CC1R$, OC1REF 为无效电平 (OC1REF=0), 否则为有效电平 (OC1REF=1); 向下计数时, 若 $TIM_CNT > TIM_CC1R$, OC1REF 为有效电平 (OC1REF=1), 否则为无效电平 (OC1REF=0)。
OC1CE	7	W/R	0	<p>比较输出 OC1REF 清除控制寄存器:</p> <p>0: OC1REF 不受 PWM_ETRF 输入的影响。</p> <p>1: PWM_ETRF 输入有效电平时, 设置 OC1REF 为无效电平。</p>

PWM 通道 CC1 输入捕获/输出比较模式控制寄存器（输入捕获模式）(TIM_CC1MR)

参数名	比特位	属性	复位值	描述
CC1MS	1-0	W/R	00	捕获/比较功能选择寄存器： 00：CC1 通道被配置为输出比较功能； 01：CC1 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自通道 CC1 的输入信号 CH1IN。 10：CC1 通道被配置为输入捕获功能，输入捕获信号 CH1INC 来自通道 CC0 的输入信号 CH0IN。 11：CC1 通道被配置为输入捕获功能，输入捕获信号 CH1INC 来自 TRC。 注：CC1MS 仅在通道关闭（寄存器 TIMx_CCENR 的 CC1E=0）时才是可写的，因此在使用通道 CC1 时，应该先配置 CC1MS，再配置 CC1E 为 1。
IC1PDIV	3-2	W/R	0	通道 CC1 输入捕获信号预分频控制寄存器，这两位定义了 CC1 输入信号的预分频系数。 00：不分频。 01：每 2 个事件触发一次捕获。 10：每 4 个事件触发一次捕获。 11：每 8 个事件触发一次捕获。
RESERVED	7-4	R	0	保留

11.6.10TIM 通道 CC2 输入捕获/输出比较模式控制寄存器 (TIM_CC2MR) [0xDB]

Table 172 TIM 通道 CC2 输入捕获/输出比较模式控制寄存器(输出比较模式)(TIM_CC2MR)

参数名	比特位	属性	复位值	描述
CC2MS	1-0	W/R	00	捕获/比较功能选择寄存器： 00：CC2 通道被配置为输出比较功能； 01：CC2 通道被配置为输入捕获功能，输入捕获信号 CH2INC 来自通道 CC2 的输入信号 CH2IN。

参数名	比特位	属性	复位值	描述
				<p>10: CC2 通道被配置为输入捕获功能, 输入捕获信号 CH2INC 来自通道 CC3 的输入信号 CH3IN。</p> <p>11: CC2 通道被配置为输入捕获功能, 输入捕获信号 CH2INC 来自 TRC。</p> <p>注: CC2MS 仅在通道关闭 (寄存器 TIMx_CCENR 的 CC2E=0) 时才是可写的, 因此在使用通道 CC2 时, 应该先配置 CC2MS, 再配置 CC2E 为 1。</p>
OC2FE	2	W/R	0	<p>输出比较快速使能控制寄存器, 该位用于加快 CC2 输出对触发输入事件的响应:</p> <p>0: 根据计数器与 TIM_CC2R 的值, CC2 正常输出。当触发输入有一个有效沿时, 激活 CC2 输出的最小延时为 5 个时钟周期。</p> <p>1: 输入信号有效沿的作用就像发生了一次比较匹配, 此时 CC2 的输出被设置为计数器与比较值匹配时的电平, 而与比较结果无关。触发输入的有效沿和 CC2 输出间的延时被缩短为 3 个时钟周期。</p> <p>OC2FE 只在通道配置为 PWM 模式 1 或 PWM 模式 2 时起作用。</p>
OC2PEN	3	W/R	0	<p>输出比较寄存器的预装载使能控制寄存器:</p> <p>0: 禁止寄存器 TIM_CC2R 的预装载功能, 可随时将数据写入寄存器 TIMx_CC2R, 且新值立即生效。</p> <p>1: 开启寄存器 TIMx_CC2R 的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CC2R 的预装载值在更新事件产生时被载入影子寄存器中。</p> <p>注 1: 一旦 LOCK 级别设为 3, 并且通道</p>

参数名	比特位	属性	复位值	描述
				被配置为输出比较模式时，该位不能被修改。 注 2：若未使能输出比较寄存器的预装载功能，当改变输出比较寄存器的值时，PWM 模式下可能会出现不规则的输出。单脉冲模式下的 PWM 输出不会出现这种情况。
OC2MS	6-4	W/R		输出模式选择寄存器，该位定义了输出参考信号 OC2REF 的动作，而 OC2REF 决定了 OC2、OC2N 的输出，从而决定输出管脚上的 PWM 输出波形。OC2REF 为高电平有效，而 OC2 和 OC2N 的有效电平取决于 CC2P 和 CC2NP 位的配置。 000：冻结。 <ul style="list-style-type: none"> ● 输出比较寄存器 TIM0_CC2R 与计数器间的比较结果对 OC2REF 不起作用。 001：匹配时输出有效电平 <ul style="list-style-type: none"> ● 当输出比较寄存器 TIMx_CC2R 与计数器的值相同时，设置 OC2REF 为有效电平（即高电平）。 010：匹配时输出无效电平 <ul style="list-style-type: none"> ● 当输出比较寄存器 TIMx_CC2R 与计数器的值相同时，设置 OC2REF 为无效电平（即低电平）。 011：翻转。 <ul style="list-style-type: none"> ● 当输出比较寄存器 TIMx_CC2R 与计数器的值相同时，OC2REF 信号翻转。 100：强制输出无效电平。

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> ● 当配置此种输出模式时，立即将 OC2REF 设置为无效电平。 <p>101: 强制输出有效电平。</p> <ul style="list-style-type: none"> ● 当配置此种输出模式时，立即将 OC2REF 设置为有效电平。 <p>110: PWM 模式 1</p> <ul style="list-style-type: none"> ● 边沿计数模式下：向上计数时，若 $TIM_CNT < TIMx_CC2R$，OC2REF 为有效电平（OC2REF=1），否则为无效电平（OC2REF=0）；向下计数时，若 $TIM_CNT > TIM_CC2R$，OC2REF 为无效电平（OC2REF=0），否则为有效电平（OC2REF=1）。 ● 中央对齐计数模式下：向上计数时，若 $TIM_CNT < TIM_CC2R$，OC2REF 为有效电平（OC2REF=1），否则为无效电平（OC2REF=0）；向下计数时，若 $TIM_CNT > TIM_CC2R$，OC2REF 为无效电平（OC2REF=0），否则为有效电平（OC2REF=1）。 <p>111: PWM 模式 2</p> <ul style="list-style-type: none"> ● 边沿计数模式下：向上计数时，若 $TIM_CNT < TIM_CC2R$，OC2REF 为无效电平（OC2REF=0），否则为有效电平（OC2REF=1）；向下计数时，若 $TIM_CNT > TIM_CC2R$，OC2REF 为有效电平（OC2REF=1），否则为无效电平（OC2REF=0）。 ● 中央对齐计数模式下：向上计数时，若 $TIM_CNT < TIM_CC2R$，OC2REF 为无效电平（OC2REF=0），否则为

参数名	比特位	属性	复位值	描述
				有效电平 (OC2REF=1); 向下计数时, 若 TIM_CNT > TIM_CC2R, OC2REF 为有效电平 (OC2REF=1), 否则为无效电平 (OC2REF=0)。
OC2CE	7	W/R	0	比较输出 OC2REF 清除控制寄存器: 0: OC2REF 不受 PWM_ETRF 输入的影响。 1: PWM_ETRF 输入有效电平时, 设置 OC2REF 为无效电平。

PWM 通道 CC2 输入捕获/输出比较模式控制寄存器 (输入捕获模式) (TIM_CC2MR)

参数名	比特位	属性	复位值	描述
CC2MS	1-0	W/R	00	捕获/比较功能选择寄存器: 00: CC2 通道被配置为输出比较功能; 01: CC2 通道被配置为输入捕获功能, 输入捕获信号 CH2INC 来自通道 CC2 的输入信号 CH2IN。 10: CC2 通道被配置为输入捕获功能, 输入捕获信号 CH2INC 来自通道 CC3 的输入信号 CH3IN。 11: CC2 通道被配置为输入捕获功能, 输入捕获信号 CH2INC 来自 TRC。 注: CC2MS 仅在通道关闭 (寄存器 TIM_CCENR 的 CC2E=0) 时才是可写的, 因此在使用通道 CC2 时, 应该先配置 CC2MS, 再配置 CC2E 为 1。
IC2PDIV	3-2	W/R	0	通道 CC2 输入捕获信号预分频控制寄存器, 这两位定义了 CC2 输入信号的预分频系数。 00: 不分频。 01: 每 2 个事件触发一次捕获。 10: 每 4 个事件触发一次捕获。 11: 每 8 个事件触发一次捕获。

参数名	比特位	属性	复位值	描述
RESERVED	7-4	R	0	保留

11.6.11TIM 通道 CC3 输入捕获/输出比较模式控制寄存器 (TIM_CC3MR) [0xDC]

Table 173 TIM 通道 CC3 输入捕获/输出比较模式控制寄存器(输出比较模式)(TIM_CC3MR)

参数名	比特位	属性	复位值	描述
CC3MS	1-0	W/R	00	<p>捕获/比较功能选择寄存器:</p> <p>00: CC3 通道被配置为输出比较功能;</p> <p>01: CC3 通道被配置为输入捕获功能, 输入捕获信号 CH3INC 来自通道 CC3 的输入信号 CH3IN。</p> <p>10: CC3 通道被配置为输入捕获功能, 输入捕获信号 CH3INC 来自通道 CC2 的输入信号 CH2IN。</p> <p>11: CC3 通道被配置为输入捕获功能, 输入捕获信号 CH3INC 来自 TRC。</p> <p>注: CC3MS 仅在通道关闭 (寄存器 TIM0_CCENR 的 CC3E=0) 时才是可写的, 因此在使用通道 CC3 时, 应该先配置 CC3MS, 再配置 CC3E 为 1。</p>
OC3FE	2	W/R	0	<p>输出比较快速使能控制寄存器, 该位用于加快 CC3 输出对触发输入事件的响应:</p> <p>0: 根据计数器与 CC3R 的值, CC3 正常输出。当触发输入有一个有效沿时, 激活 CC3 输出的最小延时为 5 个时钟周期。</p> <p>1: 输入信号有效沿的作用就像发生了一次比较匹配, 此时 CC3 的输出被设置为计数器与比较值匹配时的电平, 而与比较结果无关。触发输入的有效沿和 CC3 输出间的延时被缩短为 3 个时钟周期。</p> <p>OC3FE 只在通道配置为 PWM 模式 1 或 PWM 模式 2 时起作用。</p>
OC3PEN	3	W/R	0	输出比较寄存器的预装载使能控制寄存

参数名	比特位	属性	复位值	描述
				<p>器：</p> <p>0：禁止寄存器 TIMx_CC3R 的预装载功能，可随时将数据写入寄存器 TIMx_CC3R，且新值立即生效。</p> <p>1：开启寄存器 TIMx_CC3R 的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CC3R 的预装载值在更新事件产生时被载入影子寄存器中。</p> <p>注 1：一旦 LOCK 级别设为 3，并且通道被配置为输出比较模式时，该位不能被修改。</p> <p>注 2：若未使能输出比较寄存器的预装载功能，当改变输出比较寄存器的值时，PWM 模式下可能会出现不规则的输出。单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC3MS	6-4	W/R		<p>输出模式选择寄存器，该位定义了输出参考信号 OC3REF 的动作，而 OC3REF 决定了 OC3、OC3N 的输出，从而决定输出管脚上的 PWM 输出波形。OC3REF 为高电平有效，而 OC3 和 OC3N 的有效电平取决于 CC3P 和 CC3NP 位的配置。</p> <p>000：冻结。</p> <ul style="list-style-type: none"> ● 输出比较寄存器 TIM_CC3R 与计数器间的比较结果对 OC3REF 不起作用。 <p>001：匹配时输出有效电平</p> <ul style="list-style-type: none"> ● 当输出比较寄存器 TIM_CC3R 与计数器的值相同时，设置 OC3REF 为有效电平（即高电平）。 <p>010：匹配时输出无效电平</p>

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> 当输出比较寄存器 TIM_CC3R 与计数器的值相同时，设置 OC3REF 为无效电平（即低电平）。 <p>011: 翻转。</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM_CC3R 与计数器的值相同时，OC3REF 信号翻转。 <p>100: 强制输出无效电平。</p> <ul style="list-style-type: none"> 当配置此种输出模式时，立即将 OC3REF 设置为无效电平。 <p>101: 强制输出有效电平。</p> <ul style="list-style-type: none"> 当配置此种输出模式时，立即将 OC3REF 设置为有效电平。 <p>110: PWM 模式 1</p> <ul style="list-style-type: none"> 边沿计数模式下：向上计数时，若 $TIM_CNT < TIM_CC3R$，OC3REF 为有效电平（OC3REF=1），否则为无效电平（OC3REF=0）；向下计数时，若 $TIM_CNT > TIM_CC3R$，OC3REF 为无效电平（OC3REF=0），否则为有效电平（OC3REF=1）。 中央对齐计数模式下：向上计数时，若 $TIM_CNT < TIM_CC3R$，OC3REF 为有效电平（OC3REF=1），否则为无效电平（OC3REF=0）；向下计数时，若 $TIM_CNT > TIM_CC3R$，OC2REF 为无效电平（OC3REF=0），否则为有效电平（OC3REF=1）。 <p>111: PWM 模式 2</p> <ul style="list-style-type: none"> 边沿计数模式下：向上计数时，若

参数名	比特位	属性	复位值	描述
				<p>TIM_CNT < TIM_CC3R, OC3REF 为无效电平 (OC3REF=0), 否则为有效电平 (OC3REF=1); 向下计数时, 若 TIM_CNT > TIM_CC3R, OC3REF 为有效电平 (OC3REF=1), 否则为无效电平 (OC3REF=0)。</p> <ul style="list-style-type: none"> 中央对齐计数模式下: 向上计数时, 若 TIM_CNT < TIM_CC3R, OC3REF 为无效电平 (OC3REF=0), 否则为有效电平 (OC3REF=1); 向下计数时, 若 TIM_CNT > TIM_CC3R, OC3REF 为有效电平 (OC3REF=1), 否则为无效电平 (OC3REF=0)。
OC3CE	7	W/R	0	<p>比较输出 OC3REF 清除控制寄存器:</p> <p>0: OC3REF 不受 PWM_ETRF 输入的影响。</p> <p>1: PWM_ETRF 输入有效电平时, 设置 OC3REF 为无效电平。</p>

PWM 通道 CC3 输入捕获/输出比较模式控制寄存器 (输入捕获模式) (TIM_CC3MR)

参数名	比特位	属性	复位值	描述
CC3MS	1-0	W/R	00	<p>捕获/比较功能选择寄存器:</p> <p>00: CC3 通道被配置为输出比较功能;</p> <p>01: CC3 通道被配置为输入捕获功能, 输入捕获信号 CH3INC 来自通道 CC3 的输入信号 CH3IN。</p> <p>10: CC3 通道被配置为输入捕获功能, 输入捕获信号 CH3INC 来自通道 CC2 的输入信号 CH2IN。</p> <p>11: CC3 通道被配置为输入捕获功能, 输入捕获信号 CH3INC 来自 TRC。</p> <p>注: CC3MS 仅在通道关闭 (寄存器 TIM_CCENR 的 CC3E=0) 时才是可写的,</p>

参数名	比特位	属性	复位值	描述
				因此在使用通道 CC3 时，应该先配置 CC3MS，再配置 CC3E 为 1。
IC3PDIV	3-2	W/R	0	通道 CC3 输入捕获信号预分频控制寄存器，这两位定义了 CC3 输入信号的预分频系数。 00：不分频。 01：每 2 个事件触发一次捕获。 10：每 4 个事件触发一次捕获。 11：每 8 个事件触发一次捕获。
RESERVED	7-4	R	0	保留

11.6.12 通道使能控制寄存器 (TIM_CCENR) [0xDD]

Table 174 通道使能控制寄存器(TIM_CCENR)

参数名	比特位	属性	复位值	描述
CC0EN	0	W/R	0	通道 CC0 输入捕获/输出比较使能： CC0 通道配置为输出时： 0：关闭 ● OC0 禁止输出，输出引脚 PWM_CH0 的信号受寄存器 MOE、IOSS、ROSS、OIS0、OIS0N、CC0NEN、CC0P 位的控制。 1：开启 ● OC0 信号输出到对应的输出引脚 PWM_CH0，其输出受寄存器 MOE、OSSI、OSSR、OIS0、OIS0N、CC0NEN、CC0P 的控制。 CC0 通道配置为输入时： 0：捕获禁止。 1：捕获使能 ● 当捕获信号产生有效沿时，将计数器的值捕获到 TIM_CCOR 中。
CC0NEN	1	W/R	0	通道 CC0 互补路输出使能控制：

参数名	比特位	属性	复位值	描述
				0: 关闭 <ul style="list-style-type: none"> ● OC0N 禁止输出，输出引脚 PWM_CH0N 的信号受寄存器 MOE、IOSS、ROSS、OIS0、OIS0N、CC0EN、CC0NP 位的控制。 1: 开启 <ul style="list-style-type: none"> ● OC0N 信号输出到对应的输出引脚 PWM_CH0N，其输出受寄存器 MOE、OSSI、OSSR、OIS0、OIS0N、CC0EN、CC0NP 的控制。
CC1EN	2	W/R	0	通道 CC1 输入捕获/输出比较使能控制，参见 CC0EN 描述。
CC1NEN	3	W/R	0	通道 CC1 互补输出使能控制，参见 CC0NEN 描述。
CC2EN	4	W/R	0	通道 CC2 输入捕获/输出比较使能控制，参见 CC0EN 描述。
CC2NEN	5	W/R	0	通道 CC2 互补输出使能控制，参见 CC0NEN 描述。
CC3EN	6	W/R	0	通道 CC3 输入捕获/输出比较使能控制，参见 CC0EN 描述。
RESERVED	7	R	0	保留

11.6.13 通道极性控制寄存器 (TIM_CCPS) [0xDE]

Table 175 通道极性控制寄存器(TIM_CCPS)

参数名	比特位	属性	复位值	描述
CC0P	0	W/R	0	通道 CC0 的极性： <ul style="list-style-type: none"> ● 通道 CC0 配置为输出时： <ul style="list-style-type: none"> 0: OC0 (PWM_CH0) 的高电平为有效电平。 1: OC0 (PWM_CH0) 的低电平为有效电平。 ● CC0 通道配置为输入时：

参数名	比特位	属性	复位值	描述
				<p>该位选择是 CH0IN 还是 CH0IN 的反相信号作为触发或捕获信号。</p> <p>0: 不反相: CH0IN 的上升沿作为有效边沿, CH0IN 的高电平作为有效电平。</p> <p>1: 反相: CH0IN 的下沿作为有效边沿, CH0IN 的低电平作为有效电平。</p> <p>注: 一旦 LOCK 级别设置为 2 或 3, 该位不能被修改。</p>
CC0NP	1	W/R	0	<p>通道 CC0 互补输出极性:</p> <p>0: OC0N (PWM_CH0N) 的高电平为有效电平。</p> <p>1: OC0N (PWM_CH0N) 的低电平为有效电平。</p> <p>注: 一旦 LOCK 级别设置为 2 或 3 或 CC3MS =0 时, 则该位不能被修改。</p>
CC1P	2	W/R	0	通道 CC1 极性, 参见 CC0P 描述。
CC1NP	3	W/R	0	通道 CC1 互补输出极性控制寄存器, 参见 CC0NP 描述。
CC2P	4	W/R	0	通道 CC2 极性, 参见 CC0P 描述。
CC2NP	5	W/R	0	通道 CC2 互补输出极性, 参见 CC0NP 描述。
CC3P	6	W/R	0	通道 CC3 极性, 参见 CC0P 描述。
RESERVED	7	R	0	保留

11.6.14TIM 死区时间寄存器 (TIM_DTG) [0xDF]

Table 176 TIM 死区时间寄存器(TIM_DTG)

参数名	比特位	属性	复位值	描述
DTG	7-0	W/R	0	<p>死区发生器死区时间设置寄存器, 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间:</p> <p>若 $DTG[7:5] = "0xx"$: $DT = DTG[6:0] \times T_{dts_clk}$</p> <p>若 $DTG[7:5] = "10x"$: $DT = (64 + DTG[5:0]) \times 2 \times T_{dts_clk}$</p>

参数名	比特位	属性	复位值	描述
				若 $DTG[7:5] = "110"$: $DT = (32 + DTG[4:0]) \times 8 \times T_{dts_clk}$ 若 $DTG[7:5] = "111"$: $DT = (32 + DTG[4:0]) \times 16 \times T_{dts_clk}$

11.6.15 TIM 周期值自动装载寄存器的低 8 位 (TIM_ARRL) [0xE1]

Table 177 TIM 周期值自动装载寄存器的低 8 位 (TIM_ARRL)

参数名	比特位	属性	复位值	描述
ARRL	7-0	W/R	0	周期值自动重装载值的低 8 位。若寄存器 TIM_CONR2 的 ARPLE 位配置为 1, 寄存器的预装载功能有效, 当产生更新事件时, 该寄存器的值被装载到影子寄存器, 更新事件包括: <ul style="list-style-type: none"> ● 计数器上溢/下溢 ● 配置寄存器 TIM_EGR 的 UG 位为 1 ● 从模式控制器产生的复位信号 当自动重装载的值为 0 时, 计数器不工作。

11.6.16 TIM 周期值自动装载寄存器的高 8 位 (TIM_ARRH) [0xE2]

Table 178 TIM 周期值自动装载寄存器的高 8 位 (TIM_ARRH)

参数名	比特位	属性	复位值	描述
ARRH	7-0	W/R	0	周期值自动重装载值的高 8 位。若寄存器 TIM_CONR2 的 ARPLE 配置为 1, 寄存器的预装载功能有效, 当产生更新事件时, 该寄存器的值被装载到影子寄存器, 更新事件包括: <ul style="list-style-type: none"> ● 计数器上溢/下溢 ● 配置寄存器 TIM_EGR 的 UG 位为 1 ● 从模式控制器产生的复位信号 当自动重装载的值为 0 时, 计数器不工作。

11.6.17 TIM 预分频寄存器的低 8 位 (TIM_PSCL) [0xE3]

Table 179 TIM 预分频寄存器的低 8 位 (TIM_PSCL)

参数名	比特位	属性	复位值	描述
CNTPSCL	7-0	W/R	0	<p>预分频寄存器的低 8 位, 该寄存器包含了当更新事件产生时装入预分频影子寄存器的值, 该寄存器的预装载功能始终有效, 更新事件包括:</p> <ul style="list-style-type: none"> ● 计数器上溢/下溢 ● 设置 UG 位为 1 ● 从模式控制器产生的复位信号

11.6.18TIM 预分频寄存器的高 8 位 (TIM_PSCH) [0xE4]

Table 180 TIMx 预分频寄存器的高 8 位 (TIM_PSCH)

参数名	比特位	属性	复位值	描述
CNTPSCH	7-0	W/R	0	<p>预分频器值的高 8 位, 该寄存器包含了当更新事件产生时装入预分频影子寄存器的值, 该寄存器的预装载功能始终有效, 更新事件包括:</p> <ul style="list-style-type: none"> ● 计数器上溢/下溢 ● 设置 UG 位为 1 ● 从模式控制器产生的复位信号

11.6.19TIM 重复计数寄存器 (TIM_RCR) [0xE5]

Table 181 TIM 重复计数寄存器(TIM_RCR)

参数名	比特位	属性	复位值	描述
RCR	7-0	W/R	0	<p>周期计数次数的值, 开启了预装载功能后, 这些位用于设置比较寄存器的更新速率, 如果允许产生更新中断, 则会同时影响产生更新中断的速率。</p> <p>每次向下计数器 REP_CNT 达到 0, 会产生一个更新事件并且计数器 REP_CNT 重新从 RCR 开始计数。由于 REP_CNT 只有在周期更新事件发生时才重载重复值, 因此对此寄存器写入的新值只在下次更新事件发生时才起作用。</p>

参数名	比特位	属性	复位值	描述
				这意味着在 PWM 模式下, RCR +1 对应着: --在边沿对齐模式下, PWM 周期的数目。 --在中央对齐模式下, PWM 半周期的数目。

11.6.20 通道 CC0 捕获/比较寄存器的低 8 位 (TIM_CC0RL) [0xE6]

Table 182 通道 CC0 捕获/比较寄存器的低 8 位(TIM_CC0RL)

参数名	比特位	属性	复位值	描述
CC0RL	7-0	W/R	0	通道 CC0 捕获/向上计数比较寄存器的低 8 位: <ul style="list-style-type: none"> ● 若 CC0 通道配置为输出: 此寄存器包含了装入比较影子寄存器的值 (预装载值)。 如果未选择预装载功能, 其立即被装入捕获/比较影子寄存器。否则, 只有当更新事件发生时, 此预装载值才被装载入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值, 根据比较结果在 OC0 端口上输出信号。 ● 若 CC0 通道配置为输入: 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

11.6.21 通道 CC0 捕获/比较寄存器的高 8 位 (TIM_CC0RH) [0xE7]

Table 183 通道 CC0 捕获/比较寄存器的高 8 位(TIM_CC0RH)

参数名	比特位	属性	复位值	描述
CC0RH	7-0	W/R	0	通道 CC0 捕获/向上计数比较寄存器的高 8 位: <ul style="list-style-type: none"> ● 若 CC0 通道配置为输出: 此寄存器包含了装入比较影子寄存器的值 (预装载值)。 如果未选择预装载功能, 其立即被装入捕获/比较影子寄存器中。否则, 只

参数名	比特位	属性	复位值	描述
				<p>有当更新事件发生时,此预装载值才被装载入捕获/比较影子寄存器中。影子寄存器包含了与计数器比较的值,根据比较结果在 OC0 端口上输出信号。</p> <ul style="list-style-type: none"> ● 若 CC0 通道配置为输入: 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的高 8 位。

11.6.22 通道 CC1 捕获/比较寄存器的低 8 位 (TIM_CC1RL) [0xE9]

Table 184 通道 CC1 捕获/比较寄存器的低 8 位(TIM_CC1RL)

参数名	比特位	属性	复位值	描述
CC1RL	7-0	W/R	0	<p>通道 CC1 捕获寄存器/向上计数比较寄存器的低 8 位:</p> <ul style="list-style-type: none"> ● 若 CC1 通道配置为输出: 此寄存器包含了装入比较影子寄存器的值 (预装载值)。 如果未选择预装载功能,其立即被装入捕获/比较影子寄存器。否则,只有当更新事件发生时,此预装载值才被装载入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值,根据比较结果在 OC1 端口上输出信号。 ● 若 CC1 通道配置为输入: 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

11.6.23 通道 CC1 捕获/比较寄存器的高 8 位 (TIM_CC1RH) [0xEA]

Table 185 通道 CC1 捕获/比较寄存器的高 8 位(TIM_CC1RH)

参数名	比特位	属性	复位值	描述
CC1RH	7-0	W/R	0	<p>通道 CC1 捕获寄存器/向上计数比较寄存器的高 8 位:</p> <ul style="list-style-type: none"> ● 若 CC1 通道配置为输出: 此寄存器包含了装入比较影子寄存

参数名	比特位	属性	复位值	描述
				器的值（预装载值）。 <p>如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC1 端口上输出信号。</p> <ul style="list-style-type: none"> ● 若 CC1 通道配置为输入： 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

11.6.24 通道 CC2 捕获/比较寄存器的低 8 位 (TIM_CC2RL) [0xEB]

Table 186 通道 CC2 捕获/比较寄存器的低 8 位(TIM_CC2RL)

参数名	比特位	属性	复位值	描述
CC2RL	7-0	W/R	0	通道 CC2 捕获寄存器/向上计数比较寄存器的低 8 位： <ul style="list-style-type: none"> ● 若 CC2 通道配置为输出： 此寄存器包含了装入比较影子寄存器的值（预装载值）。 如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC2 端口上输出信号。 ● 若 CC2 通道配置为输入： 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

11.6.25 通道 CC2 捕获/比较寄存器的高 8 位 (TIM_CC2RH) [0xEC]

Table 187 通道 CC2 捕获/比较寄存器的高 8 位(TIM_CC2RH)

参数名	比特位	属性	复位值	描述
CC2RH	7-0	W/R	0	通道 CC2 捕获寄存器/向上计数比较寄存

参数名	比特位	属性	复位值	描述
				器的高 8 位： <ul style="list-style-type: none"> ● 若 CC2 通道配置为输出： <p>此寄存器包含了装入比较影子寄存器的值（预装载值）。</p> <p>如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC2 端口上输出信号。</p> ● 若 CC2 通道配置为输入： <p>此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。</p>

11.6.26 通道 CC3 捕获/比较寄存器的低 8 位 (TIM_CC3RL) [0xED]

Table 188 通道 CC3 捕获/比较寄存器的低 8 位(TIM_CC3RL)

参数名	比特位	属性	复位值	描述
CC3RL	7-0	W/R	0	通道 CC3 捕获寄存器/向上计数比较寄存器的低 8 位： <ul style="list-style-type: none"> ● 若 CC3 通道配置为输出： <p>此寄存器包含了装入比较影子寄存器的值（预装载值）。</p> <p>如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC3 端口上输出信号。</p> ● 若 CC3 通道配置为输入： <p>此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。</p>

11.6.27 通道 CC3 捕获/比较寄存器的高 8 位 (TIM_CC3RH) [0xEE]

Table 189 通道 CC3 捕获/比较寄存器的高 8 位(TIM_CC3RH)

参数名	比特位	属性	复位值	描述
CC3RH	7-0	W/R	0	通道 CC3 捕获寄存器/向上计数比较寄存器的高 8 位： <ul style="list-style-type: none"> ● 若 CC3 通道配置为输出： <p>此寄存器包含了装入比较影子寄存器的值（预装载值）。</p> <p>如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC3 端口上输出信号。</p> ● 若 CC3 通道配置为输入： <p>此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。</p>

11.6.28 TIM 刹车控制寄存器 (TIM_BRKC) [0xEF]

Table 190 TIM 刹车控制寄存器(TIM_BRKC)

参数名	比特位	属性	复位值	描述
BRKZEN	0	W/R	0	刹车时输出高阻使能寄存器： <p>0：刹车时不输出高阻；</p> <p>1：刹车时输出高阻。</p>
IOSS	1	W/R	0	空闲模式（MOE=0）下“关闭状态”（CCxEN=0 或 CCxNEN=0）输出控制，该位用于当 MOE=0 且通道设置为输出时。 <p>0：当定时器 CCx 通道处于关闭状态时（CCxEN=0 或 CCxNEN=0），输出脚 PWM_CHx 或 PWM_CHxN 的输出使能 PWM_CHxEN 和 PWM_CHxNEN 无效，PWM_CHx 或 PWM_CHxN 输出高阻。详见 Table 161 中 MOE=0 且 IOSS=0 的部分。</p> <p>1：当定时器 CCx 通道处于关闭状态时</p>

参数名	比特位	属性	复位值	描述
				<p>(CCxEN=0 或 CCxNEN=0)，输出脚 PWM_CHx 和 PWM_CHxN 的输出使能 PWM_CHxEN 和 PWM_CHxNEN 有效，PWM_CHx 和 PWM_CHxN 输出其空闲电平，详见 Table 161 中 MOE=0 且 IOSS=1 的部分。</p> <p>注：一旦 LOCK 级别设置为 2 或 3，则该位不能被修改。</p>
ROSS	2	W/R	0	<p>运行模式 (MOE=1) 下“关闭状态” (CCxEN=0 或 CCxNEN=0) 输出控制，该位用于当 MOE=1 且通道位互补输出时。没有互补输出的定时器没有 OSSR 位。</p> <p>0：当定时器 CCx 通道处于关闭状态时 (CCxEN=0 或 CCxNEN=0)，输出脚 PWM_CHx 或 PWM_CHxN 的输出使能 PWM_CHxEN 和 PWM_CHxNEN 无效，PWM_CHx 或 PWM_CHxN 输出高阻。详见 Table 161 中 MOE=1 且 ROSS=0 的部分。</p> <p>1：当定时器 CCx 通道处于关闭状态时 (CCxEN=0 或 CCxNEN=0)，输出脚 PWM_CHx 和 PWM_CHxN 的输出使能 PWM_CHxEN 和 PWM_CHxNEN 有效，PWM_CHx 和 PWM_CHxN 输出其空闲电平，详见 Table 161 中 MOE=1 且 ROSS=1 的部分。</p> <p>注：一旦 LOCK 级别设置为 2，则该位不能被修改。</p>
CMPBRKE	4	W/R	0	<p>比较器刹车功能使能寄存器：</p> <p>0：禁止刹车输入</p> <p>1：开启刹车输入</p> <p>注 1：一旦 LOCK 级别设置为 1、2 或 3，该位不能修改。</p>

参数名	比特位	属性	复位值	描述
				注 2: 此位只控制外部输入的刹车信号, 对寄存器配置的刹车事件 BG 无效。 注 3: 通过配置 17.3 节的寄存器 BKPRSEL 可以改变比较器器刹车信号的极性。
EXTBRKE	4	W/R	0	外部刹车功能使能寄存器: 0: 禁止刹车输入 1: 开启刹车输入 注 1: 一旦 LOCK 级别设置为 1、2 或 3, 该位不能修改。 注 2: 此位只控制外部输入的刹车信号, 对寄存器配置的刹车事件 BG 无效。
BRKP	5	W/R	0	刹车输入极性: 0: 刹车输入高电平为有效电平 1: 刹车输入低电平为有效电平 注: 一旦 LOCK 级别设置为 1、2 或 3, 该位不能修改。
AOE	6	W/R	0	自动输出使能: 0: MOE 只能被软件置 1; 1: MOE 能被软件置 1 或在刹车信号无效后的更新事件自动置 1。 注: 一旦 LOCK 级别设置为 1、2 或 3, 该位不能修改。
MOE	7	W/R	0	主输出使能寄存器, 一旦刹车输入有效, 该位被硬件异步清 0, 根据 AOE 的设置, 可由软件置 1 或自动置 1, 它仅对配置为输出通道有效。 0: 使定时器处于空闲状态。定时器的输出受 IOSS、OIS0、OIS0N、CC0EN、CC0EN、CC0P 和 CC0NP 位的控制。 1: 使定时器处于运行状态。定时器的输出受 ROSS、CC0EN、CC0EN、CC0P 和 CC0NP 位的控制。

11.6.29TIM 事件产生寄存器 (TIM_EGR) [0xF1]

Table 191 TIM 事件产生寄存器(TIM_EGR)

参数名	比特位	属性	复位值	描述
BG	0	W	0	产生刹车事件, 该位由软件置 1, 硬件自动清 0: 0: 无动作; 1: 产生一个刹车事件。此时 MOE=0, BIF=1, 若打开刹车中断使能, 则产生刹车中断。
TG	1	W	0	产生触发事件, 该位由软件置 1, 硬件自动清 0: 0: 无动作; 1: 产生一个触发事件。此时 TIF=1, 若打开了触发中断使能, 则产生触发中断。
COMG	2	W	0	产生寄存器 TIM_CCENR、TIM_CCPS 和 TIM_CCxMR.OCxMS 的更新事件: 0: 无动作; 1: 当 CCxPE=1 时, 更新 TIM_CCENR、TIM_CCPS 和 TIM_CCxMR.OCxMS。
CC0G	3	W	0	产生捕获/比较事件, 该位由软件置 1, 用于产生一个捕获/比较事件, 由硬件自动清 0: 0: 无动作; 1: 在通道 CC0 上产生一个捕获/比较事件 若通道 CC0 配置位输出: 设置 CC0IF=1, 若打开比较中断使能, 则产生比较中断。 若通道 CC0 配置为输入: 当前的计数器值捕获至 TIM_CC0R 寄存器, 捕获中断标志位 CC0IF 自动置 1, 若打开捕获中断使能, 则产生捕获中断, 若 CC0IF 已经为 1, 则过捕获标志位 CC0OIF 自动置 1。
CC1G	4	W	0	参见 CC0G 描述。
CC2G	5	W	0	参见 CC0G 描述。

参数名	比特位	属性	复位值	描述
CC3G	6	W	0	参见 CCOG 描述。
UG	7	W	0	产生更新事件，该位由软件置 1，由硬件自动清 0： 0：无动作； 1：初始化计数器、初始化预分频计数器、产生一个更新事件： <ul style="list-style-type: none"> ● 若在中央对齐模式或向上计数模式下，计数器被清 0； ● 若在向下计数模式下，计数器初始化为 TIM_ARR 的值； ● 初始化预分频计数器为 0。

11.6.30 相位偏移控制寄存器(TIM_PHACON) [0XF2]

Table 192 相位偏移控制寄存器(TIM_PHACON)

参数名	比特位	属性	复位值	描述
PHASEEN	0	W/R	0	不同通道间 PWM 输出相位偏移使能寄存器： 0：不同通道间 PWM 输出无相位偏移； 1：打开通道间 PWM 输出相位偏移功能。
PHASECH	2-1	W/R	00	具有相位偏移的通道选择寄存器： 00：通道 CC0 的 PWM 输出具有相位偏移； 01：通道 CC1 的 PWM 输出具有相位偏移； 10：通道 CC2 的 PWM 输出具有相位偏移； 11：通道 CC3 的 PWM 输出具有相位偏移。
RESERVED	7-3	R	0	保留。

11.6.31 TIM 中断标志寄存器 1 (TIM_ISR0) [0xF3]

Table 193 TIM 中断标志寄存器 0(TIM_ISR0)

参数名	比特位	属性	复位值	描述
BIF	0	W/R	0	刹车中断标志。一旦发生刹车事件，由硬件将该位置 1。刹车输入无效时，该位可由软件清 0。 0：无刹车事件产生；

参数名	比特位	属性	复位值	描述
				1: 产生刹车事件。
TIF	1	W/R	0	触发中断标志, 当发生触发事件时 (当在 TRGI 输入端检测到有效边沿或软件配置寄存器 TIM_EGR 的 TG 位) 由硬件将该位置 1, 由软件清 0: 0: 无触发事件产生 1: 有触发事件产生
COMIF	2	W/R	0	COM 中断标志, 一旦 COM 事件来到, 该位由硬件置 1, 由软件清 0: 0: 无 COM 事件产生 1: 有 COM 事件产生
CC0IF	3	W/R	0	互补模式下, 通道 CC0 的捕获/比较中断标志: ● 如果通道 CC0 配置为输出模式: 当计数器值与比较值匹配时由硬件置 1, 由软件清 0 0: 无匹配发生 1: 计数器与比较值匹配 ● 如果通道 CC0 配置为输入捕获模式: 当捕获事件发生时该位由硬件置 1, 由软件清 0 或通过读 TIM_CC0R 清 0 0: 无输入捕获产生 1: 输入捕获产生
CC1IF	4	W/R	0	通道 CC1 捕获/比较中断标志寄存器, 参见 CC0IF 描述
CC2IF	5	W/R	0	通道 CC2 捕获/比较中断标志寄存器, 参见 CC0IF 描述
CC3IF	6	W/R	0	通道 CC3 捕获/比较中断标志寄存器, 参见 CC0IF 描述
RESERVED	7	R	0	保留。

11.6.32 TIM 中断标志寄存器 2 (TIM_ISR1) [0xF4]

Table 194 TIM 中断标志寄存器 1(TIM_ISR1)

