

# MGEQ1C064

## 数据手册

版本: 0.12



## 特性

- AEC-Q100 2 级认证
- 1-T 80C51 中央处理器
- **MGEQ1C064** 内含 **64K** 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
  - **MGEQ1C064 默认空间配置**
    - \* AP Flash 默认映射 (61.5KB, 0000h~F5FFh)
    - \* IAP Flash 默认映射(1.0KB, F600h~F9FFh)
    - \* ISP Flash 默认映射(1.5KB, FA00h~FFFFh), ISP 引导码
- 数据存储 RAM:
  - 片内 256 字节高速缓存
  - **MGEQ1C064** -片内 **3840** 字节扩展 RAM(XRAM)
  - 扩展 RAM(XRAM)支持页选
- 双数据指针
- 外部存储总线接口(EMB)
  - 仅支持无地址模式访问(LCD 8080 I/F)
- 提供一个通道的 DMA 引擎
  - P2P, M2P, P2M
  - 存储器目标: 片内 XRAM & EMB
  - 外设目标: UART0/1/2/3, SPI, TWI0/I2C0, I2C1, ADC12, EMB & CRC16
  - 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
  - **24** 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
- **MGEQ1C064** 共有 **14/ 17** (分割模式下)个定时器
  - RTC 定时器和 WDT 定时器
  - 定时器 0、定时器 1、定时器 2、定时器 3、定时器 4
  - PCA0, 可编程计数器阵列 0
  - S0 BRG, S1 BRG, S2 BRG 和 S3 BRG
  - 如果定时器 2/3/4 用于分立模式, 则 **MGEQ1C064** 共有 **17** 个定时器
  - DMA 模块的定时器 5/6 也可以用作定时器
- **7** 个 16-位定时/计数器, 定时器 0、定时器 1、定时器 2、定时器 3 和定时器 4, 定时器 5、定时器 6
  - X12 模式和时钟输出功能
  - 定时器 0~4 具有同步控制使能、停止和重载的功能。
  - 定时器 2/3/4 新的 5 个操作模式, 它们有着 8 个时钟源和 8 个捕捉源
  - 定时器 2/3/4 可以分立为两个 8 位定时器
  - 时钟输出(CCO)在 T2CKO、T3CKO 和 T4CKO
  - 定时器 0~4 支持 PWM 模式
  - 定时器 2/3/4 支持占空比捕捉功能
  -
- **1** 个可编程 16 位计数/定时阵列(PCA0)有着 **8** 个 CCP 模块
  - PCA0 有 6 个 CCP(捕捉/比较/PWM)模块和 2 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 **144MHz** 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, 可被配置成:
    - \* 最高 **8** 通道无缓冲 10/12/16 位 PWM, 或
    - \* 最高 **8** 通道带缓冲 2~8 位 PWM, 或

- \* U 最高 4 通道带缓冲 9~16 位 PWM
- PCA0 PWM 模式 0~5 具有死区控制, 中止控制和中心对齐选择
- 8 个键盘中断(KBI)
- 输出信号中止&调制 0/1 (OBM0/1)
  - 多路调制和可选载波源
  - 支持快速事件中止功能
- 12 位单端 ADC
  - 可编程转换率高达 1.0M sps
  - MGEQ1C064 有 10 个外部输入通道和 2 个用于内部参考电压(IVR/1.4V)与内部 VSS 的通道
  - 支持 ADC 结果的窗口监测功能
  - 支持通道扫描模式
  - ADC VREF+从外部输入
- 模拟比较器 0
  - 在 ACN10 上可选内部电压参考(IVR/1.4V)
  - 4 个可选 ACPI0(+) 输入
  - 可唤醒掉电模式和 Idle 模式
  - 脉冲滤波器和输出到内部定时器捕捉源
- 模拟比较器 1/2
  - ACN11/ACN12 上可选内部电压参考(IVR/1.4V)
  - 可唤醒掉电模式和 Idle 模式
  - 脉冲滤波器和输出到内部定时器捕捉源
- 增强型 UART (S0)
  - 帧错误监测
  - 自动地址识别
  - 最大波特率可达 3.6864MHz/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
  - 模式 5 支持具有自动波特率的 LIN 总线协议
  - 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
- 2<sup>nd</sup>/3<sup>rd</sup>/4<sup>th</sup> UART (S1/ S2/ S3)
  - 最大波特率可达 1.8432/3.0MHz
  - 模式 4 支持 SPI 主机, SPICLK 速率可达 12MH
- 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 SPICLK 速率可达 12MHz
  - 支持 QPI 模式
  - 高达 5 个 SPI 主机(包括串口 S0/S1/S2/S3 模式 4)
  - 在 SPI 从机模式下支持菊花链功能
- 3 个主/从两线串行接口: TWI0/I2C0, TWI1/I2C1 和 STWI (SI2C)
  - 2 主/从硬件引擎: TWI0/I2C0 和 TWI1/I2C1
  - I2C0/I2C1 主机模式可达 1MHz 和 I2C0/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
  - 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 (CCITT-16 多项式)
  - 支持 FLASH 内容的自动 CRC
  - 可编程 CRC 的初始种子功能

- 片上调试接口(OCD)
- 最大 **44** GPIO
  - 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 (P5,P7 在 48 引脚封装下没有封出来) 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 共用 XTAL2, XTAL1 和 RST
  - 可编程通用输入输出(GPIO)的驱动力
  - 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 (144MHz / 96MHz)
- 两个低电压监测
  - BOD0: 监测 **1.7V**
  - BOD1: 选择监测电压为 4.2V/3.7V/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: 2.4V – 5.5V
  - Flash 写操作(ISP/IAP/ICP)的最低电压为 **2.4V**
- 工作频率范围: **36MHz** (最大)
  - 外部时钟输入模式, 0 – 25MHz @ 2.4V – 5.5V
  - CPU 可达 25MHz @ **2.4V** – 5.5V
  - **使用片内时钟倍频器(CKM), CPU 可达 36MHz @ 2.7V -5.5V**
- 16 字节唯一 ID 码
- 工作温度
  - 工业级(-40°C 到+105°C)\*
- 封装类型
  - LQFP48 (7 x 7 mm): MGEQ1C064AD48 (64K)

## 目录

特性.....	3
目录.....	6
图目录.....	13
表目录.....	19
1. 概述.....	22
2. 方框图.....	23
3. 特殊功能寄存器.....	24
3.1. SFR 图(页 0~F).....	24
3.2. SFR 位分配(页 0~F).....	26
3.3. 辅助 SFR 图 (P 页).....	30
3.4. 辅助 SFR 位分配(P 页).....	31
4. 引脚结构.....	32
4.1. 封装指南.....	32
4.2. 引脚描述.....	33
4.3. 功能复用.....	36
5. 8051 CPU 功能描述.....	45
5.1. CPU 寄存器.....	45
5.2. CPU 时序.....	46
5.3. CPU 寻址模式.....	47
6. 存储器组织.....	48
6.1. 片内程序存储器 (Flash).....	48
6.2. 片内数据存储器 RAM.....	49
6.3. 片内扩展 RAM (XRAM).....	50
6.4. EMB,片外数据存储器总线.....	51
6.4.1. 无地址相位模式的 MOVX 8 位数据访问.....	52
6.5. 关于 C51 编译器的声明标识符.....	53
7. XRAM 访问.....	54
7.1. MOVX 在 16 位地址上使用双 DPTR.....	54
7.2. MOVX 在 8 位地址上使用 XRPS.....	56
8. 接存储器访问控制器 (DMA).....	57
8.1. DMA 结构.....	58
8.2. DMA 操作.....	59
8.2.1. DMA 传输类型.....	59
8.2.2. DMA 传输模式.....	60
8.2.3. 传输计数和地址指针.....	60
8.2.4. 启动一个 DMA 传输.....	60
8.2.5. 暂停或停止 DMA 传输.....	61
8.2.6. DMA 中断.....	61
8.2.7. DMA 循环模式.....	61
8.2.8. DMA 的错误处理.....	61
8.2.9. 数据拷贝到 CRC16.....	61
8.2.10. 定时器 5 & 定时器 6.....	62
8.3. DMA 寄存器.....	64
8.4. 定时器 5 寄存器.....	65
8.5. 定时器 6 寄存器.....	66
9. 系统时钟.....	68
9.1. 时钟结构.....	69
9.2. 时钟源切换.....	69
9.3. 片内 CKM (PLL).....	70
9.4. 时钟丢失监测(MCD).....	70
9.5. XTAL 模式下快速唤醒.....	70
9.6. CKM 下的唤醒.....	70

9.7.	时钟寄存器 .....	71
10.	看门狗定时器(WDT) .....	75
10.1.	WDT 结构 .....	75
10.2.	空闲模式的 WDT .....	75
10.3.	掉电模式的 WDT (自动唤醒) .....	75
10.4.	WDT 寄存器 .....	76
10.5.	WDT 硬件选项 .....	78
11.	实时时钟(RTC)/系统时钟 .....	79
11.1.	RTC 寄存器 .....	80
12.	系统复位 .....	82
12.1.	复位源 .....	82
12.2.	上电复位 .....	83
12.3.	外部复位 .....	83
12.4.	软件复位 .....	83
12.5.	低电压监测复位 .....	84
12.6.	WDT 复位 .....	84
12.7.	MCD 复位 .....	85
12.8.	非法地址复位 .....	85
12.9.	堆栈指针警告复位 .....	85
13.	电源管理 .....	86
13.1.	低电压监测器 .....	86
13.2.	省电模式 .....	87
13.2.1.	慢频模式 .....	87
13.2.2.	副频模式 .....	87
13.2.3.	RTC 模式 .....	87
13.2.4.	Watch 模式 .....	87
13.2.5.	Monitor 模式 .....	87
13.2.6.	Idle 模式 .....	87
13.2.7.	掉电模式 .....	87
13.2.8.	中断唤醒掉电模式 .....	89
13.2.9.	复位唤醒掉电模式 .....	89
13.2.10.	KBI 键盘唤醒掉电模式 .....	89
13.3.	电源控制寄存器 .....	90
14.	I/O 口配置(GPIO) .....	92
14.1.	IO 结构 .....	92
14.1.1.	端口 3 准双向口结构 .....	92
14.1.2.	端口 3 推挽输出结构 .....	93
14.1.3.	端口 3 仅输入(高阻抗输入)结构 .....	93
14.1.4.	端口 3 开漏输出结构 .....	94
14.1.5.	用仅模拟输入结构 .....	94
14.1.6.	带上拉电阻的通用开漏输出 .....	95
14.1.7.	通用开漏输出结构 .....	95
14.1.8.	通用端口的数字输入结构 .....	95
14.1.9.	通用推挽输出结构 .....	96
14.1.10.	端口引脚输出驱动力选择 .....	96
14.1.11.	端口引脚输出快速驱动选择 .....	96
14.2.	I/O 端口寄存器 .....	96
14.2.1.	端口 0 寄存器 .....	97
14.2.2.	端口 1 寄存器 .....	97
14.2.3.	端口 2 寄存器 .....	98
14.2.4.	端口 3 寄存器 .....	98
14.2.5.	端口 4 寄存器 .....	98
14.2.6.	端口 5 寄存器 ((48 引脚封装不支持) .....	99
14.2.7.	端口 6 寄存器 .....	99
14.2.8.	端口 7 寄存器 (48 引脚封装不支持) .....	100
14.2.9.	端口输出驱动力控制寄存器 .....	100
14.2.10.	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.....	109
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/计数器.....	122
16.1.	定时器 0 and 定时器 1.....	122
16.1.1.	定时器 0/1 Mode 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 (外部中断触发外部中断).....	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.	波特率发生器模式(BRG).....	143
16.2.12.	定时器 2 可编程时钟输出.....	145
16.2.13.	定时器 2 寄存器.....	147
16.3.	定时器 3.....	151
16.3.1.	定时器 3 模式 0 (自动重载和外部中断).....	151
16.3.2.	定时器 3 模式 1 (外部中断触发自动重载).....	152
16.3.3.	定时器 3 模式 2 (捕捉).....	153
16.3.4.	定时器 3 模式 3 (捕捉带自动清零).....	154
16.3.5.	定时器 3 模式 6 (占空比捕捉).....	155
16.3.6.	分立定时器 3 模式 0 (自动重载和外部中断).....	156
16.3.7.	分立定时器 3 模式 1 (外部中断触发自动重载).....	157
16.3.8.	分立定时器 3 模式 2 (捕捉).....	158
16.3.9.	分立定时器 3 模式 3 (捕捉带自动清零).....	159
16.3.10.	分立定时器 3 模式 4 (8-位 PWM 模式).....	160
16.3.11.	定时器 3 可编程时钟输出.....	161
16.3.12.	定时器 3 寄存器.....	163
16.4.	定时器 4.....	167
16.4.1.	定时器 4 模式 0 (自动重载和外部中断).....	167
16.4.2.	定时器 4 模式 1 (外部中断触发自动重载).....	168
16.4.3.	定时器 4 模式 2 (捕捉).....	169
16.4.4.	定时器 4 模式 3 (捕捉带自动清零).....	170
16.4.5.	定时器 4 模式 6 (占空比捕捉).....	171
16.4.6.	分立定时器 4 模式 0 (自动重载和外部中断).....	172
16.4.7.	分立定时器 4 模式 1 (外部中断触发自动重载).....	173
16.4.8.	分立定时器 4 模式 2 (捕捉).....	174
16.4.9.	分立定时器 4 模式 3 (捕捉带自动清零).....	175
16.4.10.	分立定时器 4 模式 4 (8-位 PWM 模式).....	176
16.4.11.	定时器 4 可编程时钟输出.....	177
16.4.12.	定时器 4 寄存器.....	179
16.5.	定时器全局控制.....	183
16.5.1.	所有定时器运行的全局使能.....	183
16.5.2.	所有定时器重载的全局使能.....	183
16.5.3.	所有定时器停止的全局使能.....	184
17.	可编程计数器阵列(PCAO).....	185

17.1.	PCA 概述.....	185
17.2.	PCA 定时/计数器.....	186
17.3.	比较/捕捉模块.....	190
17.4.	PCA 操作模式.....	192
17.4.1.	捕捉模式.....	193
17.4.2.	缓冲捕捉模式.....	194
17.4.3.	16 位软件定时器模式(比较模式).....	195
17.4.4.	高速输出模式(比较输出模式).....	195
17.4.5.	缓冲 8 位 PWM 模式.....	196
17.4.6.	无缓冲 10/12/16 位 PWM 模式.....	197
17.4.7.	缓冲 10/12/16 位 PWM 模式.....	198
17.4.8.	COPM 模式.....	199
17.4.9.	缓冲 COPM 模式.....	200
17.4.10.	FIFO 数据模式.....	201
17.4.11.	增强型 PWM 控制.....	202
17.4.12.	PCA 模块输出控制.....	206
17.4.13.	可变分辨率的 PWM 中心对齐.....	212
18.	串口 0 (UART0).....	213
18.1.	串口 0 模式选择.....	213
18.2.	串口 0 模式 0.....	215
18.3.	串口 0 模式 1.....	217
18.4.	串口 0 模式 2 和模式 3.....	218
18.5.	帧错误侦测.....	218
18.6.	多处理器通讯.....	219
18.7.	自动地址识别.....	220
18.8.	波特率设置.....	222
18.8.1.	S0 的波特率选择.....	222
18.8.2.	模式 0 波特率.....	222
18.8.3.	模式 2 波特率.....	223
18.8.4.	模式 1 & 3 波特率.....	224
18.9.	串口 0 模式 4 (SPI 主机).....	236
18.10.	串口 0 寄存器.....	238
18.11.	串口 0 增强功能.....	241
18.11.1.	S0 波特率发生器(S0BRG).....	242
18.11.2.	独立波特率发生器 S0BRG 用于 S0.....	243
18.11.3.	S0 LIN 总线寄存器.....	243
18.11.4.	S0 当做 16 位定时器模式.....	244
18.11.5.	S0 当做 16 位定时器模式.....	244
18.11.6.	S0BRG 可编程时钟输出.....	245
19.	串口 1 (UART1).....	246
19.1.	串口 1 波特率发生器 (S1BRG).....	246
19.2.	串口 1 波特率设定.....	247
19.2.1.	模式 0 波特率.....	247
19.2.2.	模式 2 波特率.....	247
19.2.3.	模式 1 & 3 波特率.....	248
19.3.	串口 1 模式 4 (SPI 主机).....	252
19.4.	S1BRG 8 位定时器模式.....	254
19.5.	S1BRG 16 位定时器模式.....	254
19.6.	S1BRT 可编程时钟输出.....	255
19.7.	S1 波特率发生器用于 S0.....	256
19.8.	串口 1 寄存器.....	257
20.	串口 2 (UART2).....	260
20.1.	串口 2 波特率发生器 (S2BRG).....	260
20.2.	串口 2 波特率设定.....	261
20.2.1.	模式 0 波特率.....	261
20.2.2.	模式 2 波特率.....	261
20.2.3.	模式 1 & 3 波特率.....	262
20.3.	串口 2 模式 4(SPI 主机).....	262

