

MG82F6P32

数据手册

8 位单片机
内嵌 OPA / PGA / ACMP

版本: V1.00

特 性

- 1-T 80C51 中央处理器
- **MG82F6P32** 内含 32K 字节闪存 ROM
 - ISP 空间可以选择为 **0.5KB/1.0KB~7.5KB**
 - 灵活的 IAP 大小空间设置
 - 密码保护程序区访问
 - Flash 擦/写次数: 20,000
 - Flash 数据保留时间: 100 年 25°C
 - **MG82F6P32 默认空间配置**
 - * AP 程序空间 (29.5KB, 0000h~75FFh)
 - * IAP 数据空间 (1.0KB, F600h~F9FFh)
 - * ISP 引导码空间(1.5KB, FA00h~FFFFh), ISP Boot code
- 数据存储: **2K** 字节
 - 片内 256 字节高速缓存
 - **MG82F6P32** - 片内 **1792** 字节扩展 RAM(XRAM)
 - 扩展 RAM(XRAM)支持页选
- 双数据指针
- 提供一个信道的 DMA 引擎
 - P2P, M2P, P2M
 - 存储器目标: 片内 XRAM
 - 外设目标: UART0/1, SPI, TWI0/I2C0, I2C1, ADC12, CRC16/32
 - 定时器 5 和定时器 6 被 DMA 所用, 但是当 DMA 没有使用时, 可以把它们作为独立的定时器使用
- 中断控制器
 - **21** 个中断源, 4 级优先级
 - **4** 个带滤波器的外部中断 nINT0、nINT1、nINT2 和 nINT3
 - 所有的外部中断支持高/低或上升/下降沿触发
 - AC0, AC1, OPA0 和 OPA1 支持唤醒 MCU
- **MG82F6P32** 共有 **12/14** (分割模式下)定时器
 - RTC 定时器和 WDT 定时器
 - 定时器 0、定时器 1、定时器 2 和定时器 3
 - PTM0 (PWM 定时器 0), PCA1 (可编程计数器阵列 1)
 - S0 BRG 和 S1 BRG
 - DMA 模块的定时器 5/6 也可以用作定时器
 - 如果定时器 2/3 用于分立模式, 则 **MG82F6P32** 有 **14** 定时器
- **6** 个 16-位定时/计数, 定时器 0、定时器 1、定时器 2、定时器 3, 定时器 5 和定时器 6
 - X12 模式和时钟输出功能
 - 定时器 **2/3** 有新的 6 个操作模式, 它们有着 8 个时钟源和 8 个捕捉源
 - 定时器 **2/3** 可以分立为两个 8 位定时器
 - 时钟输出(CCO)在 T2CKO, T3CKO
 - 定时器 0~3 支持 PWM 模式
 - 定时器 2/3 支持占空比捕捉功能
- **1** 个 16 位 PWM 定时器 (PTM0)
 - PTM0 有 6 个 CP (比较/PWM)模块
 - 可重载 16 位基准计数器支持可变长度的 PWM
 - 从片内 CKM 可获得高达 **96MHz** 的时钟来源
 - 16 位软件定时器模式和高速输出模式

- 可变 8/10/12/16 位 PWM 模式, PTM0 可被配置成:
 - * 最高 6 通道无缓冲 10/12/16 位 PWM, 或
 - * 最高 6 通道带缓冲 2~8 位可变频率 PWM, 或
 - * 最高 3 通道带缓冲 9~16 位可变频率 PWM
- PWM 具有死区控制, 中止控制和中心对齐选择
- 1 个可编程 16 位计数/定时阵列(PCA1)
 - PCA1 有两个 2 CCP (捕捉/比较/PWM)模块
 - 可重载 16 位基准计数器支持可变长度的 PWM
 - 从片内 CKM 可获得高达 96MHz 的时钟来源
 - 捕捉模式, 16 位软件定时器模式和高速输出模式
 - 缓冲捕捉模式可以监控窄脉冲输入
 - 8/10/12/16 位可调 PWM 模式, PCA1 可被配置成:
 - * 最高 2 通道无缓冲 10/12/16 位 PWM, 或
 - * 最高 2 通道带缓冲 2~8 位可变频率 PWM, 或
 - * 最高 1 通道带缓冲 9~16 位可变频率 PWM
 - PWM 具有死区控制, 中止控制和中心对齐选择
- 定时器 2/3, PTM0 和 PCA1 具有同步输出信号的全局控制
- 8 个键盘中断输入(KBI)
- 12 位单端 ADC
 - 可编程转换率高达 1M(常温) sps/ 500K sps (全温度)
 - MG82F6P32 有 8 个外部输入通道和 5 个用于内部参考电压(IVR/2.4V), PGO, OP0, OP1, 1/4VDD 和内部 VSS。
 - 支持 ADC 结果的窗口监测功能
 - 支持信道扫描模式
 - ADC VREF+ 从内部 IVR 2.4V 输入
- 片上电压参考(IVR 24)
- 2 个运算放大器 OPA (OPA0/ OPA1)
 - 低输入偏移,校准后 0.5mV ~ 1mV, 3mV 工厂微调
 - 内部 1.2V 参考可为正负端输入选项作为直流偏置或用于偏移校准
 - 支持 I/O 输入和输出
 - PGA 可作为前置放大器输出到负端输入
 - 内部输出路径连接 ADC 和 AC0 输入
 - 6 位微调范围的软件偏移校准
 - 低功耗模式
 - 支持比较模式
 - 比较模式支持中断, 可以在掉电模式下用于边缘唤醒触发源
- 可编程增益放大器(PGA)
 - 低输入偏移,校准后 0.5mV ~ 1mV, 3mV 工厂微调
 - 支持 I/O 输入和输出
 - 增益: x1, x2, x4, x8, x16, x32, x64, x128
 - 低功耗模式
 - 6 位微调范围的软件偏移校准
- 模拟比较器 0 (AC0)
 - ACNIO 可选内部电压基准(IVR/2.4V)
 - 4 个可选 ACPI0(+) 输入
 - 支持 OPAn 输出到 ACPI0/ACNIO 用于信号比较
 - 掉电模式和 IDLE 模式下唤醒

- 滤波选择和输出到定时器捕获
- 迟滞电压: 0mV, $\pm 10\text{mV}$, $\pm 20\text{mV}$ and $\pm 60\text{mV}$
- 6 位偏移微调
- 模拟比较器 1 (AC1)
 - 可选内部电压基准(IVR/2.4V)
 - 掉电模式和 IDLE 模式下唤醒
 - 滤波选择和输出到定时器捕获
 - 迟滞电压: 0mV, $\pm 10\text{mV}$, $\pm 20\text{mV}$ and $\pm 60\text{mV}$
 - 6 位偏移微调
- 增强型 UART (S0)
 - 帧错误监测
 - 自动地址识别
 - 最大波特率可达 6MHz
 - 模式 4 支持 SPI 主机, SPICLK 速率可达 12MHz
 - 内置波特率发生器(S0BRG)支持不同波特率的 TX 或 RX
 - S0BRG 在定时器模式级联定时器 0/1 具有 16/24 位的定时器/计数器
 - 支持 ARGB 数据传送
- 2nd UART (S1)
 - 最大波特率可达 1.8432/3.0MHz
 - 模式 4 支持 SPI 主机, SPICLK 速率可达 12MHz
 - 提供 BMC ENDEC 用于 USB PD
 - 支持 ARGB 数据传送
- 1 个主/从 SPI 串行接口
 - SPI 主机模式下 SPICLK 速率可达 24MHz
 - SPI 从机模式下速率可达 12MHz
 - 高达 3 个 SPI 主机(包括串口 S0/S1 模式 4)
 - 在 SPI 从机模式下支持菊花链功能(模式 1)
- 3 个主/从两线串行接口: TWI0/I2C0, TWI1/I2C1 and STWI (SI2C)
 - 2 个主/从硬件引擎: TWI0/I2C0 and TWI1/I2C1
 - I2C0/I2C1 主机模式可达 1MHz 和 2C0/I2C1 从机模式可达 400KHz
 - 1 个软件 TWI/I2C、STWI/SI2C、开始/停止串行信号监测(SID)
 - I2C0/I2C1 多从机地址识别
- 可编程看门狗定时器(WDT), 时钟来源为 ILRCO, XTAL 或 SYSCLK/12
 - 通过 CPU 或上电一次性使能
 - WDT 溢出可中断 CPU 或复位 CPU
 - 掉电模式(watch 模式)下支持看门狗(WDT)功能, 用于自动唤醒功能
- 实时时钟模块 (RTC), 时钟源来自 XTAL, ILRCO, WDTPS, WDTOF, SYSCLK 或 SYSCLK/12
 - 可编程中断周期从微秒到分钟唤醒
 - 21-位长系统定时器
- 蜂鸣器功能
- 通用逻辑(GPL/CRC)
 - 位序颠倒功能
 - 16 位 CRC 引擎 (多项式: 0x1201)
 - 支持 FLASH 内容的自动 CRC
 - 可编程 CRC 的初始种子功能
 - 4b5b 编码/解码 (ENDEC)

- 32 位 CRC 引擎 (多项式: **0x04C1_1DB7**)
- 片上调试接口 (OCD)
- 在 **32/28** 脚封装中最大 **29/25** 个 GPIO
 - 默认下所有 I/O 都是模拟 I/O 模式
 - P3 可以设置成准双向口模式, 推挽输出模式, 开漏集输出模式和仅输入模式
 - P0、P1、P2、P4 和 P6 可以设置为推挽输出模式, 开漏集输出模式
 - P6.0,P6.1 和 P4.7 共享 XTAL2, XTAL1 和 RST
 - 可编程通用输入输出 (GPIO) 的驱动力和驱动速度
 - 每一个脚具有片内上拉使能
- 时钟源
 - 内部 12MHz/11.059MHz 振荡器(IHRCO): 工厂校对到±1%, 典型值
 - 外部晶振模式, 支持 32.768KHz 和时钟丢失监测(MCD)
 - 内部低功耗 32KHz RC 振荡器(ILRCO)
 - 外部时钟输入(ECKI)在 P6.0/XTAL2, 可达到 25MHz
 - 内部 RC 振荡输出在 P6.0/XTAL2
 - 片内时钟倍频器(CKM)可提供高速时钟源 **96MHz**
- 两个低电压监测
 - BOD0: 监测 **1.7V**
 - BOD1: 选择监测电压为 4.2V/3.6V/2.4V/2.0V
 - 中断 CPU 或复位 CPU
 - (BOD1)在掉电模式下唤醒 CPU
- 多种功耗模式: 掉电模式、空闲模式、慢频模式、副频模式、RTC 模式、watch 模式和 monitor 模式
 - 所有的中断能唤醒空闲(IDLE)模式
 - **23** 只引脚的 **19** 个中断源能唤醒掉电模式
 - 慢频模式和副频模式支持低速 MCU 运转
 - RTC 模式在掉电模式下支持实时时钟(RTC)恢复 CPU
 - Watch 模式在掉电模式下支持看门狗(WDT)恢复 CPU
 - Monitor 模式在掉电模式下支持 BOD1 恢复 CPU
- 工作电压范围: 1.8V – 5.5V
 - Flash 写操作(ISP/IAP/ICP)的最低电压为 **1.8V**
- 工作频率范围: **32MHz** (最大)
 - 外部时钟输入模式, 0 – 12MHz @ 2.0V – 5.5V, 0 – 25MHz @ 2.4V – 5.5V
 - CPU 可达 12MHz @ **1.8V** – 5.5V, 可达 25MHz @ **2.2V** – 5.5V
 - 使用片内时钟倍频器(CKM), CPU 可达 32MHz @ 2.7V -5.5V
 - 外设时钟(SYSCLK)可达 48MHz
- 16 字节唯一 ID 码
- 工作温度
 - 工业级(-40°C 到+105°C)*
- 封装类型
 - LQFP32 (7 x 7 mm): MG82F6P32AD32 (32K)
 - QFN32 (4 x 4 x 0.55 mm): MG82F6P32AZ32 (32K)
 - SSOP28 (150 mil): MG82F6P32AL28 (32K)

*: 抽样检测

目 录

特 性.....	3
目 录.....	7
图列表	15
表列表	19
1. 概述	21
2. 方框图.....	22
3. 特殊功能寄存器.....	23
3.1. SFR 图(页 0~F)	23
3.2. SFR 位分配 (Page 0~F)	25
3.3. 辅助 SFR 图 (P 页)	29
3.4. 辅助 SFR 位分配(P 页).....	30
4. 引脚结构	31
4.1. 封装指南.....	31
4.2. 引脚描述.....	33
4.3. 功能复用.....	35
5. 8051 CPU 功能描述.....	42
5.1. CPU 寄存器	42
5.2. CPU 时序.....	43
5.3. CPU 寻址模式	44
6. 存储器组织.....	45
6.1. 片内程序闪存.....	45
6.2. 片内数据 RAM.....	46
6.3. 片内扩展 RAM (XRAM)	48
6.4. EMB, 片外数据存储器总线.....	48
6.5. 关于 C51 编译器的声明标识符	48
7. XRAM 访问	49
7.1. MOVX 在 16 位地址上使用双 DPTR	49
7.2. MOVX 在 8 位地址上使用 XRPS.....	51
8. 直接存储器访问控制器 (DMA).....	52
8.1. DMA 结构	53
8.2. DMA 操作	54
8.2.1. DMA 传输类型	54
8.2.2. DMA 传输模式	54
8.2.3. 传输计数和地址指针	55
8.2.4. 启动一个 DMA 传输	55
8.2.5. 暂停或停止 DMA 传输	55
8.2.6. DMA 中断.....	56
8.2.7. DMA 循环模式	56
8.2.8. DMA 的错误处理	56
8.2.9. 数据拷贝到 CRC16.....	56
8.2.10. 定时器 5 & 定时器 6.....	57
8.3. DMA 寄存器.....	59
8.4. 定时器 5 寄存器.....	61
8.5. 定时器 6 寄存器.....	62
9. 系统时钟	64
9.1. 时钟结构.....	65

9.2.	时钟源切换	66
9.3.	片内 CKM (PLL)	66
9.4.	时钟丢失监测(MCD)	66
9.5.	XTAL 模式下快速唤醒	66
9.6.	CKM 下的唤醒	66
9.7.	时钟寄存器	67
10.	看门狗定时器(WDT)	71
10.1.	WDT 结构	71
10.2.	空闲模式下的 WDT	71
10.3.	掉电模式下的 WDT (自动唤醒)	71
10.4.	WDT 寄存器	72
10.5.	WDT 硬件选项	74
11.	实时时钟(RTC)/系统时钟	75
11.1.	RTC 寄存器	76
12.	系统复位	78
12.1.	复位源	78
12.2.	上电复位	79
12.3.	外部复位	79
12.4.	软件复位	79
12.5.	低电压监测复位	81
12.6.	WDT 复位	83
12.7.	MCD 复位	83
12.8.	非法地址复位	84
12.9.	堆栈指针错误复位	84
13.	电源管理	85
13.1.	低电压监测器	85
13.2.	省电模式	86
13.2.1.	慢频模式	86
13.2.2.	副频模式	86
13.2.3.	RTC 模式	86
13.2.4.	Watch 模式	86
13.2.5.	Monitor 模式	86
13.2.6.	Idle 模式	86
13.2.7.	掉电模式	86
13.2.8.	中断唤醒掉电模式	88
13.2.9.	复位唤醒掉电模式	88
13.2.10.	KBI 键盘唤醒掉电模式	88
13.3.	电源控制寄存器	89
14.	I/O 口配置(GPIO)	91
14.1.	IO 结构	91
14.1.1.	端口 3 准双向口结构	91
14.1.2.	端口 3 推挽输出结构	92
14.1.3.	端口 3 仅输入(高阻抗输入)结构	92
14.1.4.	端口 3 开漏输出结构	93
14.1.5.	通用模拟输入输出结构	93
14.1.6.	带上拉电阻的通用开漏输出	94
14.1.7.	通用开漏输出结构	94
14.1.8.	通用端口的数字输入结构	94

14.1.9.	通用推挽输出结构	95
14.1.10.	GPIO 读和写-- 端口锁存器和端口状态	95
14.1.11.	复位引脚结构	96
14.1.12.	端口引脚输出驱动力选择	96
14.1.13.	端口引脚输出快速驱动选择	96
14.2.	I/O 口寄存器	97
14.2.1.	端口 0 寄存器	97
14.2.2.	端口 1 寄存器	98
14.2.3.	端口 2 寄存器	98
14.2.4.	端口 3 寄存器	99
14.2.5.	端口 4 寄存器	99
14.2.6.	端口 6 寄存器	100
14.2.7.	端口输出驱动力控制寄存器	101
14.2.8.	端口输出快速驱动控制寄存器	102
15.	中断	104
15.1.	中断结构	104
15.2.	中断源	106
15.3.	中断使能	109
15.4.	中断优先级	110
15.5.	中断处理	110
15.6.	nINTx 输入源选择和输入滤波器(x=0~3)	111
15.7.	中断寄存器	112
16.	定时器/计数器	121
16.1.	定时器 0 和定时器 1	122
16.1.1.	定时器 0/1 模式 0	122
16.1.2.	定时器 0/1 模式 1	124
16.1.3.	定时器 0/1 模式 2	125
16.1.4.	定时器 0/1 模式 3	126
16.1.5.	定时器 0/1 可编程时钟输出	127
16.1.6.	定时器 0/1 寄存器	129
16.2.	定时器 2	133
16.2.1.	定时器 2 模式 0(自动重载和外部中断)	133
16.2.2.	定时器 2 模式 1(外部中断触发自动重载)	134
16.2.3.	定时器 2 模式 2(捕捉)	135
16.2.4.	定时器 2 模式 3(捕捉带自动清零)	136
16.2.5.	定时器 2 模式 6 (占空比捕捉)	137
16.2.6.	分立定时器 2 模式 0 (自动重载和外部中断)	138
16.2.7.	分立定时器 2 模式 1 (外部中断触发自动重载)	139
16.2.8.	分立定时器 2 模式 2 (捕捉)	140
16.2.9.	分立定时器 2 模式 3 (捕捉带自动清零)	141
16.2.10.	分立定时器 2 模式 4 (8 位 PWM 模式)	142
16.2.11.	分立定时器 2 模式 6 (占空比捕捉)	143
16.2.12.	波特率发生器模式 (BRG)	144
16.2.13.	定时器 2 可编程时钟输出	146
16.2.14.	定时器 2 寄存器	148
16.3.	定时器 3	152
16.3.1.	定时器 3 模式 0(自动重载和外部中断)	152
16.3.2.	定时器 3 模式 1(外部中断触发自动重载)	153
16.3.3.	定时器 3 模式 2(捕捉)	154

16.3.4.	定时器 3 模式 3(捕捉带自动清零)	155
16.3.5.	定时器 3 模式 6 (占空比捕捉)	156
16.3.6.	分立定时器 3 模式 0 (自动重载和外部中断)	157
16.3.7.	分立定时器 3 模式 1(外部中断触发自动重载)	158
16.3.8.	分立定时器 3 模式 2(捕捉)	159
16.3.9.	分立定时器 3 模式 3 (捕捉带自动清零)	160
16.3.10.	分立定时器 3 模式 4 (8-位 PWM 模式)	161
16.3.11.	分立定时器 3 模式 6 (占空比捕捉)	162
16.3.12.	定时器 3 可编程时钟输出	163
16.3.13.	定时器 3 寄存器	165
16.4.	定时器全局控制	169
16.4.1.	所有定时器运行的全局控制	169
16.4.2.	所有定时器重载的全局控制	169
16.4.3.	所有定时器停止的全局控制	170
17.	PWM 定时器(PTM0)	171
17.1.	PTM0 概述	171
17.2.	PTM0 定时器/计数器	172
17.3.	比较模块	176
17.4.	PCAx/PTMx 操作模式	178
17.4.1.	8 位软件定时器模式 (比较模式)	178
17.4.2.	16 位软件定时器模式(比较模式)	179
17.4.3.	缓冲 16 位软件定时器模式(比较模式)	180
17.4.4.	8 位高速输出模式(比较输出模式)	181
17.4.5.	16 位高速输出模式 (比较输出模式)	182
17.4.6.	相位可变计较输出模式	183
17.4.7.	缓冲 8 位 PWM 模式	184
17.4.8.	无缓冲 10/12/16 位 PWM 模式	185
17.4.9.	缓冲 10/12/16 位 PWM 模式	186
17.4.10.	PTM0 增强型 PWM 控制	187
17.4.11.	PTM0 模块输出控制	191
17.4.12.	可变分辨率的 PWM 中心对齐	196
17.4.13.	PWM 全局控制	196
18.	可编程计数器阵列(PCA1)	197
18.1.	PCA1 概述	197
18.2.	PCA1 定时器/计数器	198
18.3.	比较/捕捉模块	202
18.4.	PCA1 操作模式	204
18.4.1.	捕捉模式	205
18.4.2.	缓冲捕捉模式	206
18.4.3.	8 位软件定时器模式 (比较模式)	207
18.4.4.	16 位软件定时器模式(比较模式)	208
18.4.5.	缓冲 16 位软件定时器模式(比较模式)	209
18.4.6.	8 位高速输出模式(输出比较模式)	210
18.4.7.	16 位高速输出模式(输出比较模式)	211
18.4.8.	相位可变计较输出模式	212
18.4.9.	缓冲 8 位 PWM 模式	213
18.4.10.	无缓冲 10/12/16 位 PWM 模式	214
18.4.11.	缓冲 10/12/16 位 PWM 模式	215
18.4.12.	PCA1 增强型 PWM 控制	216
18.4.13.	PCA1 模块输出控制	220

18.4.14.	可变分辨率的 PWM 中心对齐	223
18.4.15.	PWM 全局控制	223
19.	串口 0 (UART0)	224
19.1.	串口 0 模式选择	224
19.2.	串口 0 模式 0	225
19.3.	串口 0 模式 1	227
19.4.	串口 0 模式 2 和模式 3	228
19.5.	帧错误侦测	228
19.6.	多处理器通讯	229
19.7.	自动地址识别	229
19.8.	串口 0 模式 4 (SPI 主机)	231
19.9.	波特率设置	232
19.9.1.	S0 的波特率选择	232
19.9.2.	模式 0&4 的波特率(串行时钟频率)	233
19.9.3.	模式 2 波特率	233
19.9.4.	模式 1&3 波特率	234
19.10.	ARGB 发送器	243
19.10.1.	S0 ARGB 波特率	243
19.10.2.	S0 ARGB 总线复位时间	244
19.10.3.	S0 ARGB 中断	245
19.10.4.	S0 ARGB 寄存器	245
19.11.	串口 0 增强功能	246
19.11.1.	S0 波特率发生器 (S0BRG)	248
19.11.2.	S0 作为 8 位定时器模式	248
19.11.3.	S0BRG 可编程时钟输出	249
19.12.	串口 0 寄存器	250
20.	串口 1 (UART1)	254
20.1.	串口 1 模式选择	254
20.2.	串口 1 波特率发生器(S1BRG)	255
20.3.	串口 1 波特率设定	256
20.3.1.	模式 0 & 4 波特率(串行时钟频率)	256
20.3.2.	模式 2 波特率	256
20.3.3.	模式 1 & 3 波特率	256
20.4.	串口 1 模式 4 (SPI 主机)	260
20.5.	S1BRG 定时器模式(模式 8)	262
20.5.1.	8 位定时器模式	262
20.6.	BMC ENDEC (模式 6)	263
20.6.1.	模式 6 波特率	263
20.7.	ARGB 发送器	266
20.7.1.	S1 ARGB 波特率	266
20.7.2.	S1 ARGB 总线复位时间	266
20.7.3.	S1 ARGB 中断	267
20.7.4.	S1 ARGB 寄存器	267
20.8.	S1BRT 可编程时钟输出	268
20.9.	S1 波特率发生器用于 S0	269
20.10.	串口 1 寄存器	270
21.	串行外设接口 (SPI)	274
21.1.	典型 SPI 配置	275

21.1.1.	单主机和单从机	275
21.1.2.	双设备，既是主机也是从机	275
21.1.3.	单主机和多从机	275
21.2.	SPI 配置	276
21.2.1.	一个从机的补充注意事项	276
21.2.2.	一个主机的补充注意事项	276
21.2.3.	nSS 引脚模式改变	277
21.2.4.	发送保持寄存器满标志	277
21.2.5.	写冲突	277
21.2.6.	SPI 时钟速率选择	277
21.3.	数据模式	278
21.4.	菊花链连接	280
21.4.1.	菊花链配置	280
21.5.	SPI 寄存器	281
22.	双线串行接口 (TWI0/I2C0 & TWI1/I2C1)	284
22.1.	工作模式	285
22.1.1.	主机发送模式	285
22.1.2.	主机接收模式	285
22.1.3.	从机发送模式	286
22.1.4.	从机接收模式	286
22.1.5.	多从机地址识别	287
22.2.	混合状态	288
22.3.	使用 TWI/I2C	288
22.4.	TWI0/I2C0 寄存器	294
22.5.	TWI1/I2C1 寄存器	298
23.	串行接口侦测(STWI/SI2C)	302
23.1.	SID 结构	302
23.2.	SID 寄存器	302
24.	蜂鸣器	304
24.1.	蜂鸣器寄存器	304
25.	键盘中断 (KBI)	305
25.1.	KBI 结构	305
25.2.	KBI 寄存器	306
26.	通用逻辑(GPL-CRC)	308
26.1.	GPL-CRC 结构	308
26.2.	GPL-BOREV 结构	309
26.3.	GPL-EDC45 结构	309
26.4.	GPL 寄存器	311
27.	运算放大器 (OPA0/ OPA1)	313
27.1.	OPA 结构	313
27.2.	OPA 操作	313
27.2.1.	OPA 输入通道	313
27.2.2.	OPA 工作模式	314
27.2.3.	OPA 功率模式	314
27.2.4.	OPA 模式	314
27.2.5.	比较模式	314
27.2.6.	OPA 输入偏移微调	315
27.2.7.	Idle 和掉电模式	315

27.3. OPAn 寄存器	316
28. 可编程增益放大器, PGA	320
28.1. PGA 结构	320
28.2. PGA 操作	320
28.2.1. PGA 输入通道	320
28.2.2. PGA 操作	320
28.2.3. PGA 输入偏移微调模式	321
28.3. PGA 寄存器	321
29. 12-Bit ADC	323
29.1. ADC 结构	323
29.2. ADC 操作	324
29.2.1. ADC 参考电压 VREF+ 选择	324
29.2.2. ADC 输入通道	324
29.2.3. ADC 内部电压参考	324
29.2.4. 开启一次转换	324
29.2.5. ADC 转换率	325
29.2.6. ADC 中断	326
29.2.7. ADC 窗口检测	326
29.2.8. ADC 信道扫描模式	327
29.2.9. DMA 传输 ADC 数据	328
29.2.10. I/O 端口引脚用于 ADC 功能	329
29.2.11. 空闲和掉电模式	329
29.2.12. 如何提高 ADC 精准度	329
29.3. ADC 寄存器	330
30. 模拟比较器 0/1 (AC0/AC1)	336
30.1. AC0/AC1 结构	336
30.2. AC0/AC1 寄存器	337
31. 内部电压参考 (IVR, 2.4V)	343
31.1. IVR (2.4V) 结构	343
31.2. IVR 寄存器	343
31.3. 如何读取 IVR (1.4V) ADC 预设值	344
32. ISP 和 IAP	345
32.1. MG82F6P32 Flash 存储器配置	345
32.2. MG82F6P32 Flash 在 SP/IAP 上的访问	346
32.2.1. ISP/IAP Flash 页擦除模式	347
32.2.2. ISP/IAP Flash 字节编程模式	348
32.2.3. ISP/IAP Flash 读模式	349
32.3. ISP 操作	350
32.3.1. 硬件启动 ISP 方法	350
32.3.2. 软件启动 ISP 方法	350
32.3.3. ISP 注意事项	351
32.4. 在应用编程(IAP)	352
32.4.1. MG82F6P32 IAP 存储边界/范围	352
32.4.2. 更新 IAP-存储中的数据	353
32.4.3. AP 注意事项	354
32.5. ISP/IAP 寄存器	355
32.5.1. ISP/IAP 示例代码	357
33. P 页 SFR 访问	358

34. 辅助特殊功能寄存器	363
35. 硬件选项	372
36. 应用说明	374
36.1. 电源电路	374
36.2. 复位电路	374
36.3. XTAL 振荡电路	375
36.4. ICP 和 OCD 接口电路	376
36.5. 在芯片编程功能	377
36.6. 在线调试功能	378
37. 电气特性	379
37.1. 最大绝对额定值	379
37.2. 直流特性	380
37.3. 外部复位特性	381
37.4. 外部时钟特性	382
37.5. IHRCO 特性	382
37.6. ILRCO 特性	382
37.7. CKM 特性	383
37.8. Flash 特性	383
37.9. OPA 特性	383
37.10. PGA 特性	384
37.11. ADC 特性	385
37.12. IVR 特性	386
37.13. 模拟比较器 AC0/AC1 特性	387
37.14. 串口时序特性	388
37.15. SPI 时序特性	389
37.16. TWI/I2C 时序特性	391
38. 指令集	392
39. 封装尺寸	395
39.1. LQFP-32 (7mm X 7mm)封装尺寸	395
39.2. QFN-32 (4mm X 4mm X 0.55mm) 封装尺寸	396
39.3. SSOP28 (150 mil)封装尺寸	397
40. 版本历史	398
41. 免责声明	399

图列表

图 2-1. MG82F6P32 方框图.....	22
图 4-1. MG82F6P32 LQFP32 顶视图	31
图 4-2. MG82F6P32 QFN32 顶视图	32
图 4-3. MG82F6P32 SSOP28 顶视图	32
图 6-1. 程序存储器	45
图 6-2. 数据存储器	46
图 6-3. 内部 RAM 的低 128 字节	47
图 6-4. SFR 空间	47
图 7-1. 双 DPTR 结构	49
图 7-2. XRPS 结构	51
图 8-1. DMA 访问方框图	52
图 8-2. DMA 结构	53
图 8-3. DMA 中断	56
图 8-4. 定时器 5 结构	57
图 8-5. 定时器 6 结构	58
图 9-1. 系统时钟	65
图 10-1. 看门狗定时器	71
图 11-1. 实时时钟计数器	75
图 12-1. 系统复位源	78
图 12-2. BOD1 复位流程	81
图 12-3. BOD1 侦测软件控制	82
图 13-1. 低电压监测器 0/1	85
图 13-2. 掉电模式唤醒结构	87
图 14-1. 端口 3 准双向 I/O	92
图 14-2. 端口 3 推挽输出	92
图 14-3. 端口 3 仅输入	92
图 14-4. 端口 3 开漏输出	93
图 14-5. 带上拉电阻的通用开漏输出	93
图 14-6. 带上拉电阻的通用开漏输出	94
图 14-7. 通用开漏输出	94
图 14-8. 通用推挽输出	95
图 14-9. GPIO 读和写路径	95
图 14-10. 复位引脚输入结构	96
图 14-11. 复位引脚编程为推挽输出	96
图 15-1. 中断系统	105
图 15-2. 系统标志中断结构	108
图 15-3. nINT0~3 端口引脚选择配置	111
图 16-1. 定时器 0 模式 0 结构	122
图 16-2. 定时器 1 模式 0 结构	123
图 16-3. 定时器 0 模式 1 结构	124
图 16-4. 定时器 1 模式 1 结构	124
图 16-5. 定时器 0 模式 2 结构	125
图 16-6. 定时器 1 模式 2 结构	125
图 16-7. 定时器 0 模式 3 结构	126
图 16-8. 定时器 0 时钟输出公式	127
图 16-9. 定时器 1 时钟输出公式	127
图 16-10. 定时器 0 时钟输出模式	127
图 16-11. 定时器 0 时钟输出控制	128
图 16-12. 定时器 1 时钟输出模式	128
图 16-13. 定时器 2 模式 0 结构(自动重载和外部中断模式)	133
图 16-14. 定时器 2 模式 1 结构(外部中断触发自动重载模式)	134
图 16-15. 定时器 2 模式 2 结构(捕捉模式)	135
图 16-16. 定时器 2 模式 3 结构(捕捉带自动清零 TL2 和 TH2)	136

图 16-17. 定时器 2 模式 6 结构 (占空比捕捉)	137
图 16-18. 分立定时器 2 模式 0 结构(自动重载和外部中断).....	138
图 16-19. 分立定时器 2 模式 1 结构(外部中断触发自动重载)	139
图 16-20. 分立定时器 2 模式 2 结构 (捕捉).....	140
图 16-21. 分立定时器 2 模式 3 结构(捕捉带自动清零 TH2)	141
图 16-22. 分立定时器 2 模式 4 结构(8 位 PWM 模式)	142
图 16-23. 分立定时器 2 模式 6 结构 (占空比捕捉)	143
图 16-24. 定时器 2 波特率发生器模式	144
图 16-25. 分立定时器 2 波特率发生器模式	145
图 16-26. 定时器 2 时钟输出公式	146
图 16-27. 定时器 2 时钟输出模式.....	146
图 16-28. 分立定时器 2 时钟输出公式	147
图 16-29. 分立定时器 2 时钟输出模式	147
图 16-30. 定时器 3 模式 0 结构(自动重载和外部中断模式).....	152
图 16-31. 定时器 3 模式 1 结构(外部中断触发自动重载模式)	153
图 16-32. 定时器 3 模式 2 结构(捕捉模式)	154
图 16-33. 定时器 3 模式 3 结构 (捕捉带自动清零 TL3 和 TH3)	155
图 16-34. 定时器 3 模式 6 结构 (占空比捕捉)	156
图 16-35. 分立定时器 3 模式 0 结构(自动重载和外部中断).....	157
图 16-36. 分立定时器 3 模式 1 结构(外部中断触发自动重载)	158
图 16-37. 分立定时器 3 模式 2 结构(捕捉)	159
图 16-38. 分立定时器 3 模式 3 结构(捕捉带自动清零 TH3)	160
图 16-39. 分立定时器 3 模式 4 结构(8 位 PWM 模式)	161
图 16-40. 分立定时器 3 模式 6 结构 (占空比捕捉)	162
图 16-41. 定时器 3 时钟输出频率计算公式	163
图 16-42. 定时器 3 时钟输出模式.....	163
图 16-43. 分立定时器 3 时钟输出公式	164
图 16-44. 分立定时器 3 时钟输出模式	164
图 17-1. PTM0 方框图.....	171
图 17-2. PTM0 定时器/计数器.....	172
图 17-3. PTM0 中断系统	174
图 17-4. PCA 软件定时器模式	178
图 17-5. PTM 软件定时器模式	179
图 17-6. PCA 软件定时器模式	180
图 17-7. PTM0 高速输出模式.....	181
图 17-8. PTM 高速输出模式	182
图 17-9. PTM 高速输出模式	183
图 17-10. PTM0 缓冲 8 位 PWM 模式	184
图 17-11. PTM 无缓冲 10/12/16 位 PWM 模式	185
图 17-12. PTM 缓冲 10/12/16 位 PWM 模式 (带死区控制)	186
图 17-13. PWM 带死区控制的波形	187
图 17-14. 边沿对齐和中心对齐的 PWM 波形.....	188
图 17-15. PWM 中止控制的锁存模式波形	189
图 17-16. 中止控制的逐周期模式波形	189
图 17-17. PTM0 PWM 中止控制源	190
图 17-18. PTM0 模块输出控制.....	191
图 17-19. POEn 对齐输出控制 (例如.PWM 边沿对齐的波形).....	192
图 17-20. 可变分辨率的 PWM 中心对齐	196
图 18-1. PCA1 方框图.....	197
图 18-2. PCA1 定时器/计数器	198
图 18-3. PCA1 中断系统	200
图 18-4. PCA1 捕获模式	205
图 18-5. PCA1 缓冲捕捉模式 (BMEn1=1, n= 0)	206
图 18-6. PCA1 软件定时器模式	207
图 18-7. PCA1 软件定时器模式	208

图 18-8. PCA1 软件定时器模式	209
图 18-9. PCA1 高速输出模式	210
图 18-10. PCA1 高速输出模式	211
图 18-11. PCA1 高速输出模式	212
图 18-12. PCA1 缓冲 8 位 PWM 模式	213
图 18-13. PCA1 无缓冲 10/12/16 位 PWM 模式	214
图 18-14. PCA1 缓冲 10/12/16 位 PWM 模式 (带死区控制)	215
图 18-15. PCA1 PWM 带死区控制的波形	216
图 18-16. 边沿对齐和中心对齐的 PWM 波形	217
图 18-17. PWM 中止控制的锁存模式波形	218
图 18-18. PWM 中止控制的逐周期模式波形	218
图 18-19. PCA1 PWM 中止控制源	219
图 18-20. PCA1 模块输出控制	220
图 18-21. C1POEn 对齐输出控制 (例如 PWM 边沿对齐的波形)	220
图 18-22. 可变分辨率的 PWM 中心对齐	223
图 19-1. 模式 1 数据帧	225
图 19-2. 模式 2, 3 数据帧	225
图 19-3. 串口 0 模式 0	226
图 19-4. 模式 0 发送波形	226
图 19-5. 模式 0 接收波形	226
图 19-6. 串口模式 1, 2, 3	227
图 19-7. UART0 帧错误检测	228
图 19-8. UART0 多处理器通讯	229
图 19-9. 自动地址识别	229
图 19-10. 串口 0 模式 4, 单主机和单从机架构 (n = 0)	231
图 19-11. 串口 0 模式 4, 单主机和多从机架构 (n = 0)	231
图 19-12. 串口 0 模式 4 发送波形 (n = 0)	232
图 19-13. S0 模式 1, 2 和 3 的波特率选择	233
图 19-14. S0 ARGB 模式波特率公式: (n=0)	243
图 19-15. S0 ARGB 模式总线复位时间 (n=0)	244
图 19-16. S0 ARGB 模式中断 (n=0)	245
图 19-17. S0BRG 配置	248
图 19-18. S0 8 位定时器模式	248
图 19-19. S0BRG 时钟输出 (S0BRG 在 8 位定时器模式下)	249
图 19-20. S0BRG 时钟输出 (S0BRG 用于 UART 模式)	249
图 20-1. S1BRG 结构 (S1TME=0)	255
图 20-2. 串口 1 模式 4, 单主机和单从机架构 (n = 1)	260
图 20-3. 串口 1 模式 4, 单主机和多从机架构 (n = 1)	260
图 20-4. 串口 1 模式 4 发送波形 (n = 1)	261
图 20-5. S1BRG 8 位定时器模式结构 (S1TME=1)	262
图 20-6. S1 ARGB 模式波特率公式: (n=1)	266
图 20-7. S1 ARGB 模式总线复位时间 (n=1)	266
图 20-8. 模式 9 (ARGB) 中断 (n=1)	267
图 20-9. S1BRG 时钟输出 (S1BRG 在 8 位定时器模式)	268
图 20-10. S1BRG 时钟输出 (S1BRG 用于 UART 模式)	268
图 20-11. UART0 额外波特率源	269
图 21-1. SPI 方框图	274
图 21-2. SPI 单主机和单从机结构	275
图 21-3. SPI 双设备结构, 既是主机也是从机	275
图 21-4. SPI 单主机和多从机结构	275
图 21-5. SPI 在 CPHA=0 时从机传送格式	278
图 21-6. 在 CPHA=1 时从机传送格式	278
图 21-7. SPI 在 CPHA=0 时主机传送格式	279
图 21-8. SPI 在 CPHA=1 时主机传送格式	279

图 21-9. SPI 从机在菊花链连接结构	280
图 22-1. TWI/I2C 总线互联	284
图 22-2. TWI/I2C 方框图	284
图 22-3. 多从机地址识别	287
图 23-1. 串行接口侦测结构	302
图 24-1. 蜂鸣器发生器	304
图 25-1. 键盘中断 (KBI) 结构	305
图 26-1. CRC16 结构	308
图 26-2. CRC32 结构	309
图 26-3. BOREV 结构	309
图 26-4. EDC45 模式 0 结构 (4 位到 5 位编码)	310
图 26-5. EDC45 模式 1 结构 (5 位到 4 位解码)	310
图 27-1. OPA 方框图	313
图 28-1. PGA 方框图	320
图 29-1. ADC 方框图	323
图 29-2. ADC 中断	326
图 29-3. ADC 转换时序	326
图 29-4. ADC 窗口检测	327
图 30-1. 模拟比较器 0 方框图	336
图 30-2. 模拟比较器 1 方框图	337
图 31-1. IVR 方框图	343
图 32-1. 6P32 Flash 存储器配置	345
图 32-2. ISP/IAP 页擦除流程	347
图 32-3. ISP/IAP 页擦除操作的示例代码	347
图 32-4. ISP/IAP 字节编程流程	348
图 32-5. ISP/IAP 字节编程的示例代码	348
图 32-6. ISP/IAP 字节读取流程	349
图 32-7. ISP/IAP 字节读取的范例代码	349
图 32-8. ISP 示例代码	357
图 36-1. 电源电路	374
图 36-2. 复位电路	374
图 36-3. XTAL 振荡电路	375
图 36-4. ICP 和 OCD 接口电路	376
图 36-5. 经 ICP(MLink)的独立编程	377
图 36-6. MLink ICE 功能的系统框图	378
图 37-1. 外部时钟驱动波形	382
图 37-2. 移位寄存器模式时序波形	388
图 37-3. CPHA=0 时 SPI 主机传送波形	389
图 37-4. CPHA=1 时 SPI 主机传送波形	389
图 37-5. CPHA=0 时 SPI 从机传送波形	390
图 37-6. CPHA=1 时 SPI 从机传送波形	390
图 37-7. TWI/I2C 总线的典型应用和时序图	391
图 39-1. LQFP-32 (7mm X 7mm)封装尺寸	395
图 39-2. QFN-32 (4mm X 4mm X 0.55mm)封装尺寸	396
图 39-3. SSOP-28 (150 mil)封装尺寸	397

表列表

表 3-1. SFR 图 (页 0~F).....	23
表 3-2. SFR 位分配(页 0~F).....	25
表 3-3. 辅助 SFR 图 (P 页).....	29
表 3-4. 辅助 SFR 位分配(P 页)	30
表 4-1. 引脚描述	33
表 8-1. DMA 数据路径选择	54
表 14-1. 可用 I/O 引脚数量.....	91
表 14-2. 端口 3 配置设定	97
表 14-3. 通用端口配置设定	97
表 15-1. 中断源	104
表 15-2. 中断源标志位	106
表 15-3. 中断使能	109
表 15-4. 中断优先级	110
表 17-1. PCAx/PTMx 模块模式.....	178
表 18-1. PCA1 模块模式	204
表 19-1. 串口 0 模式选择	224
表 19-2. 串口 0 模式 4 的 SPI 模式配置	232
表 19-3. SMOD2 在模式 2 的应用标准	233
表 19-4. S0 模式 2 波特率@ F _{SYSClk} =11.0592MHz	234
表 19-5. S0 模式 2 波特率@ F _{SYSClk} =12.00MHz	234
表 19-6. SMOD2 在模式 1 和 3 使用定时器 1&SnBRG 的应用标准.....	234
表 19-7. 定时器 1 或 SnBRG 产生常用波特率@ F _{SYSClk} =11.0592MHz	235
表 19-8. 定时器 1 或 SnBRG 产生常用波特率 @ F _{SYSClk} =22.1184MHz	235
表 19-9. 定时器 1 或 SnBRG 产生常用波特率@ F _{SYSClk} =12.0MHz	236
表 19-10. 定时器 1 或 SnBRG 产生常用波特率@ F _{SYSClk} =24.0MHz	236
表 19-11. 定时器 1 或 SnBRG 产生常用波特率@ F _{SYSClk} =29.4912MHz	236
表 19-12. 定时器 1 或 SnBRG 产生常用波特率@ F _{SYSClk} =44.2368MHz	237
表 19-13. 定时器 1 或 SnBRG 产生常用波特率 @ F _{SYSClk} =32MHz	237
表 19-14. 定时器 1 或 SnBRG 产生常用波特率@ F _{SYSClk} =48.0MHz	238
表 19-15. SMOD2 在模式 1&3 使用定时器 2 的应用标准	239
表 19-16. 定时器 2 产生常用波特率@ F _{SYSClk} =11.0592MHz	239
表 19-17. 定时器 2 产生常用波特率@ F _{SYSClk} =22.1184MHz	240
表 19-18. 定时器 2 产生常用波特率 @ F _{SYSClk} =12.0MHz	240
表 19-19. 定时器 2 产生常用波特率@ F _{SYSClk} =24.0MHz	240
表 19-20. 定时器 2 产生常用波特率 @ F _{SYSClk} =29.4912MHz	241
表 19-21. 定时器 2 产生常用波特率 @ F _{SYSClk} =44.2368MHz	241
表 19-22. 定时器 2 产生常用波特率 @ F _{SYSClk} =32MHz	242
表 19-23. 定时器 2 产生常用波特率 @ F _{SYSClk} =48.0MHz	242
表 19-24. SMOD2 在分割定时器 2 模式 1&3 下的应用条件	243
表 20-1. 串口 1 模式选择	254
表 20-2. S1 模式 2 波特率@ F _{SYSClk} =11.0592MHz	256
表 20-3. S1 模式 2 波特率@ F _{SYSClk} =12.00MHz	256
表 20-4. S1BRG 产生常用波特率@ F _{SYSClk} =11.0592MHz	257
表 20-5. S1BRG 产生常用波特率@ F _{SYSClk} =22.1184MHz	257
表 20-6. S1BRG 产生常用波特率@ F _{SYSClk} =12.0MHz	257
表 20-7. S1BRG 产生常用波特率@ F _{SYSClk} =24.0MHz	258
表 20-8. S1BRG 产生常用波特率@ F _{SYSClk} =29.4912MHz	258
表 20-9. S1BRG 产生常用波特率@ F _{SYSClk} =44.2368MHz	258
表 20-10. S1BRG 产生常用波特率@ F _{SYSClk} =32.0MHz	259
表 20-11. S1BRG 产生常用波特率@ F _{SYSClk} =48.0MHz	259
表 20-12. 串口 1 模式选择	260
表 20-13. 串口 1 模式 4 的 SPI 模式配置	261
表 21-1. SPI 主机和从机选择	276

表 21-2. SPI 串行时钟速率.....	277
表 21-3. SPI 模式定义.....	278
表 22-1. TWI0/I2C0 串行时钟速率.....	296
表 22-2. TWI1/I2C1 串行时钟速率.....	299
表 26-1. 4 位/5 位格式转换表.....	309
表 27-1. OPAn 模式选择.....	314
表 27-2. OPAn 功率级别选择.....	314
表 28-1. PGA 正端输入选择.....	320
表 29-1. 启用通道扫描后添加 ADC 通道信息.....	327
表 29-2. 无通道扫描下 ADC DMA 数据传送顺序.....	328
表 29-3. 通道扫描下 ADC DMA 数据传送顺序.....	328
表 30-1. AC0 参考准位 NVRL = 0, 选择高区域.....	338
表 30-2. AC0 参考准位 NVRL = 1, 选择低区域.....	338
表 36-1. 晶体振荡电路 C1 和 C2 的参考电容.....	375
表 38-1. 指令集.....	392
表 40-1. 版本历史.....	398

1. 概述

MG82F6P32是基于80C51的高效1-T结构的单芯片微处理器，每条指令需要1~7时钟信号(比标准的8051快6~7倍)，与标准8051指令集兼容。因此在与标准8051有同样的处理能力的情况下，**MG82F6P32**只需要非常低的运行速度，同时由此能很大程度的减少耗电量。

MG82F6P32有 **32K**字节的内置Flash存储器用于保存代码。Flash存储器可以通过串行模式编程(ICP，在电路编程)或者ISP模式进行编程的能力。同时，也提供在应用编程(IAP)的能力。ISP和ICP让使用者无需从产品中取下微控制器就可以下载新的代码；IAP意味着应用程序正在运行时，微控制器能够在Flash中写入非易失数据。这些功能都由内置的电荷泵提供编程用的高压不需要外部提供高压。

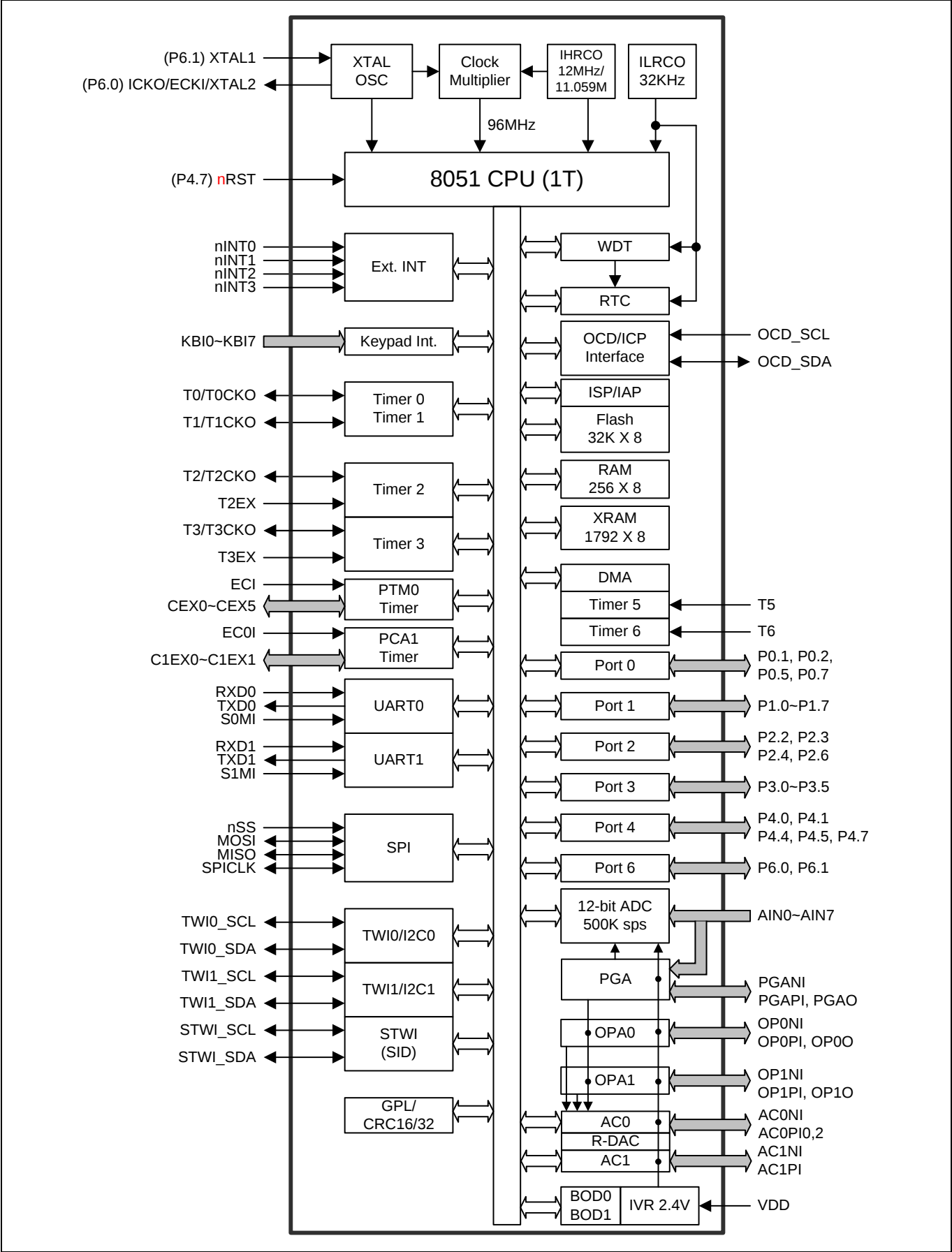
MG82F6P32除了80C52MCU的标准功能，256字节的随机存储器，4个外部中断可选高/低触发，1个多源4级中断控制，2个带ARGB以及应用于USB PD协议的MBC ENDEC的串口(UART0/1)和4个定时/计数器。另外，**MG82F6D64/32**有**29**个I/O 端口引脚，**1792**字节外部数据存储器(XRAM)，**500K sps** 12位ADC，2个OPA，1个PGA，5个16位定时器，1个**6通道**带死区控制的PTM，1个**2通道**带死区控制PWM的PCA，1个8位SPI，3个TWI/I2C (TWI0/I2C0, TWI1/I2C1 and STWI/ SI2C)，1个键盘中断，2个模拟比较器，看门狗定时器，实时时钟(RTC)模块，2个低电压监测器，1个外部时钟输入ECKI (P6.0)，1个片上振荡器(与P6.0和P6.1共享)，1个高精度的内部振荡(IHRCO)，1个片内时钟倍频器(CKM)来产生高速时钟源，1个低速的内部RC振荡器(ILRCO)和2个串口(UART0~1)用来促进多处理器的通讯，及一个速度增强设备(X2 模式)。支持三种不同DMA传输类型M2P(XRAM到外设)，P2M(外设到XRAM)和P2P(外设到外设)来增强传输和降低CPU负载。

MG82F6P32有多种工作模式可以减少耗电量：空闲模式，掉电模式，慢频模式，副频模式，RTC模式，watch 模式和 monitor 模式。在空闲模式下，CPU被冻结而外围模块和中断系统依然活动。在掉电模式下，随机存储器RAM和特殊功能寄存器SFR的值被保存，而其他所有功能被终止。最重要的是，在掉电模式下的微控制器可以被多种中断或复位唤醒。在慢频模式，用户可以通过8位的系统时钟分频器减慢系统速度以减少耗电量。选择副频模式系统时钟来自内部低速振荡器CPU 用一个特别慢的速度在运行。实时时钟(RTC)模式支持所有模式下的实时时钟功能，watch模式，在掉电模式或空闲模式下WDT溢出作为一个自动定时器来唤醒CPU。Monitor模式，在掉电模式监测电压，当电压特别低的时候会复位。

另外，**MG82F6P32** 装配有笙泉独家的(OCD)接口可以用于在线仿真(ICE)，OCD 接口提供在片内和在系统不干扰调试并且不占用任何资源。支持 ICE 应用中的几个必须的操作例如：复位、全速、停止、单步、全速到光标和断点设置。软件开发期间使用者不需要使用任何的开发板或者传统的 ICE 上应用的插头转接器，使用者只需要连接好 OCD 接口，这强有力的接口使得开发非常容易。

2. 方框图

图 2-1. MG82F6P32 方框图



3. 特殊功能寄存器

3.1. SFR 图(页 0~F)

表 3-1. SFR 图 (页 0~F)

		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8	0		CH	CCAP0H	CCAP1H	CCAP2H	CCAP3H	CCAP4H	CCAP5H
	1	P6	CH1	C1CAP0H	C1CAP1H				
	2~F								
F0	0	B	PAOE	PCAPWM0	PCAPWM1	PCAPWM2	PCAPWM3	PCAPWM4	PCAPWM5
	1			C1PWM0	C1PWM1				
	2~7								
	8		C0AOE1						
	9		C1AOE0						
	A		C1AOE1						
	B~F		--	--	--				
E8	0	P4	CL	CCAP0L	CCAP1L	CCAP2L	CCAP3L	CCAP4L	CCAP5L
	1		CL1	C1CAP0L	C1CAP1L				
	2~F								
E0	0~F	ACC	WDTCR	IFD	IFADRH	IFADRL	IFMT	SCMD	ISPCR
D8	0	CCON	CMOD	CCAPM0	CCAPM1	CCAPM2	CCAPM3	CCAPM4	CCAPM5
	1	C1CON	C1MOD	C1CAPM0	C1CAPM1				
	2~F								
D0	0	PSW	SIADR	SIDAT	SISTA	SICON	KBPATN	KBCON	KBMASK
	1		SI1ADR	SI1DAT	SI1STA	SI1CON			
	2		SIA2	SIA2M	--	SICR1			
	3		SI1A2	SI1A2M	--	SI1CR1			
	4~F		--	--	--	--			
C8	0	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	CLRL	CHRL
	1	T3CON	T3MOD	RCAP3L	RCAP3H	TL3	TH3	CL1RL	CH1RL
	3	T5CON	--	TLR5	THR5	TL5	TH5		
	4	T6CON	--	TLR6	THR6	TL6	TH6	--	--
	5~F	--	--	--	--	--	--		
C0	0	XICON	XICFG	--	ADCFG0	ADCON0	ADCDL	ADCDH	CKCON0
	1		XICFG1		ADCFG1				
	2		XICFG2		ADCFG2				
	3				ADCFG3				
	4				ADCFG4				
	5				ADCFG5				
	6				--				
	7				ADCFG7				
	8~A		--		--				
	B				ADCFG11				
	C				ADCFG12				
	D				ADCFG13				
	E				ADCFG14				
	F				--				
B8	0	IP0L	SADEN/ S0CR1	OPSTA	--	PWMCR	CRC0DA	--	CKCON1
	1			OP0CON0		PDT CRA			
	2			OP1CON0		--			
	3~4			--		--			
	5			PGA CON0		--			
	6			PGA CON1		--			
	7			OPCMP		--			
	8~9			--		--			
	A			--		C1PWMCR			
	B			--		C1PDT CR			
	C~F			--		--			
B0	0	P3	P3M0	P3M1	P4M0	--	--	--	IP0H
	1				--	--	P6M0		
	2				--	PDRV C0	--		
	3				--	PDRV C1	--		
	4~F								
A8	0~F	IE	SADDR/ EDC45			SFRPI	EIE1	EIP1L	EIP1H

A0	0	P2	AUXR0	AUXR1	AUXR2	AUXR3	EIE2	EIP2L	EIP2H
	1					AUXR4			
	2					AUXR5			
	3					AUXR6			
	4					AUXR7			
	5					AUXR8			
	6					AUXR9			
	7					AUXR10			
	8					AUXR11			
	9					AUXR12			
	D					AUXR16			
	E					AUXR17			
	F								
98	0	S0CON/ S0AGCR	S0BUF	S0BRT	S0BRC	S0CFG	S0CFG1	AC0CON	AC0MOD
	1	S1CON/ S1AGCR	S1BUF	S1BRT	S1BRC/ S1CR1	S1CFG	S1CFG6	AC1CON	AC1MOD
	2								
	3						--	AC0CR1	AC0CF0
	4						--		AC1CF0
	5~F	--	--	--	--	--	--	--	--
90	0	P1	P1M0	P1M1	P0M0	DMACR0	P2M0	BOREV	PCON1
	1			P2M1	T2MOD1		TREN0		
	2			P4M1	T3MOD1		TRLC0		
	3			P6M1			TSPC0		
	4			P0M1	--		--		
	7			P3FDC	--		--		
	8			P1FDC	--	DMACG0	--		
	9			P2FDC	--	DMADS0	--		
	A			P4FDC					
	B			P0FDC	--		--		
	F			P3M2					
88	0~F	TCON	TMOD	TL0	TL1	TH0	TH1	SFIE	XRPS
80	0~7	P0	SP	DPL	DPH	SPSTAT	SPCON	SPDAT	PCON0
	8						SPCR1		
	9~F						--		
		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F

注: 用户需要设置 SFRPI=0x00 ~ 0x0F 作为 SFR 的页访问。
(MCU 在中断时不会保留 SFRPI 的值。用户需要使用软件来保留 SFRPI 的值。)

SFRPI: SFR 页索引寄存器

SFR 页 = 0~F

SFR 地址 = 0xAC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	IDX3	IDX2	IDX1	IDX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: 保留位。写寄存器时，此位必须写“0”。

Bit 3~0: SFR 页索引

IDX[3:0]	可选页
0000	页 0
0001	页 1
0010	页 2
0011	页 3
.....	
.....	
.....	
1111	页 F

3.2. SFR 位分配 (Page 0~F)

表 3-2. SFR 位分配(页 0~F)

符号	描述	地址 (HEX)	页 (HEX)	位地址及符号								复位值
				位-7	位-6	位-5	位-4	位-3	位-2	位-1	位-0	
P0	端口 0	80	0~F	P0.7		P0.5			P0.2	P0.1		10100110
SP	堆栈指针	81	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000111
DPL	数据指针低 8 位	82	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
DPH	数据指针高 8 位	83	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SPSTAT	SPI 状态寄存器	84	0~F	SPIF	WCOL	THRF	SPIBSY	MODF				00000000
SPCON	SPI 控制寄存器	85	0~7	SSIG	SPEN	DORD	MSTR	CPOL	CPHA			00000100
SPCR1	SPI 数据寄存器	85	8	SPSOES	SPFACE			SPIOM0	SPR2	SPR1	SPR0	00000000
SPDAT	电源控制寄存器 0	86	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PCON0	定时器控制寄存器	87	0~F	SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL	00010000
TCON	定时器模式寄存器	88	0~F	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000
TMOD	定时器 0 低 8 位	89	0~F	T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0	00000000
TL0	定时器 1 低 8 位	8A	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL1	定时器 0 高 8 位	8B	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH0	定时器 1 高 8 位	8C	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH1	系统标志中断使能	8D	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SFIE	XRAM 页选择	8E	0~F	SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE	01100000
XRPS	堆栈指针	8F	0~F	0	0	0	0	0	.2	.1	.0	00000000
P1	端口 1	90	0~F	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	11111111
P1M0	P1 模式寄存器 0	91	0~F	P1M0.7	P1M0.6	P1M0.5	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0	00000000
P1M1	P1 模式寄存器 1	92	0	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	11111111
P2M1	P2 模式寄存器 1	92	1	1	P2M1.6	1	P2M1.4	P2M1.3	P2M1.2	1	1	11111111
P4M1	P4 模式寄存器 1	92	2	P4M1.7	1	P4M1.5	P4M1.4	1	1	P4M1.1	P4M1.0	11111111
P6M1	P6 模式寄存器 1	92	3	1	1	1	1	1	1	P6M1.1	P6M1.0	11111111
P0M1	P0 模式寄存器 1	92	4	P0M1.7	1	P0M1.5	1	1	P0M1.2	P0M1.1	1	11111111
P3FDC	P3 快速驱动控制	92	7	0	0	P3FDC.5	P3FDC.4	P3FDC.3	P3FDC.2	P3FDC.1	P3FDC.0	00000000
P1FDC	P1 快速驱动控制	92	8	P1FDC.7	P1FDC.6	P1FDC.5	P1FDC.4	P1FDC.3	P1FDC.2	P1FDC.1	P1FDC.0	00000000
P2FDC	P2 快速驱动控制	92	9	0	P2FDC.6	0	P2FDC.4	P2FDC.3	P2FDC.2	0	0	00000000
P4FDC	P4 快速驱动控制	92	A	P4FDC.7	0	P4FDC.5	P4FDC.4	0	0	P4FDC.1	P4FDC.0	00000000
P0FDC	P0 快速驱动控制	92	B	P0FDC.7	0	P0FDC.5	0	0	P0FDC.2	P0FDC.1	0	00000000
P3M2	P3 模式寄存器 2	92	F	1	1	P3M2.5	P3M2.4	P3M2.3	P3M2.2	P3M2.1	P3M2.0	11111111
P0M0	P0 模式寄存器 0	93	0	P0M0.7	0	P0M0.5	0	0	P0M0.2	P0M0.1	0	00000000
T2MOD1	定时器 2 模式 1 寄存器	93	1	TL2CS	TF2IG	TL2IS	T2CKS	T2MS1	CP2S2	CP2S1	CP2S0	00000000
T3MOD1	定时器 3 模式 1 寄存器	93	2	TL3CS	TF3IG	TL3IS	T3CKS	T3MS1	CP3S2	CP3S1	CP3S0	00000000
DMACR0	DMA 控制寄存器.0	94	0~7	0	0	0	0	DMAE0	DMAS0	DIE0	DCF0	00000000
DMACG0	DMA 配置寄存器.0	94	8	PDMAH	PDMAL	CRCW0	0	EXTS10	EXTS00	0	LOOP0	00000000
DMADS0	DMA 数据路径选择 0	94	9	DSS30	DSS20	DSS10	DSS00	DDS30	DDS20	DDS10	DDS00	00000000
P2M0	P2 模式寄存器 0	95	0	0	P2M0.6	0	P2M0.4	P2M0.3	P2M0.2	0	0	00000000
TREN0	定时器使能寄存器 0	95	1	TR2E	TR3E	0	0	0	0	C1CR	PTMCR	00000000
TRLC0	定时器重载控制寄存器 0	95	2	T2RLC	T3RLC	0	0	0	0	C1RLC	PTMRLC	00000000
TSPC0	定时器停止控制寄存器 0	95	3	T2SC	T3SC	0	0	0	0	C1SC	PTMSC	00000000
BOREV	位序颠倒	96	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PCON1	电源控制寄存器 1	97	0~F	SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF	00000000
S0CON	串口 0 控制寄存器	98	0	SM00 /FE	SM10	SM20	REN0	TB80	RB80	TI0	RI0	00000000
S0AGCR	串口 0 ARGB 控制	98	0	SM00	SM10	S0DTY1	S0GFIE	S0IDT11	S0IDT01	S0AGBF	S0AGBRF	00000000
S1CON	串口 1 控制寄存器	98	1~2	SM01	SM11	SM21	REN1	TB81	RB81	TI1	RI1	00000000
S1AGCR	串口 1 ARGB 控制	98	1~2	SM01	SM11	S1DTY1	S1GFIE	S1IDT11	S1IDT01	S1AGBF	S1AGBRF	00000000
S0BUF	串口 0 缓存	99	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S1BUF	串口 1 缓存	99	1~2	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S0BRT	S0 波特率定时器	9A	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S1BRT	S1 波特率定时器	9A	1~2	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S0BRC	S0 波特率计数器	9B	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S1BRC	S1 波特率计数器	9B	1~2	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S1CR1	S1 控制 1	9B	1~2	M6RER1	TXDO1	SOPWEC	TOTXD1	RMRI1	EXTFLT	TXINV1	RXINV1	00000000
S0CFG	S0 配置寄存器	9C	0	URTS	SMOD2	0	SM30	S0DOR	BTI	UTIE	SMOD3	00001000
S1CFG	S1 配置寄存器	9C	1~2	SM31	S1MOD3	S1DOR	S1TR	S1MOD1	S1TX12	S1CKOE	S1TME	00100000
S0CFG1	S0 配置寄存器 1	9D	0	SBF0	TXER0	S0SB16	ATBR0	TXRX0	SYNC0	0	0	00000000
S1CFG6	S1 配置寄存器 6	9D	1~2	PDOEC1	M6TGU1	M6TXB1	M6IDL1	M6TUR1	M6ROR1	PDOE1	M6PDE1	00010000
AC0CON	AC0 控制寄存器	9E	0	AC0LP	AC0PDX	AC0OUT	AC0F	AC0EN	AC0INV	AC0M1	AC0M0	00x00000
AC1CON	AC1 控制寄存器	9E	1	AC1LP	AC1PDX	AC1OUT	AC1F	AC1EN	AC1INV	AC1M1	AC1M0	00x00000
AC0CR1	AC0 控制寄存器 1	9E	3	GF	AC0HC1	AC0HC0		GF	AC0FLT1	AC0FLT0	AC0TR	00000000
AC0MOD	AC0 模式寄存器	9F	0	NVRS3	NVRS2	NVRS1	NVRS0	NVRL		AC0PIS1	AC0PIS0	00000000
AC1MOD	AC1 模式寄存器	9F	1	0	AC1HC1	AC1HC0	AC1NIS	NVRL1	AC1FLT1	AC1FLT0	AC1TR	00000000
P2	端口 2	A0	0~F	0	P2.6	0	P2.4	P2.3	P2.2	0	0	01011100

符号	描述	地址 (HEX)	页 (HEX)	位地址及符号								复位值
				位-7	位-6	位-5	位-4	位-3	位-2	位-1	位-0	
AUXR0	辅助寄存器 0	A1	0-F	0	0	C1BKF	PBKF	0	0	INT1H	INT0H	00000000
AUXR1	辅助寄存器 1	A2	0-F	OP1Fr	OP0Fr	CRCDS1	CRCDS0	0	AC1Fr	AC0Fr	DPS	00000000
AUXR2	辅助寄存器 2	A3	0-F	STAF	STOF	C1PLK	COPLK	T1X12	T0X12	0	0	00000000
AUXR3	辅助寄存器 3	A4	0	T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL	00000000
AUXR4	辅助寄存器 4	A4	1	T2PS1	T2PS0	T1PS1	T1PS0	AC1OE		AC0OE	--	00000000
AUXR5	辅助寄存器 5	A4	2			C0PPS1	C0PPS0	EC1IPS0	C1COPS	ECIPS0	C0COPS	00000000
AUXR6	辅助寄存器 6	A4	3	KBIHPS1	KBIHPS0	KBILPS1	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS	00000000
AUXR7	辅助寄存器 7	A4	4	--	--	C0CKOE	C1CKOE	0	0	0	0	00000000
AUXR8	辅助寄存器 8	A4	5		C1ICS1	1ICS0	0	0	S1COPS	T3PS1	T3PS0	00000000
AUXR9	辅助寄存器 9	A4	6	SIDPS1	SIDPS0	T1G1	T0G1			S1PS1	S1PS0	00000000
AUXR10	辅助寄存器 10	A4	7			SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD	00000000
AUXR11	辅助寄存器 11	A4	8			I2C1PS1	I2C1PS0	RX1S0		T1CKOE	T0CKOE	00000000
AUXR12	辅助寄存器 12	A4	9	CRCDS2	CRCM0	PDOES1	PDOES0	EDCM0	GPLC0			00000000
AUXR16	辅助寄存器 16	A4	D			C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0	00000000
AUXR17	辅助寄存器 17	A4	E							C1OPS1	C1OPS0	00000000
EIE2	扩展中断使能 2	A5	0-F	0	ETWI1	EPCA1	0	0	0	EAC1	ET3	00000000
EIP2L	扩展中断优先级 2 低位	A6	0-F	0	PTWI1L	PPCA1L	0	0	0	PAC1L	PT3L	00000000
EIP2H	扩展中断优先级 2 高位	A7	0-F	0	PTWI1H	PPCA1H	0	0	0	PAC1H	PT3H	00000000
IE	中断使能	A8	0-F	EA	EDMA	ET2	ES0	ET1	EX1	ET0	EX0	00000000
SADDR	从机地址	A9	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
EDC45	4b5b ENDEC	A9	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SFRPI	SFR 页索引	AC	0-F	0	0	0	0	IDX3	IDX2	IDX1	IDX0	00000000
EIE1	扩展中断使能 1	AD	0-F	EAC0	ETWI0	EKB	ES1	ESF	EPTM0	EADC	ESPI	00000000
EIP1L	扩展中断优先级 1 Low	AE	0-F	PAC0L	PTWI0L	PKBL	PS1L	PSFL	PPCAL	PADCL	PSPIL	00000000
EIP1H	扩展中断优先级 1 High	AF	0-F	PAC0H	PTWI0H	PKBH	PS1H	PSFH	PPCAH	PADCH	PSPIH	00000000
P3	端口 3	B0	0-F	0	0	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	00111111
P3M0	P3 模式寄存器 0	B1	0-F	0	0	P3M0.5	P3M0.4	P3M0.3	P3M0.2	P3M0.1	P3M0.0	00000000
P3M1	P3 模式寄存器 1	B2	0-F	0	0	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0	00000000
P4M0	P4 模式寄存器 0	B3	0	P4M0.7	0	P4M0.5	P4M0.4	0	0	P4M0.1	P4M0.0	10110000
PDRVC0	端口驱动控制 0	B4	2	P3DC1	P3DC0	P2DC1	P2DC0	P1DC1	P1DC0	P0DC1	P0DC0	00000000
PDRVC1	端口驱动控制 1	B4	3	0	0	0	0	0	0	P4DC1	P4DC0	00000000
P6M0	P6 模式寄存器 0	B5	1							P6M0.1	P6M0.0	00000000
IPOH	中断优先级 0 高位	B7	0-F	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	00000000
IPOH	中断优先级 0 低位	B8	0-F	PX3L	PX2L	PT2L	PSL	PT1L	PX1L	PT0L	PX0L	00000000
SADEN	从机地址掩码	B9	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S0CR1	S0 控制 1	B9	0-F	S0TR	S0TX12	S0TCK	S0RCK	S0CKOE	ARTE	0	0	00000000
OPSTA	OPA 状态寄存器	BA	0							OP1CMPF	OP0CMPF	00000000
OP0CON0	OPA0 控制寄存器 0	BA	1	OP0EN	OPPWR	OP0LP	OP0IG	OP0PIS1	OP0PIS0	OP0NIS1	OP0NIS0	00000000
OP1CON0	OPA1 控制寄存器 0	BA	2	OP1EN		OP1LP	OP1IG	OP1PIS1	OP1PIS0	OP1NIS1	OP1NIS0	00000000
PGACON0	PGA 控制寄存器 0	BA	5	PGEN		PGPL	PGAOE	PGPIS1	PGPIS0			00000000
PGACON1	PGA 控制寄存器 1	BA	6			PGENS1GN	0	PGGNM	PGGN2	PGGN1	PGGN0	00000000
OPCMP	OPA 比较模式	BA	7			OP1PDX	OP0PDX	OP1M1	OP1M0	OP0M1	OP0M0	00000000
PWMCR	PWM 控制寄存器	BC	0	PCAE	COOFS	PBKM	PBKE1.1	PBKE1.0	PBKE0.2	PBKE0.1	PBKE0.0	00000000
PDTCRA	PWM 死区控制寄存器 A	BC	1	DTPS1	DTPS0	DT.5	DT.4	DT.3	DT.2	DT.1	DT.0	00000000
C1PWMCR	PWM 控制寄存器	BC	A	PCA1E	C1OFS	C1BKM	C1BKE1.1	C1BKE1.0	C1BKE0.2	C1BKE0.1	C1BKE0.0	00000000
C1PDTCR	C1 PWM 死区控制	BC	B	C1DTPS1	C1DTPS0	C1DT.5	C1DT.4	C1DT.3	C1DT.2	C1DT.1	C1DT.0	00000000
CRC0DA	CRC0 数据输出	BD	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CKCON1	时钟控制寄存器 1	BF	0-F	XTOR	PLLOCKF	MCKSTA1	MCKSTA0	OSCSTA3	OSCSTA2	OSCSTA1	OSCSTA0	00010001
XICON	扩展中断控制	C0	0-F	INT3H	EX3	IE3	IT3	INT2H	EX2	IE2	IT2	00000000
XICFG	Ext. INT. 配置	C1	0	INT1IS1	INT1IS0	INT0IS1	INT0IS0	X3FLT	X2FLT	X1FLT	X0FLT	00000000
XICFG1	Ext. INT. 配置 1	C1	1	INT1IS2	INT0IS2	INT2IS1	INT2IS0	X3FLT1	X2FLT1	X1FLT1	X0FLT1	00000000
XICFG2	Ext. INT. 配置 2	C1	2	0	0	0	0	INT3IS2	INT3IS1	INT3IS0	INT2IS2	00000000
ADCFG0	ADC 配置寄存器 0	C3	0	ADCKS2	ADCKS1	ADCKS0	ADRJ	ACHS	SMPF	ADTM1	ADTM0	00000000
ADCFG1	ADC 配置寄存器 1	C3	1	IGADCI	EADCWI	SMPFIE	SIGN	AOS.3	AOS.2	AOS.1	AOS.0	00000000
ADCFG2	ADC 配置寄存器 2	C3	2	SHT.7	SHT.6	SHT.5	SHT.4	SHT.3	SHT.2	SHT.1	SHT.0	00000000
ADCFG3	ADC 配置寄存器 3	C3	3	ADPS1	ADPS0	ADHS	ADHS1	ARES1	ARES0	ADES0	0	01100000
ADCFG4	ADC 配置寄存器 4	C3	4	ADMINS	ADWM0	ADTM3	ADTM2	0	VRS0	DBSD	DOSD	00000000
ADCFG5	ADC 配置寄存器 5	C3	5	ASCE.7	ASCE.6	ASCE.5	ASCE.4	ASCE.3	ASCE.2	ASCE.1	ASCE.0	00000000
ADCFG7	ADC 配置寄存器 7	C3	7	0	0	0	0	ASCS3	ASCS2	ASCS1	ASCS0	00000000
ADCFG11	ADC 配置寄存器 11	C3	B	WHB.3	WHB.2	WHB.1	WHB.0	1	1	1	1	11111111
ADCFG12	ADC 配置寄存器 12	C3	C	WHB.11	WHB.10	WHB.9	WHB.8	WHB.7	WHB.6	WHB.5	WHB.4	11111111
ADCFG13	ADC 配置寄存器 13	C3	D	WLB.3	WLB.2	WLB.1	WLB.0	0	0	0	0	00000000
ADCFG14	ADC 配置寄存器 14	C3	E	WLB.11	WLB.10	WLB.9	WLB.8	WLB.7	WLB.6	WLB.5	WLB.4	00000000
ADCON0	ADC 控制 0	C4	0-F	ADCEN	ADCWI	CHS3	ADCI	ADCS	CHS2	CHS1	CHS0	00000000
ADCDL	ADC 数据低位	C5	0-F	ADCV.3	ADCV.2	ADCV.1	ADCV.0	0	0	0	0	00000000
ADCDH	ADC 数据高位	C6	0-F	ADCV.11	ADCV.10	ADCV.9	ADCV.8	ADCV.7	ADCV.6	ADCV.5	ADCV.4	00000000
CKCON0	时钟控制寄存器 0	C7	0-F	AFS	ENCKM	CKMIS1	CKMIS0	CKKS	SCKS2	SCKS1	SCKS0	00010000
T2CON	定时器 2 控制寄存器	C8	0	TF2	EXF2	RCLK/TF2L	TCLK/TL2IE	EXEN2	TR2	C/T2	CP/RL2	00000000

符号 L	描述	地址 (HEX)	页 (HEX)	位地址及符号								复位值
				位-7	位-6	位-5	位-4	位-3	位-2	位-1	位-0	
T3CON	定时器 3 控制寄存器	C8	1	TF3	EXF3	TF3L	TL3IE	EXEN3	TR3	C/T3	CP/RL3	00000000
T5CON	定时器 5 控制寄存器	C8	3	TF5	0	T5CKS1	T5CKS0	T5IE	TR5	T5GAT1	T5GAT0	00000000
T6CON	定时器 6 控制寄存器	C8	4	TF6	0	T6CKS1	T6CKS0	T6IE	TR6	T6GAT1	T6GAT0	00000000
T2MOD	定时器 2 模式寄存器	C9	0	T2SPL	TL2X12/ T2EIP	T2EXH	T2X12	TR2L	TR2LC	T2OE	T2MS0	00000000
T3MOD	定时器 3 模式寄存器	C9	1	T3SPL	TL3X12/ T3EIP	T3EXH	T3X12	TR3L	TR3LC	T3OE	T3MS0	00000000
RCAP2L	定时器 2 捕捉低位	CA	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RCAP3L	定时器 3 捕捉低位	CA	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TLR5	TL5 重载寄存器	CA	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TLR6	TL6 重载寄存器	CA	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RCAP2H	定时器 2 捕捉高位	CB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RCAP3H	定时器 3 捕捉高位	CB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
THR5	TH5 重载寄存器	CB	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
THR6	TH6 重载寄存器	CB	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL2	定时器 2 低位	CC	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL3	定时器 3 低位	CC	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL5	定时器 5 低位	CC	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL6	定时器 6 低位	CC	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH2	定时器 2 高位	CD	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH3	定时器 3 高位	CD	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH5	定时器 5 高位	CD	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH6	定时器 6 高位	CD	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CLRL	CL 重载寄存器	CE	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CL1RL	CL1 重载寄存器	CE	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CHRL	CH 重载寄存器	CF	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CH1RL	CH1 重载寄存器	CF	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PSW	程序状态字	D0	0-F	CY	AC	F0	RS1	RS0	OV	F1	P	00000000
SIADR	TWI0 地址寄存器	D1	0	.7	.6	.5	.4	.3	.2	.1	GC	00000000
SI1ADR	TWI1 地址寄存器	D1	1	.7	.6	.5	.4	.3	.2	.1	GC1	00000000
SIA2	TWI0 2 nd 地址寄存器	D1	2	.7	.6	.5	.4	.3	.2	.1	A2E	00000000
SI1A2	TWI1 2 nd 地址寄存器	D1	3	.7	.6	.5	.4	.3	.2	.1	A2E1	00000000
SIDAT	TWI0 数据寄存器	D2	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SI1DAT	TWI1 数据寄存器	D2	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SIA2M	SIA2 掩码寄存器	D2	2	SIA2M.7	SIA2M.6	SIA2M.5	SIA2M.4	SIA2M.3	SIA2M.2	SIA2M.1	1	11111111
SI1A2M	SI1A2 掩码寄存器	D2	3	SI1A2M.7	SI1A2M.6	SI1A2M.5	SI1A2M.4	SI1A2M.3	SI1A2M.2	SI1A2M.1	1	11111111
SISTA	TWI0 状态寄存器	D3	0	.7	.6	.5	.4	.3	.2	.1	.0	11111000
SI1STA	TWI1 状态寄存器	D3	1	.7	.6	.5	.4	.3	.2	.1	.0	11111000
SICON	TWI0 控制寄存器	D4	0	CR2	ENSI	STA	STO	SI	AA	CR1	CR0	00000000
SI1CON	TWI1 控制寄存器	D4	1	CR21	ENSI1	STA1	STO1	SI1	AA1	CR11	CR01	00000000
SICR1	TWI0 控制寄存器 1	D4	2					F8IE	MTSE	PAA	CR3	00000000
SI1CR1	TWI1 控制寄存器 1	D4	3					F8IE1	MTSE1	PAA1	CR31	00000000
KBPATN	键盘模式	D5	0-F	.7	.6	.5	.4	.3	.2	.1	.0	11111111
KBCON	键盘控制	D6	0-F	KBCS1	KBCS0	KBES	0	0	0	PATN SEL	KBIF	00000000
KBMASK	键盘中断掩码	D7	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCON	PTM 控制寄存器	D8	0	CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0	00000000
C1CON	PCA1 控制寄存器	D8	1	C1F	C1R					C1CF1	C1CF0	00000000
CMOD	PTM 模式寄存器	D9	0	CIDL	BME4	BME2	BME0	CPS2	CPS1	CPS0	ECF	00000000
C1MOD	PCA1 模式寄存器	D9	1	C1IDL			BME01	C1PS2	C1PS1	C1PS0	ECF1	00000000
CCAPM0	PTM 模块 0 模式	DA	0	DTE0	ECOM0	0	0	MAT0	TOG0	PWM0	ECCF0	00000000
C1CAPM0	PCA1 模块 0 模式	DA	1	DT1E0	ECOM01	CAP1P0	CAP1N0	MAT01	TOG01	PWM01	EC1CF0	00000000
CCAPM1	PTM 模块 1 模式	DB	0	0	ECOM1	0	0	MAT1	TOG1	PWM1	ECCF1	00000000
C1CAPM1	PCA1 模块 1 模式	DB	1	0	ECOM11	CAP1P1	CAP1N1	MAT11	TOG11	PWM11	EC1CF1	00000000
CCAPM2	PTM 模块 2 模式	DC	0-F	DTE2	ECOM2	0	0	MAT2	TOG2	PWM2	ECCF2	00000000
CCAPM3	PTM 模块 3 模式	DD	0-F	0	ECOM3	0	0	MAT3	TOG3	PWM3	ECCF3	00000000
CCAPM4	PTM 模块 4 模式	DE	0-F	DTE4	ECOM4	0	0	MAT4	TOG4	PWM4	ECCF4	00000000
CCAPM5	PTM 模块 5 模式	DF	0-F	0	ECOM5	0	0	MAT5	TOG5	PWM5	ECCF5	00000000
ACC	累加器	E0	0-F	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
WDTCLR	WDT 控制寄存器	E1	0-F	WREN	NSW	ENW	CLRW	WIDL	PS2	PS1	PS0	00000000
IFD	ISP Flash 数据	E2	0-F	.7	.6	.5	.4	.3	.2	.1	.0	11111111
IFADRH	ISP Flash 地址高位	E3	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
IFADRL	ISP Flash 地址低位	E4	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
IFMT	ISP 模式表	E5	0-F	MS.7	0	0	0	MS.3	MS.2	MS.1	MS.0	00000000
SCMD	ISP 系列命令	E6	0-F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
ISPCR	ISP 控制寄存器	E7	0-F	ISPEN	SWBS	SRST	CFAIL	0	0	0	0	00000000
P4	端口 4	E8	0-F	P4.7		P4.5	P4.4			P4.1	P4.0	10110011
CL	PTM 基准定时器低位	E9	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CL1	PCA1 基准定时器低位	E9	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0L	PTM 模块 0 比较低位	EA	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000

符号 L	描述	地址 (HEX)	页 (HEX)	位地址及符号								复位值
				位-7	位-6	位-5	位-4	位-3	位-2	位-1	位-0	
C1CAP0L	PCA1 模块 0 捕捉低位	EA	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP1L	PTM 模块 1 比较低位	EB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
C1CAP1L	PCA1 模块 1 捕捉低位	EB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP2L	PTM 模块 2 比较低位	EC	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP3L	PTM 模块 3 比较低位	ED	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP4L	PTM 模块 4 比较低位	EE	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP5L	PTM 模块 5 比较低位	EF	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
B	B 寄存器	F0	0~F	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00000000
PAOE	PWM 额外输出使能	F1	0~7	POE3	POE2B	POE2A	POE2	POE1	POE0B	POE0A	POE0	10011001
C0AOE1	PTM0 额外输出使能 1	F1	8	POEM0				POE5			POE4	0xxx1xx1
C1AOE0	PCA1 额外输出使能 0	F1	9					C1POE1			C1POE0	xxxx1001
C1AOE1	PCA1 额外输出使能 1	F1	A	C1POEM0								0xxxxxxx
PCAPWM0	PTM PWM0 模式	F2	0	P0RS1	P0RS0	0	0	0	P0INV	ECAP0H	ECAP0L	00000000
C1PWM0	PCA1 PWM0 模式	F2	1	P0RS11	P0RS01	0	0	0	P0INV1	ECAP0H1	ECAP0L1	00000000
PCAPWM1	PTM PWM1 模式	F3	0	P1RS1	P1RS0	0	0	0	P1INV	ECAP1H	ECAP1L	00000000
C1PWM1	PCA1 PWM1 模式	F3	1	P1RS11	P1RS01	0	0	0	P1INV1	ECAP1H1	ECAP1L1	00000000
PCAPWM2	PTM PWM2 模式	F4	0~F	P2RS1	P2RS0	0	0	0	P2INV	ECAP2H	ECAP2L	00000000
PCAPWM3	PTM PWM3 模式	F5	0~F	P3RS1	P3RS0	0	0	0	P3INV	ECAP3H	ECAP3L	00000000
PCAPWM4	PTM PWM4 模式	F6	0~F	P4RS1	P4RS0	0	0	0	P4INV	ECAP4H	ECAP4L	00000000
PCAPWM5	PTM PWM5 模式	F7	0~F	P5RS1	P5RS0	0	0	0	P5INV	ECAP5H	ECAP5L	00000000
P6	Port 6	F8	1	0	0	0	0	0	0	P6.1	P6.0	00000011
CH	PTM base timer High	F9	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CH1	PCA1 base timer High	F9	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0H	PTM 模块 0 比较高位	FA	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
C1CAP0H	PCA1 模块 0 捕捉高位	FA	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP1H	PTM 模块 1 比较高位	FB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
C1CAP1H	PCA1 模块 1 捕捉高位	FB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP2H	PTM 模块 2 比较高位	FC	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP3H	PTM 模块 3 比较高位	FD	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP4H	PTM 模块 4 比较高位	FE	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP5H	PTM 模块 5 比较高位	FF	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000

3.3. 辅助 SFR 图 (P 页)

MG82F6P32 /32 特殊功能寄存器(SFR)有一个辅助索引 P 页，它读写的方法跟标准的 8051 特殊功能寄存器的不一样。象访问 ISP/IAP 一样通过设置 IFMT 和 SCMD 来访问这个辅助的特殊功能寄存器。P 页有 256 字节有用到的为 **11 个物理字节地址**和 **6 个逻辑字节地址**。11 个物理字节地址包括 IAPLB、CKCON2、CKCON3、CKCON4、CKCON5、PCON2、PCON3、SPCON0、DCON0、RTCTM 和 RTCCR。6 个逻辑字节地址包括 PCON0、PCON1、CKCON1、WDTCR、P4 和 P6。在 0~F 页访问这 6 个逻辑地址会得到相同的 SFR 值。更多详细的信息请参考章节“**33 P 页 SFR 访问**”。

表 3-3. 辅助 SFR 图 (P 页)

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8	P6	--	--	--	--	--	--	--
F0	--	--	--	--	--	--	--	--
E8	P4	--	--	--	--	--	--	--
E0	--	WDTCR	--	--	--	--	--	--
D8	--	--	--	--	--	--	--	--
D0	--	--	--	--	--	--	--	--
C8	--	--	--	--	--	--	--	--
C0	--	--	--	--	--	--	--	CKCON0
B8	--	--	--	--	--	--	--	--
B0	--	--	--	--	--	--	--	--
A8	--	--	--	--	--	--	--	--
A0	--	--	--	--	--	--	--	--
98	--	--	--	--	--	--	--	--
90	--	--	--	--	--	--	--	PCON1
88	--	--	--	--	--	--	--	--
80	--	--	--	--	--	--	--	PCON0
78	--	--	--	--	--	--	--	--
70	--	--	--	--	--	--	--	--
68	--	--	--	--	--	--	--	--
60	--	--	--	--	--	--	--	--
58	--	--	--	--	--	--	--	--
50	--	--	--	SPHB	RTCCR	RTCTM	--	--
48	SPCON0	--	--	--	DCON0	--	--	--
40	CKCON2	CKCON3	CKCON4	--	PCON2	PCON3	--	--
38	--	--	--	--	--	--	--	--
30	--	--	--	--	--	--	--	--
28	--	--	--	--	--	--	--	--
20	--	--	--	--	--	--	--	--
18	--	--	--	--	--	--	--	--
10	--	--	--	OP0CFG0	OP1CFG0	PGACFG0	AC0CF0	AC1CF0
08	--	--	--	--	--	--	--	--
00	--	--	--	IAPLB	--	--	--	--
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F

3.4. 辅助 SFR 位分配(P 页)

表 3-4. 辅助 SFR 位分配(P 页)

符号	描述	地址	位地址和符号								复位值
			位-7	位-6	位-5	位-4	位-3	位-2	位-1	位-0	
物理字节											
IAPLB	IAP 低边界	03H	IAPLB6	IAPLB5	IAPLB4	IAPLB3	IAPLB2	IAPLB1	IAPLB0	0	
OP0CFG0	OP0 配置 0	13H	--	--	OP0OS5	OP0OS4	OP0OS3	OP0OS2	OP0OS1	OP0OS0	00xxxxxx
OP1CFG0	OP1 配置 0	14H	--	--	OP1OS5	OP1OS4	OP1OS3	OP1OS2	OP1OS1	OP1OS0	00011111
PGACFG0	PGA 配置 0	15H	--	--	PGAOS5	PGAOS4	PGAOS3	PGAOS2	PGAOS1	PGAOS0	00011111
AC0CF0	AC0 配置 0	16H	AC0OSE	AC0OS6	AC0OS5	AC0OS4	AC0OS3	AC0OS2	AC0OS1	AC0OS0	10011111
AC1CF0	AC1 配置 0	17H	AC1OSE	AC1OS6	AC1OS5	AC1OS4	AC1OS3	AC1OS2	AC1OS1	AC1OS0	10011111
CKCON2	时钟控制 2	40H	XTGS1	XTGS0	XTALE	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0	01010000
CKCON3	时钟控制 3	41H	WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	MCDS1	MCDS0	00000010
CKCON4	时钟控制 4	42H	RCSS2	RCSS1	RCSS0	RPSC2	RPSC1	RPSC0	RTCCS3	RTCCS2	00000000
PCON2	电源控制 2	44H	AWBOD1	0	BO1S1	BO1S0	BO1RE	EBOD1	BO0RE	1	0000x1x1
PCON3	电源控制 3	45H	IVREN	IVRPDE	0	SPWRE	0	0	0	0	00000000
SPCON0	SFR 页面控制 0	48H	0	P6CTL	P4CTL	WRCTL	0	CKCTL0	PWCTL1	PWCTL0	00000000
DCON0	设备控制 0	4CH	HSE	IAPO	HSE1	0	0	IORCTL	RSTIO	OCDE	10000011
SPHB	SP 高边界	53H	1	1	1	1	SPHB.3	SPHB.2	SPHB.1	SPHB.0	11111111
RTCCR	RTC 控制寄存器	54H	RTCE	RTCO	RTCRL5	RTCRL4	RTCRL3	RTCRL2	RTCRL1	RTCRL0	00111111
RTCTM	RTC 定时器寄存器	55H	RTCCS1	RTCCS0	RTCCT5	RTCCT4	RTCCT3	RTCCT2	RTCCT1	RTCCT0	01111111
逻辑字节											
PCON0	电源控制 0	87H	SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL	00010000
PCON1	电源控制 1	97H	SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF	00000000
CKCON0	时钟控制 0	C7H	AFS	ENCKM	CKMIS1	CKMIS0	CCKS	SCKS2	SCKS1	SCKS0	00010000
WDTCR	看门口控制寄存器	E1H	WREN	NSW	ENW	CLRW	WIDL	PS2	PS1	PS0	00000000
P4	端口 4	E8H	P4.7	1	P4.5	P4.4	1	1	P4.1	P4.0	11111111
P6	端口 6	F8H	1	1	1	1	1	1	P6.1	P6.0	11111111

P 页 SFR 写的示例代码：

```

ISPCR = ISPEN;           //使能 IAP/ISP
IFMT = MS2;              // P 页写, IFMT =0x04
IFADRH = 0x00;
IFADRL = SPCON0;         //设置 P 页 SFR 地址
IFD |= CKCTL0;           //设置 CKCTL0
SCMD = 0x46;             //
SCMD = 0xB9;             //
IFMT = Flash_Standby;    // IAP/ISP 等待, IFMT =0x00
ISPCR &= ~ISPEN;

```

4. 引脚结构

4.1. 封装指南

图 4-1. MG82F6P32 LQFP32 顶视图

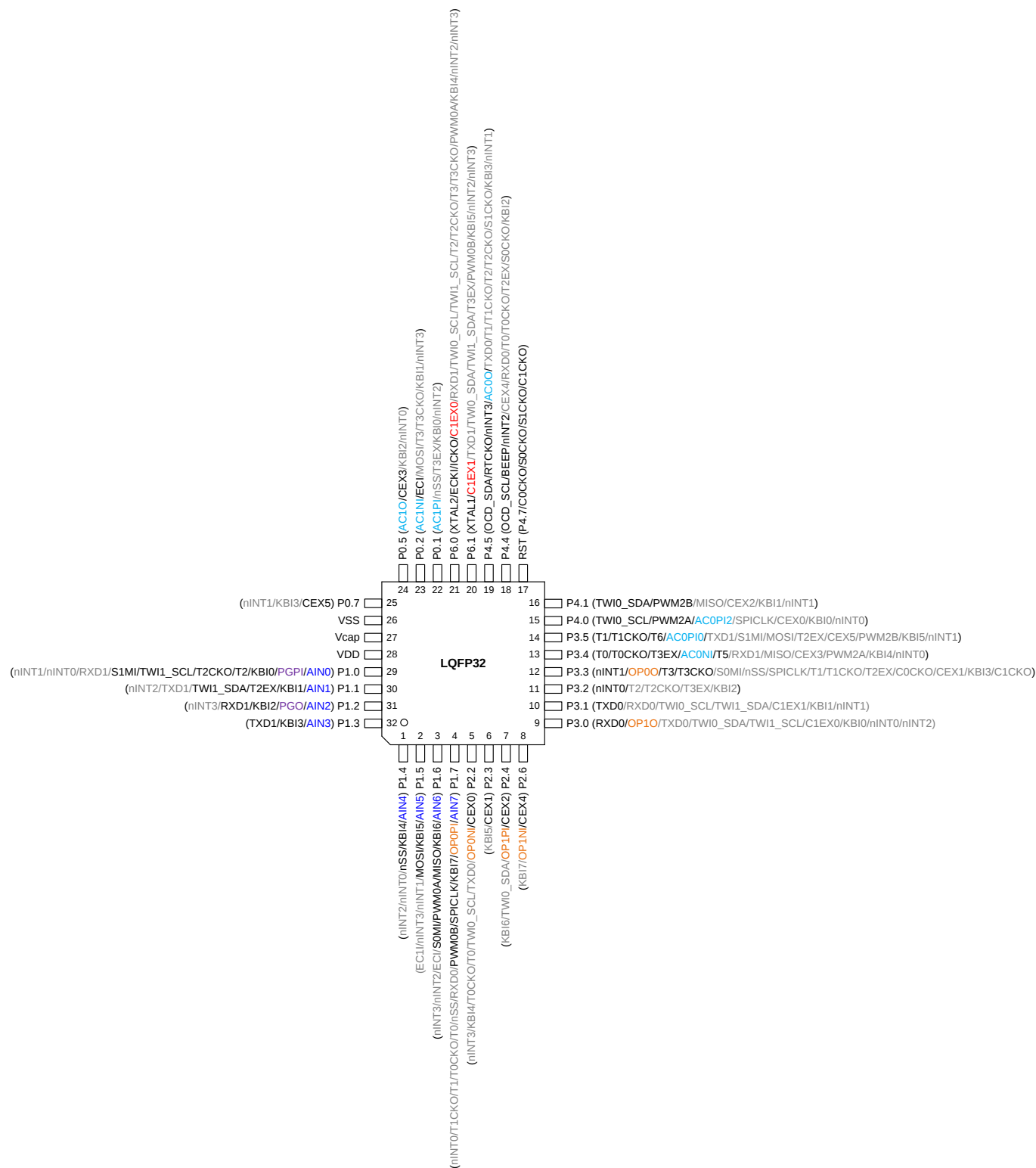


图 4-2. MG82F6P32 QFN32 顶视图

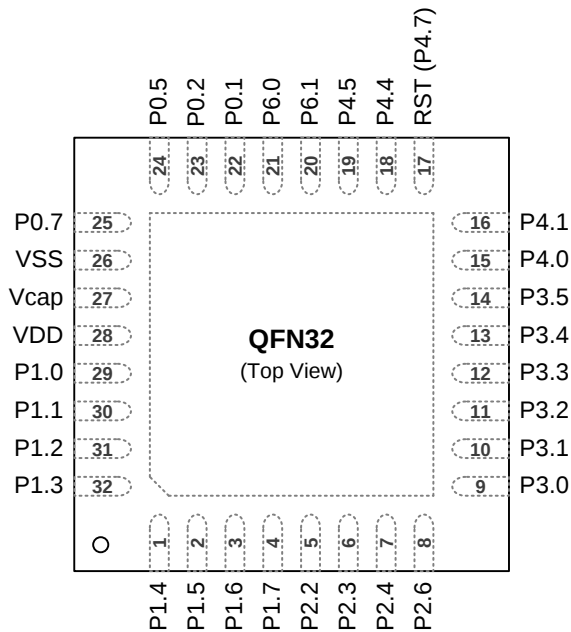
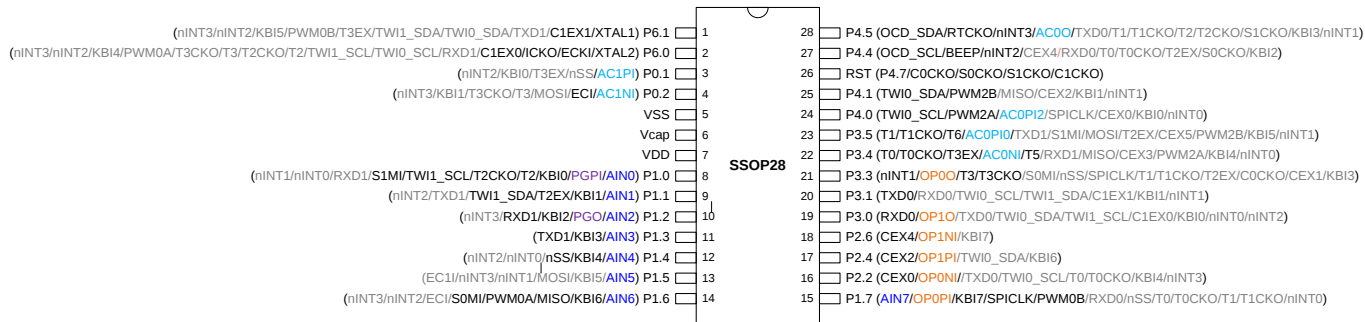


图 4-3. MG82F6P32 SSOP28 顶视图



4.2. 引脚描述

表 4-1. 引脚描述

助记符	引脚号		I/O 类型	描述
	32-Pin LQFP/ QFN	28-Pin SSOP		
P0.1 (AC1PI)	22	3	I/O	*端口 0.1. * AC1PI: 模拟比较器 1 正端输入
P0.2 (AC1NI) (ECI)	23	4	I/O	*端口 0.2. * AC1NI: 模拟比较器 1 负端输入 * ECI: PTM 外部时钟输入
P0.5 (AC1O) (CEX3)	24	--	I/O	*端口 0.5. * AC1OUT: 模拟比较器 1 输出 * CEX3: PTM 模块 3 外部 I/O
P0.7 (CEX5)	25	--	I/O	*端口 0.7. * CEX5: PTM 模块 5 外部 I/O
P1.0 (AIN0) (PGPI) (KBI0) (T2) (T2CKO) (TWI1_SCL) (S1MI)	29	8	I/O	*端口 1.0. * AIN0: ADC 通道 0 模拟输入 * PGPI: PGA 正端输入 * KBI0: 键盘输入 0. * T2: 定时/计数器 2 外部时钟输入. * T2CKO: 定时器 2 可编程时钟输出. * TWI1_SCL: TWI1/I2C1 串行时钟. * S1MI: 串口 1 SPI 主机模式数据输入
P1.1 (AIN1) (KBI1) (T2EX) (TWI1_SDA)	30	9	I/O	*端口 1.1. * AIN1: ADC 通道 1 模拟输入. * KBI1: 键盘输入 1. * T2EX: 定时/计数器 2 外部控制输入. * TWI1_SDA: TWI1/I2C1 串行数据.
P1.2 (AIN2) (PGO) (KBI2) (RXD1)	31	10	I/O	*端口 1.2. * AIN0: ADC 通道 2 模拟输入 * PGO: PGA 输出 * KBI2: 键盘输入 2. * RXD1: 串口 1 (UART1) 串行输入端口
P1.3 (AIN3) (KBI3) (TXD1)	32	11	I/O	*端口 1.3. * AIN3: ADC 通道 3 模拟输入 * KBI3: 键盘输入 3. * TXD1: 串口 1 (UART1) 串行输出端口
P1.4 (AIN4) (KBI4) (nSS)	1	12	I/O	*端口 1.4. * AIN4: ADC 通道 4 模拟输入 * KBI4: 键盘输入 4. * nSS: SPI 从机选择
P1.5 (AIN5) (KBI5) (MOSI)	2	13	I/O	*端口 1.5. * AIN5: ADC 通道 5 模拟输入. * KBI5: 键盘输入 5. * MOSI: SPI 主机输出&从机输入
P1.6 (AIN6) (KBI6) (MISO) (PWM0A) (S0MI)	3	14	I/O	*端口 1.6. * AIN6: ADC 通道 6 模拟输入 * KBI6: 键盘输入 6. * MISO: SPI 主机输入&从机输出. * PWM0A: PTM PWM0 输出副通道 A. * S0MI: 串口 0 SPI 主机模式数据输入.
P1.7 (AIN7) (OP0PI) (KBI7) (SPICLK) (PWM0B)	4	15	I/O	*端口 1.7. * AIN7: ADC 通道 7 模拟输入 * OP0PI: OPA0 正端输入 * KBI7: 键盘输入 7. * SPICLK: SPI 时钟, 用于主机输出, 从机输入 * PWM0B: PTM PWM0 输出副通道 B.
P2.2 (CEX0) (OP0NI)	5	16	I/O	*端口 2.2. * CEX0: PTM 模块 0 外部 I/O * OP0NI: OPA0 负端输入
P2.3 (CEX1)	6	--	I/O	*端口 2.3. * CEX1: PTM 模块 1 外部 I/O
P2.4 (CEX2) (OP1PI)	7	17	I/O	*端口 2.4. * CEX2: PTM 模块 2 外部 I/O * OP1PI: OPA1 正端输入

P2.6 (CEX4) (OP1NI)	8	18	I/O	*端口 2.6. * CEX4: PTM0 模块 4 外部 I/O * OP1NI: OPA1 负端输入
P3.0 (RXD0) (OP1O)	9	19	I/O	*端口 3.0. * RXD0: 串口 0 (UART0) 串行输入端口 * OP1O: OPA1 输出
P3.1 (TXD0)	10	20	I/O	*端口 3.1. * TXD0: 串口 0 (UART0) 串行输出端口
P3.2 (nINT0)	11	21	I/O	* 端口 3.2. * nINT0: 外部中断 0 输入
P3.3 (nINT1) (T3) (T3CKO) (OP0O)	12	21	I/O	*端口 3.3. * nINT1: 外部中断 1 输入 * T3: 定时/计数器 3 外部时钟输入. * T3CKO: 定时器 3 可编程时钟输出. * OP0O: OPA0 输出
P3.4 (T0) (T0CKO) (T3EX) (AC0NI) (T5)	13	22	I/O	*端口 3.4. * T0: 定时/计数器 0 外部输入. * T0CKO: 定时器 0 可编程时钟输出. * T3EX: 定时/计数器 3 外部控制输入 * AC0NI: 模拟比较器 0 负端输入. * T5: 定时/计数器 5 外部时钟输入.
P3.5 (T1) (T1CKO) (T6) (AC0PI0)	14	23	I/O	*端口 3.5. * T1: 定时/计数器 1 外部输入. * T1CKO: 定时器 1 可编程时钟输出. * T6: 定时/计数器 6 外部时钟输入. * AC0PI0: 模拟比较器 0 正端输入通道 0.
P4.0 (TWI0_SCL) (PWM2A) (AC0PI2)	15	24	I/O	*端口 4.0. * TWI0_SCL: TWI0/I2C0 串行时钟. * PWM2A: PTM0 PWM2 输出副通道 A. * AC0PI2: 模拟比较器 0 正端输入通道 2.
P4.1 (TWI0_SDA) (PWM2B)	16	25	I/O	*端口 4.1. * TWI0_SDA: TWI0/I2C0 串行数据. * PWM2B: PTM0 PWM2 输出副通道 B.
P4.4 (OCD_SCL) (BEEP) (nINT2)	18	27	I/O	*端口 4.4. * OCD_SCL: OCD 接口, 串行时钟 * BEEP: 蜂鸣器输出 * nINT2: 外部中断 2 输入
P4.5 (OCD_SDA) (RTCKO) (nINT3) (AC0O)	19	28	I/O	*端口 4.5. * OCD_SDA: OCD 接口, 串行数据 * RTCKO: RTC 可编程时钟输出. * nINT3: 外部中断 3 输入. * AC0O: 模拟比较器 0 输出.
P6.0 (XTAL2) (ECKI) (ICKO) (C1EX1)	21	2	I/O I O	*端口 6.0. * XTAL2: 片内振荡电路输出 * ECKI: 外部时钟输入模式下, 时钟输入引脚 * ICKO: 内部时钟 (MCK) 输出 * C1EX1: PCA1 模块 1 外部 I/O
P6.1 (XTAL1) (C1EX0)	20	1	I/O	*端口 6.1. * XTAL1: 片内振荡电路输入 * C1EX0: PCA1 模块 0 外部 I/O
nRST (P4.7) (C0CKO) (S0CKO) (S1CKO) (C1CKO)	17	26	I O	* nRST: 外部 RESET 输入, 低有效 * 端口 4.7 仅输出 <i>注意: 当 P4.7/nRST 用作端口引脚时, 不建议将其编程为输入, 以避免在启动期间高电平送入该引脚, 导致 MCU 锁定复位。</i> * C0CKO: PTM0 时基可编程时钟输出 * S0CKO: S0BRG 可编程时钟输出. * S1CKO: S1BRG 可编程时钟输出. * C1CKO: PCA1 时基可编程时钟输出
Vcap	27	6	I/O	* Vcap. 内部 LDO 应增加输出电容以稳定电压。连接 0.1uF 和 4.7uF 到 VSS。
VDD	28	7	P	电源输入
VSS	26	5	G	地, 0 V 参考

4.3. 功能复用

许多 I/O 引脚，除了普通的 I/O 口功能之外，还能复用其他内部功能。对于这些数码外设，默认状态是 GPIOs。然而，用户可以通过 AXRU0~AXUR3 相对应的控制位重新定义端口的功能。

AUXR3: 辅助寄存器 3

SFR 页 = 仅 0 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: T0PS1~0, 定时器 0 端口引脚选择位[1:0].

T0PS1~0	T0/T0CKO
0 0	P3.4
0 1	P4.4
1 0	P2.2
1 1	P1.7

Bit 5~4: BPOC1~0, 蜂鸣器输出控制位

BPOC[1:0]	P4.4 功能	I/O 模式
0 0	P4.4	By P4M0.4 & P4M1.4
0 1	ILRCO/32	By P4M0.4 & P4M1.4
1 0	ILRCO/16	By P4M0.4 & P4M1.4
1 1	ILRCO/8	By P4M0.4 & P4M1.4

为了使用 P4.4 的蜂鸣器功能，建议配置 P4.4 为推挽输出模式。

Bit 3: S0PS0, 串口 0 端口引脚选择位 0. (S0PS1 在 AUXR10.3)

S0PS1~0	RXD0	TXD0
0 0	P3.0	P3.1
0 1	P4.4	P4.5
1 0	P3.1	P3.0
1 1	P1.7	P2.2

Bit 2~1: TWIPS1~0, TWI0/I2C0 端口引脚选择位[1:0].

TWIPS1~0	TWI0_SCL	TWI0_SDA
0 0	P4.0	P4.1
0 1	P6.0	P6.1
1 0	P3.1	P3.0
1 1	P2.2	P2.4

AUXR4: 辅助寄存器 4

SFR 页 = 仅 1 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	0	AC0OE	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: T2PS1~0, 定时器 2 端口引脚选择位[1:0].

T2PS1~0	T2/T2CKO	T2EX
0 0	P1.0	P1.1
0 1	P3.2	P3.3
1 0	P6.0	P3.5
1 1	P4.5	P4.4

Bit 5~4: T1PS1~0, 定时器 1 端口引脚选择位 [1:0].

T1PS1~0	T1/T1CKO
0 0	P3.5
0 1	P4.5
1 0	P1.7
1 1	P3.3

Bit 3: AC1OE, AC1OUT 输出使能位

0: 禁止 AC1OUT 在引脚上输出

1: 使能 AC1OUT 输出到 P0.5.

Bit 1: AC0OE, AC0OUT 输出使能位

0: 禁止 AC0OUT 在引脚上输出

1: 使能 AC0OUT 输出到 P4.5.

AUXR5: 辅助寄存器 5

SFR 页 = 仅 2 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0PPS1	C0PPS0	EC1IPS0	C1COPS	ECIPS0	C0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: C0PPS1, {PWM2A, PWM2B}端口引脚选择位 1.

C0PPS1	PWM2A	PWM2B
0	P4.0	P4.1
1	P3.4	P3.5

Bit 4: C0PPS0, {PWM0A, PWM0B}端口引脚选择位 0.

C0PPS0	PWM0A	PWM0B
0	P1.6	P1.7
1	P6.0	P6.1

Bit 3: EC1IPS0, PCA1 ECI 端口引脚选择位 0.

EC1IPS0	EC1I
0	ECI
1	P1.5

Bit 2: C1COPS, PCA1 时钟输出(C1CKO) 端口引脚选择位

C1COPS	C1CKO
0	P4.7
1	P3.3

Bit 1: ECIPS0, PTM0 ECI 端口引脚选择位 0.

ECIPS0	ECI
0	P0.2
1	P1.6

Bit 0: C0COPS, PTM0 时钟输出 (C0CKO) 端口引脚选择位

C0COPS	C0CKO
0	P4.7
1	P3.3

AUXR6: 辅助寄存器 6

SFR 页 = 仅 3 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
KBIHPS1	KBIHPS0	KBILPS1	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: KBIHPS1~0, KBI4~7 端口引脚选择位[1:0].

KBIHPS1~0	KBI7	KBI6	KBI5	KBI4
0 0	P1.7	P1.6	P1.5	P1.4
0 1	P1.7	P1.6	P3.5	P3.4
1 0	P2.6	P2.4	P2.3	P2.2
1 1	P1.7	P1.6	P6.1	P6.0

Bit 5~4: KBILPS1~0, KBI0~3 端口引脚选择位 0.

KBILPS1~0	KBI3	KBI2	KBI1	KBI0
0 0	P1.3	P1.2	P1.1	P1.0
0 1	P3.3	P3.2	P3.1	P3.0
1 0	P0.7	P0.5	P0.2	P0.1
1 1	P4.5	P4.4	P4.1	P4.0

Bit 3: T3FCS, 保留用于芯片测试

Bit 2: T2FCS, 保留用于芯片测试

Bit 1: SnMIPS, S0MI, S1MI, S2MI & S3MI 端口引脚选择位

SnMIPS	S0MI	S1MI
0	P1.6	P1.0
1	P3.3	P3.5

Bit 0: S0COPS, S0BRG Clock Output (S0CKO) 端口引脚选择位

S0COPS	S0CKO
0	P4.7
1	P4.4

AUXR7: 辅助寄存器 7

SFR 页 = 仅 4 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0CKOE	C1CKOE	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 保留。写该寄存器时，此位必须写“0”。

Bit 5: C0CKOE, PTM 时钟输出(C0CKO) 使能

0: 禁止 PTM 时钟输出

1: 使能 PTM 时基 1/2 溢出率时钟输出

Bit 4: C1CKOE, PCA1 时钟输出(C1CKO) 使能

0: 禁止 PCA1 时钟输出

1: 使能 PCA1 时基 1/2 溢出率时钟输出

Bit 3~0: 保留。写该寄存器时，此位必须写“0”。

AUXR8: 辅助寄存器 8

SFR 页 = 仅 5 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留。写该寄存器时，此位必须写“0”。

Bit 6: C1ICS1, PCA1 输入通道 1 输入端口引脚选择位

C1ICS1	C1EX1 输入
0	C1EX1 端口引脚
1	T2EXI

Bit 5: C1ICS0, PCA1 输入通道 0 输入端口引脚选择位

C1ICS0	C1EX0 input
0	C1EX0 Port Pin
1	T3EXI

Bit 4~3: 保留。写该寄存器时，此位必须写“0”。

Bit 2: S1COPS, S1BRG 时钟输出 (S1CKO) 端口引脚选择

S1COPS	S1CKO
0	P4.7
1	P4.5

Bit 1~0: T3PS1~0, Timer 3 端口引脚选择位[1:0].

T3PS1~0	T3/T3CKO	T3EX
0 0	P3.3	P3.4
0 1	P3.3	P3.2
1 0	P0.2	P0.1
1 1	P6.0	P6.1

AUXR9: 辅助寄存器 9

SFR 页 = 仅 6 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G1	T0G1	0	0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: SID/STWI 端口引脚选择位[1:0].

SIDPS1~0	STWI_SCL	STWI_SDA
0 0	nINT1	S0MI
0 1	TWI0_SCL	TWI0_SDA
1 0	TWI1_SCL	TWI1_SDA
1 1	T2EXI	T3EXI

Bit 1~0: S1PS1~0, 串口 1 引脚选择 [1:0].

S1PS1~0	RXD1	TXD1
0 0	P1.2	P1.3
0 1	P6.0	P6.1
1 0	P1.0	P1.1
1 1	P3.4	P3.5

AUXR10: 辅助寄存器 10

SFR 页 = 仅 7 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: SPIPS1~0, SPI 端口引脚选择位 1~0.

引脚选项	SPIPS1~0	nSS	MOSI	MISO	SPICLK
0	0 0	P1.4	P1.5	P1.6	P1.7
1	0 1	P0.1	P0.2	P4.1	P4.0
2	1 0	P3.3	P1.5	P1.6	P1.7
3	1 1	P1.7	P3.5	P3.4	P3.3

Bit 3: S0PS1, 串口 0 引脚选择位 1. (此功能在 AUXR3.3, S0PS0 中已描述)

Bit 2~1: P6.0 功能配置控制位 1 和位 0, 这两位仅仅当内部 RC 振荡(IHRCO 或 ILRCO)被选择为系统时钟源时有效。在晶振模式下, P6.0 和 P6.1 用于 XTAL2 和 XTAL1。外部时钟输入模式下, P6.0 专用于时钟输入。在内部振荡模式, P6.0 为普通 I/O 或时钟源发生器提供下列选项, 当 P60OC[1:0] 索引为非 P6.0 GPIO 功能时, P6.0 将驱动内部 RC 振荡器输出为其它设备提供时钟源

P60OC[1:0]	P6.0 功能	I/O 模式
0 0	P6.0	By P6M1.0 & P6M0.0
0 1	MCK	By P6M1.0 & P6M0.0
1 0	MCK/2	By P6M1.0 & P6M0.0
1 1	MCK/4	By P6M1.0 & P6M0.0

了解详情, 请参考“9 系统时钟”。6.0 作为时钟输出功能时, 建议设置{P6M1.0, P6M0.0}为“01”来选择 P6.0 为推挽输出模式。

Bit 0: P60FD, P6.0 快速驱动。

0: P6.0 默认驱动输出。

1: P6.0 快速驱动输出使能。若 P6.0 被配置为时钟输出, 当 P6.0 输出频率大于 12MHz(5V)或者大于 6MHz(3V) 时使能此位。

AUXR11: 辅助寄存器 11

SFR 页 = 仅 8 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	I2C1PS1	I2C1PS0	RX1S0	0	T1CKOE	T0CKOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: I2C1PS1~0, TWI1/I2C1 端口引脚选择[1:0].

I2C1PS1~0	TWI1_SCL	TWI1_SDA
0 0	P1.0	P1.1
0 1	P6.0	P6.1
1 0	P3.0	P3.1
1 1	TWI0_SCL	TWI0_SDA

Bit 3: RX1S0, RXD1 选择位 0.

0: RXD1 由 S1PS1~0 (AUXR9.1~0)选择.

1: RXD1 强制选择在 AC0OUT.

AUXR12: 辅助寄存器 12

SFR 页 = 仅 9 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CRCDS2	CRCM0	PDOES1	PDOES0	EDCM0	GPLC0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: PDOES1~0, PDTXOE1 端口引脚选择位[1:0].

PDOES1~0	PDTXOE
0 0	P1.5
0 1	P3.3
1 0	P6.1
1 1	P3.2

AUXR16: 辅助寄存器 16

SFR 页 = 仅 D 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~0: C0OPS5 ~0, PTM0 输出引脚选择

	C0OPS5		C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
C0OPSn	CEX5		CEX4	CEX3	CEX2	CEX1	CEX0
0	P0.7		P2.6	P0.5	P2.4	P2.3	P2.2
1	P3.5		P3.2	P3.4	P4.1	P3.3	P4.0

AUXR17: 辅助寄存器 17

SFR 页 = 仅 E 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	0	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1~0: C1OPS1 ~0, PCA1 输出引脚选择

	C1OPS1	C1OPS0
C1OPSn	C1EX1	C1EX0
0	P6.1	P6.0
1	P3.1	P3.0

XICFG: 外部中断配置寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xC1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
INT1IS.1	INT1IS.0	INT0IS.1	INT0IS.0	X3FLT	X2FLT	X1FLT	X0FLT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: INT1IS.1~0, 与 INT1IS.2 一起决定 nINT1 输入引脚选择位如下表定义.

INT1IS.2~0	选择 nINT1 的端口引脚
0 0 0	P3.3
0 0 1	P3.1
0 1 0	P3.5
0 1 1	P4.1
1 0 0	P4.5
1 0 1	P1.5
1 1 0	P1.0
1 1 1	P0.7

Bit 5~4: INT0IS.1~0, 与 INT0IS.2 一起决定 nINT0 输入引脚选择位如下表定义

INT0IS.2~0	选择 nINT0 的端口引脚
0 0 0	P3.2
0 0 1	P3.0
0 1 0	P3.4
0 1 1	P4.0
1 0 0	P1.0
1 0 1	P1.4
1 1 0	P1.7
1 1 1	P0.5

XICFG1: 外部中断配置寄存器 1

SFR 页 = 仅 1 页

SFR 地址 = 0xC1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
INT1IS.2	INT0IS.2	INT2IS.1	INT2IS.0	X3FLT1	X2FLT1	X1FLT1	X0FLT1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: INT1IS2, 与 INT1IS.1~0 一起决定 nINT1 输入端口引脚选择位。

Bit 6: INT0IS2, 与 INT0IS.1~0 一起决定 nINT0 输入端口引脚选择位。

Bit 5~4: INT2IS.1~0, nINT2 输入引脚选择位如下表定义。

INT2IS.2~0	选择 nINT2 的端口引脚
0 0 0	P4.4
0 0 1	P3.0
0 1 0	P1.1
0 1 1	P1.6
1 0 0	P1.4
1 0 1	P6.0
1 1 0	P0.1
1 1 1	P6.1

XICFG2: 外部中断配置寄存器 2

SFR 页 = 仅 2 页

SFR 地址 = 0xC1

复位值 = XXXX-0000

7	6	5	4	3	2	1	0
--	--	--	--	INT3IS.2	INT3IS.1	INT3IS.0	INT2IS.2
W	W	W	W	R/W	R/W	R/W	R/W

Bit 3~1: INT3IS.2~0, nINT3 输入引脚选择位如下表定义

INT3IS.2~0	选择 nINT3 的端口引脚
0 0 0	P4.5
0 0 1	P2.2
0 1 0	P1.5
0 1 1	P1.6
1 0 0	P0.2
1 0 1	P6.1
1 1 0	P1.2
1 1 1	P6.0

Bit 0: INT2IS2, 与 INT2IS.1~0 一起决定 nINT2 输入端口引脚选择位

5. 8051 CPU 功能描述

5.1. CPU 寄存器

PSW: 程序状态字

SFR 页 = 0~F

SFR 地址 = 0xD0

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CY: 进位标志

AC: 辅助进位标志

F0: 用户可设定的标志位 0

RS1: 寄存器组选择位 1

RS0: 寄存器组选择位 0

OV: 溢出标志

F1: 用户可设定的标志位 1

P: 奇偶标志

程序状态字(PSW)包含反映 CPU 当前状态的几个状态位。PSW 属于特殊功能寄存器 SFR 区，包含进位标志，辅助进位标志(应用于 BCD 操作)，两个寄存器组选择位，溢出标志，奇偶标志和两个用户可设定的标志位。

进位标志，不仅有算术运算的进位功能，也充当许多布尔运算的“累加器”。

RS0 和 RS1 被用来选择 4 组中的任意一组寄存器组，详见章节“6.2 片内数据 RAM”。一些指令参考这些内存(RAM)的位置比如从 R0 到 R7。

奇偶位反映累加器内 1 的个数的状况，累加器中 1 的个数是奇数则 P=1，否则 P=0。

SP: 堆栈指针

SFR 页 = 0~F

SFR 地址 = 0x81

复位值 = 0000-0111

7	6	5	4	3	2	1	0
SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

堆栈指针保持栈顶位置，每执行一个 PUSH 指令，会自动增加，复位后默认值为 0x07。

DPL: 数据指针低字节

SFR 页 = 0~F

SFR 地址 = 0x82

复位值 = 0000-0000

7	6	5	4	3	2	1	0
DPL.7	DPL.6	DPL.5	DPL.4	DPL.3	DPL.2	DPL.1	DPL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DPL 是 16 位 DPTR 的低字节，DPTR 用来间接访问 XRAM 和程序空间。

DPH: 数据指针高字节

SFR 页 = 0~F

SFR 地址 = 0x83 复位值 = 0000-0000

7	6	5	4	3	2	1	0
DPH.7	DPH.6	DPH.5	DPH.4	DPH.3	DPH.2	DPH.1	DPH.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DPH 是 16 位 DPTR 的高字节，DPTR 用来间接访问 XRAM 和程序空间。

ACC: 累加器

SFR 页 = 0~F

SFR 地址 = 0xE0 复位值 = 0000-0000

7	6	5	4	3	2	1	0
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

算术运算的累加器

B: B Register

SFR 页 = 0~F

SFR 地址 = 0xF0 复位值 = 0000-0000

7	6	5	4	3	2	1	0
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

另一个算术运算的累加器

5.2. CPU 时序

MG82F6P32 是基于 80C51 的高效 1-T 结构的单芯片微处理器，与 8051 指令集兼容，每条指令需要 1~7 个时钟信号(比标准 8051 快 6~7 倍)。使用流线型结构同标准的 8051 结构比较大大增加了指令完成的速度，指令的时序比标准的 8051 更快。

多数 8051 执行指令，一个区别是建立在机器周期和时钟周期之间，机器周期来自 2 到 12 个时钟周期长度。然而，1-T 结构的 80C51 执行指令是基于单独的时钟周期时序。所有指令时序被指定在时钟周期期间。关于 1T-80C51 指令更详细的说明，请参考“38 指令集”，这里有每一条指令的助记符、字节数、时钟周期数。

5.3. CPU 寻址模式

直接寻址(DIR)

直接寻址时操作数用指令中一个8位地址的区域表示，只有内部数据存储器 and 特殊功能寄存器可以直接寻址。

间接寻址(IND)

间接寻址时指令用一个包含操作数地址的寄存器表示，内部和外部存储器均可间接寻址。

8 位地址的地址寄存器可以是选中区的 R0 或 R1 或堆栈指针，16 位地址的地址寄存器只能是 16 位的“数据指针”寄存器-DPTR。

寄存器指令寻址(REG)

包含从 R0 到 R7 的寄存器区可以被某些指令存取，这些指令的操作码中用 3 位寄存器说明。存取寄存器的指令有更高的代码效率，因为这种模式减少了一个地址字节。当指令被执行时，其中被选取的区一个 8 位寄存器被存取。执行时，用 PSW 寄存器中两位区选择位来选择四分之一区。

特殊寄存器指令寻址

一些指令具有一个特定的寄存器，例如，一些指令常用于累加器，或数据指针等等，所以没有需要指向它的地址字节。操作码本身就行了。

立即寻址(IMM)

常量的数值可以在程序存储器中跟随操作码。

索引寻址

索引寻址只能访问程序存储器，且只读。这种寻址模式用查表法读取程序存储器。一个 16 位基址寄存器(数据指针 DPTR 或程序计数器 PC)指向表的基地址，累加器提供偏移量。程序存储器中表项目地址由基地址加上累加器数据后形成。另一种索引寻址方式是利用“case jump”指令。跳转指令中的目标地址是基地址加上累加器数据后的值。

6. 存储器组织

像所有的 80C51 一样，MG82F6P32 2 的程序存储器和数据存储器的地址空间是分开的，这样 8 位微处理器可以通过一个 8 位的地址快速而有效的访问数据存储器。

程序存储器(ROM)只能读取，不能写入。最大可以达到 32K 字节。在 MG82F6P32 中，所有的程序存储器都是片上 Flash 存储器。因为没有设计外部程序使能(/EA)和编程使能(/PSEN)信号，所以不允许外接程序存储器。

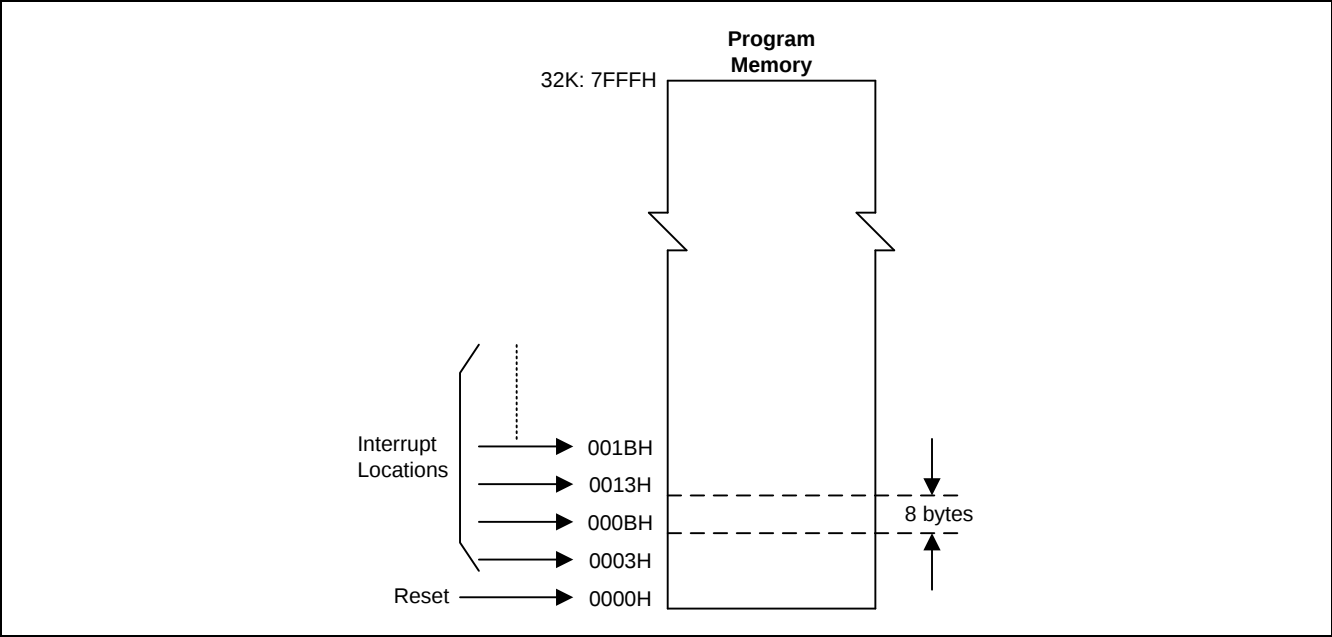
数据存储器使用与程序存储器不同的地址空间。MG82F6P32 有 256 字节的内部和 1792 字节的片上扩展存储器(XRAM)。

6.1. 片内程序闪存

程序存储器用来保存让 CPU 进行处理的程序代码，如图 6-1.所示。复位后，CPU 从地址为 0000H 的地方开始运行，用户应用代码的起始部分应该放在这里。为了响应中断，中断服务位置(被称为中断矢量)应该位于程序存储器。每个中断在程序存储器中有一个固定的起始地址，中断使 CPU 跳到这个地址运行中断服务程序。举例来说，外部中断 0 被指定到地址 0003H，如果使用外部中断 0，那么它的中断服务程序一定是从 0003H 开始的。如果中断未被使用，那么这些地址就可以被一般的程序使用。

中断服务程序的起始地址之间有 8 字节的地址间隔：外部中断 0，0003H；定时器 0，000BH；外部中断 1，0013H；定时器 1，001BH 等等。如果中断服务程序足够短，它完全可以放在这 8 字节的空间中。如果其他的中断也被使用的话，较长的中断服务程序可以通过一条跳转指令越过后面的中断服务起始地址。

图 6-1. 程序存储器



6.2. 片内数据 RAM

图 6-2 向 MG82F6P32 使用者展示了内部和外部数据存储器的空间划分。内部数据存储器被划分为三部分，通常被称为低 128 字节 RAM，高 128 字节 RAM 和 128 字节 SFR 空间。内部数据存储器的地址线只有 8 位宽，因此地址空间只有 256 字节。SFR 空间的地址高于 7FH，用直接地址访问；而用间接访问的方法访问高 128 字节的 RAM。这样虽然 SFR 和高 128 字节 RAM 占用相同的地址空间(80H—FFH)，但他们实际上是分开的。

如图 6-3 所示，低 128 字节 RAM 与所有 80C51 一样。最低的 32 字节被划分为 4 组每组 8 字节的寄存器组。指令中称这些寄存器为 R0 到 R7。程序状态字(PSW)中的两位用于选择哪组寄存器被使用。这使得程序空间能够被更有效的使用，因为对寄存器访问的指令比使用直接地址的指令短。接下来的 16 字节是可以位寻址的存储器空间。80C51 的指令集包含一个位操作指令集，这区域中的 128 位可以被这些指令直接使用。位地址从 00H 开始到 7FH 结束。

所有的低 128 字节 RAM 都可以用直接或间接地址访问，而高 128 字节 RAM 只能用间接地址访问。

图6-4给出了特殊功能寄存器(SFR)的概览。SFR包括端口寄存器，定时器和外围器件控制器，这些寄存器只能用直接地址访问。SFR 空间中有16个地址同时支持位寻址和直接寻址。可以位寻址的 SFR的地址末位是0H或8H。

图 6-2. 数据存储器

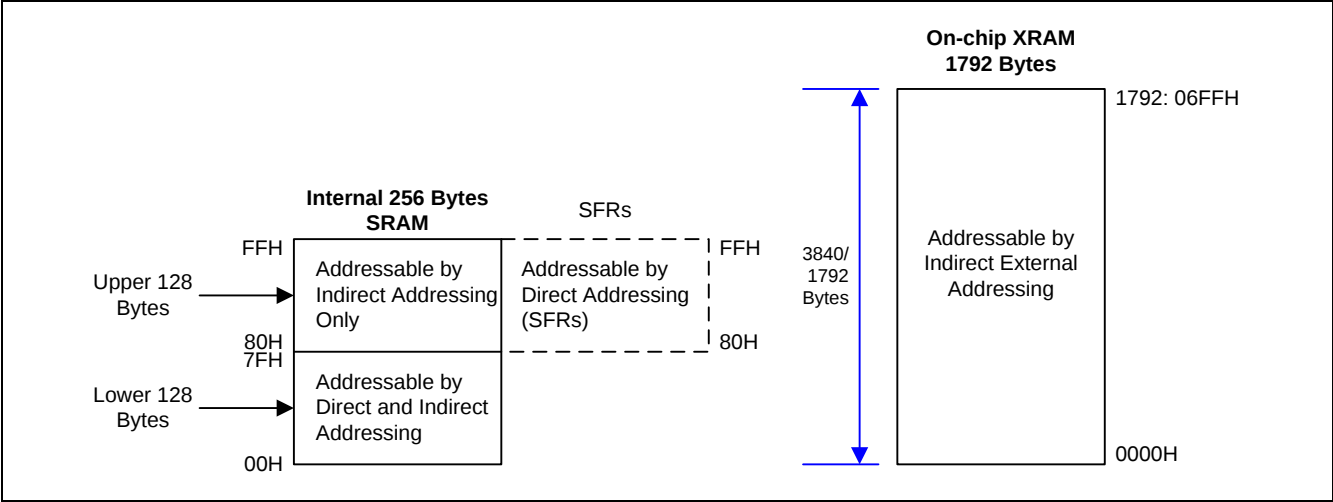


图 6-3. 内部 RAM 的低 128 字节

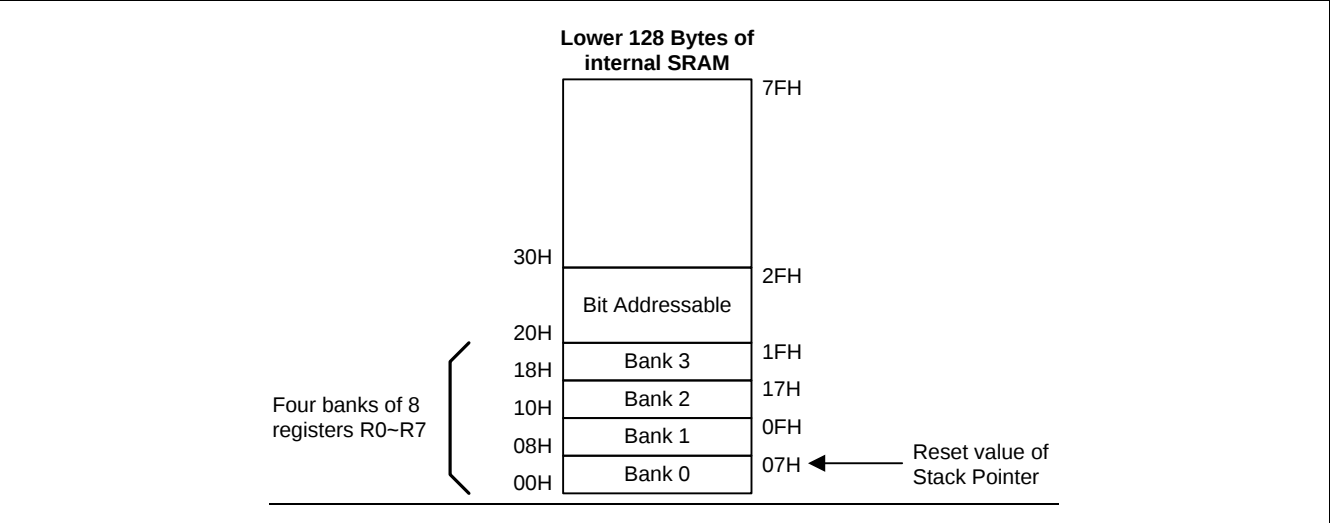
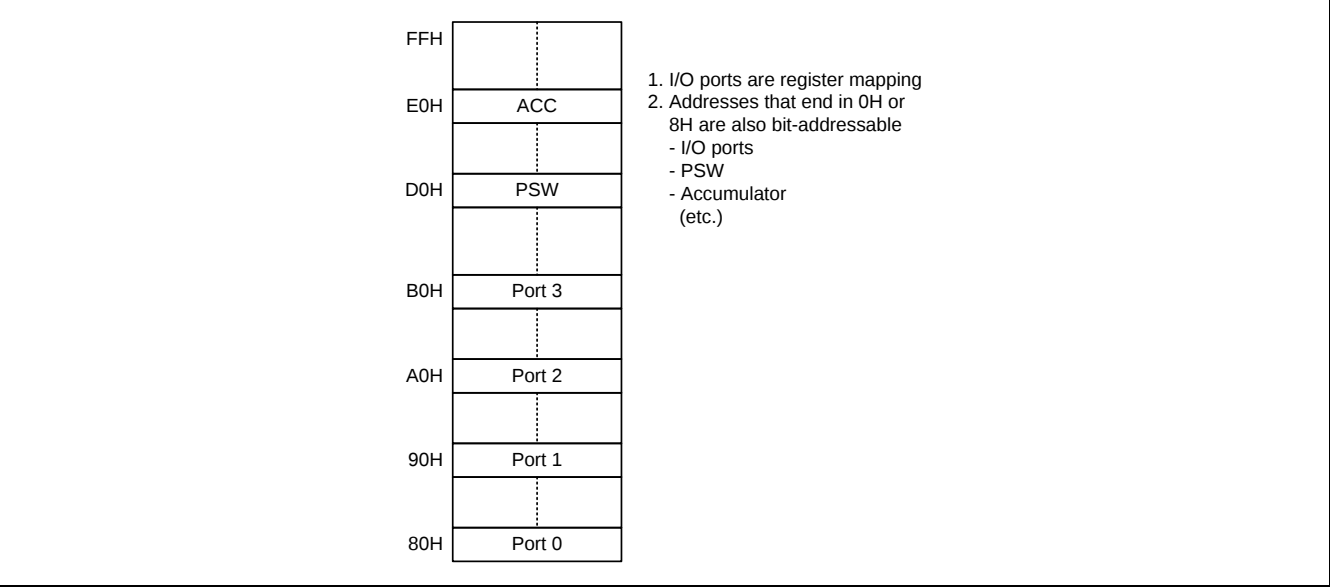


图 6-4. SFR 空间



6.3. 片内扩展 RAM (XRAM)

访问片上扩展 RAM(XRAM)，参考图 6-2，这 92 字节的 XRAM(0000H to 06FFH)可以被外部移动指令“MOVX @Ri”和“MOVX @DPTR”间接访问，在 C51 编译器中，使用“pdata”或“xdata”声明变量分配到 XRAM 中，编译后，被“pdata”或“xdata”声明过的变量将分别通过“MOVX @Ri”或“MOVX @DPTR”指令进行存取，这样 MG82F6P32 硬体才能正确访问 XRAM。

6.4. EMB, 片外数据存储器总线

MG82F6P32 不支持访问片外数据存储器

6.5. 关于 C51 编译器的声明标识符

C51 编译器的声明识别符与 MG82F6P32 存储空间的对应关系如下：

data

128 字节的内部数据存储空间(00h~7Fh)。使用除 MOVX 和 MOVC 以外的指令，可以直接或间接的访问。全部或部分的堆栈可能保存在此区域中。

idata

间接数据。256 字节的内部数据存储空间(00h~FFh)使用除 MOVX 和 MOVC 以外的指令间接访问。全部或部分的堆栈可能保存在此区域中。此区域包括 data 区和 data 区以上的 128 字节。

sfr

特殊功能寄存器。CPU 寄存器和外围部件控制/状态寄存器，只能通过直接地址访问。

xdata

外部数据或片上的扩展 RAM(XRAM)；通过“MOVX @DPTR”指令访问标准 80C51 的 64K 存储空间。MG82F6P32 有 1792 字节的片内 xdata 存储空间。

pdata

分页的外部数据(256 字节)或片上的扩展 RAM(XRAM)：重叠的 256 字节的存储器地址通过“MOVX @Ri”指令访问。MG82F6P32 有 256 字节片上 pdata 存储器它与片上 xdata 存储器共享。

code

32K 字节程序存储空间。通过“MOVC @A+DPTR”访问，作为程序部分被读取。MG82F6P32 有 32K 字节的片上程序存储器。

7. XRAM 访问

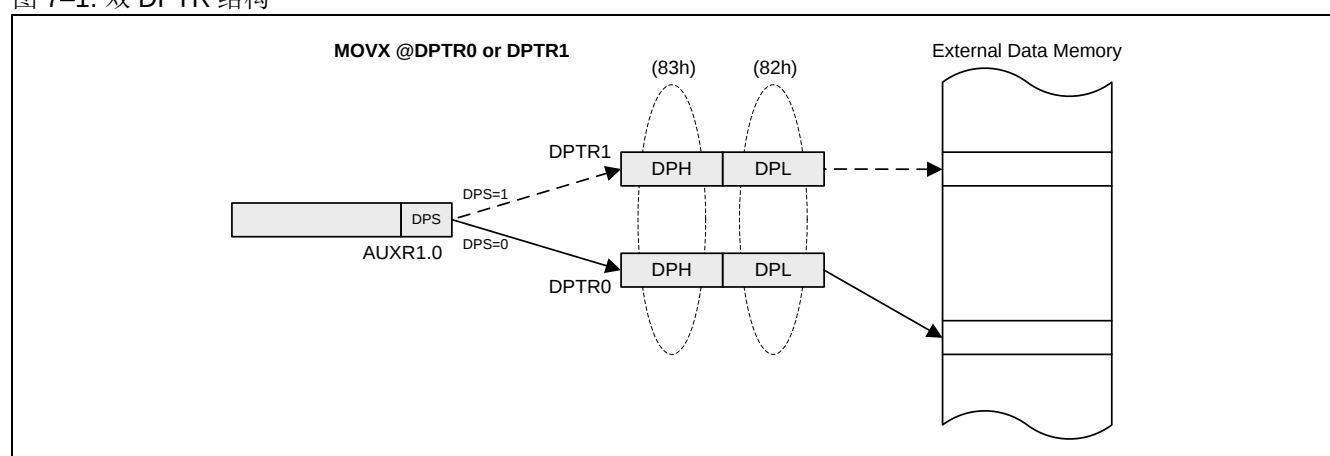
MG82F6P32 系列 MCU 内含有 **1792** 字节被映射到外部数据存储器的数据存储器(XRAM)。外部数据存储空间可以使用外部移动指令(MOVX)和数据指针(DPTR)访问，或使用(R0 或 R1)的 MOVX 间接访问模式。如果 MOVX 指令使用 8 位寻址操作(比如@R1)，16 位地址的高字节则由 XRAM 的页选择寄存器(XRPS)决定。

使用 MOVX 指令访问内部的 XRAM 存储空间。MOVX 指令有使用两种间接寻址方法。第一种方法是使用数据指针(DPTR)，一个包含外部数据存储器(XRAM)读写有效地址的 16 位寄存器。第二种方法是使用 R0 或 R1 结合 XRPS 寄存器来获取有效的外部数据存储器(XRAM)地址。

7.1. MOVX 在 16 位地址上使用双 DPTR

如图 7-1 所示的双 DPTR 结构是能让芯片指定外部数据存储器的定位地址的一种方法。有两个 16 位 DPTR 寄存器，和一个称之为 DPS(AUXR1.0)的控制位，允许在程序代码和外部存储器之间的切换。

图 7-1. 双 DPTR 结构



DPTR 指令

使用 DPS 位的六条指令参考 DPTR 的当前选择，如下：

INC DPTR	; 数据指针加 1
MOV DPTR,#data16	; DPTR 加载 16 位常量
MOV A,@A+DPTR	; 将代码字节移动到 ACC
MOVX A,@DPTR	; 移动 XRAM(16 位地址)到 ACC
MOVX @DPTR,A	; 移动 ACC 到 XRAM(16 位地址)
JMP @A+DPTR	; 直接跳转到 DPTR

AUXR1: 辅助寄存器 1

SFR 页 = 0~F

SFR 地址 = 0xA2 复位值 = 0000-0000

7	6	5	4	3	2	1	0
OP1Fr	OP0Fr	CRCDS1	CRCDS0	0	AC1Fr	AC0Fr	DPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 0: DPS, DPTR 选择位，用来在 DPTR0 和 DPTR1 之间切换。

0: 选择 DPTR0。

1: 选择 DPTR1。

DPS	Selected DPTR
0	DPTR0
1	DPTR1

DPL: 数据指针低 8 位

SFR 页 = 0~F

SFR 地址 = 0x82 复位值 = 0000-0000

7	6	5	4	3	2	1	0
DPL.7	DPL.6	DPL.5	DPL.4	DPL.3	DPL.2	DPL.1	DPL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DPL 寄存器是 16 位 DPTR 的低字节。DPTR 用于间接访问 XRAM 和闪存(Flash)存储器的编址。

DPH: 数据指针高 8 位

SFR 页 = 0~F

SFR 地址 = 0x83 复位值 = 0000-0000

7	6	5	4	3	2	1	0
DPH.7	DPH.6	DPH.5	DPH.4	DPH.3	DPH.2	DPH.1	DPH.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

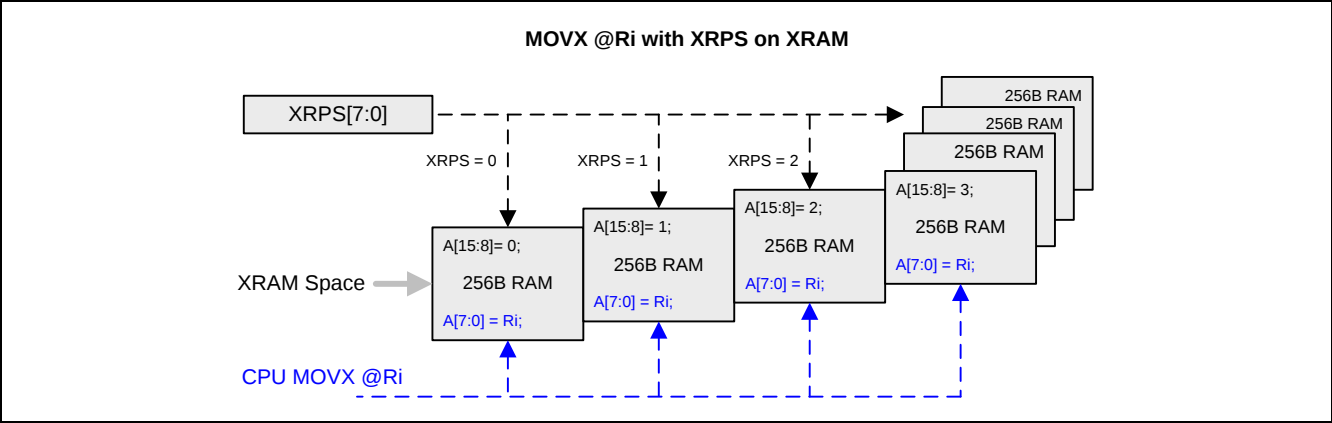
DPH 寄存器是 16 位 DPTR 的高字节。DPTR 用于间接访问 XRAM 和闪存(Flash)存储器的编址。

7.2. MOVX 在 8 位地址上使用 XRPS

MOVX 指令的 8 位地址是由 XRPS SFR 的值作为有效地址的高 8 位和 R0 或 R1 的值作为有效地址的低 8 位构成。

这个功能可以给设计者提供更高效率的代码来访问 XRAM。要访问整个范围的 XRAM，需要使用 2 字节的地址。编译器使用 DPTR 对 2 字节的地址进行编译，访问特定的内存位置，会增加更多的指令，降低效率。但是如果使用带有全局“pdata”变量搭配使用 XRPS，编译器会将其转换为 MOVX@Ri，以减少许多额外的指令，从而提高内存访问性能。

图 7-2. XRPS 结构



XRPS: XRAM 页选择寄存器

SFR 页 = 0~F

SFR 地址 = 0x8F

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	XRPS.2	XRPS.1	XRPS.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: 保留。写该寄存器时，此位必须写“0”。

Bit 3~0: XRPS, XRAM 的页选择。XRPS 寄存器是 16 位外部数据存储器地址的高字节，当用于 8 位 MOVX 指令时有效地选择 RAM 的 256 字节页。因此 XRPS 寄存器的高位(保留位)总应是零，XRPS 决定 XRAM 访问的是哪一页。在 MG82F6P32 中，XRPS 索引 7 页的 256 字节页 RAM。

例如：如果 XRPS = 0x01，则访问 XRAM 的 0x0100 到 0x01FF 地址。

8. 直接存储器访问控制器 (DMA)

直接存储器访问控制器(DMA)让数据从源头到目的传输皆不需要 CPU 介入，存取范围包含整个 XRAM 和 SFR。例如，DMA 能从 ADC12 转换结果移到 8051 XRAM。这样可以让 CPU 在这段期间并行做其它操作以提高效能。

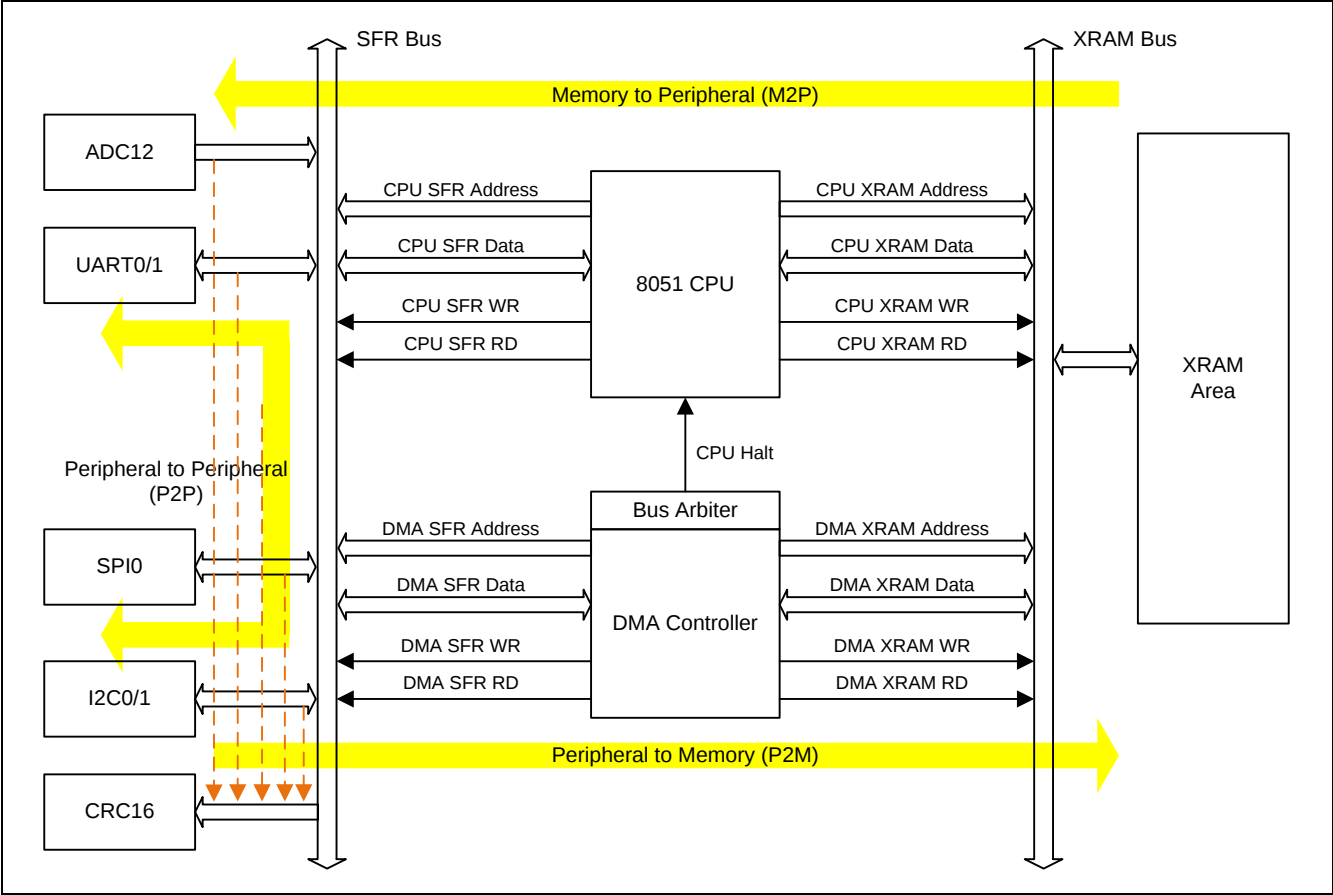
使用 DMA 能提升外设模块的效率。通过允许 CPU 在低功耗模式下不用被唤醒便可进行数据传输，以达到降低系统功耗的要求。

DMA 控制器特性如下：

- 容易使用的单通道 DMA
- 传输类型：存储器到外设(M2P)，外设到存储器(P2M)，外设到外设(P2P)
- 触发 DMA 传输的选项：CPU 软件或外部硬件
- 支持区块传输模式，传输大小可达 65536 字节
- 在 DMA 传输中可以拷贝数据到 CRC 引擎
- 在循环模式下自动初始化当前传输计数器
- DMA 传输传输过程中可以暂停传送及恢复传送
- 可以在低功耗模式运行(中断唤醒空闲模式)
- 传输结束的中断选择

DMA 传送方框图见 图 8-1.

