



JMT1808R MCU

用户手册

IRONCHIP

江苏宏云技术有限公司

www.macrocloudtec.com

0512-58191012

IRONCHIP

1	产品概述.....	20
1.1	功能描述.....	20
1.2	主要特性.....	20
1.3	框图.....	22
1.4	应用范围.....	22
1.5	引脚配置.....	23
1.5.1	LQFP48L 引脚图.....	23
1.5.2	LQFP32L 引脚图.....	24
1.5.3	QFN32L 引脚图.....	25
1.5.4	TSSOP28L 引脚图.....	26
1.5.5	引脚说明.....	27
2	电源管理 (PMU)	30
2.1	概述.....	30
2.2	低功耗控制.....	30
2.2.1	降低系统时钟频率.....	31
2.2.2	外设时钟门控.....	32
2.2.3	关闭模拟模块.....	32
2.2.4	四种低功耗模式.....	32
2.2.5	OSC 开关说明.....	34
2.3	寄存器.....	34
2.3.1	低功耗控制寄存器(PMUCTR) [0xA1].....	35
3	时钟复位管理 (CRM)	36
3.1	时钟管理.....	36
3.1.1	概述.....	36
3.1.2	框图.....	36
3.1.3	功能描述.....	37
3.1.4	寄存器描述.....	42
3.2	复位管理.....	52
3.2.1	概述.....	52
3.2.2	框图.....	53
3.2.3	功能描述.....	53
3.2.4	寄存器.....	56

4	JMT51 MCU 核	61
4.1	概述.....	61
4.2	指令集.....	61
5	JMT018 DSP 核	67
5.1	概述.....	67
5.1.1	框图.....	67
5.1.2	系统总线.....	67
5.1.3	运算单元 (ALU)	68
5.1.4	乘累加单元(MAC).....	68
5.1.5	除法单元 (DIV)	68
5.1.6	硬件循环 (HW loop)	69
5.1.7	数据传输.....	69
5.1.8	存储单元.....	69
5.2	指令集.....	69
5.2.1	数据传输指令.....	69
5.2.2	算术运算指令.....	71
5.2.3	移位运算指令.....	72
5.2.4	逻辑运算指令.....	73
5.2.5	乘法运算指令.....	73
5.2.6	并行运算指令.....	74
5.2.7	程序控制指令.....	77
5.3	数据运算器.....	78
5.3.1	通用寄存器.....	78
5.3.2	累加器 A.....	79
5.3.3	乘法.....	83
5.3.4	乘加运算.....	84
5.3.5	加减运算.....	86
5.3.6	移位运算.....	86
5.3.7	四舍五入方法.....	87
5.4	DSP 存储空间.....	87
5.4.1	数据存储空间.....	87
5.4.2	程序存储空间.....	88
5.5	DSP 工作配置.....	89

5.6	DSP 代码生成与下载.....	89
5.6.1	DSP 代码生成	89
5.6.2	DSP 代码下载	89
5.7	DSP 内部寄存器.....	90
5.7.1	DSP 通用寄存器 R0.....	90
5.7.2	DSP 通用寄存器 R1.....	91
5.7.3	DSP 通用寄存器 R2.....	91
5.7.4	DSP 通用寄存器 R3.....	91
5.7.5	DSP 通用寄存器 R4.....	91
5.7.6	DSP 通用寄存器 R5.....	91
5.7.7	DSP 通用寄存器 R6.....	91
5.7.8	DSP 通用寄存器 R7.....	91
5.7.9	X 数据空间指针寄存器 DP0.....	92
5.7.10	Y 数据空间指针寄存器 DP1.....	92
5.7.11	DSP 程序地址指针寄存器 PC.....	92
5.7.12	乘法及乘加结果寄存器 A.....	92
5.7.13	LINK 寄存器	92
5.7.14	状态寄存器 PSR.....	92
5.7.15	模式寄存器 PMR	93
5.7.16	硬件循环次数寄存器 LOOP	93
5.7.17	硬件循环指令个数寄存器 LIR	94
5.8	JMT51 访问 DSP 寄存器.....	94
5.8.1	DSP 配置寄存器(DSP_CFG) [0xD4]	95
5.8.2	DSP 运行状态寄存器(DSP_STA) [0xD1].....	95
5.8.3	PC 寄存器低字节(DSP_PCL) [0xD2]	96
5.8.4	PC 寄存器高字节(DSP_PCH) [0xD3].....	96
5.8.5	R0 寄存器低字节(DSP_R0L) [0xD9].....	96
5.8.6	R0 寄存器高字节(DSP_R0H) [0xDA]	96
5.8.7	R1 寄存器低字节(DSP_R1L) [0xDB].....	96
5.8.8	R1 寄存器高字节(DSP_R1H) [0xDC]	97
5.8.9	R2 寄存器低字节(DSP_R2L) [0xDD].....	97
5.8.10	R2 寄存器高字节(DSP_R2H) [0xDE].....	97
5.8.11	R3 寄存器低字节(DSP_R3L) [0xDF]	97
5.8.12	R3 寄存器高字节(DSP_R3H) [0xE1].....	97
5.8.13	R4 寄存器低字节(DSP_R4L) [0xE2].....	97

5.8.14	R4 寄存器高字节(DSP_R4H) [0xE3].....	97
5.8.15	R5 寄存器低字节(DSP_R5L) [0xE4].....	98
5.8.16	R5 寄存器高字节(DSP_R5H) [0xE5].....	98
5.8.17	R6 寄存器低字节(DSP_R6L) [0xE6].....	98
5.8.18	R6 寄存器高字节(DSP_R6H) [0xE7].....	98
5.8.19	R7 寄存器低字节(DSP_R7L) [0xE9].....	98
5.8.20	R7 寄存器高字节(DSP_R7H) [0xEA].....	98
5.8.21	DP0 寄存器低字节(DSP_DP0L) [0xEB].....	98
5.8.22	DP0 寄存器高字节(DSP_DP0H) [0xEC].....	99
5.8.23	DP1 寄存器低字节(DSP_DP1L) [0xED].....	99
5.8.24	DP1 寄存器高字节(DSP_DP1H) [0xEE].....	99
5.8.25	A 寄存器 7~0 比特(DSP_A0L) [0xEF].....	99
5.8.26	A 寄存器 15~8 比特(DSP_A0H) [0xF1].....	99
5.8.27	A 寄存器 23~16 比特(DSP_A1L) [0xF2].....	99
5.8.28	A 寄存器 31~24 比特(DSP_A1H) [0xF3].....	99
5.8.29	A 寄存器 39~32 比特(DSP_A2) [0xF4].....	100
5.8.30	LNKR 寄存器低字节(DSP_LNKRL) [0xF5].....	100
5.8.31	LNKR 寄存器高字节(DSP_LNKRH) [0xF6].....	100
5.8.32	PSR 寄存器低字节(DSP_PSRL) [0xF7].....	100
5.8.33	PSR 寄存器高字节(DSP_PSRH) [0xF9].....	101
5.8.34	PMR 寄存器低字节(DSP_PMRL) [0xFA].....	101
5.8.35	PMR 寄存器高字节(DSP_PMRH) [0xFB].....	101
5.8.36	LOOP 寄存器低字节(DSP_LOOPL) [0xFC].....	101
5.8.37	LOOP 寄存器高字节(DSP_LOOPH) [0xFD].....	101
5.8.38	LIR 寄存器低字节(DSP_LIRL) [0xFE].....	102
5.8.39	LIR 寄存器高字节(DSP_LIRH) [0xFF].....	102
6	中断.....	103
6.1	概述.....	103
6.2	外部中断 (INT0/INT1).....	105
6.3	中断优先级.....	105
6.4	中断处理过程.....	106
6.5	中断响应时间.....	106
6.6	中断寄存器.....	107

6.6.1	中断使能寄存器 0(IEN0) [0xA8]	107
6.6.2	中断使能寄存器 1(IEN1) [0xB8]	108
6.6.3	中断使能寄存器 2 (IEN2) [0xC0]	109
6.6.4	中断优先级寄存器 0 (IP0) [0xB9]	109
6.6.5	中断优先级寄存器 1(IP1) [0xBA]	110
6.6.6	中断优先级寄存器 2(IP2) [0xBB].....	111
7	存储器.....	112
7.1	<i>JMT51</i> 存储器.....	112
7.1.1	程序存储器.....	112
7.1.2	外部数据存储器.....	113
7.1.3	内部数据存储器.....	113
7.1.4	特殊功能寄存器.....	115
7.2	<i>JMT018</i> 存储器.....	120
7.3	存储器访问冲突说明	121
7.4	<i>JMT51</i> 核寄存器.....	122
7.4.1	ACC 寄存器(A) [0xE0]	122
7.4.2	B 寄存器(B) [0xF0].....	122
7.4.3	堆栈指针(SP) [0x81]	122
7.4.4	程序状态字寄存器(PSW) [0xD0]	123
7.4.5	数据指针低字节(DPL) [0x82]	124
7.4.6	数据指针高字节(DPH) [0x83].....	124
7.4.7	MOVX 指令设置寄存器(MOVXCON) [0x86].....	124
7.5	<i>PAGE</i> 堆栈寄存器.....	125
7.5.1	SFR PAGE 寄存器(SFRPAGE) [0x84]	125
7.5.2	<i>PAGE</i> 堆栈使能寄存器(SFRPGEN) [0x85]	126
7.5.3	<i>PAGE</i> 堆栈寄存器 0(PGSTACK0) [0x91].....	126
7.5.4	<i>PAGE</i> 堆栈寄存器 1(PGSTACK1) [0x92].....	126
7.5.5	<i>PAGE</i> 堆栈寄存器 2(PGSTACK2) [0x93].....	126
7.5.6	<i>PAGE</i> 堆栈寄存器 3(PGSTACK3) [0x94].....	127
8	Flash 存储器	128
8.1	概述.....	128
8.2	<i>FLASH</i> 操作.....	128
8.2.1	<i>FLASH</i> 数据读取	128

8.2.2	FLASH 扇区擦除.....	129
8.2.3	FLASH 编程.....	129
8.3	FLASH 寄存器.....	130
8.3.1	FLASH 控制寄存器(FLSC) [0xF9].....	130
8.3.2	FLASH 关键字寄存器(FLSK) [0xFA].....	130
9	看门狗定时器 (WDT)	131
9.1	概述.....	131
9.2	功能描述.....	131
9.3	寄存器.....	131
9.3.1	看门狗分频系数寄存器(WDTCKDIV) [0xD9].....	131
9.3.2	看门狗比较值寄存器(WDTINT) [0xDA].....	132
9.3.3	看门狗中断标志寄存器(WDTINTF) [0xDB].....	132
9.3.4	看门狗喂狗寄存器(WDTLD) [0xDC].....	133
9.3.5	看门狗启停寄存器(WDTST) [0xDD].....	133
10	实时时钟 (RTC)	134
10.1	概述.....	134
10.2	实时时钟.....	134
10.3	闹钟.....	136
10.4	RTC 时钟调校.....	136
10.5	实时时钟寄存器更新.....	137
10.6	实时时钟寄存器读取.....	137
10.7	RTC 中断.....	137
10.8	RTC 寄存器.....	138
10.8.1	RTC 控制寄存器 0(RTCCON0) [0xA9].....	138
10.8.2	RTC 控制寄存器 1(RTCCON1) [0xAA].....	139
10.8.3	RTC 时标控制寄存器(RTCTMC) [0xAB].....	140
10.8.4	RTC 秒寄存器(RTCSEC) [0xAC].....	140
10.8.5	RTC 分钟寄存器(RTCMIN) [0xAD].....	141
10.8.6	RTC 小时寄存器(RTCHOUR) [0xAE].....	141
10.8.7	RTC 星期寄存器(RTCWEEK) [0xAF].....	141
10.8.8	RTC 日寄存器(RTCDAY) [0xB1].....	141
10.8.9	RTC 月寄存器(RTCMON) [0xB2].....	141

10.8.10	RTC 年寄存器(RTCYEAR) [0xB3].....	142
10.8.11	闹钟 0 秒寄存器(AL0SEC) [0xB4].....	142
10.8.12	闹钟 0 分钟寄存器(AL0MIN) [0xB5].....	142
10.8.13	闹钟 0 小时寄存器(AL0HOUR) [0xB6].....	142
10.8.14	闹钟 0 星期寄存器(AL0WEEKS) [0xB7].....	142
10.8.15	闹钟 1 分钟寄存器(AL1MIN) [0xB9].....	143
10.8.16	闹钟 1 小时寄存器(AL1HOUR) [0xBA].....	143
10.8.17	闹钟 1 日寄存器(AL1DAY) [0xBB].....	143
10.8.18	闹钟 1 月寄存器(AL1MON) [0xBC].....	144
10.8.19	RTC 时钟调校寄存器 0(RTCADJ0) [0xBD].....	144
10.8.20	RTC 时钟调校寄存器 1(RTCADJ1) [0xBE].....	144
10.8.21	RTC 中断使能寄存器(RTCIE) [0xBF].....	144
10.8.22	RTC 中断标志寄存器(RTCIF) [0xC7].....	145
11	定时器/计数器 (TIMER)	146
11.1	概述.....	146
11.2	TIMER 管脚配置.....	146
11.3	Timer0 工作模式.....	146
11.3.1	模式 0(13 位定时器/计数器).....	146
11.3.2	模式 1(16 位定时器/计数器).....	147
11.3.3	模式 2(8 位自动重装定时器/计数器).....	147
11.3.4	模式 3(两个 8 比特定器).....	148
11.4	Timer1 工作模式.....	149
11.4.1	模式 0(13 位定时器/计数器).....	149
11.4.2	模式 1(16 位定时器/计数器).....	149
11.4.3	模式 2(8 位自动重装定时器/计数器).....	150
11.5	Timer2 工作模式.....	150
11.5.1	模式 0(13 位定时器/计数器).....	150
11.5.2	模式 1(16 位定时器/计数器).....	151
11.5.3	模式 2(16 位自动重装定时器/计数器).....	151
11.5.4	模式 3(1 个 8 比特定器/计数器).....	152
11.6	TIMER 寄存器.....	152
11.6.1	Timer0/1 控制寄存器(TCON) [0x88].....	153
11.6.2	Timer0/1 模式寄存器(TMOD) [0x89].....	154

11.6.3	Timer0 低 8 位(TL0) [0x8A]	155
11.6.4	Timer1 低 8 位(TL1) [0x8B]	155
11.6.5	Timer0 高 8 位(TH0) [0x8C].....	155
11.6.6	Timer1 高 8 位(TH1) [0x8D].....	155
11.6.7	Timer0/1 预分频控制器(TPSC) [0x8E].....	155
11.6.8	Timer2 控制寄存器(T2CON) [0xD1]	156
11.6.9	Timer2 重载寄存器低 8 位(RL2) [0xD2]	156
11.6.10	Timer2 重载寄存器高 8 位(RH2) [0xD3].....	156
11.6.11	Timer2 低 8 位(TL2) [0xD4]	157
11.6.12	Timer2 高 8 位(TH2) [0xD5].....	157
11.6.13	Timer2 预分频控制器(T2PSC) [0xD6].....	157
12	高级定时器 (PWM)	158
12.1	概述.....	158
12.2	特性.....	158
12.3	框图.....	160
12.4	输入输出引脚配置.....	160
12.5	功能描述.....	160
12.5.1	计数器.....	161
12.5.2	输入捕获功能.....	173
12.5.3	从模式控制.....	175
12.5.4	输出比较功能.....	181
12.5.5	主模式控制.....	207
12.5.6	中断.....	209
12.5.7	高级定时器同步.....	211
12.6	寄存器描述.....	216
12.6.1	TIMx 控制寄存器 0 (TIMx_CONR0) [TIM0:0xBB/TIM1:0xFB]	221
12.6.2	TIMx 控制寄存器 1 (TIMx_CONR1) [TIM0:0x89/TIM1:0xC9]	223
12.6.3	TIMx 控制寄存器 2 (TIMx_CONR2) [TIM0:0x8A/TIM1:0xCA].....	224
12.6.4	TIMx 控制寄存器 3 (TIMx_CONR3) [TIM0:0x8B/TIM1:0xCB]	225
12.6.5	TIMx 从模式控制寄存器 0 (TIMx_TGICR0) [TIM0:0x8C/TIM1:0xCC]	226
12.6.6	TIMx 从模式控制寄存器 1 (TIMx_TGICR1) [TIM0:0x8D/TIM1:0xCD].....	228
12.6.7	TIMx 中断使能寄存器 0 (TIMx_IER) [TIM0:0x8E/TIM1:0xCE]	230
12.6.8	TIMx 中断使能寄存器 1 (TIMx_IER_NOCMP) [TIM0:0xB7/TIM7:0xF7]	231

12.6.9	TIMx 通道 CC0 输入捕获/输出比较模式控制寄存器 (TIMx_CC0MR)	
	[TIM0:0x8F/TIM1:0xCF]	231
12.6.10	TIMx 通道 CC1 输入捕获/输出比较模式控制寄存器 (TIMx_CC1MR)	
	[TIM0:0x91/TIM1:0xD1]	236
12.6.11	TIMx 通道 CC2 输入捕获/输出比较模式控制寄存器 (TIMx_CC2MR)	
	[TIM0:0x92/TIM1:0xD2]	241
12.6.12	TIMx 通道 CC3 输入捕获/输出比较模式控制寄存器 (TIMx_CC3MR)	
	[TIM0:0x93/TIM1:0xD3]	245
12.6.13	TIMx 通道使能控制寄存器 (TIMx_CCENR) [TIM0:0x94/TIM1:0xD4]	249
12.6.14	TIMx 通道极性控制寄存器 (TIMx_CCPS) [TIM0:0x95/TIM1:0xD5]	251
12.6.15	TIMx 死区时间寄存器 (TIMx_DTG) [TIM0:0x96/TIM1:0xD6]	252
12.6.16	TIMx 周期值自动装载寄存器的低 8 位 (TIMx_ARRL) [TIM0:0x97/TIM1:0xD7]	
	252	
12.6.17	TIMx 周期值自动装载寄存器的高 8 位 (TIMx_ARRH) [TIM0:0x99/TIM1:0xD9]	
	253	
12.6.18	TIMx 预分频寄存器的低 8 位 (TIMx_PSCL) [TIM0:0x9A/TIM1:0xDA]	253
12.6.19	TIMx 预分频寄存器的高 8 位 (TIMx_PSCH) [TIM0:0x9B/TIM1:0xDB]	253
12.6.20	TIMx 重复计数寄存器 (TIMx_RCR) [TIM0:0x9C/TIM1:0xDC]	254
12.6.21	TIMx 通道 CC0 捕获/向上计数比较寄存器的低 8 位 (TIMx_CC0RUL)	
	[TIM0:0x9D/TIM1:0xDD]	254
12.6.22	TIMx 通道 CC0 捕获/向上计数比较寄存器的高 8 位 (TIMx_CC0RUH)	
	[TIM0:0x9E/TIM1:0xDE]	255
12.6.23	TIMx 通道 CC0 向下计数比较寄存器的低 8 位 (TIMx_CC0RDL) [TIM0:0x9F/	
	TIM1:0xDF]	256
12.6.24	TIMx 通道 CC0 向下计数比较寄存器的高 8 位 (TIMx_CC0RDH) [TIM0:0xA1/	
	TIM1:0xE1]	257
12.6.25	TIMx 通道 CC1 捕获/向上计数比较寄存器的低 8 位 (TIMx_CC1RUL)	
	[TIM0:0xA2/TIM1:0xE2]	258
12.6.26	TIMx 通道 CC1 捕获/向上计数比较寄存器的高 8 位 (TIMx_CC1RUH)	
	[TIM0:0xA3/TIM1:0xE3]	259
12.6.27	TIMx 通道 CC1 向下计数比较寄存器的低 8 位 (TIMx_CC1RDL)	
	[TIM0:0xA4/TIM1:0xE4]	260
12.6.28	TIMx 通道 CC1 向下计数比较寄存器的高 8 位 (TIMx_CC1RDH)	
	[TIM0:0xA5/TIM1:0xE5]	260
12.6.29	TIMx 通道 CC2 捕获/向上计数比较寄存器的低 8 位 (TIMx_CC2RUL)	

[TIM0:0xA6/TIM1:0xE6].....	261
12.6.30 TIMx 通道 CC2 捕获/向上计数比较寄存器的高 8 位 (TIMx_CC2RUH)	
[TIM0:0xA7/TIM1:0xE7].....	262
12.6.31 TIMx 通道 CC2 向下计数比较寄存器的低 8 位 (TIMx_CC2RDL)	
[TIM0:0xA9/TIM0:0xE9].....	263
12.6.32 TIMx 通道 CC2 向下计数比较寄存器的高 8 位 (TIMx_CC2RDH)	
[TIM0:0xAA/TIM1:0xEA].....	264
12.6.33 TIMx 通道 CC3 捕获/向上计数比较寄存器的低 8 位 (TIMx_CC3RUL)	
[TIM0:0xAB/TIM1:0xEB].....	265
12.6.34 TIMx 通道 CC3 捕获/向上计数比较寄存器的高 8 位 (TIMx_CC3RUH)	
[TIM0:0xAC/TIM1:0xEC].....	266
12.6.35 TIMx 通道 CC3 向下计数比较寄存器的低 8 位 (TIMx_CC3RDL)	
[TIM0:0xAD/TIM1:0xED].....	267
12.6.36 TIMx 通道 CC3 向下计数比较寄存器的高 8 位 (TIMx_CC3RDH)	
[TIM0:0xAE/TIM1:0xEE].....	268
12.6.37 TIMx 刹车控制寄存器 (TIMx_BRKC) [TIM0:0xAF/TIM1:0xEF].....	269
12.6.38 TIMx 事件产生寄存器 (TIMx_EGR) [TIM0:0xB1/TIM1:0xF1].....	271
12.6.39 TIMx 互补模式控制寄存器 (TIMx_CMPLMTARY) [TIM0:0xB2/TIM1:0xF2]	273
12.6.40 TIMx 中断标志寄存器 0 (TIMx_ISR0) [TIM0:0xB3/TIM1:0xF3].....	273
12.6.41 TIMx 中断标志寄存器 1 (TIMx_ISR1) [TIM0:0xB4/TIM1:0xF4].....	274
12.6.42 TIMx 中断标志寄存器 2 (TIMx_ISR2) [TIM0:0xB9/TIM1:0xF9].....	275
12.6.43 TIMx 计数器的低 8 位 (TIMx_CNTH) [TIM0:0xB5/TIM1:0xF5].....	276
12.6.44 TIMx 计数器的高 8 位 (TIMx_CNTH) [TIM0:0xB6/TIM1:0xF6].....	276
13 CORDIC 协处理器.....	277
13.1 概述.....	277
13.2 功能描述.....	278
13.2.1 输入数据格式.....	278
13.2.2 结果数据归一化.....	278
13.2.3 CORDIC 调度.....	279
13.2.4 CORDIC 中断.....	280
13.2.5 CORDIC 精度.....	280
13.3 CORDIC 寄存器.....	281
13.3.1 JMT51 MCU 调度寄存器.....	281

13.3.2	JMT018 DSP 指令调度寄存器	284
14	通用异步收发传输器 (UART)	287
14.1	概述	287
14.2	框图	287
14.3	UART 管脚配置	288
14.4	UART 工作模式	288
14.4.1	标准 UART 工作模式 0: 同步移位寄存器	288
14.4.2	标准 UART 工作模式 1: 8 位 UART, 波特率可变	289
14.4.3	标准 UART 工作模式 2: 9 位 UART, 波特率固定	290
14.4.4	标准 UART 工作模式 3: 9 位 UART, 波特率可变	290
14.4.5	38KHz 红外调制发送	291
14.4.6	IrDA(SIR)红外通信	292
14.5	UART 多机通信	292
14.6	UART 奇偶校验	293
14.7	UART 波特率计算	293
14.8	UART 寄存器	302
14.8.1	UART0 控制寄存器(S0CON) [0x98]	302
14.8.2	UART0 数据缓冲寄存器(S0BUF) [0x99]	303
14.8.3	UART0 波特率重载寄存器低 8 位(S0RELL) [0x9A]	303
14.8.4	UART0 波特率重载寄存器高 2 位(S0RELH) [0x9B]	304
14.8.5	UART0 红外控制寄存器(S0IRCON) [0x9C]	304
14.8.6	UART1 控制寄存器(S1CON) [0xF1]	305
14.8.7	UART1 数据缓冲寄存器(S1BUF) [0xF2]	305
14.8.8	UART1 波特率重载寄存器低 8 位(S1RELL) [0xF3]	306
14.8.9	UART1 波特率重载寄存器高 2 位(S1RELH) [0xF4]	306
14.8.10	UART1 红外控制寄存器(S1IRCON) [0xF5]	306
15	I2C 接口	308
15.1	概述	308
15.2	框图	308
15.3	I2C 管脚配置	308
15.4	I2C 模块连接	308
15.5	I2C 数据格式	309

15.5.1	数据有效.....	309
15.5.2	读写控制比特.....	309
15.5.3	反馈 (ACK)	310
15.5.4	起始条件和结束条件.....	310
15.5.5	数据格式.....	310
15.5.6	时钟同步.....	311
15.5.7	仲裁.....	311
15.6	I2C 时钟产生.....	312
15.7	I2C 工作模式.....	313
15.7.1	从机模式.....	313
15.7.2	主机模式.....	315
15.8	I2C 中断.....	318
15.9	I2C 寄存器.....	319
15.9.1	I2C 控制寄存器(I2CCON) [0xE1].....	319
15.9.2	I2C 从机地址寄存器低 8 位(I2CSADDRL) [0xE2]	320
15.9.3	I2C 从机地址寄存器高 2 位(I2CSADDRH) [0xE3].....	320
15.9.4	I2C 本机地址寄存器低 8 位(I2COADDRL) [0xE4].....	321
15.9.5	I2C 本机地址寄存器高 2 位(I2COADDRH) [0xE5]	321
15.9.6	I2C 工作时钟分频控制寄存器(I2CDIV) [0xE6]	321
15.9.7	I2C 时钟 SCL 低电平时间配置寄存器(I2CDUTYL) [0xE7].....	322
15.9.8	I2C 时钟 SCL 高电平时间配置寄存器(I2CDUTYH) [0xE9]	322
15.9.9	I2C 数据 SDA 保持时间配置寄存器(I2CHOLD) [0xEA]	322
15.9.10	I2C 数据写缓冲寄存器(I2CWBUF) [0xEB].....	323
15.9.11	I2C 数据读缓冲寄存器(I2CRBUF) [0xEC]	323
15.9.12	I2C 状态寄存器(I2CSTS) [0xED]	323
15.9.13	I2C 中断标志寄存器(I2CISC) [0xEE].....	324
15.9.14	I2C 中断使能寄存器(I2CIEN) [0xEF]	325
16	串行外设接口 (SPI)	326
16.1	概述.....	326
16.2	框图.....	326
16.3	SPI 管脚配置.....	326
16.4	SPI 主/从机互连.....	327
16.4.1	4 线互连模式.....	327

16.4.2	3 线互连模式.....	327
16.5	SPI 时序设置.....	328
16.6	数据发送和接收过程.....	329
16.6.1	主机模式.....	329
16.6.2	从机模式.....	331
16.7	SPI 中断.....	332
16.8	SPI 寄存器.....	333
16.8.1	SPI 控制寄存器(SPICON) [0xC9].....	333
16.8.2	SPI 中断使能寄存器(SPIIE) [0xCA].....	334
16.8.3	SPI 中断标志寄存器(SPIIF) [0xCB].....	335
16.8.4	SPI 波特率控制寄存器(SPIBR) [0xCC].....	335
16.8.5	SPI 数据缓冲寄存器(SPIBUF) [0xCD].....	336
17	模拟/数字转换 (ADC) 和可编程增益放大器 (PGA)	337
17.1	概述.....	337
17.2	ADC 转换结果访问方式.....	338
17.3	ADC 输入端口配置.....	338
17.4	可编程增益放大器 (PGA) 说明.....	339
17.5	ADC 参考电压配置.....	341
17.6	ADC 转换功能.....	341
17.7	ADC 转换结果数据格式.....	342
17.8	ADC 转换时序.....	342
17.9	模拟看门狗.....	343
17.10	外部触发事件.....	344
17.11	特殊通道序列管理.....	346
17.12	ADC 工作模式.....	347
17.12.1	单次转换+非扫描模式.....	347
17.12.2	单次转换+扫描模式.....	348
17.12.3	连续转换+非扫描模式.....	350
17.12.4	连续转换+扫描模式.....	351
17.13	双ADC 协同工作模式.....	353
17.13.1	独立模式.....	353
17.13.2	普通同步模式.....	353

17.13.3	特殊同步模式.....	354
17.13.4	普通交叉模式.....	355
17.13.5	特殊交替触发模式.....	355
17.13.6	普通同步+特殊同步模式.....	356
17.13.7	普通同步+特殊交替触发模式.....	357
17.13.8	普通交叉+特殊同步模式.....	358
17.14	中断处理.....	359
17.15	ADC 寄存器.....	360
17.15.1	JMT51 访问 ADC 寄存器.....	360
17.15.2	JMT018 访问 ADC 寄存器.....	391
18	模拟比较器 (CMP)	393
18.1	模拟比较器概述.....	393
18.2	低压检测.....	394
18.2.1	低压检测保护程序示例.....	395
18.3	模拟比较器寄存器.....	396
18.3.1	模拟比较器使能寄存器(CMPEN) [0x91].....	396
18.3.2	模拟比较器中断寄存器(CMPINT) [0x92].....	397
18.3.3	模拟比较器比较结果寄存器(CMPFLAG) [0x93].....	397
18.3.4	模拟比较器滤波选择寄存器(CMPFT) [0x94].....	398
18.3.5	模拟比较器 VDD 分压寄存器(CMPVDD) [0x95].....	398
18.3.6	模拟比较器参考电压选择寄存器(CMPREF) [0x96].....	398
19	通用输入/输出 (GPIO)	400
19.1	管脚设置.....	400
19.2	管脚功能复用.....	403
19.3	PWM 管脚特殊复用.....	407
19.3.1	PWM 输出复用.....	407
19.3.2	PWM 刹车复用.....	407
19.4	管脚数据寄存器读写.....	408
19.5	外设管脚复用配置.....	408
19.6	管脚滤波.....	415
19.7	INT0 中断.....	415
19.8	INT1 中断.....	418

19.9	PA0 唤醒.....	421
19.10	GPIO 寄存器.....	421
19.10.1	PA0 数字 I/O 控制寄存器(PA0CTRL) [0x99].....	424
19.10.2	PA1 数字 I/O 控制寄存器(PA1CTRL) [0x9A].....	425
19.10.3	PA2 数字 I/O 控制寄存器(PA2CTRL) [0x9B].....	426
19.10.4	PA3 数字 I/O 控制寄存器(PA3CTRL) [0x9C].....	427
19.10.5	PA4 数字 I/O 控制寄存器(PA4CTRL) [0x9D].....	428
19.10.6	PA5 数字 I/O 控制寄存器(PA5CTRL) [0x9E].....	428
19.10.7	PA6 数字 I/O 控制寄存器(PA6CTRL) [0x9F].....	429
19.10.8	PA7 数字 I/O 控制寄存器(PA7CTRL) [0xC3].....	430
19.10.9	PB0 数字 I/O 控制寄存器(PB0CTRL) [0xA1].....	431
19.10.10	PB1 数字 I/O 控制寄存器(PB1CTRL) [0xA2].....	432
19.10.11	PB2 数字 I/O 控制寄存器(PB2CTRL) [0xA3].....	432
19.10.12	PB3 数字 I/O 控制寄存器(PB3CTRL) [0xA4].....	433
19.10.13	PB4 数字 I/O 控制寄存器(PB4CTRL) [0xA5].....	434
19.10.14	PB5 数字 I/O 控制寄存器(PB5CTRL) [0xA6].....	435
19.10.15	PB6 数字 I/O 控制寄存器(PB6CTRL) [0xA7].....	436
19.10.16	PB7 数字 I/O 控制寄存器(PB7CTRL) [0xC4].....	436
19.10.17	PC0 数字 I/O 控制寄存器(PC0CTRL) [0xA9].....	437
19.10.18	PC1 数字 I/O 控制寄存器(PC1CTRL) [0xAA].....	438
19.10.19	PC2 数字 I/O 控制寄存器(PC2CTRL) [0xAB].....	439
19.10.20	PC3 数字 I/O 控制寄存器(PC3CTRL) [0xAC].....	440
19.10.21	PC4 数字 I/O 控制寄存器(PC4CTRL) [0xAD].....	440
19.10.22	PC5 数字 I/O 控制寄存器(PC5CTRL) [0xAE].....	441
19.10.23	PC6 数字 I/O 控制寄存器(PC6CTRL) [0xAF].....	442
19.10.24	PC7 数字 I/O 控制寄存器(PC7CTRL) [0xC5].....	443
19.10.25	PD0 数字 I/O 控制寄存器(PD0CTRL) [0xB1].....	444
19.10.26	PD1 数字 I/O 控制寄存器(PD1CTRL) [0xB2].....	444
19.10.27	PD2 数字 I/O 控制寄存器(PD2CTRL) [0xB3].....	445
19.10.28	PD3 数字 I/O 控制寄存器(PD3CTRL) [0xB4].....	446
19.10.29	PD4 数字 I/O 控制寄存器(PD4CTRL) [0xB5].....	447
19.10.30	PD5 数字 I/O 控制寄存器(PD5CTRL) [0xB6].....	448
19.10.31	PD6 数字 I/O 控制寄存器(PD6CTRL) [0xB7].....	448
19.10.32	PD7 数字 I/O 控制寄存器(PD7CTRL) [0xC6].....	449
19.10.33	PE0 数字 I/O 控制寄存器(PE0CTRL) [0xB9].....	450

19.10.34 PE1 数字 I/O 控制寄存器(PE1CTRL) [0xBA]	451
19.10.35 PE2 数字 I/O 控制寄存器(PE2CTRL) [0xBB]	452
19.10.36 PE3 数字 I/O 控制寄存器(PE3CTRL) [0xBC]	452
19.10.37 PE4 数字 I/O 控制寄存器(PE4CTRL) [0xBD]	453
19.10.38 PE5 数字 I/O 控制寄存器(PE5CTRL) [0xBE].....	454
19.10.39 PE6 数字 I/O 控制寄存器(PE6CTRL) [0xBF].....	455
19.10.40 PE7 数字 I/O 控制寄存器(PE7CTRL) [0xC7]	456
19.10.41 PG0 数字 I/O 控制寄存器(PG0CTRL) [0xC1]	456
19.10.42 PG1 数字 I/O 控制寄存器(PG1CTRL) [0xC2]	457
19.10.43 PA 转换速率寄存器(PASR) [0xC9].....	458
19.10.44 PB 转换速率寄存器(PBSR) [0xCA]	459
19.10.45 PC 转换速率寄存器(PCSR) [0xCB].....	460
19.10.46 PD 转换速率寄存器(PDSR) [0xCC]	461
19.10.47 PE 转换速率寄存器(PESR) [0xCD].....	461
19.10.48 PG 转换速率寄存器(PGSR) [0xCE]	462
19.10.49 PA 驱动能力寄存器(PADR) [0xD1].....	463
19.10.50 PB 驱动能力寄存器(PBDR) [0xD2]	464
19.10.51 PC 驱动能力寄存器(PCDR) [0xD3]	464
19.10.52 PD 驱动能力寄存器(PDDR) [0xD4].....	465
19.10.53 PE 驱动能力寄存器(PEDR) [0xD5].....	466
19.10.54 PG 驱动能力寄存器(PGDR) [0xD6].....	467
19.10.55 模拟管脚使能寄存器 0(AIOEN0) [0xCF]	467
19.10.56 模拟管脚使能寄存器 1(AIOEN1) [0xD7]	469
19.10.57 复位管脚滤波选择寄存器(NRSTFQS) [0xD9]	470
19.10.58 唤醒使能寄存器(WKUPEN) [0xDB].....	470
19.10.59 INT0 上升沿中断使能寄存器(INT0PE) [0xDC]	471
19.10.60 INT0 下降沿中断使能寄存器(INT0NE) [0xDD]	471
19.10.61 INT0 中断使能寄存器(INT0EN) [0xDE].....	472
19.10.62 INT0 中断控制寄存器(INT0CON) [0xDF].....	473
19.10.63 INT1 上升沿中断使能寄存器(INT1PE) [0xE1]	474
19.10.64 INT1 下降沿中断使能寄存器(INT1NE) [0xE2].....	475
19.10.65 INT1 中断使能寄存器(INT1EN) [0xE3].....	476
19.10.66 INT1 中断控制寄存器(INT1CON) [0xE4]	477
19.10.67 INT1 中断管脚选择寄存器 0(INT1SEL0) [0xE5].....	478
19.10.68 INT1 中断管脚选择寄存器 1(INT1SEL1) [0xE6].....	479

19.10.69	INT1 中断管脚选择寄存器 2(INT1SEL2) [0xE7].....	480
19.10.70	INT1 中断管脚选择寄存器 3(INT1SEL3) [0xE9].....	480
19.10.71	PWM 刹车选择寄存器(PWMBKS) [0xEA]	481
19.10.72	PWM 特殊复用输出分组选择寄存器(PWMOSEL) [0xF1]	481
19.10.73	PWM 特殊复用输出选择寄存器 0(PWMOSEL0) [0xF2]	482
19.10.74	PWM 特殊复用输出选择寄存器 1(PWMOSEL1) [0xF3]	483
19.10.75	PWM 特殊复用输出选择寄存器 2(PWMOSEL2) [0xF4]	483
19.10.76	PWM 特殊复用输出使能寄存器 0(PWMOMUX0) [0xF5]	484
19.10.77	PWM 特殊复用输出使能寄存器 1(PWMOMUX1) [0xF6]	485
19.10.78	PWM 特殊复用输出使能寄存器 2(PWMOMUX2) [0xF7]	486
19.10.79	PA 输入缓冲使能寄存器(PAIE) [0xF9]	487
19.10.80	PB 输入缓冲使能寄存器(PBIE) [0xFA]	488
19.10.81	PC 输入缓冲使能寄存器(PCIE) [0xFB]	489
19.10.82	PD 输入缓冲使能寄存器(PDIE) [0xFC].....	489
19.10.83	PE 输入缓冲使能寄存器(PEIE) [0xFD]	490
19.10.84	PG 输入缓冲使能寄存器(PGIE) [0xFE].....	491
19.10.85	PA 数据寄存器(P0) [0x80].....	492
19.10.86	PB 数据寄存器(P1) [0x90]	492
19.10.87	PC 数据寄存器(P2) [0xA0].....	492
19.10.88	PD 数据寄存器(P3) [0xB0].....	492
19.10.89	PE 数据寄存器(P4) [0xC8].....	492
19.10.90	PG 数据寄存器(P6) [0x98]	492
20	电气特性	493
20.1	极限参数.....	493
20.2	工作条件.....	493
20.2.1	芯片供电.....	493
20.2.2	功耗特性.....	493
20.2.3	I/O 特性	494
20.2.4	REGC 引脚特性	495
20.2.5	NRST 引脚特性.....	495
20.2.6	上电/掉电条件.....	495
20.2.7	时钟特性.....	496
20.2.8	通信接口.....	498
20.2.9	FLASH 特性	501

20.2.10	ADC 特性	501
20.2.11	PGA 特性.....	502
20.2.12	CMP 特性	502
20.2.13	内部参考电压特性.....	503
20.2.14	电气敏感性.....	503
21	封装特性	504
21.1	LQFP32L.....	504
21.2	LQFP48L.....	505
21.3	QFN32L.....	507
21.4	TSSOP28L.....	508

IRONCHIP

1 产品概述

1.1 功能描述

JMT1808R 是一款内置 MCU 和 DSP 的双核 SOC 芯片。JMT1808R 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。JMT1808R 集成了：32K 字节 Flash、2.5K 字节 SRAM、1 个高性能 CORDIC（坐标旋转矢量运算）单元、2 个高性能 16 位 PWM 模块（每个 PWM 模块具有 3 路互补加 1 路独立 PWM）、2 个独立的 12 通道 10 位 ADC、2 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 RTC、1 个 SPI、1 个 I2C、2 个 UART 通信接口。

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

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:
 - 512x16bit SRAM（PM, 和 JMT51 核共享）
 - 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
- 支持 8~24MHz 外部晶体振荡器时钟输入

外设特性:

- 42 个双向 I/O
 - 1 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 2.5K 字节数据存储器(SRAM)
 - 256x8bit 内部数据存储器
 - 1024x8bit DM2
 - 512x16bit PM(MCU 和 DSP 共享)
 - 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

- 2 个 PWM 模块
 - 每个 PWM 模块具有 3 路互补加 1 路独立 PWM (可作为 7 路独立 PWM 使用)
 - 每个 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
 - 2 个独立 12 通道 10 位 ADC
 - 支持两路独立 ADC 模式
 - 支持双 ADC 模式
 - 支持各 12 通道输入
 - 支持模拟看门狗
 - 支持多达 4 个通道结果寄存器
 - 支持外部管脚/PWM 触发
 - 采样时间延时可编程
 - 参考电压: 外部参考电压、内置参考电压、VDD 可选
 - 2 个高倍数可编程增益放大器 (PGA)
 - 支持 1/4、1/2、1、2、4、8、16、32 放大倍数可配置
 - 支持同相、反相极性
 - 2 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 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
 - 3 个 Timer、1 个 WDT、1 个 RTC
 - 1 个模拟比较器
 - 支持低压检测功能
 - 比较器输入可选择: PC0/PGA1 输出/VDDA
 - 比较器参考电压可选择
 - 上电复位、掉电复位、低压复位
- 低功耗特性:**
- 芯片工作电压
 - V_{DD} : 2.45V~5.5V
 - 支持 5 种工作模式
 - Normal、Idle、Stop、Sleep 和 Deep Sleep
- 封装:**
- LQFP48L、LQFP32L
 - QFN32L
 - TSSOP28L

1.3 框图

JMT1808R 框图如 Figure 1 所示：

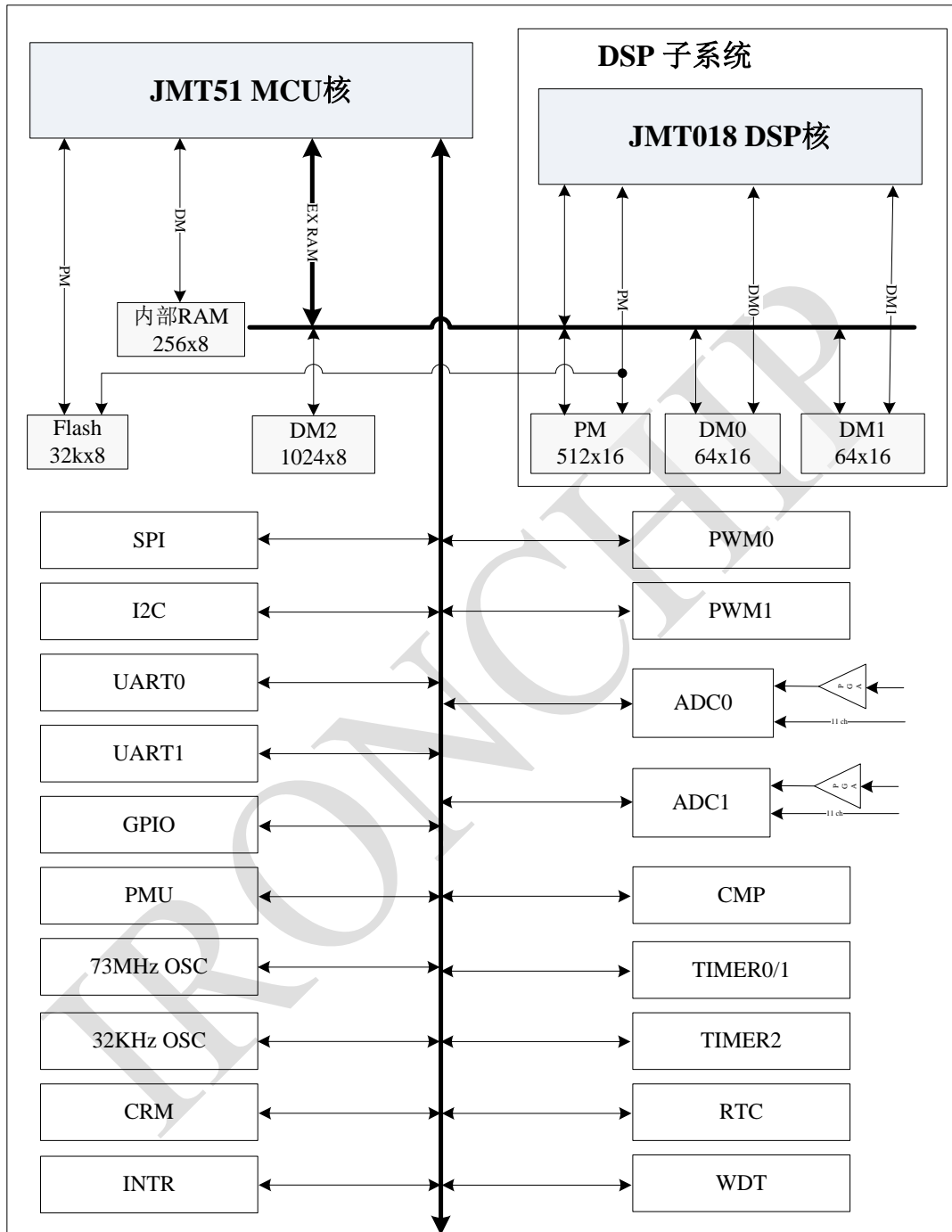


Figure 1 JMT1808R 结构框图

1.4 应用范围

JMT1808R 可应用于电机控制、太阳能逆变器、UPS 电源控制器等产品中。

JMT1808R 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、空气净化器等具有 PWM、SPWM、FOC-SVPWM 等控制模式的产品中。

JMT1808R 支持 SPWM、SVPWM 逆变器。

JMT1808R 支持 SPWM、SVPWM UPS。

JMT1808R 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，两路独立 ADC 完成电流、电压采样，每个 ADC 均配置有一路独立的 PGA 进行放大。

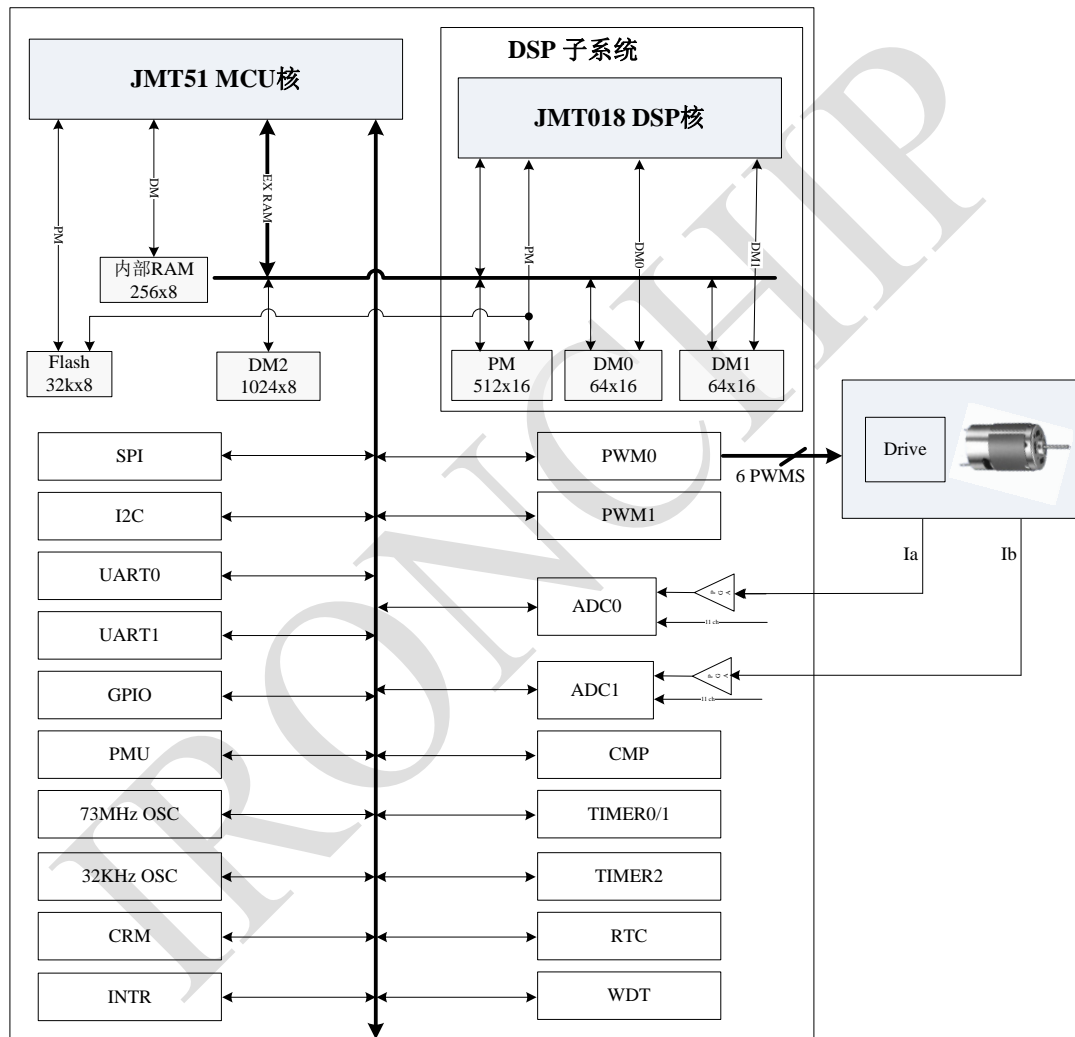


Figure 2 JMT1808R 应用连接图

1.5 引脚配置

1.5.1 LQFP48L 引脚图

JMT1808R 芯片 LQFP48L 封装如 Figure 3 所示：

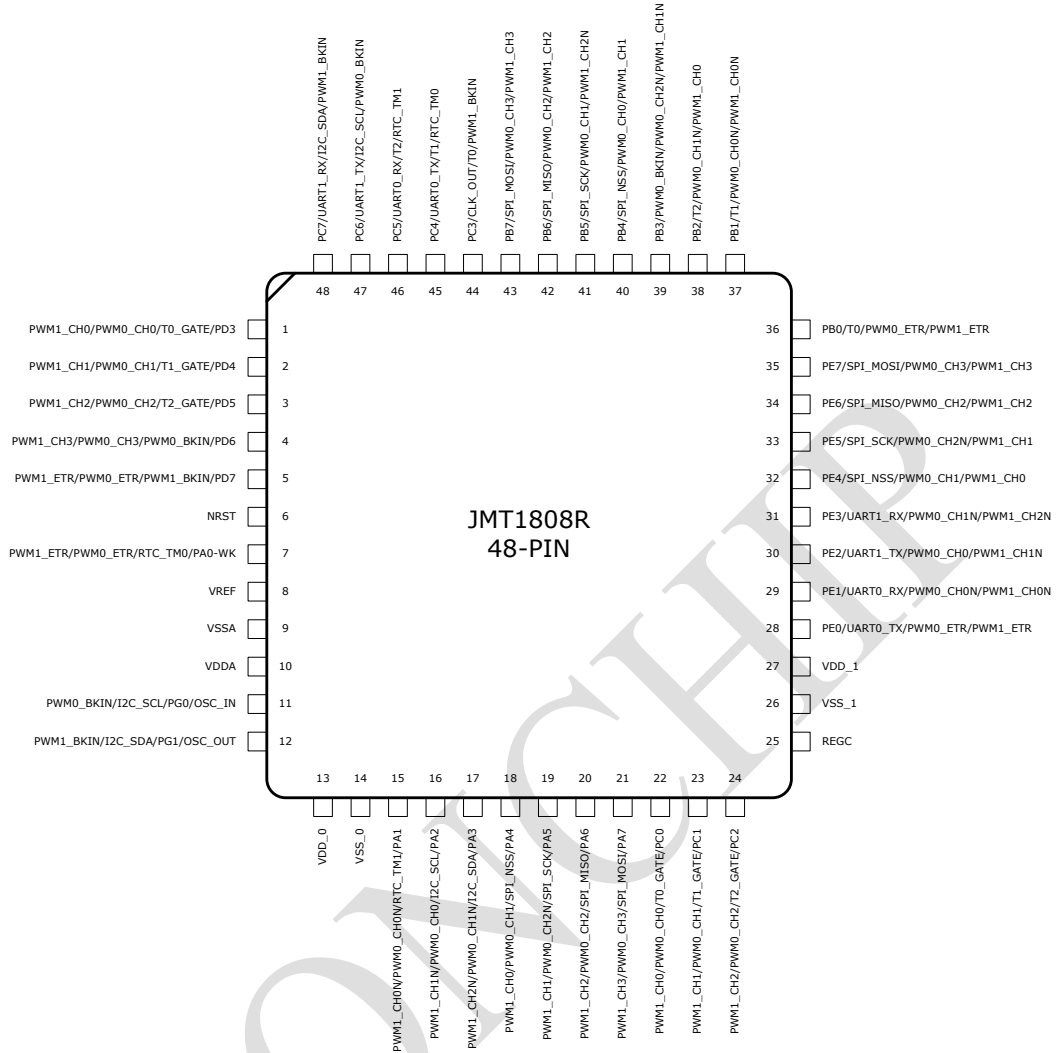


Figure 3 JMT1808R 芯片 LQFP48L 封装

1.5.2 LQFP32L 引脚图

JMT1808R 芯片 LQFP32L 封装如 Figure 4 所示:

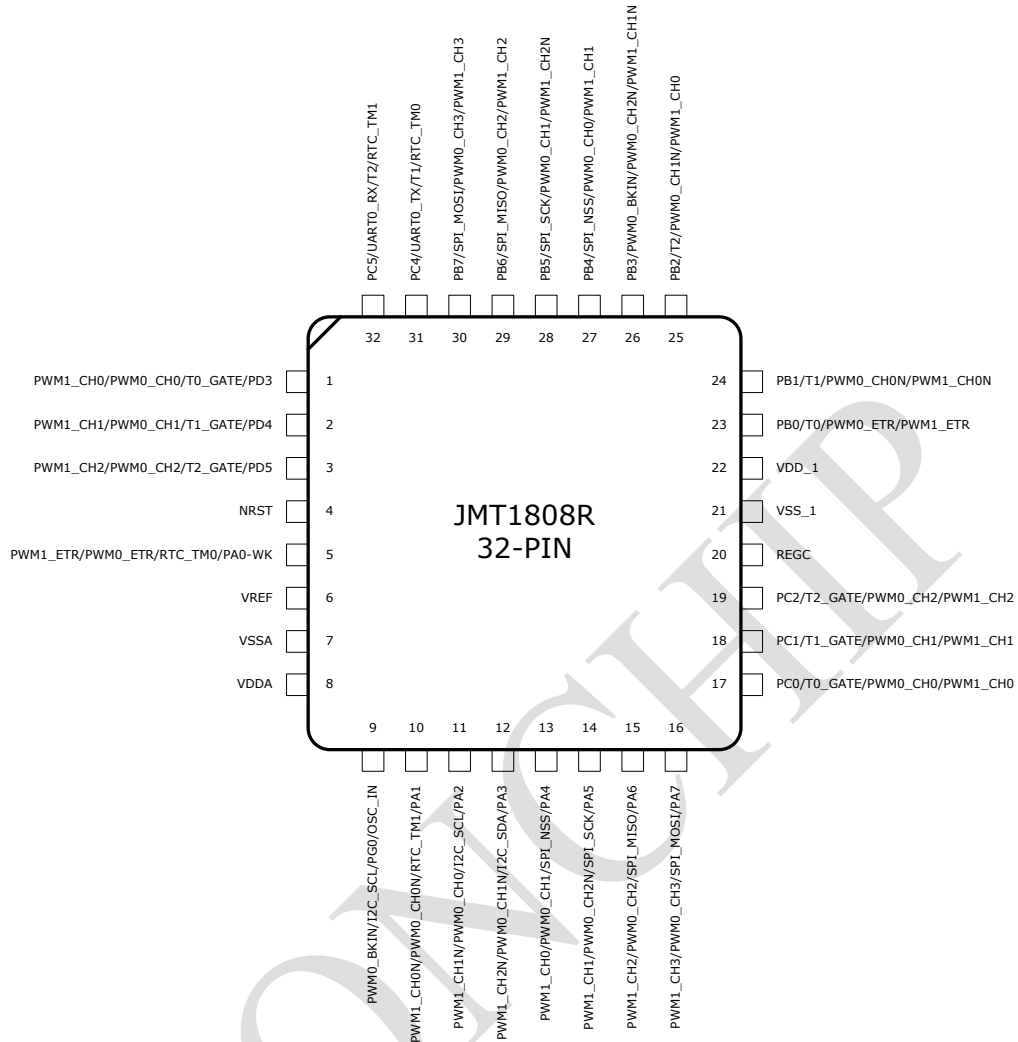


Figure 4 JMT1808R 芯片 LQFP32L 封装

1.5.3 QFN32L 引脚图

JMT1808R 芯片 QFN32L 封装如 Figure 4 所示:

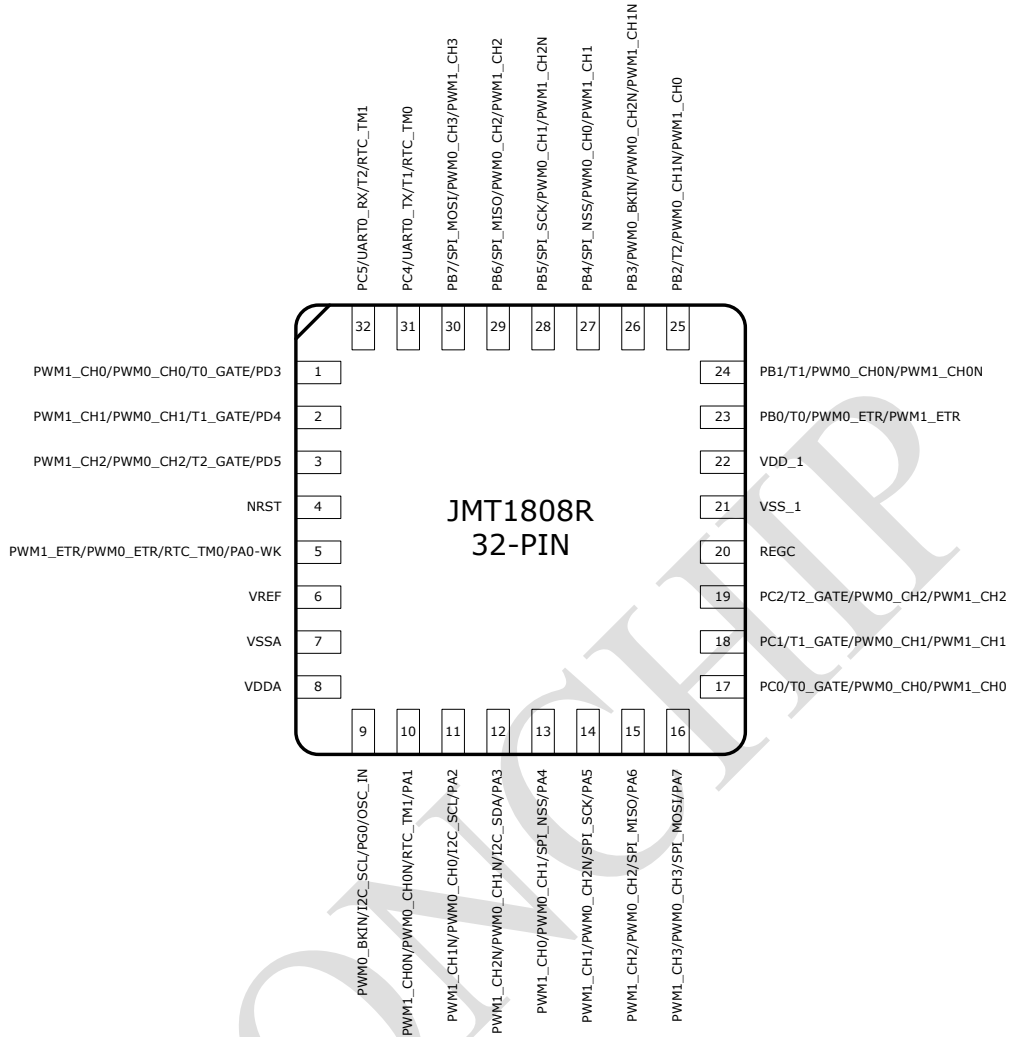


Figure 5 JMT1808R 芯片 QFN32L 封装

1.5.4 TSSOP28L 引脚图

JMT1808R 芯片 TSSOP28L 封装如 Figure 6 所示:

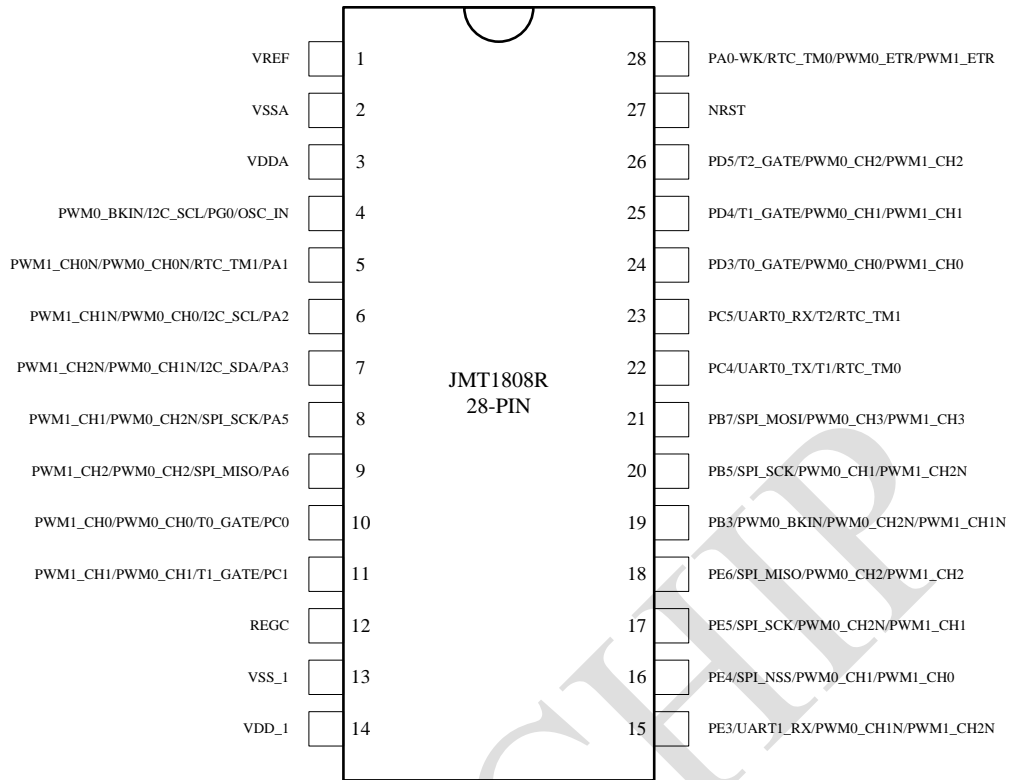


Figure 6 JMT1808R 芯片 TSSOP28L 封装

1.5.5 引脚说明

JMT1808R 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~PC2 和 PD3~PD7 为数字、模拟复用 I/O；PB0~PB7、PC3~PC7、PE0~PE7 和 PG0~PG1 为纯数字 I/O。管脚的功能复用见 Table 1:

Table 1 JMT1808R 引脚说明

序号 (LQFP48L)	序号 (LQFP32L /QFN3L)	序号 (TSSOP28L)	引脚名	功能 0 (默认)	功能 1	功能 2	功能 3
1	1	24	PD3	GPIO	T0_GATE	PWM0_CH0	PWM1_CH0
2	2	25	PD4	GPIO	T1_GATE	PWM0_CH1	PWM1_CH1
3	3	26	PD5	GPIO	T2_GATE	PWM0_CH2	PWM1_CH2
4			PD6	GPIO	PWM0_BKIN	PWM0_CH3	PWM1_CH3
5			PD7	GPIO	PWM1_BKIN	PWM0_ETR	PWM1_ETR
6	4	27	NRST				
7	5	28	PA0	GPIO -WK	RTC_TM0	PWM0_ETR	PWM1_ETR
8	6	1	VREF				

序号 (LQFP48L)	序号 (LQFP32L /QFN3L)	序号 (TSSOP28L)	引脚名	功能 0 (默认)	功能 1	功能 2	功能 3
9	7	2	VSSA				
10	8	3	VDDA				
11	9	4	OSC_IN	OSC_IN	GPIO	I2C_SCL	PWM0_BKIN
12			OSC_OUT	OSC_OUT	GPIO	I2C_SDA	PWM1_BKIN
13			VDD_0				
14			VSS_0				
15	10	5	PA1	GPIO	RTC_TM1	PWM0_CH0N	PWM1_CH0N
16	11	6	PA2	GPIO	I2C_SCL	PWM0_CH0	PWM1_CH1N
17	12	7	PA3	GPIO	I2C_SDA	PWM0_CH1N	PWM1_CH2N
18	13		PA4	GPIO	SPI_NSS	PWM0_CH1	PWM1_CH0
19	14	8	PA5	GPIO	SPI_SCK	PWM0_CH2N	PWM1_CH1
20	15	9	PA6	GPIO	SPI_MISO	PWM0_CH2	PWM1_CH2
21	16		PA7	GPIO	SPI_MOSI	PWM0_CH3	PWM1_CH3
22	17	10	PC0	GPIO	T0_GATE	PWM0_CH0	PWM1_CH0
23	18	11	PC1	GPIO	T1_GATE	PWM0_CH1	PWM1_CH1
24	19		PC2	GPIO	T2_GATE	PWM0_CH2	PWM1_CH2
25	20	12	REGC				
26	21	13	VSS_1				
27	22	14	VDD_1				
28			PE0	GPIO	UART0_TX	PWM0_ETR	PWM1_ETR
29			PE1	GPIO	UART0_RX	PWM0_CH0N	PWM1_CH0N
30			PE2	GPIO	UART1_TX	PWM0_CH0	PWM1_CH1N
31		15	PE3	GPIO	UART1_RX	PWM0_CH1N	PWM1_CH2N
32		16	PE4	GPIO	SPI_NSS	PWM0_CH1	PWM1_CH0
33		17	PE5	GPIO	SPI_SCK	PWM0_CH2N	PWM1_CH1
34		18	PE6	GPIO	SPI_MISO	PWM0_CH2	PWM1_CH2
35			PE7	GPIO	SPI_MOSI	PWM0_CH3	PWM1_CH3
36	23		PB0	GPIO	T0	PWM0_ETR	PWM1_ETR
37	24		PB1	GPIO	T1	PWM0_CH0N	PWM1_CH0N

序号 (LQFP48L)	序号 (LQFP32L /QFN3L)	序号 (TSSOP28L)	引脚名	功能 0 (默认)	功能 1	功能 2	功能 3
38	25		PB2	GPIO	T2	PWM0_CH1N	PWM1_CH0
39	26	19	PB3	GPIO	PWM0_BKIN	PWM0_CH2N	PWM1_CH1N
40	27		PB4	GPIO	SPI_NSS	PWM0_CH0	PWM1_CH1
41	28	20	PB5	GPIO	SPI_SCK	PWM0_CH1	PWM1_CH2N
42	29		PB6	GPIO	SPI_MISO	PWM0_CH2	PWM1_CH2
43	30	21	PB7	GPIO	SPI_MOSI	PWM0_CH3	PWM1_CH3
44			PC3	GPIO	CLK_OUT	T0	PWM1_BKIN
45	31	22	PC4	GPIO	UART0_TX	T1	RTC_TM0
46	32	23	PC5	GPIO	UART0_RX	T2	RTC_TM1
47			PC6	GPIO	UART1_TX	I2C_SCL	PWM0_BKIN
48			PC7	GPIO	UART1_RX	I2C_SDA	PWM1_BKIN

2 电源管理 (PMU)

2.1 概述

JMT1808R 的工作电压范围为 2.45V~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)、实时时钟 (RTC) 以及一些控制电路供电。Figure 7 为 JMT1808R 供电示意图。

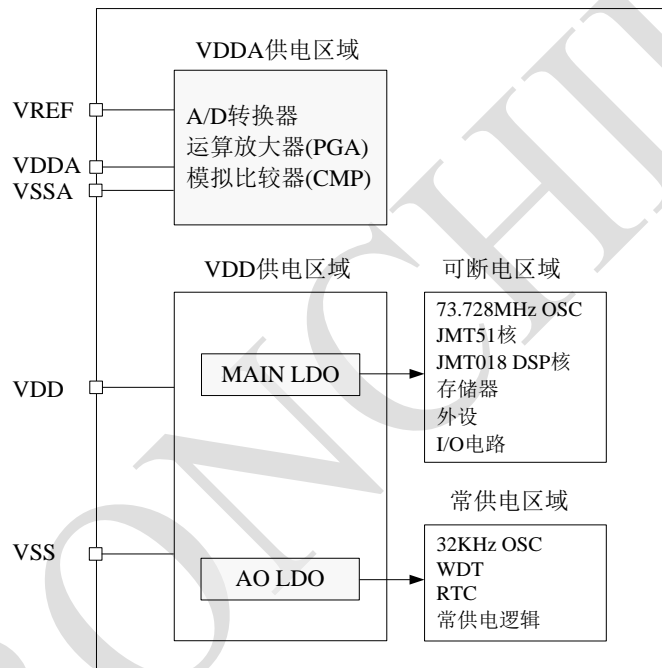


Figure 7 JMT1808R 芯片供电示意图

Figure 7 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、RTC 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¹。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。

JMT1808R 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），实时时钟（RTC）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、RTC 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，内部参考电压等

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。

¹ 建议用户在进入低功耗之前关闭外部 22.1184MHz OSC，外部 22.1184MHz OSC 默认为关闭模式

2.2.2 外设时钟门控

JMT1808R 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 3.1.4 章节门控时钟寄存器的说明。

2.2.3 关闭模拟模块

JMT1808R 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，内部 2.5V 参考电压。具体参见各自模块章节说明。若 ADC、PGA 和 CMP 模拟模块均不工作，则可以关闭内部参考电压。

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。

JMT1808R 四种低功耗模式与正常运行模式的状态转换图如 Figure 8 所示。

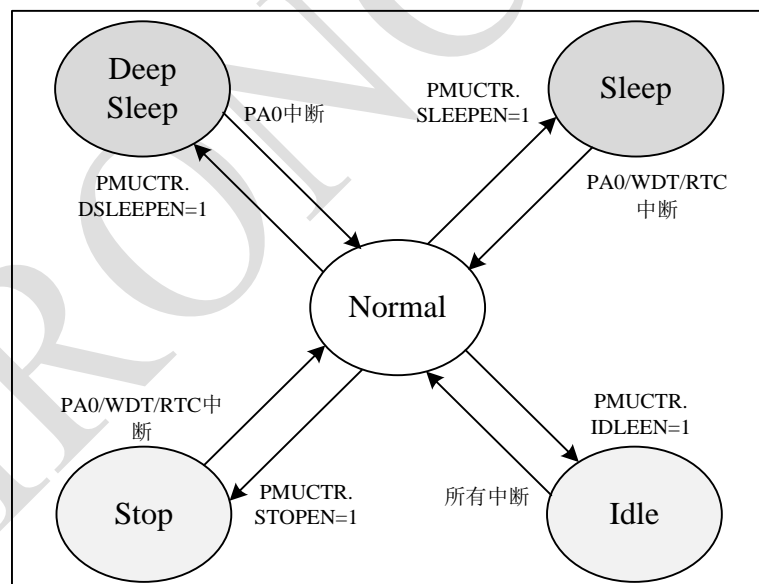


Figure 8 JMT1808R 工作模式切换场景图

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，建议编写程序时，在配置进入 Idle 后紧跟插入一条 NOP 指令。

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，建议编写程序时，在配置进入 Stop 后紧跟插入一条 NOP 指令。

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），实时时钟（RTC）正常工作。

当系统时钟源为 32KHz OSC 时，不支持进入 Stop 模式。

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
- 实时时钟（RTC）中断唤醒
- 外部管脚：PA0 作为唤醒管脚使用

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、RTC 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
- RTC 中断唤醒
- 外部管脚：PA0 作为唤醒管脚使用

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、RTC 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管脚：外部 PA0 作为唤醒管脚使用。

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：

- JMT1808R 有三个时钟，其中 73.728MHz OSC 和 22.1184MHz 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。
- 配置寄存器 OSCPD.OSC22MEN，控制 22.1184MHz OSC 工作模式。配置为 1 时正常工作，配置为 0 时进入低功耗模式（时钟关闭）。
- 对某个 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，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

- 芯片包含三个振荡器：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 的片内低频振荡器和一个输出频率为 22.1184MHz 的外部振荡器。
- 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 时钟或由外部振荡器产生的 22.1184MHz 时钟经过分频、选择和门控得到。

3.1.2 框图

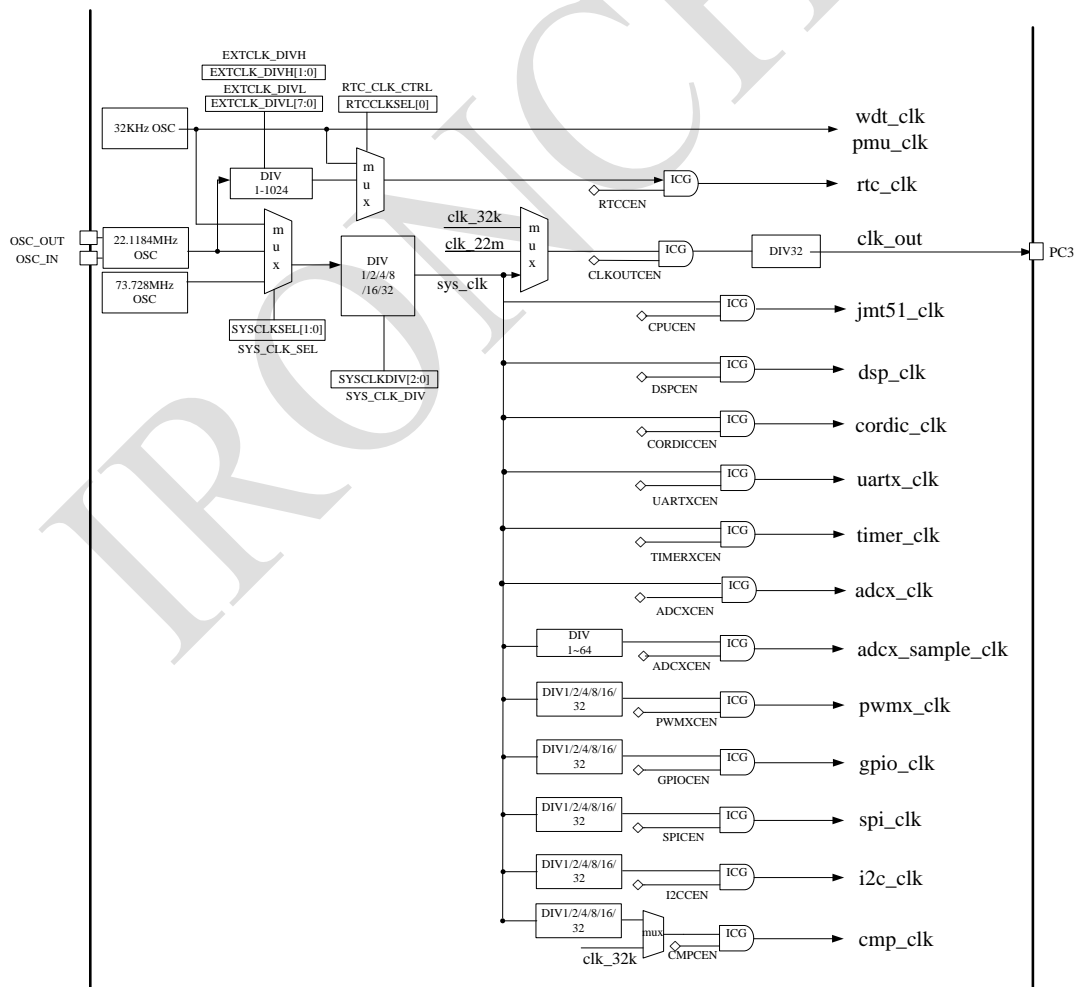


Figure 9 JMT1808R 时钟树

3.1.3 功能描述

3.1.3.1 时钟源

JMT1808R 时钟模块包含三个振荡器：输出频率为 32 KHz 的片内低频振荡器、输出频率为 73.728MHz 的片内高频振荡器和输出频率为 22.1184MHz 的外部振荡器。

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或者经过分频后作为 RTC 的时钟。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°C）。

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。

当选择片外振荡器的时钟作为系统时钟时，若使用时钟监控系统监控外部振荡器是否停振，需要打开片内高频振荡器。在时钟监控系统打开的情况下，当外部振荡器出现停振故障时，硬件会自动将系统时钟切换到片内高频振荡器产生的时钟。

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、WDT 和 RTC 模块的时钟，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°C）。

3.1.3.1.3 片外 22.1184MHz 晶体振荡器

22.1184MHz 的外部晶体振荡器可为系统提供更为精确的主时钟，相关的硬件配置可参考 Figure 10，进一步信息可参考电气特性部分的 20.2.7.3 章节。

外部晶体振荡器的开关受寄存器 OSCPD 的 OSC22MEN 位和寄存器 OSCEN 的 OSCEN 位控制，当 OSCPD 的 OSC22MEN 位和 OSCEN 寄存器的 OSCEN 位同时配置为 1 时，外部振荡器为芯片正常提供 22.1184MHz 的时钟；当寄存器 OSCPD 的 OSC22MEN 位配置为 0 时，外部振荡器产生的 22.1184MHz 时钟被门控掉；当 OSCEN 寄存器的 OSCEN 配置为 0 时，外部振荡器停止工作。

寄存器 CLK_READY 的 OSC22MRDY 位指示外部振荡器的输出时钟是否稳定。启动外

部振荡器后，当 22.1184MHz 时钟稳定时，寄存器 CLK_READY 的 OSC22MRDY 位被硬件置 1，此时时钟才能被使用。

可以通过配置寄存器 OSC_STABL 和寄存 OSC_STABH 决定等待外部振荡器时钟稳定的时间。

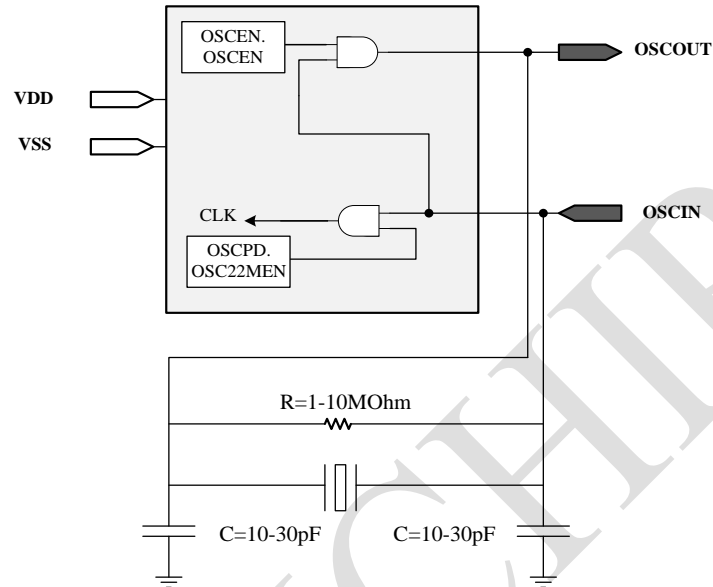


Figure 10 外部晶体振荡器连接示意图

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 寄存器选择以下三个时钟源：

- 片内 32KHz 振荡器输出的时钟。
- 片外振荡器输出的 22.1184MHz 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，可以配置寄存器 SYS_CLK_SEL 选择片外振荡器输出的时钟，步骤如下：

- 配置寄存器 OSCPD 的 OSC22MEN 为 1 并且配置寄存器 OSCEN 的 OSCEN 位为 1，使能外部振荡器时钟。
- 读取寄存器 CLK_READY，检查 CLK_READY 的 OSC22MRDY 位，若 OSC22MRDY 为 0，不能切换时钟；若 OSC22MRDY 为 1，表示外部 22.1184MHz 时钟稳定，此时可以配置 SYS_CLK_SEL 为 0x01，时钟切换至外部 22.1184MHz 时钟。

可以配置寄存器 SYS_CLK_SEL 为 0x02 选择片内 32KHz 振荡器作为系统时钟源。

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 1/2/4/8/16/32 分频。

分频后即得到 sys_clk。

3.1.3.3 时钟监控系统 (CMS)

时钟监控系统 (CMS) 监控外部振荡器是否正常工作，通过配置寄存器 OSCEN 的 CMSEN 位为 1 激活 CMS，一旦 CMS 被激活，CMS 将在片外振荡器输出时钟稳定后（寄存器 CLK_READY 的 OSC22MRDY 位为 1）被使能，并在片外振荡器时钟关闭后关闭。当外部振荡器发生故障时，CMS 做如下处理：

- 发出中断 CMSINT，允许软件完成营救操作；
- 向高级定时器 (PWM) 模块发送刹车信号；
- 将 RTC 模块的时钟由片外时钟切换至片内 32KHz 振荡器产生的时钟；
- 将系统时钟自动切换至片内 73.728MHz 振荡器产生的时钟；
- 将 OSCPД 寄存器的 OSC22MEN 清 0，门控掉片外振荡器时钟。

需要注意的是：只有当系统时钟由外部振荡器提供时，CMS 才会起作用，否则，即使激活了 CMS，当外部振荡器停振时 CMS 也不会起作用。

3.1.3.4 时钟输出

JMT1808R 允许输出时钟信号到外部 PC3 引脚，产生过程如下：

首先，做时钟选择，可通过配置寄存器 CLKOUTCTRL 选择以下三个时钟：

- 片内 32KHz 振荡器输出的时钟。
- 片外振荡器输出的 22.1184MHz 时钟。
- 系统时钟 sys_clk。

复位时，默认选系统时钟 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。

3.1.3.5 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 jmt51_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。

3.1.3.6 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`。

3.1.3.7 外设时钟

3.1.3.7.1 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。

3.1.3.7.2 AD 采样时钟和 AD 控制模块时钟

`ADx(x=0,1)`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_EN2` 的 `ADC0CEN` 位配置为 1, 打开 `AD0` 采样时钟, 配置为 0, 关闭 `AD0` 采样时钟。
- 将寄存器 `CLK_EN2` 的 `ADC1CEN` 位配置为 1, 打开 `AD1` 采样时钟, 配置为 0, 关闭 `AD1` 采样时钟。

`AD` 控制模块 `ADC0` 的时钟 `adc0_clk` 由系统时钟 `sys_clk` 经过门控得到, 通过配置 `CLK_EN2.ADC0CEN` 为 0, 可关闭 `adc0_clk`, 将 `CLK_EN2.ADC0CEN` 配置为 1, 打开 `adc0_clk`。

`AD` 控制模块 `ADC1` 的时钟 `adc1_clk` 由系统时钟 `sys_clk` 经过门控得到, 通过配置 `CLK_EN2.ADC1CEN` 为 0, 可关闭 `adc1_clk`, 将 `CLK_EN2.ADC1CEN` 配置为 1, 打开 `adc1_clk`。

3.1.3.7.1 CORDIC 时钟

`CORDIC` 模块时钟 `cordic_clk` 由 `sys_clk` 经过门控得到, 通过配置 `CLK_EN1.CORDICCEN` 为 0, 关闭 `cordic_clk`, 将 `CLK_EN1.CORDICCEN` 配置为 1, 打开 `cordic_clk`。

3.1.3.7.2 UART 时钟

`UART0` 时钟 `uart0_clk` 由系统时钟 `sys_clk` 经过门控得到, 通过配置 `CLK_EN1.UART0CEN` 为 0, 关闭 `uart0_clk`, 将 `CLK_EN1.UART0CEN` 配置为 1, 打开 `uart0_clk`。

`UART1` 时钟 `uart1_clk` 由系统时钟 `sys_clk` 经过门控得到, 通过配置 `CLK_EN1.UART1CEN` 为 0, 关闭 `uart1_clk`, 将 `CLK_EN1.UART1CEN` 配置为 1, 打开 `uart1_clk`。

3.1.3.7.3 TIMER 时钟

`TIMER` 时钟 `timer_clk` 由 `sys_clk` 经过门控得到, 通过配置 `CLK_EN1.TIMERCEN` 为 0, 关闭 `timer_clk`, 将 `CLK_EN1.TIMERCEN` 配置为 1, 打开 `timer_clk`。

3.1.3.7.4 PWM 模块时钟

`PWMx(x=0, 1)` 模块的最高工作时钟为 73.728MHz, 默认工作时钟为 9.19MHz, `PWMx`

模块的工作时钟产生过程如下：

首先，做时钟分频：

- 将 sys_clk 连接至分频器。
- PWM0 分频器可通过配置寄存器 PWM0_CLK_DIV 得到 sys_clk 时钟的 1/2/4/8/16/32 分频。
- PWM1 分频器可通过配置寄存器 PWM1_CLK_DIV 得到 sys_clk 时钟的 1/2/4/8/16/32 分频。

其次，做时钟门控：

- 将寄存器 CLK_EN2 的 PWM0CEN 位配置为 0，关闭 PWM0 模块时钟，配置为 1，打开 PWM0 模块时钟。
- 将寄存器 CLK_EN2 的 PWM1CEN 位配置为 0，关闭 PWM1 模块时钟，配置为 1，打开 PWM1 模块时钟。

3.1.3.7.5 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 1/2/4/8/16/32 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 CLK_EN1.GPIOCEN 为 0，关闭 gpio_clk，配置 CLK_EN1.GPIOCEN 为 1，打开 gpio_clk。

3.1.3.7.6 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：

首先，对 sys_clk 做时钟分频：

- 将时钟 sys_clk 链接至 SPI 分频器。
- 分频器可通过配置寄存器 SPI_CLK_DIV 得到 1/2/4/8/16/32 分频。

其次，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 CLK_EN2.SPICEN 为 0，关闭 spi_clk，配置 CLK_EN2.SPICEN 为 1，打开 spi_clk。

3.1.3.7.7 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：

首先，对 sys_clk 做时钟分频：

- 将时钟 sys_clk 链接至 I2C 分频器。
- 分频器可通过配置寄存器 I2C_CLK_DIV 得到 1/2/4/8/16/32 分频。

其次，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 CLK_EN2.I2CCEN 为 0，关闭 i2c_clk，配置 CLK_EN2.I2CCEN 为 1，打开 i2c_clk。

3.1.3.7.8 模拟比较器滤波时钟

模拟比较器滤波时钟的最高工作时钟为 73.728MHz，默认工作时钟为 9.19MHz，模拟比较器滤波时钟产生过程如下：

首先，对 sys_clk 做时钟分频：

- 将时钟 sys_clk 链接至模拟比较器分频器。
- 分频器可通过配置寄存器 CMP_CLK_CTRL 得到 1/2/4/8/16/32 分频。
- 将寄存器 CMP_CLK_CTRL 的低 3 位配置为 111，可以选择 32KHz 作为模拟比较器的滤波时钟。

其次，对分频器输出的时钟做门控：

- 将分频器的输出连接至门控单元。
- 配置 CMP_CLK_CTRL.CMPCLKEN 为 0，关闭 cmp_clk，配置 CMP_CLK_CTRL.CMPCLKEN 为 1，打开 cmp_clk。

3.1.4 寄存器描述

该模块共有 25 个寄存器，对应的 SFR PAGE 为 3，具体列表如下：

Table 5 时钟模块寄存器（SFRPAGE=0x03）

地址	寄存器名	类型	复位值	功能描述
0x8C	SYS_CLK_SEL	W/R	0x00	系统时钟选择寄存器
0x8F	SYS_CLK_DIV	W/R	0x03	系统时钟分频系数选择寄存器
0x8D	ADC_CLK_DIV	W/R	0x09	AD 采样时钟分频系数选择寄存器
0xB2	GPIO_CLK_DIV	W/R	0x00	GPIO 时钟分频系数选择寄存器
0xAB	SPI_CLK_DIV	W/R	0x00	SPI 时钟分频系数选择寄存器
0xB1	CMP_CLK_CTRL	W/R	0x00	模拟比较器滤波时钟控制寄存器

地址	寄存器名	类型	复位值	功能描述
0xAC	I2C_CLK_DIV	W/R	0x00	I2C 时钟分频系数选择寄存器
0xAD	PWM0_CLK_DIV	W/R	0x00	PWM0 时钟分频系数选择寄存器
0xAE	PWM1_CLK_DIV	W/R	0x00	PWM1 时钟分频系数选择寄存器
0x91	CLK_EN0	W/R	0x03	时钟门控寄存器 0
0xA9	CLK_EN1	W/R	0x3C	时钟门控寄存器 1
0xAA	CLK_EN2	W/R	0x00	时钟门控寄存器 2
0x9E	OSCPD	W/R	0x00	振荡器时钟控制寄存器
0xA7	OSCEN	W/R	0x00	片外 22.1184MHz 振荡器输出控制寄存器
0x8A	EXTCLK_DIVL	W/R	0xA2	片外 22.1184MHz 时钟分频系数寄存器低 8 位
0x8B	EXTCLK_DIVH	W/R	0x02	片外 22.1184MHz 时钟分频系数寄存器高 2 位
0xA4	OSC_STABL	W/R	0xE0	外部 22.1184MHz 振荡器稳定时间寄存器低 8 位
0xA5	OSC_STABH	W/R	0x2E	外部 22.1184MHz 振荡器稳定时间寄存器高 8 位
0xA6	CLK_READY	R	0x00	时钟稳定标识寄存器
0x8E	RTC_CLK_CTRL	W/R	0x06	RTC 时钟控制寄存器
0x97	TRMVAL73M	W/R	0x40	73.728MHz 时钟校准值
0x99	TRMVAL32K	W/R	0x08	32KHz 时钟校准值
0xA3	BANDGAP_CTRL	W/R	0x10	BANDGAP 校准值
0x87	ADCTRIM	W/R	0x08	ADC 内部参考电压校准值
0x9F	CLKOUTCTRL	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) [0x8C]

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	1-0	W/R	00	系统时钟选择寄存器:

参数名	比特位	属性	复位值	描述
				00/10: 选择 73.728MHz 时钟; 01: 选择片外 22.1184MHz 时钟; 11: 选择片内 32KHz 时钟。 当采用片外 22.1184MHz 振荡器作为时钟源时, 若外部振荡器发生故障, 则此寄存器自动配置为 00, 即系统时钟自动切换至片内 73.728MHz 振荡器产生的时钟。
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) [0x8F]

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	2-0	W/R	011	系统时钟分频系数选择寄存器: 000: 1 分频; 001: 2 分频; 010: 4 分频; 011: 8 分频; 100: 16 分频; 其他: 32 分频。
Reserved	7-3	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) [0x8D]

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) [0xB2]

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	2-0	W/R	000	GPIO 模块时钟分频系数选择寄存器:

参数名	比特位	属性	复位值	描述
				000: 1 分频; 001: 2 分频; 010: 4 分频; 011: 8 分频; 100: 16 分频; 其他: 32 分频。
Reserved	7-3	R	0	保留

3.1.4.5 SPI 时钟分频系数选择寄存器(SPI_CLK_DIV) [0xAB]

Table 10 SPI 时钟分频系数选择寄存器(SPI_CLK_DIV)

参数名	比特位	属性	复位值	描述
SPICLKDIV	2-0	W/R	000	SPI 模块时钟分频系数选择寄存器: 000: 1 分频; 001: 2 分频; 010: 4 分频; 011: 8 分频; 100: 16 分频; 其他: 32 分频;
Reserved	7-3	R	0	保留

3.1.4.6 模拟比较器滤波时钟控制寄存器(CMP_CLK_CTRL) [0xB1]

Table 11 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。 其他: 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:

参数名	比特位	属性	复位值	描述
				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

3.1.4.7 I2C 时钟分频系数选择寄存器(I2C_CLK_DIV) [0xAC]

Table 12 I2C 时钟分频系数选择寄存器(I2C_CLK_DIV)

参数名	比特位	属性	复位值	描述
I2CCLKDIV	2-0	W/R	000	I2C 模块时钟分频系数选择寄存器: 000: 1 分频; 001: 2 分频; 010: 4 分频; 011: 8 分频; 100: 16 分频; 其他: 32 分频。
Reserved	7-3	R	0	保留

3.1.4.8 PWM0 时钟分频系数选择寄存器(PWM0_CLK_DIV) [0xAD]

Table 13 PWM0 时钟分频系数选择寄存器(PWM0_CLK_DIV)

参数名	比特位	属性	复位值	描述
PWM0CLKDIV	2-0	W/R	000	PWM0 模块时钟分频系数选择寄存器: 000: 1 分频; 001: 2 分频; 010: 4 分频; 011: 8 分频; 100: 16 分频; 其他: 32 分频。
Reserved	7-3	R	0	保留