20.4.	S2BRG 8 位定时器模式.....	262
20.5.	S2BRG 16 位定时器模式.....	263
20.6.	S2BRT 可编程时钟输出.....	263
20.7.	串口 2 寄存器 .....	264
21.	串口 3 (UART3).....	268
21.1.	串口 3 波特率发生器 (S3BRG).....	268
21.2.	串口 3 波特率设定 .....	269
21.2.1.	模式 0 波特率 .....	269
21.2.2.	模式 2 波特率 .....	269
21.2.3.	模式 1 & 3 波特率.....	269
21.2.4.	串口 3 模式 4 (SPI 主机) .....	269
21.3.	S3BRG 8 位定时器模式.....	270
21.4.	S3BRG 16 位定时器模式.....	270
21.5.	S3BRT 可编程时钟输出.....	270
21.6.	串口 3 寄存器 .....	272
22.	串行外设接口(SPI).....	275
22.1.	典型 SPI 配置 .....	276
22.1.1.	单主机和单从机.....	276
22.1.2.	双设备, 既是主机也是从机.....	276
22.1.3.	单主机和多从机.....	276
22.2.	SPI 配置 .....	277
22.2.1.	一个从机的补充注意事项 .....	277
22.2.2.	一个主机的补充注意事项 .....	277
22.2.3.	nSS 引脚模式改变.....	277
22.2.4.	发送保持寄存器满标志.....	278
22.2.5.	写冲突 .....	278
22.2.6.	SPI 时钟速率选择.....	278
22.3.	数据模式.....	279
22.4.	菊花链连接 .....	281
22.4.1.	菊花链配置 .....	281
22.5.	QPI 模式连接.....	282
22.5.1.	QPI 模式配置 .....	282
22.6.	SPI 寄存器.....	283
23.	双线串行接口(TWI0/I2C0 & TWI1/I2C1) .....	286
23.1.	操作模式.....	287
23.1.1.	主机发送模式 .....	287
23.1.2.	主机接收模式 .....	287
23.1.3.	从机发送模式 .....	288
23.1.4.	从机接收模式 .....	288
23.1.5.	多从机地址识别.....	289
23.2.	混合状态.....	290
23.3.	使用 TWI/I2C .....	290
23.4.	TWI0/I2C0 寄存器 .....	296
23.5.	TWI1/I2C1 寄存器 .....	300
24.	串行接口侦测(STWI/SI2C).....	304
24.1.	SID 结构 .....	304
24.2.	SID 寄存器.....	304
25.	蜂鸣器 .....	306
25.1.	蜂鸣器寄存器.....	306
26.	键盘中断 (KBI).....	307
26.1.	KBI 结构 .....	307
26.2.	KBI 寄存器.....	308
27.	通用逻辑(GPL-CRC).....	310
27.1.	GPL-CRC 结构 .....	310
27.2.	GPL-BOREV 结构 .....	311
27.3.	GPL 寄存器 .....	311
28.	输出信号中止&调制 0/1 (OBM0/1) .....	313

28.1.	OBM 结构.....	314
28.2.	OBM0 寄存器.....	315
28.3.	OBM1 寄存器.....	319
29.	12 位 ADC.....	323
29.1.	ADC 结构.....	323
29.2.	ADC 操作.....	324
29.2.1.	ADC 输入通道.....	324
29.2.2.	ADC 内部电压参考.....	324
29.2.3.	开启一次转换.....	324
29.2.4.	ADC 转换率.....	325
29.2.5.	ADC 中断.....	325
29.2.6.	ADC 窗口侦测.....	326
29.2.7.	ADC 通道扫描模式.....	327
29.2.8.	DMA 传输 ADC 数据.....	327
29.2.9.	I/O 引脚用于 ADC 功能.....	327
29.2.10.	空闲和掉电模式.....	328
29.2.11.	如何提高 ADC 精准度.....	328
29.3.	ADC 寄存器.....	328
30.	模拟比较器 0/1/2 (AC0/AC1/AC2).....	335
30.1.	AC0/AC1/AC2 结构.....	335
30.2.	AC0/AC1/AC2 寄存器.....	337
31.	内部参考电压 (IVR, 1.4V).....	343
31.1.	IVR (1.4V)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.	MGEQ1C064 Flash 存储器配置.....	345
32.2.	MGEQ1C064 Flash 在 SP/IAP 上的访问.....	347
32.2.1.	ISP/IAP Flash 页擦除模式.....	348
32.2.2.	ISP/IAP Flash 字节编程模式.....	349
32.2.3.	ISP/IAP Flash 读模式.....	350
32.3.	ISP 操作.....	351
32.3.1.	硬件启动 ISP 方法.....	351
32.3.2.	软件启动 ISP 方法.....	351
32.3.3.	ISP 注意事项.....	352
32.4.	在应用编程(IAP).....	353
32.4.1.	MGEQ1C064 IAP 存储边界/范围.....	353
32.4.2.	更新 IAP-存储中的数据.....	353
32.4.3.	IAP 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 页访问.....	358
34.	辅助特殊功能寄存器.....	363
35.	硬件选项.....	373
36.	应用说明.....	375
36.1.	电源电路.....	375
36.2.	复位电路.....	375
36.3.	XTAL 振荡电路.....	376
36.4.	ICP 和 OCD 接口电路.....	376
36.5.	在芯片编程功能.....	378
36.6.	在线调试功能.....	379
37.	电气特性.....	380
37.1.	绝对最大额定值.....	380
37.2.	DC 特性.....	381
37.3.	外部时钟特性.....	383
37.4.	IHRCO 特性.....	383
37.5.	ILRCO 特性.....	383

---

37.6.	CKM 特性 .....	384
37.7.	Flash 特性 .....	384
37.8.	ADC 特性.....	385
37.9.	IVR 特性 .....	386
37.10.	模拟比较器 AC0/AC1/AC2 特性.....	387
37.11.	串口时序特性.....	389
37.12.	SPI 时序特性 .....	390
37.13.	EMB 时序特性 .....	392
38.	指令集 .....	393
39.	封装尺寸 .....	396
39.1.	LQFP-48 (7mm X 7mm) 尺寸 .....	396
40.	版本历史.....	397
41.	免责声明.....	398

## 图目录

图 2-1. MGEQ1C064 方框图.....	23
图 4-1. MGEQ1C064AD48 LQFP48 顶视图.....	32
图 6-1. 程序存储器.....	48
图 6-2. 数据存储器.....	49
图 6-3. RAM 内部 RAM 的低 128 字节.....	50
图 6-4. 特殊功能寄存器 SFR 空间.....	50
图 6-5. 无地址相位 EMB 结构.....	52
图 6-6. 无地址相位 EMB 用于 8 位数据访问.....	52
图 7-1. 双 DPTR 结构.....	54
图 7-2. XRPS 结构.....	56
图 8-1. DMA 访问图.....	57
图 8-2. DMA 结构.....	58
图 8-3. DMA 中断.....	61
图 8-4. 定时器 5 结构.....	62
图 8-5. 定时器 6 结构.....	63
图 9-1. 系统时钟.....	69
图 10-1. 看门狗定时器.....	75
图 11-1. 实时时钟计数器.....	79
图 12-1. 系统复位源.....	82
图 13-1. 低电压监测器 0/1.....	86
图 13-2. 掉电模式唤醒结构.....	88
图 14-1. 端口 3 准双向口结构.....	93
图 14-2. 端口 3 推挽输出结构.....	93
图 14-3. 端口 3 仅输入.....	93
图 14-4. 端口 3 开漏输出.....	94
图 14-5. 通用仅模拟输入.....	94
图 14-6. 带上拉电阻的通用开漏输出.....	95
图 14-7. 通用开漏输出.....	95
图 14-8. 通用推挽输出.....	96
图 15-1. 中断系统.....	105
图 15-2. 系统标志位中断配置.....	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 0 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&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 (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. 定时器 2 波特率发生器模式.....	143
图 16-24. 分立定时器 2 波特率发生器模式.....	144
图 16-25. 定时器 2 时钟输出公式.....	145
图 16-26. 定时器 2 时钟输出模式.....	145
图 16-27. 分立定时器 2 时钟输出公式.....	146
图 16-28. 分立定时器 2 时钟输出模式.....	146
图 16-29. 定时器 3 模式 0 结构(自动重载和外部中断模式).....	151
图 16-30. 定时器 3 模式 1 结构(外部中断触发自动重载模式).....	152
图 16-31. 定时器 3 模式 2 结构(捕捉模式).....	153
图 16-32. 定时器 3 模式 3 结构(捕捉带自动清零 TL3&TH3).....	154
图 16-33. 定时器 3 模式 6 (占空比捕捉).....	155
图 16-34. 分立定时器 3 模式 0 结构(自动重载和外部中断).....	156
图 16-35. 分立定时器 3 模式 1 结构(外部中断触发自动重载).....	157
图 16-36. 分立定时器 3 模式 2 结构(捕捉).....	158
图 16-37. 分立定时器 3 模式 3 结构(捕捉带自动清零 TH3).....	159
图 16-38. 分立定时器 3 模式 4 结构(8 位 PWM 模式).....	160
图 16-39. 定时器 3 时钟输出公式.....	161
图 16-40. 定时器 3 时钟输出模式.....	161
图 16-41. 分立定时器 3 时钟输出公式.....	162
图 16-42. 分立定时器 3 时钟输出模式.....	162
图 16-43. 定时器 4 模式 0 结构(自动重载和外部中断模式).....	167
图 16-44. 定时器 3 模式 1 结构(外部中断触发自动重载模式).....	168
图 16-45. 定时器 4 模式 2 结构(捕捉模式).....	169
图 16-46. 定时器 4 模式 3 结构(捕捉带自动清零 TL4 & TH4).....	170
图 16-47. 定时器 4 模式 6 结构(占空比捕捉).....	171
图 16-48. 分立定时器 4 模式 0 结构(自动重载和外部中断).....	172

图 16-49. 分立定时器 4 模式 1 结构(外部中断触发自动重载)	173
图 16-50. 分立定时器 4 模式 2 结构(捕捉)	174
图 16-51. 分立定时器 4 模式 3 结构(捕捉带自动清零 TH4)	175
图 16-52. 分立定时器 4 模式 4 结构(8 位 PWM 模式)	176
图 16-53. 定时器 4 时钟输出公式	177
图 16-54. 定时器 4 时钟输出模式	177
图 16-55. 分立定时器 4 时钟输出公式	178
图 16-56. 分立定时器 4 时钟输出模式	178
图 17-1. PCA 方框图	185
图 17-2. PCA 定时/计数器	186
图 17-3. PCA 中断系统	188
图 17-4. PCA 捕捉模式	193
图 17-5. PCA 缓冲捕捉模式(BMEn=1, n= 0, 2, 4)	194
图 17-6. PCA 缓冲捕捉模式波形	194
图 17-7. PCA 软件定时器模式	195
图 17-8. PCA 高速输出模式	195
图 17-9. PCA 缓冲 8 位 PWM 模式	196
图 17-10. PCA 无缓冲 10/12/16 位 PWM 模式	197
图 17-11. PCA 缓冲 10/12/16 位 PWM 模式 (带死区控制)	198
图 17-12. PCA COPM 模式	199
图 17-13. PCA 缓冲 COPM 模式	200
图 17-14. PCA 通道作为 FIFO 数据模式	201
图 17-15. PWM 带死区控制的波形	202
图 17-16. 边沿对齐和中心对齐的 PWM 波形	203
图 17-17. PWM 中止控制的锁存模式波形	204
图 17-18. PWM 中止控制的逐周期模式波形	204
图 17-19. PCA PWM Break control source	205
图 17-20. PCA 模块输出控制	206
图 17-21. POEn 对齐输出控制 (例如.PWM 边沿对齐的波形)	207
图 17-22. 可变分辨率的 PWM 中心对齐	212
图 18-1. 模式 1 数据帧	213
图 18-2. 模式 2,3 数据帧	214
图 18-3. 串口 0 模式 0	215
图 18-4. 模式 0 发送波形	216
图 18-5. 模式 0 接收波形	216
图 18-6. 串口模式 1, 2, 3	217
图 18-7. UART0 帧错误侦测	218
图 18-8. UART0 多处理器通讯	219
图 18-9. 自动地址识别	220
图 18-10. S0 的波特率源	222
图 18-11. 串口 0 模式 4, 单主机和单从机架构(n = 0)	236

图 18-12. 串口 0 模式 4，单主机和多从机架构( $n = 0$ ) .....	236
图 18-13. 串口 0 模式 4 发送波形 ( $n = 0$ ) .....	237
图 18-14. S0BRG 配置.....	242
图 18-15. S0 8 位定时器模式.....	244
图 18-16. S0 16 位定时器模式.....	244
图 18-17. S0BRG 时钟输出(S0BRG 为 8 位定时器模式).....	245
图 18-18. S0BRG 时钟输出(S0BRG 用于 UART 模式).....	245
图 19-1. S1BRG 结构 (S1TME=0) .....	246
图 19-2. 串口模式 4，单主机和单从机架构( $n = 1$ ).....	252
图 19-3. 串口模式 4，单主机和多从机架构 ( $n = 1$ ).....	252
图 19-4. 串口 1 模式 4 传送波形 ( $n = 1$ ) .....	253
图 19-5. S1BRG 8 位定时器模式结构 (S1TME=1) .....	254
图 19-6. S1BRG 16 位定时器模式结构(S1TME=1) .....	254
图 19-7. S1BRG 时钟输出 (S1BRG 8 位定时器模式) .....	255
图 19-8. S1BRG 时钟输出 (S1BRG 用于 UART 模式).....	255
图 19-9. UART0 额外波特率源.....	256
图 20-1. S2BRG 结构(S2TME=0) .....	260
图 20-2. S2BRG 8 位定时器模式结构(S2TME=1) .....	262
图 20-3. S2BRG 16 位定时器模式结构 (S2TME=1) .....	263
图 20-4. S2BRG 时钟输出(S2BRG 8 位定时器模式) .....	263
图 20-5. S2BRG 时钟输出 (S2BRG 用于 UART 模式).....	264
图 21-1. S3BRG 结构 (S3TME=0) .....	268
图 21-2. S3BRG 8 位定时器模式结构(S3TME=1) .....	270
图 21-3. S3BRG 16 位定时器模式结构(S3TME=1) .....	270
图 21-4. S3BRG 时钟输出(S3BRG 8 位定时器模式) .....	271
图 21-5. S3BRG 时钟输出(S3BRG 用于 UART 模式).....	271
图 22-1. SPI 方框图 .....	275
图 22-2. SPI 单主机和单从机结构.....	276
图 22-3. SPI 双设备结构，既是主机也是从机.....	276
图 22-4. SPI 单主机和多从机结构.....	276
图 22-5. SPI 在 CPHA=0 时从机传送格式 .....	279
图 22-6. SPI 在 CPHA=1 时从机传送格式 .....	279
图 22-7. SPI 在 CPHA=0 时主机传送格式 .....	280
图 22-8. SPI 在 CPHA=1 时主机传送格式 .....	280
图 22-9. SPI 从机在菊花链连接结构 .....	281
图 22-10. QPI 单主机&单从机结构.....	282
图 23-1. TWI/I2C 总线互联框图 .....	286
图 23-2. TWI/I2C 方框图 .....	286
图 23-3. 多从机地址识别.....	289
图 24-1. 串行接口侦测结构 .....	304
图 25-1. 蜂鸣器发生器.....	306

图 26-1. 键盘中断(KBI)结构 .....	307
图 27-1. CRC 结构 .....	310
图 27-2. BOREV 结构 .....	311
图 28-1. 输出信号中止&调制器(OBMn) 调制信号 .....	313
图 28-2. 输出信号中止&调制器 n (OBMn) 结构(n = 0, 1) .....	314
图 29-1. ADC 方框图 .....	323
图 29-2. ADC 中断 .....	325
图 29-3. ADC 转换时序 .....	326
图 29-4. ADC 窗口侦测 .....	326
图 30-1. 模拟比较器 0 方框图 .....	335
图 30-2. 模拟比较器 1 方框图 .....	336
图 30-3. 模拟比较器 2 方框图 .....	336
图 31-1. IVR 图解 .....	343
图 32-1. MGEQ1C064 Flash 存储器配置 .....	346
图 32-2. ISP/IAP 页擦除流程 .....	348
图 32-3. ISP/IAP 页擦除操作的示例代码 .....	348
图 32-4. ISP/IAP 字节编程流程 .....	349
图 32-5. ISP/IAP 字节编程的示例代码 .....	349
图 32-6. ISP/IAP 字节读流程 .....	350
图 32-7. ISP/IAP 字节读取的范例代码 .....	350
图 32-8. ISP 示例代码 .....	357
图 36-1. 电源电路 .....	375
图 36-2. 复位电路 .....	375
图 36-3. XTAL 振荡电路 .....	376
图 36-4. ICP 和 OCD 接口电路 .....	377
图 36-5. 经 ICP32 (MLink)的独立编程 .....	378
图 36-6. 经 ICP 的独立编程 (8051 OCD ICE Adapter) .....	378
图 36-7. MLink ICE 功能的系统框图 .....	379
图 36-8. ICE 功能的系统框图 .....	379
图 37-1. 外部时钟驱动波形 .....	383
图 37-2. 移位寄存器模式时序波形 .....	389
图 37-3. CPHA=0 时 SPI 主机传送波形 .....	390
图 37-4. CPHA=1 时 SPI 主机传送波形 .....	390
图 37-5. CPHA=0 时 SPI 从机传送波形 .....	391
图 37-6. CPHA=1 时 SPI 从机传送波 .....	391
图 37-7. EMB 读周期(无地址相位) .....	392
图 37-8. EMB 写周期(无地址相位) .....	392
图 39-1. LQFP-48 (7mm X 7mm ) 封装尺寸 .....	396