参数名	比特位	属性	复位值	描述
CC0OIF	0	W/R	0	通道 CC0 过捕获标志，仅当该通道被配置为输入捕获时，该标志可由硬件置 1，由软件清 0 0: 无过捕获发生； 1: 捕获信号产生时已发生捕获。
CC1OIF	1	W/R	0	通道 CC1 过捕获标志，参见 CC0OIF 描述
CC2OIF	2	W/R	0	通道 CC2 过捕获标志，参见 CC0OIF 描述
CC3OIF	3	W/R	0	通道 CC3 过捕获标志，参见 CC0OIF 描述
UOVERIF	4	W/R	0	计数器上溢更新中断标志： 0: 没有上溢事件发生； 1: 发生上溢事件。
DOVERIF	5	W	0	计数器下溢更新中断标志： 0: 没有下溢事件发生； 1: 发生下溢事件。
SOFTUIF	6	W/R	0	复位（配置 TIM_EGR 的 UG 位或复位模式下的从模式控制器产生的复位信号）更新中断标志： 0: 没有软件或复位更新事件发生； 1: 发生软件或复位更新事件。
RESERVED	7	R	0	保留。

11.6.33 TIM 计数器的低 8 位 (TIM_CNTL) [0xF5]

Table 195 TIM 计数器的低 8 位 (TIM_CNTL)

参数名	比特位	属性	复位值	描述
CNTL	7-0	W/R	0	计数器的低 8 位。

11.6.34 TIM 计数器的高 8 位 (TIM_CNTH) [0xF6]

Table 196 TIM 计数器的高 8 位(TIM_CNTH)

参数名	比特位	属性	复位值	描述
CNTH	7-0	W/R	0	计数器的高 8 位。

11.6.35 相位偏移长度寄存器的低 8 位(TIM_PHASEL) [0XF7]

Table 197 相位偏移长度寄存器的低 8 位(TIM_PHASEL)

参数名	比特位	属性	复位值	描述
PHASEL	7-0	W/R	0	相位偏移长度寄存器的低 8 位。

11.6.36 相位偏移长度寄存器的高 8 位(TIM_PHASEH) [0XF9]

Table 198 相位偏移长度寄存器的高 8 位(TIM_PHASEH)

参数名	比特位	属性	复位值	描述
PHASEH	7-0	W/R	0	相位偏移长度寄存器的高 8 位。

11.6.37 TIM 计数器使能寄存器(TIM_CNTEN) [0xFA]

Table 199 TIM 计数器使能寄存器 (TIM_CNTEN)

参数名	比特位	属性	复位值	描述
CNTEN	0	W/R	0	计数器使能软件控制位： 0：计数器不计数。 1：计数器计数。 注：触发模式下不需要配置，硬件自动设置 CNTEN 位。
RESERVED	7-1	R	0	保留。

12 CORDIC 协处理器

12.1 概述

CORDIC (Coordinate Rotation Digital Computer) 算法即坐标旋转数字计算方法, 主要用于三角函数、双曲线、指数、对数的计算。该算法通过基本的加和移位迭代运算代替乘法运算, 使得矢量的旋转和定向的计算不再需要三角函数、乘法、开方、反三角、指数等函数。

JMT1801ED CORDIC算法支持圆周系统中的两种计算模式: 旋转模式和向量模式。其中旋转模式用于计算任意角度的正余弦 (COS/SIN); 向量模式用于计算任意向量的反正切值 (\tan^{-1}) 以及该向量的模 $P*|a|$ 。

CORDIC算法在圆周系统中的迭代公式为:

$$\begin{aligned} X_{n+1} &= X_n - S_n Y_n 2^{-n} \\ Y_{n+1} &= Y_n + S_n X_n 2^{-n} \\ Z_{n+1} &= Z_n - \tan^{-1}(2^{-n}) \end{aligned}$$

在旋转模式中, CORDIC经过N次迭代后, 输出变为:

$$\begin{aligned} X_N &= P[X_0 \cos(Z_0) - Y_0 \sin(Z_0)] \\ Y_N &= P[X_0 \sin(Z_0) + Y_0 \cos(Z_0)] \\ Z_N &= 0 \end{aligned}$$

设定 $X_0 = 1/P$, $Y_0 = 0$, 其中 P 为一常数, 那么N次迭代后CORDIC公式的输出得到输入角度 Z_0 的余弦值 $\cos(Z_0)$ 和正弦值 $\sin(Z_0)$:

$$[X_N, Y_N, Z_N] = [\cos(Z_0), \sin(Z_0), 0]$$

在向量模式中, CORDIC经过N次迭代后, CORDIC公式的输出变为:

$$\begin{aligned} X_N &= P(\sqrt{(X_0)^2 + (Y_0)^2}) \\ Y_N &= 0 \\ Z_N &= Z_0 + \tan^{-1}(Y_0/X_0) \end{aligned}$$

设定 $Z_0 = 0$, 那么N次迭代后CORDIC公式的输出得到输入向量 (X_0, Y_0) 的反正切值 $\tan^{-1}(Y_0/X_0)$ 和向量的模 $P(\sqrt{(X_0)^2 + (Y_0)^2})$:

$$[X_N, Y_N, Z_N] = [P(\sqrt{(X_0)^2 + (Y_0)^2}), 0, \tan^{-1}(Y_0/X_0)]$$

CORDIC算法的误差主要来自于截断误差。通过对每次计算进行16次迭代, 同时内核数据使用32位位宽, JMT1801ED CORDIC协处理器的运算达到了相当高的精度。

JMT1801ED CORDIC协处理器有两种调度模式: JMT51 MCU调度、JMT018 DSP指令调度。

12.2 功能描述

12.2.1 输入数据格式

JMT1801ED CORDIC 协处理器支持两种计算模式：旋转模式和向量模式。

CORDIC 旋转模式，也即计算任意角度的正余弦(COS/SIN)，需要输入待计算的角度 Z_0 ：

- 配置寄存器(Z0_HIGH,Z0_LOW)⁵或者 Z0_DSP⁶，输入待计算正余弦的角度 Z_0 。
- Z_0 取值范围为 $[-2^{15}, (2^{15}-1)]$ ，对应的角度范围为 $[-\pi, ((2^{15}-1)/2^{15})\pi]$ 。

CORDIC 向量模式，也即计算任意向量的反正切值 (\tan^{-1}) 以及该向量的模 $P * |a|$ ，其中 $|a| = \sqrt{(X_0)^2 + (Y_0)^2}$, $P = 1.6468$ ，需要输入待计算的向量 (X_0, Y_0) 的坐标值 X_0 和 Y_0 ：

- 配置寄存器(X0_HIGH,X0_LOW)或者 X0_DSP，输入待计算向量的 X 轴坐标 X_0 。
- 配置寄存器(Y0_HIGH,Y0_LOW)或者 Y0_DSP，输入待计算向量的 Y 轴坐标 Y_0 。
- X_0 和 Y_0 取值范围为 $[-2^{15}, (2^{15}-1)]$ 。

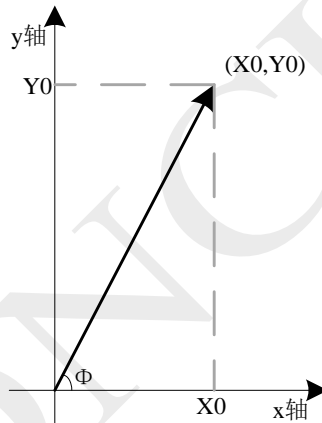


Figure 106 向量 (X_0, Y_0)

12.2.2 结果数据归一化

JMT1801ED CORDIC 协处理器旋转模式计算任意角度的正余弦 (COS/SIN)：

- 回读(YN_HIGH,YN_LOW)或者 YN_DSP 寄存器，得到待计算角度的正弦值 $\sin(Z_0)$ 。
- 回读(XN_HIGH,XN_LOW)或者 XN_DSP 寄存器，得到待计算角度的余弦值 $\cos(Z_0)$ 。
- 正余弦 (SIN/COS) 结果为有符号数，使用二进制补码的数据格式表示。
- 正余弦 (SIN/COS) 结果数据使用 0 个整数位，15 个小数位表示。
- 若正余弦 (SIN/COS) 结果为 1 时，使用最接近 1 的小数表示 (0x7fff)。

⁵(Z0_HIGH,Z0_LOW)：后缀为_HIGH 寄存器代表输入参数的高 8 位，后缀为_LOW 寄存器代表输入参数的低 8 位，下同；

⁶ Z0_DSP：后缀为_DSP 寄存器，用于 DSP 指令调度，否则用于 JMT51 调度，下同。

JMT1801ED CORDIC 协处理器向量模式计算任意向量的反正切值 (\tan^{-1}) 以及该向量的模 $P * |a|$:

- 回读(ZN_HIGH,ZN_LOW)或者 ZN_DSP 寄存器, 得到输入向量的反正切值 (\tan^{-1}), 也即得到 $\tan^{-1}(Y_0/X_0)$ 。
- 回读(XN_HIGH,XN_LOW) 或者 XN_DSP 寄存器, 得到该向量的模 $P(\sqrt{(X_0)^2 + (Y_0)^2})$ 。
- 反正切值 $\tan^{-1}(Y_0/X_0)$ 得到的是一个角度, 结果为按比例因子调整后由整数 $[-2^{14}, (2^{14}-1)]$ 代表角度 $[-\pi/2, ((2^{14}-1)/2^{14})\pi/2]$ 。
- 向量的模 $P(\sqrt{(X_0)^2 + (Y_0)^2})$ 数据使用 1 个符号位, 15 个整数位表示。
- 若向量的模 $P(\sqrt{(X_0)^2 + (Y_0)^2})$ 超出 16 位有符号数所表达的范围时, 使用 16 位有符号最大正整数表示 (0x7fff)。

12.2.3 CORDIC 调度

JMT1801ED CORDIC 协处理器可以被 JMT51 MCU 或者 JMT018 DSP 指令调度, 每次启动之后, 需要 19 个工作时钟进行迭代操作, 迭代完成之后, 可以对新的数据进行处理。

i. JMT51 MCU 调度 CORDIC:

- 配置 CORDIC 的工作模式寄存器 CORD_MOD
- 根据工作模式, 配置 CORDIC 的初始参数:
 - 若需要计算某一角度的正余弦 (SIN/COS) 值, 需要配置角度寄存器 Z0_LOW、Z0_HIGH。
 - 若需要计算某向量对应的反正切值 (\tan^{-1}) 或者模 ($P * |a|$), 需要配置向量坐标寄存器 X0_LOW、X0_HIGH、Y0_LOW、Y0_HIGH。
- 配置 CORDIC 启动寄存器 CORD_STA, 启动 CORDIC 计算
- CORDIC 完成计算之后:
 - 采用中断模式: 使能中断, 待 CORDIC 完成计算, 会产生对应的 CORDIC 中断。
 - 采用查询模式: 通过回读状态寄存器 CORD_INT, 查询 CORDIC 是否完成工作。
- CORDIC 结果保存至对应的结果寄存器, 回读结果寄存器即可得到需要的结果。
 - 角度的正余弦保存至(YN_HIGH,YN_LOW)、(XN_HIGH,XN_LOW)寄存器, 分别对应正弦 (SIN) 和余弦 (COS)。
 - 向量对应角度的反正切值保存至(ZN_HIGH,ZN_LOW)寄存器, 向量的模保存至(XN_HIG,HXN_LOW)寄存器。

ii. JMT018 DSP 指令调度 CORDIC:

- 配置 CORDIC 的工作模式寄存器 CORD_MOD_DSP
- 根据工作模式，配置 CORDIC 的初始参数。
 - 若需要计算某一角度的正余弦（SIN/COS）值，需要配置角度寄存器 Z0_DSP。
 - 若需要计算某向量对应的反正切值（ \tan^{-1} ）或者模（ $P * |a|$ ），需要配置向量坐标寄存器（X0_DSP，Y0_DSP）。
- 使用 CORDIC 指令（“STCDC”指令），启动 CORDIC 开始工作。
- 通过“STCDC”指令启动 CORDIC 之后，无需等待，即可使用 DSP 的 MOV 指令读取 CORDIC 的运算结果。
- CORDIC 结果保存至对应的结果寄存器，回读结果寄存器即可得到需要的结果：
 - 角度的正弦值保存至寄存器 YN_DSP，角度的余弦值保存至寄存器 XN_DSP。
 - 向量对应角度的反正切值保存至寄存器 ZN_DSP，向量的模保存至寄存器 XN_DSP。

注意，若需要 JMT51 和 JMT018 DSP 同时调度 CORDIC，必须保证在 CORDIC 非忙状态，才能再次调度 CORDIC，否则会导致结果不正确。JMT51 可通过回读状态寄存器 CORD_INT 查询 CORDIC 的工作状态，JMT018 DSP 可通过回读状态寄存器 CORD_STU_DSP 查询 CORDIC 的工作状态。

12.2.4 CORDIC 中断

JMT1801ED CORDIC 协处理器被 JMT51 MCU 调度时，可以使用中断模式或者查询模式：

- 中断模式：
 - 配置 IEN2.CDCINT 寄存器为 1，使能 CORDIC 中断。
 - 当 CORDIC 计算完成后产生 CORDIC 中断。
 - JMT51 收到中断之后，配置 CORD_INT 寄存器为 0，清除 CORDIC 中断。
- 查询模式：
 - 配置 CORD_STA 寄存器启动 CORDIC 工作之后，查询 CORD_INT 状态寄存器
 - 当 CORD_INT 状态寄存器为 1 时，表示 CORDIC 完成计算，配置 CORD_INT 状态寄存器为 0 清除 CORDIC 完成标志。

12.2.5 CORDIC 精度

JMT1801ED CORDIC 协处理器运算进行 16 次迭代，内部使用 32 位数据宽度进行迭代

操作，所计算的三角函数达到了相当高的精度。

使用误差的均方根来表示 JMT808R CORDIC 的精度，如下公式：

$$X_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_{理论i} - X_{计算i})^2}, N = 32768$$

JMT1801ED CORDIC 计算的正余弦和理论的正余弦 (SIN/COS) 误差的均方根 $X_{rms}=9.25e^{-5}$ 。

JMT1801ED CORDIC 计算向量反正切 (\tan^{-1}) 误差的均方根 $X_{rms}=6.45e^{-5}$ 。

JMT1801ED CORDIC 计算向量模 ($P*|a|$) 误差的均方根 $X_{rms}=0.3764$ 。

12.3 CORDIC 寄存器

JMT1801ED CORDIC 协处理器受 JMT51 MCU 和 JMT018 DSP 指令调度，寄存器不同，下面分别介绍：

12.3.1 JMT51 MCU 调度寄存器

JMT51 MCU 调度 CORDIC 协处理器，共有 16 个寄存器，包括 X/Y/Z 三个初始值寄存器 (X0/Y0/Z0)、X/Y/Z 三个结果寄存器 (XN/YN/ZN)、启动寄存器、模式选择寄存器、工作状态寄存器和中断寄存器。对应的 SFR PAGE 为 1。

Table 200 JMT51 调度 CORDIC 寄存器 (SFRPAGE=0x01)

地址	寄存器名	属性	复位值	功能描述
0xB9	X0_LOW	W/R	0x00	输入数据 X0 低 8 位
0xBA	X0_HIGH	W/R	0x00	输入数据 X0 高 8 位
0xBB	Y0_LOW	W/R	0x00	输入数据 Y0 低 8 位
0xBC	Y0_HIGH	W/R	0x00	输入数据 Y0 高 8 位
0xBD	Z0_LOW	W/R	0x00	输入数据 Z0 低 8 位
0xBE	Z0_HIGH	W/R	0x00	输入数据 Z0 高 8 位
0xC1	XN_LOW	R	0x00	运算结果 XN 低 8 位
0xC2	XN_HIGH	R	0x00	运算结果 XN 高 8 位
0xC3	YN_LOW	R	0x00	运算结果 YN 低 8 位
0xC4	YN_HIGH	R	0x00	运算结果 YN 高 8 位
0xC5	ZN_LOW	R	0x00	运算结果 ZN 低 8 位
0xC6	ZN_HIGH	R	0x00	运算结果 ZN 高 8 位
0xC9	CORD_STA	W	0x00	CORDIC 启动寄存器
0xCA	CORD_MOD	W/R	0x00	CORDIC 模式寄存器

地址	寄存器名	属性	复位值	功能描述
0xCB	CORD_STU	R	0x00	CORDIC 状态寄存器
0xCC	CORD_INT	W/R	0x00	CORDIC 中断寄存器

12.3.1.1 输入数据 X0 低 8 位(X0_LOW) [0xB9]

Table 201 输入数据 X0 低 8 位(X0_LOW)

参数名	比特位	属性	复位值	描述
X0_LOW	7-0	W/R	0	输入数据 X0 低 8 位

12.3.1.2 输入数据 X0 高 8 位(X0_HIGH) [0xBA]

Table 202 输入数据 X0 高 8 位(X0_HIGH)

参数名	比特位	属性	复位值	描述
X0_HIGH	7-0	W/R	0	输入数据 X0 高 8 位

12.3.1.3 输入数据 Y0 低 8 位(Y0_LOW) [0xBB]

Table 203 输入数据 Y0 低 8 位(Y0_LOW)

参数名	比特位	属性	复位值	描述
Y0_LOW	7-0	W/R	0	输入数据 Y0 低 8 位

12.3.1.4 输入数据 Y0 高 8 位(Y0_HIGH) [0xBC]

Table 204 输入数据 Y0 高 8 位(Y0_HIGH)

参数名	比特位	属性	复位值	描述
Y0_HIGH	7-0	W/R	0	输入数据 Y0 高 8 位

12.3.1.5 输入数据 Z0 低 8 位(Z0_LOW) [0xBD]

Table 205 输入数据 Z0 低 8 位(Z0_LOW)

参数名	比特位	属性	复位值	描述
Z0_LOW	7-0	W/R	0	输入数据 Z0 值低 8 位

12.3.1.6 输入数据 Z0 高 8 位(Z0_HIGH) [0xBE]

Table 206 输入数据 Z0 高 8 位(Z0_HIGH)

参数名	比特位	属性	复位值	描述
Z0_HIGH	7-0	W/R	0	输入数据 Z0 高 8 位

12.3.1.7 运算结果 XN 低 8 位(XN_LOW) [0xC1]

Table 207 运算结果 XN 低 8 位(XN_LOW)

参数名	比特位	属性	复位值	描述
XN_LOW	7-0	R	0	运算结果 XN 低 8 位

12.3.1.8 运算结果 XN 高 8 位(XN_HIGH) [0xC2]

Table 208 运算结果 XN 高 8 位(XN_HIGH)

参数名	比特位	属性	复位值	描述
XN_HIGH	7-0	R	0	运算结果 XN 高 8 位

12.3.1.9 运算结果 YN 低 8 位(YN_LOW) [0xC3]

Table 209 运算结果 YN 低 8 位(YN_LOW)

参数名	比特位	属性	复位值	描述
YN_LOW	7-0	R	0	运算结果 YN 低 8 位

12.3.1.10 运算结果 YN 高 8 位(YN_HIGH) [0xC4]

Table 210 运算结果 YN 高 8 位(YN_HIGH)

参数名	比特位	属性	复位值	描述
YN_HIGH	7-0	R	0	运算结果 YN 高 8 位

12.3.1.11 运算结果 ZN 低 8 位(ZN_LOW) [0xC5]

Table 211 运算结果 ZN 低 8 位(ZN_LOW)

参数名	比特位	属性	复位值	描述
ZN_LOW	7-0	R	0	运算结果 ZN 低 8 位

12.3.1.12 运算结果 ZN 高 8 位(ZN_HIGH) [0xC6]

Table 212 运算结果 ZN 高 8 位(ZN_HIGH)

参数名	比特位	属性	复位值	描述
ZN_HIGH	7-0	R	0	运算结果 ZN 高 8 位

12.3.1.13 CORDIC 启动寄存器(CORD_STA) [0xC9]

Table 213 CORDIC 启动寄存器(CORD_STA)

参数名	比特位	属性	复位值	描述
CORD_STA	0	W	0	CORDIC 启动寄存器: 0: 不操作 1: 启动一次 CORDIC 计算
Reserved	7-1	R	0	保留

12.3.1.14 CORDIC 模式寄存器(CORD_MOD) [0xCA]

Table 214 CORDIC 模式寄存器(CORD_MOD)

参数名	比特位	属性	复位值	描述
CORD_MOD	0	W/R	0	CORDIC 模式寄存器： 0: 旋转模式，计算角度正余弦 1: 向量模式，计算反正切和模
Reserved	7-1	R	0	保留

12.3.1.15 CORDIC 状态寄存器(CORD_STU) [0xCB]

Table 215 CORDIC 状态寄存器(CORD_STU)

参数名	比特位	属性	复位值	描述
CORD_STU	0	R	0	CORDIC 状态寄存器： 0: CORDIC 空闲 1: CORDIC 正在进行迭代运算
Reserved	7-1	R	0	保留

12.3.1.16 CORDIC 中断寄存器(CORD_INT) [0xCC]

Table 216 CORDIC 中断寄存器(CORD_INT)

参数名	比特位	属性	复位值	描述
CORD_INT	0	W/R	0	CORDIC 中断标志位： <ul style="list-style-type: none"> • 当有 CORDIC 中断申请时，该位置为 1 • 软件配置该位为 1 可产生软中断，在中断处理程序中需要软件写 0 来清除
Reserved	7-1	R	0	保留

12.3.2 JMT018 DSP 指令调度寄存器

JMT018 DPS 指令调度 CORDIC 协处理器,包括 X/Y/Z 三个输入数据寄存器(X0/Y0/Z0)、X/Y/Z 三个结果寄存器 (XN/YN/ZN)、模式选择寄存器和工作状态寄存器。

Table 217 JMT018 DSP 指令调度 CORDIC 寄存器

地址	寄存器名	属性	复位值	功能描述
0xC0	X0_DSP	W/R	0x0000	输入数据 X0
0xC1	Y0_DSP	W/R	0x0000	输入数据 Y0

地址	寄存器名	属性	复位值	功能描述
0xC2	Z0_DSP	W/R	0x0000	输入数据 Z0
0xC3	XN_DSP	R	0x0000	运算结果 XN
0xC4	YN_DSP	R	0x0000	运算结果 YN
0xC5	ZN_DSP	R	0x0000	运算结果 ZN
0xC6	CORD_MOD_DSP	W/R	0x0000	CORDIC 模式寄存器
0xC7	CORD_STU_DSP	R	0x0000	CORDIC 状态寄存器

12.3.2.1 输入数据 X0(X0_DSP) [0xC0]

Table 218 输入数据 X0(X0_DSP)

参数名	比特位	属性	复位值	描述
X0_DSP	15-0	W/R	0	输入数据 X0

12.3.2.2 输入数据 Y0(Y0_DSP) [0xC1]

Table 219 输入数据 Y0(Y0_DSP)

参数名	比特位	属性	复位值	描述
Y0_DSP	15-0	W/R	0	输入数据 Y0

12.3.2.3 输入数据 Z0(Z0_DSP) [0xC2]

Table 220 输入数据 Z0(Z0_DSP)

参数名	比特位	属性	复位值	描述
Z0_DSP	15-0	W/R	0	输入数据 Z0

12.3.2.4 运算结果 XN(XN_DSP) [0xC3]

Table 221 运算结果 XN(XN_DSP)

参数名	比特位	属性	复位值	描述
XN_DSP	15-0	R	0	运算结果 XN

12.3.2.5 运算结果 YN(YN_DSP) [0xC4]

Table 222 运算结果 YN(YN_DSP)

参数名	比特位	属性	复位值	描述
YN_DSP	15-0	R	0	运算结果 YN

12.3.2.6 运算结果 ZN(ZN_DSP) [0xC5]

Table 223 运算结果 ZN(ZN_DSP)

参数名	比特位	属性	复位值	描述
ZN_DSP	15-0	R	0	运算结果 ZN

12.3.2.7 CORDIC 模式寄存器(CORD_MOD_DSP) [0xC6]

Table 224 CORDIC 模式寄存器(CORD_MOD_DSP)

参数名	比特位	属性	复位值	描述
CORD_MOD_DSP	0	W/R	0	CORDIC 模式寄存器： 0: 旋转模式，计算角度正余弦 1: 向量模式，计算反正切和模
Reserved	15-1	R	0	保留

12.3.2.8 CORDIC 状态寄存器(CORD_STU_DSP) [0xC7]

Table 225 CORDIC 状态寄存器(CORD_STU_DSP)

参数名	比特位	属性	复位值	描述
CORD_STU_DSP	0	R	0	CORDIC 状态寄存器： 0: CORDIC 空闲 1: CORDIC 正在进行迭代运算
Reserved	15-1	R	0	保留

13 通用异步收发传输器 (UART)

13.1 概述

JMT1801ED 的通用异步收发传输器 (UART) 支持三种接口：

- 标准 UART 通信接口
- 38KHz 红外调制发送接口
- IrDA(SIR)红外通信接口

13.2 框图

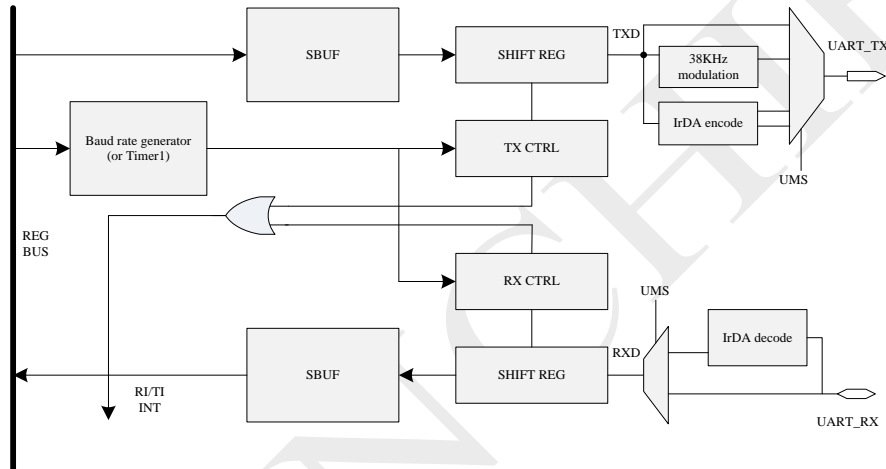


Figure 107 UART 结构框图

UART 通信接口由两个数据缓冲器、一个移位寄存器、一个串行控制寄存器、一个波特率发生器、一个 38KHz 调制和一个 IrDA 编解码器等组成。

标准 UART 通信接口共有 4 种工作模式，以供不同应用场景选用：

- 模式 0：同步移位寄存器。通过 UART_TX 发送同步时钟，通过 UART_RX 接收或发送数据，波特率是处理器频率 $f_{\text{sys_clk}}$ 的 1/12。
- 模式 1：8 位 UART。通过 UART_TX 发送或通过 UART_RX 接收 8 个数据位，波特率是可变的。
- 模式 2：9 位 UART。通过 UART_TX 发送或通过 UART_RX 接收 9 个数据位，波特率可编程为处理器频率 $f_{\text{sys_clk}}$ 的 1/32 或 1/64。
- 模式 3：9 位 UART。通过 UART_TX 发送或通过 UART_RX 接收 9 个数据位，波特率是可变的。

38KHz 红外调制发送接口，可通过标准 UART 通信接口的模式 1 和 3 实现，具体参加 13.4.5。

IrDA(SIR)红外通信接口，可通过标准 UART 通信接口的模式 1、2 和 3 实现，支持 3/16

以及 low-power(1.41-2.23us)脉宽格式，具体参见 13.4.6。

13.3 UART 管脚配置

UART 的管脚配置可参见 Table 1 引脚说明表格，管脚配置说明见 GPIO 寄存器说明。

13.4 UART 工作模式

13.4.1 标准 UART 工作模式 0：同步移位寄存器

配置 SCON 的 SM0、SM1 为 00 时，UART 工作于模式 0。此模式为同步移位寄存器，波特率固定为系统时钟频率 f_{sys_clk} 的 1/12。同步时钟由 UART_TX 输出，数据由 UART_RX 端输入或输出，发送、接收的是 8 位数据，低位在先。接口时序如 Figure 108 和 Figure 109 所示。图中 baud_clk 表示波特率示意时钟，buf_write 为软件写 SBUF 动作，SCON.TI 为发送中断标志，SCON.RI 为接收中断标志，此后类同。

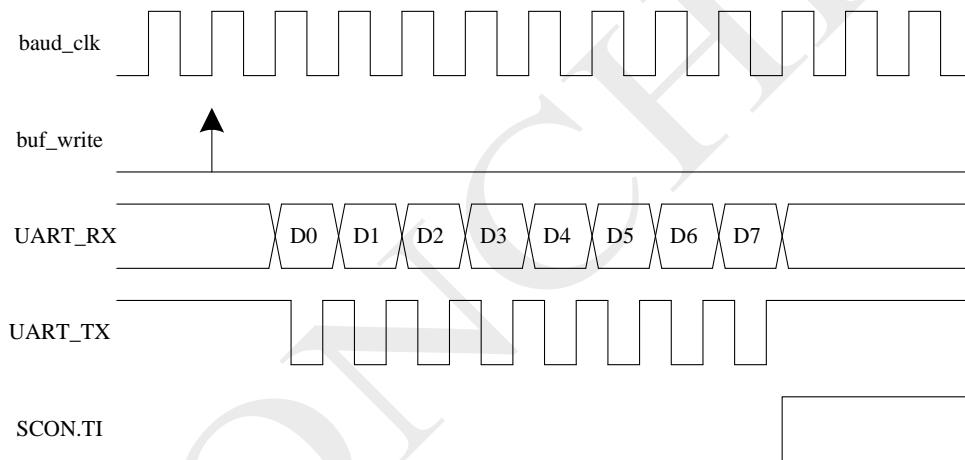


Figure 108 UART 模式 0 发送时序

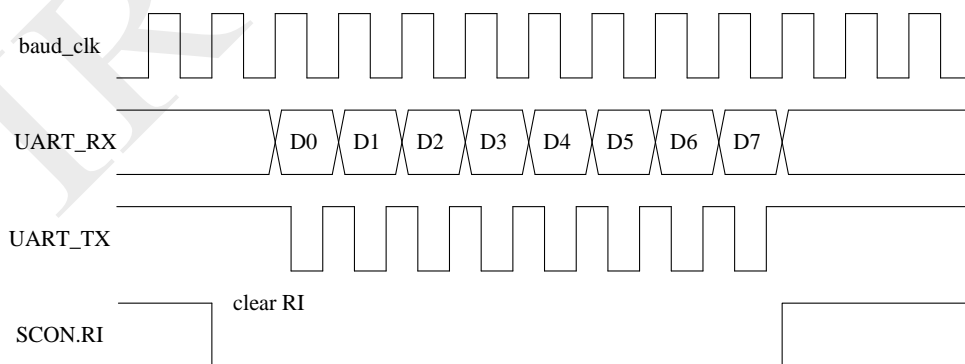


Figure 109 UART 模式 0 接收时序

模式 0 发送：当处理器执行将数据写入发送缓冲器 SBUF 时启动发送，硬件将 8 位数据以 $f_{sys_clk}/12$ 的波特率从 UART_RX 管脚输出。一帧(8 位)数据发送完毕时，SCON.TI 置 1，申请中断。再次发送数据前，必须用软件将 SCON.TI 清 0。

模式0接收: 首先将接收中断请求标志 `SCON.RI` 置 1, 然后将接收中断请求标志 `SCON.RI` 清 0, 且将允许接收控制位 `SCON.REN` 置 1, 启动串行模式 0 接收过程。启动接收过程后, `UART_RX` 为串行输入端, `UART_TX` 为同步时钟。串行接收的波特率为 $f_{sys_clk}/12$ 。当接收完成一帧数据(8 位)后, 硬件将会把 `SCON.RI` 置 1, 并发出中断申请。当再次接收时, 必须通过软件将 `SCON.RI` 清 0。

工作于模式 0 时, 必须将多机通信控制位 `SCON.SM2` 清 0, 使得对 `SCON.TB8` 位和 `SCON.RB8` 位没有影响。

模式 0 发送中断和模式 0 接收中断合并后送至 CPU, 请求中断, 软件响应中断后必须判别是 `SCON.TI` 中断请求还是 `SCON.RI` 中断请求, 并清除相应的标志位。

13.4.2 标准 UART 工作模式 1: 8 位 UART, 波特率可变

配置寄存器 `SCON` 的 `SM0`、`SM1` 为 01 时, UART 工作于模式 1。此模式为 8 位 UART 工作模式, 一帧数据为 10 位: 1 位起始位, 8 位数据位(低位在先)和 1 位停止位。

该模式下, 波特率可变, 即可根据需要进行设置。

`UART_TX` 为发送信息, `UART_RX` 为接收信息, 以全双工模式进行接收/发送。接口时序如 Figure 110 和 Figure 111 所示。

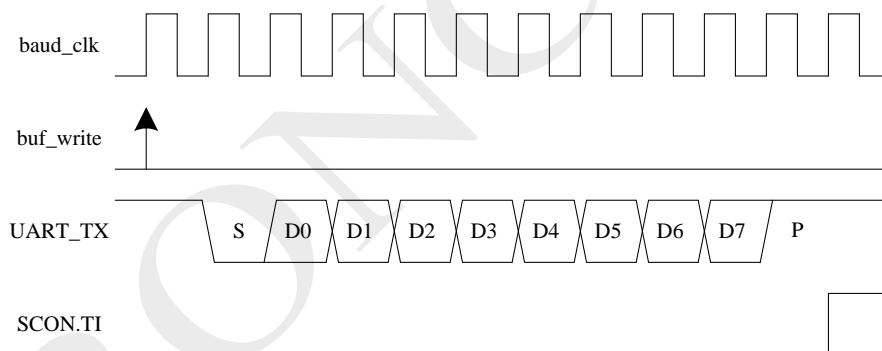


Figure 110 UART 模式 1 发送时序

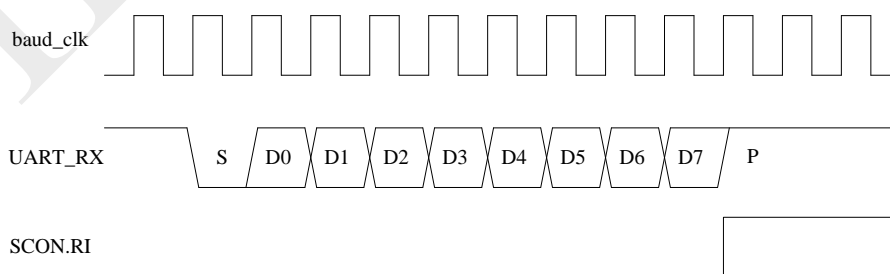


Figure 111 UART 模式 1 接收时序

UART 模式 1 中, 波特率可以由内部的波特率产生器或 Timer 1 产生, 具体参见 13.7 节。在模式 1 中, 可以用中断方式进行数据通信。当发送完一帧数据后, 硬件自动将发送中

断标志 `SCON.TI` 置 1；当接收完一帧数据后，硬件自动将接收中断标志 `SCON.RI` 置 1。在响应中断后，必须由软件清 0。

模式 1 发送中断和模式 1 接收中断合并后送至 CPU，请求中断，软件响应中断后必须判别是 `SCON.TI` 中断请求还是 `SCON.RI` 中断请求，并清除相应的标志位。

13.4.3 标准 UART 工作模式 2：9 位 UART，波特率固定

配置寄存器 `SCON` 的 `SM0`、`SM1` 为 10 时，UART 工作于模式 2。此模式为 9 位 UART 工作模式，一帧信息为 11 位：1 位起始位，8 位数据位(低位在先)，1 位可编程位(`SCON.TB8`/`SCON.RB8`，第 9 位数据)和 1 位停止位。`SCON.TB8`/`SCON.RB8` 可以作为多机通信地址标志或奇偶校验位使用。`UART_TX` 为发送信息，`UART_RX` 为接收信息，以全双工模式进行接收/发送。接口时序如 Figure 112 和 Figure 113 所示。

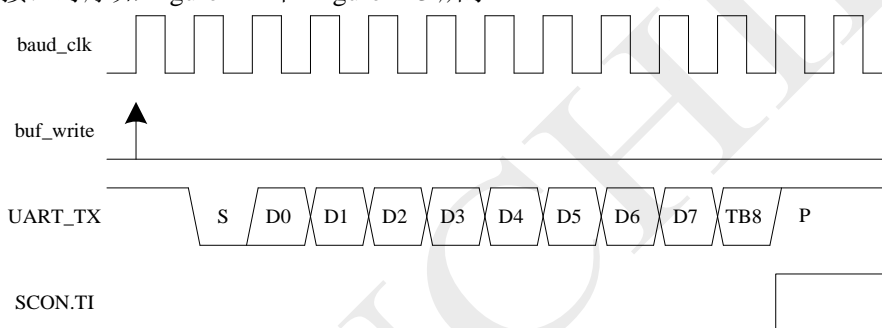


Figure 112 UART 模式 2 发送时序

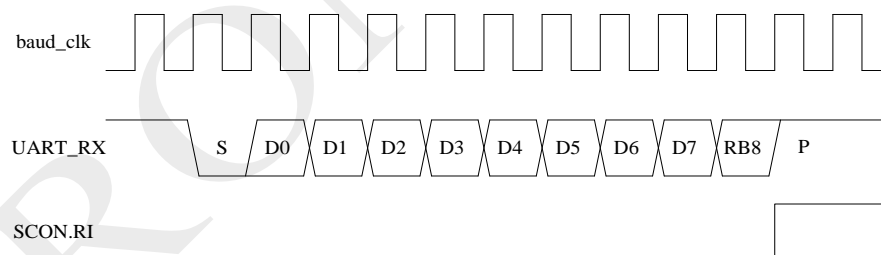


Figure 113 UART 模式 2 接收时序

配置寄存器 `SCON` 中的 `SM2`、`TB8` 可实现多机通信和奇偶校验，具体见 13.5 和 13.6 节。

模式 2 发送中断和模式 2 接收中断合并后送至 CPU，请求中断，软件响应中断后必须判别是 `SCON.TI` 中断请求还是 `SCON.RI` 中断请求，并清除相应的标志位。

13.4.4 标准 UART 工作模式 3：9 位 UART，波特率可变

配置寄存器 `SCON` 的 `SM0`、`SM1` 为 11 时，UART 工作于模式 3。此模式为 9 位 UART 工作模式，一帧信息为 11 位：1 位起始位，8 位数据位(低位在先)，1 位可编程位(`SCON.TB8`/`SCON.RB8`，第 9 位数据)和 1 位停止位。`SCON.TB8`/`SCON.RB8` 可以作为多机通信地址标志或奇偶校验位使用。`UART_TX` 为发送信息，`UART_RX` 为接收信息，以全双工模式进行接

收/发送。接口时序如 Figure 114 和 Figure 115 所示。

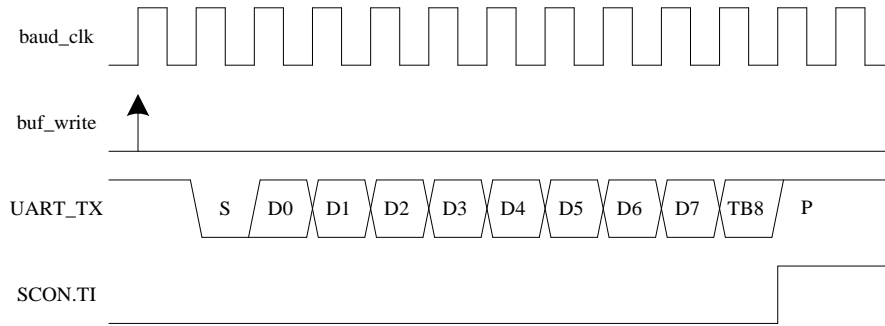


Figure 114 UART 模式 3 发送时序

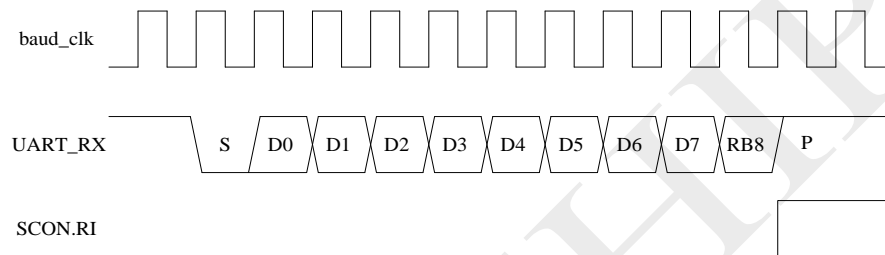


Figure 115 UART 模式 3 接收时序

UART 模式 3 中,波特率可以由内部的波特率产生器或 Timer 1 产生,具体参见 13.7 节。

配置寄存器 SCON 中的 SM2、TB8 可实现多机通信和奇偶校验,具体见 13.5 和 13.6 节。

模式 3 发送中断和模式 3 接收中断合并后送至 CPU, 请求中断, 软件响应中断后必须判别是 SCON.TI 中断请求还是 SCON.RI 中断请求, 并清除相应的标志位。