图 8-1. DMA 访问方框图



8.1. DMA 结构

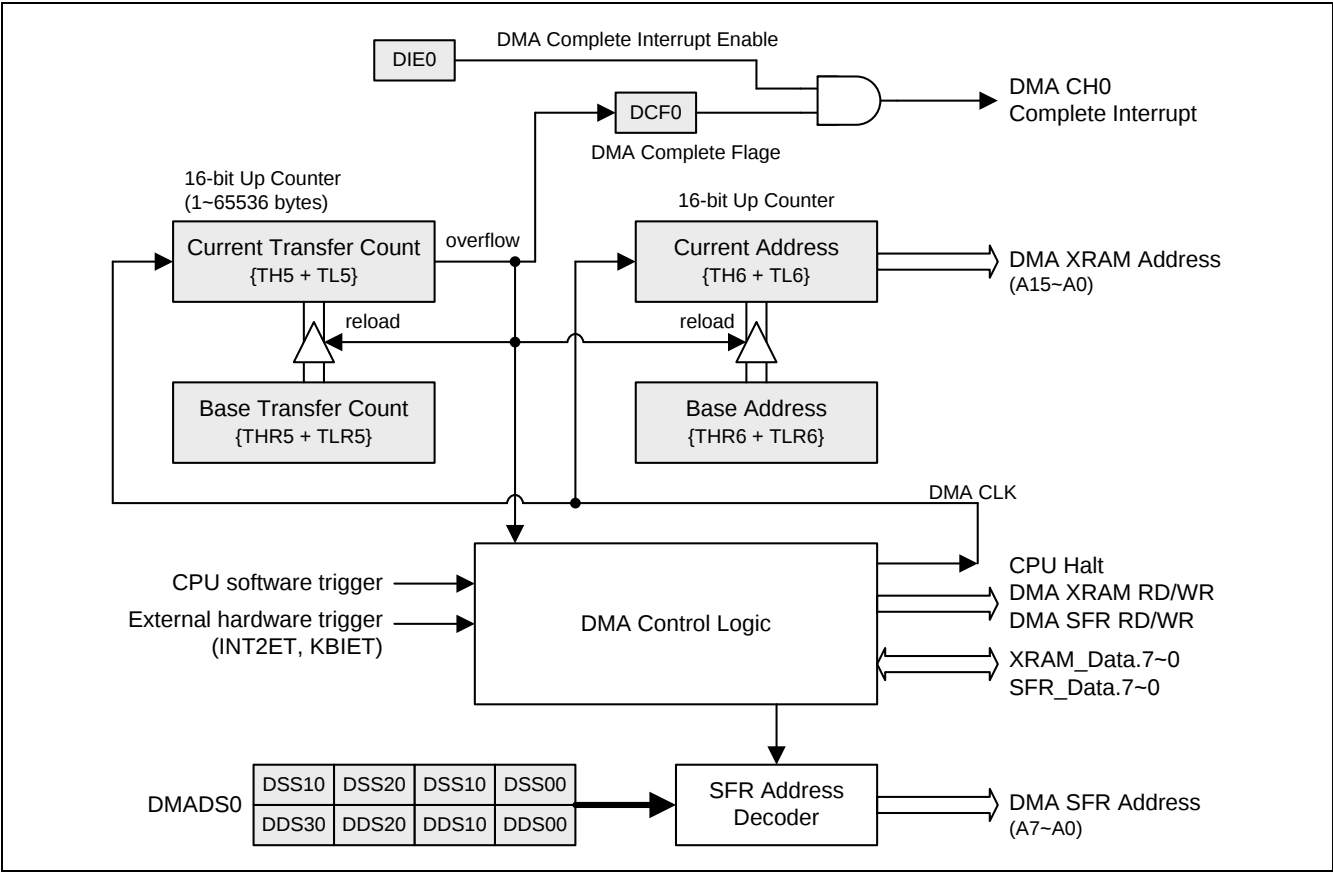
MG82F6P32 的 DMA 控制器提供一个通道 DMA 支持 3 种传输类型：传输数据从存储器(XRAM)到外设，从外设到存储器(XRAM)和从外设到外设。DMA 通道的 DMADS0 寄存器定义 DMA 传输类型配置 DMA 控制器和在外设访问时定义数据路径产生 SFR 地址。

定时器 5 和定时器 6 嵌入在 DMA 模块中。DMA 控制器支持由一个 DMA 触发，CPU 软件触发或外部硬件触发的块模式传输。传输大小从 1 字节到 65536 字节可编程，此功能由 DMA 传输计数器定时器 5 计数。如果 DMA 需要访问 XRAM，定时器 6 是 XRAM 的地址指针。当 DMA 结束一个数据交换，DMA_CLK 将触发定时器 5 加一 DMA 传输计数器和定时器 6 加一指向下一个 XRAM 地址。定时器 5 和定时器 6 仅支持向上计数。当 DMA 功能不用时，定时器 5 和定时器 6 是与通用定时器 0 一样的一个 16 位计数器。

DMACR0 和 DMACG0 是 DMA 操作模式控制的特殊功能寄存器。包括 DMA 启动，暂停，中断使能等等。在 DMA 操作章节，将详细介绍。

DMA 控制器方框图见 图 8-2

图 8-2. DMA 结构



8.2. DMA 操作

DMA 控制器需要透过用户软件配置。下面章节详述 DMA 的设置和操作。

8.2.1. DMA 传输类型

MG82F6P32 的 DMA 控制器支持 3 种类型数据传输，如下：

- M2P: XRAM 到外设
- P2M: 外设到 XRAM
- P2P:外设到外设

DMA 控制器不支持从 XRAM 到 XRAM (M2M)的数据传输。也不能访问内部数据 RAM 和 Flash ROM。透过软件设置 DMADS0 来指定 DMA 源和目的地的。当外设配置成 DMA 存取，软件不可访问此外设数据寄存器。

范例 1：
ADC12 被选择为源数据和 XRAM 选择为目的地数据。DMA 控制器会将 ADC12 的转换结果寄存器 ADCDH 和 ADCDL 内的值移动到 8051 XRAM。此 DMA 传输类型是外设到 XRAM。传送过程中软件必须避免去读取 ADCDH 和 ADCDL。

范例 2：
ADC12 被选择为源数据和 S0 TX 选择为目的地数据。DMA 控制器会将 ADC12 的转换结果寄存器 ADCDH 和 ADCDL 移动到 S0BUF。DMA 传输类型是外设到外设。传送过程中软件必须避免去读取 ADCDH、ADCDL 和写数据到 S0BUF。

DMA 数据路径选择配置见 表 8-1..

表 8-1. DMA 数据路径选择

DSS30~00 (DMADS0[7:4])	源选择	DDS30~00 (DMADS0[3:0])	目标选择
0 0 0 0	禁止	0 0 0 0	禁止
0 0 0 1	S0 RX	0 0 0 1	S0 TX
0 0 1 0	S1 RX	0 0 1 0	S1 TX
0 0 1 1		0 0 1 1	
0 1 0 0		0 1 0 0	
0 1 0 1	TWI0 RX	0 1 0 1	TWI0 TX
0 1 1 0	TWI1 RX	0 1 1 0	TWI1 TX
0 1 1 1	SPI0 RX	0 1 1 1	SPI0 TX
1 0 0 0		1 0 0 0	
1 0 0 1	ADC0	1 0 0 1	
1 0 1 0		1 0 1 0	
1 0 1 1		1 0 1 1	
1 1 0 0		1 1 0 0	
1 1 0 1		1 1 0 1	CRC
1 1 1 0		1 1 1 0	
1 1 1 1	XRAM	1 1 1 1	XRAM

注：
DMA 不支持“XRAM 到 XRAM”的传输。其它路径支持 DMA 传输

8.2.2. DMA 传输模式

MG82F6P32 的 DMA 控制器仅支持块传输模式，可透过软件或是硬件外部触发去激活一个块数据传输。在 DMA 触发之后，DMA 控制器开始移动数据直到 DMA 当前传输计数器溢出事件发生。

块数据传输大小定义在当前传输计数器{TH5+TL5}中，此计数器同时也代表当前传输数据的地址。DMA 传输支持传输大小从 1 字节到 65536 字节。在 MG82F6P32 中，一次 DMA 数据转移是从源数据到目的地数据移动一个字节。

8.2.3. 传输计数和地址指针

DMA 模块中 DMA 传输计数控制和存储器地址指针分别是定时器 5 和定时器 6。定时器 5 和定时器 6 如同通用定时器 0 一样是一个 16 位计数器(TH5 + TL5, TH6 + TL6)和 16 位重载寄存器(THR5 + TLR5, THR6 + TLR6)。如果 DMA 使能, 定时器 5 控制 DMA 传输计数和定时器 6 指向存储器地址。定时器 5 和定时器 6 总是 16 位向上计数器。

当前传输计数由{TH5 + TL5}寄存器决定转移数量。基本传输计数在{THR5 + TLR5}。支持最大传输计数到 65536。实际传输计数等于(65536 - {TH5 + TL5})的值。当前传输计数器在每一个 DMA 转移之后加一。当寄存器的值从 FFFFH 到 0000H, “DMA 传输结束”事件产生从而清零 DMAS0 停止 DMA 传输并且置位 DMA 完成标志(DCF0)。事件也重载{THR5 + TLR5}到{TH5 + TL5}为下一个 DMA 传输初始化新的当前传输计数器的起始地址。

传输计数初始化例子,

- a. 如果 DMA 大小为 65536, {TH5 + TL5}将编程为 0000H。
- b. 如果 DMA 大小为 1, {TH5 + TL5}将编程为 FFFFH。

当前地址在{TH6 + TL6}寄存器指向 XRAM 的 DMA 访问存储器地址。基于{TH6 + TL6}向上计数功能, 地址产生将加一。基地址在{THR6 + TLR6}中。每一“DMA 传输结束”事件将重载{THR6 + TLR6}到{TH6 + TL6}为下一个 DMA 传输初始化新的当前地址。当前地址覆盖整个 XRAM 存储器。

8.2.4. 启动一个 DMA 传输

MG82F6P32 很容易处理一个 DMA 控制器。启动一个 DMA 传输, 软件必须经过下列顺序来构建一个 DMA 操作:

- 1) 配置 DMADS0 来决定 DMA 传输类型和 DMA 源与目的地的数据路径。
- 2) 配置 DMA 中断和中断优先级。
- 3) 配置当前传输计数和基传输计数。
- 4) 如果 DMA 需要访问 XRAM, 则需要配置当前地址和基地址。
- 5) 配置外设为准备状态。
- 6) 设置 DMAE0 使能 DMA FSM。
- 7) 配置 DMA 触发源和触发 DMA 启动运行。
 - 如果选择软件触发, 软件设置 DMAS0 启动 DMA
 - 如果选择外部触发, 等待外部有效信号启动 DMA
- 8) 软件查询到 DMA 完成标志(DCF0)意味着 DMA 传输结束。
- 9) 写“0”到 DMAE0 结束 DMA 操作和配置 DMADS0 禁止状态。

DMA 外部触发操作中, 外部激活信号自动设置 DMAS0。不论是内部或外部触发, 当 DMA 传输结束 DMAS0 将自动清零, DMA 传输结束。

8.2.5. 暂停或停止 DMA 传输

在传输期间(设置 DMAS0 之后)写“0”到 DMAS0 一个 DMA 转移被暂停。当一个 DMA 数据转移进行中如果信道被暂停, 信道有效禁止仅在当前数据转移完成之后。重新使能 DMAS0 继续 DMA 传输。

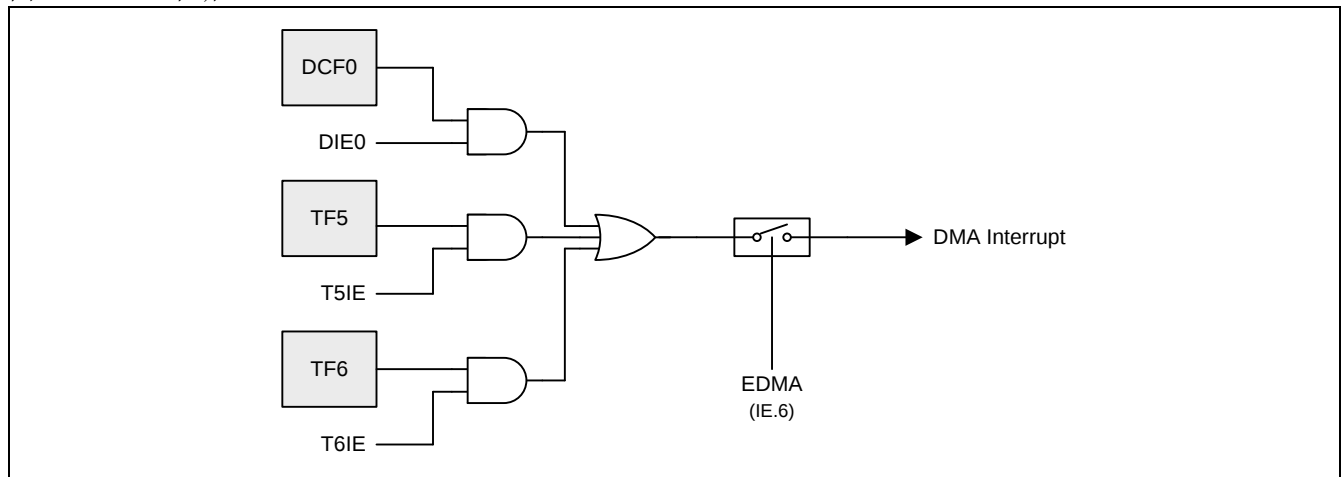
软件写“0”到 DMAE0 任何时候停止当前 DMA 传输或在 DMA 传输结束之后结束 DMA 传输。建议软件配置数据路径(DMADS0)禁止转态和清零 DMAS0。

8.2.6. DMA 中断

在任何传输模式或传输类型，当相关当前传输计数寄存器{TH5 + TL5}计数溢出 DCF0 置位。如果相关 DIE0 和 EDMA (IE.6)已置位，将产生一个 DMA 中断请求。

如果禁止 DMA 功能，DMA 模块的定时器 5 和定时器 6 是一个通用 16 位定时器。每个定时器有自己的定时器标志，TF5 和 TF6 是相关中断使能位。它们共享 DMA 中断。下面方框图展示了 DMA 中断结构。如果软件使能 DMA 传输功能，定时器 5 和定时器 6 的中断使能被禁止。

图 8-3. DMA 中断



8.2.7. DMA 循环模式

循环模式应用于处理循环缓冲和持续数据流(类似 ADC 扫描模式)。这个特性使用 DMACG0 寄存器的 LOOP 位使能。当循环模式激活，当前传输计数器自动重载基传输计数，当前地址自动重载基地址，DMA 请求继续执行不会置位 DMAS0

8.2.8. DMA 的错误处理

DMA 控制器没有任何的错误处理功能，软件须谨慎：

- 当前传输地址不能超出 XRAM 边界。MG82F6P32 的 XRAM 边界是 1792 字节(06FFH)。
- 不支持 S0, S1 奇偶检验位的产生与校验。
- 不能处理 TWI0/I2C0 和 TWI1/I2C1 的无 ACK 状态。

8.2.9. 数据拷贝到 CRC16

如果 DMA 目的地不是 CRC16 模块，使能 CRCW0，每一次 DMA 传输将拷贝数据内容到 CRC16 模块。例如，从 S0 RX 移动到 SPI0 TX 将同步填充数据到 CRC16。任何传输类型支持这个功能。

8.2.10. 定时器 5 &定时器 6

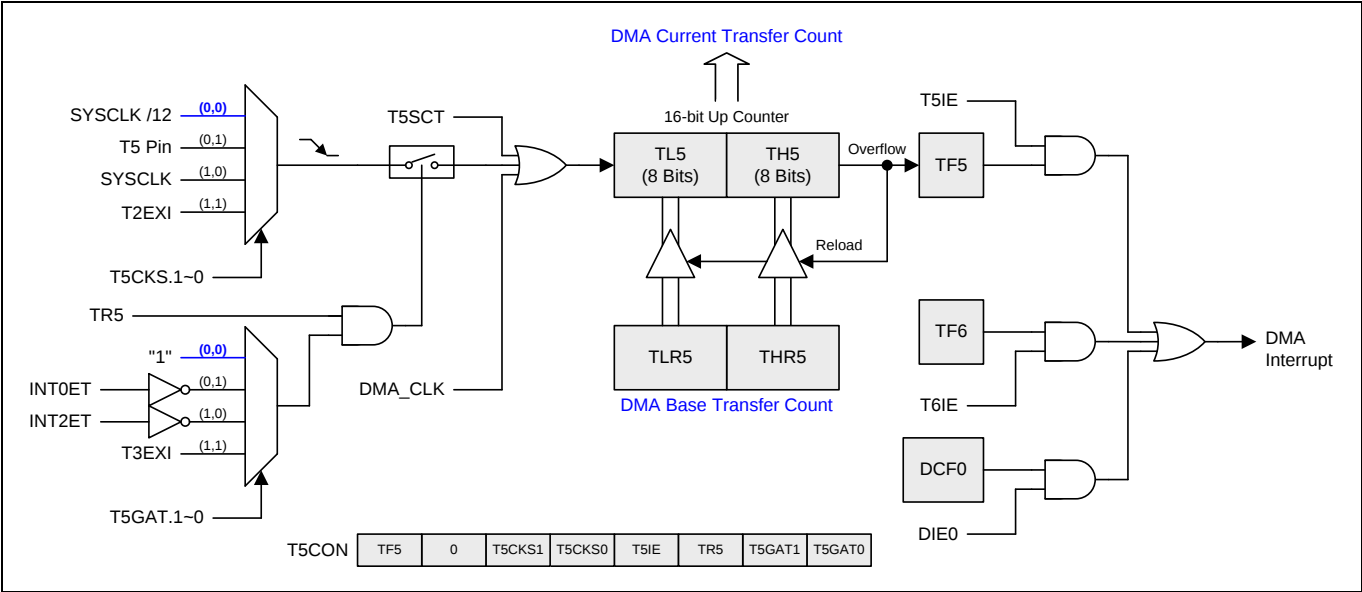
当 DMA 使能，定时器 5 是 DMA 传输计数功能。TH5 和 TL5 是当前传输计数寄存器。THR5 和 TLR5 是基传输计数寄存器。

如果 DMA 禁止，定时器 5 是一个与定时器 0 有着一样门控制功能的 16 位自动重载定时/计数器。溢出标志 TF5，是一个中断源，共享 DMA 中断向量。下图展示了定时器 5 的结构。

定时器 5 引脚配置如下：

T5
P3.4

图 8-4. 定时器 5 结构



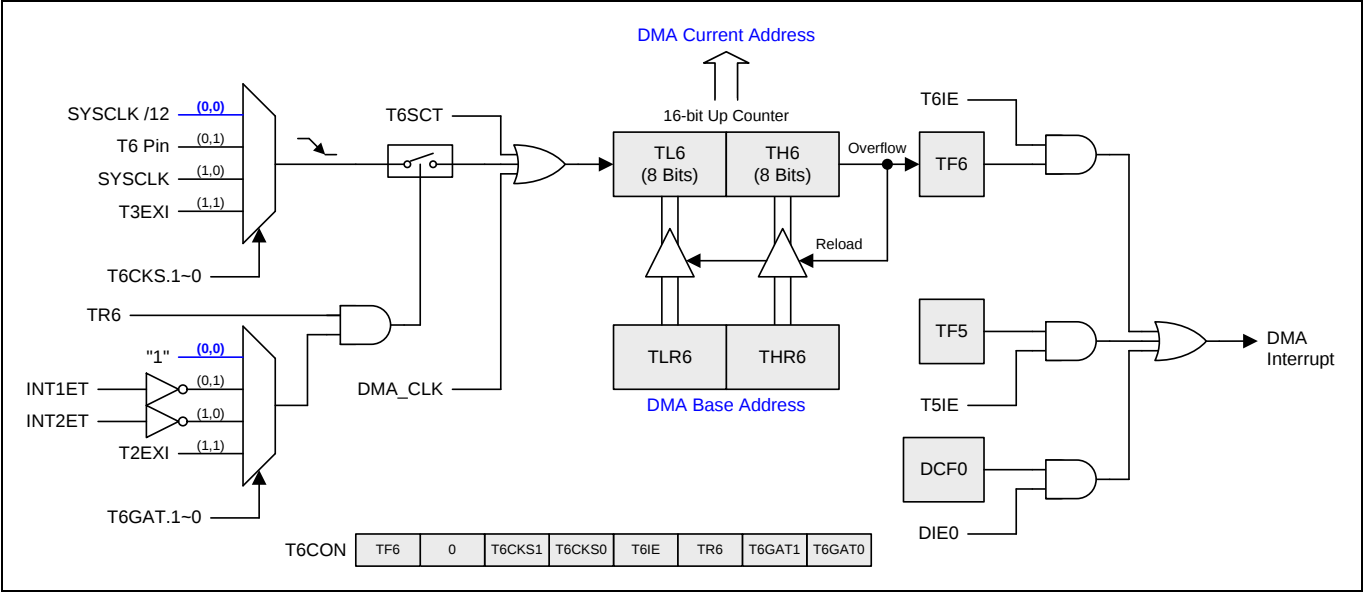
当 DMA 使能，定时器 6 是 DMA 传输地址指针功能。TH6 和 TL6 是当前地址寄存器。THR6 和 TLR6 是基地址寄存器。

如果 DMA 禁止，定时器 6 是一个与定时器 0 有着一样门控制功能的 16 位自动重载定时/计数器。溢出标志 TF6 是一个中断源并共享 DMA 中断向量。下图展示了定时器 6 的结构。

定时器 6 引脚配置如下：

T6
P3.5

图 8-5. 定时器 6 结构



8.3. DMA 寄存器

DMACR0: DMA 控制寄存器 0

SFR 页 = 0~7

SFR 地址 = 0x94

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	DMAE0	DMAS0	DIE0	DCF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: 保留。写该寄存器时，此位必须写“0”。

Bit 3: DMAE0, DMA 使能 0。

0: 清零禁止 DMA 操作。

1: 置位使能 DMA 操作。

Bit 2: DMAS0, DMA 传输启动 0。

0: 当 DMA 传输结束硬件清零。如果软件清零将暂停 DMA 传输。

1: 软件置位启动或继续 DMA 传输。

Bit 1: DIE0, DCF0 中断使能。

0: 禁止 DCF0 中断。

1: 使能 DCF0 中断共享 DMA 中断向量。

Bit 0: DCF0, DMA 完成标志 0。

0: DCF0 必须软件写“0”清零。

1: DMA 传输结束 DCF0 置位。

DMACG0: DMA 配置寄存器 0

SFR 页 = 仅 8 页

SFR 地址 = 0x94

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PDMAH	PDMAL	CRCW0	0	EXTS10	EXTS00	0	LOOP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: PDMAH/L, DMA 中断优先级控制位。

Bit 5: CRCW0, CRC16 写(拷贝)使能。

0: 禁止 DMA 数据同时拷贝到 CRC16。

1: 使能 DMA 数据同时拷贝到 CRC16。

Bit 4: 保留。写该寄存器时，此位必须写“0”。

Bit 3~2: EXTS10~00, DMA 外部触发源选择。

EXTS10, EXTS00	选择触发源
0 0	禁止, 软件触发
0 1	INT2ET
1 0	保留
1 1	KBIET

Bit 1: 保留。写该寄存器时，此位必须写“0”。

Bit 0: LOOP0。

0: 禁止 DMA 循环操作。

1: 使能 DMA 循环操作。

DMADS0: DMA 数据路径选择寄存器 0

SFR 页 = 仅 9 页

SFR 地址 = 0x94 复位值 = 0000-0000

7	6	5	4	3	2	1	0
DSS30	DSS20	DSS10	DSS00	DDS30	DDS20	DDS10	DDS00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: DMA 数据源选择.

Bit 3~0: DMA 数据目的地选择.

DSS30~00 (DMADS0[7:4])	源选择	DDS30~00 (DMADS0[3:0])	目的地选择
0 0 0 0	禁止	0 0 0 0	禁止
0 0 0 1	S0 RX	0 0 0 1	S0 TX
0 0 1 0	S1 RX	0 0 1 0	S1 TX
0 0 1 1	Reserved	0 0 1 1	Reserved
0 1 0 0	Reserved	0 1 0 0	Reserved
0 1 0 1	TWI0 RX	0 1 0 1	TWI0 TX
0 1 1 0	TWI1 RX	0 1 1 0	TWI1 TX
0 1 1 1	SPI0 RX	0 1 1 1	SPI0 TX
1 0 0 0	Reserved	1 0 0 0	Reserved
1 0 0 1	ADC0	1 0 0 1	Reserved
1 0 1 0	Reserved	1 0 1 0	Reserved
1 0 1 1	Reserved	1 0 1 1	Reserved
1 1 0 0	Reserved	1 1 0 0	Reserved
1 1 0 1	Reserved	1 1 0 1	CRC
1 1 1 0	Reserved	1 1 1 0	Reserved
1 1 1 1	On-chip XRAM	1 1 1 1	On-chip XRAM

注意 1: 当使用 DMA 来传输 ADC 数据, 请当心数据位设置, 请参考章节 29.2.9 DMA 传输 ADC 数据

8.4. 定时器 5 寄存器

T5CON: Timer 5 控制寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0xC8

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TF5	0	T5CKS1	T5CKS0	T5IE	TR5	T5GAT1	T5GAT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TF5, 定时器5溢出标志。

0: TF5必须软件清零。

1: 定时器5溢出TF5置位。

Bit 6: 保留。写该寄存器时, 此位必须写“0”。

Bit 5~4: T5CKS.1~0, 定时器5时钟源选择。

T5CKS.1~0	T5 时钟选择
00	SYSClk/12
01	T5 Pin
10	SYSClk
11	T2EXI 输入选择

T5 Pin: P3.4

Bit 3: T5IE, TF5 中断使能。

0: 禁止 TF5 中断。

1: 使能 TF5 中断共享 DMA 中断向量。

Bit 2: TR5, 定时器 5 运行控制位。

0: 清零, 停止定时/计数器 5。在启动 DMA 之前, 软件必须禁止 TR5。

1: 置位, 启动定时/计数器 5。

Bit 1~0: T5GAT.1~0, 定时器 5 的门控源选择。

T5GAT.1~0	T5 门控源
00	禁止
01	反向 INTOET
10	反向 NT2ET
11	T3EXI 输入选择

TL5: 定时器 5 低字节寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0xCC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL5.7	TL5.6	TL5.5	TL5.4	TL5.3	TL5.2	TL5.1	TL5.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH5: 定时器 5 高字节寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0xCD

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TH5.7	TH5.6	TH5.5	TH5.4	TH5.3	TH5.2	TH5.1	TH5.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TLR5: 定时器 5 低字节重载寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0xCA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TLR5.7	TLR5.6	TLR5.5	TLR5.4	TLR5.3	TLR5.2	TLR5.1	TLR5.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

THR5: 定时器5 高字节重载寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0xCB

复位值 = 0000-0000

7	6	5	4	3	2	1	0
THR5.7	THR5.6	THR5.5	THR5.4	THR5.3	THR5.2	THR5.1	THR5.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

8.5. 定时器6 寄存器

T6CON: 定时器6 控制寄存器

SFR 页 = 仅 4 页

SFR 地址 = 0xC8

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TF6	0	T6CKS1	T6CKS0	T6IE	TR6	T6GAT1	T6GAT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TF6, 定时器6溢出标志。

0: TF6必须软件清零。

1: 定时器6溢出TF6置位。

Bit 6: 保留。写该寄存器时，此位必须写“0”。

Bit 5~4: T6CKS.1~0, 定时器6时钟源选择

T6CKS.1~0	T6 时钟源
00	SYSClk/12
01	T6 Pin
10	SYSClk
11	T3EXI 输入选择

T6 Pin: P3.5

Bit3: T6IE, TF6 中断使能。

0: 禁止 TF6 中断。

1: 使能 TF6 中断共享 DMA 中断向量。

Bit2: TR6, 定时器6 运行控制位。

0: 清零，停止定时/计数器6。在启动 DMA 之前，软件必须禁止 TR6。

1: 置位，启动定时/计数器6。

Bit1~0: T6GAT.1~0, 定时器6 的门控源选择。

T6GAT.1~0	T6 门控源
00	禁止
01	反向 INT1ET
10	反向 INT2ET
11	T2EXI 输入选择

TL6: 定时器6 低字节寄存器

SFR 页 = 仅 4 页

SFR 地址 = 0xCC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL6.7	TL6.6	TL6.5	TL6.4	TL6.3	TL6.2	TL6.1	TL6.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH6: 定时器6 高字节寄存器

SFR 页 = 仅 4 页

SFR 地址 = 0xCD

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TH6.7	TH6.6	TH6.5	TH6.4	TH6.3	TH6.2	TH6.1	TH6.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
THR6.7	THR6.6	THR6.5	THR6.4	THR6.3	THR6.2	THR6.1	THR6.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

9. 系统时钟

系统时钟有 4 个时钟来源：内部快频 RC 震荡器 (IHRCO)，外部晶振，内部慢频 RC 震荡器(ILRCO) 和外部频率输入。如图 9-1 所示 MG82F6P32 系统时钟结构。

MG82F6P32 总是由 IHRCO 12MHz 启动，并且保留振荡脚用作 P6.0/P6.1 GPIO 功能。软件可以根据应用要求自由切换 4 种时钟的任意一种作为系统时钟，但必须等时钟稳定后才能切换。如果软件选择外部晶振模式，P6.0 和 P6.1 将用于 XTAL2 和 XTAL1。并且 P6.0/P6.1 GPIO 功能被禁止。在外部时钟输入模式 (ECKI)，时钟源来自 P6.0，P6.1 仍保留 GPIO 功能。

设置 XTALE (CKCON2.5)，使能外部晶振。当来自外部晶振的频率稳定后，XTOR (CKCON1.7)将自动置位。在切换系统时钟源（OSCin）到外部晶振之前，软件需要轮询这个位。XTOR 是只读的。

内建 IHRCO 提供两种频率供软件选择。通过软件置位 AFS(CKCON0.7)选择另一个频率 11.0592MHz。IHRCO 的 12MHz 和 11.059 MHz 都可以给系统时钟提供高精度的频率。详细的 IHRCO 性能，请参考章节“37.5 IHRCO”。在 IHRCO 或 ILRCO 模式，P6.0 可以作为内部 MCK 或 2 分频时钟(MCK/2)输出或 4 分频时钟(MCK/4)输出给其他系统时钟源应用。

内置 ILRCO 提供约 32KHz 的低功耗，低速频率给 WDT 和系统时钟源使用。对于需低功耗运行的软件，MCU 可以选择 ILRCO 作为系统时钟源。若要查找详细的 ILRCO 性能，请参考章节“37.6 ILRCO”。在 ILRCO 模式下，可以将 P6.0 配置为内部 MCK 或 2 分频时钟(MCK/2)输出或 4 分频时钟(MCK/4)输出为系统中的应用。

MG82F6P32 2 包含了一个时钟倍频器(CKM)产生高速时钟用于系统时钟源。**MG82F6P32** 的 CKM 应用如图 9-1 所示，它的典型输入频率是 6MHz。在使能 CKM 之前，软件必须配置 CKMIS1~0(CKCON.5~4)获得适当的 CKMI 频率用作 CKM 输入源。CKM 可以产生 CKMHI 的 4/5.33/8 倍的频率，通过设置 MCKS1~0(CKCON2.3~2)选择不同的 CKM 输出，为 MCU 提供高速操作，而无需高频时钟源。要找到详细的 CKM 性能，请参考章节“37.7 CKM”。

通过时钟分配器分配 4 种时钟源的一种作为系统时钟(SYSCLK)，如图 9-1..所示。用户能通过设置 CKCON0 寄存器的 SCKS2~SCKS0 位获得适当的系统时钟。

9.1. 时钟结构

MG82F6P32 的主要时钟系统如图 9-1 所示。CPUCLK 初始时钟源是内部高速振荡器 IHRCO 12MHz。结合时钟倍频器和分频器产生不同的频率。最大 CPUCLK 如下：

- 外部晶振模式：最大到 12MHz @ 2.0V – 5.5V；最大到 25MHz @ 2.4V – 5.5V
- CPU 最大到 12MHz @ 1.8V– 5.5V；最大到 25MHz @ 2.2V– 5.5V
- CPU 最大到 32MHz @ 2.7V -5.5V 使用片内倍频器(CKM)

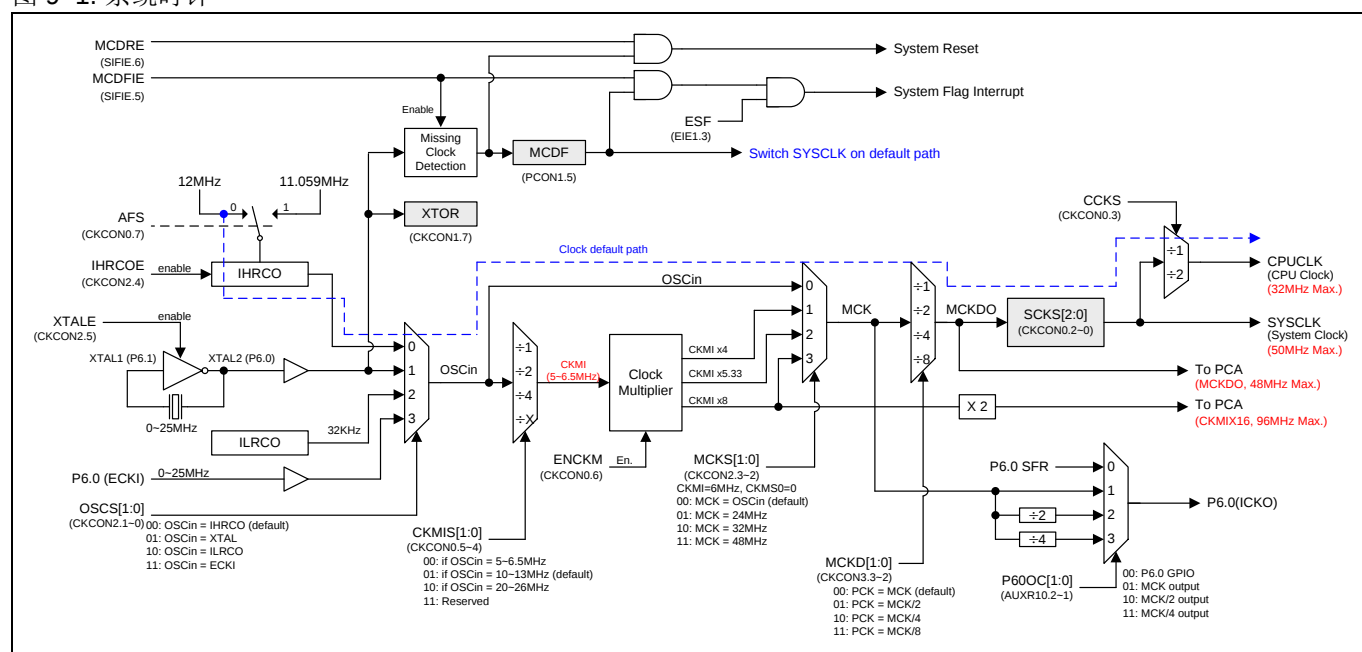
如果需要更高频的应用，当 CPUCLK > 6MHz 时 HSE(DCON0.7)需要置位。此外，当 CPUCLK>25MHz 时 HSE1 也需要置位。

系统时钟（SYSCLK）来自于外部晶振电路或内部振荡器。最大的时钟频率是 50MHz。请注意：当使用时钟倍频器(CKM)来提高 MCK 频率获得更高的 SYSCLK 时，CPUCLK 也将改变。在 MCK 频率提升之前需要置位 CCKS 来降低 CPUCLK，避免 CPUCLK 超频(CPUCLK 必须低于 25MHz 或 32MHz)。

时钟模块也提供两个时钟源给高速 PCA 应用。

- MCKDO：最大可达 48MHz
- CKMIX16：最大可达 96MHz

图 9-1. 系统时钟



9.2. 时钟源切换

系统时钟有 4 个时钟来源：内部快频 RC 震荡器(IHRCO)，外部晶振，内部慢频 RC 震荡器(ILRCO)和外部时钟输入。如图 9-1 所示 MG82F6P32 系统时钟结构。MG82F6P32 总是由 IHRCO 12MHz 启动，软件可以通过设定 OSCS[1:0]来选择时钟源，但必须等时钟稳定后才能切换。

9.3. 片内 CKM (PLL)

MG82F6P32 包含一个时钟倍增器(CKM)可以产生高速的时钟作为系统时钟。如图 9-1 所示其典型的输入频率是 6M 左右。软件必须设置 CKMIS1~0(CKCON.5~4)得到恰当的 CKMI 频率作为 CKM 的输入源。CKM 能从 CKMI 得到 4/5.33/8 倍的增益的时钟频率，设置 MCKS1~0(CKCON2.3~2)选则不同的 CKM 输出，MCK 能提供 MCU 没有的高频时钟源来高速操作。详细的 CKM 功能，请参考章节“37.7 CKM”。

9.4. 时钟丢失监测(MCD)

当使用外部晶振器作为时钟源时，它可以被时钟丢失监测器 MCD 监控，以标识振荡是否停止工作。时钟丢失监测模块由 MCD FIE 启用。如果 MCD FIE 被清零，则时钟丢失监测模块处于非活动状态。MCD F(PCON1 .5), 时钟丢失监测(MCD)标志位，由硬件置位，用于检测外部晶振输入丢失事件。在这个位上写“1”将清除 MCD F。一旦时钟丢失发生，软件必须在再次切换 OSCin 到 XTAL 之前清除 MCD F。默认情况下，MCD 事件将触发系统复位。如果用户不要此功能，清零 MCD RE 以禁用复位功能。

9.5. XTAL 模式下快速唤醒

一般外部晶振的稳定时间是 0.6ms ~ 2ms (取决于应用)。MCU 内部 IHRC 稳定时间是 200us，要快于晶振。这就可用在 XTAL 模式下唤醒的速度。在 XTAL 模式下，进入掉电模式之前用户可以将 OSCin 从 XTAL 切换到 IHRCO。

XTAL 模式下如何编程快速唤醒功能

- 设置 IHRCOE (CKCON2.4)为“1”使能 IHRCO.
- 延时 32us 等待 IHRCO 工作稳定
- 编程 OSCS[1:0] (CKCON2.1~0)为“00”，选择 IHRCO 为时钟源
- NOP x 10
- 设置 XTAL E (CKCON2.5)为“0”，禁止外部晶振电路
- MCU 进入掉电模式
-
- MCU 唤醒
- 设置 XTAL E (CKCON2.5)为“1”，使能外部晶振电路
- 轮询 XTOR (CKCON1.7)是否为“1”，等待外部晶振稳定
- 编程 OSCS[1:0] (CKCON2.1~0)为“01”，选择 XTAL 为时钟源
- NOP x 10
- 设置 IHRCOE (CKCON2.4)为“0”，禁止 IHRCO.
- 继续执行程序.....

9.6. CKM 下的唤醒

当使能 CKM 电路，它需要 100us 才能输出稳定的频率，在这个不确定频率的周期内，MCK 的输入必须保持 MCKS 在 OSCin 上，于保证系统的稳定。请参考下列流程：

时钟来源 CKM 下如何编程唤醒

- 编程 MCKS[1:0](CKCON2.3~2)为“00”选择非 CKM 输出用于时钟源
- MCU 进入掉电模式
-
- MCU 唤醒
- 延时 100us 等待 CKM 工作稳定
- 修改 MCKS[1:0](CKCON2.3~2)选择 CKM 输出用于时钟源
- 继续执行程序.....

9.7. 时钟寄存器

CKCON0: 时钟控制寄存器0

SFR 页 = 0~F & P

SFR 地址 = 0xC7

复位值 = 0001-0000

7	6	5	4	3	2	1	0
AFS	ENCKM	CKMIS1	CKMIS0	CCKS	SCKS2	SCKS1	SCKS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: AFS, 交替频率选择。

0: 选择 IHRCO 为 12MHz。

1: 选择 IHRCO 为 11.059MHz。

Bit 6: ENCKM, 使能时钟倍频器 (X8)。

0: 禁止 X8 时钟倍频器。

1: 使能 X8 时钟倍频器。

Bit 5~4: CKMIS1 ~ CKMIS0, 时钟倍频器输入选项

CKMIS[1:0]	时钟倍频器输入选项
0 0	OSCin/1 (当 OSCin = 5 ~ 7MHz)
0 1	OSCin/2 (当 OSCin = 10 ~ 14MHz)
1 0	OSCin/4 (当 OSCin = 20 ~ 28MHz)
1 1	保留

Bit 3: CCKS, CPU 时钟选择。

0: 选择 SYSCLK 为 CPU 时钟。

1: 选择 SYSCLK/2 为 CPU 时钟。

Bit 2~0: SCKS2 ~ SCKS0, 可编程系统时钟选项。

SCKS[2:0]	系统时钟(SYSCLK)
0 0 0	MCKDO/1
0 0 1	MCKDO/2
0 1 0	MCKDO/4
0 1 1	MCKDO/8
1 0 0	MCKDO/16
1 0 1	MCKDO/32
1 1 0	MCKDO/64
1 1 1	MCKDO/128

CKCON1: 时钟控制寄存器1

SFR 页 = 0~F

SFR 地址 = 0xBF

复位值 = 0001-0001

7	6	5	4	3	2	1	0
XTOR	PLLOCKF	MCKSTA1	MCKSTA0	OSCSTA3	OSCSTA2	OSCSTA1	OSCSTA0
R	R	R	R	R	R	R	R

Bit 7: XTOR, 晶振电路准备位. 只读。

0: 晶振没有准备好。

1: 晶振已准备好。当使能 XTLE, XTOR 报告晶振已达到启动计数。

Bit 6: PLLOCKF, PLL 电路锁定准备位. 只读

0: PLL 没有锁定

1: PLL 已经锁定。

Bit 5~4: MCKSTA[1:0], MCK MUX 状态

00: MCK MUX 正在切换时钟

01: MCK MUX 正在使用 OSCin 作为时钟源

10: MCK MUX 正在使用 CKMIX4/x5.33/x8 作为时钟源

11: MCU MUX 正在切换时钟

Bit 3~0: OSCSTA[3:0]

0001: OSCin MUX 正在使用 IHRCO

0010: OSCin MUX 正在使用 XTAL

0100: OSCin MUX 正在使用 ILRCO

1000: OSCin MUX 正在使用 ECKI

其它: OSCin MUX 正在切换时钟

CKCON2: 时钟控制寄存器 2

SFR 页 = 仅 P 页

SFR 地址 = 0x40

复位值 = 0101-0000

7	6	5	4	3	2	1	0
XTGS1	XTGS0	XTALE	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: XTGS1~XTGS0, 晶振增益选择位.

XTGS1, XTGS0	增益定义	晶振应用
0, 0	低增益	32.768KHz
0, 1	中等增益	2MHz ~ 25MHz
1, 0	较低增益	32.768KHz
1, 1	保留	保留

Bit 5: XTALE, 外部晶振 (XTAL) 使能

0: 禁止 XTAL 振荡电路, 在这种情况下, XTAL2 和 XTAL1 表现为端口 6.0 和 端口 6.1。

1: 使能 XTAL 振荡电路, 如果 CPU 软件设置这个位, 软件检测到 **XTOR** (CKCON1.7) 为“1”表明晶振振荡器准备好作为 OSCin 时钟选择。

Bit 4: IHRCOE, 内部高频 RC 震荡使能位

0: 禁止内部高频 RC 震荡电路

1: 使能内部快频 RC 震荡电路。如果软件设置这个位, 在 IHRCOE 位使能后, 必须等待 **32 us** IHRCOE 才能稳定输出

Bit 3~2: MCKS[1:0], MCK 时钟源选择

MCKS[1:0]	MCK 时钟源选择	OSCin = 12MHz CKMIS = [01]	OSCin = 11.059MHz CKMIS = [01]
0 0	OSCin	12MHz	11.059MHz
0 1	CKMI x4	24MHz	22.118MHz
1 0	CKMI x5.33	32MHz	29.491MHz
1 1	CKMI x8	48MHz	44.236MHz

注意: 需要设置 ENCKM = 1 使能 CKM.

注意: 需要注意 CPUCLK 和 SYSCLK 的限制。需要使用 SCKS[2:0] 和 CCKS 选择 CPUCLK 和 SYSCLK 的正确范围不要超出限制。CPUCLK ≤ 32MHz, SYSCLK ≤ 50MHz。

Bit 1~0: OSCS[1:0], OSCin 时钟源选择.

OSCS[1:0]	OSCin 时钟源选择
0 0	IHRCO
0 1	XTAL
1 0	ILRCO
1 1	ECKI, External Clock Input (P6.0) as OSCin.

CKCON3: 时钟控制寄存器 3

SFR 页 = 仅 P 页

SFR 地址 = 0x41

复位值 = 0000-0010

7	6	5	4	3	2	1	0
WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	MCDS1	MCDS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: WDTCS1~0. WDT 时钟源选择。

Bit 5: FWKP, MCU 快速唤醒控制。

0: 选择 MCU 从掉电模式正常唤醒时间大约 120us。

1: 选择 MCU 从掉电模式快速唤醒时间大约 30us。

Bit 4: WDTFS. WDT 溢出源选择。

0: 选择 WDT 位-8 溢出作为 WDT 事件源。

1: 选择 WDT 位-0 溢出作为 WDT 事件源。

Bit 3~2: MCKD[1:0], MCK 分频输出选择。

MCKD[1:0]	MCKDO 频率	如果 MCK = 12MHz	如果 MCK = 48MHz
0 0	MCKDO = MCK	MCKDO = 12MHz	MCKDO = 48MHz
0 1	MCKDO = MCK/2	MCKDO = 6MHz	MCKDO = 24MHz
1 0	MCKDO = MCK/4	MCKDO = 3MHz	MCKDO = 12MHz
1 1	MCKDO = MCK/8	MCKDO = 1.5MHz	MCKDO = 6MHz

Bit 1~0: MCDS[1:0], 保留用于测试

AUXR10: 辅助寄存器 10

SFR 页 = 仅 7 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 2~1: P6.0 功能配置控制位 1 和位 0。这两位仅仅当内部 RC 振荡 (IHRCO 或 ILRCO) 被选择为系统时钟源时有效。外部晶振模式, P6.0 和 P6.1 为 XTAL2 和 XTAL1 功能。外部时钟输入模式下, P6.0 专用于时钟输入。在内部振荡模式, P6.0 为普通 I/O 或时钟源发生器提供下列选项。当 P60OC[1:0] 索引为非 P6.0 GPIO 功能时, P6.0 将驱动内部 RC 振荡器输出为其它设备提供时钟源。

P60OC[1:0]	P60 功能	I/O 模式
00	P60	By P6M1.0 & P6M0.0
01	MCK	By P6M1.0 & P6M0.0
10	MCK/2	By P6M1.0 & P6M0.0
11	MCK/4	By P6M1.0 & P6M0.0

P6.0 作为时钟输出功能时, 建议设置{P6M1.0, P6M0.0}为“01”来选着 P6.0 为推挽输出模式。

Bit 0: P60FD, P6.0 快速驱动。

0: P6.0 默认驱动输出。

1: P6.0 快速驱动输出使能。若 P6.0 被配置为时钟输出, 当 P6.0 输出频率大于 12MHz (5V) 或者大于 6MHz (3V) 的应用时使能此位。

PCON1: 电源控制寄存器 1

SFR 页 = 0~F & P

SFR 地址 = 0x97

POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: MCDF, 时钟丢失监测标志位

0: 此位必须软件写“1”清零。软件写“0”则无操作。

1: 监测到外部晶振输入的一个丢失时钟事件此位硬件置位。此位写“1”则清零 MCDF。丢失时钟监测模块由 MCDFIE 使能。如果 MCDFIE 清零，丢失时钟监测模块则无效。一旦一个丢失时钟事件发生，在再次选择外部晶振输入之前 MCDF 必须清零。

SFIE: 系统标志中断使能寄存器

SFR 页 = 0~F

SFR 地址 = 0x8E

POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 6: MCDRE, 使能丢失时钟事件产生系统复位。

0: 禁止 MCD 事件触发系统复位。

1: 使能 MCD 事件触发系统复位。默认是使能的。

Bit 5: MCDFIE, 使能 MCDF (PCON1.5) 中断。

0: 禁止 MCDF 中断。

1: 使能 MCD 模块和使能 MCDF 中断。

DCON0: 设备控制寄存器 0

SFR 页 = 仅 P 页

SFR 地址 = 0x4C

POR = 1000-0011

7	6	5	4	3	2	1	0
HSE	IAPO	HSE1	0	0	IORCTL	RSTIO	OCDE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: HSE, 高速运行使能。

0: 选择 CPU 运行在低速模式($F_{CPUCLK} \leq 6MHz$)这样减慢内部电路从而降低功耗。

1: 使能 MCU 全速运行($F_{CPUCLK} > 6MHz$)。在 SYSCCLK 选择高频时钟(>6MHz)之前，软件必须置位 HSE 切换到用于高速运行的内部电路。

Bit 5: HSE1, 超高速运行使能。

0: 无功能。

1: 使能 MCU 超高速运行。(F_{CPUCLK} > 25MHz)当使用 HSE1=1 需要置位 HSE。

10. 看门狗定时器(WDT)

10.1. WDT 结构

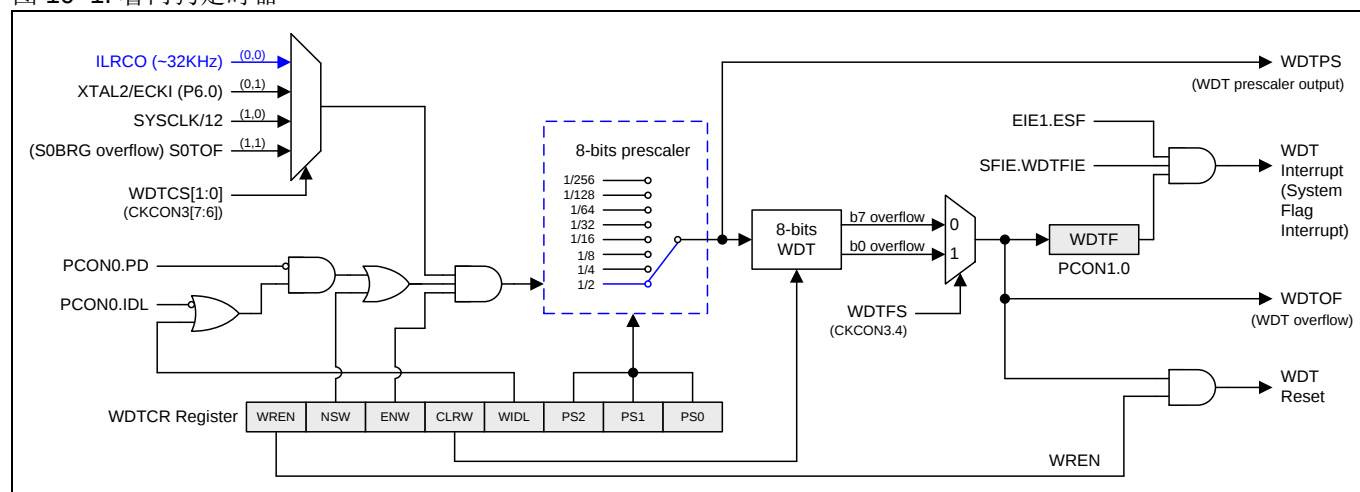
看门狗定时器(WDT) 用来使程序从跑飞或死机状态恢复的一个手段。WDT 由一个 8 位独立定时器、一个 8 分频器和一个控制寄存器(WDTCR)组成。MG82F6P32 的 WDT 结构框图如图 10-1 所示。

WDT 时钟源有四种选择。必须在启用 WDT 之前配置时钟源。默认的 WDT 时钟源是 32 KHz ILRCO。WDT 溢出将设置 WDTF (PCON1.0)，可以通过使能 WDTFIE (SFIE .0)和 ESF (EIE1.3)来配置 WDTF 以产生中断。当置位 WREN (WDTCR.7)时，溢出也会触发系统复位。为了防止 WDT 溢出，软件需要在 WDT 溢出之前向 CLRW 位 (WDTCR.4)写入“1”来清零 WDT。

一旦通过设置 ENW 位启用了 WDT，就没有办法禁用它，除非通过上电复位或在 P 页 SFR 上重置 ENW，这将清除 ENW 位。硬件（RST 引脚）复位、软件复位、WDT 复位后，WDTCR 寄存器保持以前的值不变。

WREN, NSW 和 ENW 具有一次使能生效功能，只要在通用 SFR 页写“1”则生效。P 页下的 SFR 访问 WDTCR 可以通过写“0”到 WDTCR.7~5 禁止 WREN, NSW 和 ENW。详情请参考章节“10.4 WDT”和“33 P 页 SFR 访问”。

图 10-1. 看门狗定时器



10.2. 空闲模式下的 WDT

在空闲模式下，标志位 WIDL(WDTCR.3)决定 WDT 是否计数。设置这个位能让 WDT 在空闲模式一直计数。如果硬件选项 NSWDT 使能，WDT 会一直保持计数不管位 WIDL 设置情况。

10.3. 掉电模式下的 WDT (自动唤醒)

在掉电模式下，如果 NSW(WDTCR.6)使能，ILRCO 将不会停止。MCU 进入 Watch 模式，执行自动唤醒功能。这使得 WDT 即使在掉电模式(Watch 模式)下也能继续计数。当 WDT 溢出后，它将通过软件配置唤醒 CPU 进入中断或复位。这个功能只有当 WDT 时钟源来自 ILRCO 或 P6.0 输入时才有效，P6.0 输入可以来自外部输入或使能的晶振电路(XTAL1/XTAL2)。

10.4. WDT 寄存器

WDTCR: 看门狗定时器控制寄存器

SFR 页 = 0~F & P

SFR 地址 = 0xE1

POR = XXX0-XXXX (X:取决于硬件选项)

7	6	5	4	3	2	1	0
WREN	NSW	ENW	CLRW	WIDL	PS2	PS1	PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: WREN, WDT 复位使能标志, 初始值随硬件选项 WRENO。

0: WDT 溢出不产生复位。WDT 溢出标志 WDTF 可以供软件检测或触发中断。

1: WDT 溢出产生系统复位。一旦 WREN 已经设置, 不能用软件在页 0~F 中清除, 但在 P 页中, 软件能修改其值“0”或“1”。

Bit 6: NSW, 不停止的 WDT 标志。初始值随硬件选项 NSWDT。

0: WDT 在掉电模式停止计数 MCU。

1: WDT 在掉电模式(Watch Mode)或空闲模式下永远不会停止计数 MCU。一旦 NSW 已经设置, 不能用软件在页 0~F 中清除, 但在 P 页中, 软件能修改其值“0”或“1”。

Bit 5: ENW, 使能 WDT 标志。

0: 禁止 WDT 运行。此位仅能被 POR 清除。

1: 使能 WDT。一旦 ENW 位被设置, 不能用软件在页 0~F 中清除, 但在 P 页中, 软件能修改其值“0”或“1”。

Bit 4: CLRW, WDT 清零位。

0: 写“0”到此位 WDT 没有任何操作。

1: 写“1”到此位会清除 8 位 WDT 计数器到 000H。注意此位没有必须写“0”清除.当此位设置“1”时清除 WDT 重新计数。

Bit 3: WIDL, WDT 空闲模式控制位。

0: MCU 在空闲模式下 WDT 停止计数。

1: MCU 在空闲模式下 WDT 保持计数。

Bit 2~0: PS2 ~ PS0, 选择分频器输出作 WDT 基准时钟输入。

当 WDTFS (CKCON3.4) = 0, WDT 时钟源= ILRCO 或 SYSCLK/12

PS[2:0]	Prescaler Value	WDT 时长 (WDT 时钟= ILRCO)	WDT 时长 (WDT 时钟 = SYSCLK/12) (SYSCLK = IHRCO, 12MHz)
0 0 0	2	16 ms	0.512 ms
0 0 1	4	32 ms	1.024 ms
0 1 0	8	64 ms	2.048 ms
0 1 1	16	128 ms	4.096 ms
1 0 0	32	256 ms	8.192 ms
1 0 1	64	512 ms	16.384 ms
1 1 0	128	1024 ms	32.768 ms
1 1 1	256	2048 ms	65.536 ms

当 WDTFS (CKCON3.4) = 1, WDT 时钟源= ILRCO

PS[2:0]	Prescaler Value	WDT Period (clock source = ILRCO)	
0 0 0	2	0.125 ms	245 us= 125+120
0 0 1	4	0.25 ms	370 us= 250+120
0 1 0	8	0.5 ms	620 us= 500+120
0 1 1	16	1 ms	1.12 ms= 1ms+120
1 0 0	32	2 ms	2.12 ms= 2ms+120
1 0 1	64	4 ms	4.12 ms= 4ms +120
1 1 0	128	8 ms	8.12 ms= 8ms +120
1 1 1	256	16 ms	16.12ms= 16ms+120

CKCON3: 时钟控制寄存器 3

SFR 页 = 仅 P 页

SFR 地址 = 0x41

复位值 = 0000-0010

7	6	5	4	3	2	1	0
WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	MCDS1	MCDS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: WDTCS1~0, WDT 时钟源选择位[1:0].

WDTCS1~0	WDT 时钟源
00	ILRCO
01	XTAL2/ECKI (P6.0)
10	SYSCLK/12
11	S0TOF

Bit 4: WDTFS, WDT 溢出源选择位。

0: 选择 WDT 位 8 溢出作为 WDT 事件源。

1: 选择 WDT 位 0 溢出作为 WDT 事件源。

PCON1: 电源控制寄存器 1

SFR 页 = 0~F & P

SFR 地址 = 0x97

POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 0: WDTF, WDT 溢出标志。

0: 必须由软件写“1”清除, 软件写“0”不操作。

1: 当 WDT 溢出时硬件置位此位, 写“1”清除 WDTF。

SFIE: 系统标志中断使能寄存器

SFR 页 = 0~F

SFR 地址 = 0x8E

POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 0: WDTFIE, 使能 WDTF (PCON1.0) 中断

0: 禁止 WDTF 中断。

1: 使能 WDTF 中断

10.5. WDT 硬件选项

除了由软件初始化外，WDTCR 寄存器还能在上电的时候由硬件选项 WRENO、NSWDT、HWENW、HWWIDL 和 HWPS[2:0]来自动初始化，这些选项通过通用编程器来编程，如下所述。

如果 HWENW 编程为“使能”，则硬件在上电时为 WDTCR 寄存器作如下的初始化工作：(1)位 ENW 置 1。(2)载入 WRENO 的值到 WREN 位。(3)载入 NSWDT 的值到 NSW 位。(4)载入 HWWIDL 的值到 WIDL 位。(5)载入 HWPS[2:0]的值到 PS[2:0]位。

如果 HWENW 和 WDSFWP 都被编程为“使能”，则硬件仍然会在上电时由 WDT 硬件选项初始化 WDTCR 寄存器的内容。之后，任何对 WDTCR 的位的写动作都会被忽略，除了写“1”到 WDTCR.4(CLRW)位来清 WDT 之外，即使通过对 P 页 SFR 的操作机制也不行。

WRENO:

- ☒: 使能。置位 WDTCR.WREN 以使能 WDTF 系统复位功能。
- ☐: 禁止。清除 WDTCR.WREN 以禁止 WDTF 系统复位功能。

NSWDT: WDT 不停止。

- ☒: 使能。置位 WDTCR.NSW 使能 WDT 在掉电模式(watch 模式)也保持运行。
- ☐: 禁止。清除 WDTCR.NSW 禁止 WDT 在掉电模式下(watch 模式)运行。

HWENW: 硬件载入 WDTCR 的“ENW”。

- ☒: 使能。上电时自动硬件使能看门狗定时器，并且自动加载 WRENO、NSWDT、HWWIDL 和 HWPS2~0 的值到 WDTCR 中。
- ☐: 禁止：上电时看门狗定时器(WDT)不自动使能。

HWWIDL, HWPS2, HWPS1, HWPS0:

当 HWENW 被使能，上电复位时，这四个保险丝位将被载入到特殊功能寄存器 WDTCR 中。

WDSFWP:

- ☒: 使能。特殊功能寄存器 WDTCR 中的 WREN、NSW、WIDL、PS2、PS1 和 PS0 软件写保护。
- ☐: 禁止。特殊功能寄存器 WDTCR 中的 WREN、NSW、WIDL、PS2、PS1 和 PS0 可被软件改写。

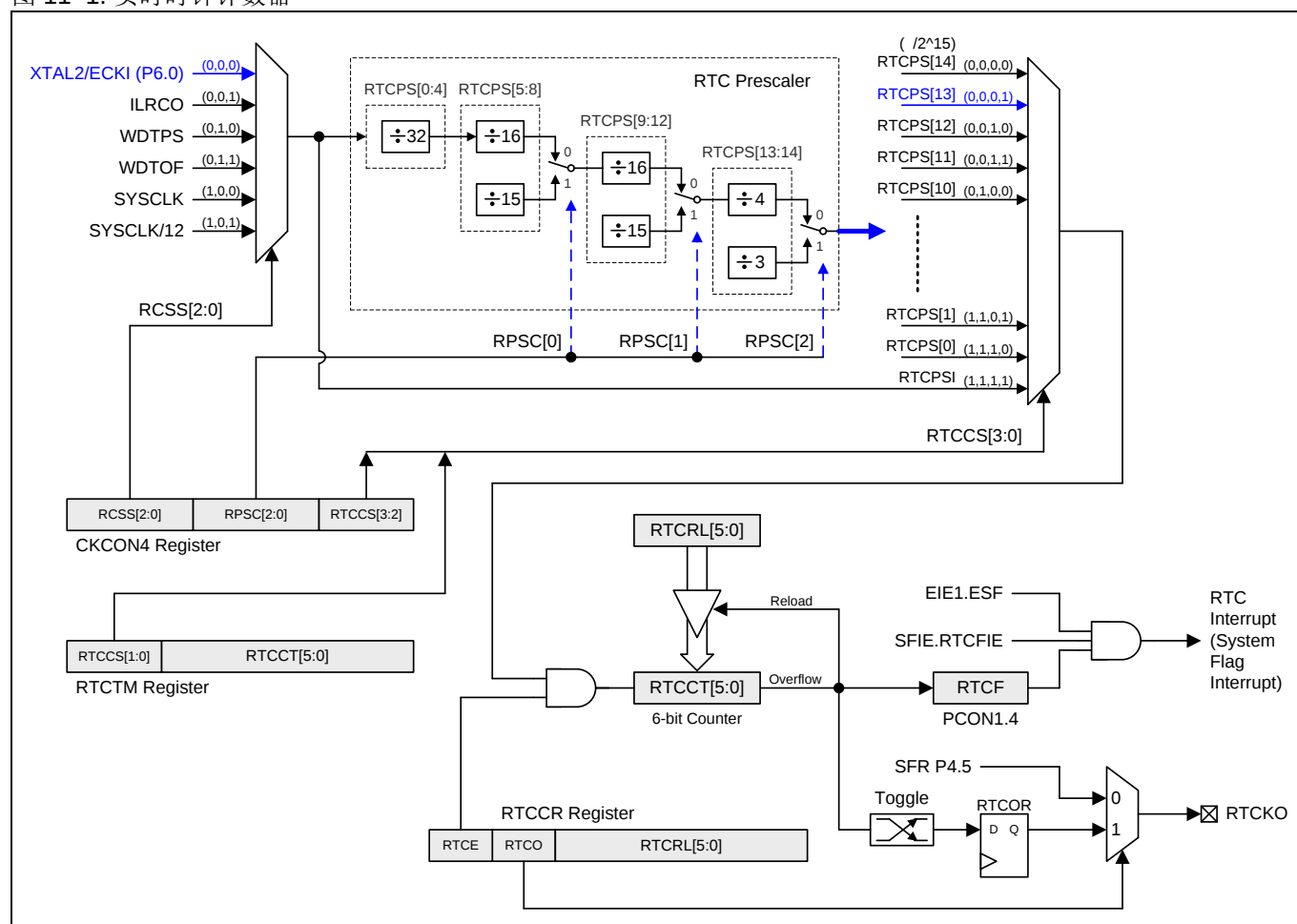
11. 实时时钟(RTC)/系统时钟

MG82F6P32 有一个简单的实时时钟，允许用户在设备的其余部分掉电时继续运行一个精确的计时器。实时时钟可以是唤醒器，也可以是中断源。实时时钟是一个最大 21 位向上计数器，由一个 0~15 位的预分频器和一个 6 位的可重载计数器组成。当它溢出时，将重载 6 位计数器，并置 **RTCF** 标志位。该预分频器的时钟源有 6 个选择，需要设置 **RCSS[2:0]** 来选择一个源，然后才能启用 **RTC**。图 11-1 显示了 **MG82F6P32** 的 **RTC** 结构。设置 **RTCCS[3:0]**, **RTCCT[5:0]**, **RTCRL[5:0]** 和 **RTCO** 必须在使能 **RTC** 功能之前，当 **RTC** 使能后，为防止时长异常，设置更改将被忽略。

使用 32.768 KHz 的晶振作为 **RTC** 模块输入将提供一个可编程的溢出周期为 0.5 秒至 64 秒。该计数器还提供一个定时器功能，该功能使用从 **SYSClk** 分频的时钟实现系统定时器功能。系统定时器功能的最大溢出周期是 $\text{SYSClk}/2^{21}$ 。ILRCO 为 **RTC** 模块提供内部时钟源。WDTPS 和 WDT0F 分别来自 **WDT** 预分频器和 **WDT** 溢出，为更长的唤醒时间需求提供了扩展的预分频器时钟源。必须在启用 **RTCE** 之前配置 **RTC** 时钟源。

如果使用 **XTAL** 振荡器作为系统时钟，那么 **RTC** 仍然使用 **P6.0** 输入作为时钟源。**RTCO** 使能 **RTC** 溢出输出到端口引脚上。只有上电复位才会将实时时钟及其相关的 **SFRs** 重置为默认状态。

图 11-1. 实时时钟计数器



11.1. RTC 寄存器

RTCCR: 实时时钟控制寄存器

SFR 页 = P

SFR 地址 = 0x54

POR = 0011-1111

7	6	5	4	3	2	1	0
RTCE	RTCO	RTCRL.5	RTCRL.4	RTCRL.3	RTCRL.2	RTCRL.1	RTCRL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: RTCE, RTC 使能。

0: 停止 RTC 计数器, RTCCT。

1: 使能 RTC 计数器并且当 RTCCT 溢出时置位 RTCF, 当 RTCE 被设置, CPU 不能访问 RTCTM, 只有当 RTCE 被清除后才能访问。

Bit 6: RTCO, RTC 输出使能。RTCKO 输出频率是(RTC 溢出率)/2。

0: 禁止 RTCKO 输出。

1: 使能 RTCKO 输出在 P4.5。

Bit 5~0: RTCRL[5:0], RTC 计数器重载值寄存器。当寄存器被 CPU 访问, 且 RTCCT 溢出时寄存器值会被重载到 RTCCT。

RTCTM: 实时时钟定时器寄存器

SFR 页 = P

SFR 地址 = 0x55

POR = 0111-1111

7	6	5	4	3	2	1	0
RTCCS.1	RTCCS.0	RTCCT.5	RTCCT.4	RTCCT.3	RTCCT.2	RTCCT.1	RTCCT.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: RTCCS.1~0, RTC 时钟选择。缺省值是“01”

RTCCS.3~0	时钟源	RTC 中断周期 如果 XTAL2/ECKI(P60) = 32768Hz	最小周期
0 0 0 0	RTCPS[14] ($/2^{15}$)	1 sec ~ 64 sec	1 sec
0 0 0 1	RTCPS[13] ($/2^{14}$)	0.5 sec ~ 32 sec	0.5 sec (默认)
0 0 1 0	RTCPS[13] ($/2^{13}$)	0.25 sec ~ 16 sec	0.25 sec
.....			
1 0 1 0	RTCPS[4] ($/2^5$)	976 us ~ 62.46 ms	976 us
1 0 1 1	RTCPS[3] ($/2^4$)	488 us ~ 31.23 ms	488 us
1 1 0 0	RTCPS[2] ($/2^3$)	244 us ~ 15.61 ms	244 us
1 1 0 1	RTCPS[1] ($/2^2$)	122 us ~ 7.805 ms	122 us
1 1 1 0	RTCPS[0] ($/2^1$)	61 us ~ 3.903 ms	61 us
1 1 1 1	RTCPSI ($/2^0$)	30.5 us ~ 1.952 ms	30.5 us

Bit 5~0: RTCCT[5:0], RTC 计数器寄存器。通过选择不同的时钟源 RTCCS[1:0]来选择 RTC 功能或系统定时功能。当计数器溢出, 置位 RTCF 标志并且 RTCFIE 使能会产生系统标志中断。最大的 RTC 溢出时间为 64 秒。

CKCON4: 时钟控制寄存器4

SFR 页 = 仅 P 页

SFR 地址 = 0x42

复位值 = 0000-0000

7	6	5	4	3	2	1	0
RCSS2	RCSS1	RCSS0	RPSC2	RPSC1	RPSC0	RTCCS3	RTCCS2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~5: RTC 时钟源选择[2:0]

RCSS2, RCSS1, RCSS0	RTC 时钟源选择
0 0 0	XTAL2/ECKI (P6.0)
0 0 1	ILRCO
0 1 0	WDTPS
0 1 1	WDTOF
1 0 0	SYSCLK
1 0 1	SYSCLK/12
1 1 0	保留
1 1 1	保留

PCON1: 电源控制寄存器1

SFR 页 = 0~F & P

SFR 地址 = 0x97

POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: RTCF, RTC 溢出标志。

0: 这位必须通过软件写“1”清除, 软件写“0”不操作。

1: 当 RTCCT 溢出此位仅仅被硬件置位, 写“1”清除 RTCF。

SFIE: 系统标志中断使能寄存器

SFR 页 = 0~F

SFR 地址 = 0x8E

POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: RTCFIE, 使能 RTCF(PCON1.4)中断。

0: 禁止 RTCF 中断。

1: 使能 RTCF 中断。如果使能。RTCF 能唤醒 CPU 在空闲模式或掉电模式。

12. 系统复位

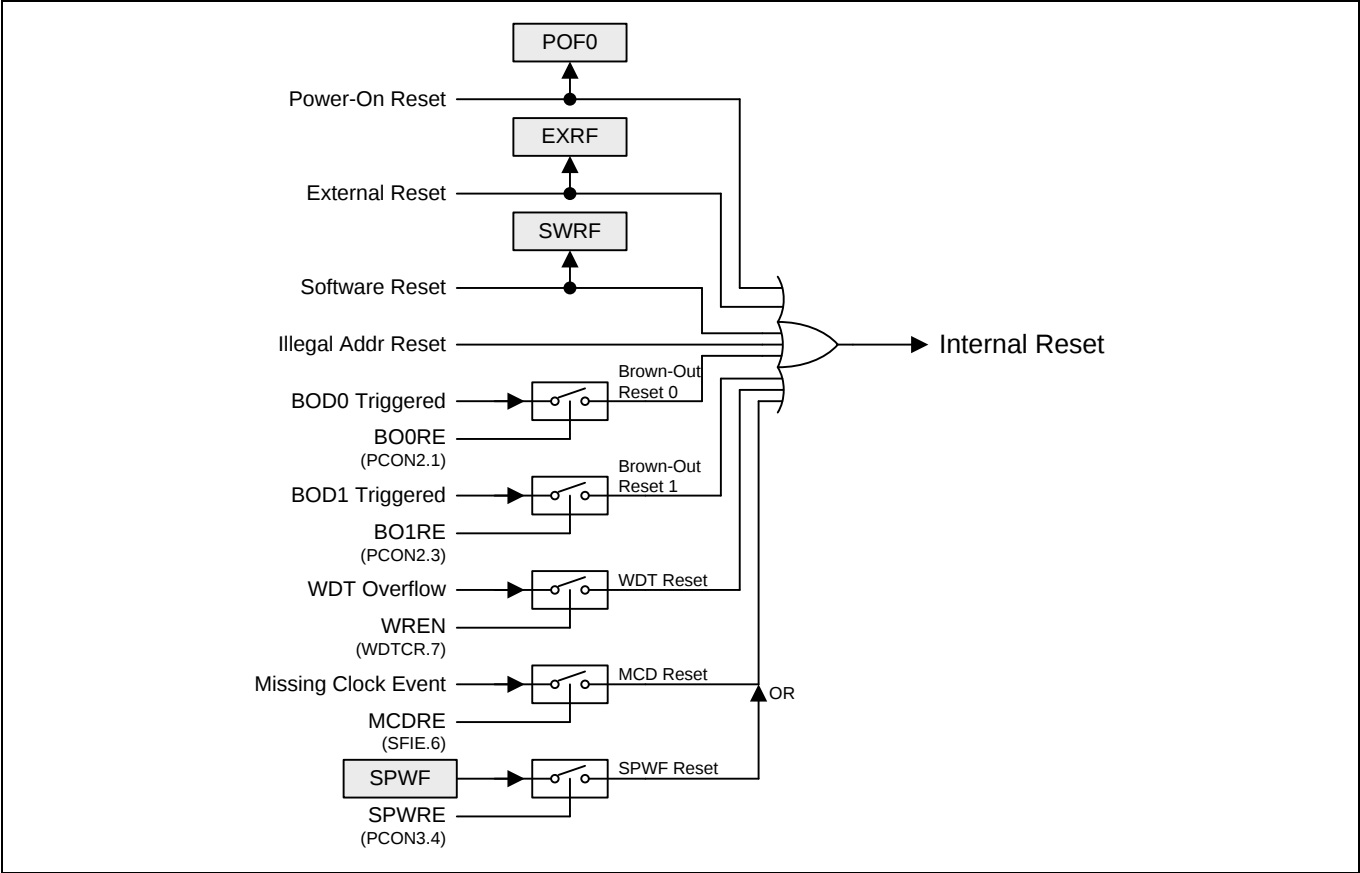
在复位期间，所有 I/O 寄存器都被设置为它们的初始值，程序根据 OR 选项从复位向量 0000H 或 ISP 起始地址开始执行。**MG82F6P32** 有 9 个复位源:上电复位、外部复位、软件复位、非法地址复位、低压监测复位 0、低压监测复位 1、WDT 复位和时钟丢失监测复位。图 12-1 显示了 **MG82F6P32** 的系统复位源。

下面的部分描述了复位产生源和相应的控制寄存器和指示标志位。

12.1. 复位源

图 12-1 介绍了 **MG82F6P32** 的复位系统和它的所有复位来源

图 12-1. 系统复位源



12.2. 上电复位

上电复位(POR)在电源上电期间用于 CPU 内部复位。CPU 在 VDD 电压上升到上电复位电压之前将保持复位状态。VDD 电压降到 POR 电压之下后 CPU 将再次进入复位状态。在一个电源周期中，如果需要再产生一次上电复位 VDD 必须降到 POR 电压之下。

PCON0: 电源控制寄存器 0

SFR 页 = 0~F & P

POR = 0001-0000

SFR 地址 = 0x87

复位值 = 000X-0000

7	6	5	4	3	2	1	0
SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: POF0, 上电复位标志 0。

0: 这标志必须通过软件清零以便认出下一个复位类型。

1: 当 VDD 从 0 伏上升到正常电压时硬件置位。POF0 也有软件置位。

上电标志 POF0 在上电过程中由硬件置“1”或当 VDD 电压降到 V_{POR} 电压之下时由硬件置“1”。它通过软件来清除但不受任何热复位(譬如: 外部 RST 引脚复位、低电压监测复位、软件(ISPCR.5)复位和 WDT 复位)的影响。它帮助用户检测 CPU 是否从上电开始运行。注意: POF0 必须由软件清除。

12.3. 外部复位

复位是通过将复位引脚保持一段时间低电平来完成的, 请参考“37.3 外部复位特性”。为了确保可靠的上电复位, nRST 引脚的硬件复位是必要的。

PCON1: 电源控制寄存器 1

SFR 页 = 0~F & P

POR = 0000-0000

SFR 地址 = 0x97

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 6: EXRF, 外部复位标志。

0: 这位必须通过软件清零, 写“1”清零, 写“0”无效。

1: 若外部复位产生则被硬件置位, 写“1”清零 EXRF。

12.4. 软件复位

软件通过对 SWRST(ISPCR.5)位写“1”触发一个系统热复位, 软件复位后, 硬件置位 SWRF 标志(PCON1.7)。SWBS 标志决定 CPU 是从 ISP 还是 AP 区域开始运行程序。

ISPCR: ISP 控制寄存器

SFR 页 = 0~F

POR = 0000-XXXX

SFR 地址 = 0xE7

7	6	5	4	3	2	1	0
ISPEN	SWBS	SWRST	CFAIL	--	--	--	--
R/W	R/W	R/W	R/W	W	W	W	W

Bit 6: SWBS, 软件执行起始选择控制。

0: 复位软件从 AP 存储区开始执行。

1: 复位软件从 ISP 存储区开始执行。

Bit 5: SWRST, 软件复位触发控制。

0: 写“0”无操作。

1: 写“1”产生软件系统复位, 它将被硬件自动清除。

PCON1: 电源控制寄存器 1

SFR 页 = 0~F & P

SFR 地址 = 0x97

POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SWRF，软件复位标志。
0: 这位必须通过软件清零，写“1”清零，写“0”无操作。
1: 软件复位产生时硬件置位此位，写“1”清零 SWRF。

12.5. 低电压监测复位

MG82F6P32 中，有两个低电压监测器(BOD0 和 BOD1)监测电源电压(VDD)，低电压监测器(BOD0) 的监测固定点为 VDD=1.7V，低电压监测器(BOD1)的监测点可以被软件选择为 VDD=4.2V, 3.6V, 2.4V 或 2.0V，如果 VDD 电压低于 BOD0 或 BOD1 监测点，则置位相关联的 BOF0 和 BOF1 标志，如果 BO0RE (PCON2.1) 被使能，BOD0 事件将触发一个 CPU 复位并置位 BOF0 指示一个低电压监测(BOD0)复位发生；如果 BO1RE (PCON2.3)被使能，BOD1 事件将触发一个 CPU 复位并置位 BOF1 指示一个低电压监测(BOD1)复位发生。

BOD1 复位注意事项

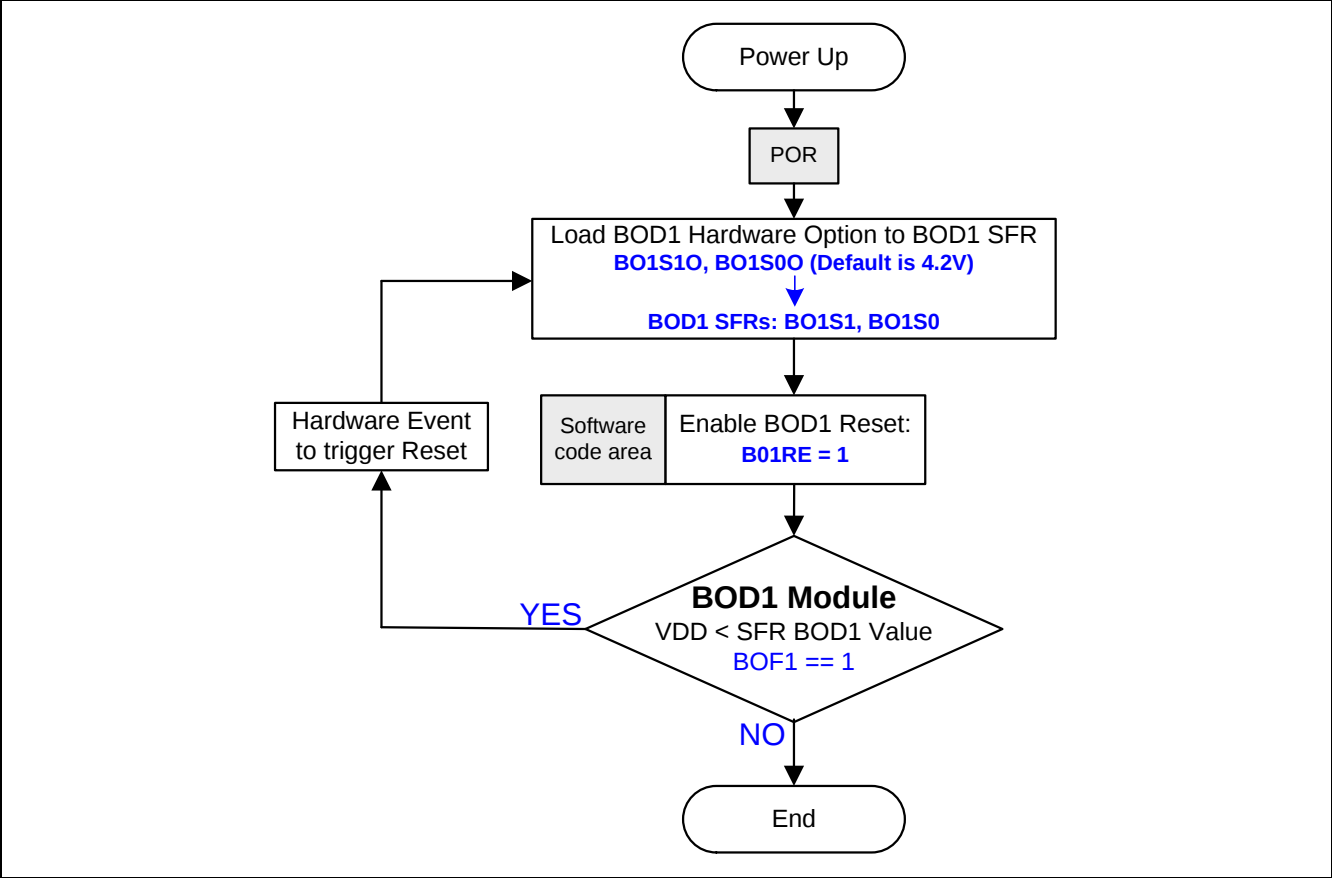
当 BOD1 复位时，它会将 BOD1 硬件选项值((BO1S10, BO1S00, 详情请参考“35 硬件选项” 章节)重新加载到 BOD1 SFR (BO1S1, BO1S0)中。但是，如果硬件选项 BO1REO 也已启用，并且用户在软件中重新设置了较低的 BOD1 级别，当 VDD 低于软件设置的级别时，将触发 BOD1 事件。通常情况下，在 VDD 电压高于软件设定值后，应该释放 BOD1 事件，但在这种情况下，设置了 BO1REO，因此将重新加载 BOD1 硬件选项值，并导致直到 VDD 高于硬件选项值才会释放 BOD1。

有两种方式控制 BOD1 复位:

- 1. 硬件选项:
 - a. 使用硬件选项设置 BOD1 侦测电压
 - b. 设置硬件选项 BO1REO 使能 BOD1 复位。用户也可以保留 BOD1 复位，并在需要时使用软件代码使能它。

图 12-2 展示了 BOD1 复位流程

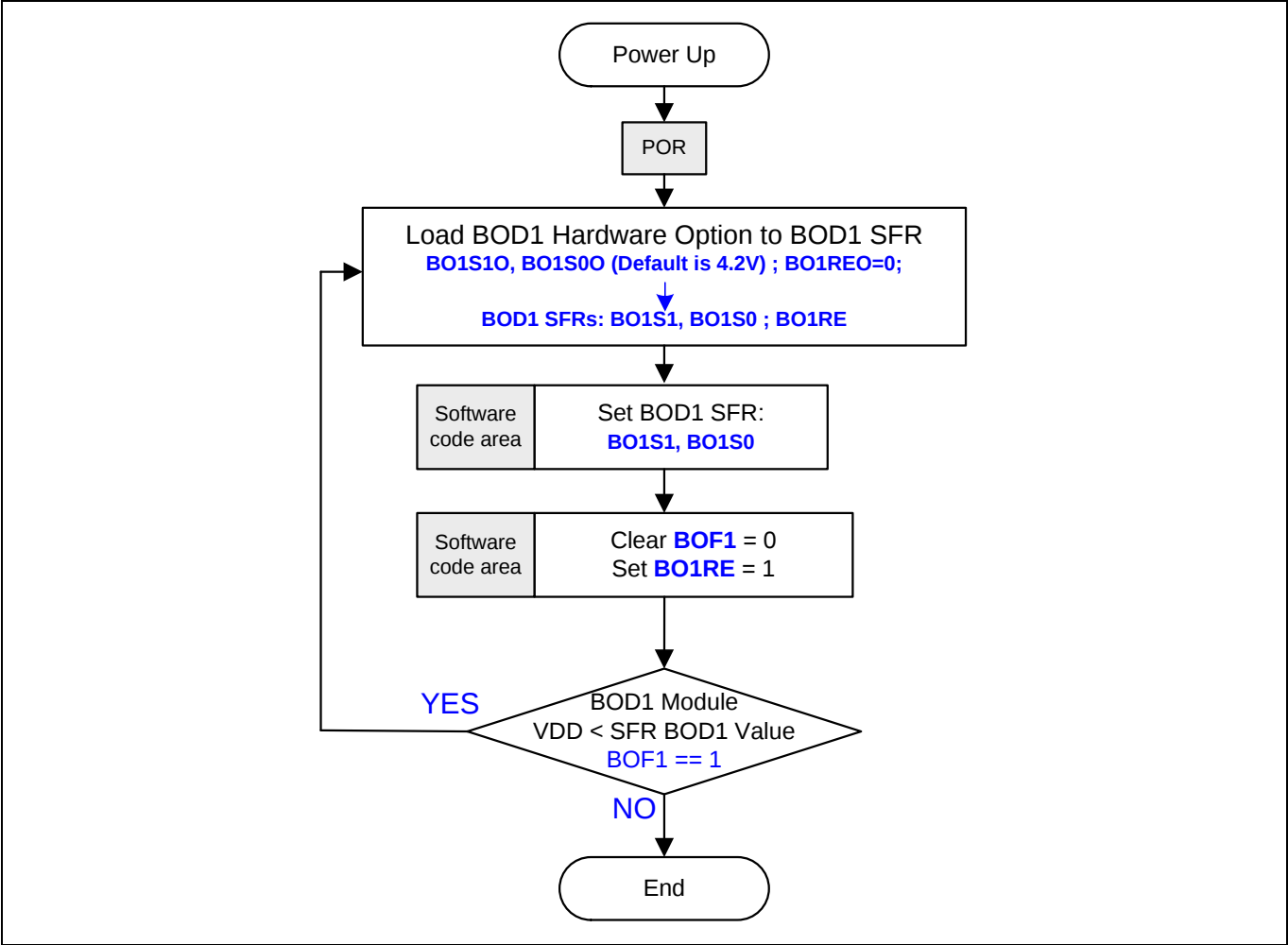
图 12-2. BOD1 复位流程



2. 软件控制 BOD1:
- a. 不使用硬件选项 BO1RE0 使能 BOD1 复位。(默认禁止)
 - b. 通过 BO1S[1:0]设置 BO1 检测电压
 - c. 清 BOD1 标志位, BOF1 = 0
 - d. 使能 BOD1 复位

图 12-3 展示了 BOD1 软件控制流程.

图 12-3. BOD1 侦测软件控制



PCON1: 电源控制寄存器 1

SFR 页 = 0~F & P

SFR 地址 = 0x97

POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 2: BOF1, 低电压监测标志 1。

0: 这位必须由软件写“1”清零。软件写“0”无效

1: 当电源电压触及到低电压监测器 1 电压时，硬件置位此位。在这个位上写“1”将清除 BO1。如果启用了 BO1RE (PCON2.3)，则 BOF1 表示发生了 BOD1 复位。

Bit 1: BOF0, 低电压监测标志 0。

0: 这位必须由软件写“1”清零。软件写“0”无效

1: 当电源电压触及到低电压监测器 0 电压时，硬件置位此位。在这个位上写“1”将清除 BOF0。如果启用了 BO0RE (PCON2.1)，则 BOF0 表示发生了 BOD0 复位。

12.6. WDT 复位

当启用 WDT 启动计数器时，WDTF 将由 WDT 溢出置位。如果 WREN(WDTCR.7)使能，WDT 溢出将触发系统复位，导致 CPU 重新启动。软件可以读取 WDTF 来识别发生了 WDT 复位。

PCON1: 电源控制寄存器 1

SFR 页 = 0~F & P

SFR 地址 = 0x97

POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 0: WDTF, WDT 溢出/复位标志。

0: 这位必须由软件写“1”清零。软件写“0”无效

1: 此位仅在 WDT 溢出时由硬件置位。在这位上写“1”将清零 WDTF。如果 WREN (WDTCR.7)使能，WDTF 表示发生了 WDT 复位。

12.7. MCD 复位

当 XTAL 为 MCU 时钟源时，XTAL 输入信号丢失将置位 MCDF，并触发系统复位。MCD 触发系统复位后，MCU 时钟源切换到 IHRCO。默认情况下，MCD 事件将触发系统复位。如果用户不要此功能，清除 MCDRE 以禁用复位功能。

PCON1: 电源控制寄存器 1

SFR 页 = 0~F & P

SFR 地址 = 0x97

POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: MCDF, 时钟丢失监测标志位

0: 这位必须由软件写“1”清零。软件写“0”无效

1: 此位仅在监测到一个时钟丢失事件后由硬件置位。在此位上写“1”清除 MCDF。时钟丢失监测模块由 MCDFIE 使能。如果 MCDFIE 被清零，时钟丢失监测模块将无效。软件在切换 OSCin 到 XTAL 之前必须清除 MCDF。

SFIE: 系统标志中断使能寄存器

SFR 页 = 0~F

SFR 地址 = 0x8E

POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 6: MCDRE, 使能时钟丢失事件产生系统复位

0: 禁止 MCD 事件触发系统复位。

1: 使能 MCD 事件触发系统复位。

Bit 5: MCDFIE, 使能 MCDF (PCON1.5) 中断。

0: 禁止 MCDF 中断。

1: 使能 MCD 模块并且使能 MCDF 中断。

12.8. 非法地址复位

在 MG82F6P32 中，如果软件程序运行到非法地址，例如超过程序 ROM 限制，将触发对 CPU 的复位。

12.9. 堆栈指针错误复位

SPHB: 堆栈指针高边界

SFR 页 = 仅 P 页

SFR 地址 = 0x53

复位值 = 1111-1111

7	6	5	4	3	2	1	0
1	1	1	1	SPHB.3	SPHB.2	SPHB.1	SPHB.0
R	R	R	R	R/W	R/W	R/W	R/W

SPHB, 用于堆栈指针警告的检测边界

如果 SPHB == 1111-1111, 当 $SP \geq 1111-1111$ 时 SPWM 将被置位.

如果 SPHB == 1111-0000, 当 $SP \geq 1111-0000$ 时 SPWM 将被置位

PCON1: 电源控制寄存器 1

SFR 页 = 0~F & P

SFR 地址 = 0x97

POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: SPWF, SP 警告标志位

0: 这位必须由软件写 “1” 清零。软件写 “0” 无效

1: 此位仅在 $SP \geq SPHB$ 时由 硬件置位。当 $SP < SPHB$ 时，在此位上写 “1” 清除 SPWF。

PCON3: 电源控制寄存器 3

SFR 页 = 仅 P 页

SFR 地址 = 0x45

POR = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	R/W	W	R/W	W	W	W	W

Bit 4: SPWRE, SPWF 触发 MCU 复位

0: 禁止 SPWF 触发 MCU 复位

1: 使能 SPWF 触发 MCU 复位

13. 电源管理

MG82F6P32 支持两个电源监测模块，低电压监测器 0(BOD0)和低电压监测器 1(BOD1)，和 7 种电源节能模式：空闲模式(IDLE)、掉电模式(Power-Down)、慢频模式、副频模式、RTC 模式、Watch 模式和 Monitor 模式。

BOD0 和 BOD1 通过 BOF0 和 BOF1 标志位报告电源状态，软件可以通过这个状态产生中断或复位。7 种电源节能模式提供不同的节能应用，通过对 CKCON0、CKCON2、CKCON3、CKCON4、CKCON5、PCON0、PCON1、PCON2、PCON3、RTCCR 和 WDTTCR 寄存器的访问来操作这些电源事件。

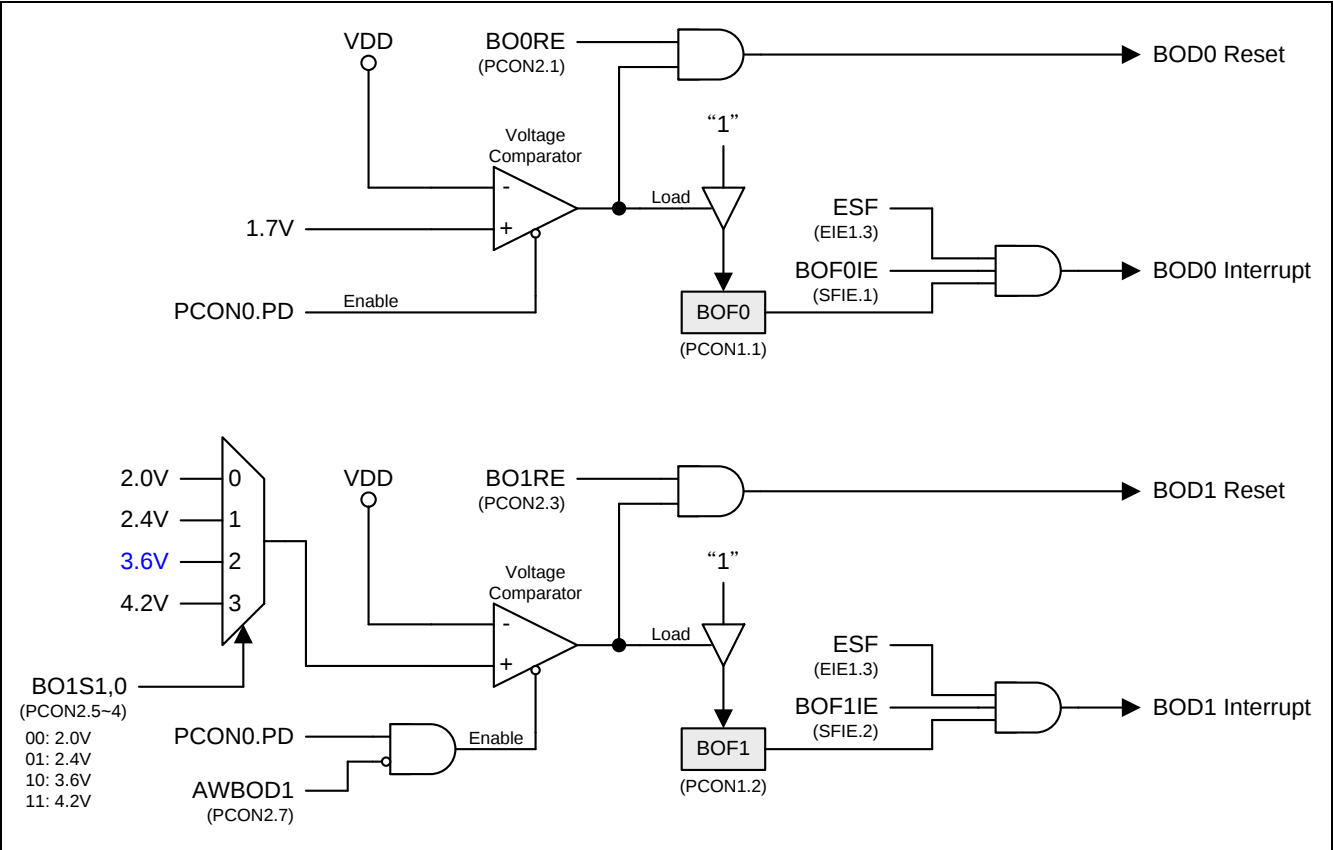
13.1. 低电压监测器

MG82F6P32 有两个低电压监测器 (BOD0& BOD1)来监测芯片电压，图 13-1 是 BOD0 和 BOD1 功能方框图，BOD0 监测固定触发电压为 VDD=1.7V，BOD1 监测 VDD 并且与软件选择的电压(4.2V/3.6V/2.4V/2.0V)做比较。当 VDD 降到触发电压以下时，BOF0 (PCON1.1)标志被置位，如果 ESF (EIE1.3)和 BOF0IE (SFIE.1)被使能，不管是普通模式或空闲模式都能产生一个中断请求以响应(BOD0)事件，BOD1 有同样的标志 BOF1，也有同样的中断功能。如果 AWBOD1 (PCON2.7)使能，这个中断(BOD1)也能唤醒掉电模式。

当 BO0RE (PCON2.1)被使能，BOD0 事件产生一个系统复位并硬件置位 BOF0 指示一个 BOD0 复位事件已经产生。在普通模式和空闲模式下 BOD0 事件能重新启动 CPU。设置相关的控制位 BO1RE(PCON2.3)，BOD1 也有同样的复位功能。如果 AWBOD1 (PCON2.7)位被使能，BOD1 也能重新启动掉电模式。

如果 BOD1 在应用中没有使用，为了节省功耗可以通过软件清除 EBOD1 (PCON2.2)来禁止 BOD1。

图 13-1. 低电压监测器 0/1



13.2. 省电模式

13.2.1. 慢频模式

程序设置 SCKS2~SCKS0(CKCON0 寄存器,参考章节“[9 系统时钟](#)”)为非 0/0/0 值,可以降低 MCU 的工作速度达到节能的目的。用户应当考虑在特殊的程序段使用合适的低速。原则上不应该影响系统的其他功能。而且,应该在普通的程序段恢复到正常的速度

13.2.2. 副频模式

设置 OSCS1~0 选择 ILRCO 作为系统时钟, MCU 的工作频率会慢下来。32KHz ILRCO 系统频率使 MCU 工作在特别慢的速度和功耗下。另外设置 SCKS2~SCKS0 位(CKCON0 寄存器, 参考章节“[9 系统时钟](#)”)用户可以使 MCU 的速度最低到 250Hz。

13.2.3. RTC 模式

MG82F6P32 有一个简单的 RTC 模块允许用户在设备部分掉电时继续运行准确的定时器。在 RTC 模式, RTC 模块作为一个“时钟”功能并且能在 RTC 溢出时唤醒芯片的掉电模式。详细描述请参考章节“[11 实时时钟\(RTC\)/系统时钟](#)”。

13.2.4. Watch 模式

如果看门狗被使能并且位 NSW 被设置, 看门狗在掉电模式保持运行支持自动唤醒功能, 这个在 **MG82F6P32** 应用中叫 Watch 模式。当 WDT 溢出, 软件选择中断或系统复位来唤醒 CPU 并硬件置位 WDTF。通过定义 WDT 预分频最大唤醒时间能到 2 秒, 更详细信息请参考章节“[10 看门狗定时器\(WDT\)](#)”和章节“[15 中断](#)”。

13.2.5. Monitor 模式

如果 AWBOD1(PCON2.3)被设置, BOD1 即使在掉电模式下, 低电压监测功能 BOD1 会有效, 这就是 **MG82F6P32** 应用中的 Monitor 模式。当 BOD1 触发到监测电压, 软件选择中断或系统复位来唤醒 CPU 并硬件置位 BOF1, 更详细信息请参考章节“[13 电源管理](#)”和章节“[15 中断](#)”。

13.2.6. Idle 模式

可以通过软件的方式置位 PCON.IDL 位, 使设备进入空闲模式。在空闲模式下, 系统不会给 CPU 提供时钟 CPU 状态、RAM、SP、PC、PSW、ACC 被保护起来。I/O 端口也保持当前的逻辑状态。空闲模式保持外部设置当有中断来时能唤醒 CPU, 空闲模式下定时器 0、定时器 1、定时器 2、定时器 3、DMA、SPI、KBI、ADC、AC0、AC1、OPA0、OPA1、PGA、S0~S1、TWI0/I2C0、TWI1/I2C1、RTC、MCD、BOD0 和 BOD1 仍然处于工作状态。在空闲模式下 PCA 和 WDT 唤醒 CPU 有条件制约。任何使能的中断源或复位都能终止空闲模式, 一个中断会退出空闲模式, 并同时进入中断服务程序, 只有在中断返回后才会开始执行进入空闲模式指令之后的程序。

当 MCU 在空闲模式和掉电模式时为了降低功耗 ADC, OPA0/1, PGA 或模拟比较器输入通道必须设置为“**仅模拟输入**”。

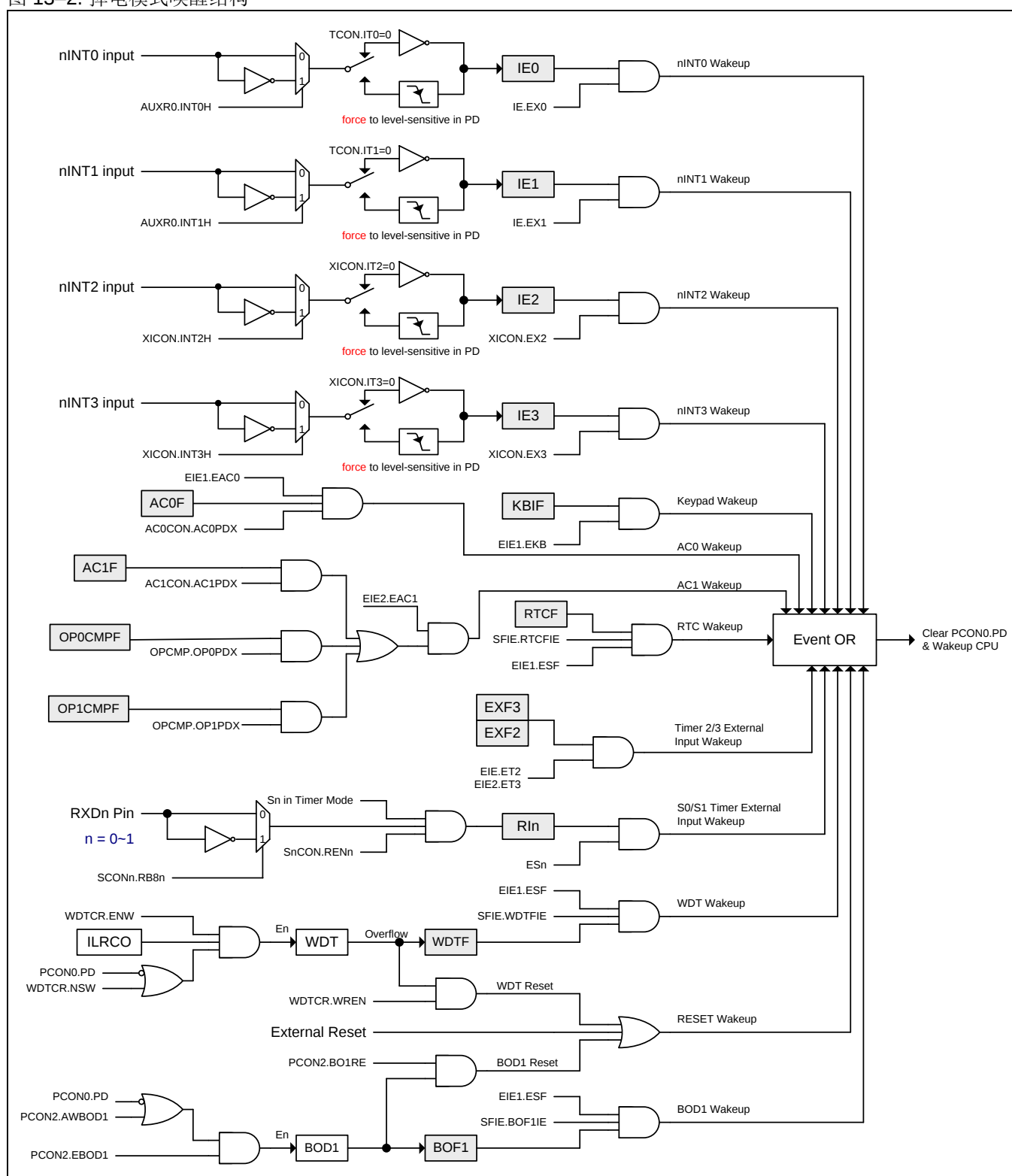
13.2.7. 掉电模式

可以通过软件的方法置位 PCON0.PD 使设备进入掉电模式。掉电模式下, 震荡停止, Flash 存储器掉电以节约电能。只有上电电路继续刷新电源。在掉电模式下, 电源电压可能会降低到 RAM 保持电压, RAM 内容将被保留; 然而, 一旦 VDD 降低, SFR 内容就不能得到保证。外部复位、上电复位、使能的外部中断、使能的 AC0~1、使能的 OPA0~1 ACMP、使能的 KBI、使能的 RTC (RTC 模式)、使能的 BOD1(monitor 模式)或使能的不停止的 WDT 都能使系统退出掉电模式。

如果有下列情况发生, 使用者至少要等 4us 后才能进入或再次进入掉电模式: 刚开始运行代码(任何形式的复位后面), 或者刚刚退出掉电模式。为了在掉电模式达到最小功耗, 软件必须设置所有的 I/O 为非悬浮状态。

图 13-2 展示了 **MG82F6P32** 在掉电模式下的唤醒机制。

图 13-2. 掉电模式唤醒结构



13.2.8. 中断唤醒掉电模式

4 个外部中断都能终止掉电模式，外部中断 nINT0、nINT1、nINT2 和 nINT3 能退出掉电模式。为了能唤醒掉电模式，中断 nINT0、nINT1、nINT2 或 nINT3 必须使能并且设置为电平触发操作，如果外部中断使能且设置是边沿触发(上升或下降)，他们会被硬件强置为电平触发(低电平或高电平)。

一个中断终止掉电模式，唤醒时间取决内部定时。当中断口产生下降沿时，掉电模式被终止，震荡重新启动，并且一个内部计数器开始计数，在内部计数器没有计满之前内部时钟不允许被应用 CPU 也不能运行指令。计数溢出后，中断服务程序开始工作，为了避免中断被重复触发，中断服务程序在返回前应该被禁止，中断口低电平应保持足够长的时间以等待系统问题。

13.2.9. 复位唤醒掉电模式

外部复位唤醒掉电模式有点类似于中断。复位脚有下降时系统退出掉电模式，震荡重新启动，且一个内部计数器开始计数。在内部计数器没有溢出之前内部时钟不允许被应用 CPU 也不能运行指令。复位脚必须保持长时间的低电平以保证系统完全复位，复位脚变高电平时开始执行程序。

值得注意的是当空闲模式被硬件复位唤醒时，前两个机器周期(内部复位没有取得控制权)，程序从进入 IDLE 模式的最后一条指令执行。这时内部硬件是禁止访问内部 RAM 的，但访问 I/O 端口没有被禁止。为了保证不可预料的写 I/O 口，在进入 IDLE 指令后不要放置写 I/O 口或外部存储器的指令。

13.2.10. KBI 键盘唤醒掉电模式

MG82F6P32 中 KBI.7~0 具有键盘中断唤醒功能，通过使能 KBI 模块的控制寄存器。软件可以设置不同的端口引脚作为 KBI 输入。更详细的 AUXR1 信息请参考章节“[34 辅助特殊功能寄存器](#)”。

通过使能 KBI 唤醒掉电模式有点类似中断唤醒。在 KBI 模式下且已经使能 KBI 中断(EIE1.5, EKB)，系统退出掉电模式，震荡重新启动，且一个内部计数器开始计数。在内部计数器没有计满之前内部时钟不允许被应用 CPU 也不能运行指令。计数溢出后，CPU 会响应 KBI 中断并执行中断服务程序。

13.3. 电源控制寄存器

PCON0: 电源控制寄存器 0

SFR 页 = 0~F & P

POR = 0001-0000

SFR 地址 = 0x87

复位值 = 000X-0000

7	6	5	4	3	2	1	0
SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: POF0, 上电标志 0。

0: 这位必须由软件写“1”清零。

1: 当上电复位产生时硬件置位此位。

Bit 1: PD, 掉电控制位。

0: CPU 清零或任何一个退出掉电模式的事件发生时硬件清零。

1: 置位则激活掉电操作。

Bit 0: IDL, 空闲模式控制位。

0: CPU 清零或任何一个退出空闲模式的事件发生时硬件清零。

1: 置位则激活空闲操作。

PCON1: 电源控制寄存器 1

SFR 页 = 0~F & P

POR = 0000-0000

SFR 地址 = 0x97

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SWRF, 软件复位标志。

0: 这位必须由软件写“1”清零。

1: 当软件复位产生时硬件置位此位。

Bit 6: EXRF, 外部复位标志。

0: 这位必须由软件写“1”清零。

1: 当外部复位产生时硬件置位此位。

Bit 5: MCDF, 时钟丢失监测标志位

0: 这位必须由软件写“1”清零。软件写“0”无效

1: 此位仅在监测到一个时钟丢失事件后由硬件置位。在此位上写“1”清除 MCDF。时钟丢失监测模块由 MCDFIE 使能。如果 MCDFIE 被清零, 时钟丢失监测模块将无效。软件在切换 OSCin 到 XTAL 之前必须清除 MCDF。

Bit 4: RTCF, RTC 溢出标志。

0: 这位必须由软件写“1”清零。写“0”无操作。

1: 当 RTCCT 溢出时此位仅由硬件置位。写“1”则清除 RTCF。

Bit 3: SPWF, SP 警告标志位。

0: 这位必须由软件写“1”清零。软件写“0”无效

1: 此位仅在 $SP \geq SPHB$ 由硬件置位。当 $SP < SPHB$ 时, 写“1”清除 SPWF。

Bit 2: BOF1, 低电压监测标志 1。

0: 这位必须由软件写“1”清零。

1: 当电源电压触及到低电压监测器 1 电压(4.2V/3.7/2.4/2.0)时, 硬件置位此位

Bit 1: BOF0, 低电压监测标志 0。

0: 这位必须由软件写“1”清零。

1: 当电源电压触及到低电压监测器 0 电压(1.7V)时, 硬件置位此位。

Bit 0: WDTF, WDT 溢出标志。

0: 这位必须由软件写“1”清零。

1: 当 WDT 溢出产生时硬件置位此位。

PCON2: 电源控制寄存器 2

SFR 页 = 仅 P 页

SFR 地址 = 0x44

POR = 0000-0101

7	6	5	4	3	2	1	0
AWBOD1	0	BO1S1	BO1S0	BO1RE	EBOD1	BO0RE	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: AWBOD1, 掉电模式(PD)下 BOD1 的唤醒。

0: 掉电模式(PD)下禁止 BOD1。

1: 掉电模式(PD)下保持 BOD1。

Bit 6: 保留。写该寄存器时, 此位必须写“0”。

Bit 5~4: BO1S[1:0], 低电压监测器 1 监测电压选择

BO1S[1:0]	BOD1 监测电压
0 0	2.0V
0 1	2.4V
1 0	3.6V
1 1	4.2V

Bit 3: BO1RE, BOD1 复位使能。

0: 当 BOF1 已经设置, 禁止低电压监测 1(BOD1)系统复位。

1: 当 BOF1 已经设置, 使能低电压监测 1(BOD1)系统复位。

Bit 2: EBOD1, 使能 BOD1 监测 VDD 下降到 BO1S1~0 设置的固定值。

0: 禁止 BOD1 监测电源电压降低芯片功耗。

1: 使能 BOD1 监测电源电压 VDD。

Bit 1: BO0RE, BOD0 复位使能。

0: 当 BOF0 已经设置, 禁止低电压监测 0(BOD0)系统复位。

1: 当 BOF0 已经设置, 使能低电压监测 0(BOD0)系统复位(VDD 触到 1.7V)。

Bit 0: 保留。写此寄存器时, 此位必须写“1”。

PCON3: 电源控制寄存器 3

SFR 页 = 仅 P 页

SFR 地址 = 0x45

POR = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: IVREN, 内部参考电压使能。

0: 禁止片内 IVR (1.4V)。

1: 使能片内 IVR (1.4V)。

Bit 6: IVREN, 掉电模式下 IVR 使能。

0: 掉电模式下禁止 IVR。

1: 掉电模式下使能 IVR。

Bit 5: 保留。写该寄存器时, 此位必须写“0”。

Bit 4: SPWRE, SPWF 触发 MCU 复位

0: 禁止 SPWF 触发 MCU 复位

1: 使能 SPWF 触发 MCU 复位

Bit 3~0: 保留。写该寄存器时, 此位必须写“0”。

14. I/O 口配置(GPIO)

MG82F6D64/32 有下列 I/O 端口: P0.1, P0.2, P0.5, P0.7, P1.0~P1.7, P2.2~P2.4, P2.6, P3.0~P3.5, P4.0~P4.1, P4.4, P4.5, P4.7 和 P6.0~P6.1。如果使能外部晶振为系统时钟或 RTC 输入, 端口 P6.0 和 P6.1 被配置成 XTAL2 和 XTAL1。如果禁止外部复位脚功能, P4.7 是有效的。准确的可用 I/O 引脚数量由封装类型决定。见表 14-1。

The **MG82F6P32** has following I/O ports: P0.1, P0.2, P0.5, P0.7, P1.0~P1.7, P2.2~P2.4, P2.6, P3.0~P3.5,

表 14-1. 可用 I/O 引脚数量

封装类型	I/O 引脚	I/O 端口数量
32-pin	P0.1, P0.2, P0.5, P0.7, P1.0~P1.7, P2.2~P2.4, P2.6, P3.0~P3.5, P4.0~P4.1, P4.4~P4.5, P4.7(nRST), P6.0(ECKI/XTAL2), P6.1(XTAL1)	29 or 28 (nRST selected) or 27 (nRST & ECKI selected) or 26 (nRST & XTAL selected)
28-pin	P0.1, P0.2, P1.0~P1.7, P2.2, P2.4, P2.6, P3.0~P3.1, P3.3~P3.5, P4.0~P4.1, P4.4~P4.5, P4.7(nRST), P6.0(ECKI/XTAL2), P6.1(XTAL1)	25 or 24 (nRST selected) or 23 (nRST & ECKI selected) or 22 (nRST & XTAL selected)

14.1. IO 结构

MG82F6P32 输入输出分成两个配置类型。第一类仅仅是端口 3 有的功能, 准双向口(标准 8051 的 I/O 端口)、推挽输出、仅输入(高阻抗输入)和集电极开漏输出。但是端口 3 和大多数其他 I/O 端口的默认模式是模拟 I/O 模式, 除了 OCD 接口(P4.4/P4.5)和复位引脚。OCD 接口的默认模式为上拉电阻开漏模式。

其它口属于第二类, 这些口可以编程为四种模式分别是模拟 I/O、带上拉电阻的开漏输出、开漏输出和推挽输出。默认设置是模拟 I/O, 也就意味着带有高阻状态的输入模式。

下面章节描述所有类型的 I/O 模式的配置

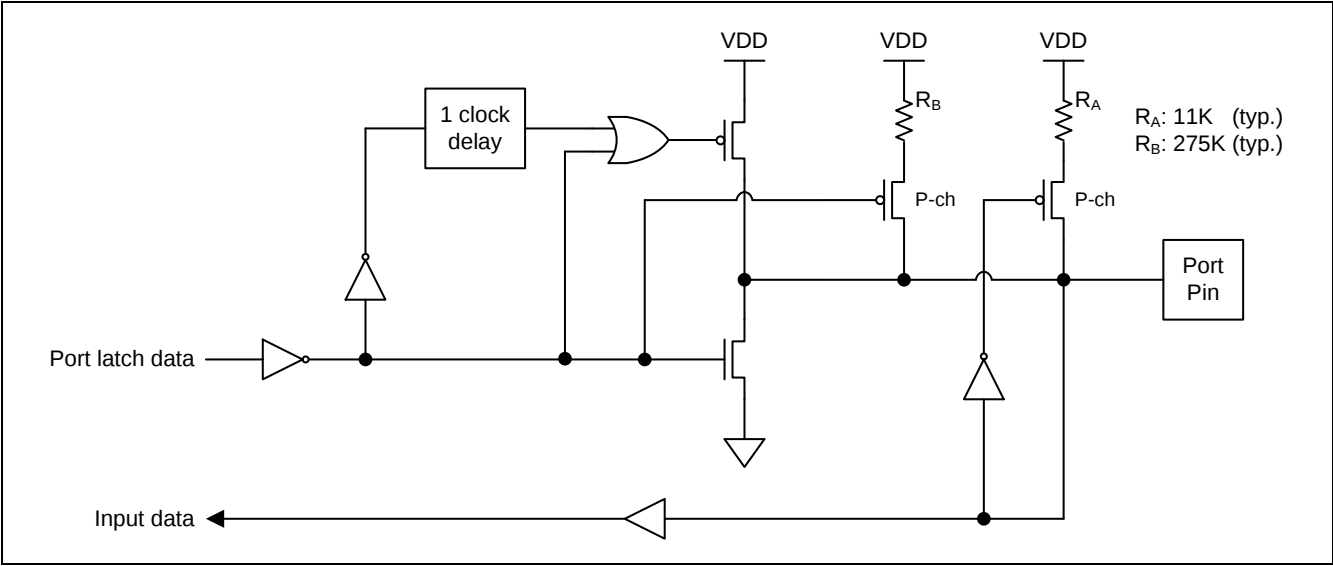
14.1.1. 端口 3 准双向口结构

端口 3 引脚工作在准双向模式时与标准 8051 端口引脚类似。一个准双向端口用作输入和输出时不需要对端口重新配置。这是因为端口输出逻辑高时, 弱上拉, 允许外部器件拉低引脚。当输出低时, 强的驱动能力可吸收大电流。在准双向输出时有三个上拉晶体管用于不同的目的。

其中的一种上拉, 称为微上拉, 只要端口寄存器的引脚包含逻辑 1 则打开。如果引脚悬空, 则这种非常弱上拉提供一个非常小的电流将引脚拉高。第二种上拉称为“弱上拉”, 端口寄存器的引脚包含逻辑 1 时且引脚自身也在逻辑电平时打开。这种上拉对准双向引脚提供主要的电流源输出为 1。如果引脚被外部器件拉低, 这个弱上拉关闭, 只剩一个微上拉。为了在这种条件下将引脚拉低, 外部器件不得不吸收超过弱上拉功率的电流, 且拉低引脚在输入的极限电压之下。第三种上拉称为“强”上拉。这种上拉用于加速准双向端口的上升沿跳变, 当端口寄存器发生从逻辑 0 到逻辑 1 跳变时, 强上拉打开一个 CPU 时钟, 快速将端口引脚拉高。

端口 3 准双向口结构如图 14-1.

图 14-1. 端口 3 准双向 I/O

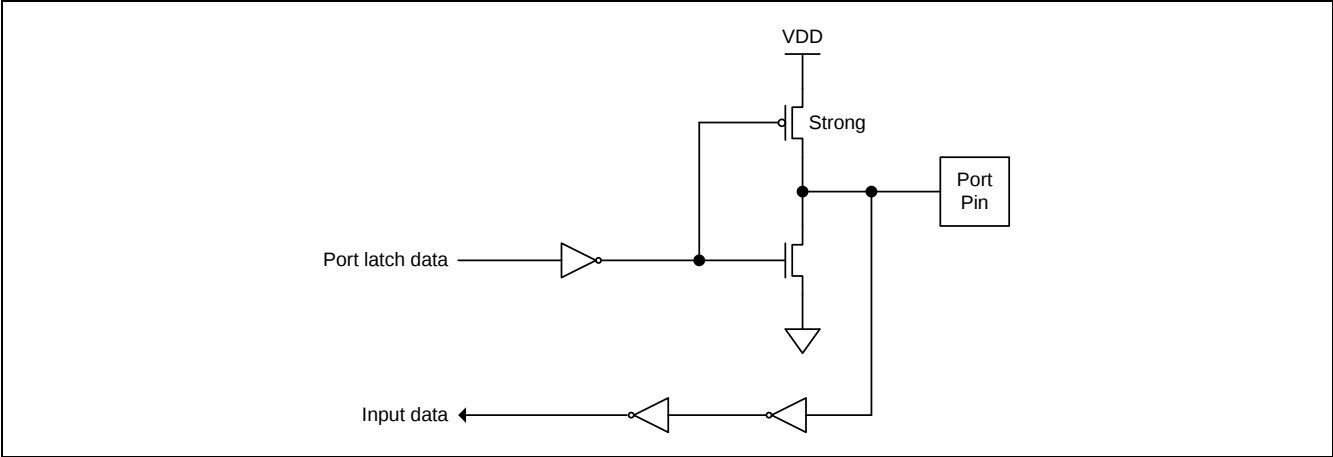


14.1.2. 端口 3 推挽输出结构

端口 3 推挽输出配置与开漏输出、准双向输出模式有着相同的下拉结构，但是当端口寄存器包含逻辑 1 时提供一个连续的强上拉。当一个端口输出需要更大的电流时可配置为推挽输出模式。另外，在这种配置下端口的输入路径与准双向模式相同。

推挽输出结构如图 14-2.

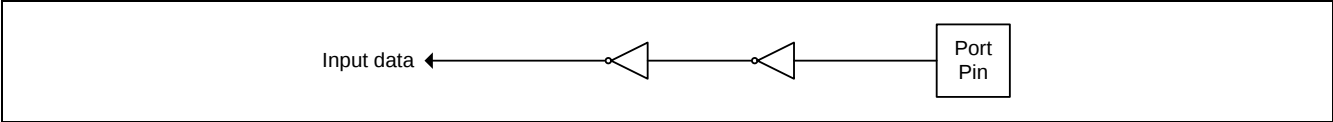
图 14-2. 端口 3 推挽输出



14.1.3. 端口 3 仅输入(高阻抗输入)结构

仅输入配置在端口 3 引脚上没有任何上拉电阻，如 图 14-3.

图 14-3. 端口 3 仅输入

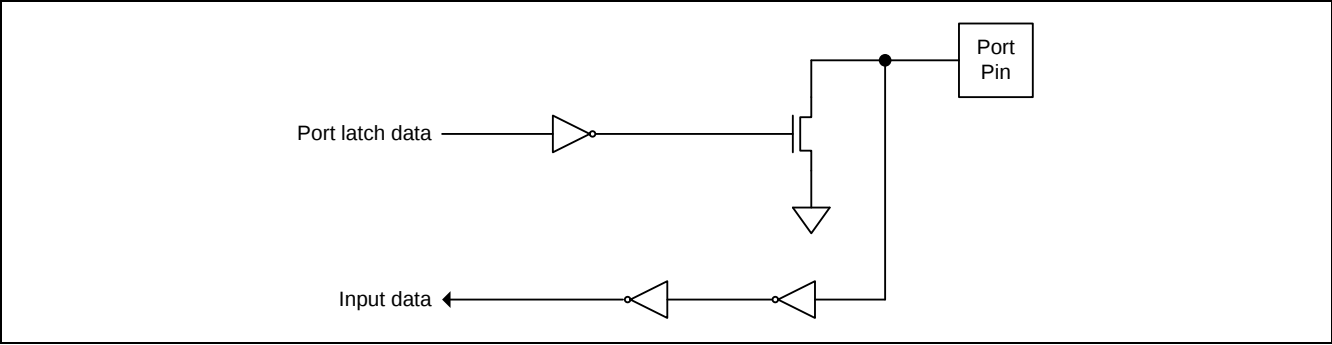


14.1.4. 端口 3 开漏输出结构

端口 3 配置为开漏输出时，当端口寄存器包含逻辑 0 时，关闭所有上拉，只有端口引脚的下拉晶体管。在应用中使用这个配置，端口引脚必须有外部上拉，典型的是将电阻接到 VDD。这个模式的下拉和准双向端口的模式相同。另外，在这种配置下端口的输入路径与准双向模式相同。

端口 3 开漏输出结构如 图 14-4.

图 14-4. 端口 3 开漏输出

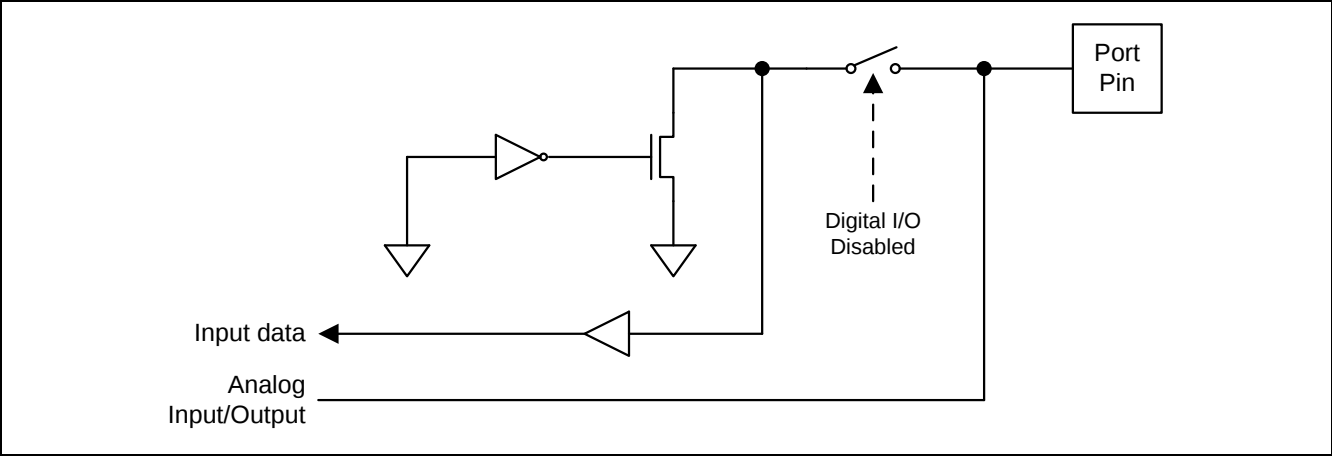


14.1.5. 通用模拟输入输出结构

在通用端口引脚上模拟输入输出结构是默认设置，包括端口 0, 1, 2, 3, 4 和 6，但不包括 OCD_ICE(P4.4 和 P4.5)接口和 nRST。对于 ADC,模拟比较器,OPA 和 PGA 的输入输出应用，用户可以保持这种结构的端口设置。如果应用在数字功能的端口引脚，用户必须把端口引脚配置成相应的结构。在模拟 I/O 模式下，读这些 I/O SFR，它将始终为 0，因为数字输入路径被禁用，并且它将绑定到低电平以防止内部浮动状态。

通用模拟输入输出结构如.

图 14-5. 带上拉电阻的通用开漏输出

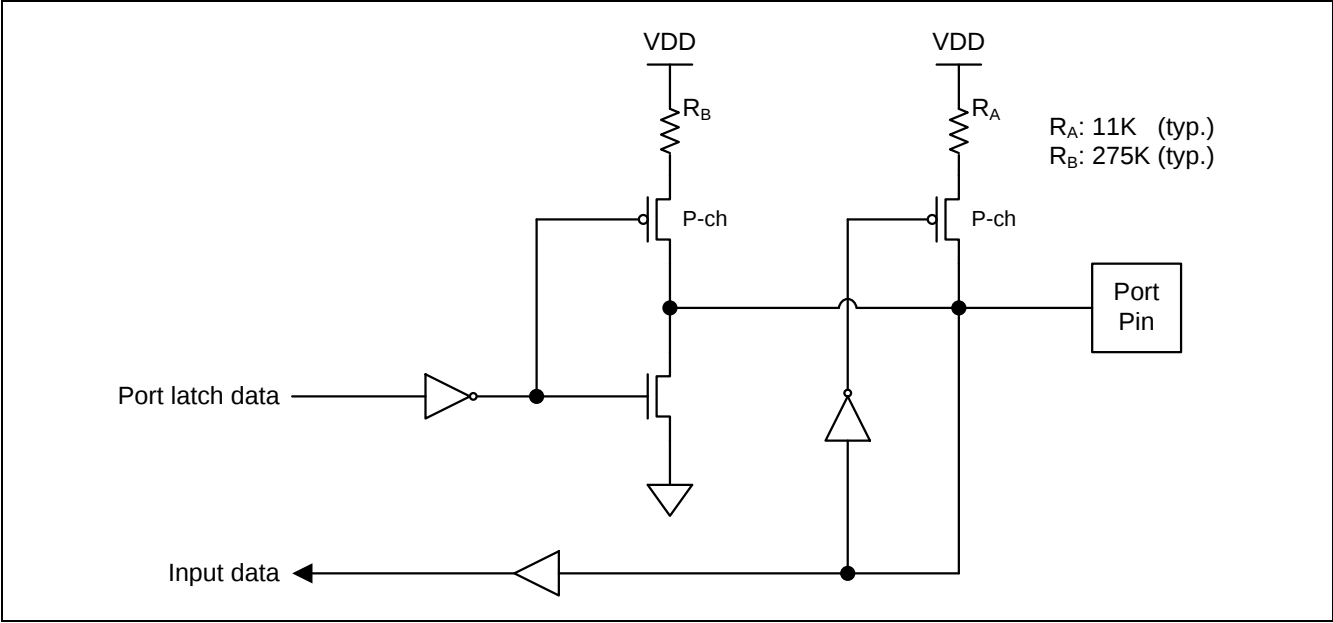


14.1.6. 带上拉电阻的通用开漏输出

带上拉电阻的开漏输出结构是在通用端口引脚上使能开漏输出模式的片内上拉电阻。

带上拉电阻的开漏输出端口结构如 图 14-6.

图 14-6. 带上拉电阻的通用开漏输出

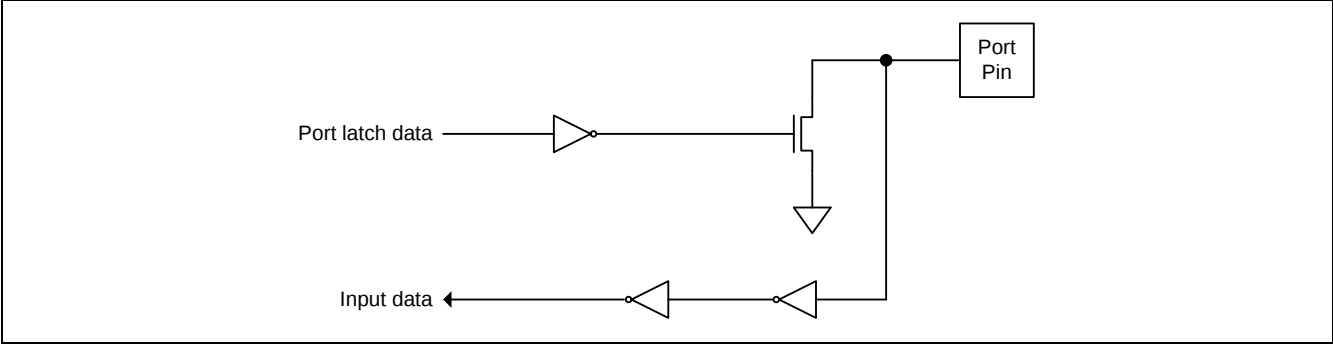


14.1.7. 通用开漏输出结构

通用开漏输出结构和端口 3 开漏输出模式一样。

通用开漏输出结构如 图 14-7.

图 14-7. 通用开漏输出



14.1.8. 通用端口的数字输入结构

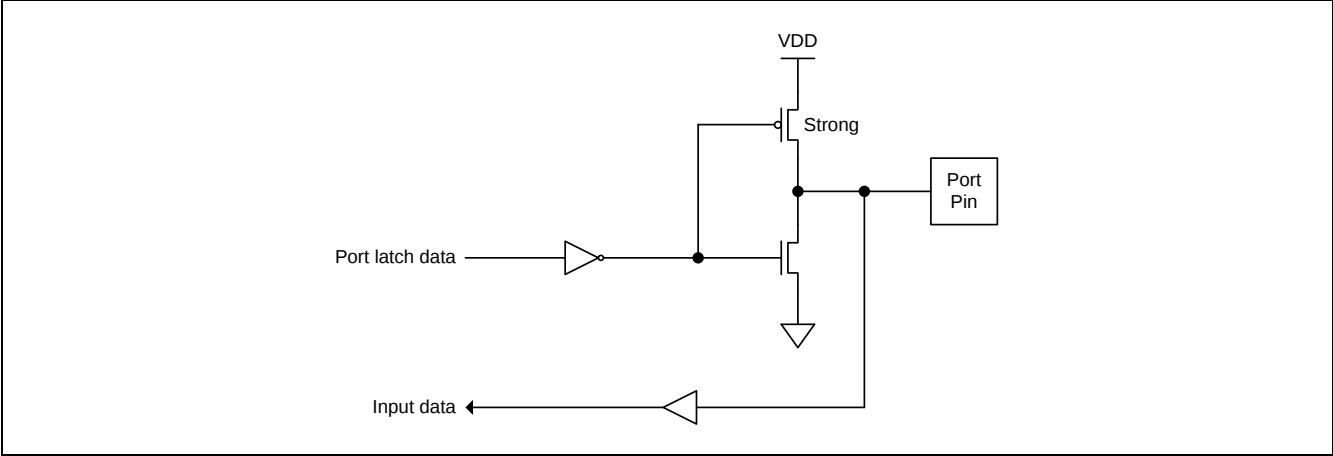
通过设置输出模式为“开漏”并且写逻辑“1”到端口数字寄存器的相应位来配置端口引脚为数字输入。例如，通过设置 $P1M0.0=0$ 、 $P1M1.0=0$ 并且 $P1.0=1$ ，这样 $P1.0$ 配置为数字输入。

14.1.9. 通用推挽输出结构

通用端口引脚上的推挽输出结构跟端口 3 的推挽输出模式一样的功能。

通用端口推挽输出结构如图 14-8.

图 14-8. 通用推挽输出



14.1.10. GPIO 读和写-- 端口锁存器和端口状态

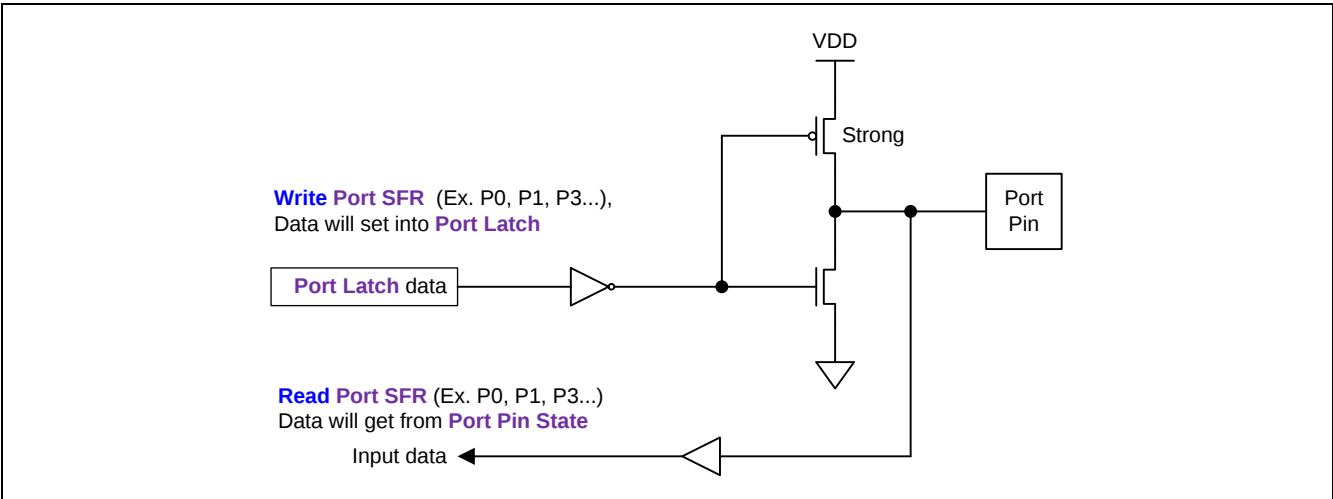
当读或写 I/O 端口 SFR 时，它将通过不同的路径，请参考图 14-9.

写数据到端口 SFR (例如, P0, P1...)，它将设置到端口锁存器

从端口状态读取数据，即当端口引脚上的电压高于 V_{IH} 时，回读数据为“1”，当端口引脚上的电压低于 V_{IL} 时，回读数据为“0”。但是回读数据不会影响端口锁存值。

例如，如果 I/O 口模式是开漏，并且写“1”到端口锁存器，但是端口引脚电压被拉低到“0”。那么回读值将为“0”，但端口锁存器保持“1”。

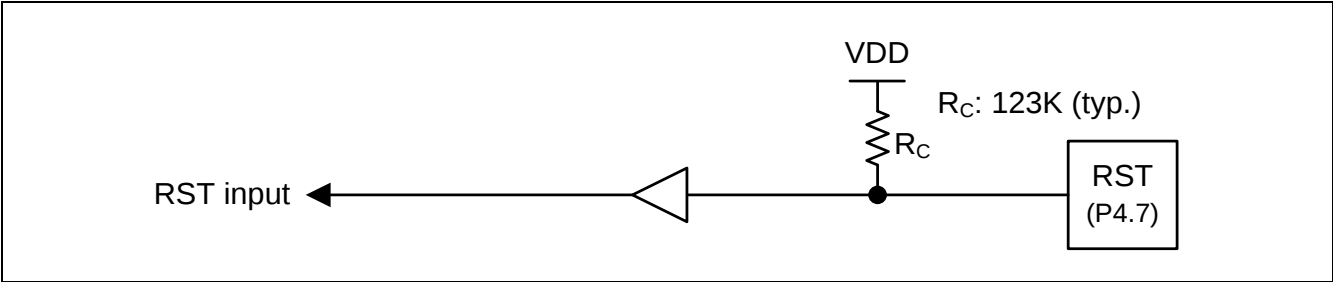
图 14-9. GPIO 读和写路径



14.1.11. 复位引脚结构

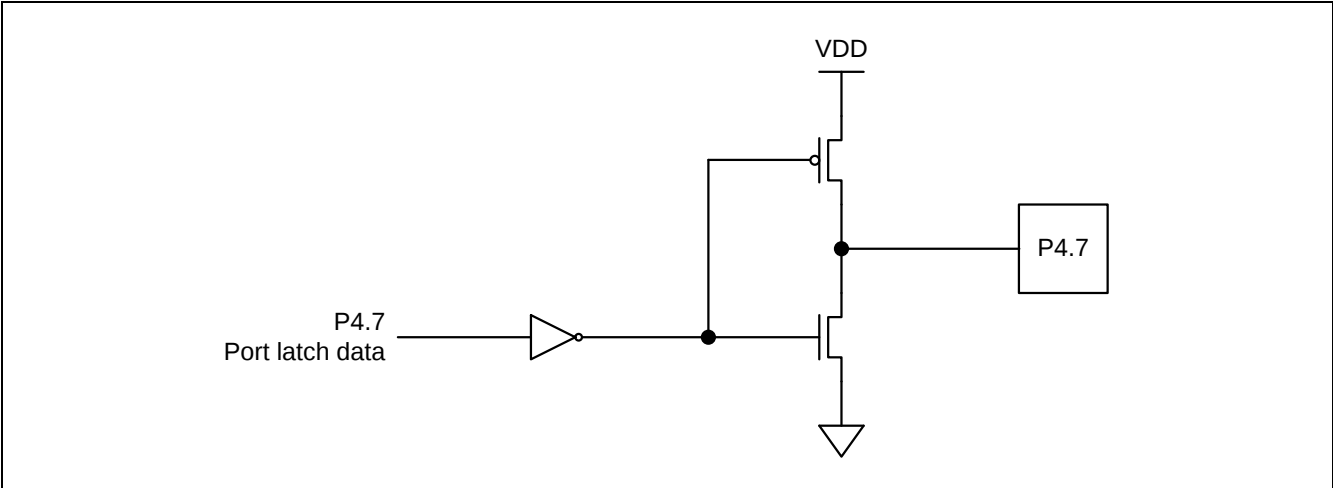
复位引脚用于外部复位触发电路。其结构如图 14-10.

图 14-10. 复位引脚输入结构



复位引脚也可以设置成输出引脚，结构如图 14-11

图 14-11. 复位引脚编程为推挽输出



14.1.12. 端口引脚输出驱动力选择

MG82F6P32 输入/输出有两种驱动力可选适合不同应用的输出阻抗。详情请参考章节“14.2.7 端口输出驱动力控制寄存器”。

14.1.13. 端口引脚输出快速驱动选择

MG82F6P32 输入/输出有两种驱动速度可选适合不同应用的输出频率。详情请参考章“14.2.8 端口输出快速驱动控制寄存器”。

14.2. I/O 口寄存器

MG82F6P32 所有的端口可通过软件个别的、独立的配置其工作模式。端口 3 有 5 种工作模式，如表 14-2。三个模式寄存器用于选择每个端口 3 引脚的类型。

表 14-2. 端口 3 配置设定

P3M2.y	P3M1.y	P3M0.y	端口模式
0	0	0	准双向
0	0	1	仅输入(高阻输入)
0	1	0	推挽输出
0	1	1	开漏输出
1	0	0	模拟 I/O(模拟)
1	其它		保留

这里 y=0~7 (端口引脚号)。寄存器 P3M0 和 P3M1 列举了每个引脚的描述。

其它的通用口引脚有四种模式见

表 14-3..。二个模式寄存器位选择每个引脚的输出类型，且系统复位之后这些端口引脚为仅模拟输入。

表 14-3.通用端口配置设定

PxM1.y	PxM0.y	端口模式
0	0	开漏输出/ 通用数字输入(端口引脚设置为“1”)
0	1	推挽输出
1	0	模拟 I/O (模拟)
1	1	带上拉电阻的开漏

这里 x=0, 1, 2, 4, 6, 7 (端口号)，y=0~7(端口引脚号)。寄存器 PxM0 和 PxM1 列举了每个引脚的描述。

14.2.1. 端口 0 寄存器

P0: 端口 0 寄存器

SFR 页 = 0~F

SFR 地址 = 0x80

复位值 = 1010-0110

7	6	5	4	3	2	1	0
P0.7	0	P0.5	0	0	P0.2	P0.1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 端口 0 输出数据通过 CPU 置位/清零。

P0M0: 端口 0 模式寄存器 0

SFR 页 = 仅 0 页

SFR 地址 = 0x93

复位值 = 0000-0000

7	6	5	4	3	2	1	0
P0M0.7	0	P0M0.5	0	0	P0M0.2	P0M0.1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P0M1: 端口 0 模式寄存器 1

SFR 页 = 仅 4 页

SFR 地址 = 0x92

复位值 = 1111-1111

7	6	5	4	3	2	1	0
P0M1.7	1	P0M1.5	1	1	P0M1.2	P0M1.1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.2. 端口 1 寄存器

P1: 端口 1 寄存器

SFR 页 = 0~F

SFR 地址 = 0x90

复位值 = 1111-1111

7	6	5	4	3	2	1	0
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 端口 1 输出数据通过 CPU 置位/清零.

P1M0: 端口 1 模式寄存器 0

SFR 页 = 0~F

SFR 地址 = 0x91

复位值 = 0000-0000

7	6	5	4	3	2	1	0
P1M0.7	P1M0.6	P1M0.5	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1M1: 端口 1 模式寄存器 1

SFR 页 = 仅 0 页

SFR 地址 = 0x92

复位值 = 1111-1111

7	6	5	4	3	2	1	0
P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.3. 端口 2 寄存器

P2: 端口 2 寄存器

SFR 页 = 0~F

SFR 地址 = 0xA0

复位值 = 0101-1100

7	6	5	4	3	2	1	0
0	P2.6	0	P2.4	P2.3	P2.2	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 端口 2 输出数据通过 CPU 置位/清零.

P2M0: 端口 2 模式寄存器 0

SFR 页 = 仅 0 页

SFR 地址 = 0x95

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	P2M0.6	0	P2M0.4	P2M0.3	P2M0.2	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2M1: 端口 2 模式寄存器 1

SFR 页 = 仅 1 页

SFR 地址 = 0x92

复位值 = 1111-1111

7	6	5	4	3	2	1	0
1	P2M1.6	1	P2M1.4	P2M1.3	P2M1.2	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.4. 端口 3 寄存器

P3: 端口 3 寄存器

SFR 页 = 0~F

SFR 地址 = 0xB0

复位值 = 0011-1111

7	6	5	4	3	2	1	0
0	0	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 端口 3 输出数据通过 CPU 置位/清零.

P3M0: 端口 3 模式寄存器 0

SFR 页 = 0~F

SFR 地址 = 0xB1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	P3M0.5	P3M0.4	P3M0.3	P3M0.2	P3M0.1	P3M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3M1: 端口 3 模式寄存器 1

SFR 页 = 0~F

SFR 地址 = 0xB2

复位值 = 0000-0000

7	6	5	4	3	2	1	0
1	1	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3M2: 端口 3 模式寄存器 2

SFR 页 = F

SFR 地址 = 0x92

复位值 = 1111-1111

7	6	5	4	3	2	1	0
1	1	P3M2.5	P3M2.4	P3M2.3	P3M2.2	P3M2.1	P3M2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.5. 端口 4 寄存器

P4: 端口 4 寄存器

SFR 页 = 0~F

SFR 地址 = 0xE8

复位值 = 1011-0011

7	6	5	4	3	2	1	0
P4.7	0	P4.5	P4.4	0	0	P4.1	P4.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 端口 4 输出数据通过 CPU 置位/清零.

P4.5 和 P4.4 有复用功能为 OCD_SDA 和 OCD_SCL.

P4.7 有复用功能为 RST 输入

P4M0: 端口 4 模式寄存器 0

SFR 页 = 仅 0 页

SFR 地址 = 0xB3

复位值 = 1011-0000

7	6	5	4	3	2	1	0
P4M0.7	0	P4M0.5	P4M0.4	0	0	P4M0.1	P4M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注: 当 P4.7/RST 用做端口引脚时, 不建议将其编程为输入使用, 以避免在启动时有高电平输入到此引脚, 导致 MCU 锁定在复位状态

P4M1: 端口 4 模式寄存器 1

SFR 页 = 仅 2 页

SFR 地址 = 0x92

复位值 = 1111-1111

7	6	5	4	3	2	1	0
P4M1.7	1	P4M1.5	P4M1.4	1	1	P4M1.1	P4M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注：当 P4.7/RST 用做端口引脚时，不建议将其编程为输入使用，以避免在启动时有高电平输入到此引脚，导致 MCU 锁定在复位状态

14.2.6. 端口 6 寄存器

P6: 端口 6 Register

SFR 页 = 仅 1 页

SFR 地址 = 0xF8

复位值 = 0000-0011

7	6	5	4	3	2	1	0
0	0	0	0	0	0	P6.1	P6.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 端口 6 输出数据通过 CPU 置位/清零.