3.1.4.9 PWM1 时钟分频系数选择寄存器(PWM1_CLK_DIV) [0xAE]

Table 14 PWM1 时钟分频系数选择寄存器(PWM1_CLK_DIV)

参数名	比特位	属性	复位值	描述
PWM1CLKDIV	2-0	W/R	000	PWM1 模块时钟分频系数选择寄存器: 000: 1 分频;

参数名	比特位	属性	复位值	描述
				001: 2 分频; 010: 4 分频; 011: 8 分频; 100: 16 分频; 其他: 32 分频。
Reserved	7-3	R	0	保留

3.1.4.10 时钟门控寄存器 0(CLK_EN0) [0x91]

Table 15 时钟门控寄存器 0(CLK_EN0)

参数名	比特位	属性	复位值	描述
CLKOUTCEN	0	W/R	1	输出时钟门控: 1: 打开输出时钟; 0: 关闭输出时钟。
FILTERCEN	1	W/R	1	GPIO 模块 PA0 和 NRST 管脚滤波时钟门控: 1: 打开 PA0 和 NRST 管脚滤波时钟; 0: 关闭 PA0 和 NRST 管脚滤波时钟。
Reserved	7-2	R	0	保留

3.1.4.11 时钟门控寄存器 1(CLK_EN1) [0xA9]

Table 16 时钟门控寄存器 1(CLK_EN1)

参数名	比特位	属性	复位值	描述
CORDICCEN	0	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
Reserved	1	R	0	保留
UART0CEN	2	W/R	1	UART0 模块时钟门控: 1: 打开 UART0 模块时钟; 0: 关闭 UART0 模块时钟。
UART1CEN	3	W/R	1	UART1 模块时钟门控: 1: 打开 UART1 模块时钟; 0: 关闭 UART1 模块时钟。
TIMERCEN	4	W/R	1	TIMER 模块时钟门控: 1: 打开 TIMER 模块时钟;

参数名	比特位	属性	复位值	描述
				0: 关闭 TIMER 模块时钟。
GPIOCEN	5	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。
Reserved	6	R	0	保留
DSPCEN	7	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。

3.1.4.12 时钟门控寄存器 2(CLK_EN2) [0xAA]

Table 17 时钟门控寄存器 2(CLK_EN2)

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
PWM0CEN	1	W/R	0	PWM0 模块时钟门控: 1: 打开 PWM0 模块时钟 0: 关闭 PWM0 模块时钟
PWM1CEN	2	W/R	0	PWM1 模块时钟门控: 1: 打开 PWM1 模块时钟 0: 关闭 PWM1 模块时钟
Reserved	4-3	R	0	保留
SPICEN	5	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。
ADC0CEN	6	W/R	0	ADC0 模块时钟门控: 1: 打开 ADC0 模块时钟; 0: 关闭 ADC0 模块时钟。
ADC1CEN	7	W/R	0	ADC1 模块时钟门控: 1: 打开 ADC1 模块时钟; 0: 关闭 ADC1 模块时钟。

3.1.4.13 振荡器时钟控制寄存器(OSCPD) [0x9E]

Table 18 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 73.728MHz 振荡器使能： 0：片内 73.728MHz 振荡器正常工作； 1：关闭片内 73.728MHz 振荡器。
OSC22MEN	2	W/R	0	片外 22.1184MHz 振荡器时钟门控： 0：关闭片外 22.1184MHz 振荡器时钟； 1：打开片外 22.1184MHz 振荡器时钟。
Reserved	7-3	R	0	保留

3.1.4.14 片外 22.1184MHz 振荡器输出控制寄存器(OSCEN) [0xA7]

Table 19 片外 22.1184MHz 振荡器输出控制寄存器(OSCEN)

参数名	比特位	属性	复位值	描述
OSCEN	0	W/R	0	片外振荡器使能： 0：片外 OSC 不使能； 1：片外 OSC 使能。
CMSINT	1	W/R	0	片外振荡器故障中断信号： 0：片外振荡器未发生故障； 1：片外振荡器发生故障。 软件写 0 清除该中断。
CMSEN	2	W/R	0	时钟监控系统使能寄存器： 0：关闭时钟监控系统； 1：打开时钟监控系统。
Reserved	7-3	R	0	保留

3.1.4.15 片外 22.1184MHz 时钟分频系数寄存器低 8 位(EXTCLK_DIVL) [0xA8]

Table 20 片外 22.1184MHz 时钟分频系数寄存器低 8 位(EXTCLK_DIVL)

参数名	比特位	属性	复位值	描述
EXTCLKDIVL	7-0	W/R	0xA2	片外 22.1184MHz 时钟分频系数的低 8 位。

3.1.4.16 片外 22.1184MHz 时钟分频系数寄存器高 2 位(EXTCLK_DIVH) [0xA8B]

Table 21 片外 22.1184MHz 时钟分频系数寄存器高 2 位(EXTCLK_DIVH)

参数名	比特位	属性	复位值	描述
EXTCLKDIVH	1-0	W/R	0x02	片外 22.1184MHz 时钟分频系数的高 2 位。
Reserved	7-2	R	0	保留

3.1.4.17 片外振荡器稳定时间寄存器低 8 位(OSC_STABL) [0xA4]

Table 22 片外振荡器稳定时间寄存器低 8 位(OSC_STABL)

参数名	比特位	属性	复位值	描述
OSCSTBTL	7-0	W/R	0xe0	片外 22.1184MHz 振荡器稳定时间寄存器低 8 位。

3.1.4.18 片外振荡器稳定时间寄存器高 8 位(OSC_STABH) [0xA5]

Table 23 片外振荡器稳定时间寄存器高 8 位(OSC_STABH)

参数名	比特位	属性	复位值	描述
OSCSTBTH	7-0	W/R	0x2e	片外 22.1184MHz 振荡器稳定时间寄存器高 8 位。

3.1.4.19 时钟稳定标识寄存器(CLK_READY) [0xA6]

Table 24 时钟稳定标识寄存器(CLK_READY)

参数名	比特位	属性	复位值	描述
Reserved	1-0	R	00	保留
OSC22MRDY	2	R	0	片外 22.1184MHz 振荡器时钟稳定标识寄存器。 0: 片外 22.1184MHz 振荡器时钟未稳定; 1: 片外 22.1184MHz 振荡器时钟稳定。
Reserved	7-3	R	0	保留

3.1.4.20 RTC 时钟控制寄存器(RTC_CLK_CTRL) [0x8E]

Table 25 RTC 时钟控制寄存器(RTC_CLK_CTRL)

参数名	比特位	属性	复位值	描述
RTCCLKSEL	0	W/R	0	RTC 模块时钟选择寄存器: 0: 选择片内 32KHz 振荡器产生的时钟; 1: 选择片外 22.1184MHz 振荡器产生的时钟作为时钟源, 进行 675 分频后得到 32768Hz 时钟。 当采用片外 22.1184MHz 振荡器作为时钟

参数名	比特位	属性	复位值	描述
				源时，若外部振荡器发生故障，则此控制位由硬件自动配置为 0，即 RTC 模块的时钟自动切换至片内 32KHz 振荡器产生的时钟。
EXTCLKDIVEN	1	W/R	1	片外 22.1184MHz 分频时钟输出使能： 0：关闭分频时钟； 1：输出分频时钟。
RTCCLKEN	2	W/R	1	RTC 模块的时钟门控寄存器： 0：关闭 RTC 模块时钟； 1：打开 RTC 模块时钟。
Reserved	7-3	R	0x00	保留

3.1.4.21 73.728MHz 时钟校准值(TRMVAL73M) [0x97]

Table 26 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

3.1.4.22 32KHz 时钟校准值(TRMVAL32K) [0x99]

Table 27 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

3.1.4.23 BANDGAP 输出值校准(BANDGAP_CTRL) [0xA3]

Table 28 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

3.1.4.24 ADC 参考电压校准值寄存器(ADCTRIM) [0x87]

Table 29 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

3.1.4.25 输出时钟控制寄存器(CLKOUTCTRL) [0x9F]

Table 30 输出时钟控制寄存器(CLKOUTCTRL)

参数名	比特位	属性	复位值	描述
Reserved	1-0	R	0	保留
CLKOUTCTRL	3-2	W/R	00	输出时钟选择寄存器： x0: 选系统时钟作为输出时钟； 01: 选片内 32KHz 时钟作为输出时钟； 11: 选片外 22.1184MHz 时钟作为输出时钟。
Reserved	7-4	R	0	保留

3.2 复位管理

3.2.1 概述

JMT1808R 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)

JMT1808R 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

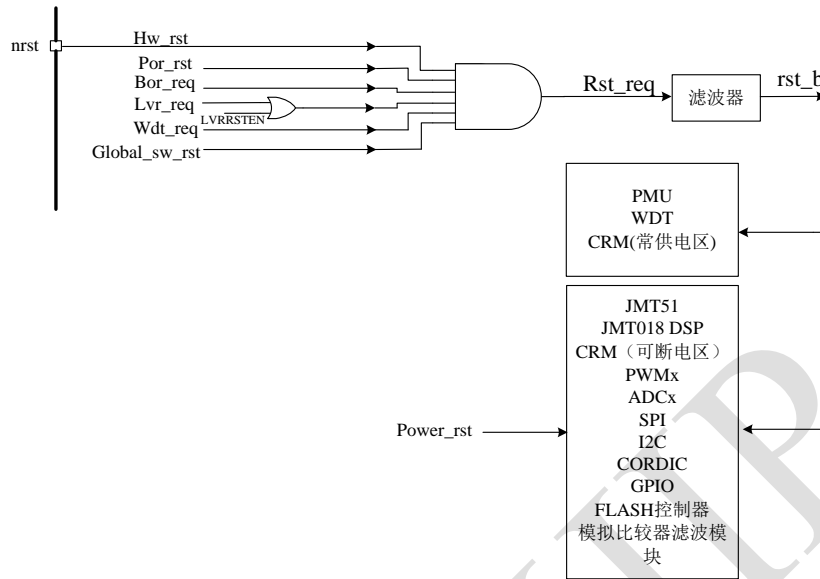


Figure 11 JMT1808R 复位

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, 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 0, 打开掉电复位检测电路, 当 CRMCTRL.BORSWEN 配置为 1, 关闭掉电复位检测电路, 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 (RTC 模块不复位), 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（RTC 模块不复位），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，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 `LVERRSTEN` 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 `LVERRSTEN` 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（RTC 模块不复位），复位结束后，JMT51 核从 FLASH 的 0 地址开始执行应用程序。

3.2.3.5 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（RTC 模块不复位），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

JMT1808R 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（RTC 模块不复位），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 35 和 Table 36。

3.2.3.9 复位信号对各模块的作用

表 Table 31 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 时钟分频寄存器 SPI_CLK_DIV、时钟使能寄存器 CLK_EN2.SPICEN 和 SPI 时钟产生单元。
- I2C 时钟分频寄存器 I2C_CLK_DIV、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。
- PWM0 时钟分频寄存器 PWM_CLK_DIV、时钟使能寄存器 CLK_EN2.PWM0CEN 和 PWM0 时钟产生单元。
- PWM1 时钟分频寄存器 PWM_CLK_DIV、时钟使能寄存器 CLK_EN2.PWM1CEN 和 PWM1 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。

表中（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 的其他的部分复位。

Table 31 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
CORDIC 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

模块	POR_ RST	HW_ RST	BOR_ RST	LVR_ RST	WDT_ RST	GALOBAL_ RST	POWER_ RST
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 滤波模块	复位	复位	复位	复位	复位	复位	复位
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 3，具体列表如下。

Table 32 复位模块寄存器(SFRPAGE = 0x03)

地址	寄存器名	属性	复位值	功能描述
0x9A	CRMCTRL	W/R	0x04	CRM 模块控制
0x96	RSTFLAG	W/R	0x20	复位标志
0x93	SWRST0	W	0x00	模块软复位 0
0x94	SWRST1	W	0x00	模块软复位 1
0x95	GLOBALRST	W/R	0x00	全局软复位

3.2.4.1 CRM 模块控制寄存器(CRMCTRL) [0x9A]

Table 33 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： 0：BOR 模块正常工作； 1：关闭 BOR 模块。

参数名	比特位	属性	复位值	描述
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。
Reserved	7-4	R	0	保留

3.2.4.2 复位标识寄存器(RSTFLAG) [0x96]

Table 34 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： 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。

参数名	比特位	属性	复位值	描述
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) [0x93]

Table 35 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； 硬件自动将此位清 0。
RTCSWRST	1	W	0	RTC 模块软复位寄存器： 0：无效； 1：对 RTC 模块进行复位； 硬件自动将此位清 0。
PWM0SWRST	2	W	0	PWM0 模块软复位寄存器： 0：无效； 1：对 PWM0 模块进行复位； 硬件自动将此位清 0。
PWM1SWRST	3	W	0	PWM1 模块软复位寄存器： 0：无效； 1：对 PWM1 模块进行复位； 硬件自动将此位清 0。
Reserved	4	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； 硬件自动将此位清 0。
I2CSWRST	6	W	0	I2C 模块软复位寄存器： 0：无效；

参数名	比特位	属性	复位值	描述
				1: 对 I2C 模块进行复位; 硬件自动将此位清 0。
Reserved	7	R	0	保留

3.2.4.4 模块软复位寄存器 1(SWRST1) [0x94]

Table 36 模块软复位寄存器 1(SWRST1)

参数名	比特位	属性	复位值	描述
GPIOSWRST	0	W	0	GPIO 模块软复位寄存器: 0: 无效; 1: 对 GPIO 模块进行复位; 硬件自动将此位清 0。
Reserved	1	R	0	保留
ADC0SWRST	2	W	0	ADC 0 模块软复位寄存器: 0: 无效; 1: 对 ADC 0 模块进行复位; 硬件自动将此位清 0。
ADC1SWRST	3	W	0	ADC 1 模块软复位寄存器: 0: 无效; 1: 对 ADC 1 模块进行复位; 硬件自动将此位清 0。
DSPSWRST	4	W	0	DSP 核软复位寄存器: 0: 无效; 1: 对 DSP 核进行复位; 硬件自动将此位清 0。
Reserved	7-5	R	0	保留

3.2.4.5 全局软复位寄存器(GLOBALRST) [0x95]

Table 37 全局软复位寄存器(GLOBALRST)

参数名	比特位	属性	复位值	描述
GLOBOSWRST	0	W	0	全芯片软复位寄存器: 0: 无效; 1: 对全芯片进行复位 (RTC 模块不复位); 硬件自动将此位清 0。

参数名	比特位	属性	复位值	描述
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。

堆栈位于内部 256 字节的数据存储器，因此堆栈深度最大可以达到 256。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。

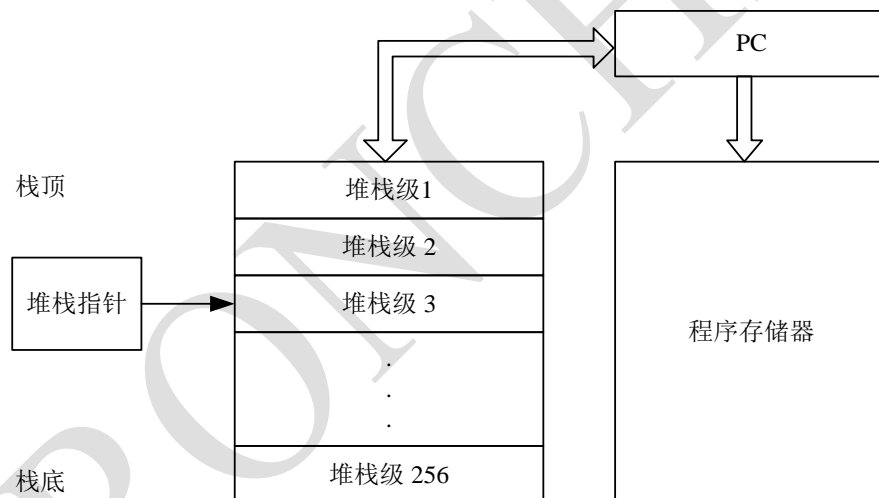


Figure 12 堆栈示意图

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 38 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 39 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
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

助记符	描述	指令码	字节
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 40 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
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

助记符	描述	指令码	字节
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 41 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
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

助记符	描述	指令码	字节
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 42 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
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	D5	3

助记符	描述	指令码	字节
	相对地址		
NOP	空操作	00	1

Table 43 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

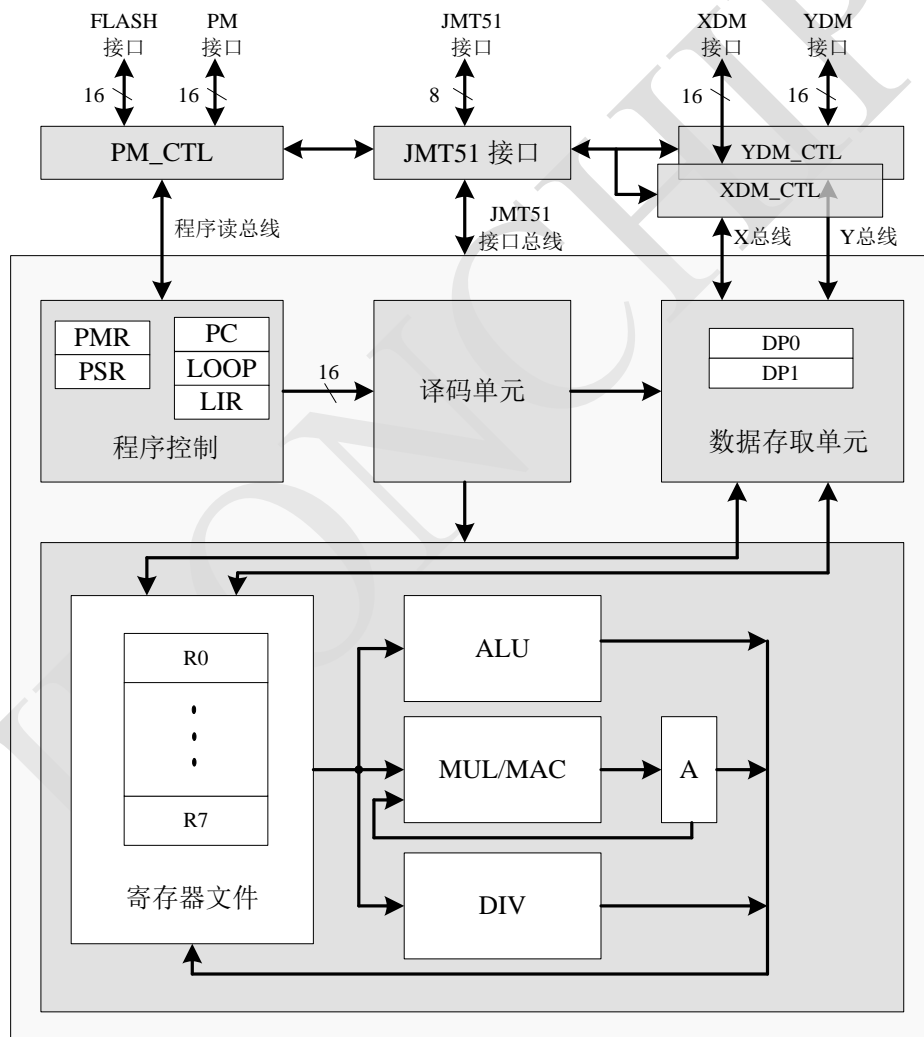


Figure 13 JMT018 DSP 核框图

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 (PM 和 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/ADCx),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/ADCx) 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 (PM 和 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 44 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
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
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 45 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
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 46 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
ASHL Rd,Rn	1	是		是		算术左移 Rn 位
LSHR Rd,Rn	1	是		是		逻辑右移 Rn 位
LSHL 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
LSHL 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 47 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 48 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
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 49 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;

						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
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 自減 1; YDM[DP1]->Rm, DP1 自減 1
IMACN Rn:XDM+,Rm:YDM+	2		是	是	有符号整数乘减并行: A-Rn*Rm->A; XDM[DP0]->Rn, DP0 自增 1; YDM[DP1]->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->A; XDM[DP0]->Rn, DP0 自減 1; YDM[DP1]->Rm, DP1 自減 1
MAC Rn:XDM+,Rm:YDM+	2		是	是	有符号小数乘加并行: A+Rn*Rm->A; XDM[DP0]->Rn, DP0 自增 1; YDM[DP1]->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->A; XDM[DP0]->Rn, DP0 自減 1; YDM[DP1]->Rm, DP1 自減 1
MACN Rn:XDM+,Rm:YDM+	2		是	是	有符号小数乘减并行: A-Rn*Rm->A; XDM[DP0]->Rn, DP0 自增 1; YDM[DP1]->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->A; XDM[DP0]->Rn, DP0 自減 1; YDM[DP1]->Rm, DP1 自減 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 50 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 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

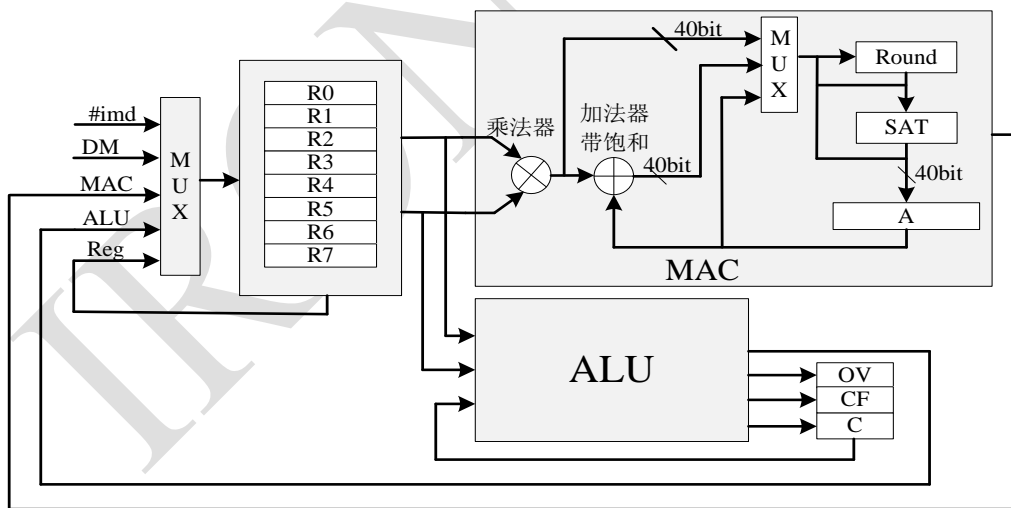


Figure 14 数据运算器框图

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。

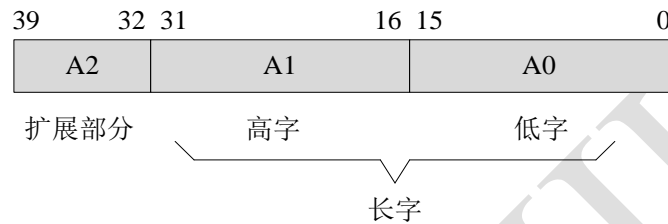


Figure 15 累加器 A 组成

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。

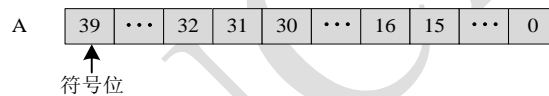


Figure 16 累加器 A 作为整数时数据结构

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。

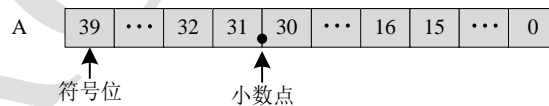


Figure 17 累加器 A 作为小数时数据结构

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。

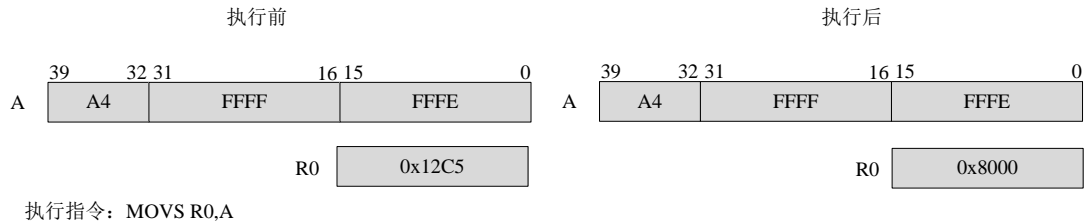


Figure 18 MOV_S Rd,A 指令(例 a)

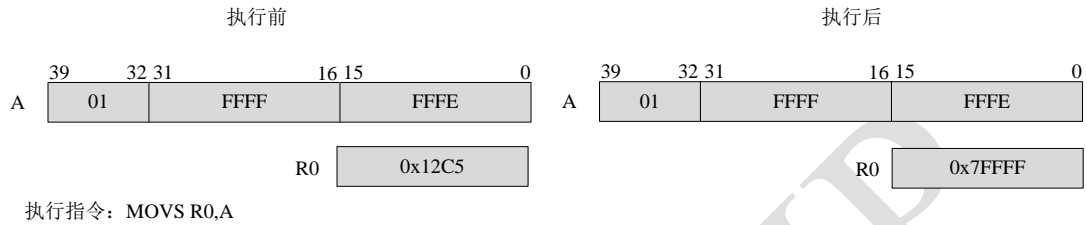


Figure 19 MOV_S Rd,A 指令(例 b)

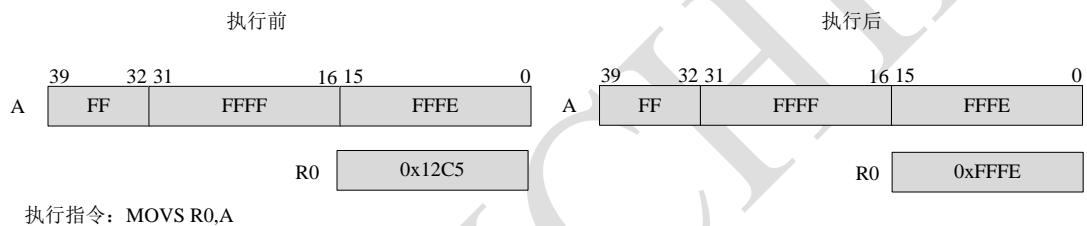


Figure 20 MOV_S Rd,A 指令(例 c)

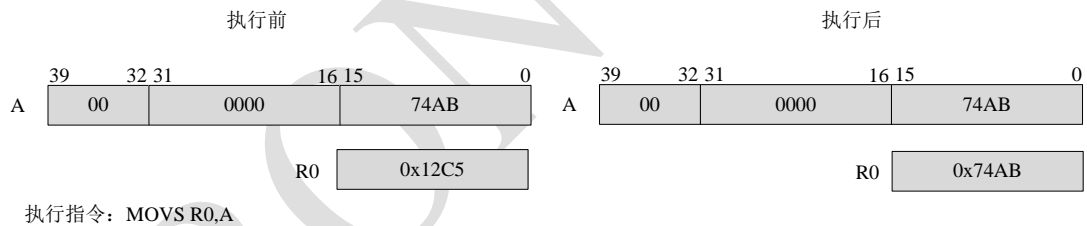


Figure 21 MOV_S Rd,A 指令(例 d)

MOV_R 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。

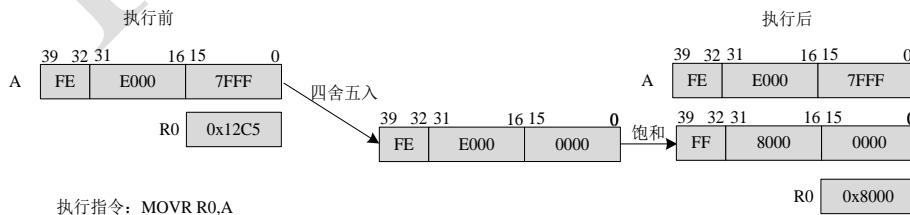


Figure 22 MOV_R Rd,A 指令(例 a)

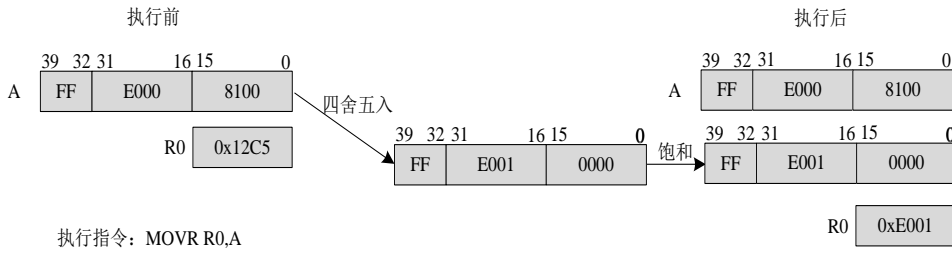


Figure 23 MOV R0,A 指令(例 b)

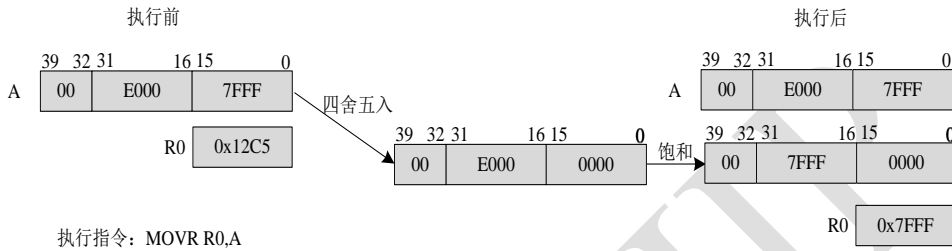


Figure 24 MOV R0,A 指令(例 c)

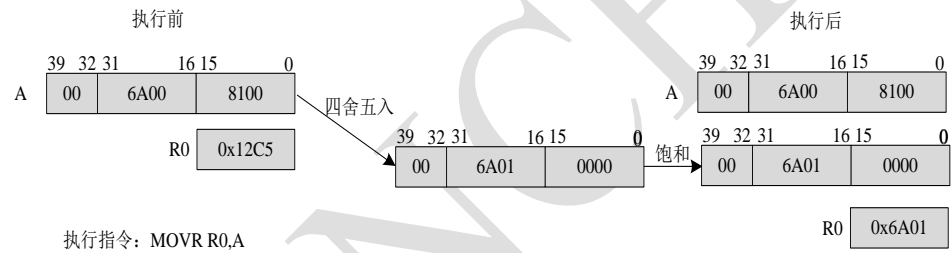


Figure 25 MOV R0,A 指令(例 d)

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。

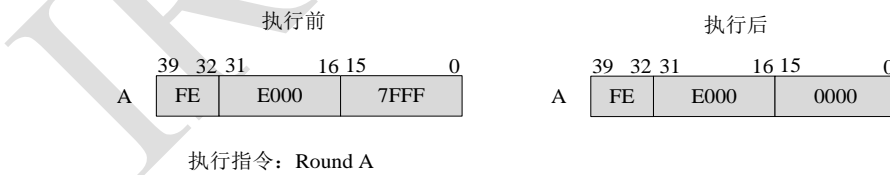


Figure 26 Round A 指令(例 a)

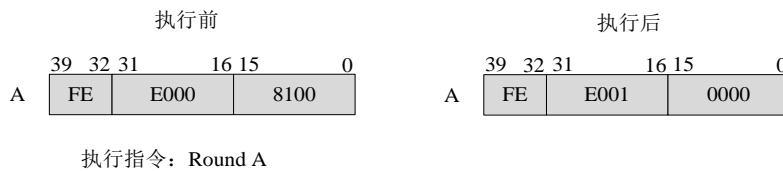


Figure 27 Round A 指令(例 b)

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。

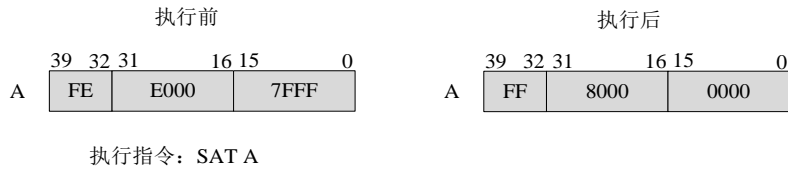


Figure 28 SAT A 指令(例 a)

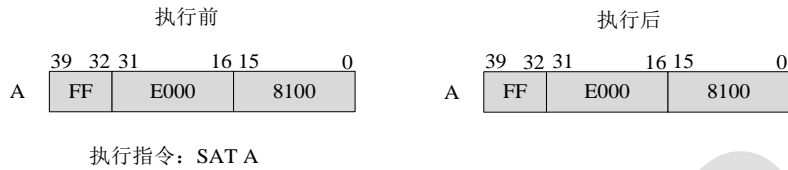


Figure 29 SAT A 指令(例 b)

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。

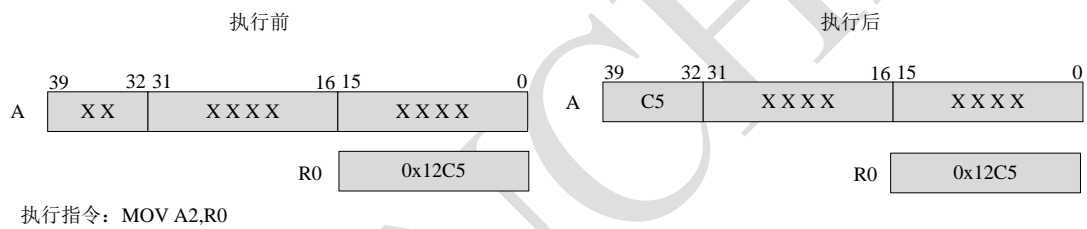


Figure 30 MOV A2,Rn 指令

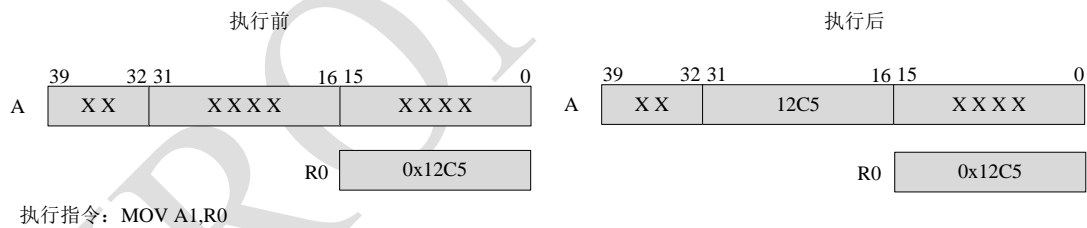


Figure 31 MOV A1,Rn 指令

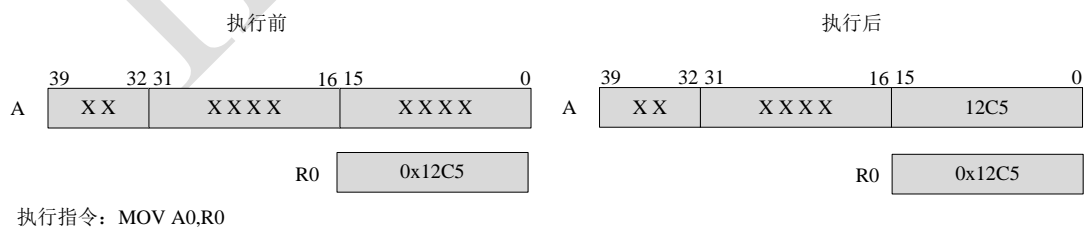


Figure 32 MOV A0,Rn 指令

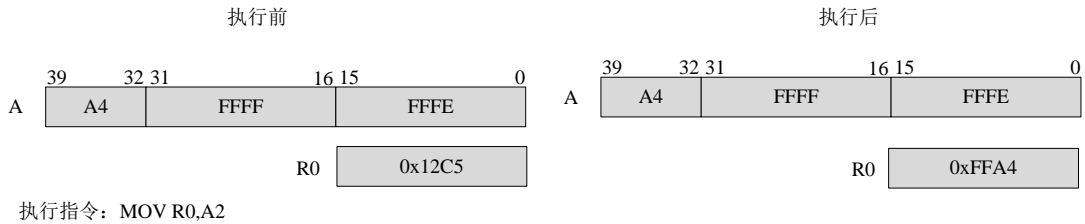


Figure 33 MOV R_d,A₂ 指令

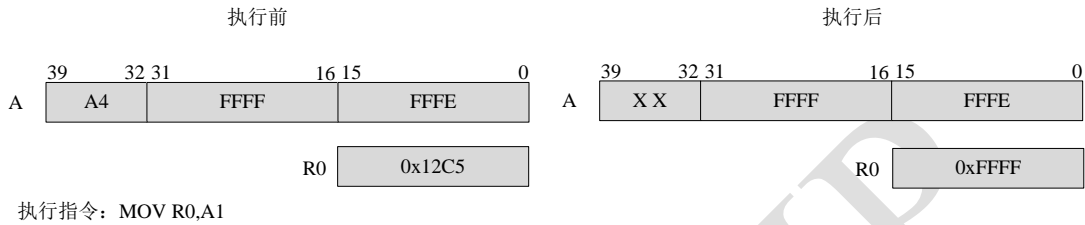


Figure 34 MOV R_d,A₁ 指令

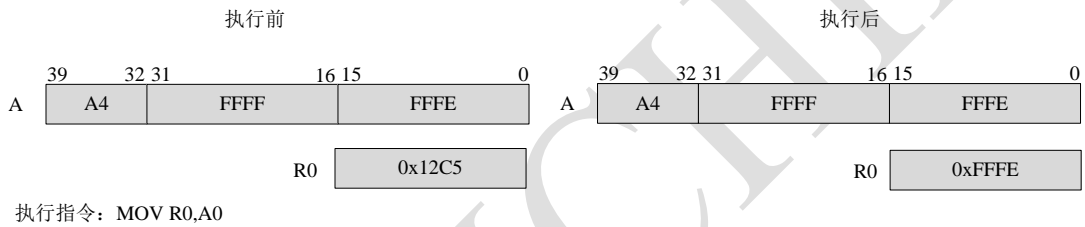


Figure 35 MOV R_d,A₀ 指令

乘加或乘减运算时，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。

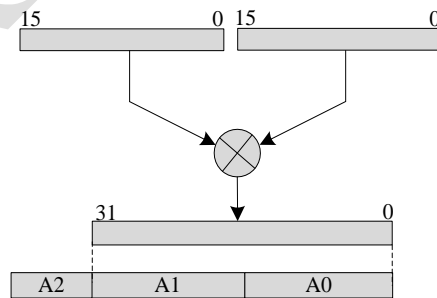


Figure 36 无符号整数相乘

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。

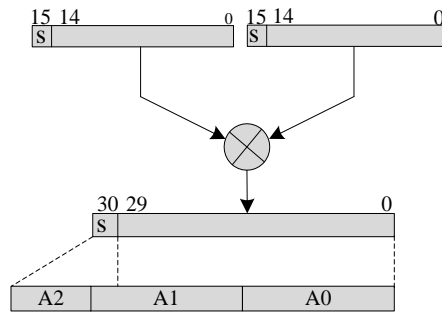


Figure 37 有符号整数相乘

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。

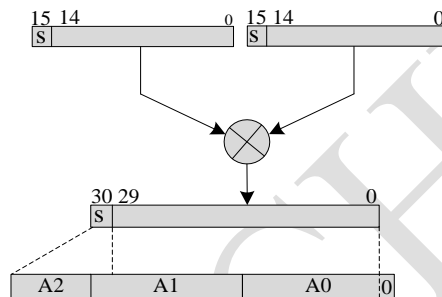


Figure 38 有符号小数相乘

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。

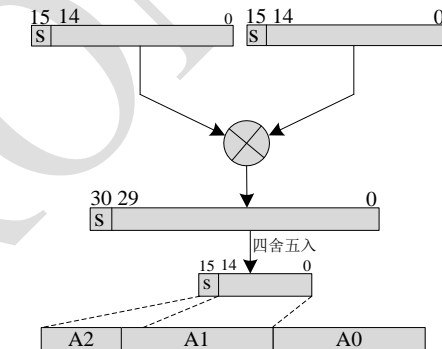


Figure 39 有符号小数相乘带四舍五入

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。

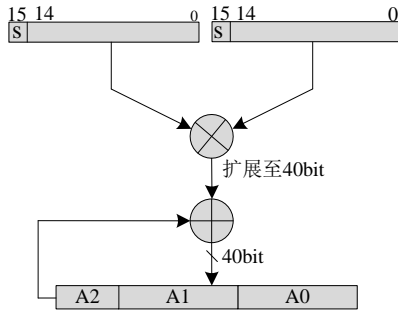


Figure 40 有符号整数乘加

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。

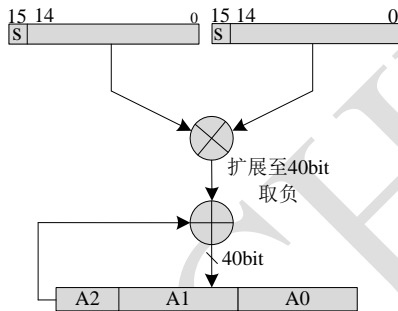


Figure 41 有符号整数乘加

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。

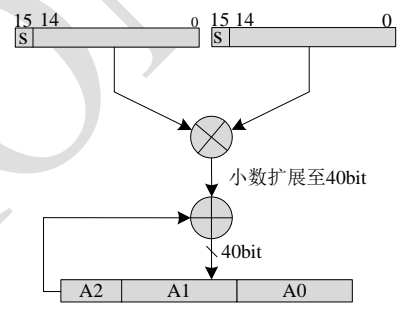


Figure 42 有符号小数乘加

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。

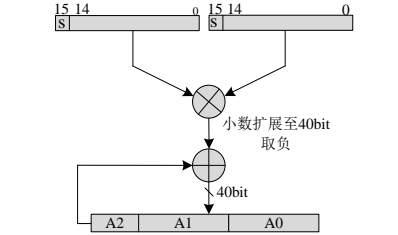


Figure 43 有符号小数乘减

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。

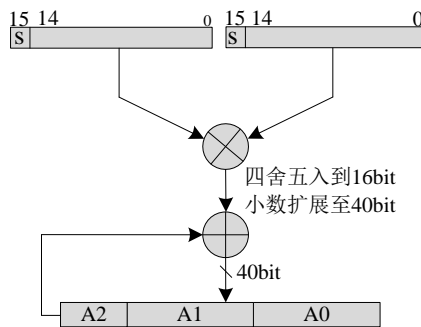


Figure 44 有符号小数乘加带四舍五入

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。

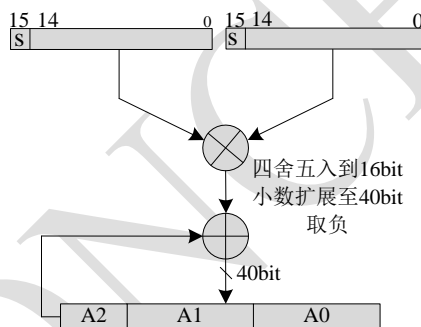


Figure 45 有符号小数乘减带四舍五入

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。

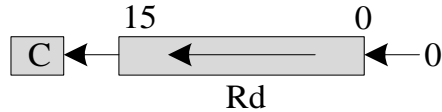


Figure 46 算术左移



Figure 47 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。

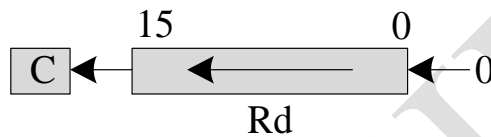


Figure 48 逻辑左移

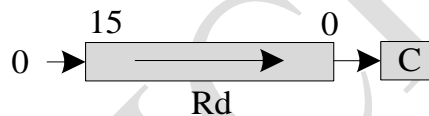


Figure 49 逻辑右移

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 程序空间包括 DSP 程序空间存储器和 FLASH 程序存储器：DSP 程序空间存储器（PM）为 512x16bit，映射为 MCU 的地址为 0x2000~0x23ff 的外部数据空间；FLASH 程序存储器为(16K-512)x16bit，映射为 MCU 的地址为 0x0400~0x7fff 的程序存储空间。

DSP 的寻址地址为 0x0000~0x01ff 时访问的程序存储空间为 PM，寻址地址为 0x0200~0x3fff 时访问的程序存储空间为 FLASH。

16 位 DSP 程序空间与 8 位 MCU 访问空间的对应关系如 Figure 50 所示。

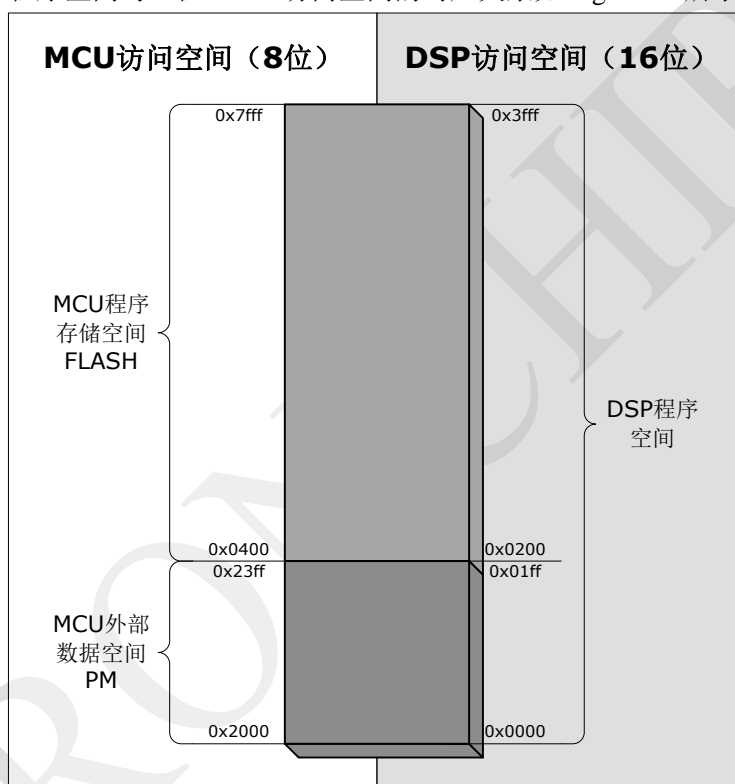


Figure 50 DSP 程序存储空间

当访问程序存储空间为 PM 时，DSP 运行最高频率为 73.728MHz，当访问程序存储空间为 FLASH 时，DSP 运行最高频率为 36.864MHz，具体见 Table 51。

Table 51 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 PM 中	73.728	73.728	73.728
	36.864	36.864	36.864
	18.432	18.432	18.432

	9.216	9.216	9.216
DSP 工作于	73.728	73.728	36.864
FLASH 中	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 PM 或 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.PCW 为 1，将配置的 PC 值更新到 DSP 的 PC 中；
- (4) 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.PCW 为 1，将配置的 PC 值更新到 DSP 的 PC 中；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 JMT1808R 的下载程序。

通过芯片的 BOOT 程序将 JMT1808R 的程序下载至 FLASH 中。

JMT018 代码可运行于 PM 或 FLASH 中：

- 运行于 PM 中：该模式下需要 JMT51 把下载至 FLASH 中的 DSP 程序搬运至 PM 中；
- 运行于 FLASH 中：该模式下 DSP 直接运行已下载至 FLASH 中的程序，无需 JMT51 核搬运 DSP 运行程序。

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 52 所示。

Table 52 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
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 53 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 54 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 55 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 56 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 57 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 58 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 59 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 60 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 61 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 62 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 63 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 64 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 65 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 LNK R 寄存器进行压栈出栈操作。

5.7.14 状态寄存器 PSR

Table 66 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

参数名	比特位	属性	复位值	描述
				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 67 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 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 68 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 69 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 70 中所列为 JMT51 访问 DSP 寄存器的地址，对应的 SFRPAGE 为 3。

Table 70 JMT51 访问 DSP 寄存器 (SFRPAGE=0x03)

地址	寄存器名	属性	复位值	功能描述
0xD4	DSP_CFG	W	0x00	DSP 配置寄存器
0xD1	DSP_STA	W/R	0x00	DSP 运行状态寄存器
0xD2	DSP_PCL	W/R	0x00	PC 寄存器低字节
0xD3	DSP_PCH	W/R	0x00	PC 寄存器高字节
0xD9	DSP_R0L	W/R	0x00	R0 寄存器低字节
0xDA	DSP_R0H	W/R	0x00	R0 寄存器高字节
0xDB	DSP_R1L	W/R	0x00	R1 寄存器低字节
0xDC	DSP_R1H	W/R	0x00	R1 寄存器高字节
0xDD	DSP_R2L	W/R	0x00	R2 寄存器低字节
0xDE	DSP_R2H	W/R	0x00	R2 寄存器高字节
0xDF	DSP_R3L	W/R	0x00	R3 寄存器低字节
0xE1	DSP_R3H	W/R	0x00	R3 寄存器高字节
0xE2	DSP_R4L	W/R	0x00	R4 寄存器低字节
0xE3	DSP_R4H	W/R	0x00	R4 寄存器高字节
0xE4	DSP_R5L	W/R	0x00	R5 寄存器低字节
0xE5	DSP_R5H	W/R	0x00	R5 寄存器高字节
0xE6	DSP_R6L	W/R	0x00	R6 寄存器低字节
0xE7	DSP_R6H	W/R	0x00	R6 寄存器高字节
0xE9	DSP_R7L	W/R	0x00	R7 寄存器低字节
0xEA	DSP_R7H	W/R	0x00	R7 寄存器高字节
0xEB	DSP_DP0L	W/R	0x00	DP0 寄存器低字节
0xEC	DSP_DP0H	W/R	0x00	DP0 寄存器高字节

地址	寄存器名	属性	复位值	功能描述
0xED	DSP_DP1L	W/R	0x00	DP1 寄存器低字节
0xEE	DSP_DP1H	W/R	0x00	DP1 寄存器高字节
0xEF	DSP_A0L	R	0x00	A 寄存器 7~0 比特
0xF1	DSP_A0H	R	0x00	A 寄存器 15~8 比特
0xF2	DSP_A1L	R	0x00	A 寄存器 23~16 比特
0xF3	DSP_A1H	R	0x00	A 寄存器 31~24 比特
0xF4	DSP_A2	R	0x00	A 寄存器 39~32 比特
0xF5	DSP_LNKRL	R	0x00	LNKR 寄存器低字节
0xF6	DSP_LNKRH	R	0x00	LNKR 寄存器高字节
0xF7	DSP_PSRL	R	0x00	PSR 寄存器低字节
0xF9	DSP_PSRH	R	0x00	PSR 寄存器高字节
0xFA	DSP_PMRL	R	0x04	PMR 寄存器低字节
0xFB	DSP_PMRH	R	0x00	PMR 寄存器高字节
0xFC	DSP_LOOPL	R	0x01	LOOP 寄存器低字节
0xFD	DSP_LOOPH	R	0x00	LOOP 寄存器高字节
0xFE	DSP_LIRL	R	0x01	LIR 寄存器低字节
0xFF	DSP_LIRH	R	0x00	LIR 寄存器高字节

5.8.1 DSP 配置寄存器(DSP_CFG) [0xD4]

Table 71 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 1: 启动 DSP 运行
PCW	1	W	0	PC 寄存器更新控制位： 0: 无效 1: 装载寄存器 DSP_PCH 和 DSP_PCL 至 DSP 的 PC 中
Reserved	7-2	R	0	保留

5.8.2 DSP 运行状态寄存器(DSP_STA) [0xD1]

Table 72 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) [0xD2]

Table 73 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) [0xD3]

Table 74 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) [0xD9]

Table 75 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) [0xDA]

Table 76 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) [0xDB]

Table 77 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) [0xDC]

Table 78 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) [0xDD]

Table 79 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) [0xDE]

Table 80 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) [0xDF]

Table 81 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) [0xE1]

Table 82 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) [0xE2]

Table 83 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) [0xE3]

Table 84 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) [0xE4]

Table 85 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) [0xE5]

Table 86 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) [0xE6]

Table 87 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) [0xE7]

Table 88 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) [0xE9]

Table 89 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) [0xEA]

Table 90 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) [0xEB]

Table 91 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) [0xEC]

Table 92 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) [0xED]

Table 93 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) [0xEE]

Table 94 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) [0xEF]

Table 95 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) [0xF1]

Table 96 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) [0xF2]

Table 97 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) [0xF3]

Table 98 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) [0xF4]

Table 99 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) [0xF5]

Table 100 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) [0xF6]

Table 101 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) [0xF7]

Table 102 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 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

参数名	比特位	属性	复位值	描述
Reserved	7-5	R	0	保留

5.8.33 PSR 寄存器高字节(DSP_PSRH) [0xF9]

Table 103 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) [0xFA]

Table 104 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) [0xFB]

Table 105 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) [0xFC]

Table 106 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) [0xFD]

Table 107 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) [0xFE]

Table 108 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) [0xFF]

Table 109 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 19 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 19 个中断源分别控制，通过配置寄存器的 IEN0.0~IEN0.7、IEN1.0~IEN1.7 及 IEN2.0~IEN2.7 比特位开启或屏蔽相应中断。
- 第三级控制为连接到第二级 19 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。

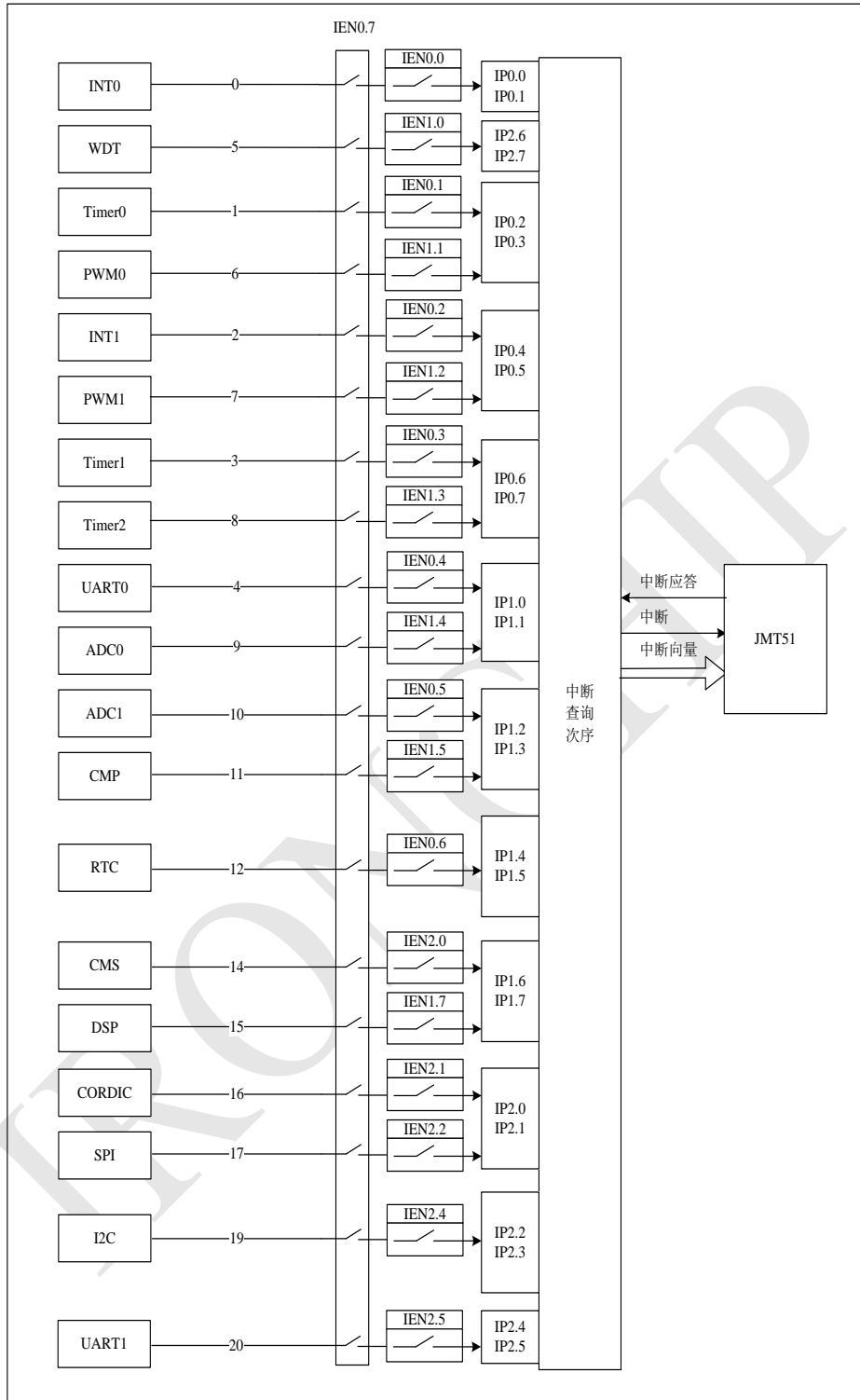


Figure 51 中断各级使能框图

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 110 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 110 所示。

Table 110 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	INT0 中断	最高 0	软件清除
1	0x0B	Timer 0 中断	2	硬件自动清除
2	0x13	INT1 中断	4	软件清除
3	0x1B	Timer 1 中断	6	硬件自动清除
4	0x23	UART0 中断	8	软件清除
5	0x2B	WDT 中断	1	软件清除
6	0x33	PWM0 中断	3	软件清除
7	0x3B	PWM1 中断	5	软件清除
8	0x43	Timer2 中断	7	硬件自动清除
9	0x4B	ADC0 中断	9	软件清除
10	0x53	ADC1 中断	10	软件清除
11	0x5B	CMP 中断	11	软件清除
12	0x63	RTC 中断	12	软件清除
14	0x73	CMS 中断	13	软件清除
15	0x7B	DSP 中断	14	软件清除
16	0x83	CORDIC 中断	15	软件清除
17	0x8B	SPI 中断	16	软件清除
19	0x9B	I2C 中断	17	软件清除
20	0xA3	UART1 中断	18	软件清除

6.2 外部中断 (INT0/INT1)

JMT51 共有 2 个外部中断：INT0、INT1，每个中断由多个 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。有关外部中断 INT0/INT1 的详细说明参见 19.7、19.8 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 110 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 111 中断优先级分组

组	中断源		描述
0	INT0 中断		通过 IP0.0、IP0.1 设置优先级
1	Timer 0 中断	PWM0 中断	通过 IP0.2、IP0.3 设置优先级
2	INT1 中断	PWM1 中断	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	UART0 中断	ADC0 中断	通过 IP1.0、IP1.1 设置优先级
5	CMP 中断	ADC1 中断	通过 IP1.2、IP1.3 设置优先级
6	RTC 中断		通过 IP1.4、IP1.5 设置优先级
7	CMS 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	UART1 中断		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。具体列表如下。

Table 112 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 113 中断使能寄存器 0(IEN0)

参数名	比特位	属性	复位值	描述
EXINT0	0	W/R	0	INT0 中断允许位： 0：禁止 INT0 中断 1：允许 INT0 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
EXINT1	2	W/R	0	INT1 中断允许位： 0：禁止 INT1 中断 1：允许 INT1 中断
ET1	3	W/R	0	Timer1 中断允许位： 0：禁止 Timer1 中断 1：允许 Timer1 中断
ES0	4	W/R	0	UART0 发送/接收中断允许位： 0：禁止 UART0 发送/接收中断 1：允许 UART0 发送/接收中断

参数名	比特位	属性	复位值	描述
ADC1INT	5	W/R	0	ADC1 中断允许位： 0: 禁止 ADC1 中断 1: 允许 ADC1 中断
RTCINT	6	W/R	0	RTC 中断允许位： 0: 禁止 OCP 中断 1: 允许 OCP 中断
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 114 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
PWM0INT	1	W/R	0	PWM0 中断允许位： 0: 禁止 PWM0 中断 1: 允许 PWM0 中断
PWM1INT	2	W/R	0	PWM1 中断允许位： 0: 禁止 PWM1 中断 1: 允许 PWM1 中断
ET2	3	W/R	0	Timer2 中断允许位： 0: 禁止 Timer2 中断 1: 允许 Timer2 中断
ADC0INT	4	W/R	0	ADC0 中断允许位： 0: 禁止 ADC0 中断 1: 允许 ADC0 中断
CMPINT	5	W/R	0	CMP 中断允许位： 0: 禁止 CMP 中断 1: 允许 CMP 中断
Reserved	6	R	0	保留

参数名	比特位	属性	复位值	描述
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 115 中断使能寄存器 2(IEN2)

参数名	比特位	属性	复位值	描述
CMSINT	0	W/R	0	CMS 中断允许位： 0: 禁止 CMS 中断 1: 允许 CMS 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
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
UART1INT	5	W/R	0	UART1 中断允许位： 0: 禁止 UART1 中断 1: 允许 UART1 中断
Reserved	7-6	R	0	保留

6.6.4 中断优先级寄存器 0 (IP0) [0xB9]

Table 116 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	INT0 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）

参数名	比特位	属性	复位值	描述
IP0.3~IP0.2	3-2	W/R	0	Timer0 中断和 PWM0 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）
IP0.5~IP0.4	5-4	W/R	0	INT1 和 PWM1 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）
IP0.7~IP0.6	3-2	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 117 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	UART0 中断和 ADC0 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）
IP1.3~IP1.2	3-2	W/R	0	ADC1 中断和 CMP 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）
IP1.5~IP1.4	5-4	W/R	0	RTC 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)
IP1.7~IP1.6	7-6	W/R	0	CMS 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 118 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)
IP2.5~IP2.4	5-4	W/R	0	UART1 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)
IP2.7~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

JMT1808R 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 52 所示。

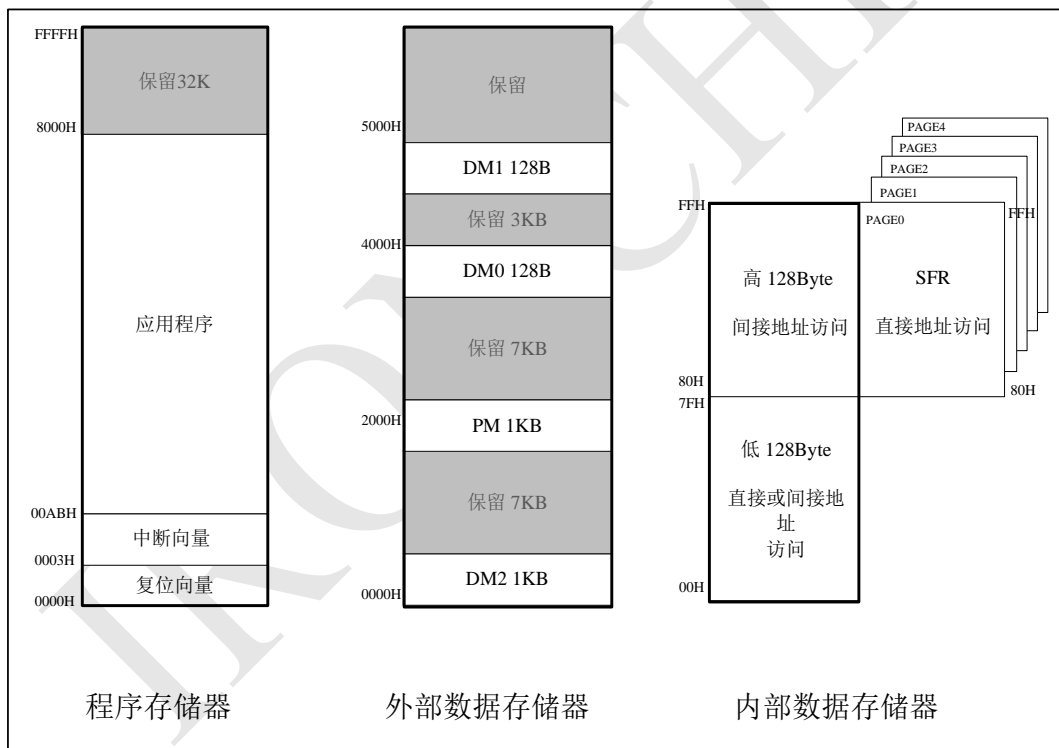


Figure 52 JMT51 存储器分布图

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 52 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 8 FLASH 存储器描述。

7.1.2 外部数据存储器

JMT51 外部数据存储器由四部分组成，如 Figure 52 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；
- PM: PM 外部存储器大小为 1K 字节，地址空间为：0x2000~0x23ff；
- 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 256 字节，地址空间为 0x00~0xff，如 Figure 53 所示。

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

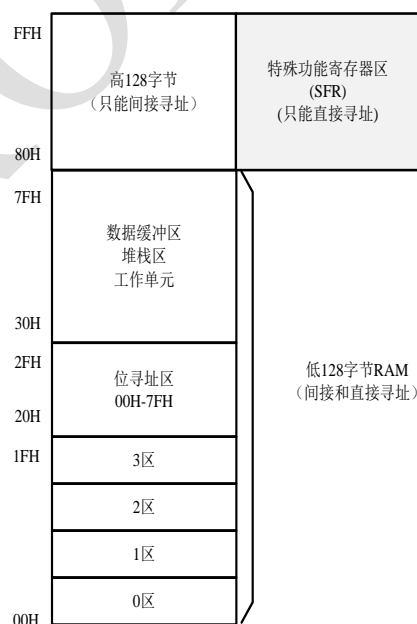


Figure 53 内部数据存储器分区

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 119 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 120 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 120 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 256 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 256 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 低 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 个字节与高 128 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。JMT1808R 芯片 SFR 分为 5 页 (PAGE)：PAGE0/PAGE1/PAGE2/PAGE3/PAGE4。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 54 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。

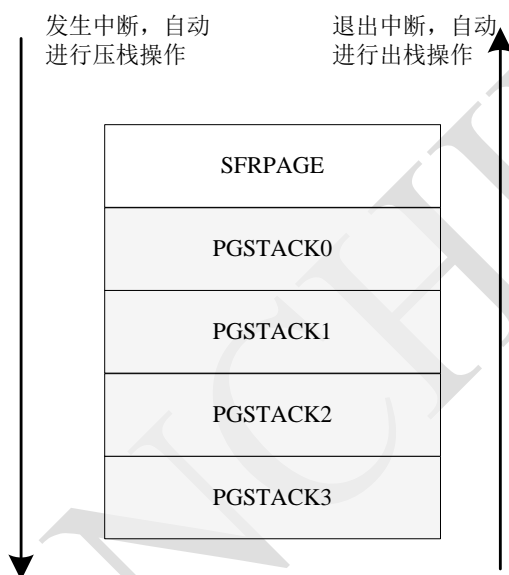


Figure 54 SFR PAGE 堆栈

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 121 列出了 JMT1808R 各模块对应的 PAGE 号：

Table 121 各模块对应的 PAGE 号

PAGE0	PAGE1	PAGE2	PAGE3	PAGE4
Timer0/1	ADC0	GPIO	CRM	PWM0
Timer2	ADC1	PAGE 堆栈	DSP	PWM1
UART0			CORDIC	
UART1				
SPI				
I2C				
PMU				
WDT				

PAGE0	PAGE1	PAGE2	PAGE3	PAGE4
RTC				
CMP				
FLASH				

Table 123、Table 124、Table 125、Table 126 和 Table 127 分别列出了 PAGE0、PAGE1、PAGE2、PAGE3 和 PAGE4 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 122 所示。

Table 122 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 123 特殊功能寄存器(PAGE0)

PAGE0								
F8H	IP2	FLSC	FLSK	FLSM				
F0H	B	S1CON	S1BUF	S1RELL	S1RELH	S1IRCON		
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		

PAGE0								
D0H	PSW	T2CON	RL2	RH2	TL2	TH2	T2PSC	
C8H	P4	SPICON	SPIIE	SPIF	SPIBR	SPIBUF		
C0H	IEN2							RTCIF
B8H	IEN1	AL1MIN	AL1HOUR	AL1DAY	AL1MON	RTCADJ0	RTCADJ1	RTCIE
B0H	P3	RTCDAAY	RTCMON	RTCYEAR	AL0SEC	AL0MIN	AL0HOUR	AL0WEEKS
A8H	IEN0	RTCCON0	RTCCON1	RTCTMC	RTCSEC	RTCMIN	RTCHOUR	RTCWEEK
A0H	P2	PMUCTR						
98H	S0CON	S0BUF	S0RELL	S0RELH	S0IRCON			
90H	P1	CMPEN	CMPINT	CMPFLAG	CMPFT	CMPVDD	CMPREF	CMPTST
88H	TCON	TMOD	TL0	TL1	TH0	TH1	TPSC	
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

Table 124 特殊功能寄存器(PAGE1)

PAGE1								
F8H	IP2							
F0H	B							
E8H	IP1							
E0H	ACC	ADC1DF			ADC1PGAC	ADC1ETC	ADC1NQTD	ADC1SQTD
D8H	IP0	ADC1SR0H	ADC1SR1L	ADC1SR1H	ADC1SR2L	ADC1SR2H	ADC1SR3L	ADC1SR3H
D0H	PSW	ADC1NQ5	ADC1NRL	ADC1NRH	ADC1SQTC	ADC1SQ0	ADC1SQ1	ADC1SR0L
C8H	P4	ADC1QLC	ADC1NQTC	ADC1NQ0	ADC1NQ1	ADC1NQ2	ADC1NQ3	ADC1NQ4
C0H	IEN2	ADC1IE	ADC1IF	ADC1WDC	ADC1HTL	ADC1HTH	ADC1LTL	ADC1LTH
B8H	IEN1							
B0H	P3							
A8H	IEN0	ADC0DF	ADC0DMC	ADC0JC	ADC0PGAC	ADC0ETC	ADC0NQTD	ADC0SQTD
A0H	P2	ADC0SR0H	ADC0SR1L	ADC0SR1H	ADC0SR2L	ADC0SR2H	ADC0SR3L	ADC0SR3H
98H	ADC1CON	ADC0NQ5	ADC0NRL	ADC0NRH	ADC0SQTC	ADC0SQ0	ADC0SQ1	ADC0SR0L
90H	P1	ADC0QLC	ADC0NQTC	ADC0NQ0	ADC0NQ1	ADC0NQ2	ADC0NQ3	ADC0NQ4
88H	ADC0CON	ADC0IE	ADC0IF	ADC0WDC	ADC0HTL	ADC0HTH	ADC0LTL	ADC0LTH
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

Table 125 特殊功能寄存器(PAGE2)

PAGE2								
F8H	IP2	PAIE	PBIE	PCIE	PDIE	PEIE	PGIE	
F0H	B	PWMOSEL	PWMOSEL0	PWMOSEL1	PWMOSEL2	PWMOMUX0	PWMOMUX1	PWMOMUX2
E8H	IP1	INT1SEL3	PWMBKS	DBGCON0	DBGCON1			
E0H	ACC	INT1PE	INT1NE	INT1EN	INT1CON	INT1SEL0	INT1SEL1	INT1SEL2
D8H	IP0	NRSTFQS	WKUPF	WKUPEN	INT0PE	INT0NE	INT0EN	INT0CON
D0H	PSW	PADR	PBDR	PCDR	PDDR	PEDR	PGDR	AIOEN1
C8H	P4	PASR	PBSR	PCSR	PDSR	PESR	PGSR	AIOEN0
C0H	IEN2	PG0CTRL	PG1CTRL	PA7CTRL	PB7CTRL	PC7CTRL	PD7CTRL	PE7CTRL
B8H	IEN1	PE0CTRL	PE1CTRL	PE2CTRL	PE3CTRL	PE4CTRL	PE5CTRL	PE6CTRL
B0H	P3	PD0CTRL	PD1CTRL	PD2CTRL	PD3CTRL	PD4CTRL	PD5CTRL	PD6CTRL
A8H	IEN0	PC0CTRL	PC1CTRL	PC2CTRL	PC3CTRL	PC4CTRL	PC5CTRL	PC6CTRL
A0H	P2	PB0CTRL	PB1CTRL	PB2CTRL	PB3CTRL	PB4CTRL	PB5CTRL	PB6CTRL
98H	P6	PA0CTRL	PA1CTRL	PA2CTRL	PA3CTRL	PA4CTRL	PA5CTRL	PA6CTRL
90H	P1	PGSTACK0	PGSTACK1	PGSTACK2	PGSTACK3			
88H	P5							
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

Table 126 特殊功能寄存器(PAGE3)

PAGE3								
F8H	IP2	DSP_PSRH	DSP_PMRL	DSP_PMRH	DSP_LOOPL	DSP_LOOPH	DSP_LIRL	DSP_LIRH
F0H	B	DSP_A0H	DSP_A1L	DSP_A1H	DSP_A2	DSP_LNKRL	DSP_LNKRH	DSP_PSRL
E8H	IP1	DSP_R7L	DSP_R7H	DSP_DP0L	DSP_DP0H	DSP_DP1L	DSP_DP1H	DSP_A0L
E0H	ACC	DSP_R3H	DSP_R4L	DSP_R4H	DSP_R5L	DSP_R5H	DSP_R6L	DSP_R6H
D8H	IP0	DSP_R0L	DSP_R0H	DSP_R1L	DSP_R1H	DSP_R2L	DSP_R2H	DSP_R3L
D0H	PSW	DSP_STA	DSP_PCL	DSP_PCH	DSP_CFG			
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	CMP_CLK_DIV	GPIO_CLK_DIV					
A8H	IEN0	CLK_EN1	CLK_EN2	SPI_CLK_DIV	I2C_CLK_DIV	PWM0_CLK_DIV	PWM1_CLK_DIV	
A0H	P2			BANDGAP_CTRL	OSC_STABL	OSC_STABH	CLK_READY	OSCCEN

PAGE3								
98H		TRMVAL32K	CRMCTRL				OSCPD	CLKOUTCTRL
90H	P1	CLK_EN0		SWRST0	SWRST1	GLOBALRST	RSTFLAG	TRMVAL72M
88H			EXTCLK_DIVL	EXTCLK_DIVH	SYS_CLK_SEL	ADC_CLK_DIV	RTC_CLK_SEL	SYS_CLK_DIV
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	ADCTRIM

Table 127 特殊功能寄存器(PAGE4)

PAGE4								
F8H	IP2	TIM1_ISR2		TIM1_CONR0				
F0H	B	TIM1_EGR	TIM1_CMLMTRARY	TIM1_ISR0	TIM1_ISR1	TIM1_CNTH	TIM1_CNTH	TIM1_IER_NOCMP
E8H	IP1	TIM1_CC2RDL	TIM1_CC2RDH	TIM1_CC3RUL	TIM1_CC3RUH	TIM1_CC3RDL	TIM1_CC3RDH	TIM1_BRKC
E0H	ACC	TIM1_CC0RDH	TIM1_CC1RUL	TIM1_CC1RUH	TIM1_CC1RDL	TIM1_CC1RDH	TIM1_CC2RUL	TIM1_CC2RUH
D8H	IP0	TIM1_ARRH	TIM1_PSCL	TIM1_PSCH	TIM1_RCR	TIM1_CC0RUL	TIM1_CC0RUH	TIM1_CC0RDL
D0H	PSW	TIM1_CC1MR	TIM1_CC2MR	TIM1_CC3MR	TIM1_CCENR	TIM1_CCPS	TIM1_DTC	TIM1_ARRL
C8H	P4	TIM1_CONR1	TIM1_CONR2	TIM1_CONR3	TIM1_TGICR0	TIM1_TGICR1	TIM1_IER	TIM1_CC0MR
C0H	IEN2							
B8H	IEN1	TIM0_ISR2		TIM0_CONR0				
B0H	P3	TIM0_EGR	TIM0_CMLMTRARY	TIM0_ISR0	TIM0_ISR1	TIM0_CNTH	TIM0_CNTH	TIM0_IER_NOCMP
A8H	IEN0	TIM0_CC2RDL	TIM0_CC2RDH	TIM0_CC3RUL	TIM0_CC3RUH	TIM0_CC3RDL	TIM0_CC3RDH	TIM0_BRKC
A0H	P2	TIM0_CC0RDH	TIM0_CC1RUL	TIM0_CC1RUH	TIM0_CC1RDL	TIM0_CC1RDH	TIM0_CC2RUL	TIM0_CC2RUH
98H		TIM0_ARRH	TIM0_PSCL	TIM0_PSCH	TIM0_RCR	TIM0_CC0RUL	TIM0_CC0RUH	TIM0_CC0RDL
90H	P1	TIM0_CC1MR	TIM0_CC2MR	TIM0_CC3MR	TIM0_CCENR	TIM0_CCPS	TIM0_DTC	TIM0_ARRL
88H		TIM0_CONR1	TIM0_CONR2	TIM0_CONR3	TIM0_TGICR0	TIM0_TGICR1	TIM0_IER	TIM0_CC0MR
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 55 所示。

JMT018 DSP 程序存储器共享 JMT51 的两部分存储器：PM 和 FLASH，通过 16 位程序读总线进行访问。JMT018 DSP 通过程序总线对 PM 的寻址地址为 0x0000~0x01ff，对 FLASH 的寻址地址为 0x0200~0x3fff。

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）ADCx 的采样结果寄存器。具体地址见章节 13.3.2 和章节 17.15.2。

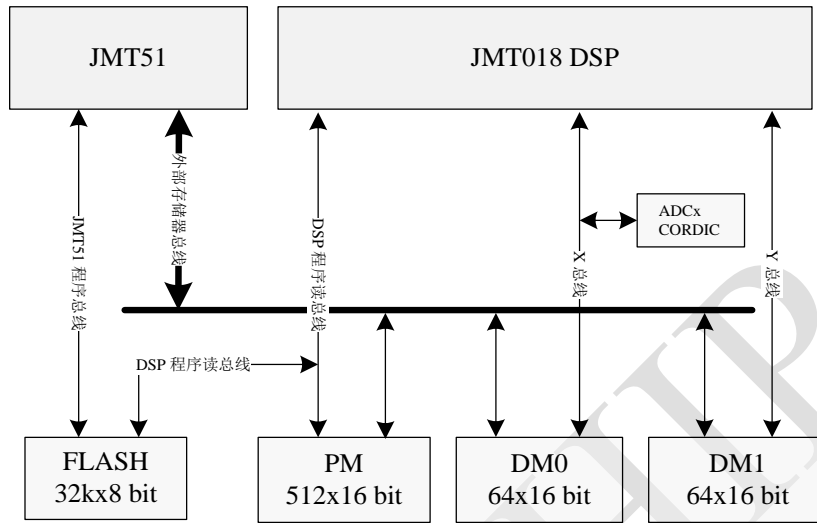


Figure 55 JMT018 DSP 存储器分布图

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（PM/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 优先权更高，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（PM/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。若 JMT51 和 JMT018 DSP 在同一时刻，访问不同块存储器，则没有该限制。JMT51 可通过读取 DSP 的工作状态寄存器，获取 DSP 是否在读取相同块存储器。例如，JMT018 DSP 正在对 PM 存储器进行读取操作，如果这时 JMT51 也需要读写 PM 存储器空间，则需要等到 DSP 释放访问，才能进行正常的读写，否则会出错；例如 JMT018 DSP 正在对 PM 存储器进行读取操作，如果这时 JMT51 需要读写 DM0 存储器空间，则无需等待，可直接访问 DM0 的地址空间。
- 若 JMT018 DSP 的程序运行于 FLASH 空间，建议在 JMT018 DSP 运行于 FLASH 的过程中，芯片不进入 IDLE 工作模式，待 JMT018 DSP 不再在 FLASH 中运行程

序，可以根据情况配置寄存器，使芯片进入各种低功耗模式。

- 若 JMT018 DSP 程序运行于 PM 存储器，则启动 JMT018 DSP 工作之后，芯片可以进入 IDLE 工作模式（也可以继续工作），若进入 IDLE 工作模式，待 JMT018 DSP 完成工作之后，JMT018 DSP 工作完成中断将唤醒 JMT51 继续工作。

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 128 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 129 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 130 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 131 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 256 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 132 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： <ul style="list-style-type: none"> ● ADD、ADDC 或 SUBB 指令引起符号位变化溢出 ● 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：

参数名	比特位	属性	复位值	描述
				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 133 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 134 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 135 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

参数名	比特位	属性	复位值	描述
				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 136 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
0x91	PGSTACK0	W/R	0x00	PAGE 堆栈寄存器 0	PAGE2
0x92	PGSTACK1	W/R	0x00	PAGE 堆栈寄存器 1	PAGE2
0x93	PGSTACK2	W/R	0x00	PAGE 堆栈寄存器 2	PAGE2
0x94	PGSTACK3	W/R	0x00	PAGE 堆栈寄存器 3	PAGE2

7.5.1 SFR PAGE 寄存器(SFRPAGE) [0x84]

Table 137 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

参数名	比特位	属性	复位值	描述
				001: PAGE1 010: PAGE2 011: PAGE3 100: PAGE4 其它: PAGE0
Reserved	7-3	R	0	保留

7.5.2 PAGE 堆栈使能寄存器(SFRPGEN) [0x85]

Table 138 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) [0x91]

Table 139 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) [0x92]

Table 140 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
Reserved	7-3	R	0	保留

7.5.5 PAGE 堆栈寄存器 2(PGSTACK2) [0x93]

Table 141 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) [0x94]

Table 142 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

IRONCHIP

8 Flash 存储器

8.1 概述

JMT1808R 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 3 个非易失性扇区（NVR0、NVR1 和 NVR2，每个 512 Bytes），其中 NVR0 部分空间用于存储芯片校准值，NVR1 部分空间用于存储芯片固件信息，NVR2 用户可自由使用。