13.4.5 38KHz 红外调制发送

UART 支持 38KHz 的红外调制发送功能。38KHz 红外调制原理为: 将 TXD 的数据‘0’调制为占空比为 1/3 的 38KHz 信号, 数据‘1’不调制。

驱动红外发送管的三极管可以是 NPN 型或 PNP 型, 通过设置 SIRCON.IRINV 控制寄存器来选择, 具体见 Figure 116, IRINV=0 对应 NPN 型, IRINV=1 对应 PNP 型。

通过改变寄存器 IRPD(见寄存器 SRELH 和 SIRCON)可调整至 38KHz 红外载波频率, 分频系数寄存器 IRPD 计算公式为:

$$IRPD = \left(\frac{f_{sys_clk}(KHz)}{(3 * 38)(KHz)} \right)$$

在该工作模式下, 发送为 38KHz 的红外调制信号, 接收为标准 UART 信号。

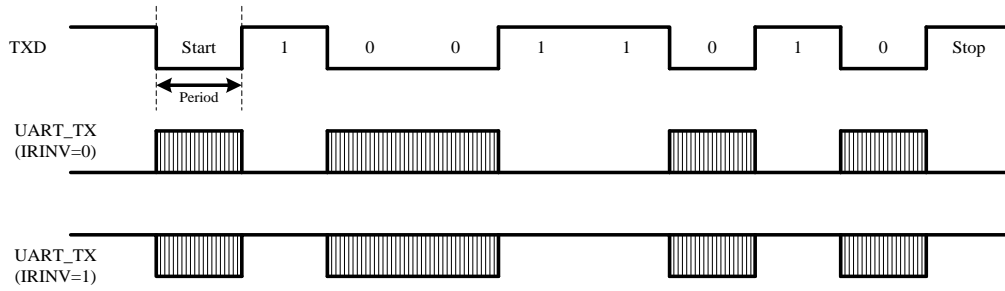


Figure 116 38KHz 红外调制发送时序

13.4.6 IrDA(SIR)红外通信

IrDA 是红外数据组织 (Infrared Data Association) 的简称, 在 IrDA 物理层中, 将数据通信按发送速率分为三类: SIR、MIR 和 FIR。串行红外 (SIR) 的速率覆盖了 RS-232 端口通常支持的速率 (9600bps~115.2Kbps)。

IrDA(SIR)为半双工通信, 有两种模式:

- 正常模式:
 - 发送时, ‘0’被调制成 3/16 比特宽度的高电平脉冲输出, 而‘1’则被转换为持续的低电平输出。
 - 接收时, ‘0’对应的输入信号为 3/16 比特宽度的低电平脉冲, ‘1’对应的输入信号为整个比特宽度的高电平。
- 低功耗模式 (low-power IrDA):
 - 发送时, ‘0’被调制成 3 倍 $f_{SLPBaud}$ 比特宽度的高电平脉冲输出, 而‘1’则被转换为持续的低电平输出。
 - 接收时, ‘0’对应的输入信号为 3 倍 $f_{SLPBaud}$ 比特宽度的低电平脉冲, ‘1’对应的输入信号为整个比特宽度的高电平。
 - 分频系数寄存器 IRPD 可根据 $f_{SLPBaud}$ 和 f_{sys_clk} 得到, 具体计算公式为:

$$IRPD = \left(\frac{f_{sys_clk}}{f_{SLPBaud}} \right)$$

其中 $1.35MHz < f_{SLPBaud} < 2.12MHz$ 。

IrDA(SIR)通信在发送和接收之间至少需要 10ms 的延迟, 这种延迟也被称为接收器建立时间。

IrDA 接收和发送时序参见 Figure 117。

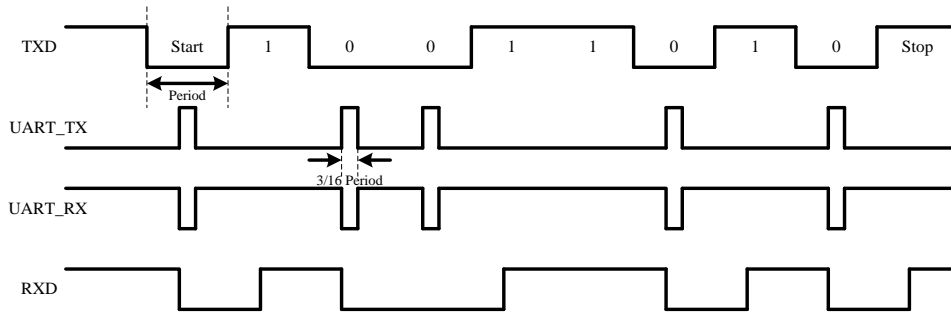


Figure 117 IrDA 接收和发送时序

13.5 UART 多机通信

UART 模式 2 和模式 3 通过使用第 9 数据位 (SCON.TB8/SCON.RB8) 可以支持一个主机与多个从机之间的多机通信。当主机要发送数据给多个从机时, 它先发送一个用于选择目标从机的地址字节。

地址字节与数据字节的区别是: 地址字节的第 9 位为逻辑 1, 数据字节的第 9 位为逻辑 0。

如果从机的寄存器 SCON.SM2 位被置 1, 则只有当 UART 接收到的第 9 位为 1 (SCON.RB8 = 1) 时, UART 才会产生接收中断。在 UART 的中断处理程序中, 软件将接收到的地址与从机自身的 8 位地址进行比较, 如果地址匹配, 从机将它的寄存器 SCON.SM2 清零, 从而允许后面接收数据字节时产生中断。未被寻址的从机保持 SCON.SM2 位为 1, 在收到后续的数据字节时不产生中断, 从而忽略接收到的数据。一旦数据块传输完毕, 被寻址的从机将它的 SCON.SM2 位重新置 1, 以忽略所有的数据传输, 直到它收到下一个地址字节。

可以将一个地址分配给多个从机, 从而允许同时向多个从机“广播”发送。

可以将多个地址分配给一个从机。

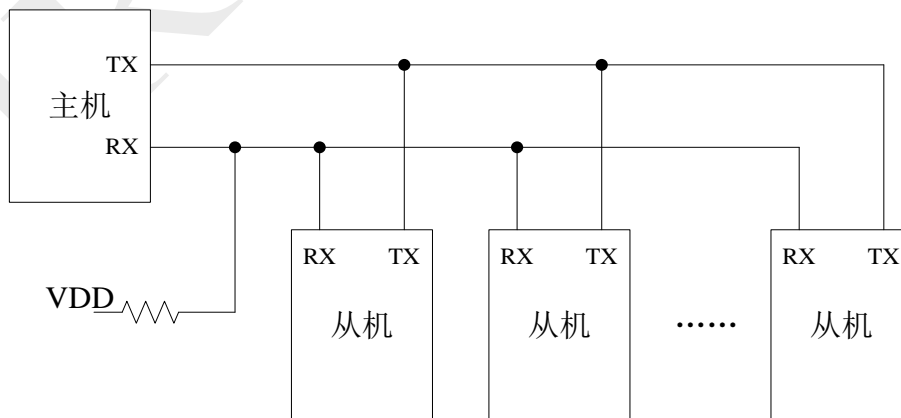


Figure 118 UART 多机通信连接方式

13.6 UART 奇偶校验

UART 模式 2 和模式 3 中每帧数据中含有第 9 数据位 (SCON.TB8/SCON.RB8)，该位可以实现奇偶校验的功能。

配置寄存器 SCON.SM2 为 0，使得 UART 工作在非多机通信模式。

发送时：SCON 中的 TB8 配置建议：将发送数据写入寄存器 A，然后将寄存器 PSW 中的奇/偶校验位 P 写入 SCON.TB8。

接收时：硬件自动把 UART 接收到的第 9 位数据写入 SCON 寄存器的 RB8，软件可将接收数据写入寄存器 A，然后将寄存器 PSW 中的奇/偶校验位 P 和 SCON 寄存器的 RB8 进行比较，来判断奇偶校验是否正确。

13.7 UART 波特率计算

UART 波特率计算：

- UART 模式 0

$$\text{baud rate} = \frac{f_{\text{sys_clk}}}{12}$$

- UART 模式 1, 其中 UART 的 UBSEL 对应 TCON.0; UART 的 SMOD 对应 TCON.2.

- 寄存器 UBSEL=0，使用 Timer1 的模式 2 作为波特率产生器：

$$\text{baud rate} = \frac{2^{\text{SMOD}} * f_{\text{sys_clk}}}{32 * (256 - \text{TH1})}$$

- 寄存器 UBSEL=1，使用 UART 自带波特率产生器：

$$\text{baud rate} = \frac{2^{\text{SMOD}} * f_{\text{sys_clk}}}{32 * (1024 - \text{SREL})}$$

Table 226 UART 模式 1 波特率 ($f_{\text{sys_clk}}=73.728\text{M}$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
9600	16	9600.0	0.0000	x	x	x
19200	136	19200.0	0.0000	16	19200.0	0.0000
38400	196	38400.0	0.0000	136	38400.0	0.0000
51200	211	51200.0	0.0000	166	51200.0	0.0000
57600	216	57600.0	0.0000	176	57600.0	0.0000
115200	236	115200.0	0.0000	216	115200.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
2400	64	2400.0	0.0000	x	x	x
4800	544	4800.0	0.0000	64	4800.0	0.0000

9600	784	9600.0	0.0000	544	9600.0	0.0000
19200	904	19200.0	0.0000	784	19200.0	0.0000
38400	964	38400.0	0.0000	904	38400.0	0.0000
51200	979	51200.0	0.0000	934	51200.0	0.0000
57600	984	57600.0	0.0000	944	57600.0	0.0000
115200	1004	115200.0	0.0000	984	115200.0	0.0000

 Table 227 UART 模式 1 波特率 ($f_{\text{sys_clk}}=36.864\text{M}$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
4800	16	4800.0	0.0000	x	x	x
9600	136	9600.0	0.0000	16	9600.0	0.0000
19200	196	19200.0	0.0000	136	19200.0	0.0000
38400	226	38400.0	0.0000	196	38400.0	0.0000
51200	234	52363.6	2.2727	211	51200.0	0.0000
57600	236	57600.0	0.0000	216	57600.0	0.0000
115200	246	115200.0	0.0000	236	115200.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
1200	64	1200.0	0.0000	x	x	x
2400	544	2400.0	0.0000	64	2400.0	0.0000
4800	784	4800.0	0.0000	544	4800.0	0.0000
9600	904	9600.0	0.0000	784	9600.0	0.0000
19200	964	19200.0	0.0000	904	19200.0	0.0000
38400	994	38400.0	0.0000	964	38400.0	0.0000
51200	1002	52363.6	2.2727	979	51200.0	0.0000
57600	1004	57600.0	0.0000	984	57600.0	0.0000
115200	1014	115200.0	0.0000	1004	115200.0	0.0000

 Table 228 UART 模式 1 波特率 ($f_{\text{sys_clk}}=18.432\text{M}$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
2400	16	2400.0	0.0000	x	x	x
4800	136	4800.0	0.0000	16	4800.0	0.0000

9600	196	9600.0	0.0000	136	9600.0	0.0000
19200	226	19200.0	0.0000	196	19200.0	0.0000
38400	241	38400.0	0.0000	226	38400.0	0.0000
51200	245	52363.6	2.2727	234	52363.6	2.2727
57600	246	57600.0	0.0000	236	57600.0	0.0000
115200	251	115200.0	0.0000	246	115200.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
600	64	600.0	0.0000	x	x	x
1200	544	1200.0	0.0000	64	1200.0	0.0000
2400	784	2400.0	0.0000	544	2400.0	0.0000
4800	904	4800.0	0.0000	784	4800.0	0.0000
9600	964	9600.0	0.0000	904	9600.0	0.0000
19200	994	19200.0	0.0000	964	19200.0	0.0000
38400	1009	38400.0	0.0000	994	38400.0	0.0000
51200	1013	52363.6	2.2727	1002	52363.6	2.2727
57600	1014	57600.0	0.0000	1004	57600.0	0.0000
115200	1019	115200.0	0.0000	1014	115200.0	0.0000

Table 229 UART 模式 1 波特率 ($f_{\text{sys_clk}}=9.216\text{M}$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
1200	16	1200.0	0.0000	x	x	x
2400	136	2400.0	0.0000	16	2400.0	0.0000
4800	196	4800.0	0.0000	136	4800.0	0.0000
9600	226	9600.0	0.0000	196	9600.0	0.0000
19200	241	19200.0	0.0000	226	19200.0	0.0000
38400	x	x	x	241	38400.0	0.0000
51200	x	x	x	245	52363.6	2.2727
57600	251	57600.0	0.0000	246	57600.0	0.0000
115200	x	x	x	251	115200.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)

300	64	300.0	0.0000	x	x	x
600	544	600.0	0.0000	64	600.0	0.0000
1200	784	1200.0	0.0000	544	1200.0	0.0000
2400	904	2400.0	0.0000	784	2400.0	0.0000
4800	964	4800.0	0.0000	904	4800.0	0.0000
9600	994	9600.0	0.0000	964	9600.0	0.0000
19200	1009	19200.0	0.0000	994	19200.0	0.0000
38400	x	x	x	1009	38400.0	0.0000
51200	x	x	x	1013	52363.6	2.2727
57600	1019	57600.0	0.0000	1014	57600.0	0.0000
115200	x	x	x	1019	115200.0	0.0000

 Table 230 UART 模式 1 波特率 ($f_{\text{sys_clk}}=4.608\text{M}$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
600	16	600.0	0.0000	x	x	x
1200	136	1200.0	0.0000	16	1200.0	0.0000
2400	196	2400.0	0.0000	136	2400.0	0.0000
4800	226	4800.0	0.0000	196	4800.0	0.0000
9600	241	9600.0	0.0000	226	9600.0	0.0000
19200	x	x	x	241	19200.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
300	544	300.0	0.0000	64	300.0	0.0000
600	784	600.0	0.0000	544	600.0	0.0000
1200	904	1200.0	0.0000	784	1200.0	0.0000
2400	964	2400.0	0.0000	904	2400.0	0.0000
4800	994	4800.0	0.0000	964	4800.0	0.0000
9600	1009	9600.0	0.0000	994	9600.0	0.0000
19200	x	x	x	1009	19200.0	0.0000

Table 231 UART 模式 1 波特率 (f_{sys_clk}=2.304M)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
300	16	300.0	0.0000	x	x	x
600	136	600.0	0.0000	16	600.0	0.0000
1200	196	1200.0	0.0000	136	1200.0	0.0000
2400	226	2400.0	0.0000	196	2400.0	0.0000
4800	241	4800.0	0.0000	226	4800.0	0.0000
9600	x	x	x	241	9600.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
300	784	300.0	0.0000	544	300.0	0.0000
600	904	600.0	0.0000	784	600.0	0.0000
1200	964	1200.0	0.0000	904	1200.0	0.0000
2400	994	2400.0	0.0000	964	2400.0	0.0000
4800	1009	4800.0	0.0000	994	4800.0	0.0000
9600	x	x	x	1009	9600.0	0.0000

- UART 模式 2, 其中 UART 的 SMOD 对应 TCON.2

$$baud\ rate = \frac{2^{SMOD} * f_{sys_clk}}{64}$$

- UART 模式 3

计算方法和模式 1 相同

13.8 UART 寄存器

UART 模块有 5 个寄存器, 对应的 SFRPAGE 为 0, 具体说明如 Table 232 所示。

Table 232 UART 寄存器 (SFRPAGE=0x00)

地址	寄存器名	属性	复位值	功能描述
0x98	SCON	W/R	0x00	UART 控制寄存器
0x99	SBUF	W/R	0x00	UART 数据缓冲寄存器
0x9A	SRELL	W/R	0x00	UART 波特率重载寄存器低 8 位
0x9B	SRELH	W/R	0x00	UART 波特率重载寄存器高 2 位
0x9C	SIRCON	W/R	0x00	UART 红外控制寄存器

13.8.1 UART 控制寄存器(SCON) [0x98]

Table 233 UART 控制寄存器(SCON)

参数名	比特位	属性	复位值	描述
RI	0	W/R	0	接收中断标志： <ul style="list-style-type: none"> • 当有接收中断申请时，该位置 1 • 软件配置该位为 1 可产生软中断，在中断处理程序中需要软件写 0 来清除
TI	1	W/R	0	发送中断标志： <ul style="list-style-type: none"> • 当有发送中断申请时，该位置为 1 • 软件配置该位为 1 可产生软中断，在中断处理程序中需要软件写 0 来清除
RB8	2	W/R	0	接收数据位 RB8： 模式 1 中若 SM2=0，则为收到的停止位 模式 2、3 中是收到的第 9 数据位
TB8	3	W/R	0	发送数据位 TB8： 模式 2、3 中是被发出去的第 9 数据位 由软件置 1/清 0
REN	4	W/R	0	允许接收位： 0：不允许接收 1：允许接收
SM2	5	W/R	0	多机交互使能位 0：不使能 1：使能
SM1	6	W/R	0	具体说明参见 Table 234
SM0	7	W/R	0	

Table 234 UART 工作方式选择

SM0	SM1	模式	描述	波特率
0	0	0	移位寄存器	$f_{\text{sys_clk}}/12$
0	1	1	8 位 UART	可变
1	0	2	9 位 UART	$f_{\text{sys_clk}}/32$ 或 $f_{\text{sys_clk}}/64$
1	1	3	9 位 UART	可变

13.8.2 UART 数据缓冲寄存器(SBUF) [0x99]

Table 235 UART 数据缓冲寄存器(SBUF)

参数名	比特位	属性	复位值	描述
SBUF	7-0	W/R	0	UART 数据缓冲寄存器： 对寄存器 SBUF 写操作，则 UART 将开始向外传输发送缓存数据； 对寄存器 SBUF 读操作，则 UART 将从串行接收缓存中读取数据

13.8.3 UART 波特率重载寄存器低 8 位(SRELL) [0x9A]

Table 236 UART 波特率重载寄存器低 8 位(SRELL)

参数名	比特位	属性	复位值	描述
SRELL	7-0	W/R	0	UART 波特率重载寄存器低 8 位： 通过配置该寄存器改变 UART 的通讯波特率。

13.8.4 UART 波特率重载寄存器高 2 位(SRELH) [0x9B]

Table 237 UART 波特率重载寄存器高 2 位(SRELH)

参数名	比特位	属性	复位值	描述
SRELH	1-0	W/R	0	UART 波特率重载寄存器高 2 位： 通过配置该寄存器改变 UART 的通讯波特率。
IRPD[9:4]	7-2	W/R	0	UART 红外调制频率控制位高 6 位

13.8.5 UART 红外控制寄存器(SIRCON) [0x9C]

Table 238 UART 红外控制寄存器(SIRCON)

参数名	比特位	属性	复位值	描述
IRPD[3:0]	3-0	W/R	0	UART 红外调制频率控制位低 4 位： 当工作在 low-power IrDA 模式时，用作 SIR 的 low-power 模式时钟分频系数，选择分频系数使 $1.35\text{MHz} < f_{\text{SLPBaud}} < 2.12\text{MHz}$ ； 当工作在 38KHz 红外调制发送模式时，用作 38KHz 时钟分频控制。 具体见 38KHz 红外调制发送和 IrDA (SIR) 红外通信描述

参数名	比特位	属性	复位值	描述
IRINV	4	W/R	0	38KHz 红外调试模式，数据‘1’不调制输出时电平： 0: 低电平 1: 高电平
LBEN	5	W/R	0	回环测试使能位： 0: 正常功能模式 1: 回环测试模式
UMS	7-6	W/R	0	UART 工作模式选择控制： 00: 标准 UART 模式 01: UART 38KHz 红外调试发送模式 10: 3/16 IrDA 模式 11: low-power IrDA 模式

14 I2C 接口

14.1 概述

JMT1801ED 的 I2C 接口支持：

- master/slave 工作模式；
- 单 master/多 master 操作；
- 7/10 位设备地址寻址；
- master/slave 模式下简单的读写操作；
- master/slave 模式下组合读写操作；
- SCL 高低电平宽度可编程；
- I2C 总线的 SCL 时钟频率为：
 - I2C 总线标准模式：100KHz
 - I2C 总线快速模式：400KHz

14.2 框图

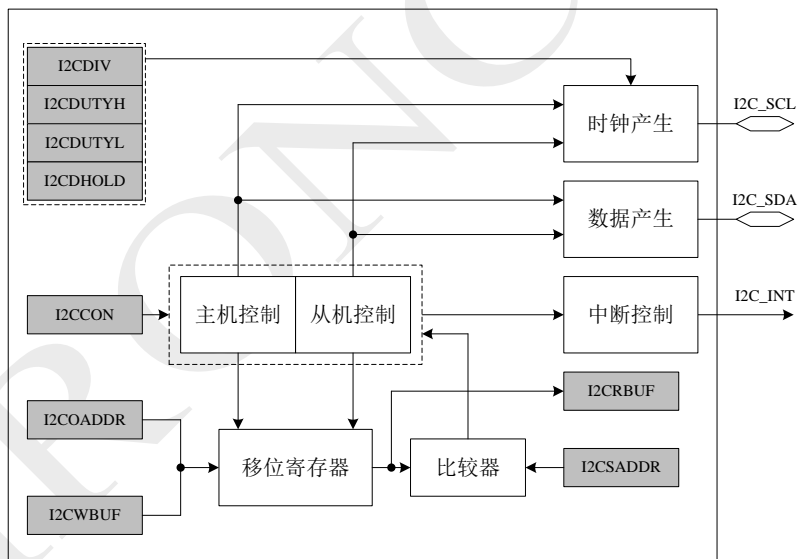


Figure 119 I2C 结构框图

14.3 I2C 管脚配置

I2C 的管脚配置可参见 Table 1 引脚说明表格，管脚配置说明见 GPIO 寄存器说明。

14.4 I2C 模块连接

每个连接到 I2C 总线的设备都有一个独一无二的地址识别。连接到 I2C 的设备都可以作为主机或从机。主机设备启动总线数据传输并产生传输的时钟信号。传输过程中，被主机寻址的设备是从机。

为了数据传输，I2C 接口有一根串行数据管脚（SDA）和一根串行时钟管脚（SCL）。两个管脚都具有漏极开路特性，来实现所需要的线与功能。它们都要用一个上拉电阻连接到正电压。总线空闲时，两个管脚都是高电平。

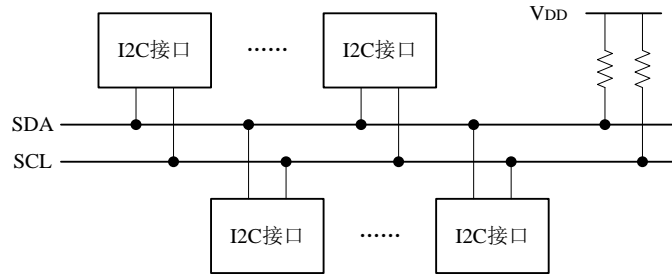


Figure 120 I2C 模块连接

14.5 I2C 数据格式

I2C 接口的时序图如下所示：

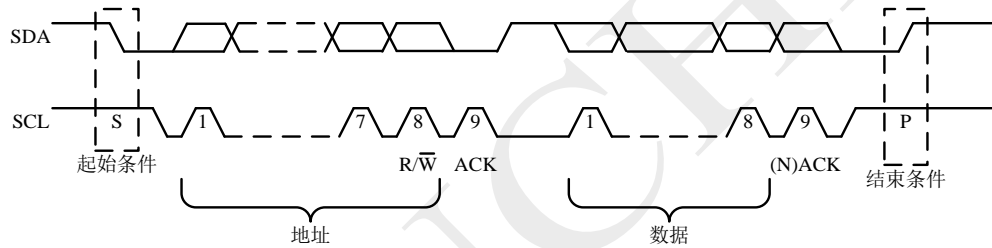


Figure 121 I2C 接口时序

14.5.1 数据有效

在传输数据的时候，SDA 线必须在时钟的高电平周期保持稳定，SDA 的高或低电平状态只有在 SCL 线是低电平时才能改变。

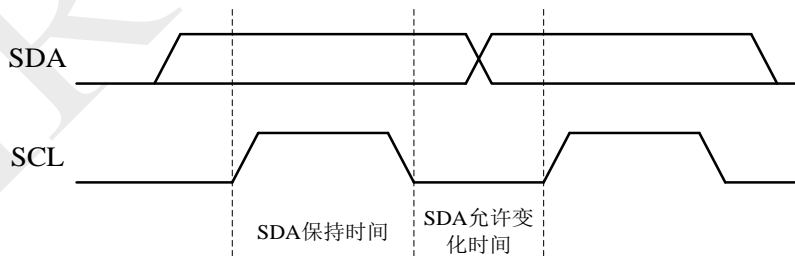


Figure 122 I2C 位传输

14.5.2 读写控制比特

在 I2C 总线中，首字节（地址或帧头字节）的第 8 位为读写控制比特（R/W）。

读写控制比特为 0 时，表示主机的写模式；读写控制比特为 1 时，表示主机的读模式。

14.5.3 反馈 (ACK)

不管是主机还是从机,在发送完 1 个字节后,都会等待接下来的一个比特作为反馈信号。

接收到的反馈信号为 0 时,表示发送字节被应答 (ACK);接收到的反馈信号为 1 时,表示发送字节未被应答 (NACK)。

14.5.4 起始条件和结束条件

在 I2C 总线中,起始条件 (S) 和结束条件 (P) 的情况:

- 起始条件指当 SCL 线是高电平时,SDA 线从高电平向低电平切换。
- 结束条件指当 SCL 线是高电平时,SDA 线由低电平向高电平切换。

起始和结束条件一般由主机产生。总线在起始条件后被认为处于忙的状态。在结束条件的某段时间后,总线被认为再次处于空闲状态。

如果产生重复起始 (Sr) 条件而不产生结束条件,总线会一直处于忙的状态。此时的起始条件 (S) 和重复起始 (Sr) 条件在功能上是一样的。

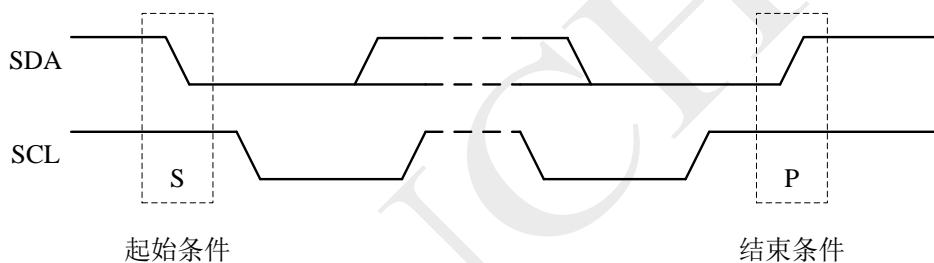


Figure 123 起始和结束条件

14.5.5 数据格式



Figure 124 数据格式

7 位地址格式,首字节由“地址”+“读写控制比特”组成。

10 位地址格式,首字节由“帧头”+“读写控制比特”组成,其中帧头由“11110”+“10 位地址的高 2 比特”组成;第二个字节由 10 位地址的低 8 比特组成。

广播格式，首字节由“0000000”+“写比特”组成。

14.5.6 时钟同步

所有主机在 SCL 线上产生它们自己的时钟来传输 I2C 总线上的信息。数据只在时钟的高电平周期有效。因此需要一个确定的时钟进行逐位仲裁。

时钟同步通过线与连接 I2C 接口到 SCL 线来执行。这就是说：SCL 线的高到低切换会使器件开始数它们的低电平周期，而且一旦器件的时钟变低电平，它会使 SCL 线保持这种状态直到到达时钟的高电平。但是，如果另一个时钟仍处于低电平周期，这个时钟的低到高切换不会改变 SCL 线的状态。因此，SCL 线被有最长低电平周期的器件保持低电平。此时，低电平周期短的器件会进入高电平的等待状态。

当所有有关的器件数完了它们的低电平周期后，时钟线被释放并变成高电平。之后，器件时钟和 SCL 线的状态没有差别。而且所有器件会开始数它们的高电平周期。首先完成高电平周期的器件会再次将 SCL 线拉低。

这样产生的同步 SCL 时钟的低电平周期由低电平时钟周期最长的器件决定而高电平周期由高电平时钟周期最短的器件决定。

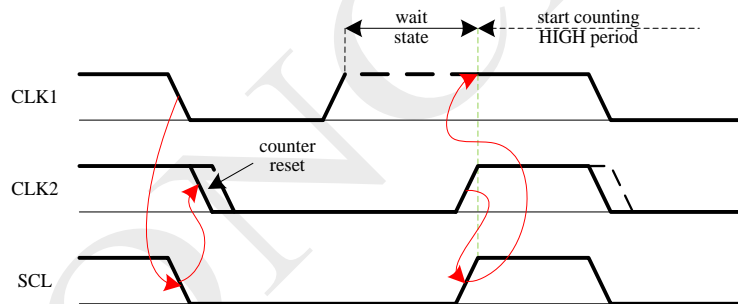


Figure 125 仲裁过程中的时钟同步

14.5.7 仲裁

主机只能在总线空闲的时候启动传输，两个或多个主机可能同时产生一个起始条件。

当 SCL 线是高电平时，仲裁在 SDA 线发生；这样在其他主机发送低电平时，发送高电平的主机将断开它的数据输出级，因为总线上的电平与它自己的电平不相同。

仲裁可以持续多位。它的第一个阶段是比较地址位。如果每个主机都尝试寻址相同的器件，仲裁会继续比较数据位（如果是主机发送模式），或者比较响应位（如果是主机接收模式）。因为 I2C 总线的地址和数据信息由赢得仲裁的主机决定，在仲裁过程中不会丢失信息。

丢失仲裁的主机立即切换到它的从机模式，置位寄存器 I2CISC.ALINT。

Figure 126 显示了两个主机的仲裁过程。图中产生 DATA1 的主机的内部数据电平与 SDA 线的实际电平不同，关断数据输出就意味着总线连接了一个高电平，这不会影响赢得仲裁的主机的数据传输。

必须特别注意的是：在串行传输时当重复起始条件或停止条件发送到 I2C 总线的时候，仲裁过程仍在进行。如果可能产生这样的情况，有关的主机必须在帧格式相同位置发送重复起始条件或停止条件。也就是说，仲裁不能在下面情况之间进行：

- 重复起始条件和数据位
- 停止条件和数据位
- 重复起始条件和停止条件

从机不被卷入仲裁过程。

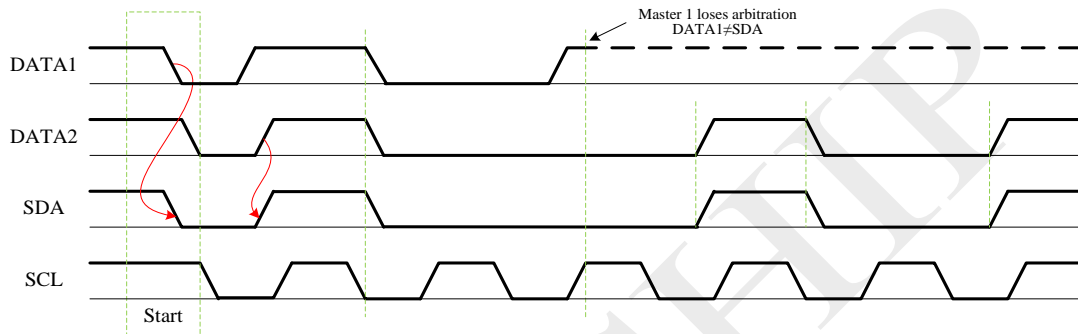


Figure 126 两个主机的仲裁过程

14.6 I2C 时钟产生

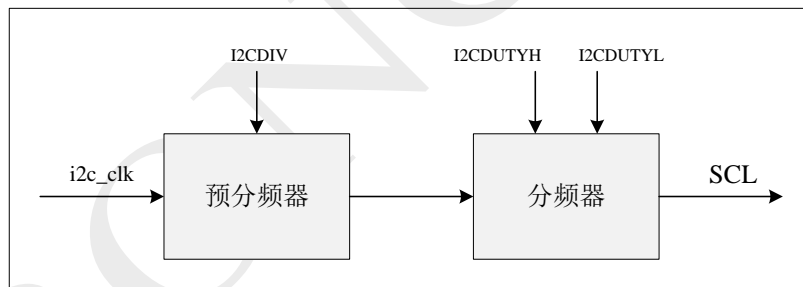


Figure 127 I2C 时钟产生示意图

主机模式下，I2C 串行时钟 SCL 由模块工作时钟 i2c_clk 经过分频得到。

SCL 高电平周期: $T_{SCL_HIGH} = T_{i2c_clk} * NDIV * (NHIGH + 1)$

SCL 低电平周期: $T_{SCL_LOW} = T_{i2c_clk} * NDIV * (NLOW + 1)$

SCL 下降沿到 SDA 变化时间: $T_{HOLD} = T_{i2c_clk} * NDIV * (NHOLD + 1)$

其中，NDIV 见寄存器 I2CDIV 说明，NHIGH 见寄存器 I2CDUTYH 说明，NLOW 见寄存器 I2CDUTYL 说明，NHOLD 见寄存器 I2CHOLD 说明。

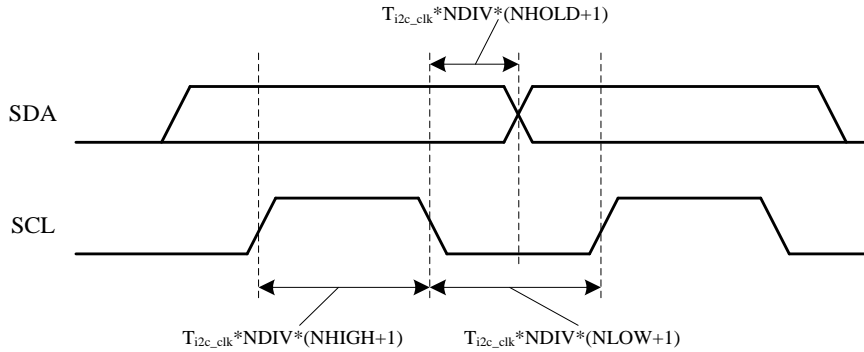


Figure 128 I2C 时钟分频

14.7 I2C 工作模式

14.7.1 从机模式

14.7.1.1 从机地址匹配

I2C 接口默认条件下处于从机模式，一旦检测到起始条件，在 SDA 线上接收到的地址被送到移位寄存器，并与 I2C 本机地址寄存器 I2COADDH 和 I2COADDL 比较，或者和广播呼叫地址比较，若比对结果一致，则主机要对本从机进行访问。

器件地址匹配的标准：

- 若 I2COADDRH.O10BEN=0，即 7 位地址时，接收的第一个字节的地址部分和 I2COADDL[6:0]一致时，器件地址匹配。
- 若 I2COADDRH.O10BEN=1，即 10 位地址时，接收的第一个字节的帧头部分为“11110”+“2 位地址”+“读写控制比特”，其中“2 位地址”和 I2COADDH[1:0]一致；接收的第二个字节和 I2COADDL[7:0]一致。当两个条件都满足时，器件地址匹配。
- 若 I2CCON.ADDR0EN=1，即广播模式使能时，接收的第一个字节为“00000000”时，器件地址匹配。

14.7.1.2 从机接收模式

当从机地址匹配后，且“读写控制比特”为 0，I2C 进入从机接收模式。

每接收一个数据，都会产生 RXINT 中断，需要读取 I2CRBUF 中数据，才能释放 SCL 线。若需继续从主机接收数据，软件参照 EV1 进行配置，产生 ACK 条件；若需停止接收数据，软件参照 EV2 进行配置，产生 NACK 条件。

图中 EV_x 代表软件响应中断之后所要完成的任务，以下类同。

EV1：（1）清除 RXINT；（2）配置 I2CCON.NACK 为 0；（3）读取 I2CRBUF 数据。

EV2：（1）清除 RXINT；（2）配置 I2CCON.NACK 为 1；（3）读取 I2CRBUF 数据。

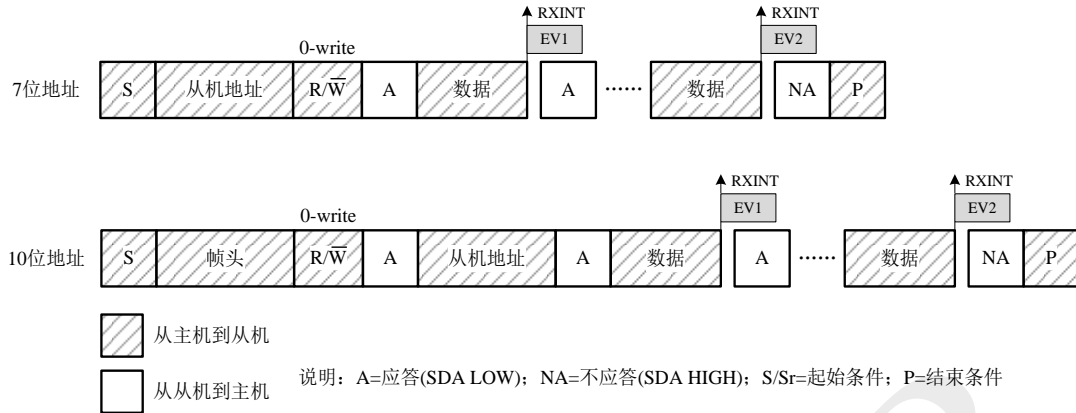


Figure 129 从机接收模式

14.7.1.3 从机发送模式

当从机地址匹配后，且“读写控制比特”为 1，I2C 进入从机发送模式。

地址匹配后，从机发送反馈信号，同时产生 TXINT 中断，需软件向寄存器 I2CWBUF 中写入数据，才能释放 SCL 线。

EV3: (1) 清除 TXINT; (2) 向 I2CWBUF 写入数据。

EV4: (1) 清除 TXINT; (2) 向 I2CWBUF 写入任意数据。

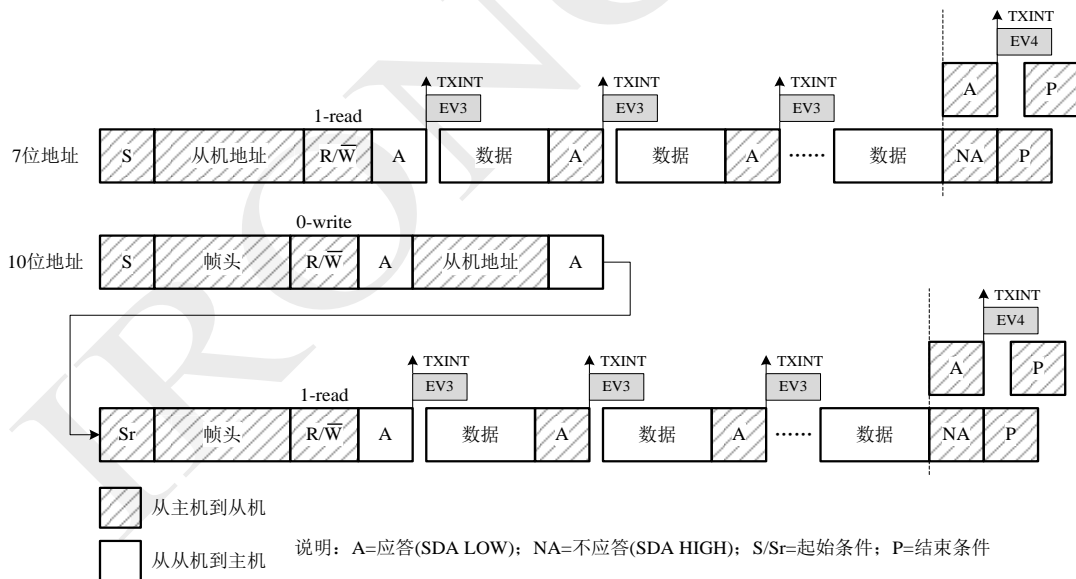


Figure 130 从机发送模式

14.7.1.4 从机混合模式

当从机地址匹配后，且“读写控制比特”为 0，I2C 进入从机接收模式。每接收一个数据，都会产生 RXINT 中断，需要读取 I2CRBUF 中数据，才能释放 SCL 线。若需继续从主机接收数据，软件参照 EV1 进行配置，产生 ACK 条件。