P6M0: 端口 6 模式寄存器 0

SFR 页 = 仅 1 页

SFR 地址 = 0xB5

复位值 = 0000-0000

7	6	5	4	3	2	1	0
7	6	5	4	3	2	P6M0.1	P6M0.0
0	0	0	0	0	0	R/W	R/W

P6M1: 端口 6 模式寄存器 1

SFR 页 = 仅 3 页

SFR 地址 = 0x92

复位值 = 1111-1111

7	6	5	4	3	2	1	0
1	1	1	1	1	1	P6M1.1	P6M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.7. 端口输出驱动力控制寄存器

MG82F6P32 所有端口引脚(除了 P4.7、P6.1 和 P6.0 之外)都有二种软件可选的驱动力。请参考端口引脚的驱动力信息

PDRVC0: 端口驱动力控制寄存器 0

SFR 页 = 仅 2 页

SFR 地址 = 0xB4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
P3DC1	P3DC0	P2DC1	P2DC0	P1DC1	P1DC0	P0DC1	P0DC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: P3DC1, 端口 3 高 4 位输出驱动力控制。

0: P3.7 ~ P3.4 输出选择为高驱动力。

1: P3.7 ~ P3.4 输出选择为低驱动力。

Bit 6: P3DC0, 端口 3 低 4 位输出驱动力控制。

0: P3.3 ~ P3.0 输出选择为高驱动力。

1: P3.3 ~ P3.0 输出选择为低驱动力。

Bit 5: P2DC1, 端口 2 高 4 位输出驱动力控制。

0: P2.7 ~ P2.4 输出选择为高驱动力。

1: P2.7 ~ P2.4 输出选择为低驱动力。

Bit 4: P2DC0, 端口 2 低 4 位输出驱动力控制。

0: P2.3 ~ P2.0 输出选择为高驱动力。

1: P2.3 ~ P2.0 输出选择为低驱动力。

Bit 3: P1DC1, 端口 1 高 4 位输出驱动力控制。

0: P1.7 ~ P1.4 输出选择为高驱动力。

1: P1.7 ~ P1.4 输出选择为低驱动力。

Bit 2: P1DC0, 端口 1 低 4 位输出驱动力控制。

0: P1.3 ~ P1.0 输出选择为高驱动力。

1: P1.3 ~ P1.0 输出选择为低驱动力。

Bit 1: P0DC1, 端口 1 高 4 位输出驱动力控制。

0: P0.7 ~ P0.4 输出选择为高驱动力。

1: P0.7 ~ P0.4 输出选择为低驱动力。

Bit 0: P0DC0, 端口 1 低 4 位输出驱动力控制。

0: P0.3 ~ P0.0 输出选择为高驱动力。

1: P0.3 ~ P0.0 输出选择为低驱动力。

PDRVC1: 端口驱动力控制寄存器 1

SFR 页 = 仅 3 页

SFR 地址 = 0xB4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	0	P4DC1	P4DC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7 ~ 2: 保留。写该寄存器时，此位必须写“0”。

Bit 1: P4DC1, 端口 4 高 4 位输出驱动力控制。

0: P4.6 ~ P4.4 输出选择为高驱动力。

1: P4.6 ~ P4.4 输出选择为低驱动力。

Bit 0: P4DC0, 端口 4 低 4 位输出驱动力控制。

0: P4.3 ~ P4.0 输出选择为高驱动力。

1: P4.3 ~ P4.0 输出选择为低驱动力。

14.2.8. 端口输出快速驱动控制寄存器

MG82F6P32 所有端口引脚(除了 P4.7/P6.1/P6.0 之外)都有二种软件可选的快速驱动。请参考端口引脚的快速驱动信息。

P3FDC: 端口 3 快速驱动控制寄存器

SFR 页 = 仅 7 页

SFR 地址 = **0x92**

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	P3FDC.5	P3FDC.4	P3FDC.3	P3FDC.2	P3FDC.1	P3FDC.0
R/W	R/W	R/W	R/W	R/W	R/W	RW	RW

Bit 5~0: 端口 3 输出快速驱动控制通过 CPU 置位/清零。

0: 禁止端口引脚输出快速驱动。

1: 使能端口引脚输出快速驱动。

P1FDC: 端口 1 快速驱动控制寄存器

SFR 页 = 仅 8 页

SFR 地址 = **0x92**

复位值 = 0000-0000

7	6	5	4	3	2	1	0
P1FDC.7	P1FDC.6	P1FDC.5	P1FDC.4	P1FDC.3	P1FDC.2	P1FDC.1	P1FDC.0
R/W	R/W	R/W	R/W	R/W	R/W	RW	RW

Bit 7~0: 端口 1 输出快速驱动控制通过 CPU 置位/清零。

0: 禁止端口引脚输出快速驱动。

1: 使能端口引脚输出快速驱动。

P2FDC: 端口 2 快速驱动控制寄存器

SFR 页 = 仅 9 页

SFR 地址 = **0x92**

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	P2FDC.6	0	P2FDC.4	P2FDC.3	P2FDC.2	0	0
R/W	R/W	R/W	R/W	R/W	R/W	RW	RW

Bit 7~0: 端口 2 输出快速驱动控制通过 CPU 置位/清零。

0: 禁止端口引脚输出快速驱动。

1: 使能端口引脚输出快速驱动。

P4FDC: 端口 4 快速驱动控制寄存器

SFR 页 = 仅 A 页

SFR 地址 = **0x92**

复位值 = 0000-0000

7	6	5	4	3	2	1	0
P4FDC.7	0	P4FDC.5	P4FDC.4	0	0	P4FDC.1	P4FDC.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 端口 4 输出快速驱动控制通过 CPU 置位/清零。

0: 禁止端口引脚输出快速驱动。

1: 使能端口引脚输出快速驱动。

P0FDC: 端口 0 快速驱动控制寄存器

SFR 页 = 仅 B 页

SFR 地址 = **0x92**

复位值 = 0000-0000

7	6	5	4	3	2	1	0
P0FDC.7	0	P0FDC.5	0	0	P0FDC.2	P0FDC.1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 端口 0 输出快速驱动控制通过 CPU 置位/清零。

0: 禁止端口引脚输出快速驱动。

1: 使能端口引脚输出快速驱动。

AUXR10: 辅助寄存器 10

SFR 页 = 仅 7 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Bit 0: P60FD, P6.0 快速驱动。
- 0: P6.0 默认驱动输出。
- 1: P6.0 快速驱动输出使能。若 P6.0 被配置为时钟输出，当 P6.0 输出频率大于 12MHz（5V）或者大于 6MHz（3V）的应用时使能此位。

15. 中断

MG82F6P32 有 **19** 个带 4 级优先级的中断源。这些中断源有几个特殊功能寄存器 SFR 与设定四个级别的中断优先级相关。这些特殊功能寄存器分别是 IE、IP0L、IP0H、EIE1、EIP1L、EIP1H、EIE2、EIP2L、EIP2H 和 XICON。IP0H(中断优先级 0 高字节)、EIP1H(扩展中断优先级 1 高字节)和 EIP2H(扩展中断优先级 2 高字节)寄存器使四个级别的中断结构合理分配。四个级别的中断优先级在处理这些中断源时更加灵活。

15.1. 中断结构

表 15-1 列出了所有的中断源。使能位被允许，中断请求时硬件会产生一个中断请求标志，当然，总中断使能位 EA (IE 寄存器) 必须使能。中断请求位能由软件置位或清零，这和硬件置位或清零结果相同。同理，中断可以由软件产生或取消，中断优先级位决定每个中断产生的优先级，多个中断同时产生时依照中断优先级顺序处理。中断向量地址表示中断服务程序的入口地址。图 15-1 展示了中断系统。每一个中断将在下面部分做简单的描述。

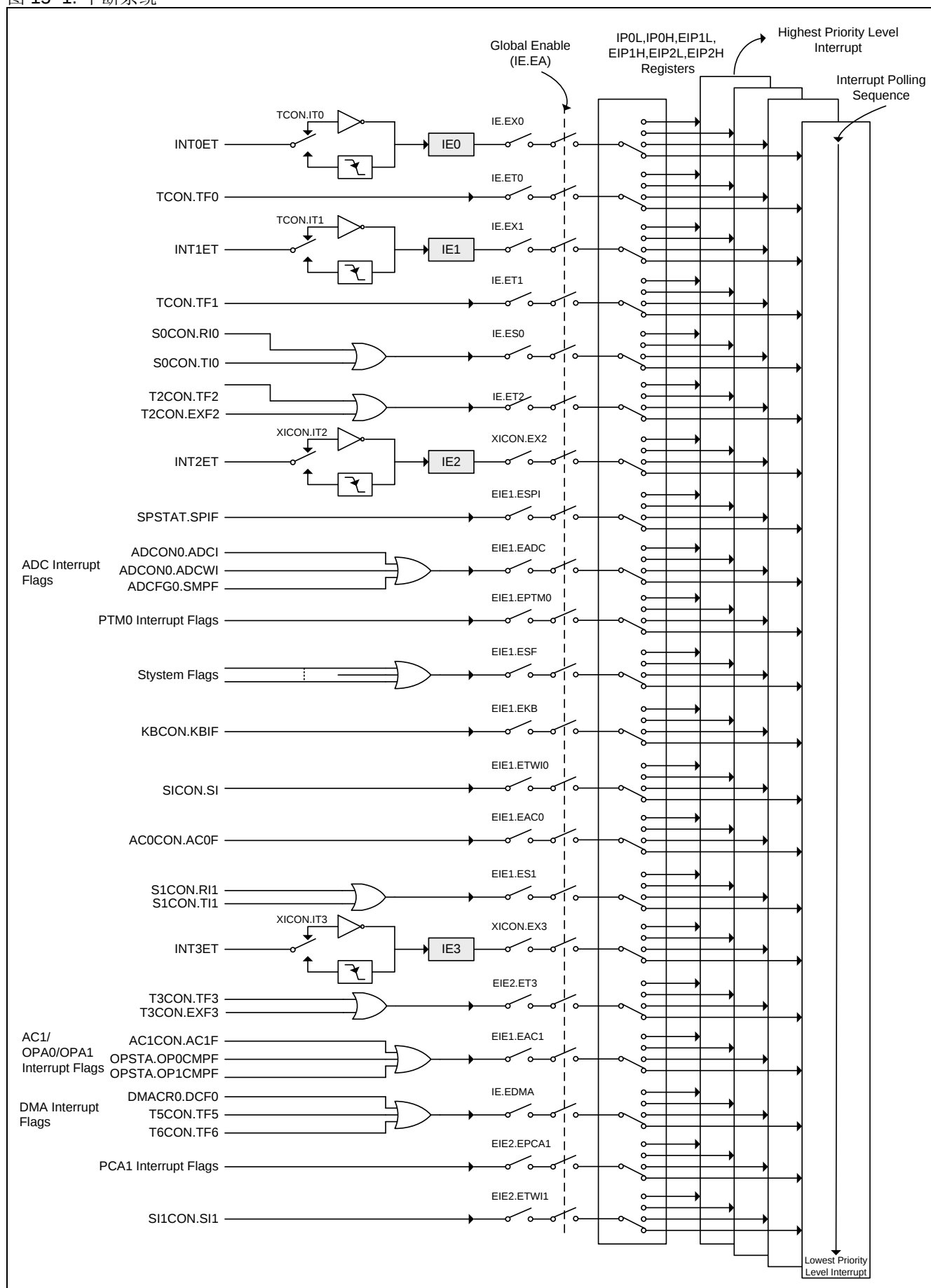
表 15-1. 中断源

序号	中断名称	使能位	请求位	优先级位	优先级	向量地址
#0	External Interrupt 0, nINT0	EX0	IE0	[PX0H, PX0L]	(Highest)	0003H
#1	Timer 0	ET0	TF0	[PT0H, PT0L]	...	000Bh
#2	External Interrupt 1, nINT1	EX1	IE1	[PX1H, PX1L]	...	0013H
#3	Timer 1	ET1	TF1	[PT1H, PT1L]	...	001BH
#4	Serial Port 0	ES0	RI0, TI0	[PSH, PSL]	...	0023H
#5	Timer 2	ET2	TF2, EXF2 (TF2L)	[PT2H, PT2L]	...	002Bh
#6	External Interrupt 2, nINT2	EX2	IE2	[PX2H, PX2L]	...	0033H
#7	SPI	ESPI	SPIF	[PSPIH, PSPIL]	...	003BH
#8	ADC	EADC	ADCI, ADCWI, SMPF	[PADCH, PADCL]	...	0043H
#9	PTM0	EPTM0	CF, CCFn (n=0~5)	[PPTMH, PPTML]	...	004Bh
#10	System Flag	ESF	(Note 1)	[PSFH, PSFL]	...	0053H
#11	Keypad Interrupt	EKB	KBIF	[PKBH, PKBL]	...	005BH
#12	TWI0/I2C0	ETWI0	SI	[PTWI0H, PTWI0L]	...	0063H
#13	Analog Comparator 0	EAC0	AC0F	[PAC0H, PAC0L]	...	006BH
#14	Serial Port 1	ES1	RI1, TI1	[PS1H, PS1L]	...	0073H
#15	External Interrupt 3, nINT3	EX3	IE3	[PX3H, PX3L]	...	007BH
#16	Timer 3	ET3	TF3, EXF3 (TF3L)	[PT3H, PT3L]	...	0083H
#17	Analog Comparator 1/ OPA0/OPA1	EAC1	AC1F, OP0CMPF, OP1CMPF	[PAC1H, PAC1L]	...	008BH
#18	DMA	EDMA	(Note 2)	[PDMAH, PDMAL]	...	0093H
#19	PCA1	EPCA1	C1F, C1CFn (n=0~1)	[PAC2H, PAC2L]	...	009BH
#20	Reserved					00A3H
#21	Reserved					00ABH
#22	Reserved					00B3H
#23	TWI1	ETWI1	SI1	[PTWI1H, PTWI1L]	(Lowest)	00BBH

注1：系统标志中断标志位包括：PCON1 寄存器的 WDTF、BOF0、BOF1、RTCF、SPWF 和 MCDF；S0CON 寄存器的 TI0；AUXR0 寄存器的 BM0F 和 BM1F；AUXR2 寄存器的 STAF 和 STOF。

注2：DMA 中断标志位包括：DCF0、TF5 和 TF6。

图 15-1. 中断系统



15.2. 中断源

表 15-2. 中断源标志位

No	源名称	请求位	位地址
#0	External Interrupt 0, nINT0	IE0	TCON.1
#1	Timer 0	TF0	TCON.5
#2	External Interrupt 1, nINT1	IE1	TCON.3
#3	Timer 1	TF1	TCON.7
#4	Serial Port 0	RI0, TI0	S0CON.0 S0CON.1
#5	Timer 2	TF2, EXF2, (TF2L)	T2CON.7 T2CON.6 T2CON.5
#6	External Interrupt 2, nINT2	IE2	XICON.1
#7	SPI	SPIF	SPSTAT.7
#8	ADC	ADCI, ADCWI, SMPF	ADCON0.4 ADCON0.6 ADCFG0.2
#9	PTM0	CF, CCFn (n=0~5)	CCON.7 CCON.5~0
#10	System Flag	WDTF, BOF0, BOF1, SPWF, RTCF, MCDF, STAF, STOF, (TI0)	PCON1.0 PCON1.1 PCON1.2 PCON1.3 PCON1.4 PCON1.5 AUXR2.7 AUXR2.6 S0CON.1
#11	Keypad Interrupt	KBIF	KBCON.0
#12	TWI0/I2C0	SI	SICON.3
#13	Analog Comparator 0	AC0F	AC0CON.4
#14	Serial Port 1	RI1, TI1	S1CON.0 S1CON.1
#15	External Interrupt 3, nINT3	IE3	XICON.5
#16	Timer 3	TF3, EXF3, (TF3L)	T3CON.7 T3CON.6 T3CON.5
#17	Analog Comparator 1 OPA0/OPA1	AC1F OP0CMPF OP1CMPF	AC1CON.4 OPSTA.0 OPSTA.1
#18	DMA	DCF0 TF5 TF6	DMACR0.0 T5CON.7 T6CON.7
#19	PCA1	C1F, C1CFn	C1CON.7 C1CON.1~0
#20	Reserved		
#21	Reserved		
#22	Reserved		
#23	TWI1/I2C1	SI1	SI1CON.3

通过 TCON 寄存器的位 IT0 和 IT1 及 XICON 寄存器的位 IT2 和 IT3 可以设定外部中断 nINT0, nINT1, nINT2 和 nINT3 为电平触发或边沿触发。实际产生这些中断的标志位是 TCON 的 IE0 和 IE1, XICON 的 IE2 和 IE3。当产生

一个外部中断时，仅当中断被激活引导进入中断服务程序由硬件将外部中断所产生的标志位清零，由外部请求源控制这个请求位，而不是片内硬件。

DMA 中断由 DCF0、TF5 和 TF6 的逻辑或产生。当进入中断服务程序时，硬件不会清除这些标志位。

TF0 和 TF1 产生定时器 0 和定时器 1 中断，多数情况下这两个标志位由它们对应的定时/计数寄存器翻转事件置位。定时器中断发生后，进入中断服务程序，硬件将清除这个标志位。

串口 0 中断由位 RI0 和位 TI0 的逻辑或产生。执行中断服务程序后不会被硬件清除须由软件清零，可以在中断服务程序中查询 RI0 和 TI0 判断是接收中断还是发送中断。

串口 1 中断由位 RI1 和位 TI1 的逻辑或产生。执行中断服务程序后不会被硬件清除须由软件清零，可以在中断服务程序中查询 RI1 和 TI1 判断是接收中断还是发送中断。

定时器/计数器 2 中断由两个标志位 TF2 或 EXF2 产生。如果定时器 2 在分立模式，TL2 的溢出将置位另一个中断标志 TF2L。跟串口一样，执行中断服务程序后这些标志位不会被硬件清除。

定时器/计数器 3 中断由两个标志位 TF3 或 EXF3 产生。如果定时器 3 在分立模式，TL3 的溢出将置位另一个中断标志 TF3L。跟串口一样，执行中断服务程序后这些标志位不会被硬件清除。

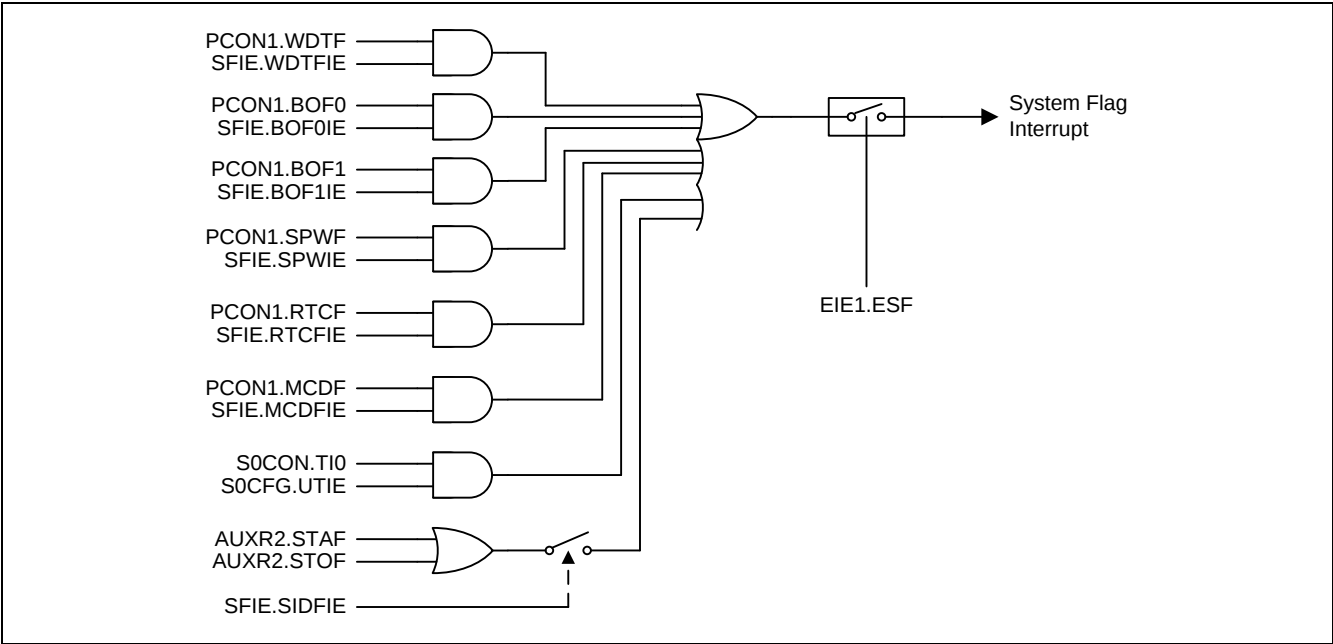
SPI 中断由寄存器 SPSTAT 里的 SPIF 位产生，SPI 引擎完成一个 SPI 传送后置该标志位。该标志位在执行中断服务程序后不会被硬件清除。

ADC 中断由寄存器 ADCON0 里的 ADCI 和 ADCWI，ADCFG0 里的 SMPF 产生。这些标志位在执行中断服务程序后不会被硬件清除。

PTM0 中断由寄存器 CCON 里的 CF、CCF5、CCF4、CCF3、CCF2、CCF1 和 CCF0 位逻辑或产生。PCA1 中断由寄存器 C1CON 里的 C1CF1 和 C1CF0 位产生。这些标志位在执行中断服务程序后不会被硬件清除。中断服务程序应当轮询这些标志位去判断是哪一个请求服务，并且在软件里清除这些中断标志位。

系统标志中断由 RTCF、BOF1、BOF0、WDTF、SPWF、MCDF、TI0、STAF 和 STOF 位产生。STAF 和 STOF 存在寄存器 AUXR2 里，由串行接口监测置位这两个标志位。串行口 TI 标志可以通过置位 UTIE 选择与系统标志中断共享中断向量。复位标志存在于寄存器 PCON1 里。RTC 计数器溢出置位 RTCF。片内低电压监测器(BOD1 和 BOD0)监测到低电压时置位 BOF1 和 BOF0。看门狗溢出置位 WDTF。SPWF 被置位表示 SP 监视器警告堆栈指针即将溢出。监测到 XTAL 时钟丢失置位 MCDF。这些标志位在执行中断服务程序后不会被硬件清除。系统标志中断结构如图 15-2 所示

图 15-2. 系统标志中断结构



键盘中断由 KBCON 寄存器的位 KBIF 来产生，KBIF 由键盘模块遇到键输入来置位。执行中断服务程序后不会被硬件清除。

TWI0/I2C0 中断由 SICON 寄存器的位 SI 来产生，SI 由 TWI0/I2C0 引擎检测到一个新的总线状态来置位。执行中断服务程序后不会被硬件清除。

TWI1/I2C1 中断由 SI1CON 寄存器的位 SI1 来产生，其功能和 TWI0/I2C 一样

AC0 中断由 AC0CON 寄存器的位 AC0F 来产生，AC0F 是通过 AC0OUT 对上升、下降或双边缘的变化检测来设置的。执行中断服务程序后不会被硬件清除。

AC1 中断由 AC1CON 寄存器的位 AC1F 来产生。

当 OPA0 和 OPA1 在 ACMP 模式下也可以产生中断。共享使能位 EAC1 来控制由 OPA0/1 引起的中断。OP0 中断由 OPASTS 中的 OP0CMPF 产生。OPA1 中断由 OPASTA 中的 OP1CMPF 产生。

所有这些中断标志都能被软件置位或清零，跟硬件置位或清零的结果是一样的。也就是说，中断能通过软件来产生也可以软件来取消。

15.3. 中断使能

表 15-3. 中断使能

No	源名称	使能位	位地址
#0	External Interrupt 0, nINT0	EX0	IE.0
#1	Timer 0	ET0	IE.1
#2	External Interrupt 1, nINT1	EX1	IE.2
#3	Timer 1	ET1	IE.3
#4	Serial Port 0	ES0	IE.4
#5	Timer 2	ET2	IE.5
#6	External Interrupt 2, nINT2	EX2	XICON.2
#7	SPI	ESPI	EIE1.0
#8	ADC	EADC	EIE1.1
#9	PTM0	EPTM0	EIE1.2
#10	System Flag	ESF	EIE1.3
#11	Keypad Interrupt	EKB	EIE1.5
#12	TWI0/I2C0	ETWI0	EIE1.6
#13	Analog Comparator 0, AC0	EAC0	EIE1.7
#14	Serial Port 1	ES1	EIE1.4
#15	External Interrupt 3, nINT3	EX3	XICON.6
#16	Timer 3	ET3	EIE2.0
#17	Analog Comparator 1/ OPA0/OPA1	EAC1	EIE2.1
#18	DMA	EDMA	IE.6
#19	PCA1	EPCA1	EIE2.5
#20	Reserved		
#21	Reserved		
#22	Reserved		
#23	TWI1/I2C1	ETWI1	EIE2.6

MG82F6P32 有 **19** 个中断源可用。每个中断源可以通过 IE、EIE1、EIE2 和 XICON 寄存器的中断使能位置位或清零各自中断使能或禁止。IE 也提供一个全局中断使能位(EA)，此位清零可以立刻禁止所有中断。如果此位置位中断由相应的中断使能位各自使能或禁止。如果此位清零则所有中断被禁止。

15.4. 中断优先级

服务中断的优先级除了有 4 个级别比 80C51 多 2 个之外跟 80C51 一样。优先级位决定每个中断的优先级(见表 15-1)。IP0L、IP0H、EIP1L、EIP1H、EIP2L 和 EIP2H 跟 4 个级别优先级中断相关。位的值和优先级的关系如表 15-4 所示。

表 15-4. 中断优先级

{IPnH.x , IPnL.x}	优先级
11	1 (最高)
10	2
01	3
00	4

每个中断源都有两个中断优先级相关位。一个位在 IPnH 寄存器另一个在 IPnL 寄存器。高优先级中断不会被低优先级中断打断。如果两个不同优先级的中断请求同时出现，较高优先级将被执行。如果相同优先级的中断请求同时出现，则按照内部优先级排序执行。同一优先级的内部优先级排序和中断向量地址如表 15-2 所示。

15.5. 中断处理

每一个系统时钟周期将采样每一个中断标志。在下一个系统时钟采样成功。如果其中一个标志在第一个周期置位，第二个周期(轮询周期)找到并且只要没有被下列条件阻止则中断系统产生一个硬件调用(LCALL)相应的中断服务程序。

阻止条件:

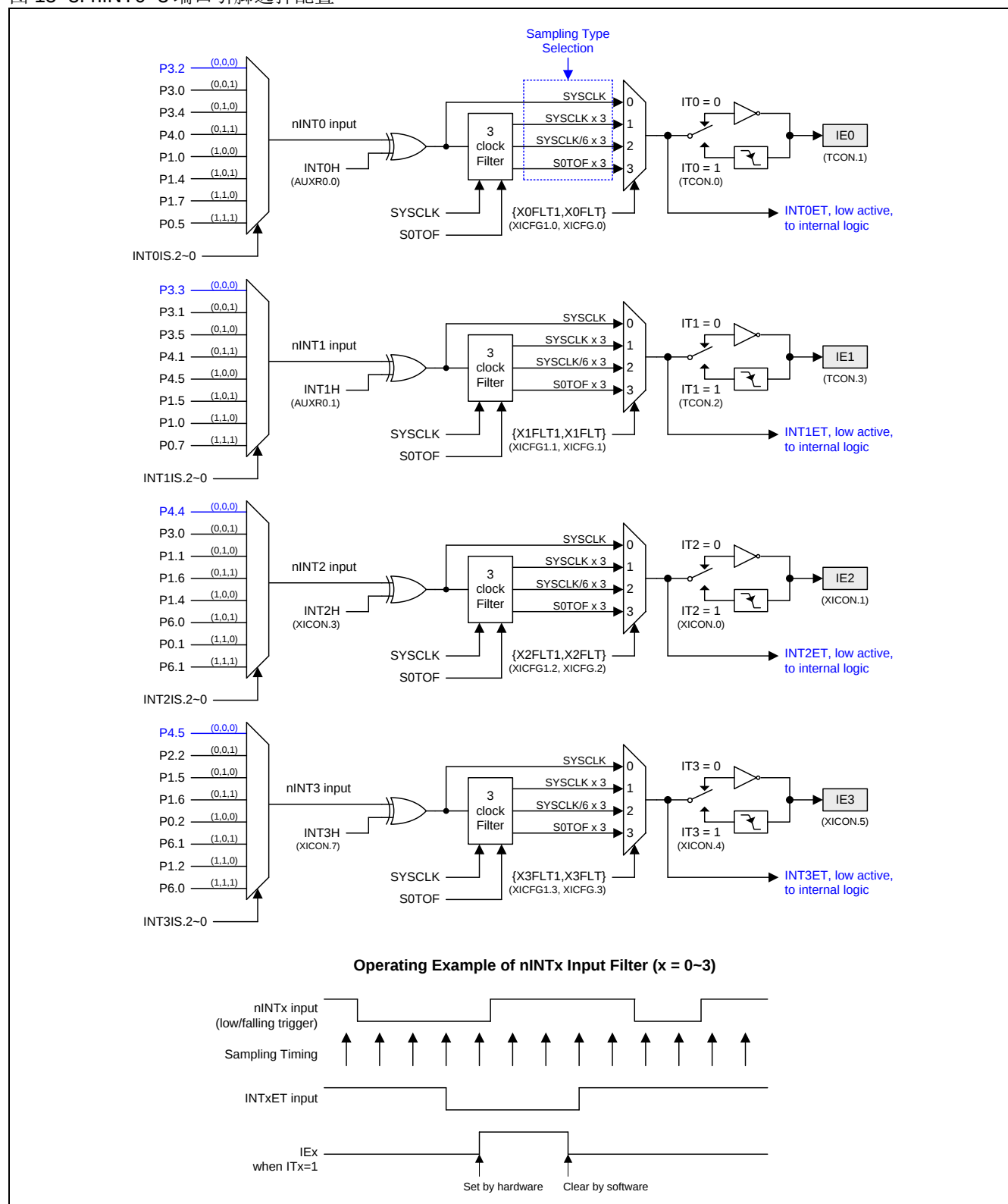
- 进行中已经有一个同级或更高级优先级的中断。
- 进行中当前周期(轮询周期)不是指令执行结束周期.
- 指令进行是 RETI 或 IE、IP0L、IP0H、EIE1、EIP1L、EIP1H、EIE2、EIP2L、EIP2H 和 XICON 寄存器的写操作

上述三个条件中的任意一个将阻止硬件中断调用(LCALL)去中断服务程序。条件 2 确保中断进入任意一个服务程序之前指令执行完毕。条件 3 确保如果在 RETI 执行或 IE 或 IP 的任何访问之后，进入中断服务程序之前至少一个或更多指令被执行。

15.6. nINTx 输入源选择和输入滤波器(x=0~3)

MG82F6P32 提供灵活的 nINT0、nINT1、nINT2 和 nINT3 源选择，通过多路复用器共享端口引脚输入，使大多数 IO 能够使能外部中断事件。它还提供了数字滤波器，以减少输入干扰。

图 15-3. nINT0~3 端口引脚选择配置



15.7. 中断寄存器

TCON: 定时/计数器 控制寄存器

SFR 页 = 0~F

SFR 地址 = 0x88

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: IE1, 外部中断 1(nINT1)请求标志。

0: 如果是边沿触发的中断则在进入中断向量后硬件清零。

1: 外部中断 1(nINT1)由边沿或电平触发(由 IT1 设置)硬件置位。

Bit 2: IT1, 外部中断 1(nINT1)类型控制位。

0: 软件清零选择低电平触发外部中断 1(nINT1)。如果 INT1H(AUXR0.1)置位, 则高电平触发外部中断 1(nINT1)。

1: 软件置位选择下降沿触发外部中断 1(nINT1)。如果 INT1H(AUXR0.1)置位, 则上升沿触发外部中断 1(nINT1)。

Bit 1: IE0, 外部中断 0(nINT0)请求标志。

0: 如果是边沿触发的中断则在进入中断向量后硬件清零。

1: 外部中断 0 (nINT0)由边沿或电平触发(由 IT0 设置)硬件置位。

Bit 0: IT0, 外部中断 0(nINT0)类型控制位。

0: 软件清零选择低电平触发外部中断 0(nINT0)。如果 INT0H(AUXR0.0)置位, 则高电平触发外部中断 0(nINT0)。

1: 软件置位选择下降沿触发外部中断 0(nINT0)。如果 INT0H(AUXR0.0)置位, 则上升沿触发外部中断 0(nINT0)。

IE: 中断使能寄存器

SFR 页 = 0~F

SFR 地址 = 0xA8

复位值 = 0000-0000

7	6	5	4	3	2	1	0
EA	EDMA	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: EA, 总中断使能位。

0: 禁止所有中断。

1: 使能所有中断。

Bit 6: EDMA, DMA 中断使能。

0: 禁止 DMA 中断。

1: 使能 DMA 中断。

Bit 5: ET2, 定时器 2 中断使能。

0: 禁止定时器 2 中断。

1: 使能定时器 2 中断。

Bit 4: ES, 串口 0 中断(UART0)使能。

0: 禁止串口 0 中断。

1: 使能串口 0 中断。

Bit 3: ET1, 定时器 1 中断使能。

0: 禁止定时器 1 中断。

1: 使能定时器 1 中断。

Bit 2: EX1, 外部中断 1(nINT1)使能。

0: 禁止外部中断 1。

1: 使能外部中断 1。

Bit 1: ET0, 定时器 0 中断使能。

0: 禁止定时器 0 中断。

1: 使能定时器 0 中断。

Bit 0: EX0, 外部中断 0(nINT0)使能。

- 0: 禁止外部中断 0。
1: 使能外部中断 0。

AUXR0: 辅助寄存器 0

SFR 页 = 0~F

SFR 地址 = 0xA1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C1BKF	PBKF	0	0	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: PBKF, PWM 中止标志。此位由 PWM 终止源使能置位。如果此位置位, 则使能的 PWM 通道 0~5 将被锁住并且输出引脚保持最初的 GPIO 状态。

0: 没有 PWM 中止事件出现。仅由软件清零。

1: PWM 中止事件出现或软件触发一个 PWM 终止。

Bit 1: INT1H, INT1 高电平/上升沿触发使能。

0: 保留 INT1 在选择的端口引脚上低电平或下降沿触发。

1: 设置 INT1 在选择的端口引脚上高电平或上升沿触发。

Bit 0: INT0H, INT0 高电平/上升沿触发使能。

0: 保留 INT0 在选择的端口引脚上低电平或下降沿触发。

1: 设置 INT0 在选择的端口引脚上高电平或上升沿触发。

XICON: 扩展中断控制寄存器

SFR 页 = 0~F

SFR 地址 = 0xC0

复位值 = 0000-0000

7	6	5	4	3	2	1	0
INT3H	EX3	IE3	IT3	INT2H	EX2	IE2	IT2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: INT3H, INT3 高电平/上升沿触发使能。

0: 保留 INT3 在选择的端口引脚上低电平或下降沿触发。

1: 设置 INT3 在选择的端口引脚上高电平或上升沿触发。

Bit 2: EX3, 外部中断 3(nINT3)使能。

0: 禁止外部中断 3。

1: 使能外部中断 3。

当 CPU 在空闲或掉电模式, 如果 EX3 使能 nINT3 事件触发 IE3 则可以唤醒 CPU; 如果 EX3 禁止, nINT3 事件触发 IE3 则不能唤醒 CPU。

Bit 5: IE3, 外部中断 3(nINT3)请求标志。

0: 如果是边沿触发的中断则在进入中断向量后硬件清零。也可以软件清零。

1: 检测到外部中断边沿硬件置位。也可以软件置位。

Bit 4: IT3, 外部中断 3(nINT3)类型控制位。

0: 软件清零选择低电平触发外部中断 3。如果 INT3H 置位, 则高电平触发外部中断 3。

1: 软件置位选择下降沿触发外部中断 3。如果 INT3H 置位, 则上升沿触发外部中断 3。

Bit 3: INT2H, INT2 高电平/上升沿触发使能。

0: 保留 INT2 在选择的端口引脚上低电平或下降沿触发。

1: 设置 INT2 在选择的端口引脚上高电平或上升沿触发。

Bit 2: EX2, 外部中断 2(nINT2)使能。

0: 禁止外部中断 2。

1: 使能外部中断 2。

当 CPU 在空闲或掉电模式, 如果 EX2 使能 nINT2 事件触发 IE2 则可以唤醒 CPU; 如果 EX2 禁止, nINT2 事件触发 IE2 则不能唤醒 CPU。

Bit 1: IE2, 外部中断 2(nINT2)请求标志。

0: 如果是边沿触发的中断则在进入中断向量后硬件清零。也可以软件清零。

1: 检测到外部中断边沿硬件置位。也可以软件置位。

Bit 0: IT2, 外部中断 2 (nINT2)类型控制位。

0: 软件清零选择低电平触发外部中断 2。如果 INT2H 置位, 则高电平触发外部中断 2。

1: 软件置位选择下降沿触发外部中断 2。如果 INT2H 置位, 则上升沿触发外部中断 2。

IP0L: 中断优先级 0 低位寄存器

SFR 页 = 0~F

SFR 地址 = 0xB8

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PX3L	PX2L	PT2L	PSL	PT1L	PX1L	PT0L	PX0L
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: PX3L, 外部中断 3 中断优先级低位。

Bit 6: PX2L, 外部中断 2 中断优先级低位。

Bit 5: PT2L, 定时器 2 中断优先级低位。

Bit 4: PSL, 串口 0(UART0)中断优先级低位。

Bit 3: PT1L, 定时器 1 中断优先级低位。

Bit 2: PX1L, 外部中断 1 中断优先级低位。

Bit 1: PT0L, 定时器 0 中断优先级低位。

Bit 0: PX0L, 外部中断 0 中断优先级低位。

IP0H: 中断优先级 0 高位寄存器

SFR 页 = 0~F

SFR 地址 = 0xB7

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: PX3H, 外部中断 3 中断优先级高位。

Bit 6: PX2H, 外部中断 2 中断优先级高位。

Bit 5: PT2H, 定时器 2 中断优先级高位。

Bit 4: PSH, 串口 0(UART0)中断优先级高位。

Bit 3: PT1H, 定时器 1 中断优先级高位。

Bit 2: PX1H, 外部中断 1 中断优先级高位。

Bit 1: PT0H, 定时器 0 中断优先级高位。

Bit 0: PX0H, 外部中断 0 中断优先级高位。

EIE1: 扩展中断使能 1 寄存器

SFR 页 = 0~F

SFR 地址 = 0xAD

复位值 = 0000-0000

7	6	5	4	3	2	1	0
EAC0	ETWIO	EKB	ES1	ESF	EPTM0	EADC	ESPI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: EAC0, 使能模拟比较器 0 (AC0) 中断。

0: 禁止 AC0 中断。

1: 使能 AC0 中断。

Bit 6: ETWIO, TWIO/ I2C0 中断使能。

0: 禁止 TWIO/ I2C0 中断。

1: 使能 TWIO/ I2C0 中断。

Bit 5: EKBI, 键盘中断使能。

0: 当键盘控制模块的 KBCON.KBIF 置位时禁止中断。

1: 当键盘控制模块的 KBCON.KBIF 置位时使能中断。

Bit 4: ES1, 串口 1 中断(UART1)使能。

0: 禁止串口 1 中断。

1: 使能串口 1 中断。

Bit 3: ESF, 系统标志中断使能。

0: 当 PCON1 的位{ RTCF、BOF1、BOF0、WDTF }置位, AUXR2 的位{STAF、STOF}置位, 或 TI0 与 UTIE 一起置位时禁止中断。

1: 当 PCON1 的位{ RTCF、BOF1、BOF0、WDTF }置位, AUXR2 的位{STAF、STOF}置位, 或 TI0 与 UTIE 一起置位时使能中断。

Bit 2: EPTM0, PTM0 中断使能。

0: 禁止 PTM0 中断。

1: 使能 PTM0 中断。

Bit 1: EADC, ADC 中断使能。

0: 当 ADC 模块的 ADCON0.ADCI 置位禁止中断。

1: 当 ADC 模块的 ADCON0.ADCI 置位使能中断。

Bit 0: ESPI, SPI 中断使能。

0: 当 SPI 模块的 SPSTAT.SPIF 置位禁止中断。

1: 当 SPI 模块的 SPSTAT.SPIF 置位使能中断。

EIP1L: 扩展中断优先级 1 低位寄存器

SFR 页 = 0~F

SFR 地址 = 0xAE

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PAC0L	PTWI0L	PKBL	PS1L	PSFL	PPCAL	PADCL	PSPIL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: PAC0L, AC0 中断优先级低位

Bit 6: PTWI0L, TWI0 中断优先级低位。

Bit 5: PKBL, 键盘中断优先级低位。

Bit 4: PS1L, 串口 1(UART1)中断优先级低位。

Bit 3: PSFL, 系统标志中断优先级低位。

Bit 2: PPCAL, PTM0 中断优先级低位。

Bit 1: PADCL, ADC 中断优先级低位。

Bit 0: PSPIL, SPI 中断优先级低位。

EIP1H: 扩展中断优先级 1 高位寄存器

SFR 页 = 0~F

SFR 地址 = 0xAF

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PAC0H	PTWI0H	PKBH	PS1H	PSFH	PPCAH	PADCH	PSPIH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: PAC0H, AC0 中断优先级高位。

Bit 6: PTWI0H, TWI0 中断优先级高位。

Bit 5: PKBH, 键盘中断优先级高位。

Bit 4: PS1H, 串口 1(UART1)中断优先级高位。

Bit 3: PSFH, 系统标志中断优先级高位。

Bit 2: PPCAH, PTM0 中断优先级高位。

Bit 1: PADCH, ADC 中断优先级高位。

Bit 0: PSPIH, SPI 中断优先级高位。

EIE2: 扩展中断使能 2 寄存器

SFR 页 = 0~F

SFR 地址 = 0xA5

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	ETWI1	EPAC1	0	0	0	EAC1	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留。写该寄存器时, 此位必须写“0”。

Bit 6: ETWI1, 使能 TWI1 中断。

0: 禁止 TWI1 中断.
1: 使能 TWI1 中断.

Bit5: EPCA1, 使能 PCA1 中断.
0: 禁止 PCA1 中断.
1: 使能 PCA1 中断.

Bit 4~2: 保留。写该寄存器时，此位必须写“0”。

Bit 1: EAC1, 使能模拟比较器 1 (AC1) 中断.
0: 禁止 AC1 中断.
1: 使能 AC1 中断.

Bit 0: ET3, 定时器 3 中断使能.
0: 禁止定时器 3 中断.
1: 使能定时器 3 中断.

EIP2L: 扩展中断优先级 2 低位寄存器

SFR 页 = 0~F

SFR 地址 = 0xA6

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	PTWI1L	PPCA1L	0	0	0	PAC1L	PT3L
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留。写该寄存器时，此位必须写“0”。
Bit 6: PTWI1L, TWI1 中断优先级低位
Bit 5: PPCA1L, PCA1 中断优先级低位
Bit 4~2: 保留。写该寄存器时，此位必须写“0”。
Bit 1: PAC1L, AC1 中断优先级低位
Bit 0: PT3L, 定时器 3 中断优先级低位

EIP2H: 扩展中断优先级 2 高位寄存器

SFR 页 = 0~F

SFR 地址 = 0xA7

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	PTWI1H	PPCA1H	0	0	0	PAC1H	PT3H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留。写该寄存器时，此位必须写“0”。
Bit 6: PTWI1H, TWI1 中断优先级高位
Bit 5: PPCA1H, PCA1 中断优先级高位
Bit 4~2: 保留。写该寄存器时，此位必须写“0”。
Bit 1: PAC1H, AC1 中断优先级高位
Bit 0: PT3H, 定时器 3 中断优先级高位

DMACG0: DMA 配置寄存器 0

SFR 页 = 仅 8 页

SFR 地址 = 0x94

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PDMAH	PDMAL	CRCW0	--	EXTS10	EXTS00	FAEN0	LOOP0
R/W	R/W	R/W	W	R/W	R/W	R/W	R/W

Bit 7: PDMAH, DMA 中断优先级高位。

Bit 6: PDMAL, DMA 中断优先级低位。

XICFG: 外展中断配置寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xC1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
INT1IS.1	INT1IS.0	INT0IS.1	INT0IS.0	X3FLT	X2FLT	X1FLT	X0FLT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: INT1IS.1~0, 与 INT1IS.2 一起定义 nINT1 的输入端口引脚选择如下表

INT1IS.2~0	nINT1 输入引脚选择
0 0 0	P3.3
0 0 1	P3.1
0 1 0	P3.5
0 1 1	P4.1
1 0 0	P4.5
1 0 1	P1.5
1 1 0	P1.0
1 1 1	P0.7

Bit 5~4: INT0IS.1~0, 与 INT0IS.2 一起定义 nINT0 的输入端口引脚选择如下表。

INT0IS.2~0	nINT0 输入引脚选择
0 0 0	P3.2
0 0 1	P3.0
0 1 0	P3.4
0 1 1	P4.0
1 0 0	P1.0
1 0 1	P1.4
1 1 0	P1.7
1 1 1	P0.5

Bit 3: X3FLT, nINT3 滤波模式控制。与 X3FLT1 (XICFG1.3) 一起选择 nINT3 输入滤波模式

X3FLT1, X3FLT	nINT3 input filter mode
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	S0TOF x 3

Bit 2: X2FLT, nINT2 滤波模式控制。与 X2FLT1 (XICFG1.2) 一起选择 nINT2 输入滤波模式

X2FLT1, X2FLT	nINT2 输入滤波模式
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	S0TOF x 3

Bit 1: X1FLT, nINT1 滤波模式控制。与 X1FLT1 (XICFG1.1)一起选择 nINT1 输入滤波模式

X1FLT1, X1FLT	nINT1 输入滤波模式
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	S0TOF x 3

Bit 0: X0FLT, nINT0 滤波模式控制。与 X0FLT1 (XICFG1.0)一起选择 nINT0 输入滤波模式

X0FLT1, X0FLT	nINT0 输入滤波模式
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	S0TOF x 3

XICFG1: 外部中断配置1 寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xC1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
INT1IS.2	INT0IS.2	INT2IS.1	INT2IS.0	X3FLT1	X2FLT1	X1FLT1	X0FLT1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: INT1IS2, 与 INT1IS.1~0 一起定义 nINT1 的输入端口引脚选择位。

Bit 6: INT0IS2, 与 INT0IS.1~0 一起定义 nINT0 的输入端口引脚选择位。

Bit 5~4: INT2IS1~0, 与 INT2IS.2 一起定义 nINT2 的输入端口引脚选择如下表

INT2IS.2~0	nINT2 输入引脚选择
0 0 0	P4.4
0 0 1	P3.0
0 1 0	P1.1
0 1 1	P1.6
1 0 0	P1.4
1 0 1	P6.0
1 1 0	P0.1
1 1 1	P6.1

Bit 3: X3FLT1, nINT3 滤波模式控制。和 X3FLT (XICFG.3)一起选择 nINT3 的输入滤波模式。参考寄存器 XICFG 有关 nINT3 输入滤波模式定义的描述。

Bit 2: X2FLT1, nINT2 滤波模式控制。和 X2FLT (XICFG.2)一起选择 nINT2 的输入滤波模式。参考寄存器 XICFG 有关 nINT2 输入滤波模式定义的描述。

Bit 1: X1FLT1, nINT1 滤波模式控制。和 X1FLT (XICFG.1)一起选择 nINT1 的输入滤波模式。参考寄存器 XICFG 有关 nINT1 输入滤波模式定义的描述。

Bit 0: X0FLT1, nINT0 滤波模式控制。和 X0FLT (XICFG.0)一起选择 nINT0 的输入滤波模式。参考寄存器 XICFG 有关 nINT0 输入滤波模式定义的描述。

XICFG2: 外部中断配置2 寄存器

SFR 页 = 仅 2 页

SFR 地址 = 0xC1

复位值 = XXXX-0000

7	6	5	4	3	2	1	0
--	--	--	--	INT3IS.2	INT3IS.1	INT3IS.0	INT2IS.2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3~1: INT3IS.2~0, nINT3 的输入端口引脚选择如下表

INT3IS.2~0	Selected Port Pin of nINT3
0 0 0	P4.5
0 0 1	P2.2
0 1 0	P1.5
0 1 1	P1.6
1 0 0	P0.2
1 0 1	P6.1
1 1 0	P1.2
1 1 1	P6.0

Bit 0: INT2IS2, 与 INT2IS.1~0 一起定义 nINT2 的输入端口引脚选择位。

SFIE: 系统标志中断使能寄存器

SFR 页 = 0~F

SFR 地址 = 0x8E

POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SIDFIE, 串行接口侦测标志中断使能。

0: 禁止 SIDF(STAF 或 STOF) 中断。

1: 使能 SIDF(STAF 或 STOF) 中断共享系统标志中断。

Bit 6: MCDRE, 时钟丢失监测事件系统复位使能

0: 禁止 MCD 事件触发系统复位

1: 使能 MCD 事件触发系统复位

Bit 5: MCDFIE, 使能 MCDF (PCON1.5) 中断

0: 禁止 MCDF 中断。

1: 使能 MCD 模块并且使能 MCDF 中断

Bit 4: RTCFIE, 使能 RTCF (PCON1.4) 中断。

0: 禁止 RTCF 中断。

1: 使能 RTCF 中断。

Bit 3: SPWIE, 使能 SPWF (PCON1.3) 中断

0: 禁止 SPWF 中断

1: 使能 SPWF 中断

Bit 2: BOF1IE, 使能 BOF1 (PCON1.2) 中断。

0: 禁止 BOF1 中断。

1: 使能 BOF1 中断。

Bit 1: BOF0IE, 使能 BOF0 (PCON1.1) 中断。

0: 禁止 BOF0 中断。

1: 使能 BOF0 中断。

Bit 0: WDTFIE, 使能 WDTF (PCON1.0) 中断。

0: 禁止 WDTF 中断。

1: 使能 WDTF 中断。

PCON1: 电源控制寄存器 1

SFR 页 = 0~F & P

SFR 地址 = 0x97

POR = 0000-0000

7	6	5	4	3	2	1	0
SWRF	EXRF	MCDF	RTCF	SPWF	BOF1	BOF0	WDTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SWRF, 软件复位标志。

0: 必须由软件写“1”才能清零。

1: 如果一个软件复位发生, 则此位硬件置位。

Bit 6: EXRF, 外部复位标志。

0: 必须由软件写“1”才能清零。

1: 如果一个外部复位发生, 则此位硬件置位。

Bit 5: MCDF, 时钟丢失监测标志。

0: 此位必须软件写“1”清零。软件写“0”则无操作。

1: 监测到外部晶振输入的一个丢失时钟事件此位硬件置位。此位写“1”则清零 MCDF。丢失时钟监测模块由 MCDFIE 使能。如果 MCDFIE 清零, 丢失时钟监测模块则无效。一旦一个丢失时钟事件发生, 在再次选择外部晶振输入之前 MCDF 必须清零。

Bit 4: RTCF, RTC 溢出标志。

0: 必须由软件写“1”才能清零。软件写“0”无操作。

1: 当 RTCCT 溢出时, 硬件置位。写“1”将清零。

Bit 3: SPWF, SP 警告标志。

0: 必须由软件写“1”才能清零。软件写“0”无操作。

1: $SP \geq SPHB$ 时此位硬件置位。当 $SP < SPHB$, 此位写“1”则清零 SPWF。

Bit 2: BOF1, 低电压监测标志 1。

0: 必须由软件写“1”才能清零。

1: 如果低电压监测器 1 监测到工作电压匹配监测电平(4.2V/3.7/2.4/2.0), 则此位硬件置位。

Bit 1: BOF0, 低电压监测标志 0。

0: 必须由软件写“1”才能清零。

1: 如果低电压监测器 0 监测到工作电压匹配监测电平(1.7V), 则此位硬件置位。

Bit 0: WDTF, WDT 溢出标志。

0: 必须由软件写“1”才能清零。

1: 如果一个 WDT 溢出发生, 则此位硬件置位。

AUXR2: 辅助寄存器 2

SFR 页 = 仅 0 页

SFR 地址 = 0xA3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
STAF	STOF	C1PLK	COPLK	T1X12	T0X12	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: STAF, STWI(SID)的起始标志侦测。

0: 软件写“0”清零。STAF 有可能在上电过程中被置位, 所以需要在软件初始化时将 STAF 清除。

1: 硬件置位, 表示在 STWI 总线上发生了一个起始动作。

Bit 6: STOF, STWI(SID)的停止标志侦测。

0: 软件写“0”清零。STOF 有可能在上电过程中被置位, 所以需要在软件初始化时将 STOF 清除。

1: 硬件置位, 表示在 STWI 总线上发生了一个停止动作。

16. 定时器/计数器

MG82F6P32 有 4 个 16 位定时器/计数器：定时器 0，定时器 1，定时器 2 和定时器 3。所有这些都配置为定时器或事件计数器。

定时器功能，定时器预分频是每 12 个时钟周期加 1。换句话说，是标准 C51 机器周期计数一次。**AUXR2.T0X12**、**AUXR2.T1X12**、**T2MOD.T2X12** 和 **T3MOD.T3X12** 可以设置定时器 0/1/2/3 每个时钟周期计数一次。这样就是标准 C51 定时器 12 倍的速度。结合 **T0C/T**、**T0XL** 和 **T0X12** 定时器 0 时钟输入可选择其它的预分频。

计数器功能，负跳变时寄存器加 1，根据相应的外部输入引脚 **T0**，**T1**，**T2** 或 **T3**。在这些功能中，每个定时器时钟周期对外部输入信号进行采样。当采样信号出现一个高电平接着一个低电平，计数加 1。当检测到跳变时，新计数值在这一时钟周期后的下一周期结束时出现在寄存器中。

16.1. 定时器 0 和定时器 1

16.1.1. 定时器 0/1 模式 0

在模式 0，定时器寄存器配置为一个 PWM 产生器。计数器所有位从全 1 翻转到全 0，置位定时器中断标志位 TFx。定时器 0 这些控制位{T0XL, T0X12, T0C/T}设置时钟源来计数。并且也用 TR0 和{T0G1, T0GATE}选择门控源来阻止触发信号停止计数。定时器 1 这些控制位{T1X12, T1C/T}设置时钟源来计数。并且也用 TR1 和{T1G1, T1GATE}选择门控源来阻止触发信号停止计数。定时器 0 和 1 的模式 0 操作是一样的。定时器 0/1 的 PWM 功能结构图见图 16-1 和图 16-2。

图 16-1. 定时器 0 模式 0 结构

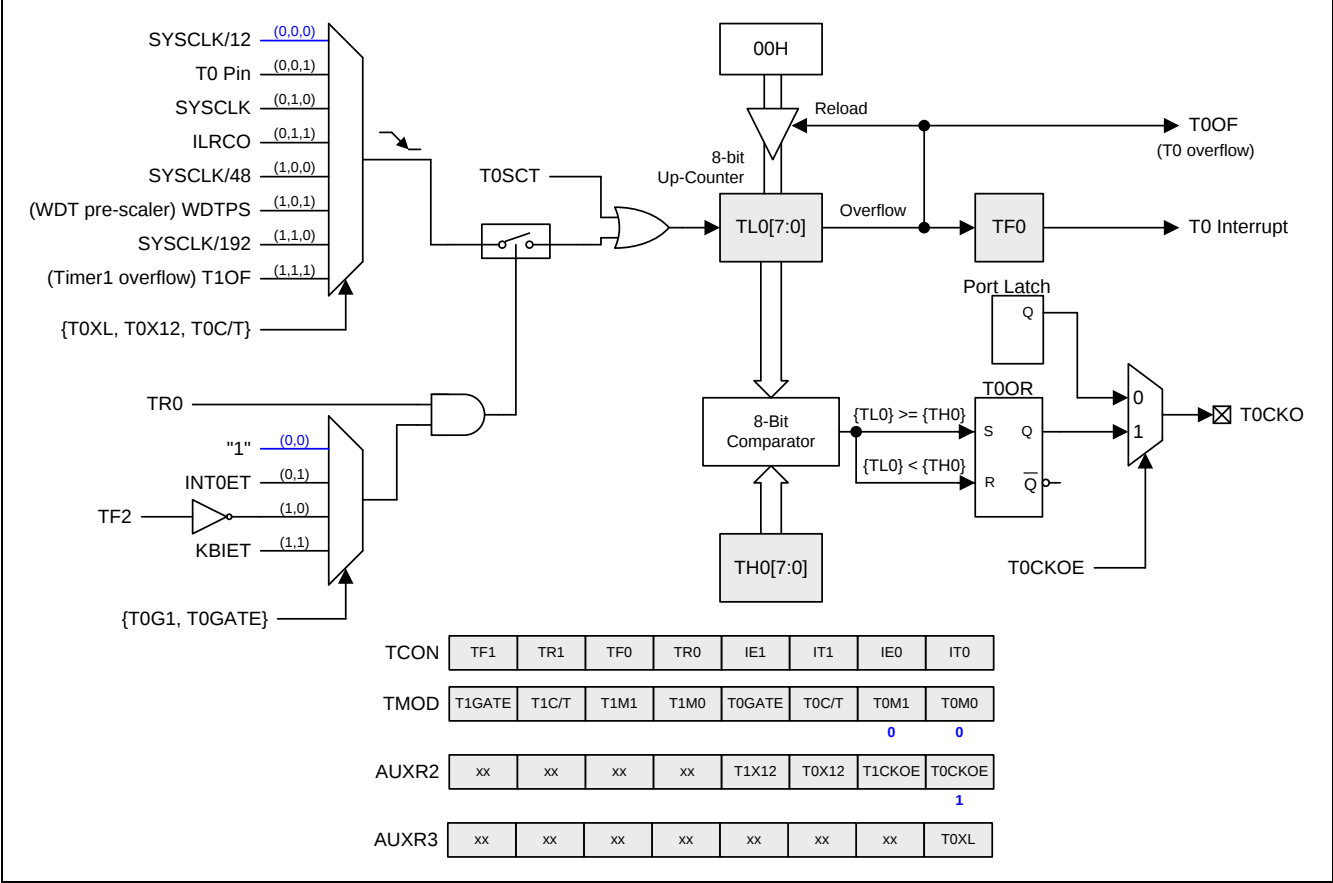
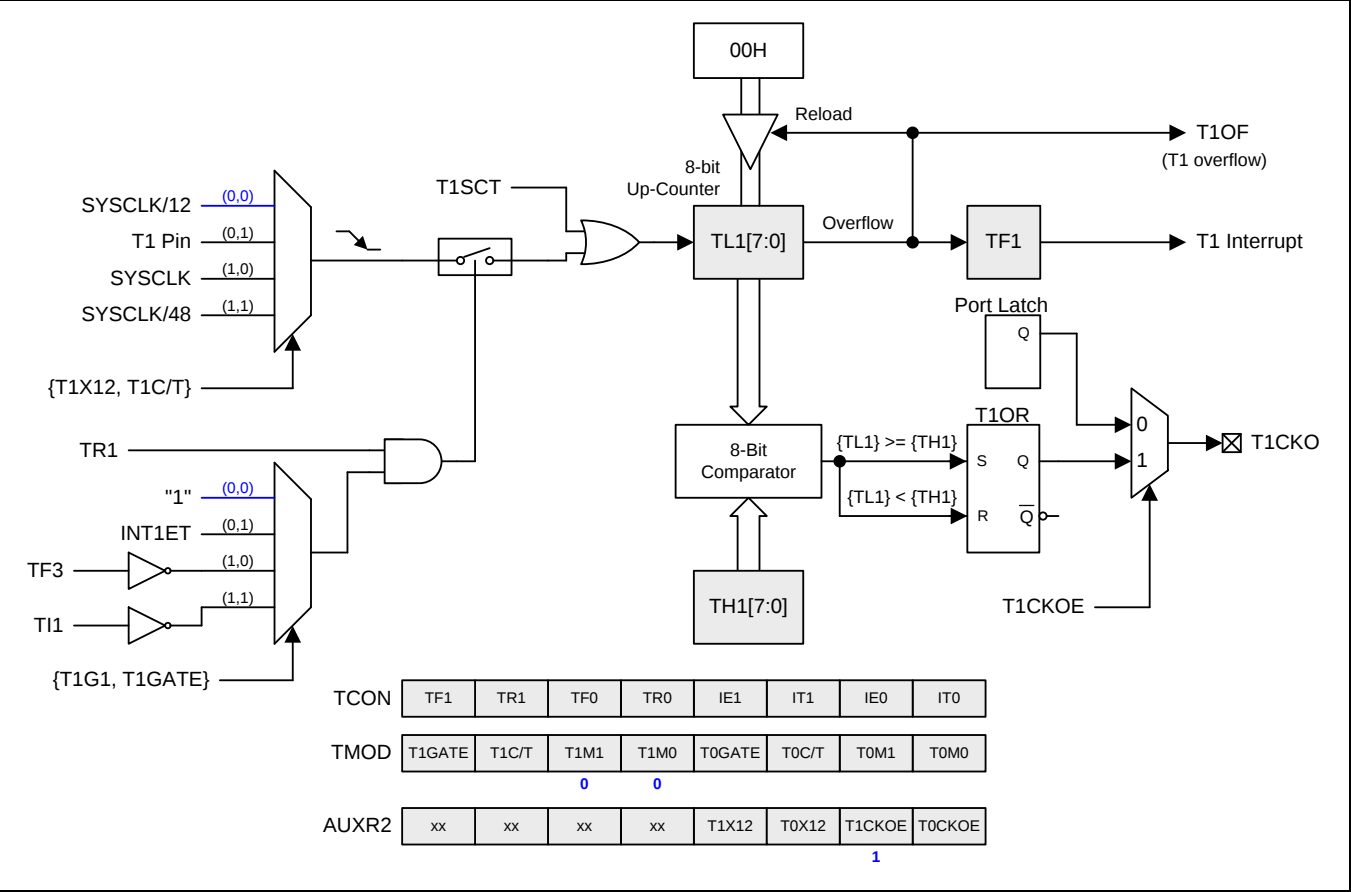


图 16-2. 定时器 1 模式 0 结构



16.1.2. 定时器 0/1 模式 1

在模式 1 定时器 0/1 配置成一个 16 位定时器或计数器。TxGATE、INTxET 和 TRx 的功能和模式 0 一样。定时器 0/1 模式 1 的结构图见图 16-3 和图 16-4。

图 16-3. 定时器 0 模式 1 结构

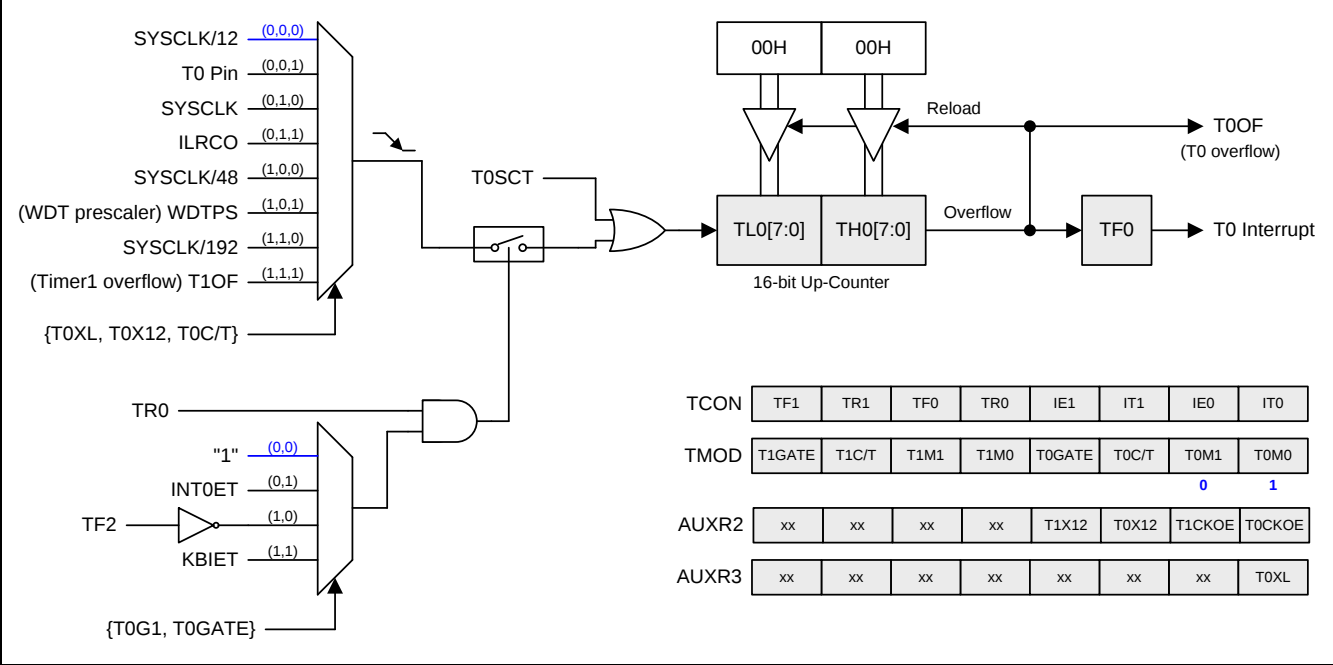
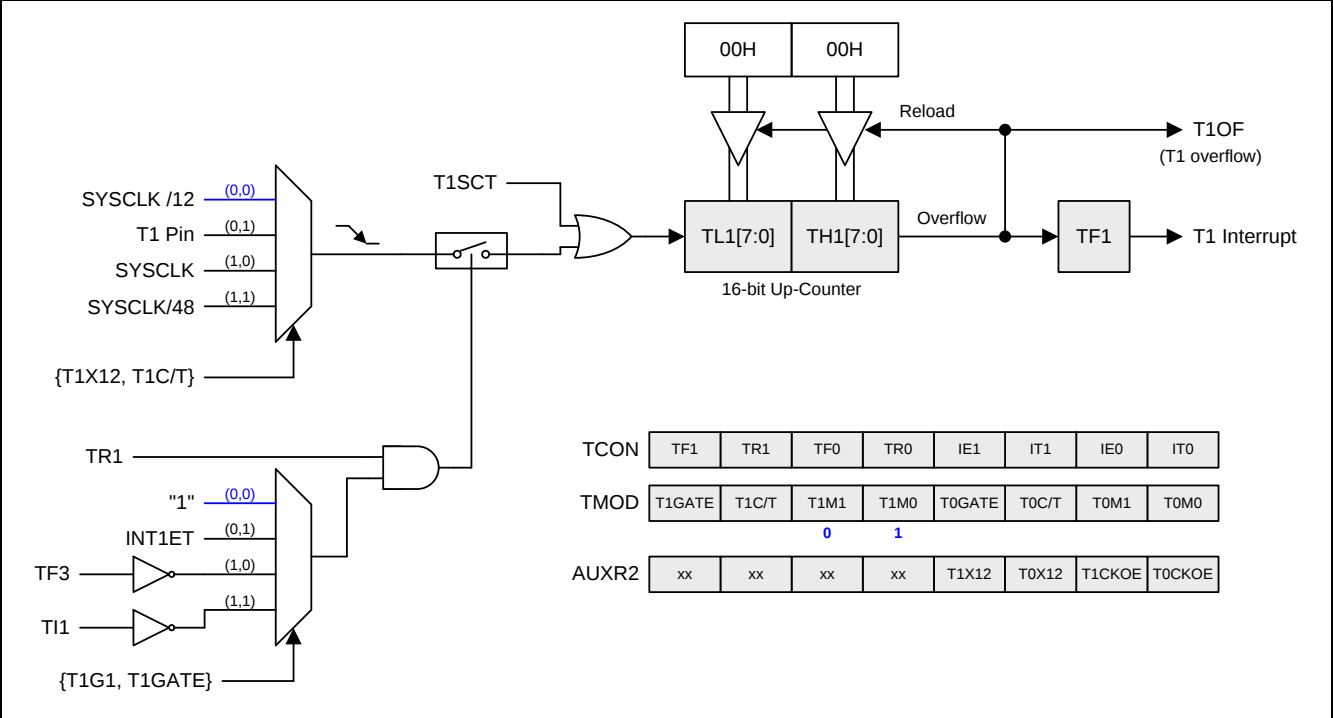


图 16-4. 定时器 1 模式 1 结构



16.1.3. 定时器 0/1 模式 2

模式 2 配置定时器寄存器为一个自动加载的 8 位计数器(TLx)。TLx 溢出不仅置位 TFx，而且也将 THx 的内容加载到 TLx，THx 内容由软件预置，加载不会改变 THx 的值。定时器 0 和 1 的模式 2 操作是一样的。定时器 0/1 模式 2 的结构图见图 16-5 和图 16-6。

图 16-5. 定时器 0 模式 2 结构

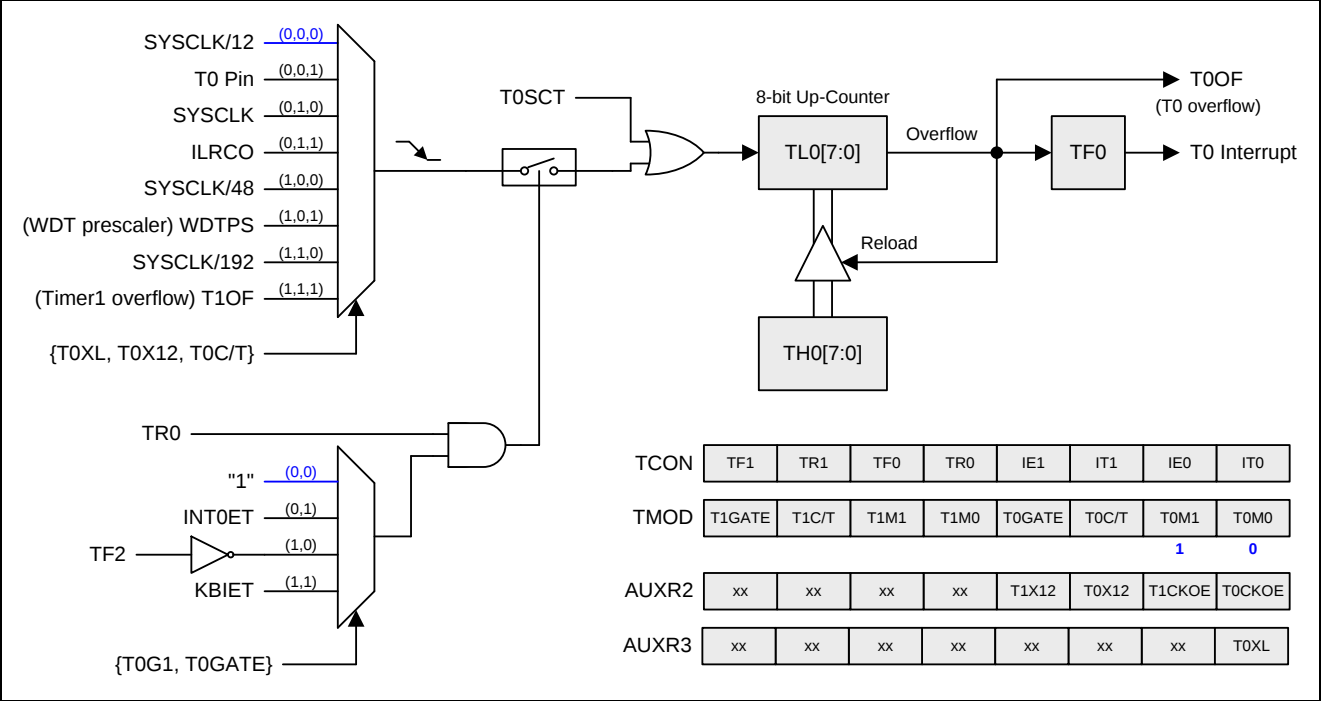
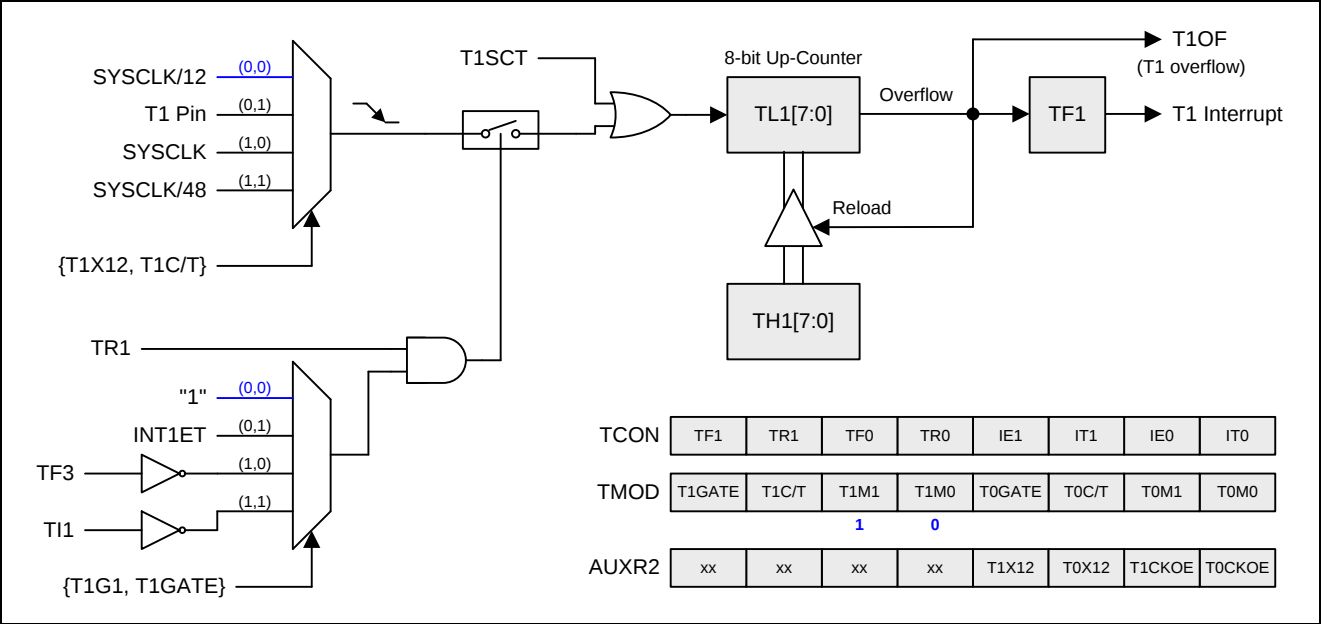


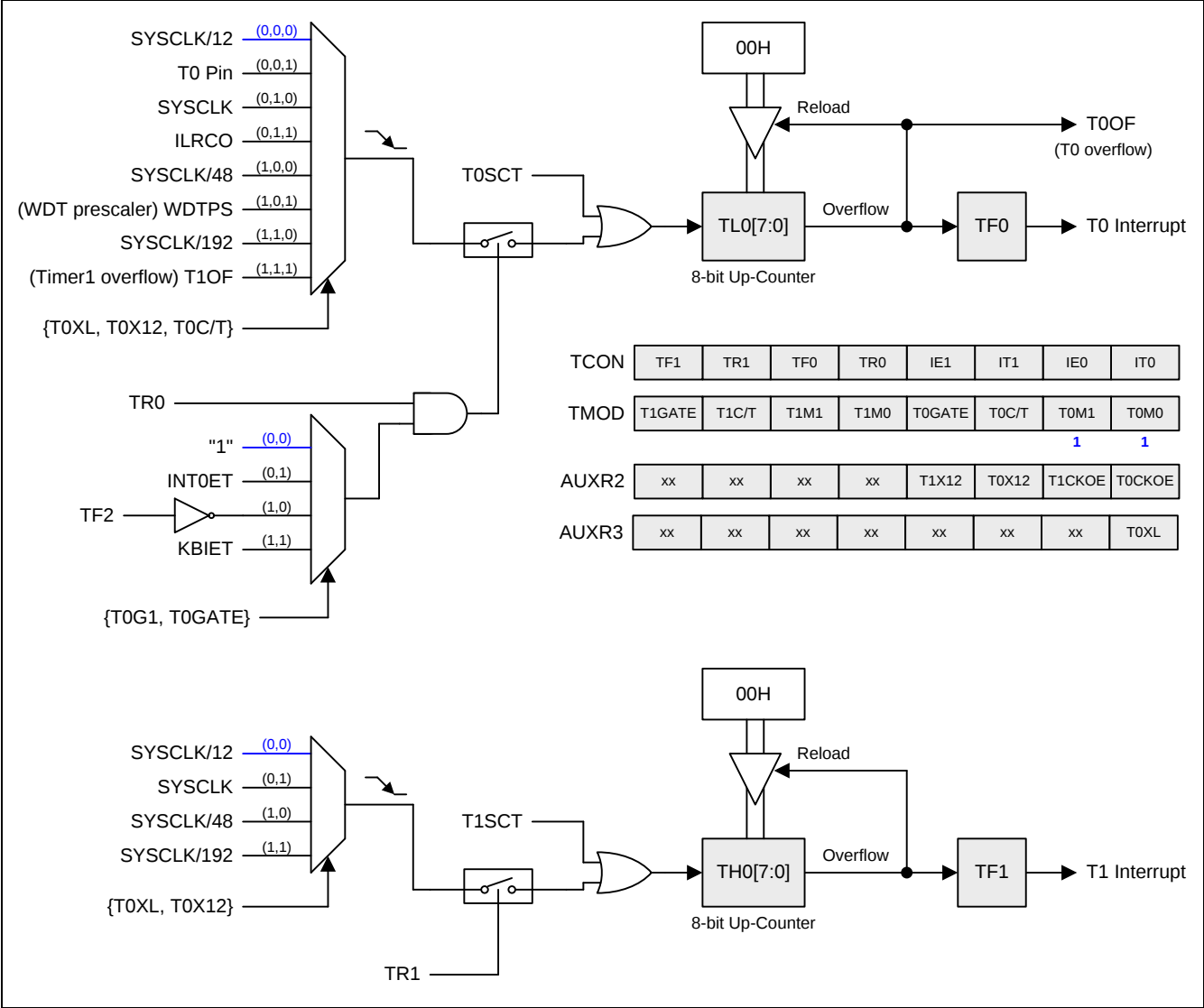
图 16-6. 定时器 1 模式 2 结构



16.1.4. 定时器 0/1 模式 3

定时器 1 在模式 3 保持计数值，效果和设置 TR1=1 一样。定时器 0 在模式 3 建立 TL0 和 TH0 两个独立的计数器。TL0 使用定时器 0 控制位：T0XL、T0X12、T0C /T、T0GATE、TR0 和 TF0。TH0 锁定为定时器功能(每个机器周期计数)且接替定时器 1 来使用 TR1 和 TF1，因从 TH0 控制定时器 1 中断。定时器 0 模式 3 的结构图见图 16-7。

图 16-7. 定时器 0 模式 3 结构



16.1.5. 定时器 0/1 可编程时钟输出

定时器 0 和 1 有一个时钟输出模式(当 TxCKOE=1)。此模式下，定时器 0 或 1 操作在 8 位自动重载占空比为 1:1 的可编程时钟发生器。产生的时钟在 P3.4 (T0CKO)和 P3.5 (T1CKO)独立输出。8 位定时器(TL0)每个输入时钟加一，在定时器 0 模块。8 位定时器(TL1)每个输入时钟加一，在定时器 1 模块。定时器从载入值到溢出重复计数。一旦溢出，(TH0,TH1)的值被载入到(TL0, TL1)同时计数。图 16-8 和图 16-9 给出了定时器 0/1 时钟输出频率公式。图 16-10 和图 16-11 给出了定时器 0 时钟输出结构和输出控制。

图 16-12 给出了定时器 1 时钟输出结构。

图 16-8. 定时器 0 时钟输出公式

$$T0 \text{ Clock-out Frequency} = \frac{T0 \text{ Clock Frequency}}{2 \times (256 - TH0)}$$

图 16-9. 定时器 1 时钟输出公式

$$T1 \text{ Clock-out Frequency} = \frac{T1 \text{ Clock Frequency}}{2 \times (256 - TH1)}$$

- 注意:
- (1) 定时器 0/1 溢出标志 TF0/1，在定时器 0/1 溢出时置位。
 - (2) 当 SYSCLK=12MHz 及选择 SYSCLK/12 为定时器 0/1 的时钟源，定时器 0/1 可编程输出频率范围从 1.95KHz 到 500KHz。
 - (3) 当 SYSCLK=12MHz 及选择 SYSCLK 为定时器 0/1 的时钟源，定时器 0/1 可编程输出频率范围从 23.44KHz 到 6MHz。

图 16-10. 定时器 0 时钟输出模式

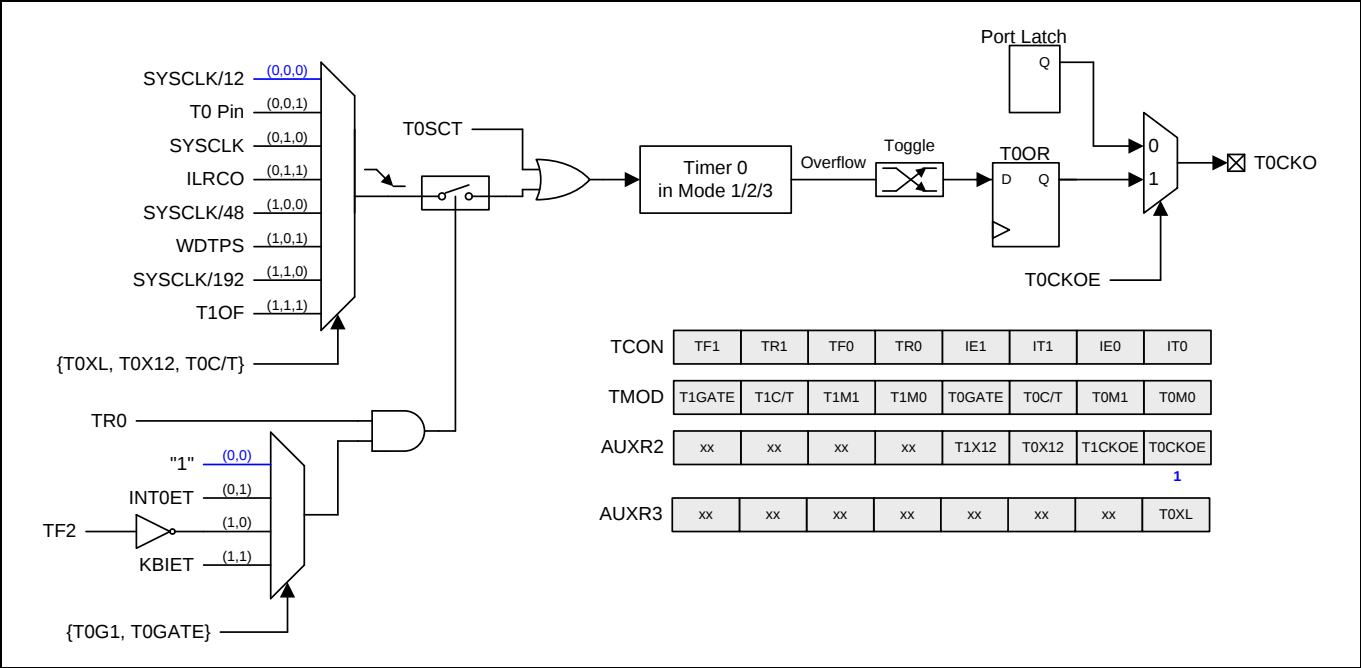


图 16-11. 定时器 0 时钟输出控制

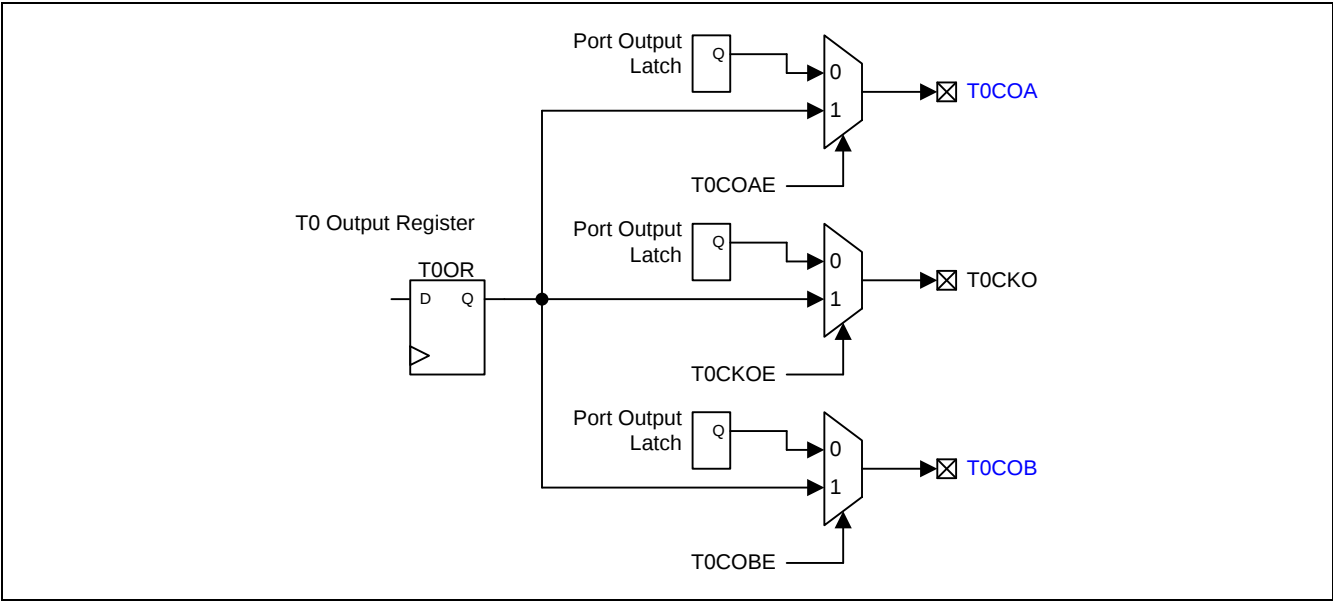
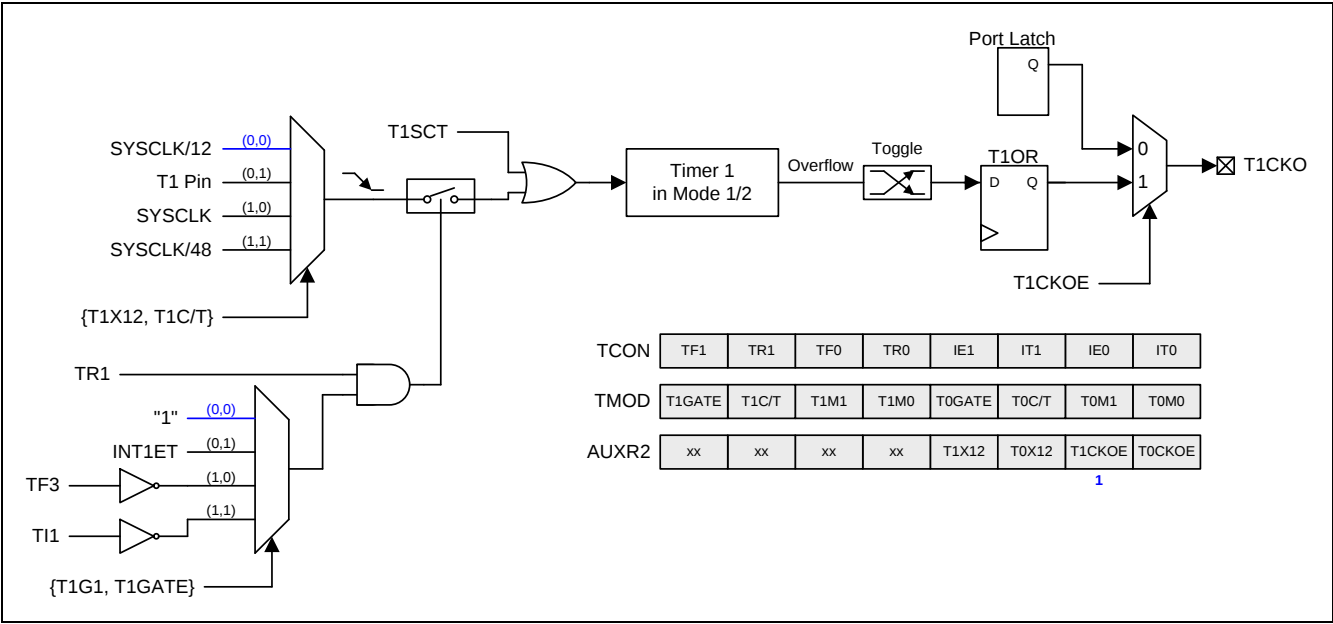


图 16-12. 定时器 1 时钟输出模式



定时器 0/1 时钟输出模式如何编程

- 选择定时器 0/1 的时钟源
- 从公式计算出 8 位自动加载值并输入到 TH0/TH1 寄存器.
- 在 TL0/TL1 寄存器输入一个跟自动加载值相同 8 位初始值.
- AUXR0 寄存器的 T0CKOE/T1CKOE 置位.
- 设置 TCON 寄存器的 TR0/TR1 位启动定时器 0/1.

时钟输出模式，定时器 0/1 溢出不会中断。这跟定时器 1 被用作波特率发生器相似。定时器 1 即可用作波特率发生器也可同时用作时钟发生器。注意，波特率和时钟输出频率都是相同的定时器 1 溢出率。因此在这类应用中软件通常禁止定时器 0/1 中断。

16.1.6. 定时器 0/1 寄存器

TCON: 定时/计数器控制寄存器

SFR 页 = 0~F

SFR 地址 = 0x88

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TF1, 定时器 1 溢出标志。

0: 处理器进入中断向量程序由硬件清零, 或软件清零。

1: 定时器/计数器 1 溢出时由硬件置位, 或软件置位。

Bit 6: TR1, 定时器 1 运行控制位。

0: 关闭定时器/计数器 1。

1: 开启定时器/计数器 1。

Bit 5: TF0, 定时器 0 溢出标志。

0: 处理器进入中断向量程序由硬件清零, 或软件清零。

1: 定时器/计数器 0 溢出时由硬件置位, 或软件置位。

Bit 4: TR0, 定时器 0 运行控制位。

0: 关闭定时器/计数器 0。

1: 开启定时器/计数器 0。

TMOD: 定时/计数器模式控制寄存器

SFR 页 = 0~F

SFR 地址 = 0x89

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

|←----- Timer1 ----->|←----- Timer0 ----->|

Bit 7: T1GATE, 定时器 1 门控制位。

T1G1, T1GATE	T1 门控源
0 0	禁止
0 1	INT1
1 0	TF3
1 1	TI1

Bit 6: T1C/T, 定时器 1 时钟源选择位。控制着 4 种时钟源的定时器 1 作为定时器或计数器。详情参见 AUXR2.T1X12 的描述。

Bit 5~4: 定时器 1 工作模式选择

T1M1	T1M0	定时器 1 工作模式
0	0	8-Bit PWM 产生器
0	1	16-Bit 定时器/计数器
1	0	8-Bit 自动重载定时器/计数器
1	1	(定时器 1) 定时器/计数器停止

Bit 3: T0GATE, 定时器 0 门控制位

T0G1, T0GATE	T0 门控源
0 0	禁止
0 1	INT0
1 0	TF2
1 1	KBI

Bit 2: T0C/T, 定时器 0 时钟源选择位。控制着 8 种时钟源的定时器 0 作为定时器或计数器。详情参见 AUXR2.T0X12 的描述。

Bit 1~0: 定时器 0 工作模式选择.

T0M1	T0M0	定时器 0 工作模式
0	0	8-Bit PWM 产生器
0	1	16-Bit 定时器/计数器
1	0	8-Bit 自动重载定时器/计数器
1	1	TL0 是 8 位定时器/计数器, TH0 锁定 8 位定时器

TL0: 定时器 0 低字节寄存器

SFR 页 = 0~F

SFR 地址 = 0x8A

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH0: 定时器 0 高字节寄存器

SFR 页 = 0~F

SFR 地址 = 0x8C

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TL1: 定时器 1 低字节寄存器

SFR 页 = 0~F

SFR 地址 = 0x8B

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH1: 定时器 1 高字节寄存器

SFR 页 = 0~F

SFR 地址 = 0x8D

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

AUXR2: 辅助寄存器 2

SFR 页 = 0~F

SFR 地址 = 0xA3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
STAF	STOF	C1PLK	COPLK	T1X12	T0X12	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: T1X12, 与 T1C/T 一起控制定时器 1 时钟源选择

T1X12, T1C/T	定时器 1 时钟源
0 0	SYSCLK/12
0 1	T1 引脚
1 0	SYSCLK
1 1	SYSCLK/48

Bit 2: T0X12, T 与 T0C/T 和 T0XL 一起控制定时器 0 时钟源选择

T0XL, T0X12, T0C/T	定时器 0 时钟源
0 0 0	SYSCLK/12
0 0 1	T0 引脚
0 1 0	SYSCLK
0 1 1	ILRCO

1 0 0	SYSCLK/48
1 0 1	WDTPS
1 1 0	SYSCLK/192
1 1 1	T1OF

AUXR3: 辅助寄存器 3

SFR 页 = 仅 0 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: T0PS1~0, 定时器 0 端口引脚选择位 [1:0].

T0PS1~0	T0/T0CKO
0 0	P3.4
0 1	P4.4
1 0	P4.6
1 1	P1.7

Bit 0: T0XL 是定时器 0 预分频控制位。T0XL 功能定义请参考 T0X12 (AUXR2.2)

AUXR4: 辅助寄存器 4

SFR 页 = 仅 1 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	0	AC0OE	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: T1PS1~0, 定时器 1 端口引脚选择位[1:0].

T1PS1~0	T1/T1CKO
0 0	P3.5
0 1	P4.5
1 0	P1.7
1 1	P3.3

AUXR9: 辅助寄存器 9

SFR 页 = 仅 6 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G1	T0G1	0	0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: T1G1, 定时器 1 门控源选择

T1G1, T1GATE	T1 门控源
00	禁止
01	INT1
10	TF3
11	TI1

Bit 4: T0G1, 定时器 0 门控源选择

T0G1, T0GATE	T0 门控源
00	禁止
01	INT0
10	TF2
11	KBI

AUXR11: 辅助寄存器 11

SFR 页 = 仅 8 页
SFR 地址 = 0xA4
复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	I2C1PS1	I2C1PS0	RX1S0	0	T1CKOE	T0CKOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1: T1CKOE, 定时器 1 时钟输出使能.
0: 禁止定时器 1 时钟输出
1: 使能定时器 1 时钟输出到 T1CKO 端口引脚.

Bit 0: T0CKOE, 定时器 0 时钟输出使能.
0: 禁止定时器 0 时钟输出
1: 使能定时器 0 时钟输出到 T0CKO 端口引脚.

16.2. 定时器 2

定时器 2 是一个 16 位定时器/计数器，既可作为一个定时器也可以作为一个事件计数器，由 T2CKS,T2X12,C/T2 选择。定时器 2 有几种通过 T2CON,T2MOD 和 T2MOD1 相关位定义的操作模式：捕捉、自动加载定时器(向上计数)、8 位 PWM、波特率发生器和可编程时钟输出。

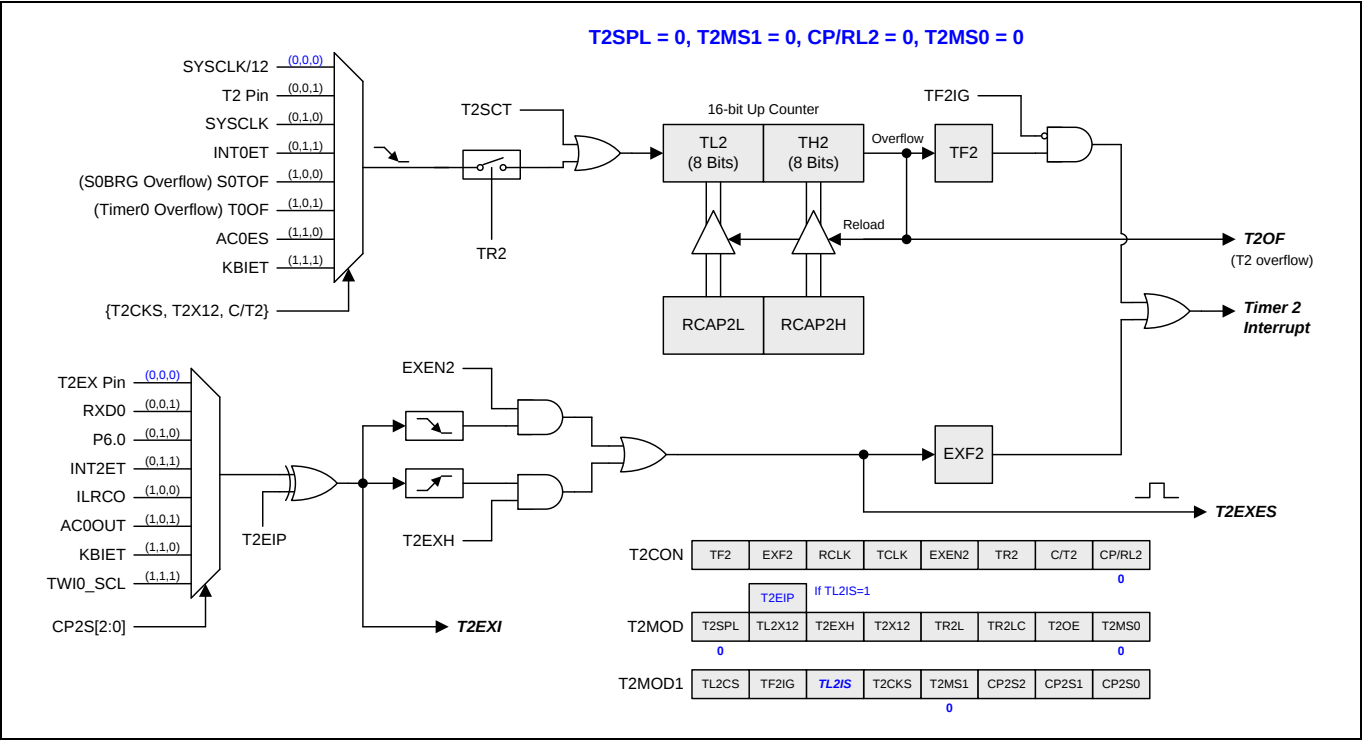
16.2.1. 定时器 2 模式 0(自动重载和外部中断)

在这个模式中，定时器 2 提供一个 16 位的自动重载定时器/计数器。TF2 是定时器 2 的溢出标志，是一个中断会被 TF2IG 阻断的定时器 2 中断源。EXEN2 使能 T2EXI 引脚的下降沿置位 EXF2，EXF2 可作为一个外部中断与 TF2 共享定时器 2 中断。T2EXI 由 8 个外部触发输入中选出一个。T2EXH 的功能与 EXEN2 一样，只是 T2EXH 使能 T2EXI 引脚的上升沿置位 EXF2。

本模块中的定时器 2 溢出事件(T2OF)可以作为时钟输入或事件源输出到其它外设。

定时器 2 模式 0 如图 16-13 所示

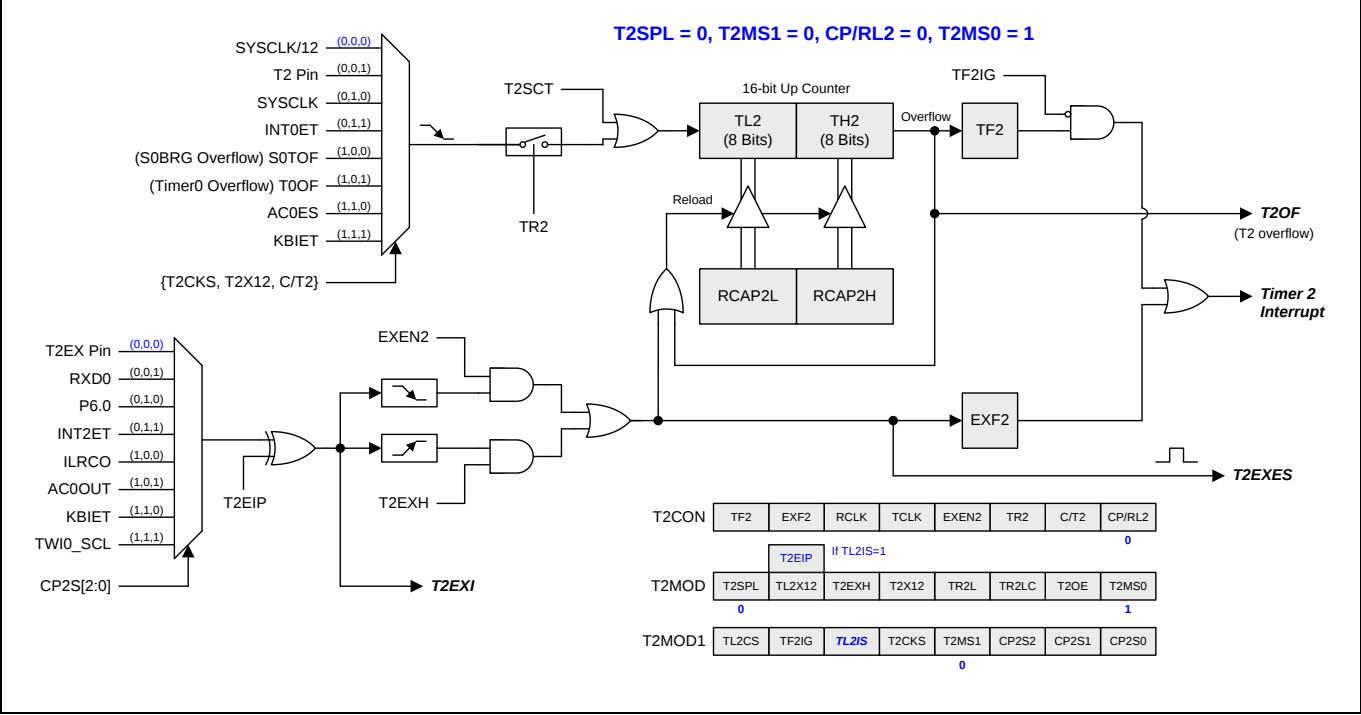
图 16-13. 时器 2 模式 0 结构(自动重载和外部中断模式)



16.2.2. 定时器 2 模式 1(外部中断触发自动重载)

如图 16-14 所示，定时器 2 模式 1，使能定时器 2 自动向上计数。本模式有 T2CON 寄存器的 EXEN2 决定的两种选择。如果 EXEN2=0，定时器 2 向上计数到 0xFFFFH 且直到溢出而置位 TF2(溢出标志)。这时定时器 2 寄存器被重载入 RCAP2L 和 RCAP2H 的 16 位数据。RCAP2L 和 RCAP2H 的值由软件预设。如果 EXEN2=1，16 位重载会被一个 T2 溢出或一个 T2EXI 的下降沿触发，T2EXI 由 8 个外部触发输入中选出，此触发同时置位 EXF2。如果定时器 2 中断使能，无论 TF2 或 EXF2 置位则产生中断。T2EXH 的功能与 EXEN2 一样，只是 T2EXH 使能 T2EXI 引脚的上升沿置位 EXF2。

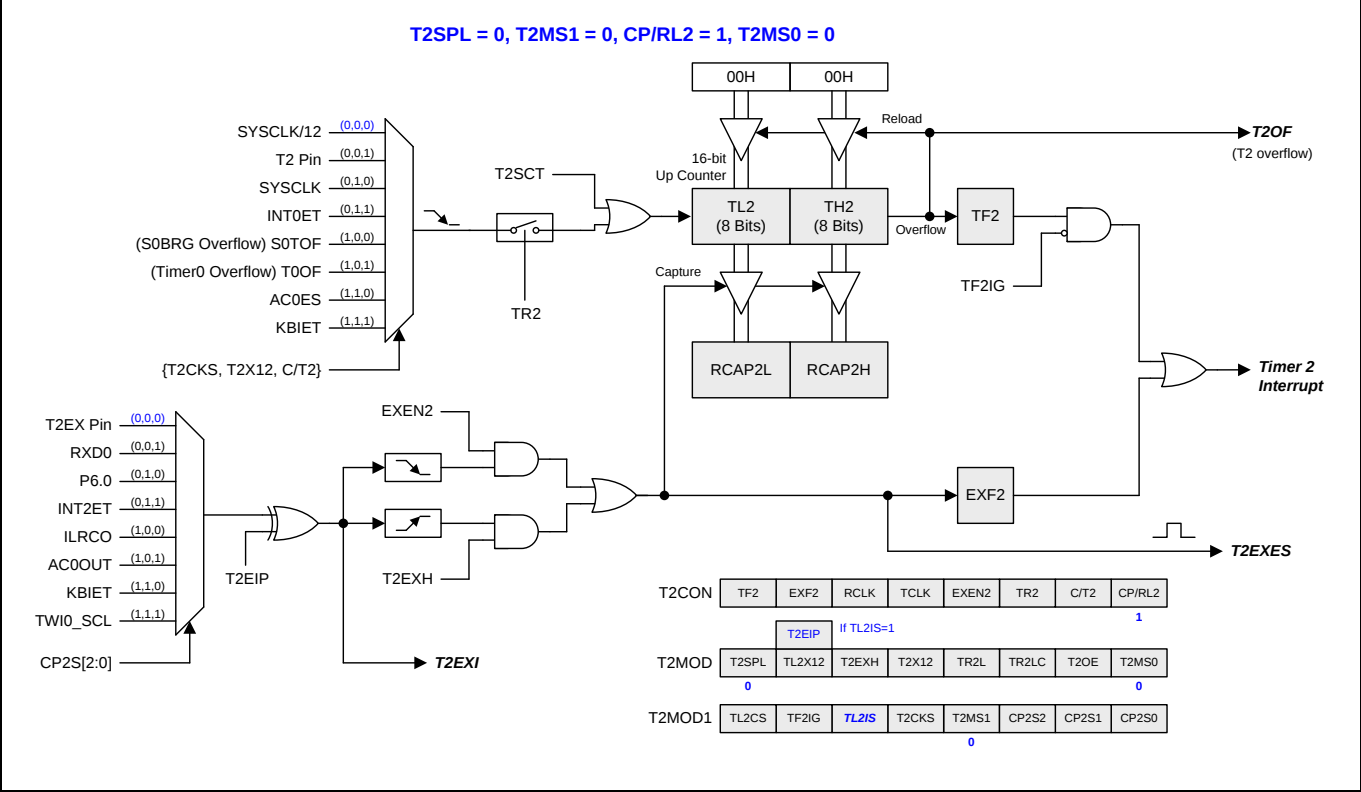
图 16-14. 定时器 2 模式 1 结构(外部中断触发自动重载模式)



16.2.3. 定时器 2 模式 2(捕捉)

由 T2CON 寄存器的位 EXEN2 做二种选择的捕捉模式如图 16-15 所示。如果 EXEN2=0，定时器 2 是一个 16 位定时器或计数器，向上溢出而置位 TF2(定时器 2 溢出标志)。TF2 常用来产生中断(IE 寄存器使能定时器 2 中断相关位)。模式 2 增加了捕捉功能，如果 EXEN2=1，T2EXI 引脚下降沿(由 8 个外部触发输入中选出)把定时器 2 寄存器 (TH2 和 TL2)各自捕捉到寄存器(RCAP2H 和 RCAP2L)。另外，T2EXI 引脚跳变使 T2CON 寄存器的 EXF2 置位，且 EXF2 跟 TF2 共享中断位置。T2EXH 的功能与 EXEN2 一样，只是 T2EXH 使能 T2EXI 引脚的上升沿置位 EXF2。

图 16-15. 定时器 2 模式 2 结构(捕捉模式)

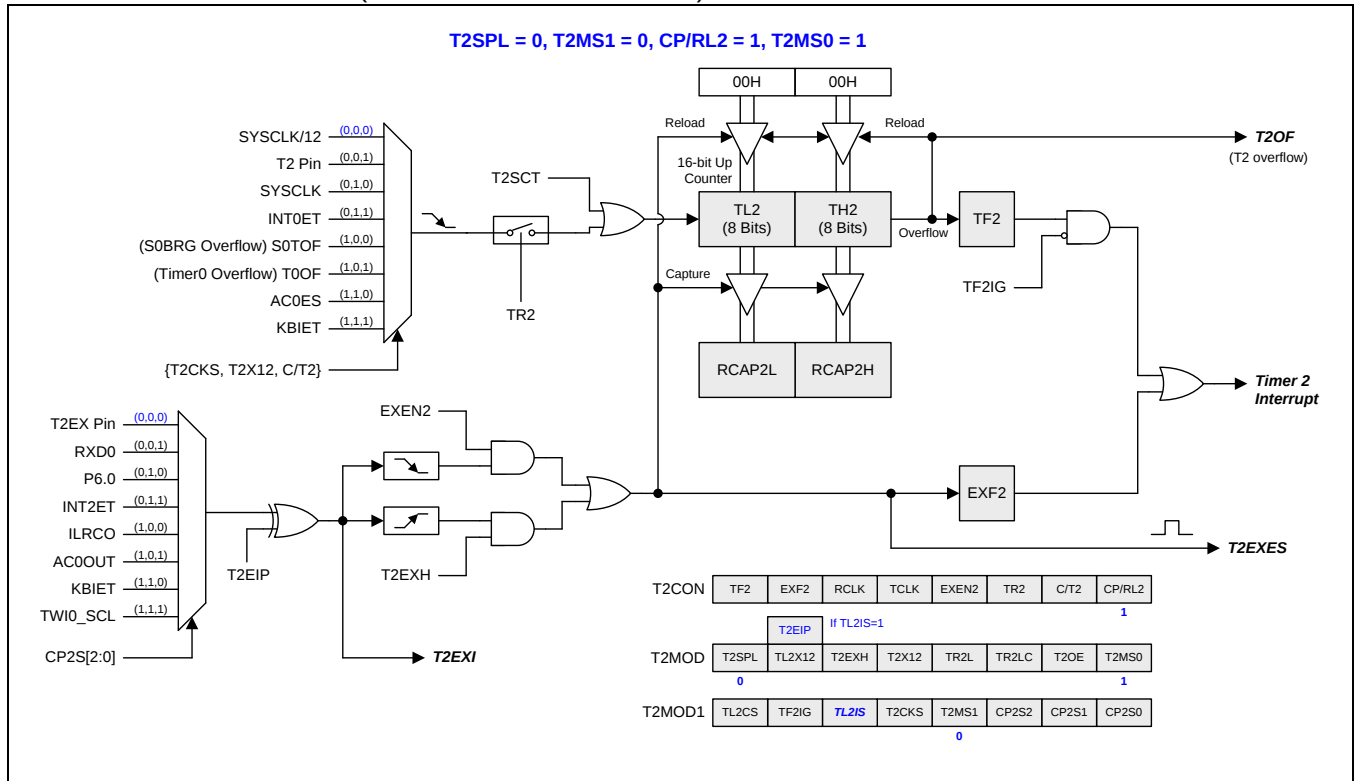


16.2.4. 定时器 2 模式 3(捕捉带自动清零)

定时器 2 模式 3 与定时器 2 模式 2 的功能相似。有一点不同的就是 T2EXES、EXF2 置位，不但定时器 2 会被捕捉而且 TL2 和 TH2 也会被清零。

定时器 2 模式 3 如图 16-16 所示。

图 16-16. 定时器 2 模式 3 结构(捕捉带自动清零 TL2 和 TH2)



16.2.5. 定时器 2 模式 6 (占空比捕捉)

定时器 2 模式 6 支持捕捉输入波形的周期时间或占空比。信号的三个边沿可以计算出周期和占空比。在占空比捕捉模式，需要将 TH2, TL2 清零。然后通过置位 TR2 开启捕捉模式，但是计数器还没有启动，它会等到第一个边缘进入到外部触发通道，例如 T2EX 引脚。这表示第一个边值是 00H。在第一次触发边缘后，计数器开始计数。请注意，T2EXI 的第一个触发边缘必须是上升沿。即使你设置了 T2EXH，第一个边沿也会被忽略以触发 EXF2。

其次，如果只想计算脉冲宽度，那么可以设置 EXEN2 通过第二个边来触发 EXF2。在本例中，当第二个边缘触发计时器将其值捕捉到 RCAP2H: RCAP2L 中时，它还触发用于中断的 EXF2，以标识它已经完成。

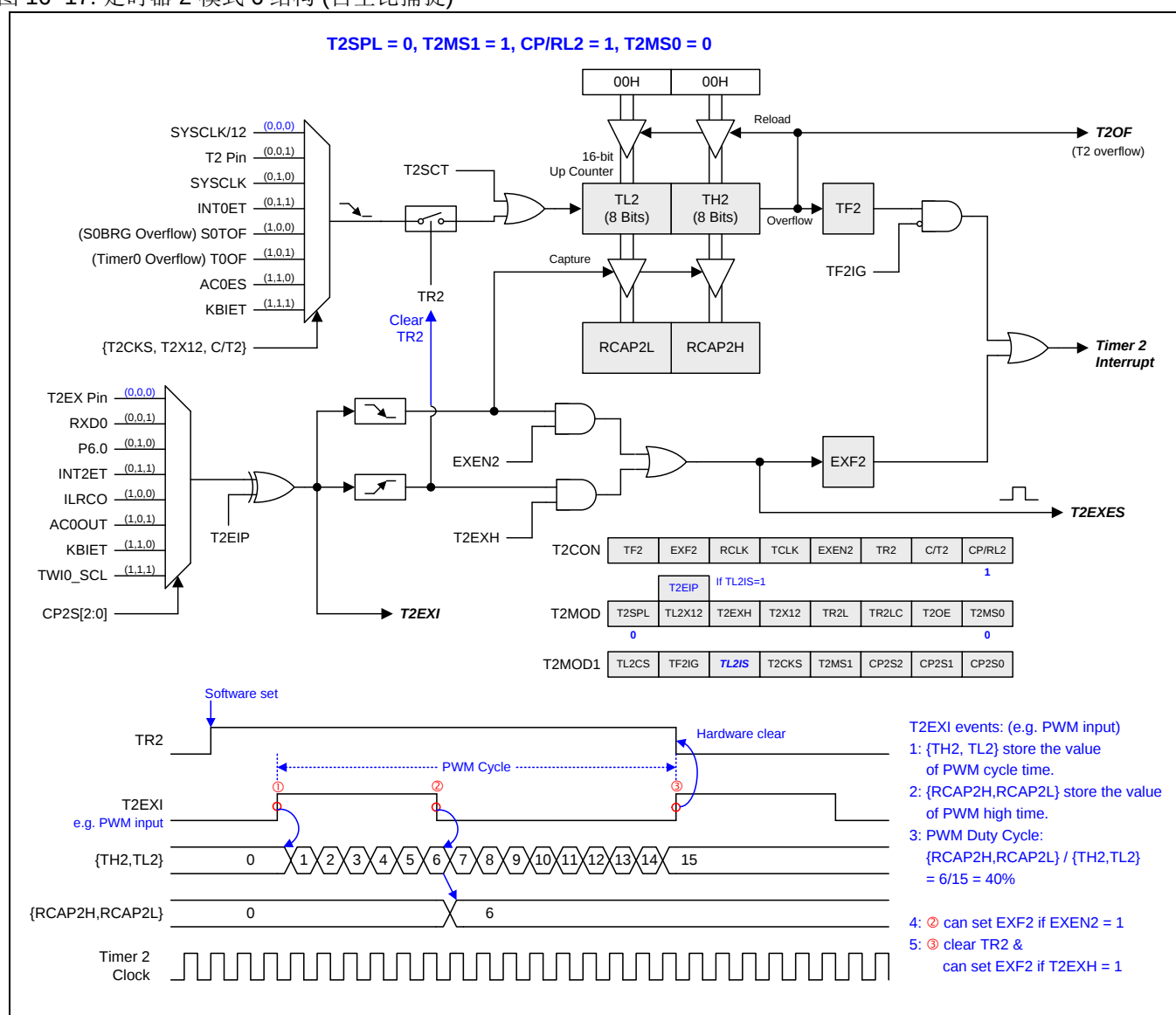
但是如果你想获得周期时间，那么就要通过清除 EXEN2 去阻止第二边沿。

当第三个边沿到来，它将自动清除 TR2 来停止计数器。

使用 TH2: TL2(第三条边)、RCAP2H: RCAP2L(第二条边)和 0 (第一条边)来计算周期时间和占空比。

定时器 2 模式 6 如 图 16-17 所示

图 16-17. 定时器 2 模式 6 结构 (占空比捕捉)



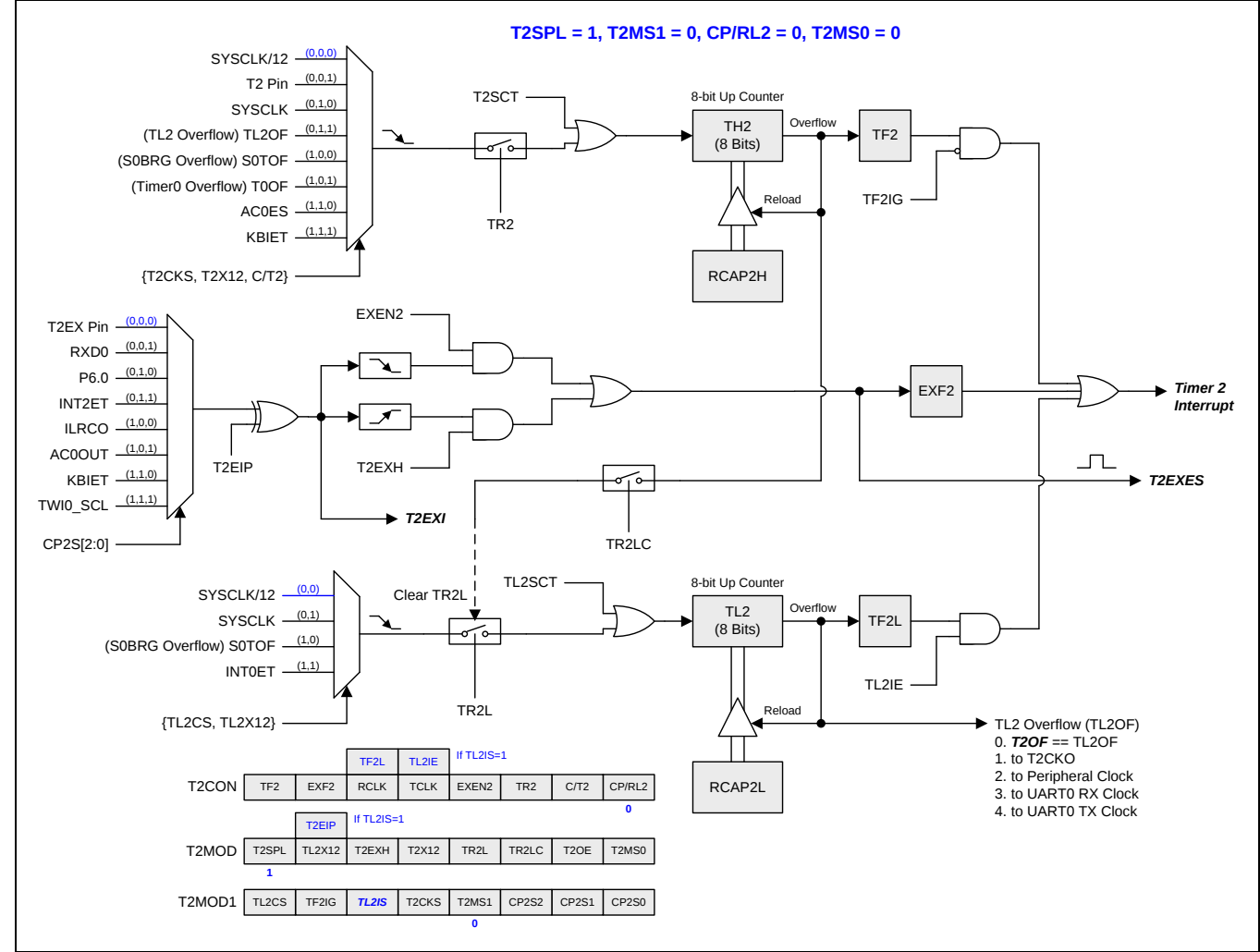
16.2.6. 分立定时器 2 模式 0 (自动重载和外部中断)

本模式中 T2SPLIT 置位，定时器 2 变为两个 8 位定时器(TH2 和 TL2)。两个 8 位定时器都是向上计数如图 16-18 所示。TH2 保存 RCAP2H 的重载值和保持 16 位模式一样的 8 个时钟源输入选择。8 位定时器功能跟 16 位模式的定时器 2 模式 0 相似。TL2 保存 4 个时钟输入选择的 RCAP2L 重载值。T2CON 的位 TR2 控制着 TH2 的运行。T2MOD 的位 TR2L 控制着 TL2 的运行。当 TR2LC 置位时 TH2 溢出会停止 TR2L 的运行。

分立模式有 3 个中断标志 EXF2、TF2 和 TF2L。EXF2 与 16 位模式一样的功能用来侦测 T2EXI 引脚的跳变。TF2IG 控制 TF2 在 TH2 从 0xFF 到 0x00 溢出时是否置位。TL2 从 0xFF 到 0x00 溢出时 TF2L 置位，TL2IE 使能中断。EXF2、TF2 和 TF2L 中断标志硬件不会清零且必须软件清零。
附带一提，16 位模式中的定时器 2 溢出事件(T2OF)将被分立模式中的 TL2 溢出事件(TL2OF)取代。

如果 T2MOD1 的位 TL2IS=0，位 T2CON.5~4 是 RCLK 和 TCLK 的功能。如果 TL2IS=1，位 T2CON.5~4 是 TF2L 和 TL2IE 的功能。

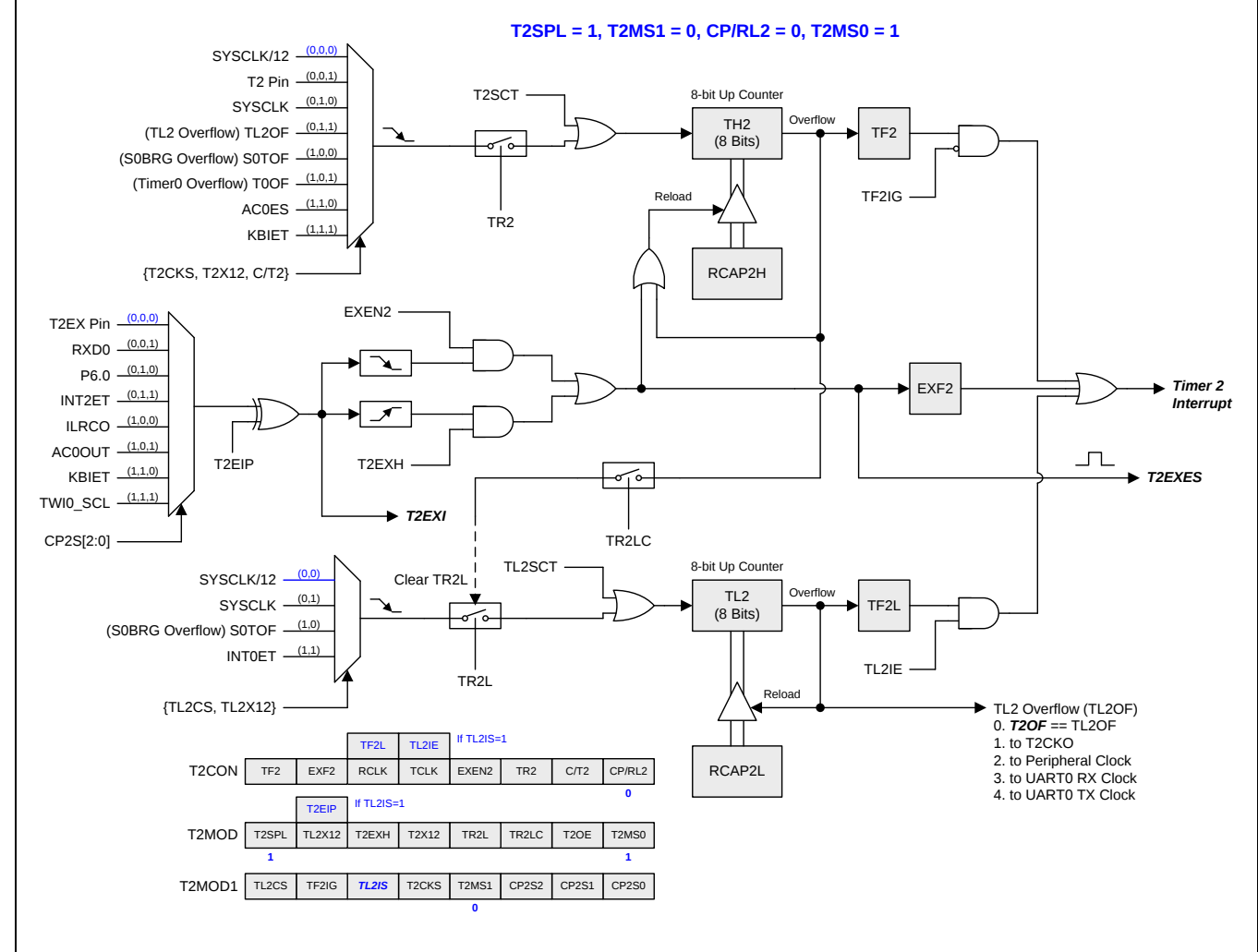
图 16-18. 分立定时器 2 模式 0 结构(自动重载和外部中断)



16.2.7. 分立定时器 2 模式 1 (外部中断触发自动重载)

本模式中 T2SPLIT 置位，定时器 2 变为两个 8 位定时器如图 16-19 所示。跟定时器 2 模式 1 相似的功能且保持与分立定时器 2 模式 0 一样的中断方式。

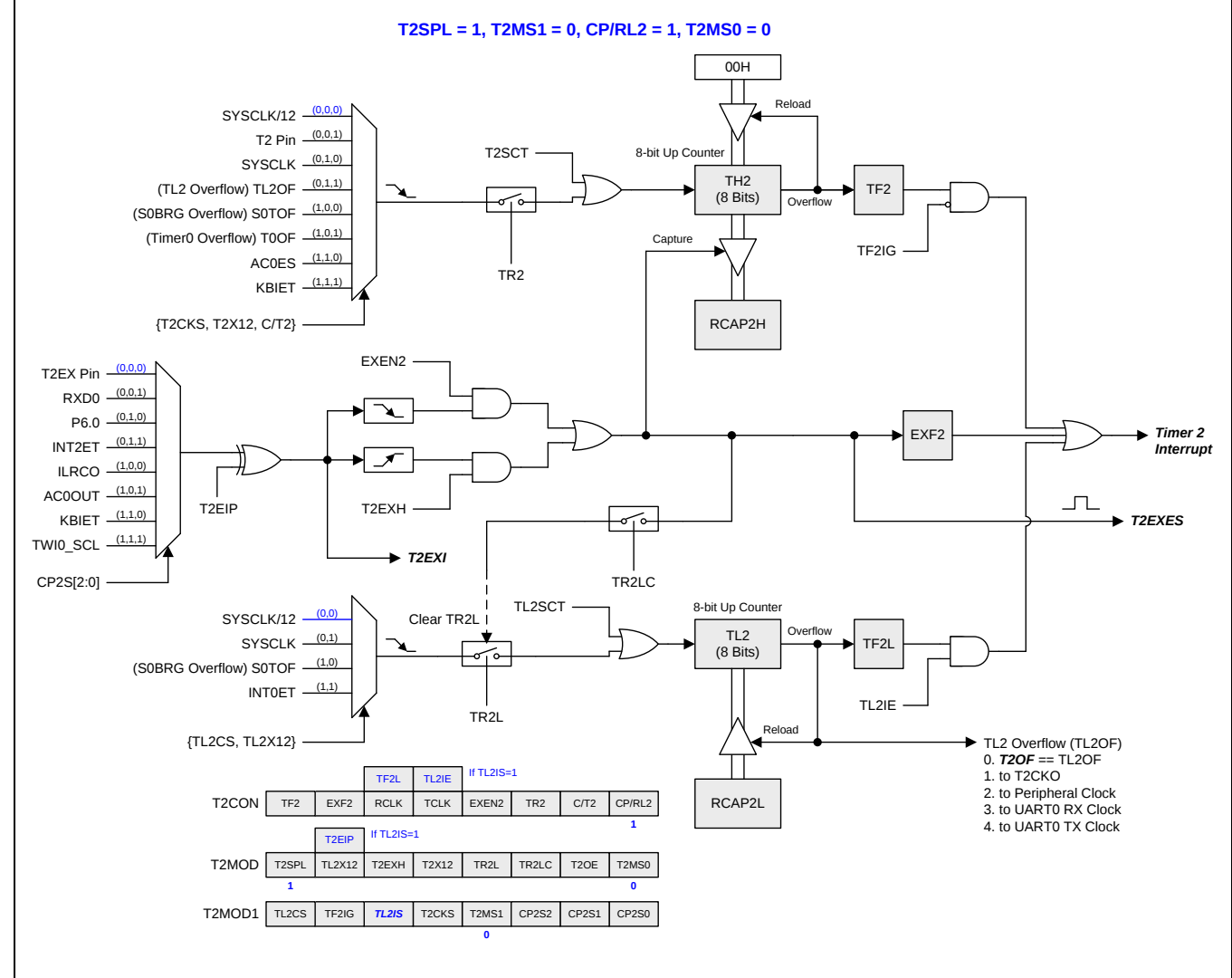
图 16-19. 分立定时器 2 模式 1 结构(外部中断触发自动重载)



16.2.8. 分立定时器 2 模式 2 (捕捉)

本模式中 T2SPLIT 置位，定时器 2 变为两个 8 位定时器如图 16-20 所示。跟定时器 2 模式 2 相似的功能且保持与分立定时器 2 模式 0 一样的中断方式。

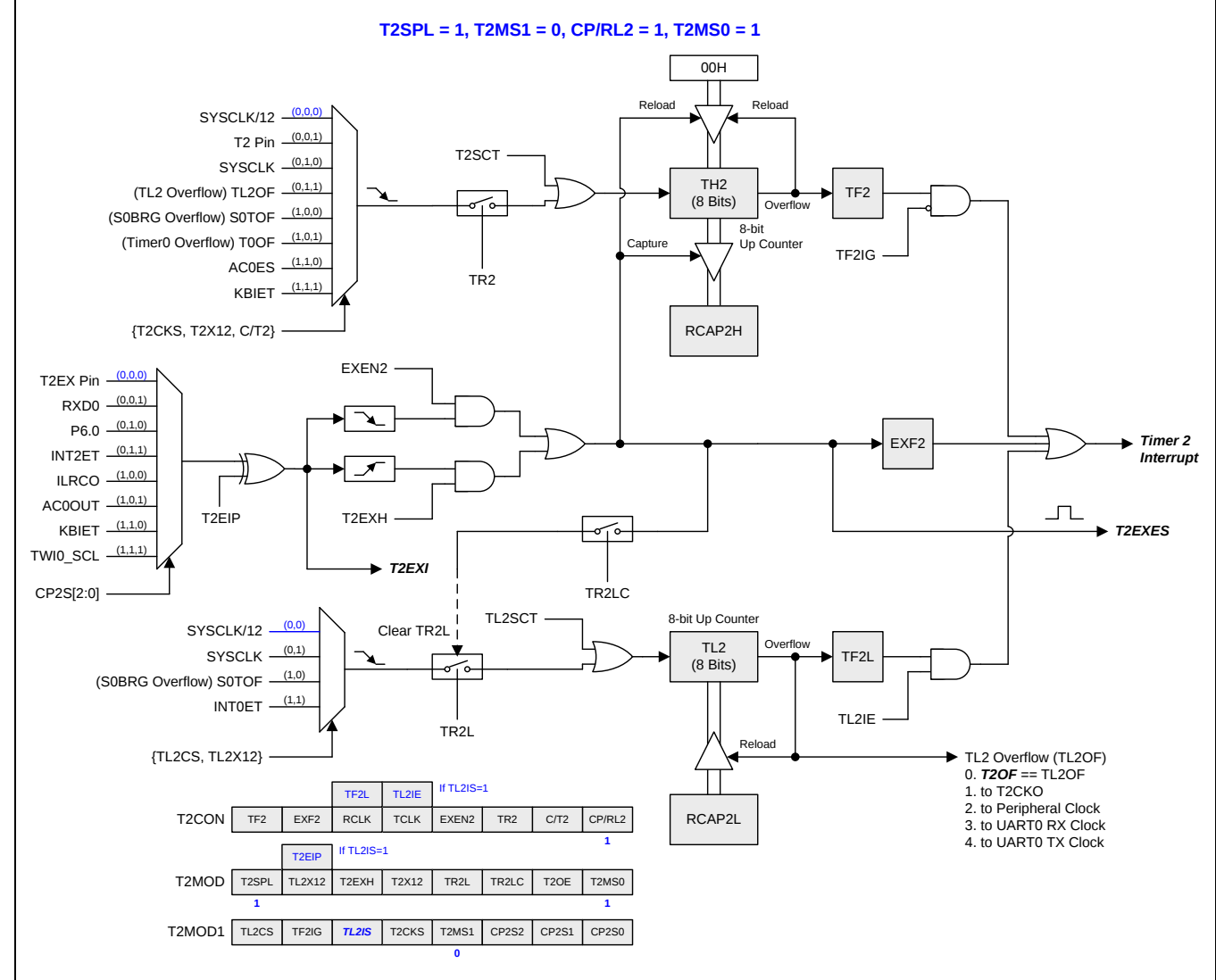
图 16-20. 分立定时器 2 模式 2 结构 (捕捉)



16.2.9. 分立定时器 2 模式 3 (捕捉带自动清零)

本模式中 T2SPLIT 置位，定时器 2 变为两个 8 位定时器如图 16-21 所示。跟定时器 2 模式 3 相似的功能且保持与分立定时器 2 模式 0 一样的中断方式。

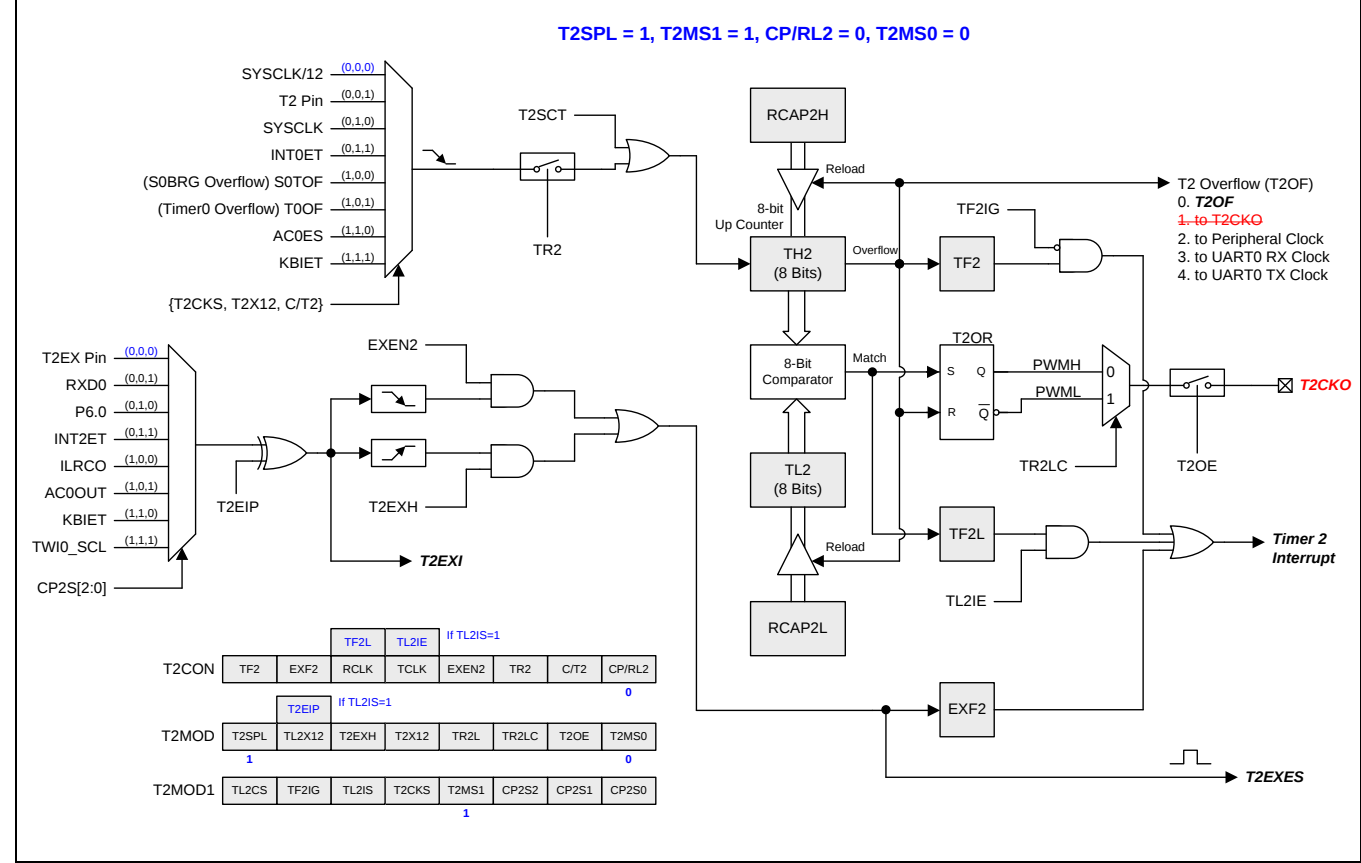
图 16-21. 分立定时器 2 模式 3 结构(捕捉带自动清零 TH2)



16.2.10. 分立定时器 2 模式 4 (8 位 PWM 模式)

本模式，定时器 2 是一个 8 位的 PWM 模式如图 16-22 所示。TH2 和 RCAP2H 相结合为一个 8 位的自动重载计数器。这二个寄存器的软件配置决定 PWM 周期。TL2 是 PWM 比较寄存器用来生成 PWM 波形。RCAP2L 是 PWM 缓冲寄存器且在此寄存器中软件更新 PWM 数据。每次 TH2 溢出事件置位 TF2 且 RCAP2L 值载入到 TL2。PWM 信号输出到 T2CKO 功能引脚且输出的开关由 T2MOD 寄存器的位 T2OE 决定。

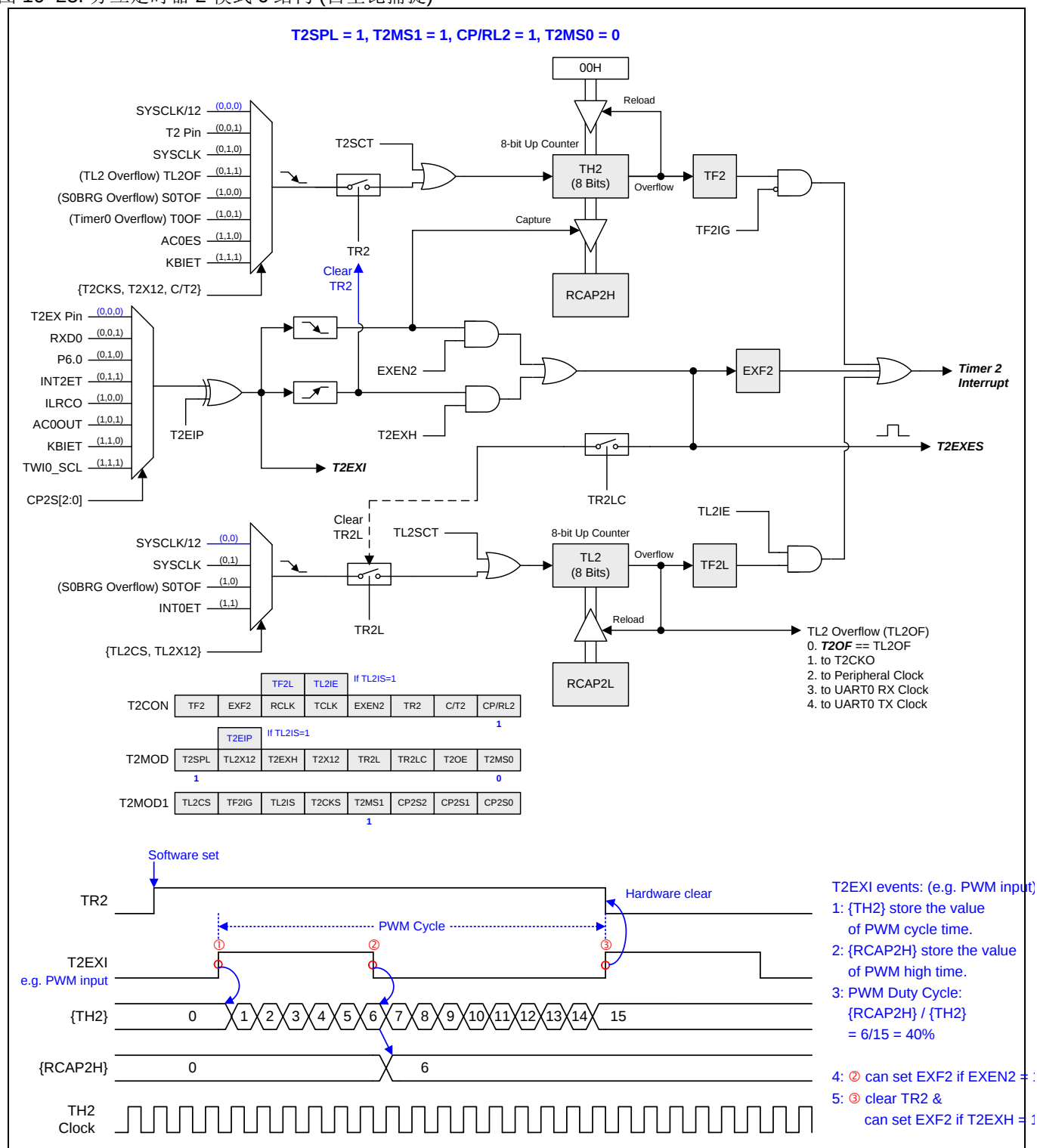
图 16-22. 分立定时器 2 模式 4 结构(8 位 PWM 模式)



16.2.11. 分立定时器 2 模式 6 (占空比捕捉)

当 T2SPL 设置在这种模式下,定时器 2 是分成两个 8 位计时器, 如图 16-23.所示。类似的功能为与定时器 2 模式 6 有着类似的功能, 并且保持同样的中断机制。

图 16-23. 分立定时器 2 模式 6 结构 (占空比捕捉)



16.2.12. 波特率发生器模式 (BRG)

如果定时器 2 工作在模式 0，T2CON 寄存器的 RCLK 和 TCLK 位允许串口发送和接收波特率源可选择定时器 1 或定时器 2。当 TCLK=0 时，定时器 1 作为串口传送波特率发生器。当 TCLK=1，定时器 2 作为串口传送波特率发生器。RCLK 对串口接收波特率有相同的功能。有了这两位，串口可以有不同的接收和发送波特率，一个通过定时器 1 来产生，另一个通过定时器 2 来产生。

图 16-24 所示定时器 2 在波特率发生器模式 UART 引擎产生 RX 和 TX 时钟(见图 19-6.)。波特率发生器模式像自动加载模式，翻转时将把寄存器 RCAP2H 和 RCAP2L 的值加载到定时器 2 的寄存器，RCAP2H 和 RCAP2L 的值由软件预置。

定时器 2 作为波特率发生器只有在 T2CON 寄存器的位 RCLK=1 和/或 TCLK=1 时有效。注意 TH2 翻转会置位 TF2，也不会产生中断。因而，当定时器 2 在波特率发生器模式时定时器中断不需要禁止。如果 EXEN2(T2 外部中断使能位)置位，T2EX(8 个定时器 2 触发输入中的一个)的负跳变将置位 EXF2(T2 外部标志位)，但是不会引起从 (RCAP2H, RCAP2L)到 (TH2, TL2)的重载。因此，当定时器 2 作为波特率发生器时，如果需要的话，T2EX 也可以作为传统的外部中断。T2EXH 的功能与 EXEN2 一样，只是 T2EXH 使能 T2EXI 引脚的正跳变侦测。

当定时器 2 在波特率发生器模式时，不能试着去读或写 TH2 和 TL2。作为一个波特率发生器，定时器 2 在 1/2 的系统时钟频率或从 T2 引脚的异步时加 1；在这些条件下，读或写操作将会不正确。寄存器 RCAP2 可以读，但是不可以写，因为写和重载重叠并引起写和/或加载错误。在访问定时器 2 或 RCAP2 寄存器之前定时器必须关闭(清零 TR2)。

注意:
当定时器 2 用作波特率发生器时，参考章节“19.9.4 模式 1&3 波特率”波特率模式 1 和 3 获取波特率设定值

如果定时器 2 在分立模式 0，TL2 和 RCAP2L 相结合为一个 8 位的波特率发生器如图 16-25 所示。TL2 溢出置位 TF2L，TL2IE 使能中断。TH2 和 RCAP2H 充当一个具有定时器 2 中断能力的自动重载定时器/计数器。

图 16-24. 定时器 2 波特率发生器模式

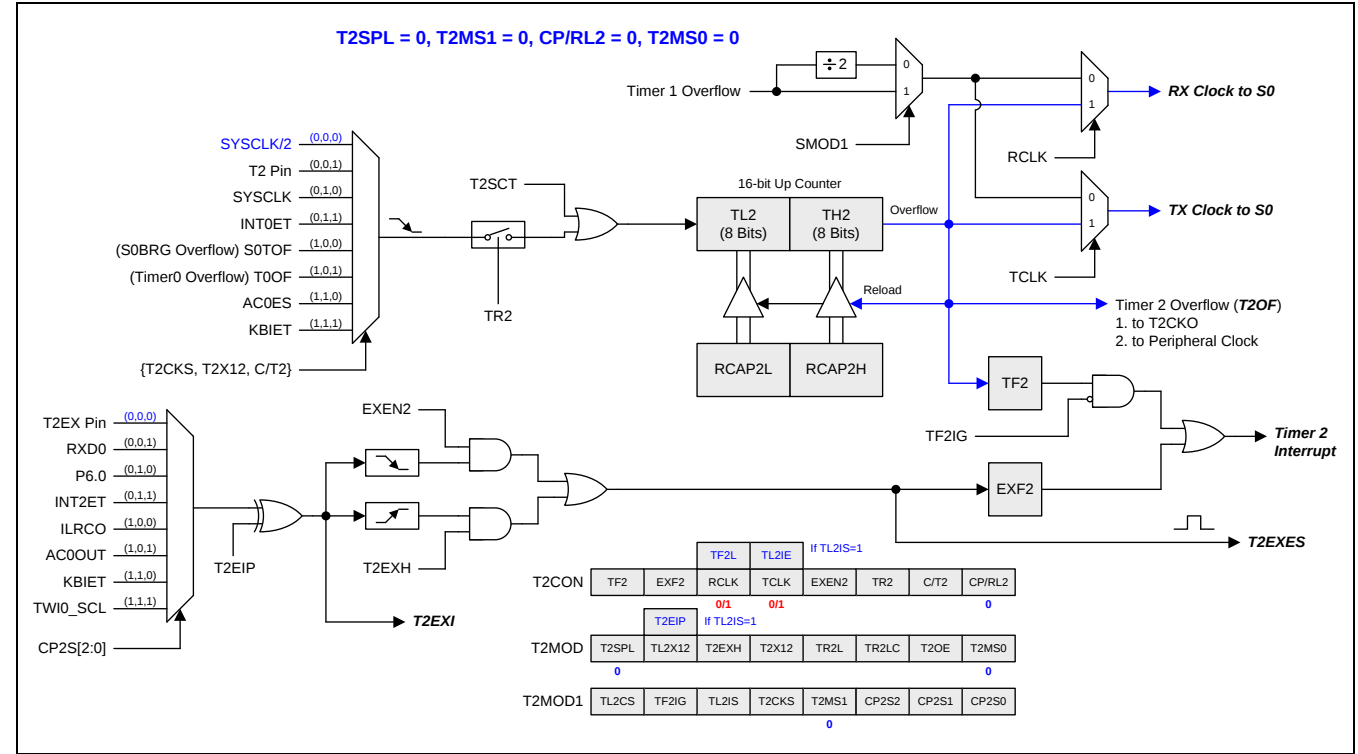
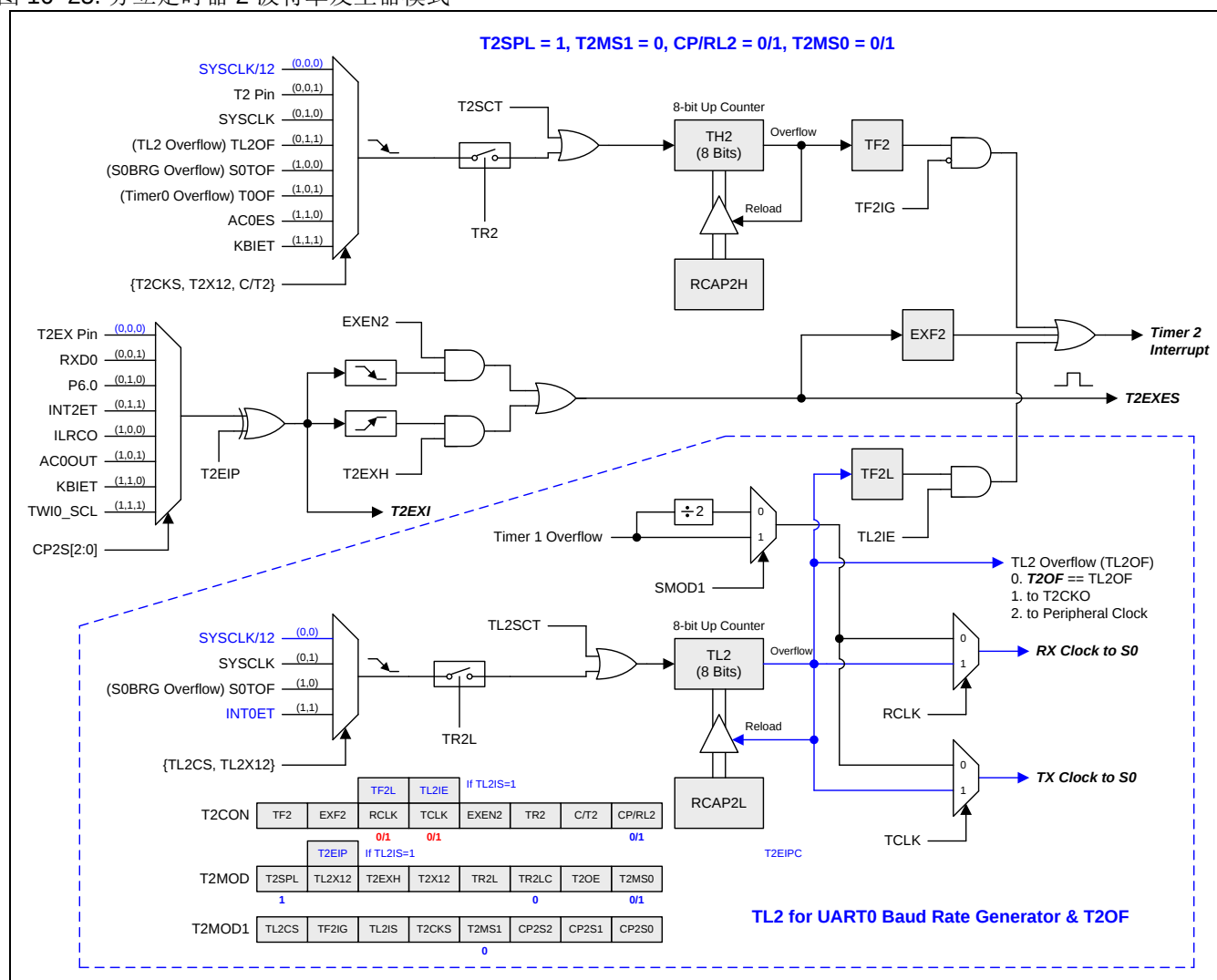


图 16-25. 分立定时器 2 波特率发生器模式



16.2.13. 定时器 2 可编程时钟输出

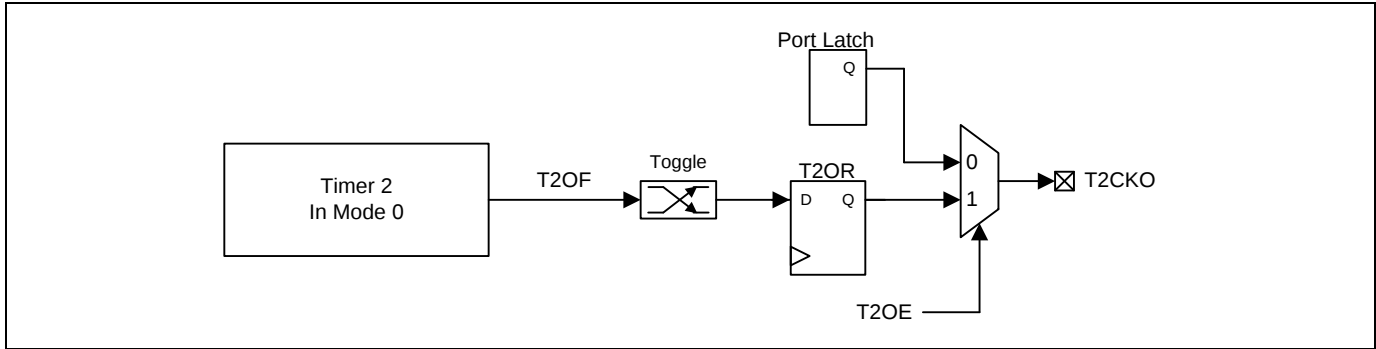
定时器 2 有一个时钟输出模式(当 CP/RL2=0 并且 T2OE=1)。在这个模式，定时器 2 运行为一个占空比为 50%的可编程时钟输出。产生的时钟从 T2CKO 输出。输入时钟((SYSCLK/2,SYSCLK/12 或 SYSCLK)使 16 位定时器 (TH2,TL2)加一。定时器从载入值到溢出重复计数。一旦溢出，(RCAP2H, RCAP2L)的值被载入到(TH2,TL2)同时计数。图 16-26 给出了定时器 2 时钟输出频率计算公式。定时器 2 的时钟输出结构如图 16-27 所示。

图 16-26. 定时器 2 时钟输出公式

$$\text{T2 Clock-out Frequency} = \frac{\text{T2 Clock Frequency}}{2 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))}$$

注意:
(1) 定时器 2 溢出标志 TF2,在定时器 2 溢出时置位可产生中断。但是 TF2 中断会被 T2MOD1 寄存器的位 TF2IG 锁住
(2) 当 SYSCLK=12MHz 及 SYSCLK/12 作为定时器 2 时钟源,定时器 2 可编程输出频率范围从 45.7Hz 到 3MHz。
(3) 当 SYSCLK=12MHz 及 SYSCLK 作为定时器 2 时钟源,定时器 2 可编程输出频率范围从 91.5Hz 到 6MHz。.

图 16-27. 定时器 2 时钟输出模式



定时器 2 时钟输出模式如何编程

- 选择定时器 2 时钟源
- 从公式计算出 16 位自动加载值并输入到 RCAP2H 和 RCAP2L 寄存器.
- 在 TH2 和 TL2 寄存器输入一个跟自动加载值相同的初始值
- T2MOD 寄存器的 T2OE 置位.
- T2CON 寄存器的 TR2 置位启动定时器 2.

在时钟输出模式，定时器 2 翻转产生中断。这和用作波特率发生器时相似。可同时使用定时器 2 作为一个波特率发生器和时钟发生器。注意，波特率和时钟输出都由定时器 2 的溢出速率来决定，中断由 TF2IG 屏蔽。.

如果定时器 2 在分立模式，时钟输出功能由 TL2 溢出产生且输出时钟频率为 TL2 溢出率的二分之一。当 TL2 溢出时 RCAP2L 是 TL2 重载值。TL2 有 4 种时钟源选择。在使能分立定时器 2 时钟输出功能之前，软件必须结束 TL2 时钟源配置。图 16-28 给出了 TL2 时钟输出频率公式。分立定时器 2 的时钟结构如图 16-29 所示。

图 16-28. 分立定时器 2 时钟输出公式

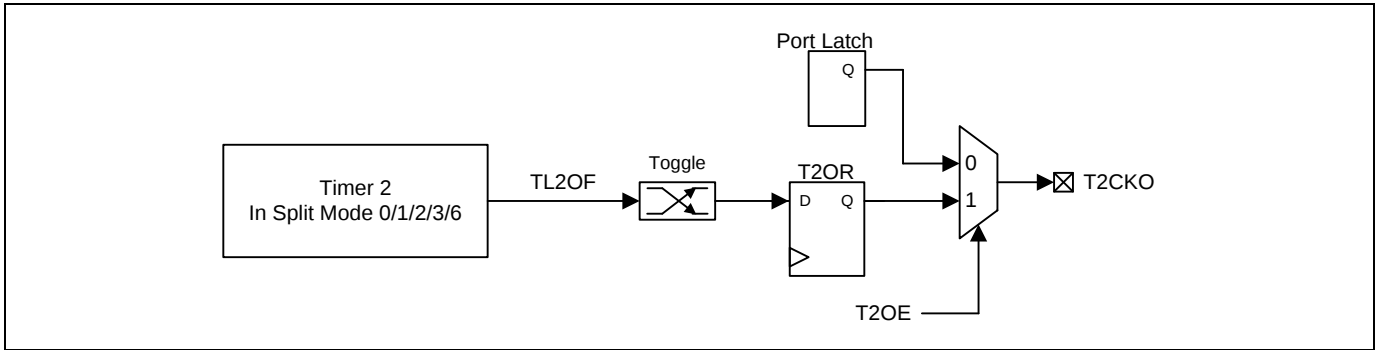
Split T2 Clock-out Frequency =

TL2 Clock Frequency

2 x (256 - RCAP2L)

注意:
(1) TL2 溢出标志 TF2L,在 TL2 溢出时置位产生中断。但是 TF2L 中断由 T2CON 寄存器的位 TL2IE 使能。
(2) 当 SYSCLK=12MHz 及 SYSCLK/12 作为 TL2 时钟源,TL2 可编程输出频率范围从 1.95KHz 到 500KHz。
(3) 当 SYSCLK=12MHz 及 SYSCLK 作为 TL2 时钟源,TL2 可编程输出频率范围从 23.44KHz 到 6MHz。

图 16-29. 分立定时器 2 时钟输出模式



分立定时器 2 时钟输出模式如何编程

- 选择 TL2 时钟源.
- 从公式计算出 8 位自动加载值并输入到 RCAP2L 寄存器.
- 在 TL2 寄存器输入一个跟自动加载值相同的初始值.
- T2MOD 寄存器的 T2OE 置位.
- T2MOD 寄存器的 TR2L 置位启动定时器 2.

在时钟输出模式，TL2 翻转产生中断。这和 TL2 用作波特率发生器时相似。可同时使用 TL2 作为一个波特率发生器和时钟发生器。注意，在分立定时器 2 模式下波特率和时钟输出都由 TL2 的溢出速率来决定。TF2L 中断由 T2CON 寄存器的 TL2IE 位使能。

16.2.14. 定时器 2 寄存器

T2CON: 定时器 2 控制寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xC8

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK/ TF2L	TCLK/ TL2IE	EXEN2	TR2	C/T2	CP/RL2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TF2, 定时器2溢出标志。

0: TF2必须软件清零。

1: 定时器2溢出TF2置位。当RCLK=1或TCLK=1时, TF2不会被置位。

Bit 6: EXF2, 定时器2外部标志。

0: EXF2必须软件清零。

1: 当EXEN2=1且在T2EX上有负跳变导致重载或捕获, 或者T2EXH=1并且在T2EX上有一个正跳变, 将置位定时器外部标志。当定时器2中断使能时, EXF2=1时将引起CPU进入定时器2中断向量程序。

必须清零 TL2IS (T2MOD1.5)才能访问 RCLK 位。

Bit 5: RCLK, 接收时钟控制位

0: 定时器1溢出用于接收时钟。

1: 定时器2溢出用于串口模式1和3的接收时钟。

必须置位 TL2IS (T2MOD1.5)才能访问 TF2L 位。

Bit 5: TF2L, 在定时器2分立模式中TL2溢出标志。

0: TF2L必须软件清零。

1: 在定时器2分立模式中TL2溢出TF2L置位。

必须清零 TL2IS (T2MOD1.5)才能访问 TCLK 位。

Bit 4: TCLK, 发送时钟控制位。

0: 定时器1溢出用于发送时钟。

1: 定时器2溢出用于串口模式1和3的发送时钟。

必须置位 TL2IS (T2MOD1.5)才能访问 TL2IE 位。

Bit 4: TL2IE, TF2L中断使能。

0: 禁止TF2L中断。

1: 使能共享定时器2中断入口的TF2L中断。

Bit 3: EXEN2, 定时器2外部使能位在T2EX引脚的负跳变。

0: 定时器2忽略T2EX引脚的负跳变事件。

1: 如果定时器2没有用作串口时钟, 在T2EX的负跳变时捕获或加载并作为结果。如果定时器2配置为串口0的时钟, T2EX保持外部信号侦测并产生 EXF2 标志响应中断。

Bit 2: TR2, 定时器2运行控制位。如果在定时器2分立模式中, 仅控制TH2。

0: 定时器2停止运行。

1: 定时器2开启运行。

Bit 1: C/T2, 定时器或计数器输入源选择位。该位和T2X12与T2CKS一起决定定时器2的输入来源。如下定义:

T2CKS, T2X12, C/T2	定时器 2 时钟源	分立模式下 TH2 时钟源
0 0 0	SYSCLK/12	SYSCLK/12
0 0 1	T2 Pin	T2 Pin
0 1 0	SYSCLK	SYSCLK
0 1 1	INT0ET	TL2OF
1 0 0	S0TOF	S0TOF
1 0 1	T0OF	T0OF
1 1 0	AC0ES	AC0ES
1 1 1	KBIET	KBIET

Bit 0: CP/RL2, 定时器 2 模式控制位。参考 T2MOD.T2MS0 的功能定义描述。.

T2MOD: 定时器 2 模式寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xC9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T2SPL	TL2X12/ T2EIP	T2EXH	T2X12	TR2L	TR2LC	T2OE	T2MS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: T2SPL, 定时器 2 分立模式控制。

0: 禁止定时器 2 的分立模式。

1: 使能定时器 2 的分立模式。

必须清零 TL2IS (T2MOD1.5)才能访问 TL2X12 位。

Bit 6: TL2X12, 定时器 2 分立模式下 TL2 时钟控制位。

TL2CS, TL2X12	TL2 时钟选择
0 0	SYSCLK/12
0 1	SYSCLK
1 0	S0TOF
1 1	INT0ET

必须置位 TL2IS (T2MOD1.5)才能访问 T2EIP 位。

Bit 6: T2EIP, T2EXI 输入信号反相控制位。

0: T2EXI 输入信号不反相

1: T2EXI 输入信号反相。

Bit 5: T2EXH, 定时器 2 外部 T2EX 引脚的正跳变使能标志。

0: 定时器 2 忽略 T2EX 引脚的正跳变事件。

1: 如果定时器 2 没有用作串口 0 时钟, 在 T2EX 的正跳变时捕获或加载并作为结果。如果定时器 2 配置为串口 0 的时钟, T2EX 保持外部信号侦测并产生 EXF2 旗标响应中断。

Bit 4: T2X12, 定时器 2 时钟源选择。参考 C/T2 的功能定义描述。

Bit 3: TR2L, 在定时器 2 分立模式中, TL2 运行控制位。

0: 停止 TL2。

1: 使能 TL2。

Bit 2: TR2LC, TR2L 清除控制位。

0: 禁止硬件事件清零 TR2L。

1: 使能 TH2 溢出(定时器 2 在模式 0/1)或者捕获输入(定时器 2 在模式 2/3)时自动清零 TR2L。

Bit 1: T2OE, 定时器 2 时钟输出使能位。

0: 禁止定时器 2 时钟输出。

1: 使能定时器 2 时钟输出。

Bit 0: T2MS0, 定时器 2 模式选择位 0.

T2MS1, CP/RL2, T2MS0	定时器 2 模式选择
0 0 0	模式 0: 自动重载和外部中断
0 0 1	模式 1: 自动重载带外部中断
0 1 0	模式 2: 捕捉模式
0 1 1	模式 3: 捕捉模式自动清零
1 0 0	模式 4: 8-Bit PWM (如果 T2SPL = 1)
1 1 0	模式 6: 占空比捕捉
其它	保留

T2MOD1: 定时器 2 模式寄存器 1

SFR 页 = 仅 1 页

SFR 地址 = 0x93

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL2CS	TF2IG	TL2IS	T2CKS	T2MS1	CP2S2	CP2S1	CP2S0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TL2CS, 在定时器 2 分立模式下的 TL2 时钟选择选择。参考 T2MOD.TL2X12 的功能描述。

Bit 6: TF2IG, TF2 中断忽略。

0: 使能 TF2 中断。默认是使能的。

1: 禁止 TF2 中断。

Bit 5: TL2IS, TF2L 和 TL2IE 访问控制。

0: 使能 RCLK 和 TCLK 的访问功能在位 T2CON.5~4。

1: 使能 TF2L 和 TL2IE 的访问功能在位 T2CON.5~4。

Bit 4: T2CKS, 定时器 2 时钟输入选择。参考 C/T2 的功能描述。

Bit 3: T2MS1, 定时器 2 模式选择位 1。参考 T2MOD.T2MS0 的功能描述。

Bit 2~0: CP2S.2~0, 此 3 位定义定时器 2 的捕捉源选择。

CP2S.2~0	定时器 2 捕捉源选择
0 0 0	T2EX 引脚
0 0 1	RXD0
0 1 0	P6.0 引脚
0 1 1	INT2ET
1 0 0	ILRCO
1 0 1	AC0OUT
1 1 0	KBIET
1 1 1	TWI0_SCL

TL2: 定时器 2 低字节寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xCC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH2: 定时器2 高字节寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xCD

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RCAP2L: 定时器2 捕捉低字节寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xCA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RCAP2H: 定时器2 捕捉高字节寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xCB

复位值 = 0000-0000

7	6	5	4	3	2	1	0
RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

AUXR4: 辅助寄存器 4

SFR 页 = 仅 1 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	AC1FLT1	AC0OE	AC0FLT1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: T2PS1~0, 定时器 2 端口引脚选择 [1:0].