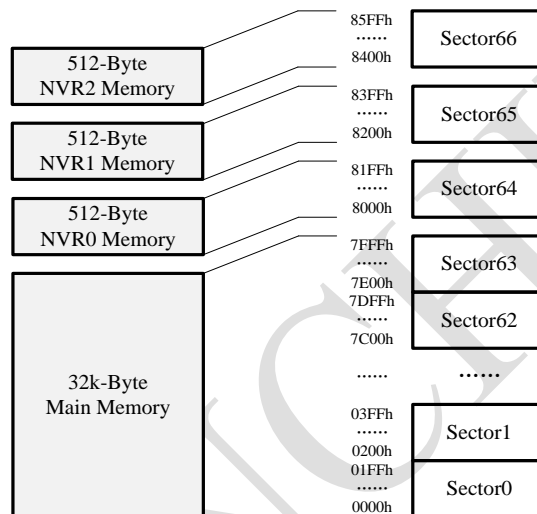


Figure 56 FLASH Memory

FLASH 特性如 Table 143 所示。

Table 143 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，NVR1 和 NVR2 读取
- 支持 MAIN，NVR0，NVR1 和 NVR2 扇区（SECTOR）擦除
- 支持 MAIN，NVR0，NVR1 和 NVR2 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
- “MOVC A,@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“MOVC A,@ 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;
```
- “MOVC A,@ 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, 具体列表如下。

Table 144 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 145 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 146 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: 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: 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, 具体说明如下。

Table 147 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 148 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:

参数名	比特位	属性	复位值	描述
				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 149 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 150 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:

参数名	比特位	属性	复位值	描述
				<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 151 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 152 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 实时时钟 (RTC)

10.1 概述

JMT1808R 的 RTC 具有以下特性:

- 提供实时时钟和日历, 包括年, 月, 日, 星期, 小时, 分钟和秒, 可以计时到 100 年
- 实时时钟寄存器 (RTCSEC, RTCMIN, RTCHOUR, RTCWEEK, RTCDAY, RTCMON, RTCYEAR) 自动检测配置是否合法
- 自动解决闰年问题
- 高精度数字时钟调校功能, 时钟精度为 $\pm 0.119\text{ppm}$
- 固定周期中断功能, 周期可为: 0.5 秒到 1 个月
- 闹钟 0 中断功能, 闹钟 0 时间寄存器包括: 日期 (ALOWEEKS), 小时 (ALOHOUR), 分钟 (ALOMIN), 秒 (ALOSEC)
- 闹钟 1 中断功能, 闹钟 1 时间寄存器包括: 月 (AL1MON), 日 (AL1DAY), 小时 (AL1HOUR), 分钟 (AL1MIN)
- 2 个可编程时标输出至管脚 (RTC_TM0 和 RTC_TM1), 其中时标的周期、输出极性可配置

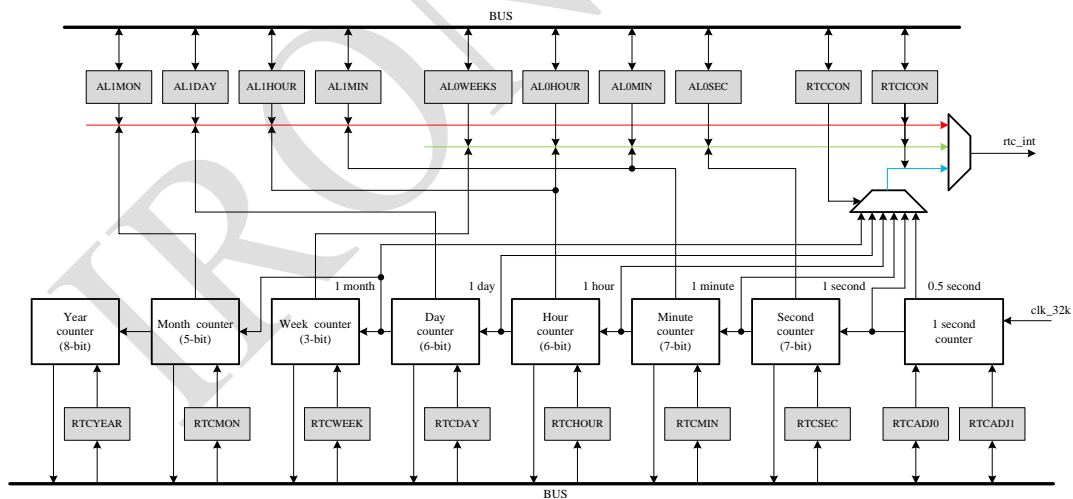


Figure 57 RTC 结构框图

10.2 实时时钟

RTC 提供实时时钟和日历, 包括年, 月, 日, 星期, 小时, 分钟和秒, 可以计时到 100 年, 时间计时范围如 Table 153 所示:

Table 153 RTC 计时范围

计时名称	计时值	描述
秒	00~59	
分钟	00~59	
小时	00~23	24 小时制
星期	00~06	00: 星期日; 01: 星期一; 02: 星期二; 03: 星期三; 04: 星期四; 05: 星期五; 06: 星期六
日	01~29	闰年二月
	01~28	非闰年二月
	01~30	四月, 六月, 九月, 十一月
	01~31	一月, 三月, 五月, 七月, 八月, 十月, 十二月
月	01~12	
年	00~99	能被 4 整除的年份是闰年

实时时钟初始化流程如 Figure 58 所示:

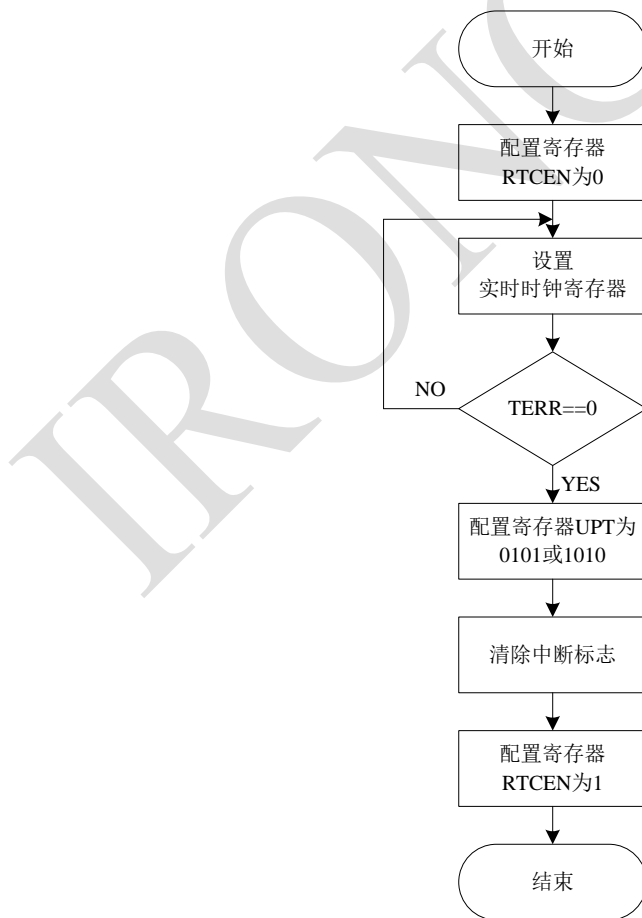


Figure 58 RTC 初始化流程

10.3 闹钟

RTC 具有两个闹钟功能，分别为：

- 闹钟 0 中断功能，可配置：日期（星期一到星期日），小时，分钟，秒
- 闹钟 1 中断功能，可配置：月，日，小时，分钟

闹钟 0 配置流程如 Figure 59 所示：

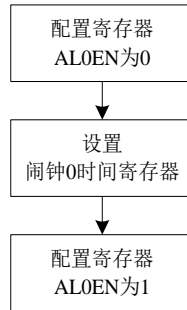


Figure 59 闹钟 0 配置流程

闹钟 1 配置流程如 Figure 60 所示：



Figure 60 闹钟 1 配置流程

10.4 RTC 时钟调校

RTC 时钟调校功能通过每 256 秒，对时钟进行一定量的增减来实现，并将调校值平均到每秒中。可以实现高达 $\pm 0.119\text{ppm}$ 的时钟精度，对应时间的误差为 ± 0.0103 秒/天。

通过设置 RTC 时钟调校寄存器 RTCADJ0 和 RTCADJ1（为了数据同步，需要先配置 RTCADJ1，后配置 RTCADJ0，校准值才能正常生效），调整需要调整的时钟数（ADJ）和调整方向（ADJDIR）。如果不需要调校，将 ADJ 设置为 0。

10.5 实时时钟寄存器更新

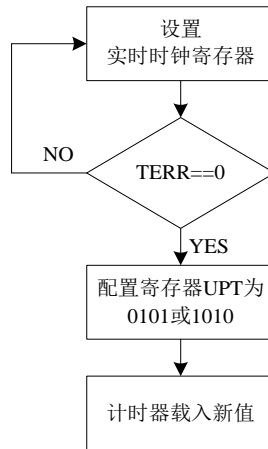


Figure 61 实时时钟寄存器更新流程

10.6 实时时钟寄存器读取

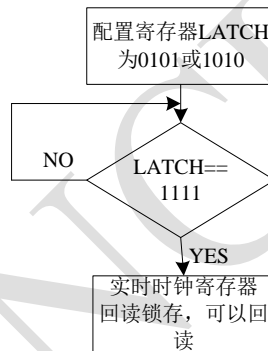


Figure 62 实时时钟寄存器读取流程

若未锁存过，实时时钟寄存器回读值均为 0；若未配置 LATCH 寄存器直接回读，实时时钟寄存器回读值为上次锁存内容。

10.7 RTC 中断

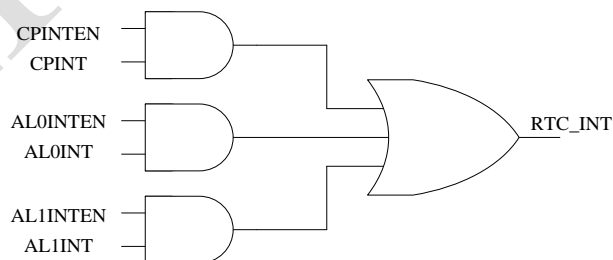


Figure 63 RTC 中断

RTC 有 3 个中断源：固定周期中断（CPINT）、闹钟 0 中断（AL0INT）和闹钟 1 中断（AL1INT）。

固定周期中断的中断周期可通过寄存器 RTCCON0.CPS 配置。

10.8 RTC 寄存器

RTC 模块有 22 个寄存器，对应的 SFRPAGE 为 0，具体说明如下。

Table 154 RTC 寄存器 (SFRPAGE=0x00)

地址	寄存器名	类型	复位值	功能描述
0xA9	RTCCON0	W/R	0x00	RTC 控制寄存器 0
0xAA	RTCCON1	W/R	0x00	RTC 控制寄存器 1
0xAB	RTCTMC	W/R	0x00	RTC 时标控制寄存器
0xAC	RTCSEC	W/R	0x00	RTC 秒寄存器
0xAD	RTCMIN	W/R	0x00	RTC 分钟寄存器
0xAE	RTCHOUR	W/R	0x00	RTC 小时寄存器
0xAF	RTCWEEK	W/R	0x00	RTC 星期寄存器
0xB1	RTCDAY	W/R	0x00	RTC 日寄存器
0xB2	RTCMON	W/R	0x00	RTC 月寄存器
0xB3	RTCYEAR	W/R	0x00	RTC 年寄存器
0xB4	AL0SEC	W/R	0x00	闹钟 0 秒寄存器
0xB5	AL0MIN	W/R	0x00	闹钟 0 分钟寄存器
0xB6	AL0HOUR	W/R	0x00	闹钟 0 小时寄存器
0xB7	AL0WEEKS	W/R	0x00	闹钟 0 星期寄存器
0xB9	AL1MIN	W/R	0x00	闹钟 1 分钟寄存器
0xBA	AL1HOUR	W/R	0x00	闹钟 1 小时寄存器
0xBB	AL1DAY	W/R	0x01	闹钟 1 日寄存器
0xBC	AL1MON	W/R	0x01	闹钟 1 月寄存器
0xBD	RTCADJ0	W/R	0x00	RTC 时钟调校寄存器 0
0xBE	RTCADJ1	W/R	0x00	RTC 时钟调校寄存器 1
0xBF	RTCIE	W/R	0x00	RTC 中断使能寄存器
0xC7	RTCIF	W/R	0x00	RTC 中断标志寄存器

10.8.1 RTC 控制寄存器 0(RTCCON0) [0xA9]

Table 155 RTC 控制寄存器 0(RTCCON0)

参数名	比特位	属性	复位值	描述
RTCEN	0	W/R	0	RTC 计数使能控制： 0：不使能 1：使能

参数名	比特位	属性	复位值	描述
ALOEN	1	W/R	0	闹钟 0 使能控制： 0: 关闭 1: 打开
ALIEN	2	W/R	0	闹钟 1 使能控制： 0: 关闭 1: 打开
TERR	3	R	0	实时时钟寄存器配置合法标志： 0: 合法 1: 非法
CPS	6-4	W/R	0	固定周期中断选择： 000: 无固定周期 001: 0.5 秒 010: 1 秒 011: 1 分钟 100: 1 小时 101: 1 天 11x: 1 月
Reserved	7	R	0	保留

10.8.2 RTC 控制寄存器 1(RTCCON1) [0xAA]

Table 156 RTC 控制寄存器 1(RTCCON1)

参数名	比特位	属性	复位值	描述
UPT	3-0	W/R	0	实时时钟寄存器更新： 实时时钟寄存器配置完成后，写 0101 或 1010 产生更新脉冲； 当回读为 1111 后，表示配置的实时时钟寄存器合法，且更新完成
LATCH	7-4	W/R	0	实时时钟寄存器锁存： 写 0101 或 1010 产生锁存脉冲； 当回读为 1111 后，表示实时时钟寄存器锁存完成，可以回读实时时钟寄存器

10.8.3 RTC 时标控制寄存器(RTCTMC) [0xAB]

Table 157 RTC 时标控制寄存器(RTCTMC)

参数名	比特位	属性	复位值	描述
TM0S	2-0	W/R	0	RTC_TM0 时标类型选择: 000: 调校前 clk_32k 的 64 分频, 占空比 1:1 001: 秒切换信号, 脉宽 clk_32k 周期 010: 分切换信号, 脉宽 clk_32k 周期 011: 时切换信号, 脉宽 clk_32k 周期 100: 日切换信号, 脉宽 clk_32k 周期 101: 月切换信号, 脉宽 clk_32k 周期 110: 闹钟 0 匹配信号, 脉宽 clk_32k 周期 111: 闹钟 1 匹配信号, 脉宽 clk_32k 周期
TM0P	3	W/R	0	RTC_TM0 时标极性选择: 0: 正脉冲 1: 负脉冲
TM1S	6-4	W/R	0	RTC_TM1 时标类型选择: 000: 调校前 clk_32k 的 64 分频, 占空比 1:1 001: 秒切换信号, 脉宽 clk_32k 周期 010: 分切换信号, 脉宽 clk_32k 周期 011: 时切换信号, 脉宽 clk_32k 周期 100: 日切换信号, 脉宽 clk_32k 周期 101: 月切换信号, 脉宽 clk_32k 周期 110: 闹钟 0 匹配信号, 脉宽 clk_32k 周期 111: 闹钟 1 匹配信号, 脉宽 clk_32k 周期
TM1P	7	W/R	0	RTC_TM1 时标极性选择: 0: 正脉冲 1: 负脉冲

10.8.4 RTC 秒寄存器(RTCSEC) [0xAC]

Table 158 RTC 秒寄存器(RTCSEC)

参数名	比特位	属性	复位值	描述
SEC	6-0	W/R	0	RTC 秒寄存器, BCD 编码, 范围 00~59
Reserved	7	R	0	保留

10.8.5 RTC 分钟寄存器(RTCMIN) [0xAD]

Table 159 RTC 分钟寄存器(RTCMIN)

参数名	比特位	属性	复位值	描述
MIN	6-0	W/R	0	RTC 分钟寄存器, BCD 编码, 范围 00~59
Reserved	7	R	0	保留

10.8.6 RTC 小时寄存器(RTCHOUR) [0xAE]

Table 160 RTC 小时寄存器(RTCHOUR)

参数名	比特位	属性	复位值	描述
HOUR	5-0	W/R	0	RTC 小时寄存器, BCD 编码, 范围 00~23
Reserved	7-6	R	0	保留

10.8.7 RTC 星期寄存器(RTCWEEK) [0xAF]

Table 161 RTC 星期寄存器(RTCWEEK)

参数名	比特位	属性	复位值	描述
WEEK	2-0	W/R	000	RTC 星期寄存器, BCD 编码, 范围 0~6: 000: 星期日 001: 星期一 010: 星期二 011: 星期三 100: 星期四 101: 星期五 110: 星期六
Reserved	7-3	R	0	保留

10.8.8 RTC 日寄存器(RTCDAY) [0xB1]

Table 162 RTC 日寄存器(RTCDAY)

参数名	比特位	属性	复位值	描述
DAY	5-0	W/R	000000	RTC 日寄存器, BCD 编码, 范围 01~31
Reserved	7-6	R	0	保留

10.8.9 RTC 月寄存器(RTCMON) [0xB2]

Table 163 RTC 月寄存器(RTCMON)

参数名	比特位	属性	复位值	描述
MON	4-0	W/R	00000	RTC 月寄存器, BCD 编码, 范围 01~12

参数名	比特位	属性	复位值	描述
Reserved	7-5	R	0	保留

10.8.10 RTC 年寄存器(RTCYEAR) [0xB3]

Table 164 RTC 年寄存器(RTCYEAR)

参数名	比特位	属性	复位值	描述
YEAR	7-0	W/R	0	RTC 年寄存器, BCD 编码, 自动识别闰年, 范围 00~99

10.8.11 闹钟 0 秒寄存器(AL0SEC) [0xB4]

Table 165 闹钟 0 秒寄存器(AL0SEC)

参数名	比特位	属性	复位值	描述
AL0SEC	6-0	W/R	0	闹钟 0 秒寄存器, BCD 编码, 范围 00~59
Reserved	7	R	0	保留

10.8.12 闹钟 0 分钟寄存器(AL0MIN) [0xB5]

Table 166 闹钟 0 分钟寄存器(AL0MIN)

参数名	比特位	属性	复位值	描述
AL0MIN	6-0	W/R	0	闹钟 0 分钟寄存器, BCD 编码, 范围 00~59
Reserved	7	R	0	保留

10.8.13 闹钟 0 小时寄存器(AL0HOUR) [0xB6]

Table 167 闹钟 0 小时寄存器(AL0HOUR)

参数名	比特位	属性	复位值	描述
AL0HOUR	5-0	W/R	0	闹钟 0 小时寄存器, BCD 编码, 范围 00~23
Reserved	7-6	R	0	保留

10.8.14 闹钟 0 星期寄存器(AL0WEEKS) [0xB7]

Table 168 闹钟 0 星期寄存器(AL0WEEKS)

参数名	比特位	属性	复位值	描述
AL0WS0	0	W/R	0	闹钟 0 星期日是否有效: 0: 无效 1: 有效
AL0WS1	1	W/R	0	闹钟 0 星期一是否有效: 0: 无效

参数名	比特位	属性	复位值	描述
				1: 有效
ALOWS2	2	W/R	0	闹钟 0 星期二是否有效: 0: 无效 1: 有效
ALOWS3	3	W/R	0	闹钟 0 星期三是否有效: 0: 无效 1: 有效
ALOWS4	4	W/R	0	闹钟 0 星期四是否有效: 0: 无效 1: 有效
ALOWS5	5	W/R	0	闹钟 0 星期五是否有效: 0: 无效 1: 有效
ALOWS6	6	W/R	0	闹钟 0 星期六是否有效: 0: 无效 1: 有效
Reserved	7	R	0	保留

10.8.15 闹钟 1 分钟寄存器(AL1MIN) [0xB9]

Table 169 闹钟 1 分钟寄存器(AL1MIN)

参数名	比特位	属性	复位值	描述
AL1MIN	6-0	W/R	0	闹钟 1 分钟寄存器, BCD 编码, 范围 00~59
Reserved	7	R	0	保留

10.8.16 闹钟 1 小时寄存器(AL1HOUR) [0xBA]

Table 170 闹钟 1 小时寄存器(AL1HOUR)

参数名	比特位	属性	复位值	描述
AL1HOUR	5-0	W/R	0	闹钟 1 小时寄存器, BCD 编码, 范围 00~23
Reserved	7-6	R	0	保留

10.8.17 闹钟 1 日寄存器(AL1DAY) [0xBB]

Table 171 闹钟 1 日寄存器(AL1DAY)

参数名	比特位	属性	复位值	描述
AL1DAY	5-0	W/R	0	闹钟 1 日寄存器, BCD 编码, 范围 01~31
Reserved	7-6	R	0	保留

10.8.18 闹钟 1 月寄存器(AL1MON) [0xBC]

Table 172 闹钟 1 月寄存器(AL1MON)

参数名	比特位	属性	复位值	描述
AL1MON	4-0	W/R	0	闹钟 1 月寄存器, BCD 编码, 范围 01~12
Reserved	7-5	R	0	保留

10.8.19 RTC 时钟调校寄存器 0(RTCADJ0) [0xBD]

Table 173 RTC 时钟调校寄存器 0(RTCADJ0)

参数名	比特位	属性	复位值	描述
ADJ[7:0]	7-0	W/R	0	RTC 时钟调校值

10.8.20 RTC 时钟调校寄存器 1(RTCADJ1) [0xBE]

Table 174 RTC 时钟调校寄存器 1(RTCADJ1)

参数名	比特位	属性	复位值	描述
ADJ[11:8]	3-0	W/R	0	RTC 时钟调校值
Reserved	6-4	R	0	保留
ADJDIR	7	W/R	0	RTC 时钟调校方向

10.8.21 RTC 中断使能寄存器(RTCIE) [0xBF]

Table 175 RTC 中断使能寄存器(RTCIE)

参数名	比特位	属性	复位值	描述
CPINTEN	0	W/R	0	固定周期中断使能: 0: 关闭 1: 打开
ALOINTEN	1	W/R	0	闹钟 0 中断使能: 0: 关闭 1: 打开
ALIINTEN	2	W/R	0	闹钟 1 中断使能: 0: 关闭 1: 打开

参数名	比特位	属性	复位值	描述
Reserved	7-3	R	0	保留

10.8.22 RTC 中断标志寄存器(RTCIF) [0xC7]

Table 176 RTC 中断标志寄存器(RTCIF)

参数名	比特位	属性	复位值	描述
CPINT	0	W/R	0	固定周期中断标志： <ul style="list-style-type: none"> ● 当有固定周期中断申请时，该位置为 1 ● 软件写 0 清除 ● 不支持写 1 软中断
ALOINT	1	W/R	0	闹钟 0 中断标志： <ul style="list-style-type: none"> ● 当有闹钟 0 中断申请时，该位置为 1 ● 软件写 0 清除 ● 不支持写 1 软中断
ALIINT	2	W/R	0	闹钟 1 中断标志： <ul style="list-style-type: none"> ● 当有闹钟 1 中断申请时，该位置为 1 ● 软件写 0 清除 ● 不支持写 1 软中断
Reserved	7-3	R	0	保留

11 定时器/计数器 (TIMER)

11.1 概述

JMT1808R 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

11.2 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。

11.3 Timer0 工作模式

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

11.3.1 模式 0(13 位定时器/计数器)

配置寄存器 TMOD 的标志位 $TMOD[1:0]^2=00$, 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 组成一个 13 位的计数器, 当 Timer0 溢出时寄存器位 TFO 置位, 同时产生 Timer0 溢出中断。程

² TMOD[1:0]代表寄存器 TMOD 的低两位

序进入中断后，该位被自动清零。

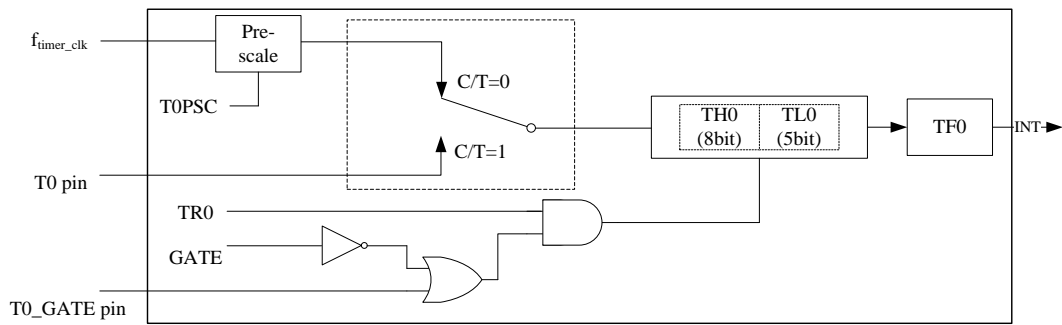


Figure 64 Timer0 模式 0 实现框图

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：

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

11.3.2 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。

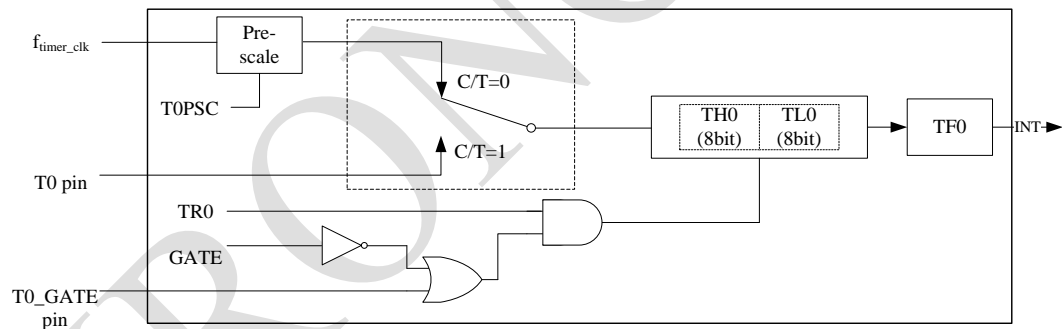


Figure 65 Timer0 模式 1 实现框图

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：

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

11.3.3 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。

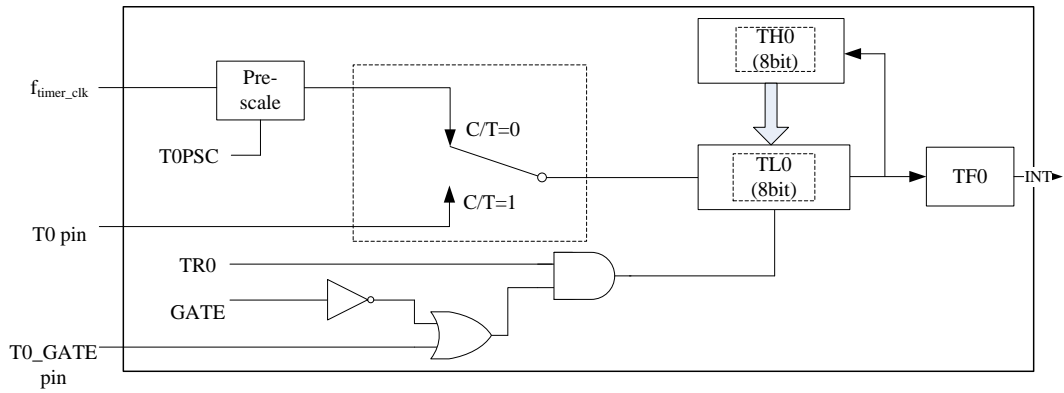


Figure 66 Timer0 模式 2 实现框图

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:

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

11.3.4 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。

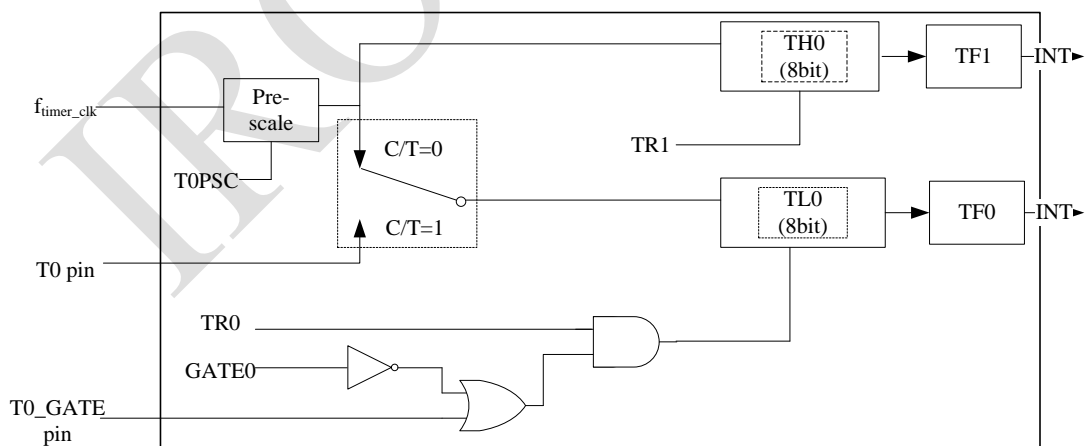


Figure 67 Timer0 模式 3 实现框图

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:

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

11.4 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。

11.4.1 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。

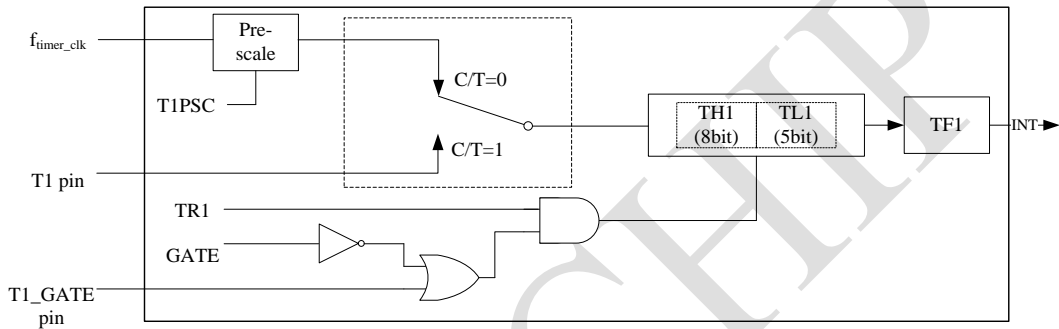


Figure 68 Timer1 模式 0 实现框图

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：

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

11.4.2 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。

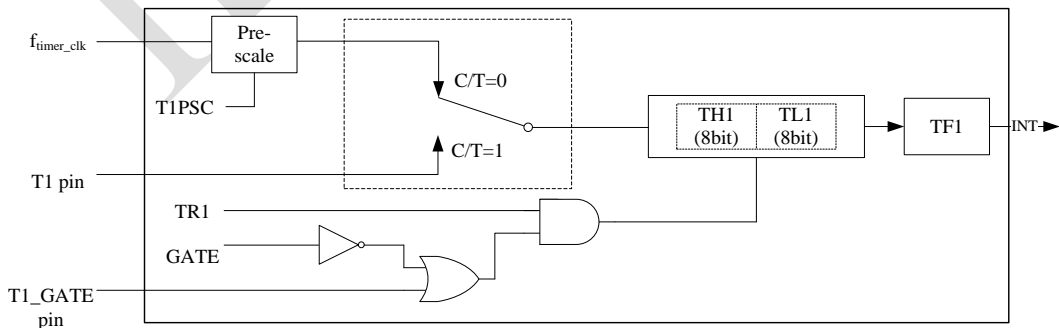


Figure 69 Timer1 模式 1 实现框图

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：

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

11.4.3 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。

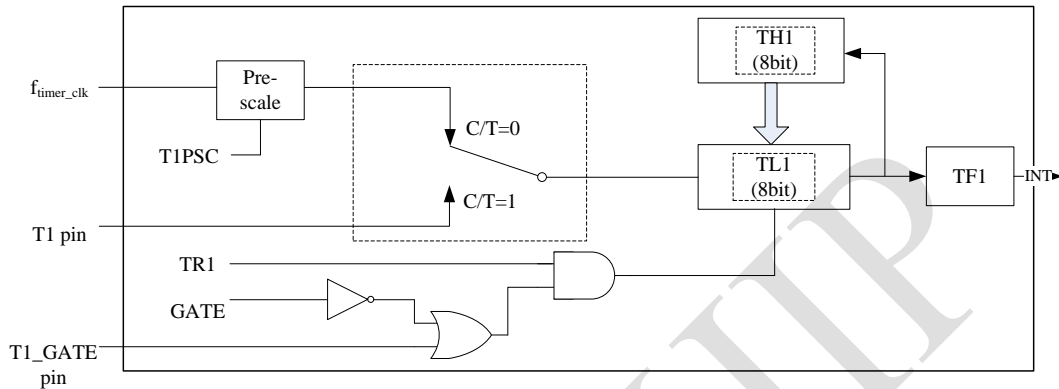


Figure 70 Timer1 模式 2 实现框图

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：

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

11.5 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。

11.5.1 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。

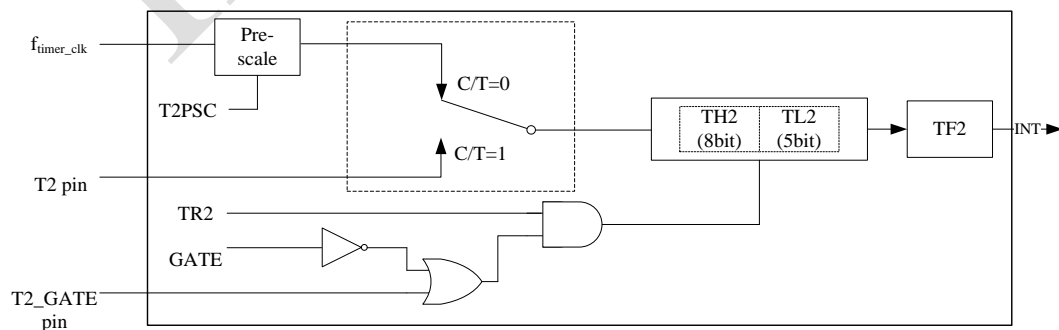


Figure 71 Timer2 模式 0 实现框图

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：

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

11.5.2 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。

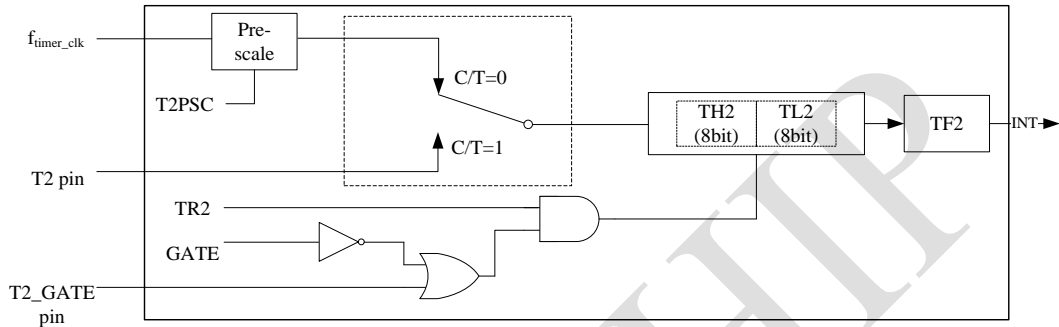


Figure 72 Timer2 模式 1 实现框图

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

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

11.5.3 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。

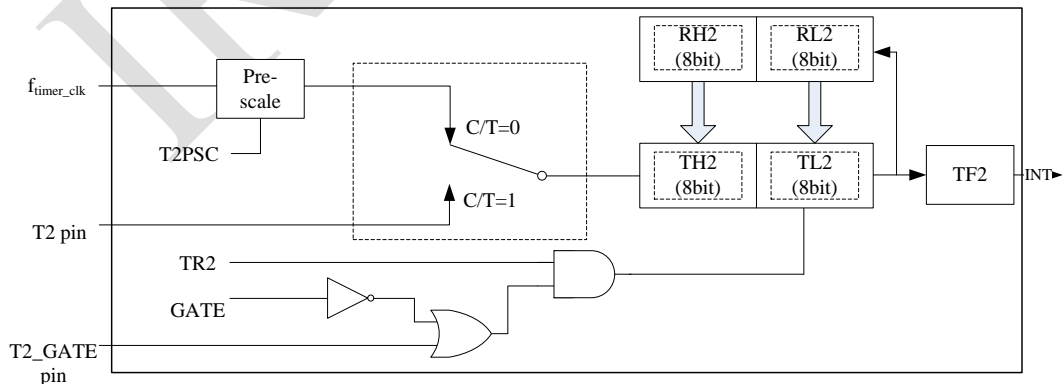


Figure 73 Timer2 模式 2 实现框图

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

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

11.5.4 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。

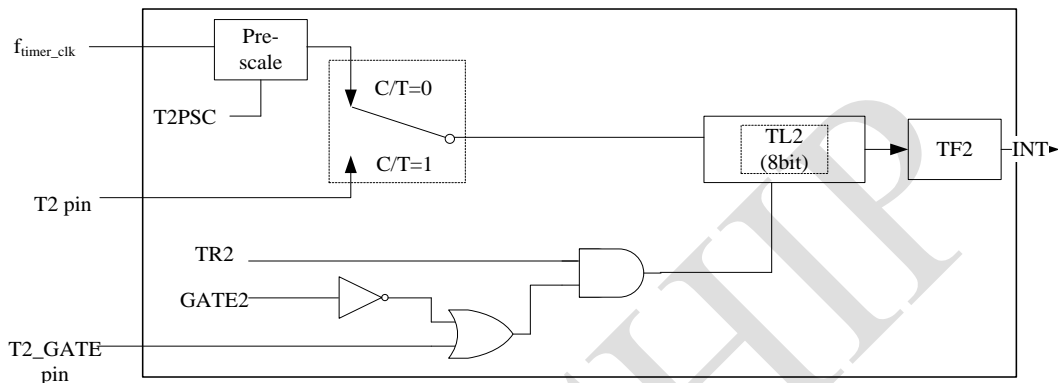


Figure 74 Timer2 模式 3 实现框图

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:

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

11.6 TIMER 寄存器

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

Table 177 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

地址	寄存器名	类型	复位值	功能描述
0xD4	TL2	W/R	0x00	Timer2 低 8 位
0xD5	TH2	W/R	0x00	Timer2 高 8 位
0xD6	T2PSC	W/R	0x00	Timer2 预分频控制器

11.6.1 Timer0/1 控制寄存器(TCON) [0x88]

Table 178 Timer0/1 控制寄存器(TCON)

参数名	比特位	属性	复位值	描述
UBSEL0	0	W/R	0	UART0 波特率产生选择位： 0: 使用 Timer 1 产生波特率 1: 使用 SORELL 和 SORELH 寄存器计数产生波特率
UBSEL1	1	W/R	0	UART1 波特率产生选择位： 0: 使用 Timer 1 产生波特率 1: 使用 SIRELL 和 SIRELH 寄存器计数产生波特率
SMOD0	2	W/R	0	UART0 双倍波特率使能控制位： 1: 使用双倍波特率 0: 正常模式
SMOD1	3	W/R	0	UART1 双倍波特率使能控制位： 1: 使用双倍波特率 0: 正常模式
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

参数名	比特位	属性	复位值	描述
				1: 溢出

11.6.2 Timer0/1 模式寄存器(TMOD) [0x89]

Table 179 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 180 Timer0/1 工作模式选择描述
M1	1	W/R	0	
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 180 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 180 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 TFO 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。

11.6.3 Timer0 低 8 位(TL0) [0x8A]

Table 181 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

11.6.4 Timer1 低 8 位(TL1) [0x8B]

Table 182 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

11.6.5 Timer0 高 8 位(TH0) [0x8C]

Table 183 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

11.6.6 Timer1 高 8 位(TH1) [0x8D]

Table 184 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

11.6.7 Timer0/1 预分频控制器(TPSC) [0x8E]

Table 185 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

11.6.8 Timer2 控制寄存器(T2CON) [0xD1]

Table 186 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 187 Timer2 工作模式选择
M1	1	W/R	0	
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 187 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。

11.6.9 Timer2 重载寄存器低 8 位(RL2) [0xD2]

Table 188 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

11.6.10 Timer2 重载寄存器高 8 位(RH2) [0xD3]

Table 189 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

11.6.11 Timer2 低 8 位(TL2) [0xD4]

Table 190 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

11.6.12 Timer2 高 8 位(TH2) [0xD5]

Table 191 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

11.6.13 Timer2 预分频控制器(T2PSC) [0xD6]

Table 192 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

12 高级定时器 (PWM)

12.1 概述

JMT1808R 包含两个完全相同的高级定时器 TIM0 (PWM0) 和 TIM1 (PWM1), 这两个高级定时器相互独立, 它们可以同时工作, 它们也相互连接, 可以用一个定时器驱动另一个定时器。

每个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, 也可以产生 7 路独立的 PWM 输出波形。

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。

12.2 特性

本定时器的特性:

- 16 位上、下和上/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。
- 在独立模式下可以输出 7 路独立的 PWM 波形。
- 本定时器包含 8 个 16 位的捕获/比较寄存器, 在互补 PWM 输出时, 中央计数模式

下向上计数过程计数器和 $TIMx_CCxRU^3$ 比较，向下计数过程计数器和 $TIMx_CCxRD$ 比较。

- 在独立模式下，前 6 路独立的 PWM 输出对应的比较寄存器为： $TIMx_CC0RU$ 、 $TIMx_CC0RD$ 、 $TIMx_CC1RU$ 、 $TIMx_CC1RD$ 、 $TIMx_CC2RU$ 和 $TIMx_CC2RD$ ，第 7 路 PWM 输出对应的比较寄存器为 $TIMx_CC3RU$ 和 $TIMx_CC3RD$ 。
- 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。
- 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

³ $TIMx$ 或 $PWMx$ 表示高级定时器 $TIM0$ 或高级定时器 $TIM1$ ，其中 $x=0, 1$ 。
 $CCxRU$ 或 $CCxRD$ 的 x 表示通道标号， $x=0, 1, 2, 3$ 。

12.3 框图

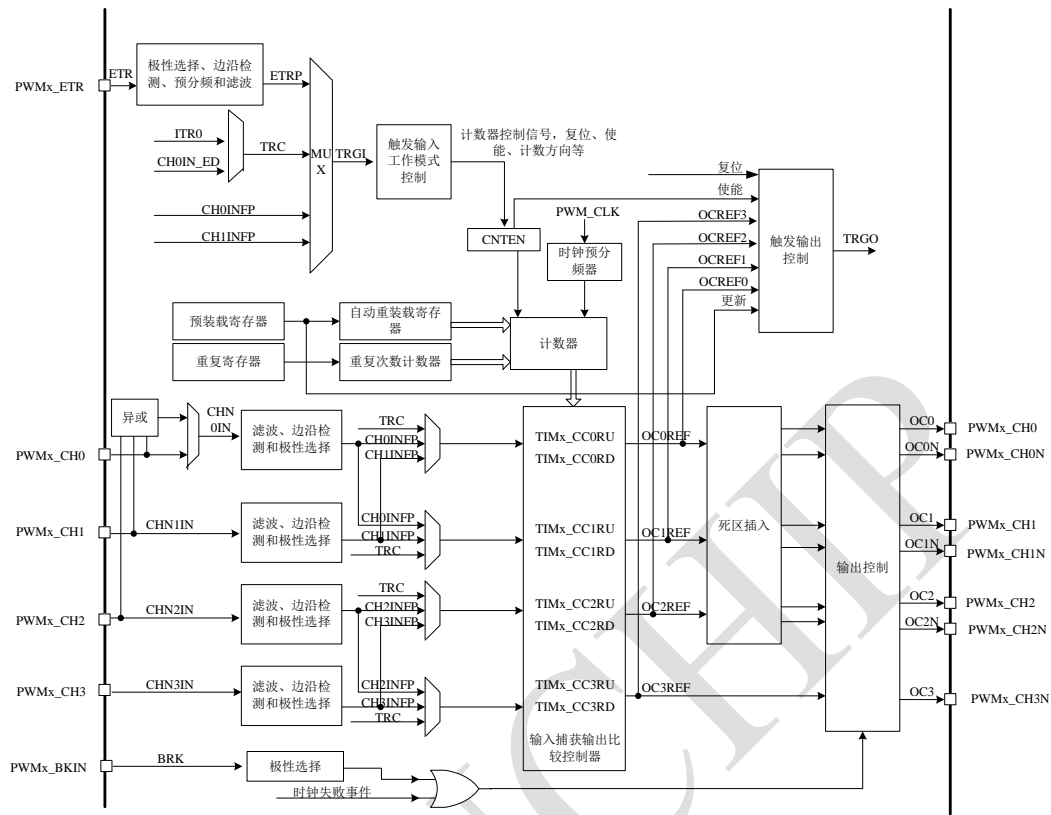


Figure 75 JMT1808R 高级定时器 (TIMx/PWMx) 框图

12.4 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 19.2~19.5 节。

当高级定时器配置为输入捕获功能时（寄存器 TIMx_CCxMR⁴的 CCxMS 位配置为 01/10/11），外部信号通过 PWMx_CHx⁵或 PWMx_ETR 通道输入，刹车信号通过 PWMx_BKIN 通道输入。

当高级定时器配置为输出比较功能时（寄存器 TIMx_CCxMR 的 CCxMS 位配置为 00），高级定时器的输出信号通过 PWMx_CHx 和 PWMx_CHxN 输出。

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

12.5 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。

⁴TIMx 中的 x 表示高级定时器 0 或高级定时器 1；CCxMR 中的 x 比较通道标号，x=0, 1, 2, 3。

⁵PWMx 中的 x 表示高级定时器 0 或高级定时器 1；CHx 中的 x 比较通道标号，x=0, 1, 2, 3。

12.5.1 计数器

12.5.1.1 计数器的计数周期

高级定时器 TIMx 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（TIMx_ARR 是其预装载寄存器），根据 TIMx_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（TIMx_ARR）。

12.5.1.2 计数器的使能控制

计数器使能 CEN 受到寄存器 TIMx_CONR2 的 CNTEN 位和从模式控制器控制，具体控制方式如下：

- 当寄存器 TIMx_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 CEN 只受 TIMx_CONR2 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 TIMx_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 TIMx_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 TIMx_TGICR0 的 SMS 位配置为其他值时，计数器使能 CEN 同时受 TIMx_CONR2 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 TIMx_CONR2 的 CNTEN 位置 1 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 TIMx_CONR2 的 CNTEN 位最后配置。

12.5.1.3 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 TIMx_PSCL 和 TIMx_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 76 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。

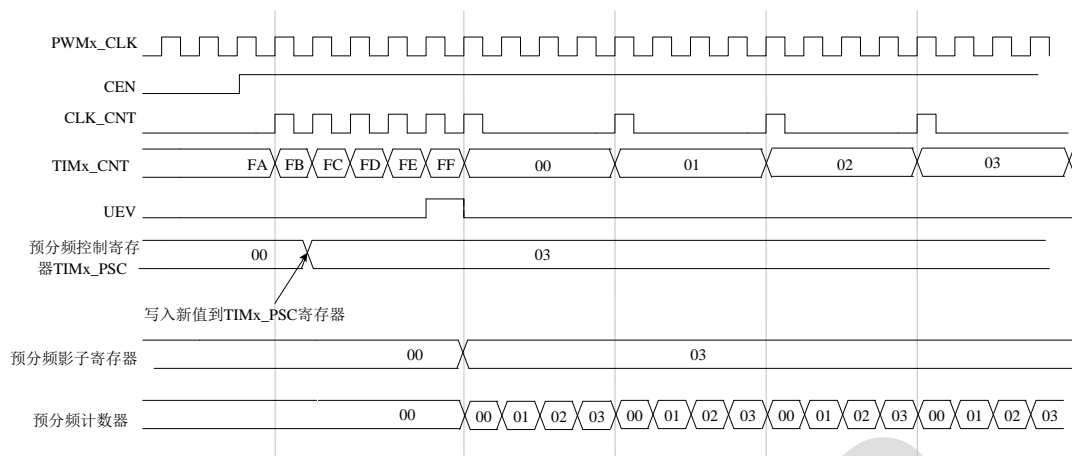


Figure 76 预分频器的参数由 1 分频变为 4 分频时，计数器的时序图

12.5.1.4 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 `TIMx_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 `TIMx_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 `TIMx_CONR2` 寄存器的 `CNTDIR` 位，计数器的方向由硬件自动设置。

当寄存器 `TIMx_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，`TIMx_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。

12.5.1.4.1 向上计数模式

当寄存器 `TIMx_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（`TIMx_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（`TIMx_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 `TIMx_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 `TIMx_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 `TIMx_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 TIMx_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（TIMx_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（TIMx_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 TIMx_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（TIMx_PSCL 和 TIMx_PSCH 寄存器）的值。

Figure 77 和 Figure 78 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 77 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 78 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。

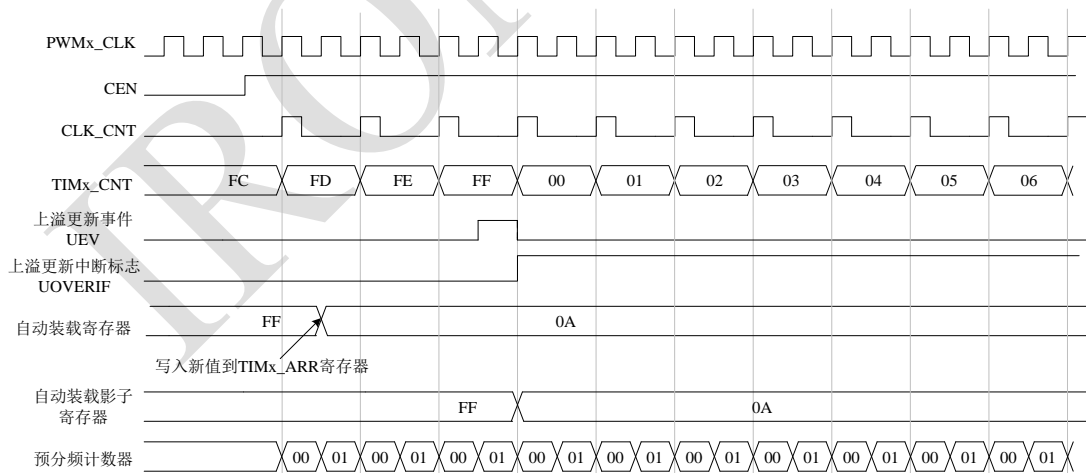


Figure 77 向上计数模式下，自动装载寄存器的预装载功能使能时计数器的时序图

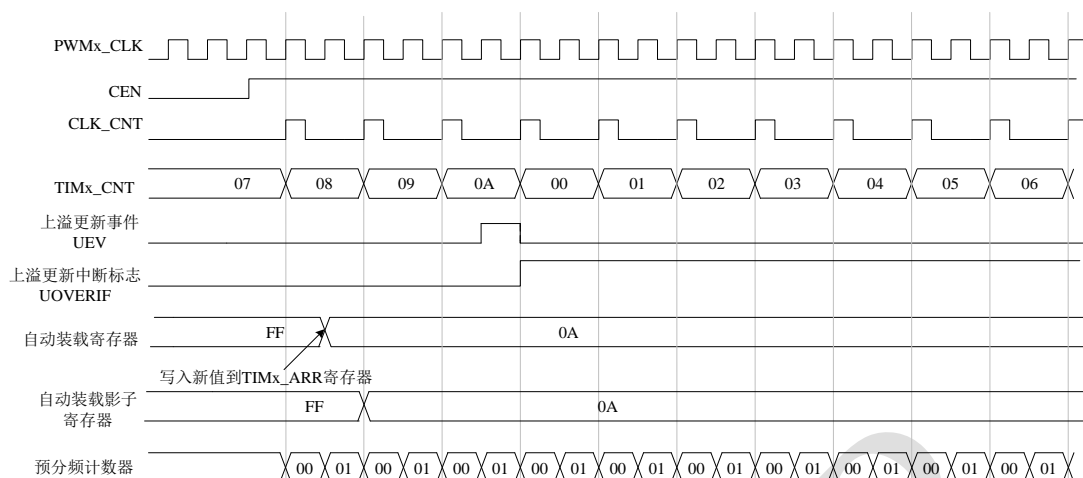


Figure 78 向上计数模式下，当自动装载寄存器的预装载功能不使能时计数器的时序图

12.5.1.4.2 向下计数模式

当寄存器 `TIMx_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 `TIMx_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 `TIMx_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 `TIMx_ARR`。

更新事件产生的规则：

- 设置寄存器 `TIMx_CONR3` 中的 `DOWNUDIS` 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 `TIMx_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 `TIMx_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 `DOVERIF` 自动置 `1`，若下溢更新中断使能（`TIMx_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（`TIMx_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 TIMx_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(TIMx_PSCL 和 TIMx_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 79 为向下计数模式下计数器的时序图。

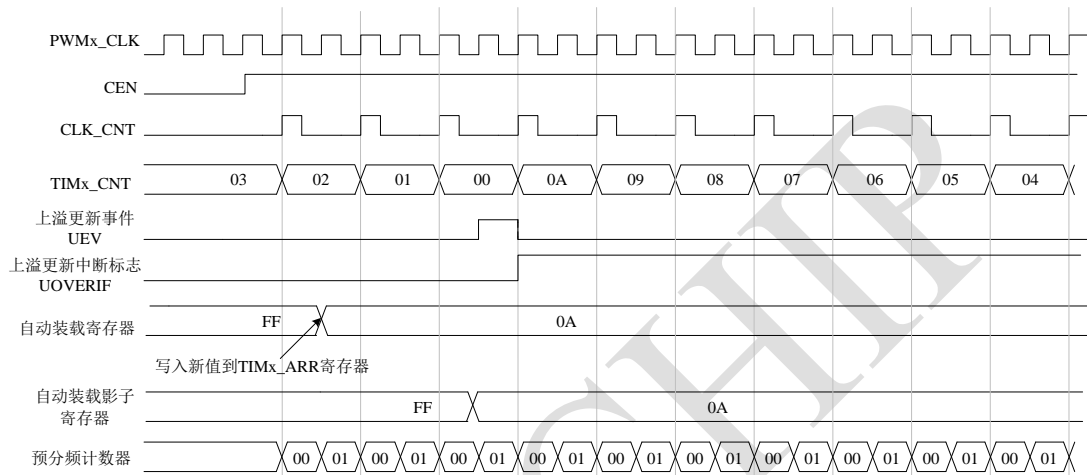


Figure 79 向下计数时计数器的时序图

12.5.1.4.3 中央对齐计数模式

寄存器 TIMx_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 TIMx_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 (TIMx_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 TIMx_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 TIMx_CONR3 中的 UPOUDIS 位为 1，可以禁止上溢更新事件的生产，设置寄存器 TIMx_CONR3 中的 DOWNOUDIS 位为 1，可以禁止下溢更新事件的生产。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 TIMx_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 TIMx_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（TIMx_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（TIMx_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（TIMx_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 TIMx_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（TIMx_PSCL 和 TIMx_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，因此下一个周期将是预期的值。Figure 80 和 Figure 81 为中央计数模式下计数器的时序。Figure 80 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 81 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。

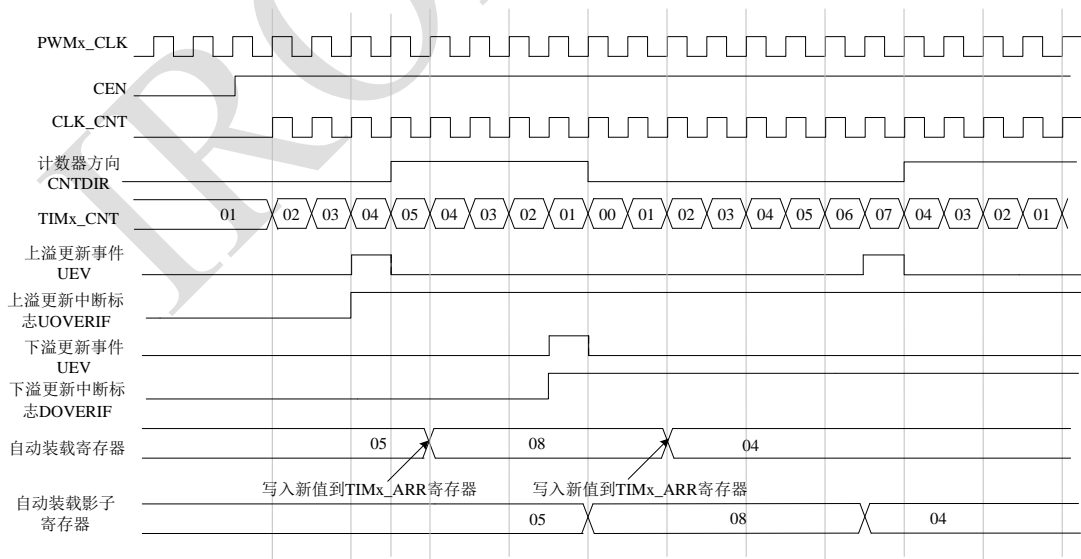


Figure 80 中央计数模式下，自动装载寄存器的预装载功能使能时计数器的时序图

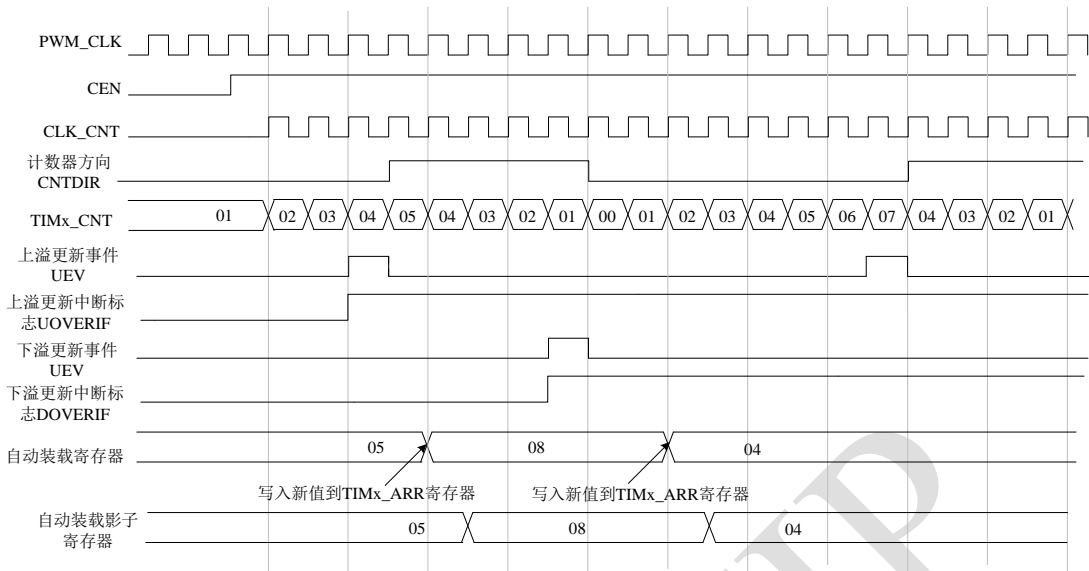


Figure 81 中央计数模式下，自动装载寄存器的预装载功能不使能时计数器的时序图

12.5.1.5 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 TIMx_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 TIMx_RCR 的内容被重新载入到周期计数器。

Figure 82 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 TIMx_CRC 配置为 0 时，每当计数器溢出时产生更新事件，当寄存器 TIMx_CRC 配置为 1 时，计数器连续 2 次溢出时产生更新事件，当 TIMx_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 83 为向下计数模式下周期计数控制器对更新事件的控制。

Figure 84 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 85 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。

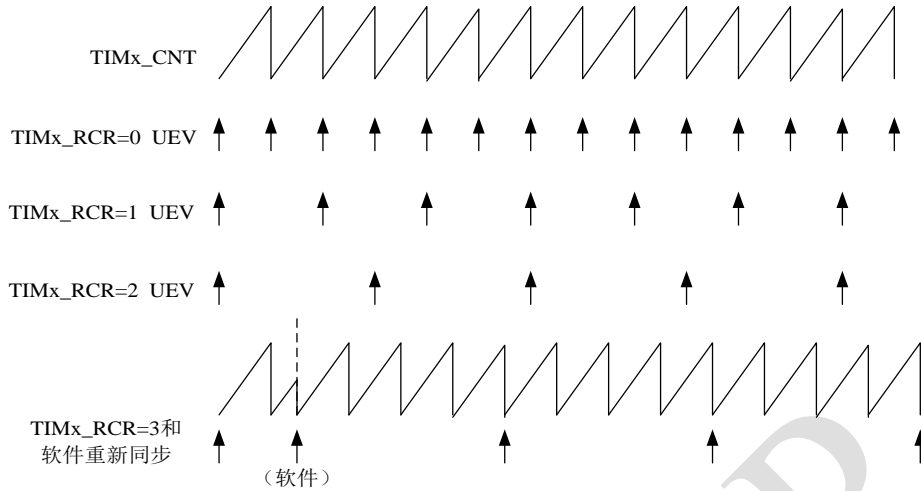


Figure 82 向上计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

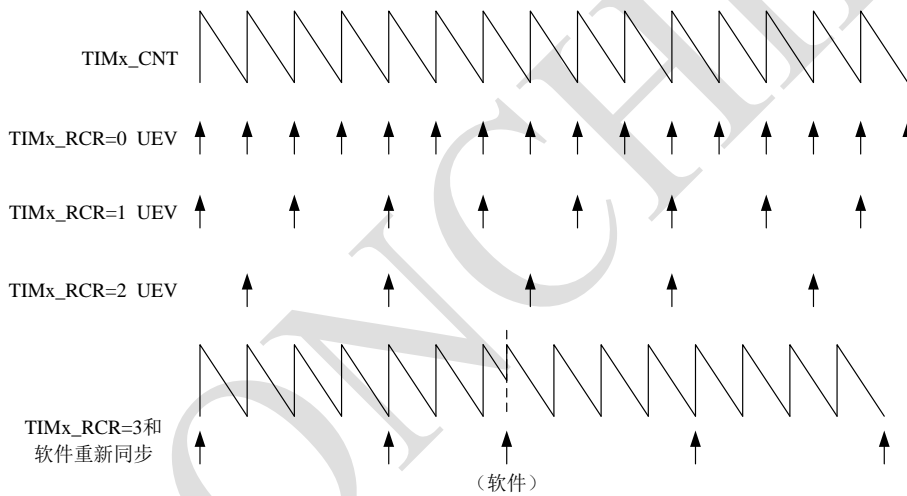


Figure 83 向下计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

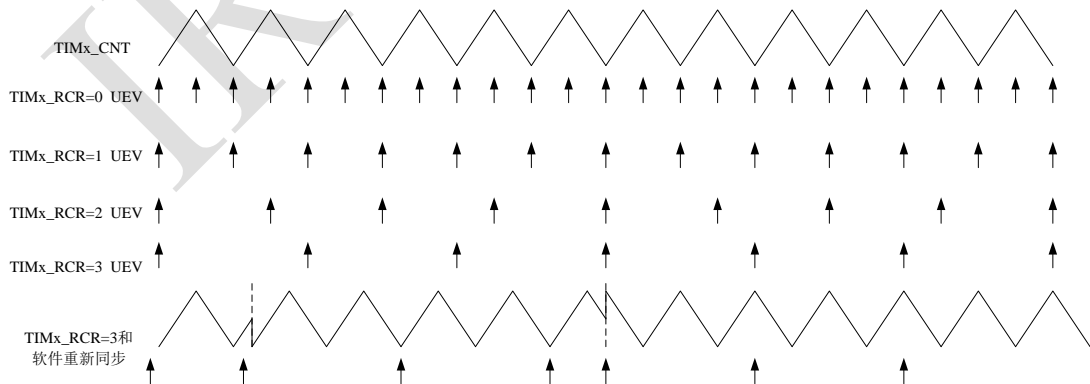


Figure 84 中央对齐计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

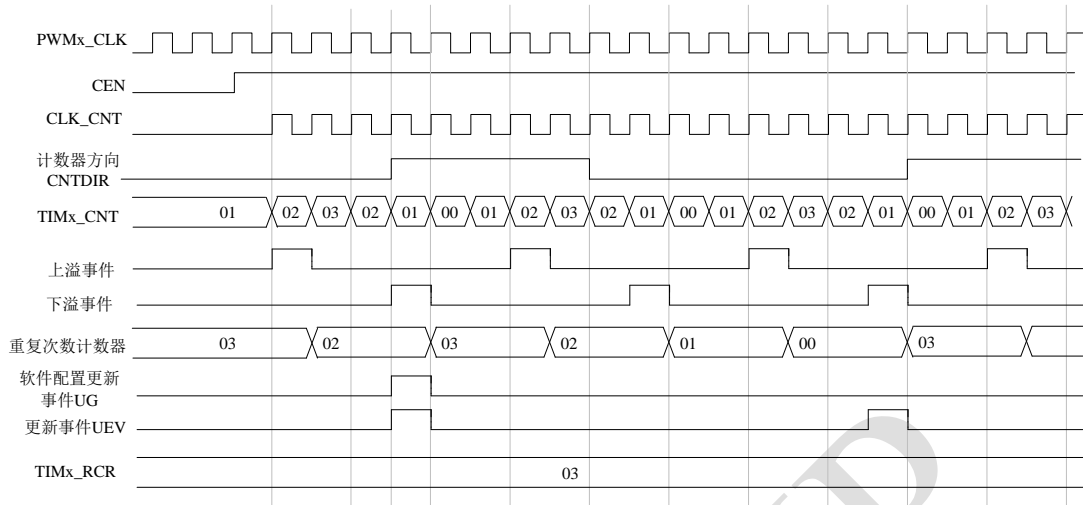


Figure 85 中央对齐模式下，更新事件的产生时序图

12.5.1.6 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 PWMx_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 PWMx_ETR 提供的输入信号 (外部时钟模式 2)。
- 内部触发输入 (ITR0)：在这种方式下，可以选用另一个定时器作为本定时器的预分频器。

12.5.1.6.1 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 PWMx_CLK_DIV 寄存器对时钟进行 1/2/4/8/16/32 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 TIMx_PSCL 和 TIMx_PSCH 实现。

将 TIMx_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 TIMx_CONR2 寄存器的 CNTEN 位配置为 1 时，计数器开始计数。

12.5.1.6.2 外部时钟源模式 1

当寄存器 TIMx_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 PWMx_CHx 输入信号的每个上升沿或下降沿计数，通过配置 TIMx_TGICR0 寄存器的 TRGS 位可以选择输入管脚 PWMx_CHx 的信号。Figure 86 为选择输入脚 PWMx_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，高级定时器首先对输入信号 CH0IN 进行滤波，通过配置寄存器 TIMx_CC0MR 的 ICF 位可以选择滤波长度和滤波时钟，滤波后得到信号 CH0INF，本例中不需要对外部信号进行滤波，配置 ICF 位为 0。

- 通过配置寄存器 TIMx_CC0MR 中的 CC0MS 位可以 CH0INF 和通道 CC1 的输入信号 CH1INF 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。
- 将滤波后的信号 CH0INF 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 TIMx_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 TIMx_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 TIMx_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 TIMx_CONR2 的 CNTEN 位为 1，启动计数器。

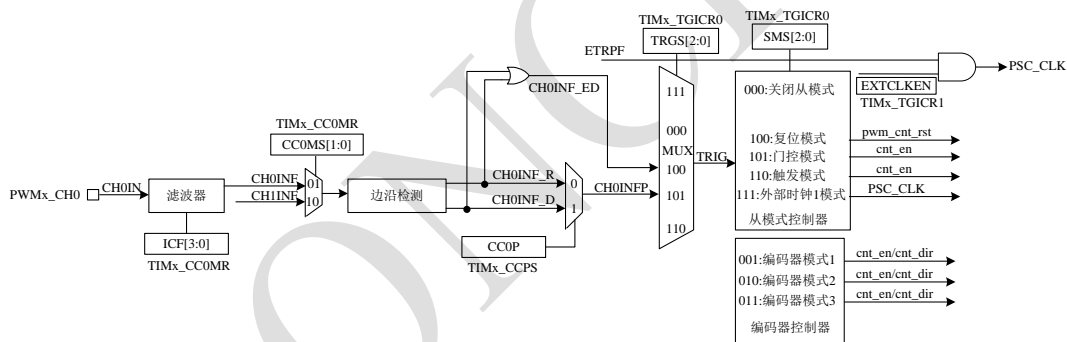


Figure 86 输入脚 PWMx_CH0 的信号作为外部时钟的连接例子

Figure 87 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，计数器记一次数。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。

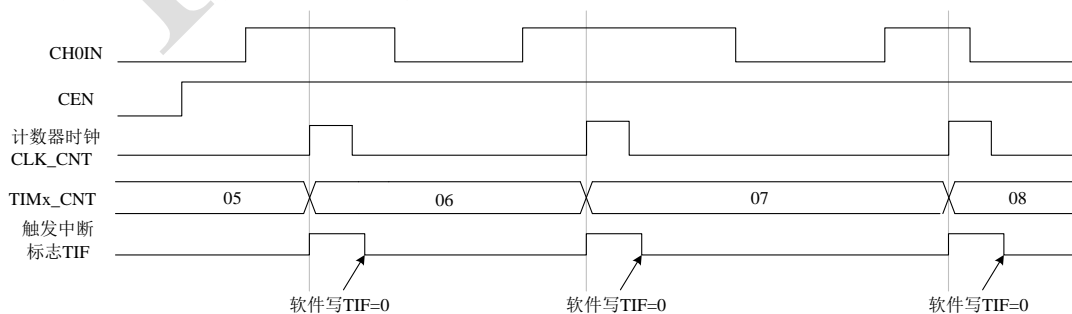


Figure 87 输入脚 PWMx_CH0 的信号作为外部时钟的时序图

12.5.1.6.3 外部时钟源模式 2

将寄存器 TIMx_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 PWMx_ETR 的每个上升沿或下降沿驱动计数。Figure 88 为外部触发信号 PWMx_ETR 作为外部时钟源的例子，具体配置为：

- 配置寄存器 TIMx_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 PWMx_ETRP。
- 配置寄存器 TIMx_TGICR1 的 EXTDIV 位，对 PWMx_ETRP 进行分频，得到信号 PWMx_ETRPDIV。
- 配置寄存器 TIMx_TGICR1 的 EXTFLT 位，对 PWMx_ETRPDIV 进行滤波，最后得到信号 PWMx_ETRPF。
- 配置寄存器 TIMx_TGICR1 的 EXTCLKEN 位为 1，将计数器时钟配置为外部时钟源模式 2。

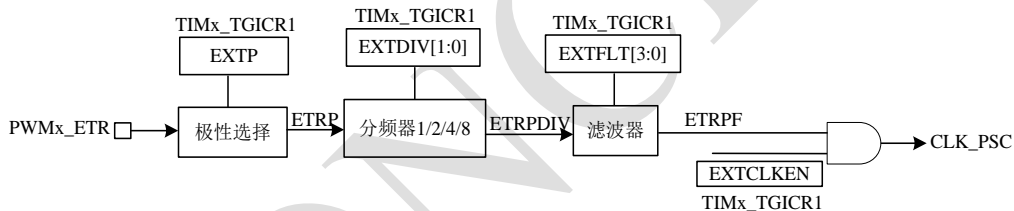


Figure 88 外部触发信号作为外部时钟源

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 PWMx_ETR 作为信号源效果是一样的，都是在 PWMx_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 89 为外部时钟模式 2 下计数器的时序图，外部信号 PWMx_ETR 出现一次上升沿，则计数器计数一次。PWMx_ETR 信号的上升沿和实际的时钟上升沿的延时取决于 PWMx_ETR 输入通道的重同步电路。

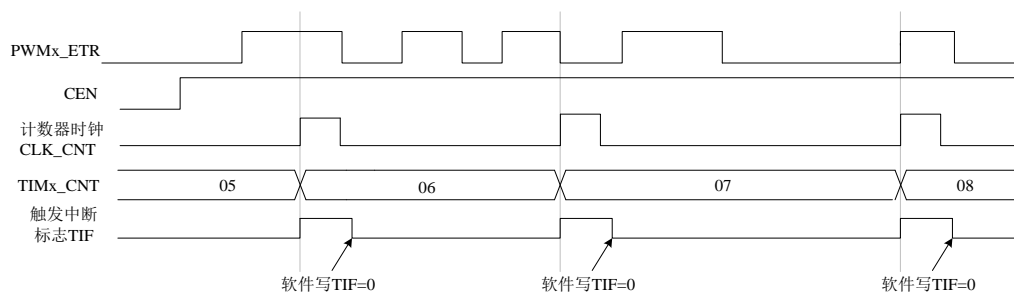


Figure 89 外部时钟模式 2 下计数器的时序图

12.5.1.6.4 内部触发输入

将 TIMx_TGICR0 的 SMS 位配置为 111，同时将 TRGS 位配置为 000，定时器的预分频器可以在 ITR0 信号的每个上升沿或下降沿计数，ITR0 来自于另一个高级定时器的 TRGO 信号，通过这种方式，一个定时器可以作为另一个定时器的预分频器。详见 12.5.7.1 节。

12.5.1.7 寄存器更新及安全级别设置

12.5.1.7.1 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。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；
- 在读/写寄存器时必须读/写低字节寄存器。

12.5.1.7.2 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：当向寄存器写入新值时，新值立即生效。

预装载模式：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，TIMx 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。

TIMx 中所有具有预装载功能的寄存器有：TIMx_PSC（无立即更新模式）、TIMx_ARR、TIMx_CCxRU、TIMx_CCxRD、TIMx_CCENR、TIMx_CCPS 和寄存器 TIMx_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四类：

- 只有预装载模式，由更新事件 UEV 更新寄存器，这些寄存器有：TIMx_PSCH 和 TIMx_PSCL。
- 由寄存器 TIMx_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：TIMx_ARRH 和 TIMx_ARRL。

- 由 TIMx_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: TIMx_CCxRUH、TIMx_CCxRUL、TIMx_CCxRDH、TIMx_CCxRDL。
- 由 TIMx_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 TIMx_CONR0 寄存器的 CCUS 位决定)寄存器,这些寄存器有: TIMx_CCENR、TIMx_CCPS 和 TIMx_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(TIMx_PSCH 和 TIMx_PSCL 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。

12.5.1.7.3 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 TIMx_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 TIMx_BRKC.LOCK 对寄存器的保护分为三个级别:

- 当寄存器 TIMx_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 TIMx_DTG 寄存器、TIMx_BRKC 的 BRKE、BRKP、AOE 位、TIMx_CONR1 寄存器。
- 当寄存器 TIMx_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 TIMx_CCPS 寄存器,不能写入 TIMx_BRKC 的 ROSS 和 IOSS 位。
- 当寄存器 TIMx_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 TIMx_CCxMR 的 OCxMS 和 OCxPEN 位。

12.5.2 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。

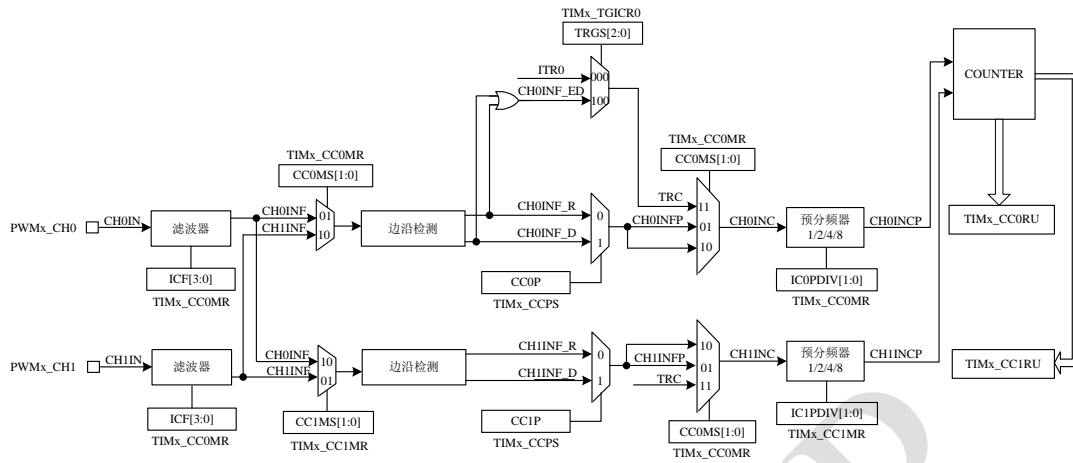


Figure 90 输入捕获通道

如 Figure 90 所示，当寄存器 TIMx_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 TIMx_CC0RU 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。
- 外部信号输入后，高级定时器首先对输入信号 CH0IN 进行滤波，通过配置寄存器 TIMx_CC0MR 的 ICF 位可以选择滤波长度和滤波时钟，滤波后得到信号 CH0INF。
- 通过配置寄存器 TIMx_CC0MR 中的 CC0MS 位可以选 CH0INF 或通道 CC1 的输入信号 CH1INF 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 TIMx_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 TIMx_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 CH0INFP。
- 通道 CC1 的输入信号经过滤波、边沿检测和边沿选择后可以得到信号 CH1INFP。
- 通过配置寄存器 TIMx_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 TIMx_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 TIMx_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 TIMx_CC0RU 中的捕获值时,先读取寄存器 TIMx_CC0RUH 再读取 TIMx_CC0RUL, Figure 91 为软件读取捕获寄存器 TIMx_CC0RU 的过程。

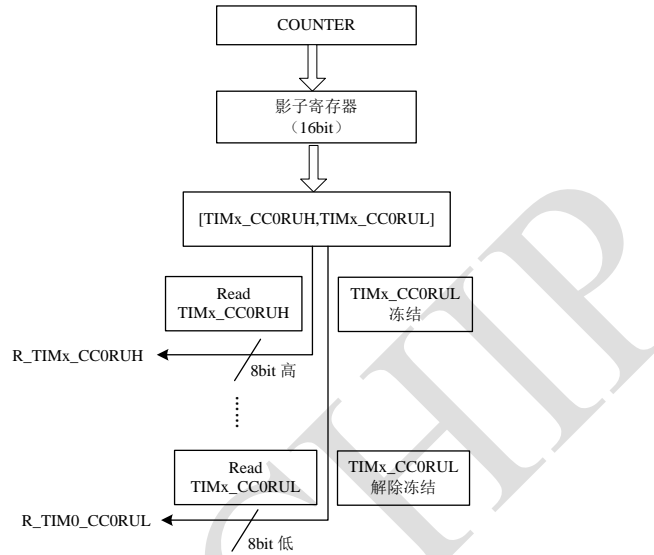


Figure 91 向软件读取捕获寄存器

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,硬件自动将捕获寄存器的预装载寄存器 TIMx_CC0RUH 和 TIMx_CC0RUL 冻结,等完成低 8 位数据读取后解除冻结,此时才可以再次发生捕获。因此,在读取捕获数据时必须先读取数据的高 8 位部分再读取低 8 位部分,否则会导致捕获寄存器被冻结,不可以再次捕获数据。

当产生捕获信号 CH0INCP 时,寄存器 TIMx_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 TIMx_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。

12.5.3 从模式控制

12.5.3.1 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 92 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 PWMx_ETR, 触发信号产生过程为:

- 配置寄存器 TIMx_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 TIMx_TGICR1 的 EXTDIV 位，对 EXTRP 进行分频，得到信号 PWMx_ETRPDIV。
- 配置寄存器 TIMx_TGICR1 的 EXTFLT 位，对 PWMx_ETRPDIV 进行滤波，最后得到信号 PWMx_ETRPF。
- PWMx_ETRPF 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 TIMx_TGICR1 的 EXTCLKEN 位配置为 1 时，外部时钟模式 2 打开，PWMx_ETRPF 也可作为计数器预分频器的时钟 PSC_CLK。
- 另一个高级定时器的输出信号 TIMx_TRGO。
- 通道 CC0 的双边沿信号 CH0IN_ED，产生过程见 12.5.2 部分说明。
- 通道 CC0 的单边沿信号 CH0INFP，产生过程见 12.5.2 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 12.5.2 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，TIMx_TRGO 和 CH0IN_ED 信号还可以作为捕获信号。

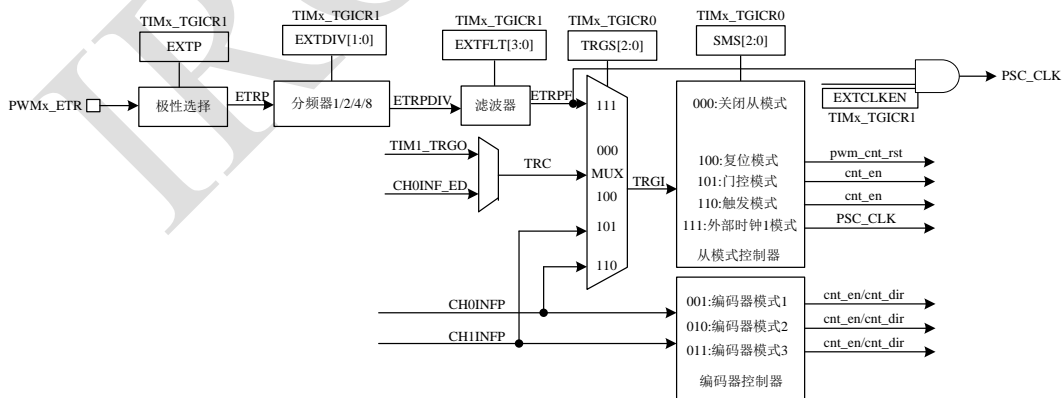


Figure 92 从模式控制器示意图

12.5.3.2 从模式控制

配置寄存器 TIMx_TGICR0 的 TRGS 位，从 ETRPF、TIM1_TRGO、CH0INF_ED、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 TIMx_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_x_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 TIMx_CONR2 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 TIMx_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 TIMx_CONR2 的 URC 位配置为 0，则寄存器 TIMx_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 TIMx_EGR 的 UG 事件；若 TIMx_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 TIMx_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 为无效平时，计数器停止计数（但不复位）。此时 TIMx_CONR2 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 TIMx_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 TIMx_CONR2 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 TIMx_CONR2 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 TIMx_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 TIMx_CONR2 的 CNTEN 位需要配置为 1。
- 外部时钟源模式 2：配置寄存器 TIMx_TGICR1 的 EXTCLKEN 位为 1，PWMx_ETR 的有效沿驱动计数器计数。此时 TIMx_CONR2 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 时一个脉冲信号，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 PWMx_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 PWMx_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 PWMx_ETR。

12.5.3.3 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 TIMx_CC0MR 的 IC0FS 位、TIMx_CCPS 的 CC0P 位，对输入的编码器信号进行滤波和极性选择，得到信号 CH0INFP。
- 根据实际需要配置寄存器 TIMx_CC1MR 的 IC1FS 位、TIMx_CCPS 的 CC1P 位，对输入的编码器信号进行滤波和极性选择，得到信号 CH1INFP。
- 配置寄存器 TIMx_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 TIMx_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 TIMx_CONR2 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 193 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 193 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 边沿计数	高	不计数	不计数	向上计数	向下计数
	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 TIMx_ARR 寄存器的自动装载值之间连续计数（根据 Table 193 中的方向，或者是 0 到 TIMx_ARR 计数，或者是 TIMx_ARR 到 0 计数），所以在开始计数之前必须配置 TIMx_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 93 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 93 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 PWMx_CH0 和 PWMx_CH2，并且将增量编码器的输出信号连接至 PWMx_CH0 和 PWMx_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 TIMx_CCENR 的 CC0EN 和 CC1EN 位为 1，使能通道 CC0 和 CC1 输入。
- 配置寄存器 TIMx_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 TIMx_CC0MR 的 IC0FS 位可以选择 CH0IN 的滤波的长度和滤波时钟，配置寄存器 TIMx_CC1MR 的 IC1FS 位可以选择 CH1IN 的滤波的长度和滤波时钟，本例中不对输入信号滤波，配置 TIMx_CC0MR. IC0FS=0、TIMx_CC1MR. IC1FS=0。

- 配置寄存器 `TIMx_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 `TIMx_CONR2` 的 `CNTEN` 位为 `1`, 使能计数器。

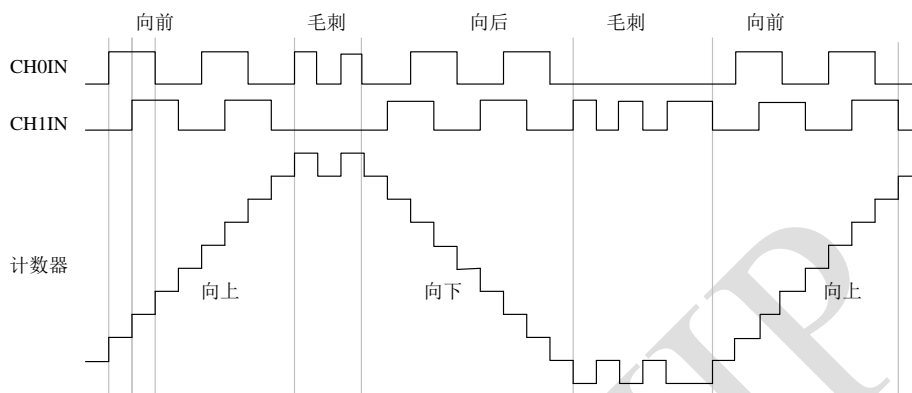


Figure 93 高级定时器作为增量编码接口的例子

12.5.3.4 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 `TIMx_ARR` 配置为最大值。
- 假设对输入的信号不需要滤波, 寄存器 `TIMx_CC0MR` 的 `IC0FS` 位配置为 `0000`。
- 配置寄存器 `TIMx_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 `TIMx_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 `TIMx_CC0MR` 的 `IC0PDIV` 位为 `00`。
- 配置寄存器 `TIMx_CCPS` 的 `CC1PS` 位为 `1`, 配置寄存器 `TIMx_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 `TIMx_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 `TIMx_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 `TIMx_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 `TIMx_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 `TIMx_CONR2` 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 94 所示。

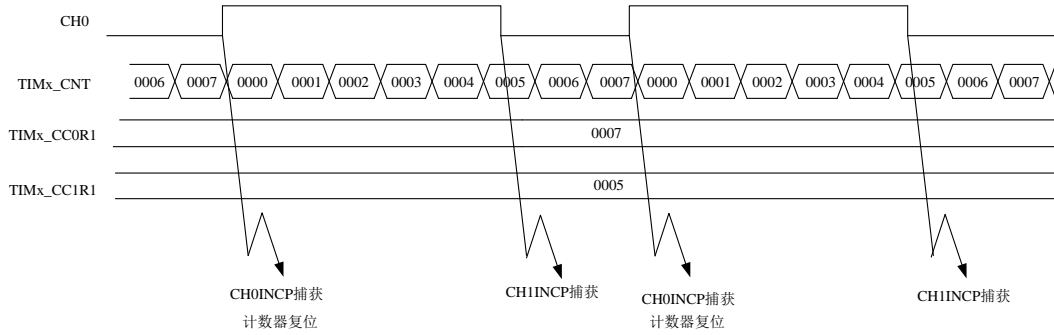


Figure 94 PWM 输入模式时序图

12.5.4 输出比较功能

将寄存器 TIMx_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 95 所示，通道 CC3 只能输出一路 PWM 波形，输出控制过程如 Figure 96 所示。

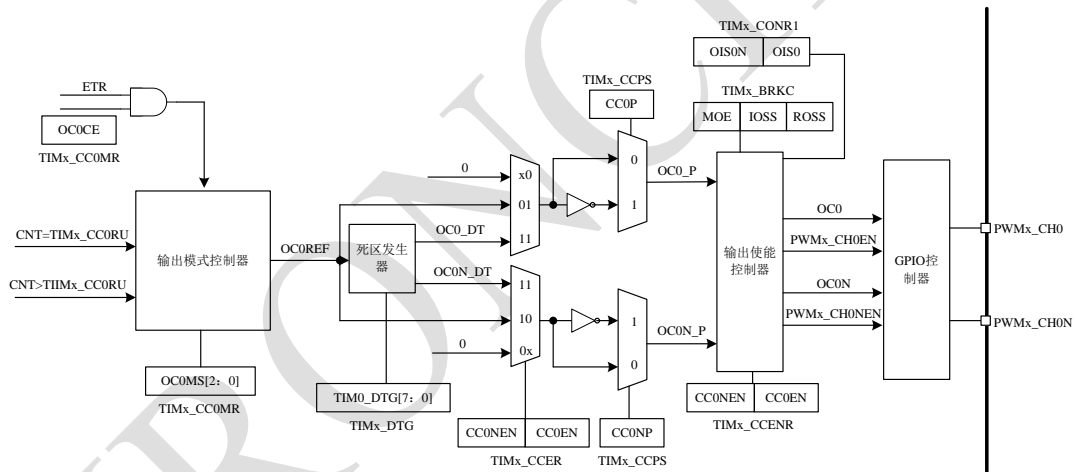


Figure 95 通道 CC0 输出控制

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 TIMx_CC0MR 的 CC0MS 位为 00，将 CC0 通道配置为输出功能。
- 配置寄存器 TIMx_CC0MR 的 OC0MS 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 12.5.4.1 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 TIMx_CC0MR 的 OC0CE 位配为 1 时，外部信号 PWMx_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 TIMx_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 TIMx_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 TIMx_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 TIMx_BRKC 中 MOE 位、IOSS、ROSS 位、寄存器 TIMx_CCER 中的 CC0ENN、CC0NEN、寄存器 TIMx_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 12.5.4.3 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 PWMx_CH0EN 和 PWMx_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 TIMx_CC0RU (或 TIMx_CC0RD) 比较成功, 则寄存器 TIMx_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。

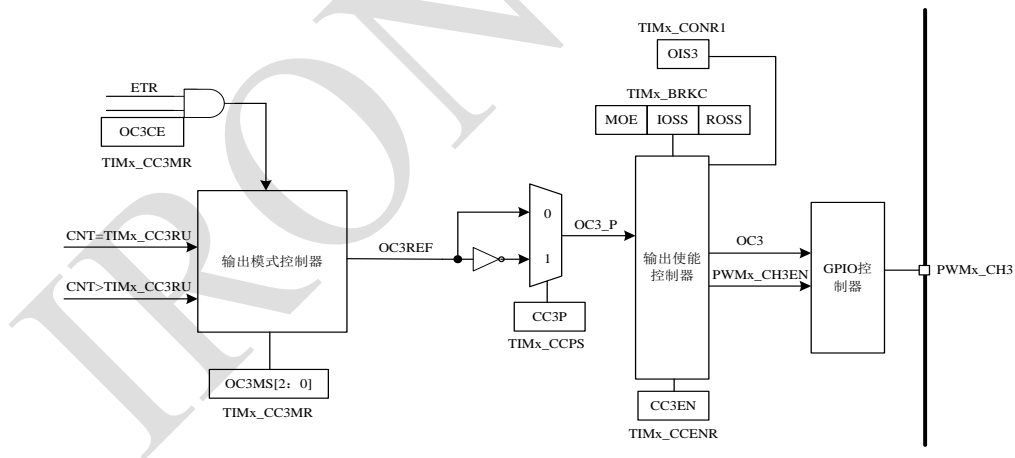


Figure 96 通道 CC3 输出控制

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 96 所示, 通道 CC3 的输出控制过程为:

- 配置寄存器 TIMx_CC3MR 的 CC3MS 位为 00, 将 CC3 通道配置为输出功能。
- 配置寄存器 TIMx_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 12.5.4.1 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 TIMx_CC3MR 的 OC3CE 位配置为 1 时, 外部信号 PWMx_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 TIMx_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 TIMx_BRKC 中 MOE 位、IOSS、ROSS 位、寄存器 TIMx_CCER 中的 CC3EN、寄存器 TIMx_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 12.5.4.3 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 PWMx_CH3EN, 将 OC3 和 PWMx_CH3EN 送入 GPIO 控制模块, 在 GPIO 模式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 TIMx_CC3RU(TIMx_CC3RD)比较成功, 则寄存器 TIMx_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。

12.5.4.1 输出模式控制

12.5.4.1.1 强置输出模式

在输出模式下(寄存器 TIMx_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 TIMx_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 TIMx_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 TIMx_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 TIMx_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 TIMx_CCxRU 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 97 为当高级定时器配置为强置输出模式时的时序图。

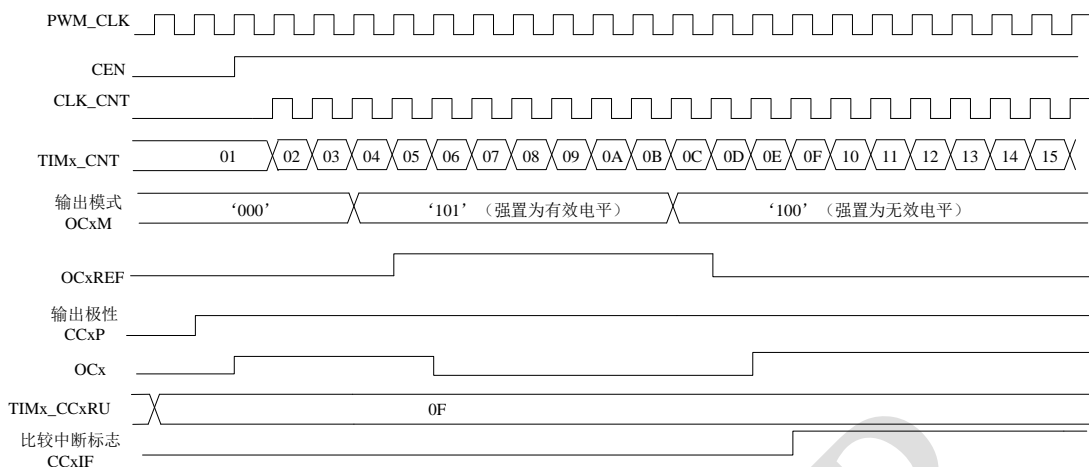


Figure 97 高级定时器强置输出模式时序图

12.5.4.1.2 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 ($TIMx_CNT = TIMx_CCxRU$):
 - 若寄存器 $TIMx_CCxMR$ 的 $OCxMS$ 位配置为 000，则 $OCxREF$ 保持不变， OCx 保持不变。
 - 若寄存器 $TIMx_CCxMR$ 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 $TIMx_CCxMR$ 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 $TIMx_CCxMR$ 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 $TIMx_ISR0$ 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 $TIMx_CONR2$ 的 $ARPLE$ 位和 $TIMx_CCxMR$ 的 $OCxPEN$ 位，确定 $TIMx_ARR$ 寄存器、 $TIMx_CCxRU$ 和 $TIMx_CCxRD$ 寄存器的预装载功能是否打开。
- 配置自动装载寄存器 $TIMx_ARR$ 、捕获比较寄存器 $TIMx_CCxRU$ 和 $TIMx_CCxRD$ ，都是遵循先配置高 8 位再配置低 8 位的规则。

- 配置寄存器 `TIMx_CCxMR` 的 `OCxMS` 位，选择输出模式，配置 `TIMx_CCPS` 寄存器，选择输出极性。
- 配置寄存器 `TIMx_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 `TIMx_IER` 的 `CCxIE` 位。
- 配置寄存器 `TIMx_EGR` 的 `UG` 位和 `COMG` 位为 1，更新具有预装载功能的寄存器。
- 设置 `TIMx_CONR2` 寄存器的 `CNTEN` 位，启动计数器。

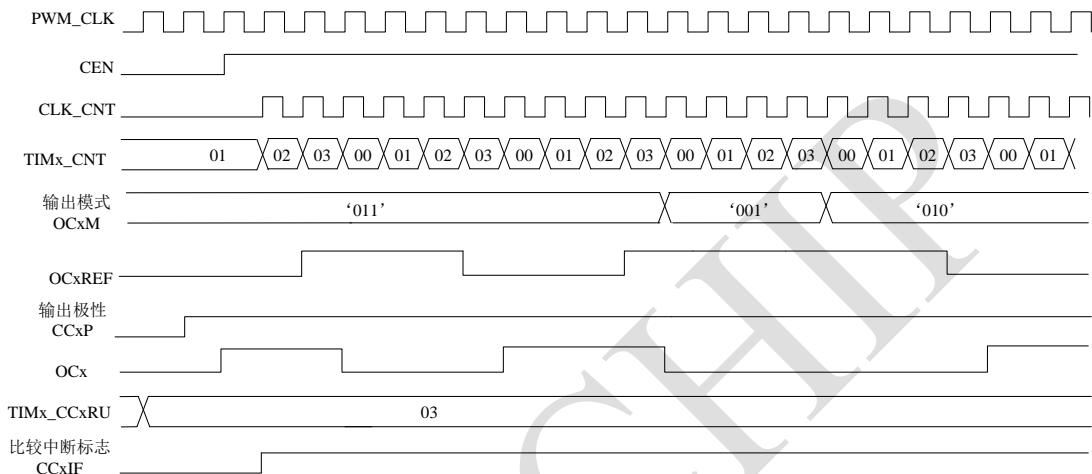


Figure 98 高级定时器比较输出模式

12.5.4.1.3 PWM 输出模式

脉冲宽度调制模式 (PWM) 可以产生一个由 `TIMx_ARR` 寄存器确定周期、由 `TIMx_CCxRU` 寄存器确定 (或与 `TIMx_CCxRD` 共同确定) 占空比的信号。

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