## 表目录

表 3.1. SFR 图 (页 0~F) .....	24
表 3.2. SFR 位分配(页 0~F).....	26
表 3.3.辅助 SFR 图 (P 页).....	30
表 3.4.辅助 SFR 位分配(P 页).....	31
表 4.1. 引脚描述.....	33
表 8.1. DMA 数据路径选择 .....	59
表 14.1.可用 I/O 引脚数量.....	92
表 14.2.端口 3 配置设定.....	96
表 14.3. 通用端口配置设定 .....	96
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 .....	109
表 17.1. PCA 模块模式 .....	192
表 18.1. 串口 0 模式选择.....	213
表 18.2. SMOD2 在模式 2 的应用标准 .....	223
表 18.3. S0 模式 2 波特率 @ F <sub>SYSClk</sub> =11.0592MHz .....	223
表 18.4. S0 模式 2 波特率 @ F <sub>SYSClk</sub> =12.00MHz .....	223
表 18.5. SMOD2 在模式 1 和 3 使用定时器 1 的应用标准.....	224
表 18.6. 定时器 1 产生的常用波特率@ F <sub>SYSClk</sub> =11.0592MHz.....	224
表 18.7. 定时器 1 产生的高波特率@ F <sub>SYSClk</sub> =11.0592MHz .....	224
表 18.8. 定时器 1 产生的常用波特率@ F <sub>SYSClk</sub> =22.1184MHz.....	225
表 18.9. 定时器 1 产生的高波特率@ F <sub>SYSClk</sub> =22.1184MHz .....	225
表 18.10. 定时器 1 产生的常用波特率@ F <sub>SYSClk</sub> =12.0MHz.....	225
表 18.11. 定时器 1 产生的高波特率@ F <sub>SYSClk</sub> =12.0MHz .....	226
表 18.12. 定时器 1 产生的常用波特率@ F <sub>SYSClk</sub> =24.0MHz.....	226
表 18.13. 定时器 1 产生的高波特率@ F <sub>SYSClk</sub> =24.0MHz .....	226
表 18.14. 定时器 1 产生的常用波特率@ F <sub>SYSClk</sub> =29.4912MHz.....	226
表 18.15. . 定时器 1 产生的高波特率@ F <sub>SYSClk</sub> =29.4912MHz .....	227
表 18.16. 定时器 1 产生的常用波特率@ F <sub>SYSClk</sub> =44.2368MHz.....	227
表 18.17. 定时器 1 产生的高波特率@ F <sub>SYSClk</sub> =44.2368MHz .....	227
表 18.18. 定时器 1 产生的常用波特率@ F <sub>SYSClk</sub> =32MHz.....	228
表 18.19. 定时器 1 产生的高波特率@ F <sub>SYSClk</sub> =32MHz .....	228
表 18.20. 定时器 1 产生的常用波特率@ F <sub>SYSClk</sub> =48.0MHz.....	228
表 18.21. 定时器 1 产生的高波特率@ F <sub>SYSClk</sub> =48.0MHz .....	229
表 18.22. SMOD2 在模式 1 和 3 使用定时器 2 的应用标准.....	230
表 18.23. 定时器 2 产生的常用波特率@ F <sub>SYSClk</sub> =11.0592MHz.....	230
表 18.24. 定时器 2 产生的高波特率@ F <sub>SYSClk</sub> =11.0592MHz .....	230
表 18.25. 定时器 2 产生的常用波特率@ F <sub>SYSClk</sub> =22.1184MHz.....	231

表 18.26. 定时器 2 产生的高波特率@ $F_{SYSCLK}=22.1184MHz$ .....	231
表 18.27. 定时器 2 产生的常用波特率@ $F_{SYSCLK}=12.0MHz$ .....	231
表 18.28. 定时器 2 产生的高波特率@ $F_{SYSCLK}=12.0MHz$ .....	232
表 18.29. 定时器 2 产生的常用波特率@ $F_{SYSCLK}=24.0MHz$ .....	232
表 18.30. 定时器 2 产生的高波特率@ $F_{SYSCLK}=24.0MHz$ .....	232
表 18.31. 定时器 2 产生的常用波特率@ $F_{SYSCLK}=29.4912MHz$ z.....	232
表 18.32. 定时器 2 产生的高波特率@ $F_{SYSCLK}=29.4912MHz$ .....	233
表 18.33. 定时器 2 产生的常用波特率@ $F_{SYSCLK}=44.2368MHz$ .....	233
表 18.34. 定时器 2 产生的高波特率@ $F_{SYSCLK}=44.2368MHz$ .....	233
表 18.35. 定时器 2 产生的常用波特率@ $F_{SYSCLK}=32MHz$ .....	234
表 18.36. 定时器 2 产生的高波特率@ $F_{SYSCLK}=32MHz$ .....	234
表 18.37. 定时器 2 产生的常用波特率@ $F_{SYSCLK}=48.0MHz$ .....	234
表 18.38. 定时器 2 产生的高波特率@ $F_{SYSCLK}=48.0MHz$ .....	235
表 18.39. SMOD2 在分割定时器 2 模式 1&3 下的应用条件.....	235
表 18.40. SMOD2 在 S0BRG 模式 1&3 下的应用条件 .....	235
表 18.41.串口模式 4 SPI 模式映射配置 .....	237
表 19.1. S1 模式 2 波特率@ $F_{SYSCLK}=11.0592MHz$ .....	247
表 19.2. S1 模式 2 波特率@ $F_{SYSCLK}=12.00MHz$ .....	247
表 19.3. S1BRG 产生的常用波特率@ $F_{SYSCLK}=11.0592MHz$ .....	248
表 19.4. S1BRG 产生的常用波特率@ $F_{SYSCLK}=22.1184MHz$ .....	248
表 19.5. S1BRG 产生的常用波特率@ $F_{SYSCLK}=12.0MHz$ .....	249
表 19.6. S1BRG 产生的常用波特率@ $F_{SYSCLK}=24.0MHz$ .....	249
表 19.7. S1BRG 产生的常用波特率@ $F_{SYSCLK}=29.4912MHz$ .....	249
表 19.8. S1BRG 产生的常用波特率@ $F_{SYSCLK}=44.2368MHz$ .....	250
表 19.9. S1BRG 产生的常用波特率@ $F_{SYSCLK}=32.0MHz$ .....	250
表 19.10. S1BRG 产生的常用波特率@ $F_{SYSCLK}=48.0MHz$ .....	251
表 19.11. 串口 1 模式选择 .....	252
表 19.12.串口 1 模式 4 的 SPI 模式配置 .....	253
表 19.13. S1BRG 产生的常用波特率 for S0 @ $F_{SYSCLK}=11.0592MHz$ .....	256
表 20.1. S2 模式 2 波特率@ $F_{SYSCLK}=11.0592MHz$ .....	261
表 20.2. S2 模式 2 波特率@ $F_{SYSCLK}=12.00MHz$ .....	261
表 21.1. S3 模式 2 波特率@ $F_{SYSCLK}=11.0592MHz$ .....	269
表 21.2. S3 模式 2 波特率@ $F_{SYSCLK}=12.00MHz$ .....	269
表 22.1. SPI 主机和从机选择 .....	277
表 22.2. SPI 串行时钟速率.....	278
表 22.3. SPI 模式定义.....	279
表 23.1. TWI0/I2C0 串行时钟速率 .....	297
表 23.2. TWI1/I2C1 串行时钟速率 .....	301
表 30.1 NVRL = 0, 选择高区域 .....	338
表 30.2. NVRL = 1, 选择低区域 .....	338
表 36.1.振荡电路的电容 C1 & C2 参照表 .....	376

表 38.1. 指令集 ..... 393

## 1. 概述

**MGEQ1C064**符合 AEC-Q100 2 级认证,可在  $-40^{\circ}\text{C}$  至  $105^{\circ}\text{C}$  工作温度、 $2.4\text{V}\sim 5.5\text{V}$  工作电压范围内运行,并搭载 LIN 接口以实现稳定可靠的通信。该芯片是基于高性能 1-T 架构 80C51 CPU 的单片机,指令执行周期为  $1\sim 7$  个时钟周期(执行速度约为标准 8051 的  $6\sim 7$  倍),且指令集与 8051 完全兼容。因此,在与标准 8051 性能相同的情况下,**MGEQ1C064** 可工作在更低的时钟频率下,从而大幅降低功耗。

**MGEQ1C064**有 64K 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。这些功能都由内建的电荷泵提供编程用的高压不需要外部提供高压。

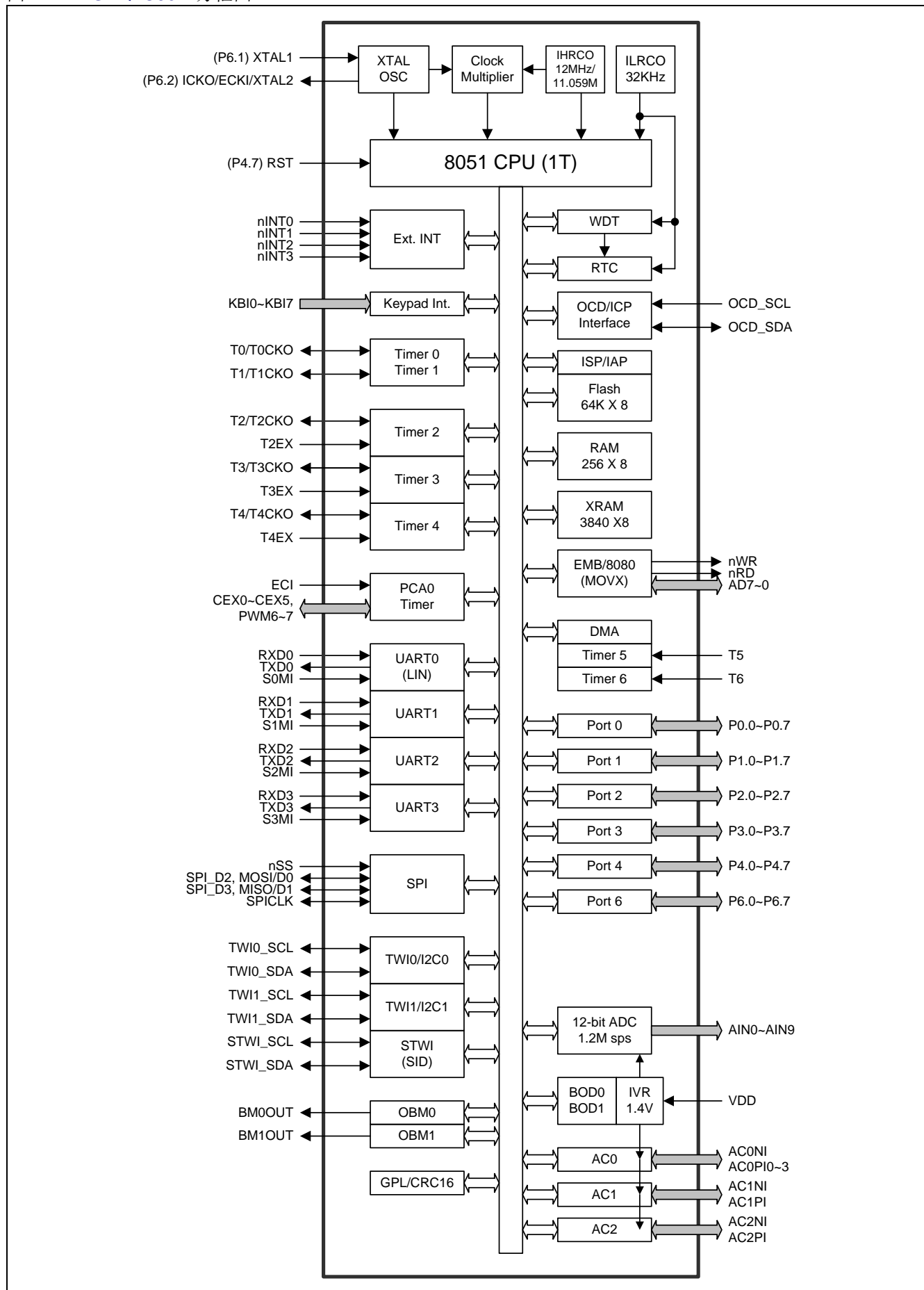
**MGEQ1C064**除了 80C52MCU 的标准功能,256 字节的随机存储器,4 个外部中断可选高/低触发,1 个多源 4 级中断控制和 5 个定时/计数器),**MGEQ1C064**有 48 个额外的 I/O 端口引脚,3840 字节外部数据存储(XRAM),1.0M sps 12 位 ADC,5 个 16 位定时器,1 个 8 通道带死区控制的 PCA(PCA0),1 个 8 位 SPI,3 个 TWI/I2C (TWI0/I2C0, TWI1/I2C1 and STWI/ SI2C),1 个键盘中断,3 个模拟比较器,看门狗定时器,实时时钟(RTC)模块,2 个低电压监测器,1 个外部时钟输入 ECK1 (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 4 个串口(UART0~3)用来促进多处理器的通讯, LIN 总线模式及一个速度增强设备(X2/X4 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。

**MGEQ1C064**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,使用者可以通过 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,**MGEQ1C064**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. MGEQ1C064 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	P5	CH	CCAP0H	CCAP1H	CCAP2H	CCAP3H	CCAP4H	CCAP5H
	1	P6		CCAP6H	CCAP7H				
	2	P7		--	--				
	3~F	--		--	--				
F0	0	B	PAOE	PCAPW M0	PCAPWM1	PCAPWM2	PCAPWM3	PCAPWM4	PCAPWM5
	1			PCAPW M6	PCAPWM7				
	2~F								
E8	0	P4	CL	CCAP0L	CCAP1L	CCAP2L	CCAP3L	CCAP4L	CCAP5L
	1			CCAP6L	CCAP7L				
	2~F			--	--				
E0	0~F	ACC	WDTCR	IFD	IFADRH	IFADRL	IFMT	SCMD	ISPCR
D8	0	CCON	CMOD	CCAPM0	CCAPM1	CCAPM2	CCAPM3	CCAPM4	CCAPM5
	1			CCAPM6	CCAPM7				
	2~F			--	--				
D0	0	PSW	SIADR	SIDAT	SISTA	SICON	KBPATN	KBCON	KBMASK
	1		SI1ADR	SI1DAT	SI1STA	SI1CON			
	2		SIA2	SIA2M	--	--			
	3		SI1A2	SI1A2M	--	--			
	4~F		--	--	--	--			
C8	0	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	CLRL	CHRL
	1	T3CON	T3MOD	RCAP3L	RCAP3H	TL3	TH3		
	2	T4CON	T4MOD	RCAP4L	RCAP4H	TL4	TH4		
	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		ADCFG6						
	B		ADCFG11						
	C		ADCFG12						
	D		ADCFG13						
	E		ADCFG14						
F	--								
B8	0	IP0L	SADEN/ SOCR1	--	--	PWMCR	CRC0DA	RTCCR	CKCON1
	1					PDTCRA			
	2					BM0CR			
	3					BM0SE			
	4					BM0LCR			
	5					BM0HCR			
	6					BM1CR			
	7					BM1SE			
	8					BM1LCR			
	9					BM1HCR			
	A~F					--			
B0	0	P3	P3M0	P3M1	P4M0	--	P5M0	RTCTM	IP0H
	1				--	P6M0			
	2				--	PDRVC0	P7M0		
	3				--	PDRVC1	EMBTC		
	4~F				--	--	--		
A8	0~F	IE	SADDR	--	--	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				
	A					AUXR13				
	B					AUXR14				
	C~F					--				
	98					0				S0CON
1		S1CON	S1BUF	S1BRT	S1BRC	S1CFG	AC1CON	AC1MOD		
2		S2CON	S2BUF	S2BRT	S2BRC	S2CFG	--	AC2CON	AC2MOD	
3		S3CON	S3BUF	S3BRT	S3BRC	S3CFG	--	--	--	
4		S3CON	S3BUF	S3BRT	S3BRC	S3CFG	--	--	--	
5~F	--	--	--	--	--	--	--	--		
90	0	P1	P1M0	P1M1	P0M0	DMACR0	P2M0	BOREV	PCON1	
	1			P2M1	T2MOD1		TREN0			
	2			P4M1	T3MOD1		TRLC0			
	3			P6M1	T4MOD1		TSPC0			
	4			P0M1	--		--			
	5			P5M1	--		--			
	6			P7M1	--		--			
	7			P3FDC	--		--			
	8			P1FDC	--		DMACG0			
	9			P2FDC	--		DMADS0			
	A			P4FDC	--		--			
	B			P0FDC	--		--			
	C			P6FDC	--		--			
	D			P5FDC	--		--			
	E			P7FDC	--		--			
	F			--	--		--			
88	0~F	TCON	TMOD	TL0	TL1	TH0	TH1	SFIE	XRPS	
80	0~F	P0	SP	DPL	DPH	SPSTAT	SPCON	SPDAT	PCON0	
		0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