T2PS1~0	T2/T2CKO	T2EX
00	P1.0	P1.1
01	P3.2	P3.3
10	P6.7	P5.7
11	P4.5	P4.4

16.3. 定时器 3

定时器 3 是一个 16 位定时器/计数器，既可作为一个定时器也可以作为一个事件计数器，由 T3CKS, T3X12 和 C/T3 选择。定时器 3 有几种工作模式:捕捉、自动重新加载(向上计数)、8 位 PWM 和可编程时钟输出，这些都是由 T3CON、T3MOD 和 T3MOD1 寄存器中的位来选择的。

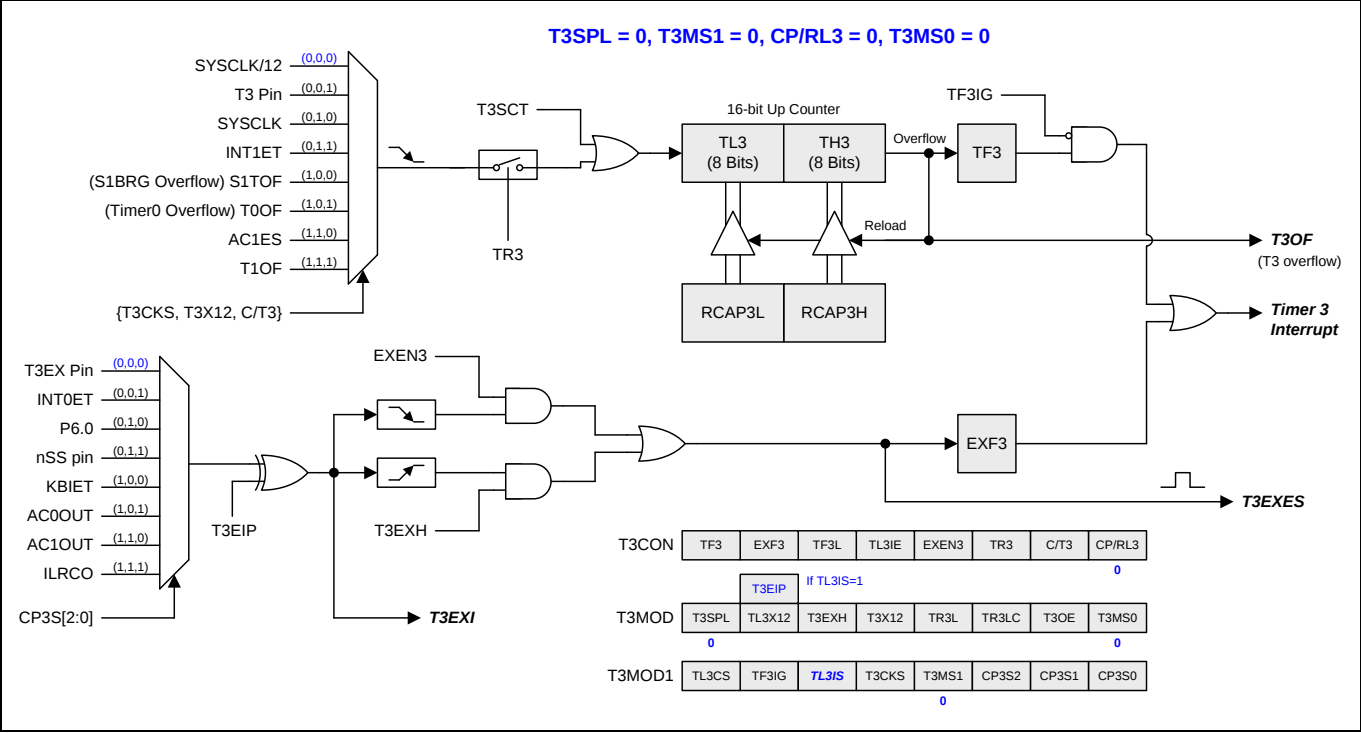
16.3.1. 定时器 3 模式 0(自动重载和外部中断)

在这个模式中，定时器 3 提供一个 16 位的自动重载定时器/计数器。TF3 是定时器 3 的溢出标志，是一个中断会被 TF3IG 阻断的定时器 3 中断源。EXEN3 使能 T3EXI 引脚的下降沿置位 EXF3，EXF3 可作为一个外部中断与 TF3 共享定时器 3 中断。T3EXI 由 8 个外部触发输入中选出一个。T3EXH 的功能与 EXEN3 一样，只是 T3EXH 使能 T3EXI 引脚的上升沿置位 EXF3。

本模块中的定时器 3 溢出事件(T3OF)可以作为时钟输入或事件源输出到其它外设。

定时器 3 模式 0 如图 16-30 所示

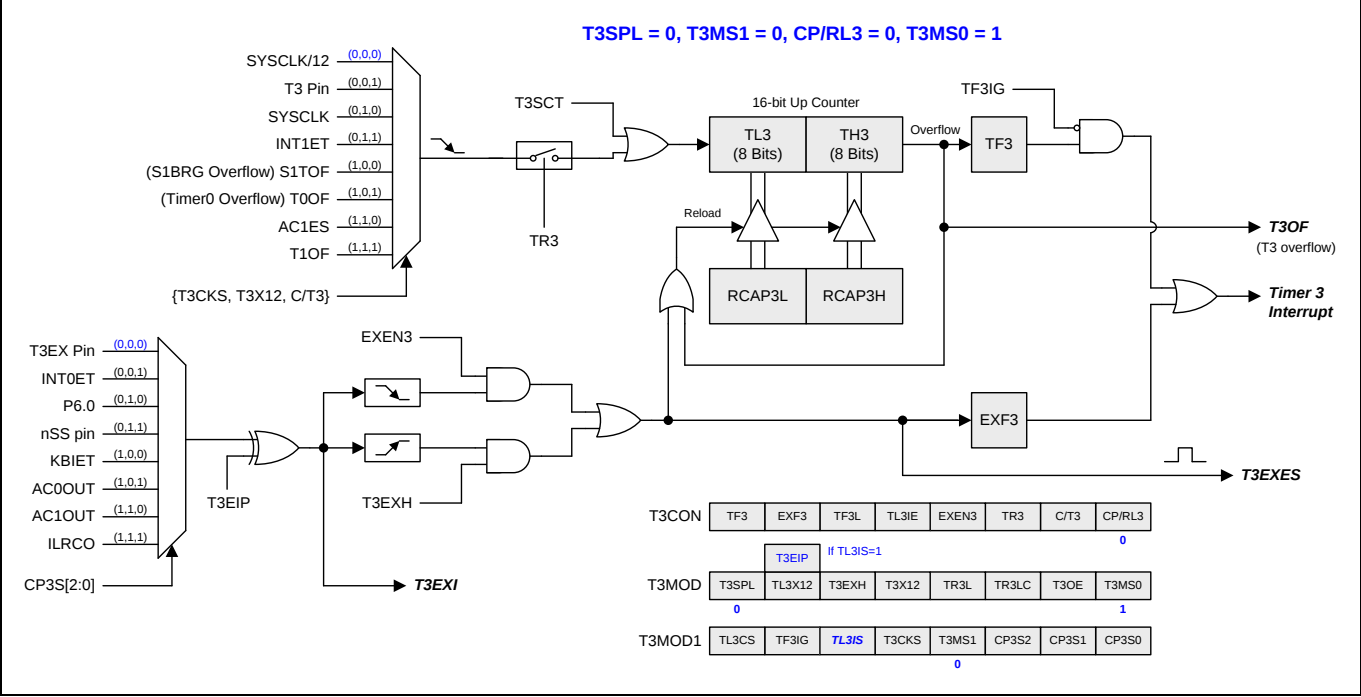
图 16-30. 定时器 3 模式 0 结构(自动重载和外部中断模式)



16.3.2. 定时器 3 模式 1(外部中断触发自动重载)

如图 16-31 所示，定时器 3 模式 1，使能定时器 3 自动向上计数。本模式有 T3CON 寄存器的 EXEN3 决定的两种选择。如果 EXEN3=0，定时器 3 向上计数到 0xFFFFH 且直到溢出而置位 TF3(溢出标志)。这时定时器 3 寄存器被重载入 RCAP3L 和 RCAP3H 的 16 位数据。RCAP3L 和 RCAP3H 的值由软件预设。如果 EXEN3=1，16 位重载会被一个 T3 溢出或一个 T3EXI 的下降沿触发，T3EXI 由 8 个外部触发输入中选出 一个，此触发同时置位 EXF3。如果定时器 3 中断使能，无论 TF3 或 EXF3 置位则产生中断。T3EXH 的功能与 EXEN3 一样，只是 T3EXH 使能 T3EXI 引脚的上升沿置位 EXF3。

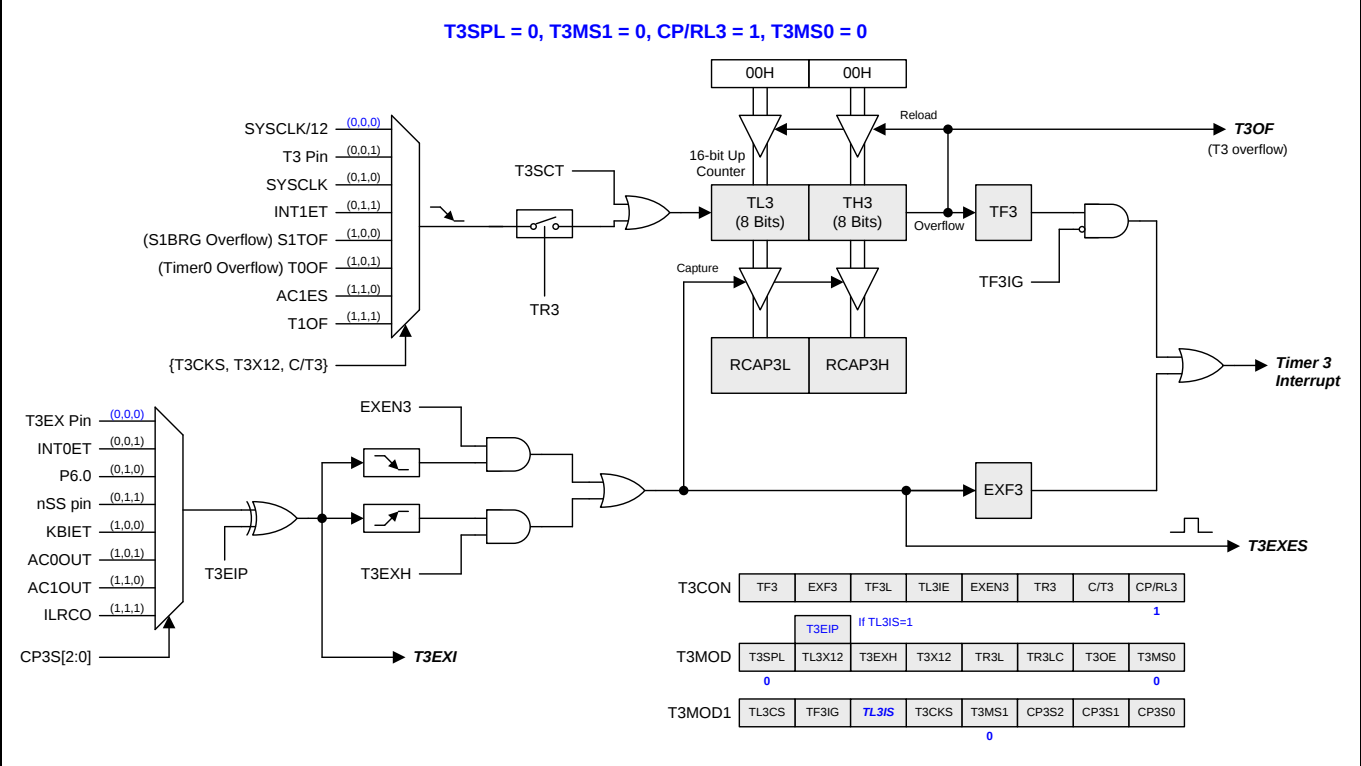
图 16-31. 定时器 3 模式 1 结构(外部中断触发自动重载模式)



16.3.3. 定时器 3 模式 2(捕捉)

由 T3CON 寄存器的位 EXEN3 做二种选择的捕捉模式如图 16-32 所示。如果 EXEN3=0，定时器 3 是一个 16 位定时器或计数器，向上溢出而置位 TF3(定时器 3 溢出标志)。TF3 常用来产生中断(IE 寄存器使能定时器 3 中断相关位)。模式 2 增加了捕捉功能，如果 EXEN3=1，T3EXI 引脚下降沿(由 8 个外部触发输入中选出)把定时器 3 寄存器 (TH3 和 TL3)各自捕捉到寄存器(RCAP3H 和 RCAP3L)。另外，T3EXI 引脚跳变使 T3CON 寄存器的 EXF3 置位，且 EXF3 跟 TF3 共享中断位置。T3EXH 的功能与 EXEN3 一样，只是 T3EXH 使能 T3EXI 引脚的上升沿置位 EXF3。

图 16-32. 定时器 3 模式 2 结构(捕捉模式)

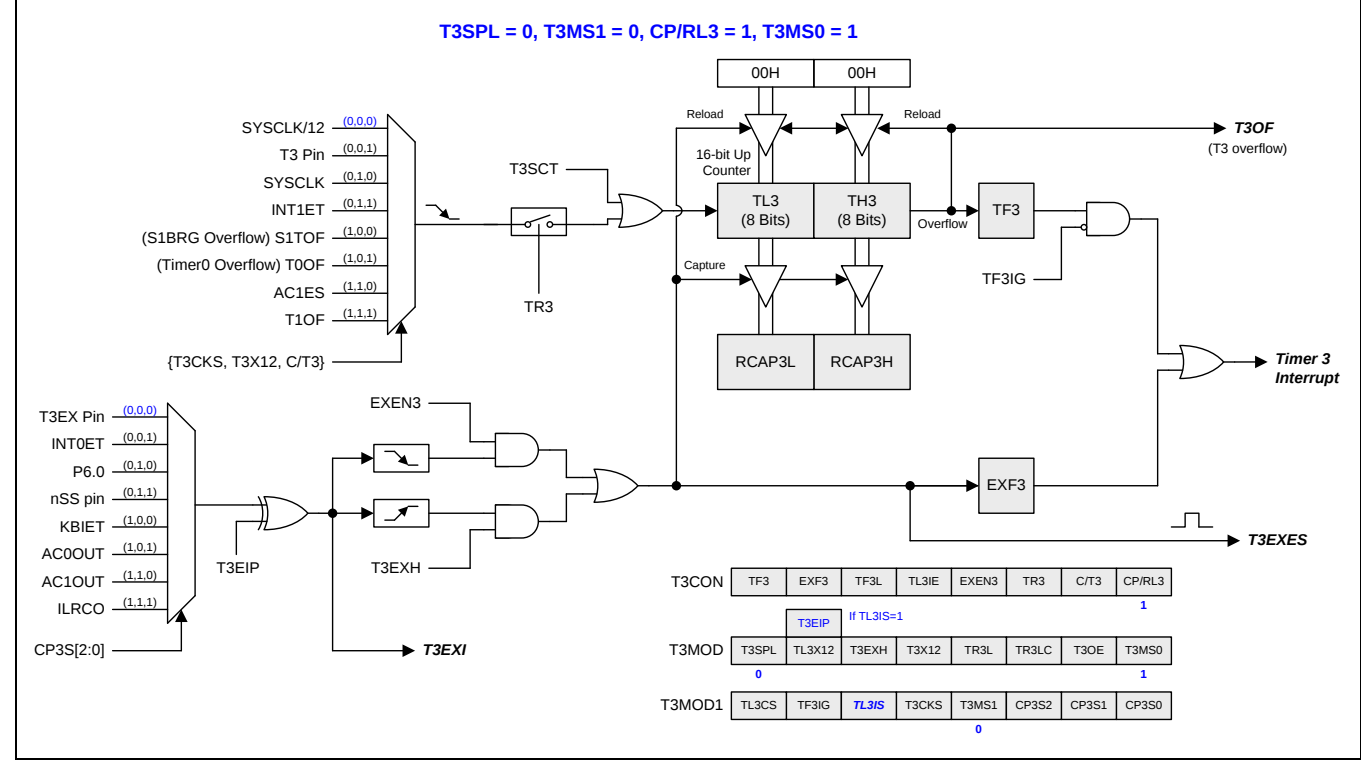


16.3.4. 定时器 3 模式 3(捕捉带自动清零)

定时器 3 模式 3 与定时器 3 模式 2 的功能相似。有一点不同的就是 T3EXES、EXF3 置位，不但定时器 3 会被捕捉而且 TL3 和 TH3 也会被清零。

定时器 3 模式 3 如图 16-33 所示

图 16-33. 定时器 3 模式 3 结构 (捕捉带自动清零 TL3 和 TH3)



16.3.5. 定时器 3 模式 6 (占空比捕捉)

定时器 3 模式 6 支持捕捉输入波形的周期时间或占空比。信号的三个边沿可以计算出周期和占空比。在占空比捕捉模式，需要将 TH3, TL3 清零。然后通过置位 TR3 开启捕捉模式，但是计数器还没有启动，它会等到第一个边缘进入到外部触发通道，例如 T3EX 引脚。这表示第一个边值是 00H。在第一次触发边缘后，计数器开始计数。请注意，T3EXI 的第一个触发边缘必须是上升沿。即使你设置了 T3EXH，第一个边沿也会被忽略以触发 EXF3。

其次，如果只想计算脉冲宽度，那么可以设置 EXEN3 通过第二个边来触发 EXF3。在本例中，当第二个边缘触发计时器将其值捕捉到 RCAP3H: RCAP3L 中时，它还触发用于中断的 EXF3，以标识它已经完成。

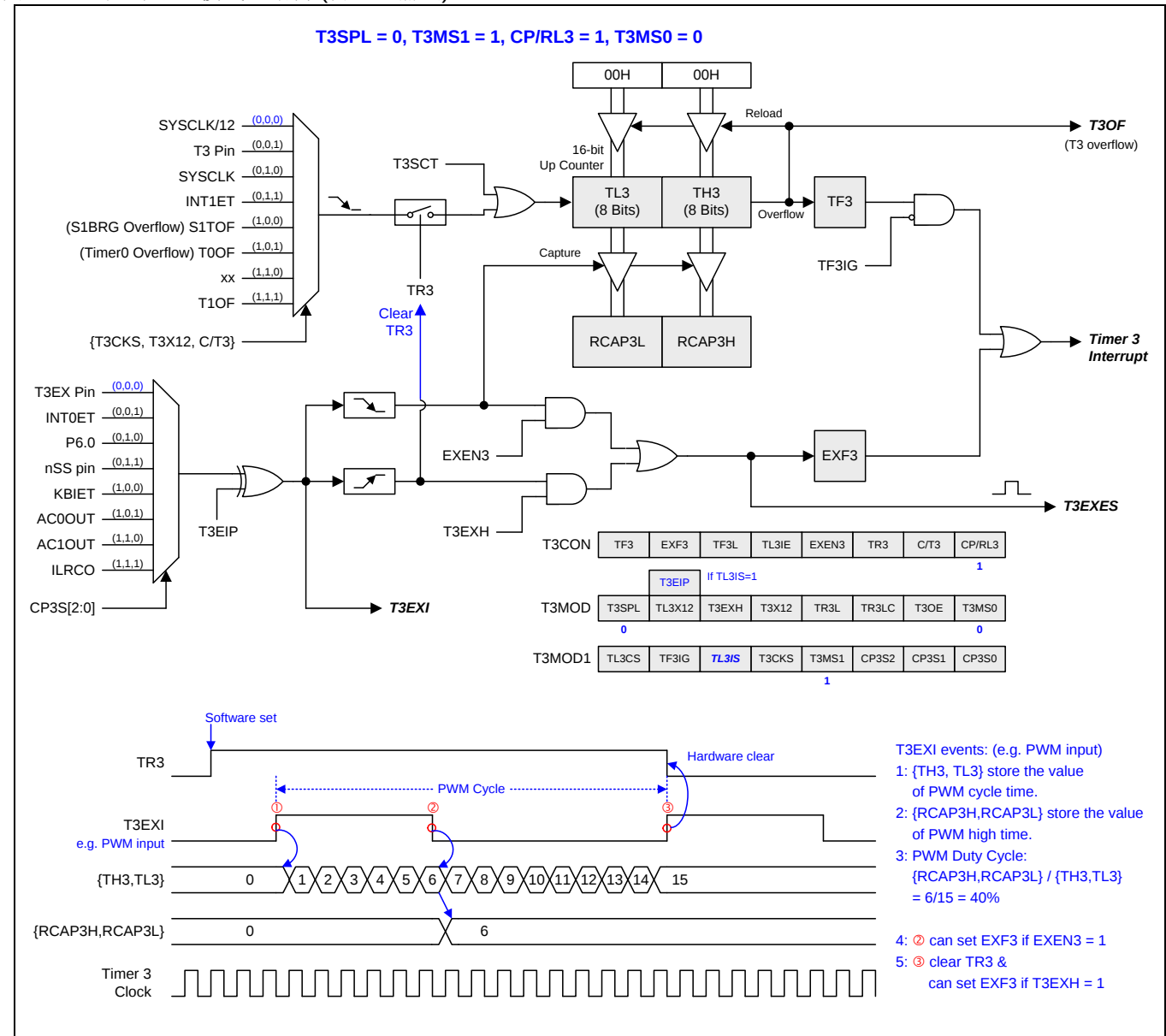
但是如果你想获得周期时间，那么就要通过清除 EXEN3 去阻止第二边沿。

当第三个边沿到来，它将自动清除 TR3 来停止计数器。

使用 TH3: TL3(第三条边)、RCAP3H: RCAP3L(第二条边)和 0 (第一条边)来计算周期时间和占空比。

定时器 3 模式 6 如 图 16-34 所示

图 16-34. 定时器 3 模式 6 结构 (占空比捕捉)



16.3.6. 分立定时器 3 模式 0 (自动重载和外部中断)

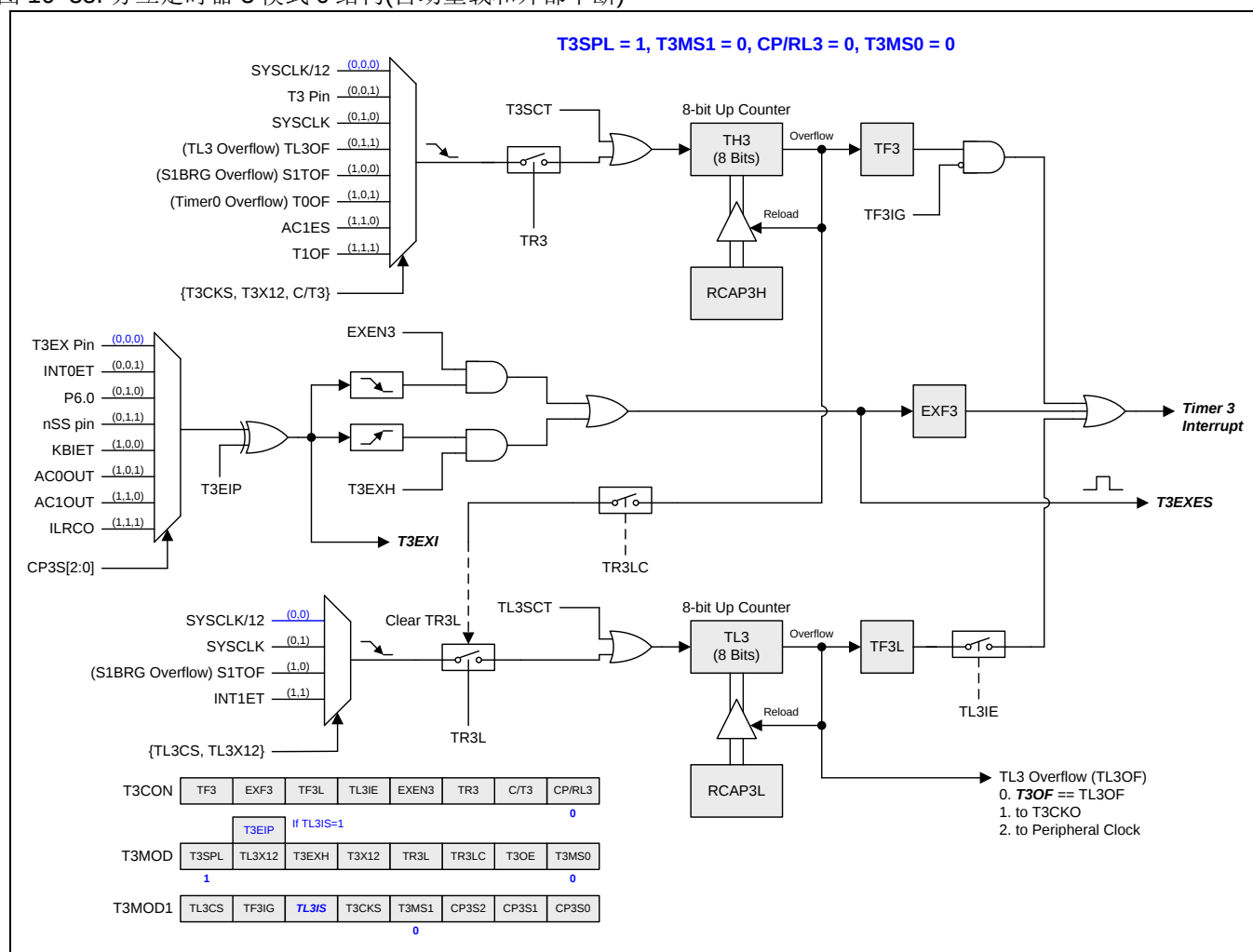
本模式中 T3SPLIT 置位，定时器 3 变为两个 8 位定时器（TH3 和 TL3）。两个 8 位定时器都是向上计数如图 16-35 所示。TH3 保存 RCAP3H 的重载值和保持 16 位模式一样的 8 个时钟源输入选择。8 位定时器功能跟 16 位模式的定时器 3 模式 0 相似。TL3 保存 4 个时钟输入选择的 RCAP3L 重载值。T3CON 的位 TR3 控制着 TH3 的运行。T3MOD 的位 TR3L 控制着 TL3 的运行。当 TR3LC 置位时 TH3 溢出会停止 TR3L 的运行。

分立模式有 3 个中断标志 EXF3、TF3 和 TF3L。EXF3 与 16 位模式一样的功能用来侦测 T3EXI 引脚的跳变。TF3IG 控制 TF3 在 TH3 从 0xFF 到 0x00 溢出时是否置位。TL3 从 0xFF 到 0x00 溢出时 TF3L 置位，TL3IE 使能中断。EXF3、TF3 和 TF3L 中断标志硬件不会清零且必须软件清零。

附带一提，16 位模式中的定时器 3 溢出事件(T3OF)将被分立模式中的 TL3 溢出事件(TL3OF)取代。

如果 T3MOD1 的位 TL3IS=0，位 T3MOD.6 是 TL3X12 的功能。如果 TL3IS=1，位 T3MOD.6 是 T3EIP 的功能。

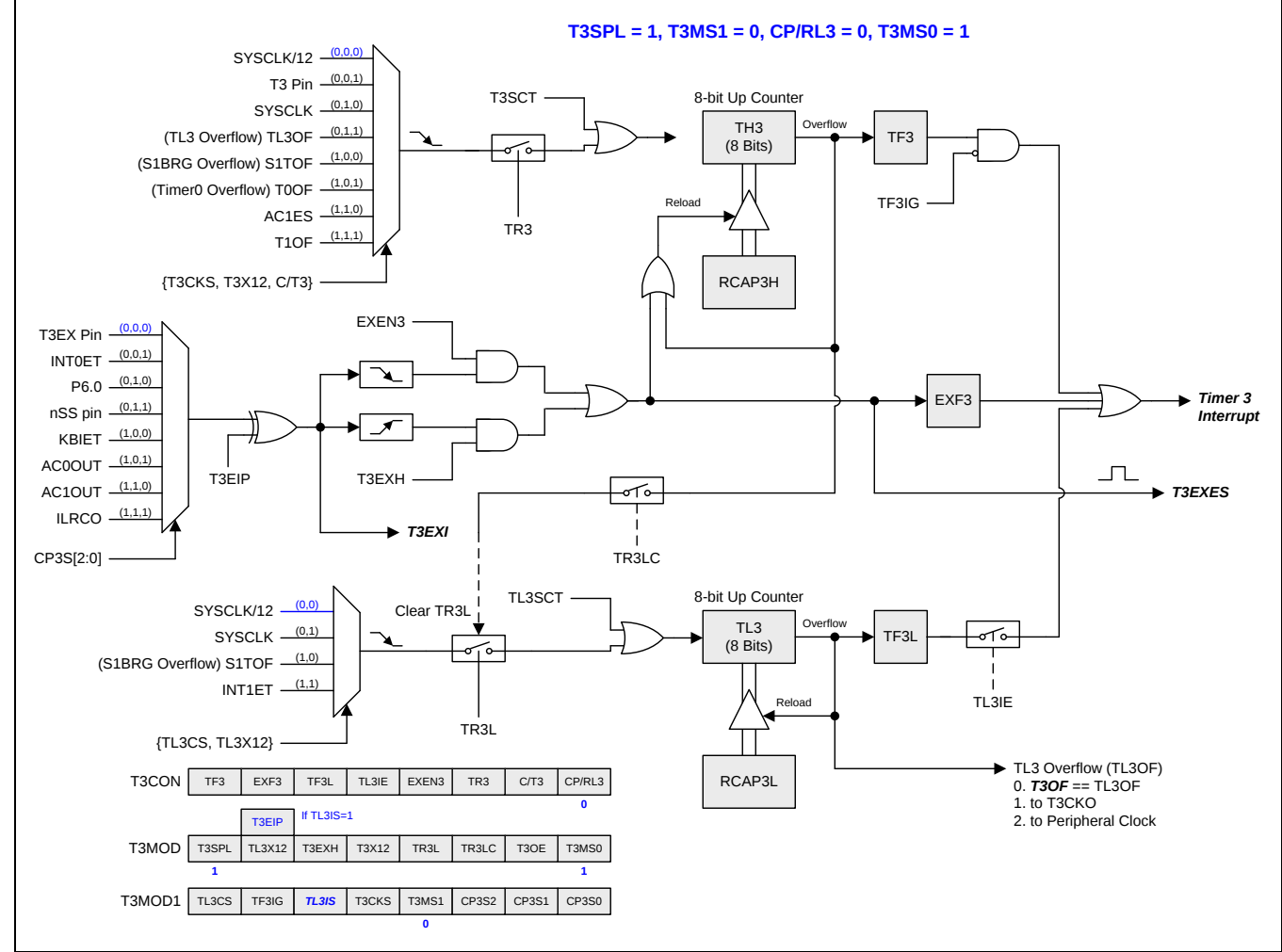
图 16-35. 分立定时器 3 模式 0 结构(自动重载和外部中断)



16.3.7. 分立定时器 3 模式 1(外部中断触发自动重载)

本模式中 T3SPLIT 置位，定时器 3 分立为两个 8 位定时器如图 16-36 所示。跟定时器 3 模式 1 相似的功能且保持与分立定时器 3 模式 0 一样的中断方式。

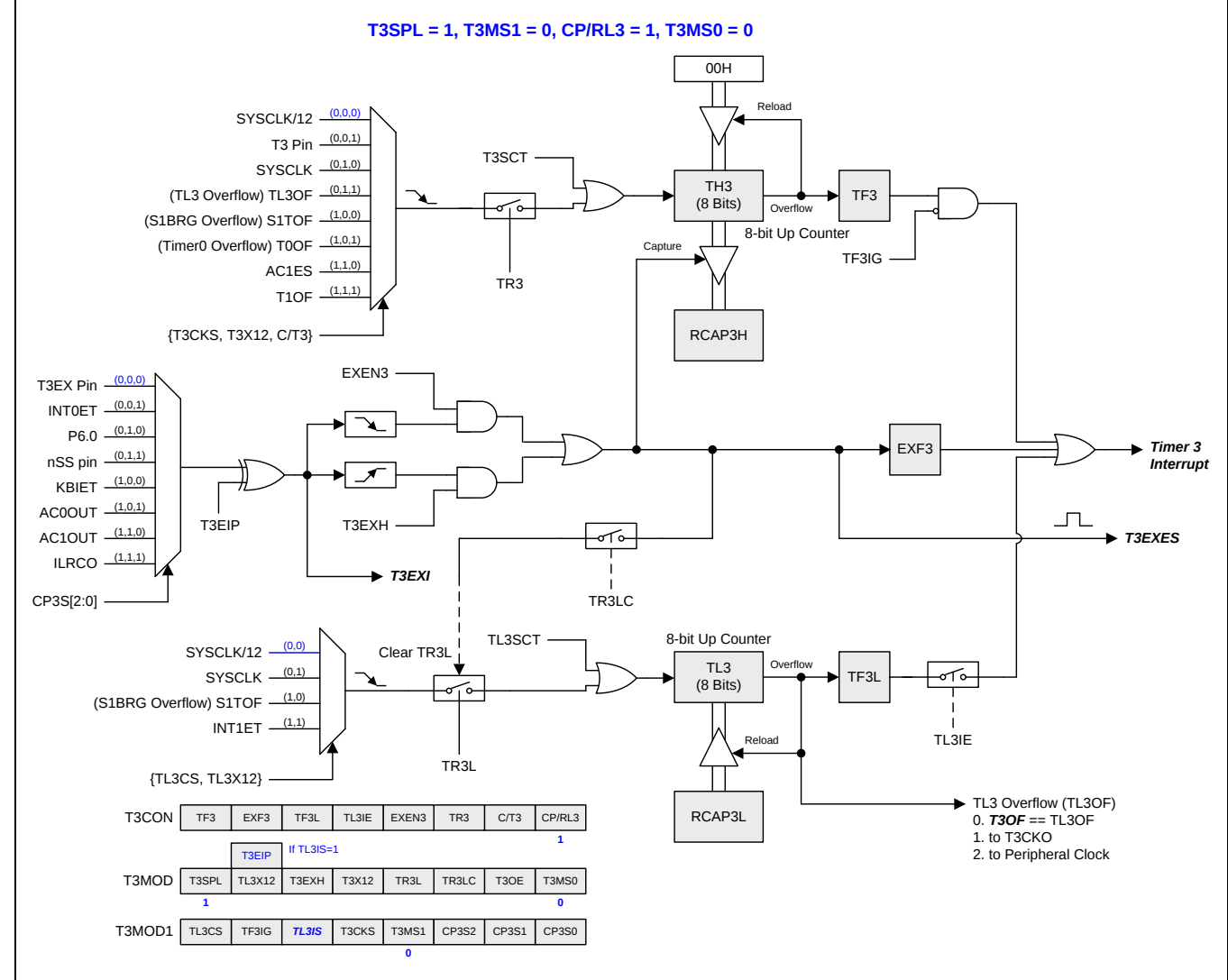
图 16-36. 分立定时器 3 模式 1 结构(外部中断触发自动重载)



16.3.8. 分立定时器 3 模式 2(捕捉)

本模式中 T3SPLIT 置位，定时器 3 分立为两个 8 位定时器如图 16-37 所示。跟定时器 3 模式 2 相似的功能且保持与分立定时器 3 模式 0 一样的中断方式。

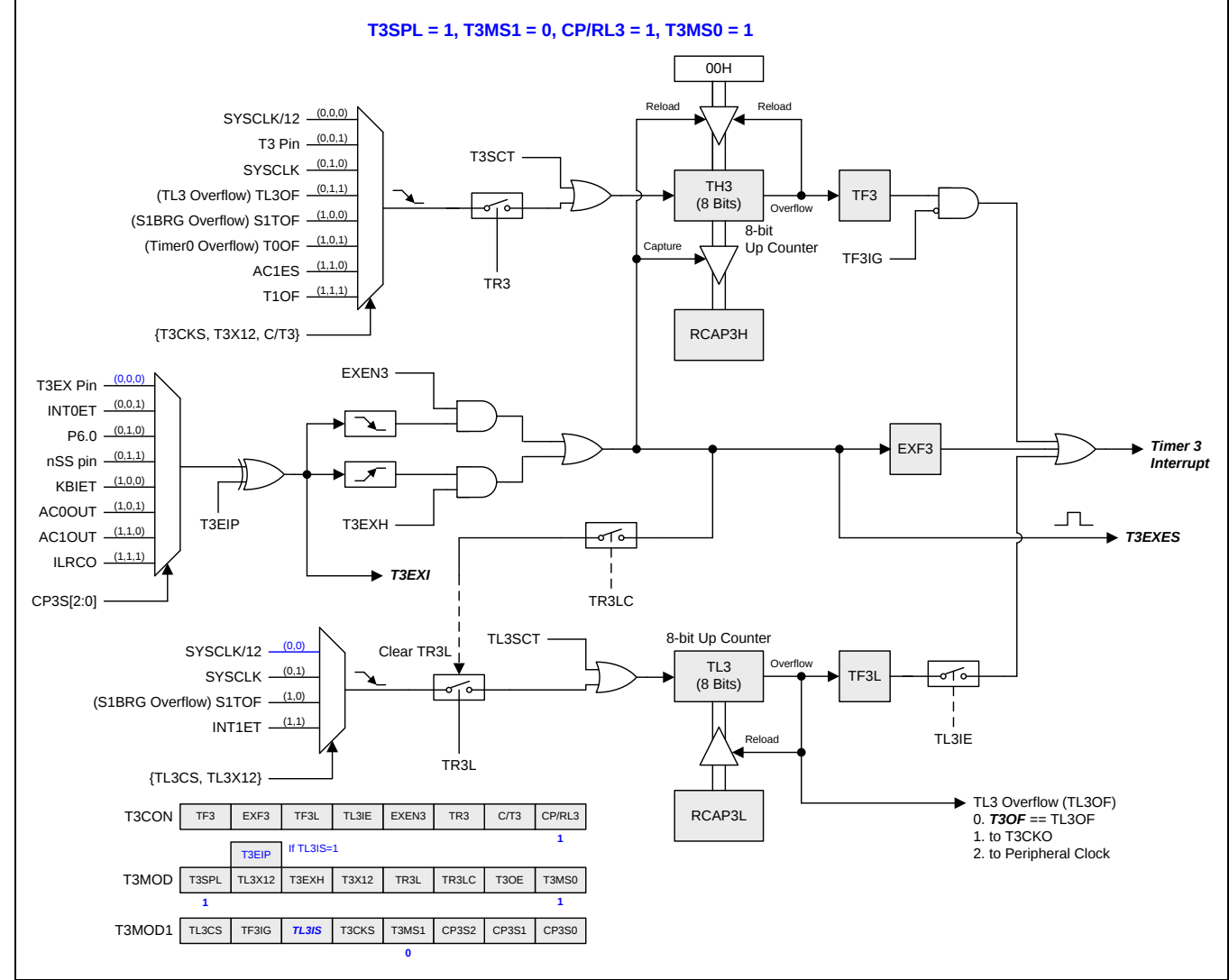
图 16-37. 分立定时器 3 模式 2 结构(捕捉)



16.3.9. 分立定时器 3 模式 3 (捕捉带自动清零)

本模式中 T3SPLIT 置位，定时器 3 分立为两个 8 位定时器如图 16-38..所示。跟定时器 3 模式 3 相似的功能且保持与分立定时器 3 模式 0 一样的中断方式。

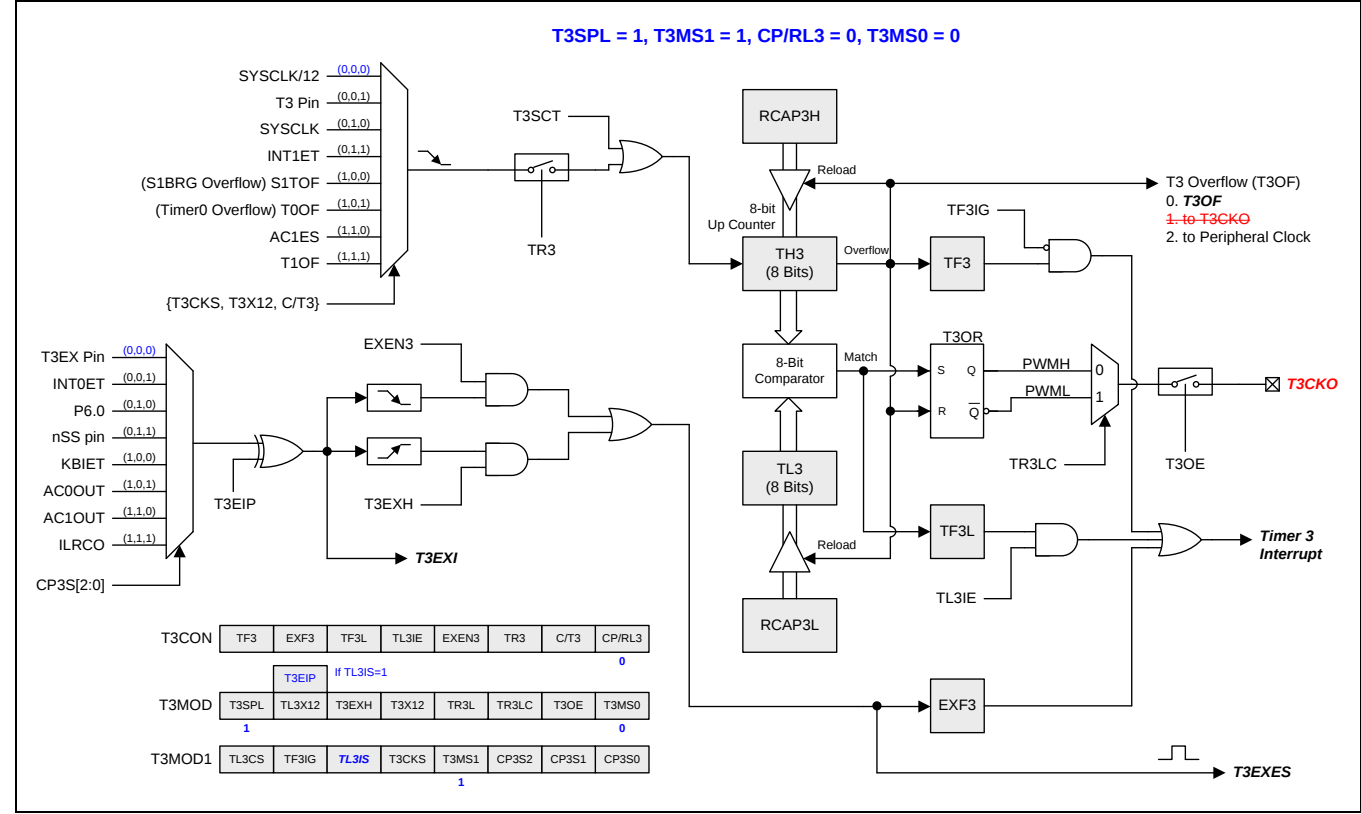
图 16-38. 分立定时器 3 模式 3 结构(捕捉带自动清零 TH3)



16.3.10. 分立定时器 3 模式 4 (8-位 PWM 模式)

本模式，定时器 3 是一个 8 位的 PWM 模式如图 16-39 所示。TH3 和 RCAP3H 相结合为一个 8 位的自动重载计数器。这二个寄存器的软件配置决定 PWM 周期。TL3 是 PWM 比较寄存器用来生成 PWM 波形。RCAP3L 是 PWM 缓冲寄存器且在此寄存器中软件更新 PWM 数据。每次 TH3 溢出事件置位 TF3 且 RCAP3L 值载入到 TL3。PWM 信号输出到 T3CKO 功能引脚且输出的开关由 T3MOD 寄存器的位 T3OE 决定。

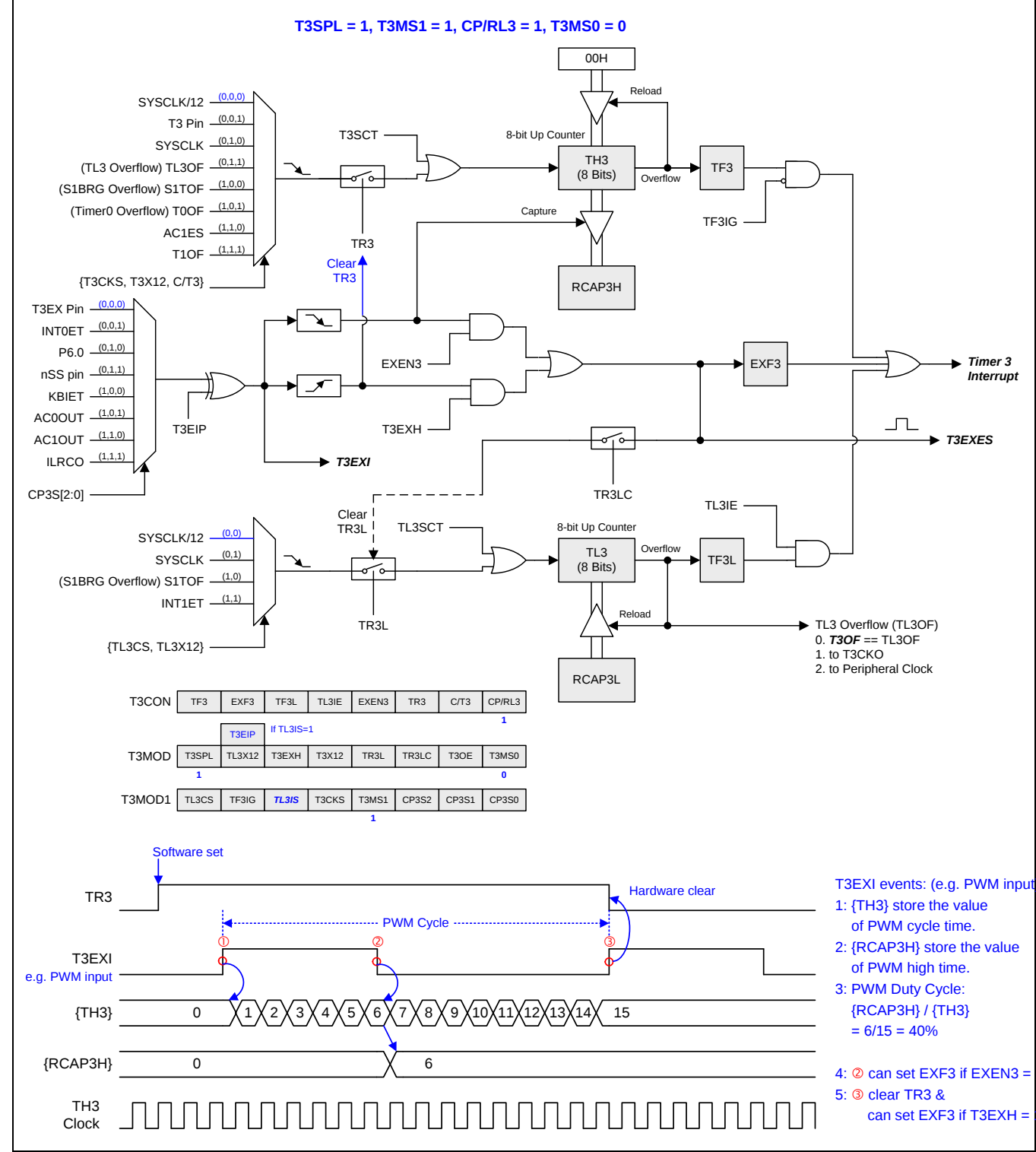
图 16-39. 分立定时器 3 模式 4 结构(8 位 PWM 模式)



16.3.11. 分立定时器 3 模式 6 (占空比捕捉)

当 T3SPL 设置在这种模式下,定时器 3 是分成两个 8 位计时器, 如图 16-40..所示。类似的功能为与定时器 3 模式 6 有着类似的功能, 并且保持分立定时器 3 模式 0 同样的中断机制。

图 16-40. 分立定时器 3 模式 6 结构 (占空比捕捉)



16.3.12. 定时器 3 可编程时钟输出

定时器 3 有一个时钟输出模式(当 CP/RL3=0 并且 T3OE=1)。在这个模式，定时器 3 运行为一个占空比为 50%的可编程时钟输出。产生的时钟从 T3CKO 输出。输入时钟(SYSCLK/2 或 SYSCLK)使 16 位定时器(TH3, TL3)加一。定时器从载入值到溢出重复计数。一旦溢出，(RCAP3H, RCAP3L)的值被载入到(TH3, TL3)同时计数。图 16-41 给出了定时器 3 时钟输出频率计算公式。定时器 3 的时钟输出结构如图 16-42 所示。

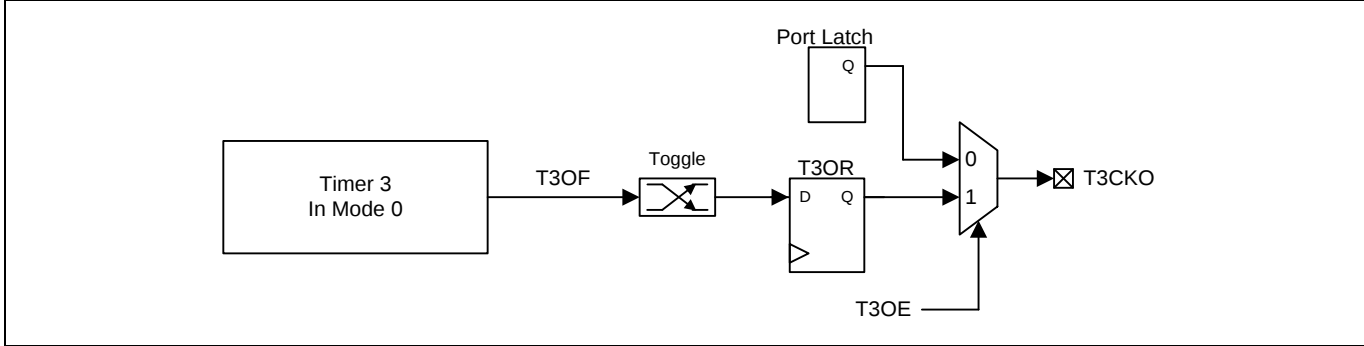
图 16-41. 定时器 3 时钟输出频率计算公式

$$\text{T3 Clock-out Frequency} = \frac{\text{T3 Clock Frequency}}{2 \times (65536 - (\text{RCAP3H}, \text{RCAP3L}))}$$

注意:

- (1) 定时器 3 溢出标志 TF3,在定时器 3 溢出时置位可产生中断。但是 TF3 中断会被 T3MOD1 寄存器的位 TF3IG 锁住。
- (2) 当 SYSCLK=12MHz 及 SYSCLK/2 作为定时器 3 时钟源, 定时器 3 可编程输出频率范围从 45.7Hz 到 3MHz。
- (3) 当 SYSCLK=12MHz 及 SYSCLK 作为定时器 3 时钟源, 定时器 3 可编程输出频率范围从 91.5Hz 到 6MHz。

图 16-42. 定时器 3 时钟输出模式



定时器 3 时钟输出模式如何编程

- 选择定时器 3 时钟源
- 从公式计算出 16 位自动加载值并输入到 RCAP3H 和 RCAP3L 寄存器.
- 在 TH3 和 TL3 寄存器输入一个跟自动加载值相同的初始值
- T3MOD 寄存器的 T3OE 置位.
- T3CON 寄存器的 TR3 置位启动定时器 3.

在时钟输出模式，定时器 3 翻转产生 TF3 中断。中断可以被 TF3IG 阻止。

如果定时器 3 在分立模式，时钟输出功能由 TL3 溢出产生且输出时钟频率为 TL3 溢出率的二分之一。当 TL3 溢出时 RCAP3L 是 TL3 重载值。TL3 有 4 种时钟源选择。在使能分立定时器 3 时钟输出功能之前，软件必须结束 TL3 时钟源配置。图 16-43 给出了 TL3 时钟输出频率公式。分立定时器 3 的时钟输出结构如图 16-44 所示。

图 16-43. 分立定时器 3 时钟输出公式

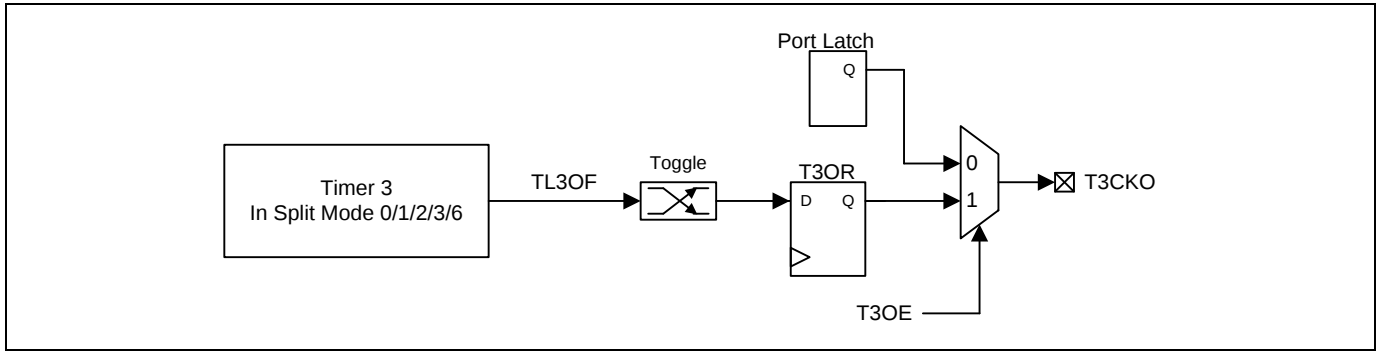
Split T3 Clock-out Frequency =

TL3 Clock Frequency

3 x (256 - RCAP3L)

注意:
(1) TL3 溢出标志 TF3L,在 TL3 溢出时置位产生中断。但是 TF3L 中断由 T3CON 寄存器的位 TL3IE 使能。
(2) 当 SYSCLK=12MHz 及 SYSCLK/12 作为 TL3 时钟源,TL3 可编程输出频率范围从 1.95KHz 到 500KHz。
(3) 当 SYSCLK=12MHz 及 SYSCLK 作为 TL3 时钟源,TL3 可编程输出频率范围从 23.44KHz 到 6MHz。

图 16-44. 分立定时器 3 时钟输出模式



分立定时器 3 时钟输出模式如何编程

- 选择 TL3 时钟源.
- 从公式计算出 8 位自动加载值并输入到 RCAP3L 寄存器.
- 在 TL3 寄存器输入一个跟自动加载值相同的初始值.
- T3MOD 寄存器的 T3OE 置位.
- T3MOD 寄存器的 TR3L 置位启动定时器 3.

在时钟输出模式，TL3 翻转不会产生中断。这和 TL3 用作波特率发生器时相似。可同时使用 TL3 作为一个波特率发生器和时钟发生器。注意，在分立定时器 3 模式下波特率和时钟输出都由 TL3 的溢出速率来决定。TF3L 中断由 T3CON 寄存器的 TL3IE 位使能。

16.3.13. 定时器 3 寄存器

T3CON: 定时器 3 控制寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xC8

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TF3	EXF3	TF3L	TL3IE	EXEN3	TR3	C/T3	CP/RL3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TF3, 定时器 3 溢出标志

0: TF3 必须软件清零。

1: 定时器 3 溢出 TF3 置位。

Bit 6: EXF3, 定时器 3 外部标志。

0: EXF3 必须软件清零。

1: 当 EXEN3=1 且在 T3EX 上有负跳变导致重载或捕获, 或者 T3EXH=1 并且在 T3EX 上有一个正跳变, 将置位定时器外部标志。当定时器 3 中断使能时, EXF3=1 时将引起 CPU 进入定时器 3 中断向量程序。当 MCU 在掉电模式下定时器 3 中断使能时, EXF3 被设置成具有唤醒 MCU 能力的电平触发。

Bit 5: TF3L, 在定时器 3 分立模式中 TL3 溢出标志。

0: TF3L 必须软件清零。

1: 在定时器 3 分立模式中 TL3 溢出 TF3L 置位。

Bit 4: TL3IE, TF3L 中断使能。

0: 禁止 TF3L 中断。

1: 使能共享定时器 3 中断入口的 TF3L 中断。

Bit 3: EXEN3, 定时器 3 外部使能位在定时器 3 外部输入引脚的负跳变。

0: 定时器 3 忽略定时器 3 外部输入引脚的负跳变事件。

1: 在定时器 3 外部输入的负跳变时捕获或加载并作为结果。如果定时器 3 配置为没有捕捉或重载的模式 0, 定时器 3 外部输入保持外部信号侦测并产生 EXF3 标志响应定时器 3 中断。

Bit 2: TR3, 定时器 3 运行控制位。如果在定时器 3 分立模式中, 仅控制 TH3。

0: 定时器/计数器 3 停止运行。

1: 定时器/计数器 3 开启运行。

Bit 1: C/T3, 定时器 3 时钟或计数器输入源选择位。和 T3X12 与 T3CKS 一起决定定时器 3 的输入来源。如下定义:

T3CKS, T3X12, C/T3	定时器 3 时钟源	分立模式下 TH3 时钟选择
0 0 0	SYSCLK/12	SYSCLK/12
0 0 1	T3 Pin	T3 Pin
0 1 0	SYSCLK	SYSCLK
0 1 1	INT1ET	TL3OF
1 0 0	S1TOF	S1TOF
1 0 1	T0OF	T0OF
1 1 0	AC1ES	AC1ES
1 1 1	T1OF	T1OF

Bit 0: CP/RL3, 定时器 3 模式控制位。参考 T3MOD.T3MS0 的功能定义描述。

T3MOD: 定时器3 模式寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xC9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T3SPL	TL3X12/ T3EIP	T3EXH	T3X12	TR3L	TR3LC	T3OE	T3MS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: T3SPL, 定时器 3 分立模式控制。

0: 禁止定时器 3 的分立模式。

1: 使能定时器 3 的分立模式。

必须清零 TL3IS (T3MOD1.5)才能访问 TL3X12 位。

Bit 6: TL3X12, 定时器 3 分立模式下 TL3 时钟控制位。.

TL3CS, TL3X12	TL3 时钟选择
0 0	SYSCLK/12
0 1	SYSCLK
1 0	S1TOF
1 1	INT1ET

必须置位 TL3IS (T3MOD1.5)才能访问 T3EIP 位。

Bit 6: T3EIP, T3EXI 输入信号反相控制位。

0: T3EXI 输入信号不反相

1: T3EXI 输入信号反相。

Bit 5: T3EXH, 定时器 3 外部 T3EX 引脚的正跳变使能标志。

0: 定时器 3 忽略 T3EX 引脚的正跳变事件。

1: 允许在 T3EX 的正跳变时捕获或加载并作为结果且置位 EXF3。

Bit 4: T3X12, 定时器 3 时钟源选择。参考 C/T3 的功能定义描述。

Bit 3: TR3L, 在定时器 3 分立模式中, TL3 运行控制位。

0: 停止 TL3。

1: 使能 TL3。

Bit 2: TR3LC, TR3L 清除控制位。

0: 禁止硬件事件清零 TR3L。

1: 使能 TH3 溢出(定时器 3 在模式 0/1)或者捕获输入(定时器 3 在模式 2/3)时自动清零 TR3L。

Bit 1: T3OE, 定时器 3 时钟输出使能位。

0: 禁止定时器 3 时钟输出。

1: 使能定时器 3 时钟输出。

Bit 0: T3MS0, 定时器 3 模式选择位 0。

T3MS1, CP/RL3, T3MS0	定时器 3 模式选择
0 0 0	模式 0: 自动重载和外部中断
0 0 1	模式 1: 自动重载带外部中断
0 1 0	模式 2: 捕捉模式
0 1 1	模式 3: 定时器 3 捕捉带自动清零
1 0 0	模式 4: 8 位 PWM (T3SPL = 1)
1 1 0	模式 6: 占空比捕捉
Others	保留

T3MOD1: 定时器3 模式寄存器 1

SFR 页 = 仅 2 页

SFR 地址 = 0x93

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL3CS	TF3IG	TL3IS	T3CK2	T3MS1	CP3S2	CP3S1	CP3S0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TL3CS, 在定时器3分立模式下的 TL3 时钟选择选择。参考 T3MOD.TL3X12 的功能描述。

Bit 6: TF3IG, TF3 中断忽略。

0: 使能 TF3 中断。默认是使能的。

1: 禁止 TF3 中断。

Bit 5: TL3IS, TL3X12/T3EIP 访问控制

0: 使能 T3MOD.6 访问 TL3X12

1: 使能 T3MOD.6 访问 T3EIP

Bit 4: T3CKS, 定时器3时钟输入选择。参考 C/T3 的功能描述。

Bit 3: T3MS1, 定时器3模式选择位 1。参考 T3MOD. T3MS0 的功能描述。

Bit 2~0: CP3S.2~0, 此 3 位定义定时器3的捕捉源选择。

CP3S.2~0	定时器3 捕捉源选择
0 0 0	T3EX 引脚
0 0 1	INT0ET
0 1 0	P6.0 引脚
0 1 1	nSS 引脚
1 0 0	KBIET
1 0 1	AC0OUT
1 1 0	AC1OUT
1 1 1	ILRCO

TL3: 定时器3 低字节寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xCC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL3.7	TL3.6	TL3.5	TL3.4	TL3.3	TL3.2	TL3.1	TL3.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH3: 定时器3 高字节寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xCD

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TH3.7	TH3.6	TH3.5	TH3.4	TH3.3	TH3.2	TH3.1	TH3.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RCAP3L: 定时器3 捕捉低字节寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xCA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
RCAP3L.7	RCAP3L.6	RCAP3L.5	RCAP3L.4	RCAP3L.3	RCAP3L.2	RCAP3L.1	RCAP3L.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RCAP3H: 定时器3 捕捉高字节寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xCB

复位值 = 0000-0000

7	6	5	4	3	2	1	0
RCAP3H.7	RCAP3H.6	RCAP3H.5	RCAP3H.4	RCAP3H.3	RCAP3H.2	RCAP3H.1	RCAP3H.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

AUXR8: 辅助寄存器 8

SFR 页 = 仅 5 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1~0: T3PS1~0, 定时器 3 端口引脚选择 [1:0].

T3PS1~0	T3/T3CKO	T3EX
0 0	P3.3	P3.4
0 1	P3.3	P3.2
1 0	P0.2	P0.1
1 1	P6.0	P6.1

16.4. 定时器全局控制

当应用要求所有定时器在同步模式下工作时，可以设置该寄存器来启动、重载和停止所有定时器。

16.4.1. 所有定时器运行的全局控制

当应用要求所有定时器在同步模式下工作时，仅需置位 TREN0 的 TRxE 或 TRxLE 来同时启动定时器。在写入“1”之后，这些寄存器将被硬件自动清零。

TREN0: 定时器运行使能寄存器 0

SFR 页 = 仅 1 页

SFR 地址 = 0x95

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TR2E	TR3E	0	0	0	0	C1CR	PTMCR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TR2E, 这个位上写“1”设置 TR2 使能(TR2=1)。在写入“1”之后，这个位被硬件自动清零。这个位上写“0”无作用。

Bit 6: TR3E, 这个位上写“1”设置 TR3 使能(TR3=1)。在写入“1”之后，这个位被硬件自动清零。这个位上写“0”无作用。

Bit 5~2: 保留。写该寄存器时，此位必须写“0”。

Bit 1: C1CR, 这个位上写“1”设置 PCA1 的 C1R 使能(C1R =1)。在写入“1”之后，这个位被硬件自动清零。这个位上写“0”无作用。

Bit 0: PTMCRE, 这个位上写“1”设置 PTM0 的 CR 使能(CR =1)。在写入“1”之后，这个位被硬件自动清零。这个位上写“0”无作用。

16.4.2. 所有定时器重载的全局控制

TRLC0: 定时器重载控制寄存器 0

SFR 页 = 仅 2 页

SFR 地址 = 0x95

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T2RLC	T3RLC	0	0	0	0	C1RLC	PTMRLC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: T2RLC, 这个位上写“1”强制 T2 重载. 在写入“1”之后，这个位被硬件自动清零。这个位上写“0”无作用。

Bit 6: T3RLC, 这个位上写“1”强制 T3 重载. 在写入“1”之后，这个位被硬件自动清零。这个位上写“0”无作用。

Bit 5~2: 保留。写该寄存器时，此位必须写“0”。

Bit 1: C1RLC, 定时器 1 模式 2 下，这个位上写“1”强制 PCA1 重载。在写入“1”之后，这个位被硬件自动清零。这个位上写“0”无作用。

Bit 0: PTMRLC, 定时器 0 模式 2 下，这个位上写“1”强制 PTM0 重载。在写入“1”之后，这个位被硬件自动清零。这个位上写“0”无作用。

16.4.3. 所有定时器停止的全局控制

TSPC0: 定时器停止控制寄存器 0

SFR 页 = 仅 3 页

SFR 地址 = 0x95

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T2SC	T3SC	0	0	0	0	C1SC	PTMSC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: T2SC，这个位上写“1”设置 TR2 禁止(TR2=0)。在写入“1”之后，这个位被硬件自动清零。这个位上写“0”无作用。

Bit 6: T3SC，这个位上写“1”设置 TR3 禁止(TR3=0)。在写入“1”之后，这个位被硬件自动清零。这个位上写“0”无作用。

Bit 5~2: 保留。写该寄存器时，此位必须写 “0”。

Bit 1: C1SC，这个位上写 “1” 设置 PCA1 的 C1R 禁止(C1R =0)。在写入 “1” 之后，这个位被硬件自动清零。这个位上写 “0” 无作用。

Bit 0: PTMSC，这个位上写 “1” 设置 PTM0 的 CR 禁止(CR =0)。在写入 “1” 之后，这个位被硬件自动清零。这个位上写 “0” 无作用。

17.PWM 定时器(PTM0)

MG82F6P32 配备有一个可编程计数器阵列(PCAx)和一个 PWM 定时器(PTM0)模块，该功能与标准定时/计数器相比以更少的 CPU 占用提供了更多的定时能力。它的优点包括减少了软件复杂度并提高了精度。

17.1. PTM0 概述

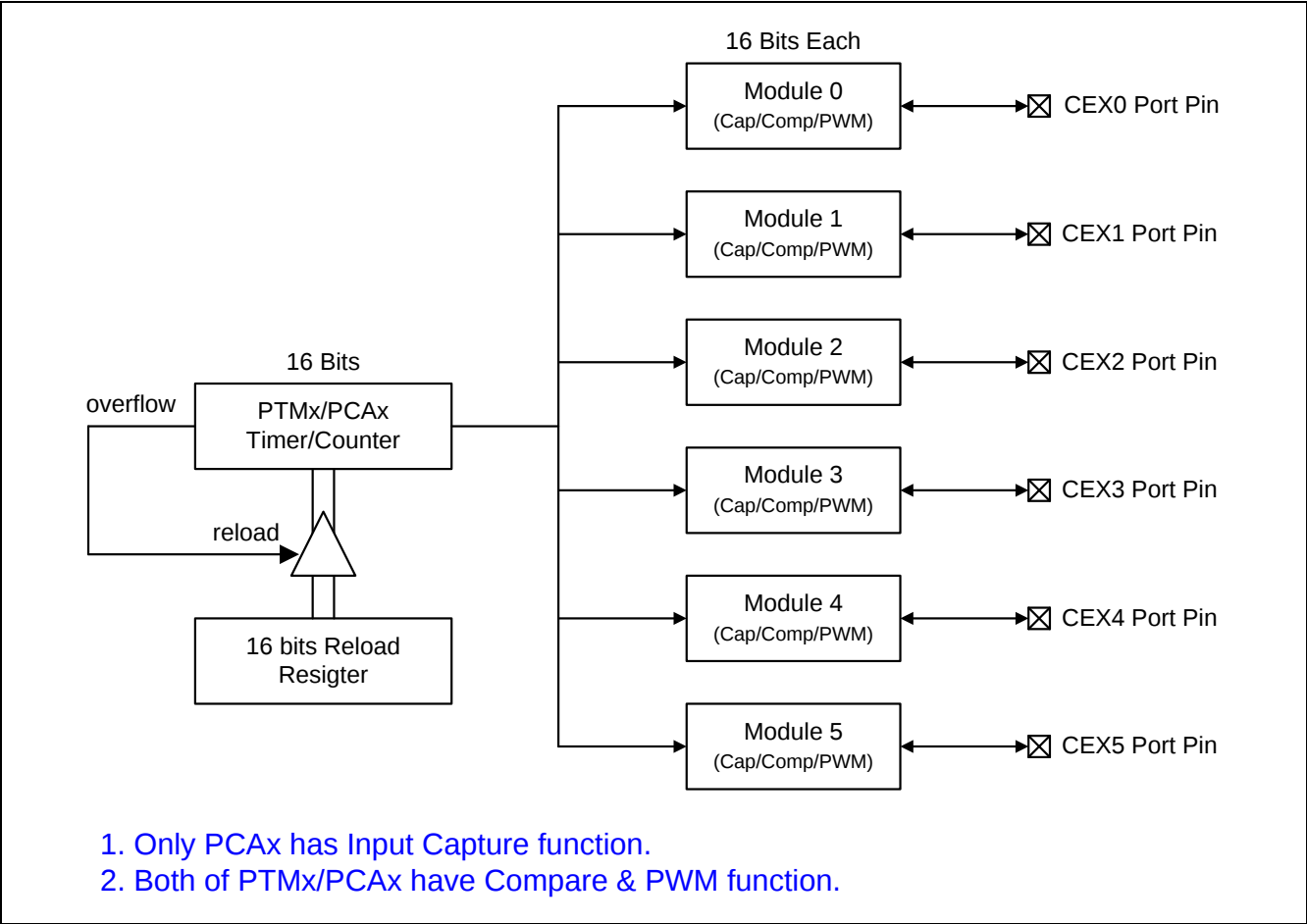
PTM0 由一个专用定时/计数器作为一个 6 组比较/ PWM 模块的时间基准。PTM0 的功能方框图如图 17-1 所示。需要注意的是 PTM0 定时器和模块都是 16 位的。如果一个外部事件同一个模块关联，那么该功能就和相应的端口引脚共享。若某模块没有使用端口引脚，这个引脚还可以用作标准 I/O。

PTM0 模块 0~5 中的任一组都可以编程为如下任意模式：

- 软件定时器(比较)
- 高速输出(比较输出)
- 相位可变比较输出
- 脉宽调制(PWM)输出

所有这些模式将在后面的章节进行详细讨论。这里，让我们先看看如何设置 PTM0 定时器和模块。

图 17-1. PTM0 方框图



17.2. PTM0 定时器/计数器

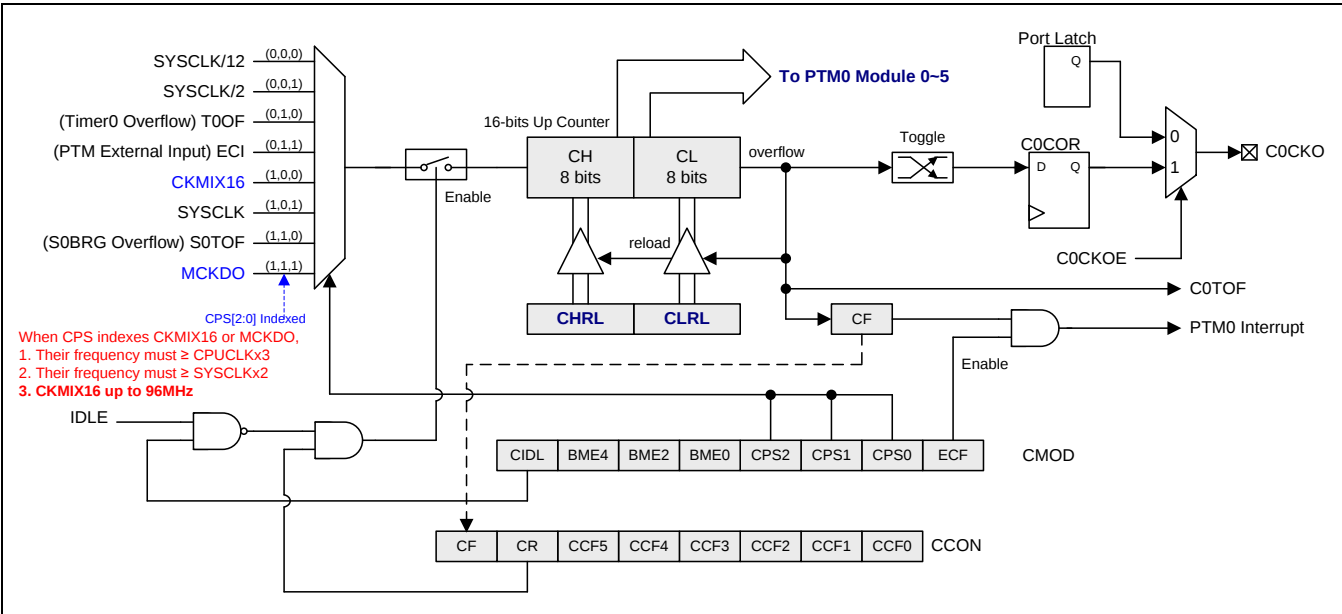
PTM0 的定时器/计数器由一个可以自动重载的 16 位定时器由寄存器 CH 和 CL(计数值的高低字节), CHRL 和 CLRL(重载寄存器的高低字节)组成, 如图 17-2 所示。{CH + CL}计数器每一次溢出 CHRL 和 CLRL 被重载入 CH 和 CL, 这样可以改变 PCA 周期时间提供可变的 PWM 周期, 比如 7 位或 9 位的 PWM。

{CH + CL}是所有模块的共有时间基准, 它的时钟输入可以从以下来源选择:

- 1/12 系统时钟频率,
- 1/2 系统时钟频率,
- 定时器 0 溢出, 可以让低频时钟源输入到定时器,
- 外部时钟输入, ECI 引脚的负跳变,
- CKMIX16, 参考章节“9.1 时钟结构”,
- 直接从系统时钟,
- S0BRG 溢出, S0TOF,
- MCKDO, 参考章节“9.1 时钟结构”。

特殊功能寄存器 CMOD 包含了计数脉冲选择位(CPS2、CPS1 和 CPS0)来指定 PTM0 定时器时钟源。当 CPS[2:0]选择 CKMIX16 或 MCKDO, 输入时钟源的频率必须 $\geq$ CPUCLK x3 或 $\geq$ SYSCLK x2。这个寄存器也包括了 ECF 位来使能计数器{CH+CL}溢出中断。并且计数器溢出切换 C0COR, 当 C0CKOE 使能将输出到端口引脚。此外, 用户可以在空闲模式下设置计数器待机位(CIDL), 来关闭 PTM0 定时器。这样可以进一步降低空闲模式下的功耗。

图 17-2. PTM0 定时器/计数器



CMOD: PTM0 计数器模式寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xD9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CIDL	BME4	BME2	BME0	CPS2	CPS1	CPS0	ECF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: CIDL, PTM0 计数器空闲模式控制。

0: PTM0 计数器在空闲模式下继续运行。

1: 空闲模式下关闭 PTM0 计数器。

Bit 6: BME4, PTM0 模块 4/5 缓冲模式使能。仅在捕捉模式, PWM 模式或 COPM 模式下的 PCA0 模块 4 和 5 有效。

0: PTM0 模块 4/5 禁止缓冲模式。

1: PTM0 模块 4/5 使能缓冲模式。

Bit 5: BME2, PCA0 模块 2/3 缓冲模式使能。仅在捕捉模式, PWM 模式或 COPM 模式下的 PCA0 模块 2 和 3 有效。

0: PTM0 模块 2/3 禁止缓冲模式。

1: PTM0 模块 2/3 使能缓冲模式。

Bit 4: BME0, PCA0 模块 0/1 缓冲模式使能。仅在捕捉模式, PWM 模式或 COPM 模式下的 PCA0 模块 0 和 1 有效。

0: PCA0 模块 0/1 禁止缓冲模式。

1: PCA0 模块 0/1 使能缓冲模式。

Bit 3~1: CPS2-CPS0, PTM0 计数器时钟源选择位。

CPS2	CPS1	CPS0	PTM0 时钟源
0	0	0	内部时钟, (system clock)/12
0	0	1	内部时钟, (system clock)/2
0	1	0	定时器 0 溢出
0	1	1	来自 ECI 引脚的外部时钟
1	0	0	CKMIX16 输出
1	0	1	内部时钟, (system clock)/1
1	1	0	S0BRT 溢出
1	1	1	MCK 分频器输出, MCKDO

注意: 当 CPS 选择 CKMIX16 或 MCKDO, 需要下面条件:

1. 时钟源的频率必须 $\geq \text{CPUCLK} \times 3$ 。
2. 时钟源的频率必须 $\geq \text{SYSCLK} \times 2$ 。

Bit 0: ECF, 使能 PTM0 计数器溢出中断。

0: 当 CF 位(CCON 寄存器中)置位时禁止中断。

1: 当 CF 位(CCON 寄存器中)置位时使能中断。

如下所示的 CCON 寄存器包含 PTM0 运行控制位和 PTM0 定时器与每个模块的标志。要运行 PTM0, CR 位 (CCON.6) 必须软件置位。要关闭 PTM0, 可以清除该位。PTM0 计数器溢出时, CF(CCON.7) 置位, 并且若 CMOD 寄存器的 ECF 置位, 还会产生一个中断。CF 位只能软件清零。CCF0 到 CCF5 是模块 0 到模块 5 的各自中断标志位, 当发生一个匹配或捕获事件时, 硬件置位, 这些位必须软件清零。PTM0 中断系统如图 17-3.. 所示。

CCON: PTM0 计数器控制寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xD8

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: CF, PTM0 计数器溢出标志

0: 只能软件清零。

1: 计数器溢出时硬件置位。CF 标志在 CMOD 寄存器的 ECF 置位时会产生一个中断。CF 可以硬件或软件置位。

Bit 6: CR, PTM0 计数器运行控制位。

0: 软件清零关闭 PTM0 计数器。

1: 软件置位开启 PTM0 计数器。

Bit 5: CCF5, PTM0 模块 5 中断标志。

0: 只能软件清零。

1: 当发生一个匹配或捕获事件时, 硬件置位。

Bit 4: CCF4, PTM0 模块 4 中断标志。

0: 只能软件清零。

1: 当发生一个匹配或捕获事件时, 硬件置位。

Bit 3: CCF3, PTM0 模块 3 中断标志。

0: 只能软件清零。

1: 当发生一个匹配或捕获事件时, 硬件置位。

Bit 2: CCF2, PTM0 模块 2 中断标志。

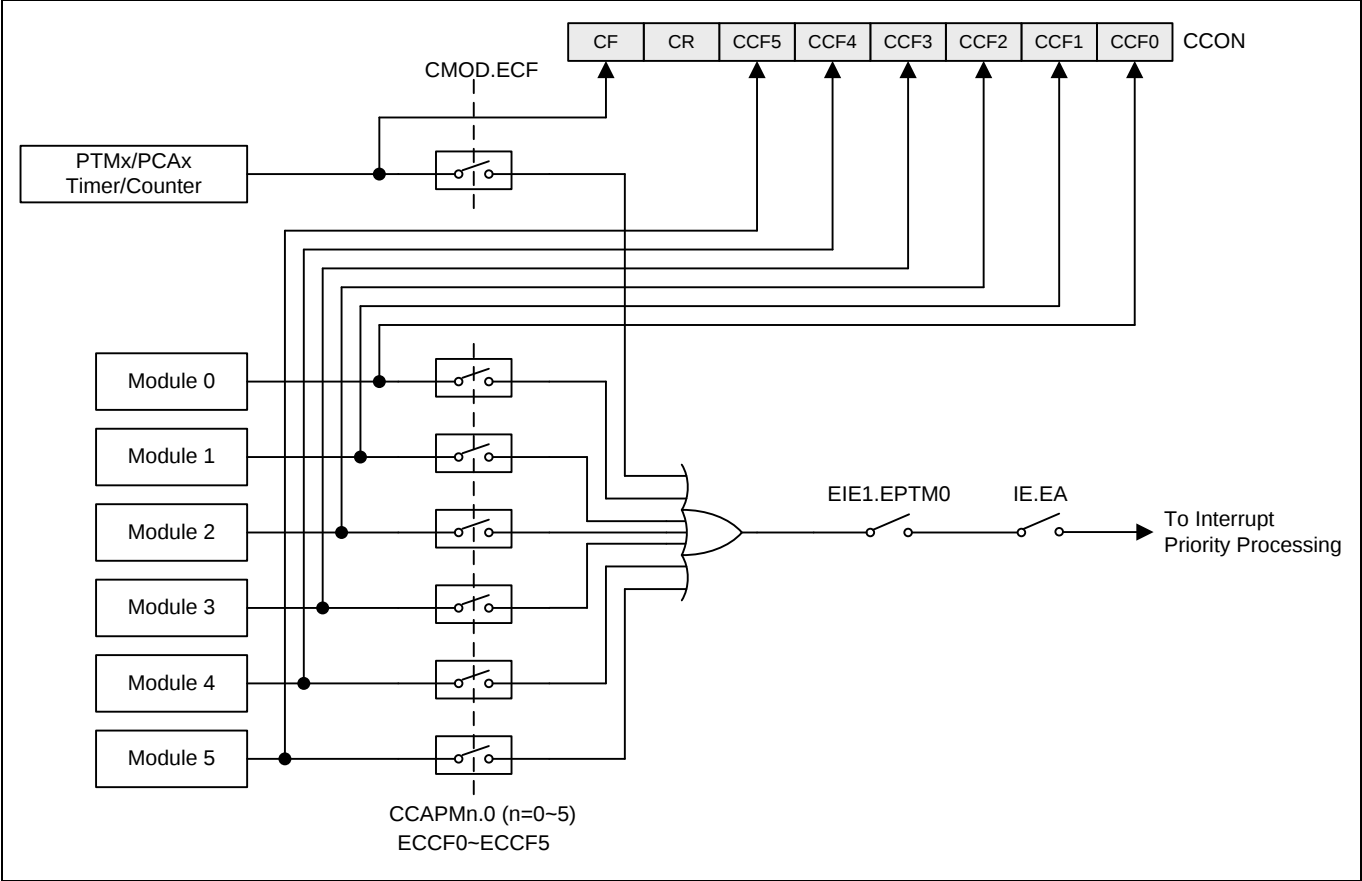
0: 只能软件清零。

1: 当发生一个匹配或捕获事件时, 硬件置位。

Bit 1: CCF1, PTM0 模块 1 中断标志。

- 0: 只能软件清零。
1: 当发生一个匹配或捕获事件时，硬件置位。
- Bit 0: CCF0，PTM0 模块 0 中断标志。
0: 只能软件清零。
1: 当发生一个匹配或捕获事件时，硬件置位。

图 17-3. PTM0 中断系统



CH: PTM0 基准时钟高字节

SFR 页 = 仅 0 页

SFR 地址 = 0xF9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CH.7	CH.6	CH.5	CH.4	CH.3	CH.2	CH.1	CH.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CL: PTM0 基准时钟低字节

SFR 页 = 仅 0 页

SFR 地址 = 0xE9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CL.7	CL.6	CL.5	CL.4	CL.3	CL.2	CL.1	CL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CHRL: PTM0 CH 重载寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xCF

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CHRL.7	CHRL.6	CHRL.5	CHRL.4	CHRL.3	CHRL.2	CHRL.1	CHRL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: CHRL, CH 的重载值.

CLRL: PTM0 CL 重载寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xCE

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CLRL.7	CLRL.6	CLRL.5	CLRL.4	CLRL.3	CLRL.2	CLRL.1	CLRL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: CLRL, CL 的重载值.

17.3. 比较模块

比较模块 0~5 中的每一组都有一个模式寄存器，叫做 CCAPMn(n=0、1、2、3、4、5)，用来选择其工作模式。ECCFn 位控制当模块中断标志置位时每个模块的中断使能。

CCAPMn: PTM 模块比较寄存器, n=0~5

SFR 页 = 仅 0 页

SFR 地址 = 0xDA~0xDF

复位值 = 0000-0000

7	6	5	4	3	2	1	0
DTE _n	ECOM _n	0	0	MAT _n	TOG _n	PWM _n	ECCF _n
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: DTE_n, 使能 PWMH_n/PWML_n 输出对的死区时间控制。此位仅在 n= 0、2 和 4 有效且当 PWM 信道操作在缓冲模式死区时间功能激活。信道的缓冲模式由 CMOD 的 BME0、BME2 或 BME4 使能

0: PWM_n 输出禁止死区时间控制。

1: PWM_n 输出使能死区时间控制。

Bit 6: ECOM_n, 比较器使能。

0: 禁止数字比较器功能。

1: 使能数字比较器功能。

Bit 3: MAT_n, 匹配控制。

0: 禁止数字比较器匹配事件去置位 CCF_n。

1: PCA 计数器同相应模块的比较/捕获寄存器匹配时 CCON 寄存器的 CCF_n 置位。

Bit 2: TOG_n, 切换控制。

0: 禁止数字比较器匹配事件去切换 CEX_n。

1: PTM 计数器同相应模块的比较/捕获寄存器匹配时使 CEX_n 引脚切换。

Bit 1: PWM_n, PWM 控制。

0: 禁止 PTM 模块中的 PWM 模式。

1: 使能 PWM 功能并使 CEX_n 引脚用作脉宽调制输出。

Bit 0: ECCF_n, 使能 CCF_n 中断。

0: 禁止 CCON 寄存器中的比较标志位 CCF_n 产生中断。

1: 使能 CCON 寄存器中的比较标志位 CCF_n 产生中断。

每个模块都有一对 8 位比较寄存器(CCAPnH, CCAPnL)。这些寄存器用来存储一个比较时间产生的时间。当模块用于 PWM 模式时，除这两个寄存器之外，一个扩展寄存器 PCAPWM_n 用来扩展输出占空比的范围，扩展的范围从 0%到 100%，步距是 1/256。关于 10/12/16 位 PWM 详情请参考章节 17.4.8 和 17.4.9。

CCAPnH: PTM0 模块 n 捕捉高寄存器高字节, n=0~5

SFR 页 = 仅 0 页

SFR 地址 = 0xFA~0xFF

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CCAPnH.7	CCAPnH.6	CCAPnH.5	CCAPnH.4	CCAPnH.3	CCAPnH.2	CCAPnH.1	CCAPnH.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CCAPnL: PTM0 模块 n 捕捉高寄存器低字节, n=0~5

SFR 页 = 仅 0 页

SFR 地址 = 0xEA~0xEF

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CCAPnL.7	CCAPnL.6	CCAPnL.5	CCAPnL.4	CCAPnL.3	CCAPnL.2	CCAPnL.1	CCAPnL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCAPWMn: PWM 模式辅助寄存器, n=0~5

SFR 页 = 仅 0 页

SFR 地址 = 0xF2~0xF7

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PnRS1	PnRS0	0	0	0	PnINV	ECAPnH	ECAPnL
R/W	R/W	W	W	R/W	R/W	R/W	R/W

Bit 7~6: PnRS1~0, PWMn 分辨率设置位 1~0。

00: 8 位 PWMn, 当[CH, CL]计数从 XXXX-XXXX-1111-1111 到 XXXX-XXXX-0000-0000 时溢出激活。

01: 10 位 PWMn, 当[CH, CL]计数从 XXXX-XX11-1111-1111 到 XXXX-XX00-0000-0000 时溢出激活。

10: 12 位 PWMn, 当[CH, CL]计数从 XXXX-1111-1111-111 到 XXXX-0000-0000-0000 时溢出激活。

11: 16 位 PWMn, 当[CH, CL]计数从 1111-1111-1111-1111 到 0000-0000-0000-0000 时溢出激活。

Bit 5~3: 保留。写该寄存器时, 此位必须写“0”。

Bit 2: PnINV, 比较/PWM 输出(C0PnOR)在 CEXn 引脚上反转。

0: 比较/PWM 输出(C0PnOR)不反转。

1: 比较/PWM 输出(C0PnOR)反转。

Bit 1: ECAPnH, 扩展第 9 位(MSB), 联合 CCAPnH 形成 9 位寄存器用于 PWM 模式。

Bit 0: ECAPnL, 扩展第 9 位(MSB), 联合 CCAPnL 形成 9 位寄存器用于 PWM 模式。

17.4. PCAx/PTMx 操作模式

不同 PTM0 功能对应的 CCAPMn 寄存器设置如表 17-1 所示

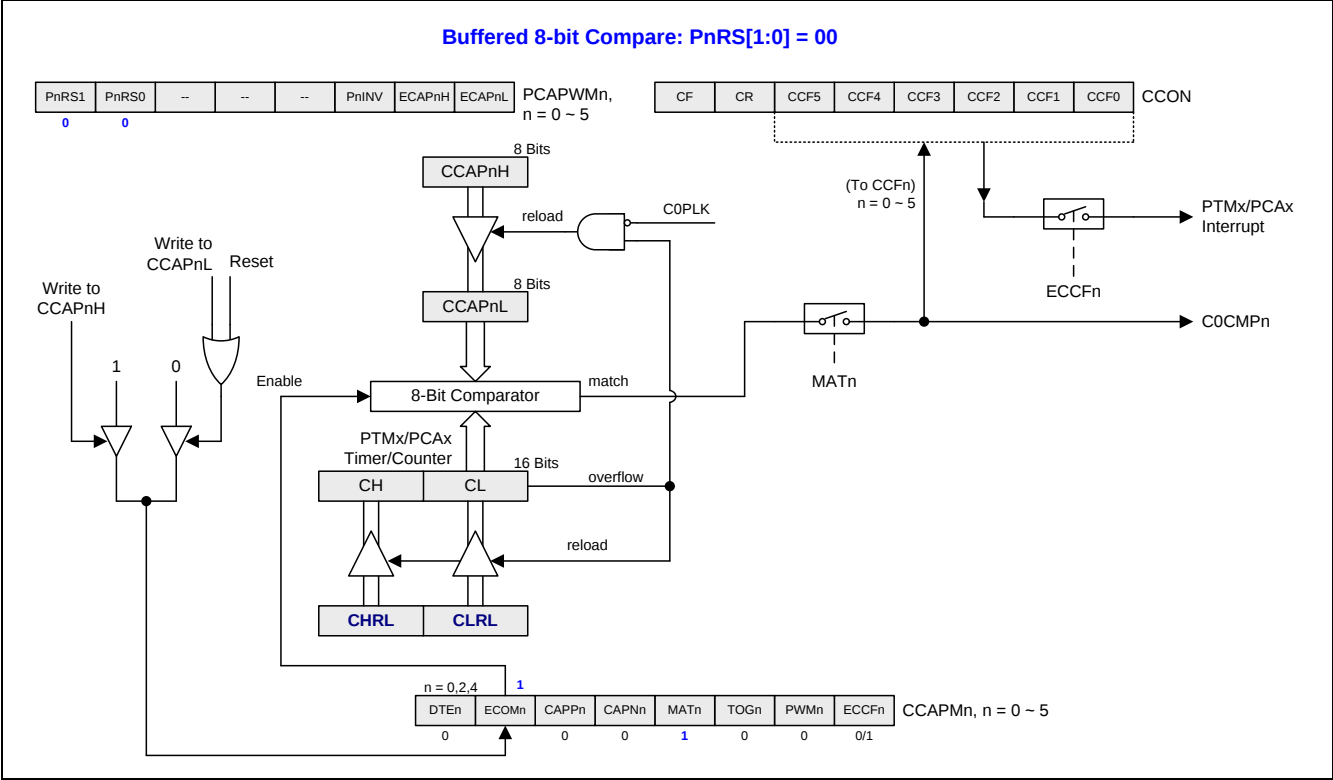
表 17-1. PCAx/PTMx 模块模式

ECOMn	MATn	TOGn	PWMn	BMEn	模块功能
0	0	0	0	0	无操作
1	1	0	0	0	软件定时器(比较)
1	1	0	0	1	缓冲软件定时器(比较)
1	1	1	0	0	高速输出(输出比较)
1	1	1	0	1	相位可变比较输出
1	0/1	0	1	0	脉宽调制器 (PWM)
1	0/1	0	1	1	缓冲 10/12/16 位 PWM

17.4.1. 8 位软件定时器模式 (比较模式)

PTM 模块可以通过设置 CCAPMn 寄存器的 ECOM 位和 MAT 位来作为一个软件定时器使用。PTM 定时器与模块的捕获寄存器值进行比较，若相等且当 CCFn 和 ECCFn 位同时设置时会产生一个中断信号。

图 17-4. PCA 软件定时器模式

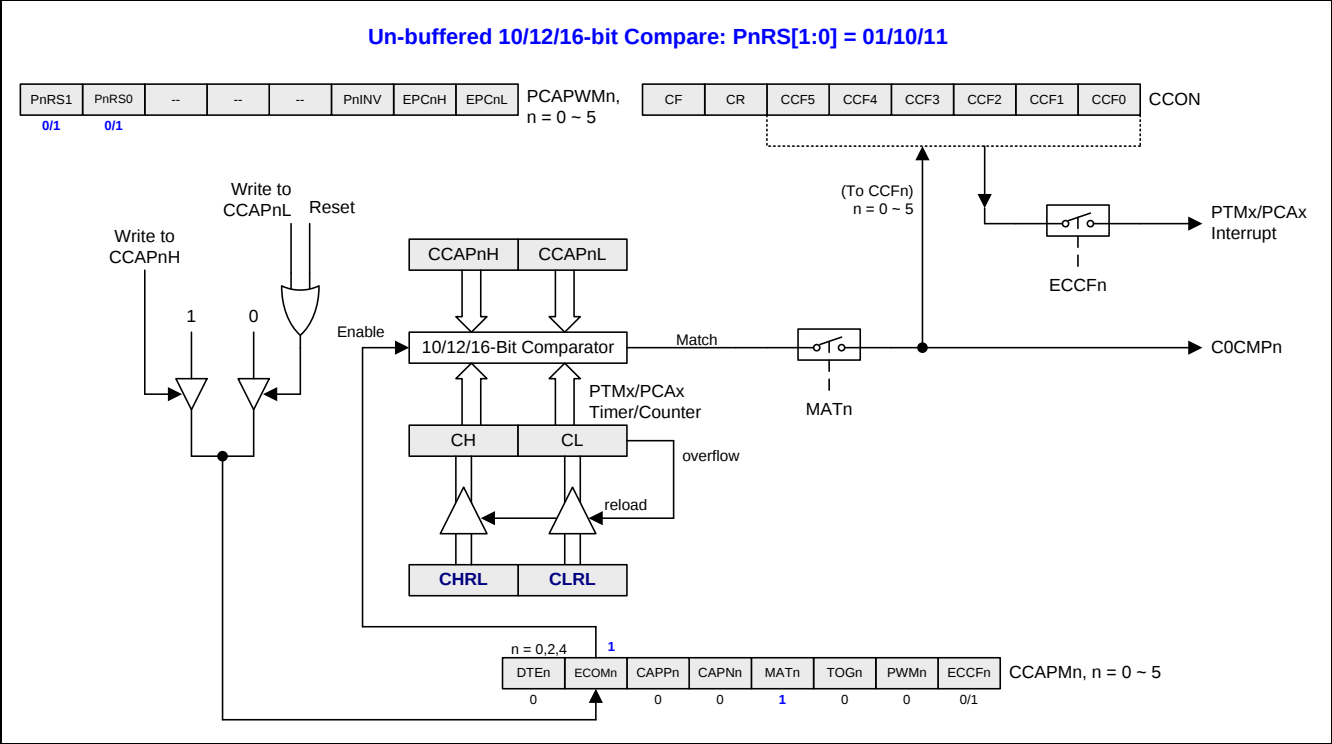


17.4.2. 16 位软件定时器模式(比较模式)

PTM 模块可以通过设置 CCAPMn 寄存器的 ECOM 位和 MAT 位来作为一个软件定时器使用。PTM 定时器与模块的捕获寄存器值进行比较，若相等且当 CCFn 和 ECCFn 位同时设置时会产生一个中断信号。

由于 CL 计数器的变化比较频繁，在将比较值写入 CCAPnL 时可能会造成不匹配。为了防止写 CCAPnL 时不匹配，它会在写命令后自动将 ECOMn 设置为 0 以禁用比较器。对 CCAPnH 执行写命令后，ECOMn 恢复为 1。如果应用程序只更改 CCAPnL，则应该在写入 CCAPnL 后将 ECOMn 设置为 1 以恢复比较器

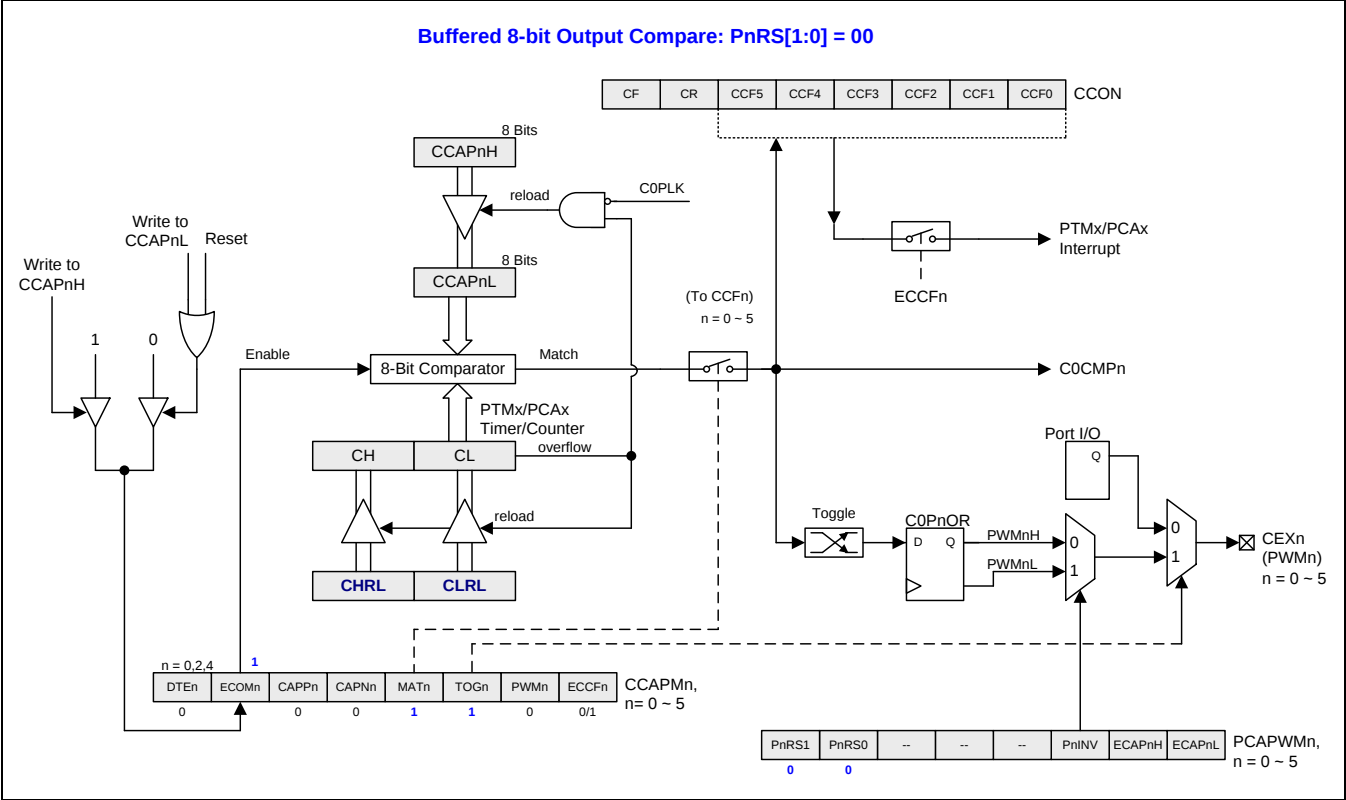
图 17-5. PTM 软件定时器模式



17.4.4. 8 位高速输出模式(比较输出模式)

这种模式下，每当 PTM 计数器与模块捕获寄存器(CCAPnH 和 CCAPnL)值相等时，与 PCA 模块相关联的 CEX 输出就会触发。为激活这种模式，CCAPMn 寄存器的 TOG、MAT 和 ECOM 位必须都置为 1。

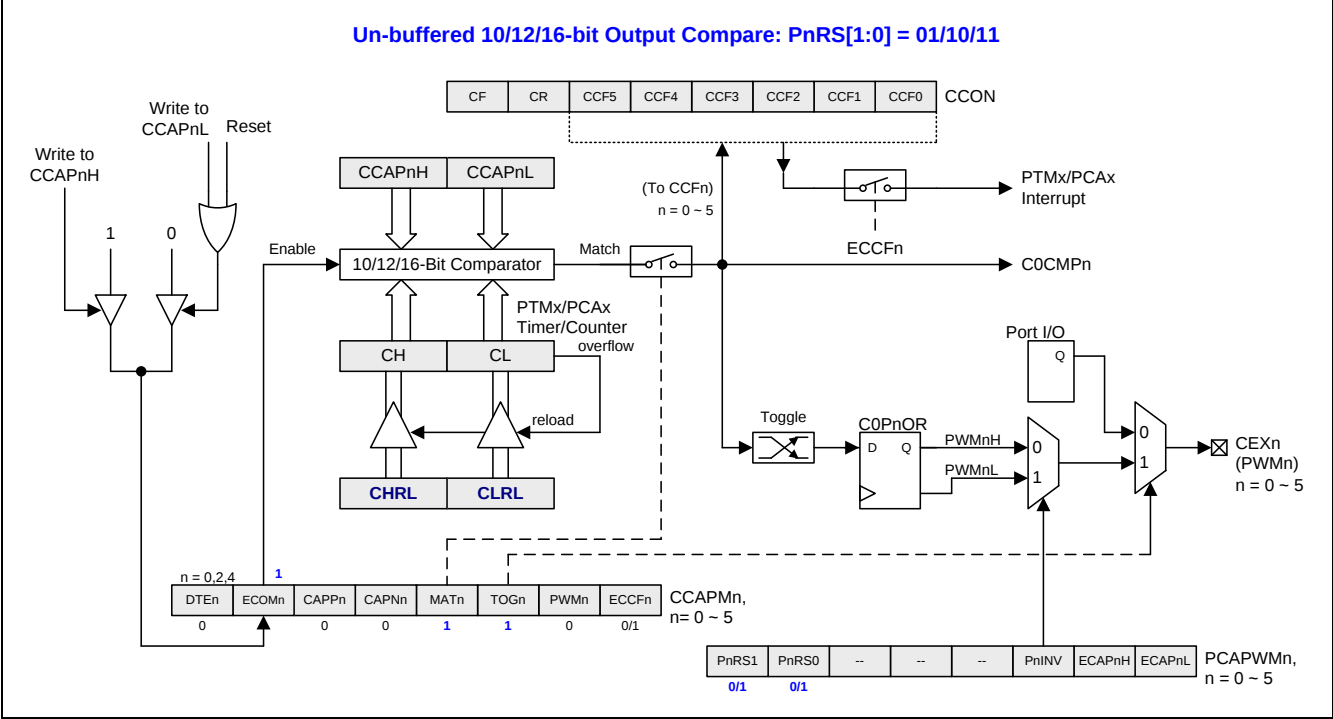
图 17-7. PTM0 高速输出模式



17.4.5. 16 位高速输出模式 (比较输出模式)

这种模式下，每当 PTM 计数器与模块捕获寄存器(CCAPnH 和 CCAPnL)值相等时，与 PCA 模块相关联的 CEX 输出就会触发。为激活这种模式，CCAPMn 寄存器的 TOG、MAT 和 ECOM 位必须都置为 1。

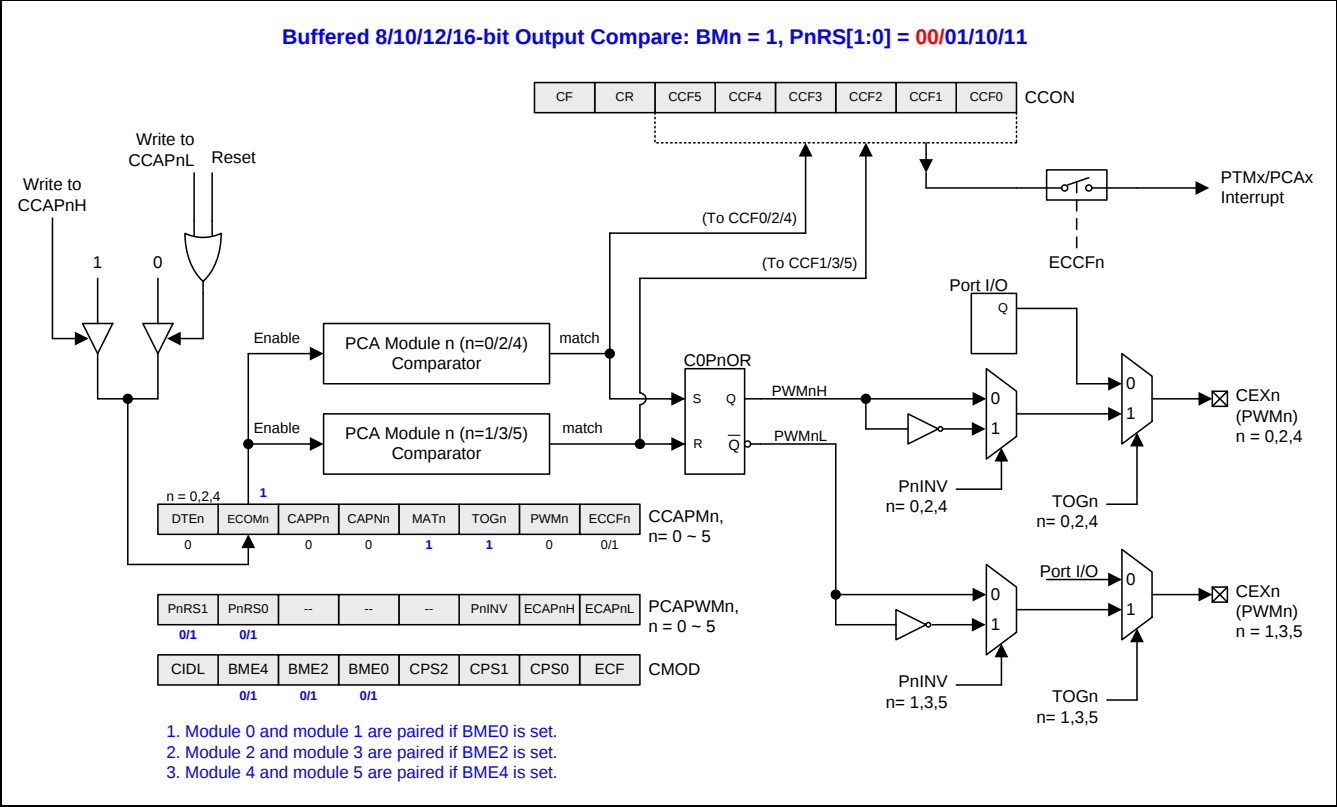
图 17-8. PTM 高速输出模式



17.4.6. 相位可变比较输出模式

这种模式下，每当 PTM 计数器与模块捕获寄存器(CCAPnH 和 CCAPnL)值相等时，与 PCA 模块相关联的 CEX 输出就会触发。为激活这种模式，CCAPMn 寄存器的 TOG、MAT 和 ECOM 位必须都置为 1。

图 17-9. PTM 高速输出模式



17.4.7. 缓冲 8 位 PWM 模式

所有 PTM0 模块都可用作 PWM 输出。输出频率取决于 PTM0 定时器的时钟源。所有的模块都有相同的输出频率，因为它们共享 PTM0 定时器。

占空比取决于模块捕获寄存器 CCAPnL 与扩展的第 9 位 ECAPnL 的值。当 9 位数据{0,[CL]}值小于{ ECAPnL, [CCAPnL] }组成的 9 位数据时，输出低电平，相等或大于时输出高电平。

当 CL 从 0xFF 到 0x00 溢出时，{ ECAPnL, [CCAPnL] } 的值使用{ ECAPnH,[CCAPnH] }的值重载，这样可以允许无异常的情况下更新 PWM。模块的 CCAPMn 寄存器 PWMn 和 ECOMn 位必须置位以使能 PWM 模式。

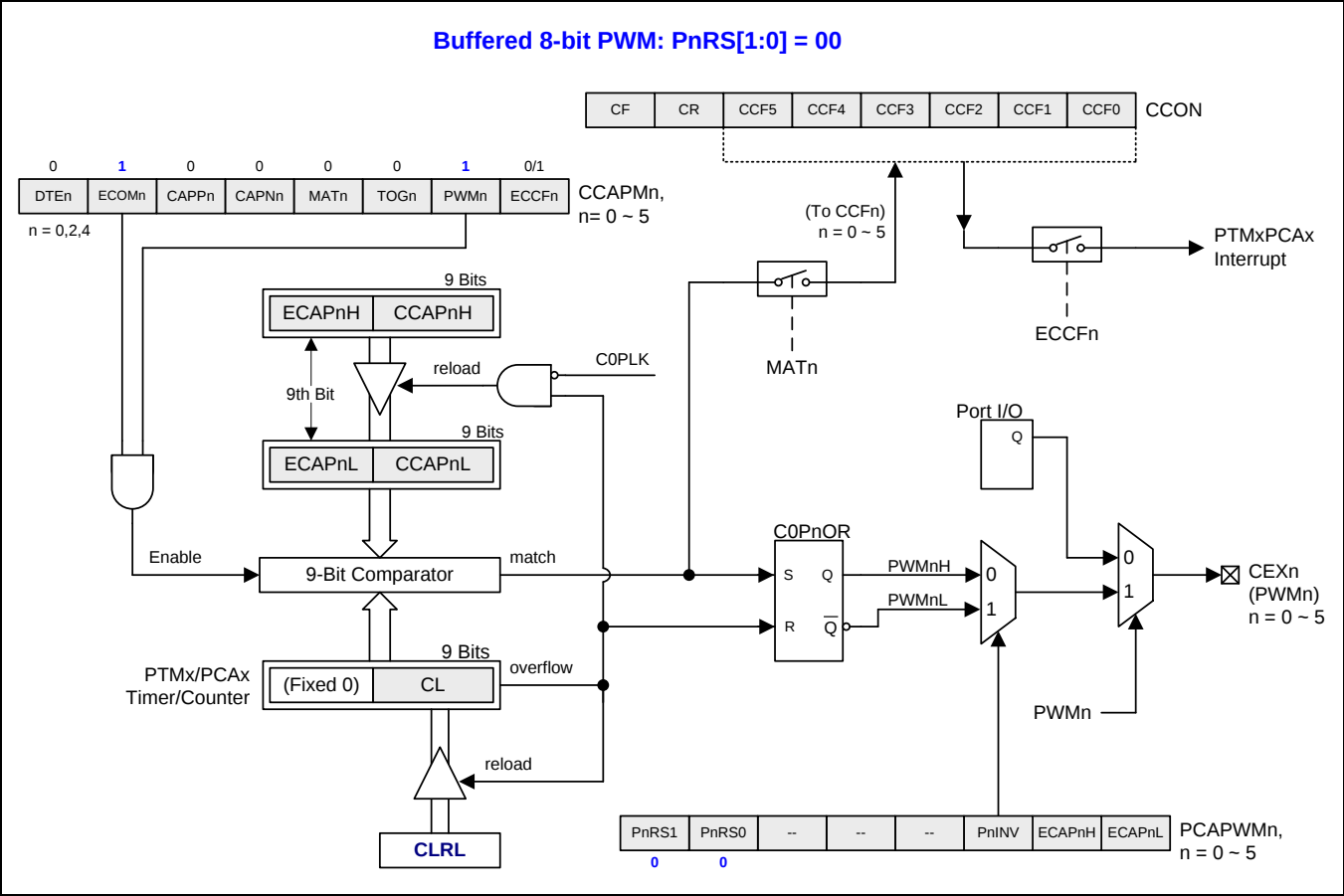
使用 9 位比较，输出的占空比可以真正实现从 0%到 100%可调。占空比计算公式如下：

占空比 = 1 - { ECAPnH, [CCAPnH] } / 256.

这里， [CCAPnH]是 CCAPnH 寄存器的 8 位值， ECAPnH(PCAPWMn 寄存器的第 1 位)是 1 位值。所以， { ECAPnH, [CCAPnH] } 组成了 9 位比较器用的 9 位值。

- 例如，
- a. 若 ECAPnH=0 且 CCAPnH=0x00 (即 9 位值, 0x000)， 占空比为 100%.
 - b. 若 ECAPnH=0 且 CCAPnH=0x40 (即 9 位值, 0x040)， 占空比为 75%.
 - c. 若 ECAPnH=0 且 CCAPnH=0xC0 (即 9 位值, 0x0C0)， 占空比为 25%.
 - d. 若 ECAPnH=1 且 CCAPnH=0x00 (即 9 位值, 0x100)， 占空比为 0%.

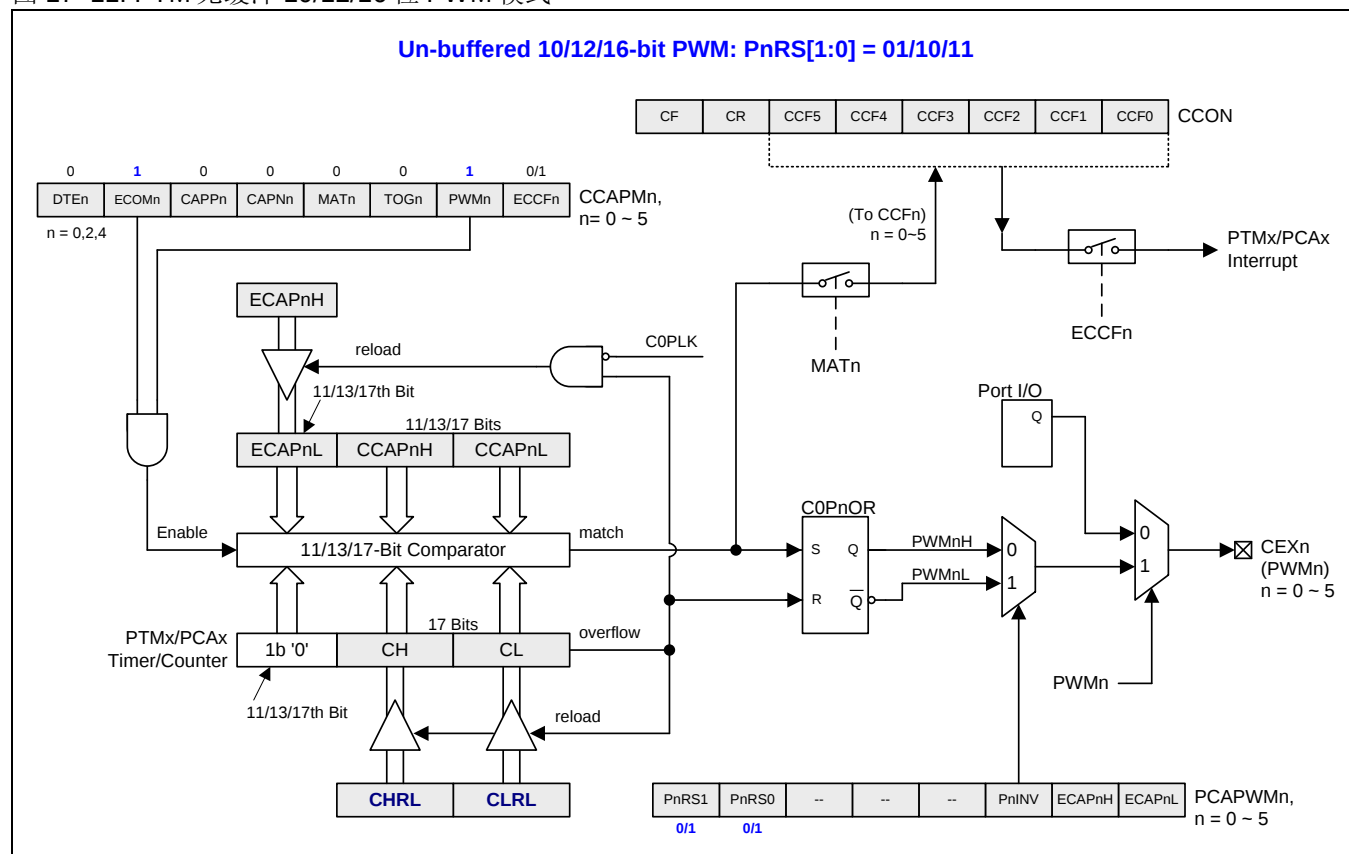
图 17–10. PTM0 缓冲 8 位 PWM 模式



17.4.8. 无缓冲 10/12/16 位 PWM 模式

PTM 提供可变的 PWM 模式以增强控制能力。有额外无缓冲的 10/12/16 位 PWM 被分配给每一路及每一路 PWM 有不同的分辨率操作能力。

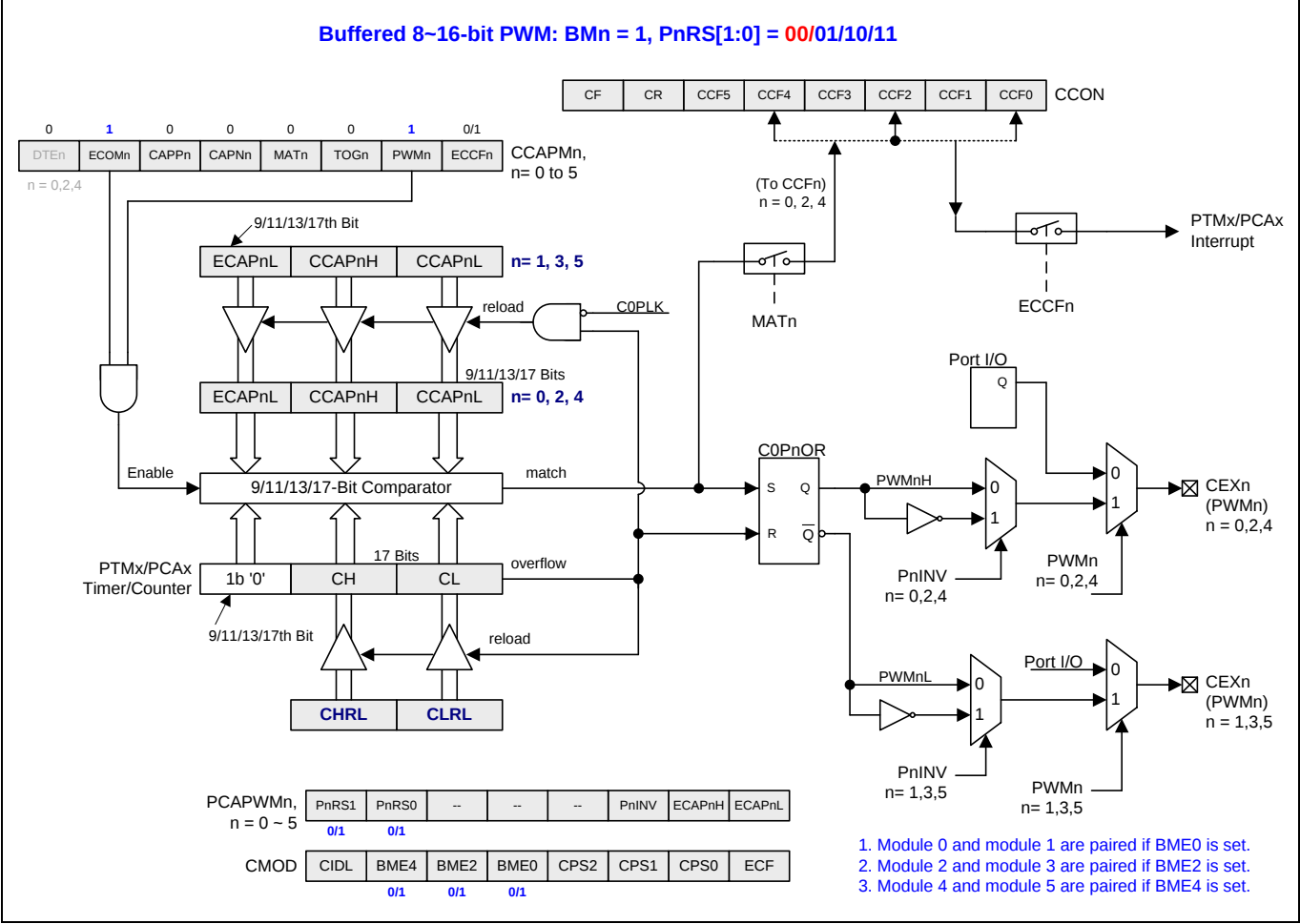
图 17-11. PTM 无缓冲 10/12/16 位 PWM 模式



17.4.9. 缓冲 10/12/16 位 PWM 模式

如果使用 10/12/16 位 PWM 模式，在将数据写入 CCAPnH 和 CCAPnL 时，将会导致意外的占空比，因为 8 位 CPU 每次只能写一个字节。要完成完全设置，需要两个写周期，当第一个字节被写入时，比较器将输出意料之外的占空比。如果应用程序在更改占空比时需要精确控制，则需要使用缓冲 PWM 模式。

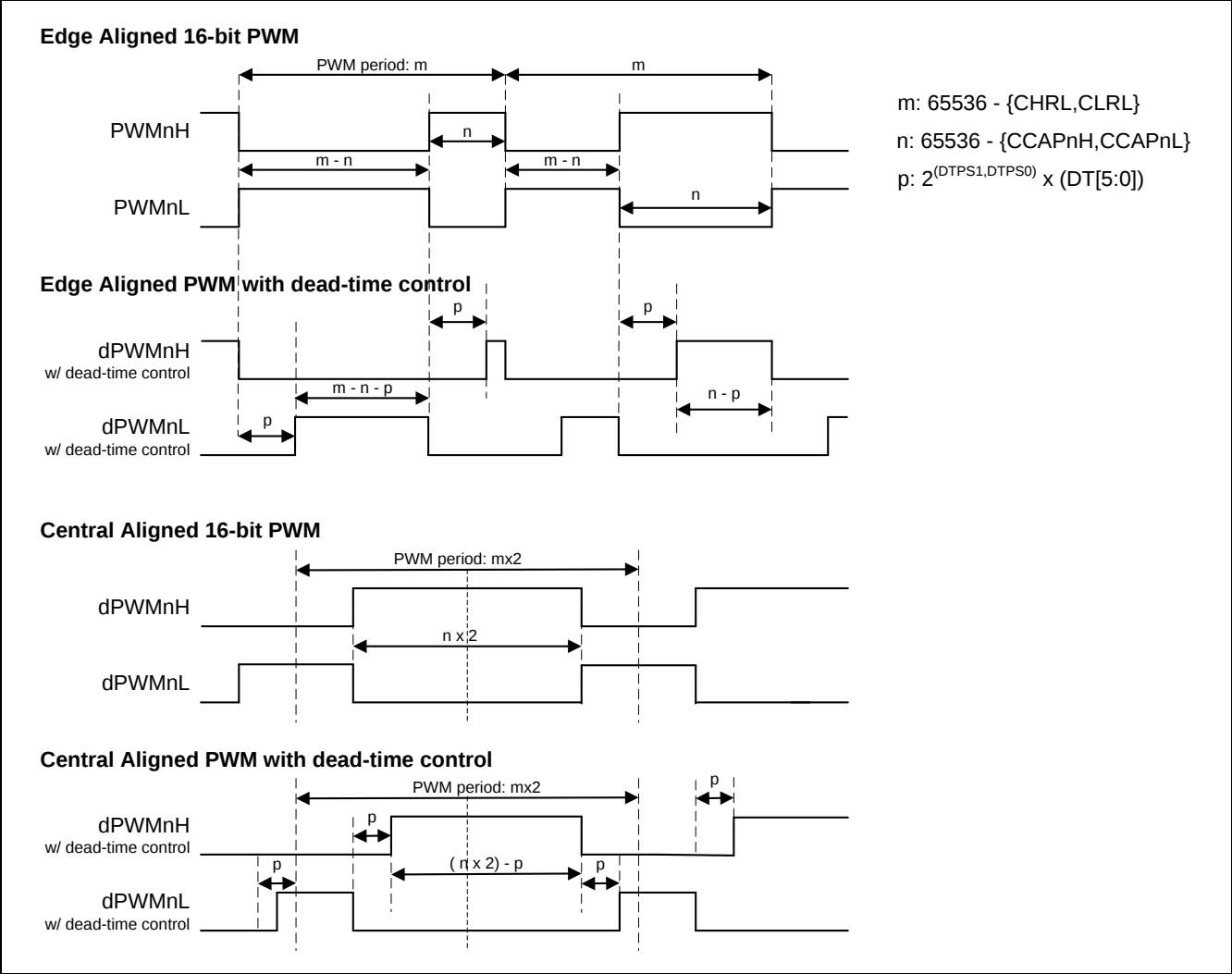
图 17-12. PTM 缓冲 10/12/16 位 PWM 模式 (带死区控制)



17.4.10. PTM0 增强型 PWM 控制

PCA 提供可变的 PWM 模式增强 PWM 应用的控制能力。这些额外的 10/12/16 位 PWM 可以分配到每个信道且每个 PWM 信道可以同时运行在不同的分辨率和不同的相位延时。

图 17–13. PWM 带死区控制的波形



CCAPMn: PTM 模块比较捕捉寄存器, n=0~5

SFR 页 = 仅 0 页

SFR 地址 = 0xDA~0xDF

复位值 = 0000-0000

7	6	5	4	3	2	1	0
DTE _n	ECOM _n	0	0	MAT _n	TOG _n	PWM _n	ECCF _n
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: DTE_n, 使能 PWM_nH/PWM_nL 输出对的死区时间控制。此位仅在 n= 0、2 和 4 有效且当 PWM 信道操作在缓冲模式死区时间功能激活。信道的缓冲模式由 CMOD 的 BME0、BME2 或 BME4 使能

0: PWM_n 输出禁止死区时间控制。

1: PWM_n 输出使能死区时间控制。

PDTCRA: PWM 死区控制寄存器 A

SFR 页 = 仅 1 页

SFR 地址 = 0xBC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
DTPS1	DTPS0	DT5	DT4	DT3	DT2	DT1	DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: DTPS1~0, 死区计数器的时钟预分频.

DTPS[1:0]	预分频选择
00	SYSCLK
01	SYSCLK/2
10	SYSCLK/4
11	SYSCLK/8

Bit 5~0: DT5~0, 死区时间控制位

DT[5:0]	死区时间
000000	禁止死区时间
000001	预分频器时钟 X 1
000010	预分频器时钟 X 2
000011	预分频器时钟 X 3
.....	
111110	预分频器时钟 X 62
111111	预分频器时钟 X 63

PWMCR: PWM 控制寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xBC

复位值 = 0000-0000

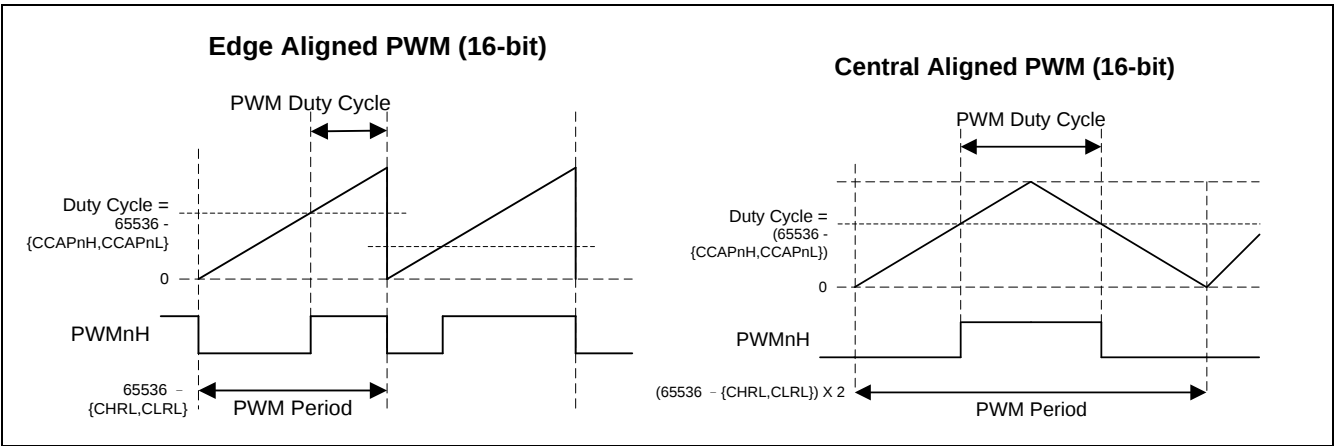
7	6	5	4	3	2	1	0
PCAE	C0OFS	PBKM	PBKE1.1	PBKE1.0	PBKE0.2	PBKE0.1	PBKE0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: PCAE, PWM 中心对齐使能. PCAE 控制使能的 PWM 通道为中心对齐调制包括 PWM 的缓冲模式或非缓冲模式. 在这个 PWM 模式下, PWM 频率是半边沿对齐模式. 此功能仅在 PWM0~5 有效.

0: 设置 PWM 功能为边沿对齐调制.

1: 使能 PWM 功能为中心对齐调制. 仅支持 CHRL 和 CLRL 设定的 8/10/12/16 位分辨率.

图 17-14. 边沿对齐和中心对齐的 PWM 波形



Bit 6: C0OFS, PTM0 溢出标志选择.

0: PWM 中心对齐周期的顶部 CF 置位.

1: PWM 中心对齐周期的低部 CF 置位.

Bit 5: PBKM, PWM 中止模式选择.

0: 锁存模式.

1: 逐周期模式.

图 17-15. PWM 中止控制的锁存模式波形

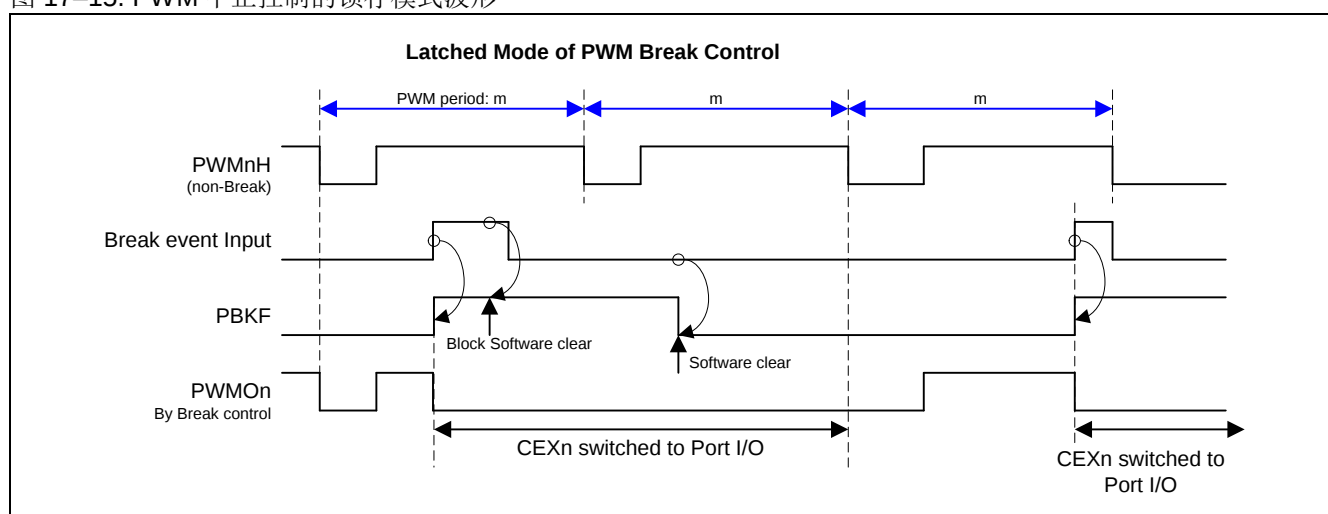
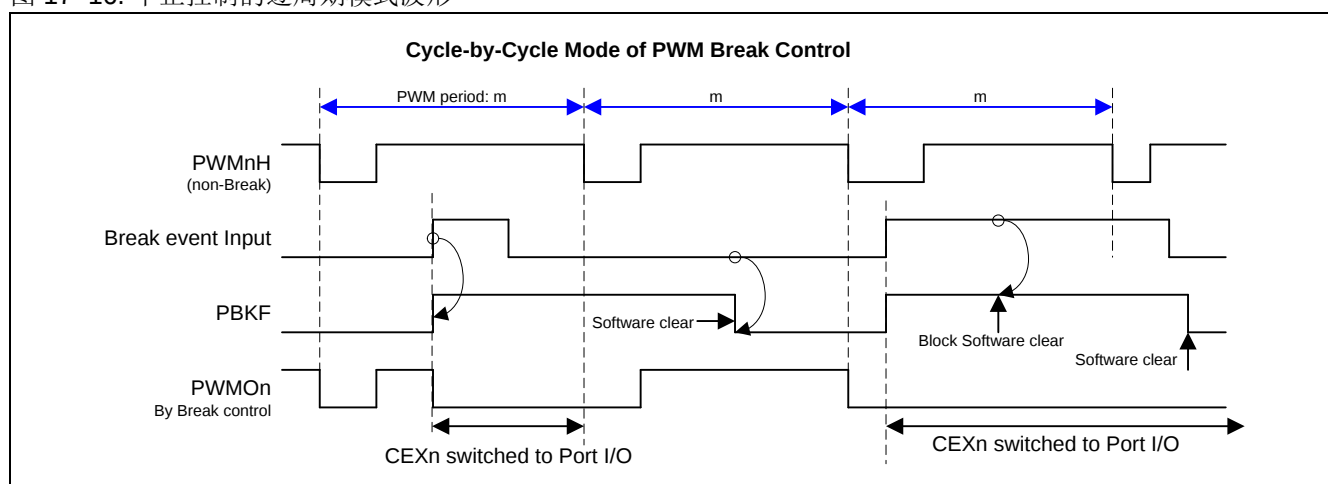


图 17-16. 中止控制的逐周期模式波形



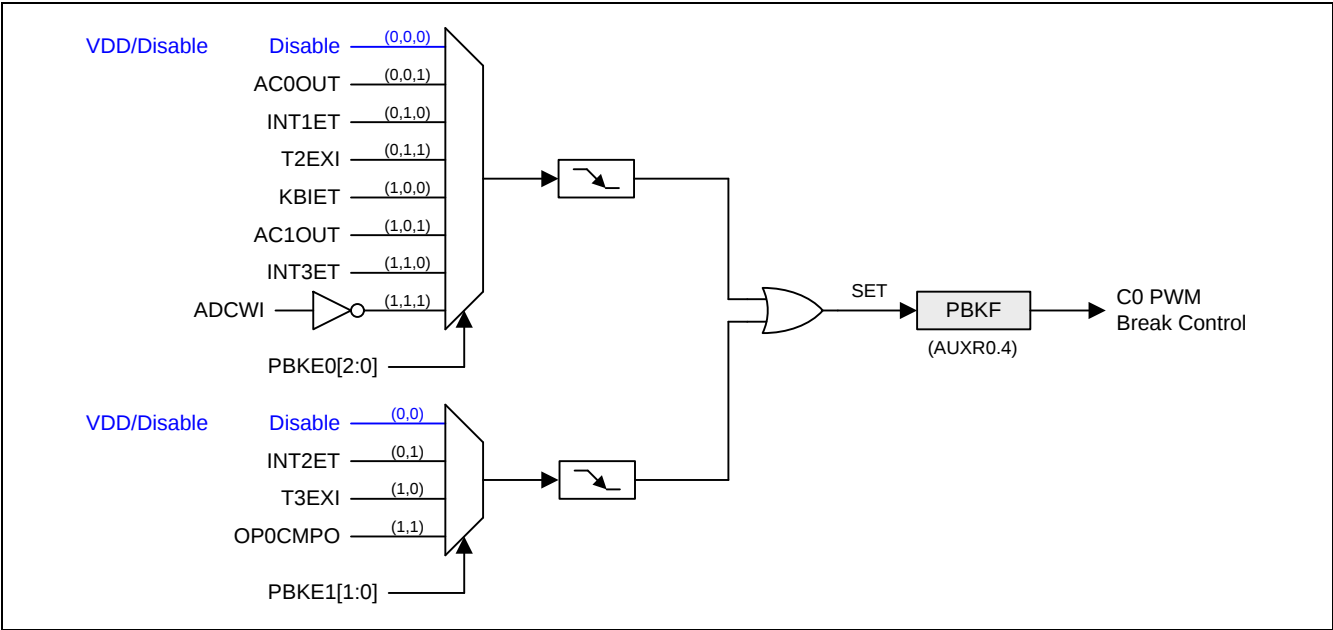
Bit 4~3: PBKE1.1~0, 中止使能 1 选择。此功能仅在 CEXn 输出模式有效(n=0~5)

PBKE1[1:0]	PWM 中止源
0 0	禁止 PWM 中止源 1
0 1	INT2ET, nINT2 激活
1 0	T3EXI
1 1	OP0ES

Bit 2~0: PBKE0.2~0, PWM 中止使能 0 选择。此功能仅在 CEXn 输出模式有效(n=0~5)

PBKE0[2:0]	PWM 中止源
0 0 0	禁止 PWM 中止源 0
0 0 1	AC0OUT
0 1 0	INT1ET, nINT1 激活
0 1 1	T2EXI
1 0 0	KBIET, KBI 匹配激活
1 0 1	AC1OUT
1 1 0	INT3ET, nINT3 激活
1 1 1	ADCWI 激活

图 17-17. PTM0 PWM 中止控制源



AUXR0: 辅助寄存器 0

SFR 页 = 0~F

SFR 地址 = 0xA1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C1BKF	PBKF	0	0	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: PBKF, PWM 中止事件标志。此位由 PWM 中止源使能设置。如果此位设置，则使能的 PWM 通道 0~5 将被锁住并且输出引脚保持最初的 GPIO 状态。
0: 没有 PWM 中止事件出现。仅由软件清零。
1: PWM 中止事件出现或软件触发一个 PWM 中止。

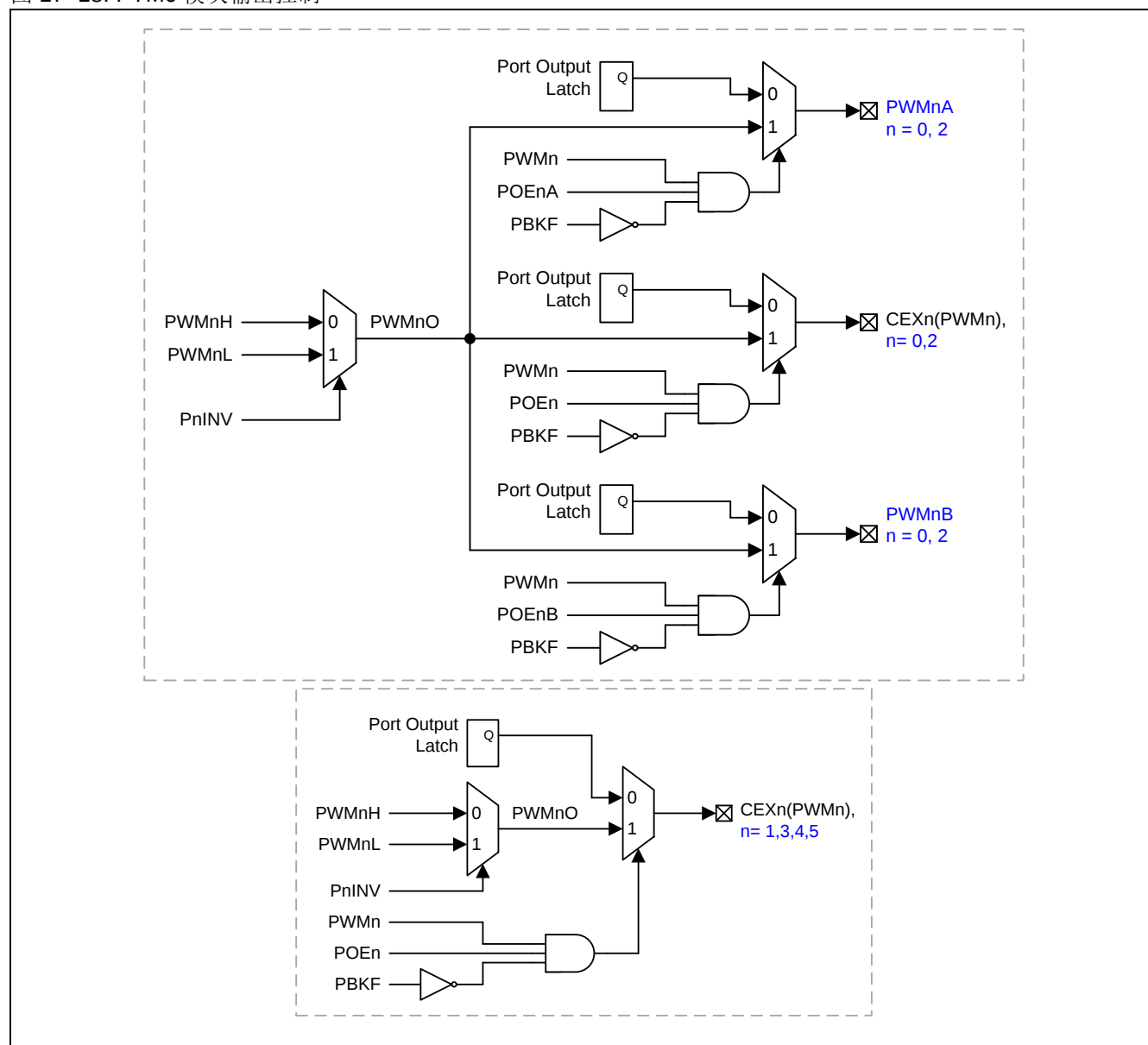
17.4.11. PTM0 模块输出控制

PTM0 模块具有多种输出控制模式，可以选择用于不同的应用。CEXn (n=1、3、4、5、6、7) 可编程为一般 I/O 端口或 PTM0 模块(PWM) 1、3、4、5、6 和 7 的输出。当 PWM 被分配到 CEXn 时，PnINV 可以在正常或反向之间切换 PWM 信号。POEn 可以用来启用或禁用端口引脚输出 PWM 信号。

CEXn (n=1、3、4、5) 可以使用 PBKF，PWM 中止标志，中止 PWM 输出。如果此位设置，则使能的 PWM 通道 0~5 将被锁住并且输出引脚保持最初的 GPIO 状态。

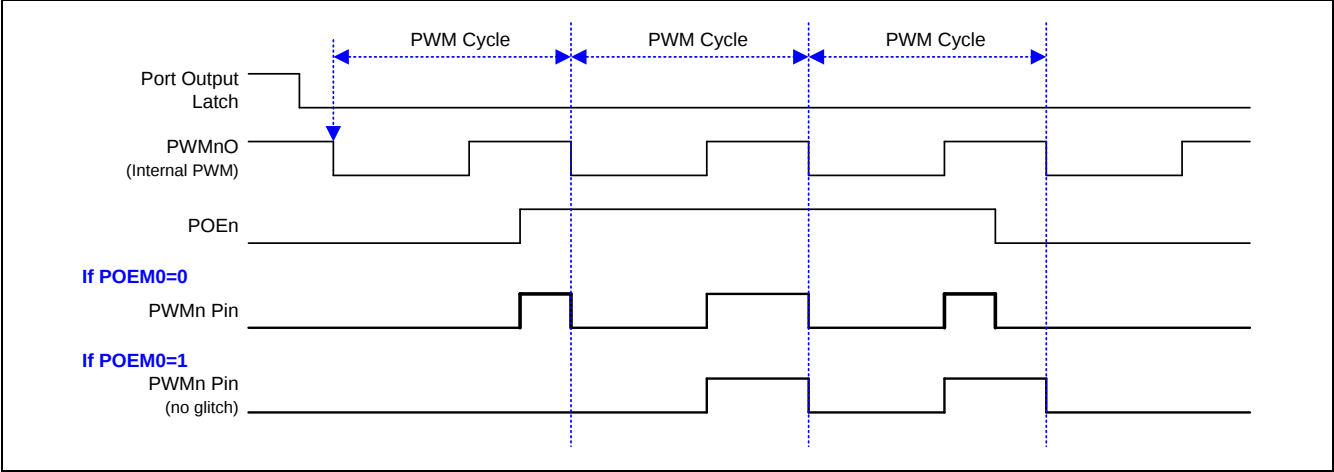
另外，PCA 模块 0 和 2 在不同的端口引脚有 2 个“克隆”信号。这三个相同的 PWM 信号可以被 POEnA 或 POEnB 或 PWMn 屏蔽，用于需要相位控制的应用程序。

图 17-18. PTM0 模块输出控制



MG82F6P32 的 POEM0 控制 POEn 输出时序与 PWM 周期对齐。这个对齐功能的配置和波形如图 17-19 所示。

图 17-19. POEn 对齐输出控制 (例如.PWM 边沿对齐的波形)



PAOE: PWM 额外输出使能寄存器

SFR 页 = 0~7

SFR 地址 = 0xF1

复位值 = 1001-1001

7	6	5	4	3	2	1	0
POE3	POE2B	POE2A	POE2	POE1	POE0B	POE0A	POE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: POE3, PTM0 PWM3 主通道(PWM3O)输出控制。

0: 禁止 PWM3O 在端口引脚输出。

1: 使能 PWM3O 在端口引脚输出。默认是使能。

Bit 6: POE2B, PTM0 PWM2 3rd通道(PWM2B)输出控制。

0: 禁止 PWM2B 在端口引脚输出。默认是禁止。

1: 使能 PWM2B 在端口引脚输出。

Bit 5: POE2A, PTM0 PWM2 2nd通道(PWM2A)输出控制。

0: 禁止 PWM2A 在端口引脚输出。默认是禁止。

1: 使能 PWM2A 在端口引脚输出。

Bit 4: POE2, PTM0 PWM2 主通道(PWM2O)输出控制。

0: 禁止 PWM2O 在端口引脚输出。

1: 使能 PWM2O 在端口引脚输出。默认是使能。

Bit 3: POE1, PTM0 PWM1 主通道(PWM1O)输出控制。

0: 禁止 PWM1O 在端口引脚输出。

1: 使能 PWM1O 在端口引脚输出。默认是使能。

Bit 2: POE0B, PTM0 PWM0 3rd通道(PWM0B)输出控制。

0: 禁止 PWM0B 在端口引脚输出。默认是禁止。

1: 使能 PWM0B 在端口引脚输出。

Bit 2: POE0B, PTM0 PWM0 3rd通道(PWM0B)输出控制。

0: 禁止 PWM0B 在端口引脚输出。默认是禁止。

1: 使能 PWM0B 在端口引脚输出。

Bit 0: POE0, PTM0 PWM0 主通道(PWM0O)输出控制。

0: 禁止 PWM0O 在端口引脚输出。

1: 使能 PWM0O 在端口引脚输出。默认是使能。

C0AOE1: PWM 额外输出使能寄存器 1

SFR 页 = 8

SFR 地址 = 0xF1

复位值 = 0XXX-1001

7	6	5	4	3	2	1	0
POEM0				POE5			POE4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: POEM0, PTM0 POEn 控制 0.

0: POEn 功能在 CPU 写入后立即生效

1: POEn 功能与 PWM 周期对齐

Bit 3: POE5, PCA PWM5 主通道(PWM5O)输出控制。

0: 禁止 PWM5O 在端口引脚输出。

1: 使能 PWM5O 在端口引脚输出。默认是使能。

Bit 0: POE4, PCA PWM4 主通道(PWM4O)输出控制。

0: 禁止 PWM4O 在端口引脚输出。

1: 使能 PWM4O 在端口引脚输出。默认是使能。

AUXR5: 辅助寄存器 5

SFR 页 = 仅 2 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0PPS1	C0PPS0	EC1IPS0	C1COPS	ECIPS0	C0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: C0PPS1, {PWM2A, PWM2B} 端口引脚选择 1.

C0PPS1	PWM2A	PWM2B
0	P4.0	P4.1
1	P3.4	P3.5

Bit 4: C0PPS0, {PWM0A, PWM0B} 端口引脚选择 0.

C0PPS0	PWM0A	PWM0B
0	P1.6	P1.7
1	P6.0	P6.1

Bit 3: EC1IPS0, PCA1 ECI 端口引脚选择 0.

EC1IPS0	ECI
0	ECI
1	P1.5

Bit 2: C1COPS, PCA1 时钟输出 (C1CKO) 端口引脚选择

C1COPS	C1CKO
0	P4.7
1	P3.3

Bit 1: ECIPS0, PTM0 ECI 端口引脚选择 0.