在从机接收数据过程中，未发送 NACK 条件，而再次检测到 Sr 条件，且从机地址匹配，“读写控制比特”为 1，I2C 进入从机发送模式。从机发送反馈信号，同时产生 TXINT 中断，需软件向寄存器 I2CWBUF 中写入数据，才能释放 SCL 线。

EV1: (1) 清除 RXINT; (2) 读取 I2CRBUF 数据。

EV3: (1) 清除 TXINT; (2) 向 I2CWBUF 写入数据。

EV4: (1) 清除 TXINT; (2) 向 I2CWBUF 写入任意数据。

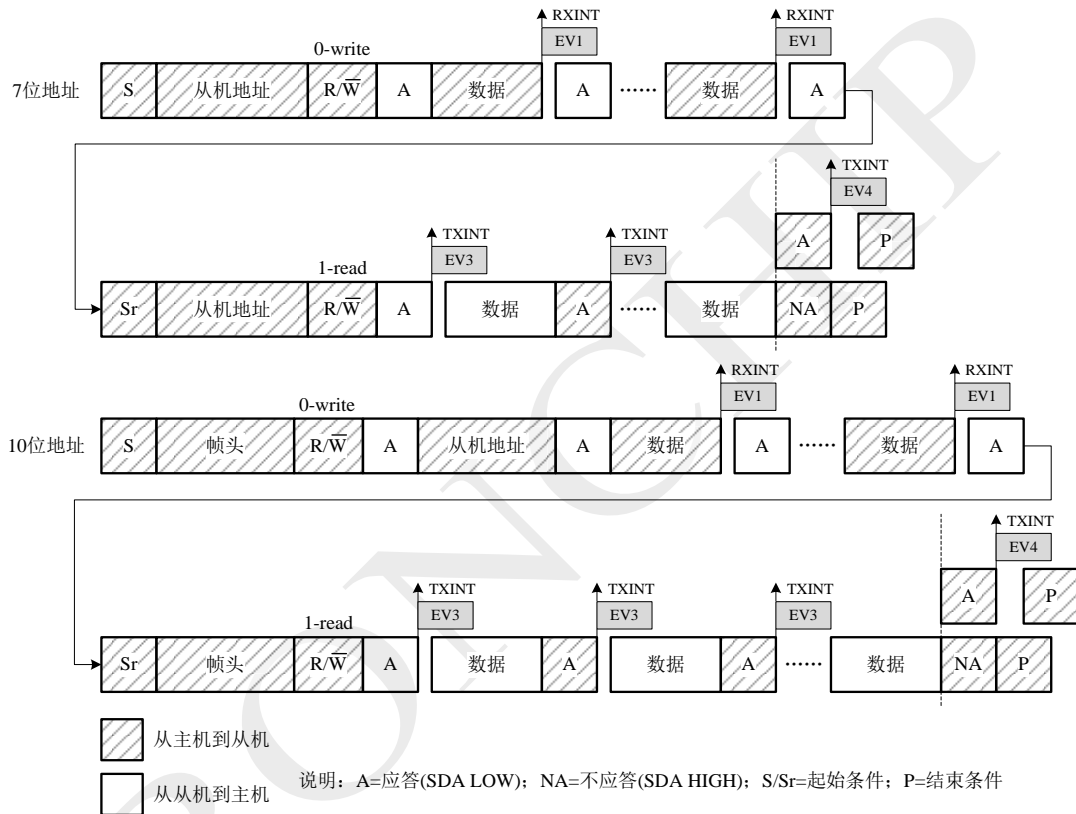


Figure 131 从机混合模式

14.7.2 主机模式

14.7.2.1 主机发送从机地址

配置 I2CCON.MSSEL 为 1，I2C 接口进入主机模式。

根据地址格式，配置 I2CSADDRH 和 I2CSADDRL:

- 7 位地址：配置 I2CSADDRH.S10BEN 为 0，向 I2CSADDRL[6:0]中写入要访问的从机 7 位地址。
- 10 位地址：配置 I2CSADDRH.S10BEN 为 1，向 I2CSADDRH[1:0]和 I2CSADDRL[7:0]中写入要访问的从机 10 位地址。
- 广播模式：向 I2CSADDRL[6:0]中写入“0000000”。

14.7.2.2 主机发送模式

配置寄存器 I2CCON.RW 为 0，配置寄存器 I2CCON.START 为 1，I2C 主机启动主机发送模式。若发送的从机地址为“000000”，则为广播模式。

若 I2C 总线中存在相应地址的从机，并发出反馈，I2C 模块产生 TXINT 中断，向 I2CWBUF 中写入数据，才能释放 SCL 线。若接收到 ACK 条件并需向从机发送数据，软件参照 EV1 进行配置；若接收到 ACK 条件并需停止发送数据，产生 STOP 条件，则参照 EV2 进行配置；若接收到 NACK 条件，硬件自动产生 STOP 条件。

EV1：（1）清除 TXINT；（2）向 I2CWBUF 写入主机向从机发送的数据。

EV2：（1）清除 TXINT （2）配置 I2CCON.STOP 为 1，主机产生 STOP 条件。

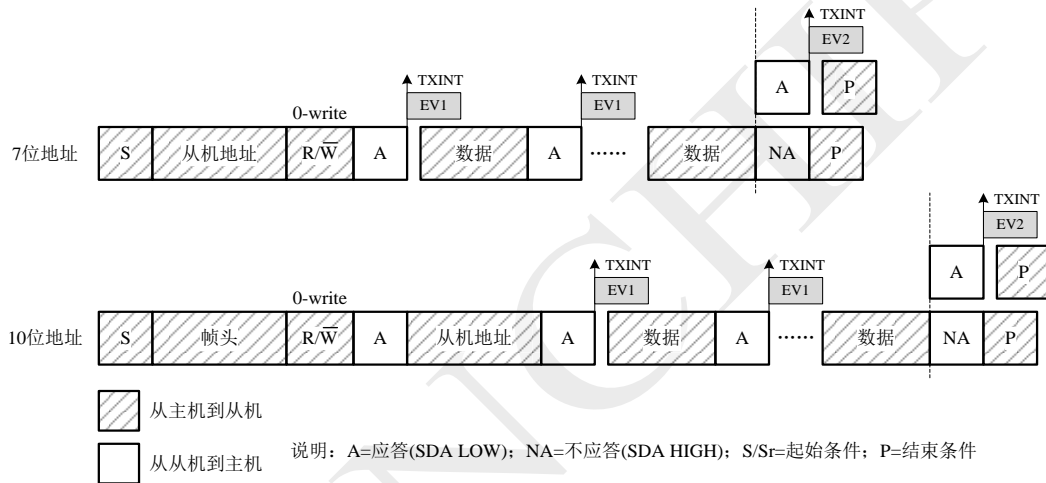


Figure 132 主机发送模式

14.7.2.3 主机接收模式

若为 7 位地址接收模式，配置寄存器 I2CCON.RW 为 1，配置寄存器 I2CCON.START 为 1，I2C 主机启动主机接收模式；若为 10 位地址接收模式，配置寄存器 I2CCON.RW 为 0，配置寄存器 I2CCON.START 为 1，I2C 总线中存在相应地址的从机，并发出反馈，帧头和从机地址发送完成后，会产生 TXINT 中断，软件参照 EV5 进行配置，产生 Sr 条件，并发送帧头信息，I2C 主机启动主机接收模式。

若 I2C 总线中存在相应地址的从机，并发出反馈，I2C 主机开始接收数据，每接收一个数据，都会产生 RXINT 中断，需要读取 I2CRBUF 中数据，才能释放 SCL 线。若需继续接收数据参照 EV3 进行配置；若需产生 STOP 条件，则参照 EV4 进行配置。

EV3：（1）清除 RXINT；（2）配置 I2CCON.NACK 为 0，产生 ACK 条件；（3）读取 I2CRBUF 数据。

EV4：（1）清除 RXINT；（2）配置 I2CCON.NACK 为 0 或 1，产生 ACK 或 NACK 条件；（3）配置 I2CCON.STOP 为 1，产生 STOP 条件；（4）读取 I2CRBUF 数据。

EV5：（1）清除 TXINT；（2）配置 I2CCON.START 为 1；（3）配置 I2CCON.RW 为 1。

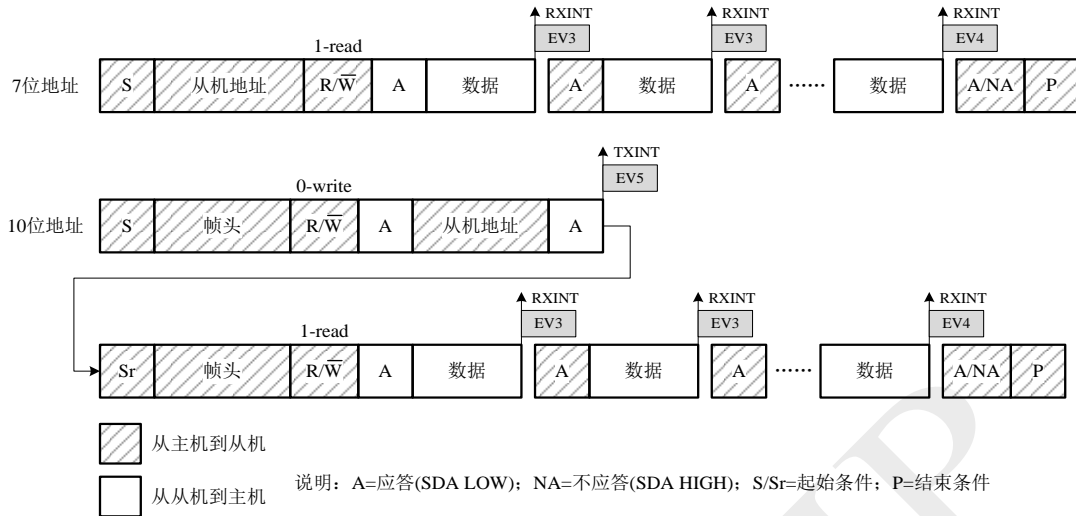


Figure 133 主机接收模式

14.7.2.4 主机混合模式

配置寄存器 I2CCON.RW 为 0，配置寄存器 I2CCON.START 为 1，I2C 主机启动主机发送模式。

若 I2C 总线中存在相应地址的从机，并发出反馈，I2C 模块产生 TXINT 中断，向 I2CWBUF 中写入数据，才能释放 SCL 线。若需向从机发送数据，软件参照 EV1 进行配置。

数据发送完后，参照 EV5 进行配置，产生 Sr（重新启动）条件，使 I2C 主机再次启动进入主机接收模式。

I2C 主机接收到从机反馈，并开始接收数据，每接收一个数据，都会产生 RXINT 中断，需要读取 I2CRBUF 中数据，才能释放 SCL 线。若需继续接收数据参照 EV3 进行配置，若需产生 STOP 条件，则参照 EV4 进行配置。

EV1: (1) 清除 TXINT; (2) 向 I2CWBUF 写入数据。

EV5: (1) 清除 TXINT; (2) 配置 I2CCON.START 为 1; (3) 配置 I2CCON.RW 为 1。

EV3: (1) 清除 RXINT; (2) 配置 I2CCON.NACK 为 0; (3) 读取 I2CRBUF 数据。

EV4: (1) 清除 RXINT; (2) 配置 I2CCON.NACK 为 0 或 1; (3) 配置 I2CCON.STOP 为 1; (4) 读取 I2CRBUF 数据。

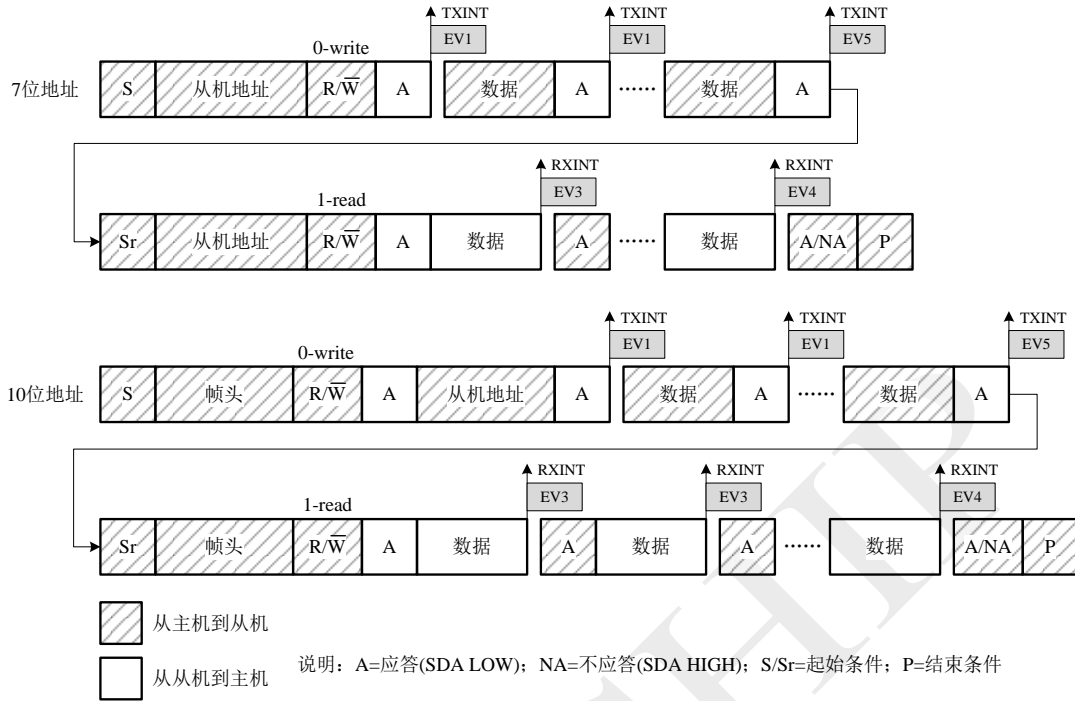


Figure 134 主机混合模式

14.8 I2C 中断

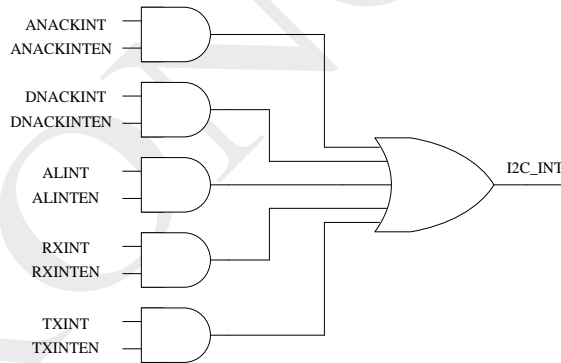


Figure 135 I2C 中断

I2C 有 5 个中断源：地址 NACK 中断 (ANACKINT)、数据 NACK 中断 (DNACKINT)、仲裁丢失中断 (ALINT)、接收中断 (RXINT) 和发送中断 (TXINT)。

I2C 中断配置及处理：

- 配置 I2CIEN 寄存器使能相关中断源；
- 配置 IEN2.I2CINT 寄存器为 1，使能 I2C 中断；
- 当产生 I2C 中断后，软件需要回读 I2CISC 寄存器，来判断 I2C 中断源，根据不同中断源，向 I2CISC 寄存器相应位写 0 清除，并做相应处理。

14.9 I2C 寄存器

I2C 模块有 14 个寄存器，对应的 SFRPAGE 为 0，具体说明如 Table 239 所示。

Table 239 I2C 寄存器 (SFRPAGE=0x00)

地址	寄存器名	类型	复位值	功能描述
0xE1	I2CCON	W/R	0x00	I2C 控制寄存器
0xE2	I2CSADDRL	W/R	0x00	I2C 从机地址寄存器低 8 位
0xE3	I2CSADDRH	W/R	0x00	I2C 从机地址寄存器高 2 位
0xE4	I2COADDRL	W/R	0x00	I2C 本机地址寄存器低 8 位
0xE5	I2COADDRH	W/R	0x00	I2C 本机地址寄存器高 2 位
0xE6	I2CDIV	W/R	0x02	I2C 工作时钟分频控制寄存器
0xE7	I2CDUTYL	W/R	0x0e	I2C 时钟 SCL 低电平时间配置寄存器
0xE9	I2CDUTYH	W/R	0x0e	I2C 时钟 SCL 高电平时间配置寄存器
0xEA	I2CHOLD	W/R	0x06	I2C 数据 SDA 保持时间配置寄存器
0xEB	I2CWBUF	W/R	0x00	I2C 数据写缓冲寄存器
0xEC	I2CRBUF	R	0x00	I2C 数据读缓冲寄存器
0xED	I2CSTS	R	0x42	I2C 状态寄存器
0xEE	I2CISC	W/R	0x00	I2C 中断标志寄存器
0xEF	I2CIEN	W/R	0x00	I2C 中断使能寄存器

14.9.1 I2C 控制寄存器(I2CCON) [0xE1]

Table 240 I2C 控制寄存器(I2CCON)

参数名	比特位	属性	复位值	描述
MSSEL	0	W/R	0	主机/从机模式选择： 0：从机模式 1：主机模式
START	1	W/R	0	产生起始条件，仅在主机模式下有效，当检测到总线上的起始条件后，硬件自动清除该位： 0：无效 1：产生起始条件
STOP	2	W/R	0	产生停止条件，仅在主机模式下有效，当检测到总线上的停止条件后，硬件自动清除该位：

参数名	比特位	属性	复位值	描述
				0: 无效 1: 产生停止条件
NACK	3	W/R	0	产生无效应答, 当检测到总线上的起始条件或停止条件后, 硬件自动清除该位: 0: 无效 1: 产生无效应答
RW	4	W/R	0	I2C 总线读写传输控制, 仅在主机模式下有效: 0: 写 (发送) 操作 1: 读 (接收) 操作
ADDR0EN	5	W/R	0	广播呼叫使能, 仅在从机模式下有效: 0: 禁止, 不产生应答 1: 使能, 产生应答
LBM	6	W/R	0	在主机模式下, I2C 回环测试模式使能: 0: 不使能 1: 使能
Reserved	7	R	0	保留

14.9.2 I2C 从机地址寄存器低 8 位(I2CSADDRL) [0xE2]

Table 241 I2C 从机地址寄存器低 8 位(I2CSADDRL)

参数名	比特位	属性	复位值	描述
I2CSADDRL	7-0	W/R	0	从机地址: 7 位从机地址模式时, 低 7 位为从机地址; 10 位从机地址模式时, 为从机地址的低 8 位

14.9.3 I2C 从机地址寄存器高 2 位(I2CSADDRH) [0xE3]

Table 242 I2C 从机地址寄存器高 2 位(I2CSADDRH)

参数名	比特位	属性	复位值	描述
I2CSADDRH	1-0	W/R	0	从机地址: 7 位从机地址模式时, 该位无效; 10 位从机地址模式时, 为从机地址的高 2 位

参数名	比特位	属性	复位值	描述
Reserved	6-2	R	0	保留
S10BEN	7	W/R	0	10 位从机地址模式使能： 0: 7 位从机地址模式 1: 10 位从机地址模式

14.9.4 I2C 本机地址寄存器低 8 位(I2COADDRL) [0xE4]

Table 243 I2C 本机地址寄存器低 8 位(I2COADDRL)

参数名	比特位	属性	复位值	描述
I2COADDRL	7-0	W/R	0	本机地址： 7 位本机地址模式时，低 7 位为本机地址； 10 位本机地址模式时，为本机地址的低 8 位

14.9.5 I2C 本机地址寄存器高 2 位(I2COADDRH) [0xE5]

Table 244 I2C 本机地址寄存器高 2 位(I2COADDRH)

参数名	比特位	属性	复位值	描述
I2COADDRH	1-0	W/R	0	本机地址： 7 位本机地址模式时，该位无效； 10 位本机地址模式时，为本机地址的高 2 位
Reserved	6-2	R	0	保留
O10BEN	7	W/R	0	10 位本机地址模式使能： 0: 7 位本机地址模式 1: 10 位本机地址模式

14.9.6 I2C 工作时钟分频控制寄存器(I2CDIV) [0xE6]

Table 245 I2C 工作时钟分频控制寄存器(I2CDIV)

参数名	比特位	属性	复位值	描述
NDIV	7-0	W/R	00000010	I2C 工作时钟分频控制： 00000000: 1 分频 00000001: 1 分频 00000010: 2 分频 00000011: 3 分频

参数名	比特位	属性	复位值	描述
			 11111110: 254 分频 11111111: 255 分频

14.9.7 I2C 时钟 SCL 低电平时间配置寄存器(I2CDUTYL) [0xE7]

Table 246 I2C 时钟 SCL 低电平时间配置寄存器(I2CDUTYL)

参数名	比特位	属性	复位值	描述
NLOW	7-0	W/R	00001110	I2C 时钟 SCL 低电平时间配置: 00000000: $3 * NDIV * T_{i2c_clk}$ 00000001: $3 * NDIV * T_{i2c_clk}$ 00000010: $3 * NDIV * T_{i2c_clk}$ 00000011: $4 * NDIV * T_{i2c_clk}$ 11111110: $255 * NDIV * T_{i2c_clk}$ 11111111: $256 * NDIV * T_{i2c_clk}$

14.9.8 I2C 时钟 SCL 高电平时间配置寄存器(I2CDUTYH) [0xE9]

Table 247 I2C 时钟 SCL 高电平时间配置寄存器(I2CDUTYH)

参数名	比特位	属性	复位值	描述
NHIGH	7-0	W/R	00001110	I2C 时钟 SCL 高电平时间配置: 00000000: $3 * NDIV * T_{i2c_clk}$ 00000001: $3 * NDIV * T_{i2c_clk}$ 00000010: $3 * NDIV * T_{i2c_clk}$ 00000011: $4 * NDIV * T_{i2c_clk}$ 11111110: $255 * NDIV * T_{i2c_clk}$ 11111111: $256 * NDIV * T_{i2c_clk}$

14.9.9 I2C 数据 SDA 保持时间配置寄存器(I2CHOLD) [0xEA]

Table 248 I2C 数据 SDA 保持时间配置寄存器(I2CHOLD)

参数名	比特位	属性	复位值	描述
NHOLD	7-0	W/R	00001110	I2C 数据 SDA 保持时间配置, 即 SDA 在 SCL 拉低后数据的保持时间, 在时序满足的情况下, 尽量小于 $(NLOW+1)/2$:

参数名	比特位	属性	复位值	描述
				00000000: $1 * NDIV * T_{i2c_clk}$ 00000001: $1 * NDIV * T_{i2c_clk}$ 00000010: $2 * NDIV * T_{i2c_clk}$ 00000011: $3 * NDIV * T_{i2c_clk}$ 11111110: $254 * NDIV * T_{i2c_clk}$ 11111111: $255 * NDIV * T_{i2c_clk}$

14.9.10 I2C 数据写缓冲寄存器(I2CWBUF) [0xEB]

Table 249 I2C 数据写缓冲寄存器(I2CWBUF)

参数名	比特位	属性	复位值	描述
I2CWBUF	7-0	W/R	0	I2C 数据写缓冲

14.9.11 I2C 数据读缓冲寄存器(I2CRBUF) [0xEC]

Table 250 I2C 数据读缓冲寄存器(I2CRBUF)

参数名	比特位	属性	复位值	描述
I2CRBUF	7-0	R	0	I2C 数据读缓冲

14.9.12 I2C 状态寄存器(I2CSTS) [0xED]

Table 251 I2C 状态寄存器(I2CSTS)

参数名	比特位	属性	复位值	描述
BUSY	0	R	0	I2C 总线忙状态指示: 0: 总线空闲 1: 总线忙
WBUFS	1	R	1	写缓冲状态指示: 0: 写缓冲满 1: 写缓冲空
RBUFS	2	R	0	读缓冲状态指示: 0: 读缓冲空 1: 读缓冲满
ADDR0	3	R	0	广播呼叫使能且在从机模式的情况下, 指示是否发生广播呼叫: 0: 未发生广播呼叫

参数名	比特位	属性	复位值	描述
				1: 发生广播呼叫
AAS	4	R	0	本机被作为从机寻址标志位, 发生广播呼叫时, 该位置 1: 0: 本机未被寻址 1: 本机被寻址
SDIR	5	R	0	本机被作为从机寻址时, 本机传输方向指示: 0: 接收 1: 发送
SCL	6	R	1	I2C 时钟 SCL 线状态指示: 0: 低电平 1: 高电平
Reserved	7	R	0	保留

14.9.13 I2C 中断标志寄存器(I2CISC) [0xEE]

Table 252 I2C 中断标志寄存器(I2CISC)

参数名	比特位	属性	复位值	描述
ANACKINT	0	W/R	0	地址未被应答中断标志位: <ul style="list-style-type: none"> ● 主机模式发送从机地址后, 未被应答时, 该位置为 1 ● 需要软件写 0 清除 ● 不支持写 1 软中断
DNACKINT	1	W/R	1	数据未被应答中断标志位: <ul style="list-style-type: none"> ● 数据发送后, 未被应答时, 该位置为 1 ● 需要软件写 0 清除 ● 不支持写 1 软中断
ALINT	2	W/R	0	丢失仲裁中断标志位: <ul style="list-style-type: none"> ● 主机模式时, 丢失仲裁后, 该位置为 1 ● 需要软件写 0 清除 ● 不支持写 1 软中断
RXINT	3	W/R	0	数据接收中断标志位: <ul style="list-style-type: none"> ● 接收到数据后, 该位置为 1

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> ● 需要软件写 0 清除 ● 不支持写 1 软中断
TXINT	4	W/R	0	数据发送请求中断标志位： <ul style="list-style-type: none"> ● 存在数据发送请求时，该位置为 1 ● 需要软件写 0 清除 ● 不支持写 1 软中断
Reserved	7-5	R	0	保留

14.9.14 I2C 中断使能寄存器(I2CIEN) [0xEF]

Table 253 I2C 中断使能寄存器(I2CIEN)

参数名	比特位	属性	复位值	描述
ANACKINTEN	0	W/R	0	地址未被应答中断使能： 0：不使能 1：使能
DNACKINTEN	1	W/R	1	数据未被应答中断使能： 0：不使能 1：使能
ALINTEN	2	W/R	0	丢失仲裁中断使能： 0：不使能 1：使能
RXINTEN	3	W/R	0	数据接收中断使能： 0：不使能 1：使能
TXINTEN	4	W/R	0	数据发送请求中断使能： 0：不使能 1：使能
Reserved	7-5	R	0	保留

15 串行外设接口 (SPI)

15.1 概述

JMT1801ED 的串行外设接口 (SPI) 具有以下特征:

- 同步串行传输
- 支持 4 线全双工或 3 线半双工传输
- 8 位数据帧格式
- 数据 MSB 在前和 LSB 在前可选择
- 支持主机或从机模式
- 时钟极性和相位可配置
- 片选极性可配置
- 可触发接收/发送中断和接收溢出中断
- 支持回环测试模式
- 主机波特率可配置
- 主机发送最高频率为 $f_{\text{spi_clk}}/2$, 主机接收最高频率为 $f_{\text{spi_clk}}/4$
- 从机发送最高频率为 $f_{\text{spi_clk}}/4$, 从机接收最高频率为 $f_{\text{spi_clk}}/4$

15.2 框图

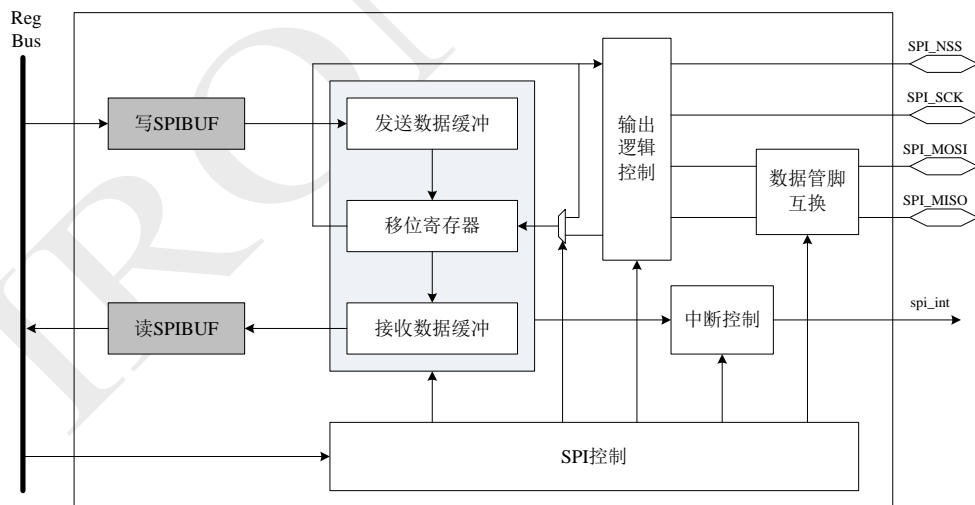


Figure 136 SPI 结构框图

15.3 SPI 管脚配置

SPI 的管脚配置可参见 Table 1 引脚说明表格, 管脚配置说明见 GPIO 寄存器说明。

15.4 SPI 主/从机互连

SPI 的互连方式有 4 种：4 线单从机，4 线多从机，3 线单从机和 3 线多从机。

15.4.1 4 线互连模式

4 线互连模式分为单从机和多从机方式。

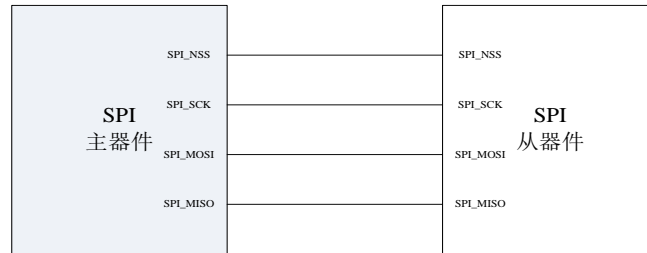


Figure 137 4 线单从机互连

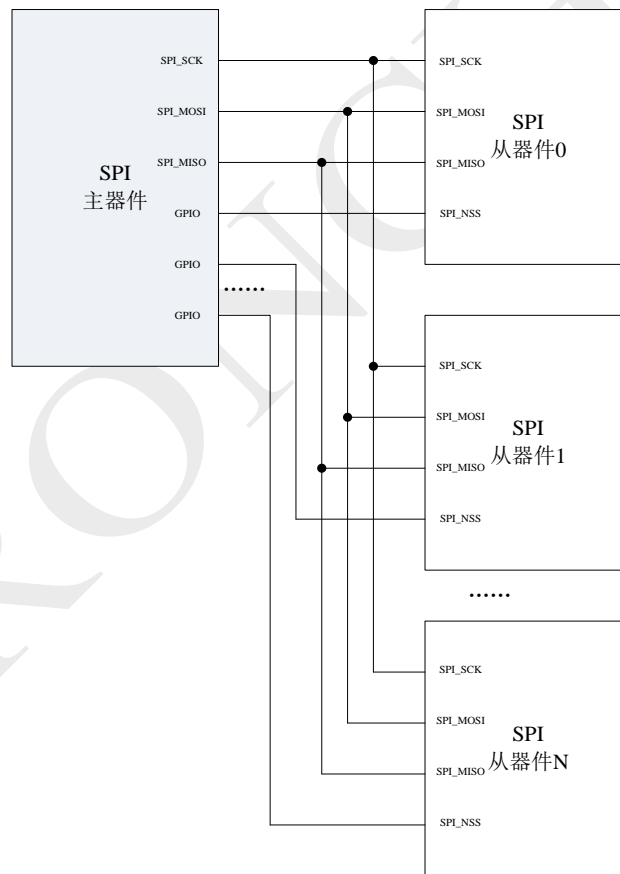


Figure 138 4 线多从机互连

15.4.2 3 线互连模式

3 线互连模式分为单从机和多从机方式。在 3 线互连模式下，寄存器 SPIBR.SDIOSWAP 可以将 SPI 的 SPI_MOSI 和 SPI_MISO 管脚功能进行互换。

JMT1801ED SPI 作为主机时，SPI_MOSI 默认作为主机数据收发管脚，可配置寄存器 SPIBR.SDIOSWAP 为 1，使用 SPI_MISO 作为主机数据收发管脚。

JMT1801ED SPI 作为从机时，SPI_MISO 默认作为从机的数据收发管脚，可配置寄存器 SPIBR.SDIOSWAP 为 1，使用 SPI_MOSI 作为主机数据收发管脚。

如 Figure 139 所示为 3 线默认连接方式：

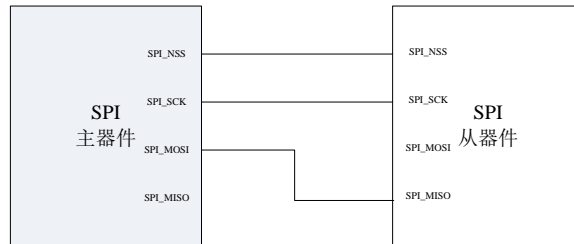


Figure 139 3 线单从机互连

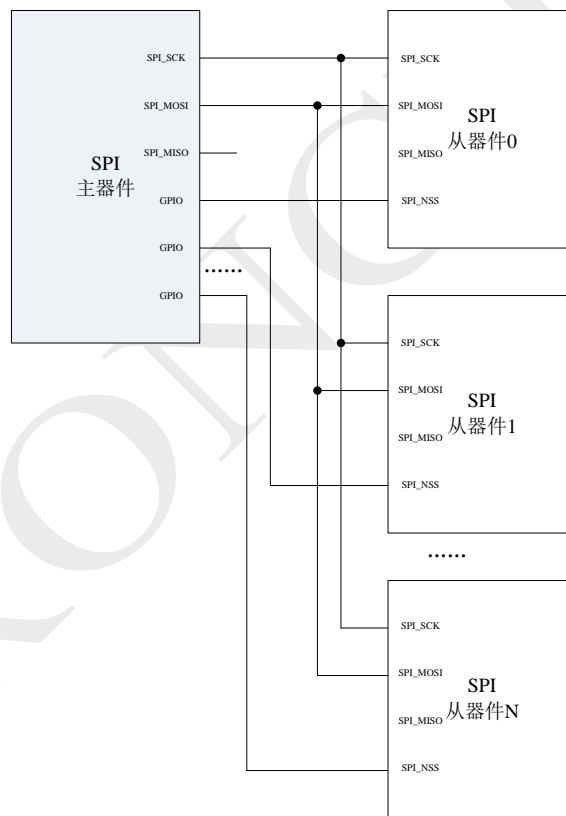


Figure 140 3 线多从机互连

15.5 SPI 时序设置

SPI 的时序配置如下：

- 配置寄存器 SPICON.PHA 选择时钟 SPI_SCK 相位；
- 配置寄存器 SPICON.POL 选择时钟 SPI_SCK 极性；
- 配置寄存器 SPICON.FRP 选择片选 SPI_NSS 极性；

- 配置寄存器 SPICON.LSB 选择数据 SPI_MOSI 和 SPI_MISO 格式。

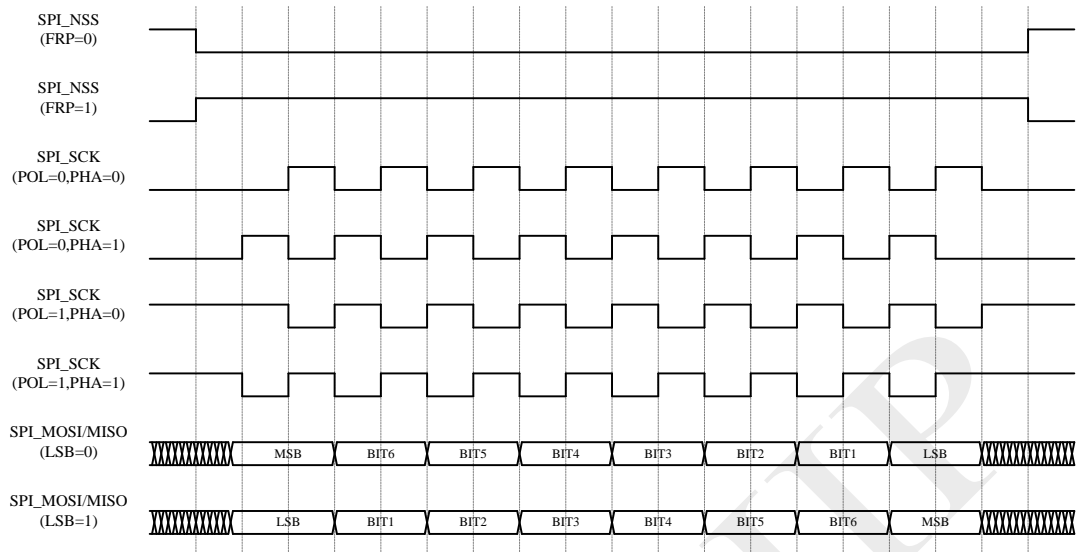


Figure 141 SPI 时序

SPI 从机根据主机提供的 SPI_SCK 时钟发送数据，由于管脚和线路的延时，主机按照默认的相位去采样 SPI_MISO 数据可能来不及，SPI 模块通过配置 SPISPC 寄存器，可以选择不同的主机采样相位，如图 Figure 142 所示。

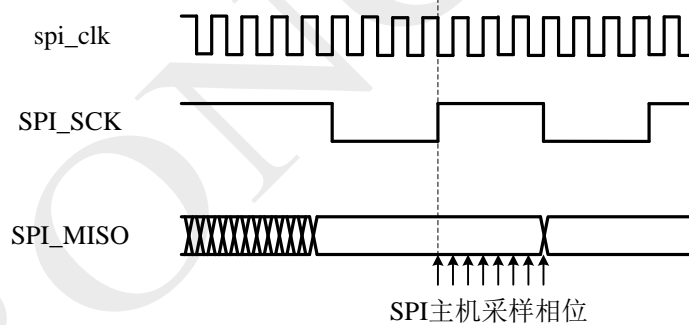


Figure 142 SPI 主机采样相位

15.6 数据发送和接收过程

15.6.1 主机模式

设置寄存器 SPICON.MS 为 0，使 SPI 工作于主机模式。

SPI 通过 SPI_SCK 管脚为整个串行通信提供时钟，数据从 SPI_MOSI 管脚输出，从 SPI_MISO 管脚读入，并通过 SPI_NSS 管脚发送从机 SPI 设备的片选，SPI_NSS 管脚在数据传输之前被置为有效值，在数据传输完毕后被置为无效值。

SPI 总线上的数据传输都由 SPI 主机启动。数据写入 SPIBUF 寄存器来启动在 SPI_MOSI 管脚的数据传输，数据发送顺序可通过 SPICON.LSB 寄存器设置，同时，SPI_MISO 管脚上的接收数据延时 SPISPC（通过寄存器 SPISPC 配置）个 spi_clk 时钟后，移位到内部移位寄