\*: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 = xxxx-0000

7	6	5	4	3	2	1	0
--	--	--	--	IDX3	IDX2	IDX1	IDX0
W	W	W	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(页 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	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111
SP	堆栈指针	81	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00001111
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	--	QPIEN	SPR2/ QDOE	00000000
SPCON	SPI控制寄存器	85	0~F	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	00000100
SPDAT	SPI数据寄存器	86	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
PCON0	电源控制寄存器0	87	0~F	SMOD1	SMOD0	GF	POF0	GF1	GF0	PD	IDL	00010000
TCON	定时器控制寄存器	88	0~F	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000
TMOD	定时器模式寄存器	89	0~F	T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0	00000000
TL0	定时器0低8位	8A	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TL1	定时器1低8位	8B	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH0	定时器0高8位	8C	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
TH1	定时器1高8位	8D	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
SFIE	系统标志中断使能	8E	0~F	SIDFIE	MCDRE	MCDFIE	RTCFIE	SPWIE	BOF1IE	BOF0IE	WDTFIE	01100000
XRPS	XRAM页选择	8F	0~F	--	--	--	--	.3	.2	.1	.0	xxxx0000
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	P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0	11111111
P4M1	P4模式寄存器1	92	2	P4M1.7	P4M1.6	P4M1.5	P4M1.4	P4M1.3	P4M1.2	P4M1.1	P4M1.0	11111111
P6M1	P6模式寄存器1	92	3	P6M1.7	P6M1.6	P6M1.5	P6M1.4	P6M1.3	P6M1.2	P6M1.1	P6M1.0	11111111
P0M1	P0模式寄存器 1	92	4	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0	11111111
P5M1	P5模式寄存器1	92	5	P5M1.7	P5M1.6	P5M1.5	P5M1.4	P5M1.3	P5M1.2	P5M1.1	P5M1.0	11111111
P7M1	P7模式寄存器r 1	92	6	--	--	--	--	--	P7M1.2	P7M1.1	P7M1.0	11111111
P3FDC	P3 快速驱动控制	92	7	P3FDC.7	P3FDC.6	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	P2FDC.7	P2FDC.6	P2FDC.5	P2FDC.4	P2FDC.3	P2FDC.2	P2FDC.1	P2FDC.0	00000000
P4FDC	P4 快速驱动控制	92	A	--	P4FDC.6	P4FDC.5	P4FDC.4	P4FDC.3	P4FDC.2	P4FDC.1	P4FDC.0	00000000
P0FDC	P0 快速驱动控制	92	B	P0FDC.7	P0FDC.6	P0FDC.5	P0FDC.4	P0FDC.3	P0FDC.2	P0FDC.1	P0FDC.0	00000000
P6FDC	P6 快速驱动控制	92	C	P6FDC.7	P6FDC.6	P6FDC.5	P6FDC.4	P6FDC.3	P6FDC.2	--	--	00000000
P5FDC	P5 快速驱动控制	92	D	P5FDC.7	P5FDC.6	P5FDC.5	P5FDC.4	P5FDC.3	P5FDC.2	P5FDC.1	P5FDC.0	00000000
P7FDC	P7 快速驱动控制	92	E	--	--	--	--	--	P7FDC.2	P7FDC.1	P7FDC.0	00000000
P0M0	P0 模式寄存器 0	93	0	P0M0.7	P0M0.6	P0M0.5	P0M0.4	P0M0.3	P0M0.2	P0M0.1	P0M0.0	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
T4MOD1	定时器4模式1寄存器.	93	3	TL4CS	TF4IG	TL4IS	T4CKS	T4MS1	CP4S2	CP4S1	CP4S0	00000000
DMACR0	DMA控制寄存器. 0	94	0~7	--	--	--	--	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	P2M0.7	P2M0.6	P2M0.5	P2M0.4	P2M0.3	P2M0.2	P2M0.1	P2M0.0	00000000
TREN0	定时器使能寄存器0	95	1	TR4LE	TR3LE	TR2LE	TR4E	TR3E	TR2E	TR1E	TROE	00000000
TRLC0	定时器重载控制寄存器0	95	2	TL4RLC	TL3RLC	TL2RLC	T4RLC	T3RLC	T2RLC	T1RLC	T0RLC	00000000
TSPC0	定时器停止控制寄存器0	95	3	TL4SC	TL3SC	TL2SC	T4SC	T3SC	T2SC	T1SC	T0SC	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
S1CON	串口 1 控制寄存器	98	1~2	SM01	SM11	SM21	REN1	TB81	RB81	TI1	RI1	00000000
S2CON	串口 2 控制寄存器	98	3	SM02	SM12	SM22	REN2	TB82	RB82	TI2	RI2	00000000
S3CON	串口 3 控制寄存器	98	4	SM03	SM13	SM23	REN3	TB83	RB83	TI3	RI3	00000000
S0BUF	串口0 缓存	99	0	.7	.6	.5	.4	.3	.2	.1	.0	xxxxxxxx
S1BUF	串口1 缓存	99	1~2	.7	.6	.5	.4	.3	.2	.1	.0	xxxxxxxx
S2BUF	串口2 缓存	99	3	.7	.6	.5	.4	.3	.2	.1	.0	xxxxxxxx
S3BUF	串口3 缓存	99	4	.7	.6	.5	.4	.3	.2	.1	.0	xxxxxxxx
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
S2BRT	S2波特率定时器	9A	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S3BRT	S3波特率定时器	9A	4	.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
S2BRC	S2波特率计数器	9B	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S3BRC	S3波特率计数器	9B	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000
S0CFG	S0 配置寄存器	9C	0	URTS	SMOD2	URM0X3	SM30	S0DOR	BTI	UTIE	SMOD3	00001000

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	
S1CFG	S1 配置寄存器	9C	1~2	SM31	S1M0X3	S1DOR	S1TR	S1MOD1	S1TX12	S1CKOE	S1TME	00100000
S2CFG	S2 配置寄存器	9C	3	SM32	S2M0X3	S2DOR	S2TR	S2MOD1	S2TX12	S2CKOE	S2TME	00100000
S3CFG	S3 配置寄存器	9C	4	SM33	S3M0X3	S3DOR	S3TR	S3MOD1	S3TX12	S3CKOE	S3TME	00100000
S0CFG1	S0 配置寄存器 1 (LINCFG)	9D	0	SBF0	TXER0	S0SB16	ATBR0	TXRX0	SYNC0	--	--	000000xx
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
AC2CON	AC2 控制寄存器	9E	2	AC2LP	AC2PDX	AC2OUT	AC2F	AC2EN	AC2INV	AC2M1	AC2M0	00x00000
AC0MOD	AC0 模式寄存器	9F	0	NVRS3	NVRS2	NVRS1	NVRS0	NVRL	AC0FLT	AC0PIS1	AC0PIS0	00000000
AC1MOD	AC1 模式寄存器	9F	1	--	--	--	AC1NIS	NVRL1	AC1FLT	--	--	xxx000xx
AC2MOD	AC2 模式寄存器	9F	2	--	--	--	AC2NIS	NVRL2	AC2FLT	--	--	xxx000xx
P2	端口2	A0	0~F	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	11111111
AUXR0	辅助寄存器 0	A1	0~F	P60OC1	P60OC0	P60FD	PBKF	BM1F	BM0F	INT1H	INT0H	00000000
AUXR1	辅助寄存器 1	A2	0~F	--	--	CRCDS1	CRCDS0	--	--	--	DPS	00000000
AUXR2	辅助寄存器 2	A3	0~F	STAF	STOF	--	C0PLK	T1X12	T0X12	T1CKOE	T0CKOE	00000000
AUXR3	辅助寄存器 3	A4	0	T0PS1	T0PS0	BPOC1	BPOC0	S0PS0	TWIPS1	TWIPS0	TOXL	00000000
AUXR4	辅助寄存器 4	A4	1	T2PS1	T2PS0	T1PS1	T1PS0	AC1OE	AC1FLT1	AC0OE	AC0FLT1	00000000
AUXR5	辅助寄存器 5	A4	2	C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	C0PS1	C0PS0	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	POE5	POE4	C0CKOE	SPI0M0	BM1OE1	BM1OE0	BM0OE1	BM0OE0	11000000
AUXR8	辅助寄存器 8	A4	5	POE7	POE6	C0PPS2	--	--	S1COPS	T3PS1	T3PS0	11000000
AUXR9	辅助寄存器 9	A4	6	SIDPS1	SIDPS0	T1G1	T0G1	C0FDC1	C0FDC0	S1PS1	S1PS0	00000000
AUXR10	辅助寄存器 10	A4	7	AC1HC0	AC0HC0	SPIPS1	SPIPS0	S0PS1	SPFACE	TWICF	PAA	11000000
AUXR11	辅助寄存器 11	A4	8	I2C1CF	PAA1	I2C1PS1	I2C1PS0	--	POEM0	C0M0	C0OFS	00000000
AUXR12	辅助寄存器 12	A4	9	--	--	T0COBE	T0COAE	--	--	--	--	00000000
AUXR13	辅助寄存器 13	A4	A	--	S3PS0	--	S2PS0	--	T4FCS	T4PS1	T4PS0	00000000
AUXR14	辅助寄存器 14	A4	B	ADLPS1	ADLPS0	--	--	RWPS0	AC2HC0	AC2OE	AC2FLT1	00000000
EIE2	扩展中断使能 2	A5	0~F	EAC2	ETW11	--	ES3	ET4	ES2	EAC1	ET3	00000000
EIP2L	扩展中断优先级 2低位	A6	0~F	PAC2L	PTW11L	--	PS3L	PT4L	PS2L	PAC1L	PT3L	00000000
EIP2H	扩展中断优先级 2高位	A7	0~F	PAC2H	PTW11H	--	PS3H	PT4H	PS2H	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
SFRPI	SFR 页索引	AC	0~F	--	--	--	--	IDX3	IDX2	IDX1	IDX0	xxx00000
EIE1	扩展中断使能 1	AD	0~F	EAC0	ETW10	EKB	ES1	ESF	EPCA	EADC	ESPI	00000000
EIP1L	扩展中断优先级 1低位	AE	0~F	PAC0L	PTW10L	PKBL	PS1L	PSFL	PPCAL	PADCL	PSPI1	00000000
EIP1H	扩展中断优先级 1高位	AF	0~F	PAC0H	PTW10H	PKBH	PS1H	PSFH	PPCAH	PADCH	PSPIH	00000000
P3	端口 3	B0	0~F	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	11111111
P3M0	P3 模式寄存器 0	B1	0~F	P3M0.7	P3M0.6	P3M0.5	P3M0.4	P3M0.3	P3M0.2	P3M0.1	P3M0.0	00000000
P3M1	P3 模式寄存器 1	B2	0~F	P3M1.7	P3M1.6	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	P4M0.6	P4M0.5	P4M0.4	P4M0.3	P4M0.2	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	P7DC0	P6DC1	P6DC0	P5DC1	P5DC0	P4DC1	P4DC0	00000000
P5M0	P5 模式寄存器 0	B5	0	P5M0.7	P5M0.6	P5M0.5	P5M0.4	P5M0.3	P5M0.2	P5M0.1	P5M0.0	00000000
P6M0	P6 模式寄存器 0	B5	1	P6M0.7	P6M0.6	P6M0.5	P6M0.4	P6M0.3	P6M0.2	P6M0.1	P6M0.0	00000000
P7M0	P7 模式寄存器 0	B5	2	--	--	--	--	--	P7M0.2	P7M0.1	P7M0.0	00000000
EMBTC	EMB 时序控制	B5	3	1	0	0	0	RWSH	RWS2	RWS1	RWS0	10000000
RTCTM	RTC 定时器寄存器	B6	0~F	RTCCS1	RTCCS0	RTCCT5	RTCCT4	RTCCT3	RTCCT2	RTCCT1	RTCCT0	01111111
IP0H	中断优先级 0 高位	B7	0~F	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	00000000
IP0L	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	--	--	00000000
PWMCR	PWM 控制寄存器	BC	0	PCAE	EXDT	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
BM0CR	OBM0 控制寄存器	BC	2	BM0OUT	BM0SW	BM0IE	BM0POL	M0FLT.1	M0FLT.0	BM0M.1	BM0M.0	01000000
BM0SE	OBM0 开关使能寄存器	BC	3	BM0E1.3	BM0E1.2	BM0E1.1	BM0E1.0	BM0E0.3	BM0E0.2	BM0E0.1	BM0E0.0	00000000
BM0LCR	OBM0 低通道控制寄存器	BC	4	BM0LCP	--	--	--	BM0LS.3	BM0LS.2	BM0LS.1	BM0LS.0	00000000
BM0HCR	OBM0 高通道控制寄存器	BC	5	BM0HCP	--	--	--	BM0HS.3	BM0HS.2	BM0HS.1	BM0HS.0	00000000
BM1CR	OBM1 控制寄存器	BC	6	BM1OUT	BM1SW	BM1IE	BM1POL	M1FLT.1	M1FLT.0	BM1M.1	BM1M.0	01000000
BM1SE	OBM1 开关使能寄存器	BC	7	BM1E1.3	BM1E1.2	BM1E1.1	BM1E1.0	BM1E0.3	BM1E0.2	BM1E0.1	BM1E0.0	00000000
BM1LCR	OBM1 低通道控制寄存器	BC	8	BM1LCP	--	--	--	BM1LS.3	BM1LS.2	BM1LS.1	BM1LS.0	00000000
BM1HCR	OBM1 高通道控制寄存器	BC	9	BM1HCP	--	--	--	BM1HS.3	BM1HS.2	BM1HS.1	BM1HS.0	00000000
CRC0DA	CRC0 数据输出	BD	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
RTCCR	RTC 控制寄存器	BE	0~F	RTCE	RTCO	RTCRL5	RTCRL4	RTCRL3	RTCRL2	RTCRL1	RTCRL0	00111111
CKCON1	时钟控制寄存器1	BF	0~F	XTOR	--	MCKSTA 1	MCKSTA 0	OSCSTA 3	OSCSTA 2	OSCSTA1	OSCSTA 0	00000000
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

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		
XICFG2	Ext. INT.配置 2	C1	2	--	--	--	--	INT3IS2	INT3IS1	INT3IS0	INT2IS2	xxxx0000	
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	EADCW1	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	--	--	ARES1	ARES0	ADES0	--	01000000	
ADCFG4	ADC 配置寄存器 4	C3	4	--	ADWM0	ADTM3	ADTM2	--	--	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	
ADCFG6	ADC 配置寄存器 6	C3	6	ASCE.15	ASCE.14	ASCE.13	ASCE.12	ASCE.11	ASCE.10	ASCE.9	ASCE.8	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	ADCW1	CHS3	ADCI	ADCS	CHS2	CHS1	CHS0	00000000	
ADCDL	ADC 数据低位	C5	0~F	ADCV.3	ADCV.2	ADCV.1	ADCV.0	--	--	--	--	0000xxxx	
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	CCKS	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	
T3CON	定时器 3 控制寄存器	C8	1	TF3	EXF3	TF3L	TL3IE	EXEN3	TR3	C/T3	CP/RL3	00000000	
T4CON	定时器 4 控制寄存器	C8	2	TF4	EXF4	TF4L	TL4IE	EXEN4	TR4	C/T4	CP/RL4	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	
T4MOD	定时器 4 模式寄存器	C9	2	T4SPL	TL4X12/ T4EIP	T4EXH	T4X12	TR4L	TR4LC	T4OE	T4MS0	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	
RCAP4L	定时器4捕捉低位	CA	2	.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	
RCAP4H	定时器4捕捉高位	CB	2	.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	
TL4	定时器4低位	CC	2	.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低位	CD	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000	
TH3	定时器3低位	CD	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000	
TH4	定时器4低位	CD	2	.7	.6	.5	.4	.3	.2	.1	.0	00000000	
TH5	定时器5低位	CD	3	.7	.6	.5	.4	.3	.2	.1	.0	00000000	
TH6	定时器6低位	CD	4	.7	.6	.5	.4	.3	.2	.1	.0	00000000	
CLRL	CL重载寄存器	CE	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000	
CHRL	CH重载寄存器	CF	0~F	.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第二 地址寄存器	D1	2	.7	.6	.5	.4	.3	.2	.1	A2E	00000000	
SI1A2	TWI1第二 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	--	11111111	
SI1A2M	SI1A2 掩码寄存器	D2	3	SI1A2M.7	SI1A2M.6	SI1A2M.5	SI1A2M.4	SI1A2M.3	SI1A2M.2	SI1A2M.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	
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	--	--	--	PATN_	SEL	KBIF	00000000
KBMASK	键盘中断掩码	D7	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000	
CCON	PCA 控制寄存器	D8	0~F	CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0	00000000	

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	
CMOD	PCA 模式寄存器	D9	0~F	CIDL	BME4	BME2	BME0	CPS2	CPS1	CPS0	ECF	00000000
CCAPM0	PCA 模块0 模式	DA	0	DTE0	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	00000000
CCAPM6	PCA 模块6 模式	DA	1	BME6	ECOM6	--	CAPN6	MAT6	TOG6	PWM6	ECCF6	00x00000
CCAPM1	PCA 模块1 模式	DB	0	--	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1	x0000000
CCAPM7	PCA 模块7 模式	DB	1	--	ECOM7	--	CAPN7	MAT7	TOG7	PWM7	ECCF7	x0x00000
CCAPM2	PCA 模块2 模式	DC	0~F	DTE2	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2	00000000
CCAPM3	PCA 模块3 模式	DD	0~F	--	ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3	x0000000
CCAPM4	PCA 模块4 模式	DE	0~F	DTE4	ECOM4	CAPP4	CAPN4	MAT4	TOG4	PWM4	ECCF4	00000000
CCAPM5	PCA 模块5 模式	DF	0~F	--	ECOM5	CAPP5	CAPN5	MAT5	TOG5	PWM5	ECCF5	x0000000
ACC	累加器	E0	0~F	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
WDTCR	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	--	--	--	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	xxxxxxx
ISPCR	ISP 控制寄存器	E7	0~F	ISPEN	SWBS	SRST	CFAIL	--	--	--	--	00000xxx
P4	端口 4	E8	0~F	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	11111111
CL	PCA 基准定时器低位	E9	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0L	PCA 模块0 捕捉低位	EA	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP6L	PCA 模块6 捕捉低位	EA	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP1L	PCA 模块1 捕捉低位	EB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP7L	PCA 模块7 捕捉低位	EB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP2L	PCA 模块2 捕捉低位	EC	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP3L	PCA 模块3 捕捉低位	ED	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP4L	PCA 模块4 捕捉低位	EE	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP5L	PCA 模块5 捕捉低位	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~F	POE3	POE2B	POE2A	POE2	POE1	POE0B	POE0A	POE0	10011001
PCAPWM0	PCA PWM0 模式	F2	0	P0RS1	P0RS0	--	--	--	P0INV	ECAP0H	ECAP0L	00xxx000
PCAPWM6	PCA PWM6模式	F2	1	P6RS1	P6RS0	--	--	CCF6	P6INV	ECAP6H	ECAP6L	00xx0000
PCAPWM1	PCA PWM1模式	F3	0	P1RS1	P1RS0	--	--	--	P1INV	ECAP1H	ECAP1L	00xxx000
PCAPWM7	PCA PWM7模式	F3	1	P7RS1	P7RS0	--	--	CCF7	P7INV	ECAP7H	ECAP7L	00xx0000
PCAPWM2	PCA PWM2模式	F4	0~F	P2RS1	P2RS0	--	--	--	P2INV	ECAP2H	ECAP2L	00xxx000
PCAPWM3	PCA PWM3模式	F5	0~F	P3RS1	P3RS0	--	--	--	P3INV	ECAP3H	ECAP3L	00xxx000
PCAPWM4	PCA PWM4模式	F6	0~F	P4RS1	P4RS0	--	--	--	P4INV	ECAP4H	ECAP4L	00xxx000
PCAPWM5	PCA PWM5模式	F7	0~F	P5RS1	P5RS0	--	--	--	P5INV	ECAP5H	ECAP5L	00xxx000
P5	端口 5	F8	0	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0	11111111
P6	端口 6	F8	1	P6.7	P6.6	P6.5	P6.4	P6.3	P6.2	P6.1	P6.0	11111111
P7	端口 7	F8	2	--	--	--	--	--	P7.2	P7.1	P7.0	11111111
CH	PCA 基准定时器高位	F9	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP0H	PCA 模块0 捕捉高位	FA	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP6H	PCA 模块6 捕捉高位	FA	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP1H	PCA 模块1 捕捉高位	FB	0	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP7H	PCA 模块7 捕捉高位	FB	1	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP2H	PCA 模块2 捕捉高位	FC	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP3H	PCA 模块3 捕捉高位	FD	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP4H	PCA 模块4 捕捉高位	FE	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000
CCAP5H	PCA 模块5 捕捉高位	FF	0~F	.7	.6	.5	.4	.3	.2	.1	.0	00000000