`TIMx_CCMRx` 寄存器 `OCxPEN` 位控制比较寄存器 `TIMx_CCxRU` 和 `TIMx_CCxRD` 的预装载使能。`TIMx_CONR2` 的 `ARPLE` 位控制自动重装载寄存器 `TIMx_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 `TIMx_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 `TIMx_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 `PWMx_CHx` 或 `PWMx_CHxN` 通过 `CCxEN`、`CCxNEN`、`MOE`、`IOSS`、`ROSS`、`OISx` 和 `OISxN` 位的组合控制，详见 12.5.4.3.1 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 $TIMx_CNT < TIMx_CCxRU$ ($TIMx_CCxRD$) 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 $TIMx_CNT < TIMx_CCxRU$ ($TIMx_CCxRD$) 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 TIMx_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 TIMx_CNT < TIMx_CCxRU 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 TIMx_CCxRU 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 99 为 TIMx_ARR=8 且计数器向上计数时 PWM 模式 1 的输出波形。

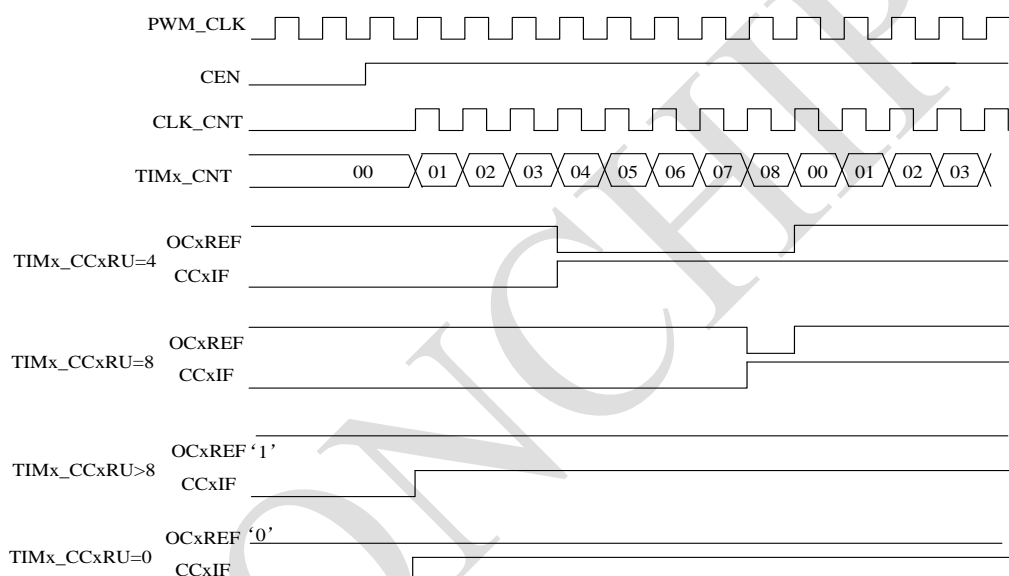


Figure 99 向上计数模式下，PWM 模式 1 输出

● 向下计数模式下的 PWM 模式 1

当 TIMx_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 TIMx_CNT > TIM0_CCxRU 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 TIMx_CCxRU 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 100 为 TIMx_ARR=8 时向上计数器时 PWM 模式 1 的输出波形。

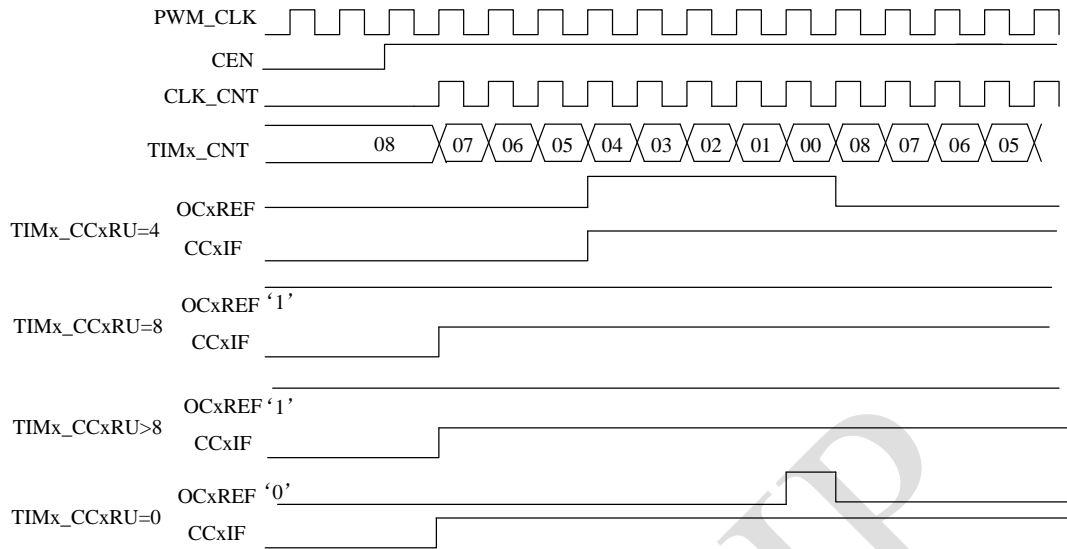


Figure 100 向下计数模式下，PWM 模式 1 输出

- 中央对齐计数模式下的 PWM 模式 1

当 $TIMx_CONR2$ 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， $TIMx_CONR2$ 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 $TIMx_CNT < TIMx_CCxRU$ 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 $TIMx_CNT > TIMx_CCxRD$ 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 101 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。

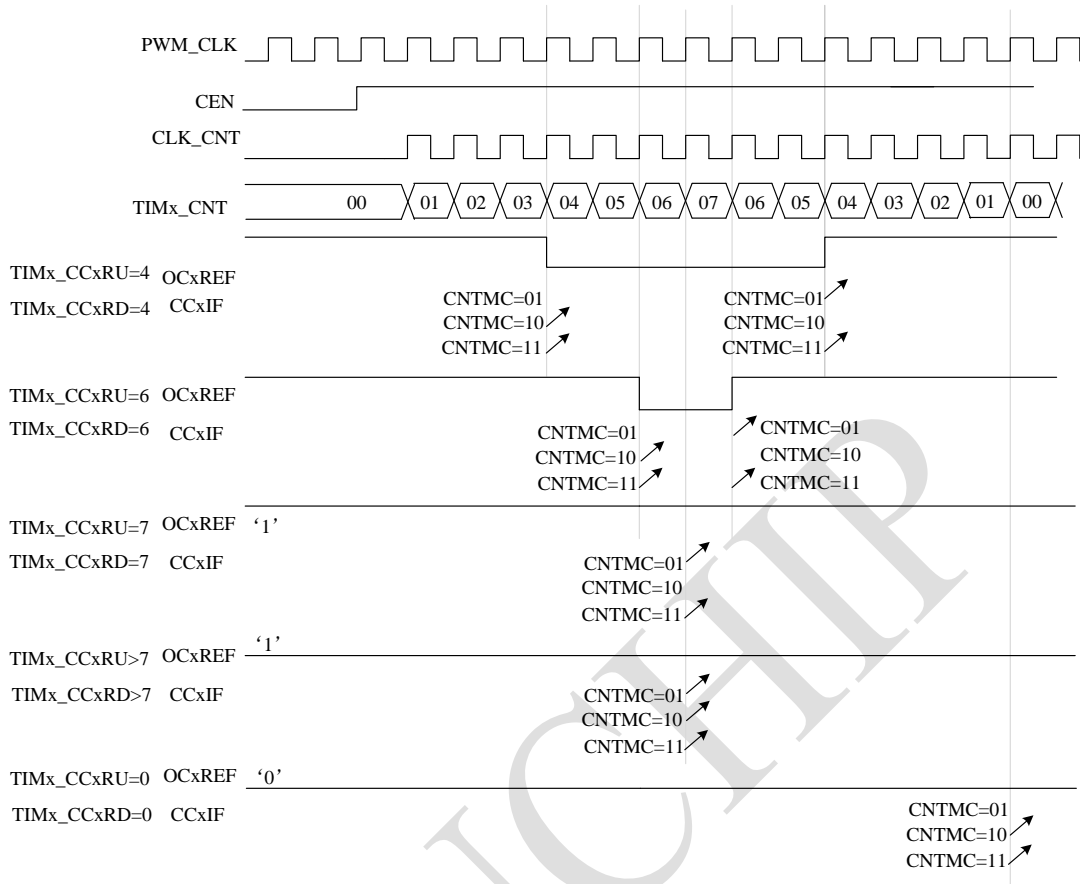


Figure 101 中央对齐计数模式下，PWM 模式 1 输出

每个通道由两个输出比较寄存器 $TIMx_CCxRU$ 和 $TIMx_CCxRD$ ，在互补模式 ($TIMx_CMPLMTARY=0x01$) 下且计数器工作在中央对齐计数模式时，计数器在向上计数过程中，计数器和 $TIMx_CCxRU$ 比较，计数器向下计数时，计数器和 $TIMx_CCxRD$ 比较，每一个通道根据比较结果输出两路互补的 PWM 波形，互补模式下，PWM 输出波形如 Figure 102 所示。

若定时器工作在独立模式 ($TIMx_CMPLMTARY=0x00$) 下，定时器输出 7 路独立的 PWM 波形， $TIMx_CCxRU$ 作为第一、第三、第五路和第七路的比较寄存器， $TIMx_CCxRD$ 作为第二、第四、第六路和第七路的比较寄存器。独立模式下，第一路和第二路 PWM 输出属于第一个通道 CC0，受到第一个通道控制寄存器 $TIMx_CC0MR$ 的控制；第三路和第四路 PWM 输出属于第二个通道 CC1，受到第二个通道控制寄存器 $TIMx_CC1MR$ 的控制；第五路和第六路 PWM 输出属于第三个通道 CC2，受到第三个通道控制寄存器 $TIMx_CC2MR$ 的控制；第七路 PWM 输出属于第四个通道 CC3，受到第四个通道控制寄存器 $TIMx_CC3MR$ 的控制，独立模式下，PWM 的输出波形如 Figure 103 所示。

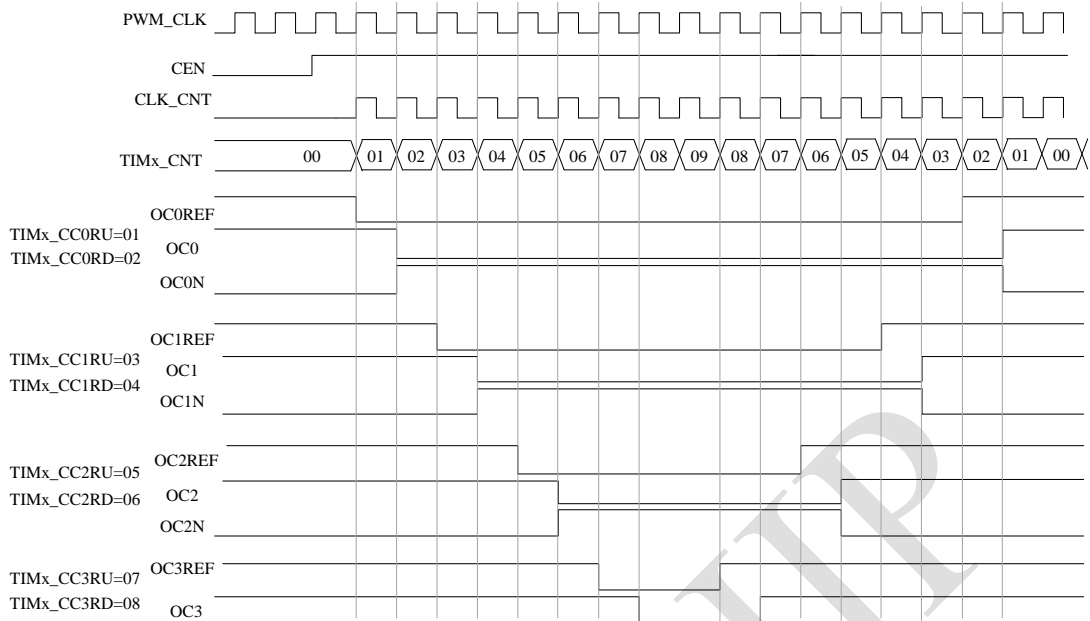


Figure 102 互补模式下，PWM 模式 1 输出波形

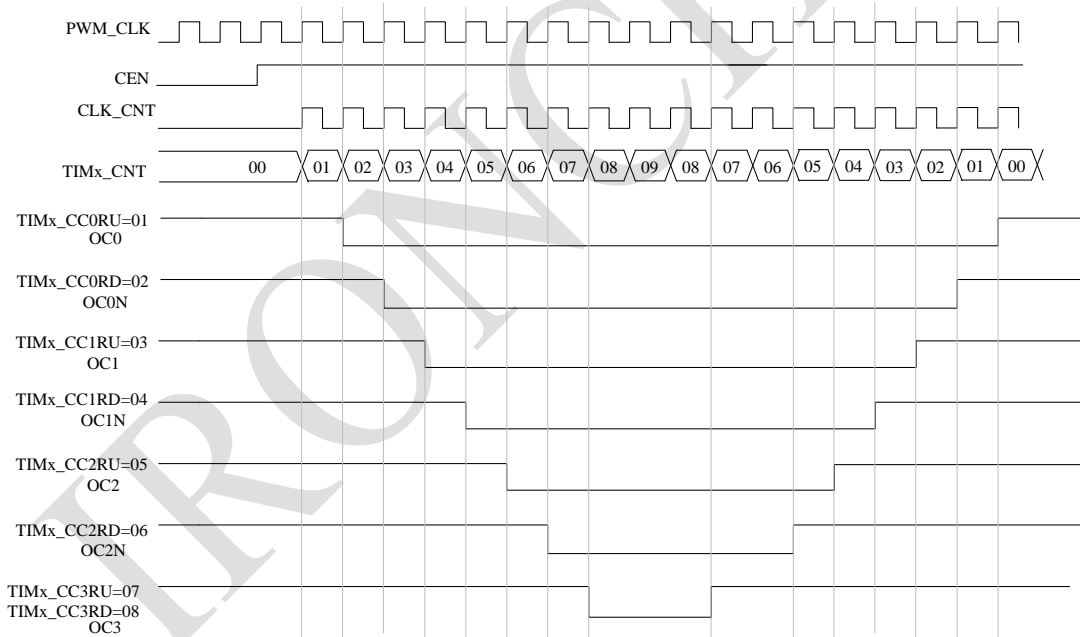


Figure 103 独立模式下，PWM 模式 1 输出波形

在中央对齐计数模式下，若将寄存器 TIMx_CONR3 的 SYMM 位配置为 1，则互补模式（TIMx_CMPLMTARY=0x01）下的 PWM 输出是对称的，即向上计数过程和向下计数过程的比较值是相同的（TIMx_CCxRU=TIM0_CCxRD），此时，软件可以只配置 TIMx_CCxRU。若寄存器 TIMx_CONR3 的 SYMM 位配置为 0，则互补模式（TIMx_CMPLMTARY=0x01）下的 PWM 输出可以是非对称的也可以是对称的，即向上计数过程和向下计数过程的比较值可以是不同的（TIMx_CCxRU≠TIM0_CCxRD）也可以是相同的，此时软件需要配置寄存器 TIMx_CCxRU 和 TIMx_CCxRD。

由此可见，在互补模式下，每一路的 PWM 输出可以是对称的也可以是非对称的，在独立模式下，前六路的 PWM 输出只可能是对称的，这是因为每一路只有一个输出比较寄存器，第七路的 PWM 输出可能是对称的也可能是非对称的，因为第七路仍有两个比较寄存器 TIMx_CC3RU 和 TIMx_CC3RD。值得注意的是在配置输出比较寄存器 TIMx_CCxRU 或 TIMx_CCxRD 时，应先配置输出比较寄存器的预装载模式（TIMx_CCMRx 寄存器 OCxPEN 位）和对称方式（TIMx_CONR3 的 SYMM 位）。

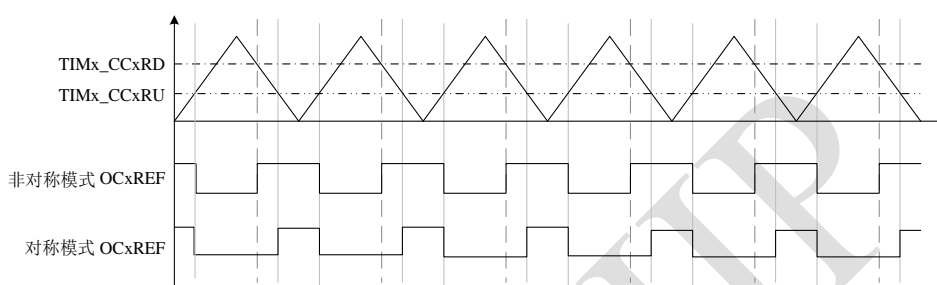


Figure 104 中央对齐计数模式下，对称模式和非对称模式的 PWM 波形输出

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 TIMx_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 TIMx_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 TIMx_EGR 寄存器的 UG 位。
 - 然后，配置寄存器 TIMx_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 TIMx_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 TIMx_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 TIMx_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 TIMx_EGR 寄存器中的 UG 位），不要在计数过程中修改计数器的值。

12.5.4.2 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它们的特性（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 TIMx_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 TIMx_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 TIMx_BRKC 寄存器的 MOE 位、IOSS、ROSS 位和 TIMx_CONR1 寄存器的 OISx、OISxN 位的控制，详见 12.5.4.3.1 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 105—Figure 107 显示了带死区的互补输出与当前参考信号 OCxREF 之间的关系。

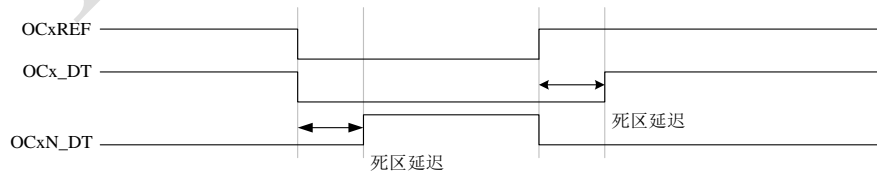


Figure 105 正常死区插入

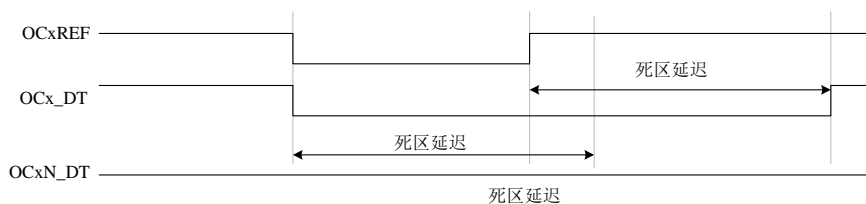


Figure 106 死区时间大于 OCxREF 无效电平宽度时的死区插入



Figure 107 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）。

12.5.4.3 输出使能控制

12.5.4.3.1 互补模式下的输出使能控制

在互补模式下，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 PWMx_CHxEN、PWMx_CHxNEN 和每一通道互补输出 OCx、OCxN。

若输出使能信号 PWMx_CHxEN 为 1，则相应的引脚输出 OCx，若 PWMx_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 PWMx_CHxNEN 对引脚的控制同理。

输出使能信号 PWMx_CHxEN 和 PWMx_CHxNEN 受到 TIMx_BRKC 寄存器的 MOE、IOSS、ROSS 位和 TIMx_CCENR 寄存器的 CCxEN、CCxNEN 位控制，具体控制方式如 Table 194 和 Table 195 所示。

高级定时器的管脚输出 PWMx_CHx 和 PWMx_CHxN 受到 TIM0_BRKC 寄存器的 MOE 位、IOSS 位、ROSS 位、TIMx_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 PWMx_CHx 和 PWMx_CHxN 的输出值如 Table 194 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 PWMx_CHx 和 PWMx_CHxN 的输出值如 Table 195 所示。

寄存器 TIMx_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 194 和 Table 195 可以看出，当 MOE 位为 1 时，管脚的 PWM 输出使能（PWMx_CHxEN/PWMx_CHxNEN）受寄存器 TIMx_BRKC 的 ROSS 位和 TIMx_CCENR 的

CCxEN、CCxNEN 位控制，ROSS 位表示运行模式下（MOE=1）关闭状态（TIMx_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 PWMx_CHxEN/PWMx_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 PWMx_CHxEN/PWMx_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 PWMx_CHxEN/PWMx_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 PWMx_CHxEN/PWMx_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 TIMx_BRKC 的 IOSS 位和 TIMx_CCEN 的 CCxEN、CCxNEN 位控制，IOSS 位表示空闲模式下（MOE=0）关闭状态（TIMx_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 PWMx_CHxEN/PWMx_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 PWMx_CHxEN 和 PWMx_CHxNEN 均有效，管脚输出 PWM 模块的输出值 OCx/OCxN。

Table 194 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	PWMx_CHx 输出状态	PWMx_CHxN 输出状态
1	X	0	0	0	输出禁止（引脚输出高阻） PWMx_CHxEN =0; PWMx_CHx=高阻	输出禁止（引脚输出高阻） PWMx_CHxNEN =0; PWMx_CHxN=高阻;
			0	1	输出禁止（引脚输出高阻） PWMx_CHxEN =0 PWMx_CHx=高阻	PWMx_CHxNEN =1; PWMx_CHxN=OCxN =CCxNP;
			1	0	PWMx_CHxEN =1; PWMx_CHx=OCx =CCxP;	输出禁止（引脚输出高阻） PWMx_CHxNEN =0; PWMx_CHxN=高阻;
			1	1	PWMx_CHxEN =1	PWMx_CHxNEN =1;

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

CEN=0						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWMx_CHx 输出状态	PWMx_CHxN 输出状态
0		X	1	0	<ul style="list-style-type: none"> ● CCxP=CCxNP 时： 若 $OISx=OISxN$, $PWMx_CHx=OCx=CCxP$, $PWMx_CHxN=OCxN=CCxNP$; 若 $OISx \neq OISxN$, $PWMx_CHx=OCx=OISx$, $PWMx_CHxN=OCxN=OISxN$. ● 当 $CCxP \neq CCxNP$ 时： 若 $OISx=OISxN$, 则 $PWMx_CHx=OCx=OISx$, $PWMx_CHxN=OCxN=OISxN$; 若 $OISx \neq OISxN$, 则 $PWMx_CHx=OCx=CCxP$, $PWMx_CHxN=OCxN=CCxNP$. 	
			1	1		

Table 195 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	CCxEN 位	CCxNEN 位	PWMx_CHx 输出状态	PWMx_CHxN 输出状态
1	X	0	0	0	输出禁止（引脚输出高阻） $PWMx_CHxEN=0$; $PWMx_CHx=$ 高阻;	输出禁止（引脚输出高阻） $PWMx_CHxNEN=0$; $PWMx_CHxN=$ 高阻;
			0	1	输出禁止（引脚输出高阻） $PWMx_CHxEN=0$; $PWMx_CHx=$ 高阻;	$PWMx_CHxNEN=1$; $PWMx_CHxN=OCxN=OCxREF \wedge CCxNP$
			1	0	$PWMx_CHxEN=1$; $PWMx_CHx=OCx=OCxREF \wedge CCxP$	输出禁止（引脚输出高阻） $PWMx_CHxNEN=0$; $PWMx_CHxN=$ 高阻

CEN=1							
控制位					输出状态		
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWMx_CHx 输出状态	PWMx_CHxN 输出状态	
			1	1	PWMx_CHxEN =1; PWMx_CHx=OCx =OCxREF + 死区 + 极性	PWMx_CHxNEN =1; PWMx_CHxN=OCxN =OCxREF + 死区 + 极性	
			1	0	0	关闭状态(输出使能有效, 输出无效电平) PWMx_CHxEN =1; PWMx_CHx=OCx = CCxP	关闭状态(输出使能有效, 输出无效电平) PWMx_CHxNEN =1; PWMx_CHxN=OCxN = CCxNP
				0	1	关闭状态(输出使能有效, 输出无效电平) PWMx_CHxEN =1; PWMx_CHx=OCx = CCxP	PWMx_CHxNEN =1; PWMx_CHxN=OCxN = OCxREF ^ CCxNP
				1	0	PWMx_CHxEN =1; PWMx_CHx =OCx =OCxREF ^ CCxP	关闭状态(输出使能有效, 输出无效电平) PWMx_CHxNEN =1; PWMx_CHxN=OCxN = CCxNP
				1	1	PWMx_CHxEN =1; PWMx_CHx =OCx =OCxREF + 死区 + 极性	PWMx_CHxNEN =1; PWMx_CHxN =OCxN =OCxREF + 死区 + 极性
0	0	X	0	0	输出禁止(引脚输出高阻)		
			0	1	PWMx_CHxEN = 0; PWMx_CHxNEN = 0;		
			1	0	PWMx_CHx =高阻;		

CEN=1						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWMx_CHx 输出状态	PWMx_CHxN 输出状态
1	1		1	1	PWMx_CHx N=高阻;	
			0	0	关闭状态 (输出使能有效, 输出无效电平)	
			0	1	PWMx_CHxEN = 1, PWMx_CHxNEN = 1;	
			1	0	● 当 CCxP=CCxNP 时: 若 OISx=OISxN, PWMx_CHx=OCx =CCxP, PWMx_CHxN=OCxN =CCxNP; 若 OISx≠OISxN, PWMx_CHx=OCx =OISx, PWMx_CHxN=OCxN =OISxN。	
			1	1	● 当 CCxP≠CCxNP 时: 若 OISx=OISxN, 则 PWMx_CHx=OCx =OISx, PWMx_CHxN=OCxN =OISxN; 若 OISx≠OISxN, 则 PWMx_CHx=OCx =CCxP, PWMx_CHxN=OCxN =CCxNP。	

由 Table 194 和 Table 195 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 TIMx_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 TIMx_CONR1。
- 2) 配置寄存器 TIMx_CONR3 的 SYMM 位(控制比较寄存器 TIMx_CCxRU 和 TIMx_CCxRD 的配置模式)。
- 3) 配置 TIMx_CONR2 的 ARPLE 位和 TIMx_CONR0 的 CCPE 位,控制相应寄存器的预装载功能。
- 4) 配置各个通道输入捕获/输出比较模式控制寄存器 TIMx_CCxMR。
- 5) 配置周期寄存器 TIMx_ARR、比较/捕获寄存器 TIMx_CCxRU、TIMx_CCxRD、通道使能寄存器 TIMx_CCxEN、死区控制等各种寄存器。
- 6) 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 TIMx_EGR 的 UG 位为 1, COMG 位为 1, 装载各个预装载寄存器)。
- 7) 配置 TIMx_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)。

8) 配置 TIMx_CONR2 寄存器的 CNTEN 位，启动 PWM 计数器。

12.5.4.3.2 独立模式下的输出使能控制

当寄存器 TIMx_CMPLMTARY 的 CMPEN 位配置为 0 时，定时器输出 7 路独立的 PWM 波形，此时管脚上的输出使能 PWMx_CHxEN/PWMx_CHxNEN 与寄存器 CCxE/CCxNE 的值相同，当 CCxE/CCxNE 配置为 0 时，管脚上的输出使能 PWMx_CHxEN/PWMx_CHxNEN 为 0，相应管脚输出高阻，当 CCxE/CCxNE 配置为 1 时，管脚上的输出使能 PWMx_CHxEN/PWMx_CHxNEN 有效，相应管脚输出 PWM 模块的输出值 OxC/OCxN。独立模式下，高级定时器引脚输出和输出使能如 Table 196 和 Table 197 所示

Table 196 独立模式下，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	PWMx_CHx 输出状态	PWMx_CHxN 输出状态
1	X	X	0	0	输出禁止（引脚输出高阻） PWMx_CHxEN=0; PWMx_CHx=高阻;	输出禁止（引脚输出高阻） PWMx_CHxNEN=0; PWMx_CHxN=高阻;
			0	1	输出禁止（引脚输出高阻） PWMx_CHxEN=0; PWMx_CHx=高阻;	PWMx_CHxNEN=1; PWMx_CHxN=OCxN=CCxNP;
			1	0	PWMx_CHxEN=1; PWMx_CHx=OCx=CCxP;	输出禁止（引脚输出高阻） PWMx_CHxNEN=0; PWMx_CHxN=高阻;
			1	1	PWMx_CHxEN=1; PWMx_CHx=OCx=CCxP;	PWMx_CHxNEN=1; PWMx_CHxN=OCxN=CCxNP;
0	X	X	0	0	输出禁止（引脚输出高阻） PWMx_CHxEN=0;	输出禁止（引脚输出高阻） PWMx_CHxNEN=0;

CEN=0						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWMx_CHx 输出状态	PWMx_CHxN 输出状态
					PWMx_CHx=高阻;	PWMx_CHxN=高阻;
			0	1	输出禁止（引脚输出高阻） PWMx_CHxEN=0; PWMx_CHx=高阻;	PWMx_CHxNEN =1; PWMx_CHxN=OCxN =OISxN;
			1	0	PWMx_CHxEN =1; PWMx_CHx=OCx =OISx;	输出禁止（引脚输出高阻） PWMx_CHxNEN=0; PWMx_CHxN=高阻;
			1	1	PWMx_CHxEN =1; PWMx_CHx=OCx =OISx;	PWMx_CHxNEN =1; PWMx_CHxN=OCxN =OISxN;

Table 197 独立模式下，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	CCxEN 位	CCxNEN 位	PWMx_CHx 输出状态	PWMx_CHxN 输出状态
			0	0	输出禁止（引脚输出高阻） PWMx_CHxEN=0; PWMx_CHx=高阻;	输出禁止（引脚输出高阻） PWMx_CHxNEN=0; PWMx_CHxN=高阻;
1	X	X	0	1	输出禁止（引脚输出高阻） PWMx_CHxEN =0; PWMx_CHx=高阻;	PWMx_CHxNEN =1; PWMx_CHxN=OCxN =OCxREF ^ CCxNP;
			1	0	PWMx_CHxEN =1; PWMx_CHx=OCx	输出禁止（引脚输出高阻）

CEN=1						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWMx_CHx 输出状态	PWMx_CHxN 输出状态
					=OCxREF ^ CCxP;	PWMx_CHxNEN =0; PWMx_CHxN=高阻;
			1	1	PWMx_CHxEN =1; PWMx_CHx=OCx =OCxREF ^ CCxP;	PWMx_CHxNEN =1; PWMx_CHxN=OCxN =OCxREF ^ CCxNP;
0	X	X	0	0	输出禁止（引脚输出高阻） PWMx_CHxEN=0; PWMx_CHx=高阻;	输出禁止（引脚输出高阻） PWMx_CHxNEN=0; PWMx_CHxN=高阻;
			0	1	输出禁止（引脚输出高阻） PWMx_CHxEN=0; PWMx_CHx=高阻;	PWMx_CHxNEN =1; PWMx_CHxN=OCxN =OISxN;
			1	0	PWMx_CHxEN =1; PWMx_CHx=OCx =OISx;	输出禁止（引脚输出高阻） PWMx_CHxNEN=0; PWMx_CHxN=高阻;
			1	1	PWMx_CHxEN =1; PWMx_CHx=OCx =OISx;	PWMx_CHxNEN =1; PWMx_CHxN=OCxN =OISxN;

12.5.4.4 刹车功能

当使用刹车功能时，依据相应的控制位（TIMx_BRKC 寄存器的 MOE、IOSS 和 ROSS，TIMx_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 PWMx_CHx 和 PWMx_CHxN 不会同时处于有效电平。

12.5.4.4.1 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 TIMx_EGR 的 BG 位为 1）对

PWM 输出进行刹车。

- 时钟失败事件。时钟失败事件由时钟复位控制器中的时钟监控系统产生，详见 3.1.3.3 节说明。

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 TIMx_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 TIMx_BRKC 中的 BKP 位选择。

12.5.4.4.2 刹车过程

当刹车事件发生时，硬件做如下处理：

- 独立模式下：

MOE 位被异步地立即清零，每一个通道输出由寄存器 TIMx_CONR2 的 OISx 位或 OISxN 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 OISx、OISxN 位以及极性指示的电平驱动输出端口，详见 Table 194 和 Table 195 的 MOE=0 部分。即使在这种情况下，PWMx_CHx 和 PWMx_CHxN 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 TIMx_IER 中的 BIE 位，则产生一个中断。

- 如果寄存器 TIMx_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。

12.5.4.4.3 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。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 TIMx_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 TIMx_BRKC 寄存器的 ROSS 位设置为 1，则无论 CCxE 和 CCxNE 配置为使能与否，PWMx_CHx 和 PWMx_CHxN 管脚的输出使能都为高；若 TIMx_BRKC 寄存器

的 ROSS 位设置为 0, 则当 CCxEN 配置为有效时, PWMx_CHx 管脚的输出使能有效, 否则, PWMx_CHx 管脚的输出使能无效, PWMx_CHx 管脚输出高阻, PWMx_CHxN 的输出管脚控制同理。

当 MOE 位为 0 时, PWM 输出处于空闲模式 (刹车后的状态), 若 TIMx_BRKC 寄存器的 IOSS 位设置为 1, 则无论 CCxE 和 CCxNE 配置为使能与否, PWMx_CHx 和 PWMx_CHxN 管脚的输出使能都有效; 若 TIMx_BRKC 寄存器的 IOSS 位设置为 0, 则 PWMx_CHx 和 PWMx_CHxN 管脚的输出使能设置为无效, PWMx_CHx 和 PWMx_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, PWMx_CHx 和 PWMx_CHxN 管脚上的输出如 Table 194 所示; 当 CNTEN 为 1 时, 定时器处于工作状态, PWMx_CHx 和 PWMx_CHxN 管脚上的输出如 Table 195 所示。刹车后 PWMx_CHx 和 PWMx_CHxN 管脚上的输出状态见 Table 194 和 Table 195 中的 MOE=0 部分。Figure 108 显示了响应刹车的输出实例。

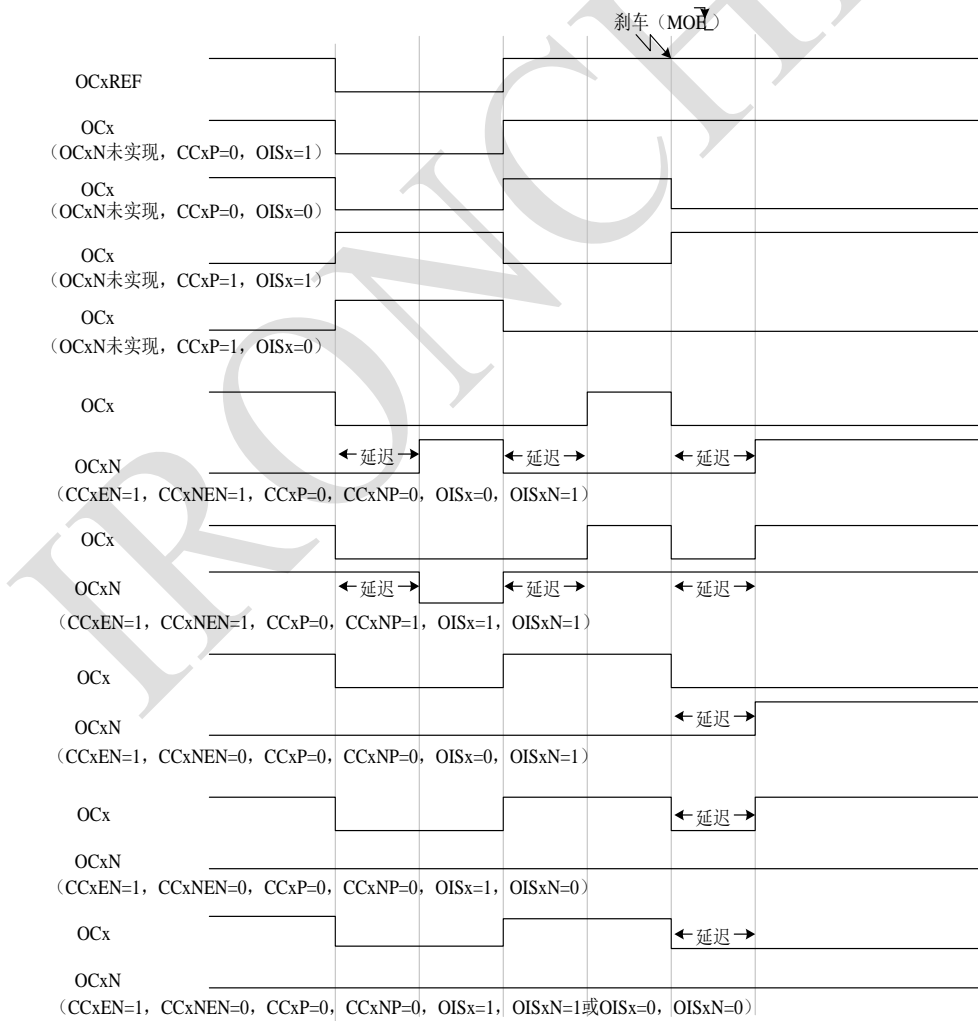


Figure 108 高级定时器刹车输出

12.5.4.5 外部事件 (PWMx_ETR) 清除 OCxREF 信号

对于一个给定的通道，设置寄存器 TIMx_CCxMR 中的 OCxCE 位为 1，能够用 PWMx_ETR 通道输入信号的有效电平把 OCxREF 信号清为低电平，PWMx_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。PWMx_ETR 通道配置如下：

- 关闭 PWMx_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) 和外部触发信号的滤波器 (EXTFLT)。

Figure 109 显示了当 PWMx_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。

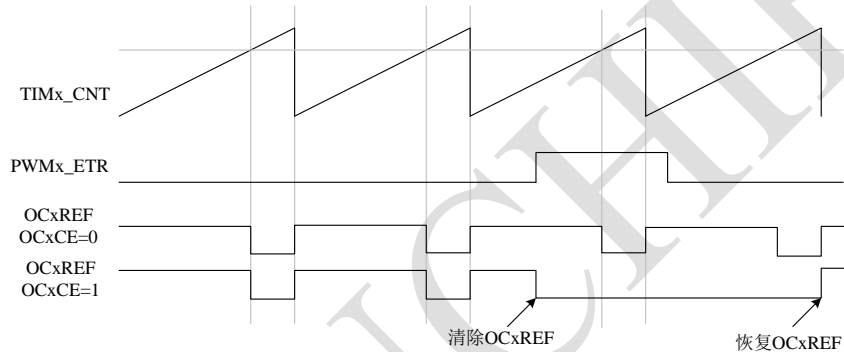


Figure 109 外部信号 PWMx_ETR 清除 OCxREF

12.5.4.6 霍尔传感器接口功能

使用高级定时器 TIMx 产生 PWM 信号驱动马达时可以用另一个高级定时器作为“接口定时器”来连接霍尔传感器。配置 TIMx_CONR0 寄存器的 CH0INSEL 位为 1，选择定时器的三个输入脚 (PWMx_CH0、PWMx_CH1、PWMx_CH2) 异或后的信号连接至 CC0 输入通道，“接口定时器”捕获这个信号。

配置“接口定时器”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。

“接口定时器”TIMx 上的通道 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。

“接口定时器”TIMx 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 送到另一个高级定时器 TIMx，在 TIMx 中利用此波形触发 TIMx 的 COM 事件。

12.5.4.7 六步 PWM

通过高级定时器 TIMx 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。下面以 TIM0 作为 PWM 输出,使用 TIM1 作为霍尔信号接口产生 COM 事件为例阐述六步 PWM 的控制过程。

霍尔信号产生 COM 事件的方式有以下两种:

- 第一种方式:将三路霍尔信号连接到“接口定时器”TIM1,在TIM1产生一个触发信号,然后通过TRGO传送到TIM0。若寄存器TIM0_CONR0的CCUS为0,TRGO送入TIM0后引起触发中断,在中断处理程序中设置六步PWM相应相位的输出控制寄存器,并配置TIM0_EGR寄存器的COM位进行换相;若寄存器TIM0_CONR0的CCUS位为1,TRGO进入TIM0后触发COM事件进行PWM换相,同时引起触发中断,在中断处理程序中配置相应相位的输出控制寄存器。
- 第二种方式: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1:霍尔信号连接到定时器TIM1,在任一霍尔输入变化指定的延迟后,改变高级定时器TIM0的配置,实现TIM0的六步PWM输出。Figure 110为TIM1作为霍尔传感器接口驱动TIM0产生六步PWM的示意图。

- 将TIM1的计数器周期值配置为最大值,即配置寄存器TIM1_ARRH为0xFF,配置寄存器TIM1_ARRL为0xFF。
- 配置寄存器TIM1_CONR0的CH0INSEL位为1,将TIM0_CH0、TIM0_CH1和TIM0_CH2管脚上的信号异或后连接到通道CC0。
- 配置寄存器TIM1_TGICR0的TRGS位为100,选择CH0INF_ED作为从模式控制器的触发输入信号。
- 配置寄存器TIM1_TGICR0的SMS位为100,将从模式控制器配置为复位模式。
- 配置寄存器TIM1_CC0MR的CC0MS位为11,将CH0INF_ED选为捕获信号。
- 配置寄存器TIM1_CC3MR的CC3MS位为00,将通道CC3配置为输出模式。
- 配置寄存器TIM1_CC3MR的OC3MS位为111,将通道CC3配置为PWM模式2。
- 配置寄存器TIM1_CC3RU指定延迟产生一个正脉冲,该寄存器的取值范围为0x01至两个相邻霍尔信号跳变的间隔。

- 配置寄存器 TIM1_CONR0 的 MMC 位为 111，将正脉冲 OC3REF 选择为 TRGO 输出。
- 配置 TIM0 的周期值寄存器 TIM0_ARR、输出模式控制寄存器 TIM0_CCxMR、极性控制寄存器 TIM0_CCPS、通道使能控制寄存器 TIM0_CCENR、比较值寄存器 TIM0_CCxRU 或 TIM0_CCxRD 等。
- 配置寄存器 TIM0_TGICR0 的 TRGS 位为 000，选择 ITR0（TIM1 的 TRGO 信号）作为 TIM0 从模式控制器的触发输入信号。
- 配置寄存器 TIM0_CONR0 的 CCPE 位为 1，使能比较输出控制寄存器的预装载功能。
- 配置寄存器 TIM0_IER 的 TRIE 位为 1，使能触发中断。在触发中断处理程序中通过配置寄存器 TIM0_CCENR、TIM0_CCPS 实现对六步 PWM 每个相位的控制，同时在触发中断服务程序中配置 TIM0_EGR 的 COM 位，产生 COM 事件，更新 TIM0 的输出控制寄存器。
- 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。

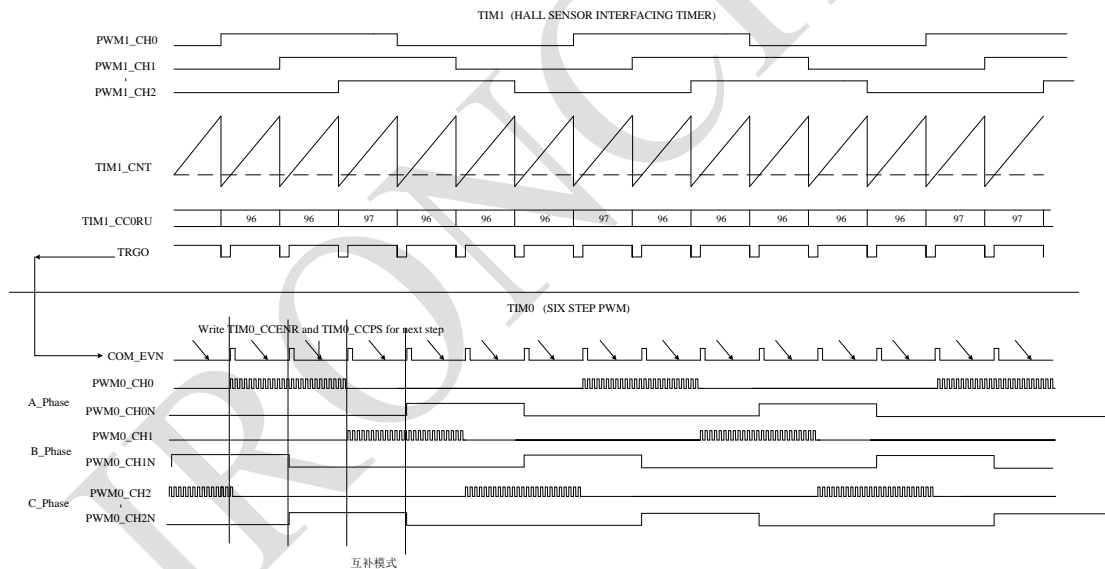


Figure 110 TIM1 作为霍尔传感器接口驱动 TIM0 产生六步 PWM 的示意图

例 2：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 TIM0 的 COM 事件来改变 TIM0 的输出配置。

- 配置 TIM0 的周期值寄存器 TIM0_ARR、输出模式控制寄存器 TIM0_CCxMR、极性控制寄存器 TIM0_CCPS、通道使能控制寄存器 TIM0_CCENR、比较值寄存器 TIM0_CCxRU 或 TIM0_CCxRD 等。
- 配置寄存器 TIM0_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 INT0EN 为 0x07，配置寄存器 INTOPE 为 0x07，配置寄存器 INTONE 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 TIM0_CCENR、TIM0_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 TIM0_EGR 的 COM 位，产生 COM 事件，更新 TIM0 的输出控制寄存器。

12.5.4.8 单脉冲模式

将寄存器 TIMx_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 TIMx_CONR2 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（TIMx_CCxRU）需满足以下条件：

- 向上计数模式：TIMx_CNT < TIMx_CCxRU < TIMx_ARR
- 向下计数模式：TIMx_CNT > TIMx_CCxRU

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_CC1RU，TIM0_CC1RU = 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。

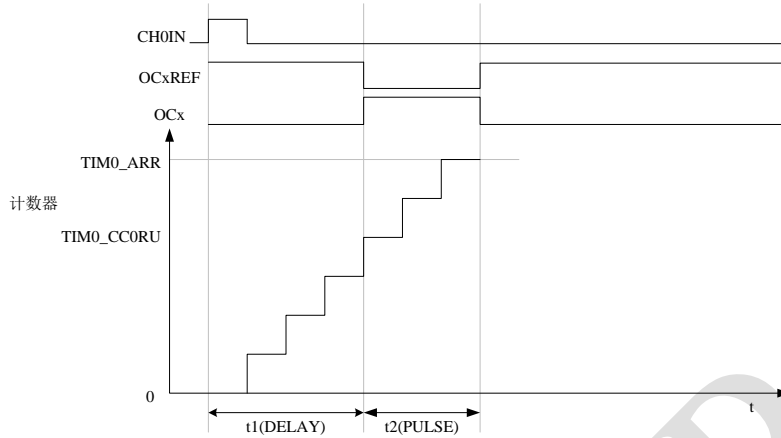


Figure 111 单脉冲模式的例子

在单脉冲模式下，将从模式设置为触发模式时，当检测到 PWMx_CHx 输入脚有效边沿时硬件自动使能计数器的使能位，然后计数器与比较值 (TIMx_CCxRU) 进行比较产生输出脉冲，这些操作需要一定的时钟周期，因此它限制了可得到的最小延迟，如果要以最小延时输出波形，可以设置 TIMx_CCMR 寄存器的 OCxFE 位，此时 OCxREF 直接响应激励而不再依赖比较的结果，输出的波形与比较匹配的波形一样，OCxFE 只在通道配置为 PWM 模式 1 和 PWM 模式 2 时起作用。

12.5.5 主模式控制

定时器工作在主模式下，可产生触发信号 TRGO 去控制其他模块，其他模块包括另一个定时器和 AD 采样模块。Figure 112 为主模式控制器示意图。

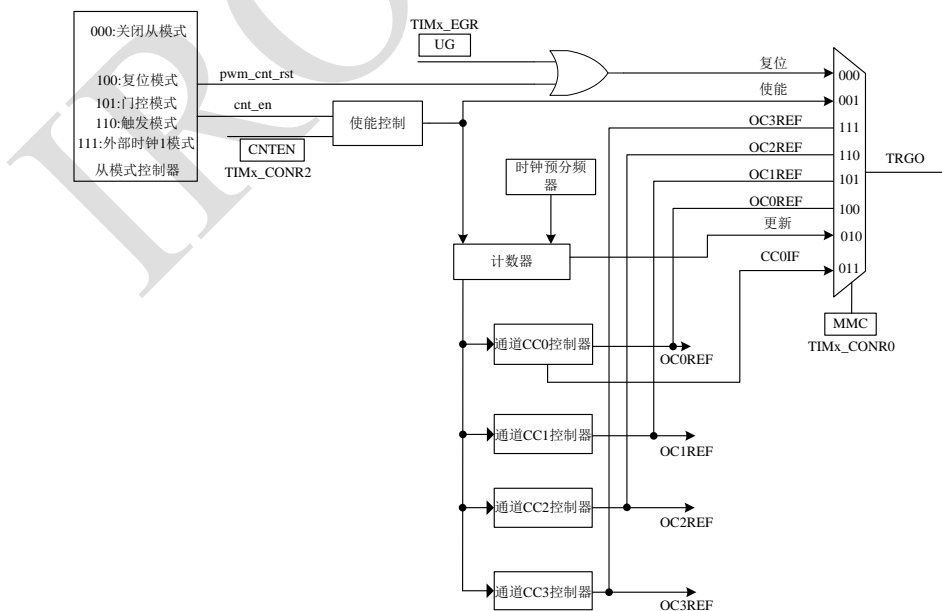


Figure 112 主模式控制示意图

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 TIMx_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 TIMx_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。将此信号通过 TIMx_TRGO 送给 AD 进行触发采样时, 每来一次复位信号, TIMx_TRGO 翻转一次, 因此, 在使用此信号触发 AD 采样时, 需要将 AD 的寄存器 ADCxETC 的 NQTPOSEN、NQTNEGEN 位或 SQTPOSEN、SQTNEGEN 位置 1, 即同时选择触发信号的上升沿和下降沿触发 AD 采样。
- 配置寄存器 TIMx_CONR0 的 MMC 位为 001, 选择计数器的使能信号作为触发输出 TIMx_TRGO。将此信号通过 TIMx_TRGO 送给 AD 进行触发采样时, 每次使能信号有效时, TIMx_TRGO 翻转一次, 因此, 在使用此信号触发 AD 采样时, 需要将 AD 的寄存器 ADCxETC 的 NQTPOSEN、NQTNEGEN 位或 SQTPOSEN、SQTNEGEN 位置 1, 即同时选择触发信号的上升沿和下降沿触发 AD 采样。
- 配置寄存器 TIMx_CONR0 的 MMC 位为 011, 选择通道 CC0 的比较标志 CC0IF 作为触发输出 TIMx_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 TIMx_CC0RU 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 TIMx_TRGO。将此信号通过 TIMx_TRGO 送给 AD 进行触发采样时, 每次 CC0IF 置位时, TIMx_TRGO 翻转一次, 因此, 在使用此信号触发 AD 采样时, 需要将 AD 的寄存器 ADCxETC 的 NQTPOSEN、NQTNEGEN 位或 SQTPOSEN、SQTNEGEN 位置 1, 即同时选择触发信号的上升沿和下降沿触发 AD 采样。
- 配置寄存器 TIMx_CONR0 的 MMC 位为 010, 选择更新事件 UEV 作为触发输出。将此信号通过 TIMx_TRGO 送给 AD 进行触发采样时, 每次发生更新事件时, TIMx_TRGO 翻转一次, 因此, 在使用此信号触发 AD 采样时, 需要将 AD 的寄存器 ADCxETC 的 NQTPOSEN、NQTNEGEN 位或 SQTPOSEN、SQTNEGEN 位置 1, 即同时选择触发信号的上升沿和下降沿触发 AD 采样。
- 配置寄存器 TIMx_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 TIMx_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 TIMx_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 TIMx_CONR0 的 MMC 位为 111, 选择通道 CC3 的基准参考信号 OC3REF 作为触发输出。

触发输出 TIMx_TRGO 除了可以控制另一个定时器，还可以触发 AD 采样，由定时器发出的触发 AD 采样的 CC 事件有：

- 定时器 TIM0 的触发输出 TIM0_TRGO；
- 定时器 TIM1 的触发输出 TIM1_TRGO；
- 定时器 TIM0 的通道 CC0 的主路输出 TIM0_OC0；
- 定时器 TIM0 的通道 CC1 的主路输出 TIM0_OC1；
- 定时器 TIM0 的通道 CC2 的主路输出 TIM0_OC2；
- 定时器 TIM0 的通道 CC3 的主路输出 TIM0_OC3；
- 定时器 TIM1 的通道 CC0 的主路输出 TIM1_OC0；
- 定时器 TIM1 的通道 CC1 的主路输出 TIM1_OC1；
- 定时器 TIM1 的通道 CC2 的主路输出 TIM1_OC2；
- 定时器 TIM1 的通道 CC3 的主路输出 TIM1_OC3；

12.5.6 中断

12.5.6.1 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。
- 刹车使能有效时收到时钟失败事件（当使用外部时钟时，外部 22.1184MHz 晶振停振）。
- 软件配置寄存器 TIMx_EGR 的 BG 位为 1，产生软刹车事件。

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 TIMx_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 TIMx_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 TIMx_CCxMR 的 CCxMS 位配置为 01/10/11），软件配置寄存器 TIMx_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 TIMx_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 TIMx_CCxMR 的 CCxMS 位配置为 00），软件配置寄存器 TIMx_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 TIMx_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。若已打开通上溢更新中断使能，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 TIMx_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。若已打开通下溢更新中断使能，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 TIMx_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。若已打开复位更新中断使能，复位更新中断产生的条件有：
- 设置 TIMx_EGR 寄存器的 UG 位为 1。
 - 从模式控制器产生的复位信号。
- 8) 独立模式下第 x 路（x=1~7）PWM 比较中断。若已打开中断使能，独立模式下第 x 路（x=1~7）PWM 比较中断产生的条件有：
- 计数器与比较值匹配时。

各个中断源产生后，在 TIMx 中进行中断合并得到中断 PWMx_INT，将 PWMx_INT 送入 JMT51，在 JMT51 中经过一级门控后得到 JMT51 中的 TIMx 中断，Figure 113 为中断产生过程。

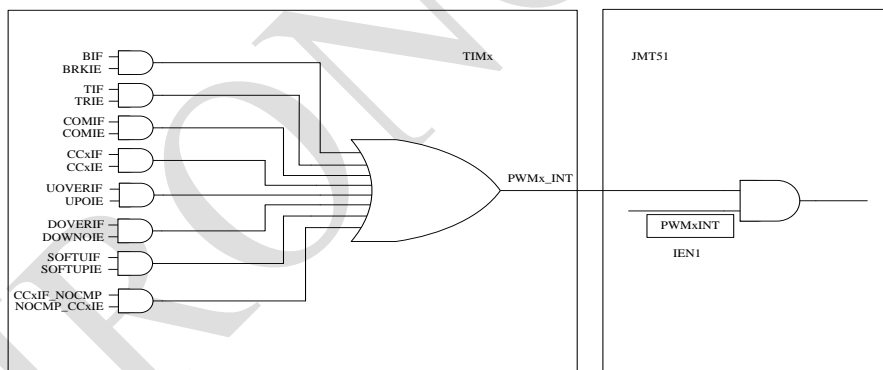


Figure 113 JMT1808R 高级定时器中断

12.5.6.2 中断配置及处理

TIMx 中断配置及处理过程如下：

- 配置中断使能寄存器 TIMx_IER、TIMx_IER_NOCMP 以及寄存器 TIMx_CONR3 的中断使能位，使能相应的中断源。
- 配置寄存器 IEN1 的 PWM0INT 位或 PWM1INT 位，使能高级定时器中断。
- 当产生 TIMx 中断后，软件需要回读中断标志寄存器 TIMx_ISR0、TIMx_ISR1 或 TIMx_ISR2，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。

12.5.7 高级定时器同步

JMT1808R 的两个定时器是互连的，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

Figure 114 显示了两个定时器的连接，处于主模式的定时器 TIMx 将触发信号 TIMx_TRGO 输出，然后接到处于从模式的定时器 TIMy 的触发输入 ITR0，这样通过定时器 TIMx 的主模式控制和定时器 TIMy 的从模式控制可以实现两个定时器的连接。

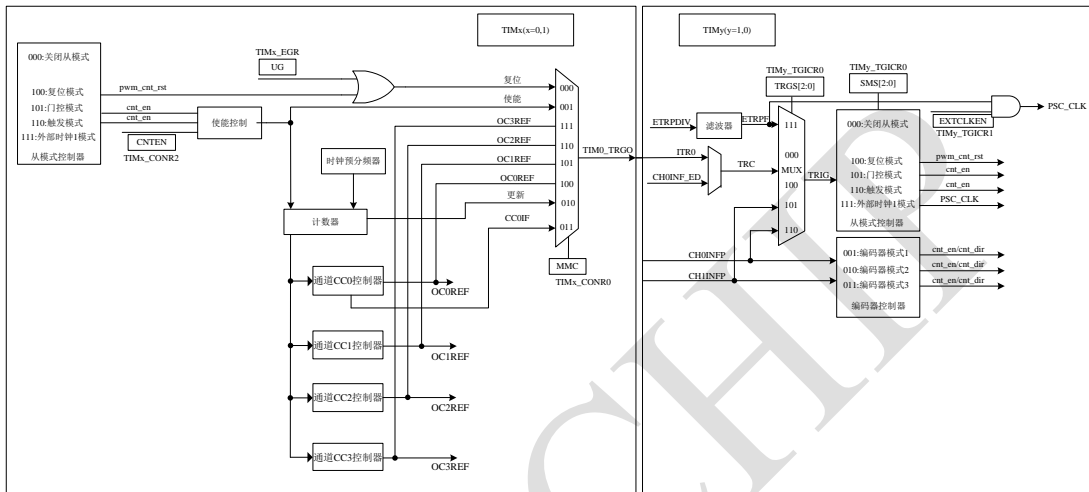


Figure 114 JMT1808R 两个高级定时器连接

12.5.7.1 一个定时器作为另一个定时器的预分频器

配置定时器 TIM0 作为定时器 TIM1 的预分频器，实际上是将定时器 TIM0 的触发输出（TRGO）作为定时器 TIM1 的外部时钟触发计数器计数。

通过配置寄存器 TIM0_CONR0 的 MMC 位可以选择不同的信号作为 TIM1 的计数器时钟。例如配置 TIM0_CONR0 的 MMC 为 1xx，则选择 OCxREF 作为 TIM0 的触发输出，OCxREF 的上升沿触发 TIM1 的计数器计数；配置 TIM0_CONR0 的 MMC 为 010，则选择 TIM0 的更新事件（UEV）作为 TIM0 的触发输出。下面以更新事件（UEV）触发为例阐述具体的控制过程：

- 配置定时器 TIM0 的主模式控制器。配置 TIM0 的 TIM0_CONR0 寄存器的 MMC 位为 010，选择定时器 TIM0 的更新事件 UEV 作为触发输出 TIM0_TRGO。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 TRGS 位为 000，选择 TIM0 的触发输出 TIM0_TRGO 作为 TIM1 的从模式触发输入。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 SMS 位为 111，将 TIM1 的从模式配置为外部时钟模式 1。
- 配置定时器 TIM1 的 TIM1_CONR2 的 CNTEN 位为 1，启动定时器 TIM1。
- 配置定时器 TIM0 的 TIM0_CONR2 的 CNTEN 位为 1，启动定时器 TIM0。

经过以上配置后，TIM0 更新事件 UEV 的上升沿会触发 TIM1 的计数器计数，这样实现

了定时器 TIM0 作为定时器 TIM1 的预分频器的功能，可以通过改变自动重载寄存器 TIM0_ARR 的值来改变分频系数。

Figure 115 显示了 TIM0 的更新事件触发 TIM1 计数的过程。

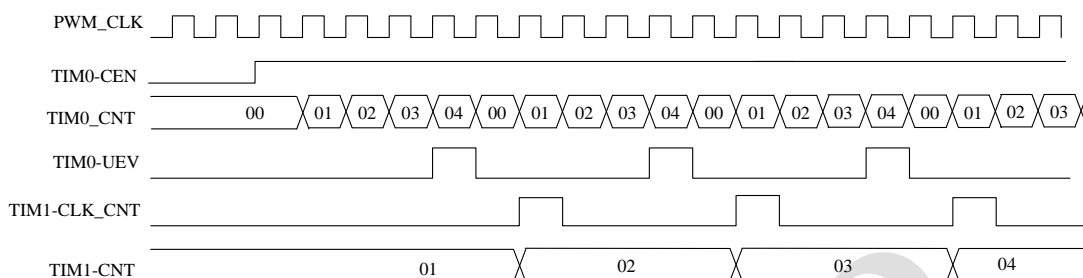


Figure 115 TIM0 的更新事件触发 TIM1 计数

12.5.7.2 一个定时器启动另一个定时器

通过配置可以使用定时器 TIM0 启动定时器 TIM1 的计数器。配置 TIM0_CONR0 寄存器的 MMC 位为不同的值，可以选择不同的信号作为 TIM1 计数器的启动信号。下面以 TIM0 的更新事件启动定时器 TIM1 为例说明详细的控制过程。

TIM0_CONR0 的 MMC 为 010，选择 TIM0 的更新事件 (UEV) 作为 TIM0 的触发输出去启动 TIM1，一旦 TIM0 产生更新事件，定时器 TIM1 的计数器从当前的数值开始计数。在收到触发信号时，定时器 TIM1 的 TIM1_CONR2 寄存器的 CNTEN 位被自动置位，同时计数器开始计数，将 TIM1_CONR2 寄存器的 CNTEN 位置 0，则 TIM1 的计数器停止计数。具体的配置过程如下：

- 配置定时器 TIM0 的主模式控制器。配置 TIM0 的 TIM0_CONR0 寄存器的 MMC 位为 010，选择定时器 TIM0 的更新事件 UEV 作为触发输出 TIM0_TRGO。
- 配置定时器 TIM0 的周期寄存器 TIM0_ARR。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 TRGS 位为 000，选择 TIM0 的触发输出 TIM0_TRGO 作为 TIM1 的从模式触发输入。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 SMS 位为 110，将 TIM1 的从模式配置为触发模式。
- 配置定时器 TIM0 的 TIM0_CONR2 的 CNTEN 位为 1，启动定时器 TIM0。

Figure 116 显示了定时器 TIM0 的更新事件启动 TIM1 计数器的过程。

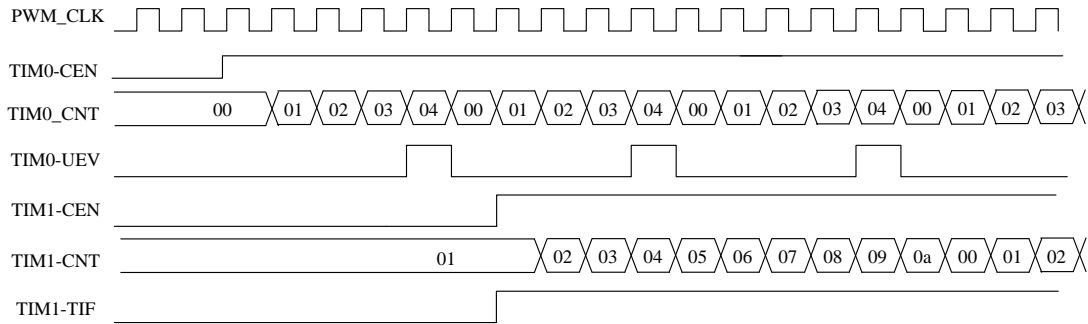


Figure 116 TIM0 的更新事件启动 TIM1

12.5.7.3 一个定时器使能另一个定时器

配置 TIM0_CONR0 寄存器的 MMC 位为不同的值，可以选择 TIM0 的不同信号作为 TIM1 计数器的使能信号。以使用定时器 TIM0 的基准参考信号 OCxREF 作为 TIM1 的计数器门控信号为例阐述详细的控制过程。

配置 TIM0 的 TIM0_CONR0 寄存器的 MMC 位为 100，选择定时器 TIM0 的 OC0REF 作为触发输出去控制 TIM1 的计数器使能，当 TIM0 的 OCxREF 为有效电平时 TIM1 的计数器计数，当 TIM0 的 OCxREF 为无效电平时 TIM1 的计数器停止计数。具体控制过程如下：

- 配置定时器 TIM0 的主模式控制器，配置 TIM0 的 TIM0_CONR0 寄存器的 MMC 位为 100，选择定时器 TIM0 的 OC0REF 作为触发输出 TIM0_TRGO。
- 配置定时器 TIM0 的计数周期寄存器 TIM0_ARR、输入输出比较寄存器 TIM0_CC0RU。
- 配置 TIM0_CC0MR 的 CC0MS 位为 000，将 TIM0 配置为输出比较模式。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 TRGS 位为 000，选择 TIM0 的触发输出 TIM0_TRGO 作为 TIM1 的从模式触发输入。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 SMS 位为 101，将 TIM1 的从模式配置为门控模式。
- 配置定时器 TIM1 的 TIM1_CONR2 的 CNTEN 位为 1，启动定时器 TIM1。
- 配置定时器 TIM0 的 TIM0_CONR2 的 CNTEN 位为 1，启动定时器 TIM0。

Figure 117 显示了定时器 TIM0 的 OCxREF 信号控制 TIM1 计数器的过程。

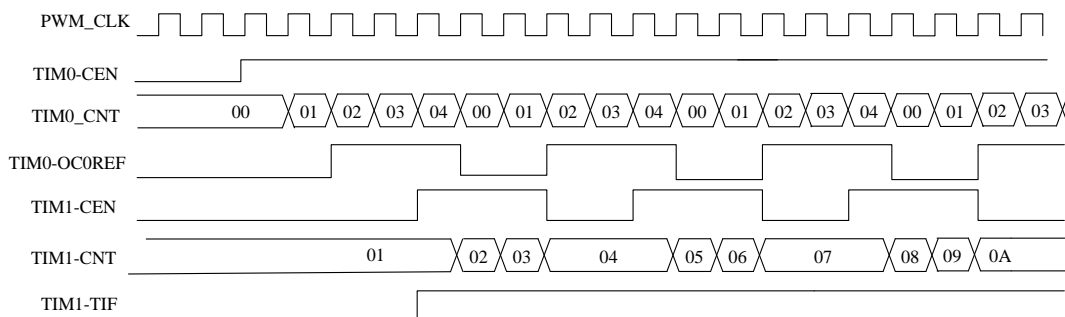


Figure 117 TIM0 的 OCOREF 控制 TIM1 计数器计数

Figure 117 的例子中，在定时器 TIM1 启动之前，它们的计数器和预分频器未初始化，因此它们从当前的数值开始计数。可以在启动定时器 TIM0 之前复位这两个定时器，并且可以向计数器中写入任意数值，使它们从给定的数值开始计数，写 TIMx_EGR 寄存器的 UG 位即可复位定时器，具体控制过程如下：

- 配置定时器 TIM0 的主模式控制器。配置 TIM0 的 TIM0_CONR0 寄存器的 MMC 位为 000，选择定时器 TIM0 的计数器使能作为触发输出 TIM0_TRGO。
- 配置定时器 TIM0 的计数周期寄存器 TIM0_ARR、输入输出比较寄存器 TIM0_CC0RU。
- 配置 TIM0_CC0MR 的 CC0MS 位为 000，将 TIM0 配置为输出比较模式。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 TRGS 位为 000，选择 TIM0 的触发输出 TIM0_TRGO 作为 TIM1 的从模式触发输入。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 SMS 位为 101，将 TIM1 的从模式配置为门控模式。
- 配置 TIM0_EGR 寄存器的 UG 位为 1，复位定时器 TIM0。
- 配置 TIM1_EGR 寄存器的 UG 位为 1，复位定时器 TIM1。
- 配置计数器的初始值。例如配置 TIM1_CNTH 寄存器为 0x00，配置 TIM1_CNTL 寄存器为 0x05。
- 配置定时器 TIM1 的 TIM1_CONR2 的 CNTEN 位为 1，启动定时器 TIM1。
- 配置定时器 TIM0 的 TIM0_CONR2 的 CNTEN 位为 1，启动定时器 TIM0。
- 计数一段时间后，配置定时器 TIM0 的 TIM0_CONR2 的 CNTEN 位为 0，停止定时器 TIM0，同时也停止定时器 TIM1。

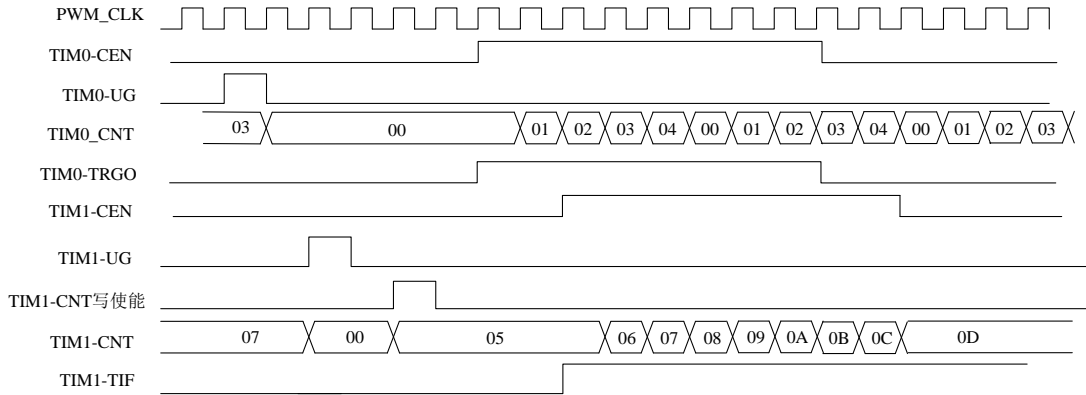


Figure 118 TIM0 的使能信号 CEN 控制 TIM1 计数器计数

12.5.7.4 一个定时器复位另一个定时器

配置 TIM0_CONR0 寄存器的 MMC 位为不同的值，可以选择 TIM0 的不同信号作为 TIM1 计数器的复位信号。使用 TIM0 的更新事件复位 TIM1 的计数器为例对控制过程进行详细阐述。

通过配置，可以使用定时器 TIM0 的更新事件 UEV 作为复位信号来初始化 TIM1 的计数器。具体控制过程如下：

- 配置定时器 TIM0 的主模式控制器。配置 TIM0 的 TIM0_CONR0 寄存器的 MMC 位为 010，选择定时器 TIM0 的更新事件 UEV 作为触发输出 TIM0_TRGO。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 TRGS 位为 000，选择 TIM0 的触发输出 TIM0_TRGO 作为 TIM1 的从模式触发输入。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 SMS 位为 100，将 TIM1 的从模式配置为复位模式。
- 配置定时器 TIM1 的 TIM1_CONR2 的 CNTEN 位为 1，启动定时器 TIM1。
- 配置定时器 TIM0 的 TIM0_CONR2 的 CNTEN 位为 1，启动定时器 TIM0。

改变 TIM0_CONR0 寄存器的 MMC 位域的配置，可以选择定时器 TIM0 的不同的信号初始化定时器 TIM1 的计数器。Figure 119 显示了 TIM0 的更新事件初始化 TIM1 计数器的过程。

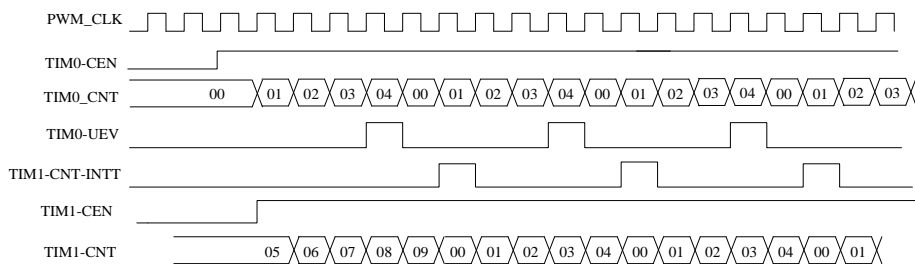


Figure 119 TIM0 的更新事件复位 TIM1

12.5.7.5 使用一个外部触发信号同时启动两个定时器

若将定时器 TIM0 配置为主从模式，可以使用定时器 TIM0 外部触发信号 CH0IN 同时启动 TIM0 和定时器 TIM1。具体控制过程如下：

- 配置定时器 TIM0 的从模式控制器。配置定时器 TIM0 的 TIM0_TGICR0 寄存器的 MSM 位为 1，使能定时器 TIM0 的主从模式；配置定时器 TIM0 的 TIM0_TGICR0 寄存器的 SMS 位为 110，将 TIM0 的从模式配置为触发模式；配置 TIM0 的 TIM0_TGICR0 寄存器的 TRGS 位为 101，选择定时器 TIM0 的 CH0IN 作为从模式触发输入信号。
- 配置定时器 TIM0 的主模式控制器。配置 TIM0 的 TIM0_CONR0 寄存器的 MMC 位为 000，选择定时器 TIM0 的计数器使能作为触发输出 TIM0_TRGO。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 TRGS 位为 000，选择 TIM0 的触发输出 TIM0_TRGO 作为 TIM1 的从模式触发输入。
- 配置定时器 TIM1 的 TIM1_TGICR0 寄存器的 SMS 位为 110，将 TIM1 的从模式配置为触发模式。

Figure 120 显示了 TIM0 的外部信号 CH0IN 同时启动 TIM0 和 TIM1 的过程。

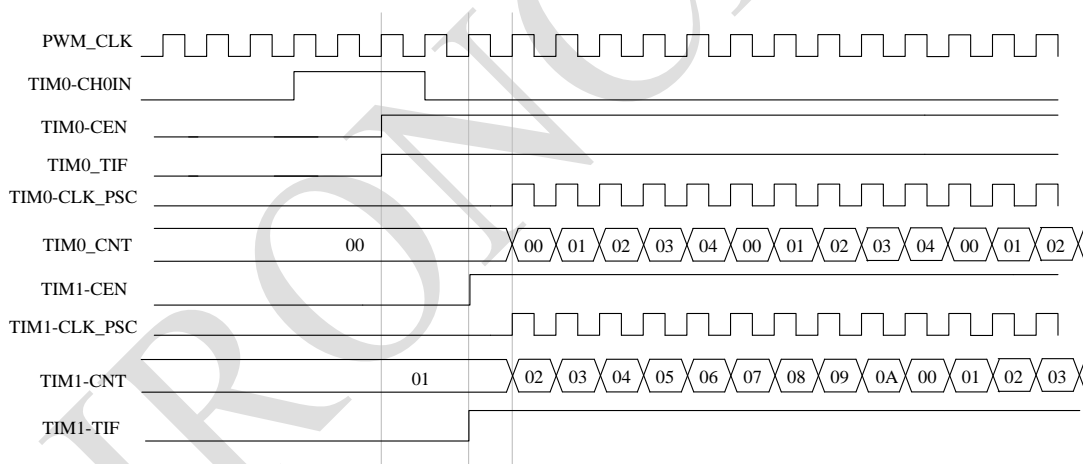


Figure 120 外部信号 PWMx_ETR 同步启动 TIM0 和 TIM1

12.6 寄存器描述

每个高级定时器包含 44 个寄存器，TIM0 的寄存器在 Table 198 中列出，TIM1 的寄存器在 Table 199 中列出，两个高级定时器的寄存器除了地址不同外，功能相同。高级定时器的寄存器对应的 SFR PAGE 为 4。

Table 198 TIM0 寄存器 (SFRPAGE=0x04)

地址	寄存器名	属性	复位值	功能描述
0xBB	TIM0_CONR0	W/R	0x00	TIM0 控制寄存器 0
0x89	TIM0_CONR1	W/R	0x00	TIM0 控制寄存器 1

地址	寄存器名	属性	复位值	功能描述
0x8A	TIM0_CONR2	W/R	0x00	TIM0 控制寄存器 2
0x8B	TIM0_CONR3	W/R	0x20	TIM0 控制寄存器 3
0x8C	TIM0_TGICR0	W/R	0x00	从模式控制寄存器 0
0x8D	TIM0_TGICR1	W/R	0x00	从模式控制寄存器 1
0x8E	TIM0_IER	W/R	0x00	TIM0 中断使能寄存器 0
0xB7	TIM0_IER_NOCMP	W/R	0x00	TIM0 中断使能寄存器 1
0x8F	TIM0_CC0MR	W/R	0x00	TIM0 通道 CC0 输入捕获/输出比较模式控制寄存器
0x91	TIM0_CC1MR	W/R	0x00	TIM0 通道 CC1 输入捕获/输出比较模式控制寄存器
0x92	TIM0_CC2MR	W/R	0x00	TIM0 通道 CC2 输入捕获/输出比较模式控制寄存器
0x93	TIM0_CC3MR	W/R	0x00	TIM0 通道 CC3 输入捕获/输出比较模式控制寄存器
0x94	TIM0_CCENR	W/R	0x00	通道使能控制寄存器
0x95	TIM0_CCPS	W/R	0x00	通道极性控制寄存器
0x96	TIM0_DTG	W/R	0x00	TIM0 死区时间寄存器
0x97	TIM0_ARRL	W/R	0x00	TIM0 周期值自动装载寄存器的低 8 位
0x99	TIM0_ARRH	W/R	0x00	TIM0 周期值自动装载寄存器的高 8 位
0x9A	TIM0_PSCL	W/R	0x00	TIM0 预分频寄存器的低 8 位
0x9B	TIM0_PSCH	W/R	0x00	TIM0 预分频寄存器的高 8 位
0x9C	TIM0_RCR	W/R	0x00	TIM0 重复计数寄存器
0x9D	TIM0_CC0RUL	W/R	0x00	通道 CC0 捕获/向上计数比较寄存器的低 8 位
0x9E	TIM0_CC0RUH	W/R	0x00	通道 CC0 捕获/向上计数比较寄存器的高 8 位
0x9F	TIM0_CC0RDL	W/R	0x00	通道 CC0 向下计数比较寄存器的低 8 位
0xA1	TIM0_CC0RDH	W/R	0x00	通道 CC0 向下计数比较寄存器的高 8 位

地址	寄存器名	属性	复位值	功能描述
0xA2	TIM0_CC1RUL	W/R	0x00	通道 CC1 捕获/向上计数比较寄存器的低 8 位
0xA3	TIM0_CC1RUH	W/R	0x00	通道 CC1 捕获/向上计数比较寄存器的高 8 位
0xA4	TIM0_CC1RDL	W/R	0x00	通道 CC1 向下计数比较寄存器的低 8 位
0xA5	TIM0_CC1RDH	W/R	0x00	通道 CC1 向下计数比较寄存器的高 8 位
0xA6	TIM0_CC2RUL	W/R	0x00	通道 CC2 捕获/向上计数比较寄存器的低 8 位
0xA7	TIM0_CC2RUH	W/R	0x00	通道 CC2 捕获/向上计数比较寄存器的高 8 位
0xA9	TIM0_CC2RDL	W/R	0x00	通道 CC2 向下计数比较寄存器的低 8 位
0xAA	TIM0_CC2RDH	W/R	0x00	通道 CC2 向下计数比较寄存器的高 8 位
0xAB	TIM0_CC3RUL	W/R	0x00	通道 CC3 捕获/向上计数比较寄存器的低 8 位
0xAC	TIM0_CC3RUH	W/R	0x00	通道 CC3 捕获/向上计数比较寄存器的高 8 位
0xAD	TIM0_CC3RDL	W/R	0x00	通道 CC3 向下计数比较寄存器的低 8 位
0xAE	TIM0_CC3RDH	W/R	0x00	通道 CC3 向下计数比较寄存器的高 8 位
0xAF	TIM0_BRKC	W/R	0x00	TIM0 刹车控制寄存器
0xB1	TIM0_EGR	W	0x00	TIM0 事件产生寄存器
0xB2	TIM0_CMPLMTARY	W/R	0x01	TIM0 互补模式控制寄存器
0xB3	TIM0_ISR0	W/R	0x00	TIM0 中断标志寄存器 0
0xB4	TIM0_ISR1	W/R	0x00	TIM0 中断标志寄存器 1
0xB9	TIM0_ISR2	W/R	0x00	TIM0 中断标志寄存器 2
0xB5	TIM0_CNTL	W/R	0x00	TIM0 计数器的低 8 位
0xB6	TIM0_CNTH	W/R	0x00	TIM0 计数器的高 8 位

Table 199 TIM1 寄存器 (SFRPAGE=0x04)

地址	寄存器名	属性	复位值	功能描述
0xFB	TIM1_CONR0	W/R	0x00	TIM1 控制寄存器 0
0xC9	TIM1_CONR1	W/R	0x00	TIM1 控制寄存器 1
0xCA	TIM1_CONR2	W/R	0x00	TIM1 控制寄存器 2
0xCB	TIM1_CONR3	W/R	0x20	TIM1 控制寄存器 3
0xCC	TIM1_TGICR0	W/R	0x00	从模式控制寄存器 0
0xCD	TIM1_TGICR1	W/R	0x00	从模式控制寄存器 1
0xCE	TIM1_IER	W/R	0x00	TIM1 中断使能寄存器 0
0xF7	TIM1_IER_NOCMP	W/R	0x00	TIM1 中断使能寄存器 1
0xCF	TIM1_CC0MR	W/R	0x00	TIM1 通道 CC0 输入捕获/输出比较模式控制寄存器
0xD1	TIM1_CC1MR	W/R	0x00	TIM1 通道 CC1 输入捕获/输出比较模式控制寄存器
0xD2	TIM1_CC2MR	W/R	0x00	TIM1 通道 CC2 输入捕获/输出比较模式控制寄存器
0xD3	TIM1_CC3MR	W/R	0x00	TIM1 通道 CC3 输入捕获/输出比较模式控制寄存器
0xD4	TIM1_CCENR	W/R	0x00	通道使能控制寄存器
0xD5	TIM1_CCPS	W/R	0x00	通道极性控制寄存器
0xD6	TIM1_DTG	W/R	0x00	TIM1 死区时间寄存器
0xD7	TIM1_ARRL	W/R	0x00	TIM1 周期值自动装载寄存器的低 8 位
0xD9	TIM1_ARRH	W/R	0x00	TIM1 周期值自动装载寄存器的高 8 位
0xDA	TIM1_PSCL	W/R	0x00	TIM1 预分频寄存器的低 8 位
0xDB	TIM1_PSCH	W/R	0x00	TIM1 预分频寄存器的高 8 位
0xDC	TIM1_RCR	W/R	0x00	TIM1 重复计数寄存器
0xDD	TIM1_CC0RUL	W/R	0x00	通道 CC0 捕获/向上计数比较寄存器的低 8 位
0xDE	TIM1_CC0RUH	W/R	0x00	通道 CC0 捕获/向上计数比较寄存器的高 8 位

地址	寄存器名	属性	复位值	功能描述
0xDF	TIM1_CC0RDL	W/R	0x00	通道 CC0 向下计数比较寄存器的低 8 位
0xE1	TIM1_CC0RDH	W/R	0x00	通道 CC0 向下计数比较寄存器的高 8 位
0xE2	TIM1_CC1RUL	W/R	0x00	通道 CC1 捕获/向上计数比较寄存器的低 8 位
0xE3	TIM1_CC1RUH	W/R	0x00	通道 CC1 捕获/向上计数比较寄存器的高 8 位
0xE4	TIM1_CC1RDL	W/R	0x00	通道 CC1 向下计数比较寄存器的低 8 位
0xE5	TIM1_CC1RDH	W/R	0x00	通道 CC1 向下计数比较寄存器的高 8 位
0xE6	TIM1_CC2RUL	W/R	0x00	通道 CC2 捕获/向上计数比较寄存器的低 8 位
0xE7	TIM1_CC2RUH	W/R	0x00	通道 CC2 捕获/向上计数比较寄存器的高 8 位
0xE9	TIM1_CC2RDL	W/R	0x00	通道 CC2 向下计数比较寄存器的低 8 位
0xEA	TIM1_CC2RDH	W/R	0x00	通道 CC2 向下计数比较寄存器的高 8 位
0xEB	TIM1_CC3RUL	W/R	0x00	通道 CC3 捕获/向上计数比较寄存器的低 8 位
0xEC	TIM1_CC3RUH	W/R	0x00	通道 CC3 捕获/向上计数比较寄存器的高 8 位
0xED	TIM1_CC3RDL	W/R	0x00	通道 CC3 向下计数比较寄存器的低 8 位
0xEE	TIM1_CC3RDH	W/R	0x00	通道 CC3 向下计数比较寄存器的高 8 位
0xEF	TIM1_BRKC	W/R	0x00	TIM1 刹车控制寄存器
0xF1	TIM1_EGR	W/R	0x00	TIM1 事件产生寄存器
0xF2	TIM1_CMPLMTARY	W/R	0x01	TIM1 互补模式控制寄存器
0xF3	TIM1_ISR0	W/R	0x00	TIM1 中断标志寄存器 0

地址	寄存器名	属性	复位值	功能描述
0xF4	TIM1_ISR1	W/R	0x00	TIM1 中断标志寄存器 1
0xF9	TIM1_ISR2	W/R	0x00	TIM1 中断标志寄存器 2
0xF5	TIM1_CNTRL	W/R	0x00	TIM1 计数器的低 8 位
0xF6	TIM1_CNTH	W/R	0x00	TIM1 计数器的高 8 位

12.6.1 TIMx 控制寄存器 0 (TIMx_CONR0) [TIM0:0xBB/TIM1:0xFB]

Table 200 TIMx 控制寄存器 0 (TIMx_CONR0)

参数名	比特位	属性	复位值	描述
CLKDIV	1-0	W/R	0	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。 00: $T_{dts_clk} = T_{pwm_clk}$ 01: $T_{dts_clk} = 2 \times T_{pwm_clk}$ 10: $T_{dts_clk} = 4 \times T_{pwm_clk}$ 11: 保留
CHOINSEL	2	W/R	0	通道 CC0 输入信号选择寄存器： 0: PWMx_CH0 管脚上的信号连接到通道 CC0 1: PWM x_CH0、PWM x_CH1 和 PWM x_CH2 管脚上的信号异或后连接到通道 CC0
MMC	5-3	W/R	0	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）： 000: 复位信号 <ul style="list-style-type: none"> 当 TIMx_EGR 寄存器的 UG 置位时，或由从模式控制器产生复位时，复位信号作为触发输出（TRGO）。 001: 使能信号 <ul style="list-style-type: none"> 计数器使能信号被用于作为触发输出（TRGO）。

参数名	比特位	属性	复位值	描述
				<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 TIMx_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 101: 通道 CC1 的基准参考信号 OC1REF 110: 通道 CC2 的基准参考信号 OC2REF 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 的有效沿更新。</p>
CCPE	7	W/R	0	<p>捕获/比较控制位预装载使能控制寄存器：</p> <p>0: 捕获/比较控制位 CCxE, CCxNE, CCxP, CCxNP 和 OCxMS 不是预装载的。</p> <p>1: 捕获/比较控制位 CCxE, CCxNE, CCxP, CCxNP 和 OCxMS 位是预装载的，设置该</p>

参数名	比特位	属性	复位值	描述
				位后，这些寄存器只在设置了 COM 位或 TRGI 的有效沿更新。

12.6.2 TIMx 控制寄存器 1 (TIMx_CONR1) [TIM0:0x89/TIM1:0xC9]

Table 201 TIMx 控制寄存器 1 (TIMx_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 (TIMx_BRKC 寄存器) 级别 1、2 或 3, 该位不能被修改。
OISON	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 (TIMx_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 OISON。
OIS2	4	W/R	0	通道 CC2 输出的空闲状态 (OC2 输出状态), 参见 OIS0。
OIS2N	5	W/R	0	通道 CC2 输出的空闲状态 (OC2N 输出状态), 参见 OISON。
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

12.6.3 TIMx 控制寄存器 2 (TIMx_CONR2) [TIM0:0x8A/TIM1:0xCA]

Table 202 TIMx 控制寄存器 2 (TIMx_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 TIMx_ARR 寄存器预装载功能。 1：打开 TIMx_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，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

参数名	比特位	属性	复位值	描述
				更新中断使能，则同时产生一个更新中断： <ul style="list-style-type: none"> ● 计数器上溢/下溢 ● 设置 TIMx_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： <ul style="list-style-type: none"> ● 计数器上溢/下溢 ● 设置 TIMx_EGR 寄存器的 UG 位 ● 从模式控制器产生的复位信号。
Reserved	6	W/R	0	保留
CNTEN	7	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。

12.6.4 TIMx 控制寄存器 3 (TIMx_CONR3) [TIM0:0x8B/TIM1:0xCB]

Table 203 TIMx 控制寄存器 3(TIMx_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中断。

参数名	比特位	属性	复位值	描述
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 TIMx_EGR 寄存器的 UG 位为 1。 ● 从模式控制器产生的复位信号。
SYMM	5	W/R	1	中央对齐模式下比较寄存器配置模式： 0：非对称模式，中央对齐模式下，比较寄存器 TIMx_CCxRU 和 TIMx_CCxRD 分开独立配置，可以配置为相同或不同的值。 1：对称模式，中央对齐模式下，比较寄存器 TIMx_CCxRU 和 TIMx_CCxRD 的值相同，此种模式下只需要配置 TIMx_CCxRU，配置 TIMx_CCxRU 的同时硬件将 TIMx_CCxRD 的值配置为 TIMx_CCxRU 的值。 注：此寄存器需在比较寄存器 TIMx_CCxRU 和 TIMx_CCxRD 配置前配置。
Reserved	7-6	R	0	保留

12.6.5 TIMx 从模式控制寄存器 0 (TIMx_TGICR0) [TIM0:0x8C/TIM1:0xCC]

Table 204 TIMx 从模式控制寄存器 0 (TIMx_TGICR0)

参数名	比特位	属性	复位值	描述
MSM	0	W/R	0	主从模式控制： 0：无作用 1：外部触发输入可以同步启动两个定时器。将外部触发信号连接至第一个定时器，将第一个定时器的使能信号输出至第二个定时器作为触发输入，当外部触发输入有效时，两个定时器同时开始计数。
TRGS	3-1	W/R	0	从模式控制器触发信号选择寄存器： 000：另一个高级定时器的触发输出信号 TRGO。 001：0； 010：0； 100：通道 CC0 的边沿检测信号 CH0INFP_ED； 101：通道 CC0 滤波后的定时器输入 CH0INFP； 110：通道 CC1 滤波后的定时器输入 CH1INFP； 其他：外部触发输入 PWMx_ETR。
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 (TIMx_CONR2.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

参数名	比特位	属性	复位值	描述
				<p>计数器在 CH0INFP0 的边沿向上/向下计数。</p> <p>011: 编码器模式 3</p> <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。 <p>100: 复位模式</p> <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。 <p>101: 门控模式</p> <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)。 <p>110: 触发模式</p> <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。 <p>111: 外部时钟模式 1</p> <ul style="list-style-type: none"> ● 选中的触发输入的有效沿驱动计数器计数。 <p>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。</p>
Reserved	7	R	0	保留

12.6.6 TIMx 从模式控制寄存器 1 (TIMx_TGICR1) [TIM0:0x8D/TIM1:0xCD]

Table 205 TIMx 从模式控制寄存器 1 (TIMx_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:

参数名	比特位	属性	复位值	描述
				0: PWMx_ETR 不反相, 高电平或上升沿有效。 1: PWMx_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 PWMx_ETRF 信号的有效沿驱动。 注 1: 设置 EXTCLKEN 位与选择外部时钟模式 1 并将 TRGI 连到 PWMx_ETRF 具有相同功效。 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 PWMx_ETRF。
EXTDIV	3-2	W/R	0	外部触发信号 PWMx_ETR 预分频: 00: 1 分频 01: 2 分频 10: 4 分频 11: 8 分频
EXTFLT	7-4	W/R	0	外部触发信号 PWMx_ETR 滤波时钟频率和滤波周期数目 (N) 选择 0000: 不滤波 0001: 滤波时钟频率为 f_{pwm_clk} , N=2 0010: 滤波时钟频率为 f_{pwm_clk} , N=4 0011: 滤波时钟频率为 f_{pwm_clk} , N=8 0100: 滤波时钟频率为 $f_{dts_clk}/2$, N=6 0101: 滤波时钟频率为 $f_{dts_clk}/2$, N=8 0110: 滤波时钟频率为 $f_{dts_clk}/4$, N=6 0111: 滤波时钟频率为 $f_{dts_clk}/4$, N=8 1000: 滤波时钟频率为 $f_{dts_clk}/8$, N=6 1001: 滤波时钟频率为 $f_{dts_clk}/8$, N=8 1010: 滤波时钟频率为 $f_{dts_clk}/16$, N=5

参数名	比特位	属性	复位值	描述