寄存器的最低比特。当数据的 8 个比特发送完后，接收数据转移到接收缓冲中，可以通过 SPIBUF 寄存器读取。

15.6.1.1 4 线主机模式

4 线主机模式收发数据流程：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 0，使 SPI 工作于主机模式
- 设置寄存器 SPICON.TWI 为 0，选择 4 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 设置寄存器 SPIBR，选择 SPI 波特率
- 设置寄存器 SPICON.LSB，选择 SPI 数据比特顺序
- 设置寄存器 SPISPC，选择主机采样延时
- 向寄存器 SPIBUF 中写入发送数据，启动 SPI 传输
- 等待 SPI 中断或查询 SPIIF.TCINT 标志位置 1 后，传输完成，可通过 SPIBUF 读取接收的从机数据

15.6.1.2 3 线主机模式

3 线主机模式发送数据流程：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 0，使 SPI 工作于主机模式
- 设置寄存器 SPICON.TWI 为 0，选择 3 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 设置寄存器 SPIBR，选择 SPI 波特率
- 设置寄存器 SPICON.LSB，选择 SPI 数据比特顺序
- 向寄存器 SPIBUF 中写入发送数据，启动 SPI 传输
- 等待 SPI 中断或查询 SPIIF.TCINT 标志位置 1 后，传输完成

3 线主机模式接收数据流程：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 0，使 SPI 工作于主机模式

- 设置寄存器 SPICON.TWI 为 0，选择 3 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 1，关闭 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 设置寄存器 SPIBR，选择 SPI 波特率
- 设置寄存器 SPICON.LSB，选择 SPI 数据比特顺序
- 设置寄存器 SPISPC，选择主机采样延时
- 向寄存器 SPIBUF 中写入任意数据，启动 SPI 传输
- 等待 SPI 中断或查询 SPIIF.TCINT 标志位置 1 后，传输完成，可通过 SPIBUF 读取接收的从机数据

15.6.2 从机模式

设置寄存器 SPICON.MS 为 1，使 SPI 工作于从机模式。

SPI 通过 SPI_SCK 管脚接收串行时钟，并只在 SPI_NSS 管脚为有效值时，从 SPI_MOSI 管脚读入数据，并向 SPI_MISO 管脚输出数据。

SPI 从机只能被动的收发数据，不能作为通信的发起者。当检测到 SPI_NSS 片选有效时，SPI 从机开始根据 SPI_SCK 管脚的时钟从 SPI_MOSI 管脚读取数据，移位到内部移位寄存器的最低比特，同时，启动将 SPIBUF 中数据在 SPI_MISO 管脚上的传输，数据发送顺序可通过 SPICON.LSB 寄存器设置。当数据的 8 个比特接收完后，接收数据转移到接收缓冲中，可以通过 SPIBUF 寄存器读取。

15.6.2.1 4 线从机模式

4 线从机模式收发数据流程：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 1，使 SPI 工作于从机模式
- 设置寄存器 SPICON.TWI 为 0，选择 4 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 设置寄存器 SPIBR，选择 SPI 波特率
- 设置寄存器 SPICON.LSB，选择 SPI 数据比特顺序
- 向寄存器 SPIBUF 中写入发送数据，硬件将根据 SPI 总线 SPI_NSS 和 SPI_SCK 状态，从 SPI_MOSI 接收数据，向 SPI_MISO 发送数据

- 等待 SPI 中断或查询 SPIIF.TCINT 标志位置 1 后，传输完成，可通过寄存器 SPIBUF 读取接收的主机数据

15.6.2.2 3 线从机模式

3 线从机模式发送数据流程：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 1，使 SPI 工作于从机模式
- 设置寄存器 SPICON.TWI 为 1，选择 3 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 设置寄存器 SPIBR，选择 SPI 波特率
- 设置寄存器 SPICON.LSB，选择 SPI 数据比特顺序
- 向 SPIBUF 中写入发送数据，硬件将根据 SPI 总线 SPI_NSS 和 SPI_SCK 状态，向 SPI_MISO 发送数据
- 等待 SPI 中断或查询 SPIIF.TCINT 标志位置 1 后，传输完成

3 线从机模式接收数据流程：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 1，使 SPI 工作于从机模式
- 设置寄存器 SPICON.TWI 为 1，选择 3 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 1，关闭 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 设置寄存器 SPIBR，选择 SPI 波特率
- 设置寄存器 SPICON.LSB，选择 SPI 数据比特顺序
- 硬件根据 SPI 总线 SPI_NSS 和 SPI_SCK 状态，从 SPI_MISO 接收数据
- 等待 SPI 中断或查询 SPIIF.TCINT 标志位置 1 后，传输完成，可通过寄存器 SPIBUF 读取接收的主机数据

15.7 SPI 中断

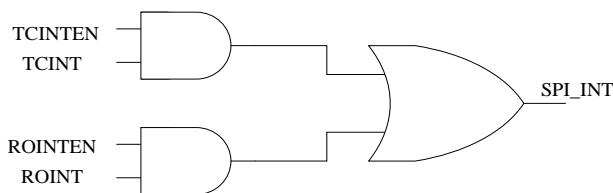


Figure 143 SPI 中断

SPI 有 2 个中断源：发送/接收完成中断 (TCINT) 和接收溢出中断 (ROINT)。其中 TCINT 使能为寄存器 SPIIE.TCINTEN，ROINT 使能为寄存器 SPIIE.ROINTEN。

当 SPI 发送或接收数据完成时，TCINT 标志位置 1。

当 SPI 接收数据缓存发生溢出时，即数据接收后放入 SPIBUF 未被读取，另一个接收数据又将之前接收到的数据覆盖了，ROINT 标志位置 1。

SPI 中断配置及处理：

- 配置 SPIIE 寄存器使能相关中断源；
- 配置 IEN2.SPIINT 寄存器为 1，使能 SPI 中断；
- 当产生 SPI 中断后，软件需要回读 SPIIF 寄存器，来判断 SPI 中断源，根据不同中断源，向 SPIIF 寄存器相应位写 0 清除，并做相应处理。

15.8 SPI 寄存器

SPI 模块有 5 个寄存器，对应的 SFRPAGE 为 0，具体说明如 Table 254 所示。

Table 254 SPI 寄存器 (SFRPAGE=0x00)

地址	寄存器名	类型	复位值	功能描述
0xC9	SPICON	W/R	0x02	SPI 控制寄存器
0xCA	SPIIE	W/R	0x00	SPI 中断使能寄存器
0xCB	SPIF	W/R	0x00	SPI 中断标志寄存器
0xCC	SPIBR	W/R	0x01	SPI 波特率控制寄存器
0xCD	SPIBUF	W/R	0x00	SPI 数据缓冲寄存器
0xCE	SPISPC	W/R	0x00	SPI 主机采样控制寄存器

15.8.1 SPI 控制寄存器(SPICON)[0xC9]

Table 255 SPI 控制寄存器(SPICON)

参数名	比特位	属性	复位值	描述
SPIEN	0	W/R	0	SPI 使能控制：

参数名	比特位	属性	复位值	描述
				0: 不使能 SPI 1: 使能 SPI
MS	1	W/R	1	主机/从机模式选择控制: 0: 主机模式 1: 从机模式
POL	2	W/R	0	SPI 时钟极性控制: 0: SPI_SCK 在非活动期间的极性为低电平 1: SPI_SCK 在非活动期间的极性为高电平
PHA	3	W/R	0	SPI 相位控制: 0: 采样时钟第一个跳变出现在每帧第一比特传输中间处 1: 采样时钟第一个跳变出现在每帧第一比特传输开始处
FRP	4	W/R	0	SPI 片选信号极性控制: 0: SPI 片选低电平有效 1: SPI 片选高电平有效
TWI	5	W/R	0	SPI 3/4 线模式选择: 0: 4 线模式 1: 3 线模式
LSB	6	W/R	0	SPI 数据 LSB 使能控制: 0: MSB, 先发送数据高位 1: LSB, 先发送数据低位
SOD	7	W/R	0	SPI 输出关闭控制: 0: SPI 输出使能 1: SPI 输出关闭

寄存器 TWI (三线 SPI 模式使能), 需要寄存器 SOD 来配合实现三线 SPI:

- 当 SPI 工作于主机模式时, SPI_MOSI 作为输入输出, 主机发送时需要配置寄存器 SOD 为 0, 主机接收时需要配置寄存器 SOD 为 1;
- 当 SPI 工作于从机模式时, SPI_MISO 作为输入输出, 从机发送时需要配置寄存器 SOD 为 0, 从机接收时需要配置寄存器 SOD 为 1。

Figure 144 所示为两个 JMT1801ED SPI 工作于 3 线模式下的主从连接关系, 当主机发送从机接收, 则设置主机的 SOD 为 0, 从机的 SOD 为 1; 当主机接收从机发送, 则设置主机

的 SOD 为 1，从机的 SOD 为 0。

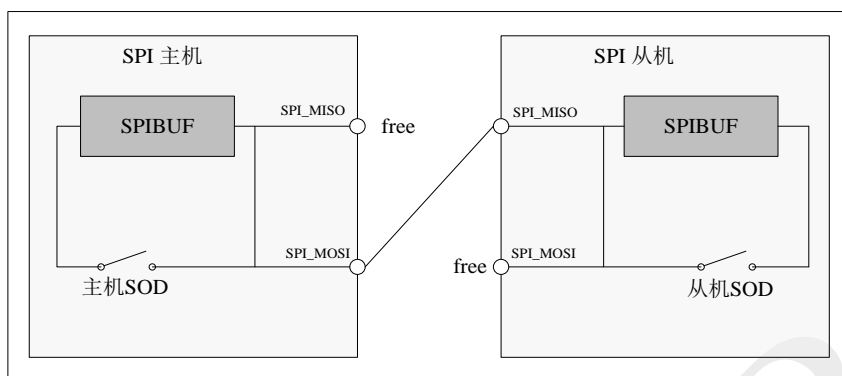


Figure 144 3 线 SPI 连接

15.8.2 SPI 中断使能寄存器(SPIIE)[0xCA]

Table 256 SPI 中断使能寄存器(SPIIE)

参数名	比特位	属性	复位值	描述
TCINTEN	0	W/R	0	SPI 发送/接收完成中断使能： 0：不使能 1：使能
ROINTEN	1	W/R	0	SPI 接收溢出中断使能： 0：不使能 1：使能
Reserved	7-2	R	0	保留

15.8.3 SPI 中断标志寄存器(SPIIF)[0xCB]

Table 257 SPI 中断标志寄存器(SPIIF)

参数名	比特位	属性	复位值	描述
TCINT	0	W/R	0	SPI 发送/接收完成中断标志位： <ul style="list-style-type: none"> ● 当有 SPI 发送/接收完成中断申请时，该位置为 1 ● 软件写 0 清除 ● 不支持写 1 软中断
ROINT	1	W/R	0	SPI 接收溢出中断标志位： <ul style="list-style-type: none"> ● 当有 SPI 接收溢出中断申请时，该位置为 1 ● 软件写 0 清除

参数名	比特位	属性	复位值	描述
				● 不支持写 1 软中断
Reserved	7-2	R	0	保留

15.8.4 SPI 波特率控制寄存器(SPIBR)[0xCC]

Table 258 SPI 波特率控制寄存器(SPIBR)

参数名	比特位	属性	复位值	描述
SPIBR	5-0	W/R	0x01	SPI 波特率控制，计算公式如下： $f_{SPI} = \frac{f_{spi_clk}}{2 * (SPIBR + 1)}$
SDIOSWAP	6	W/R	0	SPI 数据信号 SPI_MISO 和 SPI_MOSI 功能互换使能： 0: 不互换 1: 互换
LBM	7	W/R	0	SPI 回环模式使能，只在 Master 模式下有效： 0: 不使能 1: 使能

15.8.5 SPI 数据缓冲寄存器(SPIBUF) [0xCD]

Table 259 SPI 数据缓冲寄存器(SPIBUF)

参数名	比特位	属性	复位值	描述
SPIBUF	7-0	W/R	0	SPI 数据缓冲寄存器： <ul style="list-style-type: none"> ● 主机模式下，对寄存器 SPIBUF 写操作，则 SPI 将开始向外传输发送缓存数据，从机模式下，则为待发送数据 ● 对寄存器 SPIBUF 读操作，则 SPI 将从接收缓存中读取数据

15.8.6 SPI 主机采样控制寄存器(SPISPC)[0xCE]

Table 260 SPI 主机采样控制寄存器(SPISPC)

参数名	比特位	属性	复位值	描述
SPISPC	2-0	W/R	0	SPI 主机采样 SPI_MISO 上数据的延时个数： 000: 不延时采样

参数名	比特位	属性	复位值	描述
				001: 延时 1 个 spi_clk 采样 010: 延时 2 个 spi_clk 采样 011: 延时 3 个 spi_clk 采样 100: 延时 4 个 spi_clk 采样 101: 延时 5 个 spi_clk 采样 110: 延时 6 个 spi_clk 采样 111: 延时 7 个 spi_clk 采样
Reserved	7-3	R	0	保留

16 模拟/数字转换（ADC）和可编程增益放大器（PGA）

16.1 概述

JMT1801ED 中包括一个采用逐次逼近型的模数转换器（SAR-ADC）和一个可编程增益放大器（PGA）。

SAR-ADC 的分辨率为 12 位，采样率最高为 800 KSPS，采样周期最小为 1.25us，有 16 个模拟输入端供选择，结构框图如 Figure 145 所示。

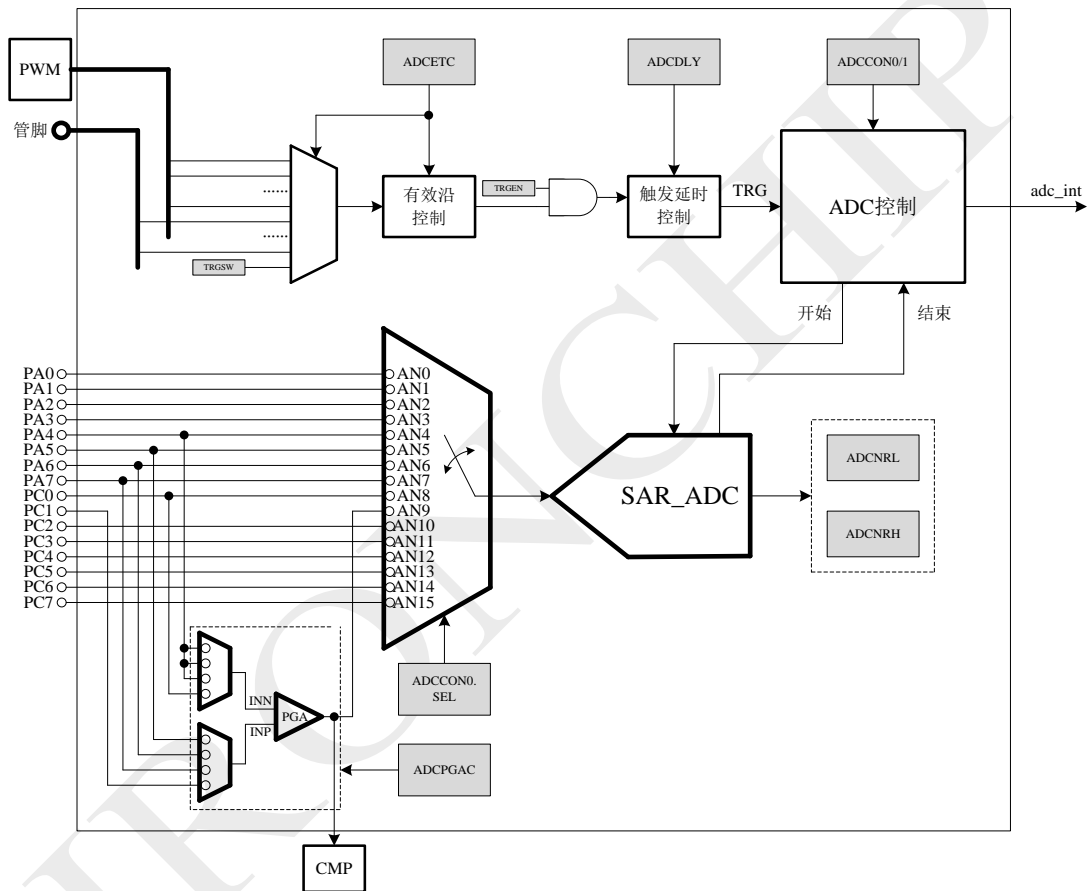


Figure 145 ADC 结构框图

16.2 ADC 转换结果访问方式

ADC 的所有转换结果寄存器有两种访问方式：

- JMT51 访问。JMT51 通过 8 位特殊功能寄存器总线读取结果寄存器；
- JMT018 DSP 指令访问。JMT018 DSP 通过 16 位 X 数据总线读取结果寄存器。

JMT51 和 JMT018 DSP 访问 ADC 转换结果的寄存器地址不同，具体见章节 16.11ADC 寄存器。

16.3 ADC 输入端口配置

ADC 的模拟输入端参见 Table 261:

Table 261 ADC 输入端对应关系

输入端 ANx	对应模拟输入
AN0	PA0
AN1	PA1
AN2	PA2
AN3	PA3
AN4	PA4
AN5	PA5
AN6	PA6
AN7	PA7
AN8	PC0
AN9	PA5, PA6, PA7, PC1 四选一后经 PGA 的输出
AN10	PC2
AN11	PC3
AN12	PC4
AN13	PC5
AN14	PC6
AN15	PC7

其中 PA0~PA7 和 PC0~PC7 为芯片模拟输入管脚。当这些管脚作为模拟管脚使用时，需将相应通道的寄存器 AIOEN 置位，隔断数字对模拟的影响。

16.4 可编程增益放大器（PGA）说明

芯片内置 1 个可编程增益放大器（PGA），PA5，PA6，PA7 和 PC1 经过四选一后连接到 PGA 的正输入端，其中，PA5，PA6，PA7 对应的负输入端为 PA4，PC1 对应的负输入端为 PC0。PGA 具体实现如 Figure 146 所示：

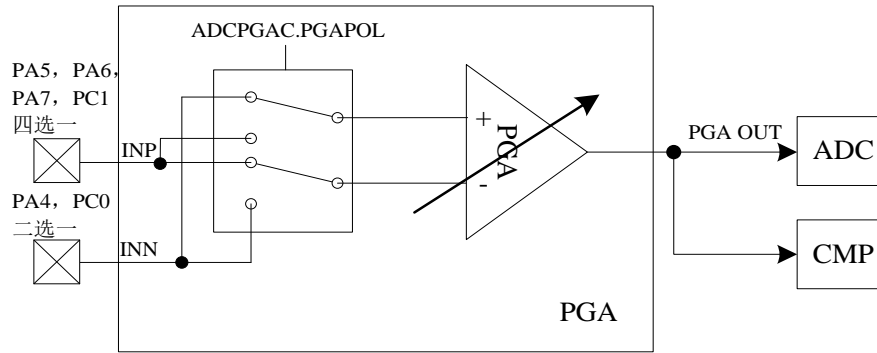


Figure 146 运算放大器内部框图

PGA 支持 1/2、1、2、4、8、16、32、64 倍放大，配置寄存器 ADCPGAC.PGAMUL 可选择运放的放大倍数，具体参见该寄存器说明。

配置寄存器 ADCPGAC.PGAPOL 可切换 PGA 的输入极性，也即调整 PGA 正负端的输入连接关系，如 Figure 146 所示：

- 寄存器 ADCPGAC.PGAPOL 为 0（反相）：PGA 正端接 INN，负端接 INP；
- 寄存器 ADCPGAC.PGAPOL 为 1（同相）：PGA 正端接 INP，负端接 INN。

PGA 输入电压为 0 时，对应的 PGA 输出电压为 ADC 参考电压 VREF 的一半，也即 VREF/2，PGA 输入输出关系为：

- 反相： $V_OUT = VREF/2 - (PGA_MUL_NUM * V_IN)$
- 同相： $V_OUT = VREF/2 + (PGA_MUL_NUM * V_IN)$

其中，V_OUT 为 PGA 的输出电压，VREF 为 ADC 的参考电压，PGA_MUL_NUM 为 PGA 的放大倍数，V_IN 为 PGA 的输入端 INP 和 INN 之间的电压差值（取值范围 -300mV~VREF）。

Figure 147 给出了 PGA 配置为同相时，放大倍数为 1/2、1 和 2 场景下，PGA 输入和 PGA 输出的电压关系图。

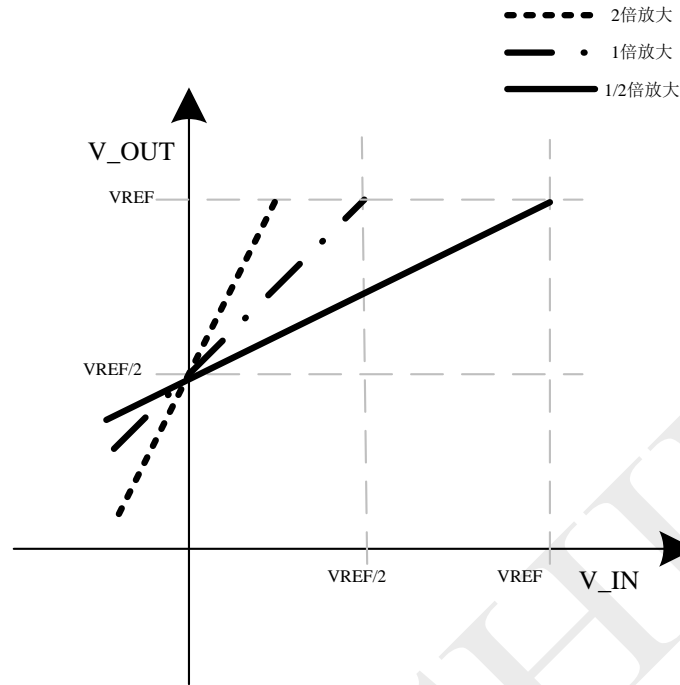


Figure 147 运算放大器同相，输入输出关系

Figure 148 给出了 PGA 配置为反相时，放大倍数为 1/2、1 和 2 场景下，PGA 输入和 PGA 输出的电压关系图。

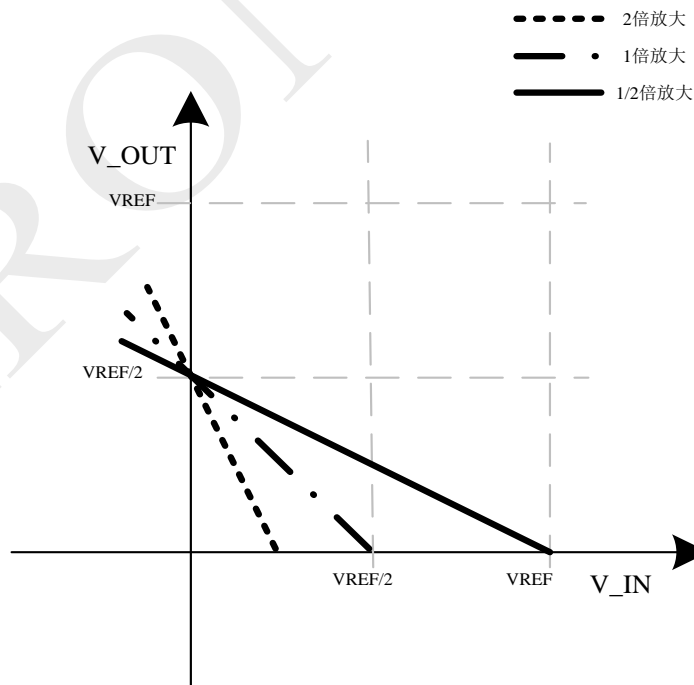


Figure 148 运算放大器反相，输入输出关系

配置寄存器 ADCPGAC.PGAPD，可对 PGA 工作模式进行设置：

- 0：配置 PGA 进入正常工作模式（默认）。
- 1：配置 PGA 进入低功耗模式。

16.5 ADC 参考电压配置

ADC 的参考电压为芯片外部输入参考电压。

16.6 ADC 转换功能

芯片的 12 位 A/D 转换器，转换的最大值可达 0xFFF。由于模拟输入最大值等于 VREF 的电压值，因此每一位可表示 VREF/4096 的模拟输入值。下图显示 A/D 转换器模拟输入值和数字输出值之间理想的转换功能。

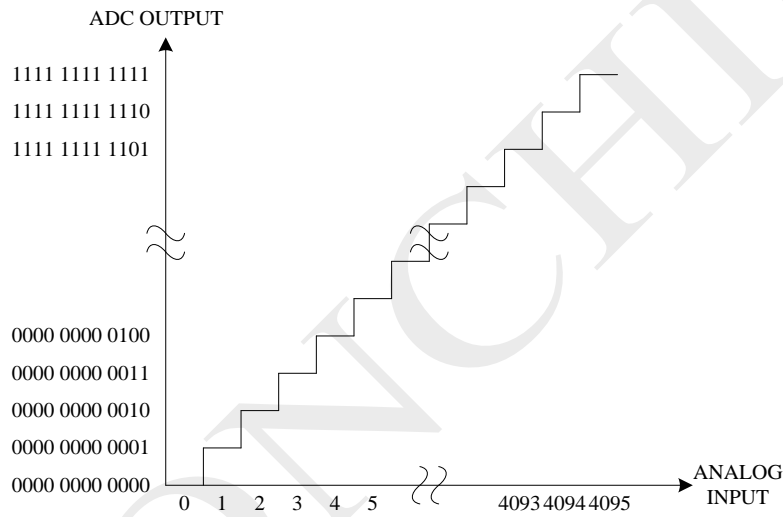


Figure 149 ADC 转换功能

16.7 ADC 转换结果数据格式

ADC 原始转换结果为 12 位数据，该 12 位数据经过数据位扩展至 16 位后放入结果寄存器。扩展后的 16 位数据有 4 种数据格式，可通过配置寄存器 ADCCON1.ADCDF 进行选择，4 种数据格式参见 Figure 150 所示。

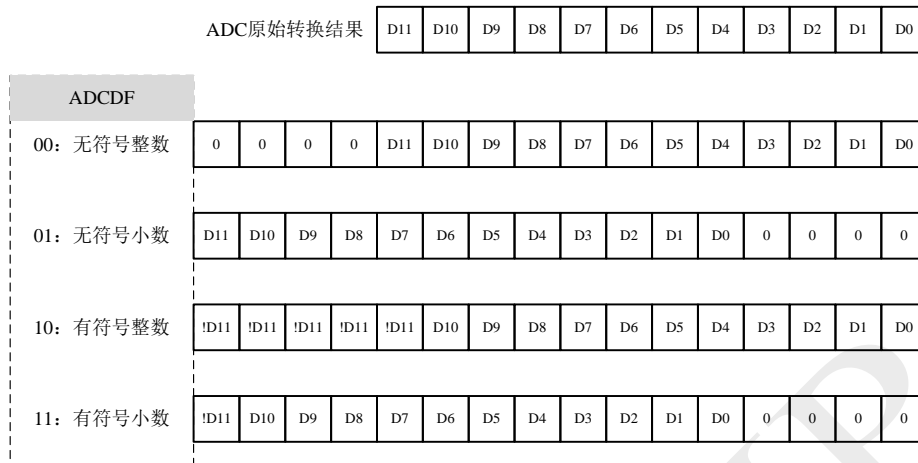


Figure 150 ADC 数据格式

16.8 ADC 转换时序

每次 ADC 转换包括采样时间和转换时间：采样时间的 `adc_sample_clk` 时钟周期数可通过寄存器 `ADCCON1.SAMPCLK` 配置；转换时间固定为 12 个 `adc_sample_clk` 时钟周期。

单次 ADC 时序如 Figure 151 所示。

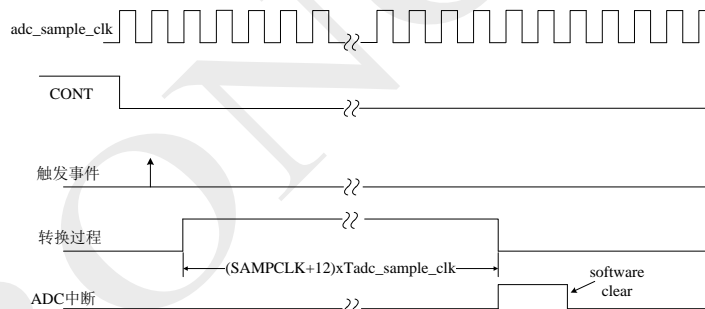


Figure 151 单次 ADC 时序图

连续 ADC 时序如 Figure 152 所示。

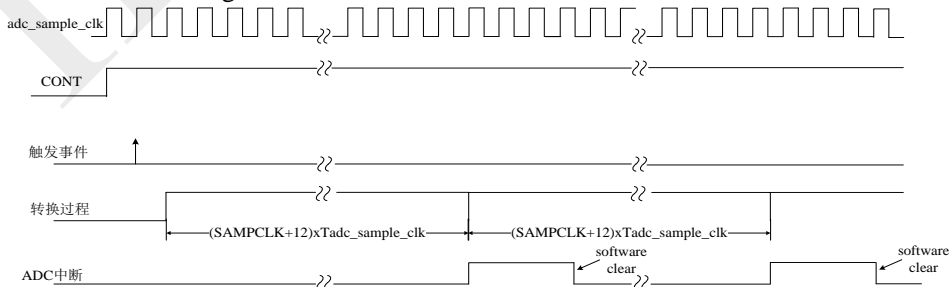


Figure 152 连续 ADC 时序图

16.9 触发事件

触发事件包括软件触发事件和外部触发事件。

16.9.1 软件触发事件

通过配置寄存器 ADCCON0.SWST 为 1，可以立即触发 ADC 转换。

16.9.2 外部触发事件

外部触发事件包括 PWM 触发、管脚边沿触发和 TRGSW 软件触发，配置过程如下：

- 配置寄存器 ADCETC.TRGSEL[3:0]，选择外部触发事件；
- 配置寄存器 ADCETC.TRGPE 和 ADCETC.TRGNE，选择外部触发事件的有效沿；
- 配置寄存器 ADCDLY，选择外部触发事件启动 ADC 采样的延时；
- 配置寄存器 ADCETC.TRGGEN 为 1，使能外部触发事件。

外部触发事件启动 ADC 采样延时功能，如 Figure 153 所示。

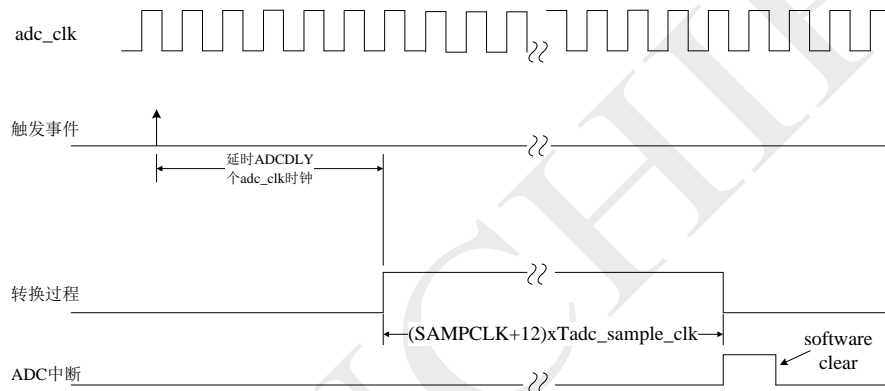


Figure 153 外部触发事件延时功能

Table 262 ADC 外部触发事件

触发源	类型	TRGSEL
EXINT_0	外部引脚输入滤波后信号，见 Figure 164	0000
EXINT_1		0001
EXINT_2		0010
EXINT_3		0011
EXINT_4		0100
EXINT_5		0101
EXINT_6		0110
EXINT_7		0111
TIM_OC0	PWM 内部信号，见章节 11.5.5 主模式控制	1000
TIM_OC1		1001
TIM_OC2		1010
TIM_OC3		1011
TIM_TRGO		1100

触发源	类型	TRGSEL
TRGSW	软件控制	1101

16.10 ADC 操作流程

A/D 转换的操作流程如下：

- 配置 ADCON1 寄存器中的 SAMPCLK，选择采样时间
- 配置 ADCETC 寄存器中的 TRGSEL，选择触发事件
- 配置 ADCETC 寄存器中的 TRGPE 和 TRGNE，选择触发事件有效沿
- 配置 ADCON0 寄存器中的 SEL，选择模拟输入通道
- 配置 ADCON0 寄存器中的 CONT，选择单次或者连续模式
- 配置 ADCON0 寄存器中的 ADCEN，使能 ADC 模块
- 软件触发事件或外部触发事件启动 AD 转换
- 完成 AD 转换后，ADCINT 信号置位，产生 ADC 中断；同时，AD 转换的结果存放在结果寄存器 ADCRL 和 ADCRH 中

16.11 ADC 寄存器

16.11.1 JMT51 访问 ADC 寄存器

ADC 模块有 7 个寄存器，对应的 SFRPAGE 为 0，具体说明如 Table 263 所示。ADC 的结果寄存器是 16 位，而 MCU 的数据总线是 8 位，为了便于寄存器读写，将 16 位寄存器拆分为两个 8 位寄存器，即高字节寄存器 ADCRH 和低字节寄存器 ADCRL。读写时需遵循以下原则：先读写高字节寄存器再读写低字节寄存器。

Table 263 JMT51 访问 ADC 寄存器（SFRPAGE=0x00）

地址	寄存器名	类型	复位值	功能描述
0x91	ADCCON0	W/R	0x00	ADC 控制寄存器 0
0x92	ADCCON1	W/R	0x00	ADC 控制寄存器 1
0x93	ADCPGAC	W/R	0x02	放大器控制寄存器
0x94	ADCETC	W/R	0x60	ADC 外部触发事件控制寄存器
0x95	ADCDLY	W/R	0x00	ADC 外部触发延时控制寄存器
0x96	ADCRL	R	0x00	ADC 结果寄存器低 8 位
0x97	ADCRH	R	0x00	ADC 结果寄存器高 8 位

16.11.1.1 ADC 控制寄存器 0(ADCCON0)[0x91]

Table 264 ADC 控制寄存器 0(ADCCON0)

参数名	比特位	属性	复位值	描述
SEL	3-0	W/R	0	ADC 通道选择：

参数名	比特位	属性	复位值	描述
				0000: PA0 0001: PA1 0010: PA2 0011: PA3 0100: PA4 0101: PA5 0110: PA6 0111: PA7 1000: PC0 1001: PA5, PA6, PA7, PC1 四选一后经 PGA 的输出 1010: PC2 1011: PC3 1100: PC4 1101: PC5 1110: PC6 1111: PC7
ADCEN	4	W/R	0	ADC 使能位: 0: 关闭 ADC 1: 打开 ADC
CONT	5	W/R	0	ADC 模式控制: 0: 单次模式 1: 连续模式
ADCINT	6	W/R	0	ADC 中断标志位: <ul style="list-style-type: none"> • 当有 ADC 中断申请时, 该 bit 置为 1 • 软件配置该 bit 为 1 可产生软中断, 在中断处理程序中需要软件写 0 来清除
SWST	7	W/R	0	ADC 软件触发事件启动: 0: 不启动 1: 启动 ADC 转换

16.11.1.2 ADC 控制寄存器 1(ADCCON1)[0x92]

Table 265 ADC 控制寄存器 1(ADCCON1)

参数名	比特位	属性	复位值	描述
ADCDF	1-0	W/R	0	ADC 结果数据格式: 00: 无符号整数 01: 无符号小数 10: 有符号整数 11: 有符号小数
ADCPD	2	W/R	0	软件控制 ADC 低功耗使能: 0: 不使能 1: 使能
SAMPSEL	3	W/R	1	采样模式控制: 0: 测试模式 1: 正常模式
SAMPCLK	7-4	W/R	0	采样时间的 adc_sample_clk 时钟数选择: 0000: 4 0001: 5 0010: 6 0011: 7 0100: 8 0101: 9 0110: 10 0111: 11 1000: 16 1001: 32 1010: 48 1011: 64 1100: 96 1101: 128 1110: 192 1111: 256

16.11.1.3 放大器控制寄存器(ADCPGAC)[0x93]

Table 266 放大器控制寄存器(ADCPGAC)

参数名	比特位	属性	复位值	描述
PGAMUL	2-0	W/R	010	PGA 运放倍数选择: 000: 1/2 倍 001: 1 倍 010: 2 倍 011: 4 倍 100: 8 倍 101: 16 倍 其他: 保留
PGAPOL	3	W/R	0	PGA 极性寄存器: 0: 反相 1: 同相
PGAMUX	5-4	W/R	0	PGA 输入引脚选择: 00: INP—PA5; INN—PA4 01: INP—PA6; INN—PA4 10: INP—PA7; INN—PA4 11: INP—PC1; INN—PC0
PGATRIM	6	W/R	0	PGA 校准使能: 0: 不使能 1: 使能 注: 仅用于出厂前校准
PGAPD	7	W/R	0	PGA 的低功耗控制: 0: 正常工作模式 1: 低功耗模式

16.11.1.4 ADC 外部触发事件控制寄存器(ADCETC)[0x94]

Table 267 ADC 外部触发事件控制寄存器(ADCETC)

参数名	比特位	属性	复位值	描述
TRGSEL	3-0	W/R	0	外部触发事件选择: 0000: EXINT_0 0001: EXINT_1 0010: EXINT_2 0011: EXINT_3

参数名	比特位	属性	复位值	描述
				0100: EXINT_4 0101: EXINT_5 0110: EXINT_6 0111: EXINT_7 1000: TIM_OC0 1001: TIM_OC1 1010: TIM_OC2 1011: TIM_OC3 1100: TIM_TRGO 1101: TRGSW 其他: 保留
TRGEN	4	W/R	0	外部触发事件使能: 0: 不使能 1: 使能 注: 切换外部触发事件时, 需要先关闭触发使能
TRGPE	5	W/R	1	外部触发事件上升沿使能: 0: 不使能 1: 使能
TRGNE	6	W/R	1	外部触发事件下降沿使能: 0: 不使能 1: 使能
TRGSW	7	W/R	0	配置 TRGSW 为 1 作为外部触发事件: 0: 不启动 ADC 转换 1: 启动 ADC 转换 注: 该启动仅在 TRGSEL 选中 TRGSW 为外部触发事件, 且 TRGEN 有效时生效

16.11.1.5 ADC 外部触发延时控制寄存器(ADC DLY)[0x95]

Table 268 ADC 外部触发延时控制寄存器(ADC DLY)

参数名	比特位	属性	复位值	描述
ADC DLY	7-0	W/R	0	外部触发事件启动 ADC 采样延时 adc_clk 周期个数, 当延时时间大于外部触发事件

参数名	比特位	属性	复位值	描述
				周期时间时，外部触发事件将不起作用。

16.11.1.6 ADC 结果寄存器低 8 位(ADCRL)[0x96]

Table 269 ADC 结果寄存器低 8 位(ADCRL)

参数名	比特位	属性	复位值	描述
ADCRL	7-0	R	0	ADC 结果低 8 位

16.11.1.7 ADC 结果寄存器高 8 位(ADCRH)[0x97]

Table 270 ADC 结果寄存器高 8 位(ADCRH)

参数名	比特位	属性	复位值	描述
ADCRH	7-0	R	0	ADC 结果高 8 位

16.11.2 JMT018 访问 ADC 寄存器

JMT018 DSP 通过 16 位 X 数据空间总线访问 ADC 结果寄存器，寄存器列表如 Table 271 所示。

Table 271 JMT018 访问 ADC 寄存器

地址	寄存器名	属性	复位值	功能描述
0x40	ADCR	R	0x0000	ADC 结果寄存器

16.11.2.1 ADC 结果寄存器[0x40]

Table 272 ADC 结果寄存器(ADCR)