ECIPS0	ECI
0	P0.2
1	P1.6

Bit 0: C0COPS, PTM0 时钟输出 (C0CKO) 端口引脚选择

C0COPS	C0CKO
0	P4.7
1	P3.3

AUXR7: 辅助寄存器 7

SFR 页 = 仅 4 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0CKOE	C1CKOE	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: C0CKOE, PTM0 时钟输出使能

0: 禁止 PTM0 时钟输出。

1: 使能 PTM0 基准定时器溢出率二分频的时钟输出。

AUXR16: 辅助寄存器 16

SFR 页 = 仅 D 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~0: C0OPS5 ~0, PTM 输出引脚选择

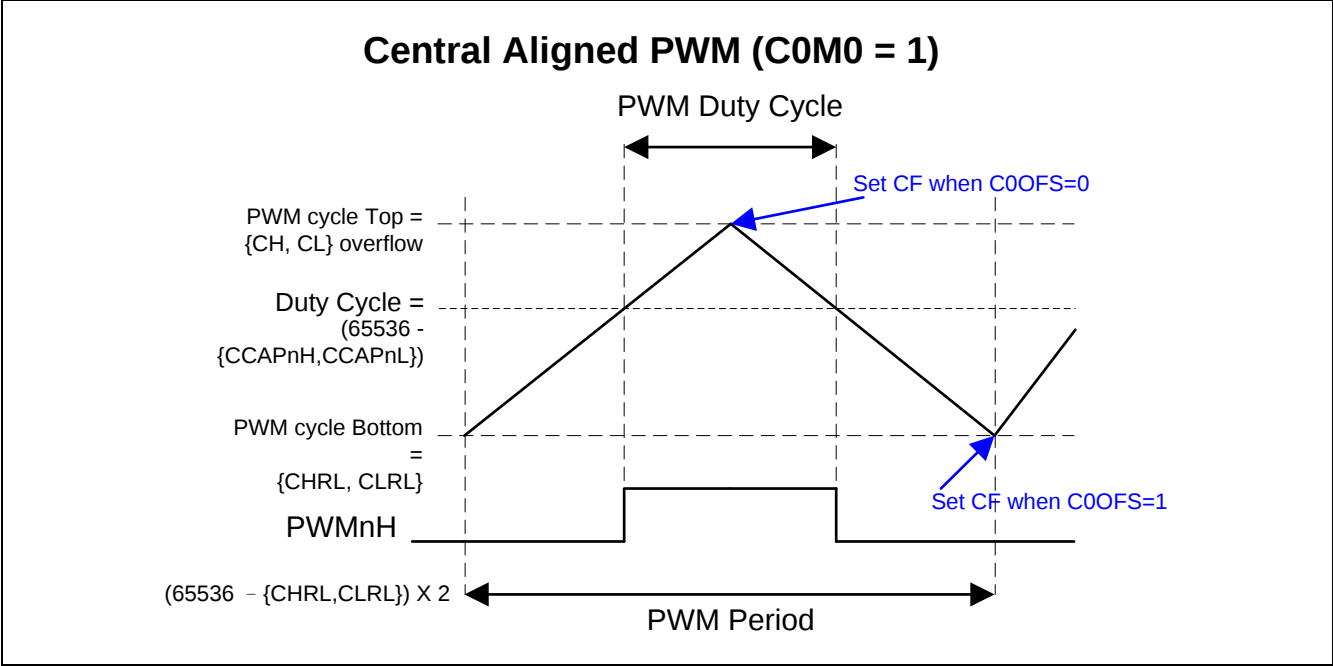
	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
C0OPSn	CEX5	CEX4	CEX3	CEX2	CEX1	CEX0
0	P0.7	P2.6	P0.5	P2.4	P2.3	P2.2
1	P3.5	P3.2	P3.4	P4.1	P3.3	P4.0

17.4.12. 可变分辨率的 PWM 中心对齐

在章节“PTM0 增强型 PWM 控制”中，定义 PWM 中心对齐仅支持 8/10/12/16 位分辨率。并且在那个模式，所有 PCA 功能，如捕捉或比较在非 PWM 模块中是有效的。

请注意，当使用中心对齐 PWM 时，我们建议将 PWM 模块设置为 16 位模式，基础定时器需要使用 16 位 0xFFFF 减去该值，以防止出现意外错误。

图 17-20. 可变分辨率的 PWM 中心对齐



PWMCR: PWM 控制寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xBC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PCAE	C0OFS	PBKM	PBKE1.1	PBKE1.0	PBKE0.2	PBKE0.1	PBKE0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 6: C0OFS, PTM0 溢出标志选择。
0: PWM 中心对齐周期的顶部 CF 置位。
1: PWM 中心对齐周期的低部 CF 置位。

17.4.13. PWM 全局控制

PTM0、PCA1 和定时器 2~3 可采用全局控制方式同步频率和相位。请参考“17.4.13 PWM 全局控制”。

18. 可编程计数器阵列(PCA1)

MG82F6P32 带有一个可编程计数器阵列(PCA1)，该功能与标准定时/计数器相比以更少的 CPU 占用提供了更多的定时能力。它的优点包括减少了软件复杂度并提高了精度。

18.1. PCA1 概述

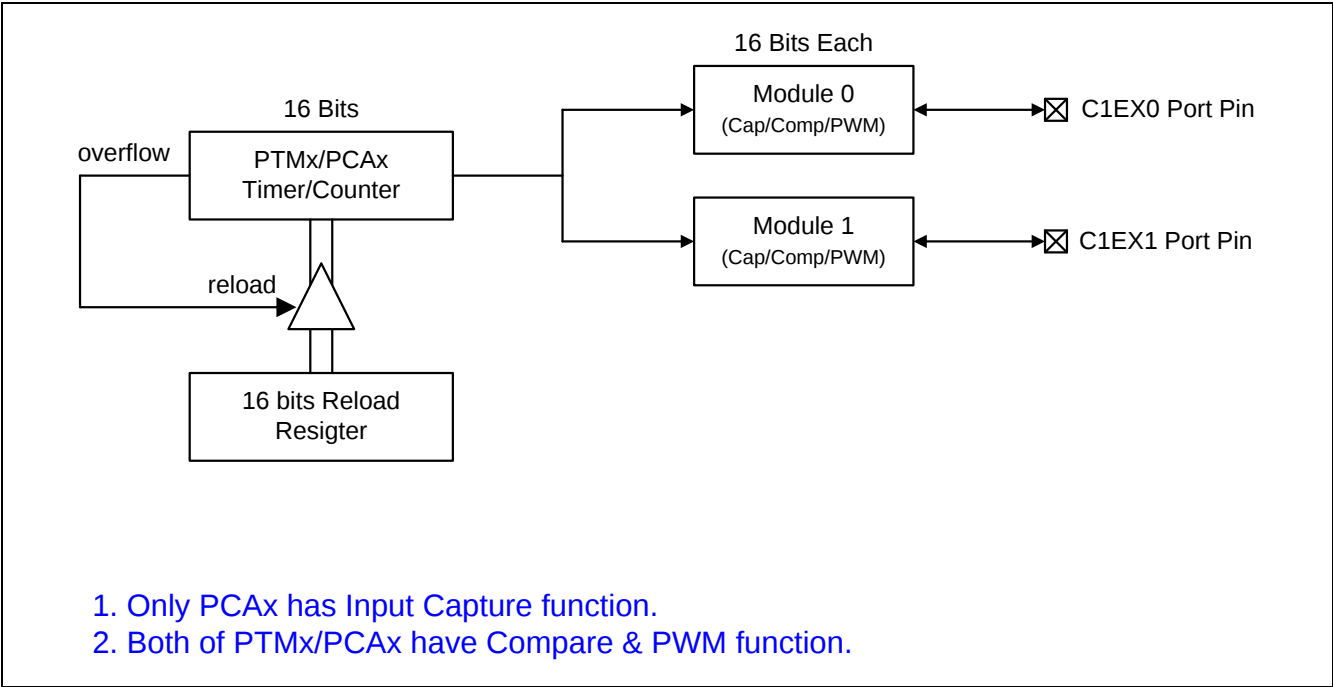
PCA1 由一个专用定时/计数器作为一个 2 组比较/捕获/PWM 模块的时间基准。PCA1 的功能方框图如图 18-1 所示。需要注意的是 PCA1 定时器和模块都是 16 位的。如果一个外部事件同一个模块关联，那么该功能就和相应的端口引脚共享。若某模块没有使用端口引脚，这个引脚还可以用作标准 I/O。

PCA1 模块 0~1 可以编程为如下任意模式：

- 上升和/或下降沿捕获
- 软件定时器(比较)
- 高速输出(比较输出)
- 脉宽调制(PWM)输出
- 脉宽调制匹配的比较输出(COPM)

PCA1 模块 0~1 支持上面列出的模式外增加了捕获功能。所有这些模式将在后面的章节进行详细讨论。这里，让我们先看看如何设置 PCA1 定时器和模块。

图 18-1. PCA1 方框图



18.2. PCA1 定时器/计数器

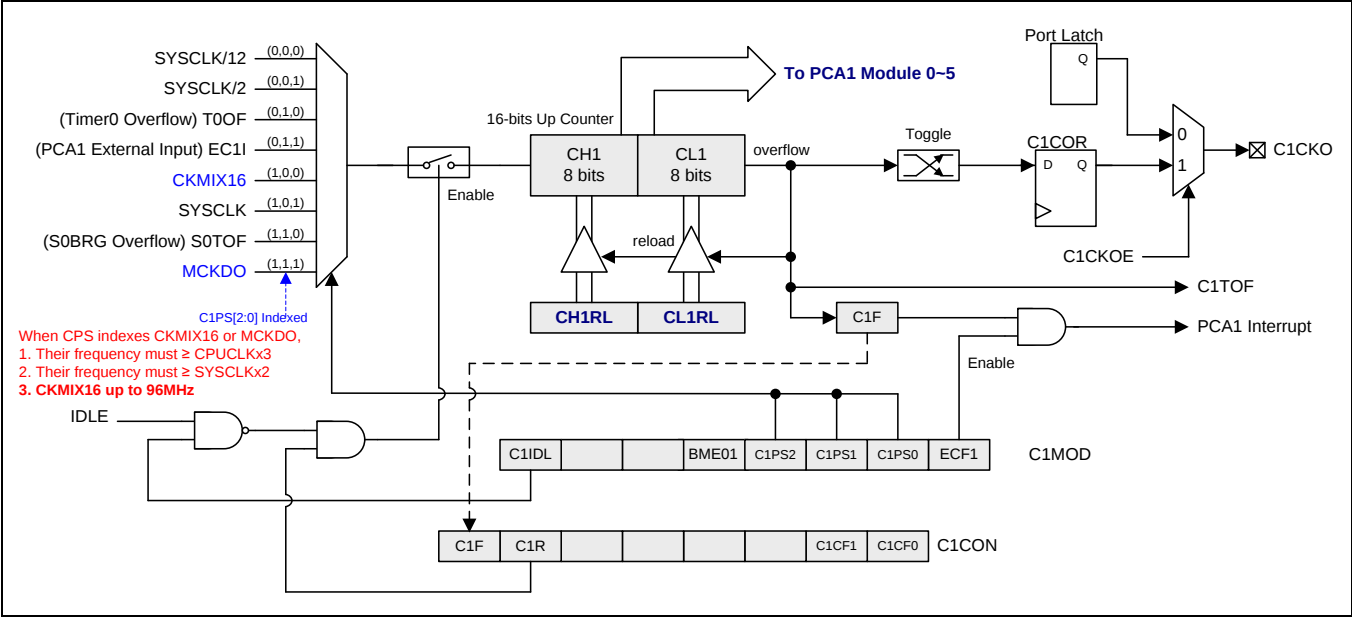
PCA1 的定时器/计数器由一个可以自动重载的 16 位定时器由寄存器 CH 和 CL(计数值的高低字节), CHRL 和 CLRL(重载寄存器的高低字节)组成, 如图 18-2 所示。{CH + CL}计数器每一次溢出 CHRL 和 CLRL 被重载入 CH 和 CL, 这样可以改变 PCA1 周期时间提供可变的 PWM 周期, 比如 7 位或 9 位的 PWM。

{CH + CL}是所有模块的共有时间基准, 它的时钟输入可以从以下来源选择:

- 1/12 系统时钟频率,
- 1/2 系统时钟频率,
- 定时器 0 溢出, 可以让低频时钟源输入到 PCA 定时器,
- 外部时钟输入, ECI 引脚的负跳变,
- CKMIX16, 参考章节 “9.1 时钟结构”,
- 直接从系统时钟,
- S0BRG 溢出, S0TOF,
- MCKDO, 参考章节 “9.1 时钟结构”。

特殊功能寄存器 C1MOD 包含了计数脉冲选择位(C1PS2、CP1S1 和 C1PS0)来指定 PCA1 定时器时钟源。当 C1PS[2:0]选择 CKMIX16 或 MCKDO, 输入时钟源的频率必须 $\geq$ CPUCLK x3 或 $\geq$ SYSCLK x2。这个寄存器也包括了 ECF 位来使能计数器{CH1+CL1}溢出中断。并且计数器溢出切换 C1COR, 当 C1CKOE 使能将输出到端口引脚。此外, 用户可以在空闲模式下设置计数器待机位(C1IDL), 来关闭 PCA1 定时器。这样可以进一步降低空闲模式下的功耗。

图 18-2. PCA1 定时器/计数器



C1MOD: PCA1 计数器模式寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xD9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
C1IDL	0	0	BME01	CPS21	CPS11	CPS01	ECF1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: C1IDL, PCA1 计数器空闲模式控制。

0: PCA1 计数器在空闲模式下继续运行。

1: 空闲模式下关闭 PCA1 计数器。

Bit 4: BME0, PCA1 模块 0/1 缓冲模式使能。仅在捕捉模式, PWM 模式或 COPM 模式下的 PCA1 模块 0 和 1 有效。

0: PCA1 模块 0/1 禁止缓冲模式。

1: PCA1 模块 0/1 使能缓冲模式。

Bit 3~1: C1PS2-C1PS0, PCA 计数器时钟源选择位

CPS21	CPS11	CPS01	PCA1 时钟源
0	0	0	内部时钟, (system clock)/12
0	0	1	内部时钟, (system clock)/2
0	1	0	定时器 0 溢出
0	1	1	来自 ECI 引脚的外部时钟
1	0	0	CKMIX16 输出
1	0	1	内部时钟, (system clock)/1
1	1	0	S0BRT 溢出
1	1	1	MCK 分频器输出, MCKDO

注意: 当 CPS 选择 CKMIX16 或 MCKDO, 需要下面条件:

1. 时钟源的频率必须 $\geq$ CPUCLK x3。
2. 时钟源的频率必须 $\geq$ SYSCCLK x2。

Bit 0: ECF1, 使能 PCA1 计数器溢出中断。

0: 当 C1F 位(C1CON 寄存器中)置位时禁止中断。

1: 当 C1F 位(C1CON 寄存器中)置位时使能中断。

如下所示的 C1CON 寄存器包含 PCA1 运行控制位和 PCA1 定时器与每个模块的标志。要运行 PCA1, C1R 位(C1CON.6)必须软件置位。要关闭 PCA1, 可以清除该位。PCA1 计数器溢出时, 1CF(C1CON.7)置位, 并且若 C1MOD 寄存器的 ECF 置位, 还会产生一个中断。C1F 位只能软件清零。C1CF0 到 C1CF15 是模块 0 到模块 15 的各自中断标志位, 当发生一个匹配或捕获事件时, 硬件置位, 这些位必须软件清零。PCA1 中断系统如图 18-3 所示。

C1CON: PCA1 计数器控制寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xD8

复位值 = 0000-0000

7	6	5	4	3	2	1	0
C1F	C1R	0	0	0	0	C1CF1	C1CF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: C1F, PCA1 计数器溢出标志

0: 只能软件清零。

1: 计数器溢出时硬件置位。C1F 标志在 C1MOD 寄存器的 ECF1 置位时会产生一个中断。C1F 可以硬件或软件置位。

Bit 6: C1R, PCA1 计数器运行控制位。

0: 软件清零关闭 PCA1 计数器。

1: 软件置位开启 PCA1 计数器。

Bit 5~2: 保留。写该寄存器时, 此位必须写“0”。

Bit 1: C1CF1, PCA1 模块 1 中断标志。

0: 只能软件清零。

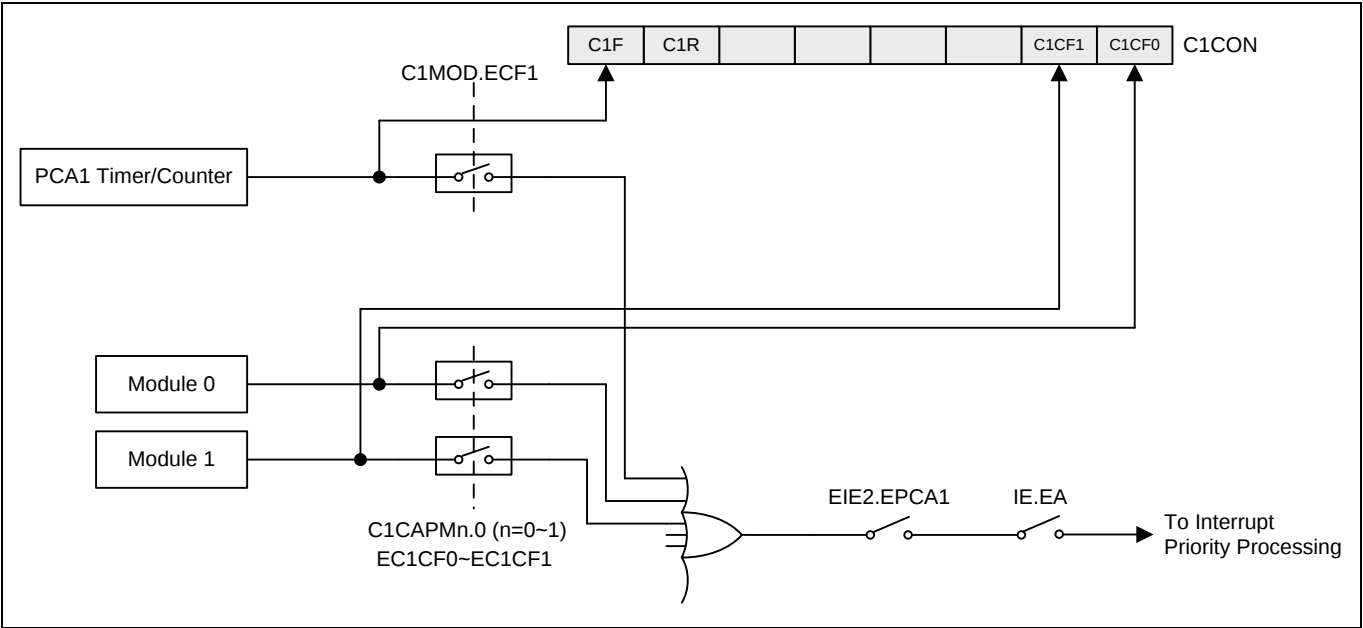
1: 当发生一个匹配或捕获事件时, 硬件置位。

Bit 0: C1CF0, PCA1 模块 0 中断标志。

0: 只能软件清零。

1: 当发生一个匹配或捕获事件时, 硬件置位。

图 18-3. PCA1 中断系统



C1PWMn: PWM 模式辅助寄存器, n=0~1

SFR 页 = 仅 1 页

SFR 地址 = 0xF2

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PnRS11	PnRS01	--	--	--	PnINV1	ECAPnH1	ECAPnL1
R/W	R/W	W	W	R/W	R/W	R/W	R/W

Bit 7~6: PnRS1~0, PWMn 分辨率设置位 1~0。

00: 8 位 PWMn, 当[CH, CL]计数从 XXXX-XXXX-1111-1111 到 XXXX-XXXX-0000-0000 时溢出激活。

01: 10 位 PWMn, 当[CH, CL]计数从 XXXX-XX11-1111-1111 到 XXXX-XX00-0000-0000 时溢出激活。

10: 12 位 PWMn, 当[CH, CL]计数从 XXXX-1111-1111-111 到 XXXX-0000-0000-0000 时溢出激活。

11: 16 位 PWMn, 当[CH, CL]计数从 1111-1111-1111-1111 到 0000-0000-0000-0000 时溢出激活。

Bit 5~3: 保留。写该寄存器时, 此位必须写“0”。

Bit 2: PnINV1, 比较/PWM 输出(C1PnOR)在 C1EXn 引脚上反转。

0: 比较/PWM 输出(C1PnOR)不反转。

1: 比较/PWM 输出(C1PnOR)反转。

Bit 1: ECAPnH1, 扩展第 9 位(MSB), 联合 C1CAPnH 形成 9 位寄存器用于 PWM 模式。

Bit 0: ECAPnL1, 扩展第 9 位(MSB), 联合 C1CAPnL 形成 9 位寄存器用于 PWM 模式。

CH1: PCA1 基准时钟高字节

SFR 页 = 仅 1 页

SFR 地址 = 0xF9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CH1.7	CH1.6	CH1.5	CH1.4	CH1.3	CH1.2	CH1.1	CH1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CL1: PCA1 基准时钟低字节

SFR 页 = 仅 1 页

SFR 地址 = 0xE9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CL1.7	CL1.6	CL1.5	CL1.4	CL1.3	CL1.2	CL1.1	CL1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CH1RL: PCA1 CH 重载寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xCF

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CH1RL.7	CH1RL.6	CH1RL.5	CH1RL.4	CH1RL.3	CH1RL.2	CH1RL.1	CH1RL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: CH1RL, CH1 的重载值

CL1RL: PCA1 CL 重载寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xCE

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CL1RL.7	CL1RL.6	CL1RL.5	CL1RL.4	CL1RL.3	CL1RL.2	CL1RL.1	CL1RL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: CL1RL, CL1 的重载值

18.3. 比较/捕捉模块

比较/捕获模块 0~7 中的每一组都有一个模式寄存器，叫做 C1CAPMn(n=0、1)，用来选择其工作模式。EC1CFn 位控制当模块中断标志置位时每个模块的中断使能。

C1CAPMn: PCA1 模块比较/捕捉寄存器, n=0~1

SFR 页 = 仅 1 页

SFR 地址 = 0xDA ~ 0xDB

复位值 = 0000-0000

7	6	5	4	3	2	1	0
DT1En	ECOMn1	CAP1Pn	CAP1Nn	MATn1	TOGn1	PWMn1	EC1CFn
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: DTE1n, 使能 PWMHn/PWMLn 输出对的死区时间控制。此位仅在 n= 0 有效且当 PWM 信道操作在缓冲模式死区时间功能激活。信道的缓冲模式由 C1MOD 的 BME01 使能

0: PWMn 输出禁止死区时间控制。

1: PWMn 输出使能死区时间控制。

Bit 6: ECOMn1, 比较器使能。

0: 禁止数字比较器功能。

1: 使能数字比较器功能。

Bit 5: CAP1Pn, 正跳变捕捉使能。

0: 禁止 PCA1 捕捉功能在 C1EXn 引脚上正跳变侦测。

1: 使能 PCA1 捕捉功能在 C1EXn 引脚上正跳变侦测。

Bit 4: CAP1Nn, 负跳变捕捉使能。模块 6 和 7 不支持捕捉模式。

0: 禁止 PCA1 捕捉功能在 C1EXn 引脚上负跳变侦测。

1: 使能 PCA1 捕捉功能在 C1EXn 引脚上负跳变侦测。

Bit 3: MATn1, 匹配控制。

0: 禁止数字比较器匹配事件去置位 C1CFn。

1: PCA1 计数器同相应模块的比较/捕获寄存器匹配时 C1CON 寄存器的 C1CFn 置位。

Bit 2: TOGn1, 切换控制。

0: 禁止数字比较器匹配事件去切换 C1EXn。

1: PCA1 计数器同相应模块的比较/捕获寄存器匹配时使 C1EXn 引脚切换。

Bit 1: PWMn1, PWM 控制。

0: 禁止 PCA1 模块中的 PWM 模式。

1: 使能 PWM 功能并使 C1EXn 引脚用作脉宽调制输出。

Bit 0: EC1CFn, 使能 C1CFn 中断。

0: 禁止 C1CON 寄存器中的比较/捕获标志位 C1CFn 产生中断。

1: 使能 C1CON 寄存器中的比较/捕获标志位 C1CFn 产生中断。

注意: CAP1Nn (1CCAPMn.4) 位和 CAP1Pn (C1CAPMn.5) 位决定了捕捉输入的信号脉冲沿, 若这两位同时设置, 则正负跳变都会发生捕获。

每个模块都有一对 8 位比较/捕获寄存器(C1CAPnH, C1CAPnL)。这些寄存器用来存储一个捕捉事件发生的时间或者一个比较时间产生的时间。当模块用于 PWM 模式时, 除这两个寄存器之外, 一个扩展寄存器 C1PWMn 用来扩展输出占空比的范围, 扩展的范围从 0%到 100%, 步距是 1/256。关于 10/12/16 位 PWM 详情请参考章节 17.4.8 和 17.4.9。

C1CAPnH: PCA1 模块 n 捕捉高寄存器高字节, n=0~1

SFR 页 = 仅 1 页

SFR 地址 = 0xFA~0xFB

复位值 = 0000-0000

7	6	5	4	3	2	1	0
C1CAPnH.7	C1CAPnH.6	C1CAPnH.5	C1CAPnH.4	C1CAPnH.3	C1CAPnH.2	C1CAPnH.1	C1CAPnH.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C1CAPnL: 模块 n 捕捉高寄存器低字节, n=0~1

SFR 页 = 仅 1 页

SFR 地址 = 0xEA~0xEB

复位值 = 0000-0000

7	6	5	4	3	2	1	0
C1CAPnL.7	C1CAPnL.6	C1CAPnL.5	C1CAPnL.4	C1CAPnL.3	C1CAPnL.2	C1CAPnL.1	C1CAPnL.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

C1PWMn: PCA1 PWM 模式辅助寄存器 n=0~1

SFR 页 = 仅 1 页

SFR 地址 = 0xF2~0xF3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PnRS11	PnRS01	--	--	--	PnINV1	ECAPnH1	ECAPnL1
R/W	R/W	W	W	R/W	R/W	R/W	R/W

Bit 7~6: PnRS11~0, PWMn 分别率设置位 1~0。

00: 8 位 PWMn, 当[CH1, CL1]计数从 XXXX-XXXX-1111-1111 到 XXXX-XXXX-0000-0000 时溢出激活。

01: 10 位 PWMn, 当[CH1, CL1]计数从 XXXX-XX11-1111-1111 到 XXXX-XX00-0000-0000 时溢出激活。

10: 12 位 PWMn, 当[CH1, CL1]计数从 XXXX-1111-1111-1111 到 XXXX-0000-0000-0000 时溢出激活。

11: 16 位 PWMn, 当[CH1, CL1]计数从 1111-1111-1111-1111 到 0000-0000-0000-0000 时溢出激活。

Bit 5~4: 保留。写该寄存器时, 此位必须写“0”。

Bit 2: PnINV1, 比较/PWM 输出(C1PnOR)在 C1EXn 引脚上反转。

0: 比较/PWM 输出(C1PnOR)不反转。

1: 比较/PWM 输出(C1PnOR)反转。

Bit 1: ECAPnH1, 扩展第 9 位(MSB), 联合 C1CAPnH 形成 9 位寄存器用于 PWM 模式。

Bit 0: ECAPnL1, 扩展第 9 位(MSB), 联合 C1CAPnL 形成 9 位寄存器用于 PWM 模式。

18.4. PCA1 操作模式

不同 PCA1 功能对应的 C1CAPMn 寄存器设置如表 18-1 所示。

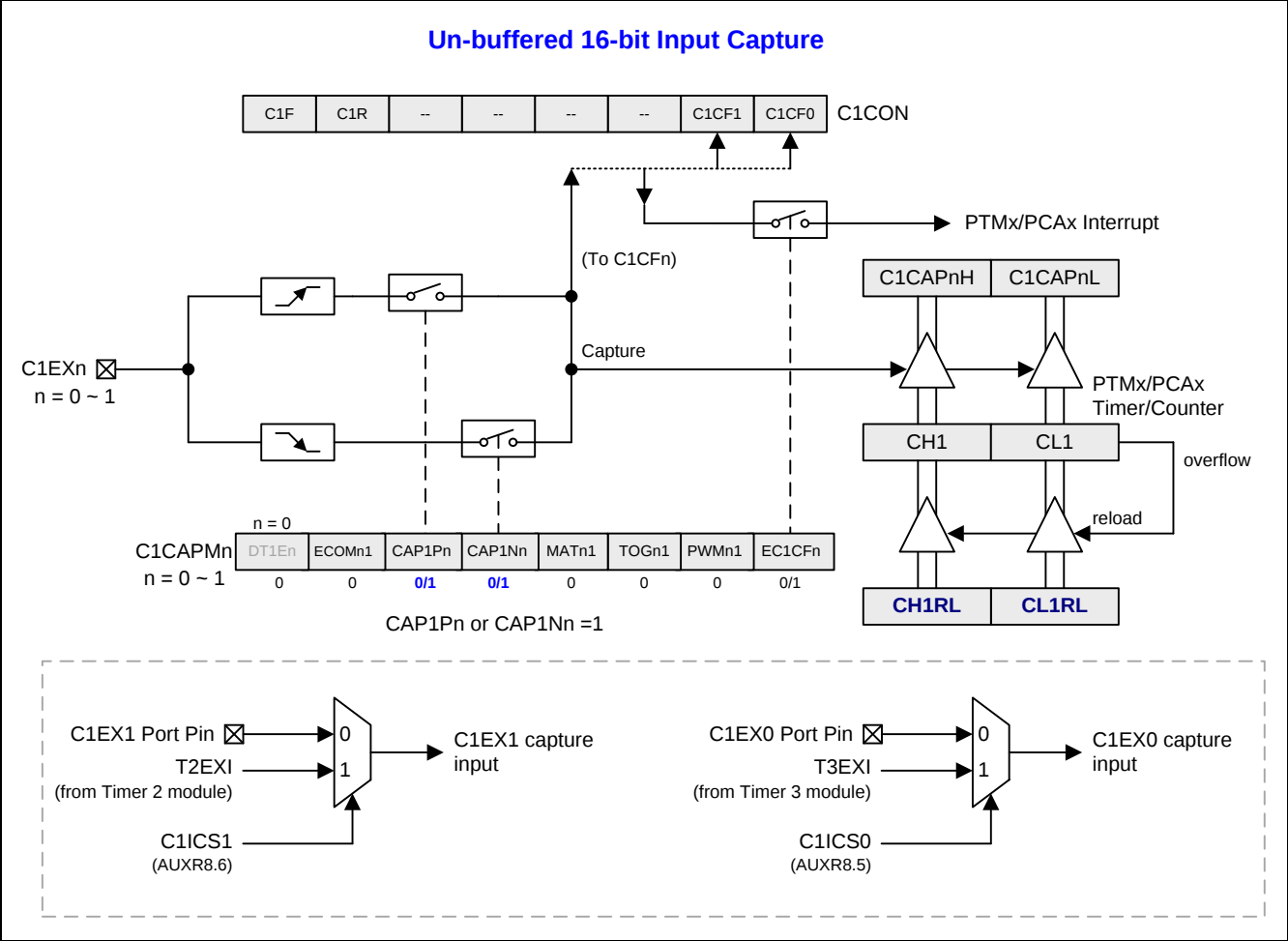
表 18-1. PCA1 模块模式

ECOMn1	CAP1Pn	CAP1Nn	MATn1	TOGn1	PWMn1	BMEEn1	模块功能
0	0	0	0	0	0	0	无操作
X	1	0	0	0	0	0	C1EXn 上上升沿输入捕捉
X	0	1	0	0	0	0	C1EXn 上下降沿输入捕捉
X	1	1	0	0	0	0	C1EXn 上跳变输入捕捉
X	1/0	0/1	0	0	0	1	带缓冲的 C1EXn 上跳变输入捕捉
1	0	0	1	0	0	0	软件定时器(比较)
1	0	0	1	0	0	1	带缓冲的软件定时器(比较)
1	0	0	1	1	0	0	高速输出(输出比较)
1	0	0	1	1	0	1	带缓冲的高速输出(输出比较)
1	0	0	0/1	0	1	0	脉宽调制器(PWM)
1	0	0	0/1	0	1	1	带缓冲的 10/12/16 位 PWM

18.4.1. 捕捉模式

要让某一PCA1模块工作在捕捉模式，CAP1N和CAP1P任何一位或两位必须置位。外部C1EX输入会在每次跳变时采样。当有效跳变发生时，PCA1硬件会将PCA1计数器寄存器值(CH1和CL1)载入到模块的捕捉寄存器(C1CAPnH和C1CAPnL)。若模块的C1CFn和EC1CFn标志同时置位，会产生一个中断。

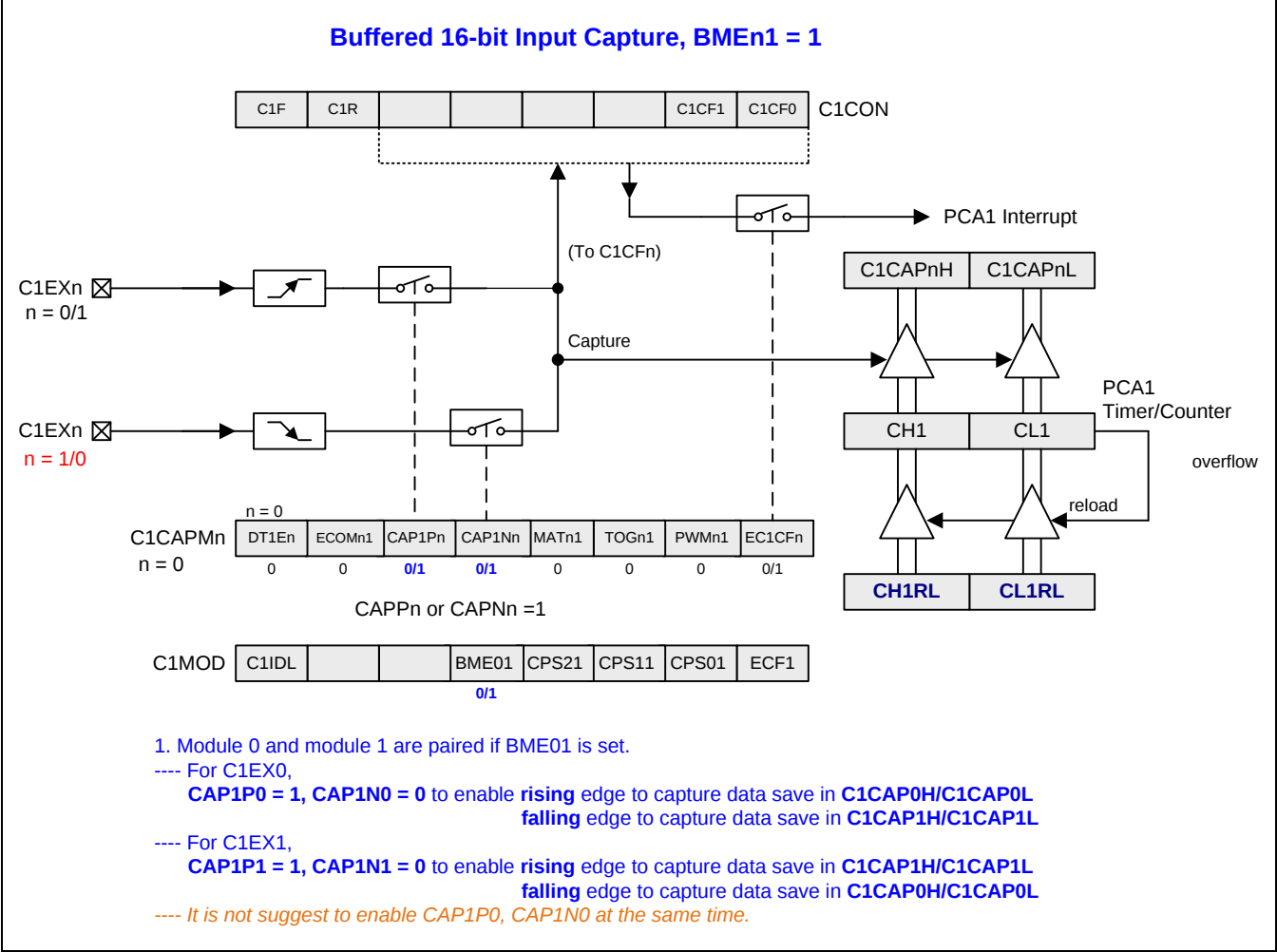
图 18-4. PCA1 捕捉模式



18.4.2. 缓冲捕捉模式

为了捕捉窄的输入信号，缓冲捕捉模式是必要的。如果使能，将把奇数模块捕捉数据寄存器(C1CAPnH, C1CAPnL, n= 1)送到偶数模块捕捉数据寄存器(通道 0)。这不影响模块 0 的捕捉操作。BME0 使能通道 0/1 的缓冲操作。

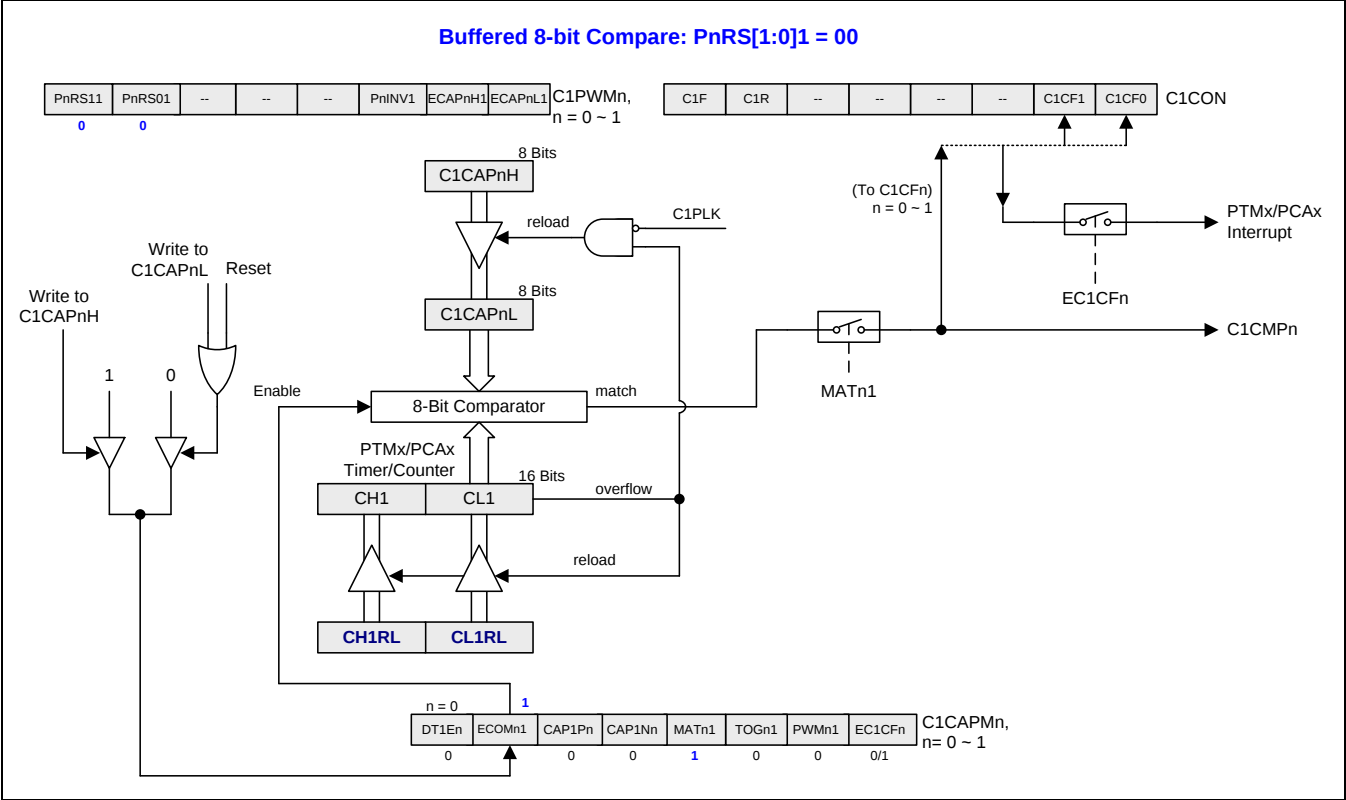
图 18-5. PCA1 缓冲捕捉模式 (BMEn1=1, n= 0)



18.4.3. 8 位软件定时器模式 (比较模式)

PCA1 模块可以通过设置 C1CAPMn 寄存器的 ECOMn1 位和 MATn1 位来作为一个软件定时器使用。PCA1 定时器与模块的捕获寄存器值进行比较，若相等且 C1CFn 和 EC1CFn 位同时设置时会产生一个中断信号。

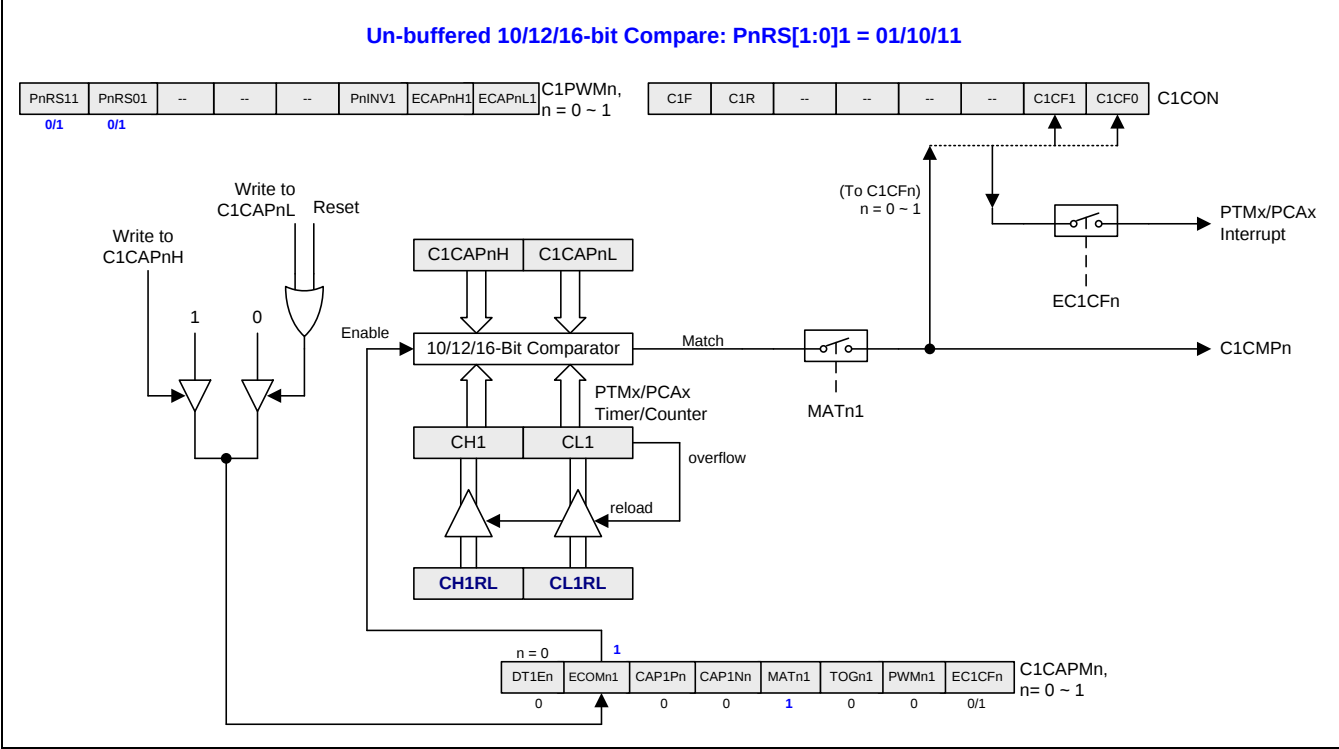
图 18–6. PCA1 软件定时器模式



18.4.4. 16 位软件定时器模式(比较模式)

PCA1 模块可以通过设置 C1CAPMn 寄存器的 ECOMn1 位和 MATn1 位来作为一个软件定时器使用。PCA1 定时器与模块的捕获寄存器值进行比较，若相等且当 C1CFn 和 EC1CFn 位同时设置时会产生一个中断信号。

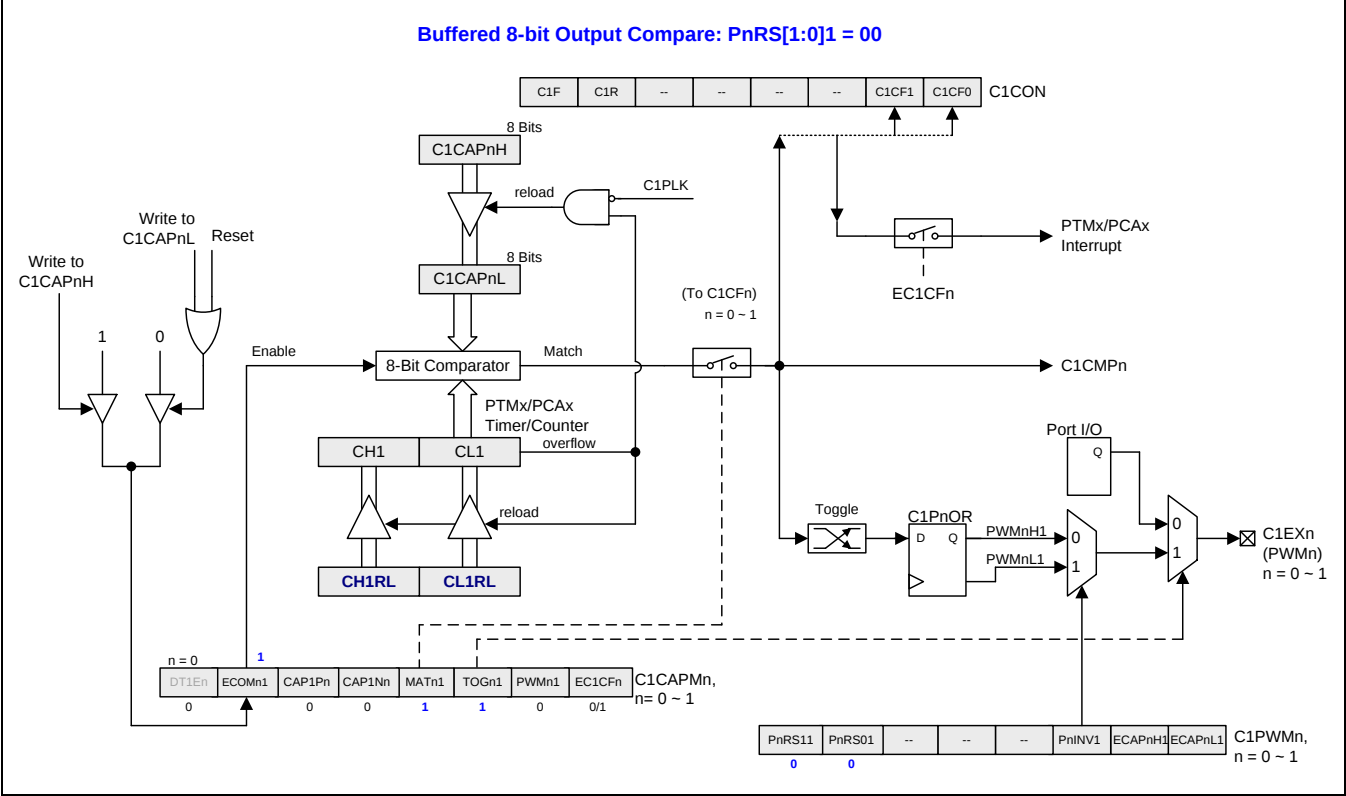
图 18–7. PCA1 软件定时器模式



18.4.6. 8 位高速输出模式(输出比较模式)

这种模式下，每当PCA1计数器与模块捕获寄存器值相等时，与PCA1模块相关联的C1EX输出就会触发。为激活这种模式，C1CAPMn 寄存器的TOGn1、MATn1 和 ECOMn1位必须都置为1。

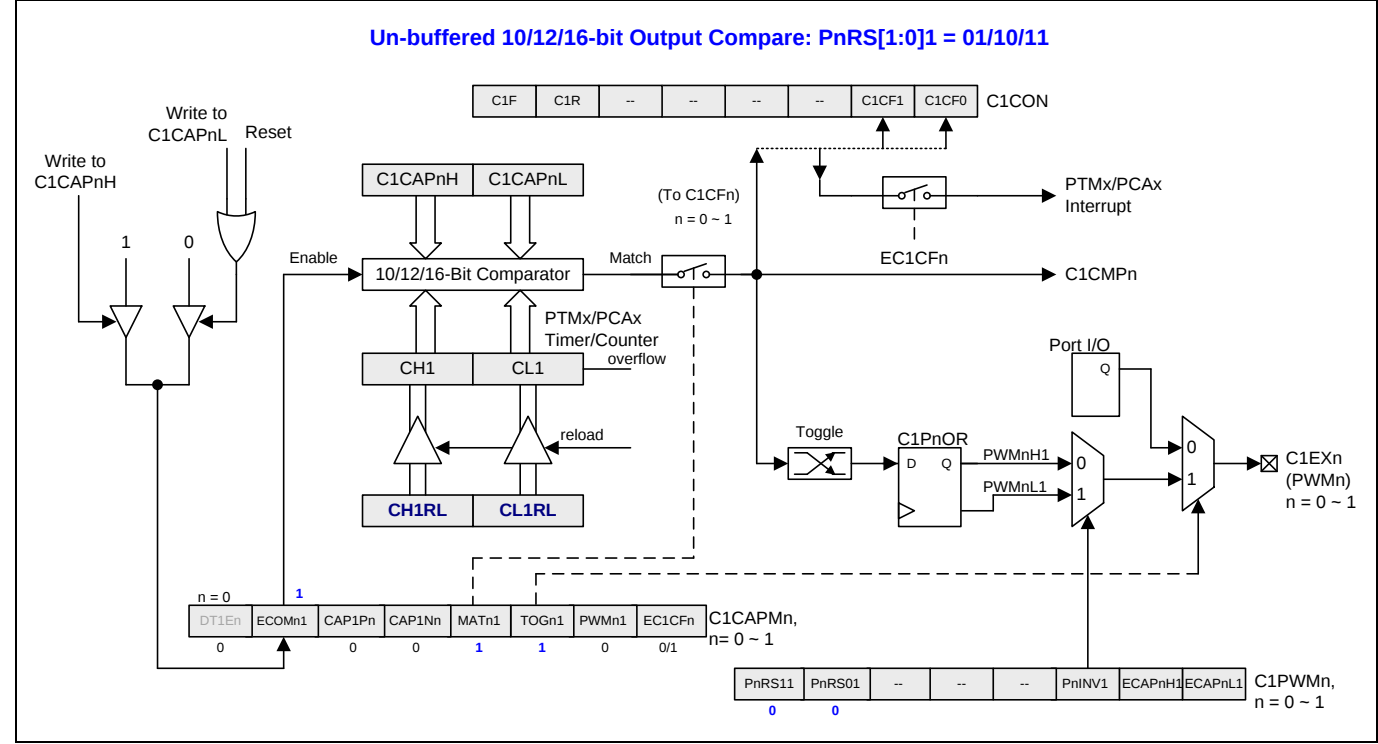
图 18–9. PCA1 高速输出模式



18.4.7. 16 位高速输出模式(输出比较模式)

这种模式下，每当PCA1计数器与模块捕获寄存器值相等时，与PCA1模块相关联的C1EX输出就会触发。为激活这种模式，C1CAPMn 寄存器的TOGn1、MATn1 和 ECOMn1位必须都置为1。

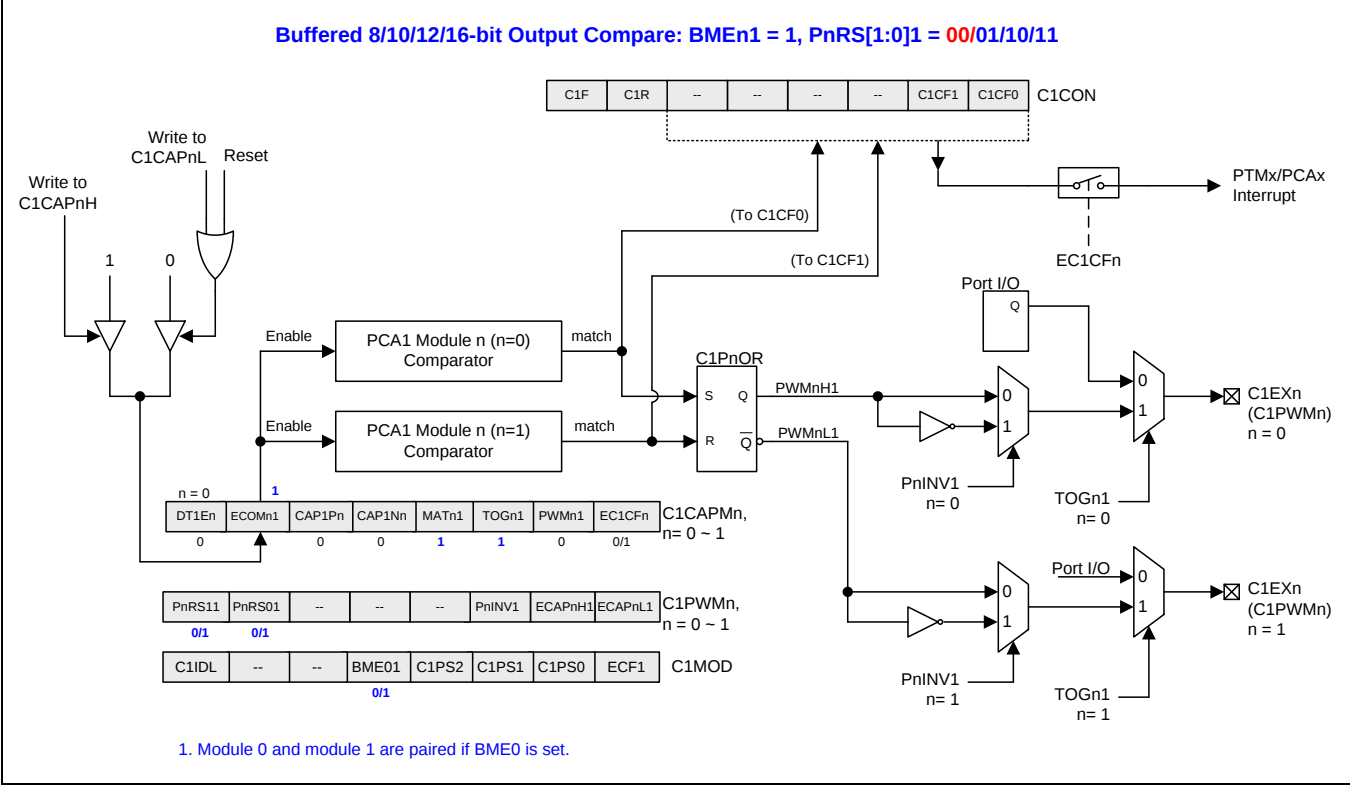
图 18–10. PCA1 高速输出模式



18.4.8. 相位可变比较输出模式

这种模式下，每当 PCA1 计数器与模块捕获寄存器值相等时，与 PCA1 模块相关联的 C1EX 输出就会触发。为激活这种模式，C1CAPMn 寄存器的 TOGn1、MATn1 和 ECOMn1 位必须都置为 1。

图 18–11. PCA1 高速输出模式



18.4.9. 缓冲 8 位 PWM 模式

所有 PCA1 模块都可用作 PWM 输出。输出频率取决于 PCA1 定时器的时钟源。所有的模块都有相同的输出频率，因为它们共享 PCA1 定时器。

占空比取决于模块捕获寄存器 C1CAPnL 与扩展的第 9 位 ECAPnL1 的值。当 9 位数据{0,[C1L]}值小于{ ECAPnL1, [C1CAPnL] }组成的 9 位数据时，输出低电平，相等或大于时输出高电平。

当 C1L 从 0xFF 到 0x00 溢出时，{ ECAPnL1, [C1CAPnL] } 的值使用{ ECAPnH1,[C1CAPnH] }的值重载，这样可以允许无异常的情况下更新 PWM。模块的 C1CAPMn 寄存器 PWMn 和 ECOMn1 位必须置位以使能 PWM 模式。

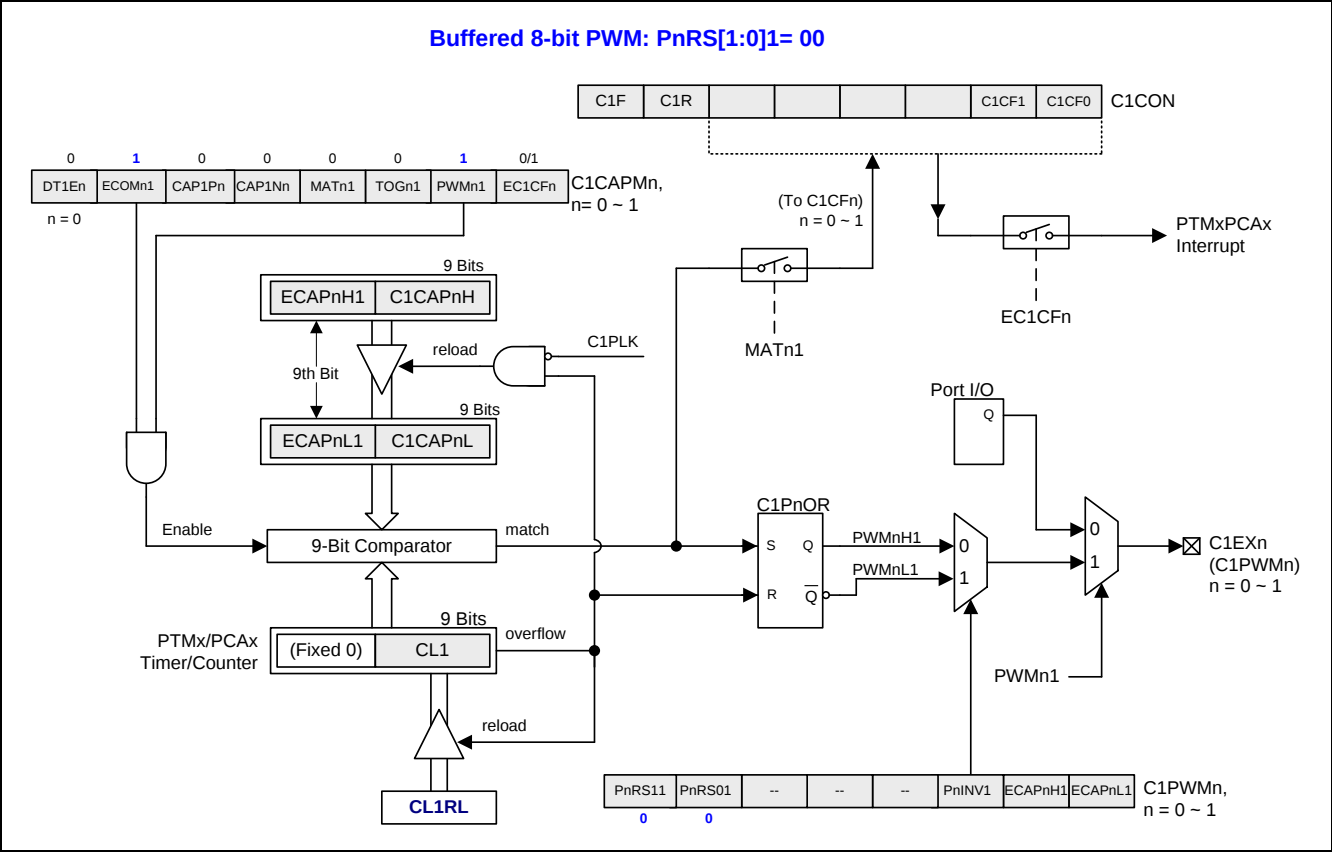
使用 9 位比较，输出的占空比可以真正实现从 0%到 100%可调。占空比计算公式如下：

$$\text{占空比} = 1 - \{ \text{ECAPnH}, [\text{CCAPnH}] \} / 256.$$

这里， [C1CAPnH]是 C1CAPnH 寄存器的 8 位值， ECAPnH1(C1PWMn 寄存器的第 1 位)是 1 位值。所以， { ECAPnH1, [C1CAPnH] } 组成了 9 位比较器用的 9 位值。

- 例如，
- a. 若 ECAPnH=0 且 CCAPnH=0x00 (即 9 位值, 0x000)， 占空比为 100%.
 - b. 若 ECAPnH=0 且 CCAPnH=0x40 (即 9 位值, 0x040)， 占空比为 75%.
 - c. 若 ECAPnH=0 且 CCAPnH=0xC0 (即 9 位值, 0x0C0)， 占空比为 25%.
 - d. 若 ECAPnH=1 且 CCAPnH=0x00 (即 9 位值, 0x100)， 占空比为 0%.

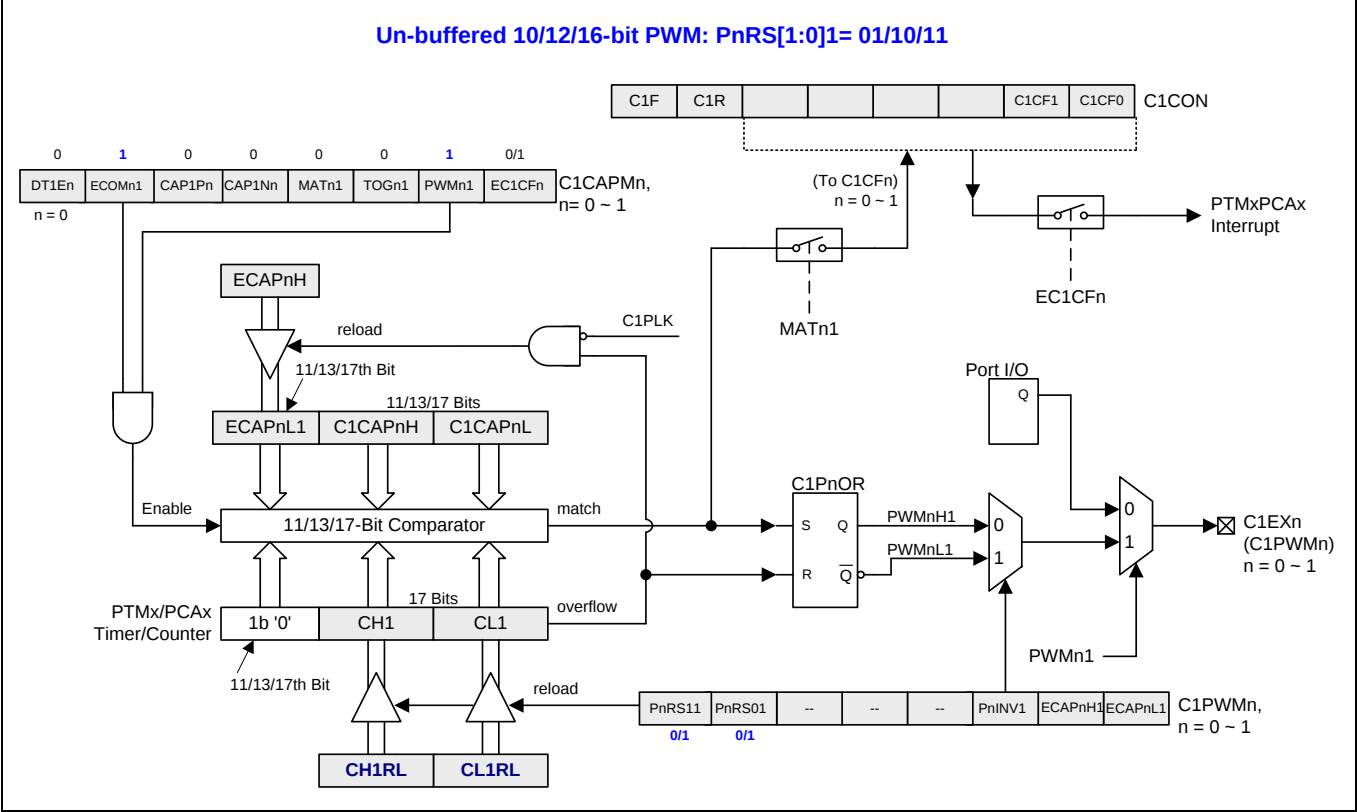
图 18–12. PCA1 缓冲 8 位 PWM 模式



18.4.10. 无缓冲 10/12/16 位 PWM 模式

PCA1 提供可变的 PWM 模式以增强控制能力。有额外无缓冲的 10/12/16 位 PWM 被分配给每一路及每一路 PWM 有不同的分辨率操作能力。

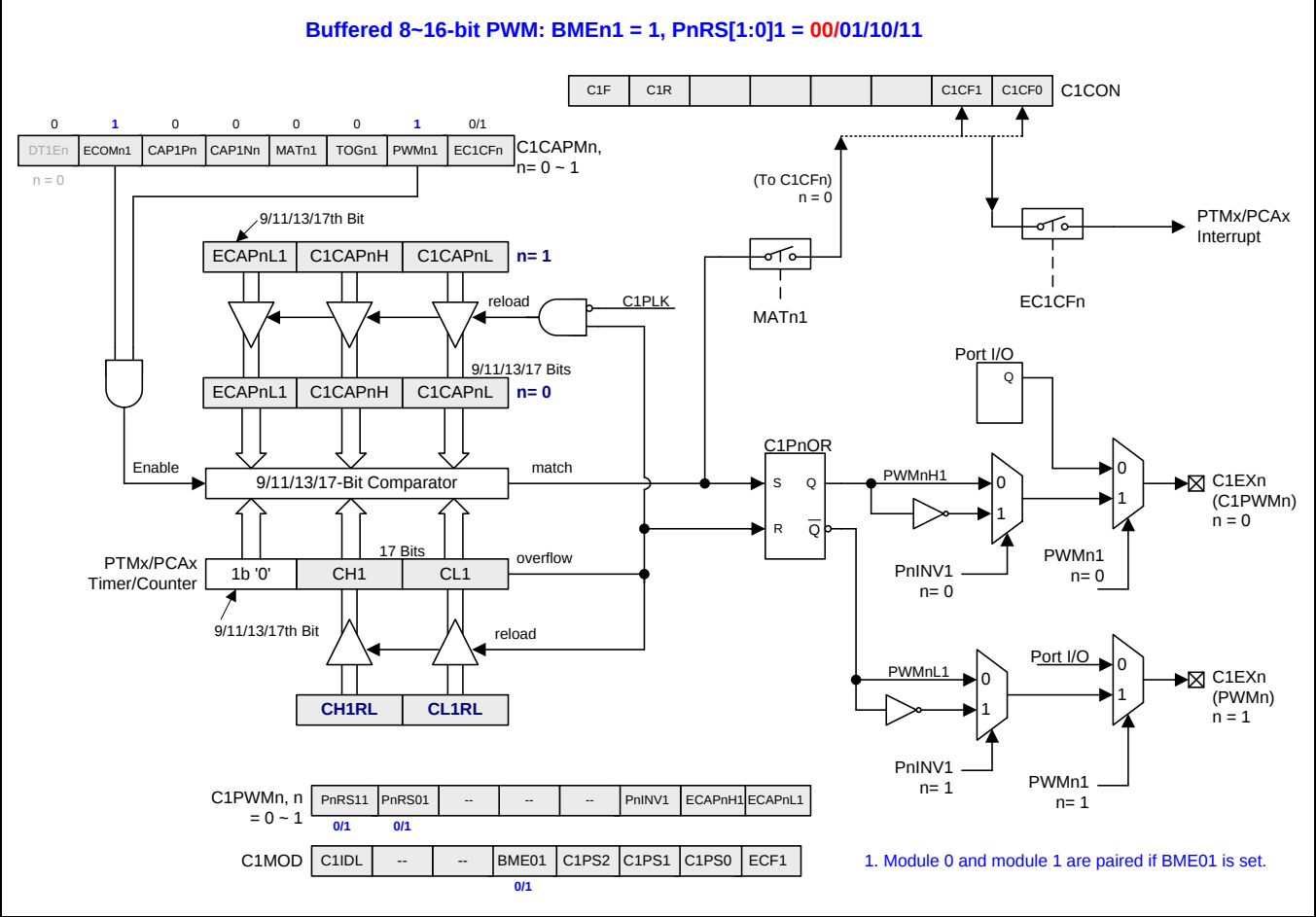
图 18–13. PCA1 无缓冲 10/12/16 位 PWM 模式



18.4.11. 缓冲 10/12/16 位 PWM 模式

如果使用 10/12/16 位 PWM 模式，在将数据写入 C1CAPnH 和 C1CAPnL 时，将会导致意外的占空比，因为 8 位 CPU 每次只能写一个字节。要完成完全设置，需要两个写周期，当第一个字节被写入时，比较器将输出意料之外的占空比。如果应用程序在更改占空比时需要精确控制，则需要使用缓冲 PWM 模式。

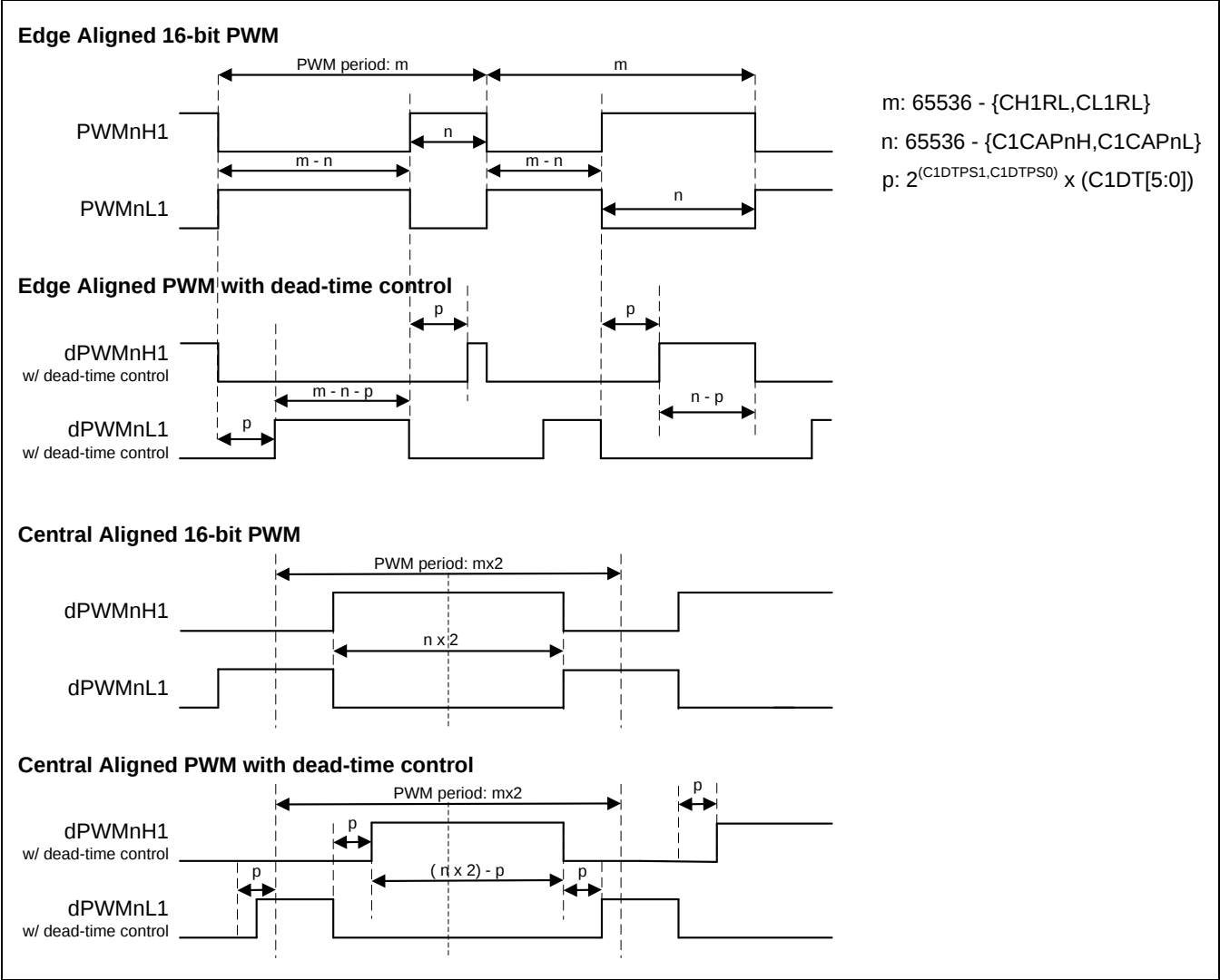
图 18-14. PCA1 缓冲 10/12/16 位 PWM 模式 (带死区控制)



18.4.12. PCA1 增强型 PWM 控制

PCA1 提供可变的 PWM 模式增强 PWM 应用的控制能力。这些额外的 10/12/16 位 PWM 可以分配到每个信道且每个 PWM 信道可以同时运行在不同的分辨率和不同的相位延时。

图 18–15. PCA1 PWM 带死区控制的波形



C1CAPMn: PCA 模块比较捕捉寄存器, n=0~1

SFR 页 = 仅 1 页

SFR 地址 = 0xDA~0xDF

复位值 = 0000-0000

7	6	5	4	3	2	1	0
DT1En	ECOMn1	CAP1Pn	CAP1Nn	MATn1	TOGn1	PWMn1	ECCFn1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: DT1En, 使能 PWMHn/PWMLn 输出对的死区时间控制。此位仅在 n= 0、2 和 4 有效且当 PWM 信道操作在缓冲模式死区时间功能激活。信道的缓冲模式由 C1MOD 的 BME01 使能

0: PWMn 输出禁止死区时间控制。

1: PWMn 输出使能死区时间控制。

C1PDTCCR: PCA1 死区控制寄存器-A

SFR 页 = 仅 B 页

SFR 地址 = 0xBC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
C1DTPS1	C1DTPS0	C1DT.5	C1DT.4	C1DT.3	C1DT.2	C1DT.1	C1DT.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: C1DTPS1~0, 死区计数器的时钟预分频

C1DTPS[1:0]	预分频选择
00	SYSCLK
01	SYSCLK/2
10	SYSCLK/4
11	SYSCLK/8

Bit 5~0: C1DT5~0, 死区时间控制位

DT[5:0]	死区时间
000000	禁止死区时间
000001	预分频器时钟 X 1
000010	预分频器时钟 X 2
000011	预分频器时钟 X 3
.....	
111110	预分频器时钟 X 62
111111	预分频器时钟 X 63

C1PWMCR: PAC1 控制寄存器

SFR 页 = 仅 A 页

SFR 地址 = 0xBC

复位值 = 0000-0000

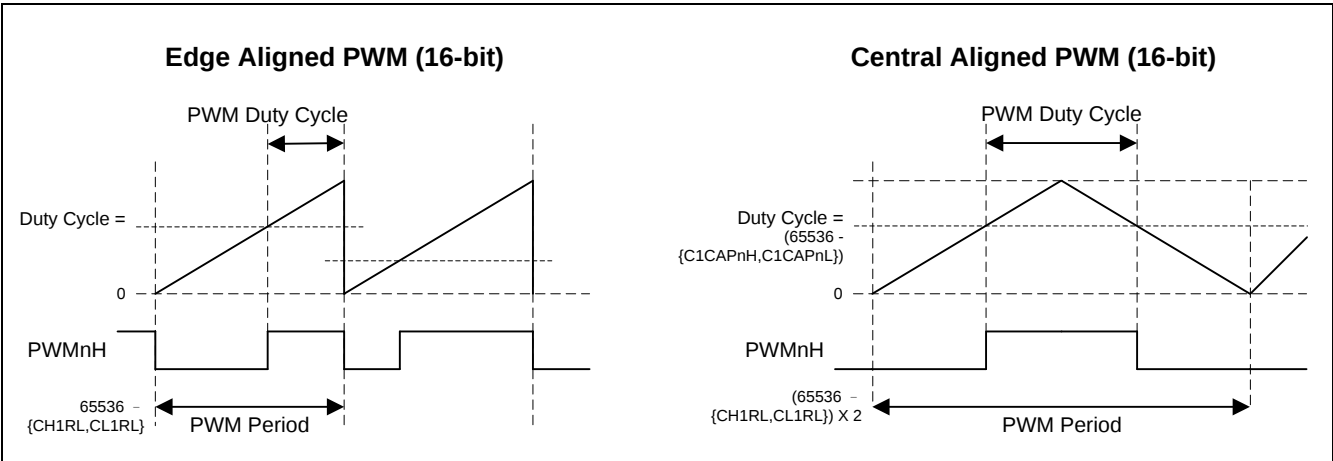
7	6	5	4	3	2	1	0
PCA1E	C1OFS	C1BKM	C1BKE1.1	C1BKE1.0	C1BKE0.2	C1BKE0.1	C1BKE0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: PCA1E, PWM 中心对齐使能。PCA1E 控制使能的 PWM 通道为中心对齐调制包括 PWM 的缓冲模式或非缓冲模式。在这个 PWM 模式下, PWM 频率是半边沿对齐模式。此功能仅在 PWM0~1 激活。

0: 设置 PWM 功能为边沿对齐调制。

1: 使能 PWM 功能为中心对齐调制。建议将所有 PWM 通道设置为相同的分辨率。

图 18-16. 边沿对齐和中心对齐的 PWM 波形



Bit 6: C1OFS, PCA1 溢出标志选择。
0: PWM 中心对齐周期的顶部 CF 置位。
1: PWM 中心对齐周期的低部 CF 置位。

Bit 5: C1BKM, PWM 中止模式选择。
0: 锁存模式。
1: 逐周期模式。

图 18–17. PWM 中止控制的锁存模式波形

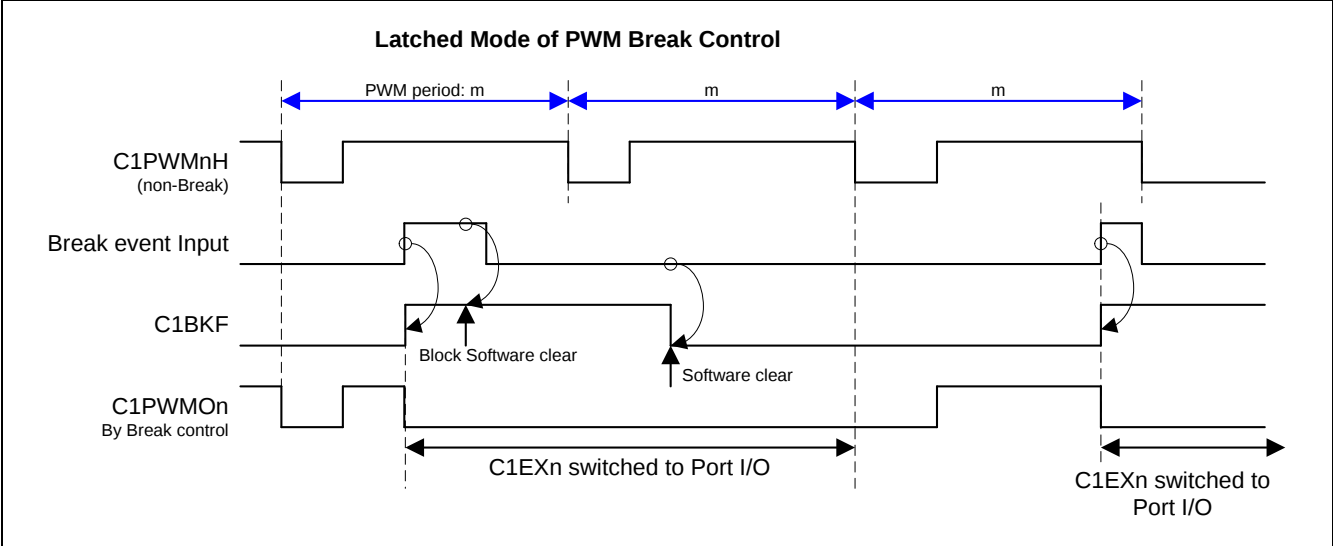
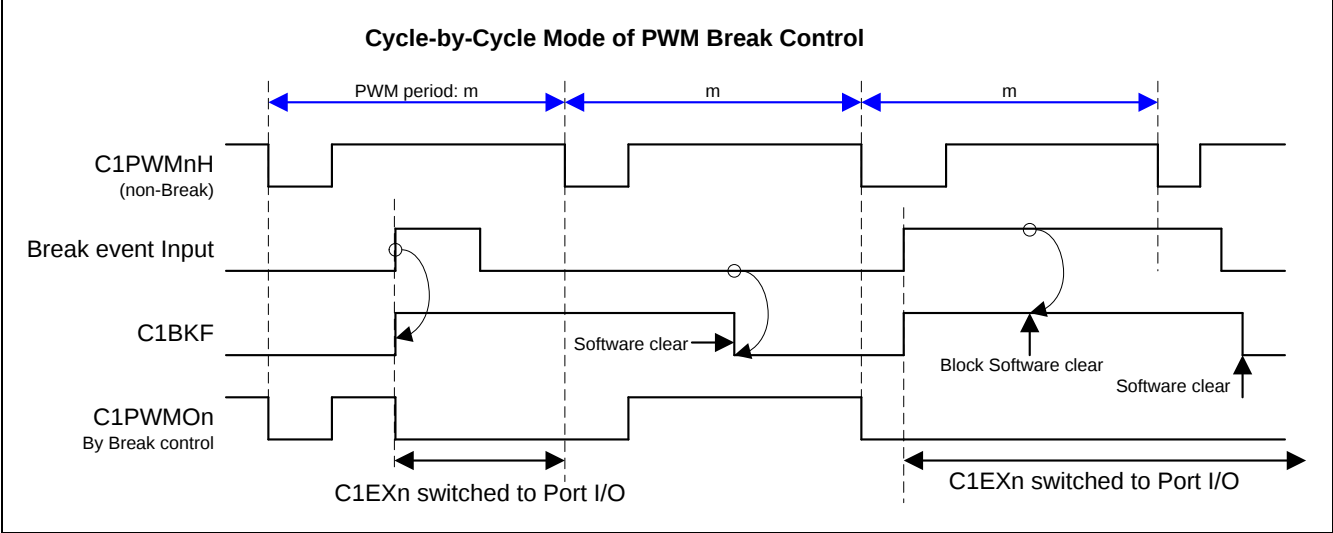


图 18–18. PWM 中止控制的逐周期模式波形



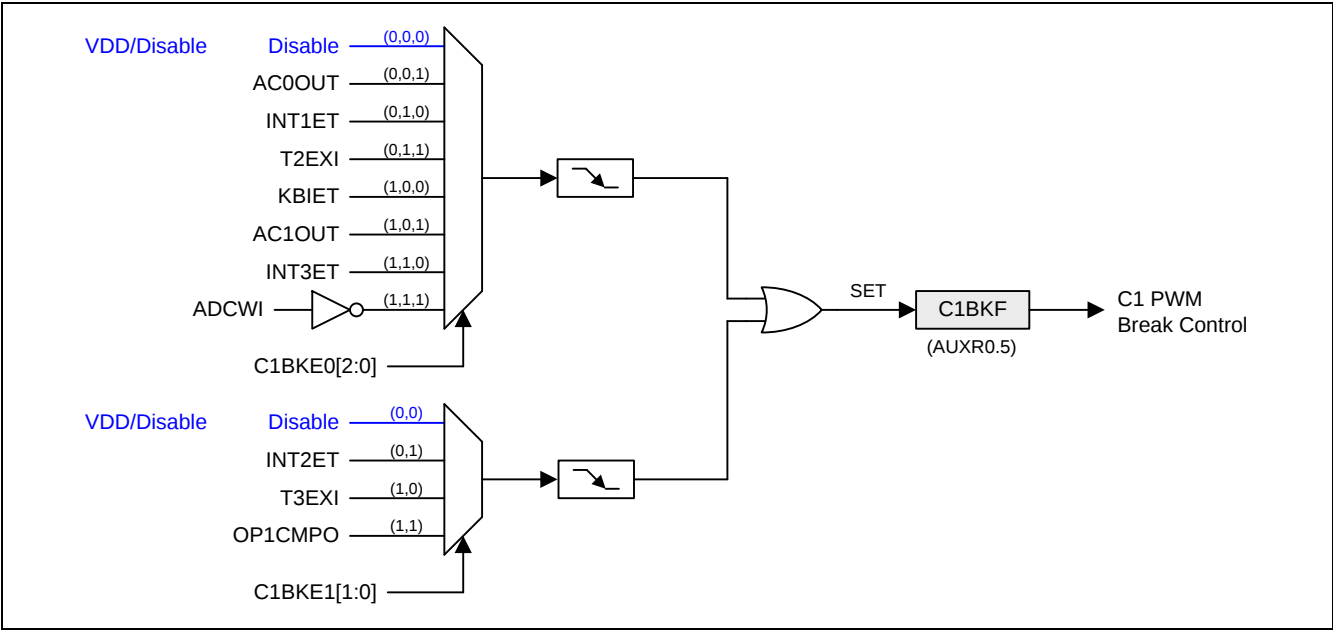
Bit 4~3: C1BKE1.1~0, PWM 中止使能 1 选择。此功能仅在 CEXn 输出模式有效 (n=0~1).

C1BKE1[1:0]	PWM 中止源
0 0	禁止 PWM 中止源 1
0 1	INT2ET, nINT2 激活
1 0	T3EXI
1 1	OP1ES

Bit 2~0: C1BKE0.2~0, PWM 中止使能 0 选择。此功能仅在 CEXn 输出模式有效(n=0~1).

C1BKE0[2:0]	PWM 中止源
0 0 0	禁止 PWM 中止源 0
0 0 1	AC0OUT
0 1 0	INT1ET, nINT1 激活
0 1 1	T2EXI
1 0 0	KBIET, KBI 匹配激活
1 0 1	AC1OUT
1 1 0	INT3ET, nINT3 激活
1 1 1	ADCWI 激活

图 18–19. PCA1 PWM 中止控制源



AUXR0: 辅助寄存器 0

SFR 页 = 0~F

SFR 地址 = 0xA1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C1BKF	PBKF	0	0	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

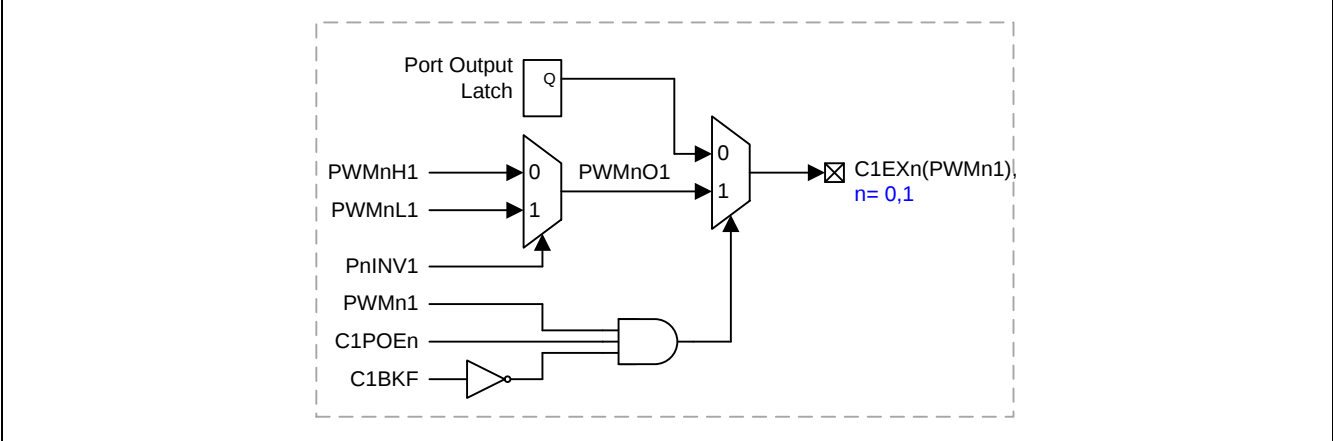
Bit 5: C1BKF, PWM 中止事件标志。此位由 PWM 中止源使能设置。如果此位设置，则使能的 PWM 通道 0~1 将被锁住并且输出引脚保持最初的 GPIO 状态。
0: 没有 PWM 中止事件出现。仅由软件清零。
1: PWM 中止事件出现或软件触发一个 PWM 中止。

18.4.13. PCA1 模块输出控制

PCA 模块具有多种输出控制模式，可以选择用于不同的应用。CEXn (n=1)可编程为一般 I/O 端口或 PCA1 模块 PWM1 的输出。当 PWM 被分配到 C1EXn 时，PnINV1 可以在正常或反向之间切换 PWM 信号。POEn 可以用来启用或禁用端口引脚输出 PWM 信号。

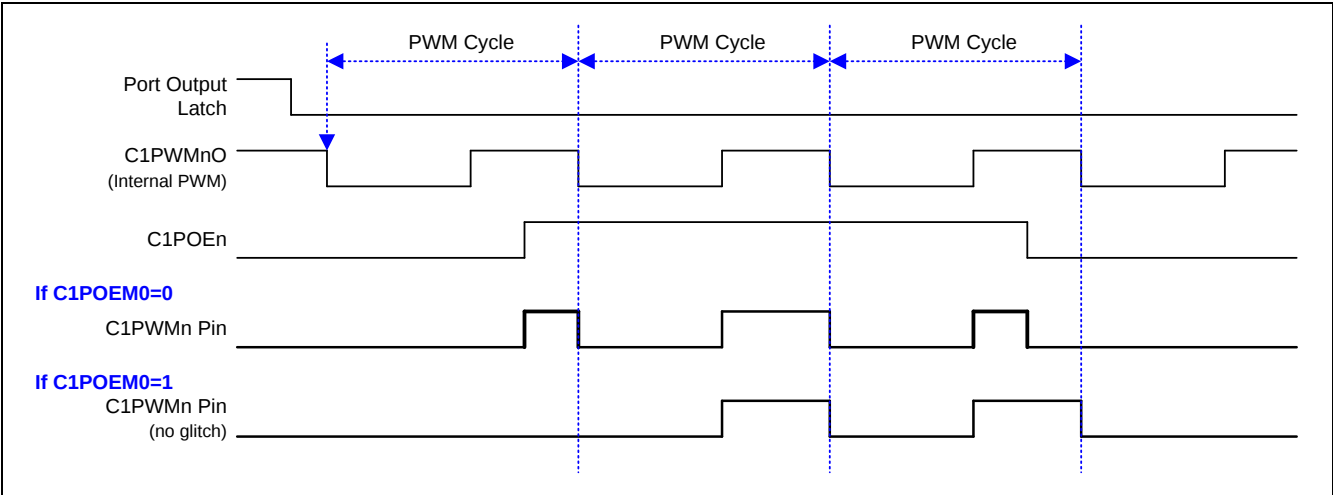
C1EXn (n=1)可以使用 C1BKF，PWM 中止标志，中止 PWM 输出。如果此位设置，则使能的 PWM 通道 0~1 将被锁住并且输出引脚保持最初的 GPIO 状态。

图 18–20. PCA1 模块输出控制



MG82F6P32 的 C1POEM0 控制 C1POEn 输出时序与 PWM 周期对齐。这个对齐功能的配置和波形如图 18–21 所示。

图 18–21. C1POEn 对齐输出控制 (例如.PWM 边沿对齐的波形)



C1AOE0: PWM 额外输出使能寄存器

SFR 页 = 仅 8 页

SFR 地址 = 0xF1

复位值 = XXXX-1001

7	6	5	4	3	2	1	0
0	0	0	0	C1POE1	0	0	C1POE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: 保留。写该寄存器时，此位必须写“0”。

Bit 3: C1POE1, PCA1 PWM1 主通道(C1PWM1O)输出控制。

0: 禁止 C1PWM1O 在端口引脚输出。

1: 使能 C1PWM1O 在端口引脚输出。默认是使能。

Bit 2~1: 保留。写该寄存器时，此位必须写“0”。

Bit 0: C1POE0, PCA 1PWM0 主通道(C1PWM0O)输出控制。

0: 禁止 C1PWM0O 在端口引脚输出。

1: 使能 C1PWM0O 在端口引脚输出。默认是使能。

C1AOE1: PWM 额外输出使能寄存器 1

SFR 页 = A

SFR 地址 = 0xF1

复位值 = 1XXX-XXXX

7	6	5	4	3	2	1	0
C1POEM0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 2: C1POEM0, PCA1 C1POEn 控制 0.

0: C1POEn 功能在 CPU 写入后立即生效

1: C1POEn 功能与 PWM 周期对齐

Bit 6~0: 保留。写该寄存器时，此位必须写“0”。

AUXR7: 辅助寄存器 7

SFR 页 = 仅 4 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0CKOE	C1CKOE	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: C1CKOE, PCA1 时钟输出(C1CKO)使能。

0: 禁止 PCA1 时钟输出。

1: 使能 PCA1 基准定时器溢出率二分频的时钟输出。

AUXR8: 辅助寄存器 8

SFR 页 = 仅 5 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 6: C1ICS1, PCA1 输入通道 1 输入端口引脚选择

C1ICS1	C1EX1 input
0	C1EX1 Port Pin
1	T2EXI

Bit 5: C1ICS0, PCA1 输入通道 0 输入端口引脚选择

C1ICS0	C1EX0 input
0	C1EX0 Port Pin
1	T3EXI

AUXR17: 辅助寄存器 17

SFR 页 = 仅 E 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	0	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1~0: C1OPS1 ~0, PCA1 输出引脚选择

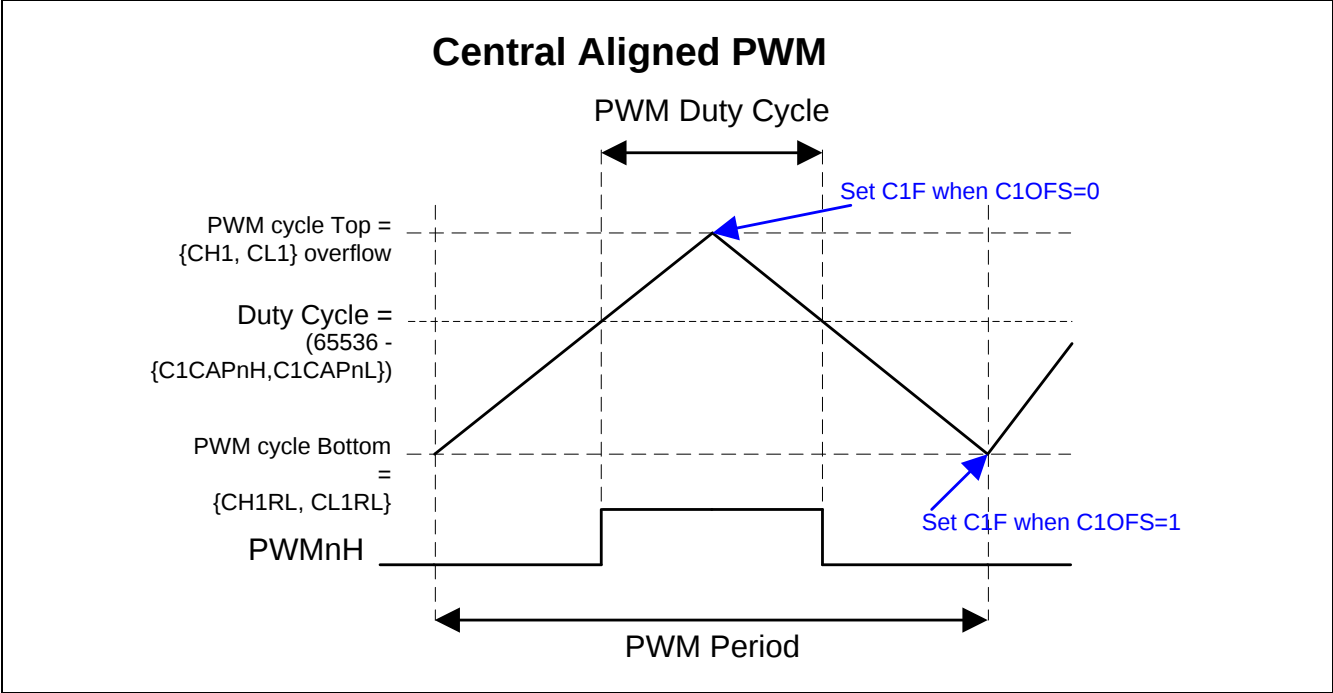
	C1OPS1	C1OPS0
C1OPSn	C1EX1	C1EX0
0	P6.1	P6.0
1	P3.1	P3.0

18.4.14. 可变分辨率的 PWM 中心对齐

在章节“PCA1 增强型 PWM 控制”中，定义 PWM 中心对齐仅支持 8/10/12/16 位分辨率。并且在那个模式，所有 PCA1 功能，如捕捉或比较在非 PWM 模块中是有效的。

请注意，当使用中心对齐 PWM 时，我们建议将 PWM 模块设置为 16 位模式，基础定时器需要使用 16 位 0xFFFF 减去该值，以防止出现意外错误。

图 18-22. 可变分辨率的 PWM 中心对齐



C1PWMCR: PCA1 PWM 控制寄存器

SFR 页 = 仅 A 页

SFR 地址 = 0xBC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PCA1E	C1OFS	C1BKM	C1BKE1.1	C1BKE1.0	C1BKE0.2	C1BKE0.1	C1BKE0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 6: C1OFS, PCA1 溢出标志选择。
0: PWM 中心对齐周期的顶部 CF1 置位。
1: PWM 中心对齐周期的低部 CF1 置位。

18.4.15. PWM 全局控制

PTM0、PCA1 和定时器 2~3 可采用全局控制方式同步频率和相位。请参考“16.4 定时器全局控制”。

19. 串口 0 (UART0)

MG82F6P32 支持全双工的串口，意思是可以同时发送和接收数据。它有一个接收缓冲，意味着在前一个接收到的字节没有从寄存器读出前，就可以开始接收第二个字节。但是，如果第一个字节在第二个字节接收完成前仍然没有被读出，则其中的一个字节将会丢失。串口的接收和发送寄存器都通过特殊寄存器 **S0BUF** 来访问。写到 **S0BUF** 加载到传送寄存器，当从 **S0BUF** 读时是一个物理上独立分离的接收寄存器。

19.1. 串口 0 模式选择

串口可以工作在 5 种标准模式和 8 个增强模式：模式 0 提供同步通讯，同时模式 1、2 和模式 3 提供异步通讯。异步通讯作为一个全双工的通用异步收发器(UART)，可以同时发送和接收，并使用不同的波特率。UART0 的模式 4 支持 SPI 主机工作，速率设置跟模式 0 一样。增强模式请参考章节 “19.11 串口 0”

表 19-1. 串口 0 模式选择

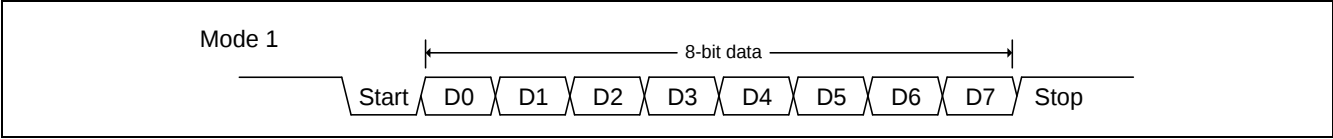
SM30,SM00, SM10	S0RCK	S0TCK	模式	功能	波特率时基	
000	0	0	0	移位寄存器	SYSClk/12 or SYSClk/4	
001	0	0	1	8 位 UART	定时器 1 或定时器 2 溢出	
010	0	0	2	9 位 UART	SYSClk/64, /32, 或 /16	
011	0	0	3	9 位 UART	定时器 1 或定时器 2 溢出	
100	0	0	4	SPI 主机	SYSClk/12 或 SYSClk/4 (SMOD1=1)	
000	0	1	增强	移位寄存器	S0BRG 溢出	
000	1	1	增强	ARGB	S0BRG 溢出	
001	0/1	0/1	增强	8 位 UART	TX 或 RX 可选 S0BRG 溢出	
010	0	1	增强	9 位 UART	TX: S0BRG 溢出 RX: SYSClk/64, /32 or /16	
010	1	0	增强	9 位 UART	TX: SYSClk/64, /32 or /16 RX: S0BRG 溢出	
010	1	1	增强	存定时器	仅定时器	
011	0/1	0/1	增强	9 位 UART	TX 或 RX 可选 S0BRG 溢出	
100	0	1	增强	SPI 主机	S0BRG 溢出	
其它				保留.		

注: Mode0~4 使用 S0RCK 和 S0TCK 的默认值，要重置为增强模式，请参考 “19.11 串口 0” 的详细描述

模式 0: 8 位数据(低位先出)通过 RXD0 传送和接收。TXD0 总是作为输出移位时钟。波特率可通过 PCON0 寄存器的 SMOD1 位选择为系统时钟频率的 1/12 或 1/4。**MG82F6P322** 串口模式 0 的时钟极性也可以软件选择。在串行数据移入或移出之前它由 P3.1 的状态决定。
图 19-4 和图 19-5 所示模式 0 的时钟极性波形。

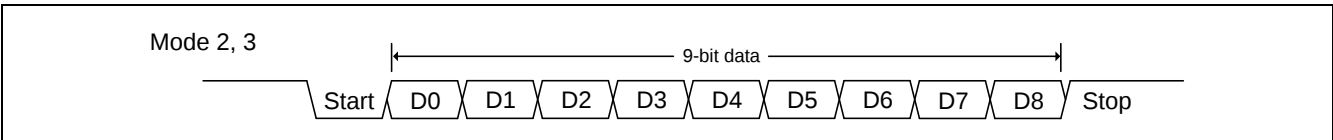
模式 1: 10 位通过 TXD0 传送或通过 RXD0 接收，数据帧包括一个起始位(0)，8 个数据位(低位优先)，和一个停止位(1) (如图 19-1 所示)。在接收时，停止位进入到专用寄存器(S0CON)的 RB80。波特率是可变的。

图 19-1. 模式 1 数据帧



模式 2: 11 位通过 TXD0 传送或通过 RXD0 接收，数据帧包括一个起始位(0)，8 个数据位(低位优先)，一个可编程的第九个数据位和一个停止位(1) (如图 19-2 所示)。在传送时，第 9 个数据位(TB80 在 S0CON 寄存器)可以分配为 0 或者 1。例如，奇偶检验位(P，在 PSW 寄存器)可以移到 TB80 中。在接收时，第九个数据位到 S0CON 寄存器中的 RB80，同时忽略停止位。波特率可以配置为 1/32 或 1/64 的系统时钟频率。

图 19-2. 模式 2, 3 数据帧



模式 3: 除了波特率是可变之外，模式 3 与模式 2 一样。

在四种模式中，使用 S0BUF 作为一个目的寄存器，可以通过任何指令发起传输。在模式 0，当 RI0=0 且 REN0=1 除了标准操作外，UART0 还能具有侦察丢失停止位的帧错误和自动地址识别的功能。

19.2. 串口 0 模式 0

串行数据通过 RXD0 读入和输出。TXD0 输出移位时钟。接收和发送 8 位数据：8 个数据位(低位优先)。波特率可通过 PCON0 寄存器中的 SMOD1 选择为系统时钟的 1/12 或 1/4。

串口模式 0 的简化功能框图如图 19-3 所示

使用 S0BUF 作为一个目的寄存器可通过任何指令来启动传输。“写到 S0BUF ”信号触发 UART0 引擎开始发送。Tx 的初始状态由 SM20 定义。SM20 =0 时，Tx 初始电平高。如果 SM20 = 1，则为低。S0BUF 里面的数据在 TXD0(P3.1)脚的每一个上升沿移出到 RXD0(P3.0)脚。八个上升沿移位时钟过后，硬件置 TI0 为 1 标志发送完成且中断向量可以由 BTI 和 UTIE 切换到系统标志中断。模式 0 发送时序见图 19-4。

当 REN0=1 和 RI0=0 时接收启动。在下一个指令周期，RX0 控制单元写 11111110 到接收移位寄存器，且在下一个时钟阶段激活接收。Rx 的初始状态可以由 S0DOR 定义。当 S0DOR =1 时，Tx 的初始电平为高。如果 S0DOR= 0，则为低。

接收是由直接 RX 时钟到 TXD0 引脚的额外输出功能的移位时钟来实现的。当接收激活时，在移位时钟的下降沿采样 RXD0(P3.0)脚并移到寄存中。八个下降沿移位时钟过后，硬件置 RI0 为 1 标志接收完成。模式 0 接收时序见图 19-5。

图 19-3. 串口 0 模式 0

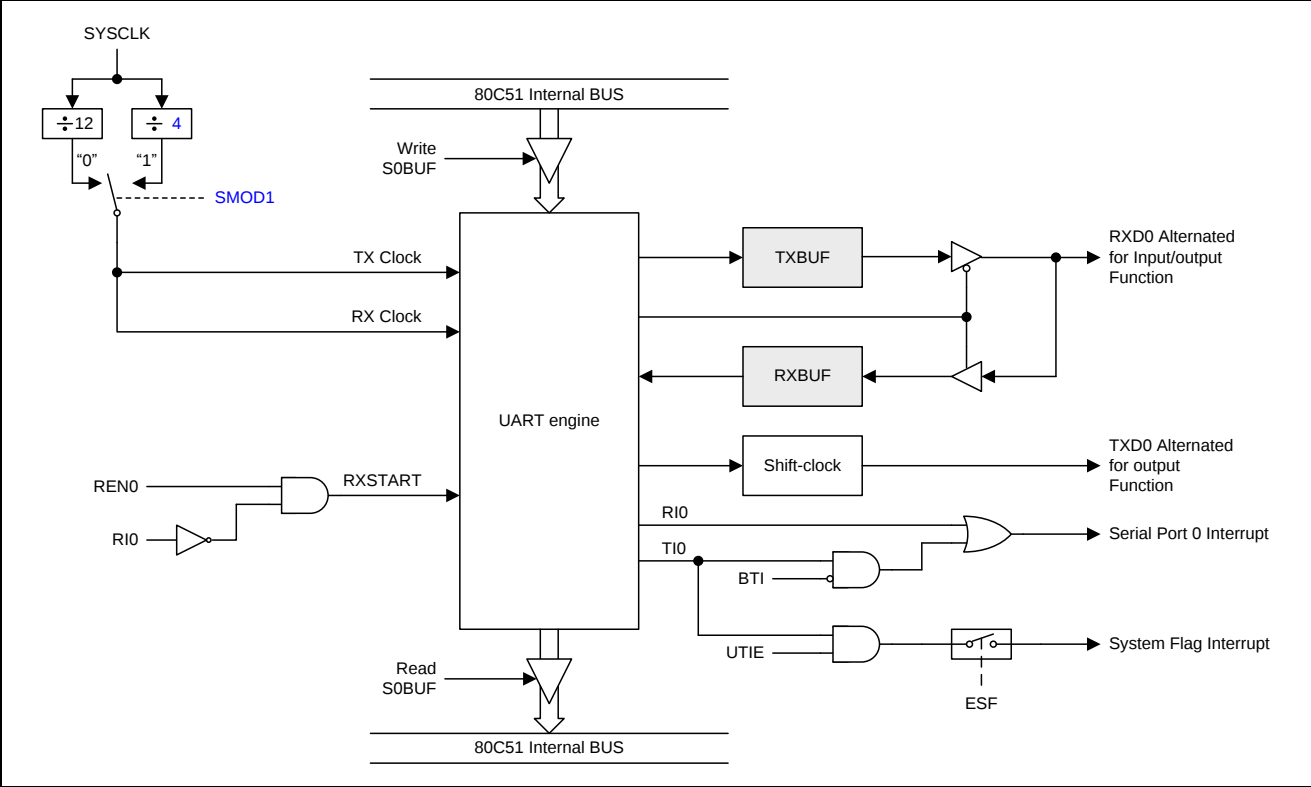


图 19-4. 模式 0 发送波形

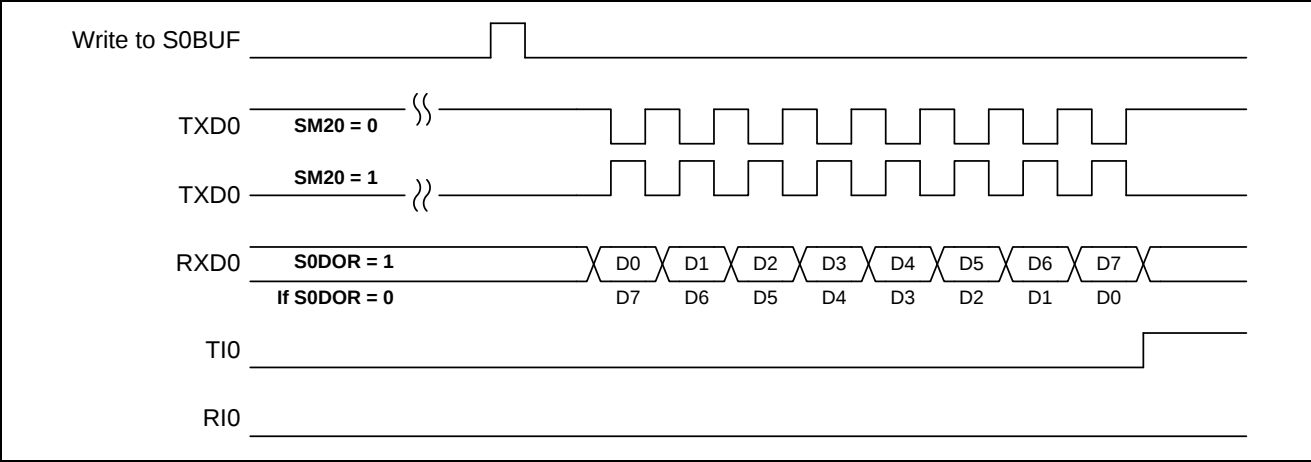
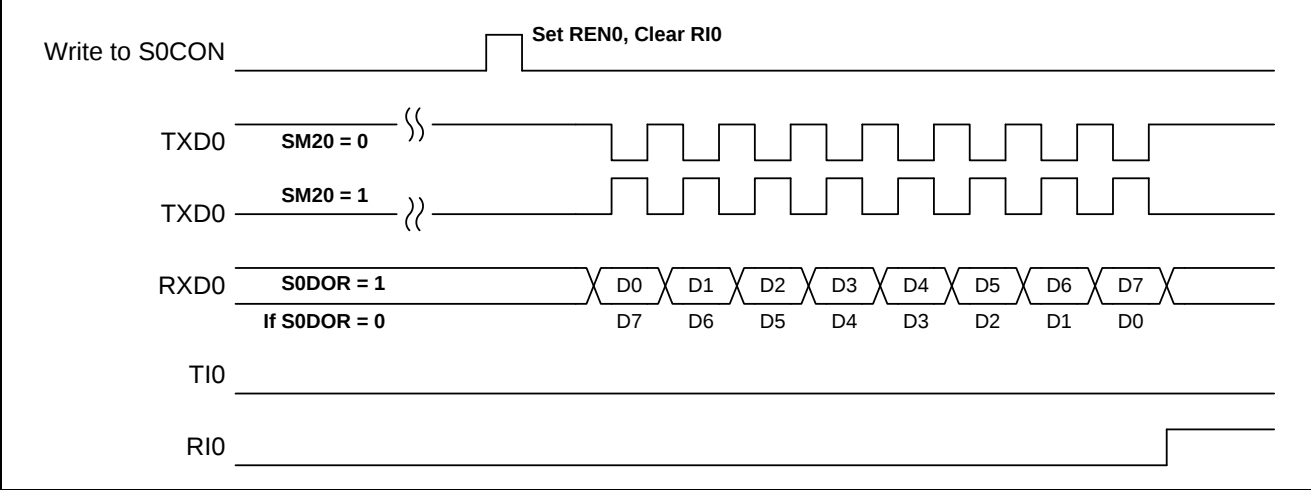


图 19-5. 模式 0 接收波形



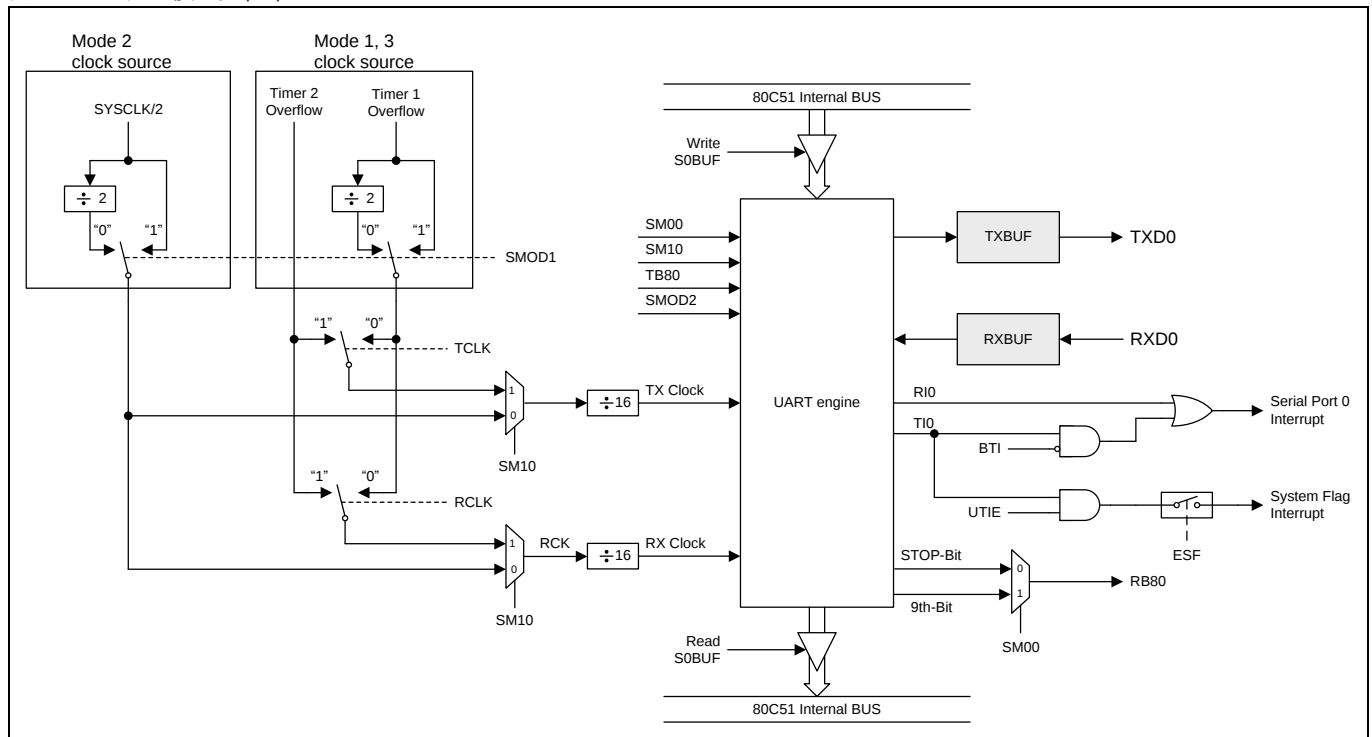
19.3. 串口 0 模式 1

通过 TXD0 发送 10 位数据或通过 RXD0 接收 10 位数据：一个起始位(0)，8 个数据位(低位先出)，和一个停止位(1)。在接收时，停止位进入 S0CON 的 RB80，波特率由定时器 1 或定时器 2 的溢出速率来决定。模式 1 数据帧时序如图 19-1 所示并且模式 1 的简化功能框图如图 19-6 所示。

使用 S0BUF 作为目的寄存器的任何指令来启动传输。“写到 S0BUF”的信号请求 UART0 引擎开始发送，当收到一个发送请求后，UART0 将在 TX 时钟的上升沿开始发送。S0BUF 中的数据从 TXD0 引脚串行输出，数据帧如图 19-1 所示及数据宽度根据 TX 时钟不同而不同。当 8 位数据发送完后，硬件将置位 TI0 表示发送结束，并且它的中断向量可以由 BTI 和 UTIE 切换到系统标志中断。

当串口 0 控制器在 RCK 采样时钟下检测到在 RXD0 有负跳变的起始位时接收开始。在 RXD0 引脚上的数据将被串行口 0 的位侦测器采样。当收到停止位后，硬件置位 RI0 表示接收结束并把停止位加载到 S0CON 寄存器的 RB80。

图 19-6. 串口模式 1, 2, 3



19.4. 串口 0 模式 2 和模式 3

通过 TXD0 传送 11 位或通过 RXD0 接收 11 位：一个起始位(0)，8 个数据位(低位在先)，一个可编程的第 9 个数据位和一个停止位(1)。在传送时，数据的第 9 位(TB80)可分配为 0 或 1。在接收时，数据的第 9 位将进入到 S0CON 的 RB80。在模式 2 波特率可编程为 1/16，1/32 或 1/64 的系统时钟频率。模式 3 可以产生可以从定时器 1 或定时器 2 产生可变的波特率。

模式 2 和 3 数据帧如图 19-2 所示，模式 2 和模式 3 的串行口功能框图如图 19-5 所示。接收部分和模式 1 相同。与模式 1 传送部分不同的仅仅是传送移位寄存器的第 9 位。

“写到 S0BUF”的信号请求 UART0 引擎加载 TB8 到发送移位寄存器的第 9 位并开始发送，当收到一个发送请求后，UART0 将在 TX 时钟的上升沿开始发送。S0BUF 中的数据从 TXD0 引脚串行输出，数据帧如图 19-2 所示及数据宽度根据 TX 时钟不同而不同。当 9 位数据发送完后，硬件将置位 TI0 表示发送结束,并且它的中断向量可以由 BTI 和 UTIE 切换到系统标志中断。

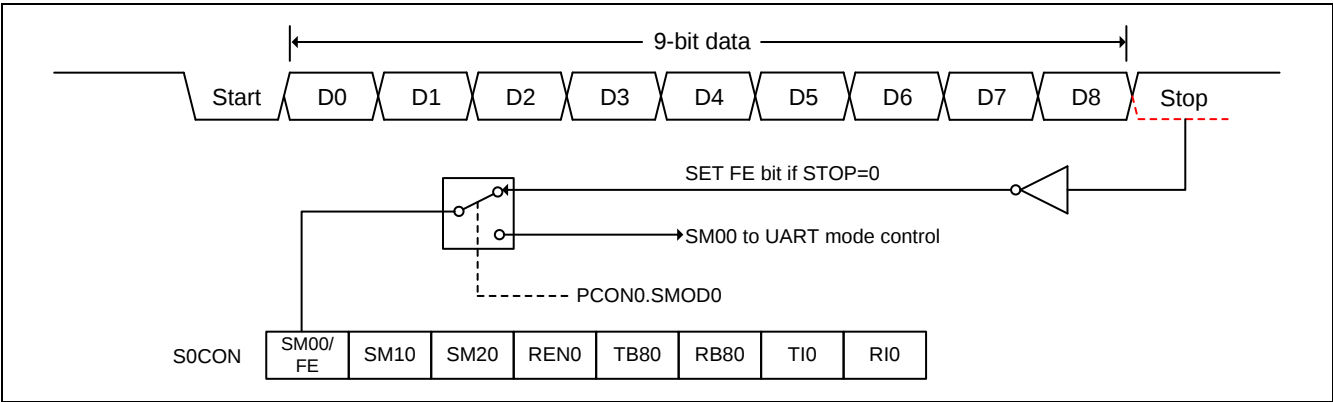
当串口 0 控制器在 RCK 采样时钟下检测到在 RXD0 有负跳变的起始位时接收开始。在 RXD0 引脚上的数据将被串口 0 的位侦测器采样。当收数据接收完后，硬件置位 RI0 表示接收结束并把第 9 位加载到 S0CON 寄存器的 RB80。

在四种模式中，使用 S0BUF 作为一个目的寄存器，可以通过任何指令发起传输。在模式 0，当 RI0=0 且 REN0=1 时启动接收。在其它模式，在 REN0=1 时，收到有负跳变的起始位时启动接收。

19.5. 帧错误侦测

开启帧错误侦测功能后，UART0 会在通讯中侦测是否丢失停止位，如果丢失一个停止位，就设置 S0CON 寄存器的 FE 标志位。FE 标志位和 SM00 标志位共享 SCON0.7，SMOD0 标志位(PCON.6)决定 S0CON.7 究竟代表哪个标志，如果 SMOD0 位(PCON0.6)置位则 S0CON.7 就是 FE 标志，SMOD0 位清零则 S0CON.7 就是 SM00 标志。当 S0CON.7 代表 FE 时，只能软件清零。参考图 19-7.。

图 19-7. UART0 帧错误侦测



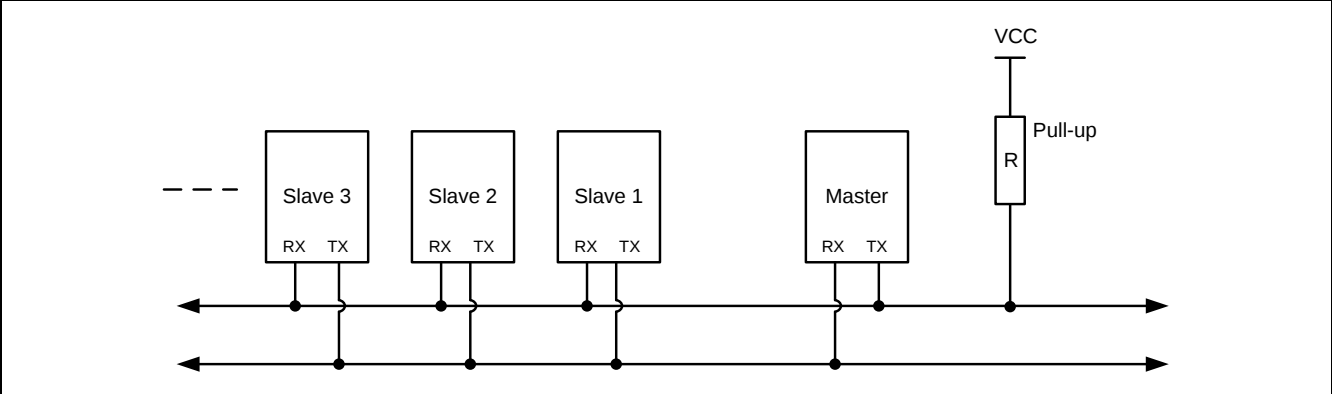
19.6. 多处理器通讯

模式 2 和 3 在用作多处理器通讯时有特殊的规定如图 19-8 所示。在这两种模式，接收 9 个数据位。第 9 个数据位存入 RB80 之后，接着进来一个停止位。端口可以编程为：在 RB80=1 时，当收到停止位后，串口中断将激活。这种特征通过设置 SM20 位(在 S0CON 寄存器中)来使能。这种方式用于多处理器系统如下：

当主处理器想传送一个数据块到多个从机中的某一个时，首先传送想要传送的目标地址标识符的地址。地址字节与数据字节的区别在于，在地址字节中第 9 位为 1，数据字节中为 0。当 SM20=1 时，收到一个数据字节将不会产生中断。然而一个地址字节将引发所有从机中断。因而所有的从机可以侦测收到的字节是否是自己的地址。从机地址将清除 SM20 位并准备好接收即将进来的所有数据。从机地址不匹配的将保持 SM20 置位，并继续他们的工作，忽略进来的数据字节。

SM20 在模式 0 和模式 1 没有影响，但是可以用来侦测停止位的有效性。在接收模式 1 中，如果 SM20=1，除非收到一个有效的停止位否则接收中断不会被激活。

图 19-8. UART0 多处理器通讯

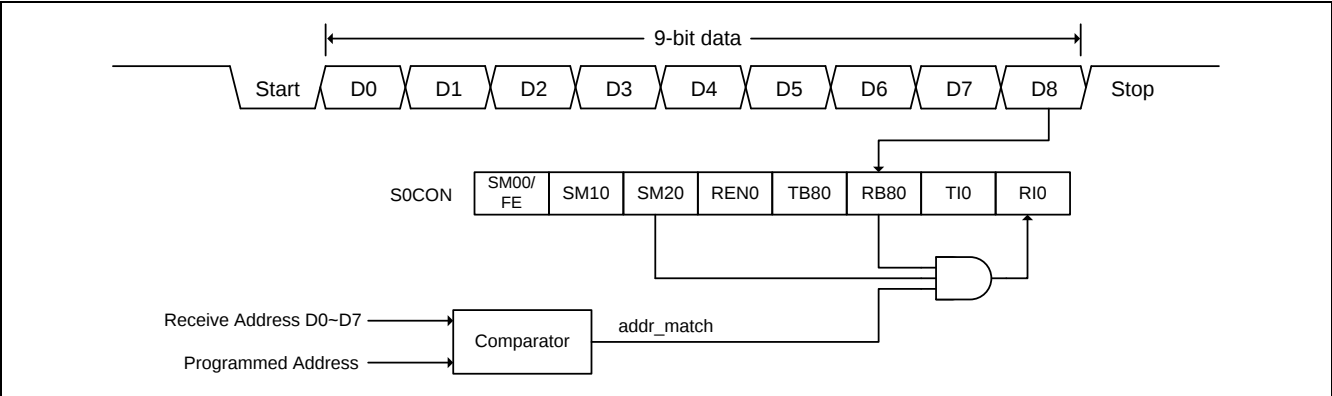


19.7. 自动地址识别

自动地址识别通过硬件比较可以让 UART0 识别串行码流中的地址部分，该功能免去了使用软件识别时需要大量代码的麻烦。该功能通过设定 S0CON 的 SM20 位来开启。

在 9 位数据 UART0 模式下，即模式 2 和模式 3，收到特定地址或广播地址时自动置位接收中断(RI0)标志，9 位模式的第 9 位信息为 1 表明接收的是一个地址而不是数据。自动地址识别功能请参考图 19-9。

图 19-9. 自动地址识别



注意:

- (1) 收到匹配地址后(addr_match=1)，清 SM20 以接收数据字节。
- (2) 收完全部数据字节后,置 SM20 为 1 以等待下一个地址。

在 8 位模式，即模式 1 下，如果 SM20 置位并且在 8 位地址与给定地址或广播地址核对一致后收到有效停止位则 RI0 置位。模式 0 是移位寄存器模式，SM20 被忽略。

使用自动地址识别功能可以让一个主机选择性的同一个或多个从机进行通讯，所有从机可以使用广播地址接收信息。两个特殊功能寄存器(SADDR 和 SADEN 地址掩码寄存器)用来定义从机地址。

SADEN 用来定义 SADDR 中的位，哪些是可用的，哪些位是“忽略”的。使用 SADEN 掩码和 SADDR 执行一个逻辑与操作创建一个“给定”地址，该地址将用作从机的地址，并且主机可以将该“给定”地址在总线上发送，以从多个从机上识别出该从机。

下面的实例帮助理解这个方案的通用性：

从机 0	从机 1
SADDR = 1100 0000	SADDR = 1100 0000
SADEN = 1111 1101	SADEN = 1111 1110
Given = 1100 00X 0	Given = 1100 00 0 X

上面的例子中 SADDR 是相同的值，而使用 SADEN 数据来区分两个从机。从机 0 要求第 0 位必须为 0，并忽略第 1 位的值；从机 1 要求第 1 位必须为 0，并忽略第 0 位的值。从机 0 的唯一地址是 1100 0010，而从机 1 的唯一地址是 1100 0001，地址 1100 0000 是可以同时寻找到从机 0 和从机 1 的。

下面一个更为复杂的系统可以寻址到从机 1 和从机 2，而不会寻址到从机 0：

从机 0	从机 1	从机 2
SADDR = 1100 0000	SADDR = 1110 0000	SADDR = 1110 0000
SADEN = 1111 1001	SADEN = 1111 1010	SADEN = 1111 1100
Given = 1100 0XX 0	Given = 1110 0X 0 X	Given = 1110 0 0 XX

上面的例子中，3 个从机的低 3 位地址不一样，从机 0 要求第 0 位必须为 0，1110 0110 可以唯一寻址从机 0；从机 1 要求第 1 位必须为 0，1110 0101 可以唯一寻址从机 1；从机 2 要求第 2 位必须为 0，它的唯一地址是 1110 0011。为了寻址到从机 0 和从机 1 而不会寻址到从机 2，可以使用地址 1110 0100，因为这个地址第 2 位是 1。

每个从机的广播地址的创建都是通过 SADDR 和 SADEN 的逻辑或，0 在结果中按忽略处理。大部分情况下，体现忽略处理是所有为 1，使用十六进制的 FF 作为广播地址。

当复位时，SADDR (SFR 地址 0xA9)和 SADEN (SFR 地址 0xB9)被加载为 0。这将生成一个“给定”地址为“忽略”(“XXXX XXXXb”)和一个“广播地址”地址为“忽略”(“XXXX XXXXb”)。这有效地禁用了自动寻址模式，并允许微控制器使用标准的 80C51 类型 UART 驱动，而这些驱动不使用该特性。

19.8. 串口 0 模式 4 (SPI 主机)

MG82F6P32 的串口 0 嵌入了一个额外的模式 4 来支持 SPI 主机引擎。模式 4 由 SM30、SM00 和 SM10 选择。请参考“表 19-1。”

SMOD1 也可控制 SPI 的传输速度。如果 SMOD1 = 0，则 SPI 的时钟频率是 SYSCLK/12。如果 SMOD1 = 1，则 SPI 的时钟频率是 SYSCLK/4。当选择 S0BRG 作为时钟源时，SPI 时钟频率不受 SMOD1 控制，等于 S0TOF/4。MG82F6P32 的 SPI 主机使用 TXD0 作为 SPICLK，RXD0 作为 MOSI，以及 S0MI 作为 MISO。而 nSS 由 MCU 软件选择在其它端口引脚。SPI 连接如图 19-10 所示。他支持多从机通讯架构见图 19-11..。

图 19-10. 串口 0 模式 4，单主机和单从机架构 (n = 0)

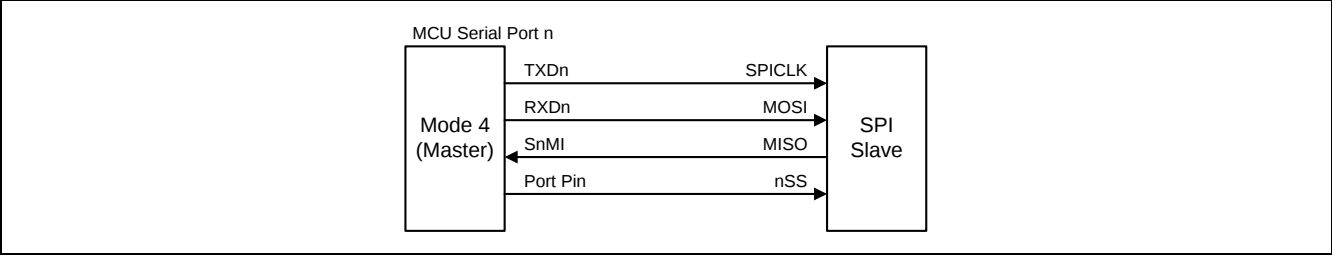
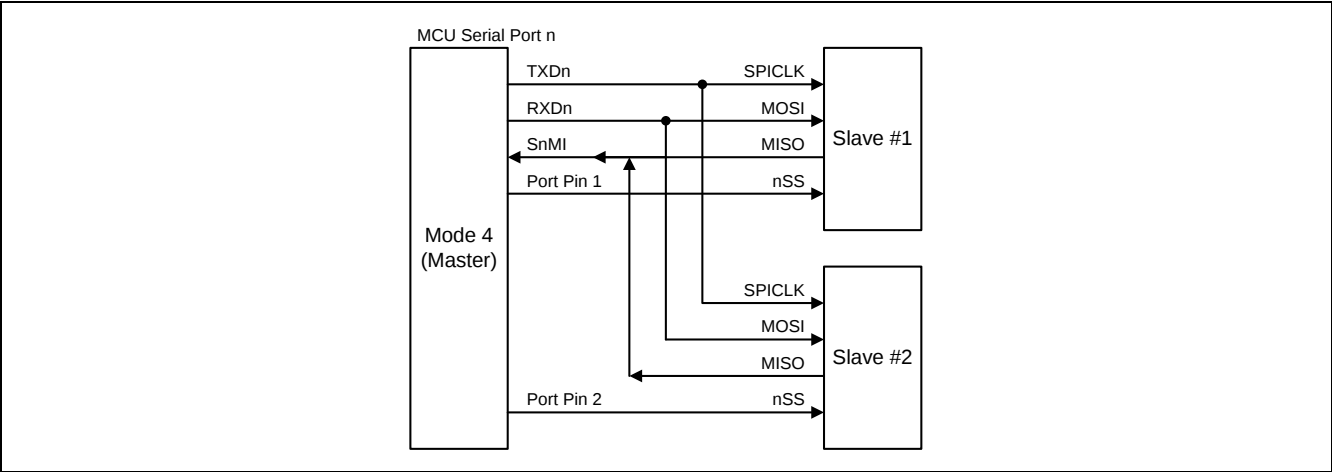


图 19-11. 串口 0 模式 4，单主机和多从机架构(n = 0)



SPI 主机能满足笙泉 MA82/84 系列 MCU 全功能 SPI 模块的传输，具有 CPOL, CPHA 和 DORD 选择。对于 CPOL 和 CPHA 条件，**MG82F6P32** 使用一种简单的方法，通过初始化 SPI 时钟分配的端口引脚(TXD0)极性来匹配它们。串行口模式 4 的 4 个 SPI 工作模式如表 19-2 所示。

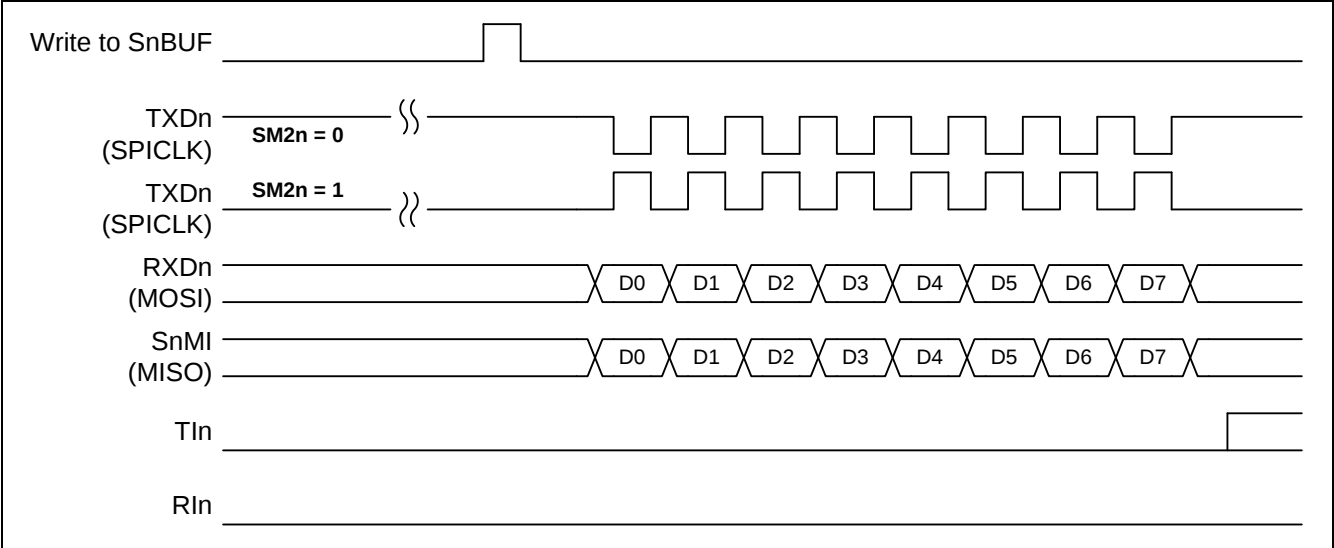
表 19-2. 串口 0 模式 4 的 SPI 模式配置

SPI 模式	CPOL	CPHA	在 SnCON 中配置 SM2n & TB8n
0	0	0	SM2n = 1, TB8n = 1
1	0	1	SM2n = 1, TB8n = 0
2	1	0	SM2n = 0, TB8n = 1
3	1	1	SM2n = 0, TB8n = 0

SPI 系列传输的位序控制(DORD)，**MG82F6P32** 提供了位序控制(S0DOR)。S0DOR 的默认值是“1”且位序控制为低位在先(LSB)。

传输由任何使用 S0BUF 作为目的寄存器的指令发起。“写入 S0BUF”信号触发 UART 引擎开始传输。S0BUF 中的数据将作为 MOSI 串行数据转移到 RXD0 引脚。SPI 移位时钟建立在用于 SPICLK 输出的 TXD0 引脚上。TxD 的初始状态可以通过 SM2n 定义。SM2n = 0 时，TxD 初始电平为高。如果 SM2n = 1，它为低。当移位时钟通过 8 个上升沿后，硬件置位 TI0，表示传输结束。S0MI 引脚上的内容将被采样并移位到移位寄存器中。然后，“读取 S0BUF”可以得到 SPI 移位数据。模式 0 传输波形如图 19-12 所示。在模式 4 中不会置位 RI0。

图 19-12. 串口 0 模式 4 发送波形(n = 0)



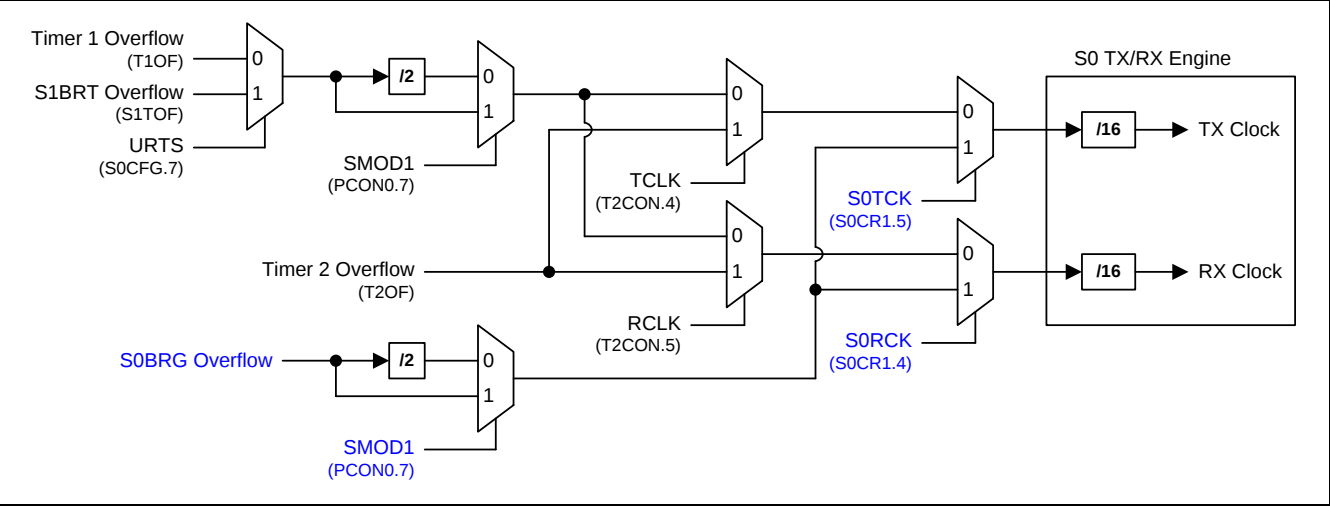
19.9. 波特率设置

位T2X12 (T2MOD.4)、T1X12 (AUXR2.3)和SMOD2 (S0CFG.6)提供一个新的波特率选项设置，如下所列：

19.9.1. S0 的波特率选择

UART0 运行在模式 1 和模式 3 时，清零 T2CON 寄存器的位 TCLK 和 RCLK 软件可选择定时器 1 作为波特率发生器。此刻，如果 URTS(S0CFG.7)置位，定时器 1 的溢出信号将被 UART1 的波特率发生器(S1BRG)取代。换句话说，一旦 RCLK=0、TCLK=0 和 URTS=1 用户可采用 S1BRG 作为 UART0 模式 1 或模式 3 的波特率发生器。这样，定时器 1 可以自由地做其它操作。当然，如果 UART1(模式 1 或模式 3)这时也在运行，那么这两个串口将有同样的波特率。

图 19–13. S0 模式 1,2 和 3 的波特率选择



19.9.2. 模式 0&4 的波特率(串行时钟频率)

S0 标准模式 0&4 的波特公式

Baud Rate = $\frac{F_{\text{SYSCLK}}}{12}$; SMOD1=0

or = $\frac{F_{\text{SYSCLK}}}{4}$; SMOD1=1

注意：
如果 **SMOD1=0**，波特率公式跟标准 8051 一样。

S0 增强模式 0&4 波特率公式(时钟源来自 S0BRG)

Baud Rate = $\frac{F_{\text{SYSCLK}}}{4 \times (256 - \text{S0BRT})}$

19.9.3. 模式 2 波特率

Mode 2 Baud Rate = $\frac{2^{\text{SMOD1}} \times 2^{(\text{SMOD2} \times 2)}}{64} \times F_{\text{SYSCLK}}$

注意：
如果 **SMOD2=0**，波特率公式跟标准 8051 一样。如果 **SMOD2=1**，波特率设置有增强功能。[表 19–3](#) 定义了模式 2 波特率发生器由 **SMOD2** 因数决定的波特率设置。

表 19–3. SMOD2 在模式 2 的应用标准

SMOD2	SMOD1	波特率	备注	推荐的最大接收误差 (%)
0	0	缺省波特率	标准功能	± 3%
0	1	双倍波特率	标准功能	± 3%
1	0	双倍波特率 x2	增强型功能	± 2%
1	1	保留		

表 19-4. S0 模式 2 波特率@ F_{SYSC}CLK=11.0592MHz

Baud Rate	SMOD2	SMOD1	Error
172,800	0	0	0.0%
345,600	0	1	0.0%
691,200	1	0	0.0%

表 19-5. S0 模式 2 波特率@ F_{SYSC}CLK=12.00MHz

Baud Rate	SMOD2	SMOD1	Error
187,500	0	0	0.0%
375,000	0	1	0.0%
750,000	1	0	0.0%

19.9.4. 模式 1&3 波特率

19.9.4.1. 使用定时器 1 和 SnBRG 作为波特率发生器

如“图 19-13.”所示, the S0 可以选择定时器 1 S0BRG 和 S1BGR 作为波特率来源

S0BRG 是内嵌波特率发生器, 其详细功能在“19.11.1 S0 波特率发生器 (S0BRG)”一节中描述。当使用 S0BRG 作为 S0 的波特率发生器时, 波特率如下所示。

MG82F6P32 的第二 UART (S1)具有独立的波特率发生器 S1BRG。S0 可以设置 URTS (S0CFG.7)选择 S1BRT 作为 UART 模式 1 和模式 3 的定时器源。有关 S0 波特率选择的详细信息, 请参阅“20.9 S1 波特率发生器用于 S0”。

Timer 1:

Mode 1, 3 Baud Rate = $\frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSC}CLK}{12 \times (256 - TH1)}$; T1X12=0

or = $\frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSC}CLK}{1 \times (256 - TH1)}$; T1X12=1

SnBRG (n = 0, 1):

Mode 1, 3 Baud Rate = $\frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSC}CLK}{12 \times (256 - SnBRT)}$; S0TX12=0

or = $\frac{2^{SMOD1} \times 2^{(SMOD2 \times 2)}}{32} \times \frac{F_{SYSC}CLK}{1 \times (256 - SnBRT)}$; S0TX12=1

注意: 当 SMOD2=0, T1X12=0 时, 波特率公式与标准 8051 相同。如果 SMOD2=1, 则有增强的波特率设置功能。表 19-6 定义了定时器 1 和 S1BGR 波特率发生器中带有 SMOD2 因子的波特率设置。S1BRG 的设置请参考“20.9”节

表 19-6. SMOD2 在模式 1 和 3 使用定时器 1&SnBRG 的应用标准

SMOD2	SMOD1	波特率	备注	推荐的最大接收误差 (%)
0	0	缺省波特率	标准功能	± 3%
0	1	双倍波特率	标准功能	± 3%
1	0	双倍波特率 X2	增强型功能	± 2%
1	1	保留		

表 19-7~ 表 19-14 列出了 8 位自动加载模式的定时器 1 中各种常用的波特率和怎样获得。对于非标准波特率, 当 F_{SYSC}CLK = 48MHz 时, 频率最大是 6MHz。

表 19-7. 定时器 1 或 SnBRG 产生常用波特率@ F_{sysclk}=11.0592MHz

TH1 or SnBRT 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T1X12=0				T1X12=1			
1200	232	208	--	0.0%	--	--	--	--
2400	244	232	--	0.0%	112	--	--	0.0%
4800	250	244	--	0.0%	184	112	--	0.0%
9600	253	250	--	0.0%	220	184	--	0.0%
14400	254	252	--	0.0%	232	208	--	0.0%
19200	--	253	--	0.0%	238	220	--	0.0%
28800	255	254	--	0.0%	244	232	--	0.0%
38400	--	--	--	--	247	238	--	0.0%
57600	--	255	--	0.0%	250	244	--	0.0%
115200	--	--	--	--	253	250	--	0.0%
230400	--	--	--	--	--	253	250	0.0%
460.8K	--	--	--	--	--	--	253	0.0%
691.2K	--	--	--	--	--	--	254	0.0%
1.3824M	--	--	--	--	--	--	255	0.0%

表 19-8. 定时器 1 或 SnBRG 产生常用波特率 @ F_{sysclk}=22.1184MHz

TH1 or SnBRT 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T1X12=0				T1X12=1			
1200	208	160	--	0.0%	--	--	--	--
2400	232	208	--	0.0%	--	--	--	--
4800	244	232	--	0.0%	112	--	--	0.0%
9600	250	244	--	0.0%	184	112	--	0.0%
14400	252	248	--	0.0%	208	160	--	0.0%
19200	253	250	--	0.0%	220	184	--	0.0%
28800	254	252	--	0.0%	232	208	--	0.0%
38400	--	253	--	0.0%	238	220	--	0.0%
57600	255	254	--	0.0%	244	232	--	0.0%
115200	--	255	--	--	250	244	--	0.0%
230400	--	--	--	--	253	250	--	0.0%
460.8K	--	--	--	--	--	253	250	0.0%
691.2K	--	--	--	--	--	--	252	0.0%
921.6K	--	--	--	--	--	--	253	0.0%
1.3824M	--	--	--	--	--	--	254	0.0%
1.8432M	--	--	--	--	--	--	--	--
2.7648M	--	--	--	--	--	--	255	0.0%

表 19-9. 定时器 1 或 SnBRG 产生常用波特率@ F_{sysclk}=12.0MHz

TH1 or SnBRT 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T1X12=0				T1X12=1			
1200	230	204	--	0.16%	--	--	--	--
2400	243	230	--	0.16%	100	--	--	0.16%
4800	--	243	--	0.16%	178	100	--	0.16%
9600	--	--	--	--	217	178	--	0.16%
14400	--	--	--	--	230	204	--	0.16%
19200	--	--	--	--	--	217	--	0.16%
28800	--	--	--	--	243	230	--	0.16%
38400	--	--	--	--	246	236	--	2.34%
57600	--	--	--	--	--	243	--	0.16%
115200	--	--	--	--	--	--	243	0.16%

表 19-10. 定时器 1 或 SnBRG 产生常用波特率@ F_{sysclk}=24.0MHz

TH1 or SnBRT 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T1X12=0				T1X12=1			
1200	204	152	--	0.16%	--	--	--	--
2400	230	204	--	0.16%	--	--	--	--
4800	243	230	--	0.16%	100	--	--	0.16%
9600	--	243	--	0.16%	178	100	--	0.16%
14400	--	--	--	--	204	152	--	0.16%
19200	--	--	--	--	217	178	--	0.16%
28800	--	--	--	--	230	204	--	0.16%
38400	--	--	--	--	--	217	--	0.16%
57600	--	--	--	--	243	230	--	0.16%
115200	--	--	--	--	--	243	--	0.16%
230400	--	--	--	--	--	--	243	0.16%

表 19-11. 定时器 1 或 SnBRG 产生常用波特率@ F_{sysclk}=29.4912MHz

TH1 or SnBRT 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T1X12=0				T1X12=1			
1200	192	128	--	0.0%	--	--	--	--
2400	224	192	--	0.0%	--	--	--	--
4800	240	224	--	0.0%	64	--	--	0.0%
9600	248	240	--	0.0%	160	64	--	0.0%
14400	--	--	--	--	192	128	--	0.0%
19200	252	248	--	0.0%	208	160	--	0.0%
28800	--	--	--	--	224	192	--	0.0%
38400	--	--	--	--	232	208	--	0.0%
57600	--	--	--	--	240	224	--	0.0%
115200	--	--	--	--	248	240	--	0.0%
230400	--	--	--	--	252	248	--	0.0%

460.8K	--	--	--	--	254	252	--	0.0%
921.6K	--	--	--	--	255	254	--	0.0%
1.8432M	--	--	--	--	--	255	254	0.0%

表 19-12. 定时器 1 或 SnBRG 产生常用波特率@ F_{sysclk}=44.2368MHz

TH1 or SnBRT 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T1X12=0				T1X12=1			
1200	160	64	--	0.0%	--	--	--	--
2400	208	160	--	0.0%	--	--	--	--
4800	232	208	--	0.0%	--	--	--	--
9600	244	232	--	0.0%	112	--	--	0.0%
14400	248	240	--	0.0%	160	64	--	0.0%
19200	250	244	--	0.0%	184	112	--	0.0%
28800	252	248	--	0.0%	208	160	--	0.0%
38400	253	250	--	0.0%	220	184	--	0.0%
57600	254	252	--	0.0%	232	208	--	0.0%
115200	255	254	--	0.0%	244	232	--	0.0%
230400	--	255	--	0.0%	250	244	--	0.0%
460.8K	--	--	--	--	253	250	--	0.0%
691.2K	--	--	--	--	--	--	--	--
921.6K	--	--	--	--	--	253	--	0.0%
1.3824M	--	--	--	--	--	--	--	--
1.8432M	--	--	--	--	--	--	--	--
2.7648M	--	--	--	--	--	255	254--	0.0%

表 19-13. 定时器 1 或 SnBRG 产生常用波特率 @ F_{sysclk}=32MHz

TH1 or SnBRT 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T1X12=0				T1X12=1			
1200	187	118	--	0.64%	--	--	--	--
2400	221	186	--	-0.79%	--	--	--	--
4800	239	222	--	2.12%	48	--	--	0.16%
9600	--	239	--	2.12%	152	48	--	0.16%
14400	--	--	--	--	187	118	--	0.64%
19200	--	--	--	--	204	152	--	0.16%
28800	--	--	--	--	221	186	--	-0.79%
38400	--	--	--	--	230	204	152	0.16%
57600	--	--	--	--	239	222	--	2.12%
115200	--	--	--	--	--	239	--	2.12%

表 19-14. 定时器 1 或 SnBRG 产生常用波特率@ F_{sysclk}=48.0MHz

TH1 or SnBRT重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T1X12=0				T1X12=1			
1200	152	48	--	0.16%	--	--	--	--
2400	204	152	--	0.16%	--	--	--	--
4800	230	204	--	0.16%	--	--	--	--
9600	243	230	--	0.16%	100	--	--	0.16%
14400	--	239	--	2.12%	152	48	--	0.16%
19200	--	243	--	0.16%	178	100	--	0.16%
28800	--	--	--	--	204	152	--	0.16%
38400	--	--	--	--	217	178	--	0.16%
57600	--	--	--	--	230	204	--	0.16%
115200	--	--	--	--	243	230	--	0.16%
230400	--	--	--	--	--	243	230	0.16%
460.8K	--	--	--	--	--	--	243	0.16%

19.9.4.2. 使用定时器 2 作为波特率发生器

当定时器 2 作波特率发生器时(T2CON 寄存器中的 TCLK 或 RCLK 任一位为 ‘1’), 波特率如下:

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD2} \times (\text{SMOD1} + 1)} \times F_{\text{SYSCLK}}}{32 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))}; \text{T2X12}=0$$

$$\text{or} = \frac{2^{\text{SMOD2} \times (\text{SMOD1} + 1)} \times F_{\text{SYSCLK}}}{16 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))}; \text{T2X12}=1$$

注意:

如果 SMOD2=0, 波特率公式跟标准 8051 一样。如果 SMOD2=1, 波特率设置有增强功能。表 19-15 定义了定时器 2 波特率发生器由 SMOD2 因数决定的波特率设置。

表 19-15. SMOD2 在模式 1&3 使用定时器 2 的应用标准

SMOD2	SMOD1	Baud Rate	Note	Recommended Max. Receive Error (%)
0	X	Default Baud Rate	Standard function	± 3%
1	0	Double Baud Rate	Enhanced function	± 2%
1	1	Reserved		

表 19-16 ~ 表 19-23 列出了各种常用的波特率, 以及如何从定时器 2 的波特率产生器模式中获得波特率。

表 19-16. 定时器 2 产生常用波特率@ F_{SYSCLK}=11.0592MHz

[RCAP2H, RCAP2L] 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T2X12=0				T2X12=1			
1200	65248	65248	--	0.0%	64960	64960	--	0.0%
2400	65392	65392	--	0.0%	65248	65248	--	0.0%
4800	65464	65464	--	0.0%	65392	65392	--	0.0%
9600	65500	65500	--	0.0%	65464	65464	--	0.0%
14400	65512	65512	--	0.0%	65488	65488	--	0.0%
19200	65518	65518	--	0.0%	65500	65500	--	0.0%
28800	65524	65524	--	0.0%	65512	65512	--	0.0%
38400	65527	65527	--	0.0%	65518	65518	--	0.0%
57600	65530	65530	--	0.0%	65524	65524	--	0.0%
115200	65533	65533	--	0.0%	65530	65530	--	0.0%
230400	--	--	65533	0.0%	65533	65533	65530	0.0%
460.8K	--	--	--	--	--	--	65533	0.0%
691.2K	--	--	65535	0.0%	--	--	65534	0.0%
921.6K	--	--	--	--	--	--	--	--
1.3824M	--	--	--	--	--	--	65535	0.0%
1.8432M	--	--	--	--	--	--	--	--
2.7648M	--	--	--	--	--	--	--	--

表 19-17. 定时器 2 产生常用波特率 @ $F_{SYSCLK}=22.1184MHz$

[RCAP2H, RCAP2L] 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T2X12=0				T2X12=1			
1200	64960	64960	--	0.0%	64384	64384	--	0.0%
2400	65248	65248	--	0.0%	64960	64960	--	0.0%
4800	65392	65392	--	0.0%	65248	65248	--	0.0%
9600	65464	65464	--	0.0%	65392	65392	--	0.0%
14400	65488	65488	--	0.0%	65440	65440	--	0.0%
19200	65500	65500	--	0.0%	65464	65464	--	0.0%
28800	65512	65512	--	0.0%	65488	65488	--	0.0%
38400	65518	65518	--	0.0%	65500	65500	--	0.0%
57600	65524	65524	--	0.0%	65512	65512	--	0.0%
115200	65530	65530	--	0.0%	65524	65524	--	0.0%
230400	65533	65533	--	0.0%	65530	65530	--	0.0%
460.8K	--	--	65533	0.0%	65533	65533	65530	0.0%
691.2K	--	--	65534	0.0%	--	--	65532	0.0%
921.6K	--	--	--	--	--	--	65533	0.0%
1.3824M	--	--	65535	0.0%	--	--	65534	0.0%
1.8432M	--	--	--	--	--	--	--	--
2.7648M	--	--	--	--	--	--	65535	0.0%

表 19-18. 定时器 2 产生常用波特率 @ $F_{SYSCLK}=12.0MHz$

[RCAP2H, RCAP2L] 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T2X12=0				T2X12=1			
1200	65224	65224	--	0.16%	64912	64912	--	0.16%
2400	65380	65380	--	0.16%	65224	65224	--	0.16%
4800	65458	65458	--	0.16%	65380	65380	--	0.16%
9600	65497	65497	--	0.16%	65458	65458	--	0.16%
14400	65510	65510	--	0.16%	65484	65484	--	0.16%
19200	65516	65516	--	2.34%	65497	65497	--	0.16%
28800	65523	65523	--	0.16%	65510	65510	--	0.16%
38400	--	--	--	--	65516	65516	--	2.34%
57600	--	--	--	--	65523	65523	--	0.16%
115200	--	--	--	--	--	--	65523	0.16%

表 19-19. 定时器 2 产生常用波特率 @ $F_{SYSCLK}=24.0MHz$

[RCAP2H, RCAP2L] 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T2X12=0				T2X12=1			
1200	64912	64912	--	0.16%	64288	64288	--	0.16%
2400	65224	65224	--	0.16%	64912	64912	--	0.16%
4800	65380	65380	--	0.16%	65224	65224	--	0.16%
9600	65458	65458	--	0.16%	65380	65380	--	0.16%

14400	65484	65484	--	0.16%	65432	65432	--	0.16%
19200	65497	65497	--	0.16%	65458	65458	--	0.16%
28800	65510	65510	--	0.16%	65484	65484	--	0.16%
38400	65516	65516	--	2.34%	65497	65497	--	0.16%
57600	65523	65523	--	0.16%	65510	65510	--	0.16%
115200	--	--	--	--	65523	65523	--	0.16%
230400	--	--	--	--	--	--	65523	0.16%

表 19–20. 定时器 2 产生常用波特率 @ F_{SYSClk}=29.4912MHz

[RCAP2H, RCAP2L] 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T2X12=0				T2X12=1			
1200	64768	64768	--	0.0%	64000	64000	--	0.0%
2400	65152	65152	--	0.0%	64768	64768	--	0.0%
4800	65344	65344	--	0.0%	65152	65152	--	0.0%
9600	65440	65440	--	0.0%	65344	65344	--	0.0%
14400	65472	65472	--	0.0%	65408	65408	--	0.0%
19200	65488	65488	--	0.0%	65440	65440	--	0.0%
28800	65504	65504	--	0.0%	65472	65472	--	0.0%
38400	65512	65512	--	0.0%	65488	65488	--	0.0%
57600	65520	65520	--	0.0%	65504	65504	--	0.0%
115200	65528	65528	--	0.0%	65520	65520	--	0.0%
230400	65532	65532	--	0.0%	65528	65528	--	0.0%
460.8K	65534	65534	--	0.0%	65532	65532	--	0.0%
921.6K	65535	65535	65534	0.0%	65534	65534	65532	0.0%
1.8432M	--	--	--	--	--	--	--	--

表 19–21. 定时器 2 产生常用波特率 @ F_{SYSClk}=44.2368MHz

[RCAP2H, RCAP2L] 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T2X12=0				T2X12=1			
1200	64384	64384	--	0.0%	63232	63232	--	0.0%
2400	64960	64960	--	0.0%	64384	64384	--	0.0%
4800	65248	65248	--	0.0%	64960	64960	--	0.0%
9600	65392	65392	--	0.0%	65248	65248	--	0.0%
14400	65440	65440	--	0.0%	65344	65344	--	0.0%
19200	65464	65464	--	0.0%	65392	65392	--	0.0%
28800	65488	65488	--	0.0%	65440	65440	--	0.0%
38400	65500	65500	--	0.0%	65464	65464	--	0.0%
57600	65512	65512	--	0.0%	65488	65488	--	0.0%
115200	65524	65524	--	0.0%	65512	65512	--	0.0%
230400	65530	65530	--	0.0%	65524	65524	--	0.0%
460.8K	65533	65533	--	0.0%	65530	65530	--	0.0%
691.2K	65534	65534	--	0.0%	65532	65532	--	0.0%
921.6K	--	--	--	--	65533	65533	--	0.0%

1.3824M	65535	65535	--	--	65534	65534	--	0.0%
1.8432M	--	--	--	--	--	--	--	--
2.7648M	--	--	--	--	65535	65535	65534	0.0%
5.5296M	--	--	--	--	--	--	65535	0.0%

表 19–22. 定时器 2 产生常用波特率 @ F_{sysclk}=32MHz

[RCAP2H, RCAP2L] 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T2X12=0				T2X12=1			
1200	64703	64703	--	0.04%	63870	63870	--	0.04%
2400	65120	65120	--	0.16%	64703	64703	--	0.04%
4800	65328	65328	--	-0.16%	65120	65120	--	0.16%
9600	65432	65432	--	-0.16%	65328	65328	--	0.16%
14400	65467	65467	--	0.64%	65398	65398	--	0.64%
19200	65484	65484	--	0.16%	65432	65432	--	0.16%
28800	65502	65502	--	2.12%	65467	65467	--	0.64%
38400	65510	65510	--	0.16%	65484	65484	--	0.16%
57600	65519	65519	--	2.12%	65502	65502	--	2.12%
115200	--	--	--	--	--	--	65501	0.64%

表 19–23. 定时器 2 产生常用波特率 @ F_{sysclk}=48.0MHz

[RCAP2H, RCAP2L] 重载值								
SMOD1	0	1	0	误差	0	1	0	误差
SMOD2	0	0	1		0	0	1	
Baud Rate	T2X12=0				T2X12=1			
1200	64286	64286	--	0.00%	63036	63036	--	0.00%
2400	64911	64911	--	0.00%	64286	64286	--	0.00%
4800	65224	65224	--	0.16%	64911	64911	--	0.00%
9600	65380	65380	--	0.16%	65224	65224	--	0.16%
14400	65432	65432	--	0.16%	65328	65328	--	0.16%
19200	65458	65458	--	0.16%	65380	65380	--	0.16%
28800	65484	65484	--	0.16%	65432	65432	--	0.16%
38400	65497	65497	--	0.16%	65458	65458	--	0.16%
57600	65510	65510	--	0.16%	65484	65484	--	0.16%
115200	65523	65523	--	0.16%	65510	65510	--	0.16%
230400	--	--	65523	0.16%	65523	65523	65510	0.16%
460.8K	--	--	--	--	--	--	65523	0.16%

19.9.4.3. 使用分割定时器 2 作为波特率发生器

当定时器 2 为分割模式，并且被用作波特率发生器时(T2CON 的 TCLK 或 RCLK 为'1'), 波特率计算公式如下

Mode 1, 3 Baud Rate =
$$\frac{2^{SMOD2 \times (SMOD1 + 1)} \times F_{SYSCLK}}{16 \times 12 \times (256 - RCAP2L)} ; TL2X12=0$$

or =
$$\frac{2^{SMOD2 \times SMOD1 + 1} \times F_{SYSCLK}}{16 \times 1 \times (256 - RCAP2L)} ; TL2X12=1$$

注: 表 19-24 在分割定时器 2 波特率发生器中使用 SMOD2 和 SMOD1 因子定义波特率设置。.