### 3.3. 辅助 SFR 图 (P 页)

**MGEQ1C064** 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	CKCON5	PCON2	PCON3	--	--
38	--	--	--	--	--	--	--	--
30	--	--	--	--	--	--	--	--
28	--	--	--	--	--	--	--	--
20	--	--	--	--	--	--	--	--
18	--	--	--	--	--	--	--	--
10	--	--	--	--	--	--	--	--
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	位地址及符号								复位值
			位-7	位-6	位-5	位-4	位-3	位-2	位-1	位-0	
<b>物理字节</b>											
IAPLB	IAP低边界	03H	IAPLB6	IAPLB5	IAPLB4	IAPLB3	IAPLB2	IAPLB1	IAPLB0	0	
CKCON2	时钟控制2	40H	--	--	--	IHRCOE	MCKS1	MCKS0	OSCS1	OSCS0	0001-0000
CKCON3	时钟控制3	41H	WDTCS1	WDTCS0	FWKP	WDTFS	MCKD1	MCKD0	--	--	00000000
CKCON4	时钟控制4	42H	RCSS2	RCSS1	RCSS0	RPSC2	RPSC1	RPSC0	RTCCS3	RTCCS2	00000000
CKCON5	时钟控制5	43H	--	--	--	--	--	--	--	CKMS0	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	--	--	SPWRE	--	--	--	--	00000000
SPCON0	SFR页面控制0	48H	--	P6CTL	P4CTL	WRCTL	--	CKCTL0	PWCTL1	PWCTL0	00000000
DCON0	设备控制0	4CH	HSE	IAPO	HSE1	--	--	IORCTL	RSTIO	OCDE	100xx011
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
<b>逻辑字节</b>											
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	--	RTCF	--	BOF1	BOF0	WDTF	0000x000
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	--	P4.5	P4.4	--	--	--	--	1x11xx11
P6	端口6	F8H	--	--	--	--	--	--	P6.1	P6.0	xxxxxx11

P 页 SFR 写的示例代码:

```

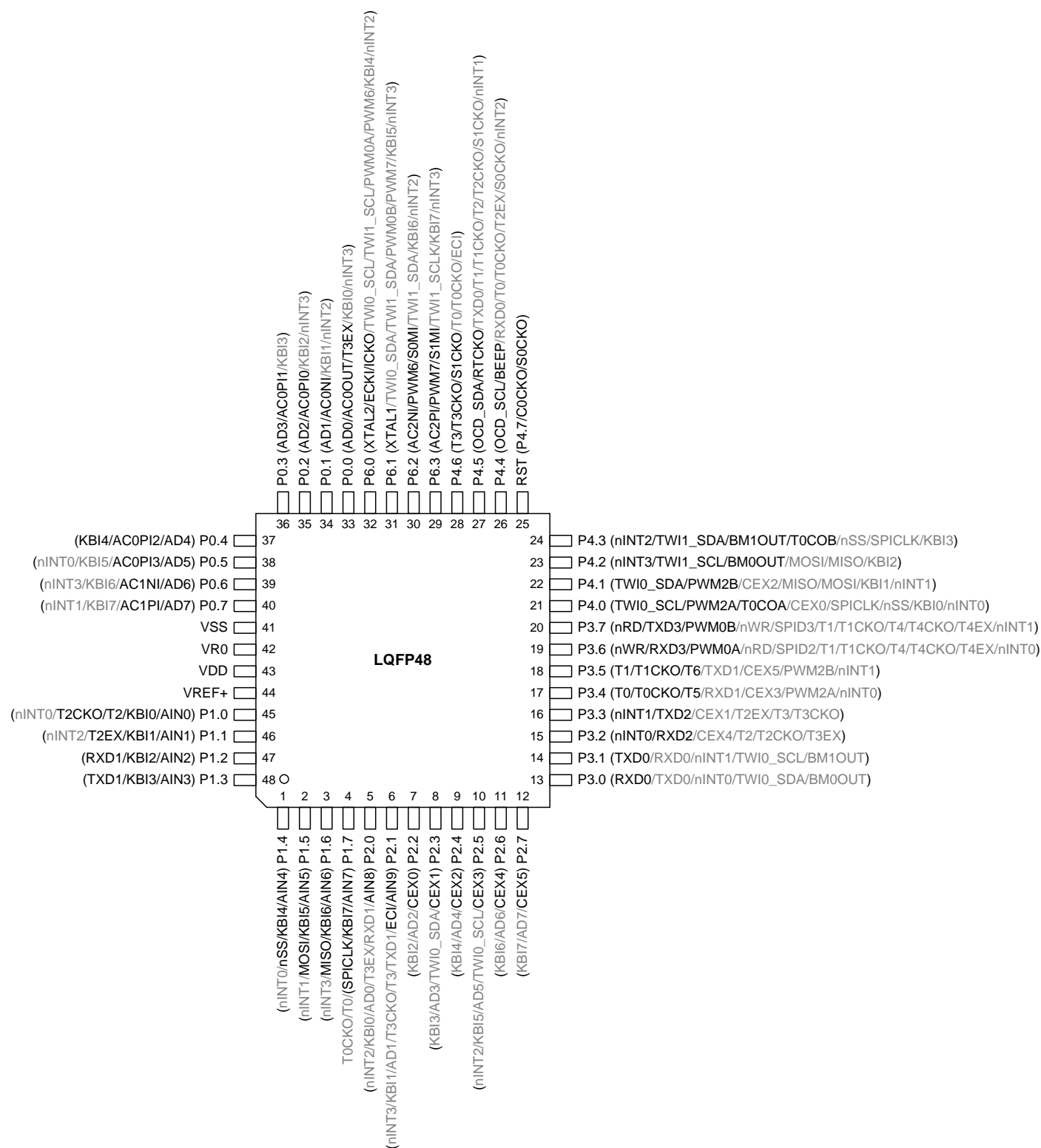
ISPCR = ISPEN;           //使能IAP/ISP
IFMT = MS2;             // P 页写, IFMT =0x04
IFADRH = 0x00;
IFADRL = SPCON0;       //设置P页 SFR地址
IFD |= CKCTL0;         //设置CKCTL0
SCMD = 0x46;           //
SCMD = 0xB9;           //
IFMT = Flash_Standby; // IAP/ISP等待, IFMT =0x00
ISPCR &= ~ISPEN;

```

## 4. 引脚结构

### 4.1. 封装指南

图 4-1. MGEQ1C064AD48 LQFP48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
	48-脚 LQFP		
<b>P0.0</b> (AD0) (AC0OUT) (T3EX)	33	I/O	* 端口 0.0. * AD0: EMB 复用 A0/D0. * AC0OUT: 模拟比较器 0 输出. * T3EX: 定时/计数器 3 外部控制输入
<b>P0.1</b> (AD1) (AC0NI)	34	I/O	* 端口 0.1. * AD1: EMB 复用 A1/D1. * AC0NI: 模拟比较器 0 负端输入
<b>P0.2</b> (AD2) (AC0PI0)	35	I/O	* 端口 0.2. * AD2: EMB 复用 A2/D2. * AC0PI0: 模拟比较器 0 正端输入通道 0
<b>P0.3</b> (AD3) (AC0PI1)	36	I/O	* 端口 0.3. * AD3: EMB 复用 A3/D3 * AC0PI1: 模拟比较器 0 正端输入通道 1.
<b>P0.4</b> (AD4) (AC0PI2)	37	I/O	* 端口 0.4. * AD4: EMB 复用 A4/D4 * AC0PI2: 模拟比较器 0 正端输入通道 2.
<b>P0.5</b> (AD5) (AC0PI3)	38	I/O	* 端口 0.5. * AD5: EMB 复用 A5/D5 * AC0PI3: 模拟比较器 0 正端输入通道 3.
<b>P0.6</b> (AD6) (AC1NI)	39	I/O	* 端口 0.6. * AD6: EMB 复用 A6/D6. * AC1NI: 模拟比较器 1 负端输入
<b>P0.7</b> (AD7) (AC1PI)	40	I/O	* 端口 0.7. * AD7: EMB 复用 A7/D7. * AC1PI: 模拟比较器 0 正端输入
<b>P1.0</b> (AIN0) (KBI0) (T2) (T2CKO)	45	I/O	* 端口 1.0. * AIN0: ADC 通道 0 模拟输入 * 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.
<b>P1.1</b> (AIN1) (KBI1) (T2EX)	46	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
<b>P1.2</b> (AIN2) (KBI2) (RXD1)	47	I/O	* 端口 1.2. * AIN0: ADC 通道 2 模拟输入 * KBI2: 键盘输入 2. * RXD1: UART1 串行输入.
<b>P1.3</b> (AIN3) (KBI3) (TXD1)	48	I/O	* 端口 1.3. * AIN3: ADC 通道 3 模拟输入 * KBI3: 键盘输入 3. * TXD1: UART1 串行输出
<b>P1.4</b> (AIN4) (KBI4) (nSS)	1	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.
<b>P1.5</b> (AIN5) (KBI5) (MOSI)	2	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, QPI 模式下数据 0 输入输出
<b>P1.6</b> (AIN6) (KBI6) (MISO)	3	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, QPI 模式下数据 1 输入输出
<b>P1.7</b> (AIN7) (KBI7) (SPICLK)	4	I/O	* 端口 1.7. * AIN7: ADC 通道 7 模拟输入 * KBI7: 键盘输入 7. * SPICLK: SPI 时钟, 主机输出, 从机输入
<b>P2.0</b> (AIN8)	5	I/O	* 端口 2.0. * AIN8: ADC 通道 8 模拟输入
<b>P2.1</b> (AIN9) (ECI)	6	I/O	* 端口 2.1. * AIN9: ADC 通道 9 模拟输入 * ECI: PCA 外部时钟输入
<b>P2.2</b> (CEX0)	7	I/O	* 端口 2.2. * CEX0: PCA0 模块 0 外部 I/O.

助记符	引脚号	I/O 类型	描述
	48-脚 LQFP		
P2.3 (CEX1)	8	I/O	* 端口 2.3. * CEX1: PCA0 模块 1 外部 I/O.
P2.4 (CEX2)	9	I/O	* 端口 2.4. * CEX2: PCA0 模块 2 外部 I/O.
P2.5 (CEX3)	10	I/O	* 端口 2.5. * CEX3: PCA0 模块 3 外部 I/O.
P2.6 (CEX4)	11	I/O	* 端口 2.6. * CEX4: PCA0 模块 4 外部 I/O.
P2.7 (CEX5)	12	I/O	* 端口 2.7. * CEX5: PCA0 模块 5 外部 I/O.
P3.0 (RXD0)	13	I/O	* 端口 3.0. * RXD0: UART0 串行输入
P3.1 (TXD0)	14	I/O	* 端口 3.1. * TXD0: UART0 串行输出.
P3.2 (nINT0) (RXD2)	15	I/O	* 端口 3.2. * nINT0: 外部中断 0 输入 * RXD2: UART2 串行输入.(MG82F6D32 不支持)
P3.3 (nINT1) (TXD2)	16	I/O	* 端口 3.3. * nINT1: 外部中断 1 输入. * TXD2: UART2 串行输出.(MG82F6D32 不支持)
P3.4 (T0) (T0CKO) (T5)	17	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 * T5: 定时/计数器 5 外部时钟输入 t
P3.5 (T1) (T1CKO) (T6)	18	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.
P3.6 (nWR) (RXD3) (PWM0A)	19	I/O	* 端口 3.6. * nWR: EMB 写信号. * RXD3: UART3 串行输入.(MG82F6D32 不支持) * PWM0A: PCA PWM0 输出副通道 A.
P3.7 (nRD) (TXD3) (PWM0B)	20	I/O	* 端口 3.7. * nRD: EMB 读信号. * TXD3: UART3 串行输出.(MG82F6D32 不支持) * PWM0B: PCA PWM0 输出副通道 B.
P4.0 (TWI0_SCL) (PWM2A) (TOCOA)	21	I/O	* 端口 4.0. * TWI0_SCL: TWI0/I2C0 串行时钟 * PWM2A: PCA PWM2 输出副通道 A. * TOCOA: T0CKO 输出副通道 A.
P4.1 (TWI0_SDA) (PWM2B)	22	I/O	* 端口 4.1. * TWI0_SDA: TWI0/I2C0 串行数据 * PWM2B: PCA PWM2 输出副通道 B.
P4.2 (nINT3) (TWI1_SCL) (BM0OUT)	23	I/O	* 端口 4.2. * nINT3: 外部中断 3 输入 * TWI1_SCL: TWI1/I2C1 串行时钟. * BM0OUT: OBM0 输出
P4.3 (nINT2) (TWI1_SDA) (BM1OUT) (TOCOB)	24	I/O	* 端口 4.3. * nINT2: 外部中断 2 输入. * TWI1_SDA: TWI1/I2C1 串行数据. * BM1OUT: OBM1 输出. * TOCOB: T0CKO 输出副通道 B.
P4.4 (OCD_SCL) (BEEP)	26	I/O	* 端口 4.4. * OCD_SCL: OCD 接口, 时钟 * BEEP: 蜂鸣器输出.
P4.5 (OCD_SDA) (RTCKO)	27	I/O	* 端口 4.5. * OCD_SDA: OCD 接口, 数据 * RTCKO: RTC 可编程时钟输出
P4.6 (T3) (T3CKO) (S1CKO)	28	I/O	* 端口 4.6. * 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 * S1CKO: S1BRG 可编程时钟输出
P6.0 (XTAL2) (ECKI)	32	I/O I O	* 端口 6.0. * XTAL2: 振荡电路输出 * ECKI: 外部时钟输入引脚

助记符	引脚号	I/O 类型	描述
	48-脚 LQFP		
(ICKO)			* ICKO: 内部时钟 (MCK) 输出.
<b>P6.1</b> (XTAL1)	31	I/O	* 端口 6.1. * XTAL1: 振荡电路输入.
<b>P6.2</b> (AC2NI) (PWM6) (S0MI)	30	I/O	* 端口 6.2. * AC2NI: 模拟比较器 2 负端输入. (MG82F6D32 不支持) * PWM6: PCA0 模块 6 PWM6 输出. * S0MI: 串口 0 SPI 主机模式数据输入
<b>P6.3</b> (AC2PI) (PWM7) (S1MI)	29	I/O	* 端口 6.3. * AC2PI: 模拟比较器 2 正端输入(MG82F6D32 不支持) * PWM7: PCA0 模块 7 PWM7 输出. * S1MI: 串口 1 SPI 主机模式数据输入.
<b>RST</b> (P4.7) (C0CKO) (S0CKO)	25	I O	* RST: 外部复位输入, 高有效 * 端口 4.7. <i>注意: 当 P4.7/RST 用作普通 I/O 引脚时, 不建议将其配置为输入模式, 以免在上电启动期间, 若该引脚输入高电平, 会导致单片机被锁定在复位状态。</i> * C0CKO: PCA 基准计数器可编程时钟输出 * S0CKO: S0BRT 可编程时钟输出
<b>VR0</b>	42	I/O	* VR0. 电压参考 0. 连接 0.1uF 和 4.7uF 到 VSS.
<b>VDD</b>	43	P	电源输入
<b>VREF+</b>	44	P	ADC VREF+外部输入
<b>VSS</b>	41	G	地. 0 V 电压参考

### 4.3. 功能复用

Many I/O pins, in addition to their normal I/O function, also serve the alternate function for internal peripherals. For the digital peripherals, all GPIOs serve the alternate function in the default state. However, the user may set the corresponding control bits in AXUR0~AUXR3 to serve their alternate function on the relocated ports.

#### 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
P60OC1	P60OC0	P60FD	PBKF	BM1F	BM0F	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 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
0 0	P60	由P6M0.0决定
0 1	MCK	由P6M0.0决定
1 0	MCK/2	由P6M0.0决定
1 1	MCK/4	由P6M0.0决定

了解详情, 请参考“9. 系统时钟”。P6.0 作为时钟输出功能时, 建议设置P6M0.0为“1”来选择 P6.0为推挽输出模式。

Bit 5: 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。

#### 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
0	0	CRCDS1	CRCDS0	0	0	0	DPS
W	W	R/W	R/W	W	W	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	0	COPLK	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	W	W	R/W	R/W	R/W	R/W

#### 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 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 (UART0) 端口引脚选择位。(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	P7.0	P7.1

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	P2.5	P2.3

#### 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端口引脚选择位[1:0].

T2PS1~0	T2/T2CKO	T2EX
0 0	P1.0	P1.1
0 1	P3.2	P3.3
1 0	P6.7	P5.7
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	P3.6
1 1	P3.7

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P7.2.

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 P0.0.