参数名	比特位	属性	复位值	描述
ADCR	15-0	R	0	ADC 结果寄存器

17 模拟比较器 (CMP)

17.1 模拟比较器概述

JMT1801ED 内置 1 个模拟比较器 (CMP)，该比较器工作原理为输入电压和参考电压进行比较，若选定比较器输入源的电压值大于选定参考电压的电压值，则比较器输出 1，反之则输出 0。

模拟比较器输出结果可以用于 PWM 刹车、OC/OD 控制、低压检测等功能。

模拟比较器参考电压来自于参考电压 VREF 管脚。

模拟比较器的输入可在芯片模拟供电 VDDA、芯片管脚 PA0 和 PGA 输出之间进行选择。

模拟比较器的比较输出，经过滤波防毛刺处理之后，送至比较结果寄存器，软件可以通过回读结果寄存器获取比较的结果。

模拟比较器的框图如 Figure 154 所示，下面具体阐述模拟比较器参考电压选择、比较器输入源选择、防毛刺滤波：

- 比较器输入源选择：配置寄存器 CMPSEL，选择比较器输入源：
 - 00：选择芯片模拟供电 VDDA 作为该比较器输入（默认）。
 - 01：选择芯片管脚 PA0 作为比较器输入。
 - 1x：选择 PGA 输出作为比较器输入。
 - 选择芯片模拟供电 VDDA 作为比较器输入时，比较器输入端电压可通过内部 DAC 对 VDDA 分压得到。通过配置 VDDDAC 寄存器，可修改比较器输入端电压大小，具体大小为： $(V_{VDDA}/64) \times (VDDDAC+1)$ ，其中 V_{VDDA} 为芯片模拟供电的电压值，VDDDAC 为软件配置该寄存器的数值。
 - 若选择 PA0 作为比较器输入，需要配置管脚 PA0 为模拟输入模式，具体配置参见 GPIO 章节 0 说明。
- 配置参考电压大小：通过配置寄存器 CMPREFDAC，可以修改比较器的参考电压大小，具体大小为：

$$(V_{REF}/64) \times (CMPREFDAC+1)$$
 其中 V_{REF} 为参考电压源的电压值，CMPREFDAC 为软件配置寄存器的数值。
- 比较器输出滤波防毛刺：
 - 硬件对模拟比较器的比较输出进行了滤波防毛刺处理；
 - 配置比较器滤波时钟控制寄存器 CMP_CLK_CTRL，选择滤波时钟频率；
 - 配置比较器滤波选择寄存器 CMPFT，选择滤波的周期数。

模拟比较器比较结果，滤波之后，做如下处理：

- 配置极性选择寄存器 BKSEL，选择合适的信号作为 PWM 的比较器刹车信号，

控制 PWM 的输出。

- 配置极性选择寄存器 OCSEL，再配置 MOS 管选型寄存器 OCMOSEL，选择合适的信号作为 OC 控制信号，可用于板级硬件保护。
- 配置极性选择寄存器 ODSEL，再配置 MOS 管选型寄存器 ODMOSEL，选择合适的信号作为 OD 控制信号，可用于板级硬件保护。
- 输出结果直接用于 LVR 低压检测，若 LVR 低压复位使能，则产生低压复位信号复位全芯片。

另外，比较器的比较结果送至结果寄存器 CMPFLAG，软件可通过回读该结果寄存器，获取比较的结果。

如果中断被使能了（寄存器 IEN1.CMPINT=1），则硬件按照以下规则产生中断，配置寄存器 CMPINT 为 0 可清除比较器中断：

- 比较器比较结果经滤波之后，和上次的结果不一致，即产生中断；
- 软件可通过查询中断类型寄存器 CMPINTTYPE，获取该中断为比较器结果从 1 变为 0 的中断，还是从 0 变为 1 的中断。

模拟比较器有两种工作模式，正常工作模式（Normal）和低功耗模式（power down），通过配置寄存器 CMPEN.CMPPD，可以选择模拟比较器的工作模式。

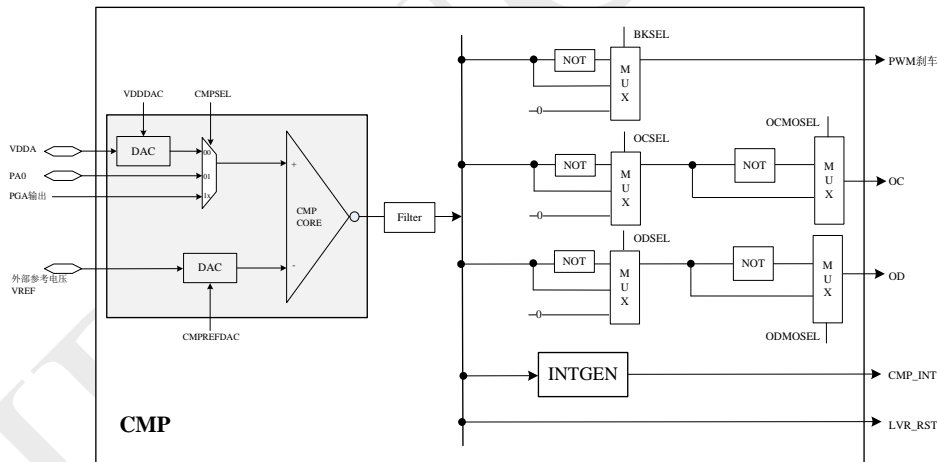


Figure 154 比较器 CMP 内部框图

17.2 低压检测

通过模拟比较器，可实现低压检测功能（LVR），该功能用于监视芯片供电，若芯片供电电压低于设定的值可上报一个中断或产生全芯片复位信号。模块实现如 Figure 155 所示。

如果芯片供电满足如下关系，也即芯片供电电压低于设定的门限值，则会产生 LVR 低压检测中断或者低压检测复位：

$$V_{VDDA} * (VDDDAC+1)/64 < V_{REF} * (CMPREFDAC+1)/64$$

其中：V_{VDDA} 为芯片供电电压，VDDDAC 为 V_{VDDA} 的分压 DAC 的寄存器值，V_{REF} 为

模拟比较器参考电压的电压值，CMPREFDAC 为比较器参考电压分压 DAC 的寄存器值。

假设比较器的参考电压为 2.5V, LVR 低压检测电平为 2.75V。则配置的寄存器 VDDDAC 和 CMPREFDAC 需要满足如下关系，才能实现芯片供电小于 2.75V 时产生低压检测中断或复位：

$$2.75 * (VDDDAC+1)/64 \geq 2.5 * (CMPREFDAC +1)/64$$

可配置 VDDDAC 为 9, CMPREFDAC 配置为 43, 也即该两个寄存器的默认值。

比较器作为 LVR 功能的配置包括：

- 根据比较器小节 17.1 描述，选择比较器输入源为芯片模拟供电 VDDA、选择参考电压、配置两个 DAC 的值、配置滤波周期数。
- 若仅需要监控全芯片供电电压是否低于门限电压，可配置比较器中断使能，中断使能之后，若电压低于监控的门限电压，则产生中断。
- 当全芯片电压低于门限电压时，若需要产生全芯片复位信号，配置寄存器 LVRRSTEN，启动 LVR 复位功能。

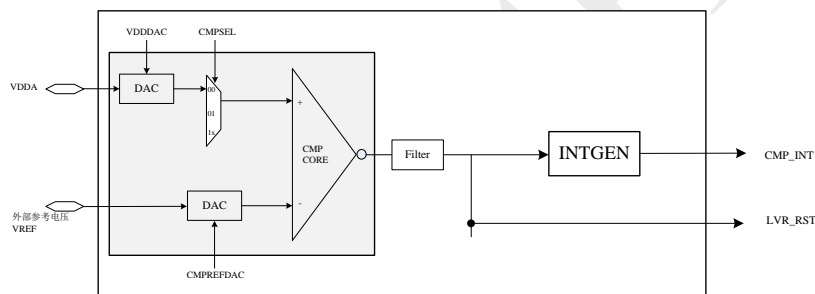


Figure 155 LVR 实现框图

17.3 模拟比较器寄存器

模拟比较器（CMP）共有 9 个寄存器。对应的 SFR PAGE 为 0，具体列表如 Table 273 所示。

Table 273 模拟比较器寄存器（SFRPAGE=0x00）

地址	寄存器名	类型	复位值	功能描述
0xB9	CMPEN	W/R	0x01	模拟比较器使能寄存器
0xBA	CMPFT	W/R	0x03	模拟比较器滤波选择寄存器
0xBB	CMPVDD	W/R	0x20	模拟比较器 VDD 分压寄存器
0xBC	CMPREF	W/R	0x12	模拟比较器参考电压选择寄存器
0xBD	CMPINT	W/R	0x00	模拟比较器中断寄存器
0xBE	CMPFLAG	W/R	0x00	模拟比较器比较结果寄存器
0xC1	OCPRSEL	W/R	0x03	OC 控制寄存器

地址	寄存器名	类型	复位值	功能描述
0xC2	ODPRSEL	W/R	0x03	OD 控制寄存器
0xC3	BKPRSEL	W/R	0x03	PWM 刹车控制寄存器

17.3.1 模拟比较器使能寄存器(CMPEN) [0xB9]

Table 274 模拟比较器使能寄存器(CMPEN)

参数名	比特位	属性	复位值	描述
CMPPD	0	W/R	1	模拟比较器工作模式： 0: 正常工作模式 1: 低功耗模式
CMPSEL	2-1	W/R	0	模拟比较器输入源选择： 00: 芯片供电 VDDA 01: 管脚 PA0 10/11: PGA 输出
Reserved	7-3	R	0	保留

17.3.2 模拟比较器滤波选择寄存器(CMPFT) [0xBA]

Table 275 模拟比较器滤波选择寄存器(CMPFT)

参数名	比特位	属性	复位值	描述
CMPFT	1-0	W/R	11	模拟比较器输出滤波周期： 00: 不滤波 01: 1 个滤波时钟周期 10: 2 个滤波时钟周期 11: 3 个滤波时钟周期
Reserved	7-2	R	0	保留

17.3.3 模拟比较器 VDD 分压寄存器(CMPVDD) [0xBB]

Table 276 模拟比较器 VDD 分压寄存器(CMPVDD)

参数名	比特位	属性	复位值	描述
VDDDAC	5-0	W/R	0x20	VDD 的 DAC 数字输入码。VDD 经过该 DAC 分压之后输出至比较器输入端，比较器输入端电压为： $(V_{VDD}/64) \times (VDDDAC+1)$ 其中 V_{VDD} 为芯片供电电压值

参数名	比特位	属性	复位值	描述
Reserved	7-6	R	0	保留

17.3.4 模拟比较器参考电压选择寄存器(CMPREF) [0xBC]

Table 277 模拟比较器参考电压选择寄存器(CMPREF)

参数名	比特位	属性	复位值	描述
CMPREFDAC	5-0	W/R	0x12	参考电压的 DAC 数字输入码。参考电压经过该 DAC 分压之后输出至比较器参考电压输入端，比较器参考电压输入端电压为： $(V_{REF}/64) \times (CMPREFDAC+1)$ 其中 V_{REF} 为参考电压源的电压值
Reserved	7-6	R	0	保留

17.3.5 模拟比较器中断寄存器(CMPINT) [0xBD]

Table 278 模拟比较器中断寄存器(CMPINT)

参数名	比特位	属性	复位值	描述
CMPINT	0	W/R	0	模拟比较器中断标志位： <ul style="list-style-type: none"> 若比较器输出结果和之前结果不一致则产生中断 软件配置该位为 1 可产生软中断，在中断处理程序中需要软件写 0 来清除
Reserved	7-1	R	0	保留

17.3.6 模拟比较器比较结果寄存器(CMPFLAG) [0xBE]

Table 279 模拟比较器比较结果寄存器(CMPFLAG)

参数名	比特位	属性	复位值	描述
CMPINTTYPE	0	W/R	0	比较器中断类型寄存器： 0: 比较器结果从 1 变为 0 中断 1: 比较器结果从 0 变为 1 中断
CMPOUT	1	R	0	模拟比较器输出结果： 0: 比较器输入电压小于参考电压 1: 比较器输入电压大于参考电压

参数名	比特位	属性	复位值	描述
Reserved	7-3	R	0	保留

17.3.7 OC 控制寄存器(OCPRSEL) [0xC1]

Table 280 OC 控制寄存器(OCPRSEL)

参数名	比特位	属性	复位值	描述
OCSEL	1-0	W/R	11	OC 基准信号极性选择： 00: 滤波之后结果取反 01: 滤波之后直通 其他: 选取 0 作为基准信号
OCMOSEL	2	W/R	0	OC 控制信号 MOS 类型选择： 0: 基准信号取反产生 OC 信号 1: 基准信号作为 OC 信号
Reserved	7-3	R	0	保留

17.3.8 OD 控制寄存器(ODPRSEL) [0xC2]

Table 281 OD 控制寄存器(ODPRSEL)

参数名	比特位	属性	复位值	描述
ODSEL	1-0	W/R	11	OD 基准信号极性选择： 00: 滤波之后结果取反 01: 滤波之后直通 其他: 选取 0 作为基准信号
ODMOSEL	2	W/R	0	OD 控制信号 MOS 类型选择： 0: 基准信号取反产生 OD 信号 1: 基准信号作为 OD 信号
Reserved	7-3	R	0	保留

17.3.9 PWM 刹车控制寄存器(BKPRSEL) [0xC3]

Table 282 PWM 刹车控制寄存器(BKPRSEL)

参数名	比特位	属性	复位值	描述
BKSEL	1-0	W/R	11	PWM 刹车基准信号极性选择： 00: 滤波之后结果取反 01: 滤波之后直通 其他: 选取 0 作为基准信号

参数名	比特位	属性	复位值	描述
Reserved	7-2	R	0	保留

IRONCHIP

18 通用输入/输出 (GPIO)

GPIO 实现的主要功能有：

- 管脚的模拟/数字选择、输入输出选择、开漏、驱动强度、上下拉等设置
- 管脚功能复用
- 管脚滤波
- 外部中断合并

正常情况下，GPIO 模块的最高输入时钟为 73.728MHz。

18.1 管脚设置

数字 I/O 实现框图如 Figure 156 所示，数字、模拟复用 I/O 实现框图如 Figure 157 所示。

为防止数字 I/O 信号对模拟 I/O 信号的影响，当数字、模拟复用 I/O 当做模拟 I/O 使用时，需将相应管脚的寄存器 AIOEN 置位。

输入使能 (IE) 由硬件自动控制：

- 数字 I/O 的 IE 固定接 1，使能输入
- 数字、模拟复用 I/O 用作模拟 I/O 时，IE 接 0，关断输入
- 数字、模拟复用 I/O 用作数字 I/O 时，IE 接 1，使能输入

管脚信号说明：

- PU：弱上拉使能
- PD：弱下拉使能
- OD：开漏输出模式使能
- IE：输入使能
- OE：输出使能
- A：数字输出 Buffer 输入
- Y：数字输入 Buffer 输出
- YA：模拟信号管脚
- SR：转换速率快慢选择
- DR：驱动能力强弱选择

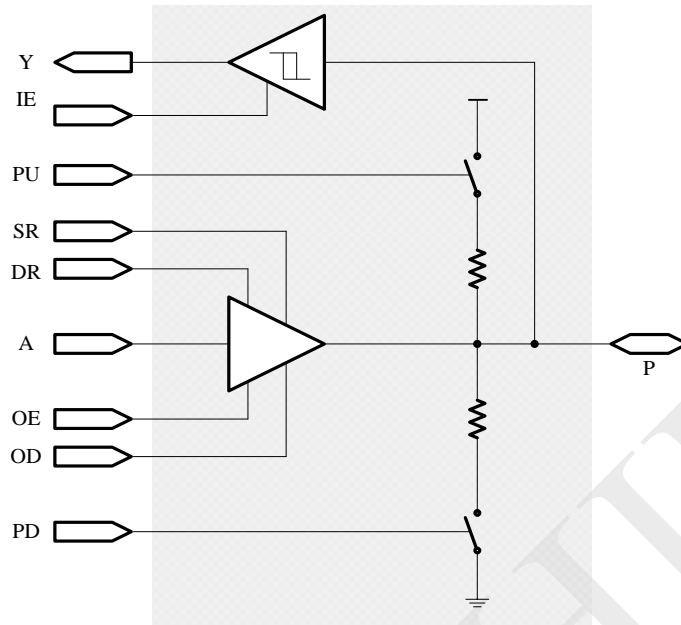


Figure 156 数字 I/O 实现框图

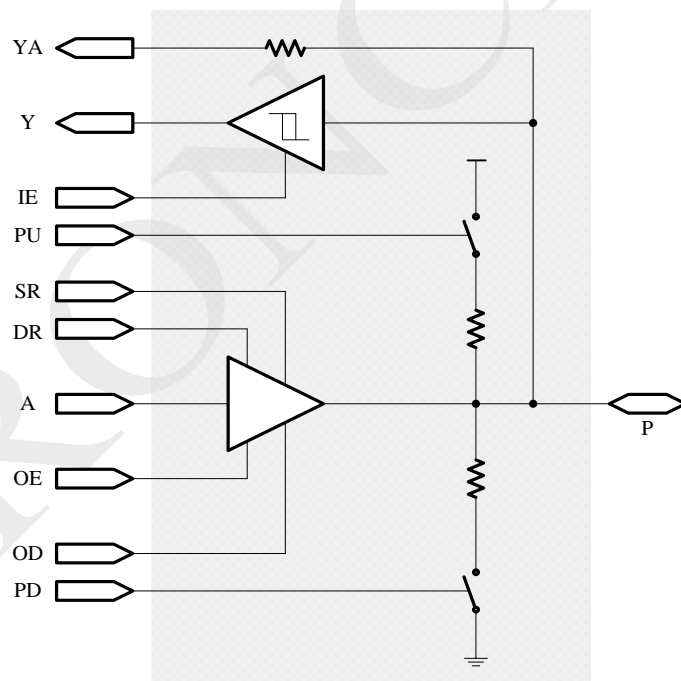


Figure 157 数字/模拟混合 I/O 实现框图

管脚真值表如 Table 283 所示，其中“W”为任意值，“Z”为高阻态，“X”为不确定态：

Table 283 管脚真值表

I/O	输入引脚								输出引脚	
	OE	IE	PU/PD	A	OD	SR	DR	P/YA	P/YA	Y
输入模式	0	1	W	W	W	W	W	1	1	1
	0	1	W	W	W	W	W	0	0	0
	0	1	1	W	W	W	W	-	H(PU) L(PD)	1(PU) 0(PD)
	0	1	0	W	W	W	W	-	Z	X
	0	0	W	W	W	W	W	1	1	0
	0	0	W	W	W	W	W	0	0	0
	0	0	1	W	W	W	W	-	H(PU) L(PD)	0
	0	0	0	W	W	W	W	-	Z	0
输出模式	1	0	W	1	0	W	W	-	1	0
	1	0	W	0	0	W	W	-	0	0
	1	0	W	0	1	W	W	-	0	0
	1	0	1	1	1	W	W	-	H(PU) L(PD)	0
	1	0	0	1	1	W	W	-	Z	0
	1	1	W	1	0	W	W	-	1	1
	1	1	W	0	0	W	W	-	0	0
	1	1	W	0	1	W	W	-	0	0
	1	1	1	1	1	W	W	-	H(PU) L(PD)	1(PU) 0(PD)
	1	1	0	1	1	W	W	-	Z	X

上升/下降转变时间：

- 上升转变时间 T_r ：信号从逻辑 0 变为逻辑 1 所需时间，逻辑 0 标准为小于供电电压的 10%，逻辑 1 的标准为大于供电电压的 90%。
- 下降转变时间 T_f ：信号从逻辑 1 变为逻辑 0 所需时间，逻辑 0 标准为小于供电电压的 10%，逻辑 1 的标准为大于供电电压的 90%。

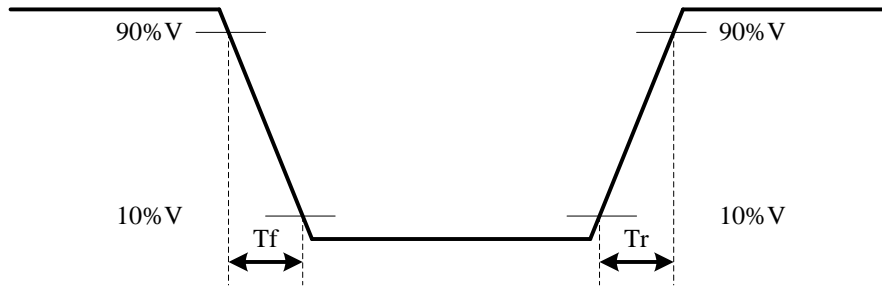


Figure 158 上升/下降转变时间示意图

传播延迟:

- 上升传播延迟 Tdr: 上升信号从 PAD 的 P 端输入到 Y 端传输所需时间
- 下降传播延迟 Tdf: 下降信号从 PAD 的 P 端输入到 Y 端传输所需时间

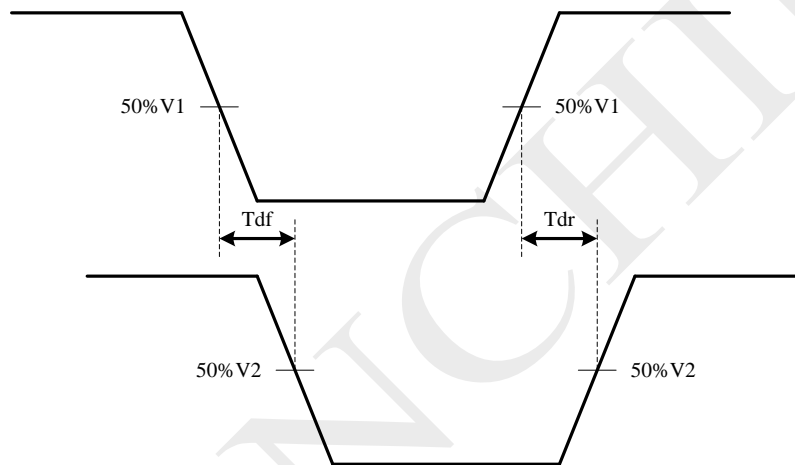


Figure 159 传播延迟示意图

转换速率 SR 对上升/下降转变时间和传播延迟影响:

- 转换速率 SR 越快, 上升/下降转变时间越小
- 转换速率 SR 越快, 传播延迟越小

驱动能力 DR 对上升/下降转变时间和传播延迟影响:

- 驱动能力 DR 越高, 上升/下降转变时间越小
- 驱动能力 DR 越高, 传播延迟越小

18.2 管脚复用

PA0~PA7, PC0~PC7 作为数字模拟复用管脚, 控制参见 Figure 160 和 Figure 161。虚线框中内容为 GPIO 模块寄存器中的控制位, 具体参见 18.9 GPIO 寄存器说明。

以上管脚作为模拟 IO 使用时, 该管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零; 作为数字 IO 使用时, 管脚的 IE, OE, PD, PU, OD 由软件控制, 参见 Figure 160 和 Figure 161 中 AIOEN 所示控制。

Figure 160 为 PA0~PA7 管脚复用框图, 产生 EXINT 中断, 且会产生低功耗唤醒(WKUP)。

Figure 161 为 PC0~PC7 管脚复用框图, 产生 EXINT 中断。

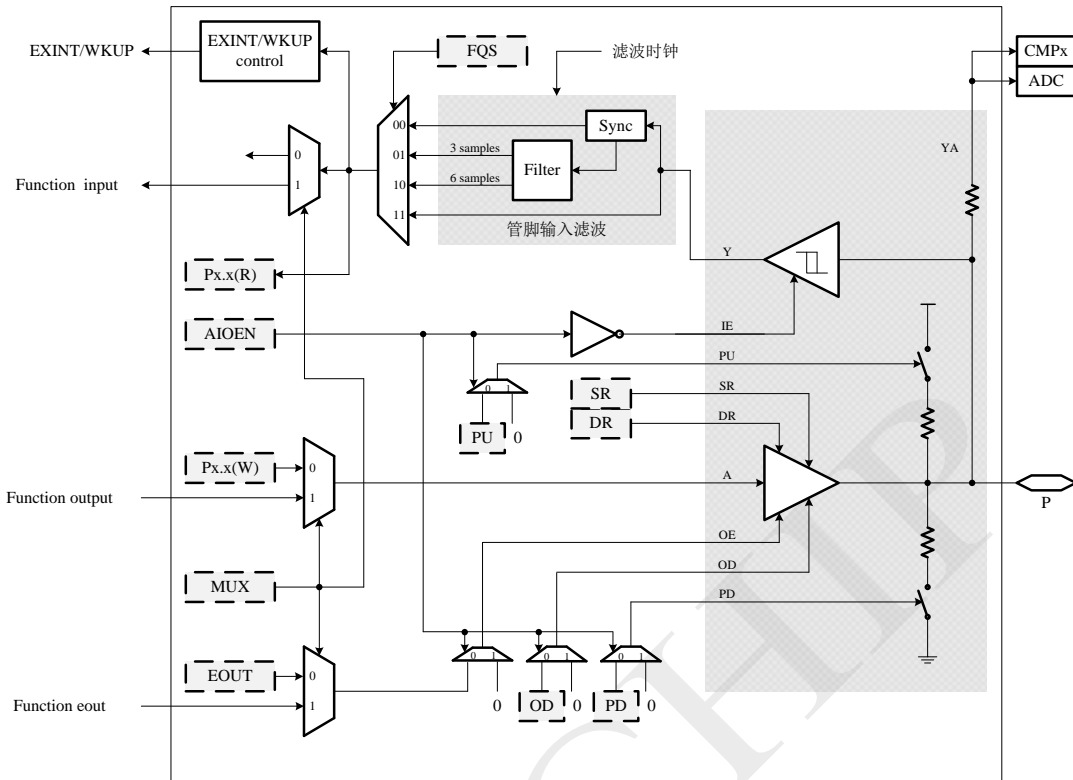


Figure 160 PA0 ~ PA7 管脚复用框图

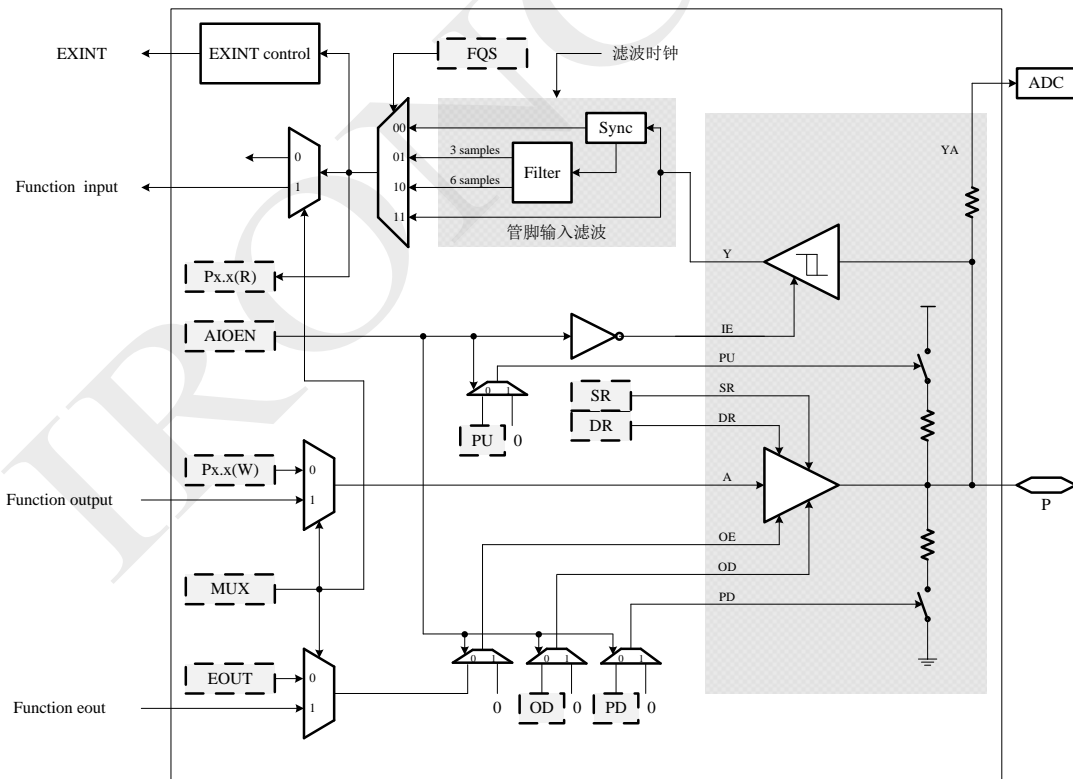


Figure 161 PC0 ~ PC7 管脚复用框图

PB0~PB7, PD0~PD7, PE0~PE7 作为纯数字管脚，控制参见 Figure 162，产生 EXINT 中断。

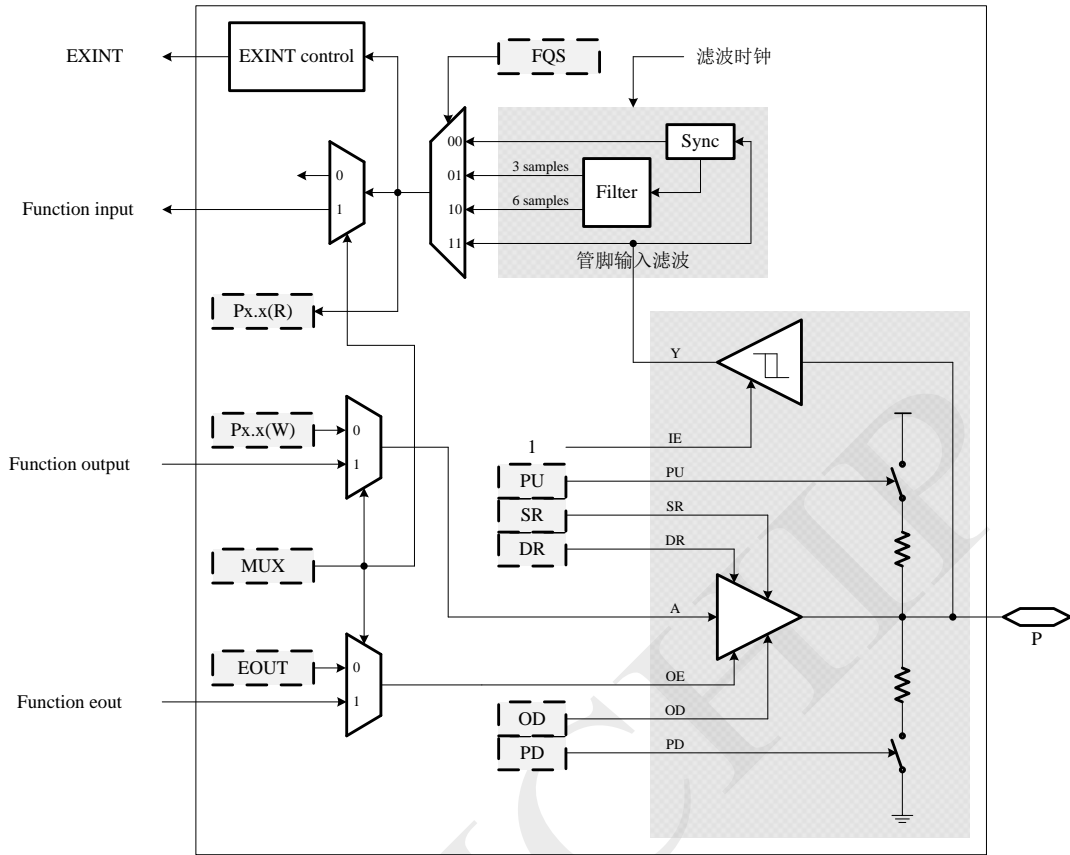


Figure 162 PB0~PB7, PD0~PD7, PE0~PE7 管脚复用框图

18.3 复用功能重映射

通过配置 MUX 寄存器，每个管脚都可以选择成复用功能；通过配置寄存器 AFREMAPO 和 AFREMAP1 可以对复用功能进行重映射。

Table 284 管脚功能复用

管脚	GPIO (Default)	复用功能	复用功能重映射
PA0	GPIO0	SPI_MISO	
PA1	GPIO1	SPI_MOSI	BZOUT
PA2	GPIO2	SPI_NSS	I2C_SCL
PA3	GPIO3	SPI_SCK	I2C_SDA
PA4	GPIO4	MOS_OC	
PA5	GPIO5	MOS_OD	
PA6	GPIO6	PWM_BKIN	
PA7	GPIO7	PWM_ETR	
PB0	GPIO8	UART_TX	BZOUT
PB1	GPIO9	UART_RX	PWM_BKIN

PB2	GPIO10	PWM_CH0	T0
PB3	GPIO11	PWM_CH0N	T0_GATE
PB4	GPIO12	PWM_CH1	T1
PB5	GPIO13	PWM_CH1N	T1_GATE
PB6	GPIO14	PWM_CH2	T2
PB7	GPIO15	PWM_CH2N	T2_GATE
PC0	GPIO16	PWM_CH0	I2C_SCL
PC1	GPIO17	PWM_CH0N	I2C_SDA
PC2	GPIO18	PWM_CH1	SPI_MISO
PC3	GPIO19	PWM_CH1N	SPI_MOSI
PC4	GPIO20	PWM_CH2	SPI_NSS
PC5	GPIO21	PWM_CH2N	SPI_SCK
PC6	GPIO22	MOS_OC	BZOUT
PC7	GPIO23	MOS_OD	PWM_BKIN
PD0	GPIO24	I2C_SCL	
PD1	GPIO25	I2C_SDA	
PD2	GPIO26	MOS_OC	
PD3	GPIO27	MOS_OD	
PD4	GPIO28	SPI_MISO	
PD5	GPIO29	SPI_MOSI	
PD6	GPIO30	SPI_NSS	
PD7	GPIO31	SPI_SCK	
PE0	GPIO32	PWM_CH0	
PE1	GPIO33	PWM_CH0N	
PE2	GPIO34	PWM_CH1	
PE3	GPIO35	PWM_CH1N	
PE4	GPIO36	PWM_CH2	
PE5	GPIO37	PWM_CH2N	
PE6	GPIO38	PWM_CH3	
PE7	GPIO39	PWM_ETR	

18.4 管脚数据寄存器读写

管脚数据寄存器（P0、P1、P2、P3 和 P4）读写遵循以下规则：

管脚数据寄存器写入：

- GPIO 输出模式下：写入将要输出到管脚的电平；
- 其它模式下：只将数据锁存到数据寄存器，不影响管脚电平。

管脚数据寄存器读取：

- 对管脚数据寄存器执行“读取-修改-写入”指令时，读取管脚数据寄存器值；
- 执行其它指令时，读取管脚输入电平值。

18.5 外设管脚复用配置

Table 285 PWM 管脚配置

管脚复用功能	说明	配置需求
PWM_BKIN	PWM 刹车输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
PWM_ETR	PWM 外部时钟输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
PWM_CHx	PWM 通道 x 输入捕获	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
	PWM 通道 x 输出比较	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
PWM0_CHxN	PWM 通道 x 互补输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。

Table 286 UART 管脚配置

管脚复用功能	说明	配置
UART_TX	UART 模式数据输出	相应管脚配置：

管脚复用功能	说明	配置
		复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
	同步移位寄存器模式时钟输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
UART_RX	UART 模式数据输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； PD 和 OD 配置为 0； PU 可根据需要配置。
	同步移位寄存器模式数据输入/输出	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。

Table 287 TIMER 管脚配置

管脚复用功能	说明	配置
T0	Timer0 外部时钟输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
T0_GATE	Timer0 外部门控输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； PD 和 OD 配置为 0； PU 可根据需要配置。
T1	Timer1 外部时钟输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。

管脚复用功能	说明	配置
T1_GATE	Timer1 外部门控输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； PD 和 OD 配置为 0； PU 可根据需要配置。
T2	Timer2 外部时钟输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
T2_GATE	Timer2 外部门控输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； PD 和 OD 配置为 0； PU 可根据需要配置。

Table 288 SPI 管脚配置

管脚复用功能	说明	配置
SPI_SCK	主机模式 SPI 时钟输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
	从机模式 SPI 时钟输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
SPI_NSS	主机模式 SPI 使能输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
	从机模式 SPI 使能输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0；

管脚复用功能	说明	配置
		PU 和 PD 可根据需要配置。
SPI_MOSI	主机模式 SPI 数据输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
	从机模式 SPI 数据输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
SPI_MISO	主机模式 SPI 数据输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
	从机模式 SPI 数据输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。

Table 289 I2C 管脚配置

管脚复用功能	说明	配置
I2C_SCL	I2C 时钟开漏输出	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 1； PD 配置为 0； PU 可根据需要配置。
I2C_SDA	I2C 数据开漏输出	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 1； PD 配置为 0； PU 可根据需要配置。

Table 290 MOS 管脚配置

管脚复用功能	说明	配置
MOS_OC	MOS 的 OC 控制输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
MOS_OD	MOS 的 OD 控制输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。

Table 291 JTAG 管脚配置

管脚复用功能	说明	配置
TCK	JTAG 时钟输入	相应管脚配置： 滤波 FQS 配置为 11，选择直通模式； 配置寄存器 JTAGEN 为 1，使管脚为 JTAG 功能，硬件自动将 PD 置 1，PU 和 OD 清 0。
TMS	JTAG 控制输入	相应管脚配置： 滤波 FQS 配置为 11，选择直通模式； 配置寄存器 JTAGEN 为 1，使管脚为 JTAG 功能，硬件自动将 PU 置 1，PD 和 OD 清 0。
TDI	JTAG 数据输入	相应管脚配置： 滤波 FQS 配置为 11，选择直通模式； 配置寄存器 JTAGEN 为 1，使管脚为 JTAG 功能，硬件自动将 PU 置 1，PD 和 OD 清 0。
TDO	JTAG 数据输出	相应管脚配置： 配置寄存器 JTAGEN 为 1，使管脚为 JTAG 功能，硬件自动将 PU，PD 和 OD 清 0。

若 JTAGEN 使能，PB4，PB5，PB6，PB7 作为 JTAG 管脚使用，PB4 弱下拉，PB5 弱上拉，PB6 弱上拉；此时 PB4CTRL，PB5CTRL，PB6CTRL，PB7CTRL 寄存器不起作用。

18.6 管脚滤波

为了消除外部环境对 I/O 的干扰而产生对全芯片工作状态的影响，芯片所有数字输入管脚均可选择滤波处理（消除干扰）。滤波实现架构框图如 Figure 163 所示。

输入管脚可经过四种不同处理方式，默认选择同步（sync）模式：

- 直通模式（bypass）模式
- 同步（Sync）模式，利用滤波时钟进行同步
- 3 个滤波时钟周期滤波（可滤除小于 2 个滤波时钟周期的脉冲）
- 6 个滤波时钟周期滤波（可滤除小于 5 个滤波时钟周期的脉冲）