表 19-24. SMOD2 在分割定时器 2 模式 1&3 下的应用条件

SMOD2	SMOD1	波特率	备注	推荐最大接收误差 (%)
0	X	默认波特率	标准功能	± 3%
1	0	双倍波特率	增强功能	± 2%
1	1	保留		

19.10. ARGB 发送器

- S0 设置为增强模式时，ARGB 数据在 TXD0 上输出。
- 由 S0BRG 可编程比特率 (S0 波特率发生器).
- 可选占空比 1/4 或 1/3
- 可选择总线复位时间检测 60us, 260us, 200us 和 320us

初始化 S0 为 ARGB 工作模式

- 置位{ SM30, SM00, SM10, S0RCK, S0TCK} 为 ARGB 模式
- 写 S0BRC/S0BRT SFR & S0TX12/SMOD1 控制位配置 ARGB 波特率
- 定义 S0DTY1 去配置 ARGB 占空比为 1/4 或 1/3.
- 在 S0IDT11 和 S0IDT01 上选择总线复位时间检测持续时间

传输 ARGB 数据帧的流程

- 等待 RI0(S0AGBRF)总线复位事件
- 清 RI0,然后开始写 ARGB 数据到 S0BUF
- 等待 TI0(S0AGBF),S0 TX 缓冲区空.
- 清 TI0, 然后写下一个 ARGB 数据到 S0BUF.....等等, 直到所有 ARGB 数据发送完成
- 等待 RI0 开始一个新的帧来传输 ARGB 数据

19.10.1. S0 ARGB 波特率

图 19-14. S0 ARGB 模式波特率公式: (n=0)

Sn ARGB Baud Rate =
$$\frac{1}{m} \times \frac{F_{SYSCLK}}{12 \times (256 - SnBRT)} ; SnTX12=0 ; m=4, \text{ if } SnDTY1 = 0$$

or =
$$\frac{1}{m} \times \frac{F_{SYSCLK}}{1 \times (256 - SnBRT)} ; SnTX12=1 ; m=3, \text{ if } SnDTY1 = 1$$

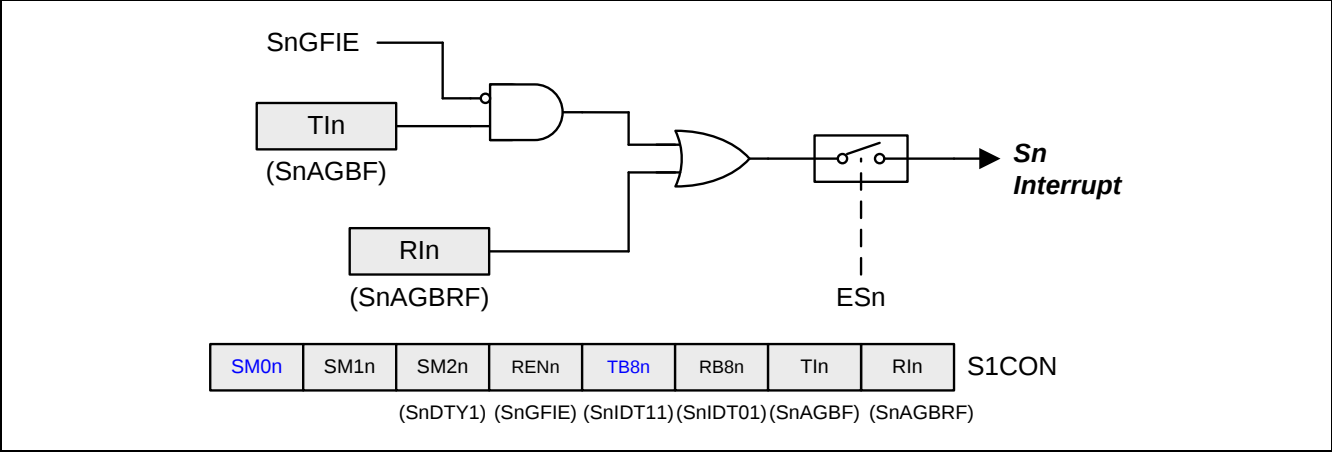
19.10.2. S0 ARGB 总线复位时间

图 19–15. S0 ARGB 模式总线复位时间 (n=0)

$$\text{Sn ARGB Reset Time} = \frac{8 \times m}{\text{Sn ARGB Baud Rate}} ; m=6/20/26/32, \text{ indexed by SnIDT1n~0n}$$

19.10.3. S0 ARGB 中断

图 19-16. S0 ARGB 模式中断(n=0)



19.10.4. S0 ARGB 寄存器

以下 UART 特殊功能寄存器被重新定义为 ARGB 操作

S0CON/ S0AGCR: 串口 0 ARGB 控制寄存器

SFR 页 = 0

SFR 地址 = 0x98

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SM00/FE	SM10	SM20	REN0	TB80	RB80	TI0	RI0
SM00	SM10	S0DTY1	S0GFIE	S0IDT11	S0IDT01	S0AGBF	S0AGBRF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: S0DTY1, ARGB 信号占空比选择

0: 选择 ARGB 信号占空比为 3/4(H)或 1/4(L).

1: 选择 ARGB 信号占空比为 2/3(H)或 1/3(L).

Bit 4: S0GFIE, 忽略 TI0 中断

0: 使能 TI0 中断。默认是使能的

1: 禁止 TI0 中断

Bit 3~2: S0IDT1~0, 选择 ARGB 总线复位时间持续时间。

S0IDT11~01	ARGB 总线复位时间 (当波特率 800K)
0 0	260us
0 1	200us
1 0	60us
1 1	320us

Bit 1: S0AGBF, 标志位报告发送保持寄存器在 ARGB 模式下完成向输出移位寄存器的传输。

0:软件写“0”清零 S0AGBF.

1: 当 S0BUF 为空时，置位 S0AGBF，如果使能 S0 中断，则产生中断。

Bit 0: S0AGBRF, ARGB 模式下 ARGB 总线复位标志位

0:软件写“0”清零 S0AGBRF

1: 当总线满足 ARGB 复位时间时，此位被置为逻辑 1。复位时间由 S0IDT1~0 定义。

19.11. 串口 0 增强功能

如果置位了 SMOD3 (S0CFG.0)， SFR 地址 0xB9 将访问 S0CR1。S0CR1 控制串口 0 的增强功能包括。:

- S0BRG 使能 S0 内置波特率发生器，S0BRG
- 使能 S0 TX 或 RX 选择 S0BRG 为波特率时基
- 使能 S0BRG 作为一般定时器
- 使能 S0 进入 ARGB 模式

S0CR1: 串口 0 控制寄存器 1 (SMOD3 = 1)

SFR 页 = 0~F

SFR 地址 = 0xB9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
S0TR	S0TX12	S0TCK	S0RCK	S0CKOE	ARTE	--	--
R/W	R/W	R/W	R/W	R/W	R/W	W	W

Bit 7: S0TR, UART0 波特率发生器控制位

0: 清零停止 S0BRG 工作

1: 置位启动 S0BRG 工作

Bit 6: S0TX12, S0BRG 时钟源选择

0: 清零选择 SYSCLK/12 作为 S0BRG 的时钟源

1: 置位选择 SYSCLK 作为 S0BRG 的时钟源

和 SM20 一起选择时钟源.

S0TX12, SM20	S0BRG 时钟选择
0 0	SYSCLK/12
0 1	T1OF
1 0	SYSCLK
1 1	T0OF

Bit 5: S0TCK, S0 控制位选择 S0BRG 溢出用于 UART0 发送时钟

0: 定时器 1 或定时器 2 溢出用于发送时钟

1: S0BRG 溢出用于发送时钟和工作模式控制

Bit 4: S0RCK, S0 控制位选择 S0BRG 溢出用于 UART0 接收时钟

0: 定时器 1 或定时器 2 溢出用于接收时钟

1: S0BRG 溢出用于接收时钟和工作模式控制

Bit 3: S0CKOE, S0BRG 时钟输出控制

0: 禁止 S0BRG 时钟输出到 S0CKO.

1: 使能 S0BRG 时钟输出到 S0CKO.

Bit 2: ARTE, 自动重复传送使能

0: 禁止自动重复传送

1: 使能自动重复传送

Bit 1~0: 保留。写该寄存器时，此位必须写“0”。

S0BRT: 串口 0 波特率定时器重载寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0x9A

复位值 = 0000-0000

7	6	5	4	3	2	1	0
S0BRT.7	S0BRT.6	S0BRT.5	S0BRT.4	S0BRT.3	S0BRT.2	S0BRT.1	S0BRT.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 波特率定时器发生器的重载值寄存器与定时器 1 的工作方式相似。

SFR 页 = 仅 0 页

SFR 地址 = 0x9B

SFR 地址 = 0x9B

SFR 地址 = 0x9B 复位值 = 0000-0000

7	6	5	4	3	2	1	0
S0BRC.7	S0BRC.6	S0BRC.5	S0BRC.4	S0BRC.3	S0BRC.2	S0BRC.1	S0BRC.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 波特率计数器寄存器与定时器 1 的工作方式相似。这个寄存器可由软件读/写。如果 S0TR (S0CR1.7) = 0, 软件写 S0BRT 数据将同时保存到 S0BRT 和 S0BRC。如果 S0TR = 1, 软件写 S0BRT 数据将不保存到 S0BRC。

19.11.1. S0 波特率发生器 (S0BRG)

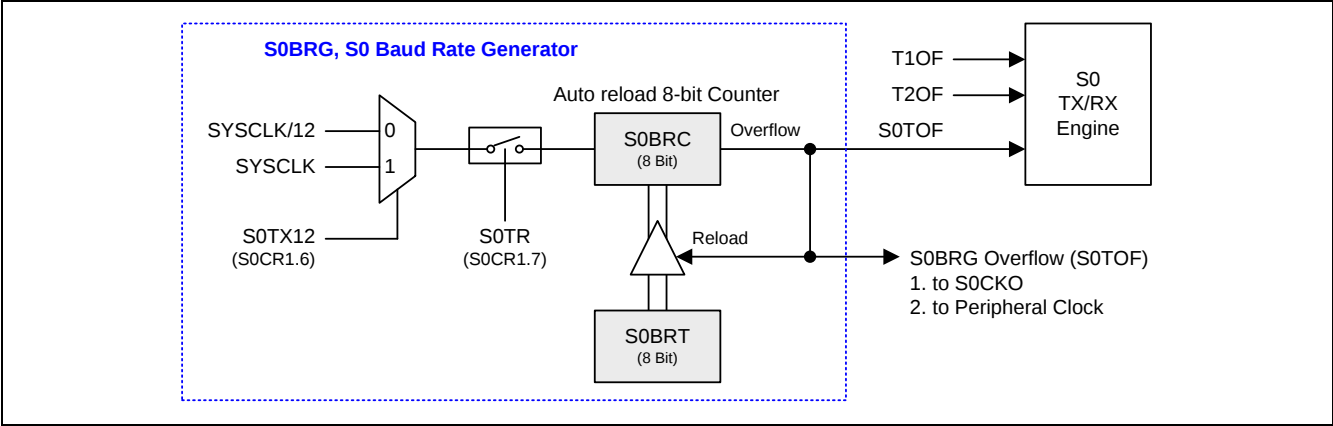
为了给 S0 更大的灵活性，可以选择 S0 波特率发生器 S0BRG 作为波特率源。**MG82F6P32** 有一个嵌入的波特率发生器来产生串口 0 操作的时钟。它由一个 8 位的向上计数器(S0BRC)和一个 8 位的重载寄存器(S0BRT)构成。S0BRC 溢出(S0TOF)是 UART0 串行引擎在所有操作模式下的时间基准且触发 S0BRT 值重载到 S0BRC 保证连续计数。

如果 S0TR= 0，软件写 S0BRT 数据将同时修改 S0BRC。在 S0TR 使能开启 S0BRC 计数之后，当 S0BRT 写入时不影响 S0BRC。修改 S1BRC 不会影响 S1BRT 的值。串口 0 波特率选择的配置请参见图 19-13..

波特率发生器也提供时钟输出的时间基准，S0CKO，为 S0BRC 溢出率的二频(S0TOF/2)。S0TOF 也供给其它外设的时钟输入切换源。无论 S0 运行还是待运行，S0BRG 总是给这些外设提供时间基准服务。

串口 0 波特率发生器配置如图 19-17.所示。

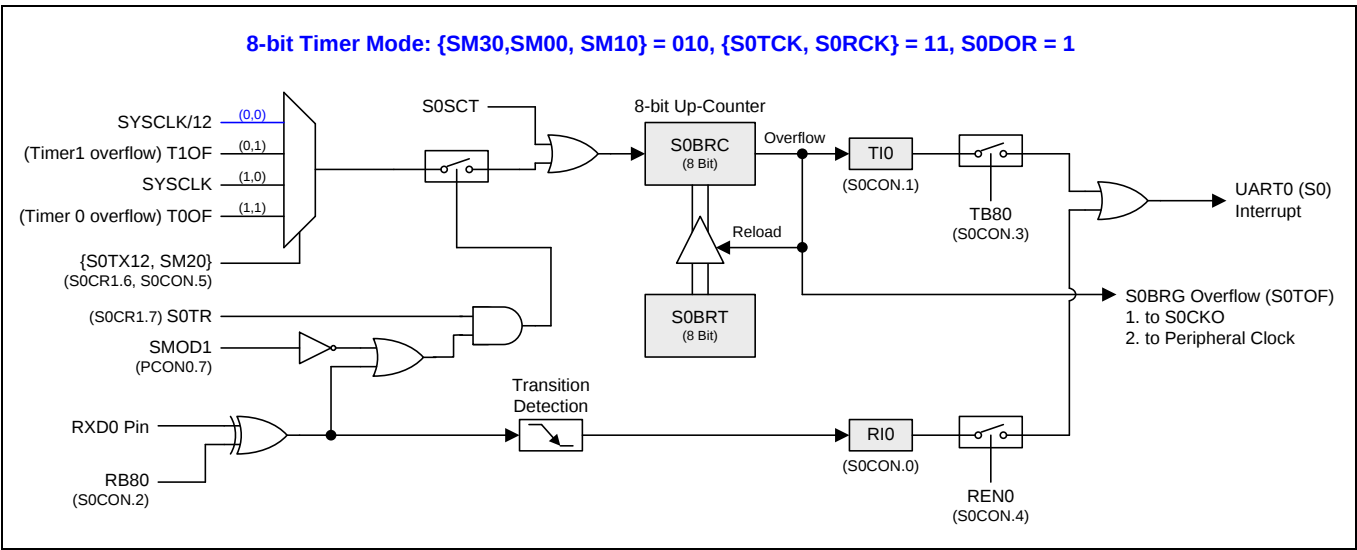
图 19-17. S0BRG 配置



19.11.2. S0 作为 8 位定时器模式

S0 8 位定时器模式如图 19-18.

图 19-18. S0 8 位定时器模式



19.11.3. S0BRG 可编程时钟输出

S0BRG 有一个时钟输出模式如图 19-19 和图 19-20.

图 19-19. S0BRG 时钟输出 (S0BRG 在 8 位定时器模式下)

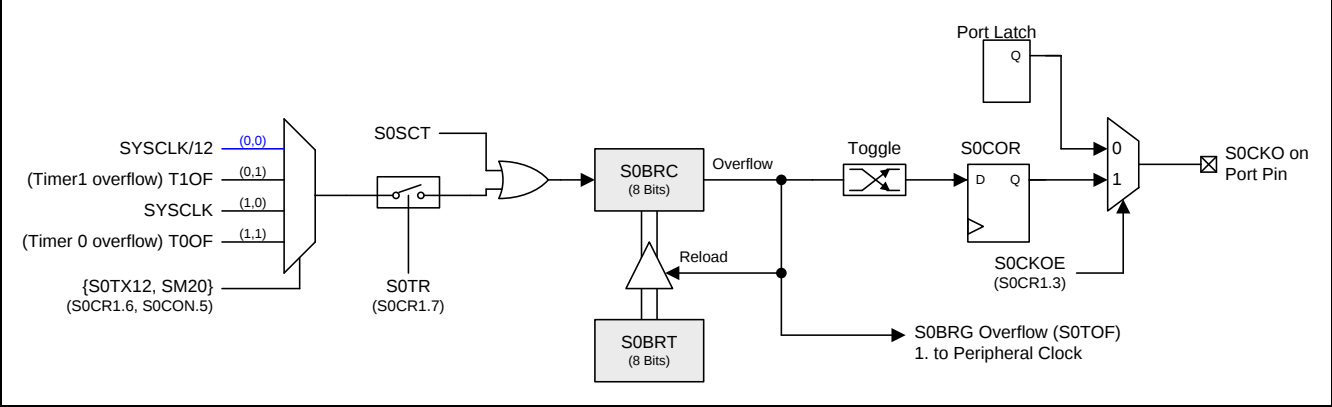
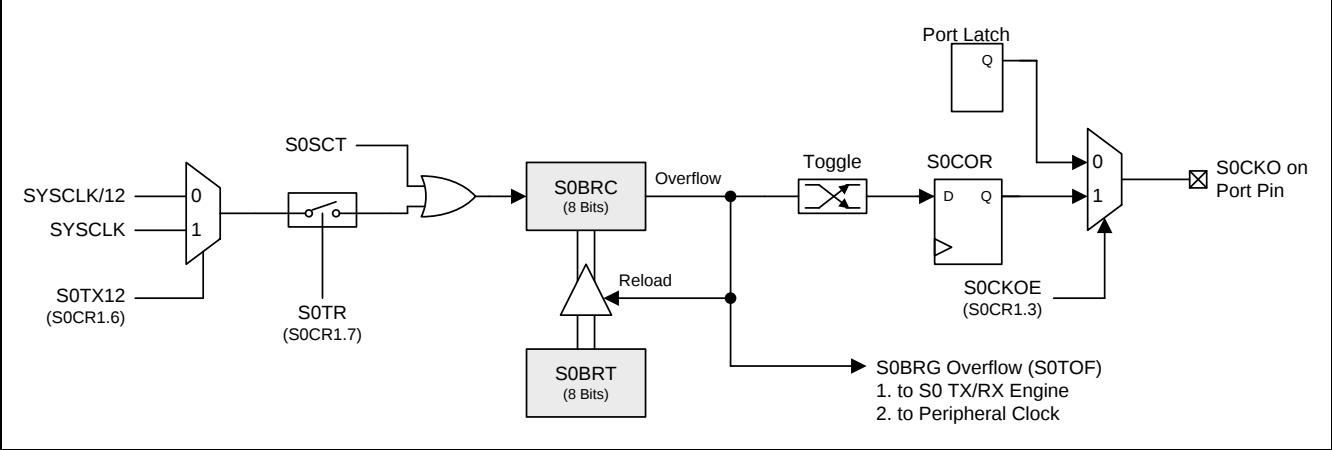


图 19-20. S0BRG 时钟输出 (S0BRG 用于 UART 模式)



AUXR6: 辅助寄存器 6

SFR 页 = 仅 3 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
KBI4PS1	KBI4PS0	KBI6PS0	KBI2PS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 0: S0COPS, S0BRG 时钟输出 (S0CKO)端口引脚选择

S0COPS	S0CKO
0	P4.7
1	P4.4

19.12. 串口 0 寄存器

串口的四种操作模式除波特率的设定之外都与标准的 8051 相同。3 个寄存器 PCON0、AUXR2 和 S0CFG 是与波特率的设定有关。

S0CON: 串口 0 控制寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0x98

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SM00/FE	SM10	SM20	REN0	TB80	RB80	T10	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: FE, 帧错误位。SMOD0 必须置位才能访问 FE 位。

0: FE 位不会被有效的帧清零, 它应当被软件清零。

1: 当检测到一个无效的停止位时, 该位被接收器置位。

Bit 7: 串口 0 模式位 0, (SMOD0 必须 = 0 才能访问位 SM00)。

Bit 6: 串口 0 模式位 1。

SM30	SM00	SM10	Mode	描述	波特率
0	0	0	0	移位寄存器	SYSClk/12 or SYSClk/4
0	0	1	1	8-位 UART	可变
0	1	0	2	9-位 UART	SYSClk/64, /32, /16, /8
0	1	1	3	9-位 UART	可变
1	0	0	4	SPI 主机	SYSClk/12 or SYSClk/4
1	0	1	5	保留	保留
1	1	0	6	保留	保留
1	1	1	7	保留	保留

Bit 5: 串口 0 模式位 2。

在模式 0 和模式 4:

Tx 初始信号电平可由 SM20 定义

0: 定义 Tx 初始信号电平为高

1: 定义 Tx 初始信号电平为低

在模式 2 和 3:

0: 禁止 SM20 功能。

1: 在模式 2 和 3 时使能地址自动识别, 如果 SM20=1 那么 RI0 将不能设置, 除非接收到的第 9 位数据(RB80)为 1, 指示是一个地址, 并且接收到的字节是本地地址或者是一个广播地址; 在模式 1, 如果 SM20=1 那么 RI0 将不能被置位除非收到一个有效的停止位, 并且接收到的字节是本地地址或者是一个广播地址。

在纯定时器模式:

与 S0TX12 一起选择时钟源

S0TX12, SM20	S0BRG 时钟选择
0 0	SYSClk/12
0 1	T10F
1 0	SYSClk
1 1	T00F

Bit 4: REN0, 使能串口接收。

0: 软件清零将禁止接收。

1: 软件置位使能接收。

Bit 3: TB80, 在模式 2 和 3 时第 9 位数据被传送, 根据需要通过软件置位或清零。

在模式 0 和模式 4, 它是 RX 数据采样沿选择

0: 采样串行时钟(TXD0)后沿的 RX 数据。

1: 采样串行时钟(TXD0)前沿的 RX 数据。

Bit 2: RB80, 在模式 2 和 3 时收到的第 9 位数据。在模式 1, 如果 SM20=0, RB80 是收到数据的停止位。在模式 0, RB80 没有使用。

Bit 1: TIO, 发送中断标志。

0: 必须由软件清零。

1: 在模式 0 时, 在第 8 位数据位时序后由硬件置位。其它模式中, 在发送停止位之初由硬件置位。

Bit 0: RIO, 接收中断标志。

0: 必须由软件清零。

1: 在模式 0 时, 在第 8 位数据位时序后由硬件置位。其它模式中(除留意 SM20 外), 在接收停止位的中间时刻由硬件置位。

S0BUF: 串口 0 缓冲寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0x99

复位值 = XXXX-XXXX

7	6	5	4	3	2	1	0
S0BUF.7	S0BUF.6	S0BUF.5	S0BUF.4	S0BUF.3	S0BUF.2	S0BUF.1	S0BUF.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 在发送和接收时作缓冲寄存器

SADDR: 从机地址寄存器

SFR 页 = 0~F

SFR 地址 = 0xA9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SADEN: 从机地址屏蔽寄存器(SMOD3 = 0)

SFR 页 = 0~F

SFR 地址 = 0xB9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SADDR 和 SADEN 组合来形成自动地址识别的要求/广播地址。事实上, SADEN 是 SADDR 的“屏蔽”寄存器。如下所示。

$$\begin{array}{rcl}
 \text{SADDR} & = & 1100\ 0000 \\
 \text{SADEN} & = & 1111\ 1101 \\
 \hline
 \text{Given} & = & 1100\ 00x\bullet
 \end{array}$$

给定的从机地址将被核对,
除了位 1 被视为“不关注”

每个从机的广播地址为 SADDR 和 SADEN 进行逻辑“或”的结果。结果中为“0”被认为“不关注”。在系统复位后, SADDR 和 SADEN 都被初始化为 0。这样生成了所有的“不关注”要求地址和所有的“不关注”广播地址。这将禁止自动地址侦测功能。

PCON0: 电源控制寄存器 0

SFR 页 = 0~F

SFR 地址 = 0x87

POR = 0001-0000

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SMOD1	SMOD0	GF	POF	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SMOD1, 双倍波特率控制位

0: 禁止 UART 双倍波特率。

1: 使能 UART 双倍波特率(模式 1、2 或 3)。

Bit 6: SMOD0, 帧错误选择。

0: S0CON.7 作 SM0 功能。

1: S0CON.7 作 FE 功能。注: 当帧错误后不管 SMOD0 什么状态 FE 都将置位。

S0CFG: 串口0 配置寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0x9C 复位值 = 0000-1000

7	6	5	4	3	2	1	0
URTS	SMOD2	0	SM30	S0DOR	BTI	UTIE	SMOD3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: URTS, UART0 定时器选择。
0: 定时器 1 或定时器 2 作为模式 1 或模式 3 的波特率发生器。
1: 当定时器 1 被选作串口 0 模式 1 或模式 3 的波特率发生器时定时器 1 溢出信号被串口 1 波特率定时器溢出信号取代。(请参考章节“19.9.4 模式 1&3 波特率.”)。

Bit 6: SMOD2, UART0 额外双倍波特率选择。
0: 禁止 UART0 额外双倍波特率。
1: 使能 UART0 额外双倍波特率。

Bit 5: 保留。写该寄存器时，此位必须写“0”。

Bit 4: SM30, 串口模式控制位 3。

Bit 3: S0DOR, 串口 0 所有操作模式的数据位序控制。
如果 S0 不是定时器模式：
0: 数据字节高位在先(MSB)传送。
1: 数据字节低位在先(LSB)传送。默认是 S0DOR 为“1”。
如果 S0 是定时器模式：
0: 设置 S0BRG 为 8 位重载定时器/计数器模式。
1: 设置 S0BRG 为 16 位定时器/计数器模式。

Bit 2: BTI, 在串口 0 中断阻止 TI0。
0: 保留 TI0 作为一个串口 0 中断源。
1: 阻止 TI0 作为一个串口 0 中断源。

Bit 1: UTIE, 在系统标志中断里使能 S0 TI0
0: 禁止在系统标志中断里中断向量共享给 TI0。
1: 设置 TI0 标志将与系统标志中断共享中断向量。

Bit 0: SMOD3, S0CR1 访问控制。
0: 禁止 S0CR1 访问。CPU 访问特殊功能寄存器地址 0xB9 为读/写 SADEN。
1: 使能 S0CR1 访问。CPU 访问特殊功能寄存器地址 0xB9 为读/写 S0CR1。

AUXR2: 辅助寄存器 2

SFR 页 = 0~F

SFR 地址 = 0xA3 复位值 = 0000-0000

7	6	5	4	3	2	1	0
STAF	STOF	C1PLK	C0PLK	T1X12	T0X12	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: T1X12, 当 C/T=0 时，定时器 1 时钟源选择。
0: 清零选择 SYSCLK/12。
1: 置位选择 SYSCLK 作时钟源。若置位，在模式 1 和模式 3 中 UART0 选择定时器 1 作波特率则速率是标准 8051 的 12 倍。

AUXR3: 辅助寄存器 3

SFR 页 = 仅 0 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: S0PS0, 串口 0 引脚选择 0. (S0PS1 在 AUXR10.3)

S0PS1~0	RXD0	TXD0
0 0	P3.0	P3.1
0 1	P4.4	P4.5
1 0	P3.1	P3.0
1 1	P1.7	P2.2

AUXR6: 辅助寄存器 6

SFR 页 = 仅 3 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
KBIHPS1	KBIHPS0	KBILPS0	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1: SnMIPS, S0MI, S1MI, S2MI & S3MI 端口引脚选择

SnMIPS	S0MI	S1MI
0	P1.6	P1.0
1	P3.3	P3.5

Bit 0: S0COPS, S0BRG 时钟输出(S0CKO) 端口引脚选择

S0COPS	S0CKO
0	P4.7
1	P4.4

20. 串口 1 (UART1)

MG82F6P32 装备有第二 UART (以后就称作 UART1)，有 5 种运行模式，模式 0 ~ 模式 4，除了下面的不同点与第一个 UART (UART0) 一样：

- (1) UART1 没有增强功能：帧错误检测和自动地址识别
- (2) UART1 使用特定的波特率定时器作为其波特率发生器(S1BRG)
- (3) UART1 使用端口 TXD1 和 RXD1 分别作为接收和发送
- (4) 波特率发生器提供切换源用于 S1CKO 和外设时钟
- (5) S1 + S1BRG 可以配置为一个带有端口变化检测的 8 位自动重载定时器

MG82F6P32 的 UART1 和 UART0 可以不同或相同模式、不同或相同通讯速率同时工作。

20.1. 串口 1 模式选择

串口 1 有 4 种标准模式和 3/4 增强模式:模式 0 提供同步通信，模式 1、2 和 3 提供异步通信。异步通信作为全双工通用异步收发器(UART)运行，可以同时以不同的波特率发送和接收。UART1 中的模式 4 支持 SPI 主操作，数据速率设置与模式 0 相同。模式 6 在发送和接收上提供 BMC ENDEC(编码器/解码器)。模式 8 是一个纯定时器功能。模式 9 支持 ARGB 发送器功能。

表 20-1. 串口 1 模式选择

S1TME	SM31	SM01, SM11	MODE	功能	波特率基准	备注
0	0	0 0	0	移位基础器	SYSCLK/12 or SYSCLK/4 (S1MOD1=1)	
0	0	0 1	1	8 位 UART	S1BRG 溢出	
0	0	1 0	2	9 位 UART	SYSCLK/64, or /32	
0	0	1 1	3	9 位 UART	S1BRG 溢出	
0	1	0 0	4	SPI 主机	SYSCLK/12 or SYSCLK/4 (S1MOD1=1)	
0	1	0 1	5	保留		
0	1	1 0	6	BMC ENDEC	S1BRG 溢出	
0	1	1 1	7	保留		
1	0	0 0	8	纯定时器	8 位定时器功能	
1	0	0 1	9	ARGB 发生器	S1BRG 溢出	
其它				保留		

20.2. 串口 1 波特率发生器(S1BRG)

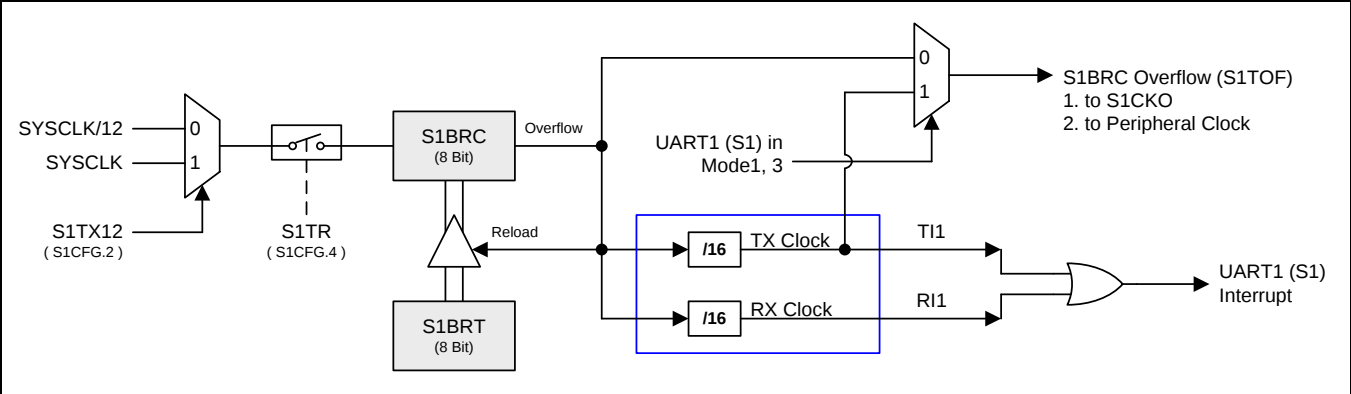
MG82F6P32 在模式 1 和模式 3 有一个嵌入式波特率发生器给串口 1 提供 UART 时钟。它由一个 8 位的向上计数器(S1BRC)和一个自动装载寄存器(S1BRT)构成。S1BRC 的溢出(S1TOF)是 UART1 在模式 1 和模式 3 串行引擎的时间基准并且触发 S1BRT 的值重载到 S1BRC 继续计数。

如果 S1TR= 0，软件写 S1BRT 数据将同时修改 S1BRC。在 S1TR 使能开启 S1BRC 计数之后，当 S1BRT 写入时不影响 S1BRC。修改 S1BRC 不会影响 S1BRT 的值。

此波特率发生器通过软件配置也能给串口 0 提供时基。这里有一个 S1BRC 溢出率的二频分(S1TOF/2)的额外时钟输出(S1CKO)。S1TOF 也供给其它外设时钟输入源切换。无论 S1 运行还是待运行，S1BRG 总是服务于这些外设的时间基准功能。

串口 1 波特率发生器结构见 图 20-1.

图 20-1. S1BRG 结构(S1TME=0)



20.3. 串口 1 波特率设定

20.3.1. 模式 0 & 4 波特率(串行时钟频率)

S1 模式 0 & 4 波特率公式: (n=1)

$$\text{Baud Rate} = \frac{F_{\text{SYSCLK}}}{12} \quad ; \text{SnMOD1}=0$$
$$\text{or} = \frac{F_{\text{SYSCLK}}}{4} \quad ; \text{SnMOD1}=1$$

20.3.2. 模式 2 波特率

S1 模式 2 波特率公式: (n=1)

$$\text{Sn Mode 2 Baud Rate} = \frac{2^{\text{SnMOD1}}}{64} \times F_{\text{SYSCLK}}$$

表 20-2. S1 模式 2 波特率@ F_{SYSCLK}=11.0592MHz

波特率	S1MOD1	Error
172800	0	0.0%
345600	1	0.0%

表 20-3. S1 模式 2 波特率@ F_{SYSCLK}=12.00MHz

波特率	S1MOD1	Error
187500	0	0.0%
375000	1	0.0%

20.3.3. 模式 1 & 3 波特率

S1 模式 1, 3 波特率公式: (n=1)

$$\text{Sn Mode 1, 3 Baud Rate} = \frac{2^{\text{SnMOD1}}}{32} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{SnBRT})} \quad ; \text{SnTX12}=0$$
$$\text{or} = \frac{2^{\text{SnMOD1}}}{32} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{SnBRT})} \quad ; \text{SnTX12}=1$$

S1MOD1	波特率	备注	建议最大接收误差 (%)
0	缺省波特率	标准功能	± 3%
1	双倍波特率	标准功能	± 3%

表 20-4 ~ 表 20-11 列出了各种常用的波特率，以及如何从串口 1 波特率发生器 S1BRG 获取波特率。

表 20-4. S1BRG 产生常用波特率@ F_{SYCLK}=11.0592MHz

10 4. S1BRT) 主用/备用板卡@ 1.535CR-11.002MHz12

S1BRT 重载值						
S1MOD1	0	1	误差	0	1	误差
Baud Rate	S1TX12=0			S1TX12=1		
1200	232	208	0.0%	--	--	--
2400	244	232	0.0%	112	--	0.0%
4800	250	244	0.0%	184	112	0.0%
9600	253	250	0.0%	220	184	0.0%
14400	254	252	0.0%	232	208	0.0%
19200	--	253	0.0%	238	220	0.0%
28800	255	254	0.0%	244	232	0.0%
38400	--	--	--	247	238	0.0%
57600	--	255	0.0%	250	244	0.0%
115200	--	--	--	253	250	0.0%
230400	--	--	--	--	253	0.0%

表 20-5. S1BRG 产生常用波特率@ F_{SYCLK}=22.1184MHz

S1BRT 重载值						
S1MOD1	0	1	误差	0	1	误差
Baud Rate	S1TX12=0			S1TX12=1		
1200	208	160	0.0%	--	--	--
2400	232	208	0.0%	--	--	0.0%
4800	244	232	0.0%	112	--	0.0%
9600	250	244	0.0%	184	112	0.0%
14400	252	248	0.0%	208	160	0.0%
19200	253	250	0.0%	220	184	0.0%
28800	254	252	0.0%	232	208	0.0%
38400	--	253	0.0%	238	220	0.0%
57600	255	254	0.0%	244	232	0.0%
115200	--	255	0.0%	250	244	0.0%
230400	--	--	--	253	250	0.0%
460.8K	--	--	--	--	253	0.0%

表 20-6. S1BRG 产生常用波特率@ F_{SYCLK}=12.0MHz

S1BRT 重载值						
S1MOD1	0	1	误差	0	1	误差
Baud Rate	S1TX12=0			S1TX12=1		
1200	230	204	0.16%	--	--	--
2400	243	230	0.16%	100	--	0.16%
4800	--	243	0.16%	178	100	0.16%
9600	--	--	--	217	178	0.16%
14400	--	--	--	230	204	0.16%
19200	--	--	--	--	217	0.16%
28800	--	--	--	243	230	0.16%
38400	--	--	--	246	236	2.34%
57600	--	--	--	--	243	0.16%

表 20-7. S1BRG 产生常用波特率@ F_{SYSLK}=24.0MHz

S1BRT 重载值						
S1MOD1	0	1	误差	0	1	误差
Baud Rate	S1TX12=0			S1TX12=1		
1200	204	152	0.16%	--	--	--
2400	230	204	0.16%	--	--	--
4800	243	230	0.16%	100	--	0.16%
9600	--	243	0.16%	178	100	0.16%
14400	--	--	--	204	152	0.16%
19200	--	--	--	217	178	0.16%
28800	--	--	--	230	204	0.16%
38400	--	--	--	--	217	0.16%
57600	--	--	--	243	230	0.16%
115200	--	--	--	--	243	0.16%

表 20-8. S1BRG 产生常用波特率@ F_{SYSLK}=29.4912MHz

S1BRT 重载值						
S1MOD1	0	1	误差	0	1	误差
Baud Rate	S1TX12=0			S1TX12=1		
1200	192	128	0.0%	--	--	--
2400	224	192	0.0%	--	--	--
4800	240	224	0.0%	64	--	0.0%
9600	248	240	0.0%	160	64	0.0%
14400	--	--	--	192	128	0.0%
19200	252	248	0.0%	208	160	0.0%
28800	--	--	--	224	192	0.0%
38400	--	--	--	232	208	0.0%
57600	--	--	--	240	224	0.0%
115200	--	--	--	248	240	0.0%
230400	--	--	--	252	248	0.0%
460.8K	--	--	--	254	252	0.0%
921.6K	--	--	--	255	254	0.0%
1.8432M	--	--	--	--	255	0.0%

表 20-9. S1BRG 产生常用波特率@ F_{SYSLK}=44.2368MHz

S1BRT 重载值						
S1MOD1	0	1	误差	0	1	误差
Baud Rate	S1TX12=0			S1TX12=1		
1200	160	64	0.0%	--	--	--
2400	208	160	0.0%	--	--	--
4800	232	208	0.0%	--	--	--
9600	244	232	0.0%	112		0.0%
14400	248	240	0.0%	160	64	0.0%
19200	250	244	0.0%	184	112	0.0%
28800	252	248	0.0%	208	160	0.0%
38400	253	250	0.0%	220	184	0.0%
57600	254	252	0.0%	232	208	0.0%
115200	255	254	0.0%	244	232	0.0%
230400	--	255	0.0%	250	244	0.0%

460.8K	--	--	--	253	250	0.0%
921.6K	--	--	--	--	253	0.0%

表 20-10. S1BRG 产生常用波特率@ F_{SYSClk}=32.0MHz

S1BRT 重载值						
S1MOD1	0	1	误差	0	1	误差
Baud Rate	S1TX12=0			S1TX12=1		
1200	187	118	0.64%	--	--	--
2400	221	186	0.79%	--	--	--
4800	239	222	2.12%	48	--	0.16%
9600	--	239	2.12%	152	48	0.16%
14400	--	--	--	187	118	0.64%
19200	--	--	--	204	152	0.16%
28800	--	--	--	221	186	-0.79%
38400	--	--	--	230	204	0.16%
57600	--	--	--	239	222	2.12%
115200	--	--	--	--	239	2.12%

表 20-11. S1BRG 产生常用波特率@ F_{SYSClk}=48.0MHz

10 11. S1BRT) 主用侧波特率@15%SLCK=48.0MHz

S1BRT 重载值						
S1MOD1	0	1	误差	0	1	误差
Baud Rate	S1TX12=0			S1TX12=1		
1200	152	48	0.16%	--	--	--
2400	204	152	0.16%	--	--	--
4800	230	204	0.16%	--	--	--
9600	243	230	0.16%	100	--	0.16%
14400	--	239	2.12%	152	48	0.16%
19200	--	243	0.16%	178	100	0.16%
28800	--	--	--	204	152	0.16%
38400	--	--	--	217	178	0.16%
57600	--	--	--	230	204	0.16%
115200	--	--	--	243	230	0.16%
230400	--	--	--	--	243	0.16%

20.4. 串口 1 模式 4 (SPI 主机)

MG82F6P32 的串口 1 内嵌 Mode 4，支持 SPI 主引擎。SM31、SM01、SM11 选择“模式 4”。MG82F6P32 的串口模式定义如表 20-12 所示。

表 20-12. 串口 1 模式选择

SM31	SM01	SM11	模式	描述	波特率
0	0	0	0	移位寄存器	SYSClk/12 or SYSClk/4
0	0	1	1	8 位 t UART	可变
0	1	0	2	9 位 UART	SYSClk/64, or /32
0	1	1	3	9 位 UART	可变
1	0	0	4	SPI Master	SYSClk/12 or SYSClk/4
1	0	1	5	保留	可变
1	1	0	6	BMC endec	可变
1	1	1	7	保留	可变

S1MOD1 还控制 SPI 传输速度。当 S1MOD1 = 1 时，SPI 时钟频率为 SYSClk/4。否则，SPI 时钟频率为 SYSClk/12。当选择 S1BRG 作为时钟源时，SPI 时钟频率不受 S1MOD1 控制，等于 S1TOF/4。

MG82F6P32 中的 SPI 主控使用 TXD1 作为 SPICLK, RXD1 作为 MOSI, S1MI 作为 MISO。nSS 由单片机软件在其他端口引脚上选择。SPI 连接如图 20-2 所示。支持如图 20-3 所示的多从机通信架构。

图 20-2. 串口 1 模式 4, 单主机和单从机架构(n = 1)

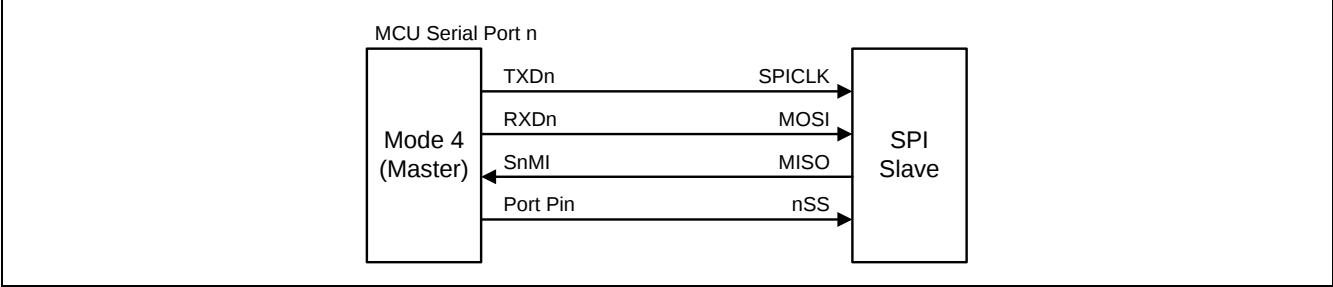
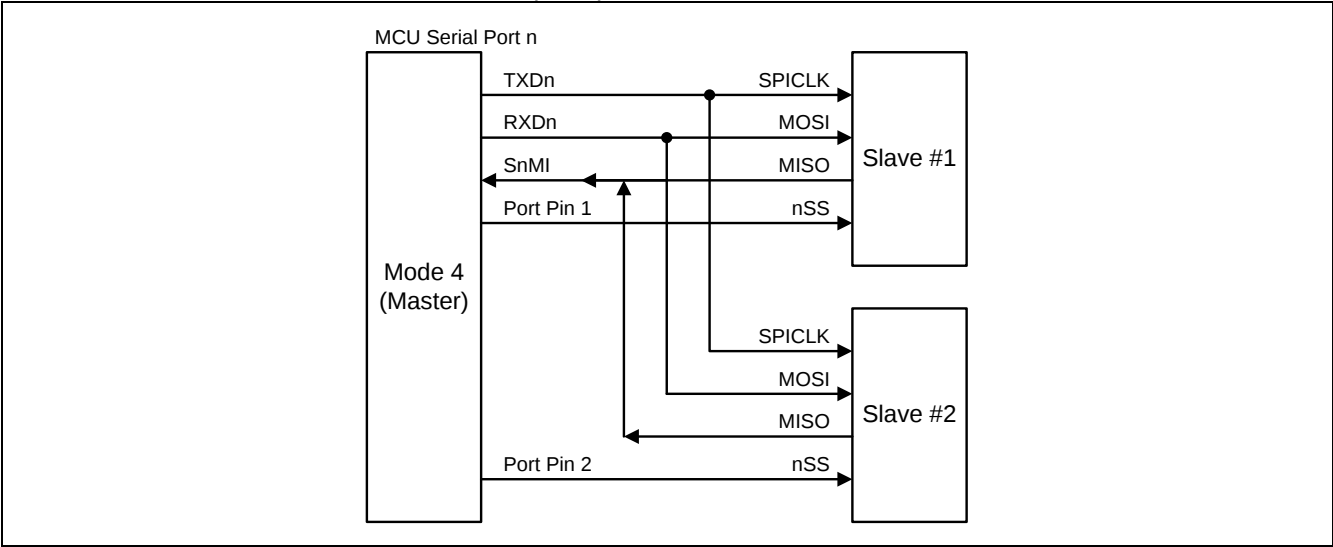


图 20-3. 串口 1 模式 4, 单主机和多从机架构 (n = 1)



SPI 主机能满足笙泉 MA82/84 系列 MCU 全功能 SPI 模块的传输，具有 CPOL, CPHA 和 DORD 选择。对于 CPOL 和 CPHA 条件，**MG82F6P32** 使用一种简单的方法，通过初始化 SPI 时钟极性来匹配它们。串行口模式 4 的 4 个 SPI 工作模式如表 20-13 所示。

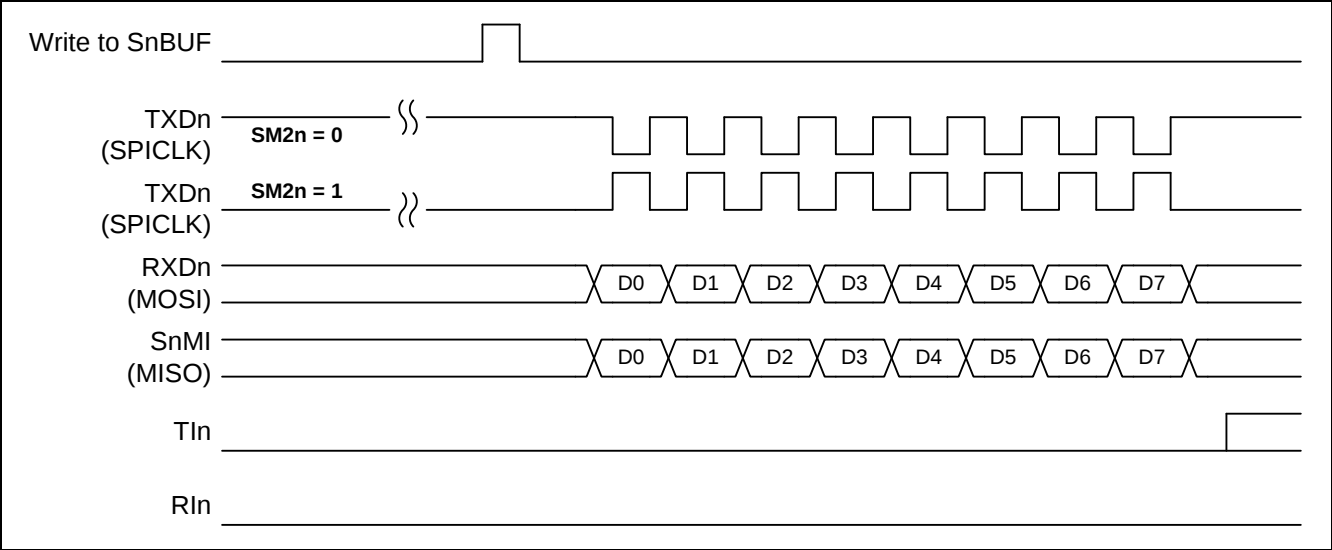
表 20-13. 串口 1 模式 4 的 SPI 模式配置

SPI 模式	CPOL	CPHA	在 SnCON 中配置 SM2n & TB8n
0	0	0	SM2n = 1, TB8n = 1
1	0	1	SM2n = 1, TB8n = 0
2	1	0	SM2n = 0, TB8n = 1
3	1	1	SM2n = 0, TB8n = 0

SPI 系列传输的位序控制(DORD)，**MG82F6P32** 提供了位序控制(S1DOR)。S1DOR 的默认值是“1”且位序控制为低位在先(LSB)。

由任何指令把 S1BUF 作为目的寄存器的使用初始化发送。“写到 S1BUF”信号触发 UART 引擎开始发送。S1BUF 的数据作为 MOSI 串行数据被移位到 RXD1 引脚。SPI 移位时钟在 TXD1 引脚上作为 SPICLK 输出。8 个移位时钟上升沿之后，硬件生效 TI1 表示发送结束。同时 S1MI 引脚也被采样且移位到移位寄存器。然后，“读取 S1BUF”会获得 SPI 的移入数据。模式 0 的发送波形如图 20-4 所示。模式 4 中 RI1 不会生效。

图 20-4. 串口 1 模式 4 发送波形 (n =1)



20.5. S1BRG 定时器模式(模式 8)

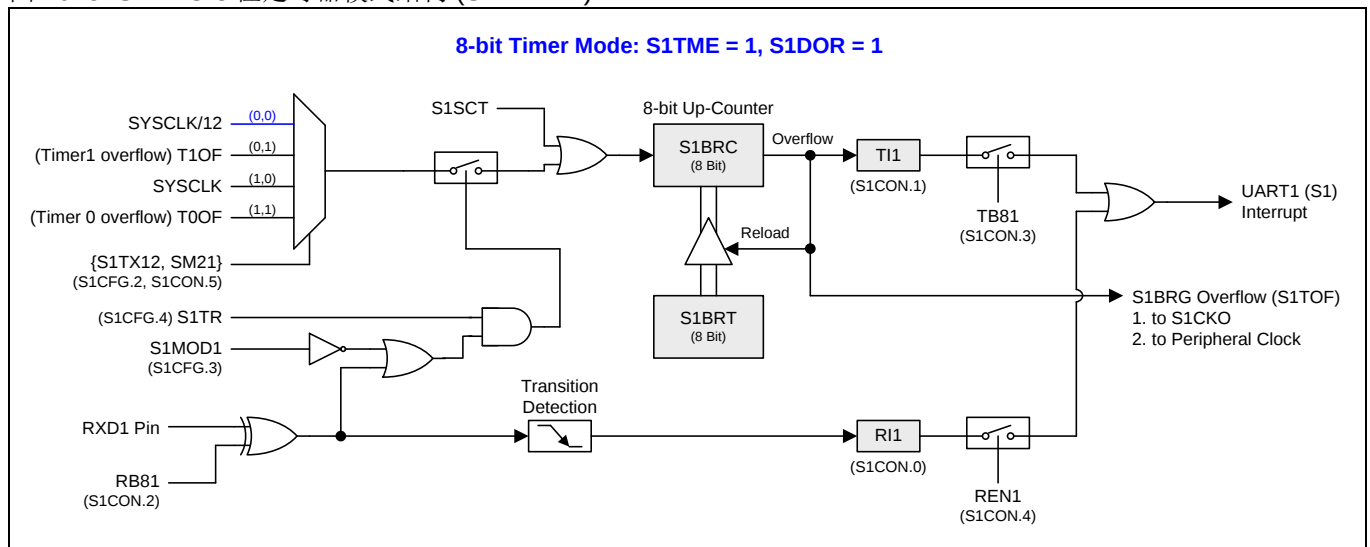
20.5.1. 8 位定时器模式

如果在应用中不需要 UART1 或软件挂起，则在 **MG82F6P32** 中设置 S1TME=1 提供 S1 波特率发生器(S1BRG)的纯定时器工作模式。这个定时器作为一个 8 位自动重载定时器运行，并提供在 TI1 (S1CON.1)上置位的溢出标志。RI1 (S1CON.0)服务于 RXD1 端口引脚上的端口变化检测器。该模式下的 TI1 和 RI1 都保持 UART1 中断资源的中断能力，并具有单独的中断启用控制(TB81 和 REN1)。RB81 选择 RXD1 端口输入的 RI1 检测电平。如果 RB81=0，则通过 REN1=1 和 RXD1 引脚下降沿检测来置位 RI1。否则，RI1 将检测 RXD1 端口引脚上的上升沿。在 MCU 掉电模式下，RI1 被强制电平敏感操作，并且如果 UART1 中断被启用，则具有唤醒 CPU 的能力。

此纯定时器模式有一个来自定时器 1 溢出的时钟输入选项，是一个级联计数器工作在一个 16 位定时器。当 S1BRC 溢出，它将是 UART0 的时钟源或切换端口引脚输出。S1CKOE=1 使能 S1CKO 输出在端口引脚并且屏蔽 RI1 中断。

S1BRG 8 位定时器模式结构如图 20-5 所示

图 20-5. S1BRG 8 位定时器模式结构 (S1TME=1)



20.6. BMC ENDEC (模式 6)

- 支持 8 位 BMC 数据 ENDEC(编码器/解码器)，编码器在 TXD1 上输出，解码器从 RXD1 输入
- 由 S1BRG 可编程比特率 (S1 波特率发生器)
- 特殊的 5 位 BMC 模式，适应 USB PD 格式
- 可配置 PDTXOE，使片外驱动 TXD1 数据

初始化 S1 到模式 6 USB PD 工作模式

- 置位{S1TME, SM31, SM01, SM11}工作模式 6, BMC ENDEC.
- 写 S1BRC/S1BRT SFR & S1TX12/S1MOD1 控制位配置 S1 波特率
- 置位 M6PDE1 为 5 位 BMC ENDEC 选择 USB PD
- 置位 REN 和 S1TR 启动 S1 TX/RX 引擎

发送一个 PD 数据帧的流程

- 写 0xAA (前导码)到 S1BUF
- 等待 TI1, 然后再次写 0xAA 到 S1BUF..... 总共发送 8 次 0xAA 到 S1BUF.
- 等待 TI1,然后发送 5 位 K-code..... 重复发送 4 次 K-code
- 如果需要,等待 TI1 发送 5 位 Data0~DataF (包括 CRC32 值)
- 等待 TI1, 然后发送 5 位 EOP K-code.
- 等待 TI1, 然后写 0xFF 到 S1BUF 结束发送

接收一个 PD 数据帧的流程

- 等待 RI 获取在 S1BUF 中的 1st 数据.如果数据位 7~5 = 100, 它的 5 位数据是非 4b5b 的 Data0~DataF.its 5 bit data is non Data0~DataF of 4b5b. 如果数据位 7~5 = 000, 它的 5 位数据是 4b5b 的 Data0~DataF (包括 CRC32 值)
- 等待 RI 获取在 S1BUF 中的 2nd 数据., 并对数据位 7~5 执行相同的检查。
-
- 等待 RI,如果数据为 0xFF, 则表示接收过程结束.

20.6.1. 模式 6 波特率

S1 模式 6 波特率公式: (n=1)

$$\text{Sn Mode 6 Baud Rate} = \frac{2^{\text{SnMOD1}}}{32} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{SnBRT})} ; \text{SnTX12}=0$$

$$\text{or} = \frac{2^{\text{SnMOD1}}}{32} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{SnBRT})} ; \text{SnTX12}=1$$

注意:

(1) 对于 SYSCLK=24MHz & S1TX12=1 & S1MOD1=1, S1BRT=0xFB 将为模式 6 产生 300K 波特率

S1CFG6: 串口 1 配置寄存器 6

SFR 页 = 1 和 2

SFR 地址 = 0x9D & S1 在模式 6

复位值 = 0001-0000

7	6	5	4	3	2	1	0
PDOEC1	M6TGU1	M6TXB1	M6IDL1	M6TUR1	M6ROR1	PDOE1	M6PDE1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

在 SFR 页面 1 或 2 访问 SFR 地址 0x9D 上的 S1CFG6 之前，软件必须将 S1 模块设置为工作模式 6。

Bit 7: PDOEC, PDOE 输出控制

0:在 PDTXOE 输出上选择低电平有效.

1: 在 PDTXOE1 输出上选择高电平有效

Bit 6: M6TGU1, S1 模式 6 下 TX 放弃标志

0: 必须软件清零

1: 由硬件设置，表示由于引擎正在进行 BMC 数据接收，因此放弃数据传输。

Bit 5: M6TXB1, S1 模式 6 下 TX (BMC 数据传送)忙状态. 仅读

0: TX 引擎不忙

1: TX 引擎忙于 S1 模式 6 的数据传输。

Bit 4: M6IDL1. BMC 总线在接收时处于 IDLE 状态。仅读

0: 引擎在 S1 模式 6 下接收 RXD1。

1: 总线在 RXD1 上空闲。

Bit 3: M6TUR1, TX 欠运行标志.

0: 必须软件清零

1: 由硬件设置，表示在 S1 模式 6 下发送最后一位数据，并且 S1 TX 缓冲区为空。

Bit 2: M6ROR1, RX 溢出标志

0: 必须软件清零

1: 由硬件设置，表示在 S1 模式 6 中接收的数据溢出。

Bit 1: PDOE1, DTXOE 输出使能，控制外部缓冲区驱动 USB PD 总线上的 TXD1。

0: 关闭端口引脚的 PDTXOE 功能

1: 使能端口引脚的 PDTXOE 功能

Bit 0: M6PDE1, USB PD BMC 功能在 S1 模式 6 下使能控制

0: 在 S1 模式 6 下关闭 USB PD BMC 功能。(8-bit BMC ENDEC)

1: 在 S1 模式 6 下使能 USB PD BMC 功能。(5-bit BMC ENDEC)

S1CFG: 串口 1 配置寄存器

SFR 页 = 1 和 2

SFR 地址 = 0x9C

复位值 = 0010-0000

7	6	5	4	3	2	1	0
SM31	S1MOD3	S1DOR	S1TR	S1MOD1	S1TX12	S1CKOE	S1TME
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 6: S1MOD3, S1MOD3 使能 S1CR1 能够访问 S1BRC 地址

S1CR1: 串口 1 控制寄存器 2

SFR 页 = 1 和 2

SFR 地址 = 0x9B & S1MOD3=1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
M6RER1	TXDO1	SOPWEC	TOTXD1	RMRI1	EXTFLT	TXINV1	RXINV1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

在 SFR 页面 1 或 2 访问 SFR 地址 0x9B 上的 S1CR1 之前，软件必须设置 S1MOD3=1。如果 S1MOD3=0，软件访问 SFR 地址 0x9B, SFR 页面 1 或 2 将是 S1BRC

Bit 7: M6RER1, BMC 接收数据错误

0: 必须软件清零

1: 由硬件设置，指示接收数据时 BMC 解码器错误

Bit 6: TXDO1, S1 模块 TXD1 信号输出状态。仅读。

0: TXD1 信号输出逻辑低电平

1: TXD1 信号输出逻辑高电平

Bit 5: SOPWEC, 1st SOP 包解码包括 E-Code.

0: 禁止 SOPWEC 功能

1: 使能 SOPWEC 功能

Bit 4: TOTXD1, 切换 TXD1 输出信号. 软件读总是“0”.

0: 在这位上软件写“0”无效

1: 软件在这个位上写“1”将切换 TXD1 信号输出状态。

Bit 3: RMRI1, 释放屏蔽 RI1 功能

0: 在 PD TX 模式下 RI1 功能被屏蔽

1: 在 PD TX 模式下使能 RI1 功能。

Bit 2: EXTFLT, 扩展输入滤波控制. (增加 TB81 以支持 S1 模式 6 下的 EXTFLT1 功能)

0: 禁止 EXTFLT 功能

1: 使能 EXTFLT 功能

Bit 1: TXINV, TXD1 输出反相

0: 禁止 TXD1 输出信号反相

1: 使能 TXD1 输出信号反相

Bit 0: RXINV, RXD1 输入反相.

0: 禁止 RXD1 输入信号反相

1: 使能 RXD1 输入信号反相

AUXR12: 辅助寄存器 12

SFR 页 = 仅 9 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CRCDS2	CRCM0	PDOES1	PDOES0	EDCM0	GPLC0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: CRCDS2. CRC0 数据端口选择位 2.

Bit 6: CRCM0, CRC 模式选择 0.

0: 选择 CRC16: 0x1021.

1: 选择 CRC32: 0x04C1_1DB7.

Bit 5~4: PDOES1~0, PDTXOE1 端口引脚选择位 [1:0].

PDOES1~0	PDTXOE
0 0	P1.5
0 1	P3.3
1 0	P6.1
1 1	P3.2

Bit 3: EDCM0. EDC45 模式选择 0.

EDCM0	模式 #	EDC45 操作
0	0	4 位到 5 位编码
1	1	5 位到 4 位解码

Bit 2: GPLC0. GPL 控制 0.

0: CPU 在 SFR 地址 0xA9 访问 SADDR.

1: CPU 在 SFR 地址 0xA9 访问 EDC45.

Bit 1~0: 保留。写该寄存器时，此位必须写“0”。

20.7. ARGB 发送器

- S1 设置为增强模式时， ARGB 数据在 TXD1 上输出。A
- 由 S1BRG 可编程比特率 (S1 波特率发生器)
- 可选占空比 1/4 或 1/3
- 可选择总线复位时间检测 60us, 260us, 200us 和 320us.

初始化 S1 为 ARGB 工作模式

- 置位{S1TME, SM31, SM01, SM11} 为 ARGB 模式
- 写 S1BRC/S1BRT SFR & S1TX12/S1MOD1 控制位配置 ARGB 波特率
- 定义 S0DTY1 去配置 ARGB 占空比为 1/4 或 1/3.
- 在 S0IDT11 和 S0IDT01 上选择总线复位时间检测持续时间

传输 ARGB 数据帧的流程

- 等待 RI1(S1AGBRF)总线复位事件
- 清 RI1,然后开始写 ARGB 数据到 S1BUF
- 等待 TI1(S1AGBF),S1 TX 缓冲区空.
- 清 TI2, 然后写下下一个 ARGB 数据到 S2BUF.....等等, 直到所有 ARGB 数据发送完成
- 等待 RI0 开始一个新的帧来传输 ARGB 数据

20.7.1. S1 ARGB 波特率

图 20–6. S1 ARGB 模式波特率公式: (n=1)

$$\text{Sn ARGB Baud Rate} = \frac{1}{m} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{SnBRT})} ; \text{SnTX12}=0 \begin{array}{l} ; m=4, \text{ if SnDTY1} = 0 \\ ; m=3, \text{ if SnDTY1} = 1 \end{array}$$
$$\text{or} = \frac{1}{m} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{SnBRT})} ; \text{SnTX12}=1 \begin{array}{l} ; m=4, \text{ if SnDTY1} = 0 \\ ; m=3, \text{ if SnDTY1} = 1 \end{array}$$

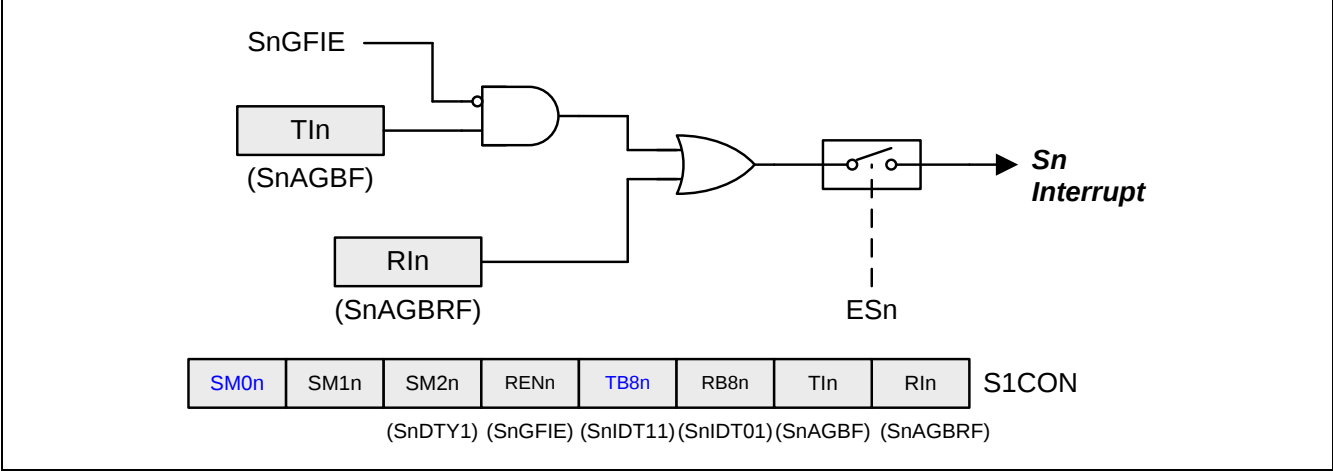
20.7.2. S1 ARGB 总线复位时间

图 20–7. S1 ARGB 模式总线复位时间 (n=1)

$$\text{Sn ARGB Reset Time} = \frac{8 \times m}{\text{Sn ARGB Baud Rate}} ; m=6/20/26/32, \text{ indexed by SnIDT1n~0n}$$

20.7.3. S1 ARGB 中断

图 20-8. 模式 9 (ARGB) 中断(n=1)



20.7.4. S1 ARGB 寄存器

以下 UART 特殊功能寄存器被重新定义为 ARGB 操作

S1CON/ S1AGCR: 串口 1 ARGB 控制寄存器

SFR 页 = 1 和 2

SFR 地址 = 0x98

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SM01	SM11	SM21	REN1	TB81	RB81	TI1	RI1
SM01	SM11	S1DTY1	S1GFIE	S1IDT11	S1IDT01	S1AGBF	S1AGBRF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: S1DTY1, ARGB 信号占空比选择

0: 选择 ARGB 信号占空比为 3/4(H)或 1/4(L).

1: 选择 ARGB 信号占空比为 2/3(H)或 1/3(L).

Bit 4: S1GFIE, 忽略 TI1 中断

0: 使能 TI1 中断。默认是使能的

1: 禁止 TI1 中断

Bit 3~2: S1IDT11~01, 选择 ARGB 总线复位时间持续时间

S1IDT11~01	ARGB 总线复位时间 (当波特率 800K)
0 0	260us
0 1	200us
1 0	60us
1 1	320us

Bit 1: S1AGBF, 标志位报告发送保持寄存器在 ARGB 模式下完成向输出移位寄存器的传输。

0: 软件写“0”清零 S1AGBF.

1: 当 S1BUF 为空时, 置位 S1AGBF, 如果使能 S1 中断, 则产生中断。

Bit 0: S1AGBRF, ARGB 模式下 ARGB 总线复位标志位.

0: 软件写“0”清零 S1AGBRF

1: 当总线满足 ARGB 复位时间时, 此位被置为逻辑 1。复位时间由 S1IDT1~0 定义。

20.8. S1BRT 可编程时钟输出

当 S1BRC 溢出时，溢出标志 S1TOF 为 S1CKO 和外围时钟提供切换源。输入时钟(SYSCLK/12 或 SYSCLK)作为 8 位定时器(S1BRC)时钟源。定时器重复计数到被载入值溢出。一旦溢出，S1BRC 载入 S1BRT 的值继续计数。
图 20-9 和图 20-10 展示了串口 1 波特率产生器的时钟输出模式方框图。时钟输出频率公式如下：

S1T Clock-out Frequency =
$$\frac{\text{SYSCLK Frequency}}{n \times (256 - \text{S1BRT})}$$

;

n=24, if S1TX12=0

;

n=2, if S1TX12=1

注意:
(1) 如 SYSCLK=12MHz 和 S1TX12=0, S1BRG 可编程输出频率范围从 1.95KHz 到 500KHz。
(2) 如 SYSCLK=12MHz 和 S1TX12=1, S1BRG 可编程输出频率范围从 23.43KHz 到 6MHz。

图 20-9. S1BRG 时钟输出(S1BRG 在 8 位定时器模式)

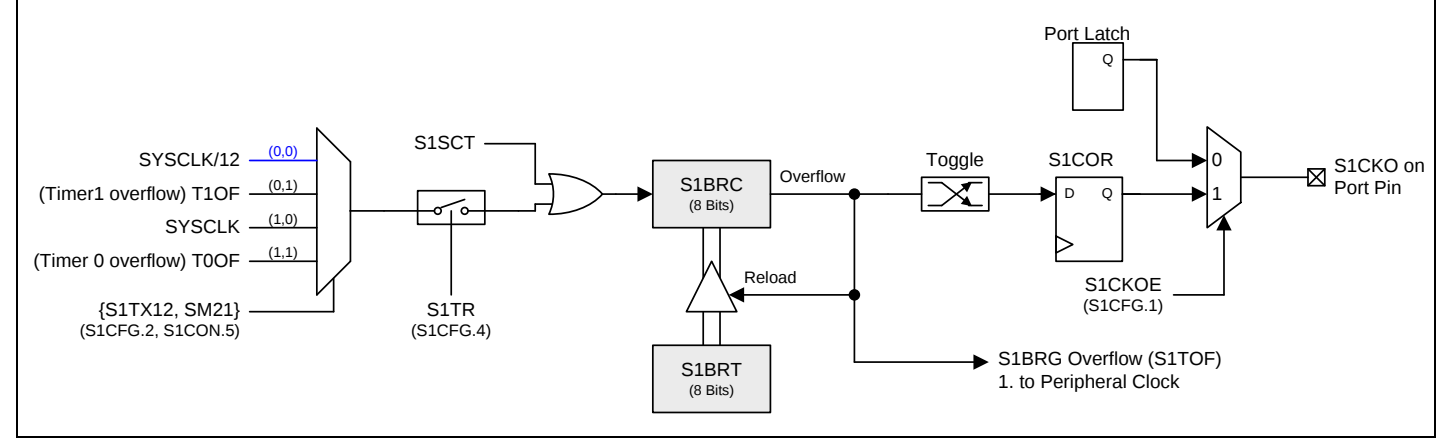
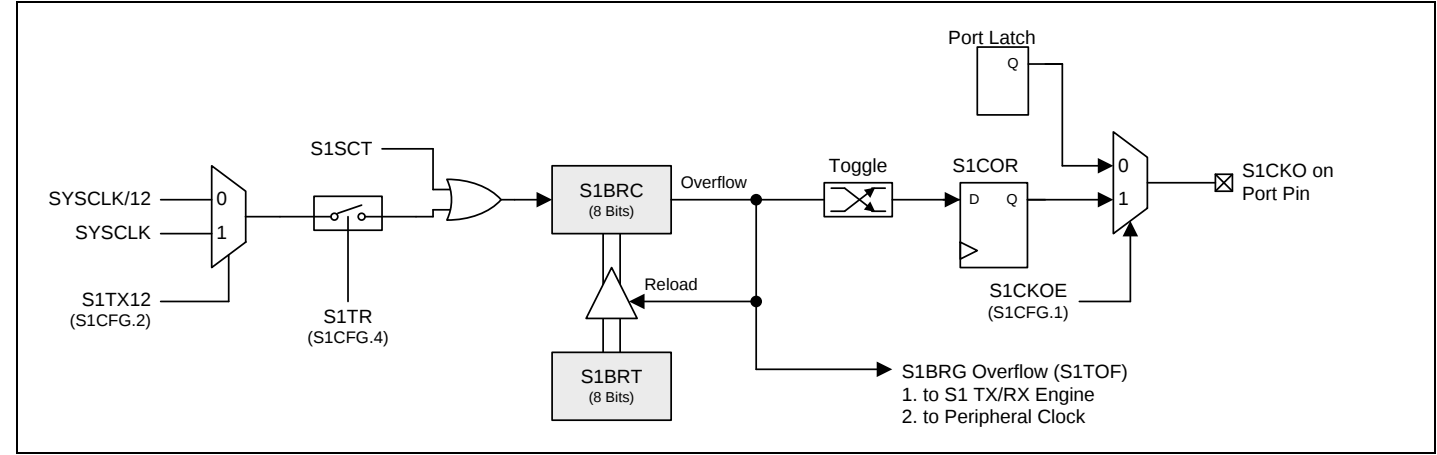


图 20-10. S1BRG 时钟输出(S1BRG 用于 UART 模式)



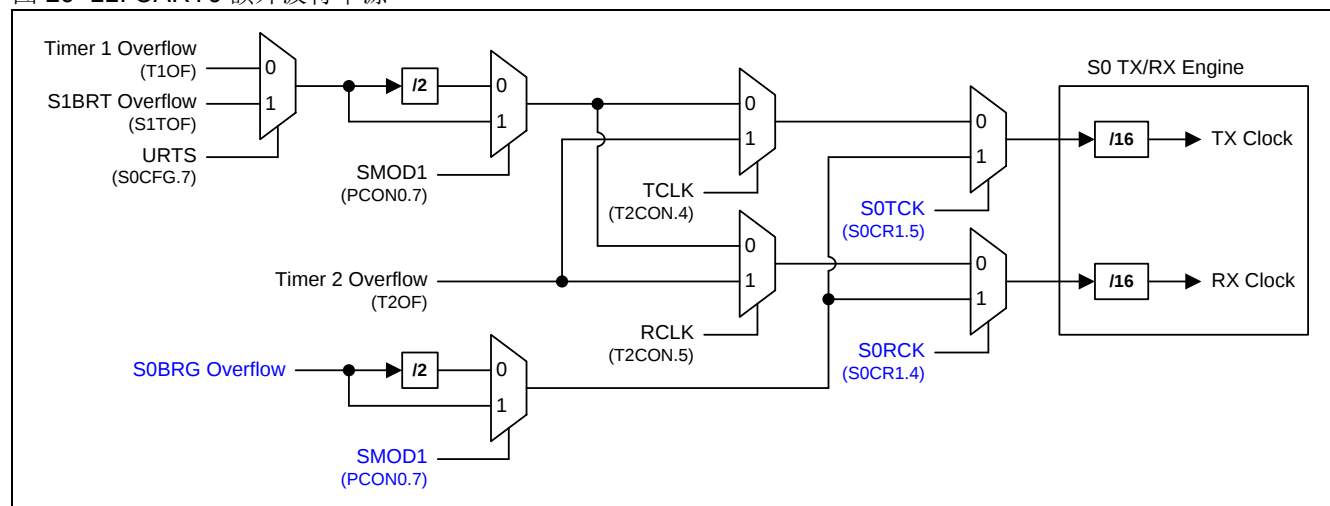
在时钟输出模式如何编程 8 位 S1BRG

- 选择 S1CFG.S1TX12 位和 S1CON.SM21 位确定 S1BRG 时钟源
- 从公式中确定 8 位重载值，并将其输入到 S1BRT 和 S1BRC 寄存器中。
- 置位 S1CFG 的 S1CKOE 位
- 置位 S1TR 去启动 S1BRC 定时器。

20.9. S1 波特率发生器用于 S0

串口 0(UART0)的模式 1 和 3 操作，软件通过 T2CON 寄存器的位 TCLK 和 RCLK 清零可以选择定时器 1 作为波特率发生器。同时，如果 URTS(S0CFG.7)置位，定时器 1 溢出信号被 UART1 波特率发生器(S1BRG)溢出信号所替代。换句话说，只要 RCLK=0, TCLK=0 和 URTS=1 用户可以采用 UART1 波特率发生器作(S1BRG)为 UART0 模式 1 或 3 的波特率发生器。这种情况下，定时器 1 也可以用作其他应用。当然，如果 UART1 的模式 1 或 3 也同时操作，则两个 UART 具有相同的波特率。

图 20-11. UART0 额外波特率源



当 S1BRG 用作串口 0(S0)波特率发生器时，波特率计算公式如下

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}} \times 2^{(\text{SMOD2} \times 2)}}{32} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{S1BRT})} ; \text{S1TX12}=0$$

$$\text{or} = \frac{2^{\text{SMOD1}} \times 2^{(\text{SMOD2} \times 2)}}{32} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{S1BRT})} ; \text{S1TX12}=1$$

注意:

详细的波特率设置请参考“19.9.4.1 使用定时器 1 和 SnBRG 作为波特率发生器”节

20.10. 串口 1 寄存器

下面的特殊功能寄存器与 UART1 的操作有关:

S1CON: 串口 1 控制寄存器

SFR 页 = 1 和 2

SFR 地址 = 0x98

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SM01	SM11	SM21	REN1	TB81	RB81	TI1	RI1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SM01, 串口 1 模式位 0

Bit 6: SM11, 串口 1 模式位 1.

SM31	SM01	SM11	模式	描述	波特率
0	0	0	0	移位寄存器	SYSCLK/12 或 SYSCLK/4
0	0	1	1	8 位 UART	可变
0	1	0	2	9 位 UART	SYSCLK/64,或 /32
0	1	1	3	9 位 UART	可变
1	0	0	4	SPI 主机	SYSCLK/12 或 SYSCLK/4
1	0	1	5	保留	保留
1	1	0	6	BMC endec	可变
1	1	1	7	保留	保留

Bit 5: 串口 1 模式位 2。

在模式 0 和模式 4:

Tx 初始信号电平可以由 SM21 定义

0: 定义 Tx 初始信号电平为高

1: 定义 Tx 初始信号电平为低

在模式 2 和 3:

0: 禁止 SM21 功能。

1: 在模式 2 和 3 时使能地址自动识别, 如果 SM21=1 那么 RI1 将不能设置, 除非接收到的第 9 位数据(RB81)为 1, 指示是一个地址, 并且接收到的字节是本机地址或者是一个广播地址; 在模式 1, 如果 SM21=1 那么 RI1 将不能被激活除非收到一个有效的停止位, 并且接收到的字节是本机地址或者是一个广播地址。

在纯定时器模式(模式 8)

与 S1TX12 一起选择时钟源.

S1TX12, SM21	S1BRG 时钟选择
0 0	SYSCLK/12
0 1	T1OF
1 0	SYSCLK
1 1	T0OF

Bit 4: REN1, 使能串行接收。

0: 软件清零将禁止接收。

1: 软件置位使能接收。

Bit 3: TB81, 在模式 2 和 3 时第 9 位数据被传送, 根据需要通过软件置位或清零。

模式 0 和模式 4 为 RX 数据采用沿的选择。

0: 采样串行时钟(TXD1)后沿的 RX 数据。

1: 采样串行时钟(TXD1)前沿的 RX 数据。

Bit 2: RB81, 在模式 2 和 3 时收到的第 9 位数据。在模式 1, 如果 SM21=0, RB81 是收到数据的停止位。在模式 0, RB81 没有使用。

Bit 1: TI1, 发送中断标志。

0: 必须由软件清零。

1: 在模式 0 时, 在第 8 位个数据位时序后由硬件置位。其它模式中, 在发送停止位之初由硬件置位。

Bit 0: RI1, 接收中断标志。

0: 必须由软件清零。

1: 在模式 0 时, 在第 8 位数据位时序后由硬件置位。其它模式中(除留意 SM21 外), 在接收停止位的中间时刻由硬件置位。

S1BUF: 串口 1 缓冲寄存器

SFR 页 = 1 和 2

SFR 地址 = 0x99

复位值 = XXXX-XXXX

7	6	5	4	3	2	1	0
S1BUF.7	S1BUF.6	S1BUF.5	S1BUF.4	S1BUF.3	S1BUF.2	S1BUF.1	S1BUF.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 在发送和接收时作缓冲寄存器

S1BRT: 串口 1 波特率定时器重载寄存器

SFR 页 = 1 和 2

SFR 地址 = 0x9A

复位值 = 0000-0000

7	6	5	4	3	2	1	0
S1BRT.7	S1BRT.6	S1BRT.5	S1BRT.4	S1BRT.3	S1BRT.2	S1BRT.1	S1BRT.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 它用于波特率定时器发生器重载变量, 工作类似于定时器 1。

S1BRC: 串口 1 波特率计数寄存器

SFR 页 = 1 和 2

SFR 地址 = 0x9B

复位值 = 0000-0000

7	6	5	4	3	2	1	0
S1BRC.7	S1BRC.6	S1BRC.5	S1BRC.4	S1BRC.3	S1BRC.2	S1BRC.1	S1BRC.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 波特率计数器寄存器与定时器 1 的工作方式相似。此寄存器可由软件读/写。如果 S1CFG1.S1TME = 0, 软件写数据到 S1BRT 同时数据存入 S1BRT 和 S1BRC。

S1CFG: 串口 1 配置寄存器

SFR 页 = 1 和 2

SFR 地址 = 0x9C

复位值 = 0010-0000

7	6	5	4	3	2	1	0
SM31	S1MOD3	S1DOR	S1TR	S1MOD1	S1TX12	S1CKOE	S1TME
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SM31, 串口 1 模式控制位 3.

SM31	SM01	SM11	模式	描述	波特率
0	0	0	0	移位寄存器	SYSCLK/12 or SYSCLK/4
0	0	1	1	8 位 UART	可变
0	1	0	2	9 位 t UART	SYSCLK/64, or /32
0	1	1	3	9 位 UART	可变
1	0	0	4	SPI 主机	SYSCLK/12 or SYSCLK/4
1	0	1	5	保留	保留
1	1	0	6	BMC endec	可变
1	1	1	7	保留	保留

Bit 5: S1DOR, 串口 1 所有操作模式的数据位序控制。

如果 S1TME = 0:

0: 数据字节高位在先(MSB)传送。

1: 数据字节低位在先(LSB)传送。默认是 S1DOR 为“1”。

如果 S1TME = 1:

0: 设置 S1BRG 为 8 位重载定时器/计数器模式。

1: 设置 S1BRG 为 16 位定时器/计数器模式。

Bit 6: S1MOD3, S1MOD3 使能在 S1BRC 地址上访问 S1CR1

Bit 4: S1TR, UART1 波特率发生器控制位.

0: 清零关闭 S1BRG。

1: 置位开启 S1BRG。

Bit 3: S1MOD1, UART1 双倍波特率选择使能位。

0: 禁止 UART1 双倍波特率功能。

1: 使能 UART1 双倍波特率功能。

S1 在模式 0 和模式 4:

0: 清零选择 SYSClk/12 作为 S1 模式 0 和模式 4 的波特率

1: 置位选择 SYSClk/4 作为 S1 模式 0 和模式 4 的波特率

Bit 2: S1TX12, UART1 波特率时钟源选择

与 SM21 一起选择时钟源.

S1TX12, SM21	S1BRG 时钟选择
0 0	SYSClk/12
0 1	T1OF
1 0	SYSClk
1 1	T0OF

Bit 1: S1CKOE, 串口 1 波特率发生器时钟输出使能。

0: 禁止 S1CKO 在端口引脚输出。

1: 使能 S1CKO 在端口引脚输出。

Bit 0: S1TME, 串口 1 波特率(BRG)定时器模式使能。

0: 保持 S1BRG 服务串口 1 (UART1)。

1: 禁止串口 1 功能并且 S1BRG 作为一个 8 位自动装载的定时器。这个模式下, 这是一个 RXD1 端口引脚变化检测器的额外功能。

AUXR9: 辅助寄存器 9

SFR 页 = 仅 6 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G0	T0G1	0	0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1~0: S1PS1~0, 串口 1 引脚选择 [1:0].

S1PS1~0	RXD1	TXD1
0 0	P1.2	P1.3
0 1	P6.0	P6.1
1 0	P1.0	P1.1
1 1	P3.4	P3.5

AUXR11: 辅助寄存器 11

SFR 页 = 仅 8 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	I2C1PS1	I2C1PS0	RX1S0	0	T1CKOE	T0CKOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: RX1S0, RXD1 选择 0.

0: RXD1 由 S1PS1~0 (AUXR9.1~0)选择.

1: 强制 RXD1 选择在 AC0OUT.

AUXR6: 辅助寄存器 6

SFR 页 = 仅 3 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
KBIHPS1	KBIHPS0	KBILPS0	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1: SnMIPS, S0MI, S1MI, S2MI & S3MI 端口引脚选择

SnMIPS	S0MI	S1MI
0	P1.6	P1.0
1	P3.3	P3.5

AUXR8: 辅助寄存器 8

SFR 页 = 仅 5 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

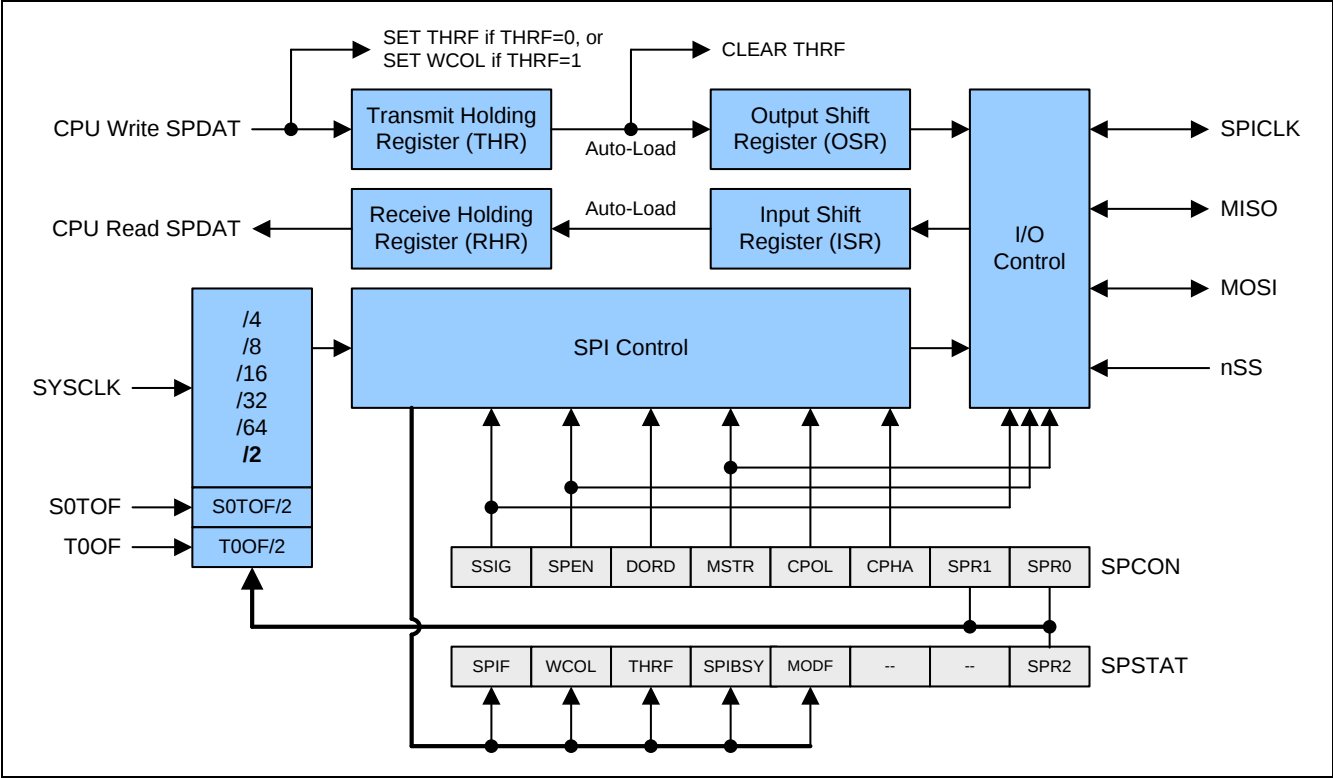
Bit 2: S1COPS, S1BRG 时钟输出(S1CKO) 端口引脚选择

S1COPS	S1CKO
0	P4.7
1	P4.5

21. 串行外设接口 (SPI)

MG82F6P32 提供了一个高速串行外设接口(SPI)。SPI 接口是一种全双工、高速同步通讯总线，有两种操作模式：主机模式和从机模式。在 48MHz 的系统时钟下主机模式支持高达 **24Mbps** 速率或从机模式高达 **12Mbps**。在 SPI 状态寄存器(SPSTAT)里有三个标志传送完成标志(SPIF)，写冲突标志(WCOL)和模式错误标志(MODF)。与传统的 SPI 相比较，一个经过特别设计的发送保持寄存器(THR)显著改善了传输效率且 THRF 标志表明 THR 是满或空。SPI 工作下忙状态由只读标志 SPIBSY 指示。

图 21-1. SPI 方框图



SPI 接口有 4 个引脚：MISO、MOSI、SPICLK 和 nSS。

- SPICLK、MOSI 和 MISO 通常将两个或多个 SPI 设备连接在一起。数据从主机到从机使用 MOSI 引脚(主出/从入)，从从机到主机使用 MISO 引脚(主入/从出)。SPICLK 信号在主机模式时输出，从机模式时输入。若 SPI 接口禁用，即 SPEN (SPCTL.6) = 0，这些引脚可以作为普通 I/O 口使用。

- nSS (/SS) 是从机选择端。典型配置中，SPI 主机可以使用其某个端口选择某一个 SPI 设备作为当前从机，一个 SPI 从机设备使用它的 /SS 引脚确定自己是否被选中。下面条件下 /SS 被忽略：

- 若 SPI 系统被禁用，即 SPEN (SPCTL.6) = 0 (复位值)
- 若 SPI 作为主机运行，即 MSTR (SPCTL.4) = 1，且 P1.4 (nSS)被配置成输出。
- 若 /SS 被设置成忽略，即 SSIG (SPCTL.7) = 1，这个端口作为普通 I/O 使用。

注意：引脚输出选项见 AUXR10 参考章节“4.3 功能复用”

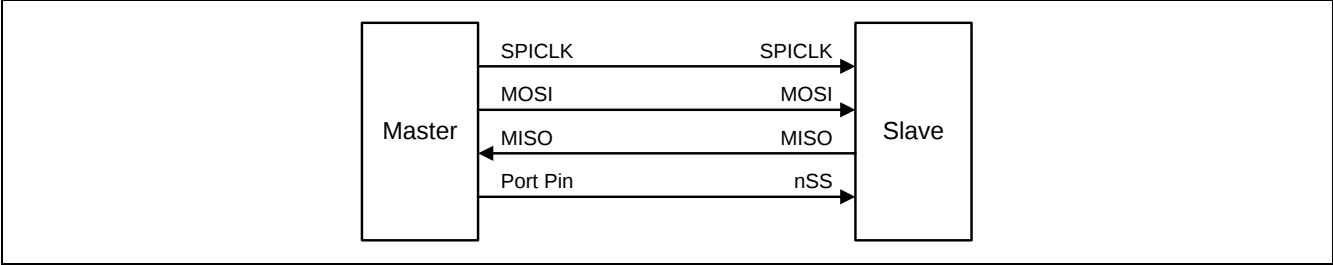
注意，即使 SPI 被配置成主机运行(MSTR=1)，它仍然可以被 nSS 引脚的低电平拉成从机(若 SSIG=0)，一旦发生这种情况，SPIF 位(SPSTAT.7)置位并且 SPEN 会被清零。（参考章节“21.2.3 nSS 引脚模式改变”）。

21.1. 典型 SPI 配置

21.1.1. 单主机和单从机

对于主机：任何端口，包括 SSGPIO，都可以用来控制从机的 nSS 片选引脚。
对于从机：SSIG 为 ‘0’，且 nSS 引脚决定该设备是否被选中。

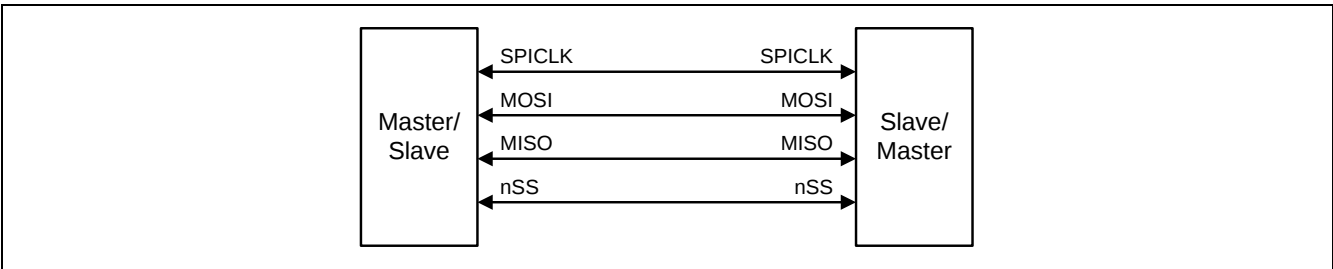
图 21-2. SPI 单主机和单从机结构



21.1.2. 双设备，既是主机也是从机

两个彼此连接的设备，均可成为主机或从机。没有 SPI 操作时，都可以被通过设置 MSTR=1、SSIG=0 和 P1.4(nSS)双向口配置成主机。任何一方要发起传输，它可以配置 P1.4 位输出并强行拉低，使另一个设备发生“被改成从机模式”事件。(参考章节“21.2.3 nSS 引脚模式改变”)。

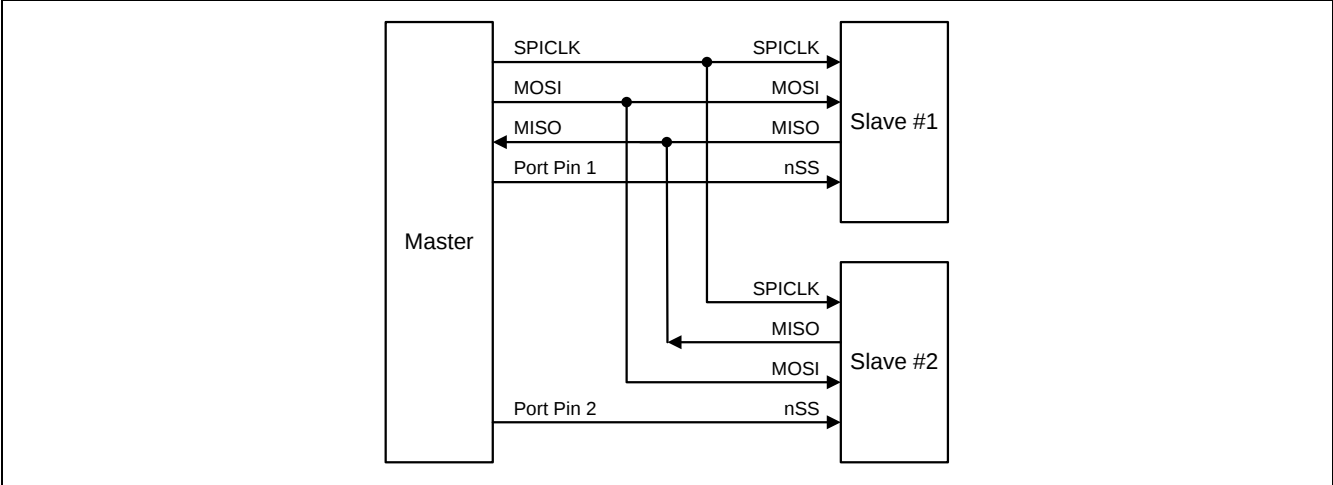
图 21-3. SPI 双设备结构，既是主机也是从机



21.1.3. 单主机和多从机

对于主机：任何端口，包括 nSSGPIO，都可以用来控制从机的/SS 片选引脚。对于所有从机：SSIG 为 ‘0’，nSS 引脚决定该设备是否被选中。

图 21-4. SPI 单主机和多从机结构



21.2. SPI 配置

表 21-1 不但列出了主机/从机模式的配置，而且列出了这些模式的用法。

表 21-1. SPI 主机和从机选择

SPEN (SPCON.6)	SSIG (SPCON.7)	nSS -pin	MSTR (SPCON.4)	模式	MISO -引脚	MOSI -引脚	SPICLK -引脚	注释
0	X	X	X	SPI 禁止	输入	输入	输入	SPI 相关端口引脚是通用端口引脚。
1	0	0	0	从机 (被选中)	输出	输入	输入	被选择为从机。
1	0	1	0	从机 (未被选中)	高阻	输入	输入	未被选中。
1	0	0	1 → 0	从机 (通过模式改变)	输出	输入	输入	模式被改为从机。 若 nSS 被拉低，MSTR 被硬件自动清'0'，且 SPEN 清零，MODF 置位。
1	0	1	1	主机 (空闲)	输入	Hi-Z	Hi-Z	MOSI 和 SPICLK 在主机待机时被置为高阻，以防止总线冲突。
				主机 (激活)		输出	输出	MOSI 和 SPICLK 在主机活动时推挽输出。
1	1	X	0	从机	输出	输入	输入	
1	1	X	1	主机	输入	输出	输出	

“X” 意味着 “忽略”。

21.2.1. 一个从机的补充注意事项

当 CPHA = 0 时，SSIG 必须为 0 且 nSS 引脚必须在每次串行字节传输前负跳变，传输结束恢复正常高电平。
注意 SPDAT 寄存器不能在 nSS 引脚低电平时写入；CPHA = 0，SSIG=1 的操作是未定义的。

当 CPHA =1 时，SSIG 可以为 0 或 1。若 SSIG=0，nSS 引脚可以在每次成功传输之间保持低电平(可以一直拉低)，这种格式有时非常适合单固定主从机配置应用。

21.2.2. 一个主机的补充注意事项

SPI 通讯中，传输总是由主机发起。若 SPI 使能(SPEN=1)并作为主机运行，写入 SPI 数据寄存器(SPDAT)数据即可启动 SPI 时钟生成器和数据传输器。大约半个到 1 个 SPI 位时间后写入 SPDAT 的数据开始出现在 MOSI 线上。

在开始传输之前，主机通过拉低相应 nSS 引脚选择一个从机作为当前从机。写入 SPDAT 寄存器的数据从主机 MOSI 引脚移出到从机的 MOSI 引脚，同时从从机 MISO 移入主机 MISO 的数据也写入到主机的 SPDAT 寄存器中。

移出 1 字节后，SPI 时钟发生器停止，置传输完成标志(SPIF)，若 SPI 中断使能则生成一个中断。主机 CPU 和从机 CPU 中的两个移位寄存器可以看成是一个分开的 16 位环形移位寄存器，数据从主机移到从机同时数据也从从机移到主机。这意味着，在一次传输过程中，主从机数据进行了交换。

21.2.3. nSS 引脚模式改变

若 SPEN=1, SSIG=0, MSTR=1 且/SS 引脚=1, SPI 使能在主机模式。这种情况下, 其他主机可以将/SS 引脚拉低来选择该设备为从机并开始发送数据过来。为避免总线冲突, 该 SPI 设备成为一个从机, MOSI 和 SPICLK 引脚被强制为输入端口, MISO 成为输出端口, SPSTAT 中 SPIF 标志置位, 若此时 SPI 中断使能, 则还会产生一个 SPI 中断。用户软件必须经常去检查 MSTR 位, 若该位被从机选择清零而用户又想要继续保持该 SPI 主机模式, 用户必须再次设置 MSTR 位, 否则, 将处于从机模式。

21.2.4. 发送保持寄存器满标志

为了提高 SPI 发送速度一个特殊设计保持寄存器(THR)可以减短 CPU 数据移动字节与字节传送的延迟时间。THRF(SPSTAT.5)置位表明 THR 的数据是有效的并且等待发送。如果 THR 是空的(THRF=0), 软件写一个字节数据到 SPDAT 数据将存储在 THR 并且 THRF 置位。如果输出移位寄存器(OSR)是空的, 硬件立刻将 THR 数据移到 OSR 并且 THRF 清零。在 SPI 主机模式, OSR 有效数据将触发 SPI 发送。在 SPI 从机模式, OSR 有效数据等待另一个 SPI 主机移出数据。如果 THR 是非空(THRF=1), 软件写一个字节数据到写冲突标志 WCOL (SPSTAT.6)将置位。

21.2.5. 写冲突

MG82F6P32 的 SPI 在发送方向和接收方向是双缓冲数据器。发送新数据在 THR 空时才能写入到缓冲器 THR。只读标志 THRF 表示 THR 是空或非空。在 THRF 为“1”时数据寄存器被写入数据冲突标志 WCOL (SPSTAT.6)将置位。这种情况下, SPDAT 写入操作将被忽略。

主机或从机检测到写冲突时, 主机异常是主机传输过程中有非空控制; 从机是在主机初始化传输没有控制结束时出现冲突。

WCOL 软件写“1”清零。

21.2.6. SPI 时钟速率选择

SPI 时钟率选择(主机模式)使用 SPCON 寄存器的 SPR1 和 SPR0 位及 SPSTAT 寄存器的 SPR2 来设置, 如表 21-2 所示。

表 21-2. SPI 串行时钟速率

SPR2	SPR1	SPR0	SPI 时钟选择	SPI 时钟率 @ SYSCLK=12MHz	SPI 时钟率 @ SYSCLK=48MHz
0	0	0	SYSCLK/4	3 MHz	12 MHz
0	0	1	SYSCLK/8	1.5 MHz	6 MHz
0	1	0	SYSCLK/16	750 KHz	3 MHz
0	1	1	SYSCLK/32	375 KHz	1.5 MHz
1	0	0	SYSCLK/64	187.5 KHz	750 KHz
1	0	1	SYSCLK/2	6 MHz	24 MHz
1	1	0	S0TOF/2	可变	可变
1	1	1	T0OF/2	可变	可变

注意:

1. SYSCLK 是系统时钟。
2. S0TOF 是 UART0 波特率发生器溢出。
3. T0OF 是定时器 0 溢出。

21.3. 数据模式

时钟相位(CPHA)位可以让用户设定数据采样和改变时的时钟沿。时钟极性位 CPOL 可以让用户设定时钟极性。下面图例显示了不同时钟相位(CPHA)。

表 21-3. SPI 模式定义

SPI Mode	CPOL	CPHA	前沿	后沿
0	0	0	采样(上升沿)	设置(下降沿)
1	0	1	设置(上升沿)	采样(下降沿)
2	1	0	采样(下降沿)	设置(上升沿)
3	1	1	设置(下降沿)	采样(上升沿)

图 21-5. SPI 在 CPHA=0 时从机传送格式

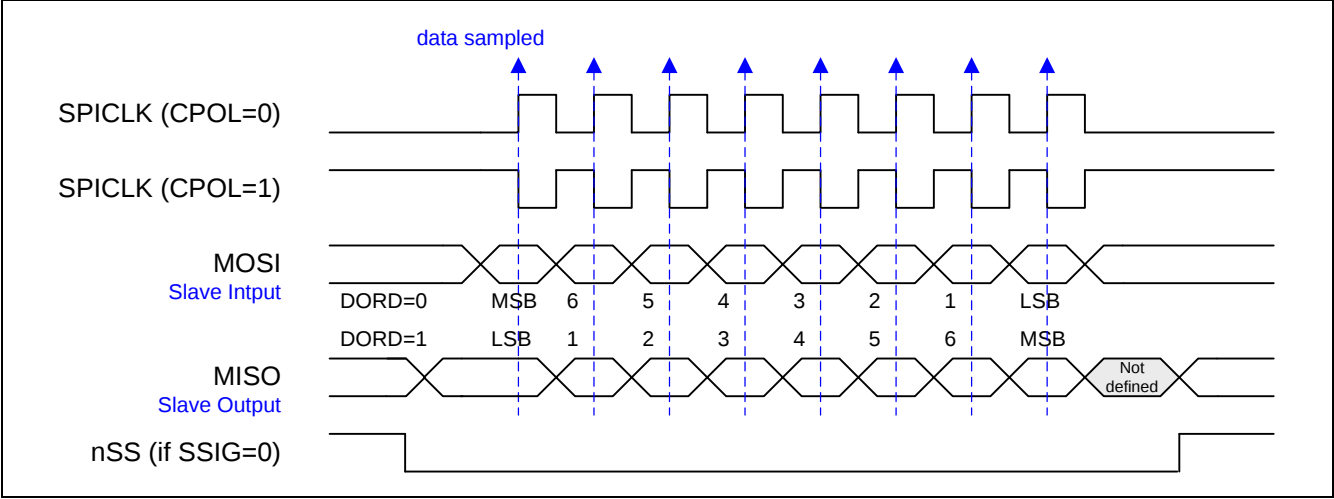


图 21-6. 在 CPHA=1 时从机传送格式

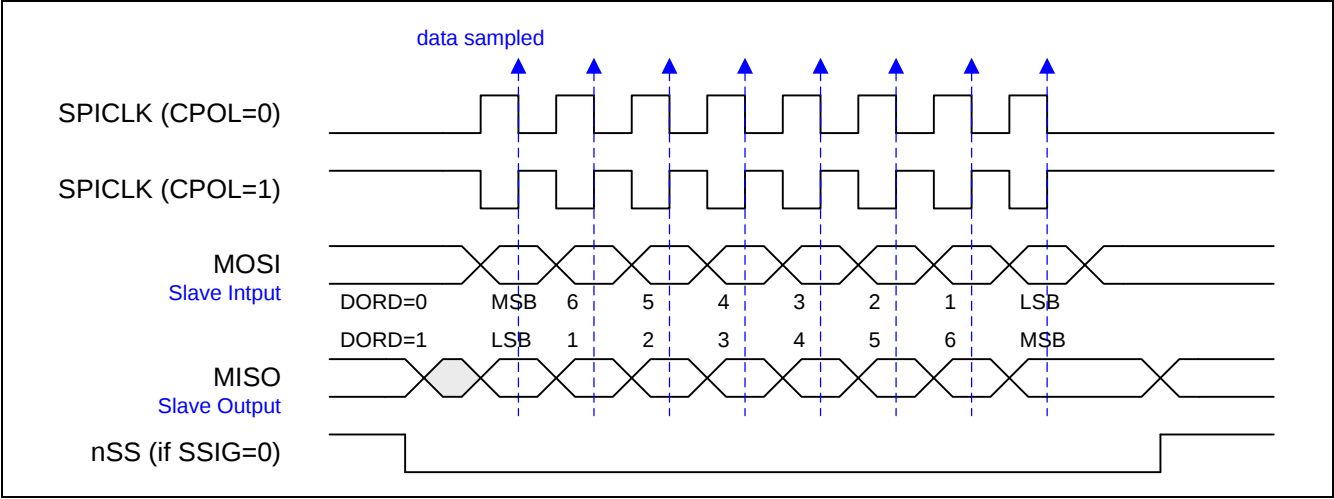


图 21-7. SPI 在 CPHA=0 时主机传送格式

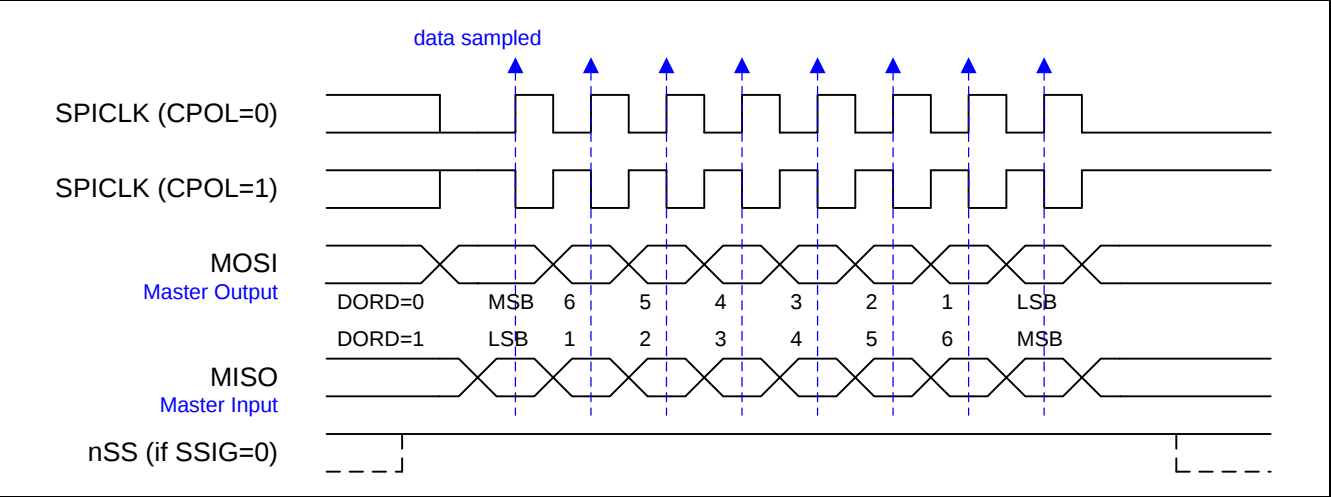
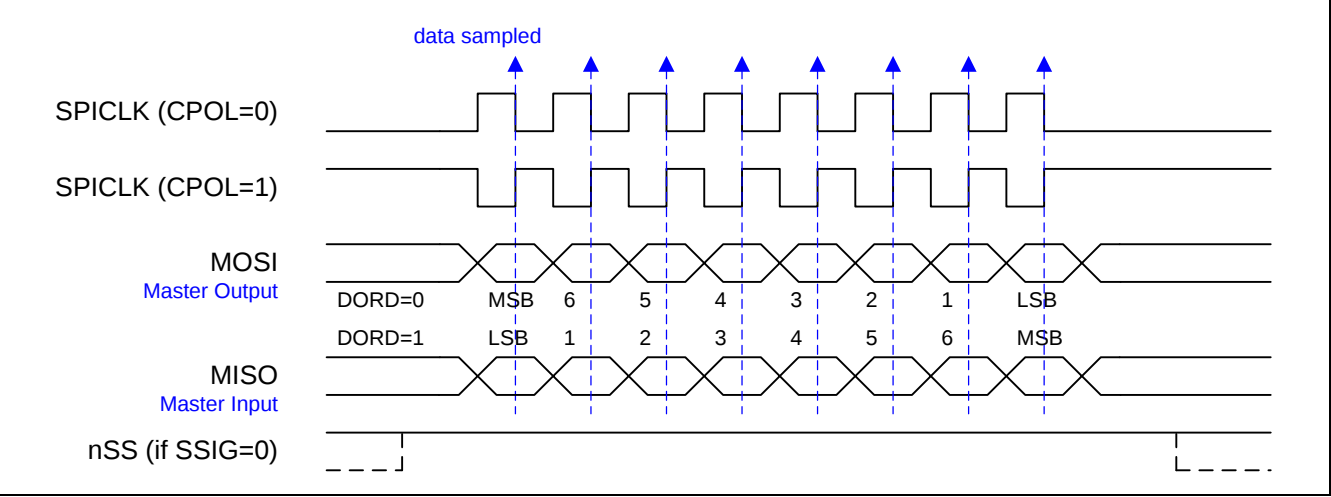


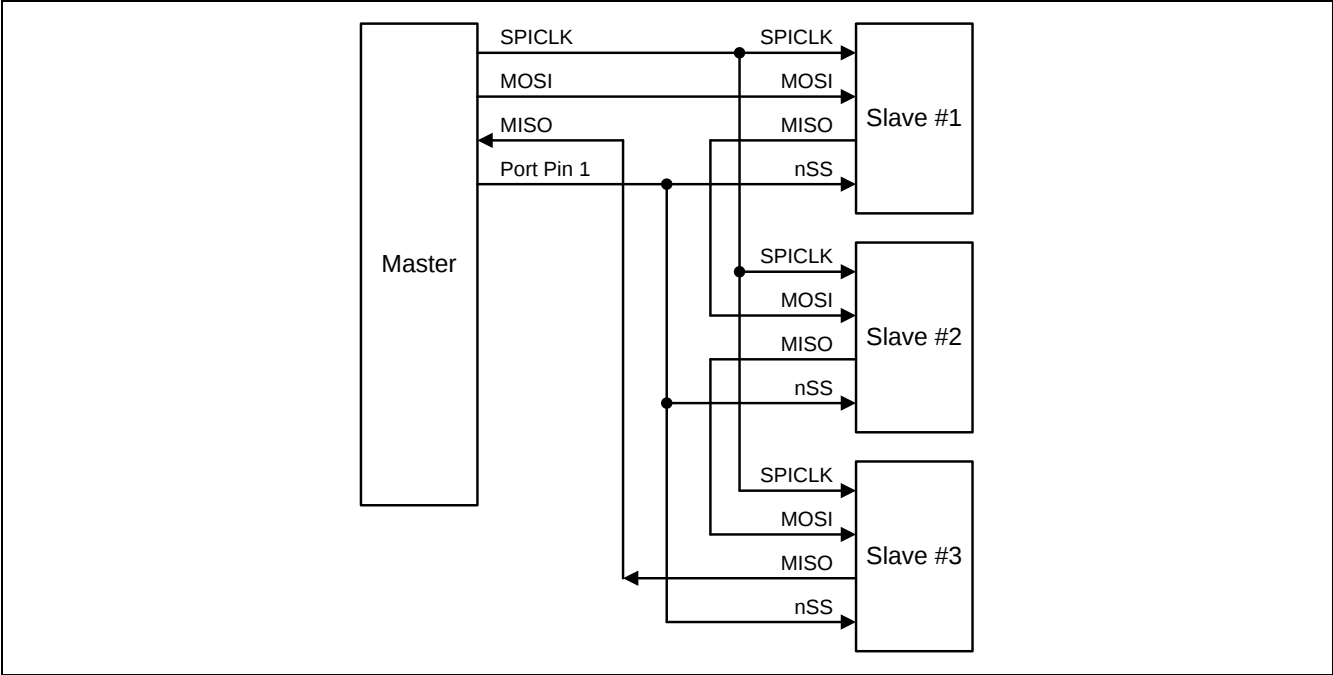
图 21-8. SPI 在 CPHA=1 时主机传送格式



21.4. 菊花链连接

如果 SPI0 被定义为从机模式，它可以连接成一个菊花链结构。第一个的从机的输出连接到第二个从机的输入，第二个从机的输出连接到第三个从机的输入，以此类推。最后一个从机的输出连接到主机的输入。在第二组时钟脉冲期间，每个从机发送出与第一组时钟脉冲接收到的信号完全相同的信号。整个链作为一个大的通信移位寄存器。菊花链功能仅需从主机接一条从机片选线 (nSS)。

图 21-9. SPI 从机在菊花链连接结构



21.4.1. 菊花链配置

如何配置 SPI 从机为菊花链

- 配置 SPCON 去定义数据模式和选择 SPI0 为从机模式.
- 置位 SPI0M0 (AUXR7.4) 使能 SPI0 为菊花链模式.
- 查询 SPIF 获得菊花链通信

21.5. SPI 寄存器

下面是 SPI 操作的相关特殊功能寄存器：

SPCON: SPI 控制寄存器

SFR 页 = 0~7

SFR 地址 = 0x85

复位值 = 0000-0100

7	6	5	4	3	2	1	0
SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SSIG, 忽略 nSS。

0: nSS 引脚决定该设备是主机还是从机。

1: MSTR 位决定该设备是主机还是从机。

Bit 6: SPEN, SPI 使能。

0: SPI 接口禁用, 所有 SPI 引脚可作为通用 I/O 口使用。

1: SPI 使能。

Bit 5: DORD, SPI 数据位序。

0: 传送数据时高位在先(MSB)。

1: 传送数据时低位在先(LSB)。

Bit 4: MSTR, 主机/从机模式选择。

0: SPI 从机模式。

1: SPI 主机模式。

Bit 3: CPOL, SPI 时钟极性选择。

0: SPICLK 空闲是为低电平。SPICLK 时钟脉冲前沿是上升沿, 而后沿是下降沿。

1: SPICLK 空闲是为高电平。SPICLK 时钟脉冲前沿是下降沿, 而后沿是上升沿。

Bit 2: CPHA, SPI 时钟相位选择。

0: /SS 引脚低电平 (SSIG=0)开始放数据并在 SPICLK 后沿改变数据。数据在 SPICLK 的前沿采样。

1: SPICLK 脉冲前沿放数据, 后沿采样。

(注: 如果 SSIG=1, CPHA 必须不为 1, 否则这个功能是没有定义的。)

Bit 1~0: SPR1-SPR0, SPI 时钟率选择位 0 和 1 (主机模式, 与 SPR2 配合使用)

SPR2	SPR1	SPR0	SPI 时钟选择	SPI 速率@ SYSCLK=12MHz	SPI 速率 @ SYSCLK=48MHz
0	0	0	SYSCLK/4	3 MHz	12 MHz
0	0	1	SYSCLK/8	1.5 MHz	6 MHz
0	1	0	SYSCLK/16	750 KHz	3 MHz
0	1	1	SYSCLK/32	375 KHz	1.5 MHz
1	0	0	SYSCLK/64	187.5 KHz	750 KHz
1	0	1	SYSCLK/2	6 MHz	24 MHz
1	1	0	S0TOF/2	可变	可变
1	1	1	T0OF/2	可变	可变

注意:

1. SYSCLK 是系统时钟。

2. S0TOF 是 UART0 波特率发生器溢出。

3. T0OF 是定时器 0 溢出。Note:

注意: 当 QPIEN 使能, SPR1~SPR0 必须软件写 “0”

SPCR1: SPI 控制寄存器 1

SFR 页 = 仅 8 页

SFR 地址 = 0x85

复位值 = 0000-0100

7	6	5	4	3	2	1	0
--	--	SPI0M0	SPSOES	SPFACE	SPR2	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: SPI0M0, SPI 模式, 菊花链模式

SPI0M0	SPI 模块工作模式
0	标准 SPI 模式
1	SPI 菊花链模式

Bit 4: SPSOES, SPI SO 提前采样沿

0: 禁止, SPI SO 一般采样沿

1: 使能. SPI SO 提前采样沿用于更快的 SPI 从机访问速率

Bit 3: SPFACE, SPIF 自动清理使能

0: 禁止, SPIF 仅软件清零

1: 使能. SPIF 在 CPU 读/写 SPDAT 操作的时候 被清零

Bit 2 ~ 0: SPR[2:0], SPI 时钟源选择

SPR2	SPR1	SPR0	SPI 时钟选择 选择	SPI 速率@ SYSCLK=12MHz	SPI 速率 @ SYSCLK=48MHz
0	0	0	SYSCLK/4	3 MHz	12 MHz
0	0	1	SYSCLK/8	1.5 MHz	6 MHz
0	1	0	SYSCLK/16	750 KHz	3 MHz
0	1	1	SYSCLK/32	375 KHz	1.5 MHz
1	0	0	SYSCLK/64	187.5 KHz	750 KHz
1	0	1	SYSCLK/2	6 MHz	24 MHz
1	1	0	S0TOF/2	可变	可变
1	1	1	T0OF/2	可变	可变

SPSTAT: SPI 状态寄存器

SFR 页 = 0~F

SFR 地址 = 0x84

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SPIF	WCOL	THRF	SPIBSY	MODF	0	0	0
R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit 7: SPIF, SPI 传输完成标志。

0: 软件写“1” SPIF 清零。

1: 当一次串行传输完成时, SPIF 位置位, 同时若 SPI 中断允许, 会产生一个中断。若 nSS 引脚在主机模式下被拉低且 SSIG=0, SPIF 位也会置位以表明“模式改变”。

Bit 6: WCOL, SPI 写冲突标志。

0: 软件写“1” WCOL 清零。

1: SPI 数据寄存器(SPDAT)在数据传输过程中被写入此位置位(见章节“21.2.5 写冲突”)。

Bit 5: THRF, 发送保持寄存器(THR)非空标志。只读。

0: 表明 THR 是“空的”。当 THR 为空时此位被硬件清零, 这意味着 THR 中的数据已经被装入移位输出寄存器进行发送, 而现在用户可以向 SPDAT 写下一个要发送的数据。

1: 表明 THR 是“非空”。当软件向 SPDAT 写数据时由硬件置位。

Bit 4: SPIBSY, SPI 忙标志。只读。

0: 表示 SPI 是空闲状态并且所有的移位寄存器是空的。

1: 置位表示 SPI 传输进行中(主机或从机)。

Bit 3: MODF, 模式错误标志。当检测到主机模式冲突时(nSS 为低电平, MSTEN=1 并且 SSIG=0), 硬件置该位为 1。如果中断使能, 就会产生一个中断。该位不会由硬件自动清零, 必须由软件写 “1” 清零。

SPDAT: SPI 数据寄存器

SFR 页 = 0~F
SFR 地址 = 0x86
复位值 = 0000-0000

7	6	5	4	3	2	1	0
(MSB)							(LSB)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SPDAT 有两个物理缓冲器供发送和接收过程中各自独立写入和读取。

AUXR10: 辅助寄存器 10

SFR 页 = 仅 7 页
SFR 地址 = 0xA4
复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: SPIPS1~0, SPI 端口选择 1~0.

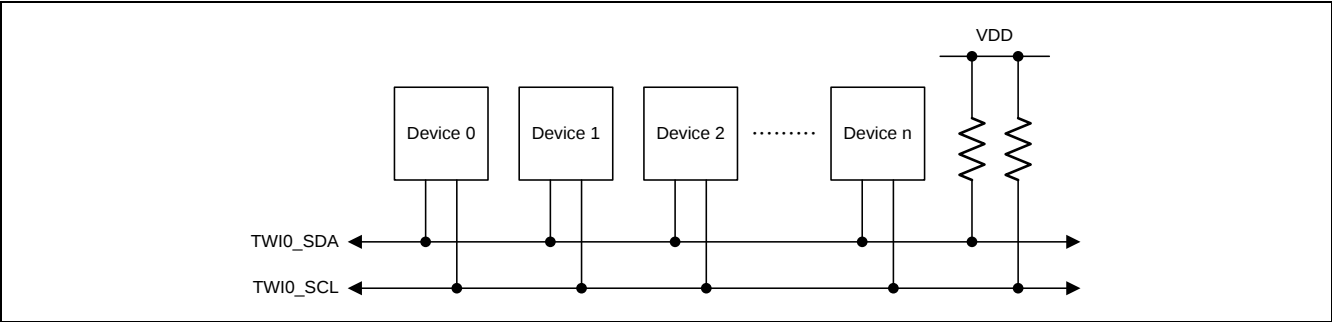
引脚选项	SPIPS1~0	nSS	MOSI	MISO	SPICLK
0	0 0	P1.4	P1.5	P1.6	P1.7
1	0 1	P0.1	P0.2	P4.1	P4.0
2	1 0	P3.3	P1.5	P1.6	P1.7
3	1 1	P1.7	P3.5	P3.4	P3.3

22. 双线串行接口 (TWI0/I2C0 & TWI1/I2C1)

双线串行接口是一个双线双向总线。双线串行接口(TWI)很适合于典型的处理器应用。**MG82F6P32** 内嵌两个 TWI/I2C ,TWI0/I2C0 和 TWI1/I2C1。两者都支持多从机地址识别。

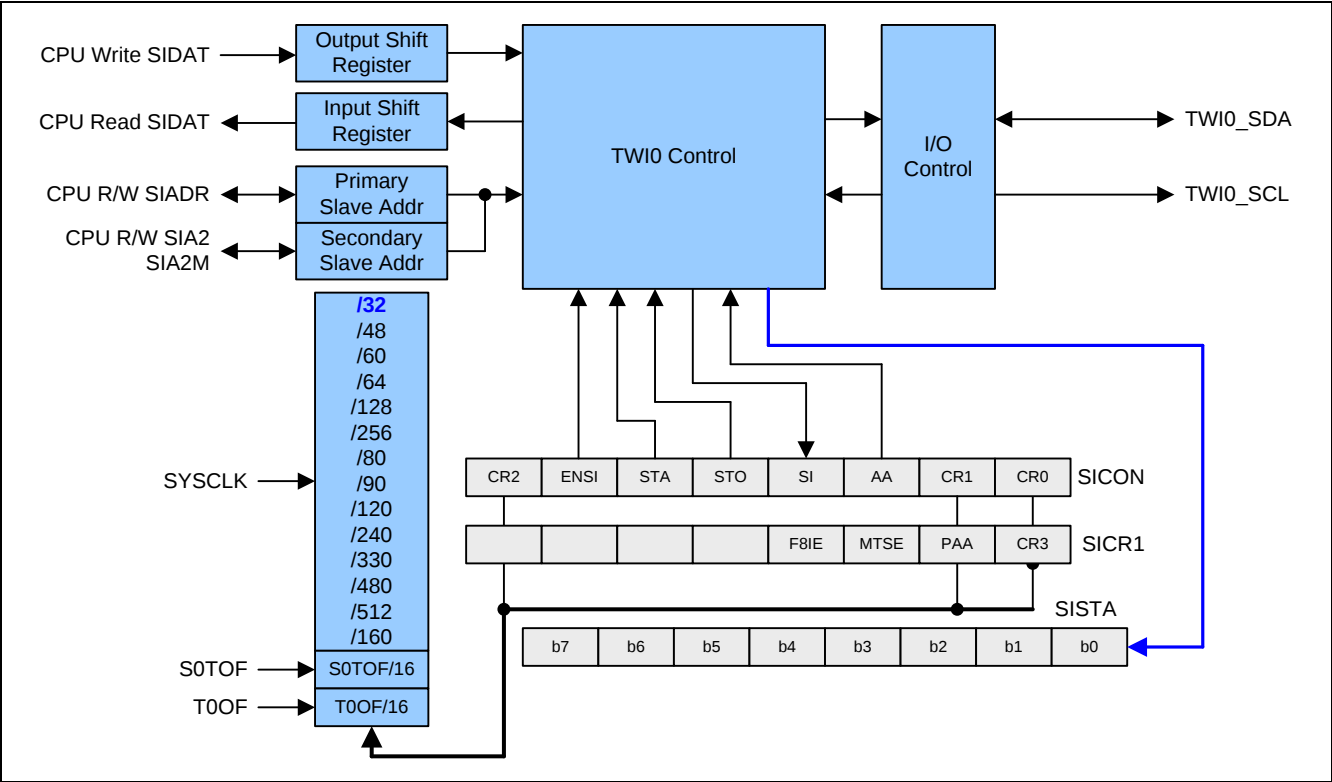
TWI/ I2C 协议允许系统设计人员只用两根双向传输线来连接多达 128 个不同的设备，一根用于时钟(TWI0_SCL)，一根用于数据(TWI0_SDA)。双线串行接口(TWI)由 TWI0_SDA(串行数据)、TWI0_SCL(串行时钟)控制产生和同步，仲裁逻辑以及起始/停止(START/STOP)。唯一需要的外部硬件就是在 TWI/ I2C 的每根传输线上添加一个上拉电阻。所有连接到总线的设备都有各自的地址，而且 TWI/ I2C 协议解决了总线仲裁的问题。

图 22-1. TWI/I2C 总线互联



TWI/ I2C 总线可以操作在主机或从机也可以是多主机。CPU 通过 SICON(串行接口控制寄存器)、SISTA(串行接口状态寄存器)、SIDAT(串行接口数据寄存器，用于发送和接收 TWI/ I2C 数据)、SIADR(串行接口从机地址寄存器)这四个特殊功能寄存器与 TWI 相连。TWI 硬件通过两根数据线与串行总线相连：SDA(串行数据线)和 SCL(串行时钟线)。

图 22-2. TWI/I2C 方框图



22.1. 工作模式

TWI/ I2C 有 4 种操作模式：1) 主机/发送模式，2) 主机/接收模式，3) 从机/发送模式 4) 从机/接收模式。SI 软件清零之后 SICON 寄存器的位 STA、STO 和 AA 决定 TWI 硬件下一个执行的是哪一个操作。当下一个操作完成，SISTA 寄存器将更新一个新状态同时 SI 也会硬件置位。现在，中断服务程序会被调用(如果 TWI/ I2C 中断使能)，软件可以通过新的状态区分需要调用哪一个子程序。

22.1.1. 主机发送模式

在主机发送模式，一些数据字节可以发送到一个从机接收器。在进入主机发送模式前，SICON 必须作如下设置：

SICON

7	6	5	4	3	2	1	0
CR2	ENSI	STA	STO	SI	AA	CR1	CR0
Bit rate	1	0	0	0	x	Bit rate	

CR0、CR1 和 CR2 定义了串行位速率。ENSI 必须设置为逻辑 1 来使能 TWI/ I2C。如果 AA 位复位，在其它设备成为总线的主机时，TWI/ I2C 将不会应答它自身的从机地址或广播地址。也就是说，如果 AA 复位，TWI/ I2C 不能进入从机模式。STA、STO 与 SI 必须复位。

置位 STA 也许可以立即进入主机发送模式。TWI/ I2C 逻辑将检测串行总线并且在总线空闲时产生一个 START 信号。发送完 START 信号后，串行中断标志(SI)将被置位，并且状态寄存器(SISTA)中的状态编码将为 08H。这个状态编码必须用于指示一个中断服务程序加载从机地址和数据方向位(SLA+W)到 SIDAT。SICON 的 SI 位必须清零，串行传输才能继续进行。

当从机地址与方向位发送完，并且接收到一个应答位后，串行中断标志(SI)会再次被置位。SISTA 可能为以下的编码：在主机模式为 18H，20H 或 38H，如果从机模式使能(AA=1)，也可以为 68H，78H 或 B0H。在这些状态编码下对应的操作将在随后的工作流程图中详细叙述。在一个重复的 START 信号后(状态编码 10H)，TWI 可以通过向 SIDAT 写入 SLA+R 进入主机接收模式。

22.1.2. 主机接收模式

在主机接收模式，可以从从机发送器接收一定数量的字节数据。SICON 也必须如主机发送模式一样初始化。开始信号发送后，中断服务程序必须向 SIDAT 写入 7 位从机地址与数据方向位(SLA+R)。SICON 的 SI 位必须清零，串行传输才能继续进行。

在从机地址与数据方向位发送完并且接收到应答位后，串行中断标志(SI)重新置位。SISTA 可能为以下的编码：在主机模式为 40H，48H 或 38H，如果从机模式使能(AA=1)，也可以为 68H，78H 或 B0H。这些状态编码下对应的操作将在随后的工作流程图中详细叙述。在一个重复的 START 信号后(状态编码 10H)，TWI 可以通过向 SIDAT 写入 SLA+W 进入主机接收模式。

22.1.3. 从机发送模式

在从机发送模式下，一些数据字节发送给主机接收。SIADR 和 SICON 必须如下初始化从机发送模式：

SIADR

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	GC

|<----- Own Slave Address ----->|

高 7 位是响应被主机寻址的 TWI/ I2C 地址。如果 LSB (GC)置位，TWI 将应答广播地址(00H)；否则将忽略广播地址。

SICON

7	6	5	4	3	2	1	0
CR2	ENSI	STA	STO	SI	AA	CR1	CR0
x	1	0	0	0	1	x	x

在从机模式下 CR0、CR1 和 CR2 不影响 TWI/ I2C。ENSI 必须置位去使能 TWI/ I2C。AA 必须置位去使能 TWI 应答自己的从机地址或广播地址。STA、STO 和 SI 必须清零。

当 SIADR 和 SICON 初始化之后，TWI/ I2C 会等待直到其从机地址被寻址并且数据方向为“1”(R)，TWI/ I2C 将工作于从机发送模式。在接收到自身的从机地址以及“R”位后，串行中断标志(SI)置位，并且可以从 SISTA 读出一个可用的状态编码。这些状态编码可以用作指示一个中断服务程序，在这些状态编码下对应的操作将在随后的工作流程图详细叙述。当 TWI/ I2C 处于主机模式时，如果仲裁失败也可能进入从机发送模式(参考 B0H 状态)。

如果在一次传输的过程中 AA 位复位，TWI/ I2C 将发送完当前字节的数据后进入 C0H 或 C8H 状态。TWI 会转换到未被寻址从机模式，如果主机继续传输，TWI/ I2C 将会忽略主机接收器，因此主机总是接收到“1”。当 AA 复位时，TWI/ I2C 不会回应其从机地址或广播地址，但是会继续监测串行总线。在任何时候可以通过置位 AA 恢复，这意味着 AA 位可用于暂时从总线中隔离 TWI/ I2C。

22.1.4. 从机接收模式

在从机接收模式，会从主机发送器接收一些数据字节。数据传送的初始化与从机发送模式一样。

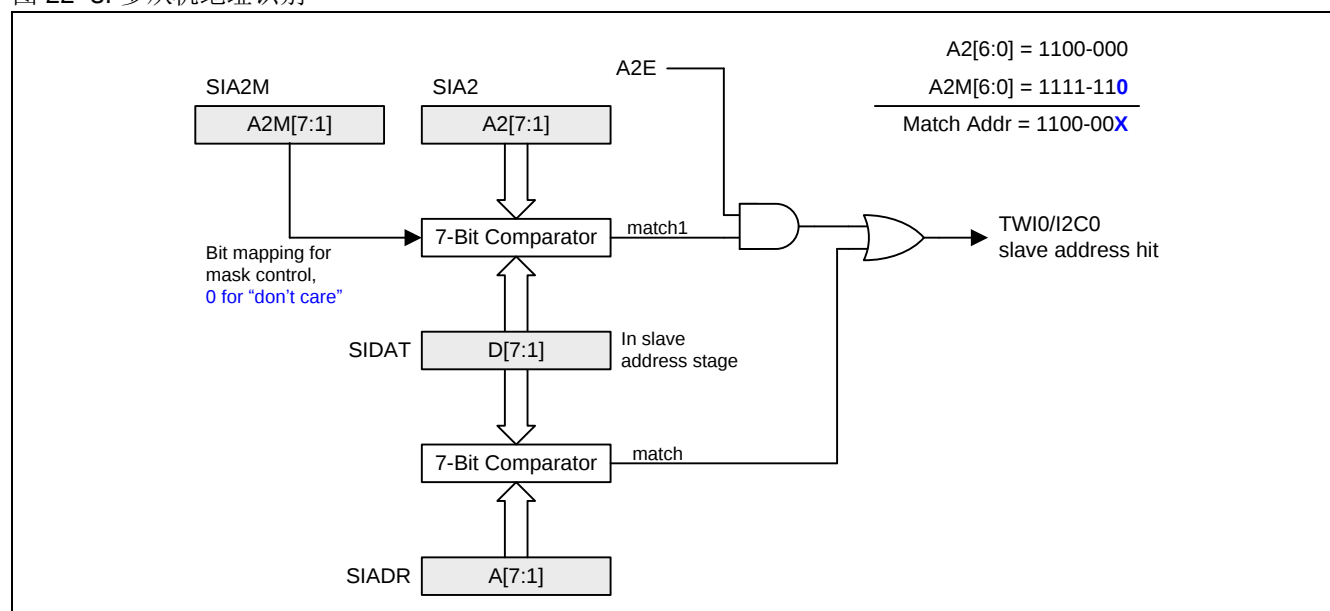
SIADR 与 SICON 初始化后，TWI/ I2C 会等待直到其从机地址被寻址并且数据方向为“0”(W)，TWI/ I2C 将工作于从机接收模式。在接收到其从机地址与“W”位后，串行中断标志(SI)置位，并且可以从 SISTA 读出一个可用的状态编码。这些状态编码可以用作指示一个中断服务程序，在这些状态编码下对应的操作将在随后的工作流程图详细叙述。当 TWI/ I2C 处于主机模式时，如果仲裁失败可能进入从机接收模式(参考状态 68H 和 78H)。

如果在一次传输的过程中 AA 位被复位，TWI/ I2C 会在接收到下一个字节后回复 NACK(逻辑 1)。当 AA 复位时，TWI/ I2C 不会响应自身的从机地址或广播地址，但是会继续监测串行总线。在任何时候可以通过置位 AA 恢复，这意味着 AA 位可用于暂时从总线中隔离 TWI/ I2C。

22.1.5. 多从机地址识别

在 **MG82F6P32** 中 SIADR 定义 TWI/I2C 的首从机地址。**MG82F6P32** 还提供带有掩码功能的第二从机地址，该掩码功能在 SIA2 和 SIA2M 上实现。在从地址掩码 SIA2M[7:1] 的位位置上的 1，使能所接收的从地址与这些位的从地址 SIA2[7:1] 之间进行比较。从地址掩码中的一个 0 表示该位将在比较中被“忽略”。在这种情况下，收到的从机地址在该位上可以是 1 或 0。

图 22-3. 多从机地址识别



22.2. 混合状态

有两个 SISTA 编码没有与已经定义的 TWI/ I2C 硬件状态对应，描述如下：

S1STA = F8H:

这个状态编码表明还没有相应的信息可用，因为串行中断标志(SI)还没有置位。这种情况发生在状态转换之间和 TWI/ I2C 未涉及串行传输时。

S1STA = 00H:

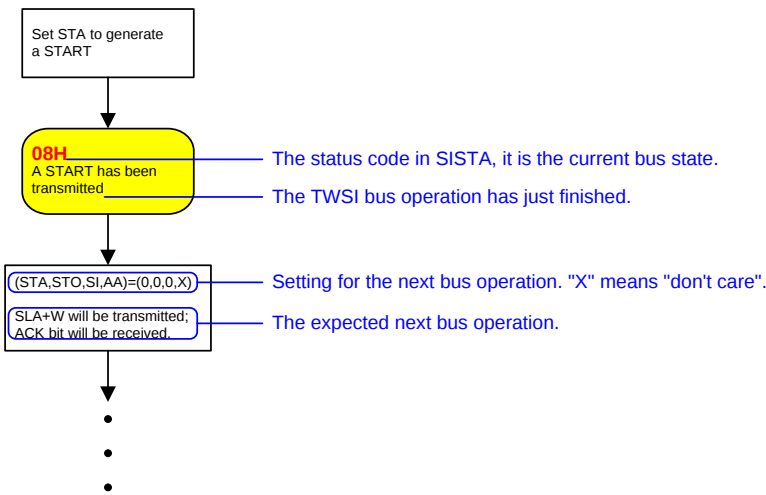
这个状态编码表明在一个 TWI/ I2C 串行传输过程中发生总线错误。当一个 START 或 STOP 信号在一帧的非法位置发送时，总线错误就会发生。例如：在传输一个字节地址、数据时，或者在应答位。总线错误也会在外界干扰扰乱内部 TWI/ I2C 信号时发生。当总线错误发生时，SI 被置位，STO 标志必须置位并且 SI 必须软件清零用来从总线错误中恢复。这会使 TWI/ I2C 进入“未被寻址”的从机状态(已定义的状态)并且清除 STO 标志(SICON 的其它位不受影响)。TWI/ I2C0_SDA 与 TWI/ I2C0_SCL 线将被释放(不会发送 STOP 信号)。

22.3. 使用 TWI/I2C

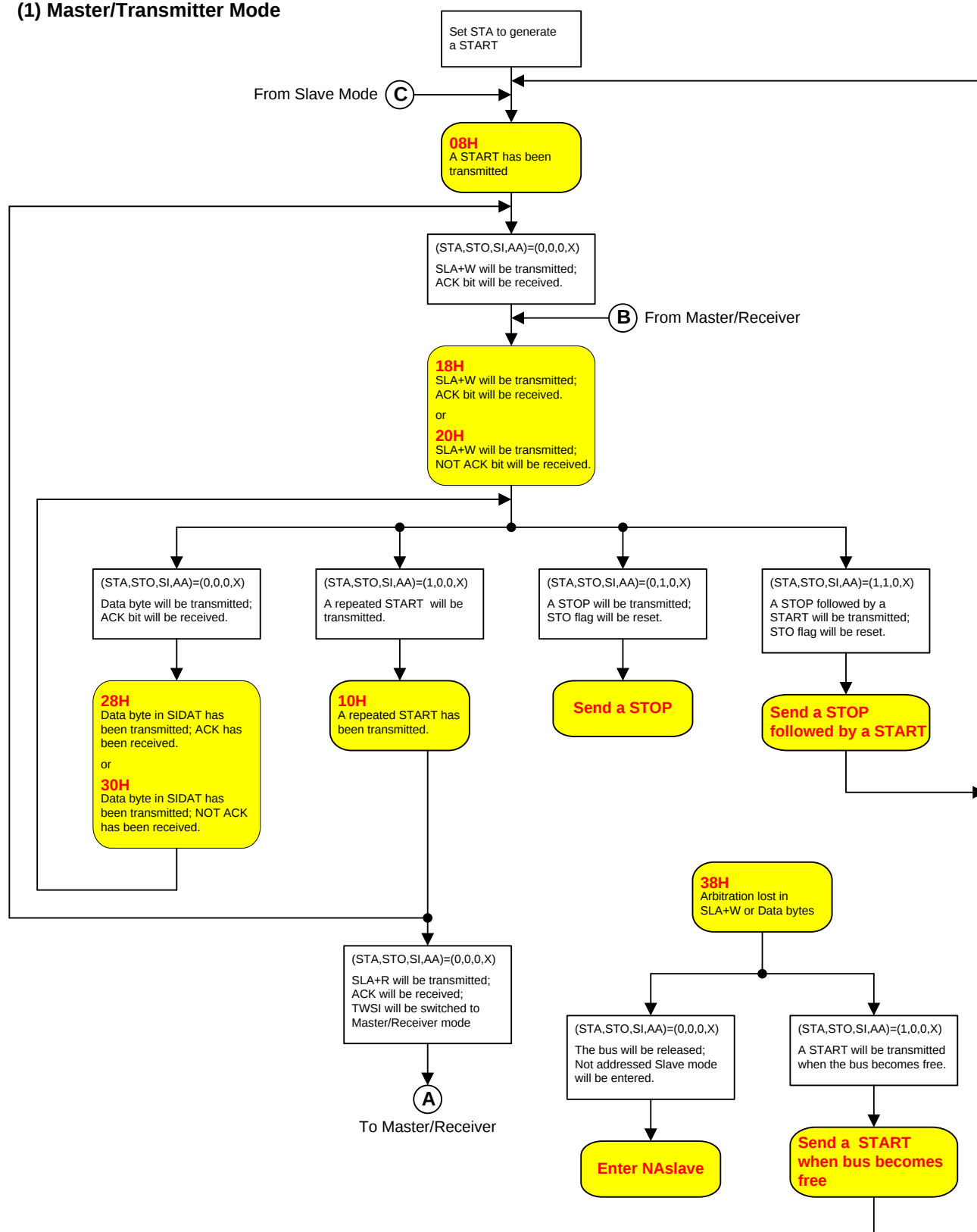
TWI/ I2C 是面向字节并且基于中断的。中断会在所有总线事件后发生，例如接收到一字节数据或发送 START 信号。因为 TWI/ I2C 是基于中断的，应用软件可以自由的一个 TWI/ I2C 字节发送的过程中处理其它工作。注意，TWI/ I2C 中断标志位(EIE1.6)与 EA 位允许应用程序选择在 SI 标志出现时是否产生中断请求。当 SI 标志出现时，表明 TWI/ I2C 已经完成一个操作并且等待程序响应。此时状态寄存器 SISTA 保存的状态编码表明 TWI/ I2C 总线的当前状态。用户程序可以通过对 STA, STO 和 AA 位(在 SICON 中)进行适当的编程来决定接下来 TWI/ I2C 总线将如何运行。

下面的操作流程图将指导用户通过“状态到状态”(state-by-state)的操作来使用 TWI/ I2C。首先，用户应该向 SIADR 写入自身的从机地址(参考前面对 SIADR 的描述)。作为主机时，在初始 SICON 后，第一步为置位“STA”来向总线产生一个 START 信号。作为从机时，在初始化 SICON 后，TWI/ I2C 等待直到被寻址。然后参考操作流程图对 SICON 的 STA, STO, SI, AA 位进行适当的编程来进行后续动作。当 SI 清零后 TWI 硬件就会进行下一步动作，因此推荐使用如下两个步骤：先对 STA, STO 与 AA 编程，然后清零 SI 位(可以使用“CLR SI”指令)来进行可靠的操作。

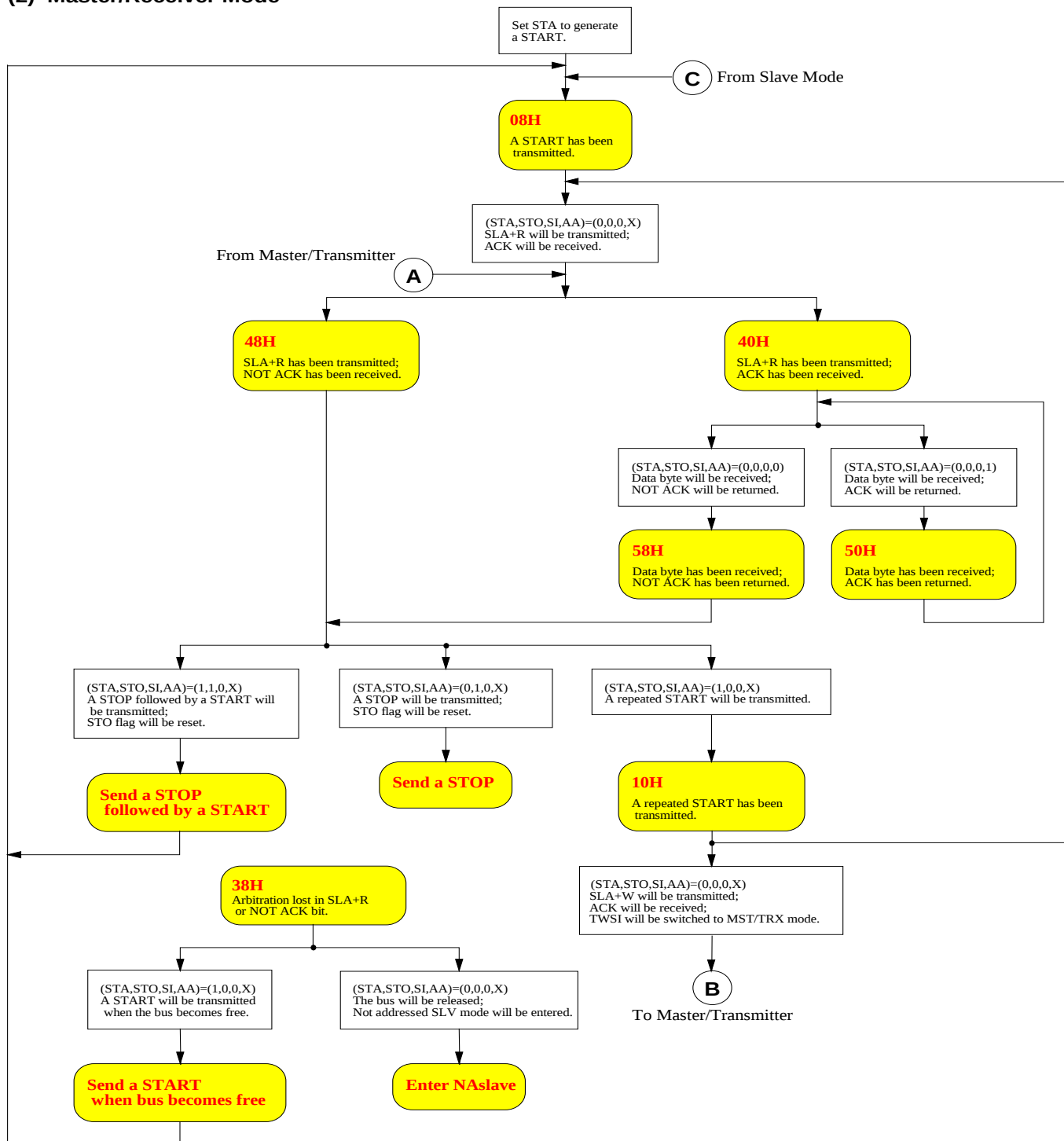
下面的图指出如何读流程图



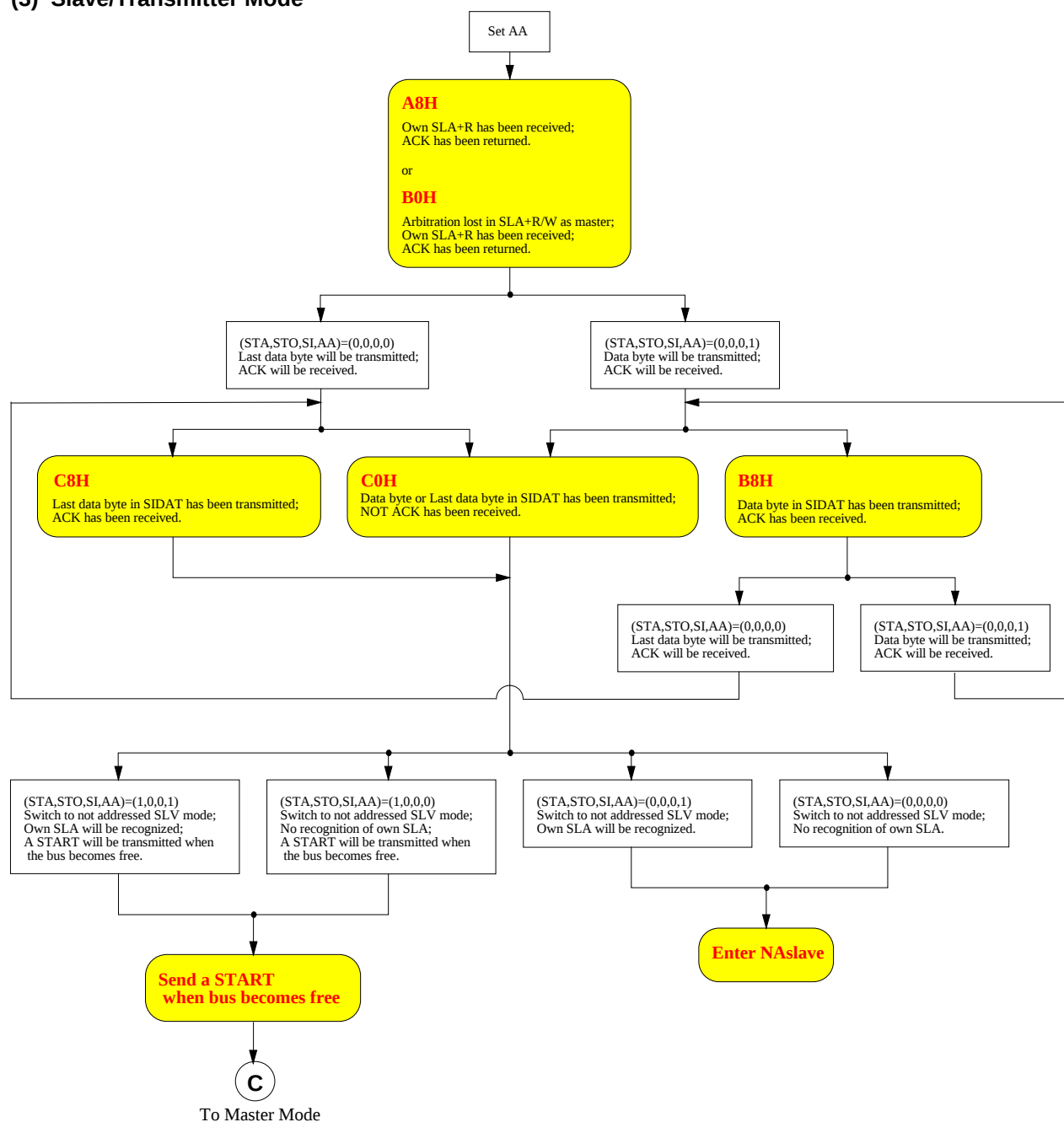
(1) Master/Transmitter Mode



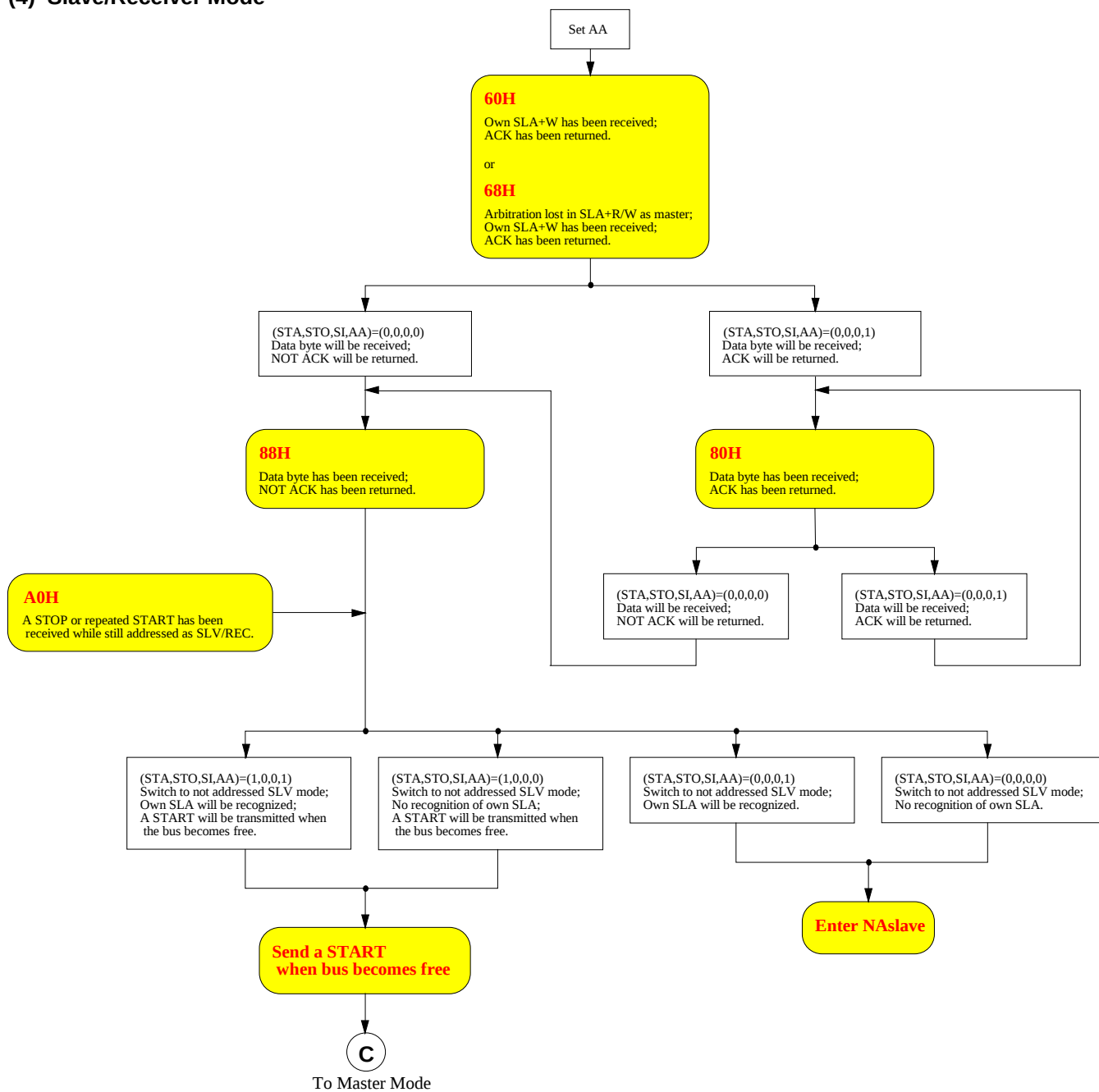
(2) Master/Receiver Mode



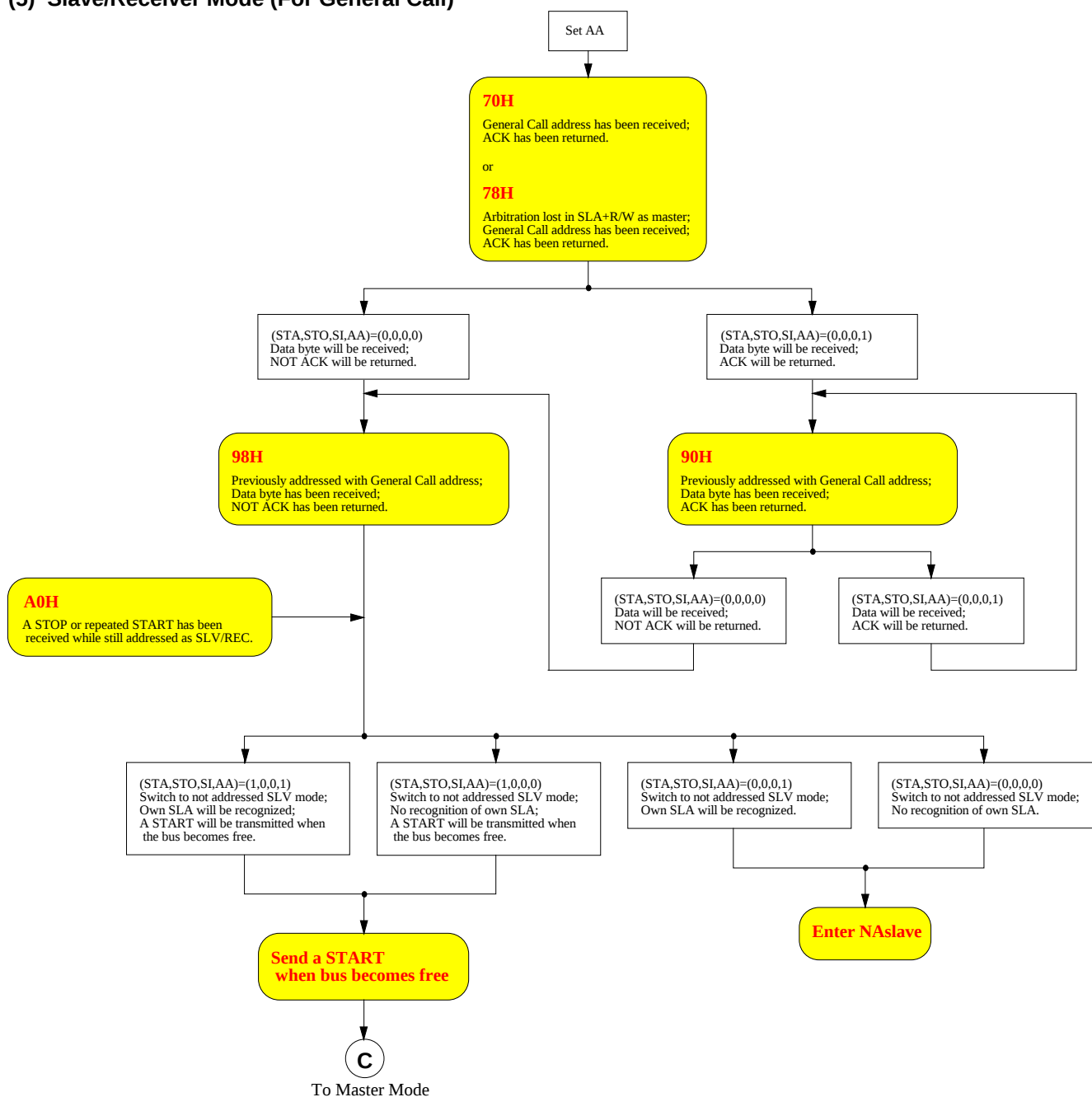
(3) Slave/Transmitter Mode



(4) Slave/Receiver Mode



(5) Slave/Receiver Mode (For General Call)



22.4. TWI0/I2C0 寄存器

SIADR: TWI0/I2C0 地址寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xD1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
A6	A5	A4	A3	A2	A1	A0	GC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~1: CPU 可以直接对此寄存器进行读写。SIADR 不受 TWI0/I2C0 硬件的影响。当 TWI0/I2C0 处于主机模式时此寄存器的值会被忽略。当处于从机模式时，寄存器的高七位必须加载本机的从机地址，最高位与从 TWI0/I2C0 总线上收到的首位相对应。

Bit 0:GC, 如果 GC 置位，广播地址(00H)会被识别，否则忽略。

SIDAT: TWI0/I2C0 数据寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xD2

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

此寄存器保存着一字节将要发送或者刚接收到的数据。在没有进行移位工作时，CPU 可以直接对此寄存器进行读写。这种情况发生在 TWI0/I2C0 正处于一个被定义的状态并且串行中断标志位(SI)置位。只要 SI 被置位，SIDAT 中的数据总是保持稳定的。在数据被移出时，总线上的数据同时移入，SIDAT 总保存着总线上出现的最后一个字节数据。因此在仲裁失败时，主机发送切换为从机接收的过程会在 SIDAT 中产生一个正确的数据。

SIDAT 与 ACK 标志位组成一个 9 位的移位寄存器，可以在移入或移出一个 8 位的数据后，跟随一个应答位。ACK 标志由 TWI0 硬件控制，CPU 访问不到。串行数据在 TWI0/I2C0_SCL 的上升沿移入 SIDAT 寄存器。当一字节的数据完全移入 SIDAT 后，SIDAT 中的数据将是可用的，并且控制逻辑会在第 9 个时钟周期返回一个应答位。串行数据在 TWI0/I2C0_SCL 的下降沿从 SIDAT 寄存器移出。

CPU 向 SIDAT 写入数据后，SD7 位将首先出现在 SDA 线上。9 个时钟周期后，SIDAT 中的 8 位数据将被发送完成，并且通过应答位返回 ACK 标志。注意发送出去的 8 位数据会移回 SIDAT。

SICON: TWI0/I2C0 控制寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xD4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CR2	ENSI	STA	STO	SI	AA	CR1	CR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CPU 可以直接读写此寄存器。其中两个位会受 TWI0/I2C0 硬件的影响：SI 位会在串行中断请求时置位，STO 位会在总线出现 STOP 状态时清零。STO 位也会在 ENS1=0 时清零。

Bit 7: CR2, TWI0/I2C0 时钟率选择位 2 (与 CR1 和 CR0 一起使用)。

Bit 6: ENSI, TWI0/I2C0 硬件使能位。

ENSI 为“0”时，TWI0/I2C0_SDA 与 TWI0_SCL 输出为高阻态，TWI0/I2C0_SDA 与 TWI0/I2C0_SCL 输入信号被忽略，TWI0/I2C0 处于未寻址从机状态，SICON 的 STO 位被强制置为“0”，但不影响其它位。TWI0/I2C0_SDA 与 TWI0/I2C0_SCL 可用作通用 I/O 引脚。ENSI 为“1”时，TWI0/I2C0 使能，TWI0/I2C0_SDA 和 TWI0/I2C0_SCL 端口锁存器(比如 P4.1 和 P4.0)必须设置为逻辑 1 并且 I/O 模式必须配置成开漏模式以用于接下来的串行通讯。

Bit 5: STA, 开始(START)标志。

当 STA 位被置位进入主机模式时，TWI0/I2C0 硬件将检查串行总线的状态，若总线空闲将产生一个 START 信号。若总线忙，TWI0/I2C0 将等待 STOP 信号出现并且在一个延迟后产生 START 信号。如果 STA 在 TWI0/I2C0 已经是处于主机模式并且一个或多个字节已被发送或接收的情况下置位，TWI0/I2C0 会发送一个重复的 START 信号。STA 可以在任何时候置位，也可以在 TWI0/I2C0 是一个被寻址的从机时置位。当 STA 位复位时，无 START 或重复的 START 信号产生。

Bit 4: STO, 停止(STOP)标志。

当 TWI0/I2C0 处于主机模式时，置位 STO 会向串行总线发送一个 STOP 信号。当在总线上检测到 STOP 信号时，TWI0/I2C0 硬件清除 STO 标志。在从机模式时，置位 STO 标志可从总线错误状态恢复。在这种情况下不会向总线发送 STOP 信号，但是 TWI0/I2C0 硬件表现就像已经接收到一个 STOP 信号，并且转换到未被寻址的从机接收模式。STOP 标志自动被硬件清零。如果 STA 与 STO 位同时置位，若 TWI0/I2C0 处于主机模式将产生一个 STOP 信号(当处于从机模式时将产生一个内部的 STOP 信号，但不发送)，接着发送一个 START 信号。

Bit 3: SI，串行中断标志。

当一个新的 TWI0/I2C0 状态出现在 SISTA 寄存器时，SI 标志会被硬件置位。如果 TWI0/I2C0 中断允许，中断服务程序将会运行。唯一不会使 SI 置位的状态是指出没有相关状态信息可以获得的 F8H。当 SI 置位时，TWI0/I2C0_SCL 线上的低电平会延长，并且串行传输暂停。TWI0/I2C0_SCL 线上的高电平不受串行中断 SI 标志影响。SI 必须由软件写“0”清零。SI 标志复位时不会产生中断请求，TWI0/I2C0_SCL 线上的时钟也不会延长。

Bit 2: AA，确认应答标志。

如果 AA 标志设为“1”，一个应答 ACK(TWI0/I2C0_SDA 低电平)将在 TWI0/I2C0_SCL 的应答时钟周期内回复，当：

- 1) 接收到本机的从机地址。
- 2) TWI0/I2C0 处于主机/接收模式时，接收到一字节的数据。
- 3) TWI0/I2C0 处于已被寻址的从机/接收模式时，接收到一字节的数据。

如果 AA 标志设为“0”，一个无应答 NACK(TWI0/I2C0_SDA 高电平)将在 TWI0/I2C0_SCL 的应答时钟周期内回复，当：

- 1) TWI0/I2C0 处于主机/接收模式时，接收到一字节的数据
- 2) TWI0/I2C0 处于已被寻址的从机/接收模式时，接收到一字节的数据

Bit 7, 1~0: CR2、CR1 和 CR0，时钟率选择位

TWI0/I2C0 处于主机模式时，这三个位决定串行时钟频率。最高主机模式时钟频率可达 **1MHz**。当 TWI0/I2C0 处于从机模式时，时钟频率并不重要，因为 TWI0/I2C0 会自动同步任何主机的时钟频率，高达 400KHz。[表 22-1](#) 给出不同的时钟速率设置：

表 22-1. TWI0/I2C0 串行时钟速率

CR3	CR2	CR1	CR0	TWI0/I2C0 时钟选择	TWI0/I2C0 时钟 100KHz 左右		TWI0/I2C0 时钟 400KHz 左右	
					系统时钟	TWI0/I2C0 时钟	系统时钟	TWI0/I2C0 时钟
0	0	0	0	SYSClk/32	N/A	N/A	12MHz	375KHz
0	0	0	1	SYSClk/48	6M Hz	125 KHz	N/A	N/A
0	0	1	0	SYSClk/60	6M Hz	100 KHz	24M Hz	400 KHz
0	0	1	1	SYSClk/64	6M Hz	93.75 KHz	24M Hz	375 KHz
0	1	0	0	SYSClk/128	12M Hz	93.75 KHz	48M Hz	375 KHz
0	1	0	1	SYSClk/256	24M Hz	93.75 KHz	N/A	N/A
0	1	1	0	S0TOF/16	可变	可变	可变	可变
0	1	1	1	T0OF/16	可变	可变	可变	可变
1	0	0	0	SYSClk/80	8M Hz	100 KHz	32M Hz	400 KHz
1	0	0	1	SYSClk/90	8M Hz	88.89 KHz	36M Hz	400 KHz
1	0	1	0	SYSClk/120	12M Hz	100 KHz	48M Hz	400 KHz
1	0	1	1	SYSClk/240	24M Hz	100 KHz	N/A	N/A
1	1	0	0	SYSClk/330	32M Hz	96.97 KHz	N/A	N/A
1	1	0	1	SYSClk/480	48M Hz	100 KHz	N/A	N/A
1	1	1	0	SYSClk/512	48M Hz	93.75 KHz	N/A	N/A
1	1	1	1	SYSClk/160	16M Hz	100 KHz	N/A	N/A

注意:

1. 最大 TWI0/I2C0 时钟速率在 1MHz 以下, 设置 SYSClk=32MHz 将产生 1MHz 时钟速率。
2. SYSClk 是系统时钟。
3. S0TOF 是 UART0 波特率发生器溢出。
4. T0OF 是定时器 0 溢出。

SICR1: TWI0/I2C0 控制寄存器 1

SFR 页 = 仅 2 页

SFR 地址 = 0xD4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
				F8IE	MTSE	PAA	CR3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: F8IE, I2C idle 状态产生 SI.