**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
C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	C0PS1	C0PS0	ECIPS0	C0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: C0IC4S0, PCA0输入通道4输入选择

C0IC4S0	CEX4 input
0	CEX4 端口 Pin
1	T2EXI

Bit 6: C0IC2S0, PCA0输入通道2输入选择

C1IC2S0	CEX2 input
0	CEX2 端口 Pin
1	T3EXI

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

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

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

C0PPS0	PWM0A	PWM0B
0	P3.6	P3.7
1	P6.0	P6.1

Bit 3: C0PS1, PCA0引脚选择位1.

C0PS1	CEX1	CEX3	CEX5
0	P2.3	P2.5	P2.7
1	P3.3	P3.4	P3.5

Bit 2: C0PS0, PCA0引脚选择位 0.

C0PS0	CEX0	CEX2	CEX4
0	P2.2	P2.4	P2.6
1	P4.0	P4.1	P3.2

Bit 1: ECIPS0, PCA0 ECI 引脚选择位 0.

ECIPS0	ECI
0	P2.1
1	P4.6

Bit 0: C0COPS, PCA0时钟输出(C0CKO) 引脚选择

C0COPS	C0CKO
0	P4.7
1	P7.2

**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	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	P0.7	P0.6	P0.5	P0.4
1 0	P2.7	P2.6	P2.5	P2.4
1 1	P6.3	P6.2	P6.1	P6.0

Bit 5~4: KBILPS1~0, KBI0~3 引脚选择位0.

KBILPS1~0	KBI3	KBI2	KBI1	KBI0
0 0	P1.3	P1.2	P1.1	P1.0
0 1	P0.3	P0.2	P0.1	P0.0
1 0	P2.3	P2.2	P2.1	P2.0
1 1	P4.3	P4.2	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 &amp; S3MI 引脚选择位

SnMIPS	S0MI	S1MI	S2MI	S3MI
0	P6.2	P6.3	P7.0	P7.1
1	P5.7	P6.7	P6.6	P6.5

Bit 0: S0COPS, S0BRG 时钟输出(S0CKO) 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 = 1100-0000

7	6	5	4	3	2	1	0
POE5	POE4	C0CKOE	SPI0M0	BM1OE1	BM1OE0	BM0OE1	BM0OE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: POE5, PCA0 PWM5主通道(PWM5O)输出控制。

0: 禁止PWM5O在端口引脚上输出。

1: 使能PWM5O在端口引脚上输出。默认是使能的。

Bit 6: POE4, PCA0 PWM4主通道(PWM4O)输出控制。

0: 禁止PWM4O在端口引脚上输出。

1: 使能PWM4O在端口引脚上输出。默认是使能的。

Bit 5: C0CKOE, PCA0时钟输出使能。

0: 禁止PCA0时钟输出。

1: PCA0基准定时器溢出率的二分之一时钟输出使能。

Bit 3~2: BM1OE1~0, BM1OUT 输出引脚选择位.

BM1OE1~0	BM1OUT 引脚
0 0	禁止
0 1	P4.3
1 0	P3.1
1 1	P7.2

Bit 1~0: BM0OE1~0, BM0OUT 输出引脚选择位.

BM0OE1~0	BM0OUT 引脚
0 0	禁止
0 1	P4.2
1 0	P3.0
1 1	P5.0

### AUXR8: 辅助寄存器 8

SFR 页 = 仅 5 页

SFR 地址 = 0xA4

复位值 = 1100-0000

7	6	5	4	3	2	1	0
POE7	POE6	C0PPS2	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	W	W	R/W	R/W	R/W

Bit 7: POE7, PCA0 PWM7主通道(PWM7O)输出控制。

0: 禁止PWM7O在端口引脚上输出。

1: 使能PWM7O在端口引脚上输出。默认是使能的。

Bit 6: POE6, PCA0 PWM6主通道(PWM6O)输出控制。

0: 禁止PWM6O在端口引脚上输出。

1: 使能PWM6O在端口引脚上输出。默认是使能的。

Bit 5: C0PPS2, {PWM6, PWM7} 引脚选择位 2.

C0PPS2	PWM6	PWM7
0	P6.2	P6.3
1	P6.0	P6.1

Bit 4~3: 保留位, 写寄存器时, 此位必须写“0”。

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

S1COPS	S1CKO
0	P4.6
1	P4.5

Bit 1~0: T3PS1~0, 定时器3 引脚选择位 [1:0].

T3PS1~0	T3/T3CKO	T3EX
0 0	P4.6	P0.0
0 1	P3.3	P3.2
1 0	P2.1	P2.0
1 1	P6.6	P6.5

### 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	C0FDC1	C0FDC0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: SID/STWI 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 引脚选择位 [1:0].

S1PS1~0	RXD1	TXD1
0 0	P1.2	P1.3
0 1	P7.1	P7.0
1 0	P2.0	P2.1
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 = 1100-0000

7	6	5	4	3	2	1	0
AC1HC0	AC0HC0	SPIPS1	SPIPS0	S0PS1	SPFACE	TWICF	PAA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5~4: SPIPS1~0, SPI 引脚选择位 [1:0].

引脚选择	SPIPS1~0	nSS	MOSI (SPID0)	MISO (SPID1)	SPICLK	SPID2	SPID3
0	0 0	P1.4	P1.5	P1.6	P1.7	P5.3	P5.4
1	0 1	P4.3	P4.2	P4.1	P4.0	P3.6	P3.7
2	1 0	P4.0	P4.1	P4.2	P4.3	P6.5	P5.7
3	1 1	P4.3	P4.1	P4.2	P4.0	P6.5	P5.7

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

Bit 1: TWICF, TWI0/I2C0 串行时钟输入滤波

0: 禁止TWICF 功能.

1: 使能TWICF 功能.

Bit 0: PAA, TWI0/I2C0 预回应.

0: 禁止PAA 功能.

1: 在TWI0/I2C0 主机RX和从机TX/RX 使用DMA传送时使能PAA功能

#### AUXR11: 辅助寄存器 11

SFR 页 = 仅 8 个

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
I2C1CF	PAA1	I2C1PS1	I2C1PS0	--	POEM0	COM0	COOFS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

I2C1PS1~0	TWI1_SCL	TWI1_SDA
0 0	P4.2	P4.3
0 1	P6.0	P6.1
1 0	P6.5	P6.6
1 1	P6.3	P6.2

#### AUXR12: 辅助寄存器 12

SFR 页 = 仅 9个

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
--	--	T0CKOB	T0CKOA	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: T0CKOB(=T0OR) 输出使能.

0: 禁止T0CKOB 输出.

1: 使能T0CKOB 在P4.3上输出.

Bit 4: T0CKOA(=T0OR) 输出使能.

0: 禁止T0CKOA 输出.

1: 使能T0CKOA 在P4.0上输出.

**AUXR13: 辅助寄存器 13**

SFR 页 = 仅 A 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	S3PS0	0	S2PS0	0	T4FCS	T4PS1	T4PS0
W	R/W	W	R/W	W	R/W	R/W	R/W

Bit 6: S3PS0, 串口 3 引脚选择位 0.

S3PS0	RXD3	TXD3
0	P3.6	P3.7
1	P6.5	P6.6

Bit 4: S2PS0, 串口 2 引脚选择位 0.

S2PS0	RXD2	TXD2
0	P3.2	P3.3
1	P6.7	P5.7

Bit 2: T4FCS, 保留用于芯片测试.

Bit 1~0: T4PS1~0, 定时器4 引脚选择位 [1:0].

T4PS1~0	T4/T4CKO	T4EX
0 0	P7.0	P7.1
0 1	P3.7	P3.6
1 0	P7.2	P5.0
1 1	P3.6	P3.7

**AUXR14: 辅助寄存器 14**

SFR 页 = 仅 B 页

SFR 地址 = 0xA4

复位值 = 0000-0100

7	6	5	4	3	2	1	0
ADLPS1	ADLPS0	0	0	RWPS0	AC2HC0	AC2OE	AC2FLT1
R/W	R/W	W	W	R/W	R/W	R/W	R/W

Bit 7~6: EMB AD总线引脚选择位(片外MOVX 时序).

ADLPS1~0	AD[7:0]
0 0	P0.7~P0.0
0 1	P2.7~P2.0
1 0	P7.2~P7.0, P5.7, P6.7~P6.4
1 1	保留

Bit 3: RWPS0, EMB nRD/nWR 引脚选择位

RWPS0	nWR	nRD
0	P3.6	P3.7
1	P3.7	P3.6

Bit 1: AC2OE, AC2OUT 在引脚上输出使能

0: 禁止AC2OUT 在引脚上输出

1: 使能AC2OUT 输出到P6.4.

**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	P3.7
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	P3.6
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	P4.3
0 0 1	P2.0
0 1 0	P1.1
0 1 1	P2.5
1 0 0	P4.4
1 0 1	P6.0
1 1 0	P0.1
1 1 1	P6.2

**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	P4.2
0 0 1	P2.1
0 1 0	P0.0
0 1 1	P1.6
1 0 0	P0.2
1 0 1	P6.1
1 1 0	P0.6
1 1 1	P6.3

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: Program Status Word

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 片内数据存储器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寄存器**

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**

**MGEQ1C064**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。操作码本身就进行了。

#### **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，MGEQ1C064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64K字节。在MGEQ1C064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。

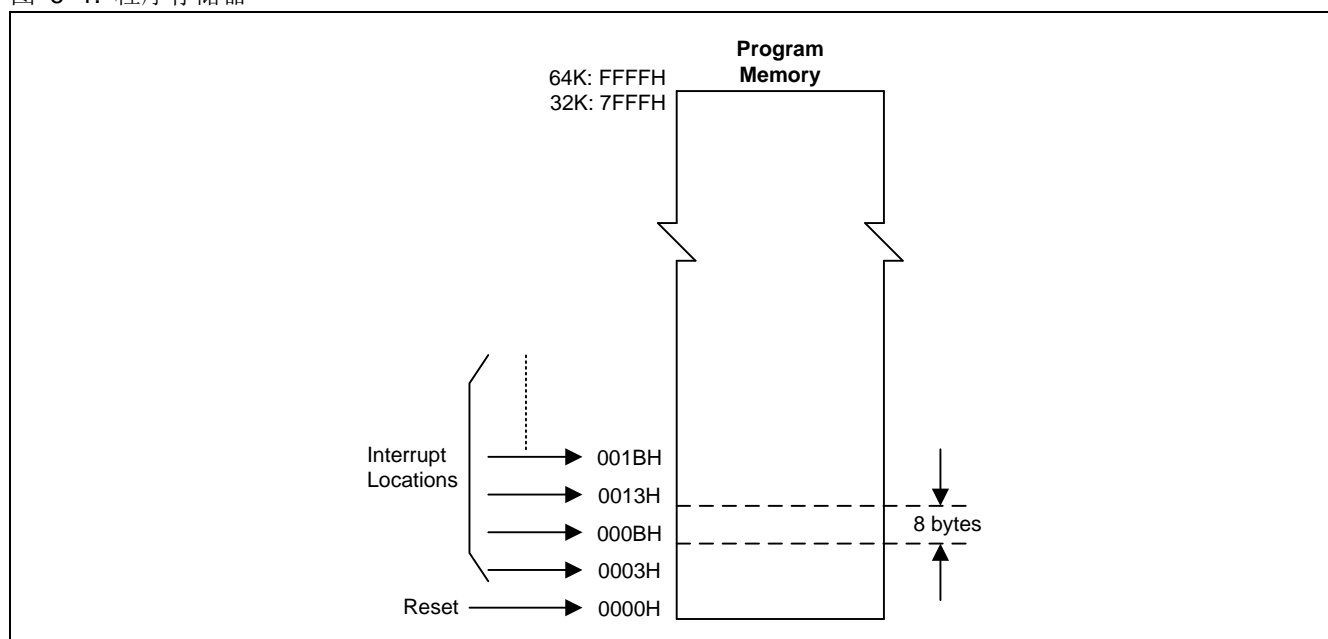
数据存储器使用与程序存储器不同的地址空间。MGEQ1C064有256字节的内部和3480字节的片上扩展存储器(XRAM)。

### 6.1. 片内程序存储器 (Flash)

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. 片内数据存储器 RAM

图 6-2 向 MGEQ1C064 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

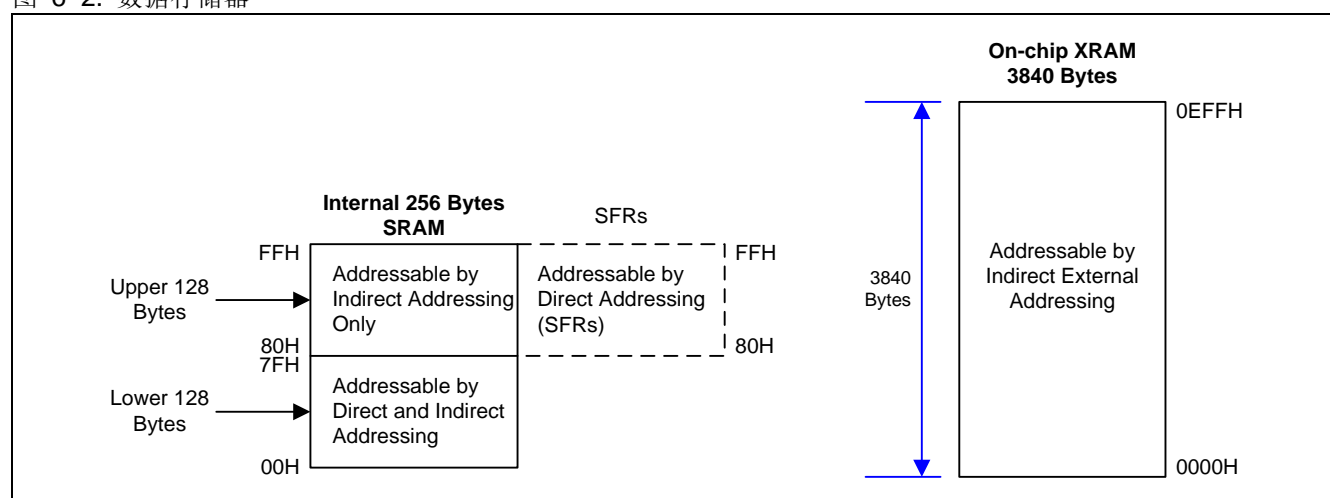


图 6-3. RAM内部RAM的低128字节

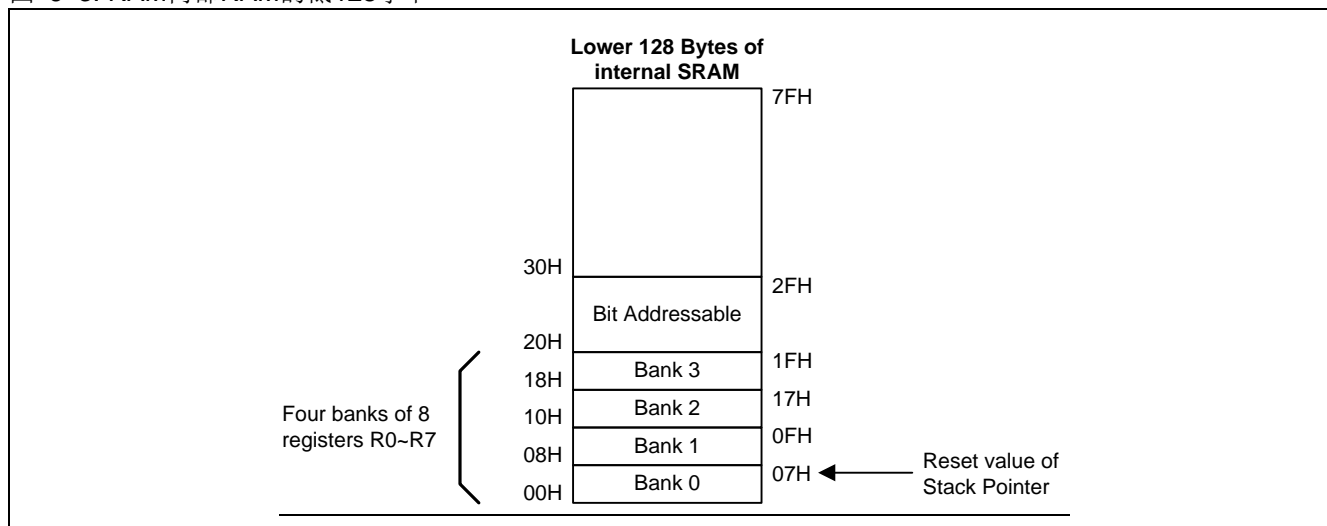
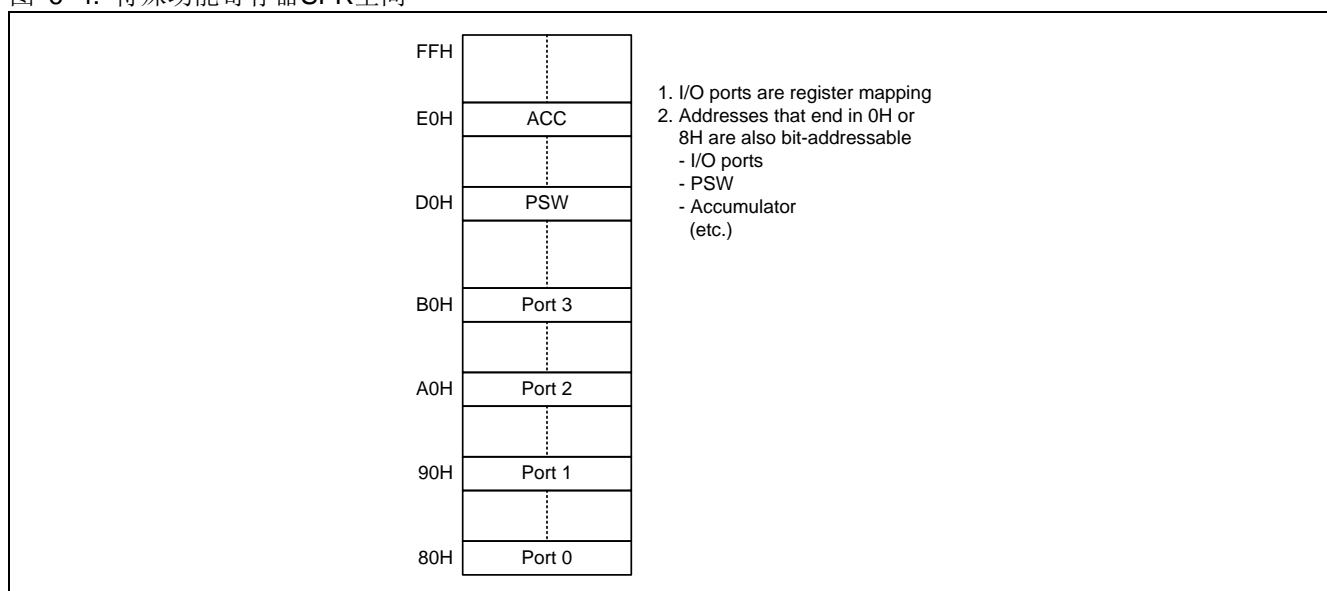