其中的滤波时钟为 gpio_clk 经过预分频后的时钟。

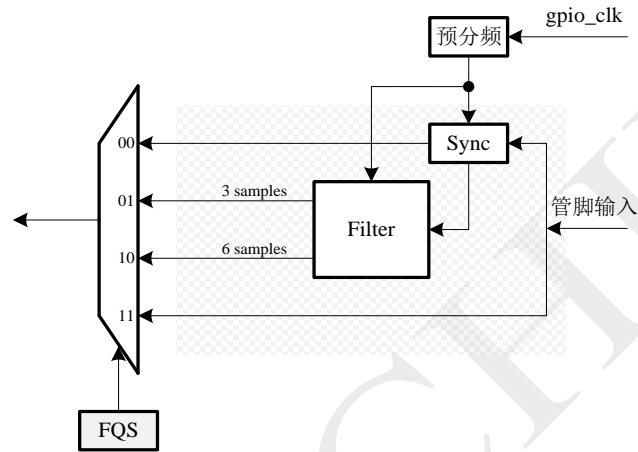


Figure 163 输入管脚滤波

18.7 EXINT 中断

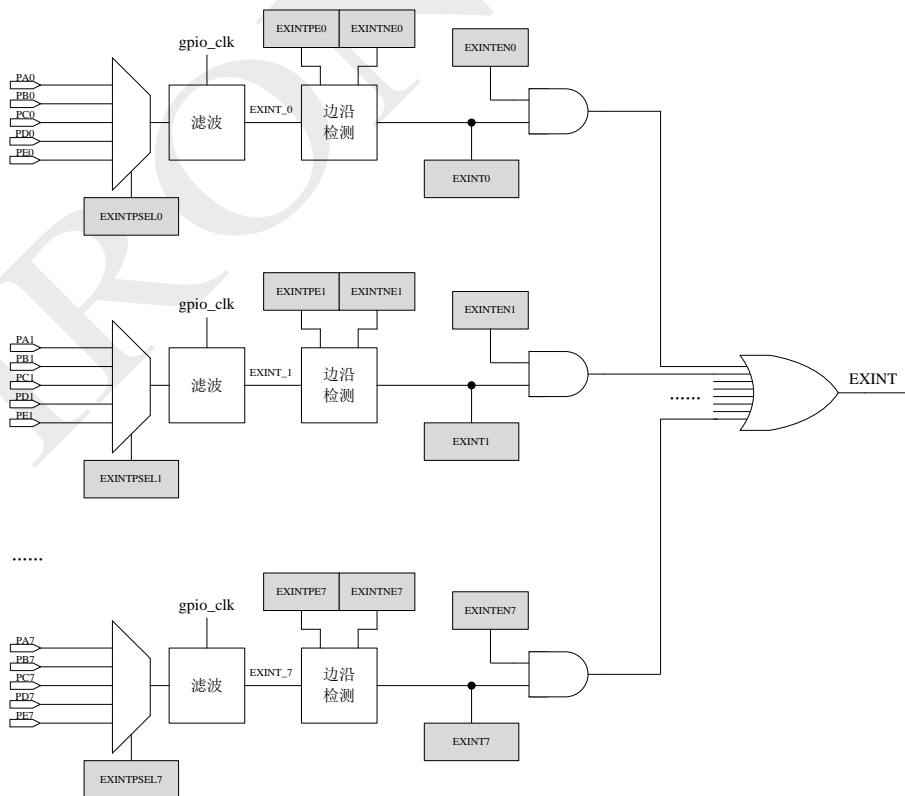


Figure 164 EXINT 中断

EXINT 中断包括 8 个外部管脚中断：

- 外部管脚中断 EXINT0:
 - 1) 配置寄存器 EXINTPSE0, 从 PA0, PB0, PC0, PD0 和 PE0 中选择 EXINT0 的中断管脚 PX0(其中 X=A, B, C, D, E)
 - 2) 配置寄存器 PX0CTRL.FQS, 设置管脚 PX0 输入滤波数
 - 3) 配置寄存器 PX0CTRL.MUX 为 0, 配置寄存器 PX0CTRL.EOUT 为 0, 设置管脚 PX0 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE0 和 EXINTNE0, 选择 EXINT0 中断类型
 - 5) 配置寄存器 EXINTEN0, 选择打开或关断 EXINT0 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
- 外部管脚中断 EXINT1:
 - 1) 配置寄存器 EXINTPSEL1, 从 PA1, PB1, PC1, PD1 和 PE1 中选择 EXINT1 的中断管脚 PX1(其中 X= A, B, C, D, E)
 - 2) 配置寄存器 PX1CTRL.FQS, 设置管脚 PX1 输入滤波数
 - 3) 配置寄存器 PX1CTRL.MUX 为 0, 配置寄存器 PX1CTRL.EOUT 为 0, 设置管脚 PX1 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE1,和 EXINTNE1, 选择 EXINT1 中断类型
 - 5) 配置寄存器 EXINTEN1, 选择打开或关断 EXINT1 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
- 外部管脚中断 EXINT2:
 - 1) 配置寄存器 EXINTPSEL2, 从 PA2, PB2, PC2, PD2 和 PE2 中选择 EXINT2 的中断管脚 PX2(其中 X= A, B, C, D, E)
 - 2) 配置寄存器 PX2CTRL.FQS, 设置管脚 PX2 输入滤波数
 - 3) 配置寄存器 PX2CTRL.MUX 为 0, 配置寄存器 PX2CTRL.EOUT 为 0, 设置管脚 PX2 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE2 和 EXINTNE2, 选择 EXINT2 中断类型
 - 5) 配置寄存器 EXINTEN2, 选择打开或关断 EXINT2 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断

- 8) 若使用查询方式, 则 5) ~7) 不用配置
- 外部管脚中断 EXINT3:
 - 1) 配置寄存器 EXINTPSEL3, 从 PA3, PB3, PC3, PD3 和 PE3 中选择 EXINT3 的中断管脚 PX3(其中 X= A, B, C, D, E)
 - 2) 配置寄存器 PX3CTRL.FQS, 设置管脚 PX3 输入滤波数
 - 3) 配置寄存器 PX3CTRL.MUX 为 0, 配置寄存器 PX3CTRL.EOUT 为 0, 设置管脚 PX3 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE3 和 EXINTNE3, 选择 EXINT3 中断类型
 - 5) 配置寄存器 EXINTEN3, 选择打开或关断 EXINT3 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
 - 外部管脚中断 EXINT4:
 - 1) 配置寄存器 EXINTPSEL4, 从 PA4, PB4, PC4, PD4 和 PE4 中选择 EXINT4 的中断管脚 PX4(其中 X= A, B, C, D, E)
 - 2) 配置寄存器 PX4CTRL.FQS, 设置管脚 PX4 输入滤波数
 - 3) 配置寄存器 PX4CTRL.MUX 为 0, 配置寄存器 PX4CTRL.EOUT 为 0, 设置管脚 PX4 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE4 和 EXINTNE4, 选择 EXINT4 中断类型
 - 5) 配置寄存器 EXINTEN4, 选择打开或关断 EXINT4 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
 - 外部管脚中断 EXINT5:
 - 1) 配置寄存器 EXINTPSEL5, 从 PA5, PB5, PC5, PD5 和 PE5 中选择 EXINT5 的中断管脚 PX5(其中 X= A, B, C, D, E)
 - 2) 配置寄存器 PX5CTRL.FQS, 设置管脚 PX5 输入滤波数
 - 3) 配置寄存器 PX5CTRL.MUX 为 0, 配置寄存器 PX5CTRL.EOUT 为 0, 设置管脚 PX5 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE5 和 EXINTNE5, 选择 EXINT5 中断类型
 - 5) 配置寄存器 EXINTEN5, 选择打开或关断 EXINT5 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置

- 外部管脚中断 EXINT6:
 - 1) 配置寄存器 EXINTPSEL6, 从 PA6, PB6, PC6, PD6 和 PE6 中选择 EXINT6 的中断管脚 PX6(其中 X= A, B, C, D, E)
 - 2) 配置寄存器 PX6CTRL.FQS, 设置管脚 PX6 输入滤波数
 - 3) 配置寄存器 PX6CTRL.MUX 为 0, 配置寄存器 PX6CTRL.EOUT 为 0, 设置管脚 PX6 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE6 和 EXINTNE6, 选择 EXINT6 中断类型
 - 5) 配置寄存器 EXINTEN6, 选择打开或关断 EXINT6 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
- 外部管脚中断 EXINT7:
 - 1) 配置寄存器 EXINTPSEL7, 从 PA7, PB7, PC7, PD7 和 PE7 中选择 EXINT7 的中断管脚 PX7(其中 X= A, B, C, D, E)
 - 2) 配置寄存器 PX7CTRL.FQS, 设置管脚 PX7 输入滤波数
 - 3) 配置寄存器 PX7CTRL.MUX 为 0, 配置寄存器 PX7CTRL.EOUT 为 0, 设置管脚 PX7 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE7 和 EXINTNE7, 选择 EXINT7 中断类型
 - 5) 配置寄存器 EXINTEN7, 选择打开或关断 EXINT7 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置

8 个外部管脚中断 EXINT0~ EXINT7 经过或逻辑组合成 EXINT 中断, 每个中断都有各自的中断使能, 组合后中断使能由寄存器 IEN0.2 控制。

硬件实现 8 个外部管脚中断 EXINT0~ EXINT7 时没有优先级, 软件可根据需要灵活编程。

18.8 PA 口唤醒

PA 口 (PA0~PA7) 可将芯片从任意低功耗模式唤醒。唤醒使能、输入滤波和唤醒有效沿可配置:

- a) 配置寄存器 PAxCTRL.FQS, 设置管脚 PA 输入滤波数;
- b) 配置寄存器 EXINTPE 和 EXINTNE, 设置 PA 唤醒有效沿, 可选择上升沿、下降沿或双沿, PA 口唤醒有效沿和 PA 中断有效沿一致。

若需要支持唤醒 DEEPSLEEP 模式, PA 管脚的滤波需要软件配置为直通模式。

18.9 GPIO 寄存器

GPIO 寄存器对应的 PAGE 如下：

- 管脚数据寄存器 P0, P1, P2, P3 和 P4 对应的 SFRPAGE 为所有 PAGE（公共寄存器），也即在所有 PAGE 配置这 5 个寄存器均生效；
- 其他寄存器对应的 SFRPAGE 为 2。

Table 292 GPIO 寄存器（SFRPAGE=0x02）

地址	寄存器名	类型	复位值	功能描述
0xB9	PA0CTRL	W/R	0x00	PA0 控制寄存器
0xBA	PA1CTRL	W/R	0x00	PA1 控制寄存器
0xBB	PA2CTRL	W/R	0x00	PA2 控制寄存器
0xBC	PA3CTRL	W/R	0x00	PA3 控制寄存器
0xBD	PA4CTRL	W/R	0x00	PA4 控制寄存器
0xBE	PA5CTRL	W/R	0x00	PA5 控制寄存器
0xBF	PA6CTRL	W/R	0x00	PA6 控制寄存器
0xC1	PA7CTRL	W/R	0x00	PA7 控制寄存器
0x89	PB0CTRL	W/R	0x04	PB0 控制寄存器
0x8A	PB1CTRL	W/R	0x04	PB1 控制寄存器
0x8B	PB2CTRL	W/R	0x00	PB2 控制寄存器
0x8C	PB3CTRL	W/R	0x00	PB3 控制寄存器
0x8D	PB4CTRL	W/R	0x30	PB4 控制寄存器
0x8E	PB5CTRL	W/R	0x30	PB5 控制寄存器
0x8F	PB6CTRL	W/R	0x30	PB6 控制寄存器
0xA9	PB7CTRL	W/R	0x00	PB7 控制寄存器
0x91	PC0CTRL	W/R	0x00	PC0 控制寄存器
0x92	PC1CTRL	W/R	0x00	PC1 控制寄存器
0x93	PC2CTRL	W/R	0x00	PC2 控制寄存器
0x94	PC3CTRL	W/R	0x00	PC3 控制寄存器
0x95	PC4CTRL	W/R	0x00	PC4 控制寄存器
0x96	PC5CTRL	W/R	0x00	PC5 控制寄存器
0x97	PC6CTRL	W/R	0x00	PC6 控制寄存器
0xAA	PC7CTRL	W/R	0x00	PC7 控制寄存器
0x99	PD0CTRL	W/R	0x00	PD0 控制寄存器
0x9A	PD1CTRL	W/R	0x00	PD1 控制寄存器

地址	寄存器名	类型	复位值	功能描述
0x9B	PD2CTRL	W/R	0x00	PD2 控制寄存器
0x9C	PD3CTRL	W/R	0x00	PD3 控制寄存器
0x9D	PD4CTRL	W/R	0x00	PD4 控制寄存器
0x9E	PD5CTRL	W/R	0x00	PD5 控制寄存器
0x9F	PD6CTRL	W/R	0x00	PD6 控制寄存器
0xAB	PD7CTRL	W/R	0x00	PD7 控制寄存器
0xA1	PE0CTRL	W/R	0x00	PE0 控制寄存器
0xA2	PE1CTRL	W/R	0x00	PE1 控制寄存器
0xA3	PE2CTRL	W/R	0x00	PE2 控制寄存器
0xA4	PE3CTRL	W/R	0x00	PE3 控制寄存器
0xA5	PE4CTRL	W/R	0x00	PE4 控制寄存器
0xA6	PE5CTRL	W/R	0x00	PE5 控制寄存器
0xA7	PE6CTRL	W/R	0x00	PE6 控制寄存器
0xAC	PE7CTRL	W/R	0x00	PE7 控制寄存器
0xAD	PADSRDR0	W/R	0xff	PAD 转换速率和驱动能力寄存器 0
0xAE	PADSRDR1	W/R	0x03	PAD 转换速率和驱动能力寄存器 1
0xB6	AFREMAP0	W/R	0x00	复用功能重映射寄存器 0
0xC3	AFREMAP1	W/R	0x00	复用功能重映射寄存器 1
0xB7	FPSC	W/R	0x00	滤波时钟预分频寄存器
0xC2	NRSTFQS	W/R	0x00	复位管脚滤波选择寄存器
0xC4	EXINTPE	W/R	0x00	EXINT 上升沿中断使能寄存器
0xC5	EXINTNE	W/R	0x00	EXINT 下降沿中断使能寄存器
0xC6	EXINTEN	W/R	0x00	EXINT 中断使能寄存器
0xB5	EXINTCON	W/R	0x00	EXINT 中断控制寄存器
0xB1	EXINTSEL0	W/R	0x00	EXINT 中断管脚选择寄存器 0
0xB2	EXINTSEL1	W/R	0x00	EXINT 中断管脚选择寄存器 1
0xB3	EXINTSEL2	W/R	0x00	EXINT 中断管脚选择寄存器 2
0xB4	EXINTSEL3	W/R	0x00	EXINT 中断管脚选择寄存器 3
0x80	P0	W/R	0xFF	PA 数据寄存器
0x90	P1	W/R	0xFF	PB 数据寄存器
0xA0	P2	W/R	0xFF	PC 数据寄存器

地址	寄存器名	类型	复位值	功能描述
0xB0	P3	W/R	0xFF	PD 数据寄存器
0xC8	P4	W/R	0xFF	PE 数据寄存器

18.9.1 PA0 控制寄存器(PA0CTRL) [0xB9]

Table 293 PA0 控制寄存器(PA0CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA0 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PA0 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PA0 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PA0 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PA0 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PA0 功能选择: 0: GPIO-PA0 (P0.0) 1: SPI_MISO
AIOEN	7	W/R	0	PA0 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA0 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.2 PA1 控制寄存器(PA1CTRL) [0xBA]

Table 294 PA1 控制寄存器(PA1CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA1 开漏使能： 0: 不使能 1: 使能
EOUT	1	W/R	0	PA1 作为 GPIO 时，输出使能： 0: 不使能 1: 使能
PU	2	W/R	0	PA1 上拉使能： 0: 不使能 1: 使能
PD	3	W/R	0	PA1 下拉使能： 0: 不使能 1: 使能
FQS	5-4	W/R	0	PA1 输入滤波选择： 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PA1 功能选择： 0: GPIO-PA1 (P0.1) 1: 复用功能-SPI_MOSI
AIOEN	7	W/R	0	PA1 作为模拟输入 IO 使能： 0: 不使能 1: 使能 该比特使能时，PA1 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.3 PA2 控制寄存器(PA2CTRL) [0xBB]

Table 295 PA2 控制寄存器(PA2CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA2 开漏使能： 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EOUT	1	W/R	0	PA2 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PA2 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PA2 下拉使能： 0：不使能 1：使能
FQS	5-4	W/R	0	PA2 输入滤波选择： 00：仅同步 01：可滤除小于 2 个滤波时钟周期的脉冲 10：可滤除小于 5 个滤波时钟周期的脉冲 11：直通
MUX	6	W/R	0	PA2 功能选择： 0：GPIO-PA2 (P0.2) 1：复用功能-SPI_NSS
AIOEN	7	W/R	0	PA2 作为模拟输入 IO 使能： 0：不使能 1：使能 该比特使能时，PA2 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.4 PA3 控制寄存器(PA3CTRL) [0xBC]

Table 296 PA3 控制寄存器(PA3CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA3 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PA3 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PA3 上拉使能：

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能
PD	3	W/R	0	PA3 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PA3 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PA3 功能选择: 0: GPIO-PA3 (P0.3) 1: 复用功能-SPI_SCK
AIOEN	7	W/R	0	PA3 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA3 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.5 PA4 控制寄存器(PA4CTRL) [0xBD]

Table 297 PA4 控制寄存器(PA4CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA4 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PA4 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PA4 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PA4 下拉使能: 0: 不使能

参数名	比特位	属性	复位值	描述
				1: 使能
FQS	5-4	W/R	0	PA4 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PA4 功能选择: 0: GPIO-PA4 (P0.4) 1: 复用功能-MOS_OC
AIOEN	7	W/R	0	PA4 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA4 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.6 PA5 控制寄存器(PA5CTRL) [0xBE]

Table 298 PA5 控制寄存器(PA5CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA5 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PA5 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PA5 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PA5 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PA5 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲

参数名	比特位	属性	复位值	描述
				10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PA5 功能选择: 0: GPIO-PA5 (P0.5) 1: 复用功能-MOS_OD
AIOEN	7	W/R	0	PA5 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA5 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.7 PA6 控制寄存器(PA6CTRL) [0xBF]

Table 299 PA6 控制寄存器(PA6CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA6 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PA6 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PA6 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PA6 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PA6 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PA6 功能选择: 0: GPIO-PA6 (P0.6)

参数名	比特位	属性	复位值	描述
				1: 复用功能-PWM_BKIN
AIOEN	7	W/R	0	PA6 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA6 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.8 PA7 控制寄存器(PA7CTRL) [0xC1]

Table 300 PA7 控制寄存器(PA7CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA7 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PA7 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	1	PA7 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PA7 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PA7 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PA7 功能选择: 0: GPIO-PA7 (P0.7) 1: 复用功能-PWM_ETR
AIOEN	7	W/R	0	PA7 作为模拟输入 IO 使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
				该比特使能时, PA7 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.9 PB0 控制寄存器(PB0CTRL) [0x89]

Table 301 PB0 控制寄存器(PB0CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB0 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PB0 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	1	PB0 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PB0 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PB0 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PB0 功能选择: 0: GPIO-PB0 (P1.0) 1: 复用功能-UARTTXD
Reserved	7	R	0	保留

18.9.10 PB1 控制寄存器(PB1CTRL) [0x8A]

Table 302 PB1 控制寄存器(PB1CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB1 开漏使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EOUT	1	W/R	0	PB1 作为 GPIO 时，输出使能： 0: 不使能 1: 使能
PU	2	W/R	1	PB1 上拉使能： 0: 不使能 1: 使能
PD	3	W/R	0	PB1 下拉使能： 0: 不使能 1: 使能
FQS	5-4	W/R	0	PB1 输入滤波选择： 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PB1 功能选择： 0: GPIO-PB1 (P1.1) 1: 复用功能-UARTRXD
Reserved	7	R	0	保留

18.9.11 PB2 控制寄存器(PB2CTRL) [0x8B]

Table 303 PB2 控制寄存器(PB2CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB2 开漏使能： 0: 不使能 1: 使能
EOUT	1	W/R	0	PB2 作为 GPIO 时，输出使能： 0: 不使能 1: 使能
PU	2	W/R	0	PB2 上拉使能： 0: 不使能 1: 使能
PD	3	W/R	0	PB2 下拉使能： 0: 不使能

参数名	比特位	属性	复位值	描述
				1: 使能
FQS	5-4	W/R	0	PB2 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PB2 功能选择: 0: GPIO-PB2 (P1.2) 1: 复用功能-PWM_CH0
Reserved	7	R	0	保留

18.9.12 PB3 控制寄存器(PB3CTRL) [0x8C]

Table 304 PB3 控制寄存器(PB3CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB3 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PB3 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PB3 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PB3 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PB3 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PB3 功能选择: 0: GPIO-PB3 (P1.3)

参数名	比特位	属性	复位值	描述
				1: 复用功能-PWM_CH0N
Reserved	7	R	0	保留

18.9.13 PB4 控制寄存器(PB4CTRL) [0x8D]

Table 305 PB4 控制寄存器(PB4CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB4 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PB4 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PB4 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PB4 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	11	PB4 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PB4 功能选择: 0: GPIO-PB4 (P1.4) 1: 复用功能-PWM_CH1
Reserved	7	R	0	保留

18.9.14 PB5 控制寄存器(PB5CTRL) [0x8E]

Table 306 PB5 控制寄存器(PB5CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB5 开漏使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EOUT	1	W/R	0	PB5 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PB5 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PB5 下拉使能： 0：不使能 1：使能
FQS	5-4	W/R	11	PB5 输入滤波选择： 00：仅同步 01：可滤除小于 2 个滤波时钟周期的脉冲 10：可滤除小于 5 个滤波时钟周期的脉冲 11：直通
MUX	6	W/R	0	PB5 功能选择： 0：GPIO-PB5（P1.5） 1：复用功能-PWM_CH1N
Reserved	7	R	0	保留

18.9.15 PB6 控制寄存器(PB6CTRL) [0x8F]

Table 307 PB6 控制寄存器(PB6CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB6 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PB6 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PB6 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PB6 下拉使能： 0：不使能

参数名	比特位	属性	复位值	描述
				1: 使能
FQS	5-4	W/R	11	PB6 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PB6 功能选择: 0: GPIO-PB6 (P1.6) 1: 复用功能-PWM_CH2
Reserved	7	R	0	保留

18.9.16 PB7 控制寄存器(PB7CTRL) [0xA9]

Table 308 PB7 控制寄存器(PB7CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB7 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PB7 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PB7 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PB7 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PB7 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PB7 功能选择: 0: GPIO-PB7 (P1.7)

参数名	比特位	属性	复位值	描述
				1: 复用功能-PWM_CH2N
Reserved	7	R	0	保留

18.9.17 PC0 控制寄存器(PC0CTRL) [0x91]

Table 309 PC0 控制寄存器(PC0CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PC0 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PC0 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PC0 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PC0 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PC0 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PC0 功能选择: 0: GPIO-PC0 (P2.0) 1: 复用功能-PWM_CH0
AIOEN	7	W/R	0	PC0 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PC0 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.18 PC1 控制寄存器(PC1CTRL) [0x92]

Table 310 PC1 控制寄存器(PC1CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PC1 开漏使能： 0: 不使能 1: 使能
EOUT	1	W/R	0	PC1 作为 GPIO 时，输出使能： 0: 不使能 1: 使能
PU	2	W/R	0	PC1 上拉使能： 0: 不使能 1: 使能
PD	3	W/R	0	PC1 下拉使能 0: 不使能 1: 使能
FQS	5-4	W/R	0	PC1 输入滤波选择： 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PC1 功能选择： 0: GPIO-PC1 (P2.1) 1: 复用功能-PWM_CH0N
AIOEN	7	W/R	0	PC1 作为模拟输入 IO 使能： 0: 不使能 1: 使能 该比特使能时，PC1 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.19 PC2 控制寄存器(PC2CTRL) [0x93]

Table 311 PC2 控制寄存器(PC2CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PC2 开漏使能： 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EOUT	1	W/R	0	PC2 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PC2 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PC2 下拉使能 0：不使能 1：使能
FQS	5-4	W/R	0	PC2 输入滤波选择： 00：仅同步 01：可滤除小于 2 个滤波时钟周期的脉冲 10：可滤除小于 5 个滤波时钟周期的脉冲 11：直通
MUX	6	W/R	0	PC2 功能选择： 0：GPIO-PC2 (P2.2) 1：复用功能-PWM_CH1
AIOEN	7	W/R	0	PC2 作为模拟输入 IO 使能： 0：不使能 1：使能 该比特使能时，PC2 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.20 PC3 控制寄存器(PC3CTRL) [0x94]

Table 312 PC3 控制寄存器(PC3CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PC3 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PC3 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PC3 上拉使能：

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能
PD	3	W/R	0	PC3 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PC3 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PC3 功能选择: 0: GPIO-PC3 (P2.3) 1: 复用功能-PWM_CH1N
AIOEN	7	W/R	0	PC3 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PC3 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.21 PC4 控制寄存器(PC4CTRL) [0x95]

Table 313 PC4 控制寄存器(PC4CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PC4 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PC4 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PC4 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PC4 下拉使能: 0: 不使能

参数名	比特位	属性	复位值	描述
				1: 使能
FQS	5-4	W/R	0	PC4 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PC4 功能选择: 0: GPIO-PC4 (P2.4) 1: 复用功能-PWM_CH2
AIOEN	7	W/R	0	PC4 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PC4 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.22 PC5 控制寄存器(PC5CTRL) [0x96]

Table 314 PC5 控制寄存器(PC5CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PC5 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PC5 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PC5 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PC5 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PC5 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲

参数名	比特位	属性	复位值	描述
				10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PC5 功能选择: 0: GPIO-PC5 (P2.5) 1: 复用功能-PWM_CH2N
AIOEN	7	W/R	0	PC5 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PC5 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.23 PC6 控制寄存器(PC6CTRL) [0x97]

Table 315 PC6 控制寄存器(PC6CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PC6 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PC6 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	1	PC6 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PC6 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PC6 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	1	PC6 功能选择: 0: GPIO-PC6 (P2.6)

参数名	比特位	属性	复位值	描述
				1: 复用功能-MOS_OC
AIOEN	7	W/R	0	PC6 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PC6 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.24 PC7 控制寄存器(PC7CTRL) [0xAA]

Table 316 PC7 控制寄存器(PC7CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PC7 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PC7 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PC7 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PC7 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PC7 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PC7 功能选择: 0: GPIO-PC7 (P2.7) 1: 复用功能-MOS_OD
AIOEN	7	W/R	0	PC7 作为模拟输入 IO 使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
				该比特使能时, PC7 的 IE,OE,PD,PU,OD 自动由硬件强制为低

18.9.25 PD0 控制寄存器(PD0CTRL) [0x99]

Table 317 PD0 控制寄存器(PD0CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PD0 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PD0 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	1	PD0 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PD0 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PD0 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PD0 功能选择: 0: GPIO-PD0 (P3.0) 1: 复用功能-I2C_SCL
Reserved	7	R	0	保留

18.9.26 PD1 控制寄存器(PD1CTRL) [0x9A]

Table 318 PD1 控制寄存器(PD1CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PD1 开漏使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EOUT	1	W/R	0	PD1 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	1	PD1 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PD1 下拉使能： 0：不使能 1：使能
FQS	5-4	W/R	0	PD1 输入滤波选择： 00：仅同步 01：可滤除小于 2 个滤波时钟周期的脉冲 10：可滤除小于 5 个滤波时钟周期的脉冲 11：直通
MUX	6	W/R	0	PD1 功能选择： 0：GPIO-PD1 (P3.1) 1：复用功能-I2C_SDA
Reserved	7	R	0	保留

18.9.27 PD2 控制寄存器(PD2CTRL) [0x9B]

Table 319 PD2 控制寄存器(PD2CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PD2 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PD2 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PD2 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PD2 下拉使能： 0：不使能

参数名	比特位	属性	复位值	描述
				1: 使能
FQS	5-4	W/R	0	PD2 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PD2 功能选择: 0: GPIO-PD2 (P3.2) 1: 复用功能-MOS_OC
Reserved	7	R	0	保留

18.9.28 PD3 控制寄存器(PD3CTRL) [0x9C]

Table 320 PD3 控制寄存器(PD3CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PD3 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PD3 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PD3 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PD3 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PD3 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PD3 功能选择: 0: GPIO-PD3 (P3.3)

参数名	比特位	属性	复位值	描述
				1: 复用功能-MOS_OD
Reserved	7	R	0	保留

18.9.29 PD4 控制寄存器(PD4CTRL) [0x9D]

Table 321 PD4 控制寄存器(PD4CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PD4 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PD4 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PD4 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PD4 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PD4 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PD4 功能选择: 0: GPIO-PD4 (P3.4) 1: 复用功能-SPI_MISO
Reserved	7	R	0	保留

18.9.30 PD5 控制寄存器(PD5CTRL) [0x9E]

Table 322 PD5 控制寄存器(PD5CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PD5 开漏使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EOUT	1	W/R	0	PD5 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PD5 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PD5 下拉使能： 0：不使能 1：使能
FQS	5-4	W/R	0	PD5 输入滤波选择： 00：仅同步 01：可滤除小于 2 个滤波时钟周期的脉冲 10：可滤除小于 5 个滤波时钟周期的脉冲 11：直通
MUX	6	W/R	0	PD5 功能选择： 0：GPIO-PD5 (P3.5) 1：复用功能-SPI_MOSI
Reserved	7	R	0	保留

18.9.31 PD6 控制寄存器(PD6CTRL) [0x9F]

Table 323 PD6 控制寄存器(PD6CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PD6 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PD6 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PD6 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PD6 下拉使能： 0：不使能

参数名	比特位	属性	复位值	描述
				1: 使能
FQS	5-4	W/R	0	PD6 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PD6 功能选择: 0: GPIO-PD6 (P3.6) 1: 复用功能-SPI_NSS
Reserved	7	R	0	保留

18.9.32 PD7 控制寄存器(PD7CTRL) [0xAB]

Table 324 PD7 控制寄存器(PD7CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PD7 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PD7 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PD7 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PD7 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PD7 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PD7 功能选择: 0: GPIO-PD7 (P3.7)

参数名	比特位	属性	复位值	描述
				1: 复用功能-SPI_SCK
Reserved	7	R	0	保留

18.9.33 PE0 控制寄存器(PE0CTRL) [0xA1]

Table 325 PE0 控制寄存器(PE0CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PE0 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PE0 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PE0 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PE0 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PE0 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PE0 功能选择: 0: GPIO-PE0 (P4.0) 1: 复用功能-PWM_CH0
Reserved	7	R	0	保留

18.9.34 PE1 控制寄存器(PE1CTRL) [0xA2]

Table 326 PE1 控制寄存器(PE1CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PE1 开漏使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EOUT	1	W/R	0	PE1 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PE1 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PE1 下拉使能： 0：不使能 1：使能
FQS	5-4	W/R	0	PE1 输入滤波选择： 00：仅同步 01：可滤除小于 2 个滤波时钟周期的脉冲 10：可滤除小于 5 个滤波时钟周期的脉冲 11：直通
MUX	6	W/R	0	PE1 功能选择： 0：GPIO-PE1 (P4.1) 1：复用功能-PWM_CH0N
Reserved	7	R	0	保留

18.9.35 PE2 控制寄存器(PE2CTRL) [0xA3]

Table 327 PE2 控制寄存器(PE2CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PE2 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PE2 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PE2 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PE2 下拉使能： 0：不使能

参数名	比特位	属性	复位值	描述
				1: 使能
FQS	5-4	W/R	0	PE2 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PE2 功能选择: 0: GPIO-PE2 (P4.2) 1: 复用功能-PWM_CH1
Reserved	7	R	0	保留

18.9.36 PE3 控制寄存器(PE3CTRL) [0xA4]

Table 328 PE3 控制寄存器(PE3CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PE3 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PE3 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PE3 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PE3 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PE3 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PE3 功能选择: 0: GPIO-PE3 (P4.3)

参数名	比特位	属性	复位值	描述
				1: 复用功能-PWM_CH1N
Reserved	7	R	0	保留

18.9.37 PE4 控制寄存器(PE4CTRL) [0xA5]

Table 329 PE4 控制寄存器(PE4CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PE4 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PE4 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PE4 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PE4 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PE4 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PE4 功能选择: 0: GPIO-PE4 (P4.4) 1: 复用功能-PWM_CH2
Reserved	7	R	0	保留

18.9.38 PE5 控制寄存器(PE5CTRL) [0xA6]

Table 330 PE5 控制寄存器(PE5CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PE5 开漏使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EOUT	1	W/R	0	PE5 作为 GPIO 时，输出使能： 0: 不使能 1: 使能
PU	2	W/R	0	PE5 上拉使能： 0: 不使能 1: 使能
PD	3	W/R	0	PE5 下拉使能： 0: 不使能 1: 使能
FQS	5-4	W/R	0	PE5 输入滤波选择： 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PE5 功能选择： 0: GPIO-PE5 (P4.5) 1: 复用功能-PWM_CH2N
Reserved	7	R	0	保留

18.9.39 PE6 控制寄存器(PE6CTRL) [0xA7]

Table 331 PE6 控制寄存器(PE6CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PE6 开漏使能： 0: 不使能 1: 使能
EOUT	1	W/R	0	PE6 作为 GPIO 时，输出使能： 0: 不使能 1: 使能
PU	2	W/R	0	PE6 上拉使能： 0: 不使能 1: 使能
PD	3	W/R	0	PE6 下拉使能： 0: 不使能

参数名	比特位	属性	复位值	描述
				1: 使能
FQS	5-4	W/R	0	PE6 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PE6 功能选择: 0: GPIO-PE6 (P4.6) 1: 复用功能-PWM_CH3
Reserved	7	R	0	保留

18.9.40 PE7 控制寄存器(PE7CTRL) [0xAC]

Table 332 PE7 控制寄存器(PE7CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PE7 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PE7 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PE7 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PE7 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PE7 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PE7 功能选择: 0: GPIO-PE7 (P4.7)

参数名	比特位	属性	复位值	描述
				1: 复用功能-PWM_ETR
Reserved	7	R	0	保留

18.9.41 PAD 转换速率和驱动能力寄存器 0(PADSRDR0) [0xAD]

Table 333 PAD 转换速率和驱动能力寄存器 0(PADSRDR0)

参数名	比特位	属性	复位值	描述
SRPA	0	W/R	1	PA0~7 转换速率控制位: 0: 转换速率快 1: 转换速率慢
DRPA	1	W/R	1	PA0~7 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
SRPB	2	W/R	1	PB0~7 转换速率控制位: 0: 转换速率快 1: 转换速率慢
DRPB	3	W/R	1	PB0~7 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
SRPC	4	W/R	1	PC0~7 转换速率控制位: 0: 转换速率快 1: 转换速率慢
DRPC	5	W/R	1	PC0~7 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
SRPD	6	W/R	1	PD0~7 转换速率控制位: 0: 转换速率快 1: 转换速率慢
DRPD	7	W/R	1	PD0~7 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低

18.9.42 PAD 转换速率和驱动能力寄存器 1(PADSRDR1) [0xAE]

Table 334 PAD 转换速率和驱动能力寄存器 1(PADSRDR1)

参数名	比特位	属性	复位值	描述
SRPE	0	W/R	1	PE0~7 转换速率控制位： 0: 转换速率快 1: 转换速率慢
DRPE	1	W/R	1	PE0~7 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
Reserved	7-2	R	0	保留

18.9.43 复用功能重映射寄存器 0(AFREMAP0) [0xB6]

Table 335 复用功能重映射寄存器 0(AFREMAP0)

参数名	比特位	属性	复位值	描述
PB0REMAP	0	W/R	0	PB0 复用功能重映射使能： 0: 不使能 1: 使能 PB0 的复用功能重映射为 BZOUT
PB1REMAP	1	W/R	0	PB1 复用功能重映射使能： 0: 不使能 1: 使能 PB1 的复用功能重映射为 PWM_BKIN
PB23REMAP	2	W/R	0	PB2 和 PB3 复用功能重映射使能： 0: 不使能 1: 使能 PB2 的复用功能重映射为 T0 PB3 的复用功能重映射为 T0_GATE
PB45REMAP	3	W/R	0	PB4 和 PB5 复用功能重映射使能： 0: 不使能 1: 使能 PB4 的复用功能重映射为 T1 PB5 的复用功能重映射为 T1_GATE
PB67REMAP	4	W/R	0	PB6 和 PB7 复用功能重映射使能： 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
				PB6 的复用功能重映射为 T2 PB7 的复用功能重映射为 T2_GATE
PC01REMAP	5	W/R	0	PC0 和 PC1 复用功能重映射使能： 0: 不使能 1: 使能 PC0 的复用功能重映射为 I2C_SCL PC1 的复用功能重映射为 I2C_SDA
PC25REMAP	6	W/R	0	PC2, PC3, PC4 和 PC5 复用功能重映射使能： 0: 不使能 1: 使能 PC2 的复用功能重映射为 SPI_MISO PC3 的复用功能重映射为 SPI_MOSI PC4 的复用功能重映射为 SPI_NSS PC5 的复用功能重映射为 SPI_SCK
PC7REMAP	7	W/R	0	PC7 复用功能重映射使能： 0: 不使能 1: 使能 PC7 的复用功能重映射为 PWM_BKIN

18.9.44 复用功能重映射寄存器 1(AFREMAP1) [0xC3]

Table 336 复用功能重映射寄存器 1(AFREMAP1)

参数名	比特位	属性	复位值	描述
PA23REMAP	0	W/R	0	PA23 复用功能重映射使能： 0: 不使能 1: 使能 PA2 的复用功能重映射为 I2C_SCL PA3 的复用功能重映射为 I2C_SDA
PA1REMAP	1	W/R	0	PA1 复用功能重映射使能： 0: 不使能 1: 使能 PA1 的复用功能重映射为 BZOUT
PC6REMAP	2	W/R	0	PC6 复用功能重映射使能：

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能 PC6 的复用功能重映射为 BZOUT
Reserved	7-3	R	0	保留

18.9.45 滤波时钟预分频寄存器(FPSC) [0xB7]

Table 337 滤波时钟预分频寄存器(FPSC)

参数名	比特位	属性	复位值	描述
FPSC	7-0	W/R	0	滤波时钟预分频系数: 滤波时钟周期 = (FPSC + 1) × T _{gpio_clk}

18.9.46 复位管脚滤波选择寄存器(NRSTFQS) [0xC2]

Table 338 复位管脚滤波选择寄存器(NRSTFQS)

参数名	比特位	属性	复位值	描述
NRSTFQS	1-0	W/R	00	复位管脚滤波选择: 00: 仅同步 01: 可滤除小于 2 个 gpio_clk 周期的脉冲 10: 可滤除小于 5 个 gpio_clk 周期的脉冲 11: 直通
Reserved	7-2	R	0	保留

18.9.47 EXINT 上升沿中断使能寄存器(EXINTPE) [0xC4]

Table 339 EXINT 上升沿中断使能寄存器(EXINTPE)

参数名	比特位	属性	复位值	描述
EXINTPE0	0	W/R	0	EXINT0 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE1	1	W/R	0	EXINT1 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE2	2	W/R	0	EXINT2 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE3	3	W/R	0	EXINT3 上升沿中断使能:

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能
EXINTPE4	4	W/R	0	EXINT4 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE5	5	W/R	0	EXINT5 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE6	6	W/R	0	EXINT6 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE7	7	W/R	0	EXINT7 上升沿中断使能: 0: 不使能 1: 使能

18.9.48 EXINT 下降沿中断使能寄存器(EXINTNE) [0xC5]

Table 340 EXINT 下降沿中断使能寄存器(EXINTNE)

参数名	比特位	属性	复位值	描述
EXINTNE0	0	W/R	0	EXINT0 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE1	1	W/R	0	EXINT1 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE2	2	W/R	0	EXINT2 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE3	3	W/R	0	EXINT3 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE4	4	W/R	0	EXINT4 下降沿中断使能: 0: 不使能

参数名	比特位	属性	复位值	描述
				1: 使能
EXINTNE5	5	W/R	0	EXINT5 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE6	6	W/R	0	EXINT6 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE7	7	W/R	0	EXINT7 下降沿中断使能: 0: 不使能 1: 使能

18.9.49 EXINT 中断使能寄存器(EXINTEN) [0xC6]

Table 341 EXINT 中断使能寄存器(EXINTEN)