0: 禁止, STSTA == 0xF8 (I2C idle 状态) 产生 SI (中断标志)

1: 使能 STSTA == 0xF8 (I2C idle 状态) 产生 SI (中断标志)

Bit 2: MTSE, 多速度使能 (禁止输入滤波); 定义输入信号数字滤波器的时钟源

0: 采用固定比例的输入信号进行数字滤波

1: 在 I2C 总线上使用 SYSClk x 3 的数字滤波器用于多速度的输入信号。

Bit 1: PAA, TWI0/I2C0 预回应。

0: 禁止 PAA 功能

1: TWI0/I2C0 主机 RX 和从机 TX/RX.在 DMA 传输上使能 PAA 功能

Bit 0: CR3, 与 CR2 ~ CR0 一起用于时钟速率选择

SISTA: TWI0/I2C0 状态寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0xD3

复位值 = 1111-1000

7	6	5	4	3	2	1	0
SIS7	SIS6	SIS5	SIS4	SIS3	SIS2	SIS1	SIS0
R	R	R	R	R	R	R	R

SISTA 是一个 8 位的只读寄存器。低三位总是为 0，高五位保存状态编码，可以组成多个可能的状态编码。当 SISTA 为 F8H 时，没有串行中断请求。SISTA 的其它值用于定义相应的 TWI0/ I2C0 状态。当进入这些状态的一种时，会请求串行中断(SI=1)。在 SI 硬件置位时，一个有效的状态编码会存于 SISTA 中。

另外，状态 00H 表示总线错误。当一个 START 或 STOP 信号在不符合规定的位置发送时会产生总线错误，如一个地址/数据的内部或者刚好在应答位上。

SIA2: TWI0/I2C0 2nd 地址寄存器

SFR 页 = 仅 2 页

SFR 地址 = 0xD1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
A2.6	A2.5	A2.4	A2.3	A2.2	A2.1	A2.0	A2E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~1: TWI0/I2C0 的 2nd 从机地址

Bit 0: A2E, 2nd 从机地址识别使能控制

0: 禁止 2nd 从机地址识别

1: 使能 2nd 从机地址识别

SIA2M: TWI0/I2C0 2nd 地址掩码寄存器

SFR 页 = 仅 2 页

SFR 地址 = 0xD2

复位值 = 1111-1111

7	6	5	4	3	2	1	0
A2M.6	A2M.5	A2M.4	A2M.3	A2M.2	A2M.1	A2M.0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SIA2 寄存器与 SIA2M 寄存器组合用于第二地址识别。实际上，SIA2M 是作为 SIA2 寄存器的掩码寄存器。示例如下：

$$\begin{array}{rcl}
 \text{SIA2}[7:1] & = & 1100 \quad 000 \\
 \text{SIA2M}[7:1] & = & 1111 \quad 110 \\
 \hline
 2^{\text{nd}} \text{ ADR}[7:1] & = & 1100 \quad 00x \longrightarrow \text{第二从机地址将被检测} \\
 & & \text{除了位 1 被“忽略”外}
 \end{array}$$

Bit 0: 保留位，写寄存器时，此位必须写“1”。

AUXR3: 辅助寄存器 3

SFR 页 = 仅 0 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 2~1: TWIPS1~0, TWI0/I2C0 端口选择 [1:0].

TWIPS1~0	TWI0_SCL	TWI0_SDA
0 0	P4.0	P4.1
0 1	P6.0	P6.1
1 0	P3.1	P3.0
1 1	P2.2	P2.4

22.5. TWI1/I2C1 寄存器

SI1ADR: TWI1/I2C1 地址寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xD1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
A61	A51	A41	A31	A21	A11	A01	GC1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~1: TWI1/I2C1 地址, CPU 可以直接对此寄存器进行读写。SI1ADR 不受 TWI1/ I2C1 硬件的影响。当 TWI1/ I2C1 处于主机模式时此寄存器的值会被忽略。当处于从机模式时, 寄存器的高七位必须加载本机的从机地址, 最高位与从 TWI1/ I2C1 总线上收到的首位相对应。

Bit 0:GC1, 如果 GC1 置位, 广播地址(00H)会被识别, 否则忽略。

SI1DAT: TWI1/I2C1 数据寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xD2

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SD71	SD61	SD51	SD41	SD31	SD21	SD11	SD01
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

此寄存器保存着一字节将要发送或者刚接收到的数据。在没有进行移位工作时, CPU 可以直接对此寄存器进行读写。这种情况发生在 TWI1/ I2C1 正处于一个被定义的状态并且串行中断标志位(SI1)置位。只要 SI1 被置位, SI1DAT 中的数据总是保持稳定的。在数据被移出时, 总线上的数据同时移入, SI1DAT 总保存着总线上出现的最后一个字节数据。因此在仲裁失败时, 主机发送切换为从机接收的过程会在 SI1DAT 中产生一个正确的数据。

SI1DAT 与 ACK 标志位组成一个 9 位的移位寄存器, 可以在移入或移出一个 8 位的数据后, 跟随一个应答位。ACK 标志由 TWI1 硬件控制, CPU 访问不到。串行数据在 TWI1/ I2C1_SCL 的上升沿移入 SI1DAT 寄存器。当一字节的数据完全移入 SI1DAT 后, SI1DAT 中的数据将是可用的, 并且控制逻辑会在第 9 个时钟周期返回一个应答位。串行数据在 TWI1/ I2C1_SCL 的下降沿从 SI1DAT 寄存器移出。

CPU 向 SI1DAT 写入数据后, SD71 位将首先出现在 SDA 线上。9 个时钟周期后, SI1DAT 中的 8 位数据将被发送完成, 并且通过应答位返回 ACK 标志。注意发送出去的 8 位数据会移回 SI1DAT。

SI1CON: TWI1/I2C1 控制寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xD4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CR21	ENSI1	STA1	STO1	SI1	AA1	CR11	CR01
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CPU 可以直接读写此寄存器。其中两个位会受 TWI1/I2C1 硬件的影响: SI1 位会在串行中断请求时置位, STO1 位会在总线出现 STOP 状态时清零。STO1 位也会在 ENSI1=0 时清零。

Bit 7: CR21, TWI1/ I2C1 时钟率选择位 2 (与 CR11 和 CR01 一起使用)。

Bit 6: ENSI1, TWI1/I2C1 硬件使能位。

ENSI1 为“0”时, TWI1/I2C1_SDA 与 TWI1_SCL 输出为高阻态, TWI1/I2C1_SDA 与 TWI1/I2C1_SCL 输入信号被忽略, TWI1/I2C1 处于未寻址从机状态, SI1CON 的 STO1 位被强制置为“0”, 但不影响其它位。TWI1/I2C1_SDA 与 TWI1/I2C1_SCL 可用作通用 I/O 引脚。ENSI1 为“1”时, TWI1/I2C1 使能, TWI1/I2C1_SDA 和 TWI1/I2C1_SCL 端口锁存器(比如 P4.3 和 P4.2)必须设置为逻辑 1 并且 I/O 模式必须配置成开漏模式以用于接下来的串行通讯。

Bit 5: STA1, 开始(START)标志。

当 STA1 位被置位进入主机模式时, TWI1/I2C1 硬件将检查串行总线的状态, 若总线空闲将产生一个 START 信号。若总线忙, TWI1/I2C1 将等待 STOP 信号出现并且在一个延迟后产生 START 信号。如果 STA1 在 TWI1/I2C1 已经是处于主机模式并且一个或多个字节已被发送或接收的情况下置位, TWI1/I2C1 会发送一个重复的 START 信号。STA1 可以在任何时候置位, 也可以在 TWI1/I2C1 是一个被寻址的从机时置位。当 STA1 位复位时, 无 START 或重复的 START 信号产生。

Bit 4: STO1, 停止(STOP)标志。

当 TWI1/I2C1 处于主机模式时，置位 STO1 会向串行总线发送一个 STOP 信号。当在总线上侦测到 STOP 信号时，TWI1/I2C1 硬件清除 STO1 标志。在从机模式时，置位 STO1 标志可从总线错误状态恢复。在这种情况下不会向总线发送 STOP 信号，但是 TWI1/I2C1 硬件表现就像已经接收到一个 STOP 信号，并且转换到未被寻址的从机接收模式。STOP 标志自动被硬件清零。如果 STA1 与 STO1 位同时置位，若 TWI1/I2C1 处于主机模式将产生一个 STOP 信号(当处于从机模式时将产生一个内部的 STOP 信号，但不发送)，接着发送一个 START 信号。

Bit 3: SI1, 串行中断标志。

当一个新的 TWI1/I2C1 状态出现在 SI1STA 寄存器时，SI1 标志会被硬件置位。如果 TWI1/I2C1 中断允许，中断服务程序将会运行。唯一不会使 SI1 置位的状态是指出没有相关状态信息可以获得的 F8H。当 SI1 置位时，TWI1/I2C1_SCL 线上的低电平会延长，并且串行传输暂停。TWI1/I2C1_SCL 线上的高电平不受串行中断 SI1 标志影响。SI1 必须由软件写“0”清零。SI1 标志复位时不会产生中断请求，TWI1/I2C1_SCL 线上的时钟也不会延长。

Bit 2: AA1, 确认应答标志。

如果 AA1 标志设为“1”，一个应答 ACK(TWI1/I2C1_SDA 低电平)将在 TWI1/I2C1_SCL 的应答时钟周期内回复，当：

- 1) 接收到本机的从机地址。
- 2) TWI1/I2C1 处于主机/接收模式时，接收到一字节的数据。
- 3) TWI1/I2C1 处于已被寻址的从机/接收模式时，接收到一字节的数据。

如果 AA1 标志设为“0”，一个无应答 NACK(TWI1/I2C1_SDA 高电平)将在 TWI1/I2C1_SCL 的应答时钟周期内回复，当：

- 1) TWI1/I2C1 处于主机/接收模式时，接收到一字节的数据
- 2) TWI1/I2C1 处于已被寻址的从机/接收模式时，接收到一字节的数据

Bit 7, 1~0: CR21, CR11 and CR01, 时钟率选择位

TWI1/I2C1 处于主机模式时，这三个位决定串行时钟频率。最高主机模式时钟频率可达 1MHz。当 TWI0/I2C0 处于从机模式时，时钟频率并不重要，因为 TWI1/I2C1 会自动同步任何主机的时钟频率，高达 400KHz。表 22-2..给出不同的时钟速率设置：

表 22-2. TWI1/I2C1 串行时钟速率

CR31	CR21	CR11	CR01	TWI1/I2C1 时钟选择	TWI1/I2C1 Clock 100KHz 左右		TWI1/I2C1 Clock 400KHz 左右	
					系统时钟	TWI0/I2C0 时钟	系统时钟	TWI0/I2C0 时钟
0	0	0	0	SYSClk/32	N/A	N/A	12MHz	375KHz
0	0	0	1	SYSClk/48	6M Hz	125 KHz	N/A	N/A
0	0	1	0	SYSClk/60	6M Hz	100 KHz	24M Hz	400 KHz
0	0	1	1	SYSClk/64	6M Hz	93.75 KHz	24M Hz	375 KHz
0	1	0	0	SYSClk/128	12M Hz	93.75 KHz	93.75 KHz	48M Hz
0	1	0	1	SYSClk/256	24M Hz	93.75 KHz	N/A	N/A
0	1	1	0	S0TOF/16	可变	可变	可变	可变
0	1	1	1	T0OF/16	可变	可变	可变	可变
1	0	0	0	SYSClk/80	8M Hz	100 KHz	32M Hz	400 KHz
1	0	0	1	SYSClk/90	8M Hz	88.89 KHz	36M Hz	400 KHz
1	0	1	0	SYSClk/120	12M Hz	100 KHz	48M Hz	400 KHz
1	0	1	1	SYSClk/240	24M Hz	100 KHz	N/A	N/A
1	1	0	0	SYSClk/330	32M Hz	96.97 KHz	N/A	N/A
1	1	0	1	SYSClk/480	48M Hz	100 KHz	N/A	N/A
1	1	1	0	SYSClk/512	48M Hz	93.75 KHz	N/A	N/A
1	1	1	1	SYSClk/160	16M Hz	100 KHz	N/A	N/A

注意：

1. 最大 TWI1/I2C1 时钟速率在 1MHz 以下，设置 SYSClk=8MHz 将产生 1MHz 时钟速率。
2. SYSClk 是系统时钟。
3. S0TOF 是 UART0 波特率发生器溢出。
4. T0OF 是定时器 0 溢出。

SI1CR1: TWI1/I2C1 控制寄存器 1

SFR 页 = 仅 3 页

SFR 地址 = 0xD4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
				F8IE1	MTSE1	PAA1	CR31
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: F8IE1, I2C1 状态产生 SI.

0: 禁止, STSTA == 0xF8 (I2C idle 状态) 产生 SI (中断标志)

1: 使能 STSTA == 0xF8 (I2C idle 状态) 产生 SI (中断标志)

Bit 2: MTSE1, 多速度使能 (禁止输入滤波); 定义输入信号数字滤波器的时钟源

0: 采用固定比例的输入信号进行数字滤波

1: 使用 SYSClk x 3 的数字滤波器用于输入信号。

Bit 1: PAA1, TWI0/I2C1 预回应.

0: 禁止 PAA1 功能

1: TWI1/I2C1 主机 RX 和从机 TX/RX.在 DMA 传输上使能 PAA1 功能

Bit 0: CR31, 与 CR21 ~ CR01 一起用于时钟速率选择

SI1STA: TWI1/I2C1 状态寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0xD3

复位值 = 1111-1000

7	6	5	4	3	2	1	0
SI1S7	SI1S6	SI1S5	SI1S4	SI1S3	SI1S2	SI1S1	SI1S0
R	R	R	R	R	R	R	R

SI1STA 是一个 8 位的只读寄存器。低三位总是为 0, 高五位保存状态编码, 可以组成多个可能的状态编码。当 SI1STA 为 F8H 时, 没有串行中断请求。SI1STA 的其它值用于定义相应的 TWI1/ I2C1 状态。当进入这些状态的一种时, 会请求串行中断(SI1=1)。在 SI1 硬件置位时, 一个有效的状态编码会存于 SI1STA 中。

另外, 状态 00H 表示总线错误。当一个 START 或 STOP 信号在不符合规定的位置发送时会产生总线错误, 如一个地址/数据的内部或者刚好在应答位上。

SI1A2: TWI1/I2C1 2nd 地址寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0xD1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
A21.6	A21.5	A21.4	A21.3	A21.2	A21.1	A21.0	A2E1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~1: TWI1/I2C1 的 2nd 从机地址.

Bit 0: A2E1, 2nd 从机地址识别使能控制

0: 禁止 2nd 从机地址识别

1: 使能 2nd 从机地址识别

SI1A2M: TWI1/I2C1 2nd 地址掩码寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0xD2

复位值 = 1111-1111

7	6	5	4	3	2	1	0
A2M1.6	A2M1.5	A2M1.4	A2M1.3	A2M1.2	A2M1.1	A2M1.0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SI1A2 寄存器与 SI1A2M 寄存器组合用于第二地址识别。实际上, SI1A2M 是作为 SI1A2 寄存器的掩码寄存器。示例如下:

SI1A2[7:1] = 1100 000
SI1A2M[7:1] = 1111 110
2nd ADR[7:1] = 1100 00x → 第二从机地址将被检测
除了位 1 被“忽略”外

Bit 0: 保留位，写寄存器时，此位必须写“1”。

AUXR11: 辅助寄存器 11

SFR 页 = 仅 8 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	I2C1PS1	I2C1PS0	RX1S0	0	T1CKOE	T0CKOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: I2C1PS1~0, TWI1/I2C1 端口引脚选择[1:0].

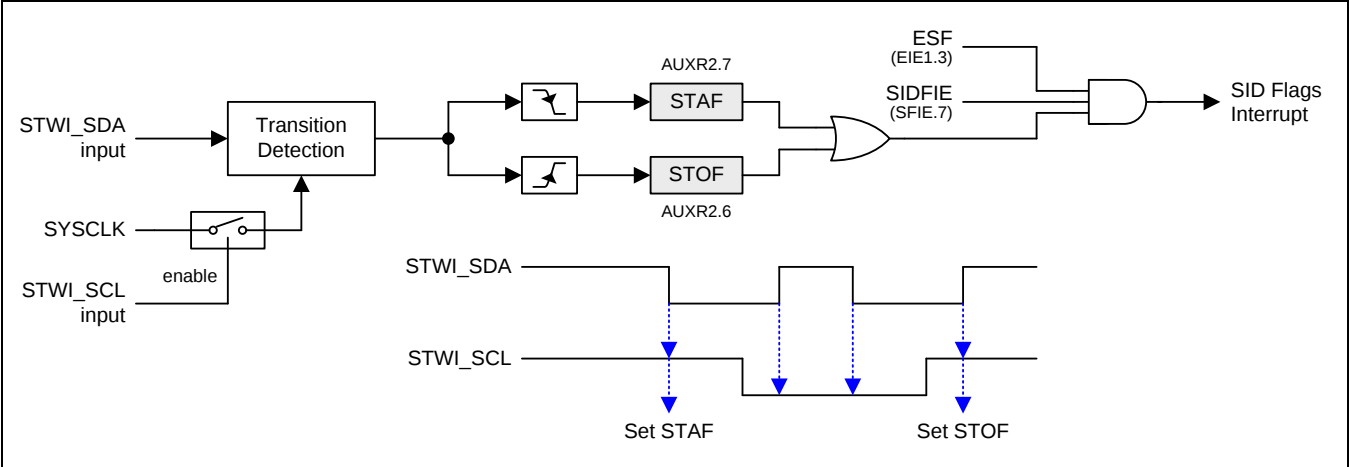
I2C1PS1~0	TWI1 SCL	TWI1 SDA
0 0	P1.0	P1.1
0 1	P6.0	P6.1
1 0	P3.0	P3.1
1 1	TWI0 SCL	TWI0 SDA

23. 串行接口侦测(STWI/SI2C)

串行接口侦测模块(SID)总是监控软件双线串行接口(STWI/SI2C)的“Start”和“Stop”状态。STWI_SCL 是串行时钟信号和 STWI_SDA 是串行数据信号。如果任何匹配条件被检测到，硬件设置 STAF 和 STOF 标志位。软件可以决定这两个标志或设置 SIDFIE (SFIE.7) 与系统标志共享中断向量。并且 STWI_SCL 位于 nINT1 将帮助 MCU 通过 nINT1 中断来判断串行数据。软件可以说使用这些资源来实施一个可变的 TWI 从机设备。

23.1. SID 结构

STAF 和 STOF 侦测的结构，中断结构和事件侦测波形如图 23-1 所示
图 23-1. 串行接口侦测结构



23.2. SID 寄存器

AUXR2: 辅助寄存器 2

SFR 页 = 0~F

SFR 地址 = 0xA3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
STAF	STOF	C1PLK	C0PLK	T1X12	T0X12	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: STAF, STWI(SID)的起始标志侦测。
0: 软件写“0”清零。STAF 有可能在上电过程中被置位，所以需要在软件初始化时将 STAF 清除。
1: 硬件置位，表示在 STWI 总线上发生了一个起始动作。

Bit 6: STOF, STWI(SID)的停止标志侦测。
0: 软件写“0”清零。STOF 有可能在上电过程中被置位，所以需要在软件初始化时将 STOF 清除。
1: 硬件置位，表示在 STWI 总线上发生了一个停止动作。

SFIE: 系统标志中断使能寄存器

SFR 页 = 0~F

SFR 地址 = 0x8E

POR = 0110-0000

7	6	5	4	3	2	1	0
SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SIDFIE, 串行接口(STWI/SI2C)侦测标志中断使能。
0: 禁止 SID 标志(STAF 或 STOF)中断。
1: 使能 SID 标志(STAF 或 STOF)中断。

AUXR9: 辅助寄存器 9

SFR 页 = 仅 6 页

SFR 地址 = 0xA4

复位值 = 0000-0000

CPR 地址 = 0x14				寄存器值 = 0000 0000			
7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G1	T0G1	0	0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

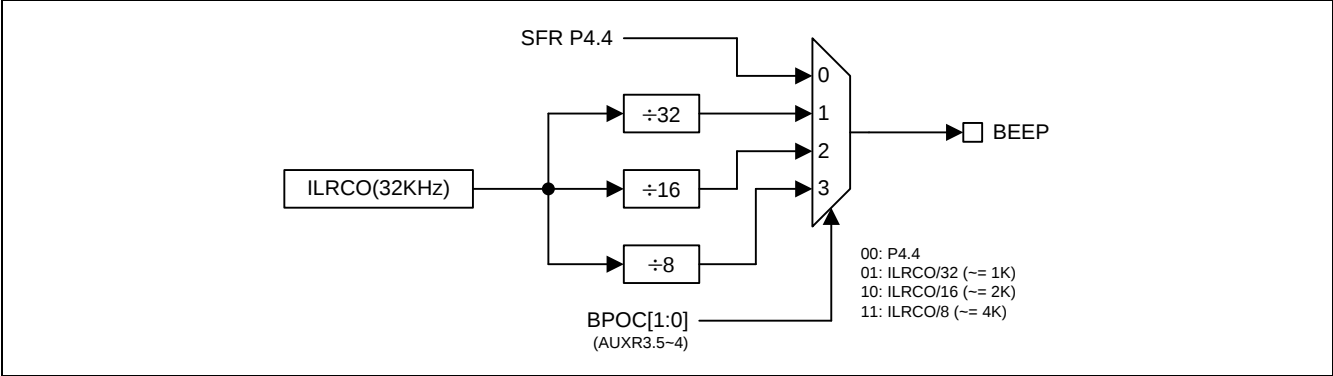
Bit 7~6: SID/STWI 端口引脚选择 [1:0].

SIDPS1~0	STWI_SCL	STWI_SDA
0 0	nINT1	SOMI
0 1	TWI0_SCL	TWI0_SDA
1 0	TWI1_SCL	TWI1_SDA
1 1	T2EXI	T3EXI

24. 蜂鸣器

蜂鸣器功能输出信号产生声音在 BEEP 引脚。信号来自 ILRCO 的分频，频率范围大约在 1、2 或 4 kHz。图 24-1 所示蜂鸣器发生器电路。但是 ILRCO 不是精确的时钟源。更详细的 ILRCO 频率偏差范围请参考章节“37.6ILRCO”。

图 24-1. 蜂鸣器发生器



24.1. 蜂鸣器寄存器

AUXR3: 辅助寄存器 3

SFR 页 = 仅 0 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: BPOC1~0, 蜂鸣器输出控制位.

BPOC[1:0]	P4.4 功能	I/O 模式
00	P4.4	By P4M0.4 & P4M1.4
01	ILRCO/32	By P4M0.4 & P4M1.4
10	ILRCO/16	By P4M0.4 & P4M1.4
11	ILRCO/8	By P4M0.4 & P4M1.4

P4.4 用于蜂鸣器功能，推荐设置 P4.4 工作在推挽输出模式。.

蜂鸣器功能使用 P4.4，在使能蜂鸣器功能之前请禁止 OCD 功能。

DCON0: 设备控制寄存器 0

SFR 页 = 仅 P 页

SFR 地址 = 0x4C

复位值 = 100x-x011

7	6	5	4	3	2	1	0
HSE	IAPO	HSE1	--GF	--GF	IORCTL	RSTIO	OCDE
R/W	R/W	R/W	W	W	W	R/W	R/W

Bit 0: OCDE, OCD 使能。

0: 在 P4.4 和 P4.5 禁止 OCD 接口。

1: 在 P4.4 和 P4.5 使能 OCD 接口。

25. 键盘中断 (KBI)

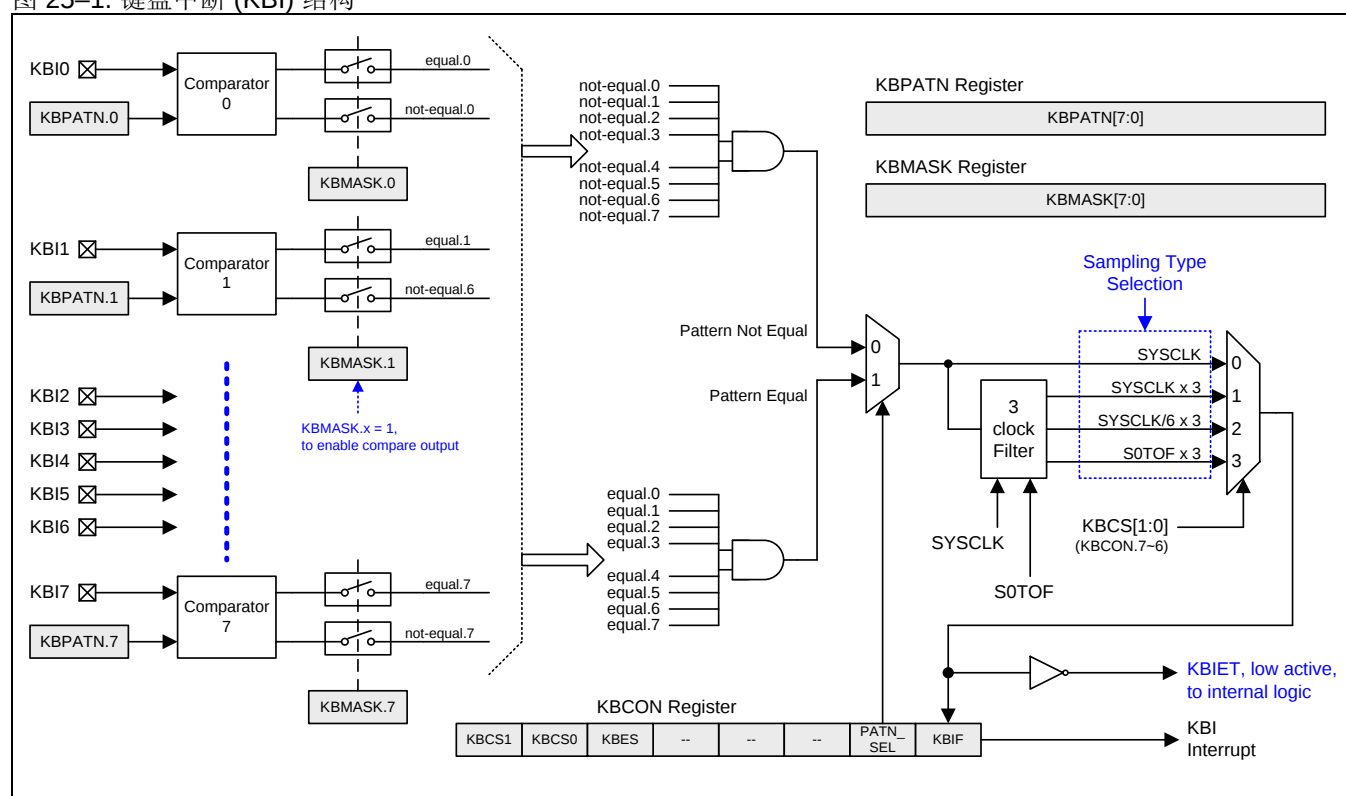
键盘中断功能主要用于当 KBI.7~0 等于或不同于某个值时产生一个中断，这个功能可以用作总线地址识别或键盘键码识别。

有 3 个特殊功能寄存器与此功能相关。键盘中断掩码寄存器(KBMASK)用来定义哪些 KBI 输入引脚可以产生中断。键盘样式寄存器(KBPATN)用来定义与键盘输入值进行比较的值，比较匹配时硬件置键盘中断控制寄存器(KBCON)中的键盘中断标志(KBIF)，若 EIE1 中的 EKBI 中断允许且 EA=1，则还会产生一个中断。键盘中断控制寄存器(KBCON)中的 PATN_SEL 位用来定义比较是“相等”还是“不等”匹配。键盘输入可以分配给不同的端口引脚，详情请参考章节“4.3 功能复用”。

为了使用键盘中断作为“键盘”中断，用户需要设置 KBPATN=0xFF 和 PATN_SEL=0(不相等)，然后将任意按键连接到 KBMASK 寄存器定义的相应端口，按下时硬件就会置位中断标志 KBIF，并当中断使能时产生中断。这个中断可以将 CPU 从空闲模式或掉电模式下唤醒。这个功能在手持设备，电池供电系统等要求低功耗而且易用的设备上特别有用。

25.1. KBI 结构

图 25-1. 键盘中断 (KBI) 结构



25.2. KBI 寄存器

下面是键盘中断(KBI)操作相关的特殊功能寄存器:

KBPATN: 键盘样式寄存器

SFR 页 = 0~F

SFR 地址 = 0xD5

复位值 = 1111-1111

7	6	5	4	3	2	1	0
KBPATN.7	KBPATN.6	KBPATN.5	KBPATN.4	KBPATN.3	KBPATN.2	KBPATN.1	KBPATN.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: KBPATN.7~0: 键盘样式, 复位值是 0xFF。

KBCON: Keypad 控制寄存器

SFR 页 = 0~F

SFR 地址 = 0xD6

复位值 = 0000-0000

7	6	5	4	3	2	1	0
KBCS1	KBCS0	KBES				PATN_SEL	KBIF
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit 7~6: KBCS1~0, KBI 滤波模式控制

KBCS1~0	KBI 输入滤波模式
00	禁止
01	SYSCLK x 3
10	SYSCLK/6 x 3
11	S0TOF x 3

Bit 5: KBES, KBI 触发模式选择。

0: 设置 KBI 模块为电平侦测模式。

1: 设置 KBI 模块为边沿侦测模式。

Bit 1: PATN_SEL, 样式匹配极性选择。

0: 键盘输入不等于 KBPATN 中用户定义的样式时产生中断。

1: 键盘输入等于 KBPATN 中用户定义样式时产生中断。

Bit 0: KBIF, 键盘中断标志。KBIF 默认值是 “1”。

0: 必须由软件写入 ‘0’ 来清零。

1: 键盘输入匹配用户定义的 KBPATN、KBMASK 和 PATN_SEL 设置条件时置位。

KBMASK: 键盘中断掩码寄存器

SFR 页 = 0~F

SFR 地址 = 0xD7

复位值 = 0000-0000

7	6	5	4	3	2	1	0
KBMASK.7	KBMASK.6	KBMASK.5	KBMASK.4	KBMASK.3	KBMASK.2	KBMASK.1	KBMASK.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

KBMASK.7: 置位时, 使能 KBI7 输入作为键盘中断源之一。

KBMASK.6: 置位时, 使能 KBI6 输入作为键盘中断源之一。

KBMASK.5: 置位时, 使能 KBI5 输入作为键盘中断源之一。

KBMASK.4: 置位时, 使能 KBI4 输入作为键盘中断源之一。

KBMASK.3: 置位时, 使能 KBI3 输入作为键盘中断源之一。

KBMASK.2: 置位时, 使能 KBI2 输入作为键盘中断源之一。

KBMASK.1: 置位时, 使能 KBI1 输入作为键盘中断源之一。

KBMASK.0: 置位时, 使能 KBI0 输入作为键盘中断源之一。

AUXR6: 辅助寄存器 6

SFR 页 = 仅 3 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
KBIHPS1	KBIHPS0	KBILPS1	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: KBIHPS1~0, KBI4~7 端口引脚选择 [1:0].

KBIHPS1~0	KBI7	KBI6	KBI5	KBI4
0 0	P1.7	P1.6	P1.5	P1.4
0 1	P1.7	P1.6	P3.5	P3.4
1 0	P2.6	P2.4	P2.3	P2.2
1 1	P1.7	P1.6	P6.1	P6.0

Bit 5~4: KBILPS1~0, KBI0~3 端口引脚选择 0.

KBILPS1~0	KBI3	KBI2	KBI1	KBI0
0 0	P1.3	P1.2	P1.1	P1.0
0 1	P3.3	P3.2	P3.1	P3.0
1 0	P0.7	P0.5	P0.2	P0.1
1 1	P4.5	P4.4	P4.1	P4.0

26. 通用逻辑(GPL-CRC)

MG82F6P32 内置一个具有 CCITT16 (CRC16 0x1021)标准的通用逻辑循环冗余检验功能。具有多种应用可编程初始值(种子值)的 8 位数据 CRC 接受流写入 CRC0DI。高字节 CRC0SH (CRCDS0~1=01)和低字节 CRC0SL (CRCDS0~1=00)设置的 16 位初始值(种子值)。结果保存在 CRC0RH (CRCDS0~1=01)和 CRC0RL (CRCDS0~1=00)。

通用逻辑循环冗余检验(GPL-CRC)还有一个通过动态检测 Flash 数据正确性的自动重载引擎直接来自 Flash 存储器的数据路径。

通用逻辑循环冗余检验(GPL-CRC)也结合数据颠倒功能。写数据到 BROVE 寄存器当从 BROVE 读取回来位序就自动颠倒了。高位在先(MSB)变成了低位在先(LSB)。

置位

置位 CRCM0 可选择 CRC32: 0x04C1_1DB7.

26.1. GPL-CRC 结构

在正常模式下，需要在 CRC0SH 和 CRC0SL 设置种子然后写数据到 CRC0DI 启动转换。

在 Flash 自动读取模式下，需要保持 CRCDS1~0 在 “0x11”。并且如下步骤所示：

- 1. 设置重载区的起始地址 IFADRH 和 IFADRL。
- 2. 结合 IAPLB(7 位)和 9 位 1-1111-1111 设置结束地址。
- 3. 设置 IFMT 寄存器(ISP/IAP Flash 模式)为 0x80 作为 Flash 自动读取模式。
- 4. 顺序将 0x46h 和 0xB9h 写入 SCMD 寄存器触发 CRC 计算。

图 26–1. CRC16 结构

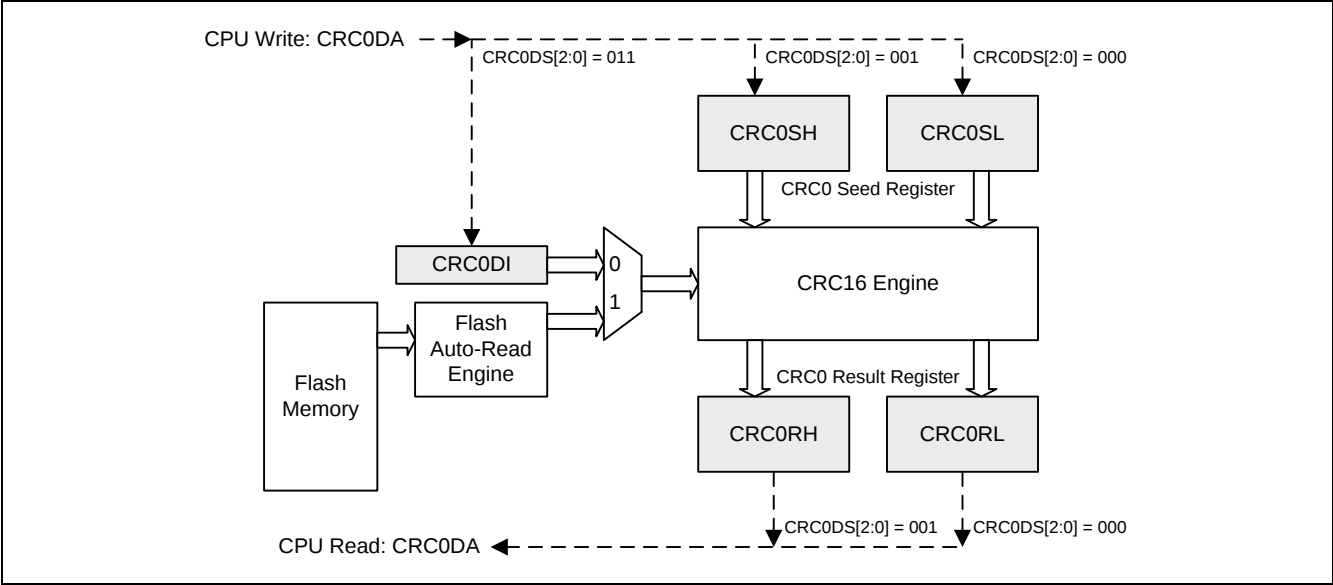
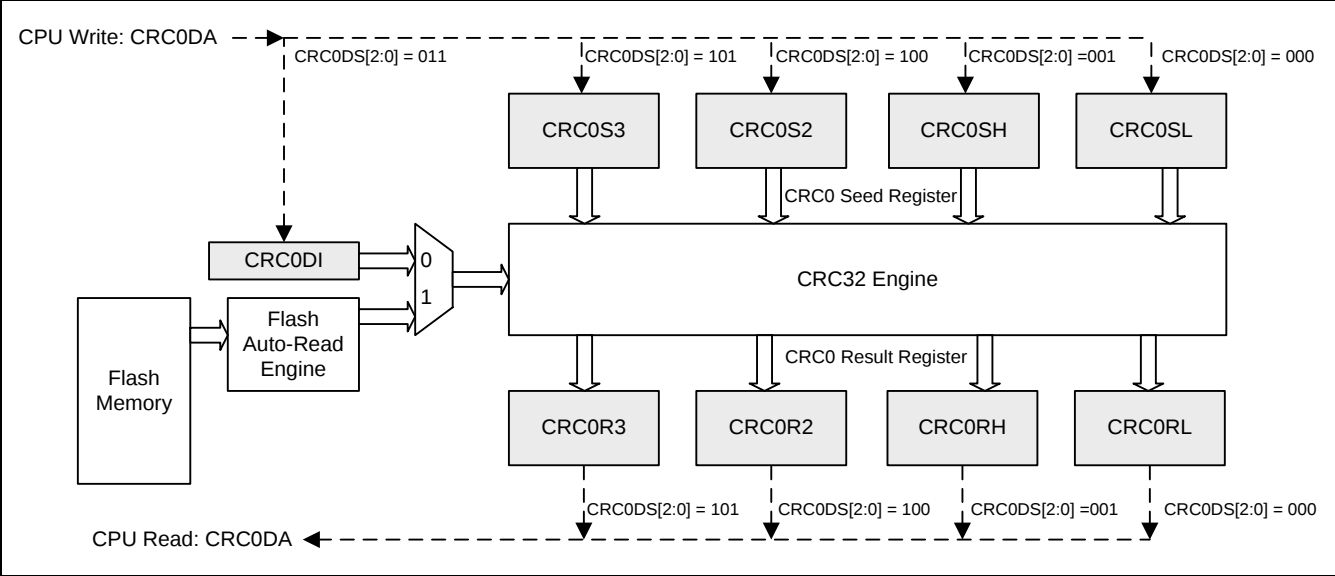
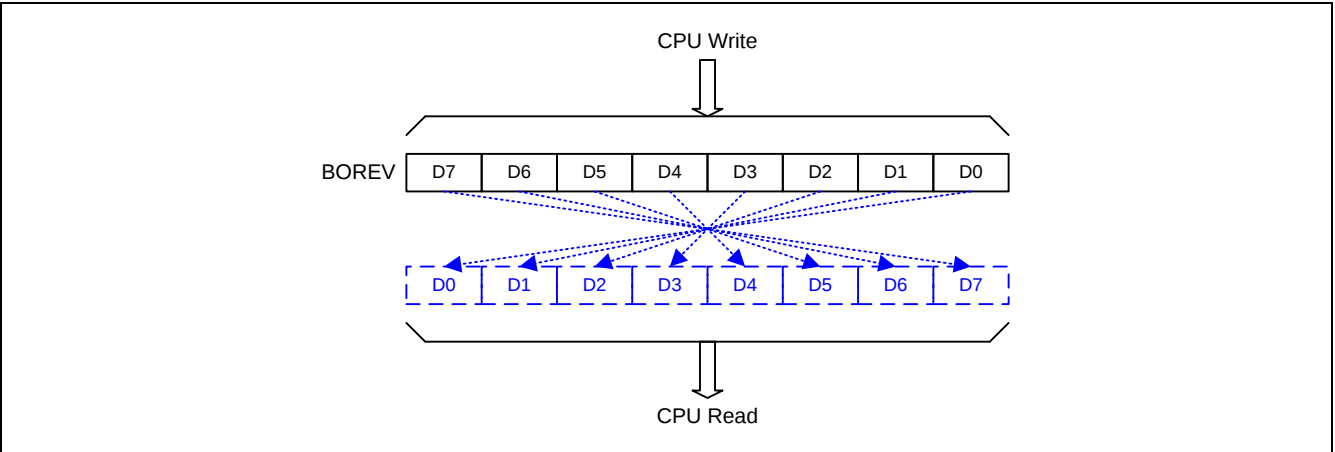


图 26–2. CRC32 结构



26.2. GPL-BOREV 结构

图 26–3. BOREV 结构



26.3. GPL-EDC45 结构

USB PD 它是使用 4 位/5 位格式的 BMC（双相标记编码）讯号进行通讯用来确保正确性。

表 26–1. 4 位/5 位格式转换表

BMC 4位/ 5位 对照表		
16进制	4 位	5 位
0	0000	11110
1	0001	01001
2	0010	10100
3	0011	10101
4	0100	01010
5	0101	01011
6	0110	01110
7	0111	01111
8	1000	10010
9	1001	10011
A	1010	10110

B	1011	10111
C	1100	11010
D	1101	11011
E	1110	11100
F	1111	11101

当要进行 USB PD 通信，可以使用 S0 来传输或接收数据。EDC45 用于 4b/5b 传输的数据读或写，与 SADDR 共享同一个地址。使用 GPLC0 来将 SFR 设定为 SADDR 或 EDC45。当 GPLC0 = 0 时，此 SFR 为 SADDR，当 GPLC0 = 1 时，该 SFR 为 EDC45。

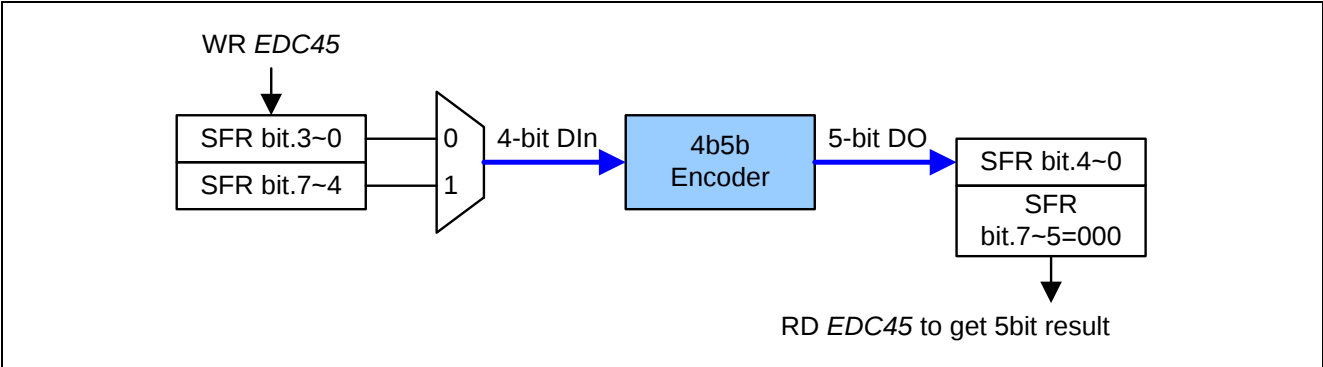
对于 USB PD 数据传输，GPLC0 应设定为 1。

GPL-EDC45 处理资料转换时，需要两笔 4 位/5 位传输数据才能完成一个传送或接收周期。

传送数据 (送出数据):

1. 设定 GPLC0 = 1，将 SFR 用作 EDC45。
2. 将 EDCM0 设定为 0，使 GPL-EDC45 操作设定为 4 位至 5 位编码器。
3. 将资料写入 SFR EDC45。
4. 读取 SFR EDC45 到 S0BUF（要使用 S0 模式 6 BMC ENDEC 模式，请参考第 20.6 节）以传输低半字节数据
5. 再次读取 SFR EDC45 到 S0BUF 以传输高半字节数据

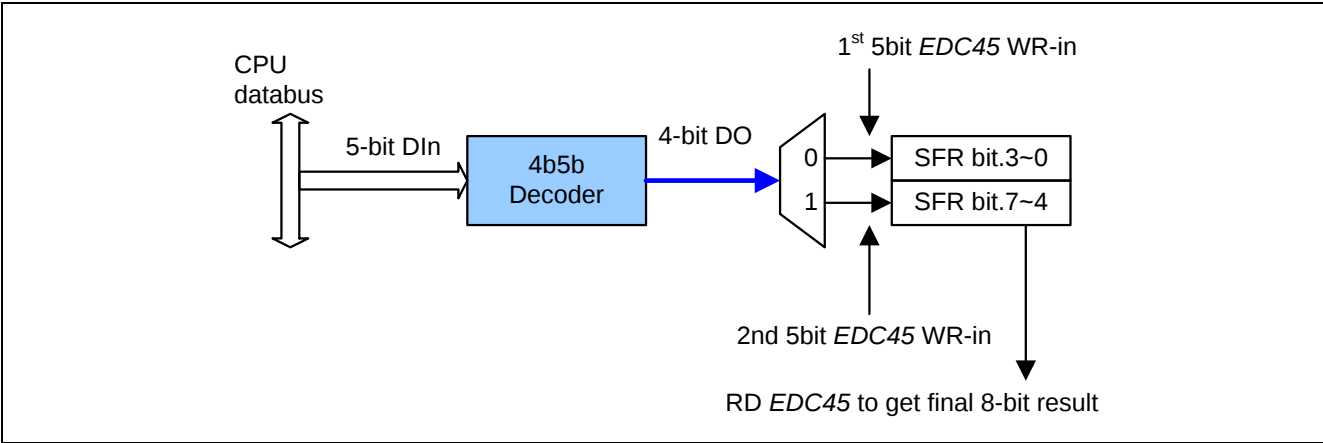
图 26–4. EDC45 模式 0 结构 (4 位到 5 位编码)



接收数据 (输入数据):

1. 设定 GPLC0 = 1，以使用 SFR EDC45。
2. 将 EDCM0 设定为 1，将 GPL-EDC45 操作设定为 5 位至 4 位译码器。
3. 将第一个数据从 S0BUF（S0 设定为模式 6 BMC ENDEC 模式，参考第 20.6 节）读取到 SFR EDC45。
4. 将第二个数据从 S0BUF 读取到 SFR EDC45。
5. 读取 SFR EDC45 以取得输入数据。

图 26–5. EDC45 模式 1 结构 (5 位到 4 位解码)



26.4. GPL 寄存器

CRC 操作相关的特殊功能寄存器如下:

CRC0DA: CRC0 数据端口

SFR 页 = 0~F

SFR 地址 = 0xB6

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CRC0DA.7	CRC0DA.6	CRC0DA.5	CRC0DA.4	CRC0DA.3	CRC0DA.2	CRC0DA.1	CRC0DA.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: CRC0 数据端口. CRC0 数据访问定义如下表:

CRCDS2~0	CPU R/W	CRC0 数据选择	描述
0 0 0	写	CRC0SL	CRC0 数据种子低位寄存器
0 0 1	写	CRC0SH	CRC0 数据种子高位寄存器.
0 1 0	写	--	保留.
0 1 1	写	CRC0DI	CRC0 数据输入寄存器
1 0 0	写	CRC0S2	CRC0 种子寄存器 2 用于 CRC32
1 0 1	写	CRC0S3	CRC0 种子寄存器 2 用于 CRC32
0 0 0	读	CRC0RL	CRC0 结果寄存器低位
0 0 1	读	CRC0RH	CRC0 结果寄存器高位
0 1 0	读	CRC0R2	CRC0 结果寄存器 2 用于 CRC32.
0 1 1	读	CRC0R3	CRC0 结果寄存器 3 用于 CRC32.
其它	读/写	--	保留

AUXR1: 辅助控制寄存器 1

SFR 页 = 0~F

SFR 地址 = 0xA2

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	CRCDS1	CRCDS0	0	0	0	DPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: CRCDS1~0. CRC0 数据端口选择位 1~0.

AUXR12: 辅助寄存器 12

SFR 页 = 仅 9 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CRCDS2	CRCM0	PDOES1	PDOES0	EDCM0	GPLC0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: CRCDS2. CRC0 数据端口选择位 2.

Bit 6: CRCM0, CRC 模式选择 0.

0: 选择 CRC16: 0x1021.

1: 选择 CRC32: 0x04C1_1DB7.

Bit 3: EDCM0. EDC45 模式选择 0.

EDCM0	Mode #	EDC45 操作
0	0	4-bit to 5-bit encoder
1	1	5-bit to 4-bit decoder

Bit 2: GPLC0. GPL 控制 0.

0: CPU 在 SFR 地址 0xA9 访问 SADDR

1: CPU 在 SFR 地址 0xA9 访问 EDC45.

BOREV: 位序颠倒数据寄存器

SFR 页 = 0~F
SFR 地址 = 0x96 复位值 = 0000-0000

7	6	5	4	3	2	1	0
BOREV.7	BOREV.6	BOREV.5	BOREV.4	BOREV.3	BOREV.2	BOREV.1	BOREV.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: BOREV7~0, 数据读/写的位序颠倒功能。
如何字节写到 BOREV 读出来位序颠倒了, 也就是写入低位在先(LSB)将变成高位在先(MSB)。例如:
在模式 0 下, 如果 0xA0 写入到 BOREV, 读回来将是 0x05。
在模式 0 下, 如果 0x01 写入到 BOREV, 读回来将是 0x80。

IFMT: ISP/IAP Flash 模式表

SFR 页 = 0~F
SFR 地址 = 0xE5 复位值 = xxxx-x000

7	6	5	4	3	2	1	0
MS.7	MS.6	MS.5	MS.4	MS.3	MS.2	MS.1	MS.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: 保留位, 写寄存器时, 此位必须写 “0000_0”。

Bit 3~0: ISP/IAP/Page-P 操作模式选择

MS[7:0]	模式
0 0 0 0-0 0 0 0	待机。
0 0 0 0-0 0 0 1	AP/IAP-存储器字节读。
0 0 0 0-0 0 1 0	AP/IAP-存储器字节编程。
0 0 0 0-0 0 1 1	AP/IAP-存储器页擦除。
0 0 0 0-0 1 0 0	P 页 SFR 写。
0 0 0 0-0 1 0 1	P 页 SFR 读。
1 0 0 0-0 0 0 0	CRC 的自动 flash 读。
1 0 0 0-0 0 0 1	Flash 字节读地址加一功能。
1 0 0 0-0 0 1 0	Flash 字节编程地址加一功能。
其它	保留。

IFMT 是用来选择闪存是用执行众多的 ISP/IAP 功能还是选择 P 页寄存器的访问。

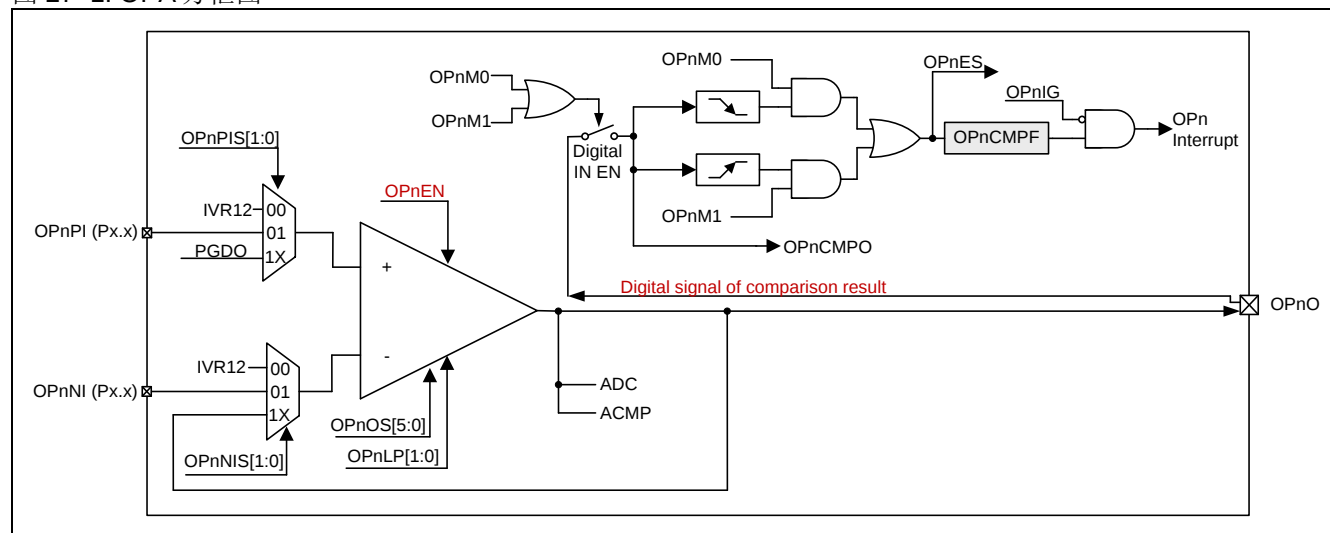
如果软件选择 CRC 的自动 flash 读模式, flash 起始地址由 IFADRH 和 IFADRL 定义。flash 结束地址在{IAPLB + 9 位 1-1111-1111}。

27. 运算放大器 (OPA0/ OPA1)

MG82F6P32 内置两个运算放大器，每个运算放大器有两个输入和一个输出。每个输入端包含三个可供选择的信号。这三个 I/O 可以连接到外部引脚，可以与任何类型的设备互连。OPA 还支持 4 级功率和比较模式。输出可连接内部 ADC 和 ACx。

27.1. OPA 结构

图 27-1. OPA 方框图



输入引脚：

OPA0 正端输入：P1.7

OPA0 负端输入：P2.2

OPA0 输出：P3.3

OPA1 正端输入：P2.4

OPA1 负端输入：P2.6

OPA1 输出：P3.0

27.2. OPA 操作

每个 OPA 都可以单独开启或关闭，并且 OPA 都有正常和低功耗两种电源模式。OPA0 和 OPA1 各有一个功率级别控制位 OPPWR。OPPWR 和 OPnLP 给出了总共 4 个功率等级可以使用

OPA 模块有两种工作模式，一种是 OPA 模式，另一种是比较(CMP)模式。在 CMP 模式下，它可以比较输入信号准位，并在 OPnCMPO 中置位比较结果。它可以用来触发中断，也可以作为一个唤醒源，在掉电时唤醒系统。

OPA 模块支持输入偏置校准以减小输入偏置。它具有工厂微调值在 MCU 初始阶段内加载。但偏移量可能根据不同的系统条件而不同，例如，VDD 电压，环境温度。用户可以将 OPA 设置为微调模式以获得优化的偏移值。

27.2.1. OPA 输入通道

两个模拟多路复用器(AMUX)选择 OPA 的正输入和负输入。正输入有三个选择，由 OPnPIS[1:0]控制，从内部带隙 1.2V 参考电压、I/O 和 PGADO 中选择信号。当与 PGA 连接时，来自 PGA 直接输出，可级联为两级系统。

负输入也有三个选择，由 OPnNIS[1:0]控制，从内部带隙 1.2V 参考电压，I/O 和 OPnO (OPA 输出)中选择信号作为单增益缓冲器。

当 OPA 模式下使用 OPA0/ OPA1，且输入或输出信号来自 I/O 时，应将这些 I/O 设置为模拟输入模式。当 OPA 关闭时，输出为高阻抗。当 OPA 处于 CMP 模式时，输出引脚将自动触发 I/O 使能数字输入路径。

27.2.2. OPA 工作模式

OPAn 有几种工作模式如 表 27-1. OPAn

表 27-1. OPAn 模式选择

OPnM1	OPnM0	模式	描述
0	0	OPA	运算放大器
0	1	CMP M0	比较模式下降沿触发
1	0	CMP M1	比较模式上升沿触发
1	1	CMP M2	比较模式双沿触发

27.2.3. OPA 功率模式

MG82F6P32 OPA 具有 4 级功率。OPPWR 是主控制寄存器同时控制 OPA0 /1 的功率级别。并且用户可以使用 OP0LP 和 OP1LP 分别获得 OPA0 和 OPA1 的低功耗模式。

表 27-2. OPAn 功率级别选择

OPPWR:OPnLP	OPAn 功耗控制
0 0	高功率,高速度
0 1	中高功率, 中高速度
1 0	中低功率, 中低速度
1 1	低功率, 低速度

27.2.4. OPA 模式

- 将负输入(OPnNI)、正输入(OPnPI)和输出(OPnO)的 I/O 模式设置为模拟输入模式
- 设置 OPnEN 使能 OPA
- 等待 OPA 稳定
- 开始使用 OPA

27.2.5. 比较模式

OPA 可以用作模拟比较模式，但其响应时间将不同于 AC0 和 AC1。
以下步骤展示了如何设置 OPA 比较模式。

- 将负输入(OPnNI)、正输入(OPnPI)的 IO 模式设置为模拟输入，输出(OPnO)自动使能数字输入路径
- 配置模式为比较模式。CMP 有 3 种模式选择不同的边沿比较结果，下降沿，上升沿和双边沿.
- 通过 OPPWR 和 OPnLP 设置 OPA 的功率模式.
- 如果想使用中断，置位 EOPn。OPnCMPF 由硬件置位，需要软件清除
- 设置 OPnEN 使能 OPA 在比较模式
- 等待 OPA 稳定
- 开始使用 OPA

27.2.6. OPA 输入偏移微调

MG82F6P32 在正常电源模式下进行了工厂调整，以降低 OPA 输入偏移电压。但是偏移量可能取决于不同的环境条件，例如 VDD 电压、环境温度和电源模式。如果用户想在特定的条件下调整偏移量，可以将 OPA 设置为比较模式，并给 NI 和 PI 相同的电压进行校准，以获得最佳的输入信号性能。

当启用 CMP 模式进行微调时，OPnO 的数字 I/O 输入路径将自动启用。在这种情况下，如果用户想要微调偏移量，则需要考虑对输出引脚的负载影响，因为微调结果将通过 I/O 引脚到数字输入显示在输出上。如果 I/O 引脚上的组件有很强的驱动或吸收电流，可能会导致从 I/O 读取时产生不正确的结果。

请按照以下步骤进行偏移校准：

方法一:对于工厂的微调模式，可以使用外部电压供电给 OPA 的 PI 和 NI；

- 将负输入(OPnNI)、正输入(OPnPI)的 IO 模式设置为模拟输入，输出(OPnO)自动使能数字输入路径
- 将 OPnNI 和 OPnPI 连接在一起，并对这两个引脚施加直流电压。直流电压应接近目标信号电压电平。
- 设置 OPnM[1:0] = 11 比较模式进行微调
- 如果要减少低功耗模式的偏移量，则需要设置 OPPWR 和 OPnLP，请参考表 27-2. OPAn
- 设置 OPnEN 使能 OPA 开始微调模式
- 等待 OPA 稳定(<5uS)
- 读 OPnO 的状态
 1. 若 OPnO = 0，增加 OPnOS[5:0]；若 OPnO = 1，则减少 OPnOS[5:0]
 2. 等待 OPnO 稳定(<600uS)
 3. 检查 OPnCMPF，如果等于 1，则表示偏移量已被修正。OPnCMPF 需要通过软件清除

方法二:对于用户的微调模式，可以使用内部电压供电给 OPA 的 PI 和 NI；

- 选择 Bandgap12 的正输入(OPnPI)，输出(OPnO)将自动启用数字输入路径
- 在这种情况下，校正偏置将基于 1.2V 作为参考电压。如果目标信号的有效电压不在此水平，则优化的偏移值可能会有微小的差异。
- 设置 OPnM[1:0] = 11 比较模式进行微调
- 如果要减少低功耗模式的偏移量，则需要设置 OPPWR 和 OPnLP，请参考表 27-2. OPAn
- 设置 OPnEN 使能 OPA 开始微调模式
- 等待 OPA 稳定(<5uS)
- 读 OPnO 的状态
 4. 若 OPnO = 0，增加 OPnOS[5:0]；若 OPnO = 1，则减少 OPnOS[5:0]
 5. 等待 OPnO 稳定(<600uS)
 6. 检查 OPnCMPF，如果等于 1，则表示偏移量已被修正。OPnCMPF 需要通过软件清除

27.2.7. Idle 和掉电模式

在掉电模式下，可以使能 OPA 在 CMP 模式下作为事件触发器唤醒系统

27.3.OPAn 寄存器

OPSTA: OPA 状态

SFR Page = 仅 0 页

SFR 地址 = 0xBA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OP1CMPF	OP0CMPF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1: OP1CMPF, OPA1 比较结果标志位

0: 该位必须通过软件在其上写“1”来清除。软件写“0”无操作。

1: 当 OPA1 处于比较模式或偏置微调模式时，当比较结果出来时，该位由硬件设置。

Bit 0: OP0CMPF, OPA0 compare result flag.

0: 该位必须通过软件在其上写“1”来清除。软件写“0”无操作。

1: 当 OPA0 处于比较模式或偏置微调模式时，当比较结果出来时，该位由硬件设置。

OP0CON0: OPA0 控制寄存器 0

SFR Page = 仅 1 页

SFR 地址 = 0xBA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
OP0EN	OPPWR	OP0LP	OP0IG	OP0PIS1	OP0PIS0	OP0NIS1	OP0NIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: OP0EN, OPA0 使能

0: 清零关闭 OPA0 模块

1: 设置打开 OPA0 模块。启用 OPA0 后至少需 5us 稳定时间。

Bit 6: OPPWR OPA0/OPA1 功率级别控制. 与 OPnLP 配合使用，适用于不同的工作功率级别。该 SFR 可同时控制 OPA0 和 OPA1 电源模式。

Bit 5: OP0LP: OPA0 低功耗模式

OPPWR:OP0LP	OPA0 功耗控制
0 0	高功率, 高速度
0 1	中高功率, 中高速度
1 0	中低功率, 中低速度
1 1	低功率, 低速度

Bit 4: OP0IG: 当启用 OPA0 比较模式时，OPA0 忽略 OP0CMPF 的比较结果以引起中断。

Bit 3 ~ 2: OP0PIS0[1:0], OPA0 正端输入信号选择

OP0PIS[1:0]	OPA0 正端输入信号选择
0 0	内部参考电压 Bandgap 1.2V (需开启 IVR24, 不保证电压精度)
0 1	OP0PI (P1.7)
1 x	PGADO

Bit 1 ~ 0: OP0NIS[1:0], OPA0 负端输入信号选择.

OP0NIS[1:0]	OPA0 负端输入信号选择
0 0	内部参考电压 Bandgap 1.2V (需开启 IVR24, 不保证电压精度)
0 1	OP0NI (P2.2)
1 x	OP0O, 工作在单增益缓冲器

OP1CON0: OPA1 控制寄存器 0

SFR Page = 仅 2 页

SFR 地址 = 0xBA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
OP1EN		OP1LP	OP1IG	OP1PIS1	OP1PIS0	OP1NIS1	OP1NIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: OP1EN, OPA1 使能

0: 清零关闭 OPA1 模块

1: 设置打开 OPA1 模块。启用 OPA10 后至少需 5us 稳定时间。

Bit 5: OP1LP, OPA1 低功耗模式

OPPWR:OP1LP	OPA1 功耗控制
0 0	高功率, 高速度
0 1	中高功率, 中高速度
1 0	中低功率, 中低速度
1 1	低功率, 低速度

Bit 4: OP1IG: 当启用 OPA1 比较模式时, OPA1 忽略 OP1CMPF 的比较结果以引起中断。

Bit 3 ~ 2: OP1PIS0[1:0], OPA1 正端输入信号选择

OP1PIS[1:0]	OPA1 正端输入信号选择
0 0	内部参考电压 Bandgap 1.2V (需开启 IVR24, 不保证电压精度)
0 1	OP1PI (P2.4)
1 x	PGADO

Bit 1 ~ 0: OP1NIS[1:0], OPA1 负端输入信号选择

OP1NIS[1:0]	OPA1 负端输入信号选择
0 0	内部参考电压 Bandgap 1.2V (需开启 IVR24, 不保证电压精度)
0 1	OP1NI (P2.6)
1 x	OP1O, 工作在单增益缓冲器

OPCMP: OPA 比较模式

SFR Page = 仅 7 页

SFR 地址 = 0xBA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	OP1PDX	OP0PDX	OP1M1	OP1M0	OP0M1	OP0M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7 ~ 6: 保留。写该寄存器时, 此位必须写“0”。

Bit 5: OP1PDX: PD 模式下 OPA1 控制

0: 在 PD 模式下编程关闭 OPA1。

1: 在 PD 模式下编程 OPA1 继续运行。

如果设置了 OP1EN、OP1PDX 和 EAC1, 则 PD 模式下的 OPA1 只能以低电平或高电平唤醒 CPU

Bit 4: OP0PDX: PD 模式下 OPA0 控制

0: 在 PD 模式下编程关闭 OPA0。

1: 在 PD 模式下编程 OPA0 继续运行。

如果设置了 OP0EN、OP0PDX 和 EAC1, 则 PD 模式下的 OPA0 只能以低电平或高电平唤醒 CPU

Bit 3~2: OP1M[1:0], OPA1 模式选择

OP0M1	OP0M0	模式	描述
0	0	OPA	运算放大器
0	1	CMP M0	下降沿触发
1	0	CMP M1	上升沿触发
1	1	CMP M2	双沿触发

Bit 1~0: OP0M[1:0], OPA0 mode selection:

OP0M1	OP0M0	模式	描述
0	0	OPA	运算放大器
0	1	CMP M0	下降沿触发
1	0	CMP M1	上升沿触发
1	1	CMP M2	双沿触发

OP0CFG0: OPA0 配置寄存器 0

SFR Page = 仅 P 页

SFR 地址 = 0x13

复位值 = 00xx-xxxx

7	6	5	4	3	2	1	0
0	0	OP0OS5	OP0OS4	OP0OS3	OP0OS2	OP0OS1	OP0OS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5 ~ 0: OP0OS[5:0] 偏置调整值

在初始化期间，将工厂调整的偏移值加载到 MCU 中。工厂的修整值在正常功率模式下被微调。如果客户希望在自定义环境下微调偏移量值，则应读取 OP0O 以了解偏移量是否已被修整。

- 如果 OP0O = 0, 增加 OP0OS[5:0]; 如果 OP0O = 1, 减小 OP0OS[5:0]
- 等待 OP0O 稳定
- 检查 OP0CMPF, 如果它被切换, 则意味着 NI 和 PI 电压是最接近的值, 这意味着偏移量已被修正。

OPnOS[5:0]	OPA 偏移调整
111111	OPAn 偏移微调基准 +31
111110	OPAn 偏移微调基准+30
.....	
100010	OPAn 偏移微调基准+2
100001	OPAn 偏移微调基准+1
100000	OPAn 偏移微调基准
011111	OPAn 偏移微调基准-1
011110	OPAn 偏移微调基准-2
.....	
000001	OPAn 偏移微调基准-31
000000	OPAn 偏移微调基准-32

OP1CFG0: OPA1 配置寄存器 0

SFR Page = 仅 P 页

SFR 地址 = 0x14

复位值 = 00xx-xxxx

7	6	5	4	3	2	1	0
0	0	OP1OS5	OP1OS4	OP1OS3	OP1OS2	OP1OS1	OP1OS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5 ~ 0: OP1OS[5:0] 偏置调整值

在初始化期间，将工厂调整的偏移值加载到 MCU 中。工厂的修整值在正常功率模式下被微调。如果客户希望在自定义环境下微调偏移量值，则应读取 OP1O 以了解偏移量是否已被修整。

- 如果 OP1O = 0, 增加 OP1OS[5:0]; 如果 OP1O = 1, 减小 OP1OS[5:0]
- 等待 OP1O 稳定
- 检查 OP1CMPF, 如果它被切换, 则意味着 NI 和 PI 电压是最接近的值, 这意味着偏移量已被修正。

OPnOS[5:0]	OPA 偏移调整
111111	OPAn 偏移微调基准 +31
111110	OPAn 偏移微调基准+30
.....	
100010	OPAn 偏移微调基准+2
100001	OPAn 偏移微调基准+1
100000	OPAn 偏移微调基准
011111	OPAn 偏移微调基准-1
011110	OPAn 偏移微调基准-2
.....	
000001	OPAn 偏移微调基准-31
000000	OPAn 偏移微调基准-32

AUXR1: 辅助控制寄存器 1

SFR Page = 0~F

SFR 地址 = 0xA2

复位值 = 0000-0000

OP1Fr	OP0Fr	CRCDS1	CRCDS0	0	AC1Fr	AC0Fr	DPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: OP1Fr. 如果 OP1CMPF 设置为 1，则 OP1Fr 也会根据硬件设置设置为 1。仅读。

Bit 6: OP0Fr. 如果 OP0CMPF 设置为 1，则 OP0Fr 也会根据硬件设置设置为 1。仅读。

Bit 2: AC1Fr. 如果 AC1F 设置为 1，则 AC1Fr 也会根据硬件设置设置为 1。仅读。

Bit 1: AC0Fr. 如果 AC0F 设置为 1, 则 AC0Fr 也会根据硬件设置设置为 1。仅读。

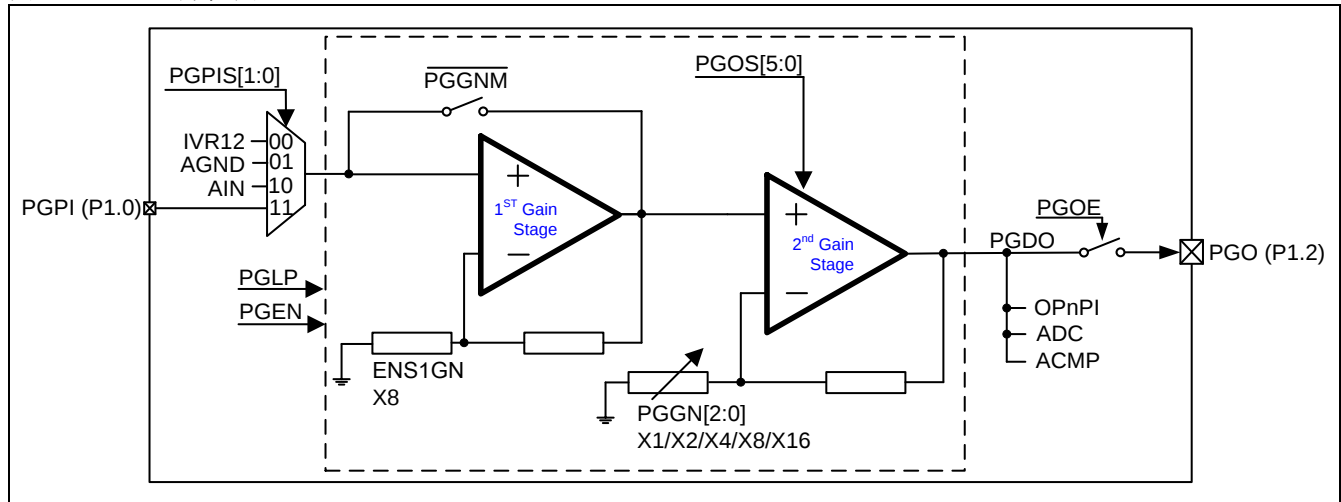
这 4 位在同一个 SFR 中，用户可以轮询这个 SFR 来同时得到这 4 个比较器的结果。

28. 可编程增益放大器, PGA

MG82F6P32 嵌入一个 PGA，它有一个正输入和一个输出。这两个 I/O 可以连接到外部引脚，可以与任何类型的设备互连。输出可以控制是否到 I/O 引脚。PGA 有两个不同增益的内部 OPA，增益可选 x1, x2, x4, x8, x16, x32, x64 和 x128。输出可连接内部 ADC、ACx、OPA0/1。

28.1. PGA 结构

图 28-1. PGA 方框图



28.2. PGA 操作

PGA 可以通过 PGEN 使能或使能，PGA 有正常和低功耗两种电源模式。

PGA 模块支持输入偏置校准，以减少输入偏置。它具有工厂修正值，将在 **MCU** 启动期间加载。但偏移量可能根据不同的系统条件而不同，例如，**VDD** 电压，环境温度。用户可以将 **PGA** 设置为修正模式以获得优化的偏移值。

28.2.1. PGA 输入通道

正输入有 4 个选择，由 PGPIS[1:0]控制，从内部 1.2V 参考电压、AGND、AIN 和 I/O 中选择信号。

表 28-1. PGA 正端输入选择

PGPIS1	PGPIS0	模式	描述
0	0	内部 1.2V 参考电压	内部 1.2V 参考电压 (需开启 IVR24, 不保证电压精度)
0	1	AGND	使用内部 AGND 用于正端输入模式.
1	0	AIN	从 ADC AMUX 输出 – AIN, 作为前置放大器与 ADC 级联使用
1	1	PGPI 引脚	I/O

28.2.2. PGA 操作

嵌入式增益用于非反相放大器

- 设置正输入(PGPI)和输出(PGO)的 I/O 模式为模拟输入模式
- 如果仅使用 PGA 作为 OPn, ADC 或 ACx 的前置放大器, 则不需要将 PGO 设置为 PGO 引脚。否则, 它应该设置 PGO 将输出分配给外部电路的 PGO 引脚。
- 配置 PGPIS0 = 1, 表示 PGPI 为正输入
- 如果增益要求低于 16, 可以使用 PGGNM = 0 跳过第一个增益阶段。

- 如果增益要求大于 16，可以设置 $PGGNM = 1$ 保持第一级增益。设 $ENS1GN = 1$ ，第 1 增益级 $\times 8$ 增益。然后通过 $PGGAIN[2:0]$ 设置第二增益级的增益，默认增益为 $\times 1, \times 2, \times 4, \times 8, \times 16$ 。结合第 1 和第 2 总最大增益高达 128。
- 配置 PGLP，如果想 PGA 在低功耗模式下使用
- 设置 PGEN 使能 PGA
- 等待 PGA 稳定
- 开始使用 PGA

28.2.3. PGA 输入偏移微调模式

MG82F6P32 在正常功率模式下对 PGA 输入偏置进行了工厂微调。但根据不同的环境条件，例如 VDD 电压、环境温度和电源模式，偏移量可能会有所不同。用户可以对 ADC 对 PGA 进行偏置微调，以获得最佳的输入信号性能。

MG82F6P32 PGA 有两个增益级，但只有第二个增益级具有偏置微调模式。每个微调步长约 0.6mV，共 64 步(+31 ~ -32)。第一阶段的偏移量可采用第二修整方法进行校准。例如，如果一级增益为 $\times 1$ ，其偏置为 +2mV，则偏置微调值可以设置在 -3 左右。如果第一阶段增益是 $\times 8$ ，那么第二个输入应该“看到”+16mV 偏移，偏移微调值可以设置在 -26 左右。如果第一级的偏移量太大，用户可以使用 ADC 记录偏移值并在软件中进行校准。

请按照以下步骤进行偏移校准：

使用微调模式：

- 设置 AIN 为 IVR 或 $1/4VDD$ ，并且使能 ADC
- 设置 $ADMINs = 0$ ，将 ADC 输入直接从 AIN 输入。
- 启动 ADC 转换并保存 ADC 结果。
- 配置 $ADMINs$ 为 1，将 ADC 输入源设置为 PGA 输出。
- 设置 $PGGNM = 0$ 以绕过第一增益级，并校准第二增益级的偏移量。
- 启动 ADC 转换，并将结果与 ADC 直接测量的值进行比较。以相反的方向调优微调代码，然后执行相同的操作，直到找到最佳解决方案。
- 更改 $PGGNM = 1$ 以启用第 1 级，并使用第 2 增益级的相同方法进行校准。
- 如果要减小低功耗模式的偏移量，则应设置为 PGLP。

28.3. PGA 寄存器

PGACON0: OPA0 控制寄存器 0

SFR Page = 仅 5 页

SFR 地址 = 0xBA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PGEN	0	PGLP	PGAOE	PGPIS1	PGPIS0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: PGEN, PGA Enable.

0: 清零关闭 PGA 模块

1: 置位开启 PGA 模块。使能后至少等待 1 μ s 稳定时间

Bit 5: PGLP, PGA 低功耗模式

0: PGA 工作在正常模式

1: PGA 工作在低功耗模式

Bit 4: PGOE, PGA 输出信号使能

0: 与 I/O 断开

1: 连接到 PGO 引脚

Bit 3 ~ 2: PGPIS[1:0], PGA 正端输入信号选择

PGPIS[1:0]	PGA 正端输入信号选择
0 0	内部 1.2V 电压参考
0 1	AGND
1 0	AIN (来自 ADC AMUX 输出)
1 1	OP0PI (P3.0)

Bit 1~0: 保留。写该寄存器时，此位必须写“0”。

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	PGENS1GN	0	PGGNM	PGGN2	PGGN1	PGGN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 保留。写该寄存器时，此位必须写“0”。

Bit 5: PGENS1GN, PGA 第一级增益控制

0: 保留.

1: PGA 增益 x8

Bit 4: 保留。写该寄存器时，此位必须写“0”。

Bit 3: PGGNM, PGA 第一级跳过控制

0: 跳过第一级 PGA,等效于设置第一级增益为 x1

1: PGA 第一级使能

Bit 2 ~ 0: PGGN[2:0], PGA 第二级增益控制.

PGGN2	PGGN1	PGGIN0	增益
0	0	0	x1
0	0	1	x2
0	1	0	x4
0	1	1	x8
1	0	0	x16
1	0	1	保留
1	1	0	保留
1	1	1	保留

PGACFG0: PGA 配置寄存器 0

SFR Page = 仅 P 页

SFR 地址 = 0x15

复位值 = 0001-1111

7	6	5	4	3	2	1	0
0	0	PGAOS5	PGAOS4	PGAOS3	PGAOS2	PGAOS1	PGAOS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 保留。写该寄存器时，此位必须写“0”。

Bit 5 ~ 0: PGAOS[5:0]微调值

- 如果有工厂调整的偏移值，将在初始化期间加载到 MCU 中。工厂微调值是在正常电源模式下调整的。如果客户想在定制环境下调整偏移值，应该读取 PGA 输出引脚 PGO 以获得当前状态。
- 如果 PGO = 0, 增加 PGAOS[5:0]; 如果 PGO = 1, 减小 PGAOS[5:0]
- 等待 PGO 稳定
- 检查 PGO, 如果它切换, 意味着 NI 和 PI 电压是最接近的值, 这意味着偏移量已被校准。

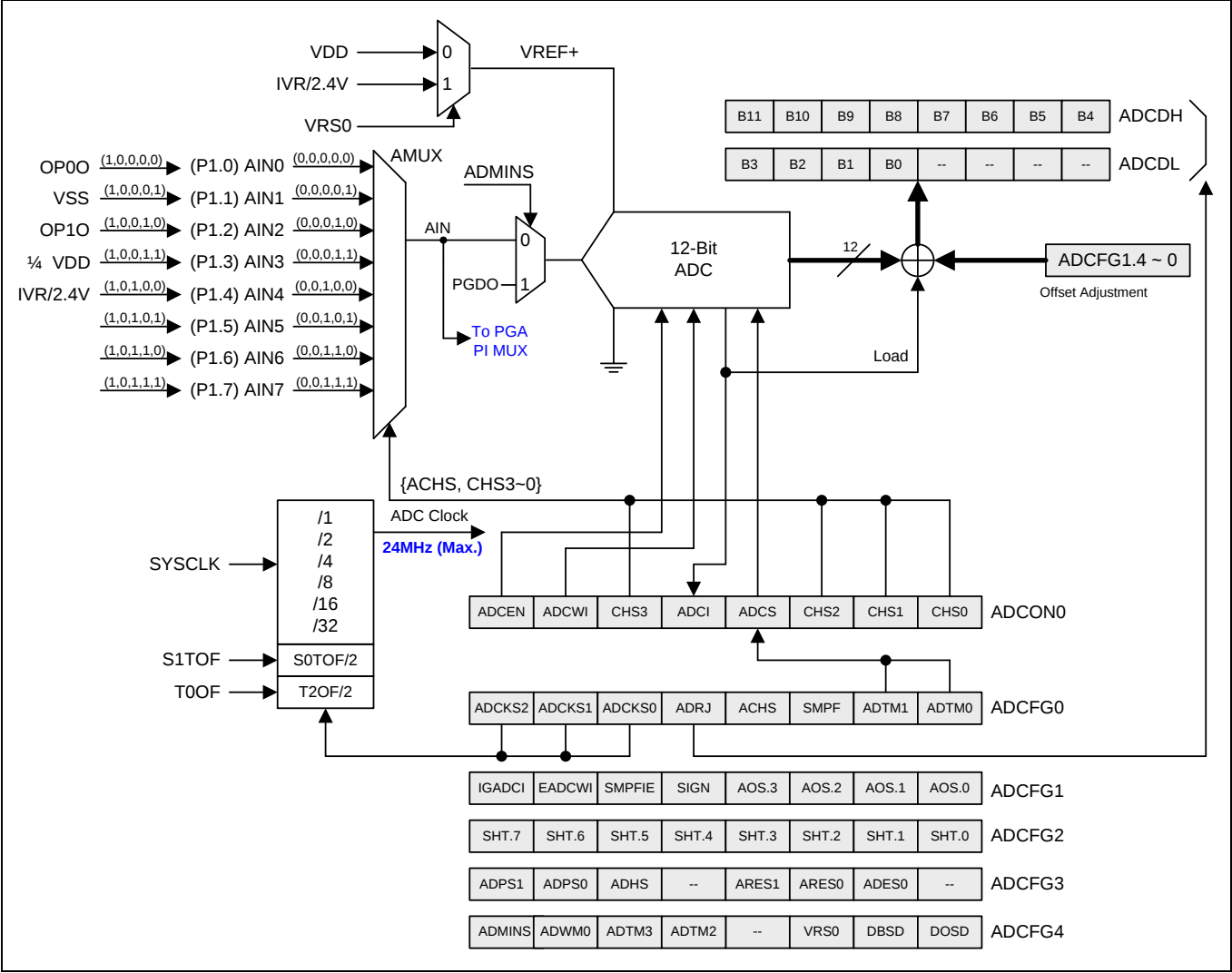
PGAOS[5:0]	PGA 偏移值调整
111111	PGAn 偏移微调基准+31
111110	PGAn 偏移微调基准+30
.....	
100010	PGAn 偏移微调基准+2
100001	PGAn 偏移微调基准+1
100000	PGAn 偏移微调基准 0
011111	PGAn 偏移微调基准-1
011110	PGAn 偏移微调基准-2
.....	
000001	PGAn 偏移微调基准-31
000000	PGAn 偏移微调基准-32

29. 12-Bit ADC

MG82F6P32 的 ADC 子系统由一个用于输入通道选择的模拟多路器(AMUX)，一个用于电压参考输入选择的 AMUX 和一个 1M(常温) sps/ 500K sps (全温度)、12 位逐次逼近型模数转换器组成。多路器(AMUX)可以通过特殊功能寄存器进行配置信道，如 图 29-1。ADC 运行在单一节点模式，并且可以配置测量 AIN0 ~ AIN7 的任何一个引脚或内部参照。仅当 ADC 控制寄存器(ADCON0)的 ADCEN 位被置逻辑 1 的时候 ADC 子系统被使能，ADCEN 设置为逻辑 0 的话 ADC 子系统关闭，将低功耗。

29.1.ADC 结构

图 29-1. ADC 方框图



29.2. ADC 操作

ADC 最大转换速度可以达到 **1M**(常温) sps/ **500K** sps (全温度)。ADC 转换时钟由 ADCFG0 寄存器的 ADCKS2~0 位决定是系统时钟分频、S0 BRG 溢出或者定时器 2 的溢出。ADC 转换时钟不能超过 **24MHz**。

转换完成后(ADCI 为 1)，转换结果从 ADC 结果寄存(ADCDH, ADCDL)中得到。作为单节点 ADC，转换结果是：

$$\text{ADC Result} = \frac{V_{\text{IN}} \times 4096}{\text{VDD Voltage}}$$

29.2.1. ADC 参考电压 VREF+选择

ADC 有两个参考电压源 VDD 和 IVR 2.4V，可由 VRS0 选择。

当使用 2.4V IVR 时，计算 LSB (least signification bit)电压会更容易。但如果用户希望获得更快的转换率，可以选择 VDD 来提供更高的电压。

29.2.2. ADC 输入通道

模拟多路复用器(AMUX)选择 ADC 的输入，允许在单端模式和一个内部参考电压(IVR, 2.4V)下测量 AIN7 ~ 0 上的任何引脚。ADC 输入通道由 ADCON0 寄存器中的 CHS3~0 和 ADCFG0 寄存器中的 ACHS 配置和选择，如图 29-1 所示。所选引脚相对于 GND 进行测量。

29.2.3. ADC 内部电压参考

默认的 ADC 参考是 VDD。如果 VDD 不是固定在某一电压，它应该读取某一电压来计算 ADC 值，从而知道该电压。

MG82F6P32 提供 3 个内部参考电压，可用于节省外部组件，并使用更少的引脚数进行其他输入/输出控制。使用 {ACHS, CHS[3:0]}选择：

1. **OP00** @ {ACHS, CHS[3:0]} = {1, 0000}, 使用 OPA0 输出作为前置放大器
2. **VSS** @ {ACHS, CHS[3:0]} = {1, 0001}, 这可以用来抵消偏置电压
3. **OP10** @ {ACHS, CHS[3:0]} = {1, 0010}, 使用 OPA 1 输出作为前置放大器
4. **¼VDD** @ {ACHS, CHS[3:0]} = {1, 0011}, 检测 VDD 或电池电压
5. **IVR (2.4V)** @ {ACHS, CHS[3:0]} = {1, 0100}, 这可用来做参考电压

参考以下步骤，使用内部电压基准计算阶跃电压(LSB 电压)：

- 1) 将模拟多路复用器(AMUX)设置为内部参考电压
- 2) 转换并存储参考电压的 ADC 值
- 3) 读回 IVR 参考值，并计算 VDD 值得到 LSB 电压
- 4) 使用 LSB 电压计算输入电压

29.2.4. 开启一次转换

在使用 ADC 功能之前，用户应：

- 1) 通过 VRS0 选择 VREF+电压
- 2) 置位 ADCEN 启动 ADC 硬件。
- 3) 通过 ADCKS2、ADCKS1 和 ADCKS0 位配置 ADC 输入时钟。
- 4) 通过位 CHS3、CHS2、CHS1 和 CHS0 选择模拟输入通道。
- 5) 配置 ADC 电压参考源
- 6) 将所选引脚配置成仅模拟输入模式。
- 7) 通过 ADRJ 位配置 ADC 转换结果输出形式。

现在，用户就可以置位 ADCS 来启动 AD 转换了。转换时间取决于 ADCKS2、ADCKS1 和 ADCKS0 位。一旦转换结束，硬件自动清除 ADCS 位，设置中断标志 ADCI，并将 12 位的转换结果按照 ADRJ 的设置存入 ADCDH 和

ADC DL。如果用户设置 ADCS 并且选择 ADC 的触发模式是 S0BRG/定时器 2 溢出或全速运行，这样 ADC 保持不断转换直到 ADCEN 清零或 ADC 配置成手动模式。

如上所述，中断标志 ADCI，由硬件设置以表明一次转换完成。因此，有两种方法检测 AD 转换是否完成：

- (1) 软件检测 ADCI 中断标志；
- (2) 设置 EIE1 寄存器 EADC 位和 IE 寄存器 EA 位使能 ADC 中断。这样，转换结束就会跳入中断服务进程。

无论(1) 或 (2)，ADCI 标志都必须在下次转换前用软件清零。

29.2.5. ADC 转换率

用户可以根据输入的模拟信号频率选择合适的转换速度。用户可以通过 ADCFG0 寄存器的 ADCKS2~0 (ADCFG0.7~5)、SHT (ADCFG2.7~0) 和 HA (ADCFG3.5) 来配置转换速率。下面公式是一个 ADC 转换的时钟个数：

用户可根据模拟输入信号的频率透过 ADHS (ADCFG3.5) 选择合适的转换速度。

该 ADC 提供 2 个传输时钟周期选项，可用于不同的最大 ADC 输入时钟速度。可使用 ADHS (ADCFG3.5) 选择如下：

1. ADHS = 0: 使用 30 个 ADC 时钟进行 ADC 转换。最大输入时钟频率应小 29.47MHz.
2. ADHS = 1: 使用 24 个 ADC 时钟进行 ADC 转换。最大输入时钟频率应小 24MHz.

用户可以通过 ADCFG0 寄存器的 ADCKS2~0 (ADCFG0.7~5)、SHT (ADCFG2.7~0) 和 HA (ADCFG3.5) 来配置转换速率。下面公式是一个 ADC 转换的时钟个数：

$$\text{ADC Conversion Rate} = \frac{\text{ADC Clock Freq.}}{(30 + X)} \quad ; X = \text{SHT}, 0 \sim 255$$

注意输入信号是交流信号(AC)， f_N ，假设采样率是 f_s ，基于奈奎斯特定理， f_s 要大于 2 倍的 f_N 确保测量的精度。

例如，

1. 为了得到 500K 的采样率：

若 $\text{SYSCLK} = 12\text{MHz}$ 并且 $\text{ADCKS} = \text{SYSCLK}$ ， $\text{SHT} = 0$ ，这样转换速率 = $12\text{MHz} / (24 + 0) = 500\text{Ksps}$ 。

(此案例，交流 (AC) 输入信号频率 f_N 要低于 250KHz 确保测量的精度。)

2. 为了得到 150K Hz 的采样率：

若 $\text{SYSCLK} = 12\text{MHz}$ 并且 $\text{ADCKS} = \text{SYSCLK}/2$ ， $\text{SHT} = 10$ ，这样转换速率 = $24\text{MHz}/4/(30+10) = 150\text{K Hz}$ 。

(案例，交流 (AC) 输入信号频率 f_N 要低于 75KHz 确保测量的精度。)

29.2.6. ADC 中断

MG82F6P32 有 3 种 ADC 中断源

- 1. ADCI, 当一个 A/D 转换完成, ADCI 置位引发一个中断。此标志中断可以被 IGADCI (ADCFG1.7)阻止。
- 2. SMPF, 当一个 ADC 通道采样和保持完成引发一个中断。此标志中断可以由 SMPFIE (ADCFG1.5)使能。
- 3. ADCWI, 在 ADC 窗口比较模式下, 当窗口比较数据匹配出现此中断标志被保持。如果此中断使能引发一个中断。此中断标志由 EADCWI(ADCFG1.6)使能。

图 29-2. ADC 中断

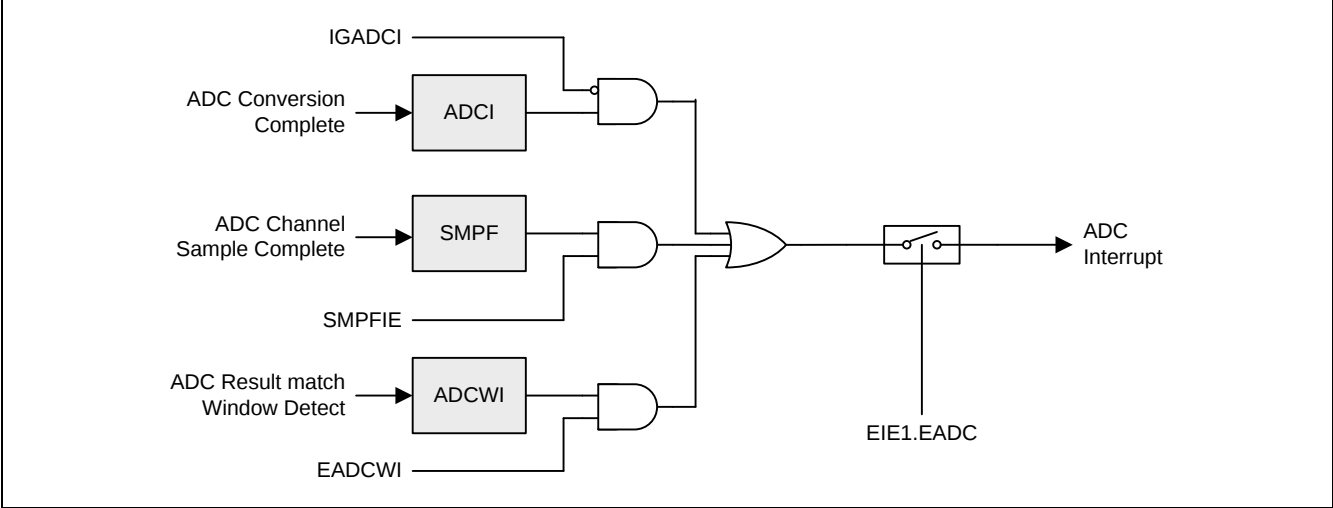
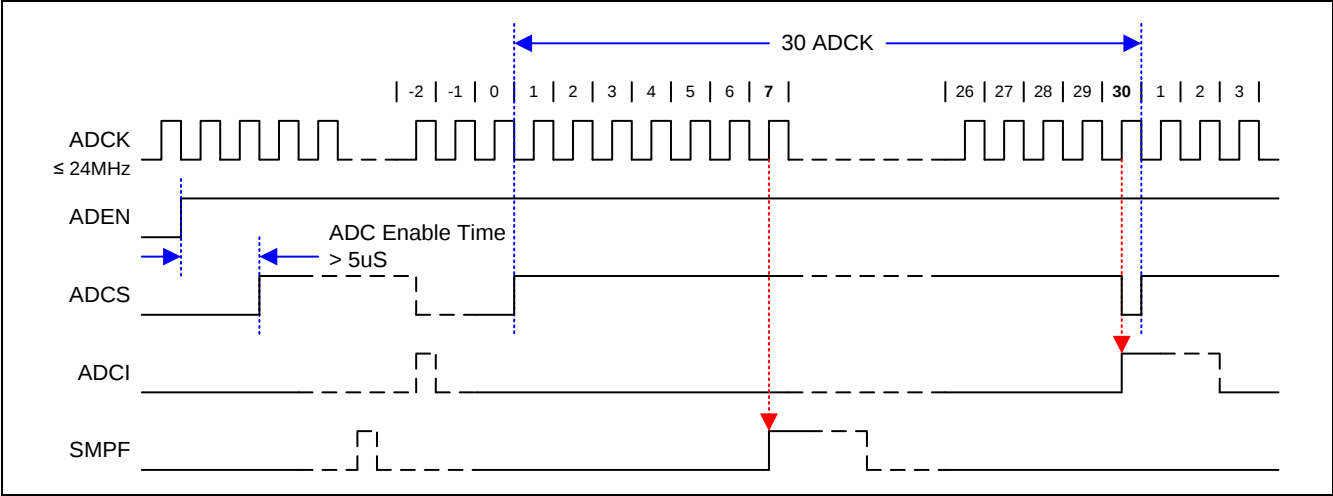


图 29-3. ADC 转换时序

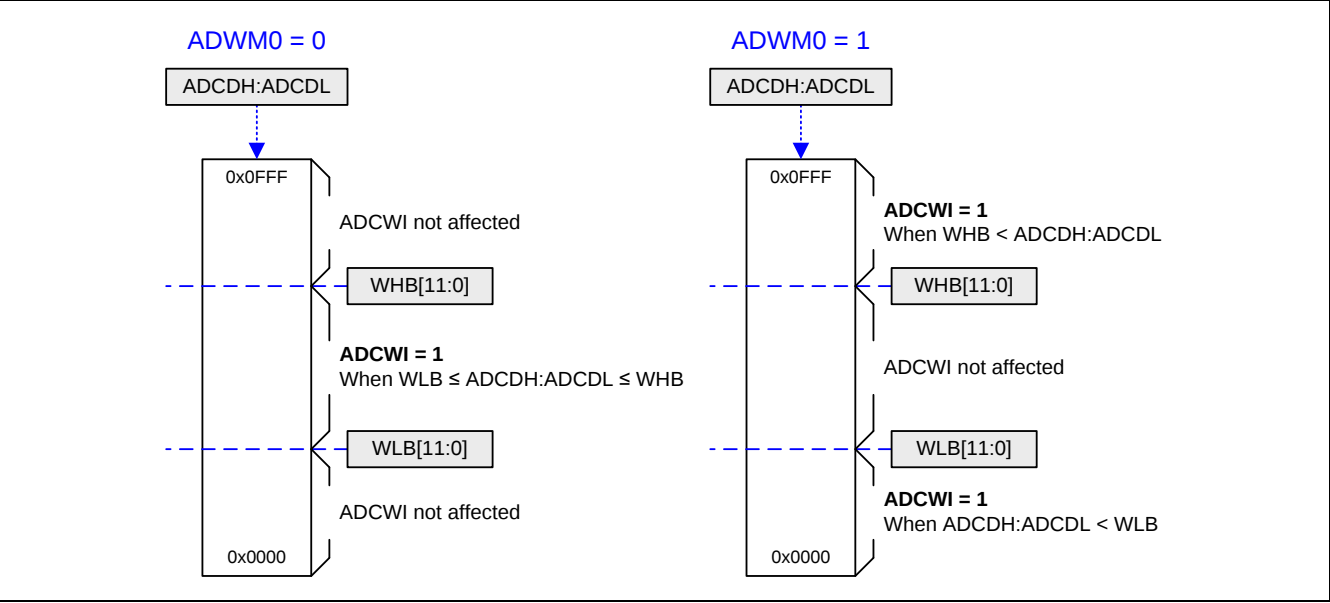


29.2.7. ADC 窗口检测

MG82F6P32 ADC 的可编程窗口侦测器根据用户设置好的阈值持续比较 ADC 输出寄存器, 侦测到期望的值将通知系统。特别是当使用中断驱动系统时可以提供更快的响应时间同时节省代码空间和减少 CPU 使用率以提高效能。窗口侦测器中断标志(ADCWI)也可以在查询模式使用。窗口高边界(WHB[11:0], {ADCFG12, ADCFG11})和低边界(WLB[11:0], {ADCFG14, ADCFG13})存放边界值。使用者可以定义想要抓取的 ADC 转换值是落在边界内或边界外。两种窗口侦测模式如下图所示:

- 1. ADWM0 = 0: 当 ADC 转换值是在边界内 ADCWI 中断标志被举起。这意味着条件 $WLB[11:0] \leq ADCDH:ADCDL \leq WHB[11:0]$ 是真, ADCWI 被举起。
- 2. ADWM0 = 1: 当 ADC 转换值是在边界外 ADCWI 中断标志被举起。这意味着条件 $WLB[11:0] > ADCDH:ADCDL$ 或 $ADCDH:ADCDL > WHB[11:0]$ 是真, ADCWI 被举起。

图 29–4. ADC 窗口检测



ADC 窗口检测的另一个应用为区别电压是大于或小于一个特定电压。

例如：

- 1.目标电压≥条件电压：ADWM0 = 0，设置条件值在 WLB 和设置 WHB = 0xFFFF。
- 2.目标电压≤条件电压：ADWM0 = 0，设置条件值在 WHB 和设置 WLB = 0。
- 3.目标电压>条件电压：ADWM0 = 1，设置条件值在 WHB 和设置 WLB = 0。
- 4.目标电压<条件电压：ADWM0 =1，设置条件值在 WLB 和设置 WHB = 0xFFFF。

29.2.8. ADC 信道扫描模式

MG82F6P32 有 8 个通道用作 ADC 输入。如果应用需要依次查看不同输入端口电压，使用 ADC 信道扫描模式容易完成和保存信道切换时间。设置 ADCFG5 的 ASCE.7 ~ ASCE.0 位指定输入通道，在 ADC 转换结束之后自动切换到下一个通道。不同的 ADC 触发模式使用此功能自动切换信道。

当信道扫描模式已启用时，通道信息将自动填入到 ADCDH 或 ADCDL 中。

表 29–1. 启用通道扫描后添加 ADC 通道信息

ADRJ	ARES[1:0] ADC Data Resolution	ADCDH	ADCDL
0	00 (12-bit)	0xFF	0xF + ADC CH Info
	01 (10-bit)	0xFF	0xC + ADC CH Info
	1x (8-bit)	0xFF	0x0 + ADC CH Info
1	00 (12-bit)	0x ADC CH Info + F	0xFF
	01 (10-bit)	0x ADC CH Info + 3	0xFF
	1x (8-bit)	0x ADC CH Info + 0	0xFF

要停止此模式，只需清除 ASCE.7 ~ ASCE.0 以禁用此功能。当 ADC 信道扫描模式开启时，请勿手动写 CHS3~0 切换通道，否则会导致意外通道被选中。如果要清除 ADCWI (ADC 窗口比较中断标志)，需要对 ADCON0 进行读修改写操作，以防止 CHS3 ~ CHS0 被修改。并且请注意，在使用此模式时，ACHS 需要为“0”，以防止内部 ADC 通道被选中

- 1) 置位 ADCEN 启动 ADC 硬件
- 2) 通过 ADCKS2、ADCKS1 和 ADCKS0 位配置 ADC 输入时钟
- 3) 将所选引脚配置成仅模拟输入模式
- 4) 通过 ADRJ 位配置 ADC 转换结果输出形式
- 5) 通过 ADCFG5 的 ASCE.7 ~ ASCE.0 选择模拟输入通道
- 6) 通过设置 ADTM [3:0]选择 ADC 触发模式

29.2.9. DMA 传输 ADC 数据

当使用 DMA 传输 ADC 数据时，需要留意下面设置：

1. DMA 控制器将先传输 ADCDL 然后 ADCDH。
2. ADRJ (ADC 结果右对齐选择)。
3. ADC 位设置：DMA 传输有 3 种位长可选，12 位、10 位和 8 位由 ARES[1:0] 选择。
4. DMA 传输 ADC 数据位。这里有 2 种 DMA 传输选择，由 DBSD 选择传输 2 字节或 1 字节。当选择 8 位，DMA 控制器自动检测传输 ADC 值保存寄存器。

例如，当 ADC 数据分辨率是 8 位模式时：

- i. ADRJ = 0 (左对齐)：ADC 值保存在 ADCDH。如果 DMA 传输模式是 8 位，则 ADCDH 将被传输。
- ii. ADRJ = 1 (右对齐)：ADC 值保存在 ADCDL。如果 DMA 传输模式是 8 位，则 ADCDL 将被传输。

例如，当 VIN = VDD 时 ADC 值是 0xFFFF，DMA 传输不同的数据组合如下：

表 29-2. 无通道扫描下 ADC DMA 数据传送顺序

ADRJ	ARES[1:0] ADC Data Resolution	DBSD ADC Data Byte Transfer by DMA	Data Transfer Order	
			1 st Data	2 nd Data
0	00 (12-bit)	0 (2-byte Data)	0xF0	0xFF
		1 (1-byte Data)	0xFF	X
	01 (10-bit)	0 (2-byte Data)	0xC0	0xFF
		1 (1-byte Data)	0xFF	X
	1x (8-bit)	0 (2-byte Data)	0x00	0xFF
		1 (1-byte Data)	0xFF	X
1	00 (12-bit)	0 (2-byte Data)	0xFF	0x 0F
		1 (1-byte Data)	0xFF	X
	01 (10-bit)	0 (2-byte Data)	0xFF	0x 03
		1 (1-byte Data)	0xFF	X
	1x (8-bit)	0 (2-byte Data)	0xFF	0x 00
		1 (1-byte Data)	0xFF	X

表 29-3. 通道扫描下 ADC DMA 数据传送顺序

ADRJ	ARES[1:0] ADC Data Resolution	DBSD ADC Data Byte Transfer by DMA	Data Transfer Order	
			1 st Data	2 nd Data
0	00 (12-bit)	0 (2-byte Data)	0xF+ADC CH Info	0xFF
		1 (1-byte Data)	0xFF	X
	01 (10-bit)	0 (2-byte Data)	0xC+ADC CH Info	0xFF
		1 (1-byte Data)	0xFF	X
	1x (8-bit)	0 (2-byte Data)	0x0 + ADC CH Info	0xFF
		1 (1-byte Data)	0xFF	X
1	00 (12-bit)	0 (2-byte Data)	0xFF	0x ADC CH Info + F
		1 (1-byte Data)	0xFF	X
	01 (10-bit)	0 (2-byte Data)	0xFF	0x ADC CH Info +3
		1 (1-byte Data)	0xFF	X
	1x (8-bit)	0 (2-byte Data)	0xFF	0x ADC CH Info +0
		1 (1-byte Data)	0xFF	X

29.2.10. I/O 端口引脚用于 ADC 功能

用作 A/D 转换的模拟输入引脚也可以保持其数字 I/O 输入输出功能。为了获得恰当的模拟性能，用作 ADC 的引脚应当禁止其数字输出，将引脚设为模拟仅输入模式。当 ADC17~0 引脚应用于模拟信号且此引脚不需要数字输入，软件设置相关引脚为仅模拟输入模式减小数字输入缓冲的功耗。模拟输入功能的端口引脚配置如“表 14-3.”所示。关于 AIN 引脚设置请参考章节“14.2 I/O”。

29.2.11. 空闲和掉电模式

在空闲和掉电模式下，若 ADC 功能打开，它将消耗一部分的电流。因此，为了降低待机和掉电模式下的功耗，可以在进入掉电和空闲模式前关闭 ADC 硬件(ADCEN =0)。

在掉电模式下，ADC 不工作。如果在空闲模式下软件触发 ADC 操作，ADC 将完成转换并置位 ADC 中断标志 ADCI。当 ADC 中断使能(EADC, EIE1.1)置位时，ADC 中断将把 CPU 从空闲模式唤醒。

29.2.12. 如何提高 ADC 精准度

使用 ADC 测量电压，其精度可能受到很多因素的影响，例如单片机 VDD 的电源噪声或基准电压的容限等。MG82F6P32 于出厂前在 VDD 等于 3.3v 的条件下校准内部参考电压-IVR，并使用 ADC 读取，将其 ADC 值并存储在 Flash 中作为预设值。用以下公式计算 AIN 电压，而不是用测量 VDD 来计算得到 1LSB 精度的电压。

- 推算 IVR 电压 (已经在 VDD=3.3V 下测量了)

$$IVR\ Voltage = \frac{IVR_{ADC_PreStored_Value} * 3300}{4096} (mV) \dots \dots \dots (1)$$

- 使用比例关系计算 I/O 引脚的电压

$$AIN\ Voltage = \frac{IVR\ Voltage * AIN_{ADC_Value}}{IVR_{ADC_Value}} (mV) \dots \dots \dots (2)$$

注: 请参考章节“31.3 如何读取 IVR (1.4V) ADC”来读取 IVR ADC 预设值。

29.3. ADC 寄存器

ADCON0: ADC 控制寄存器 0

SFR Page = 0~F

SFR 地址 = 0xC4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
ADCEN	ADCWI	CHS3	ADCI	ADCS	CHS2	CHS1	CHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: ADCEN, ADC 使能。

0: 清零而关闭 ADC 模块。

1: 开启 ADC 模块。在 ADCS 置位之前至少需要 5us 的 ADC 使能时间。

Bit 6: ADCWI, ADC 窗口比较中断标志。

0: ADC 窗口比较自最后一个标志清零之后没有匹配的数据。此标志必须软件清零。

1: ADC 窗口比较有匹配的数据此标志置位。EADCWI (ADCFG1.6)置位引发一个中断。

Bit 5: CHS3, 结合 CH2~0 选择 ADC 输入通道。

Bit 4: ADCI, ADC 中断标志。

0: 此标志必须软件清零。

1: 一次 A/D 转换完成时此标志置位, 若中断允许则还会产生一个中断。此标志中断可以被 IGADCI (ADCFG1.7)阻止。

Bit 3: ADCS, ADC 转换启动。

0: ADCS 不会被软件清零。

1: 软件置此位启动一次 A/D 转换。转换完成, ADC 硬件会自动清除 ADCS 并且 ADCI 置位。无论 ADCS 或 ADCI 为“1”时将不会开始新的 A/D 转换。

Bit 2~0: CHS2 ~ CHS0, ADC 模拟多路器输入通道选择位

单端模式:

ACHS	CHS3~0	选择通道
0	0 0 0 0	AIN0 (P1.0)
0	0 0 0 1	AIN1 (P1.1)
0	0 0 1 0	AIN2 (P1.2)
0	0 0 1 1	AIN3 (P1.3)
0	0 1 0 0	AIN4 (P1.4)
0	0 1 0 1	AIN5 (P1.5)
0	0 1 1 0	AIN6 (P1.6)
0	0 1 1 1	AIN7 (P1.7)
1	0 0 0 0	OP00
1	0 0 0 1	AVSS
1	0 0 1 0	OP10
1	0 0 1 1	1/4 VDD
1	0 1 0 0	Int. VREF (IVR/2.4V)
	其它	保留

ADCFG0: ADC 配置寄存器 0

SFR Page = 仅 0 页

SFR 地址 = 0xC3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
ADCKS2	ADCKS1	ADCKS0	ADRJ	ACHS	SMPF	ADTM1	ADTM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~5: ADC 转换时钟选择位

ADCKS[2:0]	ADC 时钟选择
0 0 0	SYSCLK
0 0 1	SYSCLK/2
0 1 0	SYSCLK/4
0 1 1	SYSCLK/8
1 0 0	SYSCLK/16
1 0 1	SYSCLK/32
1 1 0	S0TOF/2
1 1 1	T2OF/2

注意:

1. SYSCLK 是系统时钟。
2. S0TOF 是 UART0 波特率发生器溢出。
3. T2OF 是定时器 2 溢出。

Bit 4: ADRJ, ADC 结果向右对齐选择。

0: 高 8 位转换结果存在 ADCDH[7:0], 低 4 位转换结果存在 ADCDL[7:4]。

1: 高 4 位转换结果存在 ADCDH[3:0], 低 8 位转换结果存在 ADCDL[7:0]。

如果 ADRJ = 0

ADCDH: ADC 数据高字节寄存器

SFR 页 = 0~F

SFR 地址 = 0xC6

复位值 = xxxx-xxxx

7	6	5	4	3	2	1	0
(B11)	(B10)	(B9)	(B8)	(B7)	(B6)	(B5)	(B4)
R	R	R	R	R	R	R	R

ADCDL: ADC 数据低字节寄存器

SFR 页 = 0~F

SFR 地址 = 0xC5

复位值 = xxxx-xxxx

7	6	5	4	3	2	1	0
(B3)	(B2)	(B1)	(B0)	--	--	--	--
R	R	R	R	R	R	R	R

如果 ADRJ = 1

ADCDH

7	6	5	4	3	2	1	0
--	--	--	--	(B11)	(B10)	(B9)	(B8)
R	R	R	R	R	R	R	R

ADCDL

7	6	5	4	3	2	1	0
(B7)	(B6)	(B5)	(B4)	(B3)	(B2)	(B1)	(B0)
R	R	R	R	R	R	R	R

在单节点模式下, 转换结果是 12 位的无符号整数。输入的测量值从“0”到 $VDD(VREF) \times 4095/4096$ 。下表列举了向右对齐和向左对齐数据。ADCDH 和 ADCDL 寄存器没有用到的位都是“0”。

输入电压 (单节点)	ADCDH:ADCDL (ADRJ = 0)	ADCDH:ADCDL (ADRJ = 1)
$VDD \times 4095/4096$	0xFFFF	0xFFFF
$VDD \times 2048/4096$	0x8000	0x0800
$VDD \times 1024/4096$	0x4000	0x0400
$VDD \times 512/4096$	0x2000	0x0200
$VDD \times 256/4096$	0x1000	0x0100
$VDD \times 128/4096$	0x0800	0x0080
0	0x0000	0x0000

Bit 3: ACHS, ADC 辅助通道选择。结合 ACHS 和 CHS3~0 选择 ADC 输入通道。

Bit 2: SMPF, ADC 通道采样和保持标志。

0: 此标志必须软件清零。

1: 一个 ADC 通道采样和保持完成此标志置位。若中断允许则还会产生一个中断。此标志中断可以由 SMPFIE (ADCFG1.5)使能。

Bit 1~0: ADC 触发模式选择

ADTM[1:0]	ADC 转换开启选择
0 0	置位 ADCS
0 1	Timer 0 溢出
1 0	全速模式
1 1	S0 BRG 溢出

ADCFG1: ADC 配置寄存器 1

SFR 页 = 仅 1 页

SFR 地址 = 0xC3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
IGADCI	EADCWI	SMPFIE	SIGN	AOS.3	AOS.2	AOS.1	AOS.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: IGADCI, 忽略 ADCI 中断。

0: 使能 ADCI 中断。缺省是使能的。

1: 禁止 ADCI 中断。

Bit 6: EADCWI, ADCWI 中断使能。

0: 禁止 ADCWI 中断。

1: 使能 ADCWI 中断共享 ADC 中断向量。

Bit 5: SMPFIE, SMPF 中断使能。

0: 禁止 SMPF 中断。

1: 使能 SMPF 中断共享 ADC 中断向量。

Bit 4~0: SIGN 和 AOS.3~0. 寄存器值调整{ADCDH, ADCDL}中的 ADC 结果以消除偏移量。软件可以动态采集 ADC 偏置值。软件还可以将该值存储在 **MG82F6P32** IAP 区域中, 将其用作 ADC 偏移抵消的恒定参数。下表列出了 ADC 传输结果的调整值。

{Sign, AOS.[3:0]}	Value in {ADCDH, ADCDL}的值
0_1111	ADC 转换结果 + 15
0_1110	ADC 转换结果+ 14
.....	
0_0010	ADC 转换结果+ 2
0_0001	ADC 转换结果+ 1
0_0000	ADC 转换结果+ 0
1_1111	ADC 转换结果- 1
1_1110	ADC 转换结果- 2
.....	
1_0001	ADC 转换结果- 15
1_0000	ADC 转换结果- 16

ADCFG2: ADC 配置寄存器 2

SFR 页 = 仅 2 页

SFR 地址 = 0xC3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SHT.7	SHT.6	SHT.5	SHT.4	SHT.3	SHT.2	SHT.1	SHT.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: SHT[7:0], 扩展 ADC 采样时间。SHT 的值是 0~255 ADC 时钟

ADCFG3: ADC 配置寄存器 3

SFR 页 = 仅 3 页

SFR 地址 = 0xC3

复位值 = 0110-0000

7	6	5	4	3	2	1	0
ADPS1	ADPS0	ADHS	ADHS1	ARES1	ARES0	ADES0	
R/W	R/W	W	W	R/W	R/W	R/W	W

Bit 7~6: ADPS1~0, ADC 节能模式选择位 1~0

ADPS[1:0]	ADC 节能控制
0 0	高功耗, 高速度
0 1	中高功耗, 中高速度 (默认)
1 0	中低功耗, 中低速度
1 1	低功耗, 低速度

Bit 5 ~ 4: ADHS and ADHS1, ADC 速度选择

ADHS1, ADHS	ADC 速度选择
0 0	30 ADC clocks for ADC conversion
0 1	24 ADC clocks for ADC conversion
1 0	28 ADC clocks for ADC conversion
1 1	29 ADC clocks for ADC conversion

Bit 3~2: ARES1~0, ADC data Resolution selection bit 1~0.

ARES[1:0]	ADC 数据分辨率选择
0 0	12-位数据
0 1	10-位数据
1 0	8-位数据
1 1	保留

Bit 1: ADES0, ADC DMA 事件请求选择。

0: ADCI 置位请求 ADC 数据 DMA 传输。

1: ADCWI 置位请求 ADC 数据 DMA 传输。

Bit 4, 0: 保留。写该寄存器时, 此位必须写“0”。

ADCFG4: ADC 配置寄存器 4

SFR 页 = 仅 4 页

SFR 地址 = 0xC3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
ADMIN5	ADWM0	ADTM3	ADTM2	0	VRS0	DBSD	DOSD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: ADMIN5 ADC 选择从 AMUX 或 PGA 输出作为输入源。

0: ADC 输入源设置为 AIN 的 AMUX。

1: ADC 输入源设置为 PGDO, PGA 输出作为预放大器。

Bit 6: ADWM0, ADC 窗口探测器的模式选择。

0: 当 ADCDH:ADCDL 值在 WHB 和 WLB 定义的范围之内置位 ADCWI。

1: 当 ADCDH:ADCDL 值在 WHB 和 WLB 定义的范围之外置位 ADCWI。

Bit 5~4: ADC 触发模式选择位 3~2.

ADTM[3:0]	ADC 转换开启选择	输入源
0 0 0 0	置位 ADCS	软件
0 0 0 1	Timer 0 溢出(T0OF)	Timer 0
0 0 1 0	全速模式	ADC
0 0 1 1	S0 BRG 溢出(S0TOF)	S0 BRG
0 1 0 0	KBIET	KBI
0 1 0 1	INT1ET	nINT1

0 1 1 0	INT2ET	nINT2
0 1 1 1	INT3ET	nINT3
1 0 0 0	T2EXES	Timer 2
1 0 0 1	AC0ES	AC0
1 0 1 0	T3EXES	Timer 3
1 0 1 1	OP0ES	OPA0
1 1 0 0	PTM0 溢出(C0TOF)	PTM0 计数器
1 1 0 1	C0CMP5 (Note1)	TM0 CH5 比较
1 1 1 0	C1CMP0 或 C1CMP1 (Note1)	PCA1 CH0/CH1 比较
1 1 1 1	OP1ES	OPA1

Note1: C0CMPx: 参考图 17-5 & 图 17-8

Bit 3: 保留。写该寄存器时，此位必须写“0”。

Bit 2: VRS0, 电压参考, VREF+来源选择

0: VDD

1: IVR/ 2.4V

Bit 1: DBSD, DMA 传输 ADC 数据位选择

DBSD	DMA 传输 ADC 数据
0	2 字节数据
1	1 字节数据(ADRJ=0 时仅 ADCH)

Bit 0: DOSD, DMA 传输 ADC 数据字节顺序选择。

DBSD	DOSD	DMA 传输 ADC 数据字节顺序
0	0	先 ADCDL
0	1	先 ADCDH
1	0/1	ADRJ=0 时仅 ADCH

ADCFG5: ADC 配置寄存器 5

SFR 页 = 仅 5 页

SFR 地址 = 0xC3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
ASCE.7	ASCE.6	ASCE.5	ASCE.4	ASCE.3	ASCE.2	ASCE.1	ASCE.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: AIN7~AIN0 自动扫描使能。

0: 禁止 ADC 通道自动扫描。

1: 使能 ADC 通道自动扫描。

ADCFG7: ADC 配置寄存器 7

SFR 页 = 仅 7 页

SFR 地址 = 0xC3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	ASCS3	ASCS2	ASCS1	ASCS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: 保留。写该寄存器时，此位必须写“0”。

Bit 3: ASCS3 自动扫描通道选择 3。

0:使用 AIN3 作为 ADC 自动扫描通道 3

1:使用内部通道 $\frac{1}{4}$ VDD 作为 ADC 自动扫描通道 3。

Bit 2: ASCS2 自动扫描通道选择 2。

0:使用 AIN2 作为 ADC 自动扫描通道 2

1:使用内部通道 OP10 作为 ADC 自动扫描通道 2。

Bit 1: ASCS1 自动扫描通道选择 1.

0:使用 AIN1 作为 ADC 自动扫描通道 1

1:使用内部通道 VSS 作为 ADC 自动扫描通道 1.

Bit 0: ASCS0 自动扫描通道选择 0.

0:使用 AIN0 作为 ADC 自动扫描通道 0

1:使用内部通道 OP00 作为 ADC 自动扫描通道 0

ADCFG11: ADC 配置寄存器 11

SFR 页 = 仅 B 页

SFR 地址 = 0xC3

复位值 = 1111-1111

7	6	5	4	3	2	1	0
WHB.3	WHB.2	WHB.1	WHB.0	1	1	1	1
R/W	R/W	R/W	R/W	W	W	W	W

ADCFG12: ADC 配置寄存器 12

SFR 页 = 仅 C 页

SFR 地址 = 0xC3

复位值 = 1111-1111

7	6	5	4	3	2	1	0
WHB.11	WHB.10	WHB.9	WHB.8	WHB.7	WHB.6	WHB.5	WHB.4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WHB.9~0: ADC 窗口高边界值。

ADCFG13: ADC 配置寄存器 13

SFR 页 = D only

SFR 地址 = 0xC3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
WLB.3	WLB.2	WLB.1	WLB.0	0	0	0	0
R/W	R/W	R/W	R/W	W	W	W	W

ADCFG14: AD 配置寄存器 14

SFR 页 = 仅 E 页

SFR 地址 = 0xC3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
WLB.11	WLB.10	WLB.9	WLB.8	WLB.7	WLB.6	WLB.5	WLB.4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WLB.9~0: ADC 窗口低边界值。

PCON3: 电源控制寄存器 3

SFR 页 = 仅 P 页

SFR 地址 = 0x45

POR = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	R/W	W	R/W	W	W	W	W

Bit 7: IVREN, 内部电压参照使能。

0: 禁止片内 IVR (2.4V)。

1: 使能片内 IVR (2.4V)。

Bit 6 IVRPDE, IVR 可以在掉电模式下使用

0: 禁止 IVR 在掉电模式下使用

1: 使能 IVR 在掉电模式下使用

Bit 5: 保留。写该寄存器时, 此位必须写“0”。

Bit 3~0: 保留。写该寄存器时, 此位必须写“0”。

30. 模拟比较器 0/1 (AC0/AC1)

MG82F6P32 内含两个模拟比较器模块。通过比较 VIN+和 VIN-之间的电压电平，可以将模拟信号转换成数字信息。这些模块的比较结果可以发送到内部逻辑的端口引脚。

- 1. VIN+上的输入信号:
 - a. 对于 AC0:有 4 个 I/O 输入，可以由 AC0PIS[1:0]选择。
 - b. 对于 AC1: 仅有一个 I/O 输入
- 2. VIN-上有 3 种参考电压可以使用:
 - a. VDD 通过分压电阻分压成不同的电压。Table 30-1& 表 30-2 给出了详细的设置。
 - b. 从引脚 AC0NI, AC1NI 输入: 如果应用程序需要一个非常精确的电压，那么它可以在这些 I/O 引脚上使用一个精确的电压源。
 - c. IVR: 内部电压参考, 2.4V.
 - d. AC1 VIN- 可以设置 ACNI0 作为参考电压，这样就可以设置三个比较器在相同的参考电压下应用于不同的输入信号。
- 3. 结合时钟滤波器，可设定 3 种不同采样率，滤除不同噪声，减少软件去抖，提高系统整体效率。
- 4. 中断方式选择:AC0、AC1 的中断可通过上升沿、下降沿、双边沿触发。

30.1.AC0/AC1 结构

图 30-1. 模拟比较器 0 方框图

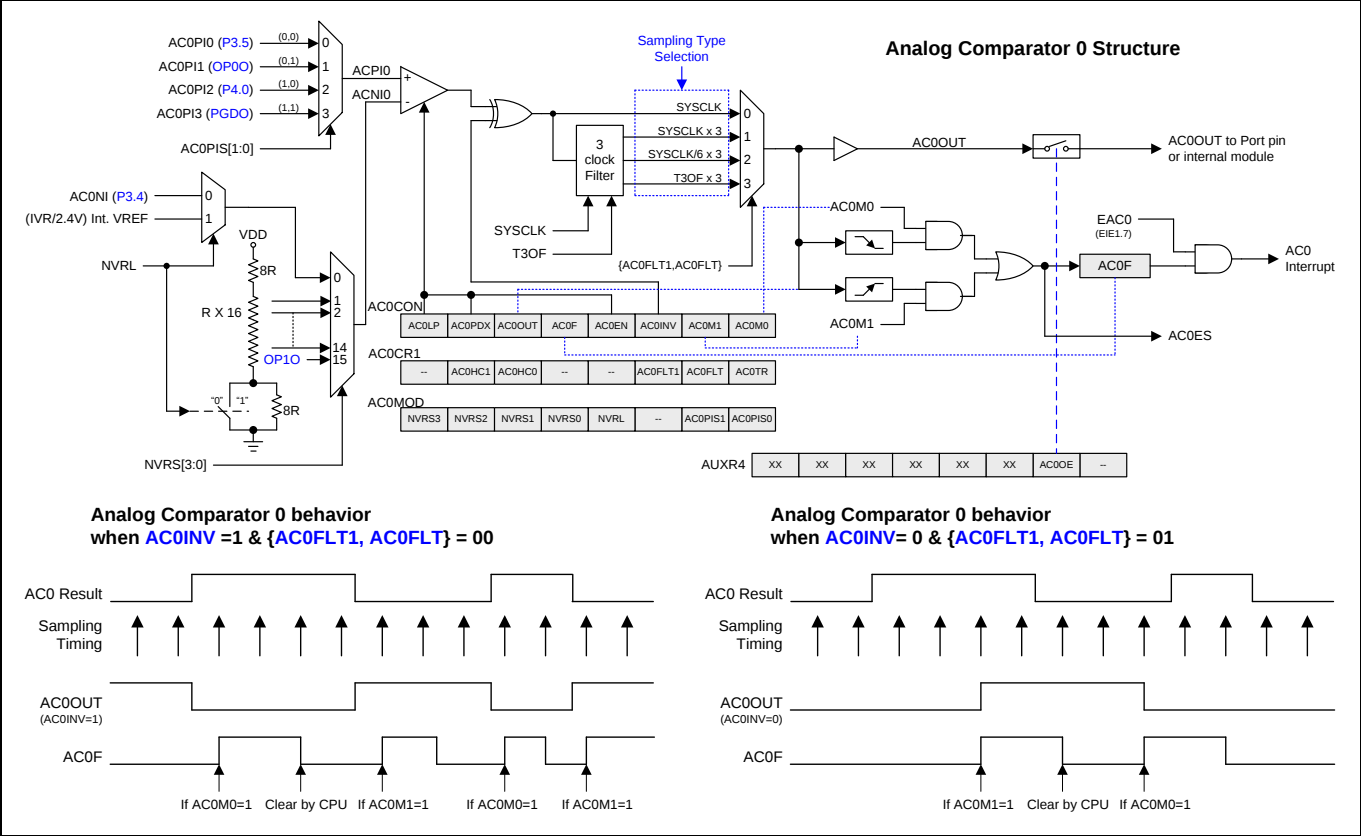
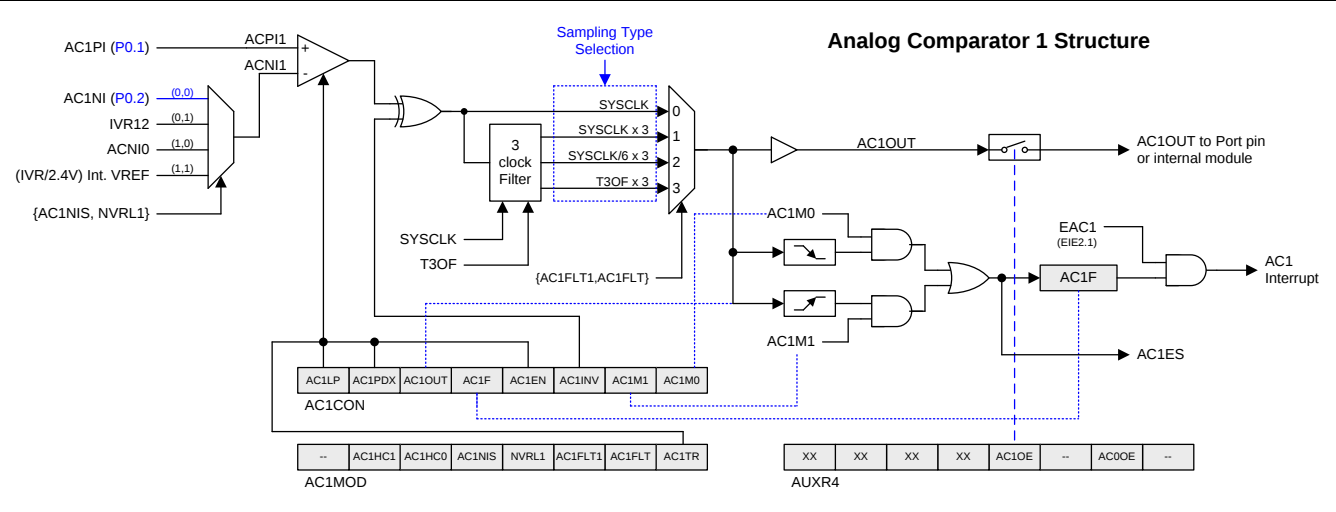


图 30-2. 模拟比较器 1 方框图



30.2. AC0/AC1 寄存器

AC0CON: 模拟比较器 0 控制& 状态寄存器

SFR Page = 仅 0 页

SFR 地址 = 0x9E

复位值 = 00X0-0000

7	6	5	4	3	2	1	0
AC0LP	AC0PDX	AC0OUT	AC0F	AC0EN	AC0INV	AC0M1	AC0M0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit 7: AC0LP, 模拟比较器 0 低功耗使能。
0:禁止 AC0 低功耗模式
1:使能 AC0 低功耗模式

Bit 6: AC0PDX, PD 模式下模拟比较器 0 控制
0: PD 模式下模拟比较器 0 关闭
1: PD 模式下模拟比较器 0 继续工作
如果 AC0EN, AC0PDX, EAC0 置位, PD 模式中的比较器只能在低电平或高电平模式下唤醒 CPU。

Bit 5: AC0OUT, 这是比较器输出的只读位

AC0 输入	AC0INV = 0	AC0INV = 1
ACPI0(+) > ACNI0(-)	AC0OUT = 1	AC0OUT = 0
ACPI0(+) < ACNI0(-)	AC0OUT = 0	AC0OUT = 1

Bit 4: AC0F. 模拟比较器 0 中断标志位
0: 这个标志位必须由软件清零。
1: 当比较器输出满足 AC0M[1:0]位所指定的条件并且 AC0EN 置位时置位。可以通过设置/清除 EIE1 的第 7 位来启用/禁用中断。

Bit 3: AC0EN. 模拟比较器 0 使能
0: 清除这个位将强制比较器输出低, 并防止进一步的事件置位 AC0F。
1: 此位置位使能比较器。

Bit 2: AC0INV, 模拟比较器 0 输出反相使能位
0: AC0 输出不反相
1: AC0 输出反相

Bit 1~0: AC0M[1:0], 模拟比较器 0 边缘触发中断模式选择

AC0M[1:0]	AC0 中断模式
0 0	保留
0 1	比较器 0 侦测输出下降沿
1 0	比较器 0 侦测输出上升沿
1 1	比较器 0 侦测输出突变

AC0MOD: 模拟比较器 0 模式寄存器

SFR 页 = 仅 0 页

SFR 地址 = 0x9F

复位值 = 0000-0000

7	6	5	4	3	2	1	0
NVRS3	NVRS2	NVRS1	NVRS0	NVRL	0	AC0PIS1	AC0PIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: NVRS[3:0], 模拟比较器 0 负端参考电压选择。如下表所示, 4 位决定模拟比较器 0 (V-)输入源:

表 30-1. AC0 参考准位 NVRL = 0, 选择高区域

NVRS[3:0]	(V-) Input	NVRS[3:0]	(V-) Input
0000	AC0NI(P3.4)	1000	16/32 VDD
0001	9/32 VDD	1001	17/32 VDD
0010	10/32 VDD	1010	18/32 VDD
0011	11/32 VDD	1011	19/32 VDD
0100	12/32 VDD	1100	20/32 VDD
0101	13/32 VDD	1101	21/32 VDD
0110	14/32 VDD	1110	22/32 VDD
0111	15/32 VDD	1111	OP1O

表 30-2. AC0 参考准位 NVRL = 1, 选择低区域

NVRS[3:0]	(V-) Input	NVRS[3:0]	(V-) Input
0000	Int. VREF (2.4V)	1000	8/24 VDD
0001	1/24 VDD	1001	9/24 VDD
0010	2/24 VDD	1010	10/24 VDD
0011	3/24 VDD	1011	11/24 VDD
0100	4/24 VDD	1100	12/24 VDD
0101	5/24 VDD	1101	13/24 VDD
0110	6/24 VDD	1110	14/24 VDD
0111	7/24 VDD	1111	Reserved

Bit 3: NVRL, 负端电压参考范围选择.

0: 选择 NVRS 在高区域

1: 选择 NVRS 在低区域

Bit 2: 保留。写该寄存器时, 此位必须写“0”。

Bit 1~0: AC0PIS[1:0], 模拟比较器 0 正端 I/O 通道选择。如下表所示, 2 位决定模拟比较器 0 (V+)输入源:

AC0PIS[1:0]	(V+)输入选择
0 0	AC0PI0(P3.5)
0 1	AC0PI1(OP0O)
1 0	AC0PI2(P4.0)
1 1	AC0PI3(PGDO)

AC0CR1: 模拟比较器 0 控制 & 状态寄存器 1

SFR Page = 仅 3 页

SFR 地址 = 0x9E

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	AC0HC1	AC0HC0	0	0	AC0FLT1	AC0FLT0	AC0TR
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留。写该寄存器时，此位必须写“0”。

Bit 6 ~ 5: AC0HC[1:0], AC0 迟滞控制

AC0HC[1:0]	输入迟滞选择
0 0	0
0 1	$\pm 10\text{mV}$
1 0	$\pm 20\text{mV}$
1 1	$\pm 60\text{mV}$

Bit 4~3: 保留。写该寄存器时，此位必须写“0”。

Bit 2 ~ 1: AC0FLT1, AC0FLT0, 模拟比较器 0 输出滤波控制。

AC0FLT1, AC0FLT0	AC0OUT 滤波模式
0 0	Disabled
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	T30F x 3

Bit 0: AC0TR, AC0 偏移电压微调模式选择

0: 正常模式

1: 偏移电压微调模式. 将自动短路 AC0 NI 和 PI, 并微调偏移值, 以检查输出切换。

AC0CF0: AC0 配置寄存器 0

SFR Page = 仅 P 页

SFR 地址 = 0x14

复位值 = 1001-1111

7	6	5	4	3	2	1	0
AC0OSE	AC0OS6	AC0OS5	AC0OS4	AC0OS3	AC0OS2	AC0OS1	AC0OS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: AC0OSE, 设置 AC0 为设置偏移模式

0: 普通模式

1: 使能 AC0 偏移校准模式。在这种模式下，它会在内部短路 NI 和 PI，用户可以在 NI 或 PI 上施加电压或使用内部参考电压进行偏置修整。

Bit 6 ~ 0: AC0OS[6:0] 偏移调整值。

在启动期间，将工厂调整的偏移值加载到 MCU 中。工厂调整值在正常电源模式下被校准。如果客户希望在自定义环境下调整偏移量值，则应该读取 AC0OUT 以了解偏移量是否已被校准。

- 如果 AC0OUT = 0, 则增加 AC0OS[6:0]; 如果 AC0OUT = 1, 则减少 AC0OS[6:0]
- 等待 AC0OUT 稳定
- 检测 AC0OUT, 如果它被切换。这意味着 NI 和 PI 电压是最接近的值，这意味着偏移量已经被校准。

ACnOS[6:0]	ACn 偏移调整值
1111111	ACn 偏移调整基值 +63
1111110	ACn 偏移调整基值 +62
.....	
1000010	ACn 偏移调整基值 +2
1000001	ACn 偏移调整基值 +1
1000000	ACn 偏移调整基值
0111111	ACn 偏移调整基值 -1

0111110	ACn 偏移调整基值-2
.....	
0000001	ACn 偏移调整基值-63
0000000	ACn 偏移调整基值-64

AC1CON: 模拟比较器 1 控制&状态寄存器

SFR Page = 仅 1 页

SFR 地址 = 0x9E

复位值 = 0000-0000

7	6	5	4	3	2	1	0
AC1LP	AC1PDX	AC1OUT	AC1F	AC1EN	AC1INV	AC1M1	AC1M0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit 7: AC1LP, 模拟比较器 1 低功耗使能..