图 6-4. 特殊功能寄存器SFR空间



### 6.3. 片内扩展 RAM (XRAM)

访问片上扩展RAM(XRAM), 参考图 6-2, 这3840字节的XRAM(0000H到0EFFH)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MGEQ1C064硬件才能正确访问XRAM。

## 6.4. EMB,片外数据存储器总线

EMB功能是由CPU MOVX指令在XRAM区域大于3840字节的地址上实现的。3840字节的XRAM空间是片内XRAM区域。EMB是用来访问片外存储器或外设的并行总线。MGEQ1C064通过EMBTC设置支持对片外读写时序的扩展控制，确保在应用中正确访问。如果CPU通过MOVX指令访问EMB，则EMBTC时序基于CPUCLK。

MGEQ1C064只提供EMB无地址模式的读写访问。MGEQ1C064中的无地址EMB也支持DMA访问。如果EMB通过DMA访问，则EMBTC时序基于SYSCLK。当EMB由DMA传输操作时，不允许CPU访问EMB，CPU必须先停止或挂起DMA，然后等待12个SYSCLK，之后再通过MOVX指令访问EMB。

建议将EMB信号线设置为以下配置：

- 将nRD/nWR 设置为推挽输出模式。
- 将数据总线AD[7:0]设置为开漏输出模式或带上拉电阻的开漏输出模式。

下列寄存器控制EMB接口功能：EMBTC配置EMB总线时序，AUXR14配置选择用于EMB 接口功能的引脚

### EMBTC: EMB Timing控制寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0xB5

复位值 = 1000-0000

7	6	5	4	3	2	1	0
1	0	0	0	RWSH	RWS2	RWS1	RWS0
R	R	R	R	R/W	R/W	R/W	R/W

Bit 7~4: 外部数据访问接口模式，需要保持为1000B

Bit 3: RWSH, EMB数据总线的写入(nWR)脉冲的设置/保持时间控制

0: nWR命令的数据总线设置/保持时间= 1 SYSCLK/CPUCLK周期

1: nWR命令的数据总线设置/保持时间= 2 SYSCLK/CPUCLK周期

Bit 2~0: RWS[2:0], EMB 读/写命令脉宽选择位

000: nRD 和 nWR 脉宽 = 1 SYSCLK/CPUCLK 周期。

001: nRD 和 nWR 脉宽 = 2 SYSCLK/CPUCLK 周期。

010: nRD 和 nWR 脉宽 = 3 SYSCLK/CPUCLK 周期。

011: nRD 和 nWR 脉宽 = 4 SYSCLK/CPUCLK 周期。

100: nRD 和 nWR 脉宽 = 5 SYSCLK/CPUCLK 周期。

101: nRD 和 nWR 脉宽 = 6 SYSCLK/CPUCLK 周期。

110: nRD 和 nWR 脉宽 = 7 SYSCLK/CPUCLK 周期。

111: nRD 和 nWR 脉宽 = 8 SYSCLK/CPUCLK 周期。

### AUXR14: 辅助寄存器 14

SFR 页 = 仅 B 页

SFR 地址 = 0xA4

复位值 = 0000-0100

7	6	5	4	3	2	1	0
ADLPS1	ADLPS0	0	0	RWPS0	AC2HC0	AC2OE	AC2FLT1
R/W	R/W	W	W	R/W	R/W	R/W	R/W

Bit 7~6: EMB AD总线端口选择(片外MOVX 指令).

ADLPS1~0	AD[7:0]
0 0	P0.7~P0.0
0 1	P2.7~P2.0
1 0	P7.2~P7.0, P5.7, P6.7~P6.4
1 1	保留

Bit 3: RWPS0, EMB nRD/nWR 端口选择.

RWPS0	nWR	nRD
0	P3.6	P3.7
1	P3.7	P3.6

### 6.4.1. 无地址相位模式的 MOVX 8 位数据访问

在这模式下，当CPU/DMA访问外部存储器或外设时是无地址相位的。

EMB的这个模式支持CPU MOVX 指令和DMA指令的访问

图 6-5.无地址相位EMB 结构

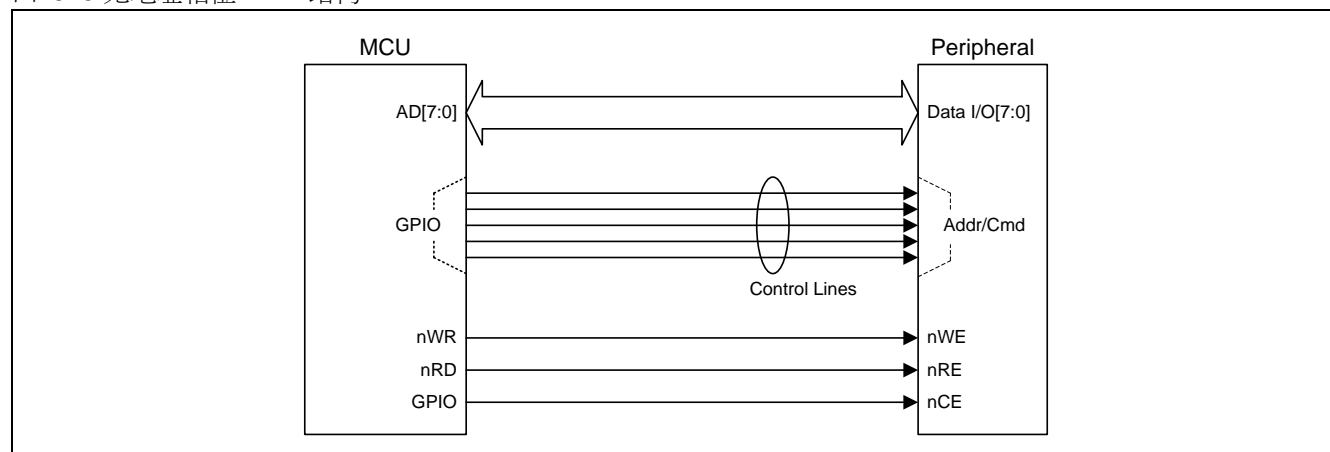
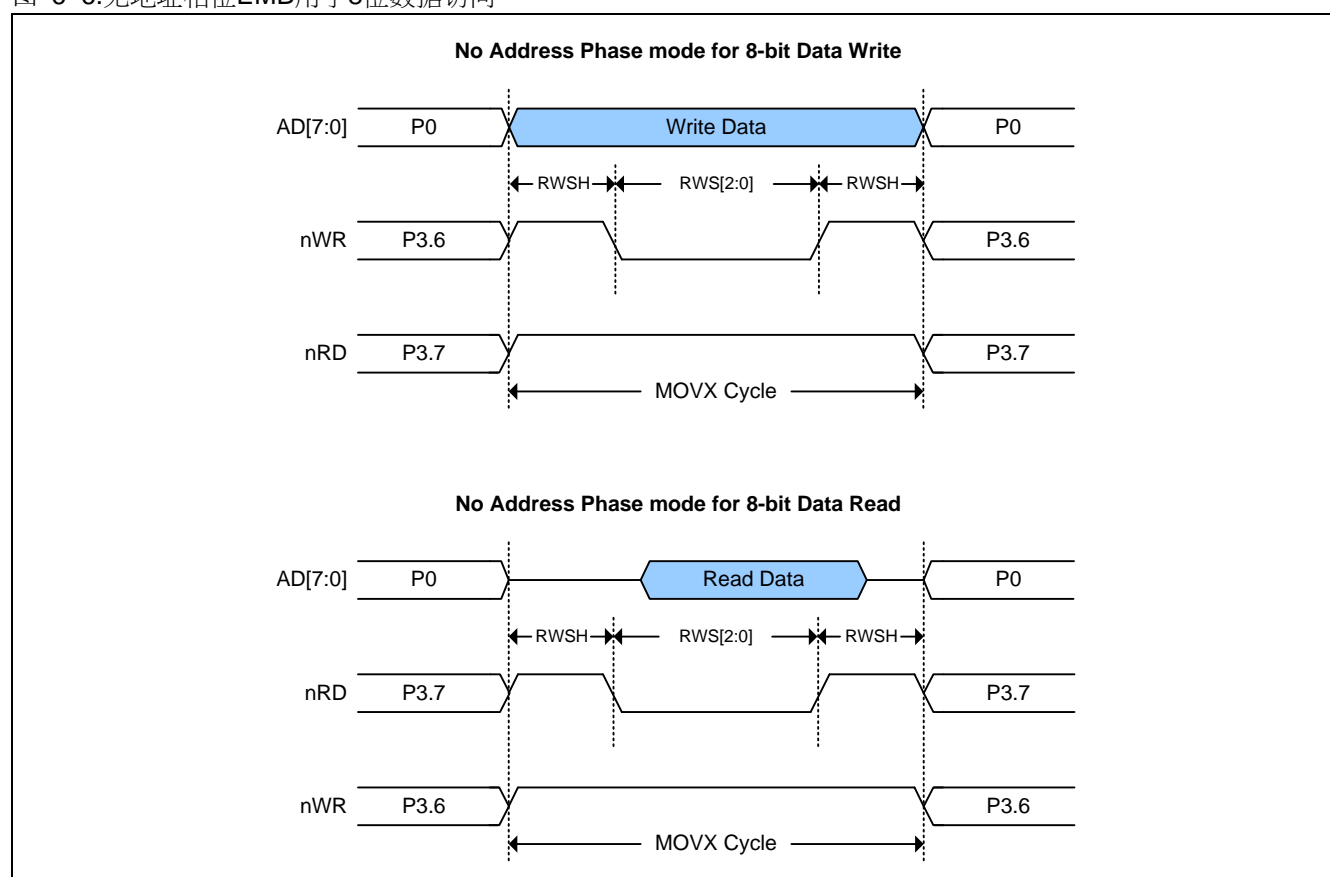


图 6-6.无地址相位EMB用于8位数据访问



## 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MGEQ1C064存储空间的对对应关系如下：

### ***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。MGEQ1C064有3840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。MGEQ1C064有256字节片上pdata存储器它与片上xdata存储器共享。

### ***code***

64K字节程序存储空间。通过“MOVC @A+DTPR”访问，作为程序部分被读取。MGEQ1C064有64K字节的片上程序存储器。

## 7. XRAM 访问

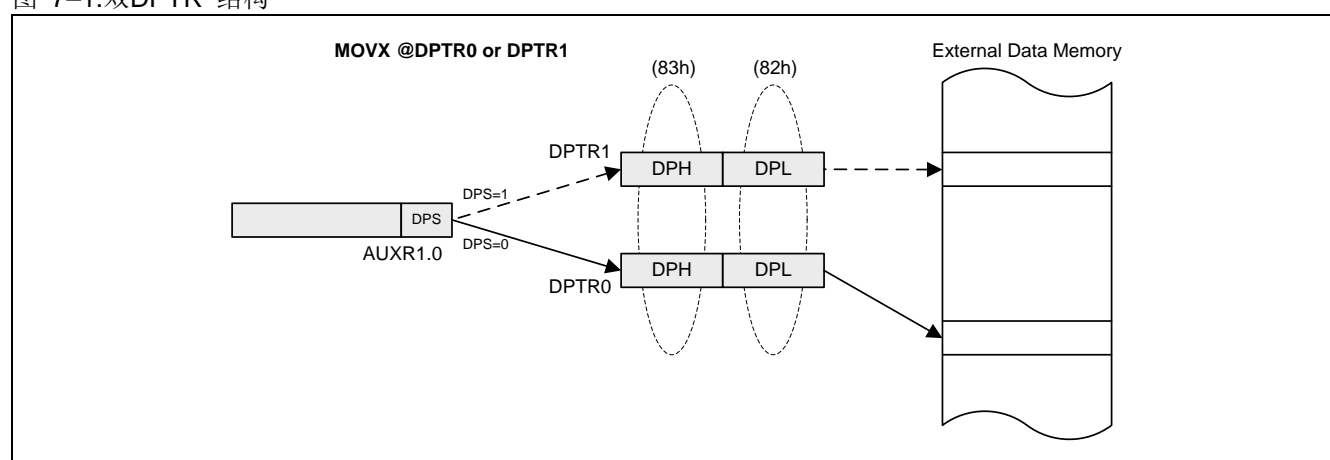
MGEQ1C064系列MCU内含有3840字节被映射到外部数据存储器的数据存储器(XRAM)。外部数据存储空间可以使用外部移动指令(MOVX)和数据指针(DPTR)访问,或使用(R0或R1)的MOVX间接访问模式。如果MOVX指令使用8位寻址操作(比如@R1),16位地址的高字节则由XRAM的页选择寄存器(XRPS)决定。

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	;移动外部RAM(16位地址)到ACC
MOVX @DPTR,A	;移动ACC到外部RAM(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
--	--	CRCDS1	CRCDS0	--	--	--	DPS
W	W	R/W	R/W	W	W	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	选择DPTR
0	DPTR0
1	DPTR1

**DPL: 数据指针低8位**

SFR 页 = 0~F

SFR 地址 = 0xA2

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 = 0xA3

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

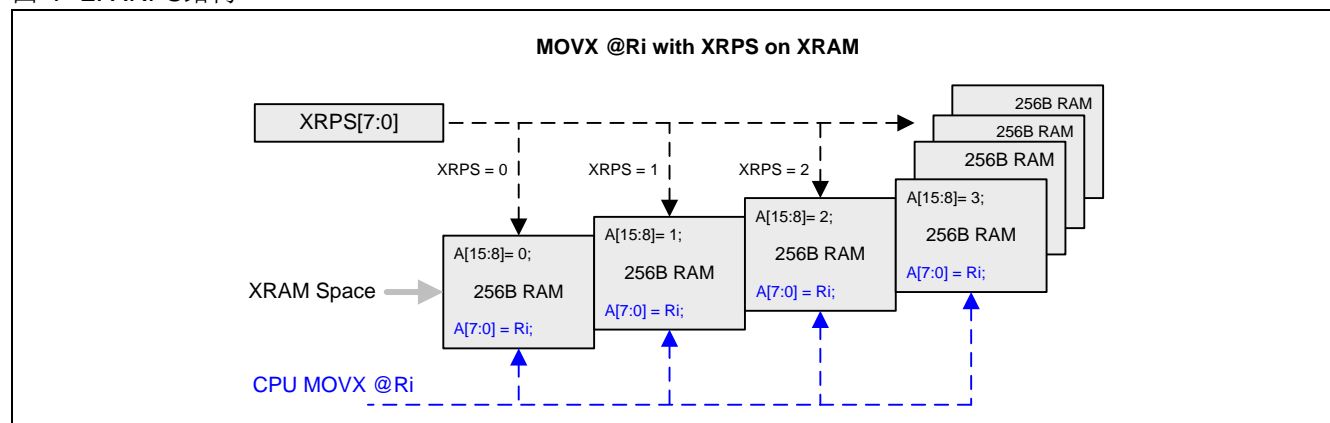
DPH寄存器是16位DPTR的高字节。DPTR用于间接访问XRAM和闪存(Flash)存储器的编址。

## 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 = XXXX-0000

7	6	5	4	3	2	1	0
--	--	--	--	XRPS.3	XRPS.2	XRPS.1	XRPS.0
W	W	W	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: 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MGEQ1C064中，XRPS索引15/7页的256字节页RAM。