参数名	比特位	属性	复位值	描述
EXINTEN0	0	W/R	0	EXINT0 中断使能: 0: 不使能 1: 使能
EXINTEN1	1	W/R	0	EXINT1 中断使能: 0: 不使能 1: 使能
EXINTEN2	2	W/R	0	EXINT2 中断使能: 0: 不使能 1: 使能
EXINTEN3	3	W/R	0	EXINT3 中断使能: 0: 不使能 1: 使能
EXINTEN4	4	W/R	0	EXINT4 中断使能: 0: 不使能 1: 使能
EXINTEN5	5	W/R	0	EXINT5 中断使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EXINTEN6	6	W/R	0	EXINT6 中断使能： 0: 不使能 1: 使能
EXINTEN7	7	W/R	0	EXINT7 中断使能： 0: 不使能 1: 使能

18.9.50 EXINT 中断控制寄存器(EXINTCON) [0xB5]

Table 342 EXINT 中断控制寄存器(EXINTCON)

参数名	比特位	属性	复位值	描述
EXINT0	0	W/R	0	EXINT0 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT0 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT1	1	W/R	0	EXINT1 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT1 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT2	2	W/R	0	EXINT2 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT2 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT3	3	W/R	0	EXINT3 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT3 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT4	4	W/R	0	EXINT4 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT4 中断申请时，该 bit 置为 1

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT5	5	W/R	0	EXINT5 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT5 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT6	6	W/R	0	EXINT6 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT6 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT7	7	W/R	0	EXINT7 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT7 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断

18.9.51 EXINT 中断管脚选择寄存器 0(EXINTSELO) [0xB1]

Table 343 EXINT 中断管脚选择寄存器 0(EXINTSELO)

参数名	比特位	属性	复位值	描述
EXINTPSEL0	2-0	W/R	0	EXINT0 中断管脚选择： 000: PA0 001: PB0 010: PC0 011: PD0 100: PE0 其他: 保留
Reserved	3	R	0	保留
EXINTPSEL1	6-4	W/R	0	EXINT1 中断管脚选择： 000: PA1 001: PB1 010: PC1

参数名	比特位	属性	复位值	描述
				011: PD1 100: PE1 其他: 保留
Reserved	7	R	0	保留

18.9.52 EXINT 中断管脚选择寄存器 1(EXINTSEL1) [0xB2]

Table 344 EXINT 中断管脚选择寄存器 1(EXINTSEL1)

参数名	比特位	属性	复位值	描述
EXINTPSEL2	2-0	W/R	0	EXINT2 中断管脚选择: 000: PA2 001: PB2 010: PC2 011: PD2 100: PE2 其他: 保留
Reserved	3	R	0	保留
EXINTPSEL3	6-4	W/R	0	EXINT3 中断管脚选择: 000: PA3 001: PB3 010: PC3 011: PD3 100: PE3 其他: 保留
Reserved	7	R	0	保留

18.9.53 EXINT 中断管脚选择寄存器 2(EXINTSEL2) [0xB3]

Table 345 EXINT 中断管脚选择寄存器 2(EXINTSEL2)

参数名	比特位	属性	复位值	描述
EXINTPSEL4	2-0	W/R	0	EXINT4 中断管脚选择: 000: PA4 001: PB4 010: PC4 011: PD4

参数名	比特位	属性	复位值	描述
				100: PE4 其他: 保留
Reserved	3	R	0	保留
EXINTPSEL5	6-4	W/R	0	EXINT5 中断管脚选择: 000: PA5 001: PB5 010: PC5 011: PD5 100: PE5 其他: 保留
Reserved	7	R	0	保留

18.9.54 EXINT 中断管脚选择寄存器 3(EXINTSEL3) [0xB4]

Table 346 EXINT 中断管脚选择寄存器 3(EXINTSEL3)

参数名	比特位	属性	复位值	描述
EXINTPSEL6	2-0	W/R	0	EXINT6 中断管脚选择: 000: PA6 001: PB6 010: PC6 011: PD6 100: PE6 其他: 保留
Reserved	3	R	0	保留
EXINTPSEL7	6-4	W/R	0	EXINT7 中断管脚选择: 000: PA7 001: PB7 010: PC7 011: PD7 100: PE7 其他: 保留
Reserved	7	R	0	保留

18.9.55 PA 数据寄存器(P0) [0x80]

Table 347 PA 数据寄存器(P0)

参数名	比特位	属性	复位值	描述
P0	7-0	W/R	0xFF	PA 数据寄存器

18.9.56 PB 数据寄存器(P1) [0x90]

Table 348 PB 数据寄存器(P1)

参数名	比特位	属性	复位值	描述
P1	7-0	W/R	0xFF	PB 数据寄存器

18.9.57 PC 数据寄存器(P2) [0xA0]

Table 349 PC 数据寄存器(P2)

参数名	比特位	属性	复位值	描述
P2	7-0	W/R	0xFF	PC 数据寄存器

18.9.58 PD 数据寄存器(P3) [0xB0]

Table 350 PD 数据寄存器(P3)

参数名	比特位	属性	复位值	描述
P3	7-0	W/R	0xFF	PD 数据寄存器

18.9.59 PE 数据寄存器(P4) [0xC8]

Table 351 PE 数据寄存器(P4)

参数名	比特位	属性	复位值	描述
P4	7-0	W/R	0xFF	PE 数据寄存器

19 电气特性

19.1 极限参数

Table 352 极限参数

符号	参数	最小值	最大值	单位
V _{DD}	供电电压	-0.3	5.5	V
V _{IN}	端口输入电压	-0.3	5.8	V
I _{VDD-VSS}	最大总电流	-	100	mA
I _{IO}	单个管脚注入电流	-10	10	mA
I _{IN(PIN)}	总注入电流	-50	50	mA
T _A	环境温度	-40	125	°C
T _J	结温度	-40	150	°C
T _{STG}	储存温度	-55	150	°C

备注：超过该“极限参数”可能导致器件永久性损坏。长时间在最大允许值或超过最大允许值的条件下工作可能影响器件的可靠性。这里只是给出能承受的最大载荷，并不意味着在此条件下器件的功能性操作无误。

19.2 工作条件

19.2.1 芯片供电

Table 353 芯片供电

符号	参数	测试条件	最小	典型	最大	单位
V _{DD}	数字部分工作电压		2.75		5.5	V
V _{DDA}	模拟部分工作电压	必须和 V _{DD} 一致	2.75		5.5	V
f _{sys_clk}	系统工作时钟	V _{DD} :2.75V~5.5V		73.728		MHz

19.2.2 功耗特性

19.2.2.1 电流特性

Table 354 电流特性

符号	参数	测试条件	最小	典型	最大	单位
I _{DD1}	工作电流 (正常模式电流)	f _{sys_clk} = 73.728MHz, 打开 ADC/PGA/CMP	22	24	26	mA
		f _{sys_clk} = 36.864MHz,	10	12	14	mA

符号	参数	测试条件	最小	典型	最大	单位
		打开 ADC/PGA/CMP				
I _{DD2}	IDLE 模式电流	f _{sys_clk} = 73.728MHz, 关闭 ADC/PGA/CMP	6	8	10	mA
I _{DD3}	IDLE 模式电流	f _{sys_clk} = 9.216 MHz, 关闭 ADC/PGA/CMP	1.5	2.5	3.5	mA
I _{DD4}	STOP 模式电流	关闭 ADC/PGA/CMP	80	90	100	uA
I _{DD5}	SLEEP 模式电流		2.9	3.1	3.3	uA
I _{DD6}	Deep sleep 模式电流		2.5	2.7	2.9	uA

19.2.2.2 低功耗唤醒时间

Table 355 低功耗唤醒时间

符号	参数	测试条件	典型	单位
t _{WUIDLE}	从 IDLE 模式唤醒时间	使用内部 73.728MHz 时钟	13.6	ns
t _{WUSTOP}	从 STOP 模式唤醒时间	使用内部 WDT 唤醒 若使用 PA 口唤醒，不考虑管脚滤波延时	9	us
t _{WUSLEEP}	从 SLEEP 模式唤醒时间	使用内部 WDT 唤醒 若使用 PA 口唤醒，不考虑管脚滤波延时	109	us
t _{WUDPSLEEP}	从 Deep sleep 模式唤醒时间	使用 PA 口唤醒	109	us

19.2.3 I/O 特性

Table 356 I/O 特性

符号	参数	测试条件	最小	典型	最大	单位
V _{IL}	I/O 输入低电平电压		-0.3		0.35V _{DD}	V
V _{IH}	I/O 输入高电平电压		0.65 V _{DD}		V _{DD} +0.3	V
V _{OL}	I/O 输出低电平电压				0.1 V _{DD}	V
V _{OH}	I/O 输出高电平电压		0.8 V _{DD}			V
I _{IO}	I/O 灌电流				20	mA
	I/O 源电流				-20	mA
R _{PU}	I/O 上拉电阻		20		100	KΩ
R _{PD}	I/O 下拉电阻		20		100	KΩ
C _{IN}	I/O 输入电容				10	pF

19.2.4 REGC 引脚特性

REGC 引脚连接芯片内部 MAIN LDO 的外部负载电容。

需在该引脚固定连接一个 1uF 的钽电容，使内部 LDO 输出稳定，如 Figure 165:

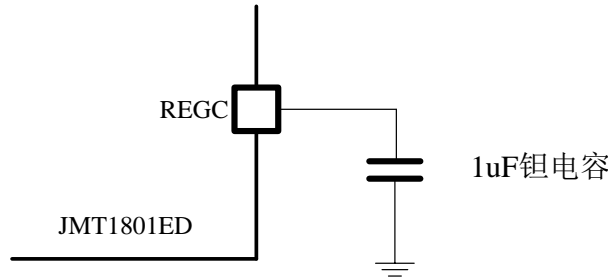


Figure 165 REGC 引脚连接示意图

19.2.5 NRST 引脚特性

NRST 引脚在芯片内部连接了一个不能断开的上拉电阻 (R_{PU})，滤波后生成内部复位信号。NRST 外部连接示意图参见 Figure 166，具体滤波配置参见寄存器 NRSTFQS 描述。

复位信号为低电平有效，输入的低电平必须满足以下两个条件，否则不会复位芯片：

- 小于 Table 356 定义的 V_{IL} 的最大值；
- 低电平宽度大于滤波的宽度。

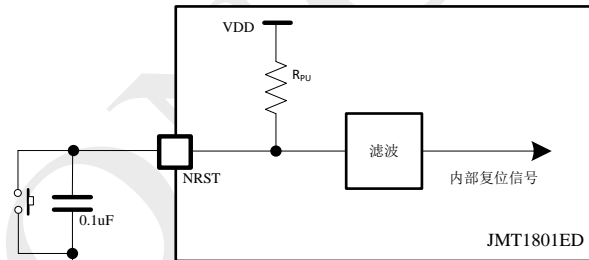


Figure 166 NRST 引脚连接示意图

19.2.6 上电/掉电条件

Table 357 上电/掉电工作条件

符号	参数	测试条件	最小	最大	单位
V_{POR}	产生 POR 复位的电压			500	mV
V_{BOR}	产生 BOR 复位的电压		2.6	2.7	V
t_{RT}	BOR 迟滞时间		30	150	us
t_{VDD}	V_{DD} 上升速率		3		V/ms

19.2.7 时钟特性

19.2.7.1 内部高速 RC 振荡器

Table 358 内部高速 RC 振荡器

符号	参数	测试条件	最小	典型	最大	单位
f_{HRC}	HRC 的频率		72.99	73.728	74.46	MHz
ACC_{HRC}	HRC 的精度	$T_A = -40 \sim 125^\circ\text{C}$	-1		+1	%
t_{HRC}	HRC 的启动时间		2	5	9	us
$I_{DD(HRC)}$	HRC 的功耗			0.8	1	mA

19.2.7.2 内部低速 RC 振荡器

Table 359 内部低速 RC 振荡器

符号	参数	测试条件	最小	典型	最大	单位
f_{LRC}	LRC 的频率		27	32	36	KHz
T_{LRC}	LRC 的启动时间				90	us
$I_{DD(LRC)}$	LRC 的功耗			0.5	1	uA

19.2.8 通信接口

19.2.8.1 I2C

JMT1801ED 的 I2C 接口符合标准 I2C 通信协议。SDA 和 SCL 不是真开漏的引脚，当配置为开漏输出时，在引脚和 VDD 之间的 PMOS 管被关闭，但仍然存在。

Table 360 I2C 接口特性

符号	参数	标准模式		快速模式		单位
		最小	最大	最小	最大	
$t_{w(SCLL)}$	SCL 时钟低电平时间	4.7		1.3		us
$t_{w(SCLH)}$	SCL 时钟高电平时间	4.0		0.6		us
$t_{su(SDA)}$	SDA 建立时间	250		100		ns
$t_h(SDA)$	SDA 数据保持时间	0		0	0.9	us
$t_r(SDA)$ $t_r(SCL)$	SDA 和 SCL 上升时间		1000	$20+0.1C_b$	300	ns
$t_f(SDA)$	SDA 和 SCL 下降时间		300		300	ns

符号	参数	标准模式		快速模式		单位
		最小	最大	最小	最大	
$t_r(SCL)$						
$t_h(START)$	起始条件保持时间	4.0		0.6		us
$t_{su}(START)$	重复起始条件建立时间	4.7		0.6		us
$t_{su}(STOP)$	停止条件建立时间	4.0		0.6		us
$t_w(STOP:START)$	停止条件至起始条件的间隔时间（总线空闲）	4.7		1.3		us
C_b	每条总线的容性负载		400		400	pF

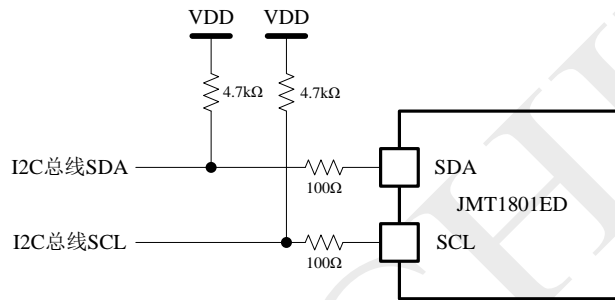


Figure 167 I2C 总线连接电路

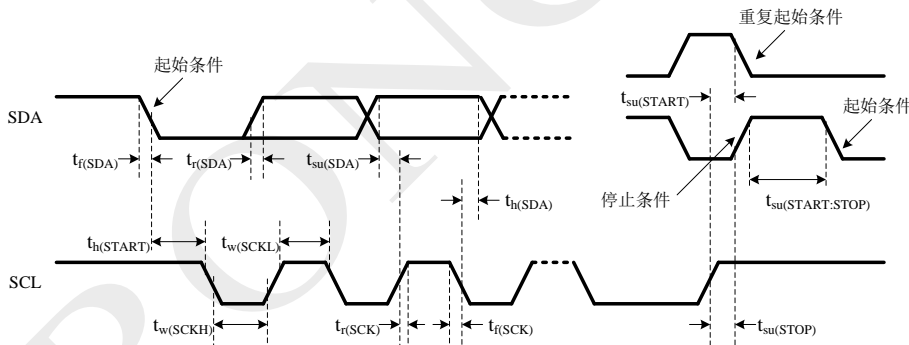


Figure 168 I2C 总线波形示意图

19.2.8.2 SPI

有关输入输出复用功能引脚（SPI_NSS、SPI_SCK、SPI_MOSI 和 SPI_MISO）的特性详情，参见 Table 356。

Table 361 SPI 特性

符号	参数	测试条件	最小	最大	单位
f_{SCK}	SPI 时钟频率	主机模式	0	18.432	MHz
$1/t_c(SCK)$		从机模式	0	18.432	
$t_r(SCK)$	SPI 时钟上升和下降时间	负载电容: $C=25pF$		9	ns

符号	参数	测试条件	最小	最大	单位
$t_r(\text{SCK})$					
$t_{su}(\text{NSS})$	NSS 建立时间	从机模式	$4t_{\text{sys_clk}}$		ns
$t_h(\text{NSS})$	NSS 保持时间	从机模式	$2t_{\text{sys_clk}}+20$		ns
$t_w(\text{SCKH})$ $t_w(\text{SCKL})$	SCK 高电平和低电平时间	主机模式, $f_{\text{sys_clk}}=73.728\text{MHz}$, 预分频系数为 1	50	60	ns
$t_{su}(\text{MI})$	数据输入建立时间	主机模式	6		ns
$t_{su}(\text{SI})$	数据输入建立时间	从机模式	6		ns
$t_h(\text{MI})$	数据输入保持时间	主机模式	6		ns
$t_h(\text{SI})$	数据输入保持时间	从机模式	6		ns
$t_a(\text{SO})$	数据输出访问时间	从机模式		$4t_{\text{sys_clk}}$	ns
$t_{dis}(\text{SO})$	数据输出禁止时间	从机模式	10		ns
$t_v(\text{SO})$	数据输出有效时间	从机模式 (使能边沿之后)		26	ns
$t_v(\text{MO})$	数据输出有效时间	主机模式 (使能边沿之后)		3	ns
$t_h(\text{SO})$	数据输出保持时间	从机模式 (使能边沿之后)	26		ns
$t_h(\text{MO})$	数据输出保持时间	主机模式 (使能边沿之后)	3		ns

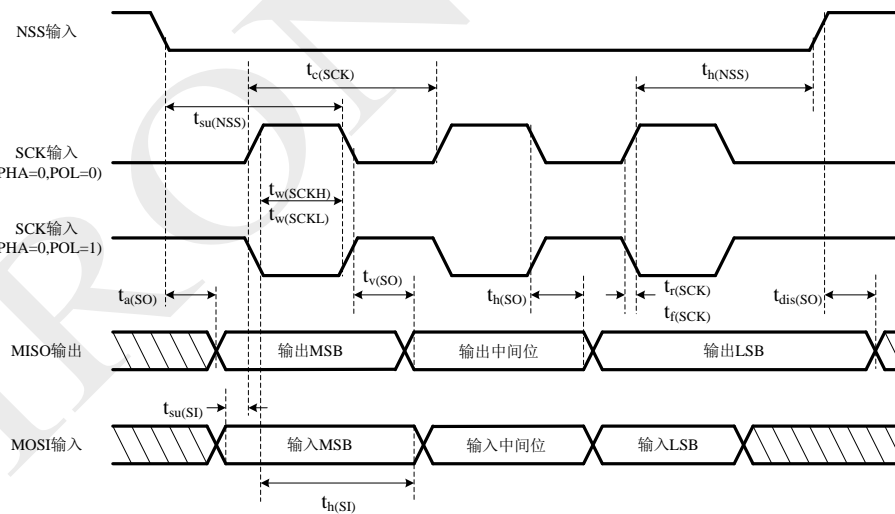


Figure 169 SPI 时序图 (从模式, PHA=0)

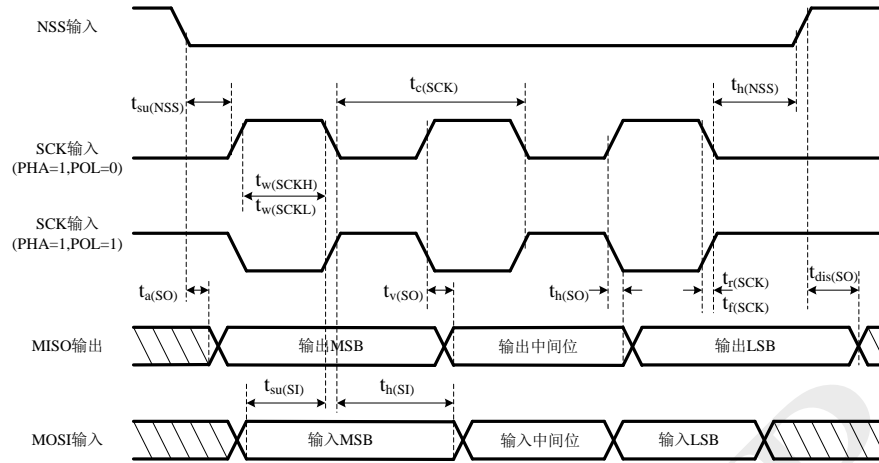


Figure 170 SPI 时序图（从模式，PHA=1）

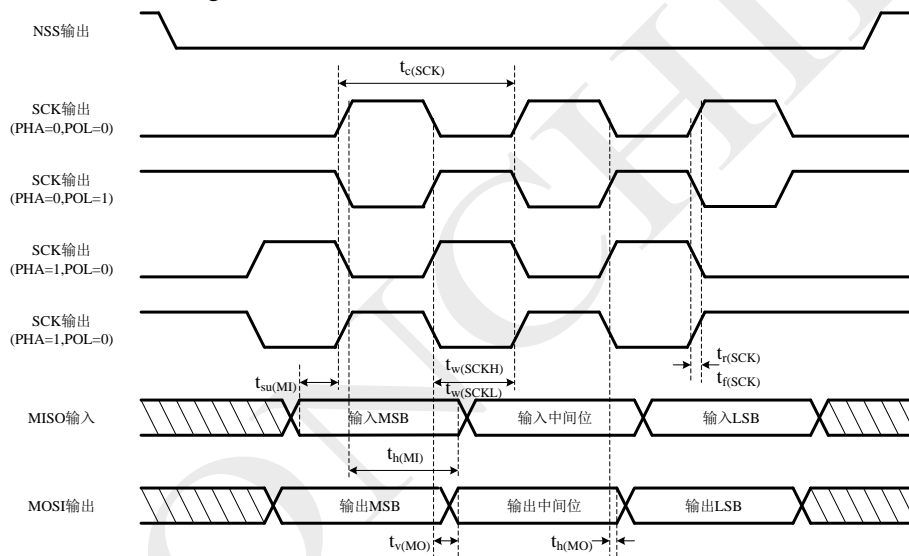


Figure 171 SPI 时序图（主模式）

19.2.9 FLASH 特性

Table 362 FLASH 特性

符号	参数	测试条件	最小	典型	最大	单位
t_{prog}	8 位的编程时间	$T_A = -40 \sim 125^\circ\text{C}$	16.5	18.5	20	us
t_{erase}	扇区（512 字节）擦除时间	$T_A = -40 \sim 125^\circ\text{C}$	4	5	6	ms
V_{prog}	编程电压		1.62	1.8	1.98	V

Table 363 FLASH 寿命和数据保存期限

符号	参数	测试条件	最小	典型	最大	单位
N_{END}	寿命	$V_{\text{prog}} = 1.8\text{V}$	20000			次
T_{DR}	数据保存期限	$T_A = 25^\circ\text{C}$	100			年

19.2.10 ADC 特性

Table 364 ADC 电气特性

符号	参数	最小	典型	最大	单位
N_R	ADC 分辨率			12	BIT
DNL	ADC 非线性微分误差	-1.0	± 0.9	1.5	LSB
INL	ADC 非线性积分误差	-3.0	± 1	2.0	LSB
E_{OFF}	ADC 偏移误差	-3.0	± 1.5	3.0	LSB
E_{GN}	ADC 增益误差		± 2	± 5	LSB
V_{REF}	ADC 参考电压	1.2	2.5	V_{DDA}	V
V_{INADC}	ADC 输入电压范围	0		V_{REF}	V
f_{SAMPLE}	ADC 采样速率	30		921.6	KSPS
$f_{adc_sample_clk}$	ADC 时钟频率	0		14.7456	MHz
$T_{adc_sample_clk}$	ADC 转换器时钟周期			67.8	ns
$T_{adc_sample_time}$	ADC 采样时间	$4/f_{SAMPLE}$			
$T_{adc_conversion_time}$	ADC 转换时间		$12/f_{SAMPLE}$		

为使得 ADC 达到更好的性能，模拟供电和参考电压的 PCB 设计建议按照 Figure 172 所示连接，其中电容建议选择瓷介电容，并且使得他们尽可能的靠近芯片。

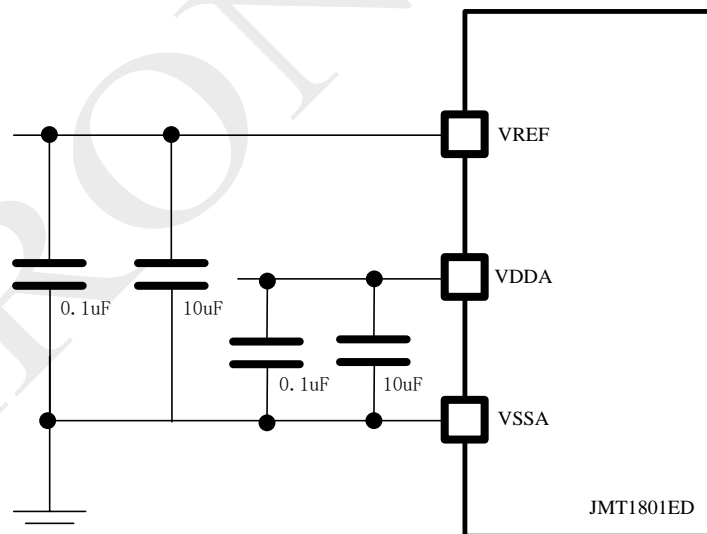


Figure 172 供电和参考电压去耦线路

19.2.11 PGA 特性

Table 365 PGA 电气特性

符号	参数	测试条件	最小	典型	最大	单位
V _{opos}	输入失调电压	校准后	-4		+4	mV
V _{cm}	共模电压范围		0		V _{DDA} -1.4	V
V _{INPGA}	输入电压范围		0		V _{DDA}	V
V _{REFPGA}	参考电压范围			1/2V _{REF}		V
PSRR	电源电压抑制比			60		dB
CMRR	共模抑制比			60		dB
SR	转换速率			2.5		V/us
GAIN	增益		1/2		64	倍
GBW	单位增益带宽			10		MHz
GAIN ERROR	增益误差		-1		+1	%
Voltage Spectral Density	电压噪声频谱密度		20		55	nV/ $\sqrt{\text{Hz}}$

19.2.12CMP 特性

Table 366 CMP 电气特性

符号	参数	测试条件	最小	典型	最大	单位
V _{cpmos}	输入失调电压		-15		+15	mV
V _{cm}	共模电压范围		0		V _{DDA} -1.4	V
V _{INCMP}	输入电压范围		0		V _{DDA}	V
V _{REFCMP}	参考电压范围		0		V _{REF}	V
PSRR	电源电压抑制比			60		dB
CMRR	共模抑制比			60		dB
t _{pd}	比较器响应时间			200		ns

19.2.13电气敏感性

19.2.13.1 ESD

ESD（静电放电敏感性试验）测试符合 JESD22-A114/C101 标准。

Table 367 ESD

符号	参数	测试条件	类型	最大	单位
V _{ESD(HBM)}	静电放电电压（人体模型）	T _A =25℃， 符合 JESD22-A114	2	6000	V
V _{ESD(CDM)}	静电放电电压（充电设备模型）	T _A =25℃，	II	500	V

符号	参数	测试条件	类型	最大	单位
		符合 JESD22-C101			

19.2.13.2 Latch Up

Latch Up（栓锁性能）测试符合 EIA/JESD 78A 集成电路栓锁标准。

Table 368 Latch Up

符号	参数	测试条件	类型
LU	静态栓锁类	T _A =125℃，符合 JESD78A	II类 A

20 封装特性

20.1 LQFP48L

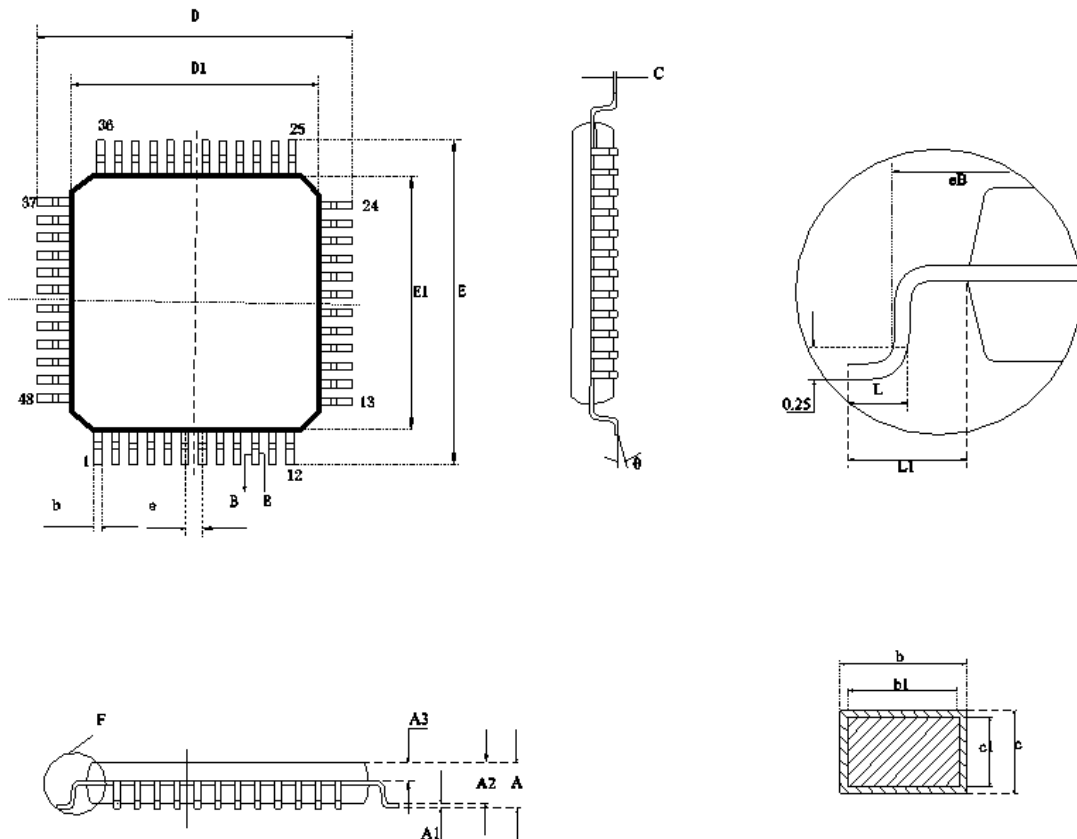


Figure 173 LQFP48L, 7X7mm 封装图

Table 369 LQFP48L, 7X7mm 机械数据

符号	毫米		
	最小	典型	最大
A			1.60
A1	0.05		0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.19		0.27
b1	0.18	0.20	0.23
c	0.13		0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20

符号	毫米		
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10		8.25
e	0.5		
L	0.45		0.75
L1	1		
θ	0°		7°

20.2 QFN40L

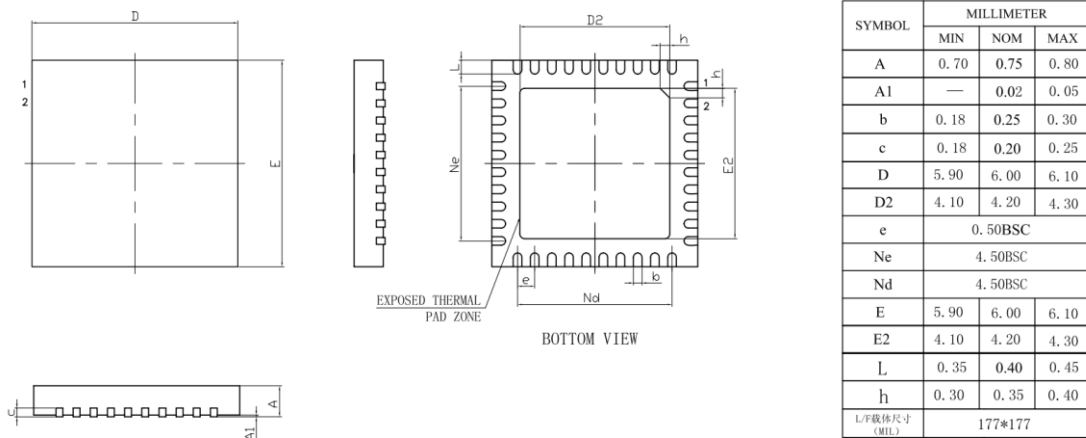


Figure 174 QFN40L, 6X6mm 封装图

20.3 QFN32L

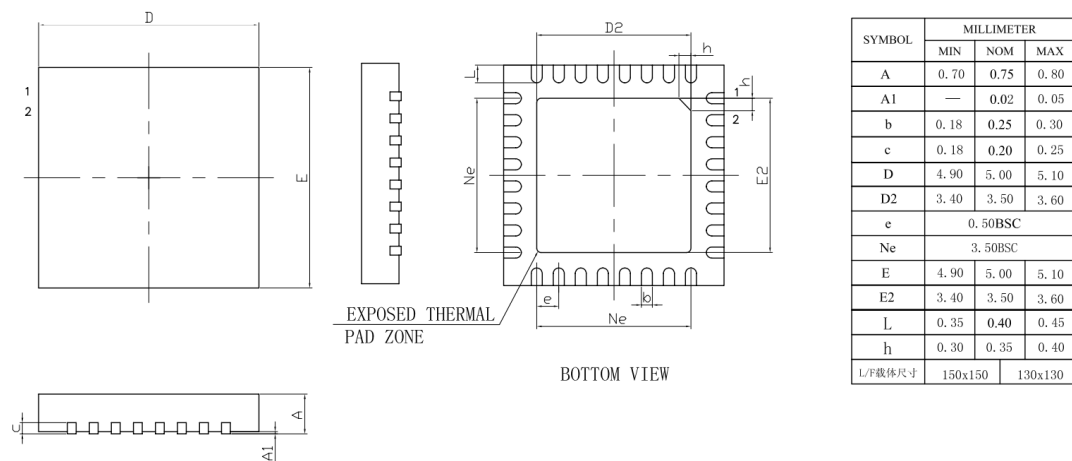


Figure 175 QFN32L, 5X5mm 封装图

20.4 TSSOP24L

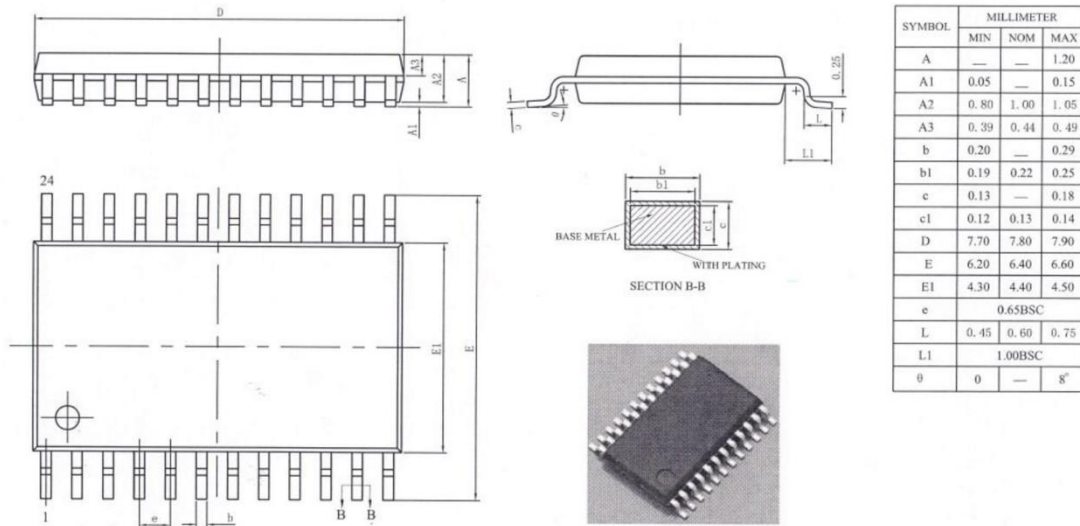


Figure 176 TSSOP24L, 7.80X4.40mm 封装图

20.5 TSSOP16L

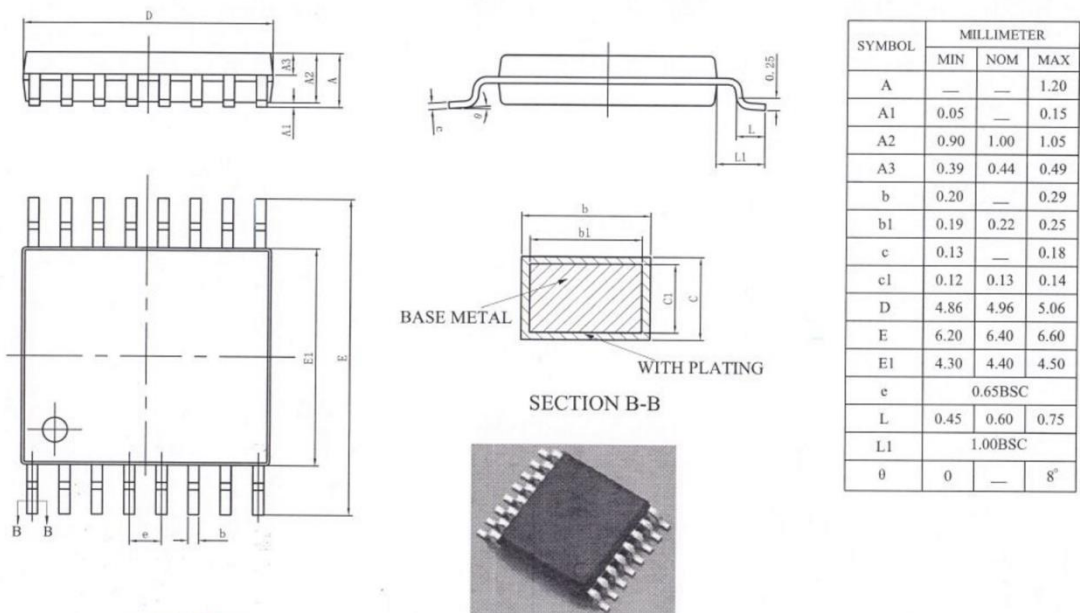


Figure 177 TSSOP16L, 4.96X4.40mm 封装图

重要提示

➤ 江苏宏云技术有限公司保留对本手册中产品在可靠性、功能和设计方面的改进作进一步说明的权利。本手册内容如有变动，恕不另行通知。订购前建议用户咨询销售代表。

➤ 江苏宏云技术有限公司的公司名称和标识都是江苏宏云技术有限公司的注册商标。

➤ 本手册中的信息仅供参考，诸如功能概要和应用电路示例旨在说明江苏宏云技术有限公司半导体器件的使用方法和操作示例；江苏宏云技术有限公司不保证参考运用时器件的正常工作。如果用户根据该信息使用器件实行相关开发，用户应承担因此引发的责任。基于上述信息的使用引起任何损失，江苏宏云技术有限公司不承担任何责任。

➤ 本手册内的任何技术信息，包括功能介绍和原理图，不应理解为使用和执行任何知识产权的许可，诸如专利权或著作权，或江苏宏云技术有限公司的其他权利。任何经由不当手段侵害江苏宏云技术有限公司专利权的公司、组织或个人，江苏宏云技术有限公司将采取一切可能的法律行动，遏止侵权行为，并追讨江苏宏云技术有限公司所受的损失、或侵权者所得的不法利益。江苏宏云技术有限公司也不保证使用该信息不侵犯任何第三方知识产权或其他权利。因使用该信息引起的对第三方知识产权或其他权利的侵权行为，江苏宏云技术有限公司不承担任何责任。

➤ 本手册所介绍的产品旨在一般用途而设计、开发和制造，包括但不限于一般的工业使用、通常办公使用、个人使用和家庭使用，不授权在以下设计、开发和制造：使用中伴随着致命风险或危险，若不加以特别高度安全保障，有可能导致对公众产生危害，甚至直接死亡、人身伤害、严重物质损失或其他损失（例如：核设施的核反应控制、航空飞行控制、空中交通控制、公共交通控制、医用维系生命系统、核武器系统的导弹发射控制等）；需要极高可靠性的应用领域（比如海底中转器和人造卫星）。注意上述领域内对使用该产品引起的用户和/或第三方的任何索赔或者损失，江苏宏云技术有限公司不承担任何责任。

➤ 半导体器件存在一定的故障发生概率。请用户对器件和设备采取冗余设计、消防设计、过电流等防护措施，及其它异常操作防护措施等安全设计，保证即使半导体器件发生故障的情况下，也不会造成人身伤害、社会损害或者重大损失。