				1011: 滤波时钟频率为 $f_{dts_clk}/16$, N=6 1100: 滤波时钟频率为 $f_{dts_clk}/16$, N=8 1101: 滤波时钟频率为 $f_{dts_clk}/32$, N=5 1110: 滤波时钟频率为 $f_{dts_clk}/32$, N=6 1111: 滤波时钟频率为 $f_{dts_clk}/32$, N=8

12.6.7 TIMx 中断使能寄存器 0 (TIMx_IER) [TIM0:0x8E/TIM1:0xCE]

Table 206 TIMx 中断使能寄存器 0 (TIMx_IER)

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

12.6.8 TIMx 中断使能寄存器 1 (TIMx_IER_NOCMP)

[TIM0:0xB7/TIM7:0xF7]

Table 207 TIMx 中断使能寄存器 1 (TIMx_IER_NOCMP)

参数名	比特位	属性	复位值	描述
NOCMP_CC0IE	0	W/R	0	独立模式下第一路 PWM 比较中断使能: 0: 禁止第一路 PWM 比较中断 1: 允许第一路 PWM 比较中断
NOCMP_CC1IE	1	W/R	0	非互补模式下第二路 PWM 比较中断使能: 0: 禁止第二路 PWM 比较中断 1: 允许第二路 PWM 比较中断
NOCMP_CC2IE	2	W/R	0	非互补模式下第三路 PWM 比较中断使能: 0: 禁止第三路 PWM 比较中断 1: 允许第三路 PWM 比较中断
NOCMP_CC3IE	3	W/R	0	非互补模式下第四路 PWM 比较中断使能: 0: 禁止第四路 PWM 比较中断 1: 允许第四路 PWM 比较中断
NOCMP_CC4IE	4	W/R	0	非互补模式下第五路 PWM 比较中断使能: 0: 禁止第五路 PWM 比较中断 1: 允许第五路 PWM 比较中断
NOCMP_CC5IE	5	W/R	0	非互补模式下第六路 PWM 比较中断使能: 0: 禁止第六路 PWM 比较中断 1: 允许第六路 PWM 比较中断
NOCMP_CC6IE	6	W/R	0	非互补模式下第七路 PWM 比较中断使能: 0: 禁止第七路 PWM 比较中断 1: 允许第七路 PWM 比较中断
Reserved	7	R	0	保留

12.6.9 TIMx 通道 CC0 输入捕获/输出比较模式控制寄存器 (TIMx_CC0MR)

[TIM0:0x8F/TIM1:0xCF]

Table 208 TIMx 通道 CC0 输入捕获/输出比较模式控制寄存器 (输出比较模式)

(TIMx_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:

参数名	比特位	属性	复位值	描述
				<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，输入捕获信号 CH0INC 来自 TRC。</p> <p>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。</p>
OC0FE	2	W/R	0	<p>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：</p> <p>0: 根据计数器与 TIMx_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。</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 CC0 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 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>
OC0PEN	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 TIMx_CC0RU 和 TIMx_CC0RD 的预装载功能，可随时将数据写入寄存器 TIMx_CC0RU 或 TIMx_CC0RD，且新值立即生效。</p> <p>1：开启寄存器 TIMx_CC0RU 和 TIMx_CC0RD 的预装载功能，读写操作仅对</p>

参数名	比特位	属性	复位值	描述
				<p>预装载寄存器操作，TIMx_CC0RU 和 TIMx_CC0RD 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。</p> <p>单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC0MS	6-4	W/R	000	<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 TIMx_CC0RU 与计数器间的比较结果对 OC0REF 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 TIMx_CC0RU 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 TIMx_CC0RU 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 TIMx_CC0RU 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

参数名	比特位	属性	复位值	描述
				<p>设置为无效电平。</p> <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 $TIMx_CNT < TIMx_CC0RU$, OC0REF 为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0); 向下计数时, 若 $TIMx_CNT > TIMx_CC0RU$, 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 $TIMx_CNT < TIMx_CC0RU$, OC0REF 为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0); 向下计数时, 若 $TIMx_CNT > TIMx_CC0RD$, OC0REF 为无效电平 (OC0REF=0), 否则为有效电平 (OC0REF=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 $TIMx_CNT < TIMx_CC0RU$, OC0REF 为无效电平 (OC0REF=0), 否则为有效电平 (OC0REF=1); 向下计数时, 若 $TIMx_CNT > TIMx_CC0RU$, 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 $TIMx_CNT < TIMx_CC0RU$, OC0REF 为无效电平 (OC0REF=0), 否则为有效电平 (OC0REF=1); 向下计数时, 若 $TIMx_CNT > TIMx_CC0RD$, OC0REF

参数名	比特位	属性	复位值	描述
				为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0)。
OC0CE	7	W/R	0	比较输出 OC0REF 清除控制: 0: OC0REF 不受 PWMx_ETRF 输入的影响。 1: PWMx_ETRF 输入有效电平时, OC0REF 输出无效电平。

TIMx 通道 CC0 输入捕获/输出比较模式控制寄存器 (输入捕获模式) (TIMx_CC0MR)

参数名	比特位	属性	复位值	描述
CC0MS	1-0	W/R	00	捕获/比较功能选择寄存器: 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。
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。
IC0FS	7-4	W/R	0	通道 CC0 输入信号 CH0IN 滤波时钟频率和滤波周期数目 (N) 选择: 0000: 不滤波

参数名	比特位	属性	复位值	描述
				0001: 滤波时钟频率为 f_{pwm_clk} , N=2 0010: 滤波时钟频率为 f_{pwm_clk} , N=4 0011: 滤波时钟频率为 f_{pwm_clk} , N=8 0100: 滤波时钟频率为 $f_{dts_clk}/2$, N=6 0101: 滤波时钟频率为 $f_{dts_clk}/2$, N=8 0110: 滤波时钟频率为 $f_{dts_clk}/4$, N=6 0111: 滤波时钟频率为 $f_{dts_clk}/4$, N=8 1000: 滤波时钟频率为 $f_{dts_clk}/8$, N=6 1001: 滤波时钟频率为 $f_{dts_clk}/8$, N=8 1010: 滤波时钟频率为 $f_{dts_clk}/16$, N=5 1011: 滤波时钟频率为 $f_{dts_clk}/16$, N=6 1100: 滤波时钟频率为 $f_{dts_clk}/16$, N=8 1101: 滤波时钟频率为 $f_{dts_clk}/32$, N=5 1110: 滤波时钟频率为 $f_{dts_clk}/32$, N=6 1111: 滤波时钟频率为 $f_{dts_clk}/32$, N=8

12.6.10TIMx 通道 CC1 输入捕获/输出比较模式控制寄存器 (TIMx_CC1MR)

[TIM0:0x91/TIM1:0xD1]

Table 209 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。
OC1FE	2	W/R	0	输出比较快速使能控制寄存器, 该位用于加快

参数名	比特位	属性	复位值	描述
				<p>CC1 输出对触发输入事件的响应:</p> <p>0: 根据计数器与 TIMx_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 TIMx_CC1RU 和 TIMx_CC1RD 的预装载功能, 可随时将数据写入寄存器 TIMx_CC1RU 或 TIMx_CC1RD, 且新值立即生效。</p> <p>1: 开启寄存器 TIMx_CC1RU 和 TIMx_CC1RD 的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CC1RU 和 TIMx_CC1RD 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。</p> <p>单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC1MS	6-4	W/R	000	<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, 从而决定输出管脚上的 PWM 输出波形。</p> <p>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>

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> ● 输出比较寄存器 TIMx_CC1RU 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 TIMx_CC1RU 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 TIMx_CC1RU 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 TIMx_CC1RU 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 $TIMx_CNT < TIMx_CC1RU$, OC1REF 为有效电平 (OC1REF=1), 否则为无效电平 (OC1REF=0); 向下计数时, 若 $TIMx_CNT > TIMx_CC1RU$, OC1REF 为无效电平 (OC1REF=0), 否则为有效电平 (OC1REF=1)。 ● 中央对齐计数模式下: 向上计数时, 若 $TIMx_CNT < TIMx_CC1RU$, OC1REF 为有效电平 (OC1REF=1), 否则为无效电平 (OC1REF=0); 向下计数时, 若 $TIMx_CNT > TIMx_CC1RD$, OC1REF 为无

参数名	比特位	属性	复位值	描述
				效电平（OC1REF=0），否则为有效电平（OC1REF=1）。 111: PWM 模式 2 <ul style="list-style-type: none"> ● 边沿计数模式下：向上计数时，若 $TIMx_CNT < TIMx_CC1RU$，OC1REF 为无效电平（OC1REF=0），否则为有效电平（OC1REF=1）；向下计数时，若 $TIMx_CNT > TIMx_CC1RU$，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 $TIMx_CNT < TIMx_CC1RU$，OC1REF 为无效电平（OC1REF=0），否则为有效电平（OC1REF=1）；向下计数时，若 $TIMx_CNT > TIMx_CC1RD$，OC1REF 为有效电平（OC1REF=1），否则为无效电平（OC1REF=0）。
OC1CE	7	W/R	0	比较输出 OC1REF 清除控制寄存器： 0: OC1REF 不受 PWMx_ETRF 输入的影响。 1: PWMx_ETRF 输入有效电平时，设置 OC1REF 为无效电平。

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

参数名	比特位	属性	复位值	描述
				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。
IC1FS	7-4	W/R	0	通道 CC1 输入信号 CH1IN 滤波时钟频率和滤波周期数目（N）选择： 0000：不滤波 0001：滤波时钟频率为 f_{pwm_clk} ，N=2 0010：滤波时钟频率为 f_{pwm_clk} ，N=4 0011：滤波时钟频率为 f_{pwm_clk} ，N=8 0100：滤波时钟频率为 $f_{dts_clk}/2$ ，N=6 0101：滤波时钟频率为 $f_{dts_clk}/2$ ，N=8 0110：滤波时钟频率为 $f_{dts_clk}/4$ ，N=6 0111：滤波时钟频率为 $f_{dts_clk}/4$ ，N=8 1000：滤波时钟频率为 $f_{dts_clk}/8$ ，N=6 1001：滤波时钟频率为 $f_{dts_clk}/8$ ，N=8 1010：滤波时钟频率为 $f_{dts_clk}/16$ ，N=5 1011：滤波时钟频率为 $f_{dts_clk}/16$ ，N=6 1100：滤波时钟频率为 $f_{dts_clk}/16$ ，N=8 1101：滤波时钟频率为 $f_{dts_clk}/32$ ，N=5 1110：滤波时钟频率为 $f_{dts_clk}/32$ ，N=6 1111：滤波时钟频率为 $f_{dts_clk}/32$ ，N=8

12.6.11 TIMx 通道 CC2 输入捕获/输出比较模式控制寄存器 (TIMx_CC2MR) [TIM0:0x92/TIM1:0xD2]

Table 210 TIMx 通道 CC2 输入捕获/输出比较模式控制寄存器 (输出比较模式)
(TIMx_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 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。
OC2FE	2	W/R	0	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： 0: 根据计数器与 TIMx_CC2RU 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。 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。 OC2FE 只在通道配置为 PWM 模式 1 或 PWM 模式 2 时起作用。
OC2PEN	3	W/R	0	输出比较寄存器的预装载使能控制寄存器： 0: 禁止寄存器 TIMx_CC2RU 和 TIMx_CC2RD 的预装载功能，可随时将数据写入寄存器 TIMx_CC2RU 或 TIMx_CC2RD，且新值立即生效。

参数名	比特位	属性	复位值	描述
				<p>1: 开启寄存器 TIMx_CC2RU 和 TIMx_CC2RD 的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CC2RU 和 TIMx_CC2RD 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。</p> <p>单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC2MS	6-4	W/R	000	<p>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。</p> <p>000: 冻结。</p> <ul style="list-style-type: none"> ● 输出比较寄存器 TIM0_CC2RU 与计数器间的比较结果对 OC2REF 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 TIMx_CC2RU 与计数器的值相同时, 设置 OC2REF 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 TIMx_CC2RU 与计数器的值相同时, 设置 OC2REF 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 TIMx_CC2RU 与计数器的值相同时, OC2REF 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 OC2REF 设置为无效电平。

参数名	比特位	属性	复位值	描述
				<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 $TIMx_CNT < TIMx_CC2RU$, OC2REF 为有效电平 (OC2REF=1), 否则为无效电平 (OC2REF=0); 向下计数时, 若 $TIMx_CNT > TIMx_CC2RU$, 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 $TIMx_CNT < TIMx_CC2RU$, OC2REF 为有效电平 (OC2REF=1), 否则为无效电平 (OC2REF=0); 向下计数时, 若 $TIMx_CNT > TIMx_CC2RD$, 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 $TIMx_CNT < TIMx_CC2RU$, OC2REF 为无效电平 (OC2REF=0), 否则为有效电平 (OC2REF=1); 向下计数时, 若 $TIMx_CNT > TIMx_CC2RU$, 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 $TIMx_CNT < TIMx_CC2RU$, OC2REF 为无效电平 (OC2REF=0), 否则为有效电平 (OC2REF=1); 向下计数时, 若 $TIMx_CNT > TIMx_CC2RD$, OC2REF 为有效电平 (OC2REF=1), 否则为无效电平 (OC2REF=0)。

参数名	比特位	属性	复位值	描述
				(OC2REF=0)。
OC2CE	7	W/R	0	比较输出 OC2REF 清除控制寄存器： 0: OC2REF 不受 PWM _x _ETRF 输入的影响。 1: PWM _x _ETRF 输入有效电平时，设置 OC2REF 为无效电平。

TIM_x 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_x_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 TIM0_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。
IC2FS	7-4	W/R	0	通道 CC2 输入信号 CH2IN 滤波时钟频率和滤波周期数目 (N) 选择： 0000: 不滤波 0001: 滤波时钟频率为 f_{pwm_clk} ，N=2

参数名	比特位	属性	复位值	描述
				0010: 滤波时钟频率为 f_{pwm_clk} , N=4 0011: 滤波时钟频率为 f_{pwm_clk} , N=8 0100: 滤波时钟频率为 $f_{dts_clk}/2$, N=6 0101: 滤波时钟频率为 $f_{dts_clk}/2$, N=8 0110: 滤波时钟频率为 $f_{dts_clk}/4$, N=6 0111: 滤波时钟频率为 $f_{dts_clk}/4$, N=8 1000: 滤波时钟频率为 $f_{dts_clk}/8$, N=6 1001: 滤波时钟频率为 $f_{dts_clk}/8$, N=8 1010: 滤波时钟频率为 $f_{dts_clk}/16$, N=5 1011: 滤波时钟频率为 $f_{dts_clk}/16$, N=6 1100: 滤波时钟频率为 $f_{dts_clk}/16$, N=8 1101: 滤波时钟频率为 $f_{dts_clk}/32$, N=5 1110: 滤波时钟频率为 $f_{dts_clk}/32$, N=6 1111: 滤波时钟频率为 $f_{dts_clk}/32$, N=8

12.6.12 TIMx 通道 CC3 输入捕获/输出比较模式控制寄存器 (TIMx_CC3MR)

[TIM0:0x93/TIM1:0xD3]

Table 211 TIMx 通道 CC3 输入捕获/输出比较模式控制寄存器 (输出比较模式)

(TIMx_CC3MR)

参数名	比特位	属性	复位值	描述
CC3MS	1-0	W/R	00	捕获/比较功能选择寄存器: 00: CC3 通道被配置为输出比较功能; 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。 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。 11: CC3 通道被配置为输入捕获功能, 输入捕获信号 CH3INC 来自 TRC。 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。
OC3FE	2	W/R	0	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:

参数名	比特位	属性	复位值	描述
				<p>0: 根据计数器与 TIMx_CC3RU 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。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	<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 TIMx_CC3RU 和 TIMx_CC3RD 的预装载功能, 可随时将数据写入寄存器 TIMx_CC3RU 或 TIMx_CC3RD, 且新值立即生效。</p> <p>1: 开启寄存器 TIMx_CC3RU 和 TIMx_CC3RD 的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CC3RU 和 TIMx_CC3RD 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。</p> <p>单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC3MS	6-4	W/R	000	<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 TIMx_CC3RU 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>

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 TIMx_CC3RU 与计数器的值相同时，设置 OC3REF 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_CC3RU 与计数器的值相同时，设置 OC3REF 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_CC3RU 与计数器的值相同时，OC3REF 信号翻转。 100: 强制输出无效电平。 <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。 101: 强制输出有效电平。 <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。 110: PWM 模式 1 <ul style="list-style-type: none"> 边沿计数模式下：向上计数时，若 $TIMx_CNT < TIMx_CC3RU$，OC3REF 为有效电平（OC3REF=1），否则为无效电平（OC3REF=0）；向下计数时，若 $TIMx_CNT > TIMx_CC3RU$，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 $TIMx_CNT < TIMx_CC3RU$，OC3REF 为有效电平（OC3REF=1），否则为无效电平（OC3REF=0）；向下计数时，若 $TIMx_CNT > TIMx_CC3RD$，OC2REF 为无效电平（OC3REF=0），否则为有效电平（OC3REF=1）。 111: PWM 模式 2

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> ● 边沿计数模式下：向上计数时，若 $TIMx_CNT < TIMx_CC3RU$，OC3REF 为无效电平（OC3REF=0），否则为有效电平（OC3REF=1）；向下计数时，若 $TIMx_CNT > TIMx_CC3RU$，OC3REF 为有效电平（OC3REF=1），否则为无效电平（OC3REF=0）。 ● 中央对齐计数模式下：向上计数时，若 $TIMx_CNT < TIMx_CC3RU$，OC3REF 为无效电平（OC3REF=0），否则为有效电平（OC3REF=1）；向下计数时，若 $TIMx_CNT > TIMx_CC3RD$，OC3REF 为有效电平（OC3REF=1），否则为无效电平（OC3REF=0）。
OC3CE	7	W/R	0	比较输出 OC3REF 清除控制寄存器： 0: OC3REF 不受 PWMx_ETRF 输入的影响。 1: PWMx_ETRF 输入有效电平时，设置 OC3REF 为无效电平。

TIMx 通道 CC3 输入捕获/输出比较模式控制寄存器（输入捕获模式）（TIMx_CC3MR）

参数名	比特位	属性	复位值	描述
CC3MS	1-0	W/R	00	捕获/比较功能选择寄存器： 00: CC3 通道被配置为输出比较功能； 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。 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。 11: CC3 通道被配置为输入捕获功能，输入捕获信号 CH3INC 来自 TRC。 注：CC3MS 仅在通道关闭（寄存器 TIM0_CCENR 的 CC3E=0）时才是可写

参数名	比特位	属性	复位值	描述
				的，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。
IC3FS	7-4	W/R	0	通道 CC3 输入信号 CH3IN 滤波时钟频率和滤波周期数目 (N) 选择： 0000：不滤波 0001：滤波时钟频率为 f_{pwm_clk} ，N=2 0010：滤波时钟频率为 f_{pwm_clk} ，N=4 0011：滤波时钟频率为 f_{pwm_clk} ，N=8 0100：滤波时钟频率为 $f_{dts_clk}/2$ ，N=6 0101：滤波时钟频率为 $f_{dts_clk}/2$ ，N=8 0110：滤波时钟频率为 $f_{dts_clk}/4$ ，N=6 0111：滤波时钟频率为 $f_{dts_clk}/4$ ，N=8 1000：滤波时钟频率为 $f_{dts_clk}/8$ ，N=6 1001：滤波时钟频率为 $f_{dts_clk}/8$ ，N=8 1010：滤波时钟频率为 $f_{dts_clk}/16$ ，N=5 1011：滤波时钟频率为 $f_{dts_clk}/16$ ，N=6 1100：滤波时钟频率为 $f_{dts_clk}/16$ ，N=8 1101：滤波时钟频率为 $f_{dts_clk}/32$ ，N=5 1110：滤波时钟频率为 $f_{dts_clk}/32$ ，N=6 1111：滤波时钟频率为 $f_{dts_clk}/32$ ，N=8

12.6.13 TIMx 通道使能控制寄存器 (TIMx_CCENR) [TIM0:0x94/TIM1:0xD4]

Table 212 TIMx 通道使能控制寄存器(TIMx_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：

参数名	比特位	属性	复位值	描述
				<p>0: 关闭</p> <ul style="list-style-type: none"> OC0 禁止输出, 输出引脚 PWM0_CH0 的信号受寄存器 MOE、IOSS、ROSS、OIS0、OIS0N、CC0NEN、CC0P 位的控制。 <p>1: 开启</p> <ul style="list-style-type: none"> OC0 信号输出到对应的输出引脚 PWMx_CH0, 其输出受寄存器 MOE、OSSI、OSSR、OIS0、OIS0N、CC0NEN、CC0P 的控制。 <p>CC0 通道配置为输入时:</p> <p>0: 捕获禁止。</p> <p>1: 捕获使能</p> <ul style="list-style-type: none"> 当捕获信号产生有效沿时, 将计数器的值捕获到 TIMx_CC0RU 中。
CC0NEN	1	W/R	0	<p>通道 CC0 互补路输出使能控制:</p> <p>0: 关闭</p> <ul style="list-style-type: none"> OC0N 禁止输出, 输出引脚 PWMx_CH0N 的信号受寄存器 MOE、IOSS、ROSS、OIS0、OIS0N、CC0EN、CC0NP 位的控制。 <p>1: 开启</p> <ul style="list-style-type: none"> OC0N 信号输出到对应的输出引脚 PWMx_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

参数名	比特位	属性	复位值	描述
				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

12.6.14 TIMx 通道极性控制寄存器 (TIMx_CCPS) [TIM0:0x95/TIM1:0xD5]

Table 213 TIMx 通道极性控制寄存器(TIMx_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 (PWMx_CH0) 的高电平为有效电平。 1: OC0 (PWMx_CH0) 的低电平为有效电平。 CC0 通道配置为输入时： 该位选择是 CH0IN 还是 CH0IN 的反相信号作为触发或捕获信号。 <ul style="list-style-type: none"> 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。 1: 反相: CH0IN 的下沿作为有效边沿, CH0IN 的低电平作为有效电平。 注：一旦 LOCK 级别设置为 2 或 3，该位不能被修改。
CC0NP	1	W/R	0	通道 CC0 互补输出极性： <ul style="list-style-type: none"> 0: OC0N (PWMx_CH0N) 的高电平为有效电平。 1: OC0N (PWMx_CH0N) 的低电平为有效电平。 注：一旦 LOCK 级别设置为 2 或 3 或 CC3MS =0 时，则该位不能被修改。
CC1P	2	W/R	0	通道 CC1 极性，参见 CC0P 描述。

参数名	比特位	属性	复位值	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

12.6.15 TIMx 死区时间寄存器 (TIMx_DTG) [TIM0:0x96/TIM1:0xD6]

Table 214 TIMx 死区时间寄存器(TIMx_DTG)

参数名	比特位	属性	复位值	描述
DTG	7-0	W/R	0	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: 若 $DTG[7:5] = "0xx"$: $DT = DTG[6:0] \times T_{dts_clk}$ 若 $DTG[7:5] = "10x"$: $DT = (64 + DTG[5:0]) \times 2 \times T_{dts_clk}$ 若 $DTG[7:5] = "110"$: $DT = (32 + DTG[4:0]) \times 8 \times T_{dts_clk}$ 若 $DTG[7:5] = "111"$: $DT = (32 + DTG[4:0]) \times 16 \times T_{dts_clk}$

12.6.16 TIMx 周期值自动装载寄存器的低 8 位 (TIMx_ARRL) [TIM0:0x97/TIM1:0xD7]

Table 215 TIMx 周期值自动装载寄存器的低 8 位 (TIMx_ARRL)

参数名	比特位	属性	复位值	描述
ARRL	7-0	W/R	0	周期值自动重载值的低 8 位。若寄存器 TIMx_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 TIMx_EGR 的 UG 位为 1 ● 从模式控制器产生的复位信号 当自动重载的值为 0 时, 计数器不工作。

12.6.17 TIMx 周期值自动装载寄存器的高 8 位 (TIMx_ARRH) [TIM0:0x99/TIM1: 0xD9]

Table 216 TIMx 周期值自动装载寄存器的高 8 位 (TIMx_ARRH)

参数名	比特位	属性	复位值	描述
ARRH	7-0	W/R	0	周期值自动重载值的高 8 位。若寄存器 TIMx_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 TIMx_EGR 的 UG 位为 1 ● 从模式控制器产生的复位信号 当自动重载的值为 0 时，计数器不工作。

12.6.18 TIMx 预分频寄存器的低 8 位 (TIMx_PSCL) [TIM0:0x9A/TIM1: 0xDA]

Table 217 TIMx 预分频寄存器的低 8 位 (TIMx_PSCL)

参数名	比特位	属性	复位值	描述
CNTPSCL	7-0	W/R	0	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： <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

12.6.19 TIMx 预分频寄存器的高 8 位 (TIMx_PSCH) [TIM0:0x9B/TIM1:0xDB]

Table 218 TIMx 预分频寄存器的高 8 位 (TIMx_PSCH)

参数名	比特位	属性	复位值	描述
CNTPSCH	7-0	W/R	0	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： <ul style="list-style-type: none"> ● 计数器上溢/下溢

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> ● 设置 UG 位为 1 ● 从模式控制器产生的复位信号

12.6.20TIMx 重复计数寄存器 (TIMx_RCR) [TIM0:0x9C/TIM1:0xDC]

Table 219 TIMx 重复计数寄存器(TIMx_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> <p>这意味着在 PWM 模式下,RCR +1 对应着：</p> <ul style="list-style-type: none"> ● 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。

12.6.21TIMx 通道 CC0 捕获/向上计数比较寄存器的低 8 位 (TIMx_CC0RUL) [TIM0:0x9D/ TIM1:0xDD]

Table 220 TIMx 通道 CC0 捕获/向上计数比较寄存器的低 8 位(TIMx_CC0RUL)

参数名	比特位	属性	复位值	描述
CC0RUL	7-0	W/R	0	<p>通道 CC0 捕获/向上计数比较寄存器的低 8 位：</p> <ul style="list-style-type: none"> ● 若 CC0 通道配置为输出： <ul style="list-style-type: none"> 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

参数名	比特位	属性	复位值	描述
				<p>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。</p> <p>1) 定时器 TIMx 工作在互补模式下: 若计数器为边沿计数模式,无论向上计数还是向下计数使都将使用 TIMx_CC0RU 作为通道 CC0 的比较寄存器(即计数器与 TIMx_CC0RU 的值比较);若计数器为中央对齐计数模式,向上计数器时计数器与 TIMx_CC0RU 的值比较,向下计数时计数器与 TIMx_CC0RD 的值比较。</p> <p>2) 定时器工作在独立模式下: TIMx_CC0RU 作为第一路 PWM 输出比较寄存器,无论计数器工作在边沿模式还是中央对齐模式下,计数器与 TIMx_CC0RU 的值比较。</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低 8 位。

12.6.22TIMx 通道 CC0 捕获/向上计数比较寄存器的高 8 位 (TIMx_CC0RUH) [TIM0:0x9E/TIM1:0xDE]

Table 221 TIMx 通道 CC0 捕获/向上计数比较寄存器的高 8 位(TIMx_CC0RUH)

参数名	比特位	属性	复位值	描述
CC0RUH	7-0	W/R	0	<p>通道 CC0 捕获/向上计数比较寄存器的高 8 位:</p> <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)。 <p>如果未选择预装载功能,其立即被装入捕获/比较影子寄存器中。否则,只有当更新事件发生时,此预装载值才被装</p>

参数名	比特位	属性	复位值	描述
				<p>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> <p>1) 定时器 TIMx 工作在互补模式下: 若计数器为边沿计数模式,无论向上计数还是向下计数使都将使用 TIMx_CC0RU 作为通道 CC0 的比较寄存器(即计数器与 TIMx_CC0RU 的值比较);若计数器为中央对齐计数模式,向上计数时计数器与 TIMx_CC0RU 的值比较,向下计数时计数器与 TIMx_CC0RD 的值比较。</p> <p>2) 定时器工作在独立模式下: TIMx_CC0RU 作为第一路 PWM 输出比较寄存器,无论计数器工作在边沿模式还是中央对齐模式下,计数器与 TIMx_CC0RU 的值比较。</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。

12.6.23 TIMx 通道 CC0 向下计数比较寄存器的低 8 位 (TIMx_CC0RDL) [TIM0:0x9F/ TIM1:0xDF]

Table 222 TIMx 通道 CC0 捕获/向上计数比较寄存器的低 8 位(TIMx_CC0RUL)

参数名	比特位	属性	复位值	描述
CC0RDL	7-0	W/R	0	<p>通道 CC0 向下计数比较寄存器预装载值的低 8 位,此寄存器在输入捕获功能下不会被使用。</p> <p>当 CC0 通道配置为输出时:</p> <ul style="list-style-type: none"> ● 若 TIMx_CMPLMTARY.CMPEN = 1 (PWM 工作在互补模式下),表示各通道成对输出 PWM 互补波形。当计数器为边沿计数模式时,不使用 TIMx_CC0RD 作为比较寄存器;当

参数名	比特位	属性	复位值	描述
				计数器为中央对齐计数模式时，计数器在向下计数时将 TIMx_CC0RD 的值与计数器比较。 ● 若 TIMx_CMPLMTARY.CMPEN = 0 (PWM 工作在独立模式下)，表示 PWM 各路独立输出，寄存器 TIMx_CC0RD 作为第二路 PWM 的比较寄存器。无论计数器工作在边沿计数模式下还是中央对齐模式下，都使用 TIMx_CC0RD 作为比较寄存器。

12.6.24TIMx 通道 CC0 向下计数比较寄存器的高 8 位 (TIMx_CC0RDH) [TIM0:0xA1/ TIM1:0xE1]

Table 223 TIMx 通道 CC0 向下计数比较寄存器的高 8 位(TIMx_CC0RDH)

参数名	比特位	属性	复位值	描述
CC0RDH	7-0	W/R	0	通道 CC0 向下计数比较寄存器预装载值的高 8 位，此寄存器在输入捕获功能下不会被使用。 当 CC0 通道配置为输出时： ● 若 TIMx_CMPLMTARY.CMPEN = 1 (PWM 工作在互补模式下)，表示各通道成对输出 PWM 互补波形。当计数器为边沿计数模式时，不使用 TIMx_CC0RD 作为比较寄存器；当计数器为中央对齐计数模式时，计数器在向下计数时将 TIMx_CC0RD 的值与计数器比较。 ● 若 TIMx_CMPLMTARY.CMPEN = 0 (PWM 工作在独立模式下)，表示 PWM 各路独立输出，寄存器 TIMx_CC0RD 作为第二路 PWM 的比较寄存器。无论计数器工作在边沿

参数名	比特位	属性	复位值	描述
				计数模式下还是中央对齐模式下,都使用 TIMx_CC0RD 作为比较寄存器。

12.6.25TIMx 通道 CC1 捕获/向上计数比较寄存器的低 8 位 (TIMx_CC1RUL) [TIM0:0xA2/TIM1:0xE2]

Table 224 TIMx 通道 CC1 捕获/向上计数比较寄存器的低 8 位(TIMx_CC1RUL)

参数名	比特位	属性	复位值	描述
CC1RUL	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: <ul style="list-style-type: none"> 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。 <p>1) 定时器 TIMx 工作在互补模式下: 若计数器为边沿计数模式,无论向上计数还是向下计数使都将使用 TIMx_CC1RU 作为通道 CC1 的比较寄存器 (即计数器与 TIMx_CC1RU 的值比较);若计数器为中央对齐计数模式,向上计数器时计数器与 TIMx_CC1RU 的值比较,向下计数时计数器与 TIMx_CC1RD 的值比较。</p> <p>2) 定时器工作在独立模式下: TIMx_CC1RU 作为第三路 PWM 输出比较寄存器,无论计数器工作在边沿模式还是中央对齐模式下,计数器与 TIMx_CC1RU 的值比较。</p> <ul style="list-style-type: none"> 若 CC1 通道配置为输入:

参数名	比特位	属性	复位值	描述
				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。

12.6.26TIMx 通道 CC1 捕获/向上计数比较寄存器的高 8 位 (TIMx_CC1RUH) [TIM0:0xA3/TIM1:0xE3]

Table 225 TIMx 通道 CC1 捕获/向上计数比较寄存器的高 8 位(TIMx_CC1RUH)

参数名	比特位	属性	复位值	描述
CC1RUH	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： <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。</p> <p>1) 定时器 TIMx 工作在互补模式下： 若计数器为边沿计数模式，无论向上计数还是向下计数都将使用 TIMx_CC1RU 作为通道 CC1 的比较寄存器（即计数器与 TIMx_CC1RU 的值比较）；若计数器为中央对齐计数模式，向上计数器时计数器与 TIMx_CC1RU 的值比较，向下计数时计数器与 TIMx_CC1RD 的值比较。</p> <p>2) 定时器工作在独立模式下： TIMx_CC1RU 作为第三路 PWM 输出比较寄存器，无论计数器工作在边沿模式还是中央对齐模式下，计数器与 TIMx_CC1RU 的值比较。</p> ● 若 CC1 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</p>

参数名	比特位	属性	复位值	描述
				捕获到的计数器值的低 8 位。

12.6.27TIMx 通道 CC1 向下计数比较寄存器的低 8 位 (TIMx_CC1RDL) [TIM0:0xA4/TIM1:0xE4]

Table 226 TIMx 通道 CC1 向下计数比较寄存器的低 8 位(TIMx_CC1RDL)

参数名	比特位	属性	复位值	描述
CC1RDL	7-0	W/R	0	<p>通道 CC1 向下计数比较寄存器预装载值的低 8 位, 此寄存器在输入捕获功能下不会被使用。</p> <p>当 CC1 通道配置为输出时:</p> <ul style="list-style-type: none"> ● 若 TIMx_CMPLMTARY.CMPEN = 1 (PWM 工作在互补模式下), 表示各通道成对输出 PWM 互补波形。当计数器为边沿计数模式时, 不使用 TIMx_CC1RD 作为比较寄存器; 当计数器为中央对齐计数模式时, 计数器在向下计数时将 TIMx_CC1RD 的值与计数器比较。 ● 若 TIMx_CMPLMTARY.CMPEN = 0 (PWM 工作在独立模式下), 表示 PWM 各路独立输出, 寄存器 TIMx_CC1RD 作为第四路 PWM 的比较寄存器。无论计数器工作在边沿计数模式下还是中央对齐模式下, 都使用 TIMx_CC1RD 作为比较寄存器。

12.6.28TIMx 通道 CC1 向下计数比较寄存器的高 8 位 (TIMx_CC1RDH) [TIM0:0xA5/TIM1:0xE5]

Table 227 TIMx 通道 CC1 向下计数比较寄存器的高 8 位(TIMx_CC1RDH)

参数名	比特位	属性	复位值	描述
CC1RDH	7-0	W/R	0	通道 CC1 向下计数比较寄存器预装载值的高 8 位, 此寄存器在输入捕获功能下不

参数名	比特位	属性	复位值	描述
				会被使用。 当 CC1 通道配置为输出时： <ul style="list-style-type: none"> ● 若 TIMx_CMPLMTARY.CMPEN = 1 (PWM 工作在互补模式下)，表示各通道成对输出 PWM 互补波形。当计数器为边沿计数模式时，不使用 TIMx_CC1RD 作为比较寄存器；当计数器为中央对齐计数模式时，计数器在向下计数时将 TIMx_CC1RD 的值与计数器比较。 ● 若 TIMx_CMPLMTARY.CMPEN = 0 (PWM 工作在独立模式下)，表示 PWM 各路独立输出，寄存器 TIMx_CC1RD 作为第四路 PWM 的比较寄存器。无论计数器工作在边沿计数模式下还是中央对齐模式下，都使用 TIMx_CC1RD 作为比较寄存器。

12.6.29 TIMx 通道 CC2 捕获/向上计数比较寄存器的低 8 位 (TIMx_CC2RUL)

[TIM0:0xA6/TIM1:0xE6]

Table 228 TIMx 通道 CC2 捕获/向上计数比较寄存器的低 8 位(TIMx_CC2RUL)

参数名	比特位	属性	复位值	描述
CC2RUL	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： <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</p>

参数名	比特位	属性	复位值	描述
				<p>OC2 端口上输出信号。</p> <p>1) 定时器 TIMx 工作在互补模式下： 若计数器为边沿计数模式，无论向上计数还是向下计数使都将使用 TIMx_CC2RU 作为通道 CC2 的比较寄存器（即计数器与 TIMx_CC2RU 的值比较）；若计数器为中央对齐计数模式，向上计数器时计数器与 TIMx_CC2RU 的值比较，向下计数时计数器与 TIMx_CC2RD 的值比较。</p> <p>2) 定时器工作在独立模式下： TIMx_CC2RU 作为第五路 PWM 输出比较寄存器，无论计数器工作在边沿模式还是中央对齐模式下，计数器与 TIMx_CC2RU 的值比较。</p> <ul style="list-style-type: none"> ● 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。

12.6.30TIMx 通道 CC2 捕获/向上计数比较寄存器的高 8 位 (TIMx_CC2RUH) [TIM0:0xA7/TIM1:0xE7]

Table 229 TIMx 通道 CC2 捕获/向上计数比较寄存器的高 8 位(TIMx_CC2RUH)

参数名	比特位	属性	复位值	描述
CC2RUH	7-0	W/R	0	<p>通道 CC2 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 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）。 <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>

参数名	比特位	属性	复位值	描述
				<p>1) 定时器 TIMx 工作在互补模式下： 若计数器为边沿计数模式，无论向上计数还是向下计数使都将使用 TIMx_CC2RU 作为通道 CC2 的比较寄存器（即计数器与 TIMx_CC2RU 的值比较）；若计数器为中央对齐计数模式，向上计数器时计数器与 TIMx_CC2RU 的值比较，向下计数时计数器与 TIMx_CC2RD 的值比较。</p> <p>2) 定时器工作在独立模式下： TIMx_CC2RU 作为第五路 PWM 输出比较寄存器，无论计数器工作在边沿模式还是中央对齐模式下，计数器与 TIMx_CC2RU 的值比较。</p> <ul style="list-style-type: none"> ● 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。

12.6.31 TIMx 通道 CC2 向下计数比较寄存器的低 8 位 (TIMx_CC2RDL) [TIM0:0xA9/TIM0:0xE9]

Table 230 TIMx 通道 CC2 向下计数比较寄存器的低 8 位(TIMx_CC2RDL)

参数名	比特位	属性	复位值	描述
CC2RDL	7-0	W/R	0	<p>通道 CC2 向下计数比较寄存器预装载值的低 8 位，此寄存器在输入捕获功能下不会被使用。</p> <p>当 CC2 通道配置为输出时：</p> <ul style="list-style-type: none"> ● 若 TIMx_CMPPLMNTARY.CMPEN = 1（PWM 工作在互补模式下），表示各通道成对输出 PWM 互补波形。当计数器为边沿计数模式时，不使用 TIMx_CC2RD 作为比较寄存器；当计数器为中央对齐计数模式时，计数器在向下计数时将 TIMx_CC2RD 的值与计数器比较。

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> 若 TIMx_CMPLMTARY.CMPEN = 0 (PWM 工作在独立模式下), 表示 PWM 各路独立输出, 寄存器 TIMx_CC2RD 作为第六路 PWM 的比较寄存器。无论计数器工作在边沿计数模式下还是中央对齐模式下, 都使用 TIMx_CC2RD 作为比较寄存器。

12.6.32 TIMx 通道 CC2 向下计数比较寄存器的高 8 位 (TIMx_CC2RDH) [TIM0:0xAA/TIM1:0xEA]

Table 231 TIMx 通道 CC2 向下计数比较寄存器的高 8 位(TIMx_CC2RDH)

参数名	比特位	属性	复位值	描述
CC2RDH	7-0	W/R	0	<p>通道 CC2 向下计数比较寄存器预装载值的高 8 位, 此寄存器在输入捕获功能下不会被使用。</p> <p>当 CC2 通道配置为输出时:</p> <ul style="list-style-type: none"> 若 TIMx_CMPLMTARY.CMPEN = 1 (PWM 工作在互补模式下), 表示各通道成对输出 PWM 互补波形。当计数器为边沿计数模式时, 不使用 TIMx_CC2RD 作为比较寄存器; 当计数器为中央对齐计数模式时, 计数器在向下计数时将 TIMx_CC2RD 的值与计数器比较。 若 TIMx_CMPLMTARY.CMPEN = 0 (PWM 工作在独立模式下), 表示 PWM 各路独立输出, 寄存器 TIMx_CC2RD 作为第六路 PWM 的比较寄存器。无论计数器工作在边沿计数模式下还是中央对齐模式下, 都使用 TIMx_CC2RD 作为比较寄存器。

12.6.33 TIMx 通道 CC3 捕获/向上计数比较寄存器的低 8 位 (TIMx_CC3RUL) [TIM0:0xAB/TIM1:0xEB]

Table 232 TIMx 通道 CC3 捕获/向上计数比较寄存器的低 8 位(TIMx_CC3RUL)

参数名	比特位	属性	复位值	描述
CC3RUL	7-0	W/R	0	<p>通道 CC3 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 CC3 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 OC3 端口上输出信号。 <p>1) 定时器 TIMx 工作在互补模式下: 若计数器为边沿计数模式, 无论向上计数还是向下计数使都将使用 TIMx_CC3RU 作为通道 CC3 的比较寄存器 (即计数器与 TIMx_CC3RU 的值比较); 若计数器为中央对齐计数模式, 向上计数器时计数器与 TIMx_CC3RU 的值比较, 向下计数时计数器与 TIMx_CC3RD 的值比较。</p> <p>2) 定时器工作在独立模式下: TIMx_CC3RU 作为第七路 PWM 输出比较寄存器。若计数器为边沿计数模式, 无论向上计数还是向下计数使都将使用 TIMx_CC3RU 作为第七路 PWM 的比较寄存器 (即计数器与 TIMx_CC3RU 的值比较); 若计数器为中央对齐计数模式, 向上计数器时计数器与 TIMx_CC3RU 的值比较, 向下计数时计数器与 TIMx_CC3RD 的值比较。</p>

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> 若 CC3 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。

12.6.34TIMx 通道 CC3 捕获/向上计数比较寄存器的高 8 位 (TIMx_CC3RUH) [TIM0:0xAC/TIM1:0xEC]

Table 233 TIMx 通道 CC3 捕获/向上计数比较寄存器的高 8 位(TIMx_CC3RUH)

参数名	比特位	属性	复位值	描述
CC3RUH	7-0	W/R	0	<p>通道 CC3 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 CC3 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 OC3 端口上输出信号。 <p>1) 定时器 TIMx 工作在互补模式下： 若计数器为边沿计数模式，无论向上计数还是向下计数使都将使用 TIMx_CC3RU 作为通道 CC3 的比较寄存器（即计数器与 TIMx_CC3RU 的值比较）；若计数器为中央对齐计数模式，向上计数器时计数器与 TIMx_CC3RU 的值比较，向下计数时计数器与 TIMx_CC3RD 的值比较。</p> <p>2) 定时器工作在独立模式下： TIMx_CC3RU 作为第七路 PWM 输出比较寄存器。若计数器为边沿计数模式，无论向上计数还是向下计数使都将使用 TIMx_CC3RU 作为第七路 PWM 的比较寄存器（即计数器与 TIMx_CC3RU 的值</p>

参数名	比特位	属性	复位值	描述
				比较); 若计数器为中央对齐计数模式, 向上计数器时计数器与 TIMx_CC3RU 的值比较, 向下计数时计数器与 TIMx_CC3RD 的值比较。 ● 若 CC3 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。

12.6.35 TIMx 通道 CC3 向下计数比较寄存器的低 8 位 (TIMx_CC3RDL) [TIM0:0xAD/TIM1:0xED]

Table 234 TIMx 通道 CC3 向下计数比较寄存器的低 8 位(TIMx_CC3RDL)

参数名	比特位	属性	复位值	描述
CC3RDL	7-0	W/R	0	通道 CC3 向下计数比较寄存器预装载值的低 8 位, 此寄存器在输入捕获功能下不会被使用。 当 CC3 通道配置为输出时: ● 若 TIMx_CMPLMTARY.CMPEN = 1 (PWM 工作在互补模式下), 表示各通道成对输出 PWM 互补波形。当计数器为边沿计数模式时, 不使用 TIMx_CC3RD 作为比较寄存器; 当计数器为中央对齐计数模式时, 计数器在向下计数时将 TIMx_CC3RD 的值与计数器比较。 ● 若 TIMx_CMPLMTARY.CMPEN = 0 (PWM 工作在独立模式下), 表示 PWM 各路独立输出, 寄存器 TIMx_CC3RD 作为第七路 PWM 的比较寄存器。若计数器为边沿计数模式, 无论向上计数还是向下计数使都将使用 TIMx_CC3RU 作为第七路 PWM 的比较寄存器 (即计数器与 TIMx_CC3RU 的值比较); 若计数器

参数名	比特位	属性	复位值	描述
				为中央对齐计数模式, 向上计数器时计数器与 TIMx_CC3RU 的值比较, 向下计数时计数器与 TIMx_CC3RD 的值比较。

12.6.36 TIMx 通道 CC3 向下计数比较寄存器的高 8 位 (TIMx_CC3RDH) [TIM0:0xAE/TIM1:0xEE]

Table 235 TIMx 通道 CC3 向下计数比较寄存器的高 8 位(TIMx_CC3RDH)

参数名	比特位	属性	复位值	描述
CC3RDH	7-0	W/R	0	<p>通道 CC3 向下计数比较寄存器预装载值的高 8 位, 此寄存器在输入捕获功能下不会被使用。</p> <p>当 CC3 通道配置为输出时:</p> <ul style="list-style-type: none"> ● 若 TIMx_CMPLMTARY.CMPEN = 1 (PWM 工作在互补模式下), 表示各通道成对输出 PWM 互补波形。当计数器为边沿计数模式时, 不使用 TIMx_CC3RD 作为比较寄存器; 当计数器为中央对齐计数模式时, 计数器在向下计数时将 TIMx_CC3RD 的值与计数器比较。 ● 若 TIMx_CMPLMTARY.CMPEN = 0 (PWM 工作在独立模式下), 表示 PWM 各路独立输出, 寄存器 TIMx_CC3RD 作为第七路 PWM 的比较寄存器。若计数器为边沿计数模式, 无论向上计数还是向下计数使都将使用 TIMx_CC3RU 作为第七路 PWM 的比较寄存器 (即计数器与 TIMx_CC3RU 的值比较); 若计数器为中央对齐计数模式, 向上计数器时计数器与 TIMx_CC3RU 的值比较, 向下计数时计数器与 TIMx_CC3RD

参数名	比特位	属性	复位值	描述
				的值比较。

12.6.37TIMx 刹车控制寄存器 (TIMx_BRKC) [TIM0:0xAF/TIM1:0xEF]

Table 236 TIMx 刹车控制寄存器(TIMx_BRKC)

参数名	比特位	属性	复位值	描述
LOCK	1-0	W/R	0	<p>寄存器锁定设置，该位为防止软件错误配置而提供写保护。</p> <p>00：锁定关闭，寄存器无写保护。</p> <p>01：锁定级别 1，不能写入 TIMx_DTG 寄存器、TIMx_BRKC 的 BRKE、BRKP、AOE 位、TIMx_CONR1 寄存器。</p> <p>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 TIMx_BRKC 的 ROSS 和 IOSS 位。</p> <p>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 TIMx_CCxMR 的 OCxMS 和 OCxPEN 位。</p> <p>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。</p>
IOSS	2	W/R	0	<p>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> <p>0：当定时器 CCx 通道处于关闭状态时（CCxEN=0 或 CCxNEN=0），输出脚 PWMx_CHx 或 PWMx_CHxN 的输出使能 PWMx_CHxEN 和 PWMx_CHxNEN 无效，PWMx_CHx 或 PWMx_CHxN 输出高阻。详见 Table 195 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（CCxEN=0 或 CCxNEN=0），输出脚 PWMx_CHx 和 PWMx_CHxN 的输出使能</p>

参数名	比特位	属性	复位值	描述
				<p>PWMx_CHxEN 和 PWMx_CHxNEN 有效，PWMx_CHx 和 PWMx_CHxN 输出其空闲电平，详见 Table 195 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	3	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 PWMx_CHx 或 PWMx_CHxN 的输出使能 PWMx_CHxEN 和 PWMx_CHxNEN 无效，PWMx_CHx 或 PWMx_CHxN 输出高阻。详见 Table 195 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 PWMx_CHx 和 PWMx_CHxN 的输出使能 PWMx_CHxEN 和 PWMx_CHxNEN 有效，PWMx_CHx 和 PWMx_CHxN 输出其空闲电平，详见 Table 195 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>
BRKE	4	W/R	0	<p>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> <p>注 2：此位只控制外部输入的刹车信号，对</p>

参数名	比特位	属性	复位值	描述
				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。

12.6.38TIMx 事件产生寄存器 (TIMx_EGR) [TIM0:0xB1/TIM1:0xF1]

Table 237 TIMx 事件产生寄存器(TIMx_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：

参数名	比特位	属性	复位值	描述
				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 TIMx_CCENR、TIMx_CCPS 和 TIMx_CCxMR.OCxMS 的更新事件: 0: 无动作; 1: 当 CCxPE=1 时, 更新 TIMx_CCENR、TIMx_CCPS 和 TIMx_CCxMR.OCxMS。
CCG	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 CCx 上产生一个捕获/比较事件 若通道 CCx 配置位输出: 设置 CCxIF=1, 若打开比较中断使能, 则产生比较中断。 若通道 CCx 配置为输入: 当前的计数器值捕获至 TIMx_CCxRU 寄存器, 捕获中断标志位 CCxIF 自动置 1, 若打开捕获中断使能, 则产生捕获中断, 若 CCxIF 已经为 1, 则过捕获标志位 CCxOIF 自动置 1。
Reserved	6-4	R	0	保留
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 TIMx_ARR 的值; ● 初始化预分频计数器为 0。

12.6.39 TIMx 互补模式控制寄存器 (TIMx_CMPLMTARY)

[TIM0:0xB2/TIM1:0xF2]

Table 238 TIMx 互补模式控制寄存器(TIMx_CMPLMTARY)

参数名	比特位	属性	复位值	描述
CMPEN	0	W/R	1	互补模式控制： 0: TIMx 各个通道的 PWM 输出是独立的，也即 OCx 和 OCxN 独立输出，只有 PWM 模式 1 和 PWM 模式 2 下可以配置为独立模式。 1: 各通道成对互补输出。
BRKZEN	1	W	0	刹车时输出高阻使能寄存器： 0: 刹车时不输出高阻； 1: 刹车时输出高阻。
Reserved	7-2	R	0	保留

12.6.40 TIMx 中断标志寄存器 0 (TIMx_ISR0) [TIM0:0xB3/TIM1:0xF3]

Table 239 TIMx 中断标志寄存器 0(TIMx_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； 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 TIMx_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

参数名	比特位	属性	复位值	描述
				志： <ul style="list-style-type: none"> ● 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_CC0RU 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

12.6.41 TIMx 中断标志寄存器 1 (TIMx_ISR1) [TIM0:0xB4/TIM1:0xF4]

Table 240 TIMx 中断标志寄存器 1(TIMx_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，该标志由硬件置 1，由软件清 0:

参数名	比特位	属性	复位值	描述
				0: 没有上溢事件发生; 1: 发生上溢事件。
DOVERIF	5	W/R	0	计数器下溢更新中断标志, 该标志由硬件置 1, 由软件清 0: 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 TIMx_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

12.6.42 TIMx 中断标志寄存器 2 (TIMx_ISR2) [TIM0:0xB9/TIM1:0xF9]

Table 241 TIMx 中断标志寄存器 2(TIMx_ISR2)

参数名	比特位	属性	复位值	描述
CC0IF_NOCMP	0	W/R	0	独立模式下, 第一路 PWM 的比较中断标志, 独立模式下只有通道 CC0 配置为输出比较模式时该位才会起作用, 当计数器与比较值匹配时该位由硬件置 1, 由软件清 0。 0: 无匹配发生; 1: 计数器与比较值匹配。
CC1IF_NOCMP	1	W/R	0	独立模式下, 第二路 PWM 的比较中断标志, 参见 CC0IF_NOCMP 描述。
CC2IF_NOCMP	2	W/R	0	独立模式下, 第三路 PWM 的比较中断标志, 参见 CC0IF_NOCMP 描述。
CC3IF_NOCMP	3	W/R	0	独立模式下, 第四路 PWM 的比较中断标志, 参见 CC0IF_NOCMP 描述。
CC4IF_NOCMP	4	W/R	0	独立模式下, 第五路 PWM 的比较中断标志, 参见 CC0IF_NOCMP 描述。
CC5IF_NOCMP	5	W/R	0	独立模式下, 第六路 PWM 的比较中断标志, 参见 CC0IF_NOCMP 描述。

参数名	比特位	属性	复位值	描述
CC6IF_NOCMP	6	W/R	0	独立模式下，第七路 PWM 的比较中断标志，参见 CC0IF_NOCMP 描述。
Reserved	7	R	0	保留

12.6.43TIMx 计数器的低 8 位 (TIMx_CNTL) [TIM0:0xB5/TIM1:0xF5]

Table 242 TIMx 计数器的低 8 位 (TIMx_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。

12.6.44TIMx 计数器的高 8 位 (TIMx_CNTH) [TIM0:0xB6/TIM1:0xF6]

Table 243 TIMx 计数器的高 8 位(TIMx_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。

13 CORDIC 协处理器

13.1 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。

JMT1808R 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 $P \cdot |a|$ 。

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, JMT1808R CORDIC 协处理器的运算达到了相当高的精度。

JMT1808R CORDIC 协处理器有两种调度模式: JMT51 MCU 调度、JMT018 DSP 指令调度。

13.2 功能描述

13.2.1 输入数据格式

JMT1808R 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)⁶或者 Z0_DSP⁷，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 $[-2^{15}, (2^{15}-1)]$ 。

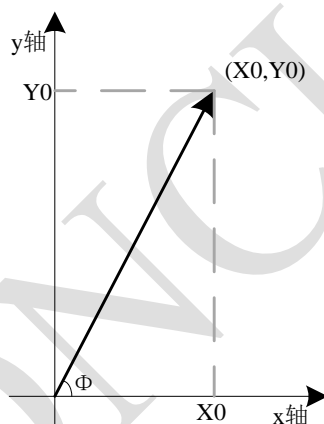


Figure 121 向量 (X_0, Y_0)

13.2.2 结果数据归一化

JMT1808R 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)。

⁶(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；

⁷ 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。

JMT1808R 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)。

13.2.3 CORDIC 调度

JMT1808R 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。

13.2.4 CORDIC 中断

JMT1808R 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。

13.2.5 CORDIC 精度

JMT1808R 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：

$$X_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_{理论i} - X_{计算i})^2}, N = 32768$$

JMT1808R CORDIC 计算的正余弦和理论的正余弦 (SIN/COS) 误差的均方根 $X_{rms}=9.25e^{-5}$ 。

JMT1808R CORDIC 计算向量反正切 (\tan^{-1}) 误差的均方根 $X_{rms}=6.45e^{-5}$ 。

JMT1808R CORDIC 计算向量模 ($P*|a|$) 误差的均方根 $X_{rms}=0.3764$ 。

13.3 CORDIC 寄存器

JMT1808R 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：

13.3.1 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 3。

Table 244 JMT51 调度 CORDIC 寄存器 (SFRPAGE=0x03)

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
0xCB	CORD_STU	R	0x00	CORDIC 状态寄存器

地址	寄存器名	属性	复位值	功能描述
0xCC	CORD_INT	W/R	0x00	CORDIC 中断寄存器

13.3.1.1 输入数据 X0 低 8 位(X0_LOW) [0xB9]

Table 245 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

13.3.1.2 输入数据 X0 高 8 位(X0_HIGH) [0xBA]

Table 246 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

13.3.1.3 输入数据 Y0 低 8 位(Y0_LOW) [0xBB]

Table 247 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

13.3.1.4 输入数据 Y0 高 8 位(Y0_HIGH) [0xBC]

Table 248 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

13.3.1.5 输入数据 Z0 低 8 位(Z0_LOW) [0xBD]

Table 249 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

13.3.1.6 输入数据 Z0 高 8 位(Z0_HIGH) [0xBE]

Table 250 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

13.3.1.7 运算结果 XN 低 8 位(XN_LOW) [0xC1]

Table 251 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

13.3.1.8 运算结果 XN 高 8 位(XN_HIGH) [0xC2]

Table 252 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

13.3.1.9 运算结果 YN 低 8 位(YN_LOW) [0xC3]

Table 253 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

13.3.1.10 运算结果 YN 高 8 位(YN_HIGH) [0xC4]

Table 254 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

13.3.1.11 运算结果 ZN 低 8 位(ZN_LOW) [0xC5]

Table 255 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

13.3.1.12 运算结果 ZN 高 8 位(ZN_HIGH) [0xC6]

Table 256 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

13.3.1.13 CORDIC 启动寄存器(CORD_STA) [0xC9]

Table 257 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

13.3.1.14 CORDIC 模式寄存器(CORD_MOD) [0xCA]