0:禁止 AC1 低功耗模式

1:使能 AC1 低功耗模式

Bit 6: AC1PDX, PD 模式下模拟比较器 1 控制

0: PD 模式下模拟比较器 1 关闭

1: PD 模式下模拟比较器 1 继续工作

如果 AC1EN, AC1PDX, EAC1 置位, PD 模式中的比较器只能在低电平或高电平模式下唤醒 CPU。

Bit 5: AC1OUT, 这是比较器输出的只读位。

AC1 输入	AC1INV = 0	AC1INV = 1
ACPI1(+) > ACNI1(-)	AC1OUT = 1	AC1OUT = 0
ACPI1(+) < ACNI1(-)	AC1OUT = 0	AC1OUT = 1

Bit 4: AC1F. 模拟比较器 1 中断标志位

0: 这个标志位必须由软件清零。

1: 当比较器输出满足 AC1M[1:0]位所指定的条件并且 AC1EN 置位时置位。可以通过设置/清除 EIE2 的第 2 位来启用/禁用中断。

Bit 3: AC1EN. 模拟比较器 1 使能

0: 清除这个位将强制比较器输出低, 并防止进一步的事件置位 AC1F。

1: 此位置位使能比较器。

Bit 2: AC1INV, 模拟比较器 1 输出反相使能位

0: AC1 输出不反相

1: AC1 输出反相

Bit 1~0: AC1M[1:0], 模拟比较器 1 中断模式。

AC1M[1:0]	AC1 中断模式
0 0	保留
0 1	比较器 1 侦测输出下降沿
1 0	比较器 1 侦测输出上升沿
1 1	比较器 1 侦测输出突变

AC1MOD: 模拟比较器 1 模式寄存器

SFR 页 = 仅 1 页

SFR 地址 = 0x9F

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	AC1HC1	AC1HC0	AC1NIS	NVRL1	AC1FLT1	AC1FLT0	AC1TR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留。写该寄存器时, 此位必须写“0”。

Bit 6 ~ 5 : AC1HC[1:0], AC1 迟滞控制

AC1HC[1:0]	输入迟滞控制
------------	--------

0 0	0
0 1	$\pm 10\text{mV}$
1 0	$\pm 20\text{mV}$
1 1	$\pm 60\text{mV}$

Bit 4~3: AC1NIS & NVRL1, 模拟比较器 1 负端输入选择

AC1NIS, NVRL1	ACNI1 选择	输入源描述
0 0	ACNI1 端口引脚	P0.2
0 1	IVR12	内部参考电压 1.2V
1 0	ACNI0	模拟比较器 0 负端输入
1 1	IVR24	内部参考电压 2.4V

Bit 2 ~ 1: AC1FLT1, AC1FLT0, 模拟比较器 1 输出滤波控制

AC1FLT1, AC1FLT0	AC1OUT 滤波模式
0 0	禁止
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	T3OF x 3

Bit 0: AC1TR, AC1 偏移电压微调模式选择

0: 正常模式

1: 偏移电压微调模式. 将自动短路 AC1 NI 和 PI, 并微调偏移值, 以检查输出切换。

AC1CF0: AC1 配置寄存器 0

SFR 页 = 仅 P 页

SFR 地址 = 0x15

RESET = 1001-1111

7	6	5	4	3	2	1	0
AC1OSE	AC1OS6	AC1OS5	AC1OS4	AC1OS3	AC1OS2	AC1OS1	AC1OS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: AC1OSE, 设置 AC1 为设置偏移模式

0: 普通模式

1: 使能 AC1 偏移校准模式。在这种模式下, 它会在内部短路 NI 和 PI, 用户可以在 NI 或 PI 上施加电压或使用内部参考电压进行偏置修整。

Bit 6 ~ 0: AC1OS[6:0] 偏移调整值。

在启动期间, 将工厂调整的偏移值加载到 MCU 中。工厂调整值在正常电源模式下被校准。如果客户希望在自定义环境下调整偏移量值, 则应该读取 AC1OUT 以了解偏移量是否已被校准。

- 如果 AC1OUT = 0, 则增加 AC1OS[6:0]; 如果 AC1OUT = 1, 则减少 AC1OS[6:0]
- 等待 AC1OUT 稳定
- 检测 AC1OUT, 如果它被切换。这意味着 NI 和 PI 电压是最接近的值, 这意味着偏移量已经被校准。

ACnOS[6:0]	ACn 偏移调整值
1111111	ACn 偏移调整基值 +63
1111110	ACn 偏移调整基值+62
.....	
1000010	ACn 偏移调整基值+2
1000001	ACn 偏移调整基值 e +1
1000000	ACn 偏移调整基值
0111111	ACn 偏移调整基值-1
0111110	ACn 偏移调整基值-2
.....	
0000001	ACn 偏移调整基值-63
0000000	ACn 偏移调整基值-64

AUXR4: 辅助寄存器 4

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	0	AC0OE	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3: AC1OE, 使能 AC1OUT 输出在端口引脚上。

0: 禁止 AC1OUT 输出在端口引脚上。

1: 使能 AC1OUT 输出在 P0.5 上

Bit 1: AC0OE, 使能 AC0OUT 输出在端口引脚上。

0: 禁止 AC0OUT 输出在端口引脚上。

1: 使能 AC0OUT 输出在 P4.5 上

PCON3: 电源控制寄存器 3

SFR 页 = 仅 P 页

SFR 地址 = 0x45

POR = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	R/W	W	R/W	W	W	W	W

Bit 7: IVREN, 内部电压参照使能。

0: 禁止片内 IVR (2.4V)。

1: 使能片内 IVR (2.4V)。

Bit 6 IVRPDE, IVR 可以在掉电模式下使用

0: 禁止 IVR 在掉电模式下使用

1: 使能 IVR 在掉电模式下使用

Bit 5: 保留。写该寄存器时, 此位必须写“0”。

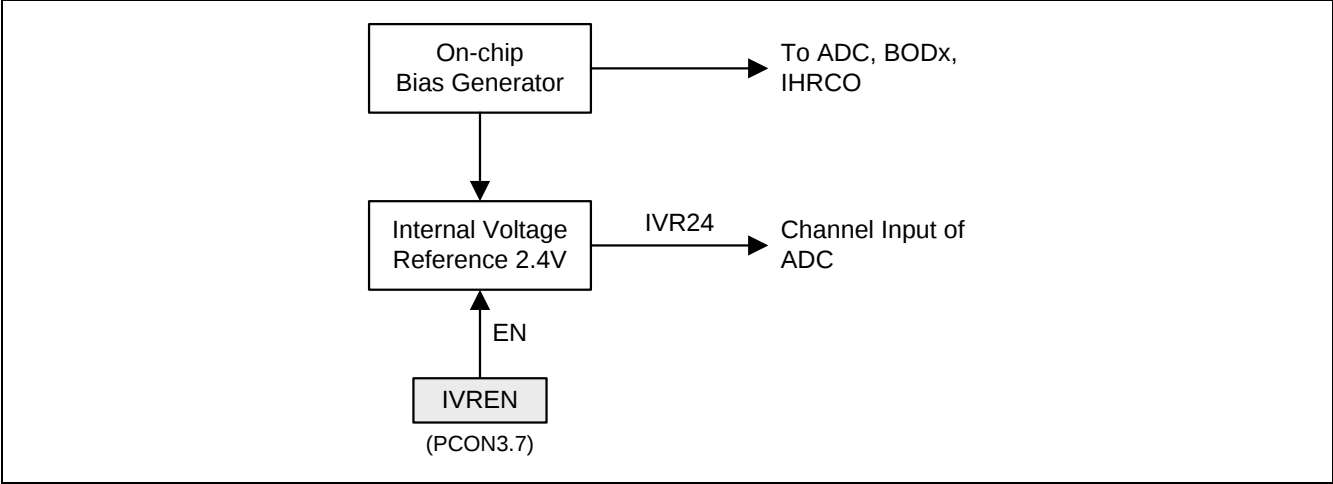
Bit 3-0: 保留。写该寄存器时, 此位必须写“0”。

31. 内部电压参考 (IVR, 2.4V)

可以将 IVR 作为 AC0 和 ADC 的参考电压。它典型的输出是 2.4V，可以由 IVREN 禁用。

31.1. IVR (2.4V)结构

图 31–1. IVR 方框图



31.2. IVR 寄存器

PCON3: 电源控制寄存器 3

SFR 页 = 仅 P 页

SFR 地址 = 0x45

POR = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	W	W	R/W	W	W	W	W

Bit 7: IVREN，内部电压参照使能。

0: 禁止片内 IVR (2.4V)。

1: 使能片内 IVR (2.4V)。

Bit 6 IVRPDE, IVR 可以在掉电模式下使用

0: 禁止 IVR 在掉电模式下使用

1: 使能 IVR 在掉电模式下使用

Bit 5: 保留。写该寄存器时，此位必须写“0”。

Bit 3~0: 保留。写该寄存器时，此位必须写“0”。

31.3. 如何读取 IVR (1.4V) ADC 预设值

IVR 在工厂已经被校准@VDD=3.3V. 并将其 ADC 值存储在 Flash 的预留区域内，供用户根据 ADC 值计算电压值。这意味着用户不需要在产线上对 ADC 进行校准。它可以节省测试时间和成本。请参考下列代码读取预存储的 IVR ADC 值，并参考章节“[29.2.1 ADC 参考电压 VREF+选择](#)”来了解如何提高 ADC 的测量精度。

```
void Get_Prestored_IVR(void)
{
    ISPCR = ISP_ENABLE;
    IFMT = 0x06;
    IFADRH = 0x00;
    IFADRL = 0xC0;

    SCMD = 0x46;
    SCMD = 0xB9;
    Trim_IVR_ADC_Value.B[0] = IFD;
    IFADRL ++;

    SCMD = 0x46;
    SCMD = 0xB9;
    Trim_IVR_ADC_Value.B[1] = IFD;

    ISPCR = ISP_DISABLE;
}
```

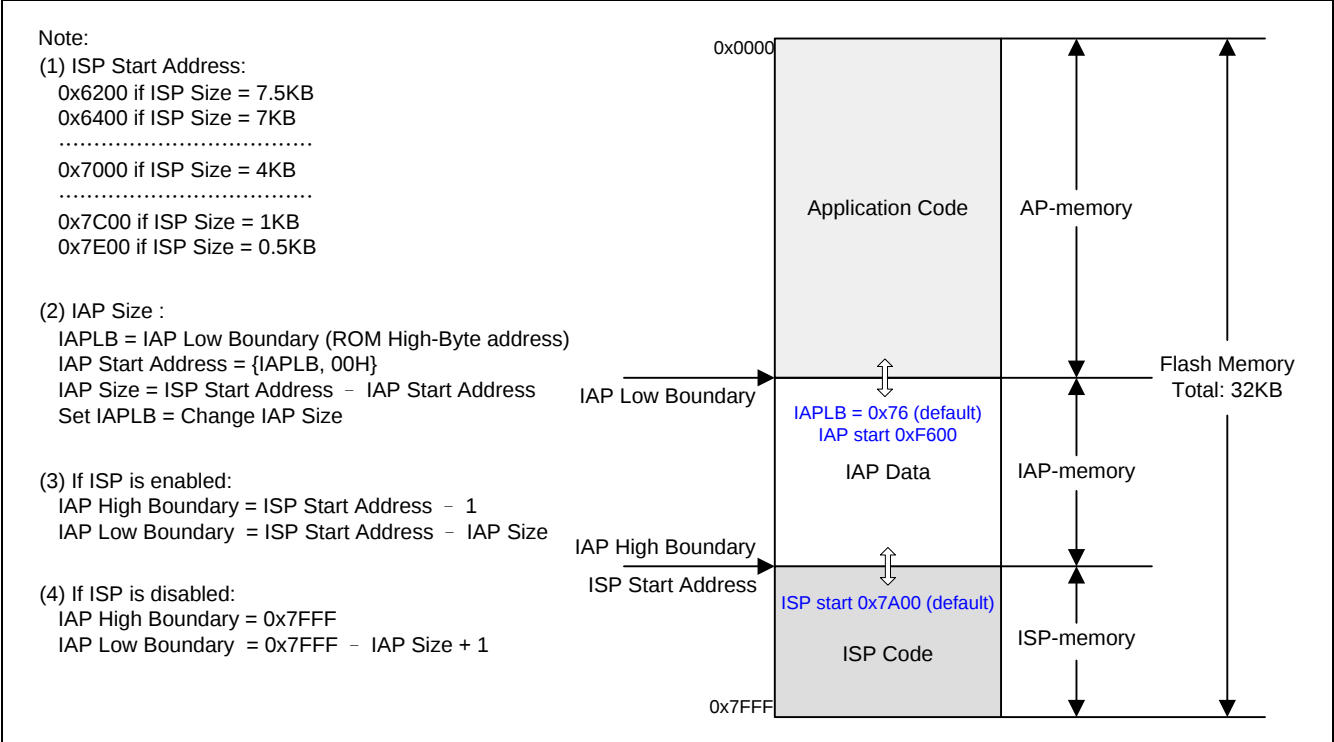
32. ISP 和 IAP

MG82F6P32 的 Flash 存储器分区为 AP-存储器, IAP-存储器和 ISP-存储器。AP-存储器用于存放用户的应用程序。IAP 用于存放非易失性应用数据, ISP-存储器用于储存在系统编程的引导程序。当 MCU 运行在 ISP 区域时, MCU 可以修改 AP 和 IAP 存储器用于程序更新。如果 MCU 运行在 AP 区域, 软件仅能修改 IAP 存储器用于更新应用数据。

32.1. MG82F6P32 Flash 存储器配置

MG82F6P32 总共有 32K 字节的 Flash, MG82F6P32 的 Flash 配置如图 32-1 所示。ISP 存储空间可以被禁止或由硬件选项 0.5KB 步距配置最大 7.5K 字节。IAP 存储空间大小由 IAP 低边界和高边界决定。IAP 低边界由 IAPLB 寄存器的值决定。IAP 高边界与 ISP 的起始地址相关, ISP 存储空间由硬件选项决定。IAPLB 寄存器值由硬件选项配置或 AP 软件编程设定。所有 AP、IAP 和 ISP 存储空间共享总 32K 字节的存储空间。

图 32-1. 6P32 Flash 存储器配置



注意:
笙泉公司 MG82F6P32 的默认 flash 存储器配置是: 1.5K ISP、1K IAP 和加密。1.5K ISP 区域是嵌入有笙泉专利的 COMBO ISP 代码通过一条线就能在线下载的 1-线 ISP 协议及串口(COM)ISP 协议。1K IAP 大小可以通过应用程序软件来重新配置。

32.2. MG82F6P32 Flash 在 SP/IAP 上的访问

MG82F6P32 给 ISP 和 IAP 应用提供三种 flash 访问模式：页擦除模式，字节编程模式及读取模式。MCU 软件使用这三种模式去更新 Flash 的数据和获取 Flash 的数据。本章展示了不同 Flash 模式的流程图和范例代码。

页擦除(每页 512 字节)

- 步骤 1: 在 ISPCR 寄存器上置位 ISPEN 使能 ISP/IAP 流程。
- 步骤 2: 在 IFMT 寄存器上设置 MS=0x03 选择页擦除模式。
- 步骤 3: 填入页地址到 IFADRH 和 IFADRL 寄存器。
- 步骤 4: 顺序地在 SCMD 寄存器写入 0x46h 然后 0xB9h 触发一个 ISP 处理。
- 步骤 5: 清零 ISPEN 和 MS=0x00 关闭 ISP/IAP 流程

字节编程

- 步骤 1: 在 ISPCR 寄存器上置位 ISPEN 使能 ISP/IAP 流程。
- 步骤 2: 在 IFMT 寄存器上设置 MS=0x02 选择字节编程模式。
- 步骤 3: 填入字节地址到 IFADRH 和 IFADRL 寄存器。
- 步骤 4: 填入被编程数据到 IFD 寄存器。
- 步骤 5: 顺序地在 SCMD 寄存器写入 0x46h 然后 0xB9h 触发一个 ISP 处理。
- 步骤 6: 清零 ISPEN 和 MS=0x00 关闭 ISP/IAP 流程

字节读取

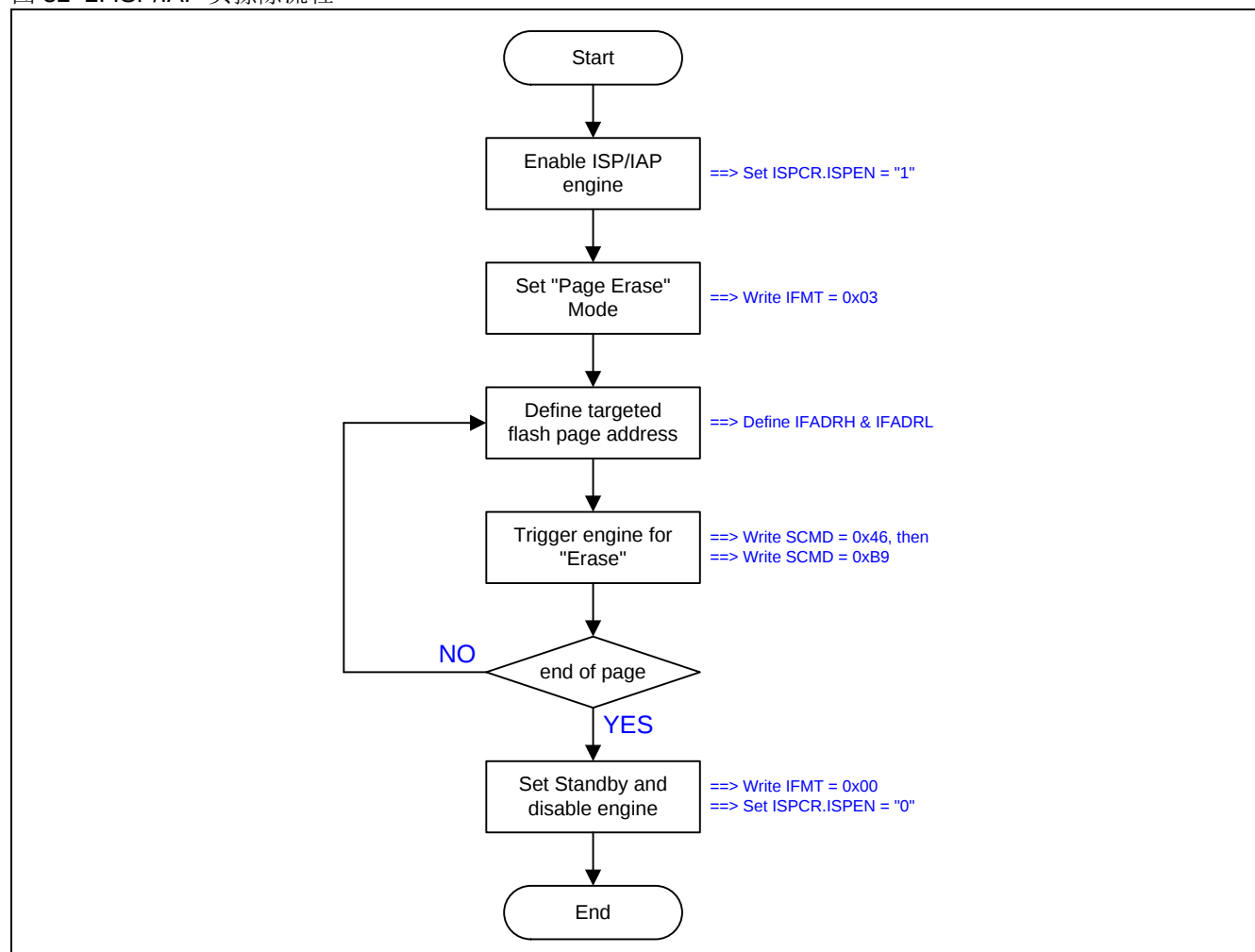
- 步骤 1: 在 ISPCR 寄存器上置位 ISPEN 使能 ISP/IAP 流程。
- 步骤 2: 在 IFMT 寄存器上设置 MS=0x01 选择字节读取模式。
- 步骤 3: 填入字节地址到 IFADRH 和 IFADRL 寄存器。
- 步骤 4: 顺序地在 SCMD 寄存器写入 0x46h 然后 0xB9h 触发一个 ISP 处理。
- 步骤 5: 现在,Flash 数据在 IFD 寄存器上。
- 步骤 5: 清零 ISPEN 和 MS=0x00 关闭 ISP/IAP 流程

MG82F6P32 的页擦除，字节编程和读取的详细描述见下面章节：

32.2.1. ISP/IAP Flash 页擦除模式

MG82F6P32 的 flash 数据任何一位只能编程为“0”。如果用户需要写“1”到 flash 数据，flash 需要擦除。但是在 MG82F6P32 的 ISP/IAP 操作中的 flash 擦除只支持“页擦除”模式，一页擦除将写“1”到一页的所有数据位。MG82F6P32 的一页有 512 个字节并且页的起始地址排列到 A8~A0=0x000。目标 flash 地址由 IFADRH 和 IFADRL 决定。这样，在 flash 页擦除模式，IFADRH.0(A8)和 IFADRL.7~0(A7~A0)必须写“0”选择正确的页地址。在 ISP/IAP 操作的 flash 页擦除流程如图 32-2 所示。

图 32-2. ISP/IAP 页擦除流程



ISP/IAP 页擦除操作的示例代码如图 32-3 所示。

图 32-3. ISP/IAP 页擦除操作的示例代码

```

MOV  ISPCR,#10000000b ; ISPCR.7 = 1, enable ISP
MOV  IFMT,#03h       ; select Page Erase Mode
MOV  IFADRH,??       ; fill [IFADRH,IFADRL] with page address
MOV  IFADRL,??       ;

MOV  SCMD,#46h       ; trigger ISP/IAP processing
MOV  SCMD,#0B9h      ;

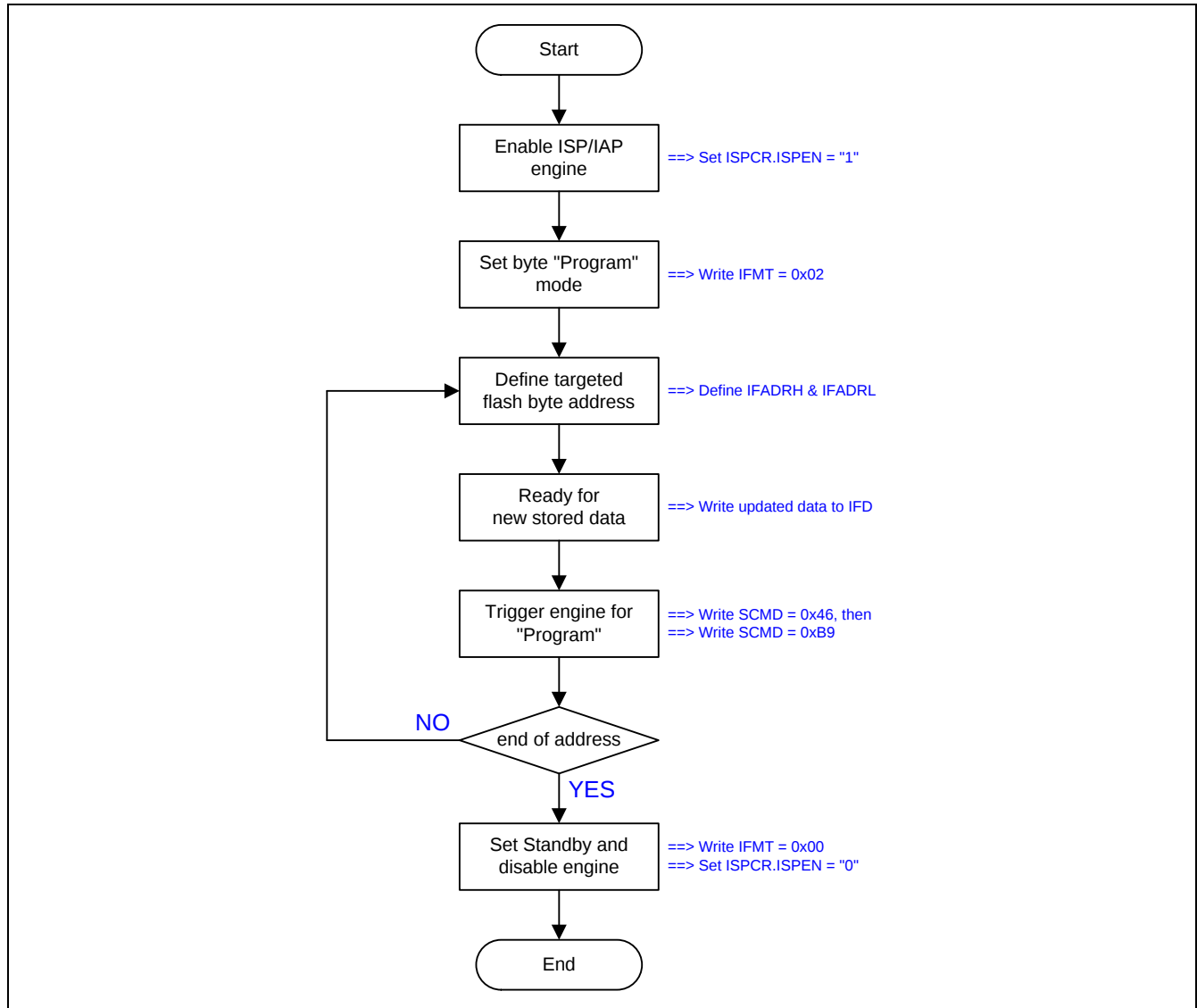
;Now, MCU will halt here until processing completed

MOV  IFMT,#00h       ; select Standby Mode
MOV  ISPCR,#00000000b ; ISPCR.7 = 0, disable ISP
  
```

32.2.2. ISP/IAP Flash 字节编程模式

MG82F6P32 编程模式提供 Flash 存储空间的字节写操作来更新数据。IFADRH 和 IFADRL 指向 Flash 的物理字节地址。IFD 存储编程到 Flash 的内容。ISP/IAP 操作的 Flash 字节编程流程如图 32-4 所示。

图 32-4. ISP/IAP 字节编程流程



ISP/IAP 字节编程操作的示例代码如图 32-5

图 32-5. ISP/IAP 字节编程的示例代码

```

MOV  ISPCR,#10000011b ; ISPCR.7=1, enable ISP
MOV  IFMT,#02h       ; select Program Mode
MOV  IFADRH,??       ; fill [IFADRH,IFADRL] with byte address
MOV  IFADRL,??       ;
MOV  IFD,??          ; fill IFD with the data to be programmed
MOV  SCMD,#46h       ; trigger ISP/IAP processing
MOV  SCMD,#0B9h      ;

```

;Now, MCU will halt here until processing completed

```

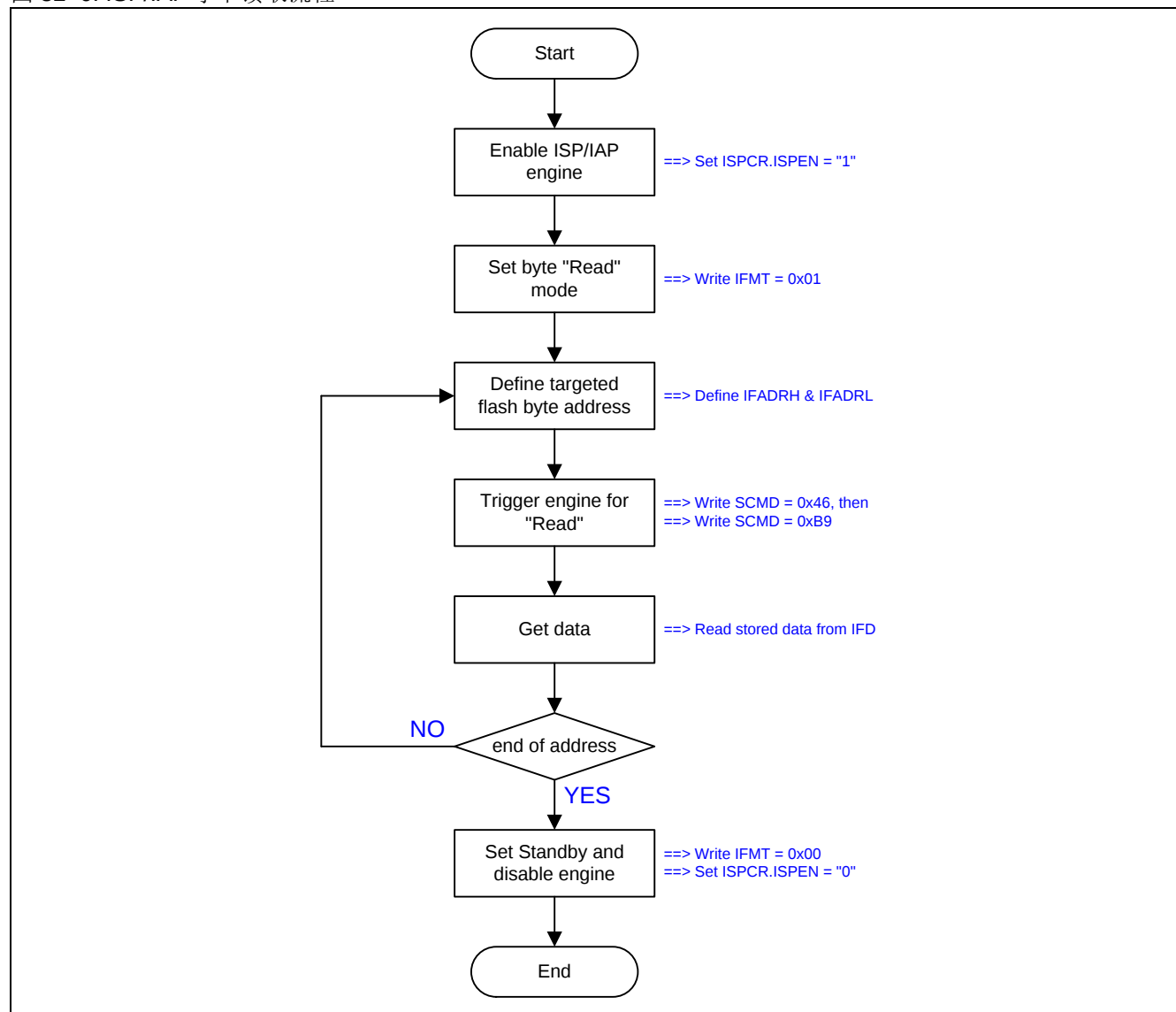
MOV  IFMT,#00h       ; select Standby Mode
MOV  ISPCR,#00000000b ; ISPCR.7 = 0, disable ISP

```

32.2.3. ISP/IAP Flash 读模式

MG82F6P32 读取模式提供从 Flash 存储空间获取已存储数据的字节读取操作。IFADRH 和 IFADRL 指向 Flash 的物理字节地址。IFD 存储从 Flash 读取到的内容。建议在数据编程或页擦除之后通过读取模式核对 Flash 数据。ISP/IAP 操作下的 Flash 字节读取流程如图 32-6 所示。

图 32-6. ISP/IAP 字节读取流程



ISP/IAP 字节读取操作的范例代码如图 32-7

图 32-7. ISP/IAP 字节读取的范例代码

```

MOV  ISPCR,#10000011b ; ISPCR.7=1, enable ISP
MOV  IFMT,#01h       ; select Read Mode
MOV  IFADRH,??       ; fill [IFADRH,IFADRL] with byte address
MOV  IFADRL,??       ;
MOV  SCMD,#46h       ; trigger ISP/IAP processing
MOV  SCMD,#0B9h      ;
;Now, MCU will halt here until processing completed
MOV  A,IFD           ; now, the read data exists in IFD
MOV  IFMT,#00h       ; select Standby Mode
MOV  ISPCR,#00000000b ; ISPCR.7 = 0, disable ISP
  
```

32.3. ISP 操作

ISP 意指在系统可编程，不需要在实际的终端产品上移除 MCU 芯片就可以更新用户的应用程序(AP 存储空间)和非易失性应用数据(IAP 存储空间)。这个可使用性就有一个宽的现场应用范围。ISP 模式使用引导程序来编程 AP 存储空间和 IAP 存储空间。

注意:

- (1) 在用 ISP 功能之前，用户必须先配置 ISP-存储器空间并用通用烧写器或[笙泉](#)的烧写器插入 ISP 代码(引导程序)到 ISP-存储器中。
- (2) ISP-存储器中的 ISP 代码只能编程 AP-存储器和 IAP-存储器。

在 ISP 操作完成之后,软件写“001”到 ISPCR.7 ~ ISPCR.5 这样会触发一个软件复位(RESET)并且使 CPU 再启动到应用程序存储空间(AP)的 0x0000 地址。

如我们所知，ISP 代码的作用就是编程 AP 存储空间和 IAP 存储空间。因此，**MCU 为了执行 ISP 代码必须从 ISP 存储空间启动**。根据 MCU 如何从 ISP 存储空间启动，有两种方法执行在系统可编程。

32.3.1. 硬件启动 ISP 方法

在上电复位时为了使 MCU 直接从 ISP 存储空间启动，MCU 的硬件选项 HWBS 和 ISP 存储空间必须使能。硬件选项的 ISP 进入方法叫做硬件访问。一旦 HWBS 和 ISP 存储空间使能,当上电复位时 MCU 总是从 ISP 存储空间启动去执行 ISP 代码(引导程序)。ISP 代码做的第一件事是核对是否有 ISP 请求。如果没有 ISP 请求，ISP 代码触发软件复位(设置 ISPCR.7~5 为“101”)使 MCU 在启动到 AP 存储空间去运行用户应用程序。

如果额外的硬件选项 HWBS2 与 HWBS 及 ISP 存储空间一起使能，MCU 在上电复位或外部复位结束之后总从 ISP 存储空间启动。通过外部复位信号提供另外一个硬件访问进入 ISP 模式。第一上电复位之后，**MG82F6P32** 通过外部复位触发而执行 ISP 操作并且不用等待下一次的上电复位，这适合不断电系统去应用硬件方法启动 ISP 功能。

32.3.2. 软件启动 ISP 方法

当 MCU 运行在 AP 存储空间时，软件访问 ISP 通过触发软件复位使 MCU 从 ISP 存储空间启动。这种情况，HWBS 或 HWBS2 不用使能。仅有的方法是当 MCU 运行在 AP 存储空间时同时设置 ISPCR.7~5 为“111”触发软件复位 MCU 从 ISP 存储空间启动。注意：ISP 存储空间必须通过硬件选项配置一个有效空间来保留 ISP 模式给软件方法启动 ISP 应用。

32.3.3. ISP 注意事项

ISP 代码开发

尽管 ISP 存储空间的 ISP 代码是可编程的，ISP 存储空间在 MCU 的 Flash 中有一个 **ISP 起始地址**(MG82F6P32 见图 32-1)，但是并不意味着你需要在你的源代码中加入这个偏移量(=ISP 起始地址)。代码偏移量硬件自动处理。用户只需像在 AP 存储空间开发应用程序一样开发。

ISP 期间的中断

在触发 ISP/IAP flash 处理之后，内部 ISP 处理时 MCU 将停止一会儿直到处理完成。此时，如果中断已使能则中断事件将排队等待服务。一旦 ISP/IAP flash 处理完成，MCU 继续运行并且如果中断标志仍然有效则排队中的中断将立即服务。不过用户需要意识到下列事项：

- (1) 当 MCU 停止在 ISP 处理时，任何中断都不能实时服务。
- (2) 低/高电平触发外部中断 nINTx，必须保持到 ISP 处理完成，否则将被忽略。

ISP 和 Idle 模式

MG82F6P32 不使用空闲模式执行 ISP 功能。反而 ISP/IAP 引擎操作 Flash 存储空间将冻结 CPU 的运行。一旦 ISP/IAP 运行结束，CPU 将继续并且推进紧跟着 ISP/AP 激活的指令。

ISP 的访问目标

如前所述，ISP 用来编程 AP 存储空间和 IAP 存储空间。一旦访问目标地址超出 IAP 存储空间的最后一个字节之外，硬件将自动忽略 ISP 处理的触发。这样 ISP 触发是无效的并且硬件不做任何事情。

ISP 的 Flash 持久期

内置 Flash 的持久期是 20,000 写周期，换句话说写周期不能超过 20,000 次。这样用户必须注意应用中需要频繁更新 AP 存储空间和 IAP 存储空间这一点。

32.4. 在应用编程(IAP)

MG82F6P32 内建一个在应用可编程(IAP)功能，当应用程序运行时在 Flash 存储空间里允许一些区域被应用成非易失性数据存储区。这个有用特点能使用在断电后还需要保存数据的应用中。这样不需要使用外部的串行 EEPROM (比如 93C46, 24C01, ..., 等等)来保存非易失性的数据。

事实上，IAP 的操作除了 Flash 存储空间被划分在不同的区域之外与 ISP 一样。ISP 操作的可编程 Flash 范围在 AP 存储空间和 IAP 存储空间，而 IAP 操作的范围只在 IAP 存储空间。

注意：

- (1) **MG82F6P32** 的 IAP 特点，软件通过写 IFMT 定义的 IAPLB 寄存器声明 IAP 存储空间。IAP 存储空间也可以通过通用的烧入器/编程器或笙泉专利的烧入器/编程器来配置 IAPLB 的初始值。
- (2) 执行 IAP 的程序代码是在 AP 存储空间并且**仅能编程 IAP 存储空间而不能编程 ISP 存储空间**

32.4.1. MG82F6P32 IAP 存储边界/范围

如果ISP 存储空间被指定，IAP存储空间范围由IAP和ISP起始地址决定如下：

$$\begin{aligned} \text{IAP 高边界} &= \text{ISP 起始地址} - 1. \\ \text{IAP 低边界} &= \text{ISP 起始地址} - \text{IAP}. \end{aligned}$$

如果ISP 存储空间没有被指定，IAP存储空间范围由下列公式决定：

$$\begin{aligned} \text{IAP 高边界} &= \mathbf{0x7FFF}. \\ \text{IAP 低边界} &= \mathbf{0x7FFF} - \text{IAP} + 1. \end{aligned}$$

例如，如果ISP存储空间是1K字节，这样ISP的起始地址是**0x7C00**，并且IAP存储空间是1K字节，此时IAP存储空间的范围就在**0x7800 ~ 0x7BFF**。**MG82F6P32**的IAP低边界由IAPLB寄存器决定，IAPLB寄存器可以在用户AP程序里用软件修改来调整IAP大小。

32.4.2. 更新 IAP-存储中的数据

ISP/IAP 相关的特殊功能寄存器见章节“[32.5 ISP/IAP](#)”。

由于 IAP 存储空间是 Flash 存储空间的一部分，Flash 擦除仅提供**页擦除**，**没有字节擦除**。为了在 IAP 存储空间更新“一个字节”，用户不能直接编程一个新数据到那个字节。正确的步骤如下：

- 步骤 1: 保存整页 flash 数据(512 字节)到包含被更新数据的 XRAM 缓冲区。
- 步骤 2: 擦除此页(使用 ISP/IAP Flash 页擦除模式)。
- 步骤 3: 在 XRAM 缓冲区修改新数据字节。
- 步骤 4: 编程 XRAM 缓冲区的被更新数据到此页(使用 ISP/IAP Flash 编程模式)。

为了读取 IAP 存储空间数据，用户可以使用**ISP/IAP Flash 读取模式**获取目标数据。

32.4.3. AP 注意事项

IAP 期间的中断

在触发 ISP/IAP flash 处理之后，内部 IAP 处理时 MCU 将停止一会儿直到处理完成。此时，如果中断已使能则中断事件将排队等待服务。一旦 ISP/IAP flash 处理完成，MCU 继续运行并且如果中断标志仍然有效则排队中的中断将立即服务。不过用户需要意识到下列事项：

- (1) 当 MCU 停止在 IAP 处理时，中断不能实时服务。
- (2) 低/高电平触发外部中断 nINTx，必须保持到 IAP 处理完成，否则将被忽略。

IAP 和空闲模式

MG82F6P32 不使用空闲模式执行 IAP 功能。反而 ISP/IAP 引擎操作 Flash 存储空间将冻结 CPU 的运行。一旦 ISP/IAP 运行结束，CPU 将继续并且推进紧跟着 ISP/AP 激活的指令。

IAP 的访问目标

如前所述，IAP 用来编程 IAP 存储空间。一旦访问目标地址不在 IAP 存储空间之内，硬件将自动忽略 IAP 处理的触发。这样 IAP 触发是无效的并且硬件不做任何事情。

读取 IAP 数据的另一种方法

IAP 存储空间读取 Flash 数据，除了使用 Flash 的读取模式之外，另一个方法是使用“MOVC A,@A+DPTR”指令。这里，DPTR 和 ACC 各自填入想要的地址和偏移量。并且访问目标必须在 IAP 存储空间内，否则读取的数据将不确定。注意使用‘MOVC’指令比使用 Flash 的读取模式更快。

IAP 的 Flash 持久期

内置 Flash 的持久期是 20,000 擦除/写周期，换句话说擦除再写周期不能超过 20,000 次。这样用户必须注意应用中需要频繁更新 IAP 存储空间这一点。

32.5. ISP/IAP 寄存器

下面专门描述ISP，IAP和P页相关的特殊功能寄存器：

IFD: ISP/IAP Flash 数据寄存器

SFR 页 = 0~F

SFR 地址 = 0xE2

复位值 = 1111-1111

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IFD 是 ISP/IAP/P 页操作的数据端口寄存器。在 ISP/IAP/P 页写操作时 IFD 的数据将被写入到期望的地址，在 ISP/IAP/P 页读操作时 IFD 的值是读到期望地址的数据。

IFADRH: ISP/IAP 地址高八位

SFR 页 = 0~F

SFR 地址 = 0xE3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IFADRH 是所有 ISP/IAP 模式下的高 8 位地址。在 P 页模式下没有定义。

IFADRL: ISP/IAP 地址低八位

SFR 页 = 0~F

SFR 地址 = 0xE4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IFADRL 是所有 ISP/IAP/P 页模式下的低 8 位地址。在 flash 页擦除时，IFADRL 可以不用理会。

IFMT: ISP/IAP Flash 模式表

SFR 页 = 0~F

SFR 地址 = 0xE5

复位值 = xxxx-x000

7	6	5	4	3	2	1	0
MS.7	MS.6	MS.5	MS.4	MS.3	MS.2	MS.1	MS.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: ISP/IAP/ P 页 操作模式选择

MS[7:0]	模式
0 0 0 0-0 0 0 0	待机
0 0 0 0-0 0 0 1	AP/IAP-存储器字节读
0 0 0 0-0 0 1 0	AP/IAP-存储器字节编程
0 0 0 0-0 0 1 1	AP/IAP-存储器页擦除
0 0 0 0-0 1 0 0	P 页 SFR 写
0 0 0 0-0 1 0 1	P 页 SFR 读
1 0 0 0-0 0 0 0	CRC 的自动 flash 读。
1 0 0 0-0 0 0 1	Flash 字节读地址加一功能。
1 0 0 0-0 0 1 0	Flash 字节编程地址加一功能。
其它	保留

IFMT 是用来选择闪存是用来执行众多的 ISP/IAP 功能还是选择 P 页寄存器的访问。

如果软件选择 CRC 的自动 flash 读模式，flash 起始地址由 IFADRH 和 IFADRL 定义。flash 结束地址在{IAPLB + 9 位 1-1111-1111}。

SCMD: 连续命令字寄存器

SFR 页 = 0~F

SFR 地址 = 0xE6复位值 = xxxx-xxxx

7	6	5	4	3	2	1	0
SCMD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCMD 是激活 ISP/IAP/P 页的命令口。如果 SCMD 连续填入 0x46h、0xB9h 并且 ISPCR.7=1，ISP/IAP/P 页被激活。

ISPCR: ISP 控制寄存器

SFR 页 = 0~F

SFR 地址 = 0xE7POR = 0000-0000

7	6	5	4	3	2	1	0
ISPEN	SWBS	SWRST	CFAIL	--	--	--	--
R/W	R/W	R/W	R/W	W	W	W	W

- Bit 7: ISPEN，ISP/IAP/P 页操作使能。
0: 所有的 ISP/IAP/P 页编程/擦除/读都是被禁止的。
1: 使能 ISP/IAP/P 页编程/擦除/读功能。
- Bit 6: SWBS，软件执行起始选择控制。
0: 复位软件从主存储区开始执行。
1: 复位软件从 ISP 存储区开始执行。
- Bit 5: SWRST，软件复位触发控制。
0: 无操作。
1: 产生软件系统复位，硬件自动清零。
- Bit 4: CFAIL，ISP/IAP 操作命令失败指示。
0: 最后一次 ISP/IAP 命令成功。
1: 最后一次 ISP/IAP 命令失败。失败的原因是闪存访问被阻止。
- Bit 3~0: 保留。写该寄存器时，此位必须写“0”。

IAPLB: IAP 低边界

SFR 页 = 仅 P 页

SFR 地址 = 0x03复位值 = 0111-000x

7	6	5	4	3	2	1	0
IAPLB							0
W	W	W	W	W	W	W	W

Bit 7~0: IAPLB 决定 IAP 存储区的最低边界。因为一个闪存页是 512 字节，所以 IAPLB 必须是偶数。为了读取 IAPLB，MCU 需要在 P 页里定义 IFADRL 地址，IMFT 模式选择 P 页读及 ISPCR.ISPEN 置位。并且在 SCMD 依次写入 0x46h 和 0xB9h，这样 IAPLB 的值就会出现在 IFD。写 IAPLB，首先 MCU 把新的 IAPLB 设定值写入 IFD；其次索引 IFADRL，选择 IMFT，使能 ISPCR.ISPEN；然后设置 SCMD。这样 IAPLB 就会更新到最新的顺序。

由 IAPLB 及 ISP 起始地址决定的 IAP 存储区如下。
IAP 低边界= IAPLB[7:0] x256
IAP 高边界= ISP 起始地址 - 1。
如果 IAPLB=0x20 且 ISP 起始地址是 0x3000，这样 IAP 存储区位于 0x2000 ~ 0x2FFF。

另外要注意一点，IAP 的低边界地址不能大于 ISP 的起始地址。

32.5.1. ISP/IAP 示例代码

ISP 操作的示例代码如图 32-8 所示。

图 32-8. ISP 示例代码

```

*****
; Demo Program for the ISP
*****
IFD    DATA 0E2h
IFADRH DATA 0E3h
IFADRL DATA 0E4h
IFMT   DATA 0E5h
SCMD   DATA 0E6h
ISPCR  DATA 0E7h
;
;      MOV  ISPCR,#10000000b ;ISPCR.7=1, enable ISP
;
;=====
; 1. Page Erase Mode (512 bytes per page)
;=====
;      ORL  IFMT,#03h  ;MS[2:0]=[0,1,1], select Page Erase Mode
;      MOV  IFADRH,??  ;fill page address in IFADRH & IFADRL
;      MOV  IFADRL,??  ;
;      MOV  SCMD,#46h  ;trigger ISP processing
;      MOV  SCMD,#0B9h ;
;      ;Now in processing...(CPU will halt here until complete)
;
;=====
; 2. Byte Program Mode
;=====
;      ORL  IFMT,#02h  ;MS[2:0]=[0,1,0], select Byte Program Mode
;      ANL  ISPCR,#0FAh ;
;      MOV  IFADRH,??  ;fill byte address in IFADRH & IFADRL
;      MOV  IFADRL,??  ;
;      MOV  IFD,??     ;fill the data to be programmed in IFD
;      MOV  SCMD,#46h  ;trigger ISP processing
;      MOV  SCMD,#0B9h ;
;      ;Now in processing...(CPU will halt here until complete)
;
;=====
; 3. Verify using Read Mode
;=====
;      ANL  IFMT,#0F9h ;MS1[2:0]=[0,0,1], select Byte Read Mode
;      ORL  IFMT,#01h  ;
;      MOV  IFADRH,??  ;fill byte address in IFADRH & IFADRL
;      MOV  IFADRL,??  ;
;      MOV  SCMD,#46h  ;trigger ISP processing
;      MOV  SCMD,#0B9h ;
;      ;Now in processing...(CPU will halt here until complete)
;      MOV  A,IFD     ;data will be in IFD
;      CJNE A,wanted,ISP_error ;compare with the wanted value
;      ...
ISP_error:
;      ...
;

```

33. P 页 SFR 访问

MG82F6P32 内建一个特别的 SFR 页 (P 页)用来存储 MCU 操作的控制寄存器。这些特殊功能寄存器在不同 IFMT 下通过 ISP/IAP 操作来访问。在 P 页访问时, IFADRH 必须设置为“00”及 IFADRL 索引 P 页内特殊功能寄存器地址。如果 IFMT=04H 则 P 页写操作, 在 SCMD 激活之后 IFD 的数据会被载入到 IFADRL 索引的特殊功能寄存器。如果 IFMT=05H 则 P 页读操作, 在 SCMD 激活之后 IFD 的数据将是 IFADRL 索引的特殊功能寄存器(SFR)的值。

以下是 P 页的 SFR 功能定义:

IAPLB: IAP 低边界

SFR 页 = P

SFR 地址 = 0x03

复位值 = 1111-111x

7	6	5	4	3	2	1	0
IAPLB							0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: IAPLB 决定 IAP 存储区的最低边界。因为一个闪存页是 512 字节, 所以 IAPLB 必须是偶数。为了读取 IAPLB, MCU 需要在 P 页里定义 IFADRL 地址, IMFT 模式选择 P 页读及 ISPCR.ISPEN 置位。并且在 SCMD 依次写入 0x46h 和 0xB9h, 这样 IAPLB 的值就会出现在 IFD。写 IAPLB, 首先 MCU 把新的 IAPLB 设定值写入 IFD; 其次索引 IFADRL, 选择 IMFT, 使能 ISPCR.ISPEN; 然后设置 SCMD。这样 IAPLB 就会更新到最新的顺序。

由 IAPLB 及 ISP 起始地址决定的 IAP 存储区如下。

IAP 低边界= IAPLB[7:0] x256

IAP 高边界= ISP 起始地址 - 1。

如果 IAPLB=0xE0 且 ISP 起始地址是 0xF000, 这样 IAP 存储区位于 0xE000 ~ 0xEFFF。

另外要注意一点, IAP 的低边界地址不能大于 ISP 的起始地址。

CKCON2: 时钟控制寄存器 2

SFR 页 = 仅 P 页

SFR 地址 = 0x40

复位值 = 0101-0000

7	6	5	4	3	2	1	0
XTGS1	XTGS0	XTALE	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: XTGS1~XTGS0, 晶振增益选择位。

XTGS1, XTGS0	增益定义	晶振应用
0, 0	低增益	32.768KHz
0, 1	中等增益	2MHz ~ 25MHz
1, 0	较低增益	32.768KHz
1, 1	保留	保留

Bit 5: XTALE,外部晶振 (XTAL)使能

0: 禁止 XTAL 振荡电路, 在这种情况下, XTAL2 和 XTAL1 表现为 Port 6.0 和 Port 6.1。

1: 使能 XTAL 振荡电路, 如果 CPU 软件设置这个位, 软件检测到 **XTOR** (CKCON1.7)为“1”表明晶振振荡器准备好作为 OSCin 时钟选择。

Bit 4: IHRCOE, 内部高频 RC 震荡使能位

0: 禁止内部高频 RC 震荡电路

1: 使能内部快频 RC 震荡电路。如果软件设置这个位, 在 IHRCOE 位使能后, 必须等待 **32 us** IHRCOE 才能稳定输出

Bit 3~2: MCKS[1:0], MCK 时钟源选择.

MCKS[1:0]	MCK 时钟源选择	OSCIn = 12MHz CKMIS = [01]	OSCIn = 11.059MHz CKMIS = [01]
0 0	OSCIn	12MHz	11.059MHz
0 1	CKMI x4	24MHz	22.118MHz
1 0	CKMI x5.33	32MHz	29.491MHz
1 1	CKMI x8	48MHz	44.236MHz

注意: 需要设置 ENCKM = 1 使能 CKM.

注意: 需要注意 CPUCLK 和 SYSCLK 的限制。需要使用 SCKS[2:0]和 CCKS 选择 CPUCLK 和 SYSCLK 的正确范围不要超出限制。CPUCLK ≤ 32MHz, SYSCLK ≤ 50MHz。

Bit 1~0: OSCS[1:0], OSCIn 时钟源选择

OSCS[1:0]	OSCIn 时钟源选择
0 0	IHRCO
0 1	XTAL
1 0	ILRCO
1 1	ECKI, 外部时钟输入 (P6.0)作为 OSCIn.

CKCON3: 时钟控制寄存器 3

SFR 页 = 仅 P 页

SFR 地址 = 0x41

复位值 = 0000-0010

7	6	5	4	3	2	1	0
WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	MCDS1	MCDS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: WDTCS1~0. WDT 时钟源选择.

Bit 5: FWKP, MCU 快速唤醒控制。

0: 选择 MCU 从掉电模式正常唤醒时间大约 120us。

1: 选择 MCU 从掉电模式快速唤醒时间大约 30us。

Bit 4: WDTFS. WDT 溢出源选择。

0: 选择 WDT 位-8 溢出作为 WDT 事件源。

1: 选择 WDT 位-0 溢出作为 WDT 事件源。

Bit 3~2: MCKD[1:0], MCK 分频输出选择.

MCKD[1:0]	MCKDO Frequency	if MCK = 12MHz	if MCK = 48MHz
0 0	MCKDO = MCK	MCKDO = 12MHz	MCKDO = 48MHz
0 1	MCKDO = MCK/2	MCKDO = 6MHz	MCKDO = 24MHz
1 0	MCKDO = MCK/4	MCKDO = 3MHz	MCKDO = 12MHz
1 1	MCKDO = MCK/8	MCKDO = 1.5MHz	MCKDO = 6MHz

Bit 1~0: MCDS[1:0], 保留用于测试

CKCON4: 时钟控制寄存器 4

SFR 页 = 仅 P 页

SFR 地址 = 0x42

复位值 = 0000-0000

7	6	5	4	3	2	1	0
RCSS2	RCSS1	RCSS0	RPSC2	RPSC1	RPSC0	RTCCS3	RTCCS2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~5: RTC 时钟源选择[2:0]

RCSS2, RCSS1, RCSS0	RTC 时钟源选择
0 0 0	ECKI (P6.0)
0 0 1	ILRCO
0 1 0	WDTPS
0 1 1	WDTOF
1 0 0	SYSCLK

1 0 1	SYSCLK / 12
1 1 0	保留
1 1 1	保留

PCON2: 电源控制寄存器 2

SFR 页 = 仅 P 页

SFR 地址 = 0x44

POR = 0000-0101

7	6	5	4	3	2	1	0
AWBOD1	0	BO1S1	BO1S0	BO1RE	EBOD1	BO0RE	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: AWBOD1, 掉电模式(PD)下 BOD1 的唤醒。

0: 掉电模式(PD)下禁止 BOD1。

1: 掉电模式(PD)下保持 BOD1。

Bit 6: 保留。写该寄存器时, 此位必须写“0”。

Bit 5~4: BO1S[1:0]. 低电压监测器 1 监测电压选择. 这两个位的初始值是从 OR1.BO1S10 和 OR1.BO1S00 加载的。

BO1S[1:0]	BOD1 侦测电平
0 0	2.0V
0 1	2.4V
1 0	3.6V
1 1	4.2V

Bit 3: BO1RE, BOD1 复位使能。

0: 当 BOF1 已经设置, 禁止低电压监测 1(BOD1)系统复位。

1: 当 BOF1 已经设置, 使能低电压监测 1(BOD1)系统复位。

Bit 2: EBOD1, 使能 BOD1 监测 VDD 下降到 BO1S1~0 设置的固定值。

0: 禁止 BOD1 监测电源电压降低芯片功耗。

1: 使能 BOD1 监测电源电压 VDD。

Bit 1: BO0RE, BOD0 复位使能。

0: 当 BOF0 已经设置, 禁止低电压监测 0(BOD0)系统复位。

1: 当 BOF0 已经设置, 使能低电压监测 0(BOD0)系统复位(VDD 触到 1.7V)。

Bit 0: 保留位。写寄存器时, 此位必须写“1”。

PCON3: 电源控制寄存器 3

SFR 页 = 仅 P 页

SFR 地址 = 0x45

POR = 0000-0000

7	6	5	4	3	2	1	0
IVREN	IVRPDE	0	SPWRE	0	0	0	0
R/W	R/W	W	R/W	W	W	W	W

Bit 7: IVREN, 内部电压参照使能。

0: 禁止片内 IVR (2.4V)。

1: 使能片内 IVR (2.4V)。

Bit 6 IVRPDE, IVR 可以在掉电模式下使用

0: 禁止 IVR 在掉电模式下使用

1: 使能 IVR 在掉电模式下使用

Bit 5: 保留。写该寄存器时, 此位必须写“0”。

Bit 4: SPWRE, SPWF 触发一个 MCU 复位(软件)。

0: 禁止 SPWF 触发 MCU 复位。

1: 使能 SPWF 触发 MCU 复位。

Bit 3~0: 保留。写该寄存器时，此位必须写“0”。

SPCON0: SFR 页控制 0

SFR 页 = 仅 P 页

SFR 地址 = 0x48

POR = 0000-0000

7	6	5	4	3	2	1	0
--	P6CTL	P4CTL	WRCTL	--	CKCTL0	PWCTL1	PWCTL0
W	R/W	R/W	R/W	W	R/W	R/W	R/W

Bit 7: 保留。写该寄存器时，此位必须写“0”。

Bit 6: P6CTL, P6 SFR 访问控制。

如果 P6CTL 置位，则 P6 禁止在 0~F 页改写。P6 在 0~F 页保持读取。但是在 SFR P 页软件拥有改写权利。

Bit 5: P4CTL, P4 SFR 访问控制。

如果 P4CTL 置位，则 P4 禁止在 0~F 页改写。P4 在 0~F 页保持读取。但是在 SFR P 页软件拥有改写权利。

Bit 4: WRCTL, WDTCSR SFR 访问控制。

如果 WRCTL 置位，则 WRCTL 禁止在 0~F 页改写。WRCTL 在 0~F 页保持读取。但是在 SFR P 页软件拥有改写权利。

Bit 3: 保留位，写寄存器时，此位必须写“0”。

Bit 2: CKCTL0, CKCON0 SFR 访问控制。

如果 CKCTL0 置位，则 CKCON0 禁止在 0~F 页改写。CKCON0 在 0~F 页保持读取。但是在 SFR P 页软件拥有改写权利。

Bit 1: PWCTL1, PCON1 SFR 访问控制。

如果 PWCTL1 置位，则 PCON1 禁止在 0~F 页改写。PCON1 在 0~F 页保持读取。但是在 SFR P 页软件拥有改写权利。

Bit 0: PWCTL0, PCON0 SFR 访问控制。

如果 PWCTL0 置位，则 PCON0 禁止在 0~F 页改写。PCON0 在 0~F 页保持读取。但是在 SFR P 页软件拥有改写权利。

DCON0: 设备控制寄存器 0

SFR 页 = 仅 P 页

SFR 地址 = 0x4C

复位值 = 1000-0011

7	6	5	4	3	2	1	0
HSE	IAPO	HSE1	0	0	IORCTL	RSTIO	OCDE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: HSE, 高速运行使能。

0: 选择 CPU 运行在低速模式($FCPUCLK \leq 6MHz$)这样减慢内部电路从而降低功耗。

1: 使能 MCU 全速运行($FCPUCLK > 6MHz$)。在 SYSCLK 选择高频时钟(>6MHz)之前，软件必须置位 HSE 切换到用于高速运行的内部电路。

Bit 6: IAPO, 仅 IAP 功能。

0: 保留 IAP 区服务于 IAP 功能和程序代码执行。

1: IAP 区禁止程序代码执行并且仅服务于 IAP 功能

Bit 5: HSE1, 超高速运行使能。

0: 无功能。

1: 使能 MCU 超高速运行。($FCPUCLK > 25MHz$)当使用 HSE1=1 需要置位 HSE。

Bit 4~3: 保留位，写寄存器时，此位必须写“0”。

Bit 2: IORCTL, GPIO 复位控制。

0: 端口 6(Port 6)所有复位事件下保持复位。

1: 如果此位置位，端口 6(Port 6)仅通过 POR/LVR/Ext_Reset/BOR0/BOR1 (如果 BOR0/1 是使能的)复位。

Bit 1: RSTIO, RST 功能为 I/O。

- 0: 选择 I/O 引脚功能为 P47。
- 1: 选择 I/O 引脚功能为外部复位输入(RST)。

Bit 0: OCDE, OCD 使能。
0: 在 P4.4 和 P4.5 禁止 OCD 接口。
1: 在 P4.4 和 P4.5 使能 OCD 接口。

SPHB: 堆栈指针高边界

SFR 页 = 仅 P 页

SFR 地址 = 0x53 复位值 = 1111-1111

7	6	5	4	3	2	1	0
1	1	1	1	SPHB.3	SPHB.2	SPHB.1	SPHB.0
R	R	R	R	R/W	R/W	R/W	R/W

SPHB, 用于堆栈指针警告的检测边界
如果 SPHB == 1111-1111, 当 $SP \geq 1111-1111$, 将置位 SPWF。
如果 SPHB == 1111-0000, 当 $SP \geq 1111-0000$, 将置位 SPWF。.

34. 辅助特殊功能寄存器

AUXR0: 辅助寄存器 0

SFR 页 = 0~F

SFR 地址 = 0xA1

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C1BKF	PBKF	0	0	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 保留。写该寄存器时，此位必须写“0”。

Bit 5: C1BKF, PCA1 PWM 中止事件标志。此位由 PWM 中止源使能设置。如果此位设置，则使能的 PWM 通道 0~1 将被锁住并且输出引脚保持最初的 GPIO 状态。

0: 没有 PWM 中止事件出现。仅由软件清零。

1: PWM 中止事件出现或软件触发一个 PWM 中止。

Bit 4: PBKF, PWM 中止标志。此位由 PWM 中止源使能置位。如果此位置位，则使能的 PWM 通道 0~5 将被锁住并且输出引脚保持最初的 GPIO 状态。

0: 没有 PWM 中止事件出现。仅由软件清零。

1: PWM 中止事件出现或软件触发一个 PWM 中止。

Bit 3~2: 保留。写该寄存器时，此位必须写“0”。

Bit 1: INT1H, INT1 高电平/上升沿触发使能。

0: 保留 INT1 在选择的端口引脚上低电平或下降沿触发。

1: 设置 INT1 在选择的端口引脚上高电平或上升沿触发。

Bit 0: INT0H, INT0 高电平/上升沿触发使能。

0: 保留 INT0 在选择的端口引脚上低电平或下降沿触发。

1: 设置 INT0 在选择的端口引脚上高电平或上升沿触发。

AUXR1: 辅助寄存器 1

SFR 页 = 0~F

SFR 地址 = 0xA2

复位值 = 0000-0000

7	6	5	4	3	2	1	0
OP1Fr	OP0Fr	CRCDS1	CRCDS0	0	AC1Fr	AC0Fr	DPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: OP1Fr. 如果 OP1CMPF 设置为 1，则 OP1Fr 也会根据硬件设置设置为 1。仅读。

Bit 6: OP0Fr. 如果 OP0CMPF 设置为 1，则 OP0Fr 也会根据硬件设置设置为 1。仅读。

Bit 5~4: CRCDS1~0. CRC0 数据端口选择 1~0。

Bit 3: 保留。写该寄存器时，此位必须写“0”。

Bit 2: AC1Fr. 如果 AC1F 设置为 1，则 AC1Fr 也会根据硬件设置设置为 1。仅读。

Bit 1: AC0Fr. 如果 AC0F 设置为 1，则 AC0Fr 也会根据硬件设置设置为 1。仅读。

Bit 0: DPS, DPTR 选择位，用来在 DPTR0 和 DPTR1 之间切换。

0: 选择 DPTR0。

1: 选择 DPTR1。

DPS	Selected DPTR
0	DPTR0
1	DPTR1

AUXR2: 辅助寄存器 2

SFR 页 = 0~F

SFR 地址 = 0xA3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
STAF	STOF	C1PLK	C0PLK	T1X12	T0X12	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: STAF, STWI/SI2C 的起始标志侦测。

0: 软件写“0”清零。

1: 硬件置位, 表示在 STWI 总线上发生了一个起始动作。

Bit 6: STOF, STWI /SI2C 的停止标志侦测。

0: 软件写“0”清零。

1: 硬件置位, 表示在 STWI 总线上发生了一个停止动作。

Bit 5: C1PLK, PCA1 缓冲 PWM/COPM 更新控制

0: 在 PCA1 时基溢出时缓冲 PWM/COPM 自动重载

1: 禁止缓冲 PWM/COPM 自动更新

Bit 4: C0PLK, PTM0 缓冲 PWM/COPM 更新控制

0: 在 PTM0 时基溢出时缓冲 PWM/COPM 自动重载

1: 禁止缓冲 PWM/COPM 自动更新

Bit 3: T1X12, 当 C/T=0 时, 定时器 1 时钟源选择。

0: 清零选择 SYSCLK/12。

1: 置位选择 SYSCLK 作时钟源。

Bit 3: T0X12, 当 C/T=0 时, 定时器 0 时钟源选择。

0: 清零选择 SYSCLK/12。

1: 置位选择 SYSCLK 作时钟源。

Bit 1~0: 保留。写该寄存器时, 此位必须写“0”。

AUXR3: 辅助寄存器 3

SFR 页 = 仅 0 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	T0XL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: T0PS1~0, 定时器 0 端口引脚选择[1:0].

T0PS1~0	T0/T0CKO
0 0	P3.4
0 1	P4.4
1 0	P2.2
1 1	P1.7

Bit 5~4: BPOC1~0, 蜂鸣器输出控制位

BPOC[1:0]	P4.4 功能	I/O 模式
0 0	P4.4	By P4M0.4 & P4M1.4
0 1	ILRCO/32	By P4M0.4 & P4M1.4
1 0	ILRCO/16	By P4M0.4 & P4M1.4
1 1	ILRCO/8	By P4M0.4 & P4M1.4

P4.4 用于蜂鸣器功能, 推荐设置 P4.4 工作在推挽输出模式。.

Bit 3: S0PS0, 串口 0 引脚选择 0. (S0PS1 at AUXR10.3)

S0PS1~0	RXD0	TXD0
0 0	P3.0	P3.1
0 1	P4.4	P4.5
1 0	P3.1	P3.0
1 1	P1.7	P2.2

Bit 2~1: TWIPS1~0, TWI0/I2C0 端口选择 [1:0].

TWIPS1~0	TWI0_SCL	TWI0_SDA
0 0	P4.0	P4.1
0 1	P6.0	P6.1
1 0	P3.1	P3.0
1 1	P2.2	P2.4

Bit 0: T0XL 是定时器 0 预分频控制位。T0XL 功能定义请参考 T0X12 (AUXR2.2)。

AUXR4: 辅助寄存器 4

SFR 页 = 仅 1 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	0	AC0OE	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: T2PS1~0, 定时器 2 端口引脚选择 [1:0].

T2PS1~0	T2/T2CKO	T2EX
0 0	P1.0	P1.1
0 1	P3.2	P3.3
1 0	P6.0	P3.5
1 1	P4.5	P4.4

Bit 5~4: T1PS1~0, 定时器 1 端口引脚选择 [1:0].

T1PS1~0	T1/T1CKO
0 0	P3.5
0 1	P4.5
1 0	P1.7
1 1	P3.3

Bit 3: AC1OE, 使能 AC1OUT 输出在端口引脚上。.

0: 禁止 AC1OUT 输出在端口引脚上。

1: 使能 AC1OUT 输出在 P0.5 上

Bit 2: 保留。写该寄存器时，此位必须写“0”。

Bit 1: AC0OE, 使能 AC0OUT 输出在端口引脚上。

0: 禁止 AC0OUT 输出在端口引脚上。

1: 使能 AC0OUT 输出在 P4.5 上

Bit 0: 保留。写该寄存器时，此位必须写“0”。

AUXR5: 辅助寄存器 5

SFR 页 = 仅 2 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0PPS1	C0PPS0	EC1IPS0	C1COPS	ECIPS0	C0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 保留。写该寄存器时，此位必须写“0”。

Bit 5: C0PPS1, {PWM2A, PWM2B} 端口引脚选择 1.

C0PPS1	PWM2A	PWM2B
0	P4.0	P4.1
1	P3.4	P3.5

Bit 4: C0PPS0, {PWM0A, PWM0B} 端口引脚选择 0.

C0PPS0	PWM0A	PWM0B
0	P1.6	P1.7
1	P6.0	P6.1

Bit 3: EC1IPS0, PCA1 ECI 端口引脚选择 0.

EC1IPS0	ECI
0	ECI
1	P1.5

Bit 2: C1COPS, PCA1 时钟输出 (C1CKO) 端口引脚选择.

C1COPS	C1CKO
0	P4.7
1	P3.3

Bit 1: ECIPS0, PTM0 ECI 端口引脚选择 0.

ECIPS0	ECI
0	P0.2
1	P1.6

Bit 0: C0COPS, PTM0 时钟输出(C0CKO) 端口引脚选择.

C0COPS	C0CKO
0	P4.7
1	P3.3

AUXR6: 辅助寄存器 6

SFR 页 = 仅 3 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
KBIHPS1	KBIHPS0	KBILPS1	KBILPS0	T3FCS	T2FCS	SnMIPS	S0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: KBIHPS1~0, KBI4~7 端口引脚选择 [1:0].

KBIHPS1~0	KBI7	KBI6	KBI5	KBI4
0 0	P1.7	P1.6	P1.5	P1.4
0 1	P1.7	P1.6	P3.5	P3.4
1 0	P2.6	P2.4	P2.3	P2.2
1 1	P1.7	P1.6	P6.1	P6.0

Bit 5~4: KBILPS1~0, KBI0~3 端口引脚选择 0.

KBILPS1~0	KBI3	KBI2	KBI1	KBI0
0 0	P1.3	P1.2	P1.1	P1.0
0 1	P3.3	P3.2	P3.1	P3.0
1 0	P0.7	P0.5	P0.2	P0.1
1 1	P4.5	P4.4	P4.1	P4.0

Bit 3: T3FCS, 保留用于芯片测试

Bit 2: T2FCS, 保留用于芯片测试

Bit 1: SnMIPS, S0MI, S1MI, S2MI & S3MI 端口引脚选择.

SnMIPS	S0MI	S1MI
0	P1.6	P1.0
1	P3.3	P3.5

Bit 0: S0COPS, S0BRG 时钟输出(S0CKO) 端口引脚选择.

S0COPS	S0CKO
0	P4.7
1	P4.4

AUXR7: 辅助寄存器 7

SFR 页 = 仅 4 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0CKOE	C1CKOE	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 保留。写该寄存器时，此位必须写“0”。

Bit 5: C0CKOE, PTM0 时钟输出使能。

0: 禁止 PTM0 时钟输出。

1: PTM0 基准定时器溢出率的二分之一时钟输出使能。

Bit 4: C1CKOE, PCA1 时钟输出使能。

0: 禁止 PCA1 时钟输出。

1: PCA1 基准定时器溢出率的二分之一时钟输出使能。

Bit 3~0: 保留。写该寄存器时，此位必须写“0”。

AUXR8: 辅助寄存器 8

SFR 页 = 仅 5 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	C1ICS1	C1ICS0	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留。写该寄存器时，此位必须写“0”。

Bit 6: C1ICS1, PCA1 输入通道 1 输入端口引脚选择

C1ICS1	C1EX1 input
0	C1EX1 Port Pin
1	T2EXI

Bit 5: C1ICS0, PCA1 输入通道 0 输入端口引脚选择.

C1ICS0	C1EX0 input
0	C1EX0 Port Pin
1	T3EXI

Bit 4~3: 保留。写该寄存器时，此位必须写“0”。

Bit 2: S1COPS, S1BRG 时钟输出 (S1CKO) 端口引脚选择.

S1COPS	S1CKO
0	P4.7
1	P4.5

Bit 1~0: T3PS1~0, Timer 3 端口引脚选择 [1:0].

T3PS1~0	T3/T3CKO	T3EX
0 0	P3.3	P3.4
0 1	P3.3	P3.2
1 0	P0.2	P0.1
1 1	P6.0	P6.1

AUXR9: 辅助寄存器 9

SFR 页 = 仅 6 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G1	T0G1	0	0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: SID/STWI 端口引脚选择[1:0].

SIDPS1~0	STWI_SCL	STWI_SDA
0 0	nINT1	S0MI
0 1	TWI0_SCL	TWI0_SDA
1 0	TWI1_SCL	TWI1_SDA
1 1	T2EXI	T3EXI

Bit 5: T1G1, 定时器 1 门控源选择.

T1G1, T1GATE	T1 门控源
00	禁止
01	INT1 激活
10	TF3 激活
11	TI1 激活

Bit 4: T0G1, 定时器 0 门控源选择

T0G1, T0GATE	T0 门控源
00	禁止
01	INT0 激活
10	TF2 激活
11	KBI 激活

Bit 3~2: 保留。写该寄存器时，此位必须写“0”。

Bit 1~0: S1PS1~0, Serial Port 1 端口引脚选择 [1:0].

S1PS1~0	RXD1	TXD1
0 0	P1.2	P1.3
0 1	P6.0	P6.1
1 0	P1.0	P1.1
1 1	P3.4	P3.5

AUXR10: 辅助寄存器 10

SFR 页 = 仅 7 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	SPIPS1	SPIPS0	S0PS1	P60OC1	P60OC0	P60FD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 保留。写该寄存器时，此位必须写“0”。

Bit 5~4: SPIPS1~0, SPI 端口引脚选择 1~0.

引脚选择	SPIPS1~0	nSS	MOSI	MISO	SPICLK
0	0 0	P1.4	P1.5	P1.6	P1.7
1	0 1	P0.1	P0.2	P4.1	P4.0
2	1 0	P3.3	P1.5	P1.6	P1.7
3	1 1	P1.7	P3.5	P3.4	P3.3

Bit 3: S0PS1, 串口 0 引脚选择位 1. (此功能在 AUXR3.3, S0PS0 中已描述)

Bit 2~1: P6.0 功能配置控制位 1 和位 0, 这两位仅仅当内部 RC 振荡(IHRCO 或 ILRCO)被选择为系统时钟源时有效。在晶振模式下, P6.0 和 P6.1 用于 XTAL2 和 XTAL1。外部时钟输入模式下, P6.0 专用于时钟输入。在内部振荡模式, P6.0 为普通 I/O 或时钟源发生器提供下列选项, 当 P60OC[1:0] 索引为非 P6.0 GPIO 功能时, P6.0 将驱动内部 RC 振荡器输出为其它设备提供时钟源

P60OC[1:0]	P6.0 功能	I/O 模式
0 0	P6.0	By P6M1.0 & P6M0.0
0 1	MCK	By P6M1.0 & P6M0.0
1 0	MCK/2	By P6M1.0 & P6M0.0
1 1	MCK/4	By P6M1.0 & P6M0.0

了解详情, 请参考“9 系统时钟”。6.0 作为时钟输出功能时, 建议设置{P6M1.0, P6M0.0}为“01”来选择 P6.0 为推挽输出模式。

Bit 0: P60FD, P6.0 快速驱动。

0: P6.0 默认驱动输出。

1: P6.0 快速驱动输出使能。若 P6.0 被配置为时钟输出, 当 P6.0 输出频率大于 12MHz(5V)或者大于 6MHz(3V) 时使能此位。

AUXR11: 辅助寄存器 11

SFR 页 = 仅 8 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	I2C1PS1	I2C1PS0	RX1S0	0	T1CKOE	T0CKOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 保留。写该寄存器时, 此位必须写“0”。

Bit 5~4: I2C1PS1~0, TWI1/I2C1 端口引脚选择[1:0].

I2C1PS1~0	TWI1_SCL	TWI1_SDA
0 0	P1.0	P1.1
0 1	P6.0	P6.1
1 0	P3.0	P3.1
1 1	TWI0_SCL	TWI0_SDA

Bit 3: RX1S0, RXD1 选择位 0.

0: RXD1 由 S1PS1~0 (AUXR9.1~0)选择.

1: RXD1 强制选择在 AC0OUT.

Bit 2: 保留。写该寄存器时, 此位必须写“0”。

Bit 1: T1CKOE, 定时器 1 时钟输出使能.

0: 禁止定时器 1 时钟输出

1: 使能定时器 1 时钟输出到 T1CKO 端口引脚.

Bit 0: T0CKOE, 定时器 0 时钟输出使能.

0: 禁止定时器 0 时钟输出

1: 使能定时器 0 时钟输出到 T0CKO 端口引脚.

AUXR12: 辅助寄存器 12

SFR 页 = 仅 9 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
CRCDS2	CRCM0	PDOES1	PDOES0	EDCM0	GPLC0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: CRCDS2. CRC0 数据端口选择位 2.

Bit 6: CRCM0, CRC 模式选择 0.

0: 选择 CRC16: 0x1021.

1: 选择 CRC32: 0x04C1_1DB7.

Bit 5~4: PDOES1~0, PDTXOE1 端口引脚选择 [1:0].

PDOES1~0	PDTXOE
0 0	P1.5
0 1	P3.3
1 0	P6.1
1 1	P3.2

Bit 3: EDCM0. EDC45 模式选择 0.

EDCM0	Mode #	EDC45 操作
0	0	4-bit to 5-bit encoder
1	1	5-bit to 4-bit decoder

Bit 2: GPLC0. GPL 控制 0.

0: CPU 在 SFR 地址 0xA9 访问 SADDR

1: CPU 在 SFR 地址 0xA9 访问 EDC45.

Bit 1~0: 保留。写该寄存器时，此位必须写“0”。

AUXR16: 辅助寄存器 16

SFR 页 = 仅 D 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 保留。写该寄存器时，此位必须写“0”。

Bit 5~0: C0OPS5 ~0, PTM 端口引脚选择.

	C0OPS5	C0OPS4	C0OPS3	C0OPS2	C0OPS1	C0OPS0
C0OPSn	CEX5	CEX4	CEX3	CEX2	CEX1	CEX0
0	P0.7	P2.6	P0.5	P2.4	P2.3	P2.2
1	P3.5	P3.2	P3.4	P4.1	P3.3	P4.0

AUXR17: 辅助寄存器 17

SFR 页 = 仅 E 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	0	C0OPS1	C0OPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~2: 保留。写该寄存器时，此位必须写“0”。

Bit 1~0: C1OPS1 ~0, PCA1 输出端口引脚选择

	C1OPS1	C1OPS0
C1OPSn	C1EX1	C1EX0
0	P6.1	P6.0
1	P3.1	P3.0

SFRPI: SFR 页面索引寄存器

SFR 页 = 0~F

SFR 地址 = 0xAC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	IDX3	IDX2	IDX1	IDX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: 保留。写该寄存器时，此位必须写“0”。

Bit 3~0: SFR 页面索引。

IDX[3:0]	可选页
0000	Page 0
0001	Page 1
0010	Page 2
0011	Page 3
.....	
.....	
.....	
1111	Page F

35. 硬件选项

MCU 的硬件选项定义了器件的性能，它不能由软件编程和控制。硬件选项仅能由通用编程器，“Megawin 8051 Writer U1”或“MLink”(这个 ICE 也支持 ICP 编程功能。参考章节“[36.5 在芯片编程功能](#)”)来编程。整片擦除后，所有的硬件选项被设置成“禁止”状态，没有配置 ISP 空间和 IAP 空间。**MG82F6P32** 有下列的硬件选项：

LOCK:

- ☒: 使能。加密上锁，使得用通用编程器读取代码锁定为 0xFF。
- ☐: 禁止。没有上锁。

ISP-存储空间:

由其指定 ISP 空间的起始地址。它的高边界由 Flash 的结束地址限定，例如：**0x3FFF**。下表列举了 ISP 空间选项。默认设定，**MG82F6P32** 空间被配置为 **1.5K**，并嵌入了 Megawin COMBO ISP 引导码,通过 Megawin 1-线 ISP 协议和串口 ISP 协议，进行在线设备 FW 更新。

ISP-空间大小	MG82F6P32 ISP 起始地址
7.5K bytes	6200
7.0K bytes	6400
6.5K bytes	6600
6.0K bytes	6800
5.5K bytes	6A00
5.0K bytes	6C00
4.5K bytes	6E00
4.0K bytes	7000
3.5K bytes	7200
3.0K bytes	7400
2.5K bytes	7600
2.0K bytes	7800
1.5K bytes	7A00
1.0K bytes	7C00
0.5K bytes	7E00
No ISP Space	--

HWBS:

- ☒: 使能。上电时，如果 ISP 空间有配置，则 MCU 从 ISP 空间启动。
- ☐: 禁止。MCU 总是从 AP 空间启动。

HWBS2:

- ☒: 使能。如果 ISP 空间有配置，不仅上电，而且所有复位都是从 ISP 空间启动。
- ☐: 禁止。由 HWBS 决定 MCU 从哪里启动。

IAP-存储空间:

IAP 存储空间指定用户定义的 IAP 空间。IAP 存储空间可以由硬件选项或者 MCU 软件修改 IAPLB 来配置。默认，被配置为 1K。

BO1S10, BO1S00:

- ☒☒: 选择 BOD1 监测电压 2.0V.
- ☒☐: 选择 BOD1 监测电压 2.4V.
- ☐☒: 选择 BOD1 监测电压 3.6V.
- ☐☐: 选择 BOD1 监测电压 4.2V.

BO0REO:

- ☒: 使能。BOD0 将触发复位事件使得 CPU 从 AP 程序起始地址运行(1.7V)。
- ☐: 禁止。BOD0 不能触发 CPU 复位。

BO1REO:

- ☒: 使能。BOD1 (4.2V、3.6V、2.4V 或 2.0V)将触发复位事件使得 CPU 从 AP 程序起始地址运行。
- ☐: 禁止。BOD1 不能触发 CPU 复位。

WRENO:

- ☒: 使能。置位 WDTCR.WREN 使能 WDTF 产生一个系统复位。
- ☐: 禁止。清零 WDTCR.WREN 禁止 WDTF 产生一个系统复位。

NSWDT: 不停止 WDT

- ☒: 使能。置位 WDTCR.NSW 在掉电模式下使能 WDT 运行(watch 模式)。
- ☐: 禁止。清零 WDTCR.NSW 在掉电模式下禁止 WDT 运行(禁止 Watch 模式)。

HWENW: 硬件加载“ENW”到 WDTCR。

- ☒: 使能。上电后使能 WDT 并且加载 WRENO、NSWDT、HWWIDL 和 HWPS2~0 的内容到 WDTCR。
- ☐: 禁止。上电后 WDT 不会自动使能。

HWWIDL, HWPS2, HWPS1, HWPS0:

当 HWENW 使能，上电后这 4 个熔丝位的内容将被加载到 WDTCR。

WDSFWP:

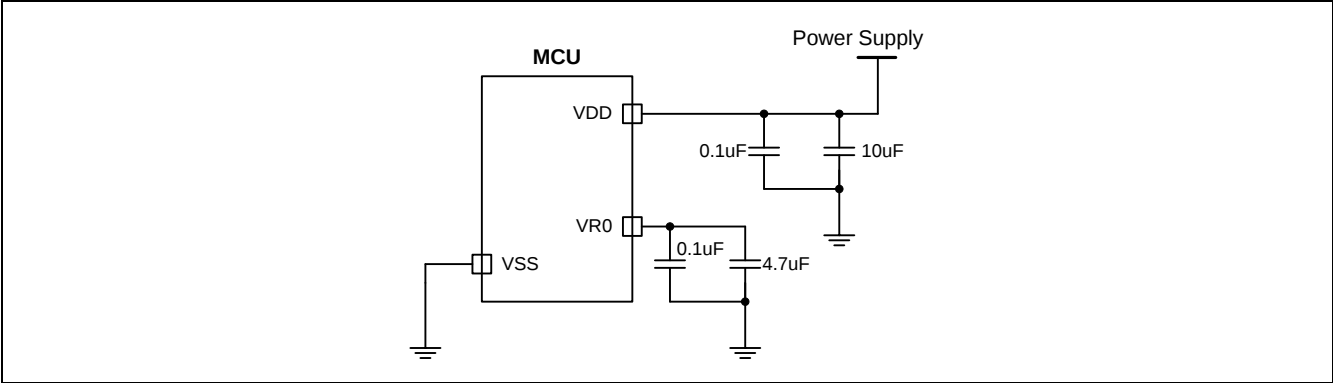
- ☒: 使能。WDT 特殊寄存器，WDTCR 的 WREN、NSW、WIDL、PS2、PS1 和 PS0 位，将被写保护。
- ☐: 禁止。WDT 特殊寄存器，WDTCR 的 WREN、NSW、WIDL、PS2、PS1 和 PS0 位，由软件自由写。

36. 应用说明

36.1. 电源电路

MG82F6P32 的工作电源变化可以从 2.0V 到 5.5V，但是增加一些外部去耦和滤波电容是必须的，如图 36-1 所示。

图 36-1. 电源电路



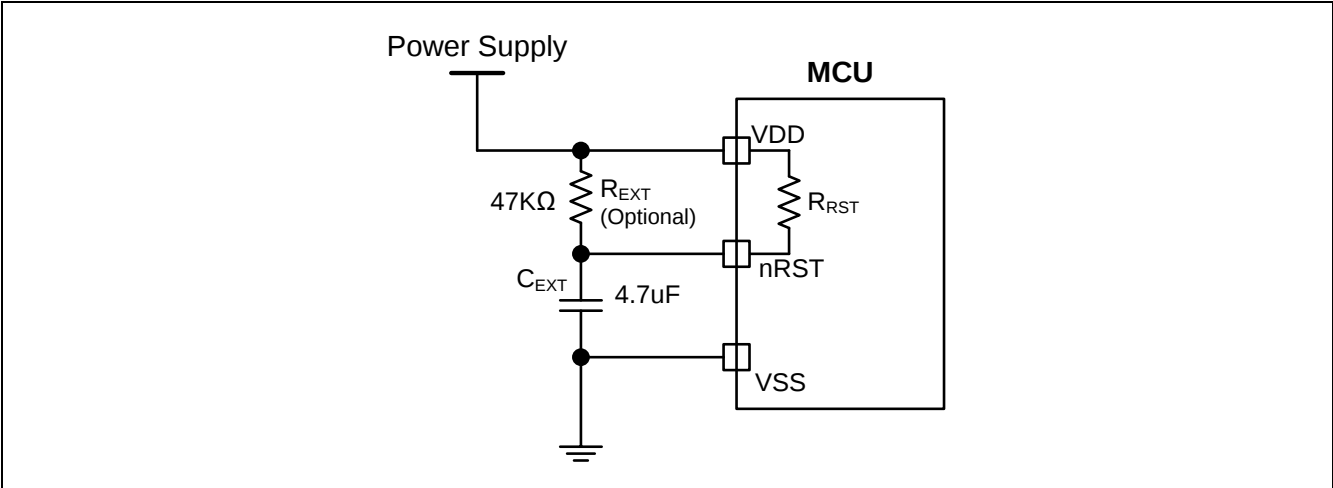
36.2. 复位电路

通常，上电可以成功产生上电复位。然而，为了上电时 MCU 产生一个可靠的复位，有必要加外部复位。外部复位电路如图 36-2 所示，它由一个连接到 VSS(地)的电容 C_{EXT} 和一个连接到 VDD(电源)的电阻组成。

一般的， R_{EXT} 是可选的，因为 $nRST$ 引脚有一个内部上拉电阻(R_{RST})。这个内部上拉电阻允许上电复位只使用一个外部电容 C_{EXT} 到 VSS。

R_{RST} 的值见章节“37.2 直流特性”

图 36-2. 复位电路



36.3. XTAL 振荡电路

为了实现成功和精确的振荡(高达 24MHz)，需要电容 C1 和 C2，如图 36-3 所示。通常，C1 和 C2 的值相同。不同频率晶体应用时，C1 和 C2 值如表 36-1 所示。

图 36-3. XTAL 振荡电路

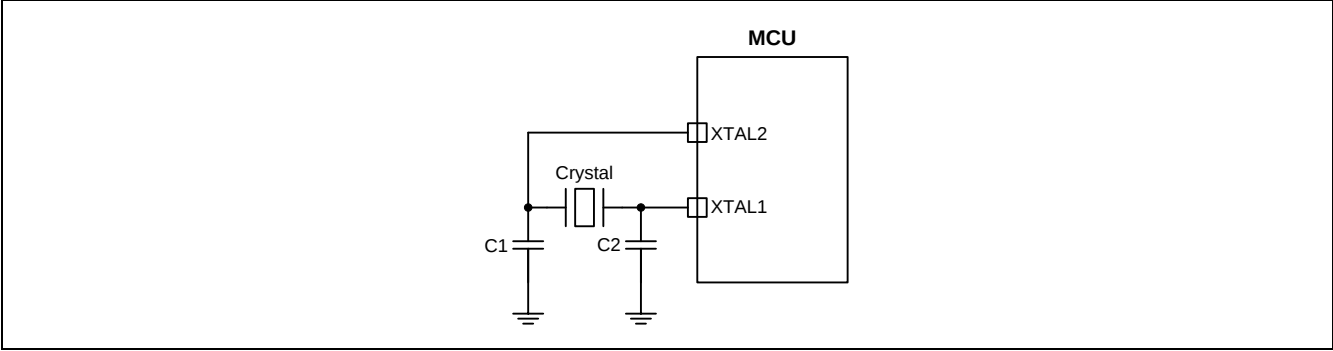


表 36-1. 晶体振荡电路 C1 和 C2 的参考电容

晶振	C1, C2 电容
16MHz ~ 25MHz	10pF
6MHz ~ 16MHz	15pF
2MHz ~ 6MHz	33pF
32768Hz	7pF

36.4. ICP 和 OCD 接口电路

MG82F6P32 包含一个笙泉专有的在芯片调试接口，它允许在元器件已经安装在产品上在芯片编程(ICP)和在线调试(OCD)。ICP 和 OCD 共享同样的接口使用一个时钟线(ICP_SCL/OCD_SCL)和一个双向数据线(ICP_SDA/OCD_SDA)完成主机与设备之间的数据传送。

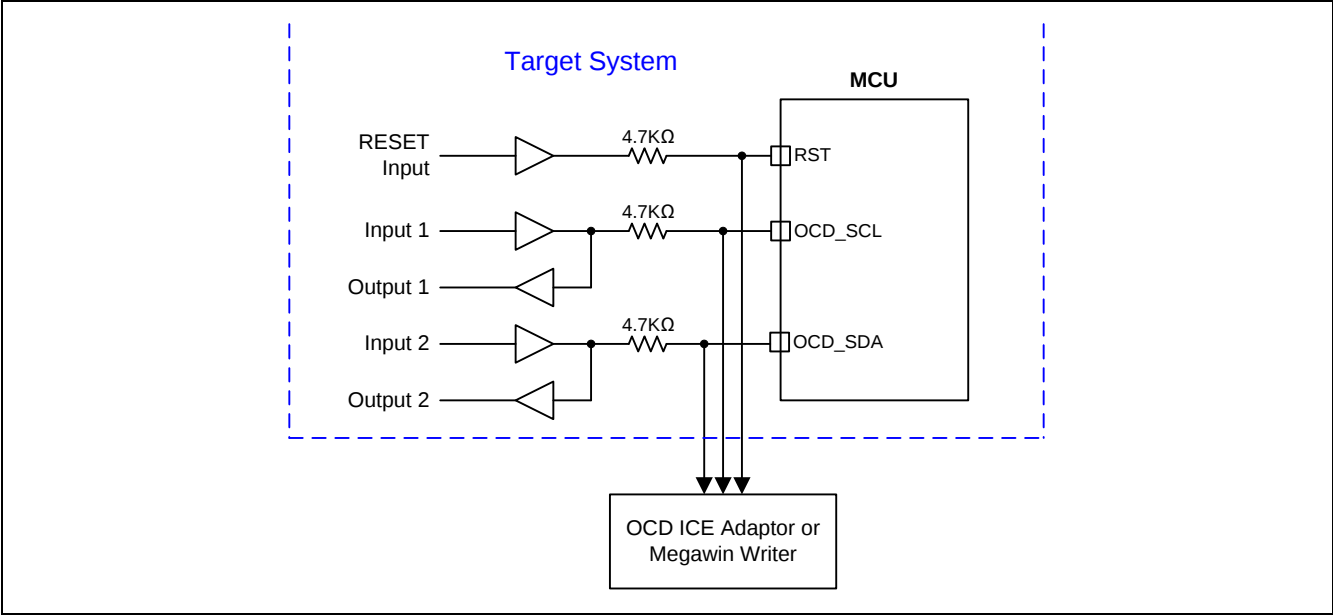
ICP 接口允许的 ICP_SCL/ICP_SDA 引脚与用户应用共享，使得可以实现在芯片 FLASH 编程。这是可行的，因为当芯片在 Halt 状态时执行 ICP 通信，此时芯片上的外围设备和用户软件都是失效的。在 halt 状态，ICP 接口能够安全的“借用” ICP_SCL (P4.4)和 ICP_SDA (P4.5)引脚。在大多应用中，必须用外部电阻来隔开 ICP 电路和用户应用电路。一种典型的隔离方法如图 36-4 所示。

强烈建议在目标系统建立 ICP 接口电路。它保留了整个软件编程和硬件选项配置的能力。

上电后，**MG82F6P32** 的 P4.4 和 P4.5 被配置成 OCD_SCL/OCD_SDA 用于在线调试功能。这是可行的，因为 OCD 通信是在 CPU Halt 状态下执行，此时用户软件是无效的。在 halt 状态，OCD 接口可以安全的使用 OCD_SCL(P4.4)和 OSC_SDA(P4.5)引脚。就像上面提到的隔离 ICP 接口，如图 36-4，用外部电阻来隔开 ICP 电路和用户应用电路。

如果用户放弃 OCD 功能，软件可以通过清零 PCON3 的位 0(OCDE)来配置 OCD_SCL 和 OCD_SDA 引脚作为 P4.4 和 P4.5。当用户想重新使用 OCD 功能，用户可以置 OCDE 为 1 来切换 P4.4 和 P4.5 到 OCD_SCL 和 OCD_SDA。或者用 ICP “擦除”在芯片 FLASH 清除用户软件来停止端口的切换。

图 36-4. ICP 和 OCD 接口电路



36.5. 在芯片编程功能

ICP，就像传统的并行编程方式，可以编程 MCU 的任何区域，包括 FLASH 和 MCU 的硬件选项。并且，得益于它专用的串行接口(经由在线调试通道)，使得 ICP 可以更新 MCU 而不用从用户的产品上卸下 MCU，就像 ISP 做的那样。

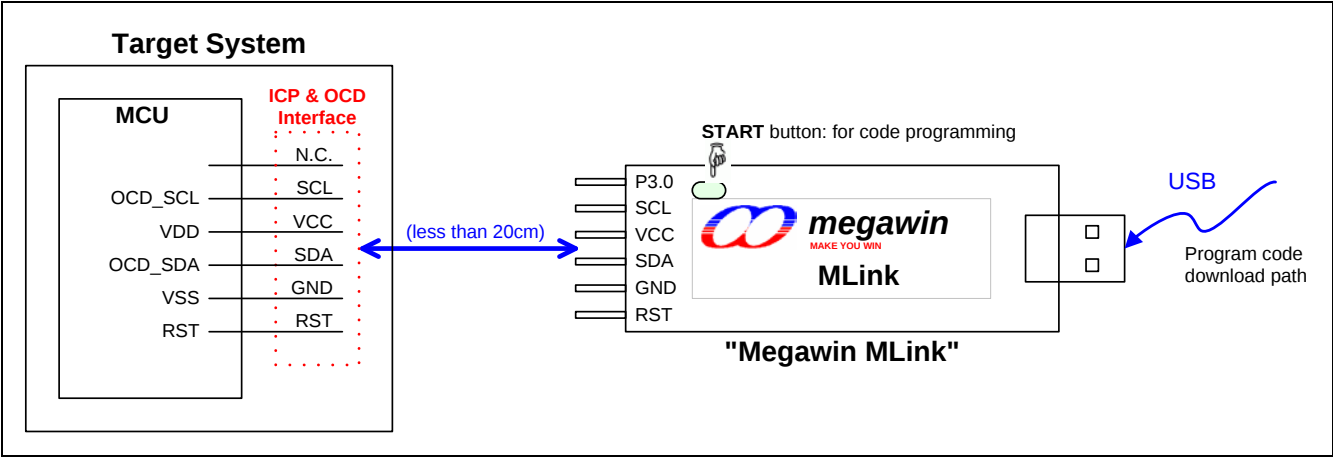
专用的 6 脚“MLink”可以支持 **MG82F6P32** 在线路编程。“MLink”有在系统的存储器来存储用户的程序和器件选项。因此，该工具可以完成一个便携的，独立的编程，而不用连线主机，如连接该工具到 PC。下面列举了 ICP 功能的特点：

特点

- 不必在目标芯片上预编程一个引导程序。
- 专用串行接口；不占用 IO 口。
- 目标芯片不必在运行状态；仅需电源。
- 便携，独立的工作，而无需主机的干预。

以上特点使得 ICP 非常有利于用户。特别的，在编程数据下载后的便携独立工作，尤其有利于没有 PC 的地方使用。ICP 独立编程的系统框图如图 36-5 所示。ICP 接口仅需 5 个引脚：SDA 线和 SCL 线是串行数据和串行时钟，用来从 6 脚“MLink”传送编程数据到目标 MCU；RST 线用来暂停 MCU；VCC 和 GND 是 6 脚“MLink”用于便携编程应用的电源输入。USB 连接器可以直接的插入 PC 的 USB 端口，用来从 PC 下载编程数据到 6 脚“MLink”。

图 36-5. 经 ICP(MLink)的独立编程



36.6. 在线调试功能

MG82F6P32 预备了一个用于在线仿真(ICE)的 Megawin 专用的在线调试(OCD)接口。这个 OCD 接口提供在芯片和系统不干扰的调试，且不占用任何的目标系统资源。支持 ICE 的几种必要操作，如复位，运行，停止，单步运行，运行到光标和断点设置。

使用 OCD 技术，Megawin 提供 “MLink” 给用户，如图 36-6 所示。用户在开发过程中不必准备任何的开发板，或者用在传统 ICE 探头的转换座。所有这些，用户仅需在系统上保留一个 6-脚的连接器用于专用的 OCD 接口：P3.0、RST、VCC、OCD_SDA、OCD_SCL 和 GND，如图 36-6 所示

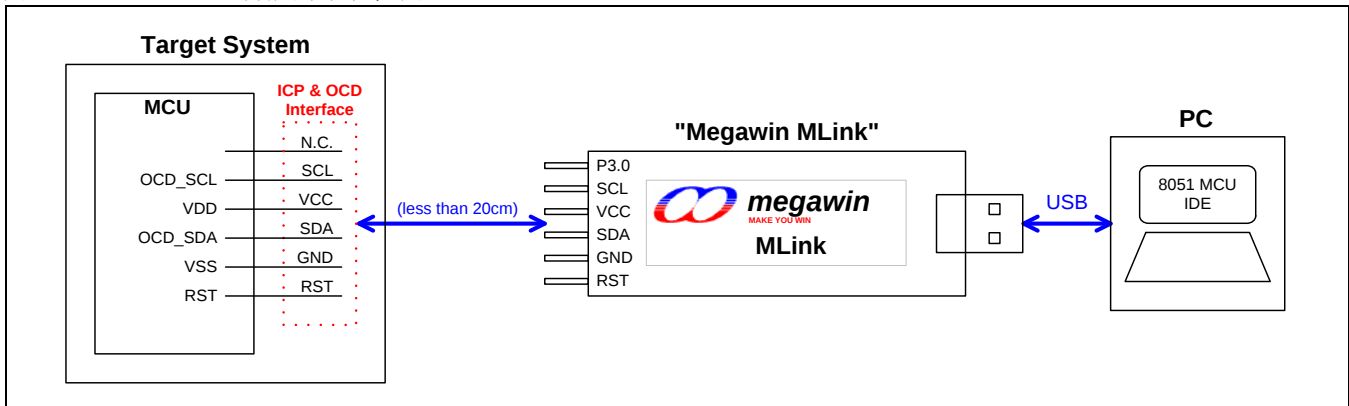
另外，最有力的功能是，它可以直接让用户的系统连接到 Keil 8051 IDE 软件进行仿真，它直接利用 Keil IDE’ s dScope-Debugger 功能。当然，所有的特点都基于你使用的 Keil 8051 IDE 软件。

注意：“Keil” 是 “Keil Elektronik GmbH and Keil Software, Inc.” 的注册商标。

特点

- 笙泉科技专用的 OCD(在芯片调试)技术
- 在芯片和在系统实时调试
- 5 针 OCD 专用串行接口，不占用目标资源
- 直接连接 Keil IDE 软件的调试功能
- USB 连接目标板与主机(PC)
- 有用的调试操作：复位，运行，停止，单步运行和运行到光标
- 可编程断点，可在仿真中插入 4 个断点
- 数个帮助调试串口：寄存器/反汇编/监视/存储区窗口
- 源代码级(汇编或 C 语言)调试能力

图 36-6. MLink ICE 功能的系统框图



注意：有关 MLink 的更多详细信息，请联系笙泉。

37. 电气特性

37.1. 最大绝对额定值

参数	范围	单位
环境温度	-40 ~ +105	°C
存储温度	-65 ~ + 150	°C
任意 GPIO 口或 RST 对地电压	-0.5 ~ VDD + 0.5	V
VDD 对地电压	-0.5 ~ +6.0	V
VDD 到地的最大电流	200	mA
任意引脚最大灌电流	40	mA

*注意：实际参数超过上述各项“绝对最大额定值”可能会对设备造成永久性损坏。这些参数是一个设备进行正常功能操作的应力额定值，任何超过上述各项的条件都不被建议，否则可能会影响设备运行的稳定性。

37.2. 直流特性

VDD = 5.0V±10%, VSS = 0V, T_A = 25 °C, 每个机器周期执行 NOP, 除非另有说明

标号	参数	测试环境	极限			单位
			最小	典型	最大	
输入/输出特性						
V _{IH1}	输入高电平(所有 I/O 口)	Except P6.0, P6.1	0.6			VDD
V _{IH2}	输入高电平(nRST、P6.0、P6.1)		0.75			VDD
V _{IL1}	输入低电平(所有 I/O 口)	Except P6.0, P6.1			0.15	VDD
V _{IL2}	输入低电平(nRST、P6.0、P6.1)				0.2	VDD
I _{IH}	输入高漏电流 (所有 I/O 口)	V _{PIN} = VDD		0	±1	uA
I _{IL1}	逻辑 0 输入电流(P3 在准双向口模式或片内上拉电阻的输入端口)	V _{PIN} = 0.4V		-20	-30	uA
I _{IL2}	逻辑 0 输入电流(所有仅输入或开漏输出口)	V _{PIN} = 0.4V		0	-1	uA
I _{H2L}	逻辑 1 到 0 输入转变电流 (P3 在准双向口模式) ⁽²⁾	V _{PIN} =1.8V		-300	-450	uA
I _{OH1}	输出高电流(P3 在准双向口模式)	VDD=5V; V _{PIN} =2.4V	-180	-260		uA
		VDD=3.3V; V _{PIN} =2.4V	-50	-80		uA
		VDD=1.8V; V _{PIN} =1.4V	-10	-15		uA
I _{OH2}	输出高电流(所有推挽输出口)	VDD=5V; V _{PIN} =2.4V	-30	-34		mA
		VDD=3.3V; V _{PIN} =2.4V	-8	-11		mA
		VDD=1.8V; V _{PIN} =1.4V	-2	-2.6		mA
I _{OH3}	输出高电流(在低驱动力时所有推挽输出口, 除 nRST 引脚外)	VDD=5V; V _{PIN} =2.4V	-8	-13.6		mA
		VDD=3.3V; V _{PIN} =2.4V	-3	-4.6		mA
		VDD=1.8V; V _{PIN} =1.4V	0.7	-1.1		mA
I _{OL1}	输出低电流(所有 I/O 口)	VDD=5V; V _{PIN} =0.4V	18	24		mA
		VDD=3.3V; V _{PIN} =0.4V	14	17		mA
		VDD=1.8V; V _{PIN} =0.4V	6	8		mA
I _{OL2}	输出低电流(在低驱动力时所有推挽输出口, 除 nRST 引脚外)	VDD=5V; V _{PIN} =0.4V	1.8	3.1		mA
		VDD=3.3V; V _{PIN} =0.4V	1.2	2.2		mA
		VDD=1.8V; V _{PIN} =0.4V	0.55	1.1		mA
R _{RST}	内部复位上拉电阻	VDD=5V	265	270	290	Kohm
		VDD=3.3V	450	460	490	Kohm
		VDD=2.1V	1430	1500	1600	Kohm
功耗						
I _{OP1}	正常模式工作电流	CPUCLK=SYSCLK = 32MHz @ IHRCO with PLL, HSE1=HSE=1		7.5		mA
I _{OP2}		CPUCKL=SYSCLK = 24MHz @ IHRCO with PLL, HSE1=0, HSE=1		6.5		mA
I _{OP3}		CPUCLK=SYSCLK = 24MHz @ XTAL 24MHz , HSE1=0, HSE=1		7.5		mA
I _{OP4}		CPUCLK=SYSCLK = 24MHz @ ECKI 24MHz , HSE1=0, HSE=1		5.8		mA
I _{OP5}		CPUCKL=SYSCLK = 12MHz @ IHRCO, HSE1=0, HSE=0		3.8		mA
I _{OP6}		CPUCLK=SYSCLK = 12MHz @ XTAL 12MHz , HSE1=0, HSE=0		4.6		mA
I _{OP7}		CPUCLK=SYSCLK = 12MHz @ ECKI 12MHz , HSE1=0, HSE=0		3.6		mA
I _{OP8}		CPUCLK=SYSCLK = 2MHz @ XTAL 2MHz		1.8		mA
I _{OP9}		CPUCLK=SYSCLK = 4MHz @ XTAL 4MHz		2.2		mA
I _{OP10}		CPUCLK=SYSCLK = 6MHz @ XTAL 6MHz		2.8		mA
I _{OP11}		CPUCLK=SYSCLK = 32KHz @ XTAL 32KHz		65		mA
I _{OPS1}	低速模式工作电流	SYSCLK = 12MHz/128 @ IHRCO		2.8		mA
I _{IDLE1}	空闲模式工作电流	SYSCLK = 12MHz @ IHRCO		1.7		mA
I _{IDLE2}		SYSCLK = 12MHz/128 @ IHRCO		0.38		mA

I _{IDLE3}		SYSCLK = 32KHz @ ILRCO		51		uA
I _{IDLE4}		SYSCLK = 32KHz @ XTAL 32KHz		57		uA
I _{SUB1}	副频模式工作电流	SYSCLK = 32KHz @ ILRCO, BOD1 disabled		4.2		uA
I _{SUB2}		SYSCLK = 32KHz/128 @ ILRCO, BOD1 disabled		4		uA
I _{WAT}	Watch 模式工作电流	WDT = 32KHz @ ILRCO in PD mode		2.5		uA
I _{MON1}	Monitor 模式工作电流	BOD1 enabled in PD mode		8.6		uA
I _{RTC1}	RTC 模式操作电流	RTC operating in PD mode, VDD = 5.0V		2.55		uA
I _{PD1}	掉电模式电流			1.5	6	uA
BOD0/BOD1 特性						
V _{BOD0}	BOD0 监测电压	T _A = -40°C to +105°C		1.7		V
V _{BOD10}	BOD1 监测电压为 2.0V	T _A = -40°C to +105°C		2.0		V
V _{BOD10}	BOD1 监测电压为 2.4V	T _A = -40°C to +105°C		2.37		V
V _{BOD11}	BOD1 监测电压为 3.6V	T _A = -40°C to +105°C		3.6		V
V _{BOD11}	BOD1 监测电压为 4.2V	T _A = -40°C to +105°C		4.2		V
I _{BOD1}	BOD1 功耗	T _A = +25°C, VDD=5.0V		6.5		uA
		T _A = +25°C, VDD=3.3V		5		
工作环境						
V _{PSR}	上电边沿速率	T _A = -40°C to +85°C	0.05			V/ms
V _{POR1}	上电复位有效电压	T _A = -40°C to +85°C			0.1	V
V _{OP1}	CPU 工作速度 0-32MHz	T _A = -40°C to +105°C	2.7		5.5	V
V _{OP2}	CPU 工作速度 0-24MHz	T _A = -40°C to +105°C	2.2		5.5	V
V _{OP3}	CPU 工作速度 0-12MHz	T _A = -40°C to +105°C	1.8		5.5	V

- (1) 数据基于特性所得, 非产品测试。
- (2) I/O 在准双向模式下, 当输入电压从高转向低并且越过阈值电压, 内部“弱上拉”将关闭。I_{H2L} 表示的是电流接近阈值电压时的电流, 详见参考““图 14-1. 端口 3 准双向 I/O””。
- (3) 所有电流流入芯片是正值, 电流流出芯片为负值。

37.3. 外部复位特性

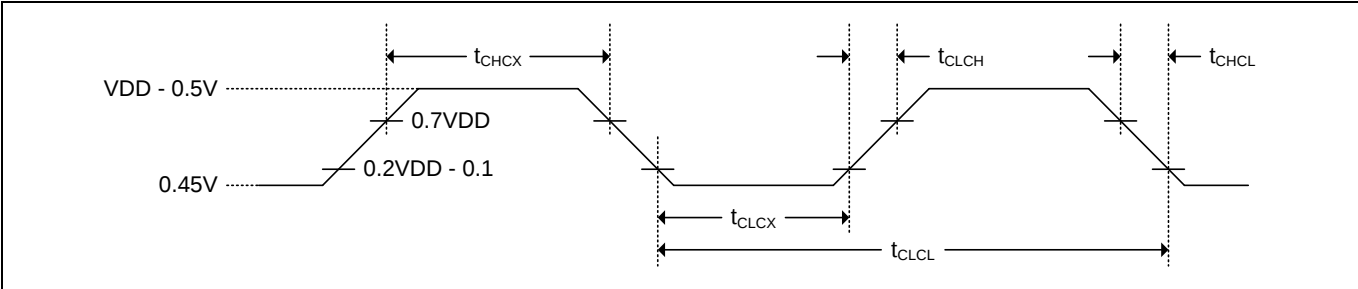
参数	测试环境	极限			单位
		最小	典型	最大	
电源电压		1.8		5.5	V
外部 nRST 产生系统复位的低电平时间	SYSCLK = IHRCO	32			uS

37.4. 外部时钟特性

VDD = 2.4V ~ 5.5V, VSS = 0V, TA = -40°C to +105°C, 除非另有说明

标号	参数	晶振				单位
		晶振模式		ECKI 模式		
		最小	最大	最小	最大	
1/t _{CLCL}	晶振频率	0.032	25	0	25	MHz
1/t _{CLCL}	晶振频率 (VDD = 2.0V ~ 5.5V)	0.032	12	0	12	MHz
t _{CLCL}	时钟周期	41.6		27.7		ns
t _{CHCX}	高时间	0.4T	0.6T	0.4T	0.6T	t _{CLCL}
t _{CLCX}	低时间	0.4T	0.6T	0.4T	0.6T	t _{CLCL}
t _{CLCH}	上升时间		5		5	ns
t _{CHCL}	下降时间		5		5	ns

图 37-1. 外部时钟驱动波形



37.5. IHRCO 特性

参数	测试环境	极限			单位
		最小	典型	最大	
电源电压		1.8		5.5	V
IHRCO 频率	TA = +25°C, AFS = 0		12		MHz
	TA = +25°C, AFS = 1		11.059		MHz
IHRCO 频率误差 (工厂校对)	TA = +25° C	-1.0		+1.0	%
	TA = -40° C to +105° C	-2.0 ⁽¹⁾		+2.0 ⁽¹⁾	%
IHRCO 启动时间	TA = -40° C to +105° C			32 ⁽¹⁾	us
IHRCO 功耗	TA = +25° C, VDD=5.0V		(1)		uA

⁽¹⁾ 数据基于特性所得, 非产品测试。

37.6. ILRCO 特性

参数	测试环境	极限			单位
		最小	典型	最大	
电源电压		1.8		5.5	V
ILRCO 频率	TA = +25° C		32		KHz
ILRCO 频率误差	TA = +25° C	-8 ⁽¹⁾		+8 ⁽¹⁾	%
	TA = -40° C to +105° C	-15 ⁽¹⁾		+15 ⁽¹⁾	%

⁽¹⁾ 数据基于特性所得, 非产品测试。

37.7. CKM 特性

参数	测试环境	极限			单位
		最小	典型	最大	
电源电压	TA = -40° C to +105° C	2.2		5.5	V
时钟输入范围	TA = -40° C to +105° C	4.5 ⁽¹⁾		6.5 ⁽¹⁾	MHz
CKM 启动时间	TA = -40° C to +105° C	30 ⁽²⁾		100 ⁽²⁾	us
CKM 功耗	TA = +25° C, VDD=5.0V, CKM = 96MHz		350		uA

(1) 数据由设计保证，非产品测试。

(2) 数据基于特性所得，非产品测试。

37.8. Flash 特性

参数	测试环境	极限			单位
		最小	典型	最大	
电源电压	TA = -40° C to +105° C	1.8		5.5	V
Flash 写 (擦除/编程)电压	TA = -40° C to +105° C	1.8		5.5	V
Flash 擦除/编程 周期	TA = -40° C to +105° C	20,000			times
Flash 数据保留期	TA = +25° C	100			year

37.9. OPA 特性

VDD=5.0V, TA= -40° C ~ +85° C, 除非另有说明

参数	测试环境	极限			单位
		最小	典型	最大	
电源电压		2.4	--	5.5	V
单个 OPA I _Q	高功率, OPPWR:OPnLP = 00	0.8	1.13	1.45	mA
	中高功率, OPPWR:OPnLP = 01	--	0.65	--	
	中低功率, OPPWR:OPnLP = 10	--	0.55	--	
	功率, OPPWR:OPnLP = 11	0.15	0.3	0.35	
偏移 V _{OS}	校准@ Vi = 1.2V		±0.5	±1.0	mV
V _{CM}	高功率, 增益误差=±0.5%	0.53	--	VDD-0.1	V
	高功率, 增益误差=±1.0%	0.30	--	VDD-0.1	
	中高功率, 增益误差=±0.5%	0.53	--	VDD-0.05	
	中高功率, 增益误差=±1.0%	0.30	--	VDD-0.05	
	中低功率, 增益误差=±0.5%	0.53	--	VDD-0.05	
	中低功率, 增益误差=±1.0%	0.30	--	VDD-0.05	
	低功率, 增益误差=±0.5%	0.63	--	VDD-0.05	
	低功率, 增益误差=±1.0%	0.43	--	VDD-0.05	
V _{OL}	高功率	0.1	--	--	V
	中高功率	0.13	--	--	
	中低功率	0.1	--	--	
	低功率	0.13	--	--	
V _{OH}	高功率	--	--	VDD-0.12	V
	中高功率	--	--	VDD-0.15	
	中低功率	--	--	VDD-0.15	
	低功率	--	--	VDD-0.20	
带宽	高功率, C _{LOAD} // R _{LOAD} = 110pF // 2k ohm (peaking, Gain=110%)	15.5	19.0	--	MHz
	高功率, C _{LOAD} // R _{LOAD} = 30pF // 2k ohm	--	--	25.0	
	高功率, C _{LOAD} // R _{LOAD} = 20pF // 2k ohm	--	--	22.0	

	中 高功率, $C_{LOAD} // R_{LOAD} = 110pF // 2k\ ohm$ (peaking ,Gain=130%)	10.5	13.0	--	
	中 低功率, $C_{LOAD} // R_{LOAD} = 110pF // 2k\ ohm$ (peaking ,Gain=116%)	8.3	10.3	--	
	低功率, $C_{LOAD} // R_{LOAD} = 110pF // 2k\ ohm$ (peaking ,Gain=136%)	8.5	9.4	--	
变化率	高功率, $C_{LOAD} // R_{LOAD} = 110pF // 2k\ ohm$ (from 20 and 80% of output voltage)	--	7.0	--	V/us
	中 高功率, $C_{LOAD} // R_{LOAD} = 110pF // 2k\ ohm$ (from 20 and 80% of output voltage)	--	7.5	--	
	中 低功率, $C_{LOAD} // R_{LOAD} = 110pF // 2k\ ohm$ (from 20 and 80% of output voltage)	--	2.7	--	
	低功率, $C_{LOAD} // R_{LOAD} = 110pF // 2k\ ohm$ (from 20 and 80% of output voltage)	--	3.6	--	
稳定时间		--	--	2.5	us
AOL					dB
CMRR					dB
PSRR					dB

注 1: 数据由设计保证, 非产品测试。

37.10. PGA 特性

VDD=5.0V, $T_A = -40^{\circ}C \sim +85^{\circ}C$, 除非另有说明

参数	测试环境	极限			单位
		最小	典型	最大	
电源电压		2.4	--	5.5	V
单个 OPA I_Q	正常功耗	0.45	0.56	0.69	mA
	低功耗	0.22	0.28	0.35	
偏移 V_{OS}	1 st	--	--	± 4.40	mV
	2 ^{ed}	--	± 0.4	± 1.0	
V_{CM} (软件校准)	G=1	0.028		VDD-0.8	V
	G=2	0.014		VDD-2.5	
	G=4	0.010	1	--	
	G=8	0.009	0.6	--	
	G=16	0.008	0.31	--	
	G=32	0.007	0.155	--	
	G=64	0.006	0.077	--	
	G=128	0.005	0.038	--	
V_{OL} (软件校准)		0.002	--	--	V
V_{OH} (软件校准)		--	--	VDD-0.1	V
正常功耗模式带宽	G=1		3.3		MHz
	G=2		7		
	G=4		5		
	G=8		4		
	G=16		2		
	G=32		3		
	G=64		--		
	G=128		--		
变化率	正常功耗	--	15	--	V/us
	低功耗	--	7.3	--	
稳定时间		--	--	1	us
CMRR					dB
PSRR					dB

注: 数据由设计保证, 非产品测试。

37.11. ADC 特性

VDD=5.0V, T_A= -40° C ~ +85° C 除非另有说明

参数	测试环境	极限			单位	
		最小	典型	最大		
电源电压		2.4		5.5	V	
分辨率			12		bits	
DC 精度 T _A = 25° C						
积分非线性	VDD = VREF+ ≥ 5V, 1M sps 功率级别: 高和中高	-2.4		2.9	LSB	
	VDD = VREF+ ≥ 2.4V, 500K sps 功率级别: 高和中高	-3.3		2.6		
差分非线性	VDD = VREF+ ≥ 5V, 1M sps 功率级别: 高和中高	-1		3.3	LSB	
	VDD = VREF+ ≥ 2.4V, 500K sps	-1		2		
偏移误差	VDD= 2.4V~5.5V		1	3.5	LSB	
DC 精度 T _A = -40° C ~ +85° C						
积分非线性	VDD = VREF+ ≥ 5V, 500K sps 功率级别: 高	-2.9		3	LSB	
	VDD = VREF+ ≥ 2.4V, 250K sps 功率级别: 高和中高	-4		3.1		
差分非线性	VDD = VREF+ ≥ 5V, 500K sps 功耗级别: 高	-1		3.5	LSB	
	VDD = VREF+ ≥ 2.4V, 250K sps	-1		2.1		
偏移误差	VDD= 2.4V~5.5V		1	3.5	LSB	
转换率						
SAR 转换时钟			12	24	MHz	
在 SAR 时钟里的转换时间			24		clocks	
建议转换率			500	1000	K sps	
模拟输入						
输入电压范围	单端点(AIN+ – GND)	0		VDD	V	
C _{ADC} 输入电容 ^{note1}			5		pF	
输入采样开关电阻 ^{note1}	VDD = 5V	Vin=0.1V	450	525	610	Ω
		Vin=2.5V	660	800	1000	Ω
		Vin=5V	950	1100	1300	
	VDD = 2.4V	Vin=0.1V	690	830	995	Ω
		Vin=1.2V	1.95	3.66	13	KΩ
		Vin=2.4V	1.5	1.9	2.4	KΩ
通道切换稳定时间						
原引脚和目标引脚电压在 VDD 与 GND 之间切换	CH0(VDD)→CH1(GND)		0		us	
	CH0(GND)→CH1(VDD)		0			
原引脚电压=VDD 切换到带下拉电阻的目标引脚	CH0(VDD)→CH1(51K 下拉)		4.41			
	CH0(VDD)→CH1(10K 下拉)		0.5			
原引脚电压=GND 切换到带上拉电阻的目标引脚	CH0(GND)→CH1(51K 上拉)		7.5			
	CH0(GND)→CH1(10K 上拉)		1.25			
原引脚电压=VDD 切换到分压电阻	CH0(VDD)→CH1(VDD/2, 51K 电阻分压)		3.91		us	
	CH0(VDD)→CH1(VDD/2, 10K 电阻分压)		3.08			
原引脚电压=GND 切换到分压电阻 (VDD/2)	CH0(GND)→CH1(VDD/2, 51K 电阻分压)		0.33			
	CH0(GND)→CH1(VDD/2, 10K 电阻分压)		0.5			
功耗						
电源电流	ADPS<1:0>=00	1.2	1.4	1.8	mA	
	ADPS<1:0>=01	--	1.38	--		
	ADPS<1:0>=10	--	1.37	--		
	ADPS<1:0>=11	--	1.33	--		

Note1: 数据由设计保证, 非产品测试。

37.12. IVR 特性

VDD=5.0V±10%, VSS=0V, TA =-40°C to +105°C, CLOAD=4.7upF/0.1ohm-ESR 除非另有说明

参数	测试环境	极限			单位
		最小	典型	最大	
电源范围					
电源电压		2.7	5.0	5.5	V
工作电流	普通功耗状态		850		uA
	低功耗状态				uA
DC 精度					
输出电压	-40°C ~ +85°C		2.4		V
散布在温度范围内	VDD = 3.3V±10mV	-15		+15	mV

37.13. 模拟比较器 AC0/AC1 特性

VDD=5.0V, T_A= -40° C ~ +85° C 除非另有说明

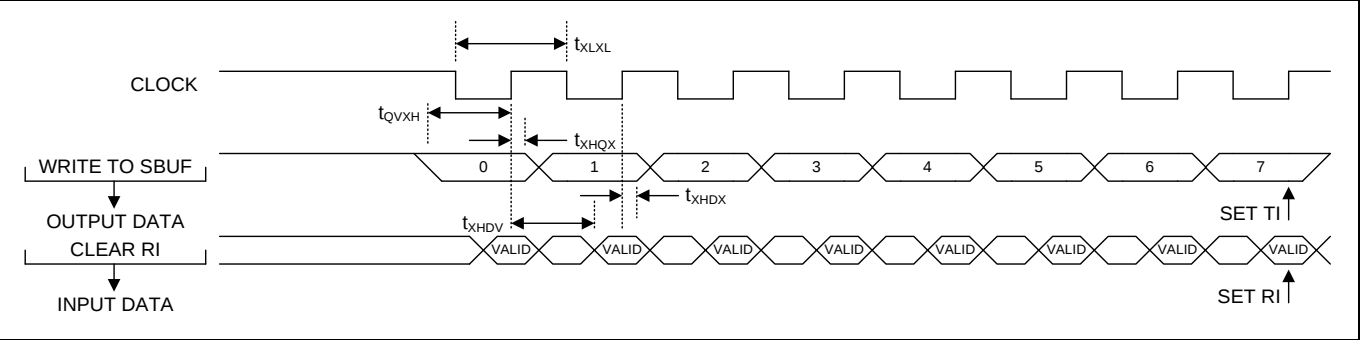
参数	测试环境	极限			单位
		最小	典型	最大	
电源范围					
电源电压		2.0	5.0	5.5	V
工作电流	普通功耗状态		57		uA
	低功耗状态		12		uA
R-Ladder (32 阶)	VDD= 5.0V		95		uA
R-Ladder (24 阶)	VDD= 5.0V		117		uA
DC 精度					
输入电压范围		0		VDD-1	V
输入偏移校准电压	VDD=5.0V		0.8		mV
输入偏移非校准电压	VDD=5.0V	-20		20	mV
输入共模电压		0.02		VDD-1	V
比较器滞环电压 (输入滞环选择=0V)	VDD= 5.0V		0		mV
	VDD= 3.0V		0		
	VDD= 2.0V		0		
比较器滞环电压 (输入滞环选择=±10mV)	VDD= 5.0V	10	18	24	mV
	VDD= 3.0V	16	21	30	
	VDD= 2.0V	18	23	32	
比较器滞环电压 (输入滞环选择=±20mV)	VDD= 5.0V	31	42	59	mV
	VDD= 3.0V	35	46	65	
	VDD= 2.0V	37	50	70	
比较器滞环电压 (输入滞环选择=±60mV)	VDD= 5.0V	80	135	190	mV
	VDD= 3.0V	75	140	195	
	VDD= 2.0V	70	130	200	
响应时间	上升(V _{DD} = 5V, Ta = 25° C, V _{OD} =50mV, V _{CM} < 50mV)		43		ns
	下降(V _{DD} = 5V, Ta = 25° C, V _{OD} =50mV, V _{CM} < 50mV)		54		ns
	上升(V _{DD} ≥ 3V, V _{OD} =100mV, 0.5V ≤ V _{CM} ≤ VDD-1.5V)		65	75	ns
	下降(V _{DD} ≥ 3V, V _{OD} =100mV, 0.5V ≤ V _{CM} ≤ VDD-1.5V)		70	100	ns
上电时间 (从掉电)	正常模式(V _{DD} = 5V, SYSCLK = CPUCLK = 12MHz)				us
	低功耗模式(V _{DD} = 5V, SYSCLK = CPUCLK = 12MHz)				us

37.14. 串口时序特性

VDD = 5.0V±10%, VSS = 0V, TA = -40°C to +105°C, 除非另有说明

参数	测试环境			单位
		最小.	最大	
t _{XLXL}	串口时钟周期	12T		T _{SYSCLK}
t _{QVXH}	设置输出数据到时钟上升沿	10T-20		ns
t _{XHQX}	上升沿后保持输出数据	T-10		ns
t _{XHDX}	上升沿后保持输入数据	5		ns
t _{XHDV}	时钟上升沿到输入数据有效		2T-10	ns

图 37-2. 移位寄存器模式时序波形



37.15. SPI 时序特性

VDD = 5.0V±10%, VSS = 0V, TA = -40°C to +105°C, 除非另有说明

标号	参数	最小.	最大	单位
主机模式时序				
1/(tMCKH + tMCKL)	SPI 时钟频率 @VDD = 3.0V ~ 5.5V		24MHz	MHz
	SPI 时钟频率 @VDD = 1.8V ~ 3.0V		16MHz	MHz
tMCKH	SPICLK 高时间	1T		TSYSCLK
tMCKL	SPICLK 低时间	1T		TSYSCLK
tMIS	MISO 有效到 SPICLK 采样边沿	10		ns
tMIH	SPICLK 转变边沿到 MISO 变化	0		ns
tMOH	SPICLK 转变边沿到 MOSI 变化		10	ns
从机模式时序				
1/(tCKH + tCKL)	SPI 时钟频率 @VDD = 3.0V ~ 5.5V		16MHz	MHz
	SPI 时钟频率 @VDD = 1.8V ~ 3.0V		12MHz	MHz
tSE	nSS 下降沿到第一个 SPICLK 边沿	2T		TSYSCLK
tSD	最后一个 SPICLK 边沿到 nSS 上升沿	2T		TSYSCLK
tSEZ	nSS 下降沿到 MISO 有效		4T	TSYSCLK
tSDZ	nSS 上升沿到 MISO 高阻		4T	TSYSCLK
tCKH	SPICLK 高时间	2T		TSYSCLK
tCKL	SPICLK 低时间	2T		TSYSCLK
tSIS	MOSI 有效到 SPICLK 采样边沿	1T		TSYSCLK
tSIH	SPICLK 采样边沿到 MOSI 变化	1T		TSYSCLK
tSOH	SPICLK 移位边沿到 MISO 变化		2T	TSYSCLK
tSLH	最后的 SPICLK 边沿到 MISO 变化 (仅 CPHA = 1)	1T	2T	TSYSCLK

图 37-3. CPHA=0 时 SPI 主机传送波形

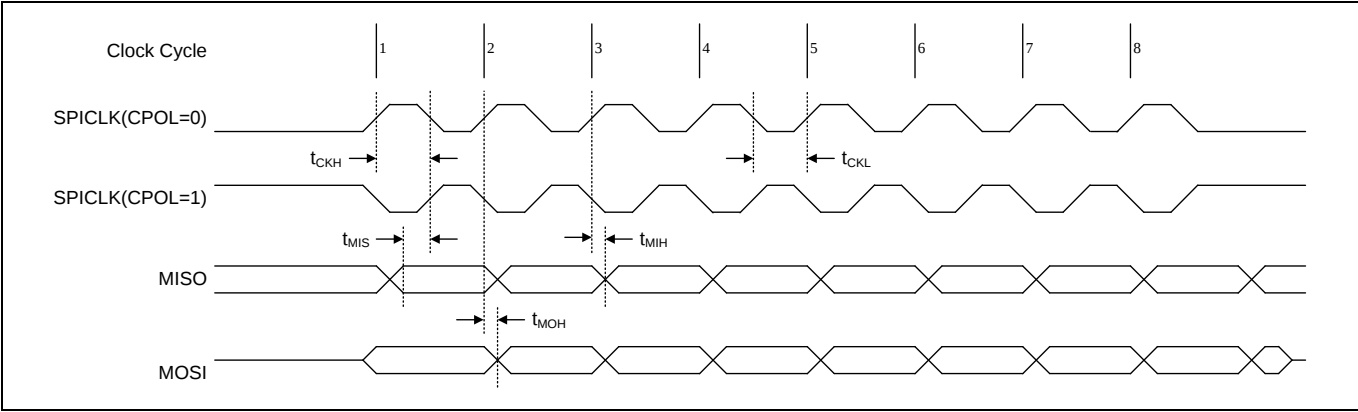


图 37-4. CPHA=1 时 SPI 主机传送波形

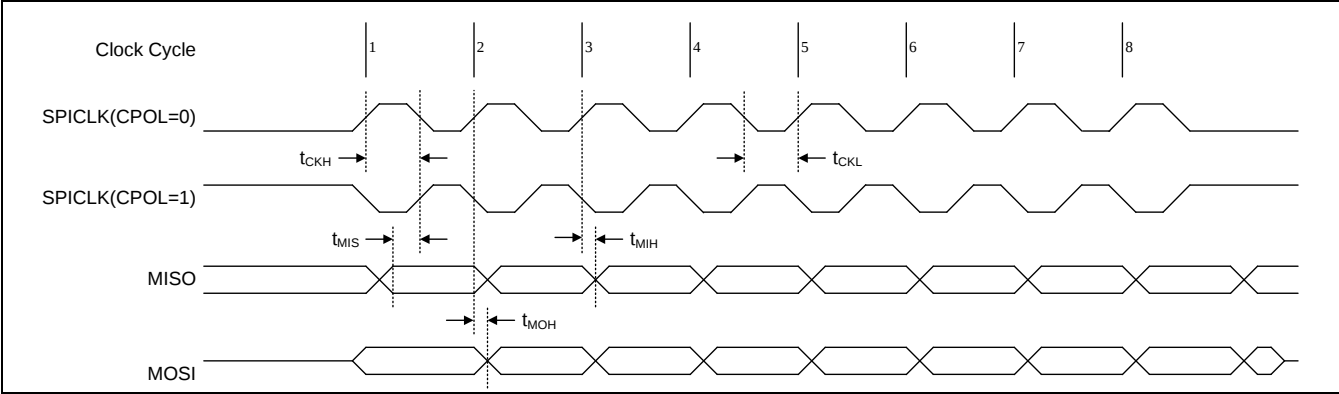


图 37-5. CPHA=0 时 SPI 从机传送波形

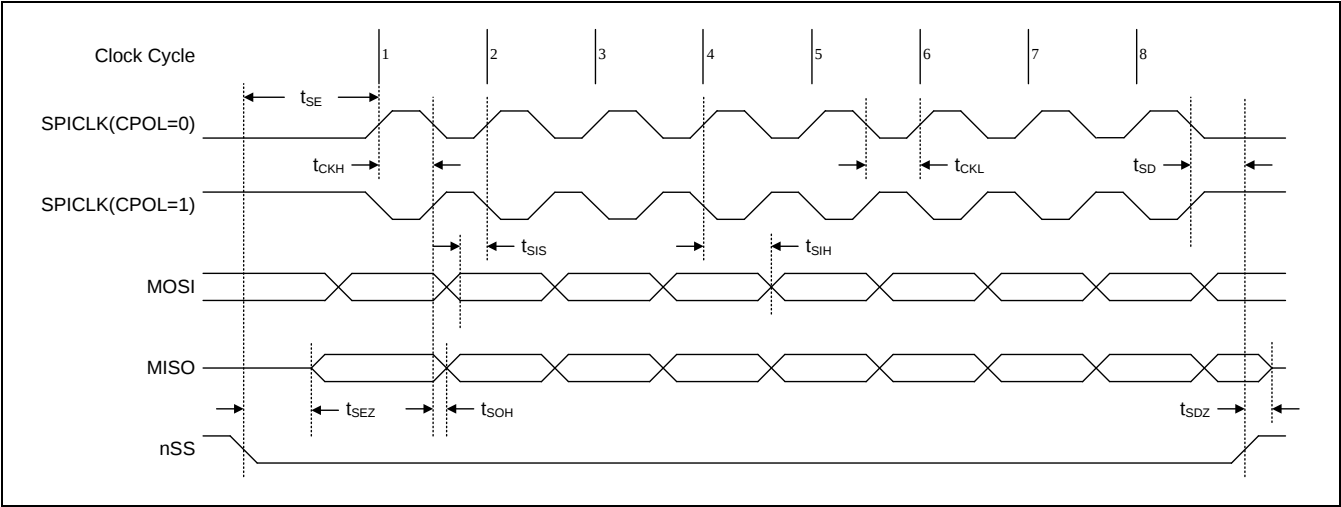
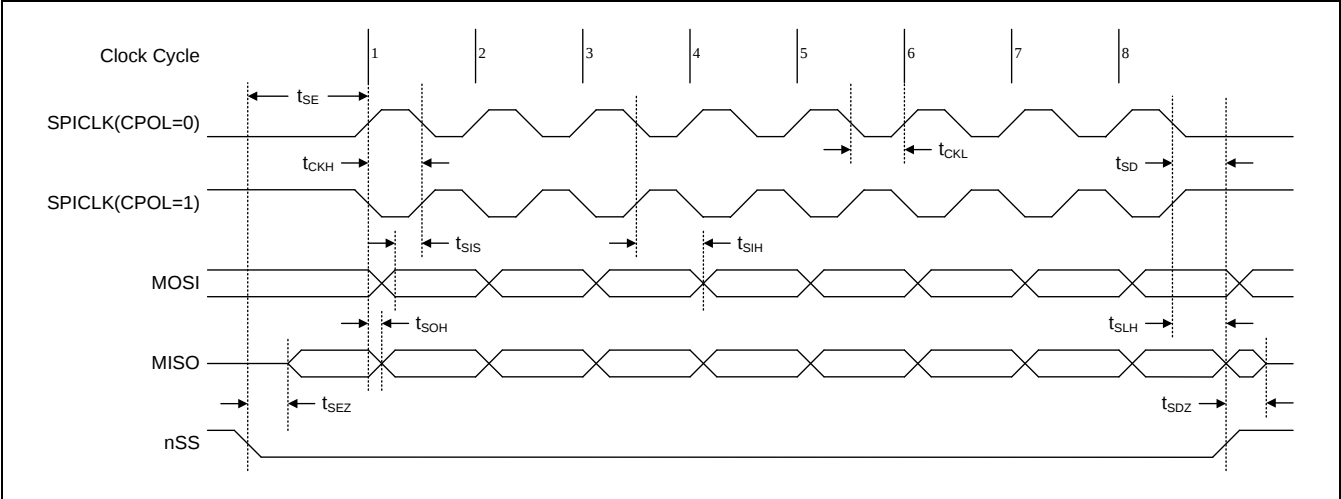


图 37-6. CPHA=1 时 SPI 从机传送波形

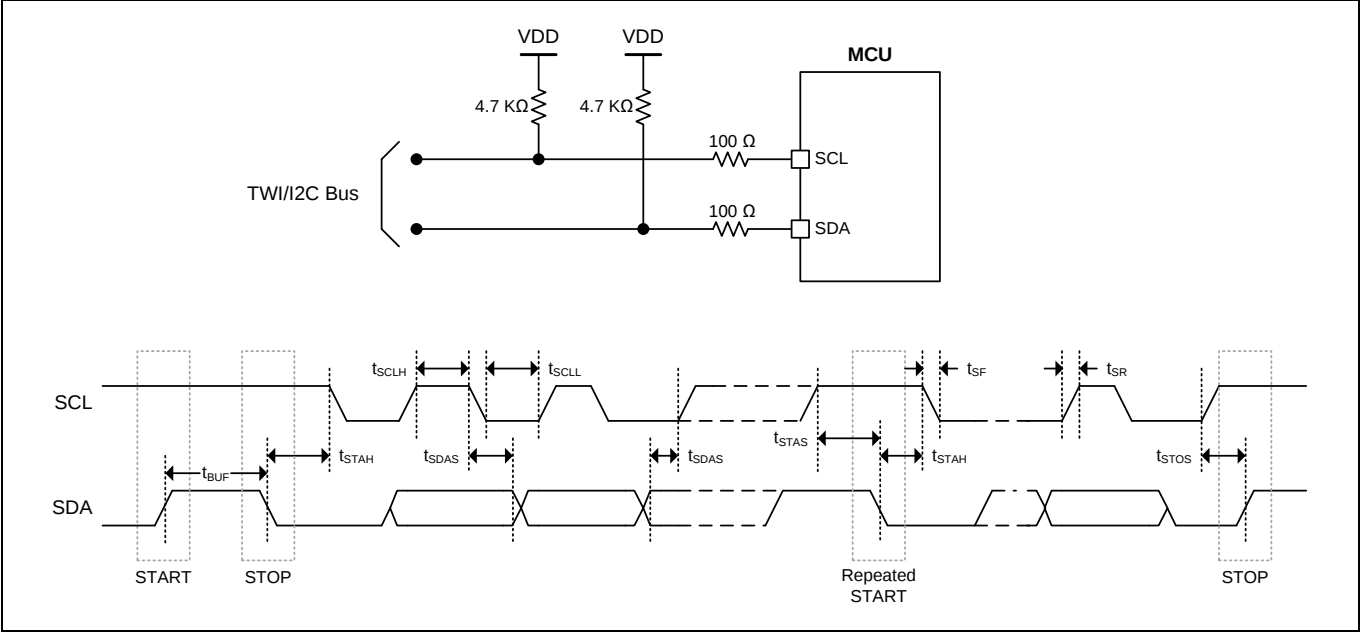


37.16. TWI/I2C 时序特性

VDD = 5.0V±10%, VSS = 0V, TA = -40°C to +105°C, 除非另有说明

标号	参数	最小.	最大	单位
主机模式时序				
tBUF	从停止到启动的总线空闲时间条件			US
tSTAH	(重复)启动条件后的保持时间			US
tSTAS	重复启动条件设置时间			US
tSTOS	停止条件设置时间			US
tSCLH	SCL 高电平时间			T _{SYSCLK}
tSCLL	SCL 低电平时间			T _{SYSCLK}
tSDAS	SDA 数据设置时间			ns
tSDAH	SDA 数据保持时间			ns
tSR	SCL/SDA 上升时间		1000 ~ 300	ns
tSF	SCL/SDA 下降数据			US
CL	SCL 和 SDA 的负载电容	400	400	pF

图 37-7. TWI/I2C 总线的典型应用和时序图



38. 指令集

表 38-1. 指令集

助记符	描述	字节	执行周期
数据传送			
MOV A,Rn	寄存器 Rn 中的内容送到累加器中	1	1
MOV A,direct	直接地址单元中的内容送到累加器中	2	2
MOV A,@Ri	工作寄存器 Ri 指向的地址单元中的内容送到累加器中	1	2
MOV A,#data	立即数送到累加器中	2	2
MOV Rn,A	累加器中内容送到寄存器 Rn 中		2
MOV Rn,direct	直接寻址单元中的内容送到寄存器 Rn 中	2	4
MOV Rn,#data	立即数直接送到寄存器 Rn 中	2	2
MOV direct,A	累加器送到直接地址单元	2	3
MOV direct,Rn	寄存器 Rn 中的内容送到直接地址单元	2	3
MOV direct,direct	直接地址单元中的内容送到另一个直接地址单元	3	4
MOV direct,@Ri	工作寄存器 Ri 指向的地址单元中的内容送到直接地址单元	2	4
MOV direct,#data	立即数送到直接地址单元	3	3
MOV @Ri,A	累加器送到以工作寄存器 Ri 指向的地址单元中	1	3
MOV @Ri,direct	直接地址单元中内容送到以工作寄存器 Ri 指向的地址单元中	2	3
MOV @Ri,#data	立即数送到以工作寄存器 Ri 指向的地址单元中	2	3
MOV DPTR,#data16	16 位常数的高 8 位送到 DPH，低 8 位送到 DPL	3	3
MOVC A,@A+DPTR	以 DPTR 为基地址变址寻址单元中的内容送到累加器中	1	4
MOVC A,@A+PC	以 PC 为基地址变址寻址单元中的内容送到累加器中	1	4
MOVX A,@Ri	片内 XRAM(8 位地址)的数据送入 ACC 中	1	3
MOVX A,@DPTR	片内 XRAM(16 位地址)的数据送入 ACC 中	1	3
MOVX @Ri,A	ACC 的数据送入片内 XRAM(8 位地址)中	1	3
MOVX @DPTR,A	ACC 的数据送入片内 XRAM(16 位地址)中	1	3
MOVX A,@Ri	外部 XRAM(8 位地址)的数据送入 ACC 中	1	3 ~ 12*Note1
MOVX A,@DPTR	外部 XRAM(16 位地址)的数据送入 ACC 中	1	3 ~ 12*Note1
MOVX @Ri,A	ACC 的数据送入外部 XRAM(8 位地址)中	1	3 ~ 12*Note1
MOVX @DPTR,A	ACC 的数据送入外部 XRAM(16 位地址)中	1	3 ~ 12*Note1
PUSH direct	直接地址单元中的数据压入堆栈中	2	4
POP direct	出栈数据送到直接地址单元中	2	3
XCH A,Rn	累加器与寄存器 Rn 中的内容互换	1	3
XCH A,direct	累加器与直接地址单元中的内容互换	2	4
XCH A,@Ri	累加器与工作寄存器 Ri 指向的地址单元中内容互换	1	4
XCHD A,@Ri	累加器与工作寄存器 Ri 指向的地址单元中内容低半字节互换	1	4
算术运算			
ADD A,Rn	将寄存器 Rn 中的内容加到累加器中	1	2
ADD A,direct	直接地址单元中的内容加到累加器中	2	3
ADD A,@Ri	寄存器工作寄存器 Ri 指向的地址单元中的内容加到累加器中	1	3
ADD A,#data	立即数加到累加器中	2	2
ADDC A,Rn	累加器与工作寄存器 Rn 中的内容、连同进位位相加，结果存在累加器中	1	2
ADDC A,direct	累加器与直接地址单元的内容、连同进位位相加，结果存在累加器中	2	3
ADDC A,@Ri	累加器与工作寄存器 Ri 指向的地址单元中的内容、连同进位位相加，结果存在累加器中	1	3
ADDC A,#data	累加器与立即数、连同进位位相加，结果存在累加器中	2	2
SUBB A,Rn	累加器与工作寄存器中的内容、连同借位位相减，结果存在累加器中	1	2
SUBB A,direct	累加器与直接地址单元中的内容、连同借位位相减，结果存在累加器中	2	3
SUBB A,@Ri	累加器与工作寄存器 Ri 指向的地址单元中内容、连同借位位相减，结果存在累加器中	1	3

助记符	描述	字节	执行周期
SUBB A,#data	累加器与立即数、连同借位相减，结果存在累加器中	2	2
INC A	累加器中的内容加 1	1	2
INC Rn	寄存器 Rn 的内容加 1	1	3
INC direct	直接地址单元中的内容加 1	2	4
INC @Ri	工作寄存器 Ri 指向的地址单元中的内容加 1	1	4
DEC A	数据指针 DPTR 的内容加 1	1	2
DEC Rn	累加器中的内容减 1	1	3
DEC direct	寄存器 Rn 中的内容减 1	2	4
DEC @Ri	直接地址单元中的内容减 1	1	4
INC DPTR	工作寄存器 Ri 指向的地址单元中的内容减 1	1	1
MUL AB	ACC 中内容与寄存器 B 中内容相乘，其结果低位存在 ACC 中、高位存在寄存器 B 中	1	4
DIV AB	ACC 中内容除以寄存器 B 中内容，商存在 ACC，而余数存在寄存器 B 中	1	5
DA A	ACC 十进制调整	1	4
逻辑运算			
ANL A,Rn	累加器和寄存器 Rn 中的内容相“与”	1	2
ANL A,direct	累加器和直接地址单元中的内容相“与”	2	3
ANL A,@Ri	累加器和工作寄存器 Ri 指向的地址单元中的内容相“与”	1	3
ANL A,#data	累加器和立即数相“与”	2	2
ANL direct,A	直接地址单元中的内容和累加器相“与”	2	4
ANL direct,#data	直接地址单元中的内容和立即数相“与”	3	4
ORL A,Rn	累加器和寄存器 Rn 中的内容相“或”	1	2
ORL A,direct	累加器和直接地址单元中的内容相“或”	2	3
ORL A,@Ri	累加器和工作寄存器 Ri 指向的地址单元中的内容相“或”	1	3
ORL A,#data	累加器和立即数相“或”	2	2
ORL direct,A	直接地址单元中的内容和累加器相“或”	2	4
ORL direct,#data	直接地址单元中的内容和立即数相“或”	3	4
XRL A,Rn	累加器和寄存器 Rn 中的内容相“异或”	1	2
XRL A,direct	累加器和直接地址单元中的内容相“异或”	2	3
XRL A,@Ri	累加器和工作寄存器 Ri 指向的地址单元中的内容相“异或”	1	3
XRL A,#data	累加器和立即数相“异或”	2	2
XRL direct,A	直接地址单元中的内容和累加器相“异或”	2	4
XRL direct,#data	直接地址单元中的内容和立即数相“异或”	3	4
CLR A	累加器内容清“0”	1	1
CPL A	累加器按位取反	1	2
RL A	累加器循环左移一位	1	1
RLC A	累加器连同进位位 CY 循环左移一位	1	1
RR A	累加器循环右移一位	1	1
RRC A	累加器连同进位位 CY 循环右移一位	1	1
SWAP A	累加器高低半字节互换	1	1
位逻辑运算			
CLR C	清“0”进位位	1	1
CLR bit	清“0”直接地址位	2	4
SETB C	置“1”进位位	1	1
SETB bit	置“1”直接地址位	2	4
CPL C	进位位求反	1	1
CPL bit	直接地址位求反	2	4
ANL C,bit	进位位和直接地址位相“与”	2	3

助记符	描述	字节	执行周期
ANL C,/bit	进位位和直接地址位相“与”	2	3
ORL C,/bit	进位位和直接地址位的反码相“与”	2	3
ORL C,/bit	进位位和直接地址位相“或”	2	3
MOV C,/bit	进位位和直接地址位的反码相“或”	2	3
MOV bit,C	直接地址位数据送入进位位	2	4
位逻辑跳转			
JC rel	进位位为“1”则转移	2	3
JNC rel	进位位为“0”则转移	2	3
JB bit,rel	直接地址位为“1”则转移	3	4
JNB bit,rel	直接地址位为“0”则转移	3	4
JBC bit,rel	直接地址位为“1”则转移，且清“0”该位	3	5
程序跳转			
ACALL addr11	绝对短调用子程序，2K 字节(页内)空间限制	2	6
LCALL addr16	绝对长调用子程序，64K 字节空间限制	3	6
RET	子程序返回	1	4
RETI	中断子程序返回	1	4
AJMP addr11	绝对短转移，2K 字节(页内)空间限制	2	3
LJMP addr16	绝对长转移，64K 字节空间限制	3	4
SJMP rel	相对转移	2	3
JMP @A+DPTR	转移到 DPTR 加 ACC 所指间接地址	1	3
JZ rel	累加器为“0”则转移	2	3
JNZ rel	累加器不为“0”则转移	2	3
CJNE A,direct,rel	累加器不等于直接地址单元的内容，则转移到偏移量所指向的地址，否则程序往下执行	3	5
CJNE A,#data,rel	累加器不等于立即数，则转移到偏移量所指向的地址，否则程序往下执行	3	4
CJNE Rn,#data,rel	寄存器 Rn 中的内容不等于立即数，则转移到偏移量所指向的地址，否则程序往下执行	3	4
CJNE @Ri,#data,rel	Ri 指向的地址单元不等于立即数，则转移到偏移量所指向的地址，否则程序往下执行	3	5
DJNZ Rn,rel	寄存器 Rn 减 1，如不等于 0，则转移到偏移量所指向的地址，否则程序往下执行	2	4
DJNZ direct,rel	直接地址单元减 1，如不等于 0，则转移到偏移量所指向的地址，否则程序往下执行	3	5
NOP	空操作指令	1	1

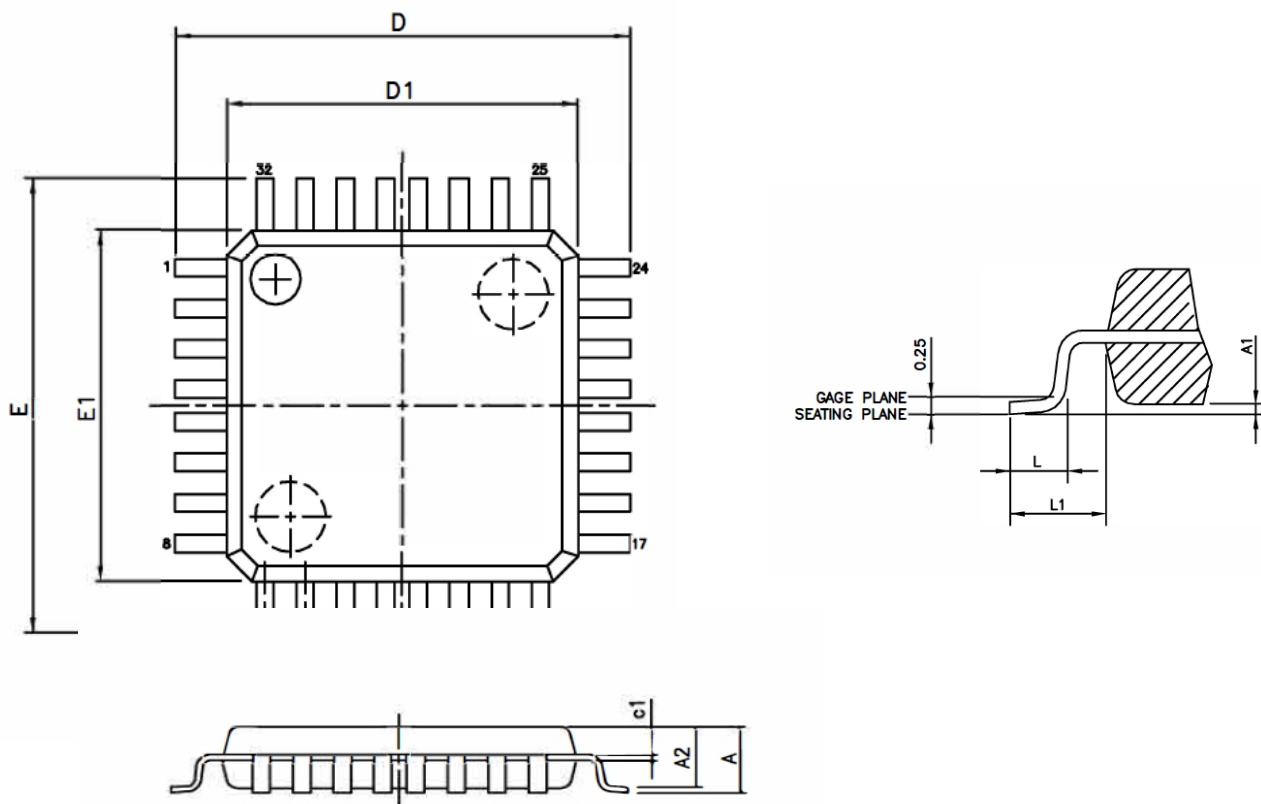
Note 1: 访问外部扩展 RAM(EMB)的周期时间是:

$$EMAI1 = 1: 3 + RW_Stretch + 2 \times RWSH; (3\sim12)$$

39. 封装尺寸

39.1. LQFP-32 (7mm X 7mm)封装尺寸

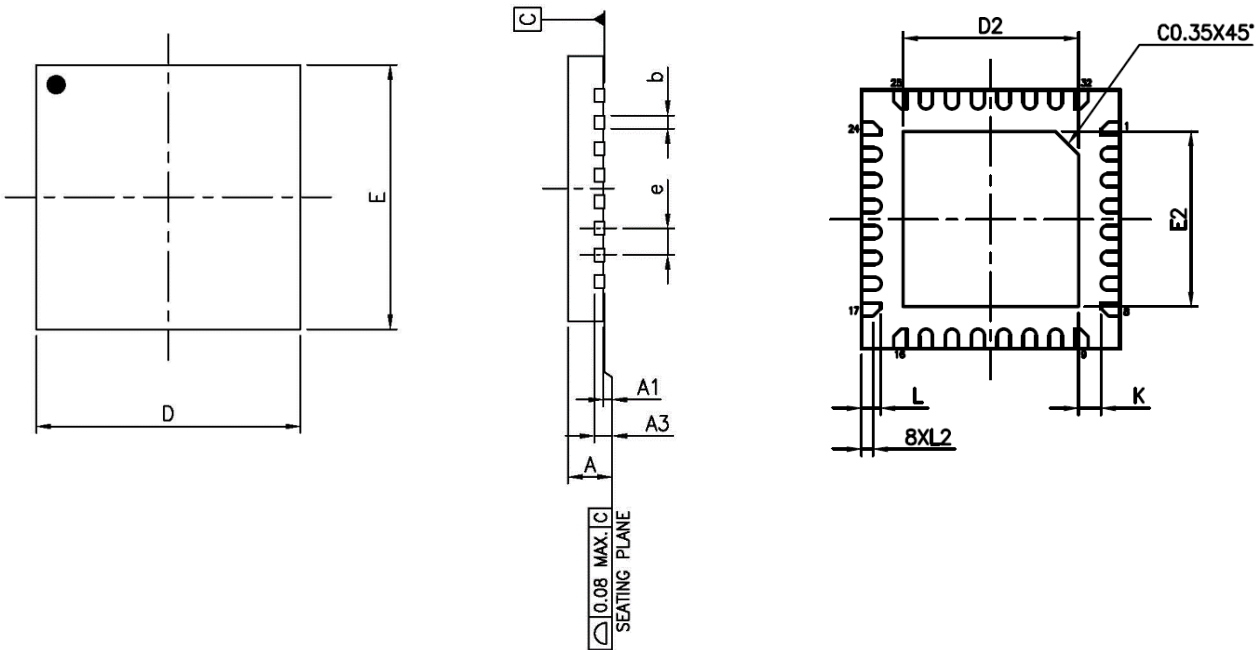
图 39-1. LQFP-32 (7mm X 7mm)封装尺寸



Unit	mm		inch	
Symbols	Min.	Max.	Min.	Max.
mm				
A	---	1.6	---	0.062
A1	0.05	0.15	0.000	0.005
A2	1.35	1.45	0.053	0.057
c1	0.09	0.16	0.003	0.006
D	9.00 BSC		0.354 BSC	
D1	7.00 BSC		0.275 BSC	
E	9.00 BSC		0.354 BSC	
E1	7.00 BSC		0.275 BSC	
e	0.8 BSC		0.0314 BSC	
b	0.30	0.45	0.011	0.017
L	0.45	0.75	0.017	0.029
L1	1REF		0.039REF	

39.2. QFN-32 (4mm X 4mm X 0.55mm) 封装尺寸

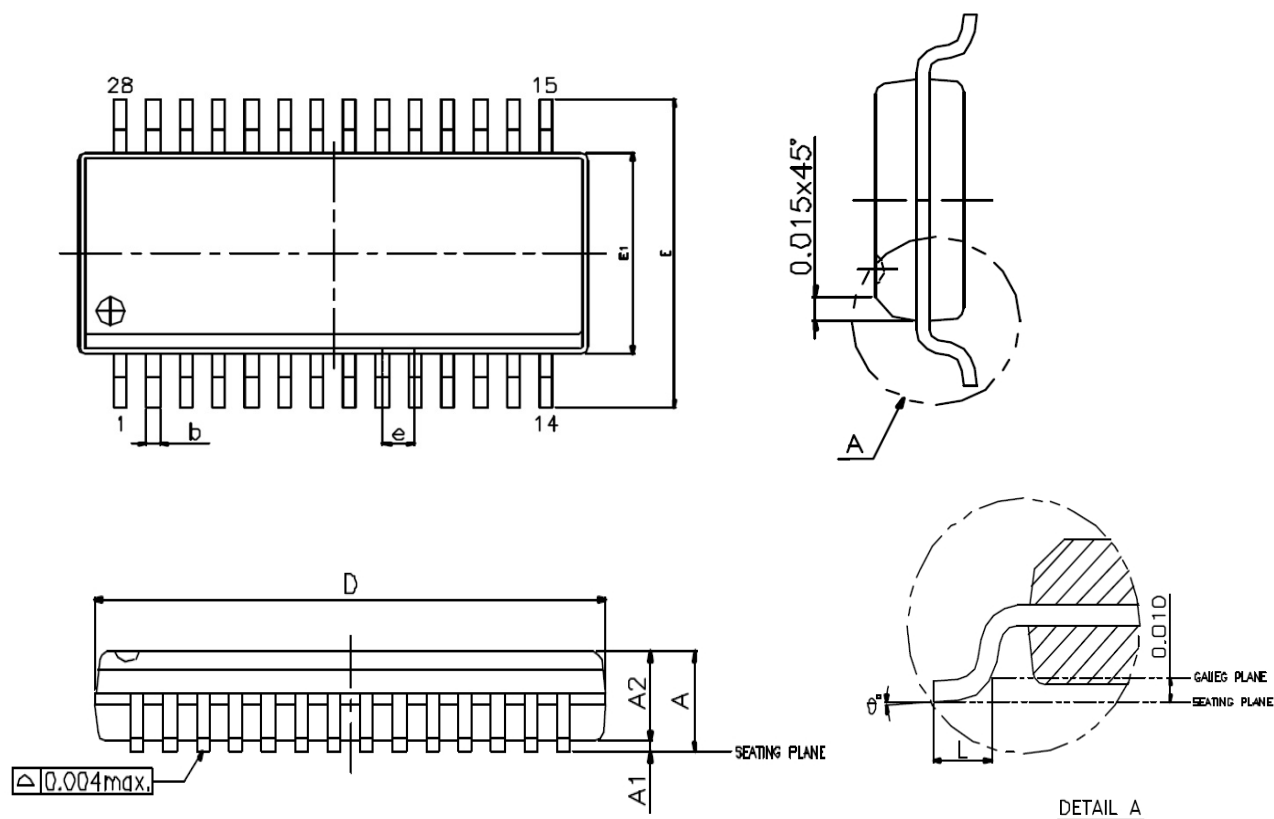
图 39-2. QFN-32 (4mm X 4mm X 0.55mm)封装尺寸



Unit	mm			inch		
JEDEC	N/A			N/A		
PKG CODE	UQFN(W432)			UQFN(W432)		
Symbols	Min.	Nom.	Max.	Min.	Nom.	Max.
A	0.50	0.55	0.60	0.020	0.022	0.024
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3	0.150 REF.			0.006 REF.		
b	0.15	0.20	0.25	0.006	0.008	0.010
D	4.00 BSC			0.157 BSC		
E	4.00 BSC			0.157 BSC		
e	0.40 BSC			0.016 BSC		
L	0.25	0.30	0.35	0.010	0.012	0.014
L2	0.13	0.18	0.23	0.005	0.007	0.009
K	0.20	----	----	0.008	----	----
D2	2.65	2.70	2.75	0.104	0.106	0.108
E2	2.65	2.70	2.75	0.104	0.106	0.108

39.3. SSOP28 (150 mil)封装尺寸

图 39-3. SSOP-28 (150 mil)封装尺寸



Symbols	Dimensions in mm	
	Min.	Max.
A	1.346	1.752
A1	0.101	0.254
A2	----	1.498
b	0.203	0.304
D	9.804	10.007
E1	3.81	3.987
e	0.635 BASIC	
E	5.791	6.198
L	0.406	1.270
θ°	0	8

40. 版本历史

表 40-1. 版本历史

版本	描述	日期
V0.20	初始发布版本	2024/02/23
V0.21	1. 修改 图 9-1 2. 修正 CPU 和 MCKDO 最大频率 3. 修正仅模拟输入为模拟 I/O 4. 在章节 14.1 和 14.1.5 增加 OCD 和 nRST 的描述 5. 修正 CRC16 结构，并增加 CRC32 结构	2024/02/29
V0.22	1. 更新 PGA 方框图 2. 更新系统方框图 3. 在章节 271.OPA 方框图里增加 I/O 引脚描述 4. 在章节 27.2.3.里修改 OPA 电源模式描述 5. 增加 OPnIG 的描述. 6. 修正 PGGAIN 为 PGGN. 7. 增加 ADMINIS 的描述 8. 修正 OP1CON0 里的 OP1PIS 和 OP1NIS 的位位置. 9. 删除 S0CFG 位 5 的描述，并且移除 S0CFG1. 10. 移除 TR4E, T4RLC, T4SC 11. 移除 PCAPWMN 的 CCFn, 它已经移到 CCON。 12. 在章节 18.4.14 里修正 C1PWMCR 位 6 的打字错误。 13. 修改 GC 和 GC1 的描述 14. 增加 SM20, SM21 的描述 15. 修正图 32-1 Flash 存储器配置里的打字错误 16. 修正 SnAGBRF 描述里的打字错误的, SnIDT11~01 17. 修正 ADCFG4 位 5~4 表里的打字错误, 1011 从 P0ES 到 OP0ES 18. 修正 AC1NIS, NVRL1 表里的打字错误 19. 在章节 17.1 里移除 PTM0 的描述 20. 修正表 30-1 里 ACONI 引脚为 P3.4, 以及 IVR2.4V 在表 30-2 21. 修正章节 16.1.6 里的 T1 引脚指派 22. 修正章节 30.2 里的 AC1MOD 读/写能力 23. 修正图 4-1 和 图 4-3 里 CEX4 复用引脚从 P44 改为 P32 24. 更新 OPA 和 PGA 电气特性 25. 增加 OP1Fr, OP0Fr, AC1Fr 和 AC0Fr 的描述 26. 修正外部复位电容与电阻的描述	2024/03/18
V1.00	1. 修正 OP1PIS0 的拼字错误 2. 增加 IVR12 的描述，IVR12 需要启用 IVR24。 3. 修改 OPA 失调微调稳定时间为 600uS。 4. 修正了第 30 章描述中 IVR 电压的错字，从 1.4V 改为 2.4V 5. 更改 AC0M[1:0]的描述以提高可读性。 6. 修改 ADC 转换率。	2024/09/09

41. 免责声明

在此，笙泉(Megawin)代表 “*Megawin Technology Co., Ltd.*”

生命支持—此产品并不是为医疗、救生或维持生命而设计的，并且当设备系统出现故障时，并不能合理地预示是否会对人身造成伤害。因此，当客户使用或出售用于上述应用的产品时，需要客户自己承担这样做的风险，笙泉公司并不会对不当地使用或出售我公司的产品而造成的任何损害进行赔偿。

更改权—笙泉保留产品的如下更改权，其中包括电路、标准单元、与/或软件 - 在此为提高设计的与/或性能的描述或内容。当产品在大批量生产时，有关变动将通过工程变更通知(ECN)进行通知。