Table 258 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

13.3.1.15 CORDIC 状态寄存器(CORD_STU) [0xCB]

Table 259 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

13.3.1.16 CORDIC 中断寄存器(CORD_INT) [0xCC]

Table 260 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

13.3.2 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 261 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

13.3.2.1 输入数据 X0(X0_DSP) [0xC0]

Table 262 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

13.3.2.2 输入数据 Y0(Y0_DSP) [0xC1]

Table 263 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

13.3.2.3 输入数据 Z0(Z0_DSP) [0xC2]

Table 264 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

13.3.2.4 运算结果 XN(XN_DSP) [0xC3]

Table 265 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

13.3.2.5 运算结果 YN(YN_DSP) [0xC4]

Table 266 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

13.3.2.6 运算结果 ZN(ZN_DSP) [0xC5]

Table 267 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

13.3.2.7 CORDIC 模式寄存器(CORD_MOD_DSP) [0xC6]

Table 268 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

13.3.2.8 CORDIC 状态寄存器(CORD_STU_DSP) [0xC7]

Table 269 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

14 通用异步收发传输器 (UART)

14.1 概述

JMT1808R 有两个通用异步收发传输器 UART0 和 UART1, 下载时必须使用复用 UART0 功能的 PC4 和 PC5 引脚。每个都支持三种接口:

- 标准 UART 通信接口
- 38KHz 红外调制发送接口
- IrDA (SIR) 红外通信接口

14.2 框图

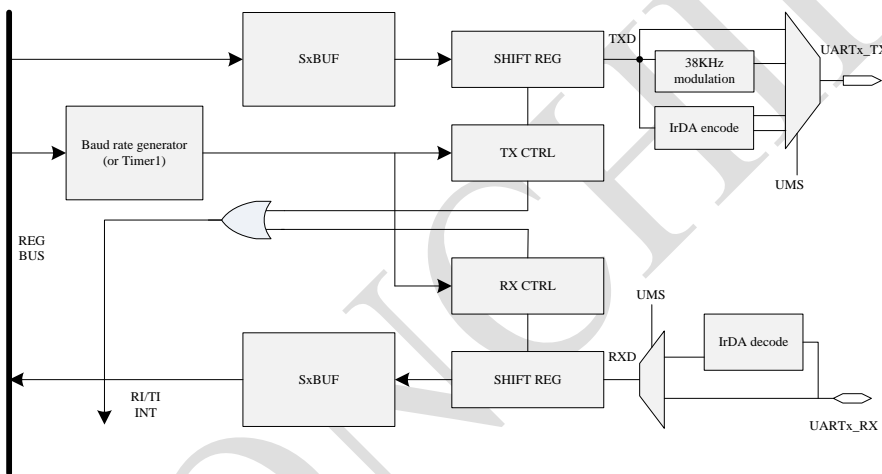


Figure 122 UART 结构框图

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 UARTx_TX 发送同步时钟, 通过 UARTx_RX 接收或发送数据, 波特率是系统时钟频率 f_{sys_clk} 的 1/12。
- 模式 1: 8 位 UART。通过 UARTx_TX 发送或通过 UARTx_RX 接收 8 个数据位, 波特率是可变的。
- 模式 2: 9 位 UART。通过 UARTx_TX 发送或通过 UARTx_RX 接收 9 个数据位, 波特率可编程为系统时钟频率 f_{sys_clk} 的 1/32 或 1/64。
- 模式 3: 9 位 UART。通过 UARTx_TX 发送或通过 UARTx_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, 具体参见 14.4.5。

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 14.4.6。

14.3 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。

14.4 UART 工作模式

14.4.1 标准 UART 工作模式 0：同步移位寄存器

配置 SxCON⁸的 SM0、SM1 为 00 时，UARTx 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 UARTx_TX 输出，数据由 UARTx_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 123 和 Figure 124 所示。图中 baud_clk 表示波特率示意时钟，buf_write 为软件写 SxBUF 动作，SxCON.TI 为发送中断标志，SxCON.RI 为接收中断标志，此后类同。

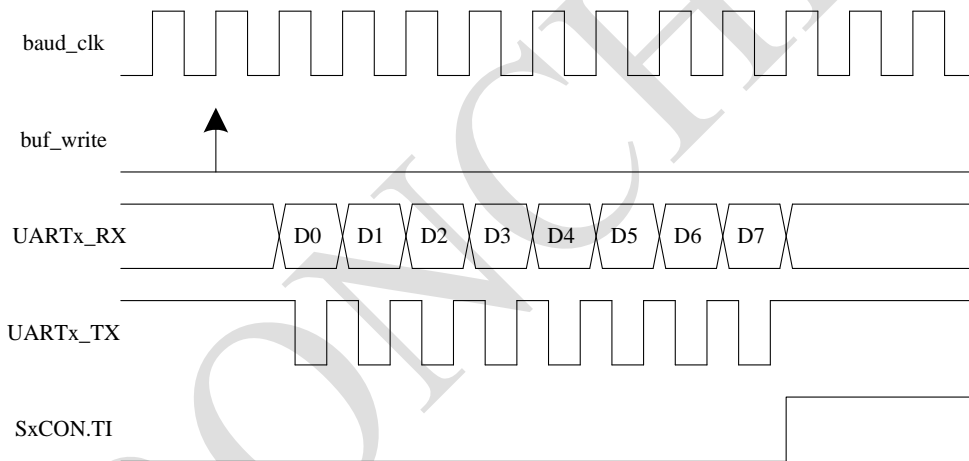


Figure 123 UART 模式 0 发送时序

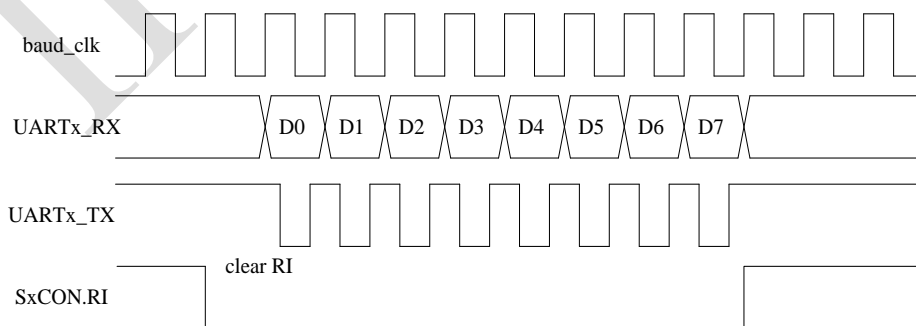


Figure 124 UART 模式 0 接收时序

模式 0 发送：当处理器执行将数据写入发送缓冲器 SxBUF 时启动发送，硬件将 8 位数据以 $f_{sys_clk}/12$ 的波特率从 UARTx_RX 管脚输出。一帧（8 位）数据发送完毕时，SxCON.TI

⁸ SxCON 中的 x=0 或 1，代表 UART0 或 UART1

置 1，申请中断。再次发送数据前，必须用软件将 SxCON.TI 清 0。

模式 0 接收：首先将接收中断请求标志 SxCON.RI 置 1，然后将接收中断请求标志 SxCON.RI 清 0，且将允许接收控制位 SxCON.REN 置 1，启动串行模式 0 接收过程。启动接收过程后，UARTx_RX 为串行输入端，UARTx_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 SxCON.RI 置 1，并发出中断申请。当再次接收时，必须通过软件将 SxCON.RI 清 0。

工作于模式 0 时，必须将多机通信控制位 SxCON.SM2 清 0，使得对 SxCON.TB8 位和 SxCON.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 MCU，请求中断，软件响应中断后必须判别是 SxCON.TI 中断请求还是 SxCON.RI 中断请求，并清除相应的标志位。

14.4.2 标准 UART 工作模式 1：8 位 UART，波特率可变

配置寄存器 SxCON 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。

UARTx_TX 为发送信息，UARTx_RX 为接收信息，以全双工模式进行接收/发送。接口时序如 Figure 125 和 Figure 126 所示。

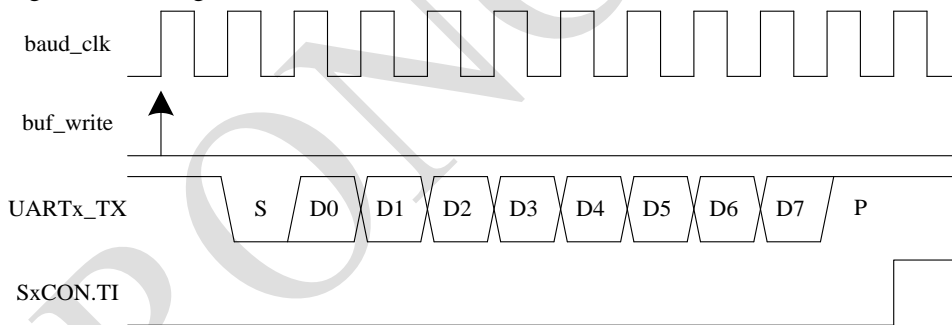


Figure 125 UART 模式 1 发送时序

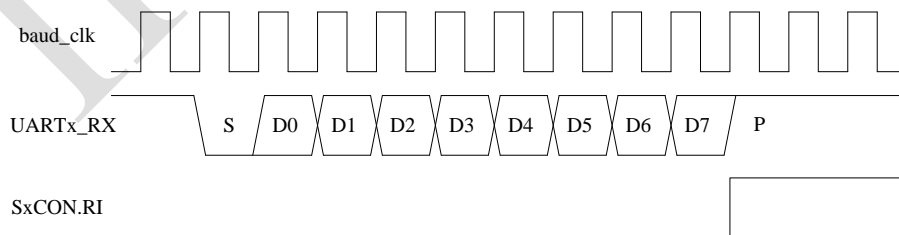


Figure 126 UART 模式 1 接收时序

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 14.7 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 SxCON.TI 置 1；当接收完一帧数据后，硬件自动将接收中断标志 SxCON.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 MCU，请求中断，软件响应中断后必须判别是 SxCON.TI 中断请求还是 SxCON.RI 中断请求，并清除相应的标志位。

14.4.3 标准 UART 工作模式 2：9 位 UART，波特率固定

配置寄存器 SxCON 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（SxCON.TB8/SxCON.RB8，第 9 位数据）和 1 位停止位。SxCON.TB8/SxCON.RB8 可以作为多机通信地址标志或奇偶校验位使用。UARTx_TX 为发送信息，UARTx_RX 为接收信息，以全双工模式进行接收/发送。接口时序如 Figure 127 和 Figure 128 所示。

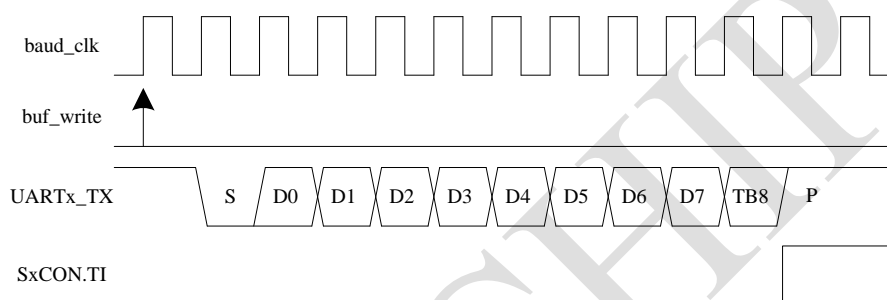


Figure 127 UART 模式 2 发送时序

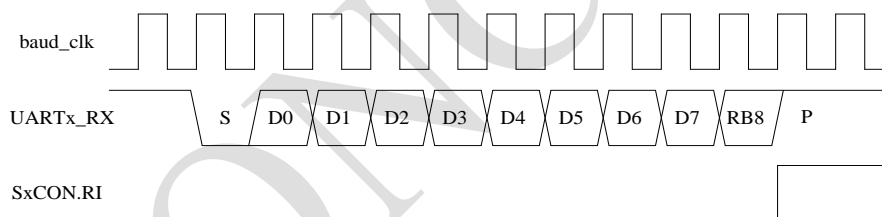


Figure 128 UART 模式 2 接收时序

配置寄存器 SxCON 中的 SM2、TB8 可实现多机通信和奇偶校验，具体见 14.5 和 14.6 节。

模式 2 发送中断和模式 2 接收中断合并后送至 MCU，请求中断，软件响应中断后必须判别是 SxCON.TI 中断请求还是 SxCON.RI 中断请求，并清除相应的标志位。

14.4.4 标准 UART 工作模式 3：9 位 UART，波特率可变

配置寄存器 SxCON 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（SxCON.TB8/SxCON.RB8，第 9 位数据）和 1 位停止位。SxCON.TB8/SxCON.RB8 可以作为多机通信地址标志或奇偶校验位使用。UARTx_TX 为发送信息，UARTx_RX 为接收信息，以全双工模式进行接收/发送。接口时序如 Figure 129 和 Figure 130 所示。

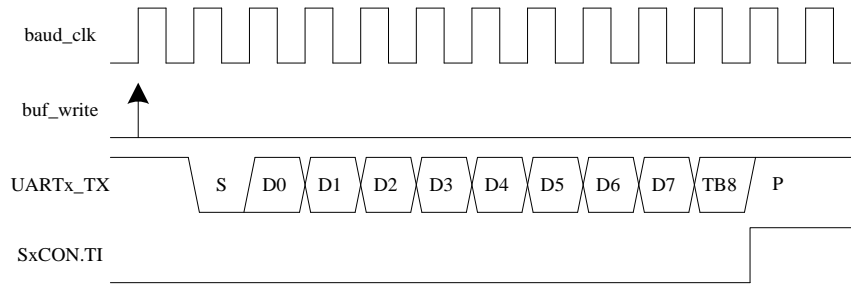


Figure 129 UART 模式 3 发送时序

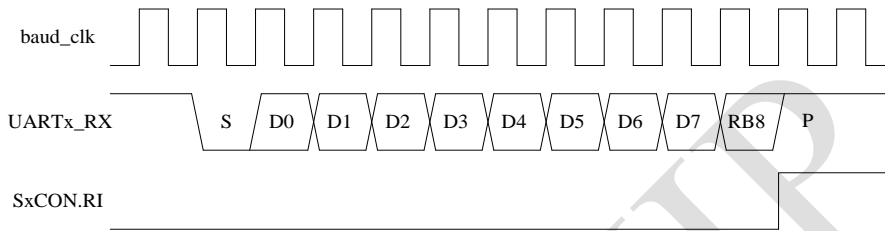


Figure 130 UART 模式 3 接收时序

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 14.7 节。

配置寄存器 SxCON 中的 SM2、TB8 可实现多机通信和奇偶校验,具体见 14.5 和 14.6 节。

模式 3 发送中断和模式 3 接收中断合并后送至 MCU,请求中断,软件响应中断后必须判别是 SxCON.TI 中断请求还是 SxCON.RI 中断请求,并清除相应的标志位。

14.4.5 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 SxIRCON.IRINV 控制寄存器来选择,具体见 Figure 131,IRINV=0 对应 NPN 型,IRINV=1 对应 PNP 型。

通过改变寄存器 IRPD (见寄存器 SxRELH 和 SxIRCON)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。

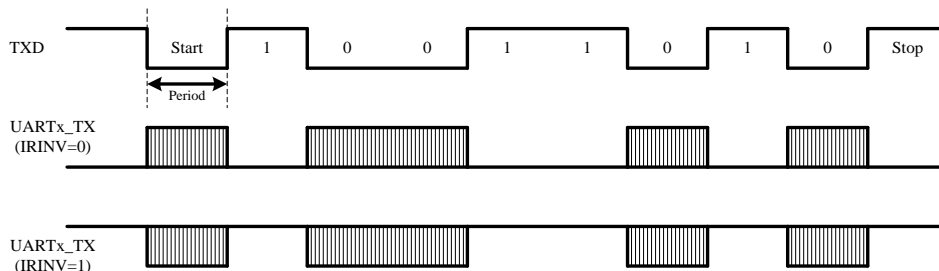


Figure 131 38KHz 红外调制发送时序

14.4.6 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 132。

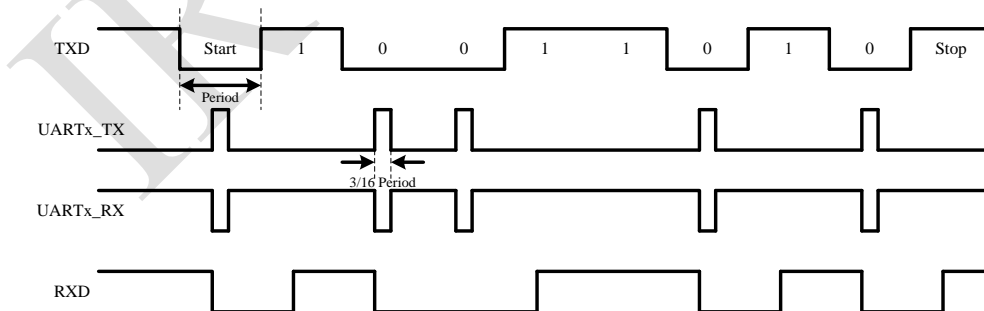


Figure 132 IrDA 接收和发送时序

14.5 UART 多机通信

UART 模式 2 和模式 3 通过使用第 9 数据位 (SxCON.TB8/SxCON.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 $SxCON.SM2$ 位被置 1，则只有当 UART 接收到的第 9 位为 1 ($SxCON.RB8 = 1$) 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 $SxCON.SM2$ 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 $SxCON.SM2$ 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 $SxCON.SM2$ 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。

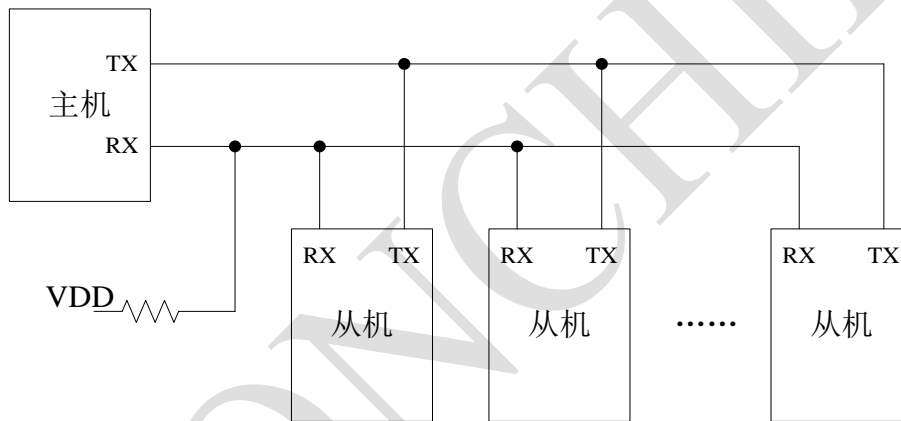


Figure 133 UART 多机通信连接方式

14.6 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 ($SxCON.TB8/SxCON.RB8$)，该位可以实现奇偶校验的功能。

配置寄存器 $SxCON.SM2$ 为 0，使得 UART 工作在非多机通信模式。

发送时： $SxCON$ 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 $SxCON.TB8$ 。

接收时：硬件自动把 UART 接收到的第 9 位数据写入 $SxCON$ 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 $SxCON$ 寄存器的 $RB8$ 进行比较，来判断奇偶校验是否正确。

14.7 UART 波特率计算

UART 波特率计算：

- UART 模式 0

$$baud\ rate = \frac{f_{sys_clk}}{12}$$

- UART 模式 1, 其中 UART0 的 UBSEL 对应 TCON.0; UART1 的 UBSEL 对应 TCON.1; UART0 的 SMOD 对应 TCON.2; UART1 的 SMOD 对应 TCON.3。

➤ 寄存器 UBSEL=0, 使用 Timer1 的模式 2 作为波特率产生器:

$$baud\ rate = \frac{2^{SMOD} * f_{sys_clk}}{32 * (256 - TH1)}$$

➤ 寄存器 UBSEL=1, 使用 UART 自带波特率产生器:

$$baud\ rate = \frac{2^{SMOD} * f_{sys_clk}}{32 * (1024 - SREL)}$$

Table 270 UART 模式 1 波特率 ($f_{sys_clk}=73.728M$)

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 271 UART 模式 1 波特率 ($f_{sys_clk}=36.864M$)

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 272 UART 模式 1 波特率 ($f_{sys_clk}=18.432M$)

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 273 UART 模式 1 波特率 ($f_{sys_clk}=9.216M$)

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 274 UART 模式 1 波特率 ($f_{sys_clk}=4.608M$)

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 275 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

Table 276 UART 模式 1 波特率 ($f_{sys_clk}=22.1184M$)

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	112	4800.0	0.0000	x	x	x
9600	184	9600.0	0.0000	112	9600.0	0.0000
19200	220	19200.0	0.0000	184	19200.0	0.0000
38400	238	38400.0	0.0000	220	38400.0	0.0000
51200	244	57600.0	2.8571	231	55296.0	-1.2571
57600	244	57600.0	0.0000	232	57600.0	0.0000
115200	250	115200.0	0.0000	244	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	448	1200.0	0.0000	x	x	x
2400	736	2400.0	0.0000	448	2400.0	0.0000
4800	880	4800.0	0.0000	736	4800.0	0.0000
9600	952	9600.0	0.0000	880	9600.0	0.0000
19200	988	19200.0	0.0000	952	19200.0	0.0000
38400	1006	38400.0	0.0000	988	38400.0	0.0000
51200	1012	57600.0	2.8571	999	55296.0	-1.2571
57600	1012	57600.0	0.0000	1000	57600.0	0.0000
115200	1018	115200.0	0.0000	1012	115200.0	0.0000

Table 277 UART 模式 1 波特率 ($f_{sys_clk}=11.0592M$)

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	112	2400.0	0.0000	x	x	x
4800	184	4800.0	0.0000	112	4800.0	0.0000
9600	220	9600.0	0.0000	184	9600.0	0.0000
19200	238	19200.0	0.0000	220	19200.0	0.0000
38400	247	38400.0	0.0000	238	38400.0	0.0000

56000	250	57600.0	2.8571	244	57600.0	2.8571
57600	250	57600.0	0.0000	244	57600.0	0.0000
115200	253	115200.0	0.0000	250	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	448	600.0	0.0000	x	x	x
1200	736	1200.0	0.0000	448	1200.0	0.0000
2400	880	2400.0	0.0000	736	2400.0	0.0000
4800	952	4800.0	0.0000	880	4800.0	0.0000
9600	988	9600.0	0.0000	952	9600.0	0.0000
19200	1006	19200.0	0.0000	988	19200.0	0.0000
38400	1015	38400.0	0.0000	1006	38400.0	0.0000
56000	1018	57600.0	2.8571	1012	57600.0	2.8571
57600	1018	57600.0	0.0000	1012	57600.0	0.0000
115200	1021	115200.0	0.0000	1018	115200.0	0.0000

 Table 278 UART 模式 1 波特率 ($f_{sys_clk}=5.5296M$)

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	112	1200.0	0.0000	x	x	x
2400	184	2400.0	0.0000	112	2400.0	0.0000
4800	220	4800.0	0.0000	184	4800.0	0.0000
9600	238	9600.0	0.0000	220	9600.0	0.0000
19200	247	19200.0	0.0000	238	19200.0	0.0000
38400	x	x	x	247	38400.0	0.0000
56000	253	57600.0	2.8571	250	57600.0	2.8571
57600	253	57600.0	0.0000	250	57600.0	0.0000
115200	x	x	x	253	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	448	300.0	0.0000	x	x	x
600	736	600.0	0.0000	448	600.0	0.0000
1200	880	1200.0	0.0000	736	1200.0	0.0000

2400	952	2400.0	0.0000	880	2400.0	0.0000
4800	988	4800.0	0.0000	952	4800.0	0.0000
9600	1006	9600.0	0.0000	988	9600.0	0.0000
19200	1015	19200.0	0.0000	1006	19200.0	0.0000
38400	x	x	x	1015	38400.0	0.0000
56000	1021	57600.0	2.8571	1018	57600.0	2.8571
57600	1021	57600.0	0.0000	1018	57600.0	0.0000
115200	x	x	x	1021	115200.0	0.0000

Table 279 UART 模式 1 波特率 ($f_{sys_clk}=2.7648M$)

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	112	600.0	0.0000	x	x	x
1200	184	1200.0	0.0000	112	1200.0	0.0000
2400	220	2400.0	0.0000	184	2400.0	0.0000
4800	238	4800.0	0.0000	220	4800.0	0.0000
9600	247	9600.0	0.0000	238	9600.0	0.0000
19200	x	x	x	247	19200.0	0.0000
56000	x	x	x	253	57600.0	2.8571
57600	x	x	x	253	57600.0	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	736	300.0	0.0000	448	300.0	0.0000
600	880	600.0	0.0000	736	600.0	0.0000
1200	952	1200.0	0.0000	880	1200.0	0.0000
2400	988	2400.0	0.0000	952	2400.0	0.0000
4800	1006	4800.0	0.0000	988	4800.0	0.0000
9600	1015	9600.0	0.0000	1006	9600.0	0.0000
19200	x	x	x	1015	19200.0	0.0000
56000	x	x	x	1021	57600.0	2.8571
57600	x	x	x	1021	57600.0	0.0000

Table 280 UART 模式 1 波特率 ($f_{sys_clk}=1.3824M$)

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	112	300.0	0.0000	x	x	x
600	184	600.0	0.0000	112	600.0	0.000
1200	220	1200.0	0.0000	184	1200.0	0.000
2400	238	2400.0	0.0000	220	2400.0	0.000
4800	247	4800.0	0.0000	238	4800.0	0.000
9600	x	x	x	247	9600.0	0.000
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	880	300.0	0.0000	736	300.0	0.0000
600	952	600.0	0.0000	880	600.0	0.0000
1200	988	1200.0	0.0000	952	1200.0	0.0000
2400	1006	2400.0	0.0000	988	2400.0	0.0000
4800	1015	4800.0	0.0000	1006	4800.0	0.0000
9600	x	x	x	1015	9600.0	0.0000

 Table 281 UART 模式 1 波特率 ($f_{sys_clk}=0.6912M$)

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	184	300.0	0.0000	112	300.0	0.0000
600	220	600.0	0.0000	184	600.0	0.0000
1200	238	1200.0	0.0000	220	1200.0	0.0000
2400	247	2400.0	0.0000	238	2400.0	0.0000
4800	x	x	x	247	4800.0	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	952	300.0	0.0000	880	300.0	0.0000
600	988	600.0	0.0000	952	600.0	0.0000
1200	1200.0	0.0000	0.0000	988	1200.0	0.0000
2400	1015	2400.0	0.0000	1006	2400.0	0.0000
4800	x	x	x	1015	4800.0	0.0000

- UART 模式 2, 其中 UART0 的 SMOD 对应 TCON.2; UART1 的 SMOD 对应 TCON.3。

$$\text{baud rate} = \frac{2^{\text{SMOD}} * f_{\text{sys_clk}}}{64}$$

- UART 模式 3

计算方法和模式 1 相同。

14.8 UART 寄存器

UART 模块有 10 个寄存器, 对应的 SFRPAGE 为 0, 具体说明如下。

Table 282 UART 寄存器 (SFRPAGE=0x00)

地址	寄存器名	属性	复位值	功能描述
0x98	S0CON	W/R	0x00	UART0 控制寄存器
0x99	S0BUF	W/R	0x00	UART0 数据缓冲寄存器
0x9A	S0RELL	W/R	0x00	UART0 波特率重载寄存器低 8 位
0x9B	S0RELH	W/R	0x00	UART0 波特率重载寄存器高 2 位
0x9C	S0IRCON	W/R	0x00	UART0 红外控制寄存器
0xF1	S1CON	W/R	0x00	UART1 控制寄存器
0xF2	S1BUF	W/R	0x00	UART1 数据缓冲寄存器
0xF3	S1RELL	W/R	0x00	UART1 波特率重载寄存器低 8 位
0xF4	S1RELH	W/R	0x00	UART1 波特率重载寄存器高 2 位
0xF5	S1IRCON	W/R	0x00	UART1 红外控制寄存器

14.8.1 UART0 控制寄存器(S0CON) [0x98]

Table 283 UART0 控制寄存器(S0CON)

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

参数名	比特位	属性	复位值	描述
				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 284
SM0	7	W/R	0	

Table 284 UART 工作方式选择

SM0	SM1	模式	描述	波特率
0	0	0	移位寄存器	$f_{sys_clk}/12$
0	1	1	8 位 UART	可变
1	0	2	9 位 UART	$f_{sys_clk}/32$ 或 $f_{sys_clk}/64$
1	1	3	9 位 UART	可变

14.8.2 UART0 数据缓冲寄存器(S0BUF) [0x99]

Table 285 UART0 数据缓冲寄存器(S0BUF)

参数名	比特位	属性	复位值	描述
SBUF	7-0	W/R	0	UART0 数据缓冲寄存器: 对寄存器 SBUF 写操作, 则 UART0 将开始向外传输发送缓存数据; 对寄存器 SBUF 读操作, 则 UART0 将从串行接收缓存中读取数据

14.8.3 UART0 波特率重载寄存器低 8 位(S0RELL) [0x9A]

Table 286 UART0 波特率重载寄存器低 8 位(S0RELL)

参数名	比特位	属性	复位值	描述
SRELL	7-0	W/R	0	UART0 波特率重载寄存器低 8 位: 通过配置该寄存器改变 UART0 的通讯波

参数名	比特位	属性	复位值	描述
				特率。

14.8.4 UART0 波特率重载寄存器高 2 位(S0RELH) [0x9B]

Table 287 UART0 波特率重载寄存器高 2 位(S0RELH)

参数名	比特位	属性	复位值	描述
SRELH	1-0	W/R	0	UART0 波特率重载寄存器高 2 位： 通过配置该寄存器改变 UART0 的通讯波特率。
IRPD[9:4]	7-2	W/R	0	UART0 红外调制频率控制位高 6 位

14.8.5 UART0 红外控制寄存器(S0IRCON) [0x9C]

Table 288 UART0 红外控制寄存器(S0IRCON)

参数名	比特位	属性	复位值	描述
IRPD[3:0]	3-0	W/R	0	UART0 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 $1.35MHz < f_{SLPBaud} < 2.12MHz$ ； 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
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	UART0 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

14.8.6 UART1 控制寄存器(S1CON) [0xF1]

Table 289 UART1 控制寄存器(S1CON)

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
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 由软件置位/复位
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 284
SM0	7	W/R	0	

14.8.7 UART1 数据缓冲寄存器(S1BUF) [0xF2]

Table 290 UART1 数据缓冲寄存器(S1BUF)

参数名	比特位	属性	复位值	描述
SBUF	7-0	W/R	0	UART1 数据缓冲寄存器： 对寄存器 SBUF 写操作，则 UART1 将开始向外传输发送缓存数据； 对寄存器 SBUF 读操作，则 UART1 将从串

参数名	比特位	属性	复位值	描述
				行接收缓存中读取数据

14.8.8 UART1 波特率重载寄存器低 8 位(S1RELL) [0xF3]

Table 291 UART1 波特率重载寄存器低 8 位(S1RELL)

参数名	比特位	属性	复位值	描述
SRELL	7-0	W/R	0	UART1 波特率重载寄存器低 8 位： 通过配置该寄存器改变 UART1 的通讯波特率。

14.8.9 UART1 波特率重载寄存器高 2 位(S1RELH) [0xF4]

Table 292 UART1 波特率重载寄存器高 2 位(S1RELH)

参数名	比特位	属性	复位值	描述
SRELH	1-0	W/R	0	UART1 波特率重载寄存器高 2 位： 通过配置该寄存器改变 UART1 的通讯波特率。
IRPD[9:4]	7-2	W/R	0	红外调制频率控制位高 6 位

14.8.10 UART1 红外控制寄存器(S1IRCON) [0xF5]

Table 293 UART1 红外控制寄存器(S1IRCON)

参数名	比特位	属性	复位值	描述
IRPD[3:0]	3-0	W/R	0	红外调制频率控制位低 4 位： 当工作在 low-powerIrDA 模式时，用作 SIR 的 low-power 模式时钟分频系数，选择分频系数使 $1.35MHz < f_{SLPBaud} < 2.12MHz$ ； 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
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：

参数名	比特位	属性	复位值	描述
				0: 正常功能模式 1: 回环测试模式
UMS	7-6	W/R	0	UART1 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

IRONCHIP

15 I2C 接口

15.1 概述

JMT1808R 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

15.2 框图

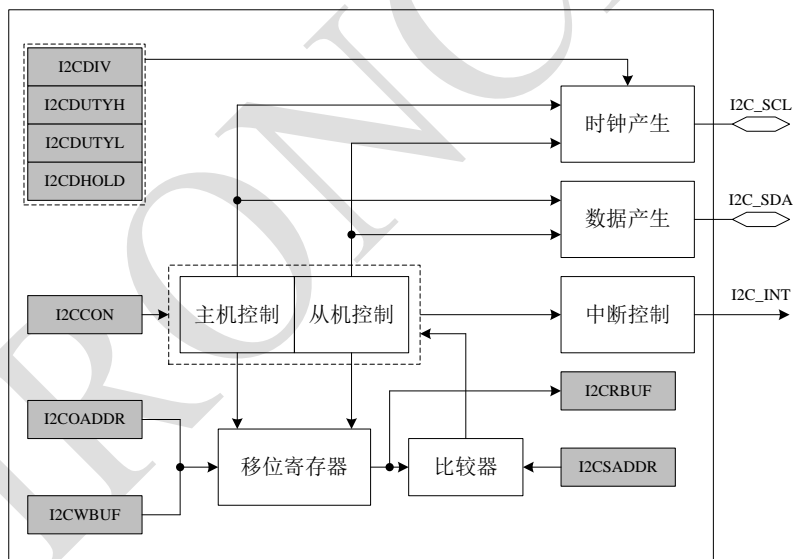


Figure 134 I2C 结构框图

15.3 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。

15.4 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。

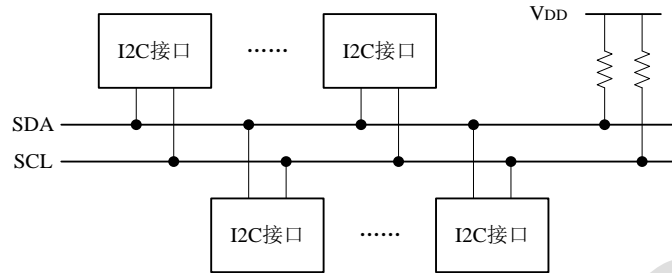


Figure 135 I2C 模块连接

15.5 I2C 数据格式

I2C 接口的时序图如下所示：

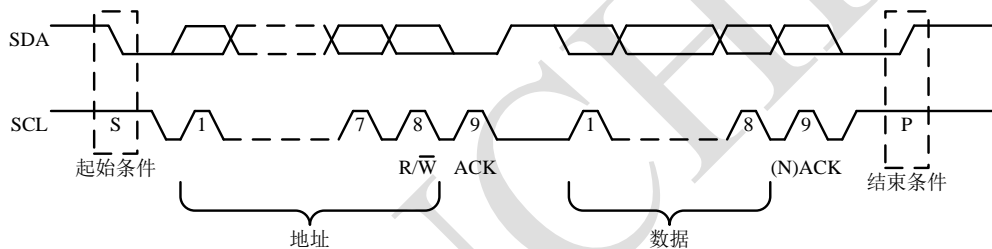


Figure 136 I2C 接口时序

15.5.1 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。

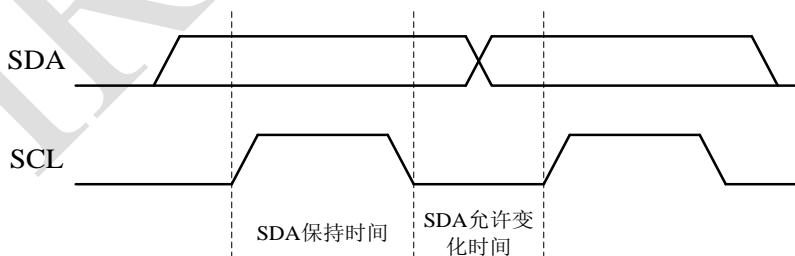


Figure 137 I2C 位传输

15.5.2 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/\bar{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。

15.5.3 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)。

15.5.4 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。

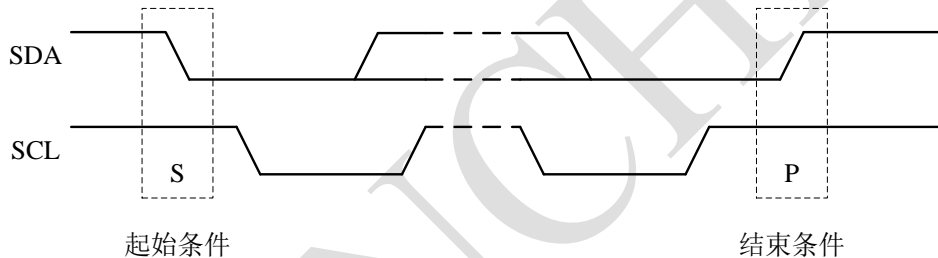


Figure 138 起始和结束条件

15.5.5 数据格式



Figure 139 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。

15.5.6 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 总线上的报文。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。

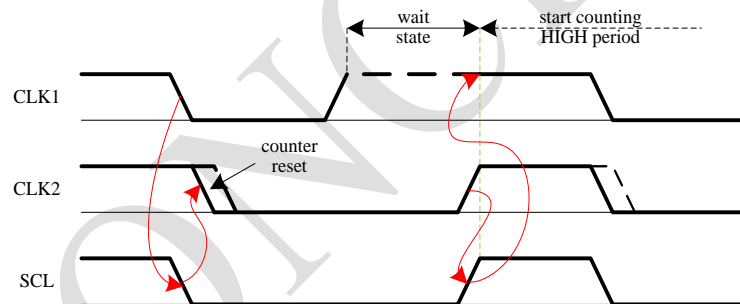


Figure 140 仲裁过程中的时钟同步

15.5.7 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 141 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。

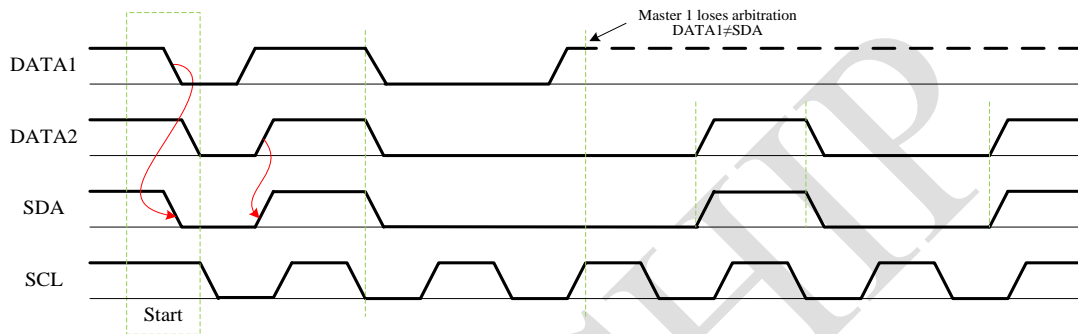


Figure 141 两个主机的仲裁过程

15.6 I2C 时钟产生

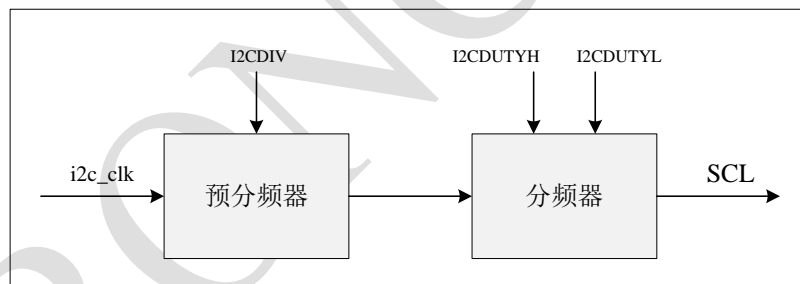


Figure 142 I2C 时钟产生示意图

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。

$$\text{SCL 高电平周期: } T_{SCL_HIGH} = T_{i2c_clk} * NDIV * (NHIGH + 1)$$

$$\text{SCL 低电平周期: } T_{SCL_LOW} = T_{i2c_clk} * NDIV * (NLOW + 1)$$

$$\text{SCL 下降沿到 SDA 变化时间: } T_{HOLD} = T_{i2c_clk} * NDIV * (NHOLD + 1)$$

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。

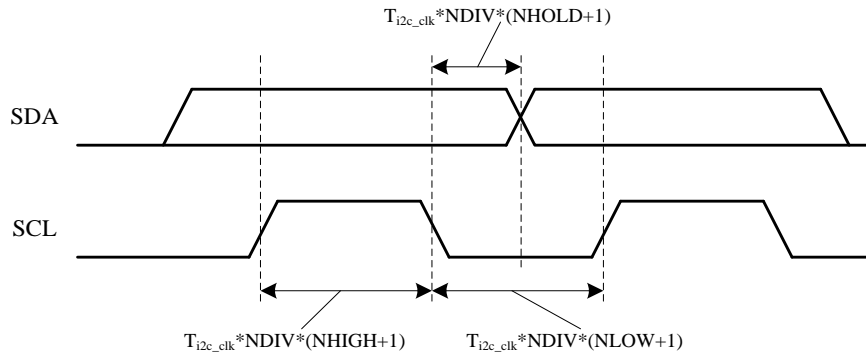


Figure 143 I2C 时钟分频

15.7 I2C 工作模式

15.7.1 从机模式

15.7.1.1 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。

15.7.1.2 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。

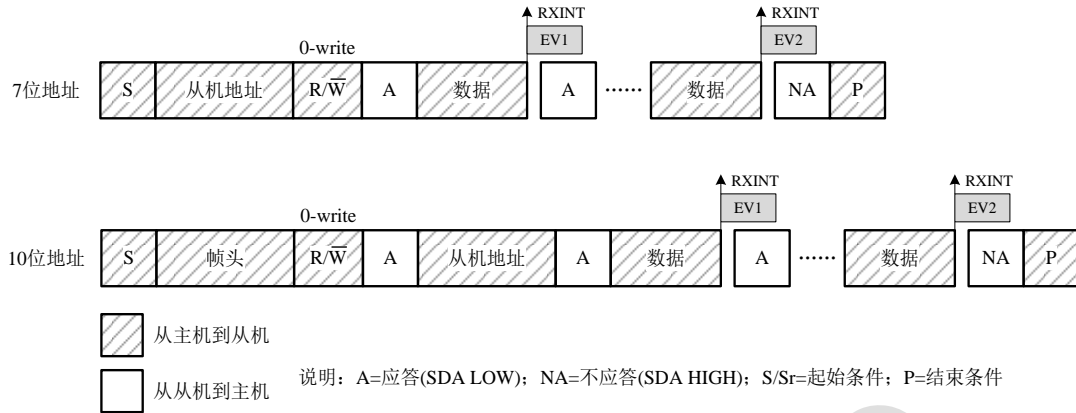


Figure 144 从机接收模式

15.7.1.3 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。

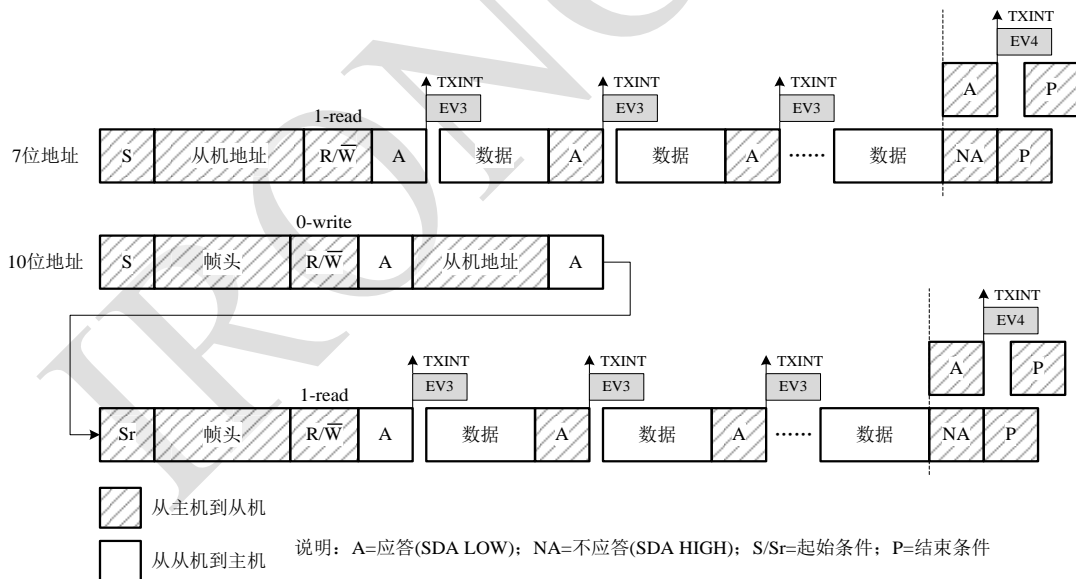


Figure 145 从机发送模式

15.7.1.4 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。

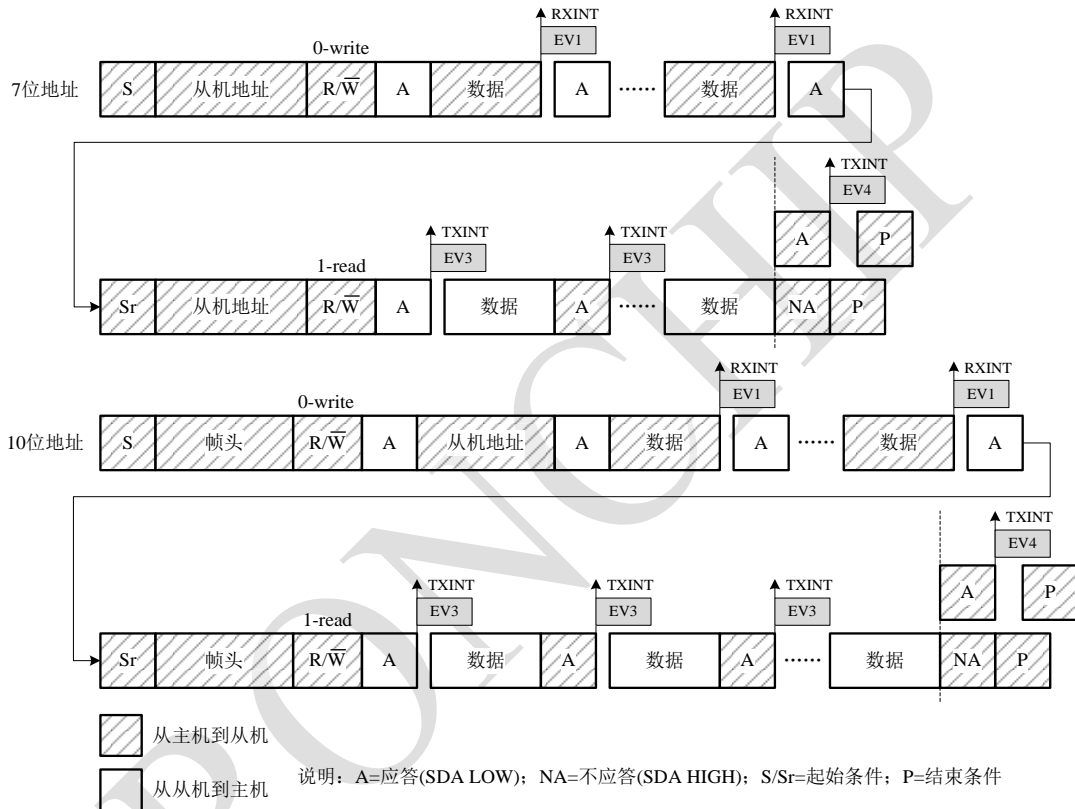


Figure 146 从机混合模式

15.7.2 主机模式

15.7.2.1 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”。

15.7.2.2 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“0000000”，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。

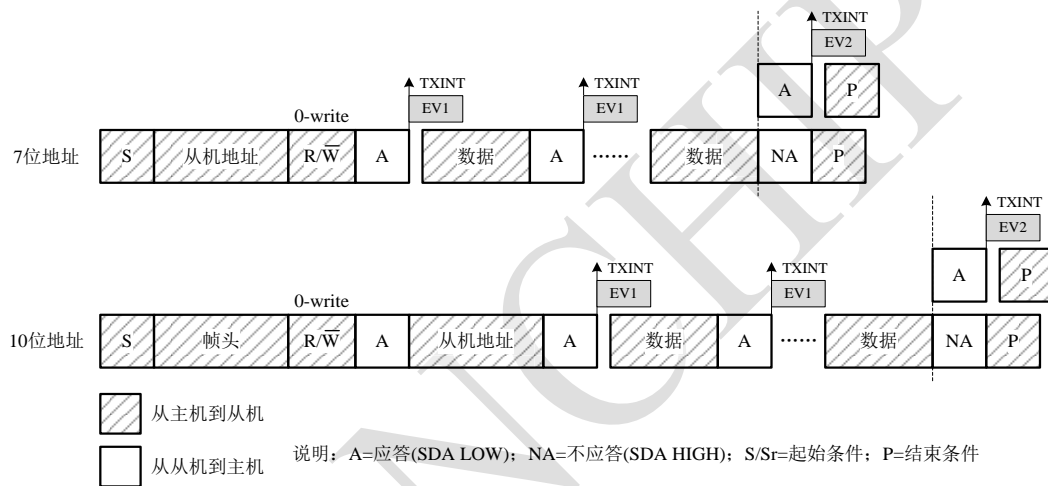


Figure 147 主机发送模式

15.7.2.3 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。

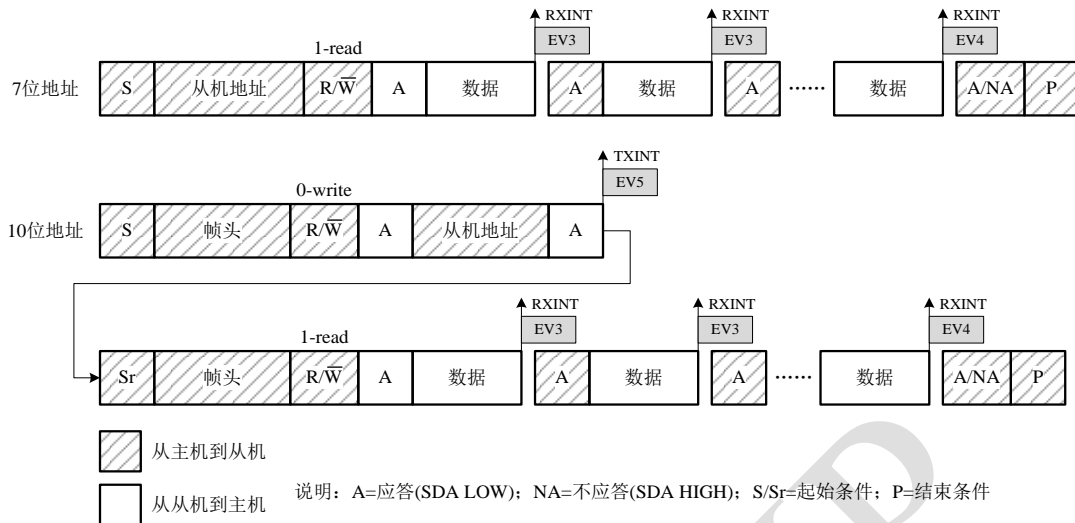


Figure 148 主机接收模式

15.7.2.4 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。

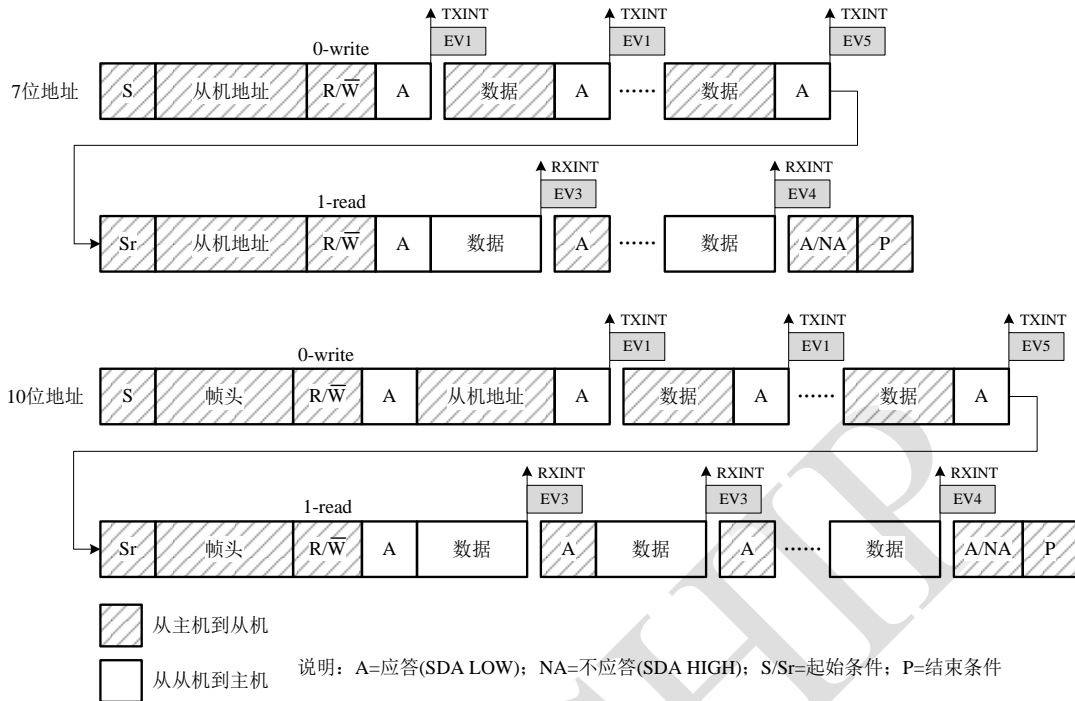


Figure 149 主机混合模式

15.8 I2C 中断

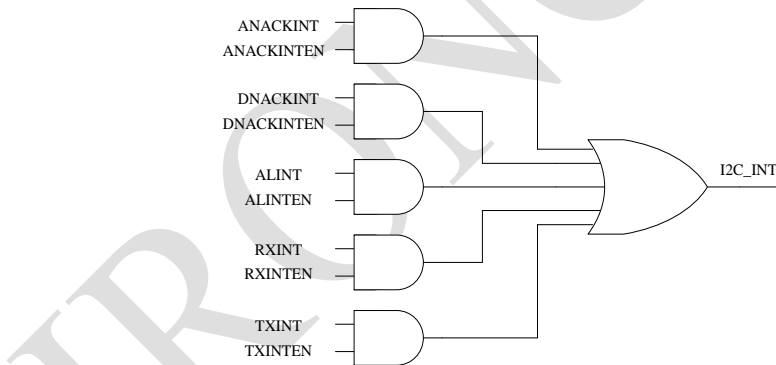


Figure 150 I2C 中断

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。

15.9 I2C 寄存器

I2C 模块有 14 个寄存器，对应的 SFRPAGE 为 0，具体说明如下。

Table 294 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

15.9.1 I2C 控制寄存器(I2CCON) [0xE1]

Table 295 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

15.9.2 I2C 从机地址寄存器低 8 位(I2CSADDRL) [0xE2]

Table 296 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

15.9.3 I2C 从机地址寄存器高 2 位(I2CSADDRH) [0xE3]

Table 297 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

15.9.4 I2C 本机地址寄存器低 8 位(I2COADDRL) [0xE4]

Table 298 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

15.9.5 I2C 本机地址寄存器高 2 位(I2COADDRH) [0xE5]

Table 299 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

15.9.6 I2C 工作时钟分频控制寄存器(I2CDIV) [0xE6]

Table 300 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	0x02	I2C 工作时钟分频控制： 0x00: 1 分频 0x01: 1 分频 0x02: 2 分频 0x03: 3 分频

参数名	比特位	属性	复位值	描述
			 0xfe: 254 分频 0xff: 255 分频

15.9.7 I2C 时钟 SCL 低电平时间配置寄存器(I2CDUTYL) [0xE7]

Table 301 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	0x0e	I2C 时钟 SCL 低电平时间配置: 0x00: $3 * NDIV * T_{i2c_clk}$ 0x01: $3 * NDIV * T_{i2c_clk}$ 0x02: $3 * NDIV * T_{i2c_clk}$ 0x03: $4 * NDIV * T_{i2c_clk}$ 0xfe: $255 * NDIV * T_{i2c_clk}$ 0xff: $256 * NDIV * T_{i2c_clk}$

15.9.8 I2C 时钟 SCL 高电平时间配置寄存器(I2CDUTYH) [0xE9]

Table 302 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	0x0e	I2C 时钟 SCL 高电平时间配置: 0x00: $3 * NDIV * T_{i2c_clk}$ 0x01: $3 * NDIV * T_{i2c_clk}$ 0x02: $3 * NDIV * T_{i2c_clk}$ 0x03: $4 * NDIV * T_{i2c_clk}$ 0xfe: $255 * NDIV * T_{i2c_clk}$ 0xff: $256 * NDIV * T_{i2c_clk}$

15.9.9 I2C 数据 SDA 保持时间配置寄存器(I2CHOLD) [0xEA]

Table 303 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	0x0e	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
				0x00: $1 * NDIV * T_{i2c_clk}$ 0x01: $1 * NDIV * T_{i2c_clk}$ 0x02: $2 * NDIV * T_{i2c_clk}$ 0x03: $3 * NDIV * T_{i2c_clk}$ 0xfe: $254 * NDIV * T_{i2c_clk}$ 0xff: $255 * NDIV * T_{i2c_clk}$

15.9.10 I2C 数据写缓冲寄存器(I2CWBUF) [0xEB]

Table 304 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

15.9.11 I2C 数据读缓冲寄存器(I2CRBUF) [0xEC]

Table 305 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

15.9.12 I2C 状态寄存器(I2CSTS) [0xED]

Table 306 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

15.9.13 I2C 中断标志寄存器(I2CISC) [0xEE]

Table 307 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

15.9.14 I2C 中断使能寄存器(I2CIEN) [0xEF]

Table 308 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

16 串行外设接口 (SPI)

16.1 概述

JMT1808R 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 $f_{spi_clk}/2$, 主机接收最高频率为 $f_{spi_clk}/4$
- 从机发送最高频率为 $f_{spi_clk}/4$, 从机接收最高频率为 $f_{spi_clk}/4$

16.2 框图

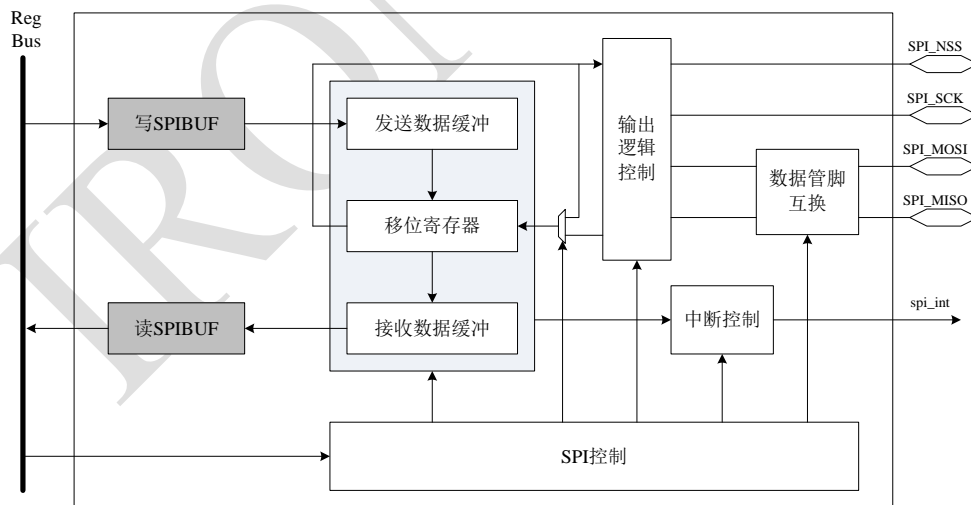


Figure 151 SPI 结构框图

16.3 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。

16.4 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。

16.4.1 4 线互连模式

4 线互连模式分为单从机和多从机方式。

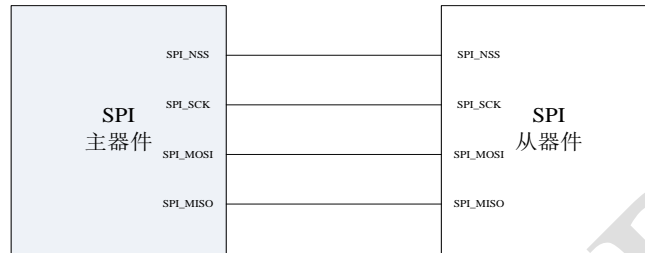


Figure 152 4 线单从机互连

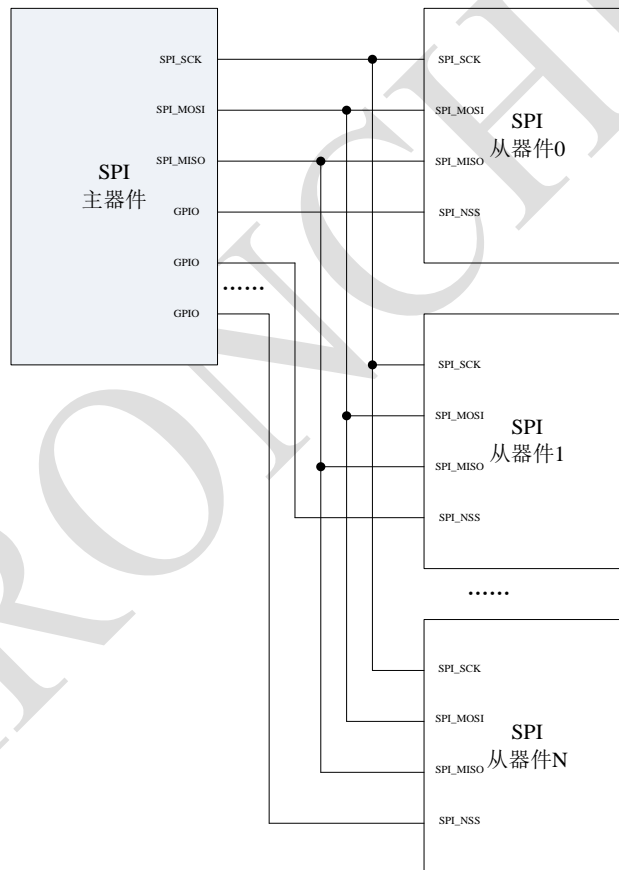


Figure 153 4 线多从机互连

16.4.2 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。

JMT1808R 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。

JMT1808R 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 作为从机数据收发管脚。

如 Figure 154 所示为 3 线默认连接方式，Figure 155 所示为从机数据管脚互换的 3 线多从机互连：

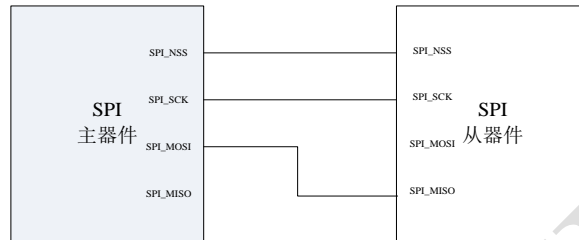


Figure 154 3 线单从机互连

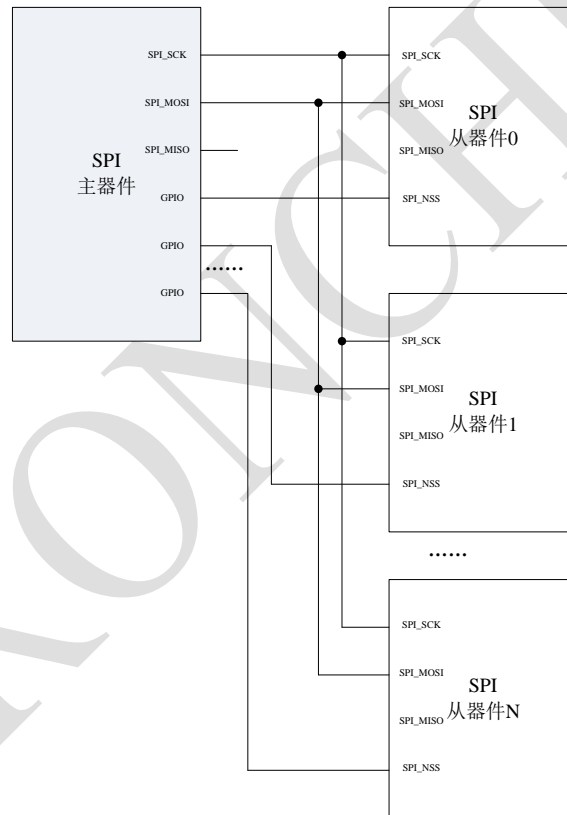


Figure 155 3 线多从机互连

16.5 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。

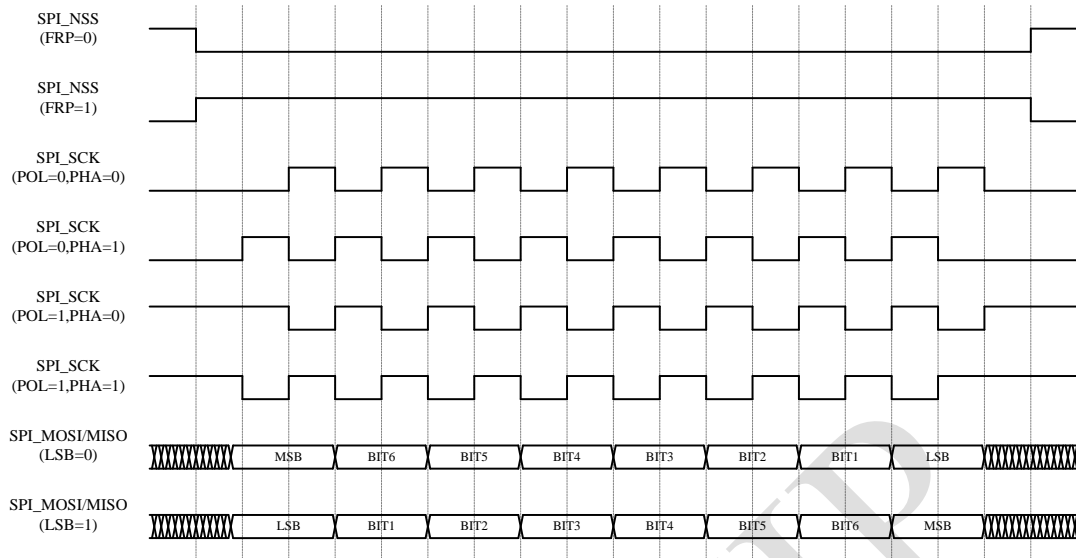


Figure 156 SPI 时序

16.6 数据发送和接收过程

16.6.1 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` 管脚上的接收数据移位到内部移位寄存器的最低比特。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。

16.6.1.1 4 线主机模式

4 线主机模式收发数据流程：

- 配置寄存器 `SPICON.SPIEN` 为 1，使能 SPI 模块
- 配置寄存器 `SPICON.MS` 为 0，使 SPI 工作于主机模式
- 配置寄存器 `SPICON.TWI` 为 0，选择 4 线 SPI 模式
- 配置寄存器 `SPICON.SOD` 为 0，使能 SPI 数据输出
- 配置寄存器 `SPICON.POL`，选择 `SPI_SCK` 极性
- 配置寄存器 `SPICON.PHA`，选择 `SPI_CLK` 相位
- 配置寄存器 `SPICON.FRP`，选择 `SPI_NSS` 极性
- 配置寄存器 `SPIBR`，选择 SPI 波特率

- 配置寄存器 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，可通过 SPIBUF 读取接收的从机数据

16.6.1.2 3 线主机模式

3 线主机模式发送数据流程：

- 配置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 配置寄存器 SPICON.MS 为 0，使 SPI 工作于主机模式
- 配置寄存器 SPICON.TWI 为 0，选择 3 线 SPI 模式
- 配置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 配置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 配置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 配置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 配置寄存器 SPIBR，选择 SPI 波特率
- 配置寄存器 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，向 SPI_MOSI（默认）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 线主机模式接收数据流程：

- 配置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 配置寄存器 SPICON.MS 为 0，使 SPI 工作于主机模式
- 配置寄存器 SPICON.TWI 为 0，选择 3 线 SPI 模式
- 配置寄存器 SPICON.SOD 为 1，关闭 SPI 数据输出
- 配置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 配置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 配置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 配置寄存器 SPIBR，选择 SPI 波特率
- 配置寄存器 SPICON.LSB，选择 SPI 数据比特顺序
- 向寄存器 SPIBUF 中写入任意数据，启动 SPI 传输，从 SPI_MOSI（默认）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 SPIBUF 读取接收的从机数据

16.6.2 从机模式

配置寄存器 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 寄存器配置。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。

16.6.2.1 4 线从机模式

4 线从机模式收发数据流程：

- 配置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 配置寄存器 SPICON.MS 为 1，使 SPI 工作于从机模式
- 配置寄存器 SPICON.TWI 为 0，选择 4 线 SPI 模式
- 配置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 配置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 配置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 配置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 配置寄存器 SPIBR，选择 SPI 波特率
- 配置寄存器 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

16.6.2.2 3 线从机模式

3 线从机模式发送数据流程：

- 配置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 配置寄存器 SPICON.MS 为 1，使 SPI 工作于从机模式
- 配置寄存器 SPICON.TWI 为 1，选择 3 线 SPI 模式
- 配置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 配置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 配置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 配置寄存器 SPICON.FRP，选择 SPI_NSS 极性

- 配置寄存器 SPIBR，选择 SPI 波特率
- 配置寄存器 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_MOSI（默认）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：

- 配置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 配置寄存器 SPICON.MS 为 1，使 SPI 工作于从机模式
- 配置寄存器 SPICON.TWI 为 1，选择 3 线 SPI 模式
- 配置寄存器 SPICON.SOD 为 1，关闭 SPI 数据输出
- 配置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 配置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 配置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 配置寄存器 SPIBR，选择 SPI 波特率
- 配置寄存器 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_MOSI（默认）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

16.7 SPI 中断

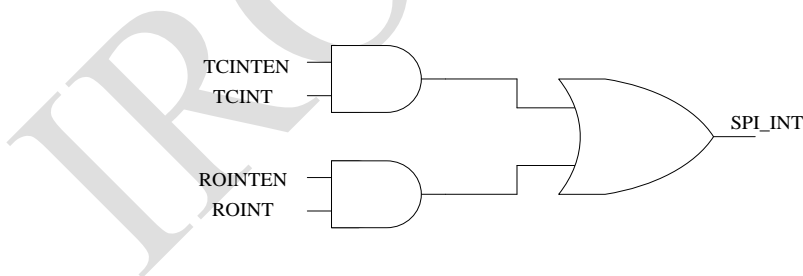


Figure 157 SPI 中断

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。

16.8 SPI 寄存器

SPI 模块有 5 个寄存器，对应的 SFRPAGE 为 0，具体说明如下。

Table 309 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

16.8.1 SPI 控制寄存器(SPICON) [0xC9]

Table 310 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： 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

参数名	比特位	属性	复位值	描述
				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 158 所示为两个 JMT1808R 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。

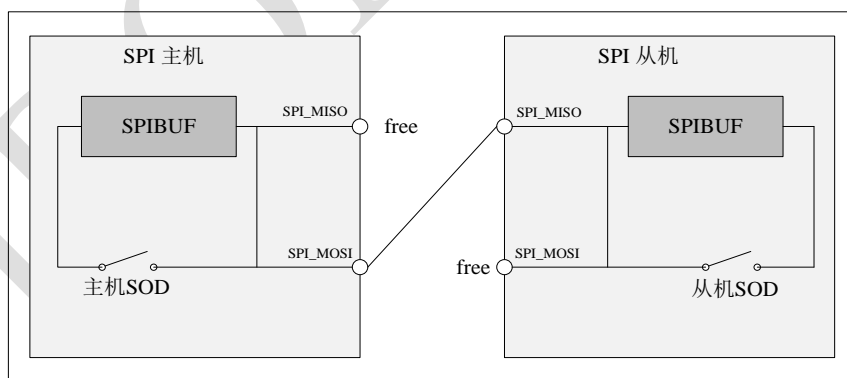


Figure 158 3 线 SPI 连接

16.8.2 SPI 中断使能寄存器(SPIIE) [0xCA]

Table 311 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

参数名	比特位	属性	复位值	描述
				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

16.8.3 SPI 中断标志寄存器(SPIIF) [0xCB]

Table 312 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 ● 不支持写 1 软中断
Reserved	7-2	R	0	保留

16.8.4 SPI 波特率控制寄存器(SPIBR) [0xCC]

Table 313 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:

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能

16.8.5 SPI 数据缓冲寄存器(SPIBUF) [0xCD]

Table 314 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	<p>SPI 数据缓冲寄存器:</p> <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

17 模拟/数字转换（ADC）和可编程增益放大器（PGA）

17.1 概述

JMT1808R 中内置两个逐次逼近型的模数转换器（SAR-ADC）：ADC0 和 ADC1；内置两个可编程增益放大器（PGA）：PGA0 和 PGA1。

SAR-ADC 的分辨率为 10 位，采样率最高为 500 KSPS，采样周期最小为 2us，每个 ADC 有 12 个模拟输入端供选择，结构框图如 Figure 159 所示。

ADCx 的转换通道序列分为普通通道序列和特殊通道序列：

- 普通通道序列，由 1~12 个模拟通道组成的通道序列，序列中的每个通道可以从 12 个模拟输入端中分别选择配置，序列长度可通过寄存器 ADCxQLC⁹.NL 配置，转换结果放入寄存器 ADCxNR 中
- 特殊通道序列，由 1~4 个模拟通道组成的通道序列，序列中的每个通道可以从 12 个模拟输入端中分别选择配置，序列长度可通过寄存器 ADCxQLC.SL 配置，转换结果分别放入对应的寄存器 ADCxSR0、ADCxSR1、ADCxSR2 和 ADCxSR3 中

ADC 转换启动有两种方式：

- 设置寄存器 ADCxCON.NQST 启动，只适用于普通通道序列
- 外部触发事件，适用于普通通道序列和特殊通道序列

JMT1808R 的两个 ADC 可以各自独立工作，也可以两个 ADC 协同工作。

ADCx 支持以下工作模式（普通通道序列和特殊通道序列均适用）：

- 单次转换+非扫描模式
- 单次转换+扫描模式
- 连续转换+非扫描模式
- 连续转换+扫描模式

两个 ADC 有以下协同工作模式：

- 独立模式
- 普通同步模式
- 特殊同步模式
- 普通交叉模式
- 特殊交替触发模式
- 普通同步+特殊同步模式
- 普通同步+特殊交替触发模式

⁹ADCx, x=0 或 1, 指示 ADC0 或 ADC1

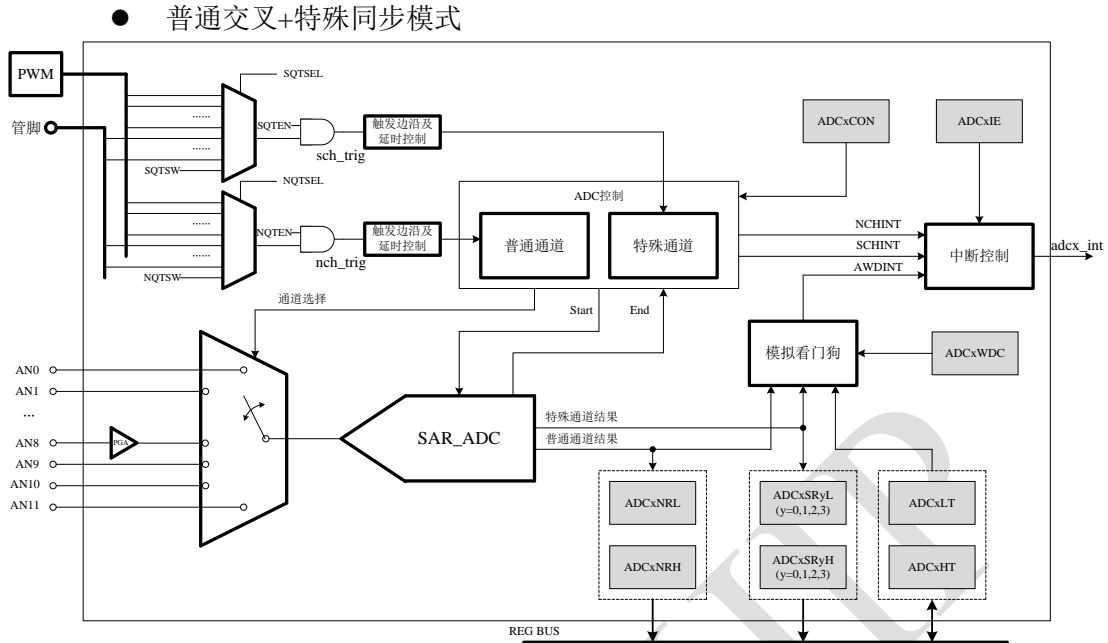


Figure 159 ADC 结构框图

17.2 ADC 转换结果访问方式

ADCx 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 17.15ADC 寄存器。

17.3 ADC 输入端口配置

2 个 ADC 的模拟输入端参见 Table 315：

Table 315 ADC 输入端对应关系

输入端 AN _x	ADC0 对应模拟输入	ADC1 对应模拟输入
AN0	PA0	PA0
AN1	PA1	PA1
AN2	PA2	PA2
AN3	PA3	PA3
AN4	PA4	PA4
AN5	PA5	PA5
AN6	PA6	PA6
AN7	PA7	PA7
AN8	PC0 经 PGA 输出	PC1 经 PGA 输出

输入端 ANx	ADC0 对应模拟输入	ADC1 对应模拟输入
AN9	PC2	PD3
AN10	PD4	PD5
AN11	PD6	PD7

其中 PA0~PA7、PC0~PC2 和 PD3~PD7 为芯片模拟输入管脚。当这些管脚作为模拟管脚使用时，需将寄存器 AIOEN0 和 AIOEN1 的相应通道置位，隔断数字对模拟的影响。

芯片内置 2 个可编程增益放大器 (PGA)，PC0 和 PC1 分别经过不同 PGA 连接到 ADC0 的 AN8 和 ADC1 的 AN8。

17.4 可编程增益放大器 (PGA) 说明

PGA 具体实现如 Figure 160 所示：

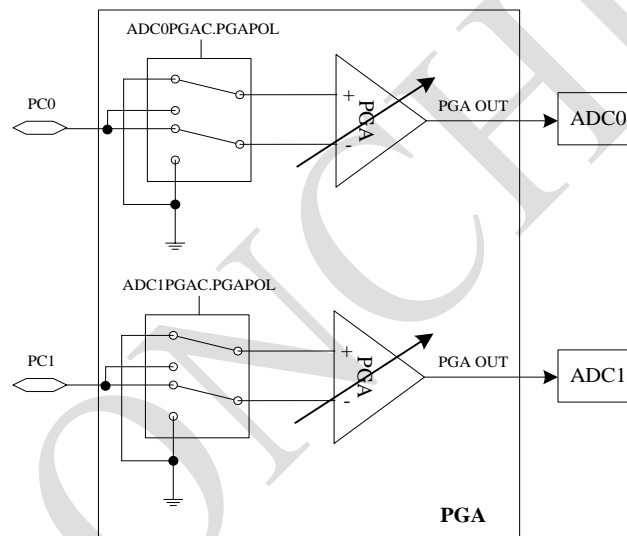


Figure 160 运算放大器内部框图

PGA 支持 1/4、1/2、1、2、4、8、16、32 倍放大，配置寄存器 ADCxPGAC.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 ADCxPGAC.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 160 所示：

- 寄存器 ADCxPGAC.PGAPOL 为 0 (反相)：PGA 正端接地，负端接相应管脚；
- 寄存器 ADCxPGAC.PGAPOL 为 1 (同相)：PGA 正端接相应管脚，负端接地。

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 的输入电压（输入电压取值范围 $-300\text{mV} \sim V_{REF}$ ）。

Figure 161 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。

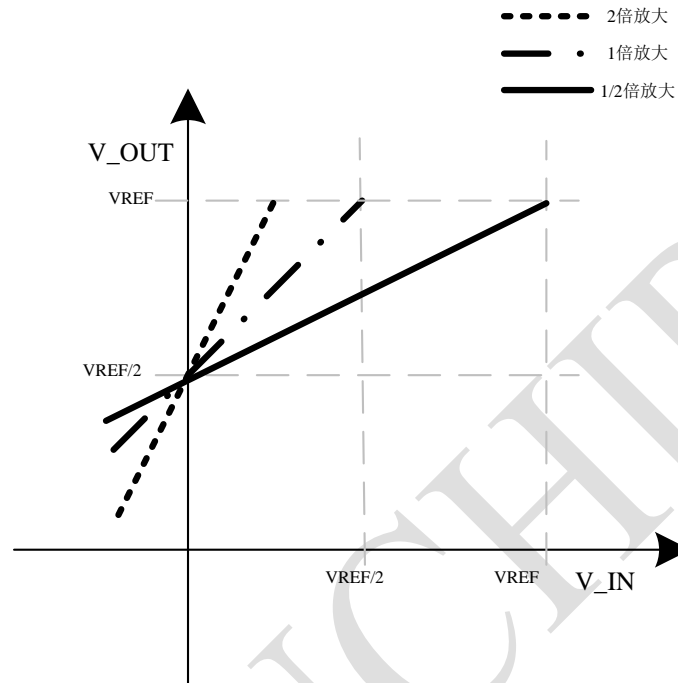


Figure 161 运算放大器同相，输入输出关系

Figure 162 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。

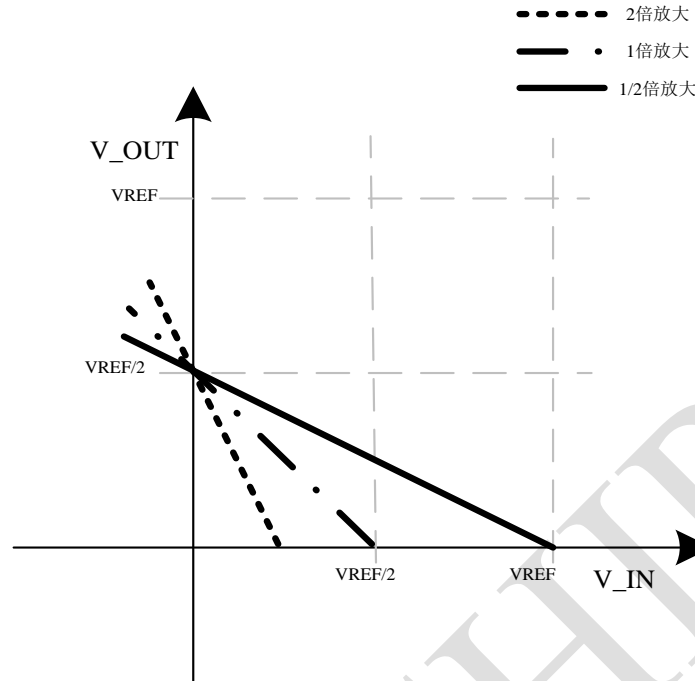


Figure 162 运算放大器反相，输入输出关系

配置寄存器 `ADCxPGAC.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。

17.5 ADC 参考电压配置

通过配置 `ADCJC.ECVREF_SEL` 寄存器，可选择 ADC 的参考电压（`VREF`）。

ADC 可供选择的参考电压（`VREF`）有：内部 2.5V 参考电压、芯片模拟供电 `VDDA`、芯片外部输入参考电压。

17.6 ADC 转换功能

芯片的 10 位 A/D 转换器，转换的最大值可达 `3FFH`。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/1024` 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。

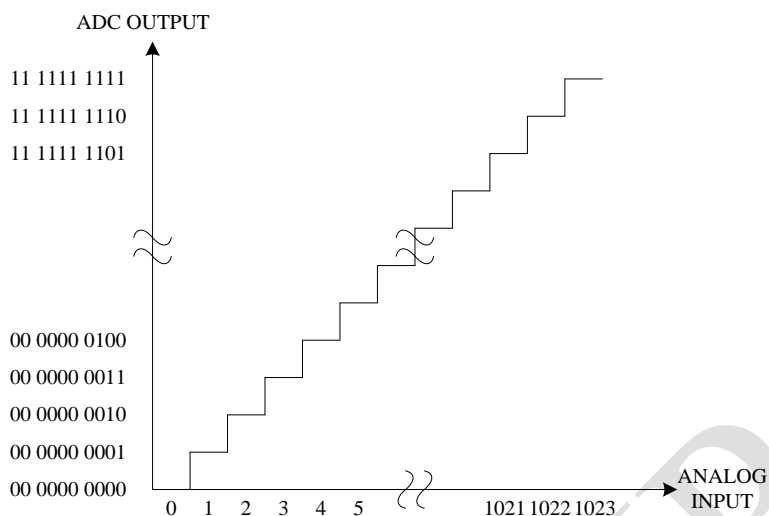


Figure 163 ADC 转换功能

17.7 ADC 转换结果数据格式

ADC 原始转换结果为 10 位数据，该 10 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 ADCxDF 进行选择，4 种数据格式参见 Figure 164 所示。

ADC原始转换结果		D9	D8	D7	D6	D5	D4	D3	D2	D1	D0					
ADCxDF																
00: 无符号整数	0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
01: 无符号小数	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0
10: 有符号整数	!D9	!D9	!D9	!D9	!D9	!D9	!D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
11: 有符号小数	!D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0

Figure 164 ADC 数据格式

17.8 ADC 转换时序

单次 ADC 时序如 Figure 165 所示。

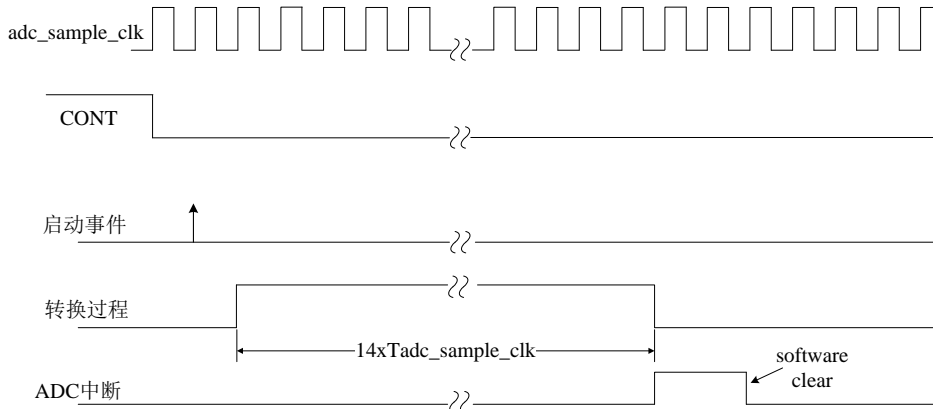


Figure 165 单次 ADC 时序图

连续 ADC 时序如 Figure 166 所示。

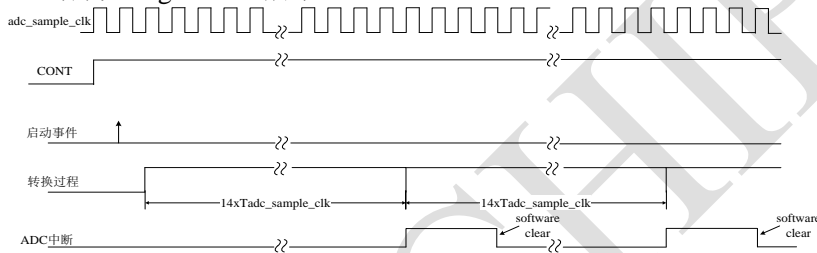


Figure 166 连续 ADC 时序图

17.9 模拟看门狗

如果被 ADC 转换的模拟电压低于低阈值或高于高阈值，模拟看门狗中断标志位 WDINT 被置 1。高阈值位于 ADCxHT 寄存器中（需要先配置高字节 ADCxHTH，再配置低字节 ADCxHTL，寄存器才能正常生效），低阈值位于 ADCxLT 寄存器中（需要先配置高字节 ADCxLTH，再配置低字节 ADCxLTL，寄存器才能正常生效）。通过设置 ADCxIE 寄存器的 WDINTEN 位来控制模拟看门狗中断使能。模拟看门狗可以作用于 1 个或多个通道。

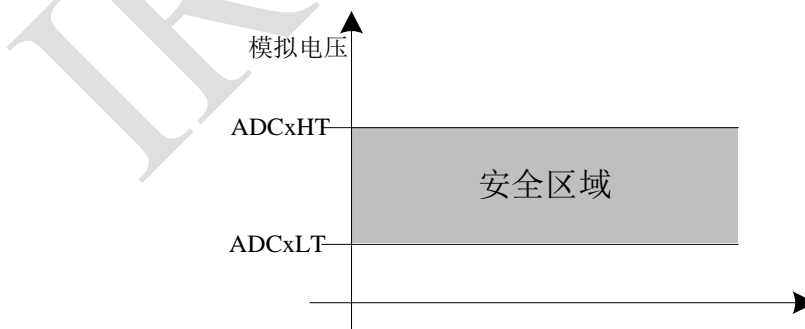


Figure 167 模拟看门狗功能

通过配置寄存器 ADCxWDC.SCHEN, ADCxWDC.NWDEN 和 ADCxWDC.SWDEN, 选择模拟看门狗监控的通道，具体参见 Table 316。

Table 316 模拟看门狗通道控制

模拟看门狗通道	SCHEN	NWDEN	SWDEN
无	x	0	0
所有特殊通道	0	0	1
所有普通通道	0	1	0
所有普通和特殊通道	0	1	1
WDCH 指定的特殊通道	1	0	1
WDCH 指定的普通通道	1	1	0
WDCH 指定的普通或特殊通道	1	1	1

17.10 外部触发事件

普通通道序列和特殊通道序列都可以由外部事件触发(例如 PWM 触发、管脚边沿触发),配置过程如下:

- 配置寄存器 ADCxNQTC.NQTEN 为 1, 使能外部事件触发普通通道序列转换; 配置寄存器 ADCxSQTC.SQTEN 为 1, 使能外部事件触发特殊通道序列转换;
- 配置寄存器 ADCxNQTC.NQTSEL[4:0]和 ADCxSQTC.SQTSEL[4:0], 选择触发的外部事件。
- 配置寄存器 ADCxETC, 选择外部触发事件的有效沿;
- 配置寄存器 ADCxNQTD, 选择普通通道外部触发事件启动 ADC 采样的延时; 配置寄存器 ADCxSQTD, 选择特殊通道外部触发事件启动 ADC 采样的延时。

外部触发事件启动 ADC 采样延时功能, 如 Figure 168 所示。

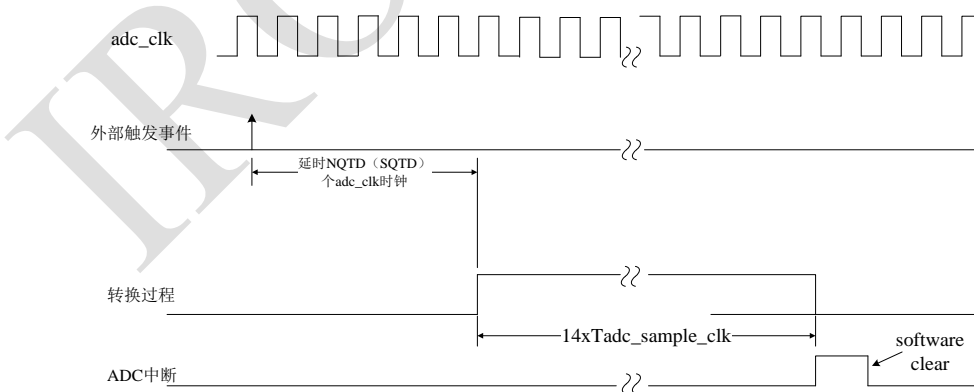


Figure 168 外部触发事件延时功能

Table 317 ADC0 和 ADC1 普通通道外部触发

触发源	类型	NQTSEL
TIM0_OC0	PWM 内部信号, 见章节	000000
TIM0_OC1	12.5.5 主模式控制	000001

触发源	类型	NQTSEL
TIM0_OC2		000010
TIM0_OC3		000011
TIM1_OC0		000100
TIM1_OC1		000101
TIM1_OC2		000110
TIM1_OC3		000111
TIM0_TRGO		001100
TIM1_TRGO		001101
INT0_0		外部引脚输入滤波后信号，见 Figure 201 INT0 中断和 Figure 202 INT1 中断
INT0_1	010001	
INT0_2	010010	
INT0_3	010011	
INT0_4	010100	
INT0_5	010101	
INT0_6	010110	
INT0_7	010111	
INT1_0	011000	
INT1_1	011001	
INT1_2	011010	
INT1_3	011011	
INT1_4	011100	
INT1_5	011101	
INT1_6	011110	
INT1_7	011111	
NQTSW	软件控制，软件配置 ADCxCON.NQTSW 为 1，产生触发信号	100000

Table 318 ADC0 和 ADC1 特殊通道外部触发

触发源	类型	NQTSEL
TIM0_OC0	PWM 内部信号，见章节 12.5.5 主模式控制	000000
TIM0_OC1		000001

触发源	类型	NQTSEL
TIM0_OC2		000010
TIM0_OC3		000011
TIM1_OC0		000100
TIM1_OC1		000101
TIM1_OC2		000110
TIM1_OC3		000111
TIM0_TRGO		001100
TIM1_TRGO		001101
INT0_0	外部引脚输入滤波后信号，见 Figure 201 INT0 中断和 Figure 202 INT1 中断	010000
INT0_1		010001
INT0_2		010010
INT0_3		010011
INT0_4		010100
INT0_5		010101
INT0_6		010110
INT0_7		010111
INT1_0		011000
INT1_1		011001
INT1_2		011010
INT1_3		011011
INT1_4		011100
INT1_5		011101
INT1_6		011110
INT1_7		011111
SQTSW	软件控制，软件配置 ADCxCON.SQTSW 为 1，产生触发信号	100000

17.11 特殊通道序列管理

特殊通道序列有以下两种触发方式：

- 当 ADCxCON.SQAUTO=0 时，需要特殊通道外部触发事件启动特殊通道采样。特殊通道触发的优先级高于普通通道，特殊通道序列的外部触发可将正在处理

的普通通道序列打断，特殊通道序列转换完成后，再继续处理普通通道序列未完成部分；如果未发生特殊通道外部触发事件，则特殊通道不进行转换。

- 当 ADCxCON.SQAUTO=1 时，不需要特殊通道外部触发事件。普通通道序列处理完后，特殊通道在其后接着自动处理。

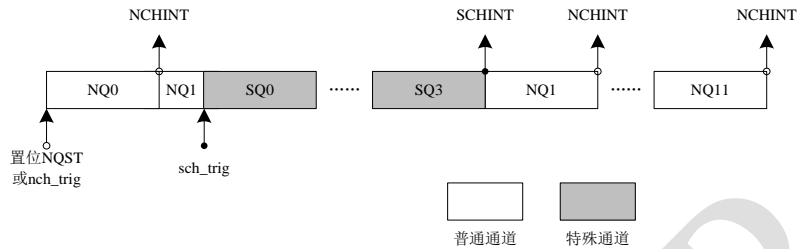


Figure 169 特殊通道序列外部触发方式

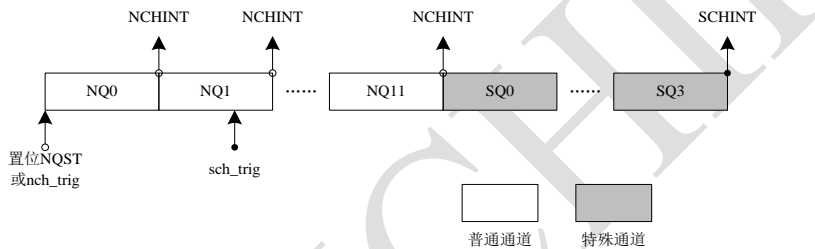


Figure 170 特殊通道序列自动触发方式

17.12 ADC 工作模式

ADCx 支持以下工作模式（普通通道序列和特殊通道序列均适用）：

- 单次转换+非扫描模式
- 单次转换+扫描模式
- 连续转换+非扫描模式
- 连续转换+扫描模式

单次转换指只对所选通道进行一次 ADC 转换；连续转换指对所选通道进行循环连续转换。非扫描模式指只对所选通道序列的第一个通道进行 ADC 转换；扫描模式指对所选通道序列的所有通道顺序进行 ADC 转换。

17.12.1 单次转换+非扫描模式

在该模式下，普通通道序列工作流程：

- 配置寄存器 ADCxCON.CONT 和 ADCxCON.SCAN 都为 0，使 ADCx 工作在单次转换模式，且关闭扫描模式；
- 配置寄存器 ADCxNQTC.NQTEN 为 1，使能外部事件触发普通通道序列转换；
- 配置寄存器 ADCxNQTC.NQTSEL[4:0]，选择触发的外部事件；
- 配置寄存器 ADCxETC，选择外部触发事件的有效沿；

- 配置寄存器 ADCxNQTD, 选择普通通道外部触发事件启动 ADC 采样的延时;
- 软件置位 ADCxCON.NQST 或发生普通通道外部触发事件, 硬件启动寄存器 ADCxNQ0.NQ0 对应的模拟通道进行 ADC 转换;
- 转换完成后, ADCxNQ0.NQ0 对应模拟通道的转换结果存储到 ADCxNRH 和 ADCxNRL 中, ADCxIF.NCHINT 硬件置位, 并产生中断(如果中断被使能);
- 中断处理, 软件清零 ADCxIF.NCHINT。

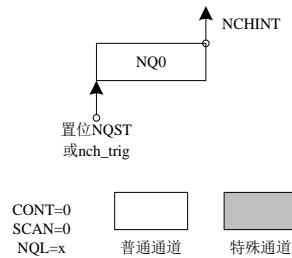


Figure 171 普通通道单次转换+非扫描模式

该模式下, 特殊通道序列工作流程:

- 配置寄存器 ADCxCON.CONT 和 ADCxCON.SCAN 都为 0, 使 ADCx 工作在单次转换模式, 且关闭扫描模式;
- 配置寄存器 ADCxSQTC.SQTEN 为 1, 使能外部事件触发特殊通道序列转换;
- 配置寄存器 ADCxSQTC.SQTSEL[4:0], 选择触发的外部事件;
- 配置寄存器 ADCxETC, 选择外部触发事件的有效沿;
- 配置寄存器 ADCxSQTD, 选择特殊通道外部触发事件启动 ADC 采样的延时;
- 如果发生特殊通道外部触发事件, 硬件启动 ADCxSQ0.SQ0 对应的模拟通道进行 ADC 转换;
- 转换完成后, ADCxSQ0.SQ0 转换结果存储到 ADCxSR0H 和 ADCxSR0L 中, ADCxIF.SCHINT 硬件置位, 并产生中断(如果中断被使能);
- 中断处理, 软件清零 ADCxIF.SCHINT。

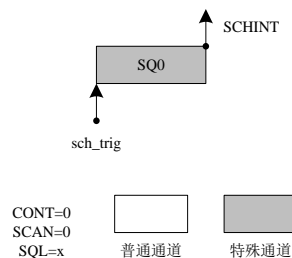


Figure 172 特殊通道单次转换+非扫描模式

17.12.2 单次转换+扫描模式

该模式下, 普通通道序列工作流程:

- 配置寄存器 ADCxCON.CONT 为 0, 使 ADCx 工作在单次转换模式; 配置寄存器 ADCxCON.SCAN 为 1, 打开扫描模式;
- 配置寄存器 ADCxNQTC.NQTEN 为 1, 使能外部事件触发普通通道序列转换;
- 配置寄存器 ADCxNQTC.NQTSEL[4:0], 选择触发的外部事件;
- 配置寄存器 ADCxETC, 选择外部触发事件的有效沿;
- 配置寄存器 ADCxNQTD, 选择普通通道外部触发事件启动 ADC 采样的延时;
- 软件置位 ADCxCON.NQST 或发生普通通道外部触发事件, 硬件启动寄存器 ADCxNQx 和 ADCxQLC.NQL 所定义的模拟通道进行 ADC 转换;
- 单个通道转换完成后, 转换结果存储到 ADCxNRH 和 ADCxNRL 中, ADCxIF.NCHINT 硬件置位, 并产生中断 (如果中断被使能), 接着硬件自动启动下一个模拟通道的转换; 产生中断后, 软件需要及时把转换结果从寄存器中读出, 不然下个通道的转换结果会将其覆盖;
- 中断处理, 软件清零 ADCxIF.NCHINT;
- 所有通道转换完成后, 转换结束。

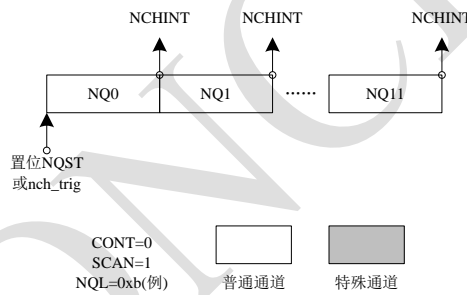


Figure 173 普通通道单次转换+扫描模式

该模式下, 特殊通道序列工作流程:

- 配置寄存器 ADCxCON.CONT 为 0, 使 ADCx 工作在单次转换模式; 配置寄存器 ADCxCON.SCAN 为 1, 打开扫描模式;
- 配置寄存器 ADCxSQTC.SQTEN 为 1, 使能外部事件触发特殊通道序列转换;
- 配置寄存器 ADCxSQTC.SQTSEL[4:0], 选择触发的外部事件;
- 配置寄存器 ADCxETC, 选择外部触发事件的有效沿;
- 配置寄存器 ADCxSQTD, 选择特殊通道外部触发事件启动 ADC 采样的延时;
- 如果发生特殊通道外部触发事件, 硬件启动 ADCxSQx 和 ADCxQLC.SQL 所定义的模拟通道进行 ADC 转换;
- 单个通道转换完成后, 自动启动下个模拟通道转换, ADCxSQ0.SQ0 转换结果存储到 ADCxSR0H 和 ADCxSR0L 中, ADCxSQ0.SQ1 转换结果存储到 ADCxSR1H 和 ADCxSR1L 中, ADCxSQ1.SQ2 转换结果存储到 ADCxSR2H 和

ADCxSR2L 中, ADCxSQ1.SQ3 转换结果存储到 ADCxSR3H 和 ADCxSR3L 中, 所有通道转换完成后, ADCxIF.SCHINT 硬件置位, 并产生中断 (如果中断被使能);

- 中断处理, 软件清零 ADCxIF.SCHINT。

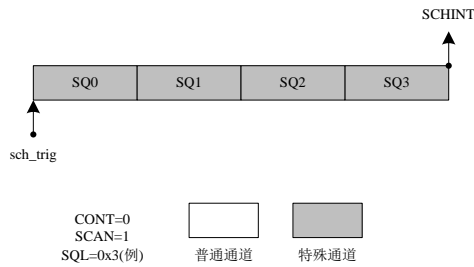


Figure 174 特殊通道单次转换+扫描模式

17.12.3 连续转换+非扫描模式

该模式下, 普通通道序列工作流程:

- 配置寄存器 ADCxCON.CONT 为 1, 使 ADCx 工作在连续转换模式; 配置寄存器 ADCxCON.SCAN 为 0, 关闭扫描模式;
- 配置寄存器 ADCxNQTC.NQTEN 为 1, 使能外部事件触发普通通道序列转换;
- 配置寄存器 ADCxNQTC.NQTSEL[4:0], 选择触发的外部事件;
- 配置寄存器 ADCxETC, 选择外部触发事件的有效沿;
- 配置寄存器 ADCxNQTD, 选择普通通道外部触发事件启动 ADC 采样的延时;
- 软件置位 ADCxCON.NQST 或发生普通通道外部触发事件, 硬件启动 ADCxNQ0.NQ0 对应的模拟通道进行 ADC 转换;
- 转换完成后, 转换结果存储到 ADCxNRH 和 ADCxNRL 中, ADCxIF.NCHINT 硬件置位, 产生中断 (如果中断被使能), 接着硬件自动启动 ADCxNQ0.NQ0 对应模拟通道的下一次 ADC 转换, 如此循环, 直到软件清零 ADCxCON.CONT。

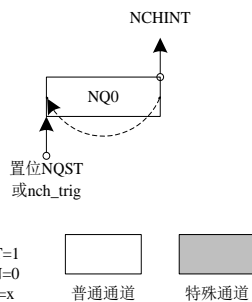


Figure 175 普通通道连续转换+非扫描模式

该模式下, 特殊通道序列工作流程:

- 配置寄存器 ADCxCON.CONT 为 1, 使 ADCx 工作在连续转换模式; 配置寄存

器 ADCxCON.SCAN 为 0，关闭扫描模式：

- 配置寄存器 ADCxSQTC.SQTEN 为 1，使能外部事件触发特殊通道序列转换；
- 配置寄存器 ADCxSQTC.SQTSEL[4:0]，选择触发的外部事件；
- 配置寄存器 ADCxETC，选择外部触发事件的有效沿；
- 配置寄存器 ADCxSQTD，选择特殊通道外部触发事件启动 ADC 采样的延时；
- 如果发生特殊通道外部触发事件，硬件启动 ADCxSQ0.SQ0 对应的模拟通道进行 ADC 转换；
- 转换完成后，转换结果存储到 ADCxSR0H 和 ADCxSR0L 中，ADCxIF.SCHINT 硬件置位，产生中断（如果中断被使能），接着硬件自动启动 ADCxSQ0.SQ0 对应模拟通道的下一次 ADC 转换，如此循环，直到软件清零 ADCxCON.CONT。

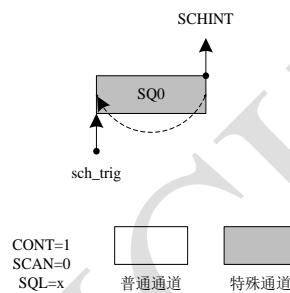


Figure 176 特殊通道连续转换+非扫描模式

17.12.4 连续转换+扫描模式

该模式下，普通通道序列工作流程：

- 配置寄存器 ADCxCON.CONT 为 1，使 ADCx 工作在连续转换模式；配置寄存器 ADCxCON.SCAN 为 1，打开扫描模式；
- 配置寄存器 ADCxNQTC.NQTEN 为 1，使能外部事件触发普通通道序列转换；
- 配置寄存器 ADCxNQTC.NQTSEL[4:0]，选择触发的外部事件；
- 配置寄存器 ADCxETC，选择外部触发事件的有效沿；
- 配置寄存器 ADCxNQTD，选择普通通道外部触发事件启动 ADC 采样的延时；
- 软件置位 ADCxCON.NQST 或发生普通通道外部触发事件，硬件启动 ADCxNQx 和 ADCxQLC.NQL 所定义的模拟通道进行 ADC 转换；
- 单个通道转换完成后，转换结果存储到 ADCxNRH 和 ADCxNRL 中，ADCxIF.NCHINT 硬件置位，并产生中断（如果中断被使能），接着硬件自动启动下一个模拟通道的转换；产生中断后，软件需要及时把转换结果从寄存器中读出，不然下个通道的转换结果会将其覆盖；
- 普通通道序列中所有通道转换完成后，硬件重新从序列的第一个通道继续转换，直到软件清零 ADCxCON.CONT。

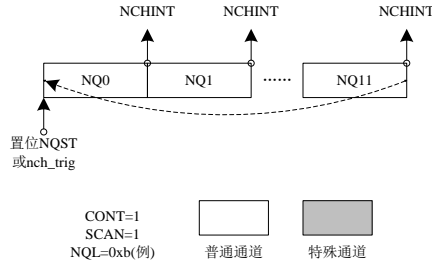


Figure 177 普通通道连续转换+扫描模式

该模式下，特殊通道序列工作流程：

- 配置寄存器 ADCxCON.CONT 为 1，使 ADCx 工作在连续转换模式；配置寄存器 ADCxCON.SCAN 为 1，打开扫描模式；
- 配置寄存器 ADCxSQTC.SQTEN 为 1，使能外部事件触发特殊通道序列转换；
- 配置寄存器 ADCxSQTC.SQTSEL[4:0]，选择触发的外部事件；
- 配置寄存器 ADCxETC，选择外部触发事件的有效沿；
- 配置寄存器 ADCxSQTd，选择特殊通道外部触发事件启动 ADC 采样的延时；
- 如果发生特殊通道外部触发事件，硬件启动 ADCxSQx 和 ADCxQLC.SQL 所定义的模拟通道进行 ADC 转换；
- 单个通道转换完成后，自动启动下个模拟通道转换，ADCxSQ0.SQ0 转换结果存储到 ADCxSR0H 和 ADCxSR0L 中，ADCxSQ0.SQ1 转换结果存储到 ADCxSR1H 和 ADCxSR1L 中，ADCxSQ1.SQ2 转换结果存储到 ADCxSR2H 和 ADCxSR2L 中，ADCxSQ1.SQ3 转换结果存储到 ADCxSR3H 和 ADCxSR3L 中，所有通道转换完成后，ADCxIF.SCHINT 硬件置位，并产生中断（如果中断被使能）；
- 中断处理，软件清零 ADCxIF.SCHINT；
- 特殊通道序列中的所有通道转换完成后，硬件重新从序列的第一个通道继续转换，直到软件清零 ADCxCON.CONT。

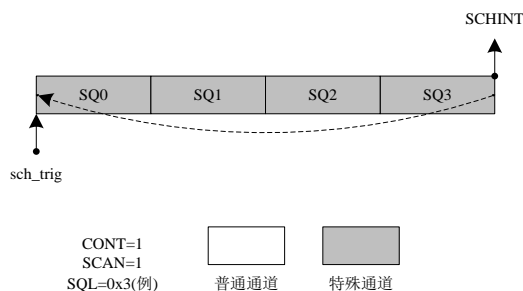


Figure 178 特殊通道连续转换+扫描模式

17.13 双 ADC 协同工作模式

两个 ADC 有以下协同工作模式：

- 独立模式
- 普通同步模式
- 特殊同步模式
- 普通交叉模式
- 特殊交替触发模式
- 普通同步+特殊同步模式
- 普通同步+特殊交替触发模式
- 普通交叉+特殊同步模式

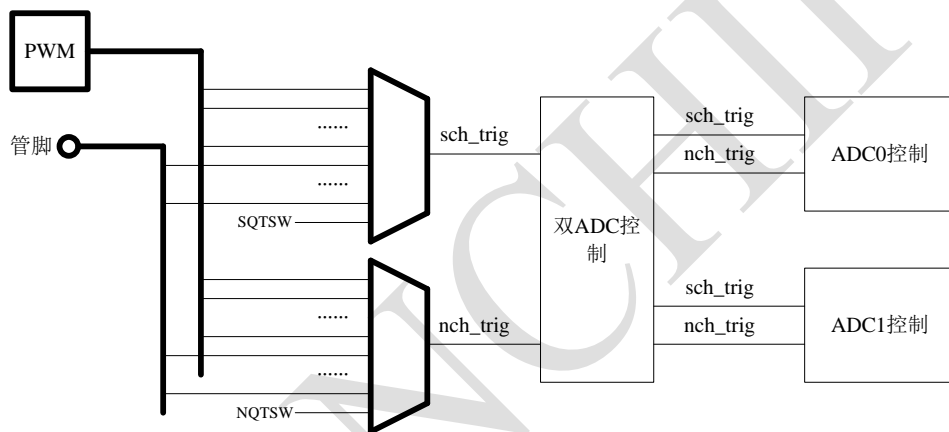


Figure 179 双 ADC 工作模式框图

17.13.1 独立模式

配置寄存器 ADCDMC.DMS 为 000，两个 ADC 独立工作，相互没有影响。

17.13.2 普通同步模式

配置寄存器 ADCDMC.DMS 为 001，ADC0 和 ADC1 工作在普通同步模式。

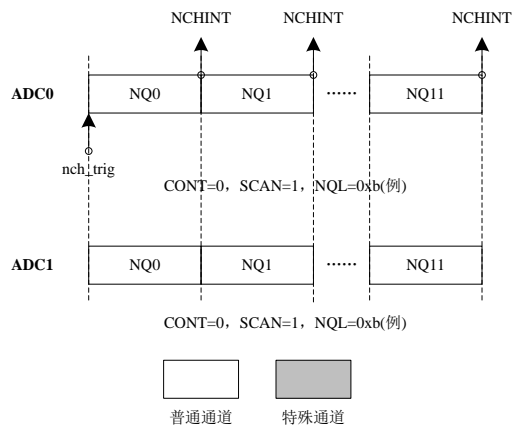


Figure 180 双 ADC 普通同步模式

该模式下, ADC0 的普通通道外部触发事件启动两个 ADC 同时工作。每个普通通道转换完成后, ADCx 的转换结果存储到 ADCxNRH 和 ADCxNRL, ADCxIF.NCHINT 硬件置位。软件只需要响应一个 ADC 的 NCHINT 中断, 同时读取两个 ADC 的转换结果。

普通同步模式下, 以下寄存器配置要求:

- 配置寄存器 ADC0CON.CONT 和 ADC1CON.CONT 为相同值
- 配置寄存器 ADC0CON.SCAN 和 ADC1CON.SCAN 为相同值
- 配置寄存器 ADC0QLC.NQL 和 ADC1QLC.NQL 为相同值

Figure 180 给出了 CONT 为 0, SCAN 为 1, NQL 为 0x0b 的场景, 即单次转换, 扫描模式下的普通同步模式。

17.13.3 特殊同步模式

配置寄存器 ADCDMC.DMS 为 010, ADC0 和 ADC1 工作在特殊同步模式。

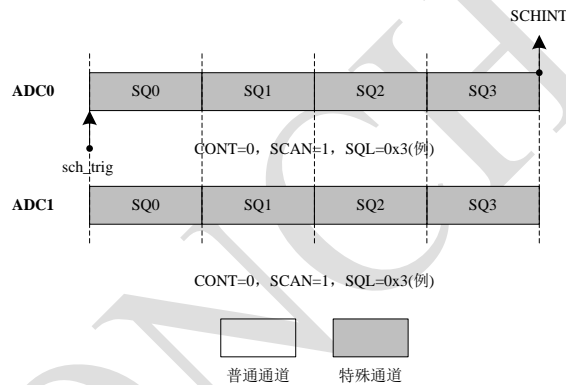


Figure 181 双 ADC 特殊同步模式

该模式下, ADC0 的特殊通道外部触发事件启动两个 ADC 同时工作。整个特殊通道序列转换完成后, ADCxSQ0.SQ0 转换结果存储到 ADCxSR0H 和 ADCxSR0L 中, ADCxSQ0.SQ1 转换结果存储到 ADCxSR1H 和 ADCxSR1L 中, ADCxSQ1.SQ2 转换结果存储到 ADCxSR2H 和 ADCxSR2L 中, ADCxSQ1.SQ3 转换结果存储到 ADCxSR3H 和 ADCxSR3L 中, ADCxIF.SCHINT 硬件置位。软件只需要响应一个 ADC 的 SCHINT 中断, 同时读取两个 ADC 的转换结果。

特殊同步模式下, 以下寄存器配置要求:

- 配置寄存器 ADC0CON.CONT 和 ADC1CON.CONT 为相同值
- 配置寄存器 ADC0CON.SCAN 和 ADC1CON.SCAN 为相同值
- 配置寄存器 ADC0QLC.SQL 和 ADC1QLC.SQL 为相同值

Figure 181 给出了 CONT 为 0, SCAN 为 1, SQL 为 0x03 的场景, 即单次转换, 扫描模式下的特殊同步模式。

17.13.4 普通交叉模式

配置寄存器 ADCDMC.DMS 为 011, ADC0 和 ADC1 工作在普通交叉模式。

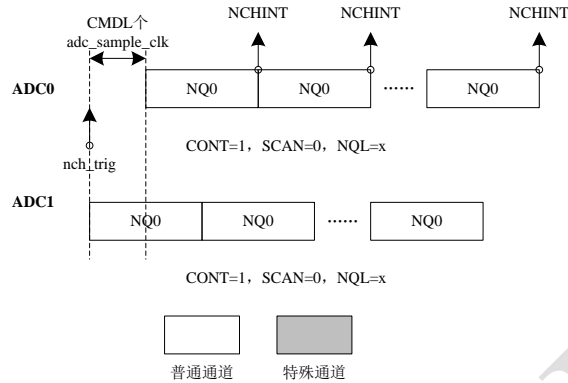


Figure 182 双 ADC 普通交叉模式

该模式下, ADC0 的普通通道外部触发事件首先启动 ADC1 普通通道转换, CMDL (通过寄存器 ADCDMC.CMDL 配置) 个 `adc_sample_clk` 时钟之后启动 ADC0 普通通道转换。每个普通通道转换完成后, ADCx 的转换结果存储到 `ADCxNRH` 和 `ADCxNRL`, `ADCxIF.NCHINT` 硬件置位。软件只需要响应 ADC0 的 NCHINT 中断, 同时读取两个 ADC 的转换结果。

普通交叉模式下, 以下寄存器配置要求:

- 配置寄存器 `ADC0CON.CONT` 和 `ADC1CON.CONT` 为相同值
- 配置寄存器 `ADC0CON.SCAN` 和 `ADC1CON.SCAN` 为相同值
- 配置寄存器 `ADC0QLC.NQL` 和 `ADC1QLC.NQL` 为相同值

Figure 182 给出了 `CONT` 为 1, `SCAN` 为 0, `NQL` 为任意值的场景, 即连续转换, 非扫描模式下的普通交叉模式。

17.13.5 特殊交替触发模式

配置寄存器 ADCDMC.DMS 为 100, ADC0 和 ADC1 工作在特殊交替触发模式。

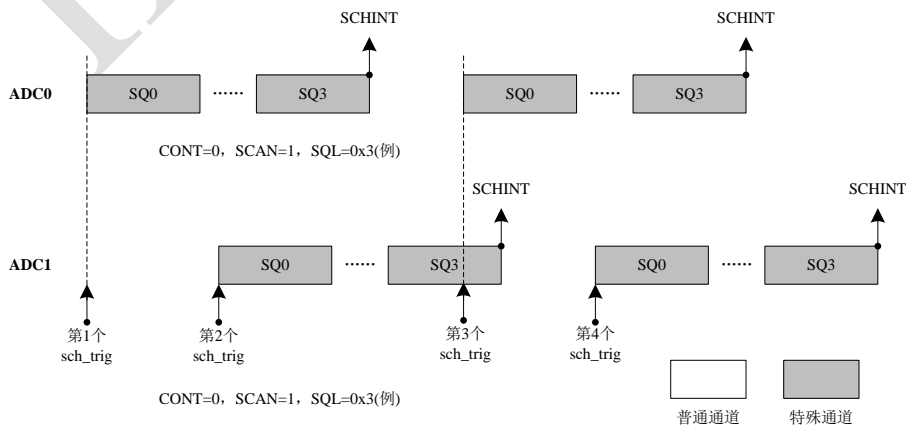


Figure 183 双 ADC 特殊交替触发模式

该模式下，ADC0 的特殊通道外部触发事件启动两个 ADC 特殊通道工作：第 1 个特殊通道触发启动 ADC0 特殊通道工作，第 2 个特殊通道触发启动 ADC1 特殊通道工作.....第奇数个特殊通道触发启动 ADC0 特殊通道工作，第偶数个特殊通道触发启动 ADC1 特殊通道工作。整个特殊通道序列转换完成后，ADCxSQ0.SQ0 转换结果存储到 ADCxSR0H 和 ADCxSR0L 中，ADCxSQ0.SQ1 转换结果存储到 ADCxSR1H 和 ADCxSR1L 中，ADCxSQ1.SQ2 转换结果存储到 ADCxSR2H 和 ADCxSR2L 中，ADCxSQ1.SQ3 转换结果存储到 ADCxSR3H 和 ADCxSR3L 中，ADCxIF.SCHINT 硬件置位。软件需要响应两个 ADC 的 SCHINT 中断，分别读取两个 ADC 的转换结果。

特殊交替触发模式下，以下寄存器配置要求：

- 配置寄存器 ADC0CON.CONT 和 ADC1CON.CONT 为相同值
- 配置寄存器 ADC0CON.SCAN 和 ADC1CON.SCAN 为相同值

Figure 183 给出了 CONT 为 0，SCAN 为 1，两个 ADC 的 SQL 均为 0x03 的场景，即单次转换，扫描模式下的特殊交替触发模式。

17.13.6 普通同步+特殊同步模式

配置寄存器 ADCDMC.DMS 为 101，ADC0 和 ADC1 工作在普通同步+特殊同步模式。

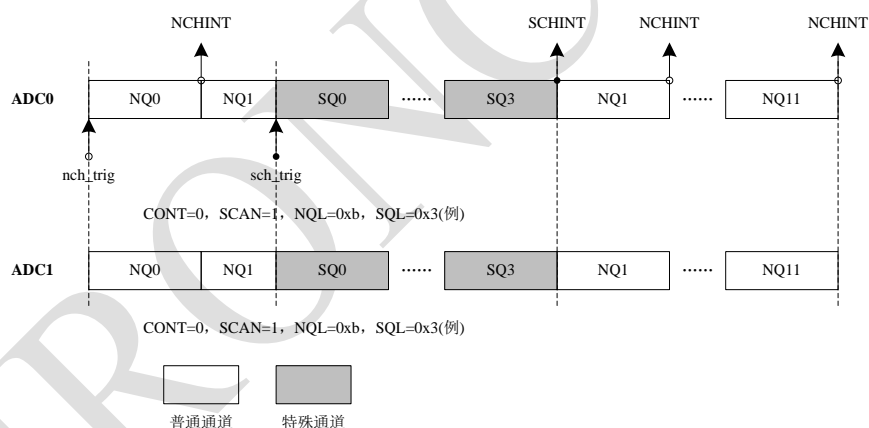


Figure 184 双 ADC 普通同步+特殊同步模式

该模式下，普通通道序列和特殊通道序列均被打开，ADC0 的普通通道外部触发事件启动两个 ADC 普通通道同时工作，ADC0 的特殊通道外部触发事件启动两个 ADC 特殊通道同时工作。特殊通道触发的优先级高于普通通道，特殊通道序列的外部触发可将正在处理的普通通道序列打断，特殊通道序列转换完成后，再继续处理普通通道序列未完成部分。

每个普通通道转换完成后，ADCx 的转换结果存储到 ADCxNRH 和 ADCxNRL 中，ADCxIF.NCHINT 硬件置位。软件只需要响应一个 ADC 的 NCHINT 中断，同时读取两个 ADC 的转换结果。

整个特殊通道序列转换完成后，ADCxSQ0.SQ0 转换结果存储到 ADCxSR0H 和 ADCxSR0L 中，ADCxSQ0.SQ1 转换结果存储到 ADCxSR1H 和 ADCxSR1L 中，ADCxSQ1.SQ2

转换结果存储到 ADCxSR2H 和 ADCxSR2L 中, ADCxSQ1.SQ3 转换结果存储到 ADCxSR3H 和 ADCxSR3L 中, ADCxIF.SCHINT 硬件置位。软件只需要响应一个 ADC 的 SCHINT 中断, 同时读取两个 ADC 的转换结果。

普通同步+特殊同步模式下, 以下寄存器配置要求:

- 配置寄存器 ADC0CON.CONT 和 ADC1CON.CONT 为相同值
- 配置寄存器 ADC0CON.SCAN 和 ADC1CON.SCAN 为相同值
- 配置寄存器 ADC0QLC.NQL 和 ADC1QLC.NQL 为相同值
- 配置寄存器 ADC0QLC.SQL 和 ADC1QLC.SQL 为相同值

Figure 184 给出了 CONT 为 0, SCAN 为 1, 两个 ADC 的 NQL 均为 0x0b, SQL 均为 0x03 的场景, 即单次转换, 扫描模式下的普通同步+特殊同步模式。

17.13.7 普通同步+特殊交替触发模式

配置寄存器 ADCDMC.DMS 为 110, ADC0 和 ADC1 工作在普通同步+特殊交替触发模式。

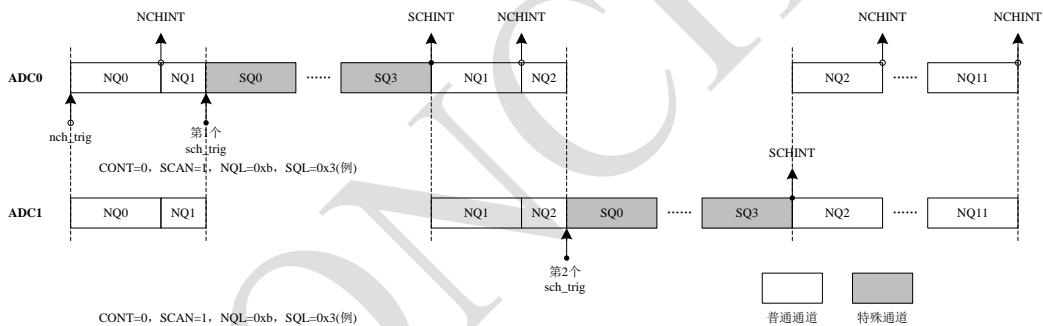


Figure 185 双 ADC 普通同步+特殊交替触发模式

该模式下, 普通通道序列和特殊通道序列均被打开, ADC0 的普通通道外部触发事件启动两个 ADC 普通通道同时工作, ADC0 的特殊通道外部触发事件启动两个 ADC 特殊通道工作: 第 1 个特殊通道触发启动 ADC0 特殊通道工作, 第 2 个特殊通道触发启动 ADC1 特殊通道工作.....第奇数个特殊通道触发启动 ADC0 特殊通道工作, 第偶数个特殊通道触发启动 ADC1 特殊通道工作。特殊通道触发的优先级高于普通通道, 特殊通道序列的外部触发可将正在处理的普通通道序列打断, 特殊通道序列转换完成后, 再继续处理普通通道序列未完成部分。

每个普通通道转换完成后, ADCx 的转换结果存储到 ADCxNRH 和 ADCxNRL 中, ADCxIF.NCHINT 硬件置位。软件只需要响应一个 ADC 的 NCHINT 中断, 同时读取两个 ADC 的转换结果。

ADCx 的整个特殊通道序列转换完成后, ADCxSQ0.SQ0 转换结果存储到 ADCxSR0H

和 ADCxSR0L 中，ADCxSQ0.SQ1 转换结果存储到 ADCxSR1H 和 ADCxSR1L 中，ADCxSQ1.SQ2 转换结果存储到 ADCxSR2H 和 ADCxSR2L 中，ADCxSQ1.SQ3 转换结果存储到 ADCxSR3H 和 ADCxSR3L 中，ADCxIF.SCHINT 硬件置位。软件需要响应两个 ADC 的 SCHINT 中断，分别读取两个 ADC 的转换结果。

普通同步+特殊交替触发模式下，以下寄存器配置要求：

- 配置寄存器 ADC0CON.CONT 和 ADC1CON.CONT 为相同值
- 配置寄存器 ADC0CON.SCAN 和 ADC1CON.SCAN 为相同值
- 配置寄存器 ADC0QLC.NQL 和 ADC1QLC.NQL 为相同值

Figure 185 给出了 CONT 为 0,SCAN 为 1,两个 ADC 的 NQL 均为 0x0b,SQL 均为 0x03 的场景，即单次转换，扫描模式下的普通同步+特殊交替触发模式。

17.13.8 普通交叉+特殊同步模式

配置寄存器 ADCDMC.DMS 为 111，ADC0 和 ADC1 工作在普通交叉+特殊同步模式。

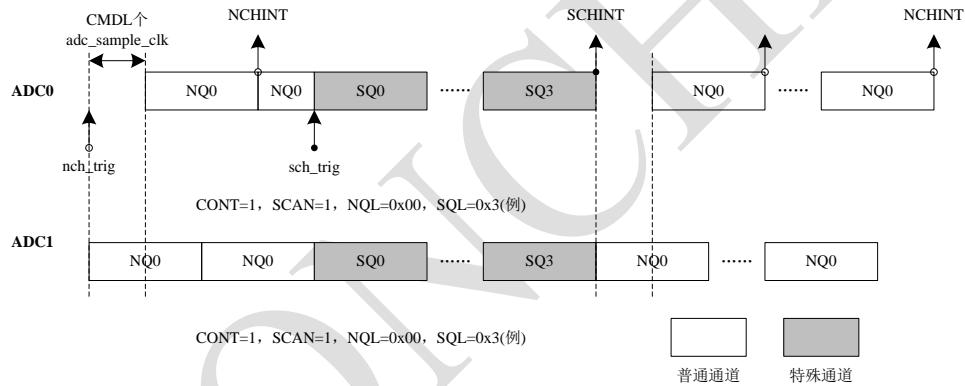


Figure 186 双 ADC 普通交叉+特殊同步模式

该模式下，普通通道序列和特殊通道序列均被打开，ADC0 的普通通道外部触发事件首先启动 ADC1 普通通道转换，CMDL(通过寄存器 ADCDMC.CMDL 配置)个 adc_sample_clk 时钟之后启动 ADC0 普通通道转换。ADC0 的特殊通道外部触发事件启动两个 ADC 特殊通道同时工作。特殊通道触发的优先级高于普通通道，特殊通道序列的外部触发可将正在处理的普通通道打断，特殊通道序列转换完成后，再继续处理普通通道序列未完成部分。

每个普通通道转换完成后，ADCx 的转换结果存储到 ADCxNRH 和 ADCxNRL 中，ADCxIF.NCHINT 硬件置位。软件只需要响应 ADC0 的 NCHINT 中断，同时读取两个 ADC 的转换结果。

整个特殊通道序列转换完成后，ADCxSQ0.SQ0 转换结果存储到 ADCxSR0H 和 ADCxSR0L 中，ADCxSQ0.SQ1 转换结果存储到 ADCxSR1H 和 ADCxSR1L 中，ADCxSQ1.SQ2 转换结果存储到 ADCxSR2H 和 ADCxSR2L 中，ADCxSQ1.SQ3 转换结果存储到 ADCxSR3H 和 ADCxSR3L 中，ADCxIF.SCHINT 硬件置位。软件只需要响应一个 ADC 的 SCHINT 中断，

同时读取两个 ADC 的转换结果。

普通交叉+特殊同步模式下，以下寄存器配置要求：

- 配置寄存器 ADC0CON.CONT 和 ADC1CON.CONT 为相同值
- 配置寄存器 ADC0CON.SCAN 和 ADC1CON.SCAN 为相同值
- 配置寄存器 ADC0QLC.NQL 和 ADC1QLC.NQL 为相同值
- 配置寄存器 ADC0QLC.SQL 和 ADC1QLC.SQL 为相同值

Figure 186 给出了 CONT 为 1,SCAN 为 1,两个 ADC 的 NQL 均为 0x00,SQL 均为 0x03 的场景，即连续转换，扫描模式下的普通交叉+特殊同步模式。

17.14 中断处理

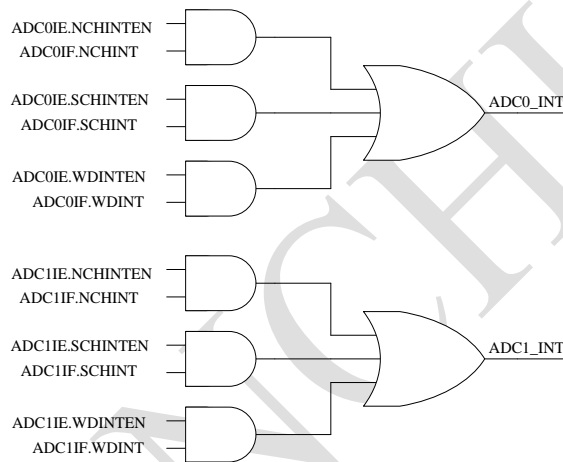


Figure 187 ADC 中断

ADC0_INT 和 ADC1_INT 各有 3 个中断源：普通通道转换完成中断（NCHINT）、特殊通道转换完成中断（SCHINT）和模拟看门狗中断（WDINT）。

- 普通通道转换完成中断：普通通道转换结束后，寄存器 ADCxIF.NCHINT 置位，产生普通通道转换完成中断；
- 特殊通道转换完成中断：特殊通道转换结束后，寄存器 ADCxIF.SCHINT 置位，产生特殊通道转换完成中断；
- 模拟看门狗中断：当指定的通道转换结果超出模拟看门狗的门限电压时，即小于 ADCxLT（ADCxLTH 和 ADCxLTL 合并），或者大于 ADCxHT（ADCxHTH 和 ADCxHTL 合并），寄存器 ADCxIF.WDINT 置位，产生模拟看门狗中断。

ADC0 中断配置及处理：

- 配置 ADC0IE 寄存器使能相关中断源；
- 配置 IEN1.ADC0INT 寄存器为 1，使能 ADC0 中断；
- 当产生 ADC0 中断后，软件需要回读 ADC0IF 寄存器，来判断 ADC0 中断源，根据不同中断源，向 ADC0IF 寄存器相应位写 0 清除，并做相应处理。

ADC1 中断配置及处理:

- 配置 ADC1IE 寄存器使能相关中断源;
- 配置 IEN0.ADC1INT 寄存器为 1, 使能 ADC1 中断;
- 当产生 ADC1 中断后, 软件需要回读 ADC1IF 寄存器, 来判断 ADC1 中断源, 根据不同中断源, 向 ADC1IF 寄存器相应位写 0 清除, 并做相应处理。

17.15 ADC 寄存器

17.15.1 JMT51 访问 ADC 寄存器

ADC 模块有 70 个寄存器, 对应的 SFRPAGE 为 1, 具体说明如下表。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(ADC0NRH, ADC0SR0H, ADC0SR1H, ADC0SR2H, ADC0SR3H, ADC1NRH, ADC1SR0H, ADC1SR1H, ADC1SR2H 和 ADC1SR3H) 和低字节寄存器 (ADC0NRL, ADC0SR0L, ADC0SR1L, ADC0SR2L, ADC0SR3L, ADC1NRL, ADC1SR0L, ADC1SR1L, ADC1SR2L 和 ADC1SR3L) 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 319 JMT51 访问 ADC 寄存器 (SFRPAGE=0x01)

地址	寄存器名	类型	复位值	功能描述
0xAA	ADCDMC	W/R	0x00	双 ADC 协同工作模式控制寄存器
0xAB	ADCJC	W/R	0x00	ADC 校准控制寄存器
0x88	ADC0CON	W/R	0x00	ADC0 控制寄存器
0xAC	ADC0PGAC	W/R	0x00	PGA0 控制寄存器
0x89	ADC0IE	W/R	0x00	ADC0 中断使能寄存器
0x8A	ADC0IF	W/R	0x00	ADC0 中断标志寄存器
0x8B	ADC0WDC	W/R	0x00	ADC0 看门狗控制寄存器
0x8C	ADC0HTL	W/R	0x00	ADC0 看门狗高阈值寄存器低 8 位
0x8D	ADC0HTH	W/R	0x00	ADC0 看门狗高阈值寄存器高 2 位
0x8E	ADC0LTL	W/R	0x00	ADC0 看门狗低阈值寄存器低 8 位
0x8F	ADC0LTH	W/R	0x00	ADC0 看门狗低阈值寄存器高 2 位
0x91	ADC0QLC	W/R	0x00	ADC0 序列长度寄存器
0x92	ADC0NQTC	W/R	0x00	ADC0 普通序列外部触发控制寄存器
0x93	ADC0NQ0	W/R	0x00	ADC0 普通序列寄存器 0
0x94	ADC0NQ1	W/R	0x00	ADC0 普通序列寄存器 1
0x95	ADC0NQ2	W/R	0x00	ADC0 普通序列寄存器 2

地址	寄存器名	类型	复位值	功能描述
0x96	ADC0NQ3	W/R	0x00	ADC0 普通序列寄存器 3
0x97	ADC0NQ4	W/R	0x00	ADC0 普通序列寄存器 4
0x99	ADC0NQ5	W/R	0x00	ADC0 普通序列寄存器 5
0x9A	ADC0NRL	R	0x00	ADC0 普通序列结果寄存器低 8 位
0x9B	ADC0NRH	R	0x00	ADC0 普通序列结果寄存器高 8 位
0x9C	ADC0SQTC	W/R	0x00	ADC0 特殊序列外部触发控制寄存器
0x9D	ADC0SQ0	W/R	0x00	ADC0 特殊序列寄存器 0
0x9E	ADC0SQ1	W/R	0x00	ADC0 特殊序列寄存器 1
0x9F	ADC0SR0L	R	0x00	ADC0 特殊序列第 1 个转换通道结果寄存器低 8 位
0xA1	ADC0SR0H	R	0x00	ADC0 特殊序列第 1 个转换通道结果寄存器高 8 位
0xA2	ADC0SR1L	R	0x00	ADC0 特殊序列第 2 个转换通道结果寄存器低 8 位
0xA3	ADC0SR1H	R	0x00	ADC0 特殊序列第 2 个转换通道结果寄存器高 8 位
0xA4	ADC0SR2L	R	0x00	ADC0 特殊序列第 3 个转换通道结果寄存器低 8 位
0xA5	ADC0SR2H	R	0x00	ADC0 特殊序列第 3 个转换通道结果寄存器高 8 位
0xA6	ADC0SR3L	R	0x00	ADC0 特殊序列第 4 个转换通道结果寄存器低 8 位
0xA7	ADC0SR3H	R	0x00	ADC0 特殊序列第 4 个转换通道结果寄存器高 8 位
0xA9	ADC0DF	W/R	0x00	ADC0 数据格式寄存器
0xAD	ADC0ETC	W/R	0x0f	ADC0 外部触发边沿控制寄存器
0xAE	ADC0NQTD	W/R	0x00	ADC0 普通通道触发延时寄存器
0xAF	ADC0SQTD	W/R	0x00	ADC0 特殊通道触发延时寄存器
0x98	ADC1CON	W/R	0x00	ADC1 控制寄存器
0xE4	ADC1PGAC	W/R	0x00	PGA1 控制寄存器
0xC1	ADC1IE	W/R	0x00	ADC1 中断使能寄存器
0xC2	ADC1IF	W/R	0x00	ADC1 中断标志寄存器
0xC3	ADC1WDC	W/R	0x00	ADC1 看门狗控制寄存器
0xC4	ADC1HTL	W/R	0x00	ADC1 看门狗高阈值寄存器低 8 位
0xC5	ADC1HTH	W/R	0x00	ADC1 看门狗高阈值寄存器高 2 位
0xC6	ADC1LTL	W/R	0x00	ADC1 看门狗低阈值寄存器低 8 位
0xC7	ADC1LTH	W/R	0x00	ADC1 看门狗低阈值寄存器高 2 位
0xC9	ADC1QLC	W/R	0x00	ADC1 序列长度寄存器
0xCA	ADC1NQTC	W/R	0x00	ADC1 普通序列外部触发控制寄存器

地址	寄存器名	类型	复位值	功能描述
0xCB	ADC1NQ0	W/R	0x00	ADC1 普通序列寄存器 0
0xCC	ADC1NQ1	W/R	0x00	ADC1 普通序列寄存器 1
0xCD	ADC1NQ2	W/R	0x00	ADC1 普通序列寄存器 2
0xCE	ADC1NQ3	W/R	0x00	ADC1 普通序列寄存器 3
0xCF	ADC1NQ4	W/R	0x00	ADC1 普通序列寄存器 4
0xD1	ADC1NQ5	W/R	0x00	ADC1 普通序列寄存器 5
0xD2	ADC1NRL	R	0x00	ADC1 普通序列结果寄存器低 8 位
0xD3	ADC1NRH	R	0x00	ADC1 普通序列结果寄存器高 8 位
0xD4	ADC1SQTC	W/R	0x00	ADC1 特殊序列外部触发控制寄存器
0xD5	ADC1SQ0	W/R	0x00	ADC1 特殊序列寄存器 0
0xD6	ADC1SQ1	W/R	0x00	ADC1 特殊序列寄存器 1
0xD7	ADC1SR0L	R	0x00	ADC1 特殊序列第 1 个转换通道结果寄存器低 8 位
0xD9	ADC1SR0H	R	0x00	ADC1 特殊序列第 1 个转换通道结果寄存器高 8 位
0xDA	ADC1SR1L	R	0x00	ADC1 特殊序列第 2 个转换通道结果寄存器低 8 位
0xDB	ADC1SR1H	R	0x00	ADC1 特殊序列第 2 个转换通道结果寄存器高 8 位
0xDC	ADC1SR2L	R	0x00	ADC1 特殊序列第 3 个转换通道结果寄存器低 8 位
0xDD	ADC1SR2H	R	0x00	ADC1 特殊序列第 3 个转换通道结果寄存器高 8 位
0xDE	ADC1SR3L	R	0x00	ADC1 特殊序列第 4 个转换通道结果寄存器低 8 位
0xDF	ADC1SR3H	R	0x00	ADC1 特殊序列第 4 个转换通道结果寄存器高 8 位
0xE1	ADC1DF	W/R	0x00	ADC1 数据格式寄存器
0xE5	ADC1ETC	W/R	0x0f	ADC1 外部触发边沿控制寄存器
0xE6	ADC1NQTD	W/R	0x00	ADC1 普通通道触发延时寄存器
0xE7	ADC1SQTD	W/R	0x00	ADC1 特殊通道触发延时寄存器

17.15.1.1 双 ADC 协同工作模式控制寄存器(ADCDMC) [0xAA]

Table 320 双 ADC 协同工作模式控制寄存器(ADCDMC)

参数名	比特位	属性	复位值	描述
DMS	2-0	W/R	0	双 ADC 协同工作模式选择： 000: 独立模式 001: 普通同步模式 010: 特殊同步模式 011: 普通交叉模式

参数名	比特位	属性	复位值	描述
				100: 特殊交替触发模式 101: 普通同步+特殊同步模式 110: 普通同步+特殊交替触发模式 111: 普通交叉+特殊同步模式
CMDL	7-3	W/R	0x03	双 ADC 协同工作模式中, 普通交叉触发模式中 ADC0 启动延时: 0x00: 不延时 0x01: 延时 1 个 adc_sample_clk 0x02: 延时 2 个 adc_sample_clk 0x0a: 延时 10 个 adc_sample_clk 0x0b~0x1f: 延时 11 个 adc_sample_clk

17.15.1.2 ADC 参考电压控制寄存器(ADCJC) [0xAB]

Table 321 ADC 参考电压控制寄存器(ADCJC)

参数名	比特位	属性	复位值	描述
PDVREF	0	W/R	0	内部参考电压的低功耗控制: 0: 正常工作模式 1: 低功耗模式 注: 该内部参考电压同时用于 ADCx、PGAx 和模拟比较器, 需要在以上模块均不工作时才能配置内部参考电压进入低功耗模式
ADREFTRIM	1	W/R	0	内部参考电压校准使能: 0: 不使能 1: 使能 注: 仅用于出厂前校准
ECVREF_SEL	3-2	W/R	0	ADCx 和 PGAx 的参考电压选择: 00: 内部 2.5V 参考电压 01: 模拟供电 VDDA 1x: 芯片外部输入参考电压 (VREF 管脚)
Reserved	7-4	R	0	保留

17.15.1.3 ADC0 控制寄存器(ADC0CON) [0x88]

Table 322 ADC0 控制寄存器(ADC0CON)

参数名	比特位	属性	复位值	描述
ADCEN	0	W/R	0	ADC0 开启: 0: 关闭 ADC0 1: 开启 ADC0
NQST	1	W	0	ADC0 普通序列转换启动: 0: 复位状态 1: 启动转换
NQTSW	2	W	0	ADC0 普通通道序列外部触发软启动: 0: 复位状态 1: 启动普通通道序列转换 注: 该启动仅在 NQTSEL 选中 NQTSW 为触发事件, 且 NQTEN 有效时生效
SQTSW	3	W	0	ADC0 特殊通道序列外部触发软启动: 0: 复位状态 1: 启动特殊通道序列转换 注: 该启动仅在 SQTSEL 选中 SQTSW 为触发事件, 且 SQTEN 有效时生效
CONT	4	W/R	0	ADC0 连续模式: 0: 单次模式 1: 连续模式
SCAN	5	W/R	0	ADC0 扫描模式: 0: 关闭扫描模式 1: 打开扫描模式
SQAUTO	6	W/R	0	ADC0 特殊通道序列触发方式管理: 0: 外部触发 1: 自动触发
ADCPD	7	W/R	0	ADC0 的低功耗控制: 0: 正常工作模式 1: 低功耗模式

17.15.1.4 PGA0 控制寄存器(ADC0PGAC) [0xAC]

Table 323 PGA0 控制寄存器(ADC0PGAC)

参数名	比特位	属性	复位值	描述
PGAMUL	2-0	W/R	010	PGA0 运放倍数选择: 000: 1/4 倍 001: 1/2 倍 010: 1 倍 011: 2 倍 100: 4 倍 101: 8 倍 110: 16 倍 111: 32 倍
PGAPOL	3	W/R	0	PGA0 极性寄存器: 0: 反相 1: 同相
PGATRIM	4	W/R	0	PGA0 校准使能: 0: 不使能 1: 使能 注: 仅用于出厂前校准
PGAPD	5	W/R	0	PGA0 的低功耗控制: 0: 正常工作模式 1: 低功耗模式
Reserved	7-6	R	0	保留

17.15.1.5 ADC0 中断使能寄存器(ADC0IE) [0x89]

Table 324 ADC0 中断使能寄存器(ADC0IE)

参数名	比特位	属性	复位值	描述
NCHINTEN	0	W/R	0	ADC0 普通通道转换完成中断使能: 0: 不使能 1: 使能
SCHINTEN	1	W/R	0	ADC0 特殊通道转换完成中断使能: 0: 不使能 1: 使能
WDINTEN	2	W/R	0	ADC0 模拟看门狗中断使能: 0: 不使能

参数名	比特位	属性	复位值	描述
				1: 使能
Reserved	7-3	R	0	保留

17.15.1.6 ADC0 中断标志寄存器(ADC0IF) [0x8A]

Table 325 ADC0 中断标志寄存器(ADC0IF)

参数名	比特位	属性	复位值	描述
NCHINT	0	W/R	0	ADC0 普通通道转换完成中断标志位： <ul style="list-style-type: none"> ● 当有普通通道转换完成中断申请时，该位置为 1 ● 不支持软中断，在中断处理程序中需要软件写 0 来清除
SCHINT	1	W/R	0	ADC0 特殊通道转换完成中断标志位： <ul style="list-style-type: none"> ● 当有特殊通道转换完成中断申请时，该位置为 1 ● 不支持软中断，在中断处理程序中需要软件写 0 来清除
WDINT	2	W/R	0	ADC0 模拟看门狗中断标志位： <ul style="list-style-type: none"> ● 当有模拟看门狗中断申请时，该位置为 1 ● 不支持软中断，在中断处理程序中需要软件写 0 来清除
Reserved	7-3	R	0	保留

17.15.1.7 ADC0 看门狗控制寄存器(ADC0WDC) [0x8B]

Table 326 ADC0 看门狗控制寄存器(ADC0WDC)

参数名	比特位	属性	复位值	描述
WDCH	3-0	W/R	0	当 SCHEN 为 1 时(单一通道模式), ADC0 看门狗通道选择: 0000: AN0 0001: AN1 0010: AN2 1010: AN10

参数名	比特位	属性	复位值	描述
				1011: AN11 其他: 保留
NWDEN	4	W/R	0	ADC0 普通通道序列模拟看门狗使能: 0: 不使能 1: 使能
SWDEN	5	W/R	0	ADC0 特殊通道序列模拟看门狗使能: 0: 不使能 1: 使能
SCHEN	6	W/R	0	ADC0 看门狗单一通道模式: 0: 所有通道上使能看门狗 1: WDCH 指定的单一通道上使能看门狗
Reserved	7	R	0	保留

17.15.1.8 ADC0 看门狗高阈值寄存器低 8 位(ADC0HTL) [0x8C]

Table 327 ADC0 看门狗高阈值寄存器低 8 位(ADC0HTL)

参数名	比特位	属性	复位值	描述
ADCHTL	7-0	W/R	0	ADC0 看门狗高阈值寄存器低 8 位

17.15.1.9 ADC0 看门狗高阈值寄存器高 2 位(ADC0HTH) [0x8D]

Table 328 ADC0 看门狗高阈值寄存器高 2 位(ADC0HTH)

参数名	比特位	属性	复位值	描述
ADCHTH	1-0	W/R	0	ADC0 看门狗高阈值寄存器高 2 位
Reserved	7-2	R	0	保留

17.15.1.10 ADC0 看门狗低阈值寄存器低 8 位(ADC0LTL) [0x8E]

Table 329 ADC0 看门狗低阈值寄存器低 8 位(ADC0LTL)

参数名	比特位	属性	复位值	描述
ADCLTL	7-0	W/R	0	ADC0 看门狗低阈值寄存器低 8 位

17.15.1.11 ADC0 看门狗低阈值寄存器高 2 位(ADC0LTH) [0x8F]

Table 330 ADC0 看门狗低阈值寄存器高 2 位(ADC0LTH)

参数名	比特位	属性	复位值	描述
ADCLTH	1-0	W/R	0	ADC0 看门狗低阈值寄存器高 2 位
Reserved	7-2	R	0	保留

17.15.1.12 ADC0 序列长度控制寄存器(ADC0QLC) [0x91]

Table 331 ADC0 序列长度控制寄存器(ADC0QLC)

参数名	比特位	属性	复位值	描述
NQL	3-0	W/R	0	ADC0 普通通道序列长度： 0000：第 1 个通道转换 0001：第 1~2 个通道转换 0010：第 1~3 个通道转换 1010：第 1~11 个通道转换 1011：第 1~12 个通道转换
SQL	5-4	W/R	0	ADC0 特殊通道序列长度： 00：第 1 个通道转换 01：第 1~2 个通道转换 10：第 1~3 个通道转换 11：第 1~4 个通道转换
Reserved	7-6	R	0	保留

17.15.1.13 ADC0 普通序列外部触发控制寄存器(ADC0NQTC) [0x92]

Table 332 ADC0 普通序列外部触发控制寄存器(ADC0NQTC)

参数名	比特位	属性	复位值	描述
NQTSEL	5-0	W/R	0	ADC0 普通通道序列外部触发事件选择： 000000：TIM0_OC0 000001：TIM0_OC1 000010：TIM0_OC2 000011：TIM0_OC3 000100：TIM1_OC0 000101：TIM1_OC1 000110：TIM1_OC2 000111：TIM1_OC3 001100：TIM0_TRGO 001101：TIM1_TRGO 010000：INT0_0 010001：INT0_1 010010：INT0_2

参数名	比特位	属性	复位值	描述
				010011: INT0_3 010100: INT0_4 010101: INT0_5 010110: INT0_6 010111: INT0_7 011000: INT1_0 011001: INT1_1 011010: INT1_2 011011: INT1_3 011100: INT1_4 011101: INT1_5 011110: INT1_6 011111: INT1_7 100000: NQTSW 其他: 保留
NQTEN	6	W/R	0	ADC0 普通通道序列外部触发使能: 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
Reserved	7	R	0	保留

17.15.1.14 ADC0 普通序列寄存器 0(ADC0NQ0) [0x93]

Table 333 ADC0 普通序列寄存器 0(ADC0NQ0)

参数名	比特位	属性	复位值	描述
NQ0	3-0	W/R	0	ADC0 普通序列第 1 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留
NQ1	7-4	W/R	0	ADC0 普通序列第 2 个转换通道源选择: 0000: AN0

参数名	比特位	属性	复位值	描述
				0001: AN1 1011: AN11 其他: 保留

17.15.1.15 ADC0 普通序列寄存器 1(ADC0NQ1) [0x94]

Table 334 ADC0 普通序列寄存器 1(ADC0NQ1)

参数名	比特位	属性	复位值	描述
NQ2	3-0	W/R	0	ADC0 普通序列第 3 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留
NQ3	7-4	W/R	0	ADC0 普通序列第 4 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.16 ADC0 普通序列寄存器 2(ADC0NQ2) [0x95]

Table 335 ADC0 普通序列寄存器 2(ADC0NQ2)

参数名	比特位	属性	复位值	描述
NQ4	3-0	W/R	0	ADC0 普通序列第 5 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留
NQ5	7-4	W/R	0	ADC0 普通序列第 6 个转换通道源选择: 0000: AN0 0001: AN1

参数名	比特位	属性	复位值	描述
			 1011: AN11 其他: 保留

17.15.1.17 ADC0 普通序列寄存器 3(ADC0NQ3) [0x96]

Table 336 ADC0 普通序列寄存器 3(ADC0NQ3)

参数名	比特位	属性	复位值	描述
NQ6	3-0	W/R	0	ADC0 普通序列第 7 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留
NQ7	7-4	W/R	0	ADC0 普通序列第 8 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.18 ADC0 普通序列寄存器 4(ADC0NQ4) [0x97]

Table 337 ADC0 普通序列寄存器 4(ADC0NQ4)

参数名	比特位	属性	复位值	描述
NQ8	3-0	W/R	0	ADC0 普通序列第 9 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留
NQ9	7-4	W/R	0	ADC0 普通序列第 10 个转换通道源选择: 0000: AN0 0001: AN1

参数名	比特位	属性	复位值	描述
				1011: AN11 其他: 保留

17.15.1.19 ADC0 普通序列寄存器 5(ADC0NQ5) [0x99]

Table 338 ADC0 普通序列寄存器 5(ADC0NQ5)

参数名	比特位	属性	复位值	描述
NQ10	3-0	W/R	0	ADC0 普通序列第 11 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留
NQ11	7-4	W/R	0	ADC0 普通序列第 12 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.20 ADC0 普通序列结果寄存器低 8 位(ADC0NRL) [0x9A]

Table 339 ADC0 普通序列结果寄存器低 8 位(ADC0NRL)

参数名	比特位	属性	复位值	描述
NRL	7-0	R	0	ADC0 普通序列结果低 8 位

17.15.1.21 ADC0 普通序列结果寄存器高 8 位(ADC0NRH) [0x9B]

Table 340 ADC0 普通序列结果寄存器高 8 位(ADC0NRH)

参数名	比特位	属性	复位值	描述
NRH	7-0	R	0	ADC0 普通序列结果高 8 位

17.15.1.22 ADC0 特殊序列外部触发控制寄存器(ADC0SQTC) [0x9C]

Table 341 ADC0 特殊序列外部触发控制寄存器(ADC0SQTC)

参数名	比特位	属性	复位值	描述
SQTSEL	5-0	W/R	0	ADC0 特殊通道序列外部触发事件选择: 000000: TIM0_OC0

参数名	比特位	属性	复位值	描述
				000001: TIM0_OC1 000010: TIM0_OC2 000011: TIM0_OC3 000100: TIM1_OC0 000101: TIM1_OC1 000110: TIM1_OC2 000111: TIM1_OC3 001100: TIM0_TRGO 001101: TIM1_TRGO 010000: INT0_0 010001: INT0_1 010010: INT0_2 010011: INT0_3 010100: INT0_4 010101: INT0_5 010110: INT0_6 010111: INT0_7 011000: INT1_0 011001: INT1_1 011010: INT1_2 011011: INT1_3 011100: INT1_4 011101: INT1_5 011110: INT1_6 011111: INT1_7 100000: SQTSW 其他: 保留
SQTEN	6	W/R	0	ADC0 特殊通道序列外部触发使能: 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

参数名	比特位	属性	复位值	描述
Reserved	7	R	0	保留

17.15.1.23 ADC0 特殊序列寄存器 0(ADC0SQ0) [0x9D]

Table 342 ADC0 特殊序列寄存器 0(ADC0SQ0)

参数名	比特位	属性	复位值	描述
SQ0	3-0	W/R	0	ADC0 特殊序列第 1 个转换通道源选择： 0000: AN0 0001: AN1 1011: AN11 其他: 保留
SQ1	7-4	W/R	0	ADC0 特殊序列第 2 个转换通道源选择： 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.24 ADC0 特殊序列寄存器 1(ADC0SQ1) [0x9E]

Table 343 ADC0 特殊序列寄存器 1(ADC0SQ1)

参数名	比特位	属性	复位值	描述
SQ2	3-0	W/R	0	ADC0 特殊序列第 3 个转换通道源选择： 0000: AN0 0001: AN1 1011: AN11 其他: 保留
SQ3	7-4	W/R	0	ADC0 特殊序列第 4 个转换通道源选择： 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.25 ADC0 特殊序列第 1 个转换通道结果寄存器低 8 位(ADC0SR0L) [0x9F]

Table 344 ADC0 特殊序列第 1 个转换通道结果寄存器低 8 位(ADC0SR0L)

参数名	比特位	属性	复位值	描述
SR0L	7-0	R	0	ADC0 特殊序列第 1 个转换通道结果寄存器低 8 位

17.15.1.26 ADC0 特殊序列第 1 个转换通道结果寄存器高 8 位(ADC0SR0H) [0xA1]

Table 345 ADC0 特殊序列第 1 个转换通道结果寄存器高 8 位(ADC0SR0H)

参数名	比特位	属性	复位值	描述
SR0H	7-0	R	0	ADC0 特殊序列第 1 个转换通道结果寄存器高 8 位

17.15.1.27 ADC0 特殊序列第 2 个转换通道结果寄存器低 8 位(ADC0SR1L) [0xA2]

Table 346 ADC0 特殊序列第 2 个转换通道结果寄存器低 8 位(ADC0SR1L)

参数名	比特位	属性	复位值	描述
SR1L	7-0	R	0	ADC0 特殊序列第 2 个转换通道结果寄存器低 8 位

17.15.1.28 ADC0 特殊序列第 2 个转换通道结果寄存器高 8 位(ADC0SR1H) [0xA3]

Table 347 ADC0 特殊序列第 2 个转换通道结果寄存器高 8 位(ADC0SR1H)

参数名	比特位	属性	复位值	描述
SR1H	7-0	R	0	ADC0 特殊序列第 2 个转换通道结果寄存器高 8 位

17.15.1.29 ADC0 特殊序列第 3 个转换通道结果寄存器低 8 位(ADC0SR2L) [0xA4]

Table 348 ADC0 特殊序列第 3 个转换通道结果寄存器低 8 位(ADC0SR2L)

参数名	比特位	属性	复位值	描述
SR2L	7-0	R	0	ADC0 特殊序列第 3 个转换通道结果寄存器低 8 位

17.15.1.30 ADC0 特殊序列第 3 个转换通道结果寄存器高 8 位(ADC0SR2H) [0xA5]

Table 349 ADC0 特殊序列第 3 个转换通道结果寄存器高 8 位(ADC0SR2H)

参数名	比特位	属性	复位值	描述
SR2H	7-0	R	0	ADC0 特殊序列第 3 个转换通道结果寄存器高 8 位

17.15.1.31 ADC0 特殊序列第 4 个转换通道结果寄存器低 8 位(ADC0SR3L) [0xA6]

Table 350 ADC0 特殊序列第 4 个转换通道结果寄存器低 8 位(ADC0SR3L)

参数名	比特位	属性	复位值	描述
SR3L	7-0	R	0	ADC0 特殊序列第 4 个转换通道结果寄存器低 8 位

17.15.1.32 ADC0 特殊序列第 4 个转换通道结果寄存器高 8 位(ADC0SR3H) [0xA7]

Table 351 ADC0 特殊序列第 4 个转换通道结果寄存器高 8 位(ADC0SR3H)

参数名	比特位	属性	复位值	描述
SR3H	7-0	R	0	ADC0 特殊序列第 4 个转换通道结果寄存器高 8 位

17.15.1.33 ADC0 结果数据格式寄存器(ADC0DF) [0xA9]

Table 352 ADC0 结果数据格式寄存器(ADC0DF)

参数名	比特位	属性	复位值	描述
ADC0DF	1-0	W/R	0	ADC0 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
Reserved	7-2	R	0	保留

17.15.1.34 ADC0 外部触发边沿控制寄存器(ADC0ETC) [0xAD]

Table 353 ADC0 外部触发边沿控制寄存器(ADC0ETC)

参数名	比特位	属性	复位值	描述
NQTPOSEN	0	W/R	1	ADC0 普通通道外部触发信号上升沿触发使能： 0: 不使能 1: 使能
NQTNEGEN	1	W/R	1	ADC0 普通通道外部触发信号下降沿触发使能： 0: 不使能 1: 使能
SQTPOSEN	2	W/R	1	ADC0 特殊通道外部触发信号上升沿触发使能： 0: 不使能 1: 使能
SQTNEGEN	3	W/R	1	ADC0 特殊通道外部触发信号下降沿触发使能： 0: 不使能

参数名	比特位	属性	复位值	描述
				1: 使能
Reserved	7-4	R	0	保留

17.15.1.35 ADC0 普通通道触发延时寄存器(ADC0NQTD) [0xAE]

Table 354 ADC0 普通通道触发延时寄存器(ADC0NQTD)

参数名	比特位	属性	复位值	描述
NQTD	7-0	W/R	0	ADC0 普通通道触发延时: $NQTD \times T_{adc_clk}$

17.15.1.36 ADC0 特殊通道触发延时寄存器(ADC0SQTD) [0xAF]

Table 355 ADC0 特殊通道触发延时寄存器(ADC0SQTD)

参数名	比特位	属性	复位值	描述
SQTD	7-0	W/R	0	ADC0 特殊通道触发延时: $SQTD \times T_{adc_clk}$

17.15.1.37 ADC1 控制寄存器(ADC1CON) [0x98]

Table 356 ADC1 控制寄存器(ADC1CON)

参数名	比特位	属性	复位值	描述
ADCEN	0	W/R	0	ADC1 开启: 0: 关闭 ADC1 1: 开启 ADC1
NQST	1	W	0	ADC1 普通序列转换启动: 0: 复位状态 1: 启动转换
NQTSW	2	W	0	ADC1 普通通道序列外部触发软启动: 0: 复位状态 1: 启动普通通道序列转换 注: 该启动仅在 NQTSEL 选中 NQTSW 为触发事件, 且 NQTEN 有效时生效
SQTSW	3	W	0	ADC1 特殊通道序列外部触发软启动: 0: 复位状态 1: 启动特殊通道序列转换 注: 该启动仅在 SQTSEL 选中 SQTSW 为触发事件, 且 SQTEN 有效时生效
CONT	4	W/R	0	ADC1 连续模式:

参数名	比特位	属性	复位值	描述
				0: 单次模式 1: 连续模式
SCAN	5	W/R	0	ADC1 扫描模式: 0: 关闭扫描模式 1: 打开扫描模式
SQAUTO	6	W/R	0	ADC1 特殊通道序列触发方式管理: 0: 外部触发 1: 自动触发
ADCPD	7	W/R	0	ADC1 的低功耗控制: 0: 正常工作模式 1: 低功耗模式

17.15.1.38PGA1 控制寄存器(ADC1PGAC) [0xE4]

Table 357 PGA1 控制寄存器(ADC1PGAC)

参数名	比特位	属性	复位值	描述
PGAMUL	2-0	W/R	010	PGA1 运放倍数选择: 000: 1/4 倍 001: 1/2 倍 010: 1 倍 011: 2 倍 100: 4 倍 101: 8 倍 110: 16 倍 111: 32 倍
PGAPOL	3	W/R	0	PGA1 极性寄存器: 0: 反相 1: 同相
PGATRIM	4	W/R	0	PGA1 校准使能: 0: 不使能 1: 使能 注: 仅用于出厂前校准
PGAPD	5	W/R	0	PGA1 的低功耗控制: 0: 正常工作模式

参数名	比特位	属性	复位值	描述
				1: 低功耗模式
Reserved	7-6	R	0	保留

17.15.1.39 ADC1 中断使能寄存器(ADC1IE) [0xC1]

Table 358 ADC1 中断使能寄存器(ADC1IE)

参数名	比特位	属性	复位值	描述
NCHINTEN	0	W/R	0	ADC1 普通通道转换完成中断使能: 0: 不使能 1: 使能
SCHINTEN	1	W/R	0	ADC1 特殊通道转换完成中断使能: 0: 不使能 1: 使能
WDINTEN	2	W/R	0	ADC1 模拟看门狗中断使能: 0: 不使能 1: 使能
Reserved	7-3	R	0	保留

17.15.1.40 ADC1 中断标志寄存器(ADC1IF) [0xC2]

Table 359 ADC1 中断标志寄存器(ADC1IF)

参数名	比特位	属性	复位值	描述
NCHINT	0	W/R	0	ADC1 普通通道转换完成中断标志位: <ul style="list-style-type: none"> ● 当有普通通道转换完成中断申请时, 该位置为 1 ● 不支持软中断, 在中断处理程序中需要软件写 0 来清除
SCHINT	1	W/R	0	ADC1 特殊通道转换完成中断标志位: <ul style="list-style-type: none"> ● 当有特殊通道转换完成中断申请时, 该位置为 1 ● 不支持软中断, 在中断处理程序中需要软件写 0 来清除
WDINT	2	W/R	0	ADC1 模拟看门狗中断标志位: <ul style="list-style-type: none"> ● 当有模拟看门狗中断申请时, 该位置为 1

参数名	比特位	属性	复位值	描述
				● 不支持软中断，在中断处理程序中需要软件写 0 来清除
Reserved	7-3	R	0	保留

17.15.1.41 ADC1 看门狗控制寄存器(ADC1WDC) [0xC3]

Table 360 ADC1 看门狗控制寄存器(ADC1WDC)

参数名	比特位	属性	复位值	描述
WDCH	3-0	W/R	0	当 SCHEN 为 1 时 (单一通道模式), ADC1 看门狗通道选择: 0000: AN0 0001: AN1 0010: AN2 1010: AN10 1011: AN11 其他: 保留
NWDEN	4	W/R	0	ADC1 普通通道序列模拟看门狗使能: 0: 不使能 1: 使能
SWDEN	5	W/R	0	ADC1 特殊通道序列模拟看门狗使能: 0: 不使能 1: 使能
SCHEN	6	W/R	0	ADC1 看门狗单一通道模式: 0: 所有通道上使能看门狗 1: WDCH 指定的单一通道上使能看门狗
Reserved	7	R	0	保留

17.15.1.42 ADC1 看门狗高阈值寄存器低 8 位(ADC1HTL) [0xC4]

Table 361 ADC1 看门狗高阈值寄存器低 8 位(ADC1HTL)

参数名	比特位	属性	复位值	描述
ADCHTL	7-0	W/R	0	ADC1 看门狗高阈值寄存器低 8 位

17.15.1.43 ADC1 看门狗高阈值寄存器高 2 位(ADC1HTH) [0xC5]

Table 362 ADC1 看门狗高阈值寄存器高 2 位(ADC1HTH)

参数名	比特位	属性	复位值	描述
ADCHTH	1-0	W/R	0	ADC1 看门狗高阈值寄存器高 2 位
Reserved	7-2	R	0	保留

17.15.1.44 ADC1 看门狗低阈值寄存器低 8 位(ADC1LTL) [0xC6]

Table 363 ADC1 看门狗低阈值寄存器低 8 位(ADC1LTL)

参数名	比特位	属性	复位值	描述
ADCLTL	7-0	W/R	0	ADC1 看门狗低阈值寄存器低 8 位

17.15.1.45 ADC1 看门狗低阈值寄存器高 2 位(ADC1LTH) [0xC7]

Table 364 ADC1 看门狗低阈值寄存器高 2 位(ADC1LTH)

参数名	比特位	属性	复位值	描述
ADCLTH	1-0	W/R	0	ADC1 看门狗低阈值寄存器高 2 位
Reserved	7-2	R	0	保留

17.15.1.46 ADC1 序列长度控制寄存器(ADC1QLC) [0xC9]

Table 365 ADC1 序列长度控制寄存器(ADC1QLC)

参数名	比特位	属性	复位值	描述
NQL	3-0	W/R	0	ADC1 普通通道序列长度： 0000：第 1 个通道转换 0001：第 1~2 个通道转换 0010：第 1~3 个通道转换 1010：第 1~11 个通道转换 1011：第 1~12 个通道转换
SQL	5-4	W/R	0	ADC1 特殊通道序列长度： 00：第 1 个通道转换 01：第 1~2 个通道转换 10：第 1~3 个通道转换 11：第 1~4 个通道转换
Reserved	7-6	R	0	保留

17.15.1.47 ADC1 普通序列外部触发控制寄存器(ADC1NQTC) [0xCA]

Table 366 ADC1 普通序列外部触发控制寄存器(ADC1NQTC)

参数名	比特位	属性	复位值	描述
NQTSEL	5-0	W/R	0	ADC1 普通通道序列外部触发事件选择： 000000: TIM0_OC0 000001: TIM0_OC1 000010: TIM0_OC2 000011: TIM0_OC3 000100: TIM1_OC0 000101: TIM1_OC1 000110: TIM1_OC2 000111: TIM1_OC3 001100: TIM0_TRGO 001101: TIM1_TRGO 010000: EXTI0_INT0 010001: EXTI1_INT0 010010: EXTI2_INT0 010011: EXTI3_INT0 010100: EXTI4_INT0 010101: EXTI5_INT0 010110: EXTI6_INT0 010111: EXTI7_INT0 011000: EXTI0_INT1 011001: EXTI1_INT1 011010: EXTI2_INT1 011011: EXTI3_INT1 011100: EXTI4_INT1 011101: EXTI5_INT1 011110: EXTI6_INT1 011111: EXTI7_INT1 100000: NQTSW 其他: 保留
NQTEN	6	W/R	0	ADC1 普通通道序列外部触发使能:

参数名	比特位	属性	复位值	描述
				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
Reserved	7	R	0	保留

17.15.1.48 ADC1 普通序列寄存器 0(ADC1NQ0) [0xCB]

Table 367 ADC1 普通序列寄存器 0(ADC1NQ0)

参数名	比特位	属性	复位值	描述
NQ0	3-0	W/R	0	ADC1 普通序列第 1 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留
NQ1	7-4	W/R	0	ADC1 普通序列第 2 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.49 ADC1 普通序列寄存器 1(ADC1NQ1) [0xCC]

Table 368 ADC1 普通序列寄存器 1(ADC1NQ1)

参数名	比特位	属性	复位值	描述
NQ2	3-0	W/R	0	ADC1 普通序列第 3 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留

参数名	比特位	属性	复位值	描述
NQ3	7-4	W/R	0	ADC1 普通序列第 4 个转换通道源选择： 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.50 ADC1 普通序列寄存器 2(ADC1NQ2) [0xCD]

Table 369 ADC1 普通序列寄存器 2(ADC1NQ2)

参数名	比特位	属性	复位值	描述
NQ4	3-0	W/R	0	ADC1 普通序列第 5 个转换通道源选择： 0000: AN0 0001: AN1 1011: AN11 其他: 保留
NQ5	7-4	W/R	0	ADC1 普通序列第 6 个转换通道源选择： 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.51 ADC1 普通序列寄存器 3(ADC1NQ3) [0xCE]

Table 370 ADC1 普通序列寄存器 3(ADC1NQ3)

参数名	比特位	属性	复位值	描述
NQ6	3-0	W/R	0	ADC1 普通序列第 7 个转换通道源选择： 0000: AN0 0001: AN1 1011: AN11

参数名	比特位	属性	复位值	描述
				其他：保留
NQ7	7-4	W/R	0	ADC1 普通序列第 8 个转换通道源选择： 0000：AN0 0001：AN1 1011：AN11 其他：保留

17.15.1.52 ADC1 普通序列寄存器 4(ADC1NQ4) [0xCF]

Table 371 ADC1 普通序列寄存器 4(ADC1NQ4)

参数名	比特位	属性	复位值	描述
NQ8	3-0	W/R	0	ADC1 普通序列第 9 个转换通道源选择： 0000：AN0 0001：AN1 1011：AN11 其他：保留
NQ9	7-4	W/R	0	ADC1 普通序列第 10 个转换通道源选择： 0000：AN0 0001：AN1 1011：AN11 其他：保留

17.15.1.53 ADC1 普通序列寄存器 5(ADC1NQ5) [0xD1]

Table 372 ADC1 普通序列寄存器 5(ADC1NQ5)

参数名	比特位	属性	复位值	描述
NQ10	3-0	W/R	0	ADC1 普通序列第 11 个转换通道源选择： 0000：AN0 0001：AN1

参数名	比特位	属性	复位值	描述
				1011: AN11 其他: 保留
NQ11	7-4	W/R	0	ADC1 普通序列第 12 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.54 ADC1 普通序列结果寄存器低 8 位(ADC1NRL) [0xD2]

Table 373 ADC1 普通序列结果寄存器低 8 位(ADC1NRL)

参数名	比特位	属性	复位值	描述
NRL	7-0	R	0	ADC1 普通序列结果低 8 位

17.15.1.55 ADC1 普通序列结果寄存器高 8 位(ADC1NRH) [0xD3]

Table 374 ADC1 普通序列结果寄存器高 8 位(ADC1NRH)

参数名	比特位	属性	复位值	描述
NRH	7-0	R	0	ADC1 普通序列结果高 8 位

17.15.1.56 ADC1 特殊序列外部触发控制寄存器(ADC1SQTC) [0xD4]

Table 375 ADC1 特殊序列外部触发控制寄存器(ADC1SQTC)

参数名	比特位	属性	复位值	描述
SQTSEL	5-0	W/R	0	ADC1 特殊通道序列外部触发事件选择: 000000: TIM0_OC0 000001: TIM0_OC1 000010: TIM0_OC2 000011: TIM0_OC3 000100: TIM1_OC0 000101: TIM1_OC1 000110: TIM1_OC2 000111: TIM1_OC3 001100: TIM0_TRGO

参数名	比特位	属性	复位值	描述
				001101: TIM1_TRGO 010000: EXTI0_INT0 010001: EXTI1_INT0 010010: EXTI2_INT0 010011: EXTI3_INT0 010100: EXTI4_INT0 010101: EXTI5_INT0 010110: EXTI6_INT0 010111: EXTI7_INT0 011000: EXTI0_INT1 011001: EXTI1_INT1 011010: EXTI2_INT1 011011: EXTI3_INT1 011100: EXTI4_INT1 011101: EXTI5_INT1 011110: EXTI6_INT1 011111: EXTI7_INT1 100000: SQTSW 其他: 保留
SQTEN	6	W/R	0	ADC1 特殊通道序列外部触发使能: 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
Reserved	7	R	0	保留

17.15.1.57 ADC1 特殊序列寄存器 0(ADC1SQ0) [0xD5]

Table 376 ADC1 特殊序列寄存器 0(ADC1SQ0)

参数名	比特位	属性	复位值	描述
SQ0	3-0	W/R	0	ADC1 特殊序列第 1 个转换通道源选择: 0000: AN0 0001: AN1

参数名	比特位	属性	复位值	描述
				1011: AN11 其他: 保留
SQ1	7-4	W/R	0	ADC1 特殊序列第 2 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.58 ADC1 特殊序列寄存器 1(ADC1SQ1) [0xD6]

Table 377 ADC1 特殊序列寄存器 1(ADC1SQ1)

参数名	比特位	属性	复位值	描述
SQ2	3-0	W/R	0	ADC1 特殊序列第 3 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留
SQ3	7-4	W/R	0	ADC1 特殊序列第 4 个转换通道源选择: 0000: AN0 0001: AN1 1011: AN11 其他: 保留

17.15.1.59 ADC1 特殊序列第 1 个转换通道结果寄存器低 8 位(ADC1SR0L) [0xD7]

Table 378 ADC1 特殊序列第 1 个转换通道结果寄存器低 8 位(ADC1SR0L)

参数名	比特位	属性	复位值	描述
SR0L	7-0	R	0	ADC1 特殊序列第 1 个转换通道结果寄存器低 8 位

17.15.1.60 ADC1 特殊序列第 1 个转换通道结果寄存器高 8 位(ADC1SR0H) [0xD9]

Table 379 ADC1 特殊序列第 1 个转换通道结果寄存器高 8 位(ADC1SR0H)

参数名	比特位	属性	复位值	描述
SR0H	7-0	R	0	ADC1 特殊序列第 1 个转换通道结果寄存器高 8 位

17.15.1.61 ADC1 特殊序列第 2 个转换通道结果寄存器低 8 位(ADC1SR1L) [0xDA]

Table 380 ADC1 特殊序列第 2 个转换通道结果寄存器低 8 位(ADC1SR1L)

参数名	比特位	属性	复位值	描述
SR1L	7-0	R	0	ADC1 特殊序列第 2 个转换通道结果寄存器低 8 位

17.15.1.62 ADC1 特殊序列第 2 个转换通道结果寄存器高 8 位(ADC1SR1H) [0xDB]

Table 381 ADC1 特殊序列第 2 个转换通道结果寄存器高 8 位(ADC1SR1H)

参数名	比特位	属性	复位值	描述
SR1H	7-0	R	0	ADC1 特殊序列第 2 个转换通道结果寄存器高 8 位

17.15.1.63 ADC1 特殊序列第 3 个转换通道结果寄存器低 8 位(ADC1SR2L) [0xDC]

Table 382 ADC1 特殊序列第 3 个转换通道结果寄存器低 8 位(ADC1SR2L)

参数名	比特位	属性	复位值	描述
SR2L	7-0	R	0	ADC1 特殊序列第 3 个转换通道结果寄存器低 8 位

17.15.1.64 ADC1 特殊序列第 3 个转换通道结果寄存器高 8 位(ADC1SR2H) [0xDD]

Table 383 ADC1 特殊序列第 3 个转换通道结果寄存器高 8 位(ADC1SR2H)

参数名	比特位	属性	复位值	描述
SR2H	7-0	R	0	ADC1 特殊序列第 3 个转换通道结果寄存器高 8 位

17.15.1.65 ADC1 特殊序列第 4 个转换通道结果寄存器低 8 位(ADC1SR3L) [0xDE]

Table 384 ADC1 特殊序列第 4 个转换通道结果寄存器低 8 位(ADC1SR3L)

参数名	比特位	属性	复位值	描述
SR3L	7-0	R	0	ADC1 特殊序列第 4 个转换通道结果寄存器低 8 位

17.15.1.66 ADC1 特殊序列第 4 个转换通道结果寄存器高 8 位(ADC1SR3H) [0xDF]

Table 385 ADC1 特殊序列第 4 个转换通道结果寄存器高 8 位(ADC1SR3H)

参数名	比特位	属性	复位值	描述
SR3H	1-0	R	0	ADC1 特殊序列第 4 个转换通道结果寄存器高 8 位

17.15.1.67 ADC1 结果数据格式寄存器(ADC1DF) [0xE1]

Table 386 ADC1 结果数据格式寄存器(ADC1DF)

参数名	比特位	属性	复位值	描述
-----	-----	----	-----	----

参数名	比特位	属性	复位值	描述
ADC1DF	1-0	W/R	0	ADC1 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
Reserved	7-2	R	0	保留

17.15.1.68 ADC1 外部触发边沿控制寄存器(ADC1ETC) [0xE5]

Table 387 ADC1 外部触发边沿控制寄存器(ADC1ETC)

参数名	比特位	属性	复位值	描述
NQTPOSEN	0	W/R	1	ADC1 普通通道外部触发信号上升沿触发使能： 0: 不使能 1: 使能
NQTNEGEN	1	W/R	1	ADC1 普通通道外部触发信号下降沿触发使能： 0: 不使能 1: 使能
SQTPOSEN	2	W/R	1	ADC1 特殊通道外部触发信号上升沿触发使能： 0: 不使能 1: 使能
SQTNEGEN	3	W/R	1	ADC1 特殊通道外部触发信号下降沿触发使能： 0: 不使能 1: 使能
Reserved	7-4	R	0	保留

17.15.1.69 ADC1 普通通道触发延时寄存器(ADC1NQTD) [0xE6]

Table 388 ADC1 普通通道触发延时寄存器(ADC1NQTD)

参数名	比特位	属性	复位值	描述
NQTD	7-0	W/R	0	ADC1 普通通道触发延时: $NQTD \times T_{adc_clk}$

17.15.1.70 ADC1 特殊通道触发延时寄存器(ADC1SQT D) [0xE7]

Table 389 ADC1 特殊通道触发延时寄存器(ADC1SQT D)

参数名	比特位	属性	复位值	描述
SQT D	7-0	W/R	0	ADC1 普通通道触发延时: $SQT D \times T_{adc_clk}$

17.15.2 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, 寄存器列表如下所示。

Table 390 JMT018 访问 ADC 寄存器

地址	寄存器名	属性	复位值	功能描述
0x40	ADC0NR	R	0x0000	ADC0 普通序列结果寄存器
0x41	ADC0SR0	R	0x0000	ADC0 特殊序列第 1 个转换通道结果寄存器
0x42	ADC0SR1	R	0x0000	ADC0 特殊序列第 2 个转换通道结果寄存器
0x43	ADC0SR2	R	0x0000	ADC0 特殊序列第 3 个转换通道结果寄存器
0x44	ADC0SR3	R	0x0000	ADC0 特殊序列第 4 个转换通道结果寄存器
0x80	ADC1NR	R	0x0000	ADC1 普通序列结果寄存器
0x81	ADC1SR0	R	0x0000	ADC1 特殊序列第 1 个转换通道结果寄存器
0x82	ADC1SR1	R	0x0000	ADC1 特殊序列第 2 个转换通道结果寄存器
0x83	ADC1SR2	R	0x0000	ADC1 特殊序列第 3 个转换通道结果寄存器
0x84	ADC1SR3	R	0x0000	ADC1 特殊序列第 4 个转换通道结果寄存器

17.15.2.1 ADC0 普通序列结果寄存器 [0x40]

Table 391 ADC0 普通序列结果寄存器(ADC0NR)

参数名	比特位	属性	复位值	描述
ADC0NR	15-0	R	0	ADC0 普通序列结果

17.15.2.2 ADC0 特殊序列第 1 个转换通道结果寄存器 [0x41]

Table 392 ADC0 特殊序列第 1 个转换通道结果寄存器(ADC0SR0)

参数名	比特位	属性	复位值	描述
ADC0SR0	15-0	R	0	ADC0 特殊序列第 1 个转换通道结果

17.15.2.3 ADC0 特殊序列第 2 个转换通道结果寄存器 [0x42]

Table 393 ADC0 特殊序列第 2 个转换通道结果寄存器(ADC0SR1)

参数名	比特位	属性	复位值	描述
ADC0SR1	15-0	R	0	ADC0 特殊序列第 2 个转换通道结果

17.15.2.4 ADC0 特殊序列第 3 个转换通道结果寄存器 [0x43]

Table 394 ADC0 特殊序列第 3 个转换通道结果寄存器(ADC0SR2)

参数名	比特位	属性	复位值	描述
ADC0SR2	15-0	R	0	ADC0 特殊序列第 3 个转换通道结果

17.15.2.5 ADC0 特殊序列第 4 个转换通道结果寄存器 [0x44]

Table 395 ADC0 特殊序列第 4 个转换通道结果寄存器(ADC0SR3)

参数名	比特位	属性	复位值	描述
ADC0SR3	15-0	R	0	ADC0 特殊序列第 4 个转换通道结果

17.15.2.6 ADC1 普通序列结果寄存器 [0x80]

Table 396 ADC1 普通序列结果寄存器(ADC1NR)

参数名	比特位	属性	复位值	描述
ADC1NR	15-0	R	0	ADC1 普通序列结果

17.15.2.7 ADC1 特殊序列第 1 个转换通道结果寄存器 [0x81]

Table 397 ADC1 特殊序列第 1 个转换通道结果寄存器(ADC1SR0)

参数名	比特位	属性	复位值	描述
ADC1SR0	15-0	W/R	0	ADC1 特殊序列第 1 个转换通道结果

17.15.2.8 ADC1 特殊序列第 2 个转换通道结果寄存器 [0x82]

Table 398 ADC1 特殊序列第 2 个转换通道结果寄存器(ADC1SR1)

参数名	比特位	属性	复位值	描述
ADC1SR1	15-0	R	0	ADC1 特殊序列第 2 个转换通道结果

17.15.2.9 ADC1 特殊序列第 3 个转换通道结果寄存器 [0x83]

Table 399 ADC1 特殊序列第 3 个转换通道结果寄存器(ADC1SR2)

参数名	比特位	属性	复位值	描述
ADC1SR2	15-0	R	0	ADC1 特殊序列第 3 个转换通道结果

17.15.2.10 ADC1 特殊序列第 4 个转换通道结果寄存器 [0x84]

Table 400 ADC1 特殊序列第 4 个转换通道结果寄存器(ADC1SR3)

参数名	比特位	属性	复位值	描述
ADC1SR3	15-0	R	0	ADC1 特殊序列第 4 个转换通道结果

18 模拟比较器 (CMP)

18.1 模拟比较器概述

JMT1808R 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。

模拟比较器参考电压可在内部参考电压和外部参考电压之间进行选择。

模拟比较器的输入可在芯片模拟供电 VDDA、芯片管脚 PC0 和 PGA1 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 188 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：

- 参考电压源选择：配置寄存器 CMPREFSEL 选择参考电压源：
 - 0：选择芯片外部参考电压 VREF 作为参考电压。
 - 1：选择芯片内部 2.5V 作为参考电压（默认）。
- 配置参考电压大小：选定参考电压源之后，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 PC0 作为比较器输入。
 - 1x：选择 PGA1 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 PC0 作为比较器输入，需要配置管脚 PC0 为模拟输入模式，具体配置参见 GPIO 章节 19 说明。
- 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，经过滤波之后，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：

- 比较器输入源选择模拟供电 **VDDA**：若 **VDDA** 经过 **DAC** 分压之后的值小于参考电压所设定的值，则产生中断。
- 比较器输入源选择 **PC0** 或者 **PGA1** 输出：若 **PC0** 或者 **PGA1** 输出的电压值大于参考电压所设定的值，则产生中断。

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。

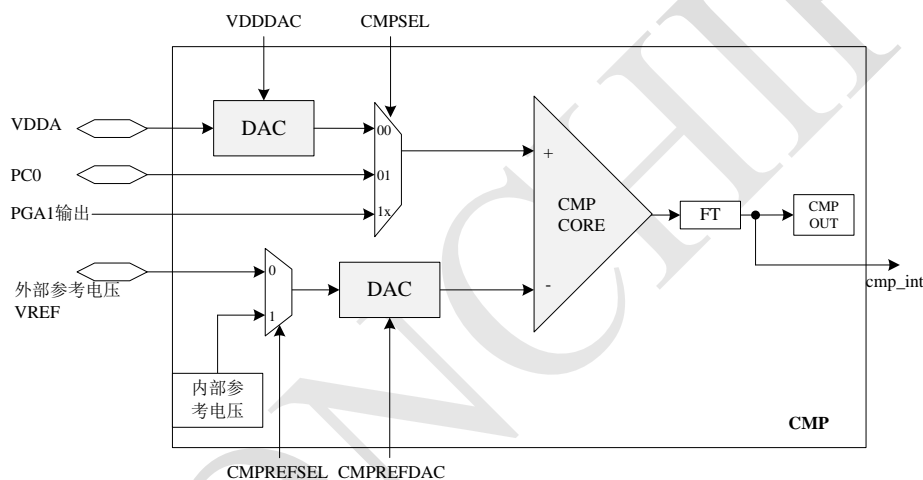


Figure 188 比较器 CMP 内部框图

18.2 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 189 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 10。

比较器作为 LVR 功能的配置包括：

- 根据比较器小节 18.1 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。
- 若需要使能当全芯片电压低于门限电压产生复位信号复位全芯片，先配置寄存器 VDDDAC 和 CMPREFDAC 产生所需的 V_{VDDA} 值，再配置寄存器 LVRRSTEN，启动 LVR 复位功能。

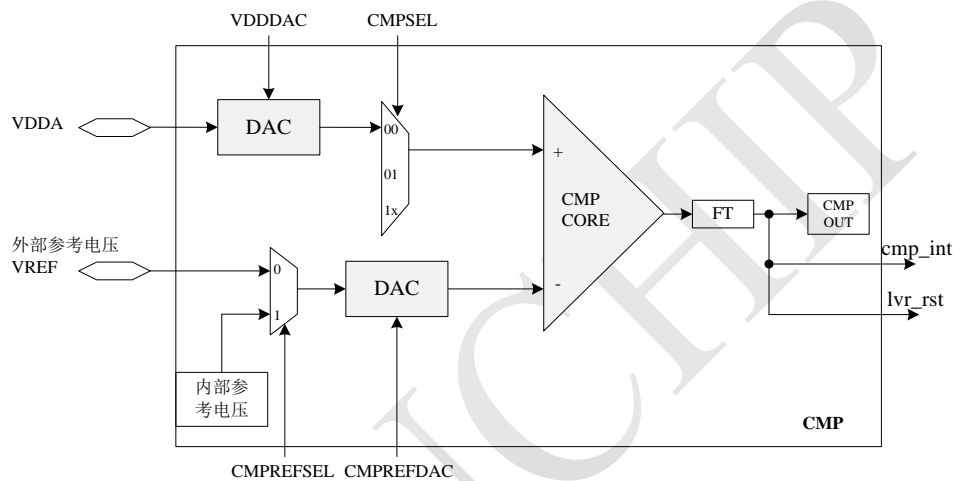


Figure 189 LVR 实现框图

18.2.1 低压检测保护程序示例

通过 CMP 模块的低压检测功能可以实现应用程序的保护，防止芯片在掉电过程中发生异常。下面为保护程序的示例，对应场景为选择芯片内部 2.5V 为比较器的参考电压，设定芯片供电电压低于 2.8V 时触发 CMP 中断进行芯片保护。用户可以根据需要参考使用此保护程序。

低压检测保护程序比较器初始化配置示例如下，其中“//”后部分为程序含义说明：

```
void cmp_init()
{
    SFRPAGE=0x03;//时钟复位管理 CRM 对应的 SFRPAGE
    CMP_CLK_CTRL=0x08;//打开比较器输出滤波时钟
    SFRPAGE=0x00;//比较器 CMP 对应的 SFRPAGE
    CMPEN    =0x00;//打开 CMP 并设置输入为芯片模拟供电 VDDA
    CMPFT    =0x03;//设置 CMP 输出滤波时钟为 3 个滤波时钟周期
    CMPVDD   =0x18;//与 CMPREF 配合设置 VDDA 的电压，此处为 2.8V
    CMPREF   =0x5B;//与 CMPVDD 配合设置 VDDA 的电压，此处为 2.8V，同时设置参
```

考电压为内部 2.5V

```
CMPINT =0x00;//初始化清除中断
CMPFLAG=0x00;//初始化清除比较结果标志
IEN1 |= 0x20;//打开 CMP 中断使能
IEN0 |= 0x80;// 打开中断总使能
}
```

低压检测保护程序比较器中断处理程序示例如下，其中“//”后部分为程序含义说明：

```
void ISR_CMP() interrupt 11//比较器中断号为 11
{
    PMUCTR=0x08;//进入 deepsleep 睡眠，进行低压保护
}
```

18.3 模拟比较器寄存器

模拟比较器（CMP）共有 6 个寄存器，包括使能寄存器、中断寄存器、比较器比较结果寄存器、滤波选择寄存器、VDD 分压寄存器和参考电压选择寄存器。对应的 SFR PAGE 为 0。

Table 401 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
0x91	CMPEN	W/R	0x00	模拟比较器使能寄存器
0x92	CMPINT	W/R	0x00	模拟比较器中断寄存器
0x93	CMPFLAG	R	0x00	模拟比较器比较结果寄存器
0x94	CMPFT	W/R	0x00	模拟比较器滤波选择寄存器
0x95	CMPVDD	W/R	0x20	模拟比较器 VDD 分压寄存器
0x96	CMPREF	W/R	0x52	模拟比较器参考电压选择寄存器

18.3.1 模拟比较器使能寄存器(CMPEN) [0x91]

Table 402 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	0	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 PC0

参数名	比特位	属性	复位值	描述
				10/11: PGA1 输出
Reserved	7-3	R	0	保留

18.3.2 模拟比较器中断寄存器(CMPINT) [0x92]

Table 403 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"> ● 当有模拟比较器中断申请时，该位置为 1 ● 若比较器输入选择寄存器 CMPSEL 配置为 0，选择芯片 VDD 时，比较器输入小于参考电压产生该中断； ● 若比较器输入选择寄存器 CMPSEL 配置为非 0 时，也即选择 PC0 或者 PGA1 输出作为输入时，比较器输入大于参考电压产生该中断； 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

18.3.3 模拟比较器比较结果寄存器(CMPFLAG) [0x93]

Table 404 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
CMPOUT	0	R	0	比较器输入源 CMPSEL 选择 VDDA 时，模拟比较器输出结果： <ul style="list-style-type: none"> 0: 比较器输入电压大于参考电压 1: 比较器输入电压小于参考电压 比较器输入源 CMPSEL 选择其它时，模拟比较器输出结果： <ul style="list-style-type: none"> 0: 比较器输入电压小于参考电压

参数名	比特位	属性	复位值	描述
				1: 比较器输入电压大于参考电压
Reserved	7-1	R	0	保留

18.3.4 模拟比较器滤波选择寄存器(CMPFT) [0x94]

Table 405 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	0	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

18.3.5 模拟比较器 VDD 分压寄存器(CMPVDD) [0x95]

Table 406 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
Reserved	7-6	R	0	保留

18.3.6 模拟比较器参考电压选择寄存器(CMPREF) [0x96]

Table 407 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
CMPREFSEL	6	W/R	1	模拟比较器参考电压选择：

参数名	比特位	属性	复位值	描述
				0: 芯片外部参考电压 (VREF) 1: 芯片内部 2.5V 参考电压
Reserved	7	R	0	保留

IRONCHIP

19 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。

19.1 管脚设置

数字 I/O 实现框图如 Figure 190 所示，数字、模拟复用 I/O 实现框图如 Figure 191 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，需将寄存器 AIOEN0 和 AIOEN1 的相应通道置位。

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

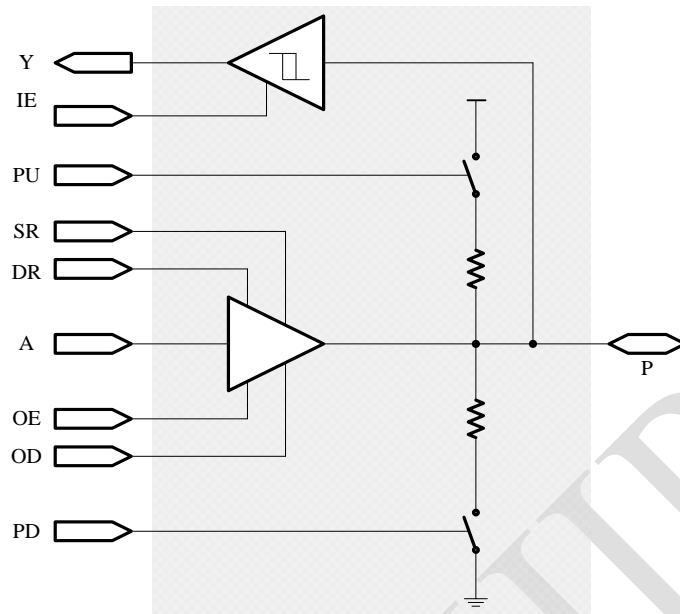


Figure 190 数字 I/O 实现框图

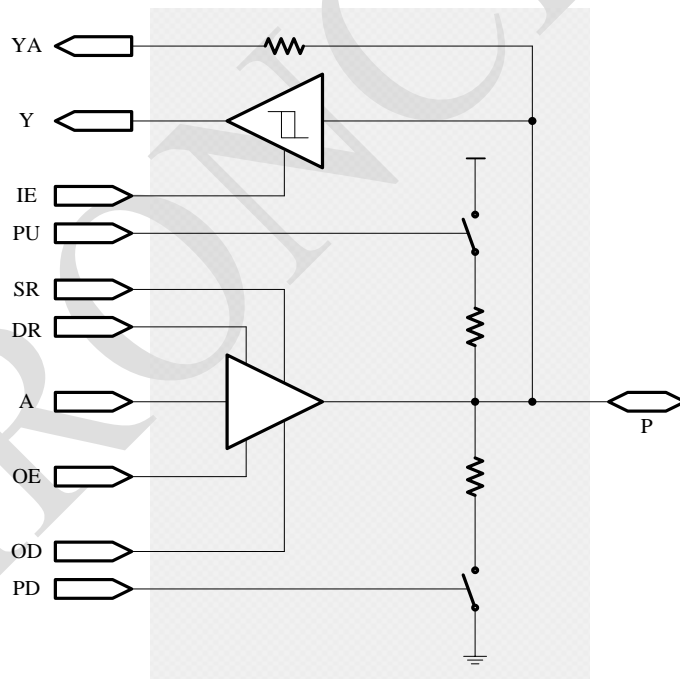


Figure 191 数字/模拟混合 I/O 实现框图

管脚真值表如 Table 408 所示，其中“W”为任意值，“Z”为高阻态，“X”为不确定态：

Table 408 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%。

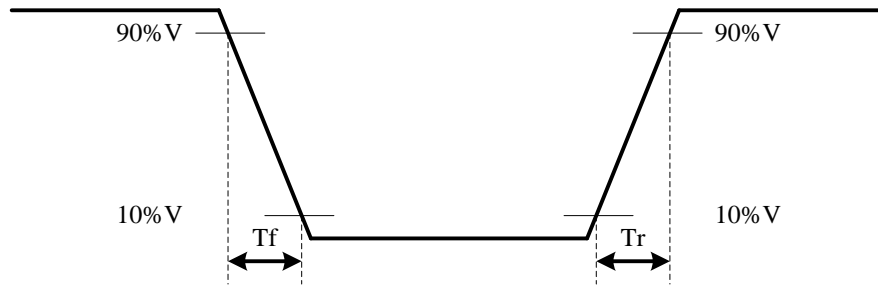


Figure 192 上升/下降转变时间示意图

传播延迟:

- 上升传播延迟 T_{dr} : 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 T_{df} : 下降信号从 PAD 的 P 端输入到 Y 端传输所需时间

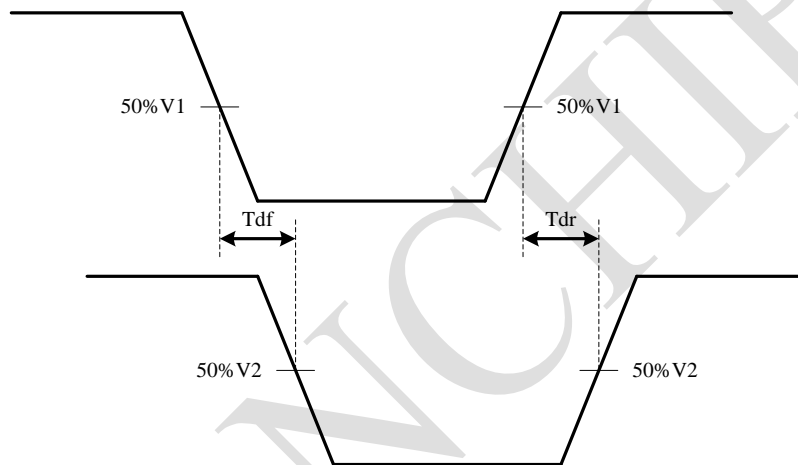


Figure 193 传播延迟示意图

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

19.2 管脚功能复用

PA0~PA7、PC0~PC2 和 PD3~PD7 作为数字模拟复用管脚, 控制参见 Figure 194 和 Figure 195, 虚线框中内容为 GPIO 模块寄存器中的控制位, 具体参见 19.9 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 194 和 Figure 195 中 AIOEN 所示控制。

Figure 194 为 PA0~PA7 管脚复用框图, 产生 INT0 中断, 且 PA0 会产生低功耗唤醒

(WKUP)。

Figure 195 为 PC0~PC2, PD3~PD7 管脚复用框图，产生 INT1 中断。

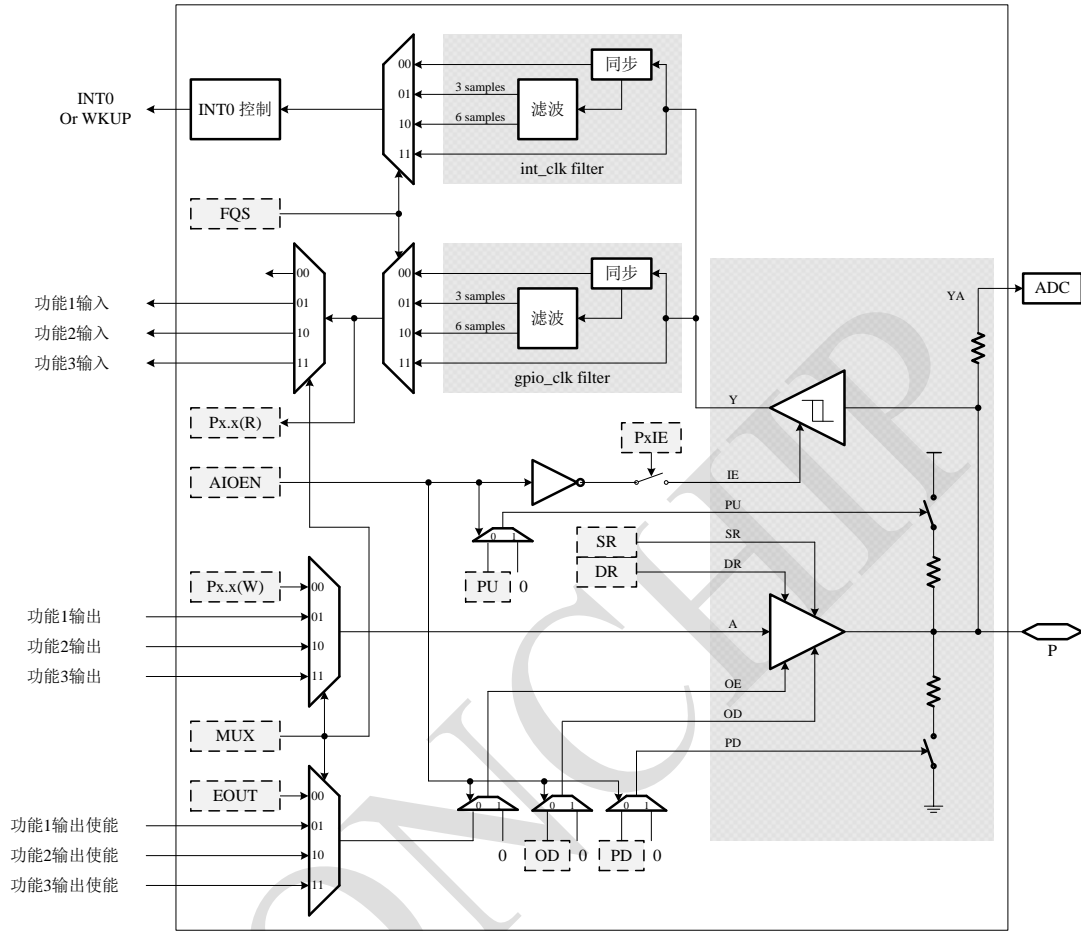


Figure 194 PA0 ~PA7 管脚复用框图

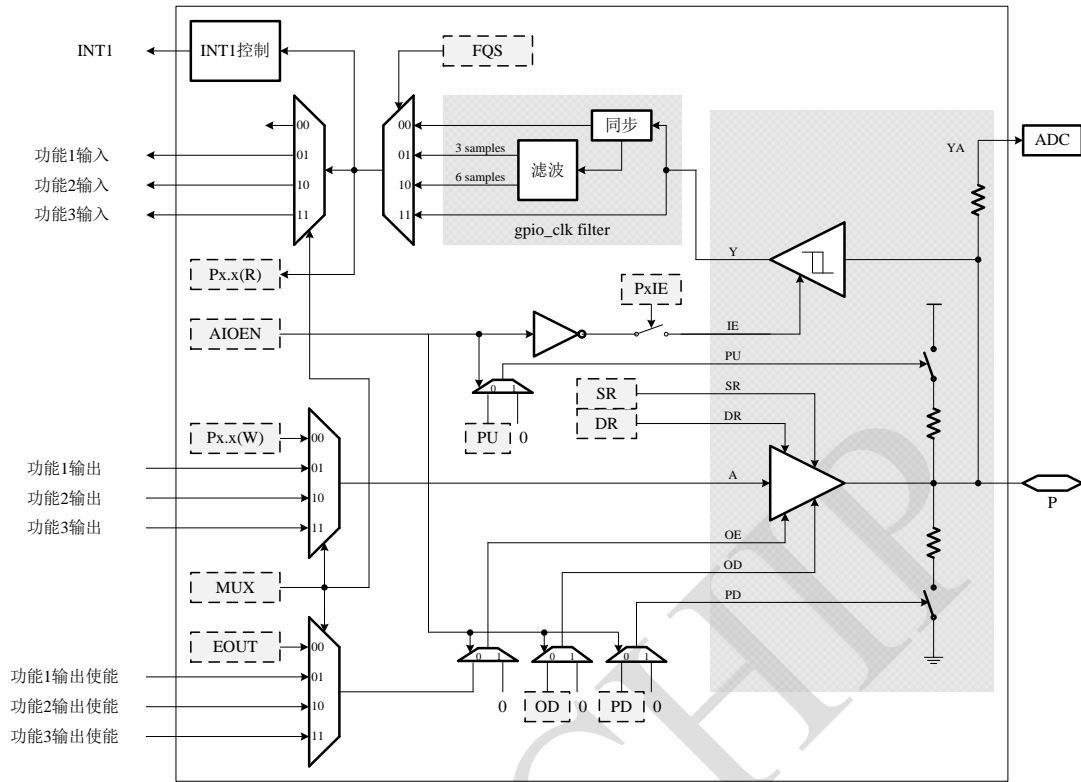


Figure 195 PC0~PC2, PD3~PD7 管脚复用框图

PB0~PB7、PC3~PC7、PD0~PD2、PE0~PE7 和 PG0~PG1 作为纯数字管脚，控制参见 Figure 196 和 Figure 197。

Figure 196 为 PB0~PB7、PC3~PC7、PD0~PD2、PE0~PE7 管脚复用框图，产生 INT1 中断。

Figure 197 为 PG0~PG1 管脚复用框图，不支持外部中断功能，不支持 GPIO 输出功能。

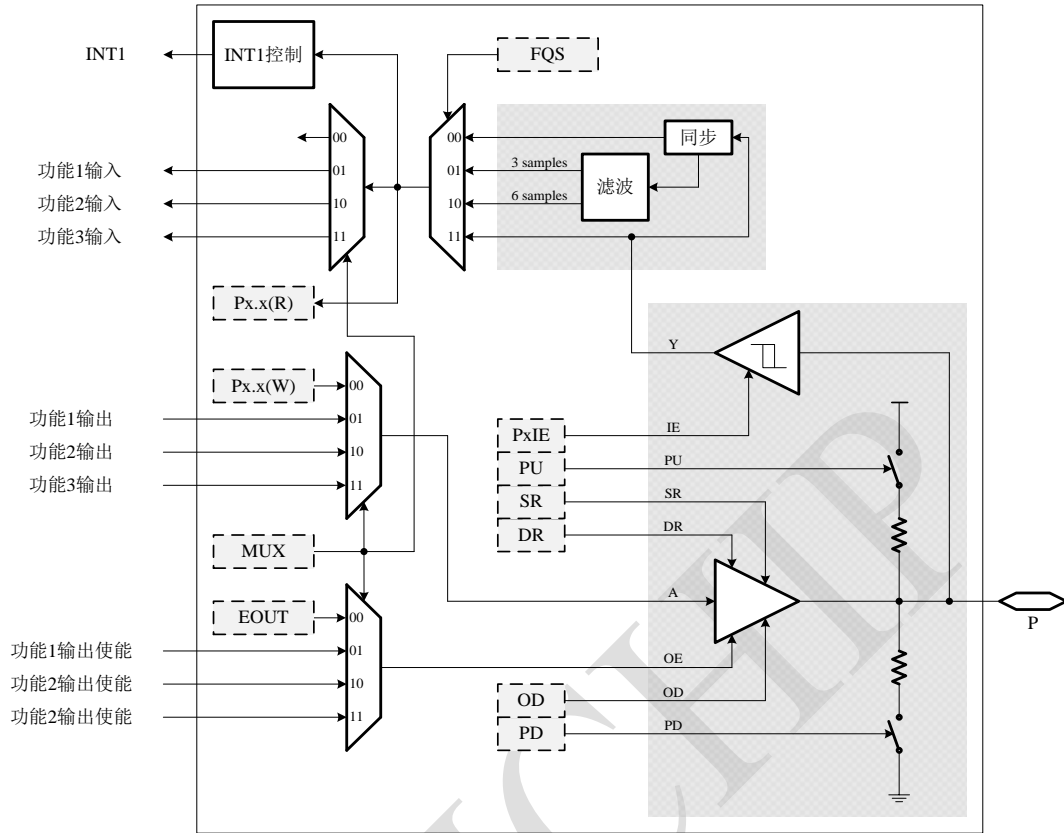


Figure 196 PB0~PB7, PC3~PC7, PD0~PD2, PE0~PE7 管脚复用框图

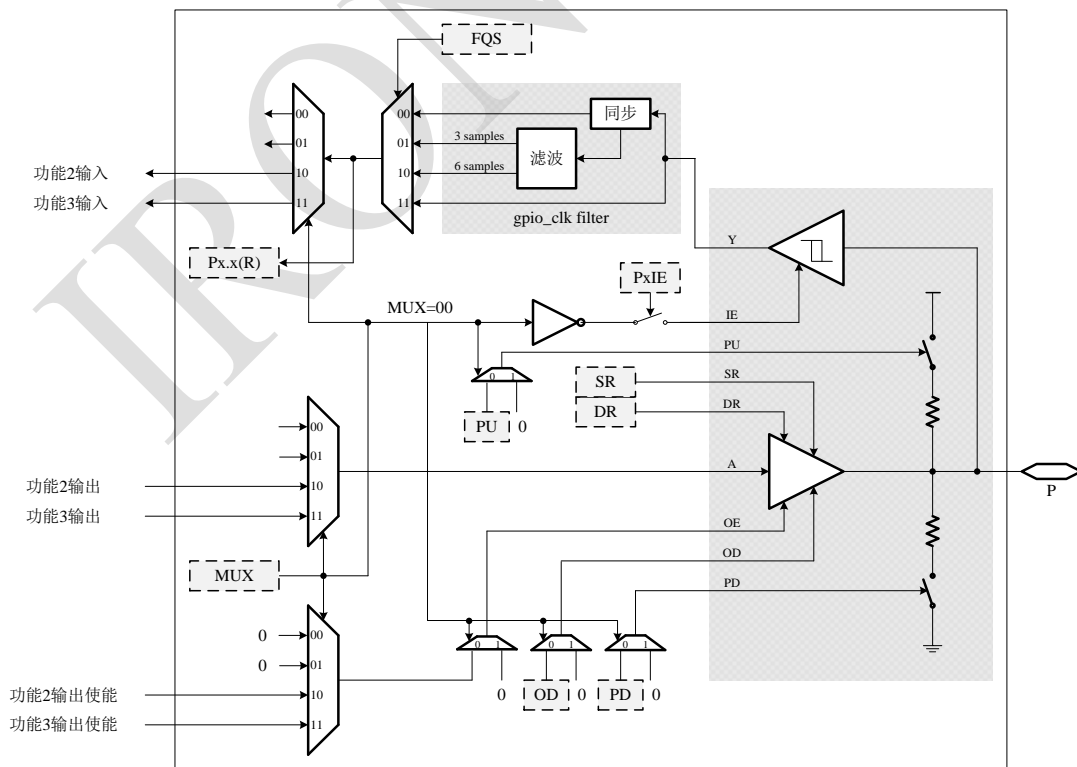


Figure 197 PG0~PG1 管脚复用框图

19.3 PWM 管脚特殊复用

19.3.1 PWM 输出复用

PWM_x 输出除了可以通过 19.2 节所述的功能 x 复用输出，也可以通过特殊通道直接输出，具体参见 Figure 198。

一般情况下，PWM 模块的管脚复用可通过 19.2 节管脚功能复用选择为 PWM 功能，也即选择 Figure 198 中的“正常功能”通道。“PWM 特殊复用”一般用于同一个 PWM 输出到多个 (X 个) 芯片管脚，驱动电容型触摸按键，通过 Y 个通道 AD 检测，得到 X × Y 个触摸按键。

PB0~PB7, PC3~PC7, PD0~PD2 和 PE0~PE7 有特殊通道直接输出 PWM 功能。

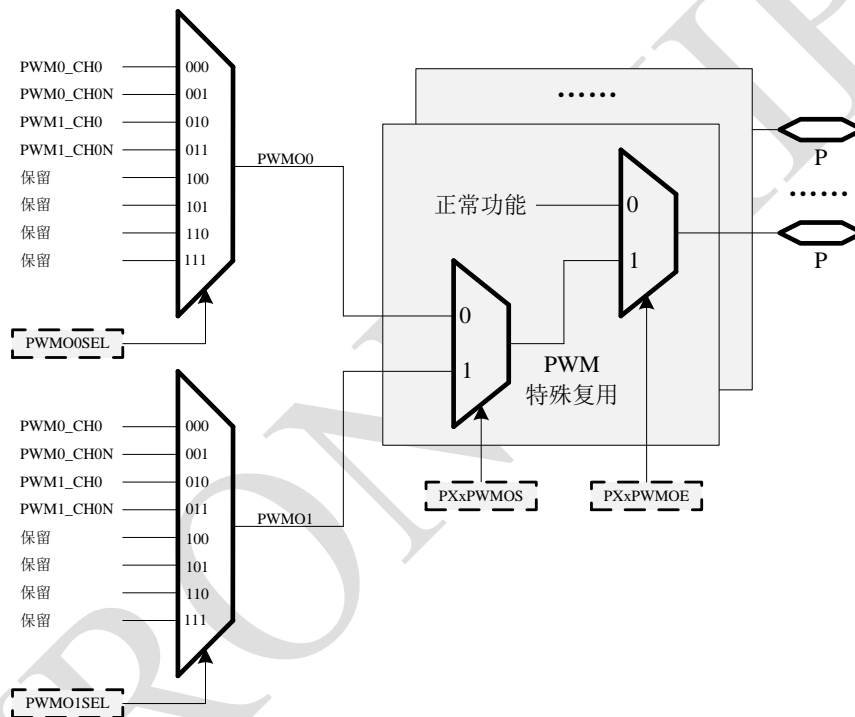


Figure 198 PWM 直接输出框图

19.3.2 PWM 刹车复用

2 个 PWM 的刹车信号，可以从 PWM0_BKIN, PWM1_BKIN 中任意选择，具体参见 Figure 199。

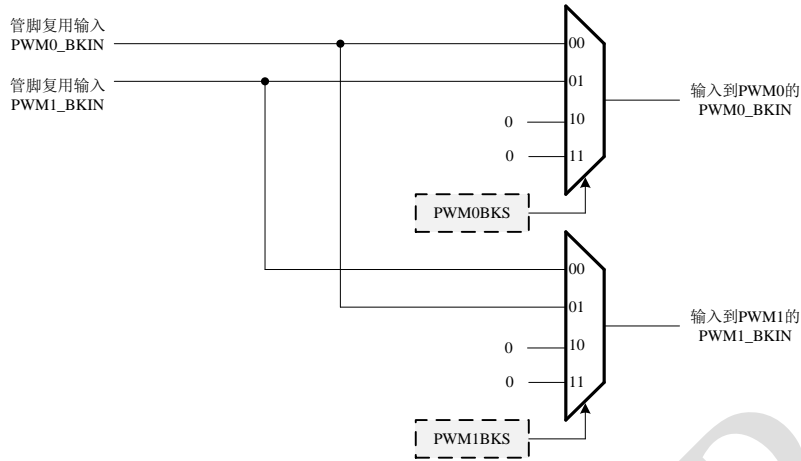


Figure 199 PWM 刹车选择

19.4 管脚数据寄存器读写

管脚数据寄存器（P0、P1、P2、P3、P4 和 P6）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。

19.5 外设管脚复用配置

Table 409 PWM0 管脚配置

管脚复用功能	说明	配置需求
PWM0_BKIN	PWM0 刹车输入	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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。
PWM0_ETR	PWM0 外部时钟输入	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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	配置需求
		PU 和 PD 可根据需要配置。
PWM0_CHx	PWM0 通道 x 输入捕获	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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。
	PWM0 通道 x 输出比较	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能； PU、PD 和 OD 配置为 0。
PWM0_CHxN	PWM0 通道 x 互补输出	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能； PU、PD 和 OD 配置为 0。

Table 410 PWM1 管脚配置

管脚复用功能	说明	配置
PWM1_BKIN	PWM1 刹车输入	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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。
PWM1_ETR	PWM1 外部时钟输入	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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。
PWM1_CHx	PWM1 通道 x 输入捕获	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能；

管脚复用功能	说明	配置
		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。
	PWM1 通道 x 输出比较	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能； PU、PD 和 OD 配置为 0。
PWM1_CHxN	PWM0 通道 x 互补输出	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能； PU、PD 和 OD 配置为 0。

Table 411 UART0 管脚配置

管脚复用功能	说明	配置
UART0_TX	UART 模式数据输出	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能； PU、PD 和 OD 配置为 0。
UART0_RX	UART 模式数据输入	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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。

Table 412 UART1 管脚配置

管脚复用功能	说明	配置
UART1_TX	UART 模式数据输出	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能； PU、PD 和 OD 配置为 0。
UART1_RX	UART 模式数据输入	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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 413 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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：

管脚复用功能	说明	配置
		PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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。
T1_GATE	Timer1 外部门控输入	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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 414 RTC 管脚配置

管脚复用功能	说明	配置
RTC_TM0	RTC 时标 0 输出	相应管脚配置：

管脚复用功能	说明	配置
		PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能； PU、PD 和 OD 配置为 0。
RTC_TM1	RTC 时标 1 输出	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能； PU、PD 和 OD 配置为 0。

Table 415 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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_MOSI	主机模式 SPI 数据输出	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能；

管脚复用功能	说明	配置
		PU、PD 和 OD 配置为 0。
	从机模式 SPI 数据输入	相应管脚配置： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 配置为该复用功能； PU、PD 和 OD 配置为 0。

Table 416 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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： PWM 输出复用使能选择为正常功能模式； 复用选择 MUX 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；

管脚复用功能	说明	配置
		PU 可根据需要配置。

19.6 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 200 所示。

输入管脚可经过四种不同处理方式，默认选择同步模式：

- 同步模式，利用 gpio_clk 进行同步
- 3 个 gpio_clk 时钟周期滤波（可滤除小于 2 个 gpio_clk 周期的毛刺）
- 6 个 gpio_clk 时钟周期滤波（可滤除小于 5 个 gpio_clk 周期的毛刺）
- 直通模式

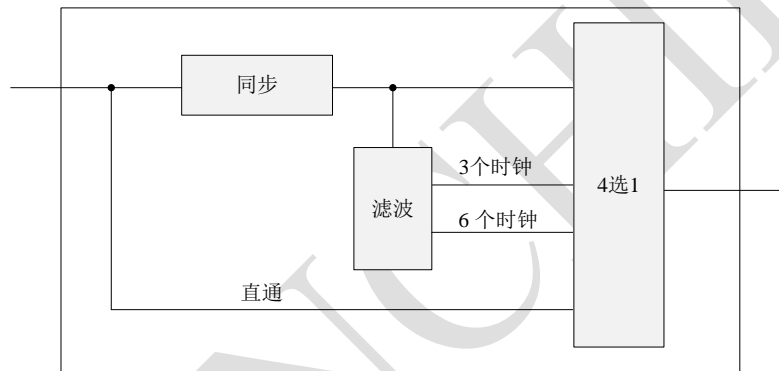


Figure 200 输入管脚滤波

19.7 INT0 中断

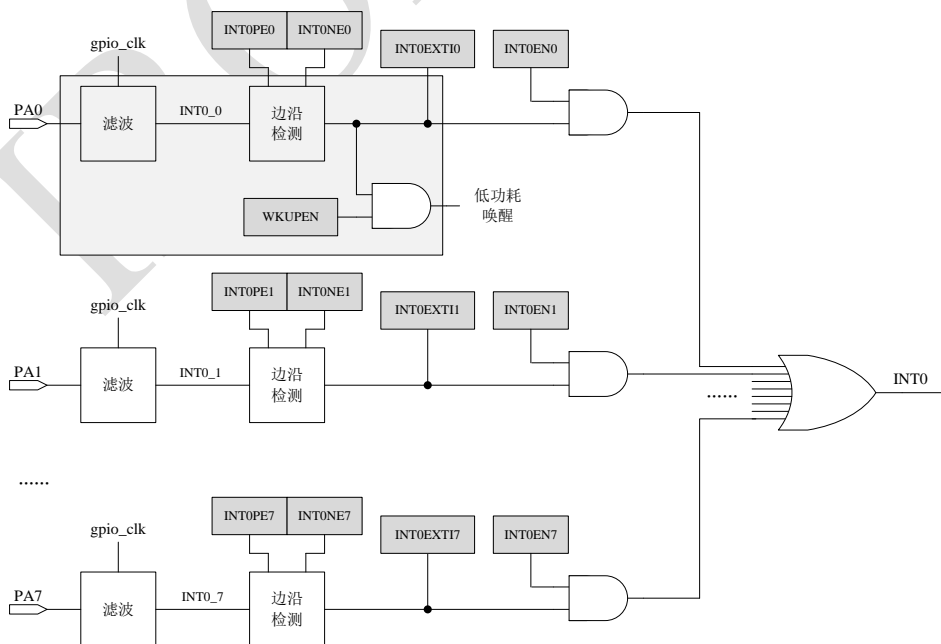


Figure 201 INT0 中断

INT0 中断包括 8 个外部管脚中断源:

- 外部管脚中断源 INT0EXTI0 (PA0):
 - 1) 配置寄存器 PA0CTRL.FQS, 设置管脚 PA0 输入滤波数
 - 2) 配置寄存器 PA0CTRL.MUX 为 00, 配置寄存器 PA0CTRL.EOUT 为 0, 设置管脚 PA0 为 GPIO 输入
 - 3) 配置寄存器 INT0PE0 和 INT0NE0, 选择 INT0EXTI0 中断类型
 - 4) 配置寄存器 INT0EN0 为 1, 打开 INT0EXTI0 中断源
 - 5) 配置寄存器 IEN0.0 为 1, 打开 INT0 中断
 - 6) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 7) 若使用查询方式, 则 4) ~6) 不用配置
- 外部管脚中断源 INT0EXTI1 (PA1):
 - 1) 配置寄存器 PA1CTRL.FQS, 设置管脚 PA1 输入滤波数
 - 2) 配置寄存器 PA1CTRL.MUX 为 00, 配置寄存器 PA1CTRL.EOUT 为 0, 设置管脚 PA1 为 GPIO 输入
 - 3) 配置寄存器 INT0PE1 和 INT0NE1, 选择 INT0EXTI1 中断类型
 - 4) 配置寄存器 INT0EN1 为 1, 打开 INT0EXTI1 中断源
 - 5) 配置寄存器 IEN0.0 为 1, 打开 INT0 中断
 - 6) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 7) 若使用查询方式, 则 4) ~6) 不用配置
- 外部管脚中断源 INT0EXTI2 (PA2):
 - 1) 配置寄存器 PA2CTRL.FQS, 设置管脚 PA2 输入滤波数
 - 2) 配置寄存器 PA2CTRL.MUX 为 00, 配置寄存器 PA2CTRL.EOUT 为 0, 设置管脚 PA2 为 GPIO 输入
 - 3) 配置寄存器 INT0PE2 和 INT0NE2, 选择 INT0EXTI2 中断类型
 - 4) 配置寄存器 INT0EN2 为 1, 打开 INT0EXTI2 中断源
 - 5) 配置寄存器 IEN0.0 为 1, 打开 INT0 中断
 - 6) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 7) 若使用查询方式, 则 4) ~6) 不用配置
- 外部管脚中断源 INT0EXTI3 (PA3):
 - 1) 配置寄存器 PA3CTRL.FQS, 设置管脚 PA3 输入滤波数
 - 2) 配置寄存器 PA3CTRL.MUX 为 00, 配置寄存器 PA3CTRL.EOUT 为 0, 设置管脚 PA3 为 GPIO 输入
 - 3) 配置寄存器 INT0PE3 和 INT0NE3, 选择 INT0EXTI3 中断类型
 - 4) 配置寄存器 INT0EN3 为 1, 打开 INT0EXTI3 中断源

- 5) 配置寄存器 IEN0.0 为 1, 打开 INTO 中断
 - 6) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 7) 若使用查询方式, 则 4) ~6) 不用配置
- 外部管脚中断源 INT0EXTI4 (PA4):
 - 1) 配置寄存器 PA4CTRL.FQS, 设置管脚 PA4 输入滤波数
 - 2) 配置寄存器 PA4CTRL.MUX 为 00, 配置寄存器 PA4CTRL.EOUT 为 0, 设置管脚 PA4 为 GPIO 输入
 - 3) 配置寄存器 INTOPE4 和 INTONE4, 选择 INT0EXTI4 中断类型
 - 4) 配置寄存器 INTOEN4 为 1, 打开 INT0EXTI4 中断源
 - 5) 配置寄存器 IEN0.0 为 1, 打开 INTO 中断
 - 6) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 7) 若使用查询方式, 则 4) ~6) 不用配置
 - 外部管脚中断源 INT0EXTI5 (PA5):
 - 1) 配置寄存器 PA5CTRL.FQS, 设置管脚 PA5 输入滤波数
 - 2) 配置寄存器 PA5CTRL.MUX 为 00, 配置寄存器 PA5CTRL.EOUT 为 0, 设置管脚 PA5 为 GPIO 输入
 - 3) 配置寄存器 INTOPE5 和 INTONE5, 选择 INT0EXTI5 中断类型
 - 4) 配置寄存器 INTOEN5 为 1, 打开 INT0EXTI5 中断源
 - 5) 配置寄存器 IEN0.0 为 1, 打开 INTO 中断
 - 6) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 7) 若使用查询方式, 则 4) ~6) 不用配置
 - 外部管脚中断源 INT0EXTI6 (PA6):
 - 1) 配置寄存器 PA6CTRL.FQS, 设置管脚 PA6 输入滤波数
 - 2) 配置寄存器 PA6CTRL.MUX 为 00, 配置寄存器 PA6CTRL.EOUT 为 0, 设置管脚 PA6 为 GPIO 输入
 - 3) 配置寄存器 INTOPE6 和 INTONE6, 选择 INT0EXTI6 中断类型
 - 4) 配置寄存器 INTOEN6 为 1, 打开 INT0EXTI6 中断源
 - 5) 配置寄存器 IEN0.0 为 1, 打开 INTO 中断
 - 6) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 7) 若使用查询方式, 则 4) ~6) 不用配置
 - 外部管脚中断源 INT0EXTI7 (PA7):
 - 1) 配置寄存器 PA7CTRL.FQS, 设置管脚 PA7 输入滤波数
 - 2) 配置寄存器 PA7CTRL.MUX 为 00, 配置寄存器 PA7CTRL.EOUT 为 0, 设置管脚 PA7 为 GPIO 输入

- 3) 配置寄存器 INT0PE7 和 INT0NE7，选择 INT0EXTI7 中断类型
- 4) 配置寄存器 INT0EN7 为 1，打开 INT0EXTI7 中断源
- 5) 配置寄存器 IEN0.0 为 1，打开 INT0 中断
- 6) 配置寄存器 IEN0.7 为 1，打开总中断
- 7) 若使用查询方式，则 4)~6) 不用配置

8 个外部管脚中断源 INT0EXTI0~INT0EXTI7 经过或逻辑组合成 INT0 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 IEN0.0 控制。

硬件实现 8 个外部管脚中断源 INT0EXTI0~INT0EXTI7 时没有优先级，软件可在 INT0 的中断服务程序中对 8 个外部管脚中断源的优先级灵活编程。

19.8 INT1 中断

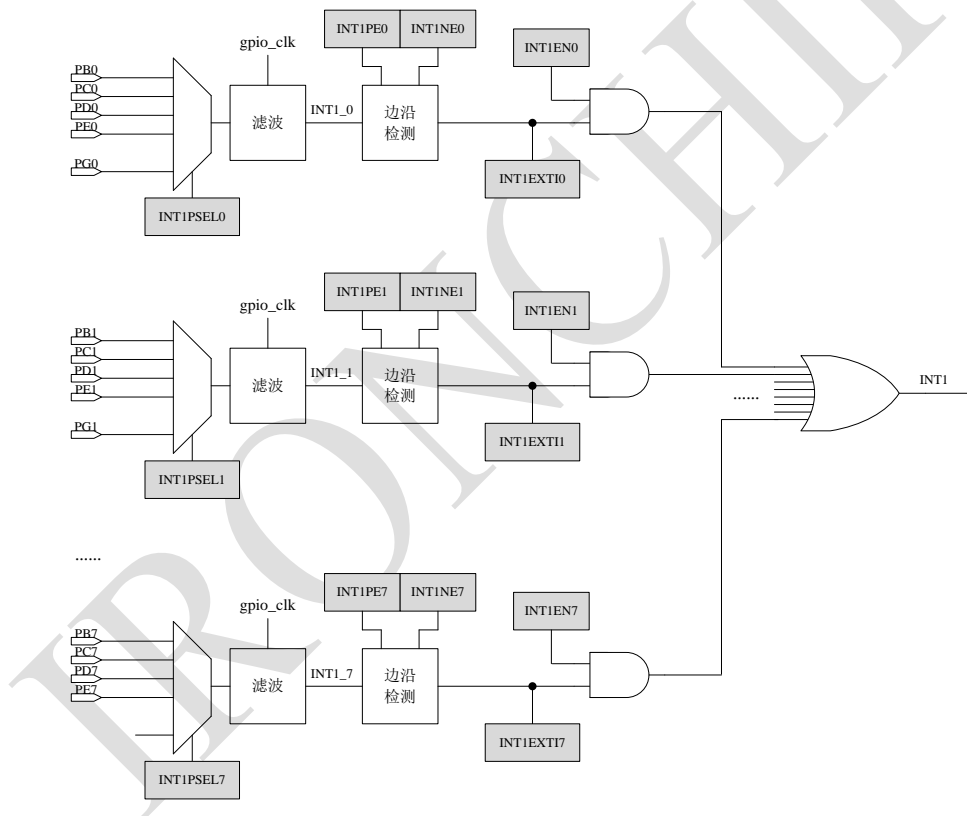


Figure 202 INT1 中断

INT1 中断包括 8 个外部管脚中断源：

- 外部管脚中断源 INT1EXTI0:
 - 1) 配置寄存器 INT1PSEL0，从 PB0, PC0, PD0, PE0 和 PG0 中选择 INT1EXTI0 的中断管脚 Px0 (其中 x=B, C, D, E, G)
 - 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 00，配置寄存器 Px0CTRL.EOUT 为 0，设置管脚 Px0 为 GPIO 输入

- 4) 配置寄存器 INT1PE0 和 INT1NE0, 选择 INT1EXTI0 中断类型
 - 5) 配置寄存器 INT1EN0 为 1, 打开 INT1EXTI0 中断源
 - 6) 配置寄存器 IEN0.2 为 1, 打开 INT1 中断
 - 7) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
- 外部管脚中断源 INT1EXTI1:
 - 1) 配置寄存器 INT1PSEL1, 从 PB1, PC1, PD1, PE1 和 PG1 中选择 INT1EXTI1 的中断管脚 Px1 (其中 x=B, C, D, E, G)
 - 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 00, 配置寄存器 Px1CTRL.EOUT 为 0, 设置管脚 Px1 为 GPIO 输入
 - 4) 配置寄存器 INT1PE1 和 INT1NE1, 选择 INT1EXTI1 中断类型
 - 5) 配置寄存器 INT1EN1 为 1, 打开 INT1EXTI1 中断源
 - 6) 配置寄存器 IEN0.2 为 1, 打开 INT1 中断
 - 7) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
 - 外部管脚中断源 INT1EXTI2:
 - 1) 配置寄存器 INT1PSEL2, 从 PB2, PC2, PD2 和 PE2 中选择 INT1EXTI2 的中断管脚 Px2 (其中 x=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 00, 配置寄存器 Px2CTRL.EOUT 为 0, 设置管脚 Px2 为 GPIO 输入
 - 4) 配置寄存器 INT1PE2 和 INT1NE2, 选择 INT1EXTI2 中断类型
 - 5) 配置寄存器 INT1EN2 为 1, 打开 INT1EXTI2 中断源
 - 6) 配置寄存器 IEN0.2 为 1, 打开 INT1 中断
 - 7) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
 - 外部管脚中断源 INT1EXTI3:
 - 1) 配置寄存器 INT1PSEL3, 从 PB3, PC3, PD3 和 PE3 中选择 INT1EXTI3 的中断管脚 Px3 (其中 x=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 00, 配置寄存器 Px3CTRL.EOUT 为 0, 设置管脚 Px3 为 GPIO 输入
 - 4) 配置寄存器 INT1PE3 和 INT1NE3, 选择 INT1EXTI3 中断类型

- 5) 配置寄存器 INT1EN3 为 1, 打开 INT1EXTI3 中断源
 - 6) 配置寄存器 IEN0.2 为 1, 打开 INT1 中断
 - 7) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
- 外部管脚中断源 INT1EXTI4:
 - 1) 配置寄存器 INT1PSEL4, 从 PB4, PC4, PD4 和 PE4 中选择 INT1EXTI4 的中断管脚 Px4 (其中 x=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 00, 配置寄存器 Px4CTRL.EOUT 为 0, 设置管脚 Px4 为 GPIO 输入
 - 4) 配置寄存器 INT1PE4 和 INT1NE4, 选择 INT1EXTI4 中断类型
 - 5) 配置寄存器 INT1EN4 为 1, 打开 INT1EXTI4 中断源
 - 6) 配置寄存器 IEN0.2 为 1, 打开 INT1 中断
 - 7) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
 - 外部管脚中断源 INT1EXTI5:
 - 1) 配置寄存器 INT1PSEL5, 从 PB5, PC5, PD5 和 PE5 中选择 INT1EXTI5 的中断管脚 Px5 (其中 x=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 00, 配置寄存器 Px5CTRL.EOUT 为 0, 设置管脚 Px5 为 GPIO 输入
 - 4) 配置寄存器 INT1PE5 和 INT1NE5, 选择 INT1EXTI5 中断类型
 - 5) 配置寄存器 INT1EN5 为 1, 打开 INT1EXTI5 中断源
 - 6) 配置寄存器 IEN0.2 为 1, 打开 INT1 中断
 - 7) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
 - 外部管脚中断源 INT1EXTI6:
 - 1) 配置寄存器 INT1PSEL6, 从 PB6, PC6, PD6 和 PE6 中选择 INT1EXTI6 的中断管脚 Px6 (其中 x=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 00, 配置寄存器 Px6CTRL.EOUT 为 0, 设置管脚 Px6 为 GPIO 输入
 - 4) 配置寄存器 INT1PE6 和 INT1NE6, 选择 INT1EXTI6 中断类型
 - 5) 配置寄存器 INT1EN6 为 1, 打开 INT1EXTI6 中断源

- 6) 配置寄存器 IEN0.2 为 1, 打开 INT1 中断
 - 7) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
- 外部管脚中断源 INT1EXTI7:
 - 1) 配置寄存器 INT1PSEL7, 从 PB7, PC7, PD7 和 PE7 中选择 INT1EXTI7 的中断管脚 Px7 (其中 x=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 00, 配置寄存器 Px7CTRL.EOUT 为 0, 设置管脚 Px7 为 GPIO 输入
 - 4) 配置寄存器 INT1PE7 和 INT1NE7, 选择 INT1EXTI7 中断类型
 - 5) 配置寄存器 INT1EN7 为 1, 打开 INT1EXTI7 中断源
 - 6) 配置寄存器 IEN0.2 为 1, 打开 INT1 中断
 - 7) 配置寄存器 IEN0.7 为 1, 打开总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置

8 个外部管脚中断源 INT1EXTI0~INT1EXTI7 经过或逻辑组合成 INT1 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 个外部管脚中断源 INT1EXTI0~INT1EXTI7 时没有优先级, 软件可在 INT1 的中断服务程序中对 8 个外部管脚中断源的优先级灵活编程。

19.9 PA0 唤醒

PA0 可将芯片从任意低功耗模式唤醒, PA0 唤醒功能参见 Figure 201。唤醒使能、输入滤波和唤醒有效沿可配置:

- 配置寄存器 WKUPEN, 使能 PA0 作为唤醒管脚;
- 配置寄存器 PA0CTRL.FQS, 设置管脚 PA0 输入滤波数;
- 配置寄存器 INTOPE0 和 INTONE0, 设置 PA0 唤醒有效沿, 可选择上升沿、下降沿或双沿, PA0 唤醒有效沿和 PA0 中断有效沿一致。

若需要支持唤醒 DEEPSLEEP 模式, PA0 管脚的滤波需要软件配置为直通模式。

19.10 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;
- 管脚数据寄存器 P6 对应的 SFRPAGE 为 2;
- 其他寄存器对应的 SFRPAGE 为 2。

Table 417 GPIO 寄存器

地址	寄存器名	类型	复位值	功能描述
0x99	PA0CTRL	W/R	0x00	PA0 数字 I/O 控制寄存器
0x9A	PA1CTRL	W/R	0x00	PA1 数字 I/O 控制寄存器
0x9B	PA2CTRL	W/R	0x00	PA2 数字 I/O 控制寄存器
0x9C	PA3CTRL	W/R	0x00	PA3 数字 I/O 控制寄存器
0x9D	PA4CTRL	W/R	0x00	PA4 数字 I/O 控制寄存器
0x9E	PA5CTRL	W/R	0x00	PA5 数字 I/O 控制寄存器
0x9F	PA6CTRL	W/R	0x00	PA6 数字 I/O 控制寄存器
0xC3	PA7CTRL	W/R	0x00	PA7 数字 I/O 控制寄存器
0xA1	PB0CTRL	W/R	0x00	PB0 数字 I/O 控制寄存器
0xA2	PB1CTRL	W/R	0x00	PB1 数字 I/O 控制寄存器
0xA3	PB2CTRL	W/R	0x00	PB2 数字 I/O 控制寄存器
0xA4	PB3CTRL	W/R	0x00	PB3 数字 I/O 控制寄存器
0xA5	PB4CTRL	W/R	0x00	PB4 数字 I/O 控制寄存器
0xA6	PB5CTRL	W/R	0x00	PB5 数字 I/O 控制寄存器
0xA7	PB6CTRL	W/R	0x00	PB6 数字 I/O 控制寄存器
0xC4	PB7CTRL	W/R	0x00	PB7 数字 I/O 控制寄存器
0xA9	PC0CTRL	W/R	0x00	PC0 数字 I/O 控制寄存器
0xAA	PC1CTRL	W/R	0x00	PC1 数字 I/O 控制寄存器
0xAB	PC2CTRL	W/R	0x00	PC2 数字 I/O 控制寄存器
0xAC	PC3CTRL	W/R	0x00	PC3 数字 I/O 控制寄存器
0xAD	PC4CTRL	W/R	0x04	PC4 数字 I/O 控制寄存器
0xAE	PC5CTRL	W/R	0x04	PC5 数字 I/O 控制寄存器
0xAF	PC6CTRL	W/R	0x00	PC6 数字 I/O 控制寄存器
0xC5	PC7CTRL	W/R	0x00	PC7 数字 I/O 控制寄存器
0xB1	PD0CTRL	W/R	0x00	PD0 数字 I/O 控制寄存器
0xB2	PD1CTRL	W/R	0x00	PD1 数字 I/O 控制寄存器
0xB3	PD2CTRL	W/R	0x00	PD2 数字 I/O 控制寄存器
0xB4	PD3CTRL	W/R	0x00	PD3 数字 I/O 控制寄存器
0xB5	PD4CTRL	W/R	0x00	PD4 数字 I/O 控制寄存器
0xB6	PD5CTRL	W/R	0x00	PD5 数字 I/O 控制寄存器

地址	寄存器名	类型	复位值	功能描述
0xB7	PD6CTRL	W/R	0x00	PD6 数字 I/O 控制寄存器
0xC6	PD7CTRL	W/R	0x00	PD7 数字 I/O 控制寄存器
0xB9	PE0CTRL	W/R	0x00	PE0 数字 I/O 控制寄存器
0xBA	PE1CTRL	W/R	0x00	PE1 数字 I/O 控制寄存器
0xBB	PE2CTRL	W/R	0x00	PE2 数字 I/O 控制寄存器
0xBC	PE3CTRL	W/R	0x00	PE3 数字 I/O 控制寄存器
0xBD	PE4CTRL	W/R	0x00	PE4 数字 I/O 控制寄存器
0xBE	PE5CTRL	W/R	0x00	PE5 数字 I/O 控制寄存器
0xBF	PE6CTRL	W/R	0x00	PE6 数字 I/O 控制寄存器
0xC7	PE7CTRL	W/R	0x00	PE7 数字 I/O 控制寄存器
0xC1	PG0CTRL	W/R	0x00	PG0 数字 I/O 控制寄存器
0xC2	PG1CTRL	W/R	0x00	PG1 数字 I/O 控制寄存器
0xC9	PASR	W/R	0xFF	PA 转换速率寄存器
0xCA	PBSR	W/R	0xFF	PB 转换速率寄存器
0xCB	PCSR	W/R	0xFF	PC 转换速率寄存器
0xCC	PDSR	W/R	0xFF	PD 转换速率寄存器
0xCD	PESR	W/R	0xFF	PE 转换速率寄存器
0xCE	PGSR	W/R	0x03	PG 转换速率寄存器
0xD1	PADR	W/R	0xFF	PA 驱动能力寄存器
0xD2	PBDR	W/R	0xFF	PB 驱动能力寄存器
0xD3	PCDR	W/R	0xFF	PC 驱动能力寄存器
0xD4	PDDR	W/R	0xFF	PD 驱动能力寄存器
0xD5	PEDR	W/R	0xFF	PE 驱动能力寄存器
0xD6	PGDR	W/R	0x03	PG 驱动能力寄存器
0xCF	AIOEN0	W/R	0x00	模拟管脚使能寄存器 0
0xD7	AIOEN1	W/R	0x00	模拟管脚使能寄存器 1
0xD9	NRSTFQS	W/R	0x00	复位管脚滤波选择寄存器
0xDB	WKUPEN	W/R	0x00	唤醒使能寄存器
0xDC	INT0PE	W/R	0x00	INT0 上升沿中断使能寄存器
0xDD	INT0NE	W/R	0x00	INT0 下降沿中断使能寄存器
0xDE	INT0EN	W/R	0x00	INT0 中断使能寄存器

地址	寄存器名	类型	复位值	功能描述
0xDF	INT0CON	W/R	0x00	INT0 中断控制寄存器
0xE1	INT1PE	W/R	0x00	INT1 上升沿中断使能寄存器
0xE2	INT1NE	W/R	0x00	INT1 下降沿中断使能寄存器
0xE3	INT1EN	W/R	0x00	INT1 中断使能寄存器
0xE4	INT1CON	W/R	0x00	INT1 中断控制寄存器
0xE5	INT1SEL0	W/R	0x00	INT1 中断管脚选择寄存器 0
0xE6	INT1SEL1	W/R	0x00	INT1 中断管脚选择寄存器 1
0xE7	INT1SEL2	W/R	0x00	INT1 中断管脚选择寄存器 2
0xE9	INT1SEL3	W/R	0x00	INT1 中断管脚选择寄存器 3
0xEA	PWMBKS	W/R	0x00	PWM 刹车选择寄存器
0xF1	PWMOSEL	W/R	0x00	PWM 特殊复用输出分组选择寄存器
0xF2	PWMOSEL0	W/R	0x00	PWM 特殊复用输出选择寄存器 0
0xF3	PWMOSEL1	W/R	0x00	PWM 特殊复用输出选择寄存器 1
0xF4	PWMOSEL2	W/R	0x00	PWM 特殊复用输出选择寄存器 2
0xF5	PWMOMUX0	W/R	0x00	PWM 特殊复用输出使能寄存器 0
0xF6	PWMOMUX1	W/R	0x00	PWM 特殊复用输出使能寄存器 1
0xF7	PWMOMUX2	W/R	0x00	PWM 特殊复用输出使能寄存器 2
0xF9	PAIE	W/R	0xFF	PA 输入缓冲使能寄存器
0xFA	PBIE	W/R	0xFF	PB 输入缓冲使能寄存器
0xFB	PCIE	W/R	0xFF	PC 输入缓冲使能寄存器
0xFC	PDIE	W/R	0xFF	PD 输入缓冲使能寄存器
0xFD	PEIE	W/R	0xFF	PE 输入缓冲使能寄存器
0xFE	PGIE	W/R	0x03	PG 输入缓冲使能寄存器
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
0xB0	P3	W/R	0xFF	PD 数据寄存器
0xC8	P4	W/R	0xFF	PE 数据寄存器
0x98	P6	R	0x03	PG 数据寄存器

19.10.1PA0 数字 I/O 控制寄存器(PA0CTRL) [0x99]

Table 418 PA0 数字 I/O 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PA0 复用功能选择: 00: P0.0(GPIO) 01: RTC_TM0 10: PWM0_ETR 11: PWM1_ETR

19.10.2 PA1 数字 I/O 控制寄存器(PA1CTRL) [0x9A]

Table 419 PA1 数字 I/O 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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PA1 复用功能选择: 00: P0.1(GPIO) 01: RTC_TM1 10: PWM0_CH0N 11: PWM1_CH0N

19.10.3PA2 数字 I/O 控制寄存器(PA2CTRL) [0x9B]

Table 420 PA2 数字 I/O 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
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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PA2 复用功能选择: 00: P0.2(GPIO) 01: I2C_SCL 10: PWM0_CH0 11: PWM1_CH1N

19.10.4PA3 数字 I/O 控制寄存器(PA3CTRL) [0x9C]

Table 421 PA3 数字 I/O 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: 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PA3 复用功能选择: 00: P0.3(GPIO)

参数名	比特位	属性	复位值	描述
				01: I2C_SDA 10: PWM0_CH1N 11: PWM1_CH2N

19.10.5PA4 数字 I/O 控制寄存器(PA4CTRL) [0x9D]

Table 422 PA4 数字 I/O 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 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PA4 复用功能选择: 00: P0.4(GPIO) 01: SPI_NSS 10: PWM0_CH1 11: PWM1_CH0

19.10.6PA5 数字 I/O 控制寄存器(PA5CTRL) [0x9E]

Table 423 PA5 数字 I/O 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PA5 复用功能选择: 00: P0.5(GPIO) 01: SPI_SCK 10: PWM0_CH2N 11: PWM1_CH1

19.10.7 PA6 数字 I/O 控制寄存器(PA6CTRL) [0x9F]

Table 424 PA6 数字 I/O 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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PA6 复用功能选择: 00: P0.6(GPIO) 01: SPI_MISO 10: PWM0_CH2 11: PWM1_CH2

19.10.8PA7 数字 I/O 控制寄存器(PA7CTRL) [0xC3]

Table 425 PA7 数字 I/O 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	0	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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PA7 复用功能选择: 00: P0.7(GPIO) 01: SPI_MOSI 10: PWM0_CH3 11: PWM1_CH3

19.10.9PB0 数字 I/O 控制寄存器(PB0CTRL) [0xA1]

Table 426 PB0 数字 I/O 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	0	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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PB0 功能选择: 00: P1.0(GPIO)

参数名	比特位	属性	复位值	描述
				01: T0 10: PWM0_ETR 11: PWM1_ETR

19.10.10 PB1 数字 I/O 控制寄存器(PB1CTRL) [0xA2]

Table 427 PB1 数字 I/O 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
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	0	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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PB1 功能选择: 00: P1.1(GPIO) 01: T1 10: PWM0_CH0N 11: PWM1_CH0N

19.10.11 PB2 数字 I/O 控制寄存器(PB2CTRL) [0xA3]

Table 428 PB2 数字 I/O 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 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PB2 功能选择: 00: P1.2(GPIO) 01: T2 10: PWM0_CH1N 11: PWM1_CH0

19.10.12PB3 数字 I/O 控制寄存器(PB3CTRL) [0xA4]

Table 429 PB3 数字 I/O 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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PB3 功能选择： 00: P1.3(GPIO) 01: PWM0_BKIN 10: PWM0_CH2N 11: PWM1_CH1N

19.10.13PB4 数字 I/O 控制寄存器(PB4CTRL) [0xA5]

Table 430 PB4 数字 I/O 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

参数名	比特位	属性	复位值	描述
FQS	5-4	W/R	0	PB4 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PB4 功能选择: 00: P1.4(GPIO) 01: SPI_NSS 10: PWM0_CH0 11: PWM1_CH1

19.10.14 PB5 数字 I/O 控制寄存器(PB5CTRL) [0xA6]

Table 431 PB5 数字 I/O 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
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	0	PB5 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PB5 功能选择: 00: P1.5(GPIO)

参数名	比特位	属性	复位值	描述
				01: SPI_SCK 10: PWM0_CH1 11: PWM1_CH2N

19.10.15 PB6 数字 I/O 控制寄存器(PB6CTRL) [0xA7]

Table 432 PB6 数字 I/O 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 1: 使能
FQS	5-4	W/R	0	PB6 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PB6 功能选择: 00: P1.6(GPIO) 01: SPI_MISO 10: PWM0_CH2 11: PWM1_CH2

19.10.16 PB7 数字 I/O 控制寄存器(PB7CTRL) [0xC4]

Table 433 PB7 数字 I/O 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PB7 功能选择: 00: P1.7(GPIO) 01: SPI_MOSI 10: PWM0_CH3 11: PWM1_CH3

19.10.17 PC0 数字 I/O 控制寄存器(PC0CTRL) [0xA9]

Table 434 PC0 数字 I/O 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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PC0 功能选择： 00: P2.0(GPIO) 01: T0_GATE 10: PWM0_CH0 11: PWM1_CH0

19.10.18 PC1 数字 I/O 控制寄存器(PC1CTRL) [0xAA]

Table 435 PC1 数字 I/O 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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PC1 功能选择: 00: P2.1(GPIO) 01: T1_GATE 10: PWM0_CH1 11: PWM1_CH1

19.10.19 PC2 数字 I/O 控制寄存器(PC2CTRL) [0xAB]

Table 436 PC2 数字 I/O 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
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PC2 功能选择: 00: P2.2(GPIO)

参数名	比特位	属性	复位值	描述
				01: T2_GATE 10: PWM0_CH2 11: PWM1_CH2

19.10.20 PC3 数字 I/O 控制寄存器(PC3CTRL) [0xAC]

Table 437 PC3 数字 I/O 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: 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PC3 功能选择: 00: P2.3(GPIO) 01: CLK_OUT 10: T0 11: PWM1_BKIN

19.10.21 PC4 数字 I/O 控制寄存器(PC4CTRL) [0xAD]

Table 438 PC4 数字 I/O 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	1	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 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PC4 功能选择: 00: P2.4(GPIO) 01: UART0_TX 10: T1 11: RTC_TM0

19.10.22 PC5 数字 I/O 控制寄存器(PC5CTRL) [0xAE]

Table 439 PC5 数字 I/O 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

参数名	比特位	属性	复位值	描述
PU	2	W/R	1	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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PC5 功能选择: 00: P2.5(GPIO) 01: UART0_RX 10: T2 11: RTC_TM1

19.10.23 PC6 数字 I/O 控制寄存器(PC6CTRL) [0xAF]

Table 440 PC6 数字 I/O 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	0	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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PC6 功能选择: 00: P2.6(GPIO) 01: UART1_TX 10: I2C_SCL 11: PWM0_BKIN

19.10.24 PC7 数字 I/O 控制寄存器(PC7CTRL) [0xC5]

Table 441 PC7 数字 I/O 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PC7 功能选择: 00: P2.7(GPIO)

参数名	比特位	属性	复位值	描述
				01: UART1_RX 10: I2C_SDA 11: PWM1_BKIN

19.10.25 PD0 数字 I/O 控制寄存器(PD0CTRL) [0xB1]

Table 442 PD0 数字 I/O 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	0	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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PD0 功能选择: 00: P3.0(GPIO) 01: T0 10: PWM0_CH0N 11: PWM1_CH0N

19.10.26 PD1 数字 I/O 控制寄存器(PD1CTRL) [0xB2]

Table 443 PD1 数字 I/O 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
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	0	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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PD1 功能选择: 00: P3.1(GPIO) 01: T1 10: PWM0_CH1N 11: PWM1_CH1N

19.10.27 PD2 数字 I/O 控制寄存器(PD2CTRL) [0xB3]

Table 444 PD2 数字 I/O 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

参数名	比特位	属性	复位值	描述
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 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PD2 功能选择： 00: P3.2(GPIO) 01: T2 10: PWM0_CH2N 11: PWM1_CH2N

19.10.28 PD3 数字 I/O 控制寄存器(PD3CTRL) [0xB4]

Table 445 PD3 数字 I/O 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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PD3 功能选择： 00: P3.3(GPIO) 01: T0_GATE 10: PWM0_CH0 11: PWM1_CH0

19.10.29 PD4 数字 I/O 控制寄存器(PD4CTRL) [0xB5]

Table 446 PD4 数字 I/O 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PD4 功能选择： 00: P3.4(GPIO)

参数名	比特位	属性	复位值	描述
				01: T1_GATE 10: PWM0_CH1 11: PWM1_CH1

19.10.30 PD5 数字 I/O 控制寄存器(PD5CTRL) [0xB6]

Table 447 PD5 数字 I/O 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
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PD5 功能选择: 00: P3.5(GPIO) 01: T2_GATE 10: PWM0_CH2 11: PWM1_CH2

19.10.31 PD6 数字 I/O 控制寄存器(PD6CTRL) [0xB7]

Table 448 PD6 数字 I/O 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 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PD6 功能选择: 00: P3.6(GPIO) 01: PWM0_BKIN 10: PWM0_CH3 11: PWM1_CH3

19.10.32 PD7 数字 I/O 控制寄存器(PD7CTRL) [0xC6]

Table 449 PD7 数字 I/O 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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PD7 功能选择： 00: P3.7(GPIO) 01: PWM1_BKIN 10: PWM0_ETR 11: PWM1_ETR

19.10.33PE0 数字 I/O 控制寄存器(PEOCTRL) [0xB9]

Table 450 PE0 数字 I/O 控制寄存器(PEOCTRL)

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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PE0 功能选择: 00: P4.0(GPIO) 01: UART0_TX 10: PWM0_ETR 11: PWM1_ETR

19.10.34 PE1 数字 I/O 控制寄存器(PE1CTRL) [0xBA]

Table 451 PE1 数字 I/O 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
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PE1 功能选择: 00: P4.1(GPIO)

参数名	比特位	属性	复位值	描述
				01: UART0_RX 10: PWM0_CH0N 11: PWM1_CH0N

19.10.35 PE2 数字 I/O 控制寄存器(PE2CTRL) [0xBB]

Table 452 PE2 数字 I/O 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 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PE2 功能选择: 00: P4.2(GPIO) 01: UART1_TX 10: PWM0_CH0 11: PWM1_CH1N

19.10.36 PE3 数字 I/O 控制寄存器(PE3CTRL) [0xBC]

Table 453 PE3 数字 I/O 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PE3 功能选择: 00: P4.3(GPIO) 01: UART1_RX 10: PWM0_CH1N 11: PWM1_CH2N

19.10.37 PE4 数字 I/O 控制寄存器(PE4CTRL) [0xBD]

Table 454 PE4 数字 I/O 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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PE4 功能选择: 00: P4.4(GPIO) 01: SPI_NSS 10: PWM0_CH1 11: PWM1_CH0

19.10.38 PE5 数字 I/O 控制寄存器(PE5CTRL) [0xBE]

Table 455 PE5 数字 I/O 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
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

参数名	比特位	属性	复位值	描述
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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PE5 功能选择: 00: P4.5(GPIO) 01: SPI_SCK 10: PWM0_CH2N 11: PWM1_CH1

19.10.39 PE6 数字 I/O 控制寄存器(PE6CTRL) [0xBF]

Table 456 PE6 数字 I/O 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 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PE6 功能选择: 00: P4.6(GPIO)

参数名	比特位	属性	复位值	描述
				01: SPI_MISO 10: PWM0_CH2 11: PWM1_CH2

19.10.40 PE7 数字 I/O 控制寄存器(PE7CTRL) [0xC7]

Table 457 PE7 数字 I/O 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 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PE7 功能选择: 00: P4.7(GPIO) 01: SPI_MOSI 10: PWM0_CH3 11: PWM1_CH3

19.10.41 PG0 数字 I/O 控制寄存器(PG0CTRL) [0xC1]

Table 458 PG0 数字 I/O 控制寄存器(PG0CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PG0 开漏使能: 0: 不使能 1: 使能
Reserved	1	R	0	保留
PU	2	W/R	0	PG0 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PG0 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PG0 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PG0 功能选择: 00: OSC_IN 01: P6.0(GPIO 输入) 10: I2C_SCL 11: PWM0_BKIN

19.10.42 PG1 数字 I/O 控制寄存器(PG1CTRL) [0xC2]

Table 459 PG1 数字 I/O 控制寄存器(PG1CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PG1 开漏使能: 0: 不使能 1: 使能
Reserved	1	R	0	保留
PU	2	W/R	0	PG1 上拉使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
PD	3	W/R	0	PG1 下拉使能： 0: 不使能 1: 使能
FQS	5-4	W/R	0	PG1 输入滤波选择： 00: 仅同步 01: 可滤除小于 2 个 gpio_clk 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
MUX	7-6	W/R	0	PG1 功能选择： 00: OSC_OUT 01: P6.1(GPIO 输入) 10: I2C_SDA 11: PWM1_BKIN

19.10.43PA 转换速率寄存器(PASR) [0xC9]

Table 460 PA 转换速率寄存器(PASR)

参数名	比特位	属性	复位值	描述
PA0SR	0	W/R	1	PA0 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PA1SR	1	W/R	1	PA1 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PA2SR	2	W/R	1	PA2 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PA3SR	3	W/R	1	PA3 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PA4SR	4	W/R	1	PA4 转换速率控制位： 0: 转换速率快 1: 转换速率慢

参数名	比特位	属性	复位值	描述
PA5SR	5	W/R	1	PA5 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PA6SR	6	W/R	1	PA6 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PA7SR	7	W/R	1	PA7 转换速率控制位： 0: 转换速率快 1: 转换速率慢

19.10.44PB 转换速率寄存器(PBSR) [0xCA]

Table 461 PB 转换速率寄存器(PBSR)

参数名	比特位	属性	复位值	描述
PB0SR	0	W/R	1	PB0 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PB1SR	1	W/R	1	PB1 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PB2SR	2	W/R	1	PB2 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PB3SR	3	W/R	1	PB3 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PB4SR	4	W/R	1	PB4 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PB5SR	5	W/R	1	PB5 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PB6SR	6	W/R	1	PB6 转换速率控制位：

参数名	比特位	属性	复位值	描述
				0: 转换速率快 1: 转换速率慢
PB7SR	7	W/R	1	PB7 转换速率控制位: 0: 转换速率快 1: 转换速率慢

19.10.45PC 转换速率寄存器(PCSR) [0xCB]

Table 462 PC 转换速率寄存器(PCSR)

参数名	比特位	属性	复位值	描述
PC0SR	0	W/R	1	PC0 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PC1SR	1	W/R	1	PC1 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PC2SR	2	W/R	1	PC2 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PC3SR	3	W/R	1	PC3 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PC4SR	4	W/R	1	PC4 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PC5SR	5	W/R	1	PC5 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PC6SR	6	W/R	1	PC6 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PC7SR	7	W/R	1	PC7 转换速率控制位: 0: 转换速率快

参数名	比特位	属性	复位值	描述
				1: 转换速率慢

19.10.46PD 转换速率寄存器(PDSR) [0xCC]

Table 463 PD 转换速率寄存器(PDSR)

参数名	比特位	属性	复位值	描述
PD0SR	0	W/R	1	PD0 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PD1SR	1	W/R	1	PD1 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PD2SR	2	W/R	1	PD2 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PD3SR	3	W/R	1	PD3 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PD4SR	4	W/R	1	PD4 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PD5SR	5	W/R	1	PD5 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PD6SR	6	W/R	1	PD6 转换速率控制位: 0: 转换速率快 1: 转换速率慢
PD7SR	7	W/R	1	PD7 转换速率控制位: 0: 转换速率快 1: 转换速率慢

19.10.47PE 转换速率寄存器(PESR) [0xCD]

Table 464 PE 转换速率寄存器(PESR)

参数名	比特位	属性	复位值	描述
PE0SR	0	W/R	1	PE0 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PE1SR	1	W/R	1	PE1 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PE2SR	2	W/R	1	PE2 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PE3SR	3	W/R	1	PE3 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PE4SR	4	W/R	1	PE4 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PE5SR	5	W/R	1	PE5 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PE6SR	6	W/R	1	PE6 转换速率控制位： 0: 转换速率快 1: 转换速率慢
PE7SR	7	W/R	1	PE7 转换速率控制位： 0: 转换速率快 1: 转换速率慢

19.10.48PG 转换速率寄存器(PGSR) [0xCE]

Table 465 PG 转换速率寄存器(PGSR)

参数名	比特位	属性	复位值	描述
PG0SR	0	W/R	1	PG0 转换速率控制位： 0: 转换速率快 1: 转换速率慢

参数名	比特位	属性	复位值	描述
PG1SR	1	W/R	1	PG1 转换速率控制位： 0：转换速率快 1：转换速率慢
Reserved	7-2	R	0	保留

19.10.49PA 驱动能力寄存器(PADR) [0xD1]

Table 466 PA 驱动能力寄存器(PADR)

参数名	比特位	属性	复位值	描述
PA0DR	0	W/R	1	PA0 驱动能力控制位： 0：驱动能力高 1：驱动能力低
PA1DR	1	W/R	1	PA1 驱动能力控制位： 0：驱动能力高 1：驱动能力低
PA2DR	2	W/R	1	PA2 驱动能力控制位： 0：驱动能力高 1：驱动能力低
PA3DR	3	W/R	1	PA3 驱动能力控制位： 0：驱动能力高 1：驱动能力低
PA4DR	4	W/R	1	PA4 驱动能力控制位： 0：驱动能力高 1：驱动能力低
PA5DR	5	W/R	1	PA5 驱动能力控制位： 0：驱动能力高 1：驱动能力低
PA6DR	6	W/R	1	PA6 驱动能力控制位： 0：驱动能力高 1：驱动能力低
PA7DR	7	W/R	1	PA7 驱动能力控制位： 0：驱动能力高 1：驱动能力低

19.10.50 PB 驱动能力寄存器(PBDR) [0xD2]

Table 467 PB 驱动能力寄存器(PBDR)

参数名	比特位	属性	复位值	描述
PB0DR	0	W/R	1	PB0 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PB1DR	1	W/R	1	PB1 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PB2DR	2	W/R	1	PB2 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PB3DR	3	W/R	1	PB3 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PB4DR	4	W/R	1	PB4 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PB5DR	5	W/R	1	PB5 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PB6DR	6	W/R	1	PB6 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PB7DR	7	W/R	1	PB7 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低

19.10.51 PC 驱动能力寄存器(PCDR) [0xD3]

Table 468 PC 驱动能力寄存器(PCDR)

参数名	比特位	属性	复位值	描述
PC0DR	0	W/R	1	PC0 驱动能力控制位： 0: 驱动能力高

参数名	比特位	属性	复位值	描述
				1: 驱动能力低
PC1DR	1	W/R	1	PC1 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PC2DR	2	W/R	1	PC2 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PC3DR	3	W/R	1	PC3 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PC4DR	4	W/R	1	PC4 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PC5DR	5	W/R	1	PC5 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PC6DR	6	W/R	1	PC6 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PC7DR	7	W/R	1	PC7 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低

19.10.52PD 驱动能力寄存器(PDDR) [0xD4]

Table 469 PD 驱动能力寄存器(PDDR)

参数名	比特位	属性	复位值	描述
PD0DR	0	W/R	1	PD0 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PD1DR	1	W/R	1	PD1 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低

参数名	比特位	属性	复位值	描述
PD2DR	2	W/R	1	PD2 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PD3DR	3	W/R	1	PD3 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PD4DR	4	W/R	1	PD4 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PD5DR	5	W/R	1	PD5 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PD6DR	6	W/R	1	PD6 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PD7DR	7	W/R	1	PD7 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低

19.10.53PE 驱动能力寄存器(PEDR) [0xD5]

Table 470 PE 驱动能力寄存器(PEDR)

参数名	比特位	属性	复位值	描述
PE0DR	0	W/R	1	PE0 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PE1DR	1	W/R	1	PE1 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PE2DR	2	W/R	1	PE2 驱动能力控制位： 0: 驱动能力高 1: 驱动能力低
PE3DR	3	W/R	1	PE3 驱动能力控制位：

参数名	比特位	属性	复位值	描述
				0: 驱动能力高 1: 驱动能力低
PE4DR	4	W/R	1	PE4 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PE5DR	5	W/R	1	PE5 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PE6DR	6	W/R	1	PE6 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PE7DR	7	W/R	1	PE7 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低

19.10.54 PG 驱动能力寄存器(PGDR) [0xD6]

Table 471 PG 驱动能力寄存器(PGDR)

参数名	比特位	属性	复位值	描述
PG0DR	0	W/R	1	PG0 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
PG1DR	1	W/R	1	PG1 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
Reserved	7-3	R	0	保留

19.10.55 模拟管脚使能寄存器 0(AIOEN0) [0xCF]

Table 472 模拟管脚使能寄存器 0(AIOEN0)

参数名	比特位	属性	复位值	描述
PA0AIOEN	0	W/R	0	PA0 作为模拟 IO 使能: 0: 不使能 1: 使能 PA0 作为模拟 IO 使用时, 此管脚的 IE,

参数名	比特位	属性	复位值	描述
				OE, PD, PU, OD 由硬件自动强制清零
PA1AIOEN	1	W/R	0	PA1 作为模拟 IO 使能: 0: 不使能 1: 使能 PA1 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PA2AIOEN	2	W/R	0	PA2 作为模拟 IO 使能: 0: 不使能 1: 使能 PA2 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PA3AIOEN	3	W/R	0	PA3 作为模拟 IO 使能: 0: 不使能 1: 使能 PA3 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PA4AIOEN	4	W/R	0	PA4 作为模拟 IO 使能: 0: 不使能 1: 使能 PA4 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PA5AIOEN	5	W/R	0	PA5 作为模拟 IO 使能: 0: 不使能 1: 使能 PA5 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PA6AIOEN	6	W/R	0	PA6 作为模拟 IO 使能: 0: 不使能 1: 使能 PA6 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PA7AIOEN	7	W/R	0	PA7 作为模拟 IO 使能:

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能 PA7 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零

19.10.56 模拟管脚使能寄存器 1(AIOEN1) [0xD7]

Table 473 模拟管脚使能寄存器 1(AIOEN1)

参数名	比特位	属性	复位值	描述
PC0AIOEN	0	W/R	0	PC0 作为模拟 IO 使能: 0: 不使能 1: 使能 PC0 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PC1AIOEN	1	W/R	0	PC1 作为模拟 IO 使能: 0: 不使能 1: 使能 PC1 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PC2AIOEN	2	W/R	0	PC2 作为模拟 IO 使能: 0: 不使能 1: 使能 PC2 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PD3AIOEN	3	W/R	0	PD3 作为模拟 IO 使能: 0: 不使能 1: 使能 PD3 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PD4AIOEN	4	W/R	0	PD4 作为模拟 IO 使能: 0: 不使能 1: 使能 PD4 作为模拟 IO 使用时, 此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零

参数名	比特位	属性	复位值	描述
PD5AIOEN	5	W/R	0	PD5 作为模拟 IO 使能： 0: 不使能 1: 使能 PD5 作为模拟 IO 使用时，此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PD6AIOEN	6	W/R	0	PD6 作为模拟 IO 使能： 0: 不使能 1: 使能 PD6 作为模拟 IO 使用时，此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零
PD7AIOEN	7	W/R	0	PD7 作为模拟 IO 使能： 0: 不使能 1: 使能 PD7 作为模拟 IO 使用时，此管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零

19.10.57 复位管脚滤波选择寄存器(NRSTFQS) [0xD9]

Table 474 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 周期的毛刺 10: 可滤除小于 5 个 gpio_clk 周期的毛刺 11: 直通
Reserved	7-2	R	0	保留

19.10.58 唤醒使能寄存器(WKUPEN) [0xDB]

Table 475 唤醒使能寄存器(WKUPEN)

参数名	比特位	属性	复位值	描述
WKUPEN	0	W/R	0	PA0 作为唤醒管脚使能： 0: 不使能 1: 使能
Reserved	7-1	R	0	保留

19.10.59INT0 上升沿中断使能寄存器(INT0PE) [0xDC]

Table 476 INT0 上升沿中断使能寄存器(INT0PE)

参数名	比特位	属性	复位值	描述
INT0PE0	0	W/R	0	INT0EXTI0 上升沿中断使能: 0: 不使能 1: 使能
INT0PE1	1	W/R	0	INT0EXTI1 上升沿中断使能: 0: 不使能 1: 使能
INT0PE2	2	W/R	0	INT0EXTI2 上升沿中断使能: 0: 不使能 1: 使能
INT0PE3	3	W/R	0	INT0EXTI3 上升沿中断使能: 0: 不使能 1: 使能
INT0PE4	4	W/R	0	INT0EXTI4 上升沿中断使能: 0: 不使能 1: 使能
INT0PE5	5	W/R	0	INT0EXTI5 上升沿中断使能: 0: 不使能 1: 使能
INT0PE6	6	W/R	0	INT0EXTI6 上升沿中断使能: 0: 不使能 1: 使能
INT0PE7	7	W/R	0	INT0EXTI7 上升沿中断使能: 0: 不使能 1: 使能

19.10.60INT0 下降沿中断使能寄存器(INT0NE) [0xDD]

Table 477 INT0 下降沿中断使能寄存器(INT0NE)

参数名	比特位	属性	复位值	描述
INT0NE0	0	W/R	0	INT0EXTI0 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
INTONE1	1	W/R	0	INT0EXTI1 下降沿中断使能: 0: 不使能 1: 使能
INTONE2	2	W/R	0	INT0EXTI2 下降沿中断使能: 0: 不使能 1: 使能
INTONE3	3	W/R	0	INT0EXTI3 下降沿中断使能: 0: 不使能 1: 使能
INTONE4	4	W/R	0	INT0EXTI4 下降沿中断使能: 0: 不使能 1: 使能
INTONE5	5	W/R	0	INT0EXTI5 下降沿中断使能: 0: 不使能 1: 使能
INTONE6	6	W/R	0	INT0EXTI6 下降沿中断使能: 0: 不使能 1: 使能
INTONE7	7	W/R	0	INT0EXTI7 下降沿中断使能: 0: 不使能 1: 使能

19.10.61 INT0 中断使能寄存器(INT0EN) [0xDE]

Table 478 INT0 中断使能寄存器(INT0EN)

参数名	比特位	属性	复位值	描述
INT0EN0	0	W/R	0	INT0EXTI0 中断使能: 0: 不使能 1: 使能
INT0EN1	1	W/R	0	INT0EXTI1 中断使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
INT0EN2	2	W/R	0	INT0EXTI2 中断使能： 0: 不使能 1: 使能
INT0EN3	3	W/R	0	INT0EXTI3 中断使能： 0: 不使能 1: 使能
INT0EN4	4	W/R	0	INT0EXTI4 中断使能： 0: 不使能 1: 使能
INT0EN5	5	W/R	0	INT0EXTI5 中断使能： 0: 不使能 1: 使能
INT0EN6	6	W/R	0	INT0EXTI6 中断使能： 0: 不使能 1: 使能
INT0EN7	7	W/R	0	INT0EXTI7 中断使能： 0: 不使能 1: 使能

19.10.62 INT0 中断控制寄存器(INT0CON) [0xDF]

Table 479 INT0 中断控制寄存器(INT0CON)

参数名	比特位	属性	复位值	描述
INT0EXTI0	0	W/R	0	INT0EXTI0 中断标志位： <ul style="list-style-type: none"> ● 当有 INT0EXTI0 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT0EXTI1	1	W/R	0	INT0EXTI1 中断标志位： <ul style="list-style-type: none"> ● 当有 INT0EXTI1 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断

参数名	比特位	属性	复位值	描述
INT0EXTI2	2	W/R	0	INT0EXTI2 中断标志位： <ul style="list-style-type: none"> ● 当有 INT0EXTI2 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT0EXTI3	3	W/R	0	INT0EXTI3 中断标志位： <ul style="list-style-type: none"> ● 当有 INT0EXTI3 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT0EXTI4	4	W/R	0	INT0EXTI4 中断标志位： <ul style="list-style-type: none"> ● 当有 INT0EXTI4 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT0EXTI5	5	W/R	0	INT0EXTI5 中断标志位： <ul style="list-style-type: none"> ● 当有 INT0EXTI5 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT0EXTI6	6	W/R	0	INT0EXTI6 中断标志位： <ul style="list-style-type: none"> ● 当有 INT0EXTI6 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT0EXTI7	7	W/R	0	INT0EXTI7 中断标志位： <ul style="list-style-type: none"> ● 当有 INT0EXTI7 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断

19.10.63INT1 上升沿中断使能寄存器(INT1PE) [0xE1]

Table 480 INT1 上升沿中断使能寄存器(INT1PE)

参数名	比特位	属性	复位值	描述
INT1PE0	0	W/R	0	INT1EXTI0 上升沿中断使能: 0: 不使能 1: 使能
INT1PE1	1	W/R	0	INT1EXTI1 上升沿中断使能: 0: 不使能 1: 使能
INT1PE2	2	W/R	0	INT1EXTI2 上升沿中断使能: 0: 不使能 1: 使能
INT1PE3	3	W/R	0	INT1EXTI3 上升沿中断使能: 0: 不使能 1: 使能
INT1PE4	4	W/R	0	INT1EXTI4 上升沿中断使能: 0: 不使能 1: 使能
INT1PE5	5	W/R	0	INT1EXTI5 上升沿中断使能: 0: 不使能 1: 使能
INT1PE6	6	W/R	0	INT1EXTI6 上升沿中断使能: 0: 不使能 1: 使能
INT1PE7	7	W/R	0	INT1EXTI7 上升沿中断使能: 0: 不使能 1: 使能

19.10.64 INT1 下降沿中断使能寄存器(INT1NE) [0xE2]

Table 481 INT1 下降沿中断使能寄存器(INT1NE)

参数名	比特位	属性	复位值	描述
INT1NE0	0	W/R	0	INT1EXTI0 下降沿中断使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
INT1NE1	1	W/R	0	INT1EXTI1 下降沿中断使能： 0: 不使能 1: 使能
INT1NE2	2	W/R	0	INT1EXTI2 下降沿中断使能： 0: 不使能 1: 使能
INT1NE3	3	W/R	0	INT1EXTI3 下降沿中断使能： 0: 不使能 1: 使能
INT1NE4	4	W/R	0	INT1EXTI4 下降沿中断使能： 0: 不使能 1: 使能
INT1NE5	5	W/R	0	INT1EXTI5 下降沿中断使能： 0: 不使能 1: 使能
INT1NE6	6	W/R	0	INT1EXTI6 下降沿中断使能： 0: 不使能 1: 使能
INT1NE7	7	W/R	0	INT1EXTI7 下降沿中断使能： 0: 不使能 1: 使能

19.10.65 INT1 中断使能寄存器(INT1EN) [0xE3]

Table 482 INT1 中断使能寄存器(INT1EN)

参数名	比特位	属性	复位值	描述
INT1EN0	0	W/R	0	INT1EXTI0 中断使能： 0: 不使能 1: 使能
INT1EN1	1	W/R	0	INT1EXTI1 中断使能： 0: 不使能 1: 使能
INT1EN2	2	W/R	0	INT1EXTI2 中断使能：

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能
INT1EN3	3	W/R	0	INT1EXTI3 中断使能: 0: 不使能 1: 使能
INT1EN4	4	W/R	0	INT1EXTI4 中断使能: 0: 不使能 1: 使能
INT1EN5	5	W/R	0	INT1EXTI5 中断使能: 0: 不使能 1: 使能
INT1EN6	6	W/R	0	INT1EXTI6 中断使能: 0: 不使能 1: 使能
INT1EN7	7	W/R	0	INT1EXTI7 中断使能: 0: 不使能 1: 使能

19.10.66 INT1 中断控制寄存器(INT1CON) [0xE4]

Table 483 INT1 中断控制寄存器(INT1CON)

参数名	比特位	属性	复位值	描述
INT1EXTI0	0	W/R	0	INT1EXTI0 中断标志位: <ul style="list-style-type: none"> ● 当有 INT1EXTI0 中断申请时, 该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT1EXTI1	1	W/R	0	INT1EXTI1 中断标志位: <ul style="list-style-type: none"> ● 当有 INT1EXTI1 中断申请时, 该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT1EXTI2	2	W/R	0	INT1EXTI2 中断标志位:

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> ● 当有 INT1EXTI2 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT1EXTI3	3	W/R	0	INT1EXTI3 中断标志位： <ul style="list-style-type: none"> ● 当有 INT1EXTI3 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT1EXTI4	4	W/R	0	INT1EXTI4 中断标志位： <ul style="list-style-type: none"> ● 当有 INT1EXTI4 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT1EXTI5	5	W/R	0	INT1EXTI5 中断标志位： <ul style="list-style-type: none"> ● 当有 INT1EXTI5 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT1EXTI6	6	W/R	0	INT1EXTI6 中断标志位： <ul style="list-style-type: none"> ● 当有 INT1EXTI6 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
INT1EXTI7	7	W/R	0	INT1EXTI7 中断标志位： <ul style="list-style-type: none"> ● 当有 INT1EXTI7 中断申请时，该位置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断

19.10.67INT1 中断管脚选择寄存器 0(INT1SEL0) [0xE5]

Table 484 INT1 中断管脚选择寄存器 0(INT1SEL0)

参数名	比特位	属性	复位值	描述
INT1PSEL0	2-0	W/R	0	INT1EXTI0 中断管脚选择: 000: PB0 001: PC0 010: PD0 011: PE0 101: PG0 其他: 保留
Reserved	3	R	0	保留
INT1PSEL1	6-4	W/R	0	INT1EXTI1 中断管脚选择: 000: PB1 001: PC1 010: PD1 011: PE1 101: PG1 其他: 保留
Reserved	7	R	0	保留

19.10.68INT1 中断管脚选择寄存器 1(INT1SEL1) [0xE6]

Table 485 INT1 中断管脚选择寄存器 1(INT1SEL1)

参数名	比特位	属性	复位值	描述
INT1PSEL2	2-0	W/R	0	INT1EXTI2 中断管脚选择: 000: PB2 001: PC2 010: PD2 011: PE2 其他: 保留
Reserved	3	R	0	保留
INT1PSEL3	6-4	W/R	0	INT1EXTI3 中断管脚选择: 000: PB3 001: PC3 010: PD3 011: PE3

参数名	比特位	属性	复位值	描述
				其他：保留
Reserved	7	R	0	保留

19.10.69INT1 中断管脚选择寄存器 2(INT1SEL2) [0xE7]

Table 486 INT1 中断管脚选择寄存器 2(INT1SEL2)

参数名	比特位	属性	复位值	描述
INT1PSEL4	2-0	W/R	0	INT1EXTI4 中断管脚选择： 000: PB4 001: PC4 010: PD4 011: PE4 其他：保留
Reserved	3	R	0	保留
INT1PSEL5	6-4	W/R	0	INT1EXTI5 中断管脚选择： 000: PB5 001: PC5 010: PD5 011: PE5 其他：保留
Reserved	7	R	0	保留

19.10.70INT1 中断管脚选择寄存器 3(INT1SEL3) [0xE9]

Table 487 INT1 中断管脚选择寄存器 3(INT1SEL3)

参数名	比特位	属性	复位值	描述
INT1PSEL6	2-0	W/R	0	INT1EXTI6 中断管脚选择： 000: PB6 001: PC6 010: PD6 011: PE6 其他：保留
Reserved	3	R	0	保留
INT1PSEL7	6-4	W/R	0	INT1EXTI7 中断管脚选择： 000: PB7

参数名	比特位	属性	复位值	描述
				001: PC7 010: PD7 011: PE7 其他: 保留
Reserved	7	R	0	保留

19.10.71 PWM 刹车选择寄存器(PWMBKS) [0xEA]

Table 488 PWM 刹车选择寄存器(PWMBKS)

参数名	比特位	属性	复位值	描述
PWM0BKS	1-0	W/R	0	PWM0 刹车信号选择: 00: PWM0_BKIN 01: PWM1_BKIN 其他: 保留
PWM1BKS	3-2	W/R	0	PWM1 刹车信号选择: 00: PWM1_BKIN 01: PWM0_BKIN 其他: 保留
Reserved	7-6	R	0	保留

19.10.72 PWM 特殊复用输出分组选择寄存器(PWMOSEL) [0xF1]

Table 489 PWM 特殊复用输出分组选择寄存器(PWMOSEL)

参数名	比特位	属性	复位值	描述
PWMO0SEL	2-0	W/R	0	PWM 特殊复用输出 PWMO0 选择: 000: PWM0_CH0 001: PWM0_CH0N 010: PWM1_CH0 011: PWM1_CH0N 其他: 保留
Reserved	3	R	0	保留
PWMO1SEL	6-4	W/R	0	PWM 特殊复用输出 PWMO1 选择: 000: PWM0_CH0 001: PWM0_CH0N 010: PWM1_CH0

参数名	比特位	属性	复位值	描述
				011: PWM1_CH0N 保留: 保留
Reserved	7	R	0	保留

19.10.73 PWM 特殊复用输出选择寄存器 0(PWMOSEL0) [0xF2]

Table 490 PWM 特殊复用输出选择寄存器 0(PWMOSEL0)

参数名	比特位	属性	复位值	描述
PB0PWMOS	0	W/R	0	PB0 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PB1PWMOS	1	W/R	0	PB1 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PB2PWMOS	2	W/R	0	PB2 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PB3PWMOS	3	W/R	0	PB3 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PB4PWMOS	4	W/R	0	PB4 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PB5PWMOS	5	W/R	0	PB5 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PB6PWMOS	6	W/R	0	PB6 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PB7PWMOS	7	W/R	0	PB7 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1

19.10.74 PWM 特殊复用输出选择寄存器 1(PWMOSEL1) [0xF3]

Table 491 PWM 特殊复用输出选择寄存器 1(PWMOSEL1)

参数名	比特位	属性	复位值	描述
PD0PWMOS	0	W/R	0	PD0 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PD1PWMOS	1	W/R	0	PD1 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PD2PWMOS	2	W/R	0	PD2 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PC3PWMOS	3	W/R	0	PC3 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PC4PWMOS	4	W/R	0	PC4 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PC5PWMOS	5	W/R	0	PC5 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PC6PWMOS	6	W/R	0	PC6 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1
PC7PWMOS	7	W/R	0	PC7 管脚 PWM 特殊复用输出源选择: 0: PWMO0 1: PWMO1

19.10.75 PWM 特殊复用输出选择寄存器 2(PWMOSEL2) [0xF4]

Table 492 PWM 特殊复用输出选择寄存器 2(PWMOSEL2)

参数名	比特位	属性	复位值	描述
PE0PWMOS	0	W/R	0	PE0 管脚 PWM 特殊复用输出源选择: 0: PWMO0

参数名	比特位	属性	复位值	描述
				1: PWM01
PE1PWMOS	1	W/R	0	PE1 管脚 PWM 特殊复用输出源选择: 0: PWM00 1: PWM01
PE2PWMOS	2	W/R	0	PE2 管脚 PWM 特殊复用输出源选择: 0: PWM00 1: PWM01
PE3PWMOS	3	W/R	0	PE3 管脚 PWM 特殊复用输出源选择: 0: PWM00 1: PWM01
PE4PWMOS	4	W/R	0	PE4 管脚 PWM 特殊复用输出源选择: 0: PWM00 1: PWM01
PE5PWMOS	5	W/R	0	PE5 管脚 PWM 特殊复用输出源选择: 0: PWM00 1: PWM01
PE6PWMOS	6	W/R	0	PE6 管脚 PWM 特殊复用输出源选择: 0: PWM00 1: PWM01
PE7PWMOS	7	W/R	0	PE7 管脚 PWM 特殊复用输出源选择: 0: PWM00 1: PWM01

19.10.76 PWM 特殊复用输出使能寄存器 0(PWMOMUX0) [0xF5]

Table 493 PWM 特殊复用输出使能寄存器 0(PWMOMUX0)

参数名	比特位	属性	复位值	描述
PB0PWMOE	0	W/R	0	PB0 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出
PB1PWMOE	1	W/R	0	PB1 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出

参数名	比特位	属性	复位值	描述
PB2PWMOE	2	W/R	0	PB2 管脚 PWM 特殊复用输出使能： 0: 正常功能 1: PWM 特殊复用输出
PB3PWMOE	3	W/R	0	PB3 管脚 PWM 特殊复用输出使能： 0: 正常功能 1: PWM 特殊复用输出
PB4PWMOE	4	W/R	0	PB4 管脚 PWM 特殊复用输出使能： 0: 正常功能 1: PWM 特殊复用输出
PB5PWMOE	5	W/R	0	PB5 管脚 PWM 特殊复用输出使能： 0: 正常功能 1: PWM 特殊复用输出
PB6PWMOE	6	W/R	0	PB6 管脚 PWM 输出源选择： 0: 正常功能 1: PWM 特殊复用输出
PB7PWMOE	7	W/R	0	PB7 管脚 PWM 特殊复用输出使能： 0: 正常功能 1: PWM 特殊复用输出

19.10.77 PWM 特殊复用输出使能寄存器 1(PWMOMUX1) [0xF6]

Table 494 PWM 特殊复用输出使能寄存器 1(PWMOMUX1)

参数名	比特位	属性	复位值	描述
PD0PWMOE	0	W/R	0	PD0 管脚 PWM 特殊复用输出使能： 0: 正常功能 1: PWM 特殊复用输出
PD1PWMOE	1	W/R	0	PD1 管脚 PWM 特殊复用输出使能： 0: 正常功能 1: PWM 特殊复用输出
PD2PWMOE	2	W/R	0	PD2 管脚 PWM 特殊复用输出使能： 0: 正常功能 1: PWM 特殊复用输出
PC3PWMOE	3	W/R	0	PC3 管脚 PWM 特殊复用输出使能：

参数名	比特位	属性	复位值	描述
				0: 正常功能 1: PWM 特殊复用输出
PC4PWMOE	4	W/R	0	PC4 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出
PC5PWMOE	5	W/R	0	PC5 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出
PC6PWMOE	6	W/R	0	PC6 管脚 PWM 输出源选择: 0: 正常功能 1: PWM 特殊复用输出
PC7PWMOE	7	W/R	0	PC7 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出

19.10.78 PWM 特殊复用输出使能寄存器 2(PWMOMUX2) [0xF7]

Table 495 PWM 特殊复用输出使能寄存器 2(PWMOMUX2)

参数名	比特位	属性	复位值	描述
PE0PWMOE	0	W/R	0	PE0 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出
PE1PWMOE	1	W/R	0	PE1 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出
PE2PWMOE	2	W/R	0	PE2 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出
PE3PWMOE	3	W/R	0	PE3 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出
PE4PWMOE	4	W/R	0	PE4 管脚 PWM 特殊复用输出使能: 0: 正常功能

参数名	比特位	属性	复位值	描述
				1: PWM 特殊复用输出
PE5PWMOE	5	W/R	0	PE5 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出
PE6PWMOE	6	W/R	0	PE6 管脚 PWM 输出复用输出使能: 0: 正常功能 1: PWM 特殊复用输出
PE7PWMOE	7	W/R	0	PE7 管脚 PWM 特殊复用输出使能: 0: 正常功能 1: PWM 特殊复用输出

19.10.79PA 输入缓冲使能寄存器(PAIE) [0xF9]

Table 496 PA 输入缓冲使能寄存器(PAIE)

参数名	比特位	属性	复位值	描述
PA0IE	0	W/R	1	PA0 输入缓冲使能: 0: 不使能 1: 使能
PA1IE	1	W/R	1	PA1 输入缓冲使能: 0: 不使能 1: 使能
PA2IE	2	W/R	1	PA2 输入缓冲使能: 0: 不使能 1: 使能
PA3IE	3	W/R	1	PA3 输入缓冲使能: 0: 不使能 1: 使能
PA4IE	4	W/R	1	PA4 输入缓冲使能: 0: 不使能 1: 使能
PA5IE	5	W/R	1	PA5 输入缓冲使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
PA6IE	6	W/R	1	PA6 输入缓冲使能: 0: 不使能 1: 使能
PA7IE	7	W/R	1	PA7 输入缓冲使能: 0: 不使能 1: 使能

19.10.80PB 输入缓冲使能寄存器(PBIE) [0xFA]

Table 497 PB 输入缓冲使能寄存器(PBIE)

参数名	比特位	属性	复位值	描述
PB0IE	0	W/R	1	PB0 输入缓冲使能: 0: 不使能 1: 使能
PB1IE	1	W/R	1	PB1 输入缓冲使能: 0: 不使能 1: 使能
PB2IE	2	W/R	1	PB2 输入缓冲使能: 0: 不使能 1: 使能
PB3IE	3	W/R	1	PB3 输入缓冲使能: 0: 不使能 1: 使能
PB4IE	4	W/R	1	PB4 输入缓冲使能: 0: 不使能 1: 使能
PB5IE	5	W/R	1	PB5 输入缓冲使能: 0: 不使能 1: 使能
PB6IE	6	W/R	1	PB6 输入缓冲使能: 0: 不使能 1: 使能
PB7IE	7	W/R	1	PB7 输入缓冲使能:

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能

19.10.81 PC 输入缓冲使能寄存器(PCIE) [0xFB]

Table 498 PC 输入缓冲使能寄存器(PCIE)

参数名	比特位	属性	复位值	描述
PC0IE	0	W/R	1	PC0 输入缓冲使能: 0: 不使能 1: 使能
PC1IE	1	W/R	1	PC1 输入缓冲使能: 0: 不使能 1: 使能
PC2IE	2	W/R	1	PC2 输入缓冲使能: 0: 不使能 1: 使能
PC3IE	3	W/R	1	PC3 输入缓冲使能: 0: 不使能 1: 使能
PC4IE	4	W/R	1	PC4 输入缓冲使能: 0: 不使能 1: 使能
PC5IE	5	W/R	1	PC5 输入缓冲使能: 0: 不使能 1: 使能
PC6IE	6	W/R	1	PC6 输入缓冲使能: 0: 不使能 1: 使能
PC7IE	7	W/R	1	PC7 输入缓冲使能: 0: 不使能 1: 使能

19.10.82 PD 输入缓冲使能寄存器(PDIE) [0xFC]

Table 499 PD 输入缓冲使能寄存器(PDIE)

参数名	比特位	属性	复位值	描述
PD0IE	0	W/R	1	PD0 输入缓冲使能： 0: 不使能 1: 使能
PD1IE	1	W/R	1	PD1 输入缓冲使能： 0: 不使能 1: 使能
PD2IE	2	W/R	1	PD2 输入缓冲使能： 0: 不使能 1: 使能
PD3IE	3	W/R	1	PD3 输入缓冲使能： 0: 不使能 1: 使能
PD4IE	4	W/R	1	PD4 输入缓冲使能： 0: 不使能 1: 使能
PD5IE	5	W/R	1	PD5 输入缓冲使能： 0: 不使能 1: 使能
PD6IE	6	W/R	1	PD6 输入缓冲使能： 0: 不使能 1: 使能
PD7IE	7	W/R	1	PD7 输入缓冲使能： 0: 不使能 1: 使能

19.10.83PE 输入缓冲使能寄存器(PEIE) [0xFD]

Table 500 PE 输入缓冲使能寄存器(PEIE)

参数名	比特位	属性	复位值	描述
PE0IE	0	W/R	1	PE0 输入缓冲使能： 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
PE1IE	1	W/R	1	PE1 输入缓冲使能: 0: 不使能 1: 使能
PE2IE	2	W/R	1	PE2 输入缓冲使能: 0: 不使能 1: 使能
PE3IE	3	W/R	1	PE3 输入缓冲使能: 0: 不使能 1: 使能
PE4IE	4	W/R	1	PE4 输入缓冲使能: 0: 不使能 1: 使能
PE5IE	5	W/R	1	PE5 输入缓冲使能: 0: 不使能 1: 使能
PE6IE	6	W/R	1	PE6 输入缓冲使能: 0: 不使能 1: 使能
PE7IE	7	W/R	1	PE7 输入缓冲使能: 0: 不使能 1: 使能

19.10.84PG 输入缓冲使能寄存器(PGIE) [0xFE]

Table 501 PG 输入缓冲使能寄存器(PGIE)

参数名	比特位	属性	复位值	描述
PG0IE	0	W/R	1	PG0 输入缓冲使能: 0: 不使能 1: 使能
PG1IE	1	W/R	1	PG1 输入缓冲使能: 0: 不使能 1: 使能
Reserved	7-2	R	0	保留

19.10.85PA 数据寄存器(P0) [0x80]

Table 502 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

19.10.86PB 数据寄存器(P1) [0x90]

Table 503 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

19.10.87PC 数据寄存器(P2) [0xA0]

Table 504 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

19.10.88PD 数据寄存器(P3) [0xB0]

Table 505 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

19.10.89PE 数据寄存器(P4) [0xC8]

Table 506 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

19.10.90PG 数据寄存器(P6) [0x98]

Table 507 PG 数据寄存器(P6)

参数名	比特位	属性	复位值	描述
P6.0	0	R	1	PG0 数据寄存器
P6.1	1	R	1	PG1 数据寄存器
Reserved	7-2	R	0	保留

20 电气特性

20.1 极限参数

Table 508 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，并不意味在此条件下器件的功能性操作无误。

20.2 工作条件

20.2.1 芯片供电

Table 509 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		2.45		5.5	V
V _{DDA}	模拟部分工作电压	必须和 V _{DD} 一致	2.45		5.5	V
f _{sys_clk}	系统工作时钟	V _{DD} :2.45V~5.5V			73.728	MHz

20.2.2 功耗特性

20.2.2.1 电流特性

Table 510 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/内部参考电压/ 外部晶体振荡器		38	50	mA

符号	参数	测试条件	最小	典型	最大	单位
		$f_{\text{sys_clk}} = 36.864\text{MHz}$, 打开 ADC/PGA/CMP/内部参考电压/ 外部晶体振荡器		19	25	mA
I _{DD2}	IDLE 模式电流	$f_{\text{sys_clk}} = 73.728\text{MHz}$, 关闭 ADC/PGA/CMP/内部参考电压/ 外部晶体振荡器	6	8	10	mA
I _{DD3}	IDLE 模式电流	$f_{\text{sys_clk}} = 9.216\text{MHz}$, 关闭 ADC/PGA/CMP/内部参考电压/ 外部晶体振荡器	1.5	2.5	3.5	mA
I _{DD4}	STOP 模式电流	关闭 ADC/PGA/CMP/内部参考电压/ 外部晶体振荡器	80	90	100	uA
I _{DD5}	SLEEP 模式电流		3	3.4	3.8	uA
I _{DD6}	Deep sleep 模式电流		2	2.4	2.8	uA

20.2.2.2 低功耗唤醒时间

Table 511 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 若使用外部管脚唤醒，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 若使用外部管脚唤醒，不考虑管脚滤波延时	109	us
t _{WUDPSLEEP}	从 Deep sleep 模式唤醒时间	使用 PA0 唤醒	109	us

20.2.3 I/O 特性

Table 512 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

符号	参数	测试条件	最小	典型	最大	单位
	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

20.2.4 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 1μF 的钽电容，使内部 LDO 输出稳定，如 Figure 10:

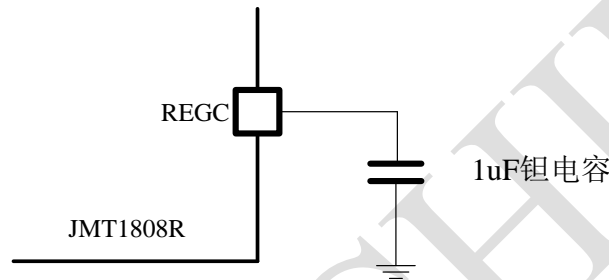


Figure 203 REGC 引脚连接示意图

20.2.5 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 204，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 512 定义的 V_{IL} 的最大值；
- 低电平宽度大于滤波的宽度。

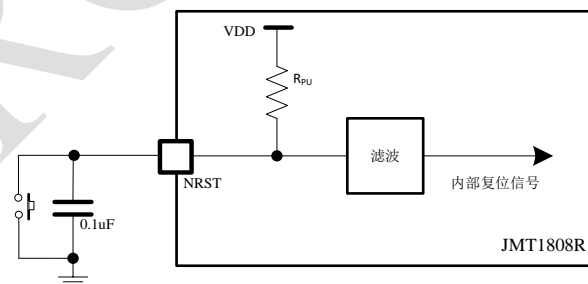


Figure 204 NRST 引脚连接示意图

20.2.6 上电/掉电条件

Table 513 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

符号	参数	测试条件	最小	最大	单位
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

20.2.7 时钟特性

20.2.7.1 内部高速 RC 振荡器

Table 514 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 =-40~125°	-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

20.2.7.2 内部低速 RC 振荡器

Table 515 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

20.2.7.3 外部时钟源

外部时钟源有两种方式：（1）连接一个外部时钟至管脚 OSC_IN；（2）连接一个谐振器至管脚 OSC_IN 和 OSC_OUT。

20.2.7.3.1 外部时钟

连接一个高速的外部时钟源至管脚 OSC_IN，可以给芯片提供工作时钟。

Table 516 外部时钟

符号	参数	测试条件	最小	典型	最大	单位
f _{EXT}	OSC_IN 输入的时钟频率		0		24	MHz
V _{EXTH}	OSC_IN 输入的时钟高电平		0.7V _{DD}		V _{DD}	V

符号	参数	测试条件	最小	典型	最大	单位
V _{EXTL}	OSC_IN 输入的时钟低电平		0		0.3V _{DD}	V
DUCY _{EXT}	占空比		40		60	%
I _L	OSC_IN 输入漏电流				1	uA

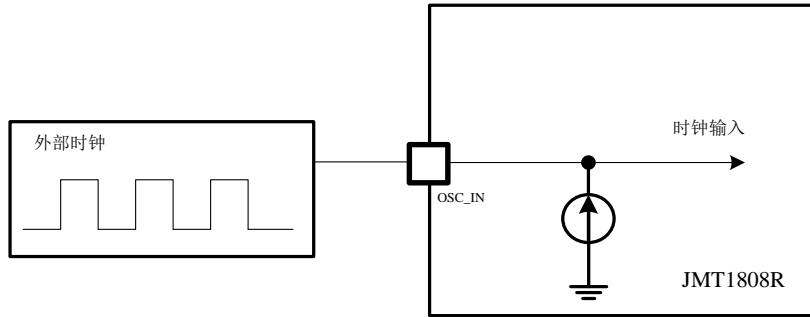


Figure 205 外部时钟输入 OSC_IN 连接示意图

20.2.7.3.2 晶体/陶瓷谐振器

使用一个 8~24MHz 的晶体/陶瓷谐振器构成振荡器产生外部时钟。在应用中，谐振器和负载电容必须尽可能的靠近振荡器的引脚，以减少输出失真和启动的稳定时间。

Table 517 晶体/陶瓷谐振器

符号	参数	测试条件	最小	典型	最大	单位
f _{OSC_IN}	振荡器频率		8	22.1184	24	MHz
R _F	反馈电阻		1		10	MΩ
C _{L1} C _{L2}	建议负载电容		10		30	pF
I ₂	内部振荡器驱动 电流				2	mA
t _{SU}	启动时间	V _{DD} 稳定之后		0.7	1	ms

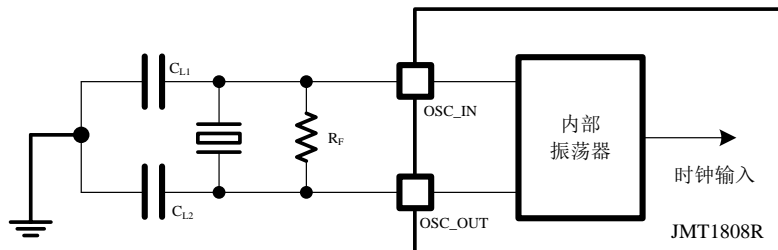


Figure 206 外部谐振器输入 OSC_IN/OSC_OUT 连接示意图

20.2.8 通信接口

20.2.8.1 I2C

JMT1808R 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 518 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
$t_f(SDA)$ $t_f(SCL)$	SDA 和 SCL 下降时间		300		300	ns
$t_h(START)$	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的时间（总线空闲）	4.7		1.3		us
C_b	每条总线的容性负载		400		400	pF

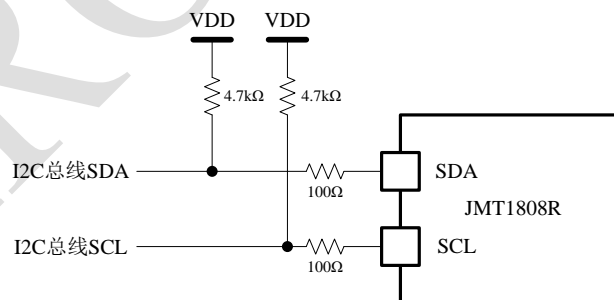


Figure 207 I2C 总线连接电路

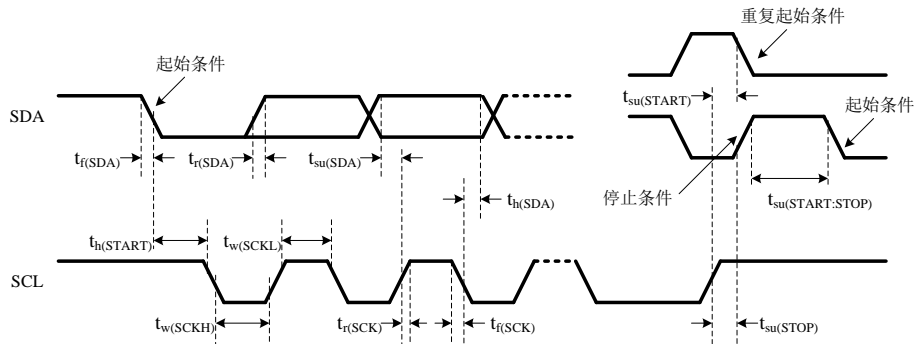


Figure 208 I2C 总线波形示意图

20.2.8.2 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 512。

Table 519 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	
$t_r(SCK)$ $t_f(SCK)$	SPI 时钟上升和下降时间	负载电容: $C=25pF$		9	ns
$t_{su}(NSS)$	NSS 建立时间	从机模式	$4t_{sys_clk}$		ns
$t_h(NSS)$	NSS 保持时间	从机模式	$2t_{sys_clk}+20$		ns
$t_w(SCKH)$ $t_w(SCKL)$	SCK 高电平和低电平时间	主机模式, $f_{sys_clk}=73.728MHz$, 预分频系数为 1	50	60	ns
$t_{su}(MI)$	数据输入建立时间	主机模式	6		ns
$t_{su}(SI)$	数据输入建立时间	从机模式	6		ns
$t_h(MI)$	数据输入保持时间	主机模式	6		ns
$t_h(SI)$	数据输入保持时间	从机模式	6		ns
$t_a(SO)$	数据输出访问时间	从机模式		$4t_{sys_clk}$	ns
$t_{dis}(SO)$	数据输出禁止时间	从机模式	10		ns
$t_v(SO)$	数据输出有效时间	从机模式 (使能边沿之后)		26	ns
$t_v(MO)$	数据输出有效时间	主机模式 (使能边沿之后)		3	ns
$t_h(SO)$	数据输出保持时间	从机模式 (使能边沿之后)	26		ns
$t_h(MO)$	数据输出保持时间	主机模式 (使能边沿之后)	3		ns

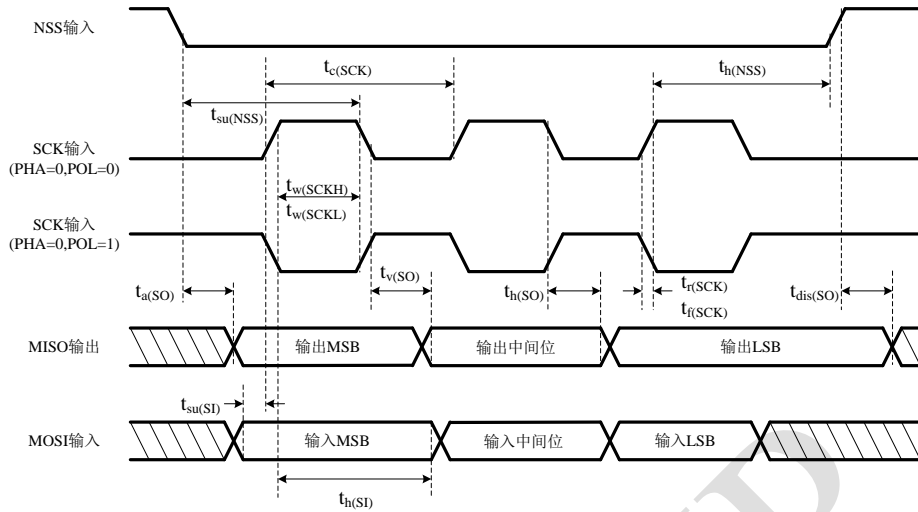


Figure 209 SPI 时序图 (从模式, PHA=0)

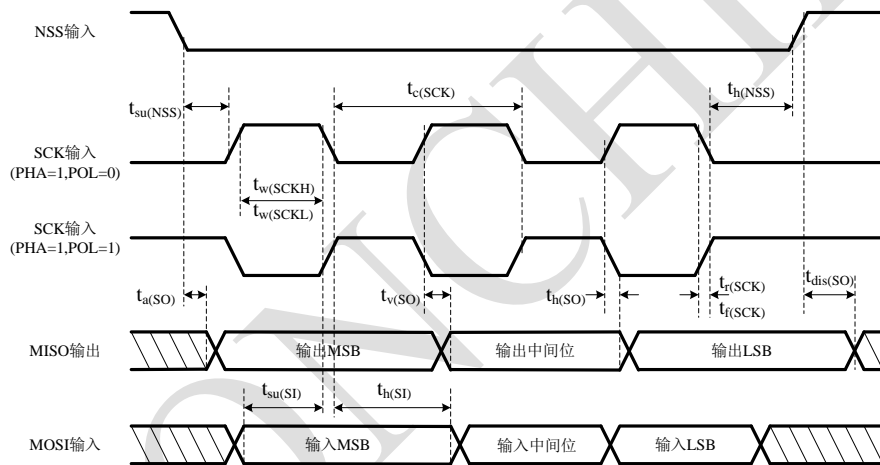


Figure 210 SPI 时序图 (从模式, PHA=1)

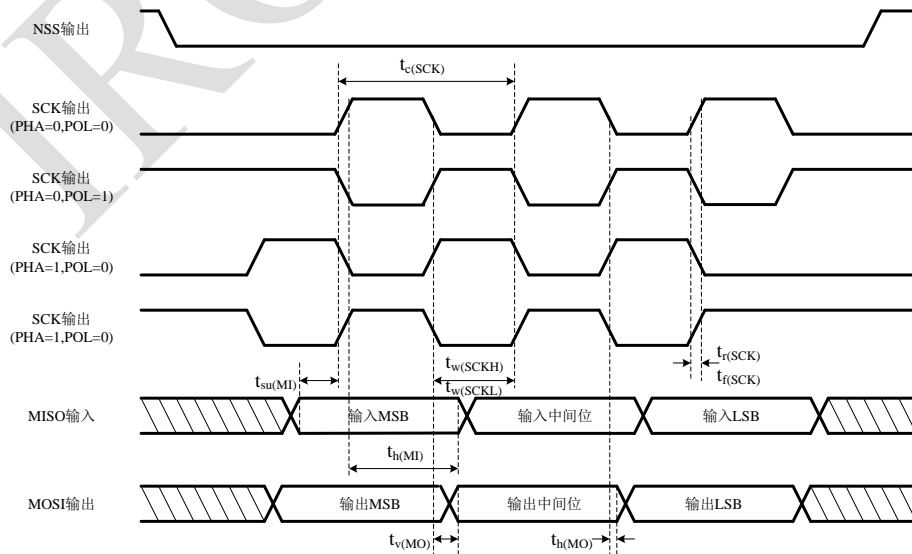


Figure 211 SPI 时序图 (主模式)

20.2.9 FLASH 特性

Table 520 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 521 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

20.2.10 ADC 特性

Table 522 ADC 电气特性

符号	参数	最小	典型	最大	单位
N_{R}	ADC 分辨率			10	BIT
D_{NL}	ADC 非线性微分误差			± 1.2	LSB
I_{NL}	ADC 非线性积分误差			± 1.5	LSB
E_{OFF}	ADC 偏移误差			± 2	LSB
E_{GN}	ADC 增益误差			± 1.5	LSB
V_{REF}	ADC 参考电压	0		V_{DDA}	V
V_{INADC}	ADC 输入电压范围	0		V_{REF}	V
f_{SAMPLE}	ADC 采样速率	0		526	KSPS
$f_{\text{adcx_sample_clk}}$	ADC 时钟频率	0		7.37	MHz
$T_{\text{adcx_sample_clk}}$	ADC 转换器时钟周期			135.6	ns
t_{adc}	A/D 转换时间		14		$T_{\text{adcx_sample_clk}}$

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 212 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。

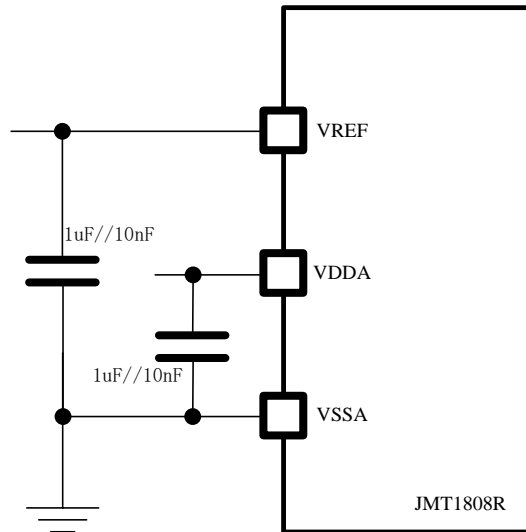


Figure 212 供电和参考电压去耦线路

20.2.11PGA 特性

Table 523 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		1/4		32	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{Hz}

20.2.12CMP 特性

Table 524 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_{INCOMP}	输入电压范围		0		V_{DDA}	V

符号	参数	测试条件	最小	典型	最大	单位
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

20.2.13 内部参考电压特性

芯片内部参考电压为 ADC、PGA 和 CMP 提供参考电压。

Table 525 内部参考电压特性

符号	参数	测试条件	最小	典型	最大	单位
V _{REFIN}	内部参考电压	-40°C < T _A < 125°C	2.4	2.5	2.6	V
V _{BG}	带隙参考缓冲电压	-40°C < T _A < 125°C	1.19	1.2	1.22	V

20.2.14 电气敏感性

20.2.14.1 ESD

ESD（静电放电敏感性试验）测试符合 JESD22-A114/C101 标准。

Table 526 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°C, 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°C, 符合 JESD22-C101	II	500	V

20.2.14.2 Latch Up

Latch Up（栓锁性能）测试符合 EIA/JESD 78A 集成电路栓锁标准。

Table 527 Latch Up

符号	参数	测试条件	类型
LU	静态栓锁类	T _A =125°C, 符合 JESD78A	II 类 A

21 封装特性

21.1 LQFP32L

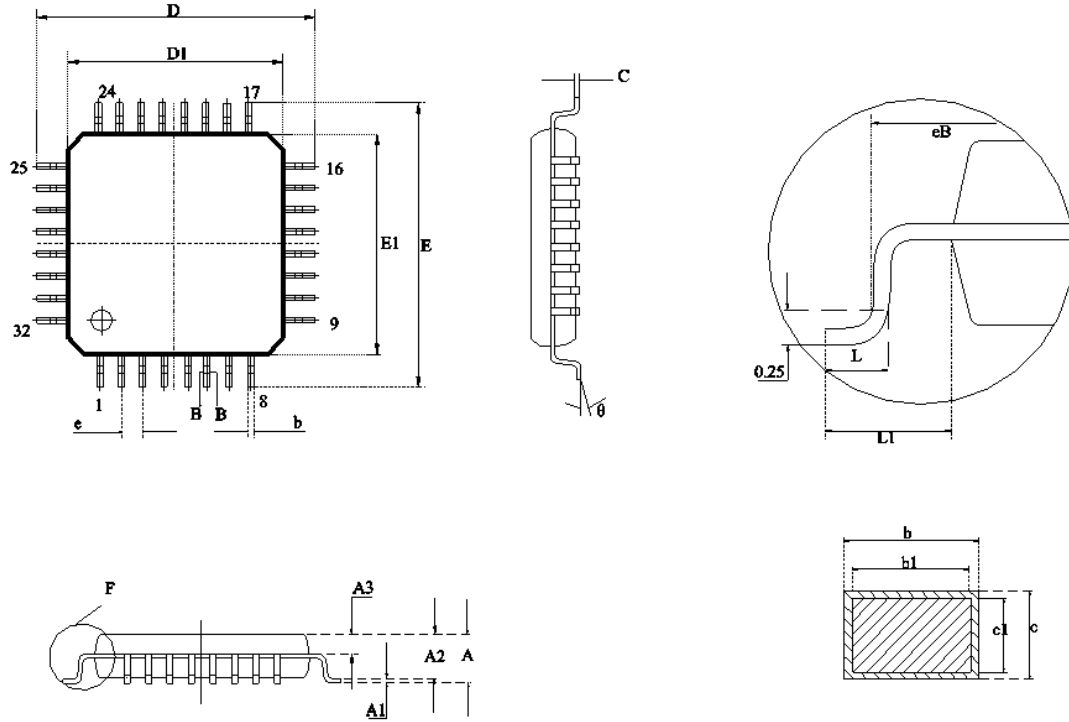


Figure 213 LQFP32L, 7X7mm 封装图

Table 528 LQFP32L, 7X7mm 机械数据

符号	毫米		
	最小	典型	最大
A			1.60
A1	0.05		0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.32		0.43
b1	0.31	0.35	0.39
c	0.13		0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20

符号	毫米		
	最小	典型	最大
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	0.8		
L	0.4		0.65
L1	1		
θ	0°		7°

21.2 LQFP48L

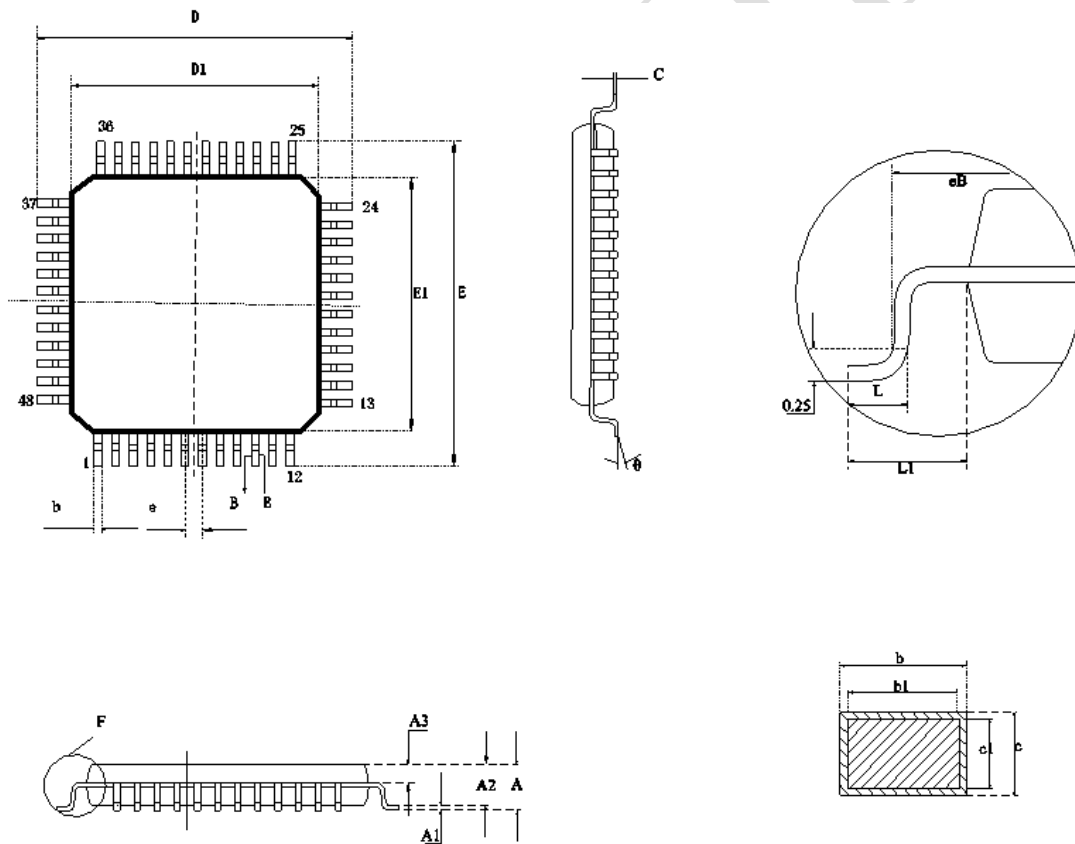


Figure 214 LQFP48L, 7X7mm 封装图

Table 529 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
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°

21.3 QFN32L

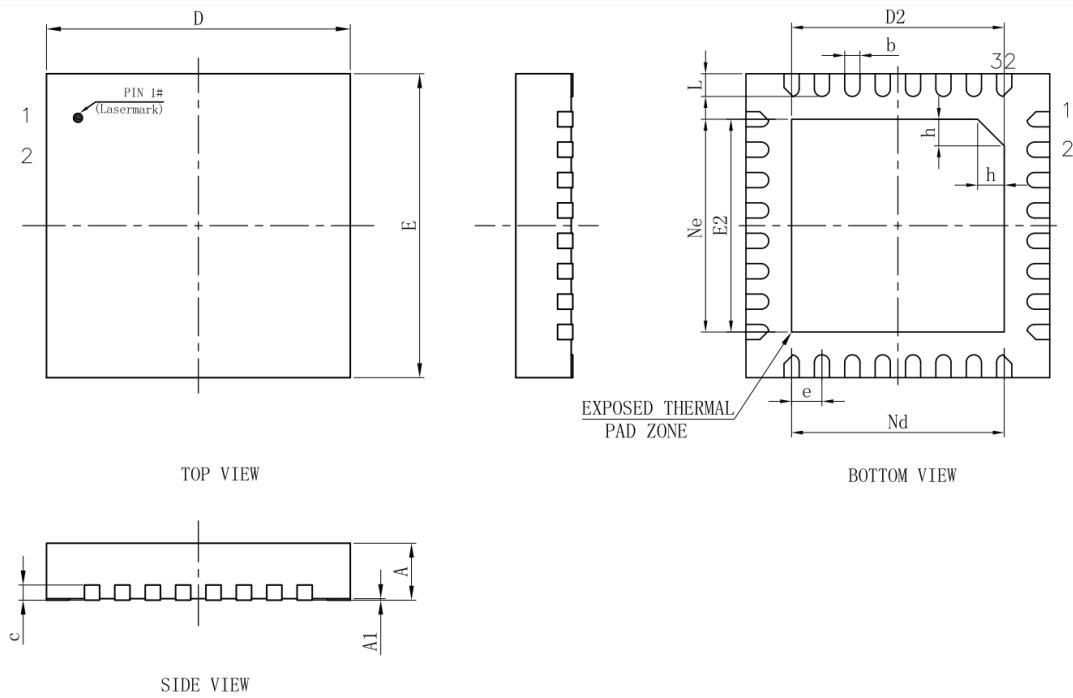


Figure 215 QFN32L, 4X4mm 封装图

Table 530 QFN32L, 4X4mm 机械数据

符号	毫米		
	最小	典型	最大
A	0.70	0.75	0.80
A1	0	0.02	0.05
b	0.15	0.20	0.25
c	0.18	0.20	0.25
D	3.90	4.00	4.10
D2	2.70	2.80	2.90
E	3.90	4.00	4.10
E2	2.70	2.80	2.90
e	0.40BSC		
Ne	2.80BSC		
Nd	2.80BSC		
L	0.25	0.30	0.35
h	0.30	0.35	0.40

21.4 TSSOP28L

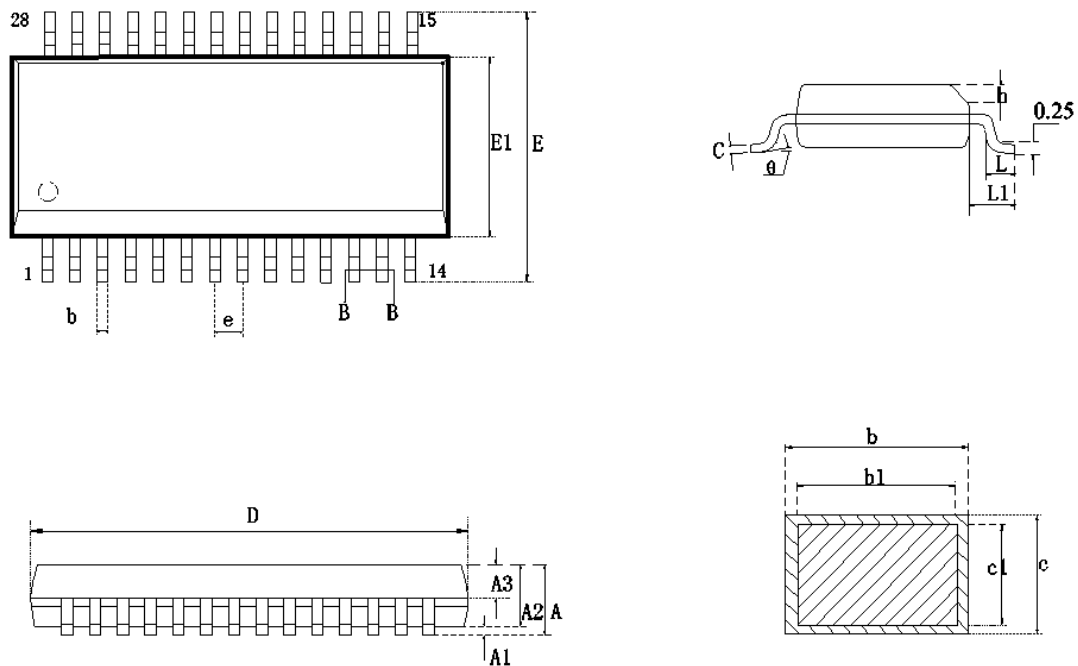


Figure 216 TSSOP28L, 9.9X3.9mm 封装图

Table 531 TSSOP28L, 9.9X3.9mm 机械数据

符号	毫米		
	最小	典型	最大
A			1.2
A1	0.05		0.15
A2	0.80		1.00
A3	0.39	0.44	0.49
b	0.20		0.29
b1	0.19	0.22	0.25
c	0.14		0.18
c1	0.12	0.13	0.14
D	9.60	9.70	9.80
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e		0.65	
L	0.45	0.60	0.75
L1		1.00	

符号	毫米		
	最小	典型	最大
θ	0°		8°

IRONCHIP

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。