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

## 8. 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在这段时间并行做其它操作以提高效能。

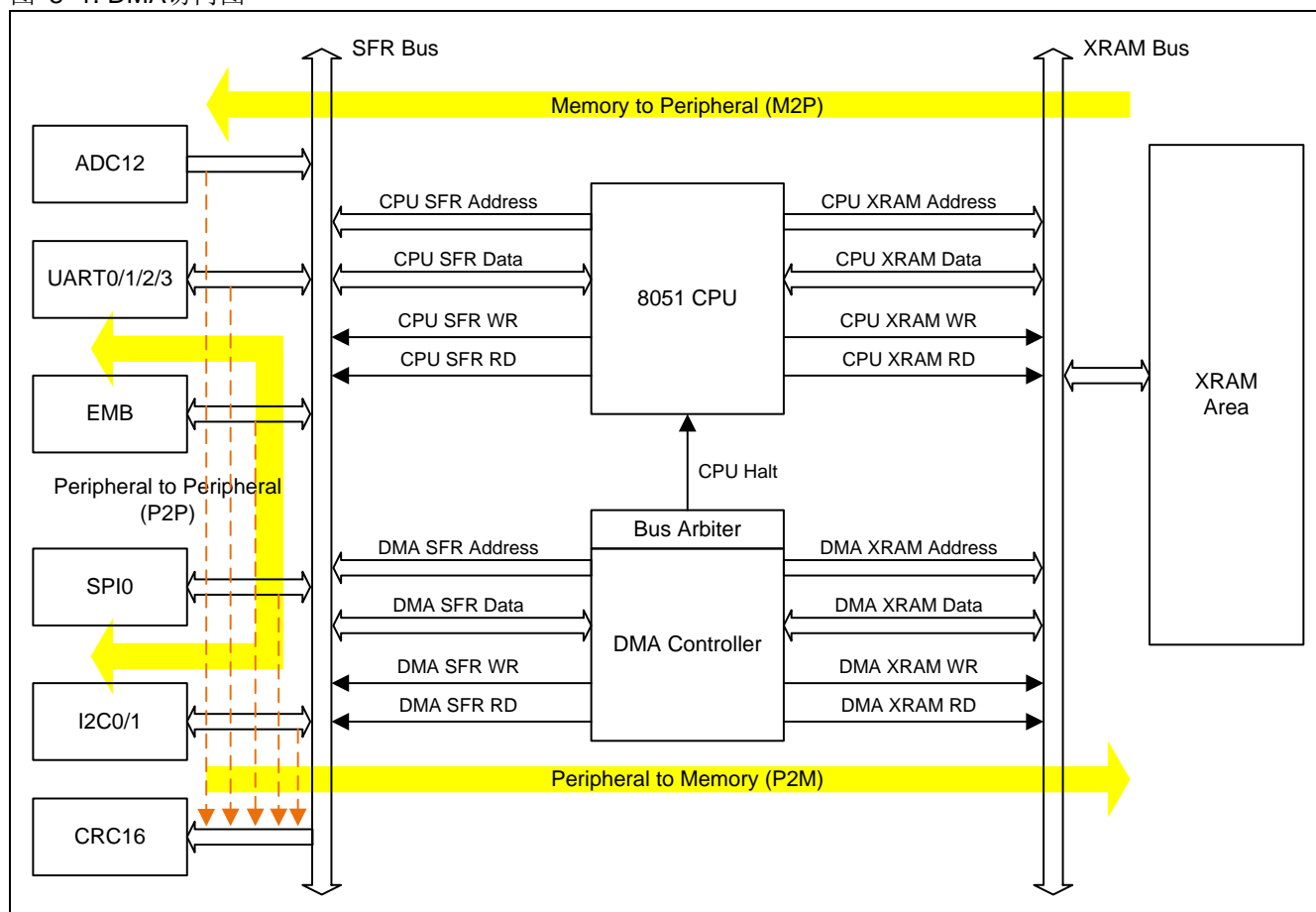
使用DMA能提升外设模块的效率。通过允许CPU在低功耗模式下不用被唤醒便可进行数据传输，以达到降低系统功耗的要求。

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访问图



## 8.1. DMA 结构

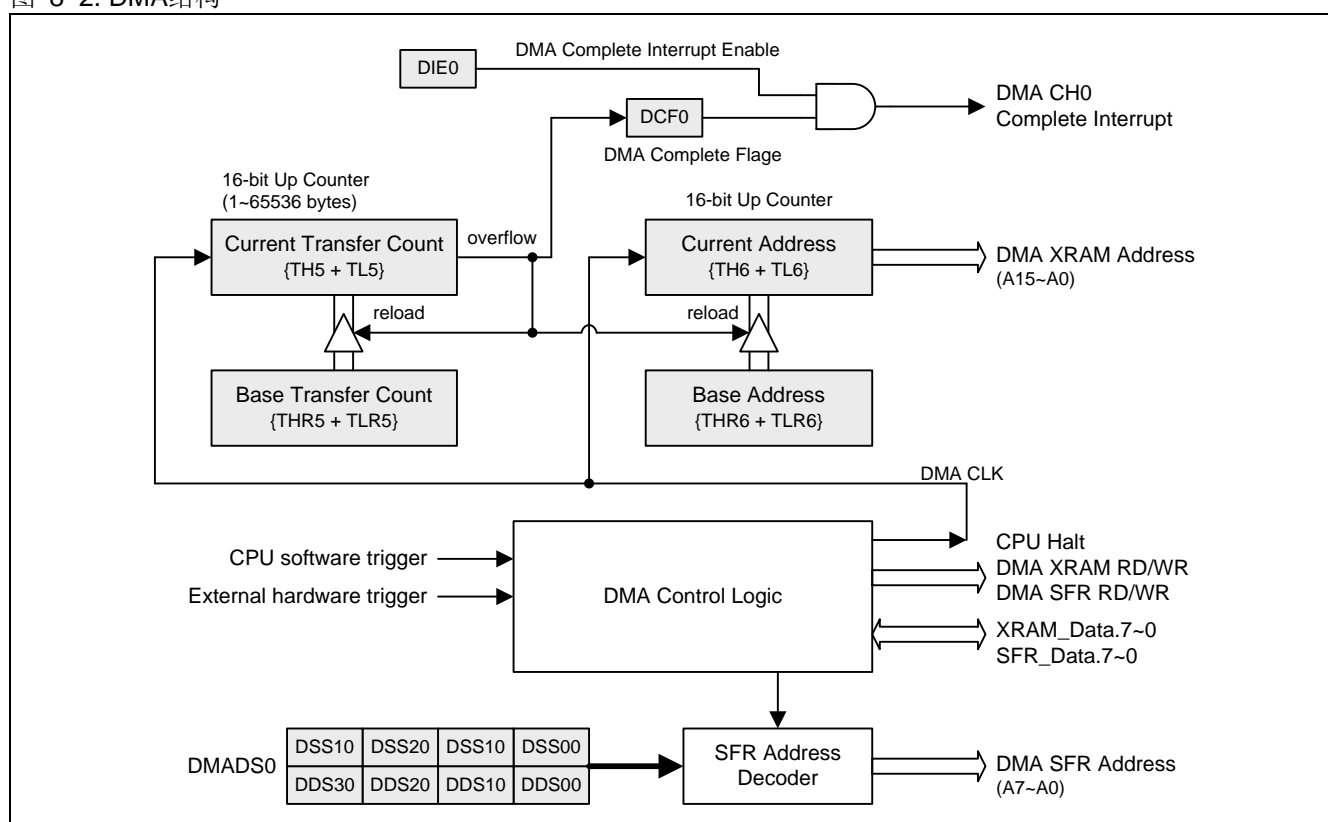
MGEQ1C064的DMA控制器提供一个通道DMA支持3种传输类型：传输数据从存储器(XRAM)到外设，从外设到存储器(XRAM)和从外设到外设。DMA通道的DMADS0寄存器定义DMA传输类型配置DMA控制器和在外设访问时定义数据路径产生SFR地址。

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

MGEQ1C064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])	目的选择
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	S2 RX	0 0 1 1	S2 TX
0 1 0 0	S3 RX	0 1 0 0	S3 TX
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 1	ADC0	1 0 0 1	保留
1 0 1 1	No Address EMB	1 0 1 1	No Address EMB
1 1 0 1	保留	1 1 0 1	CRC
1 1 1 1	XRAM	1 1 1 1	XRAM

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

### 8.2.2. DMA 传输模式

**MGEQ1C064**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**MGEQ1C064**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，

如果DMA大小为65536，{TH5 + TL5}将编程为0000H。

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

**MGEQ1C064**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通道被暂停，通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。

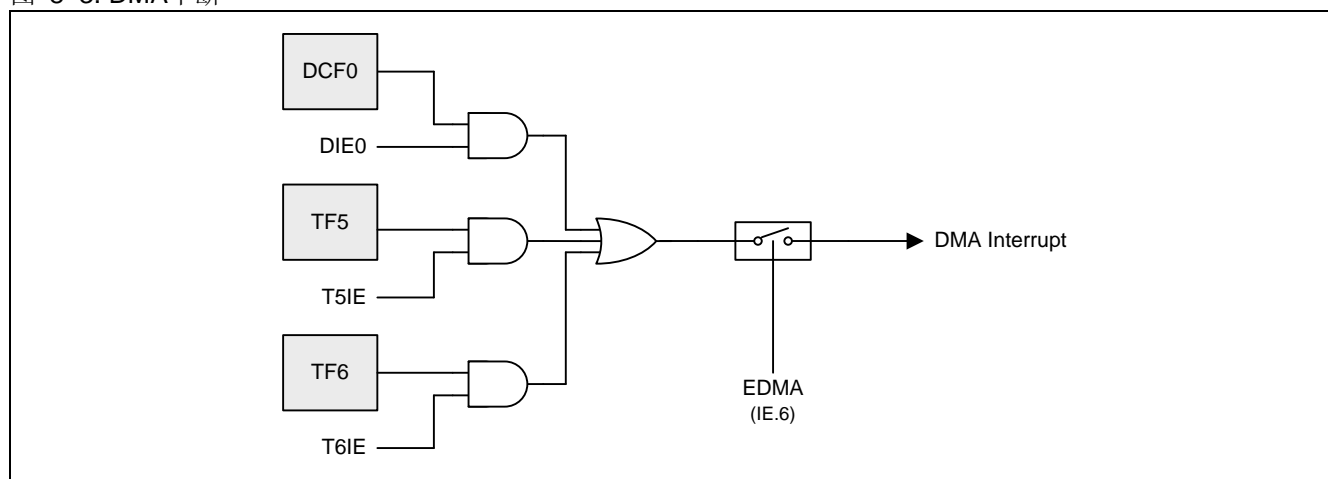
软件写“0”到DMAE0任何时候停止当前DMA传输或在DMA传输结束之后结束DMA传输。建议软件配置数据路径(DMADS0)禁止转态和清零DMAS0。

### 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。MGEQ1C064的XRAM边界是3840字节(0EFFHH)。
- 不支持S0, S1, S2和S3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。

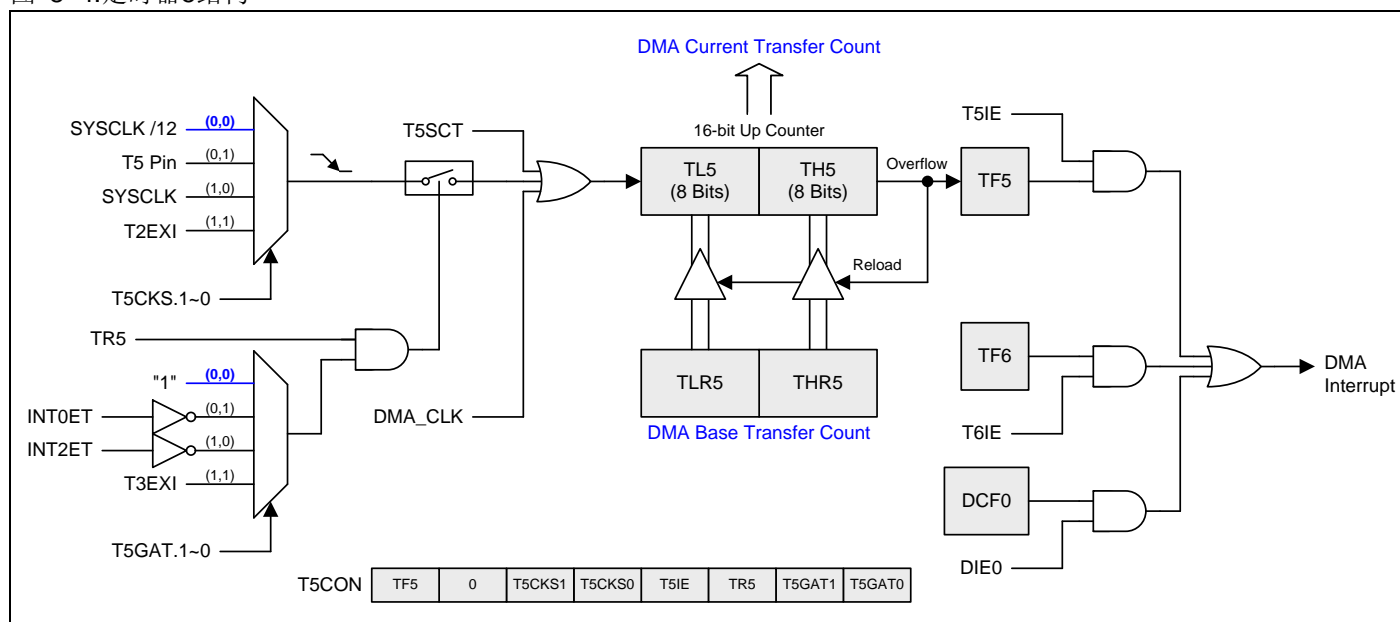
### 8.2.10. 定时器 5 & 定时器 6

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

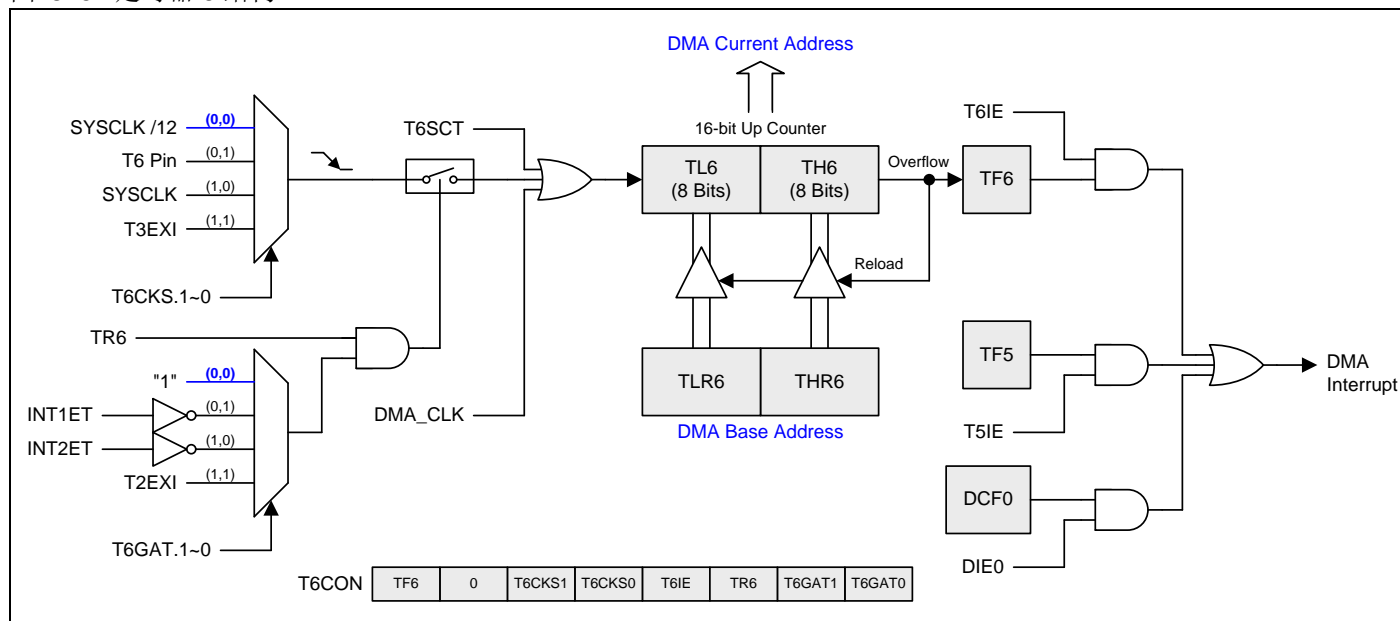


当DMA使能，定时器6是DMA传输地址指针功能。TH6和TL6是当前地址寄存器。THR6和TLR6是基地址寄存器。如果DMA禁止，定时器6是一个与定时器0有着一样门控制功能的16位自动重载定时/计数器。溢出标志TF6是一个中断源并共享DMA中断向量。下图展示了定时器6的结构。

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 = xxx0-0000

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

Bit 7~4: 保留位。软件在写入DMACR0时，必须向这些位写入"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	W	R/W	R/W	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: 保留位。软件在写入DMACG0时，必须向此位写入"0"。

Bit 3~2: EXTS10~00, DMA外部触发源选择。

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

Bit1: 保留位。软件写DMACG0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])	目的的选择
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	S2 RX	0 0 1 1	S2 TX
0 1 0 0	S3 RX	0 1 0 0	S3 TX
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 1	ADC0	1 0 0 1	保留
1 0 1 1	无地址 EMB	1 0 1 1	无地址 EMB
1 1 0 1	保留	1 1 0 1	CRC
1 1 1 1	XRAM	1 1 1 1	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 29.2.8 DMA

**8.4. 定时器 5 寄存器****T5CON: 定时器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	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: 保留位。软件在写入T5CON时, 必须向此位写入"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 引脚: 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。

Bit1~0: T5GAT.1~0, 定时器5的门控源选择

T5GAT.1~0	T5 门控选择
00	禁止
01	反向INT0ET
10	反向 INT2ET
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	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: 保留位。软件在写入T6CON时, 必须向此位写入"0"。

Bit 5~4: T6CKS.1~0, 定时器6时钟源选择

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

T6 引脚: 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 门控选择
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

#### TLR6: 定时器6低字节重载寄存器

SFR 页 = 仅 4 页

SFR 地址 = 0xCA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TLR6.7	TLR6.6	TLR6.5	TLR6.4	TLR6.3	TLR6.2	TLR6.1	TLR6.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

#### THR6: 定时器6高字节重载寄存器

SFR 页 = 仅 4 页

SFR 地址 = 0xCB

复位值 = 0000-0000

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MGEQ1C064系统时钟结构。

**MGEQ1C064**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。

内置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“37.4 IHRCO特性”。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“37.5 ILRCO特性”。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。

**MGEQ1C064**包含了一个时钟倍频器(CKM)产生高速时钟用于系统时钟源。**MGEQ1C064**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“37.6 CKM特性”。

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

MGEQ1C064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：

- 外部时钟输入模式：0 – 25MHz @ 2.4V – 5.5V
- CPU最大到25MHz @ 2.4V– 5.5V
- CPU最大到36MHz @ 2.7V -5.5V 使用片内倍频器(CKM)

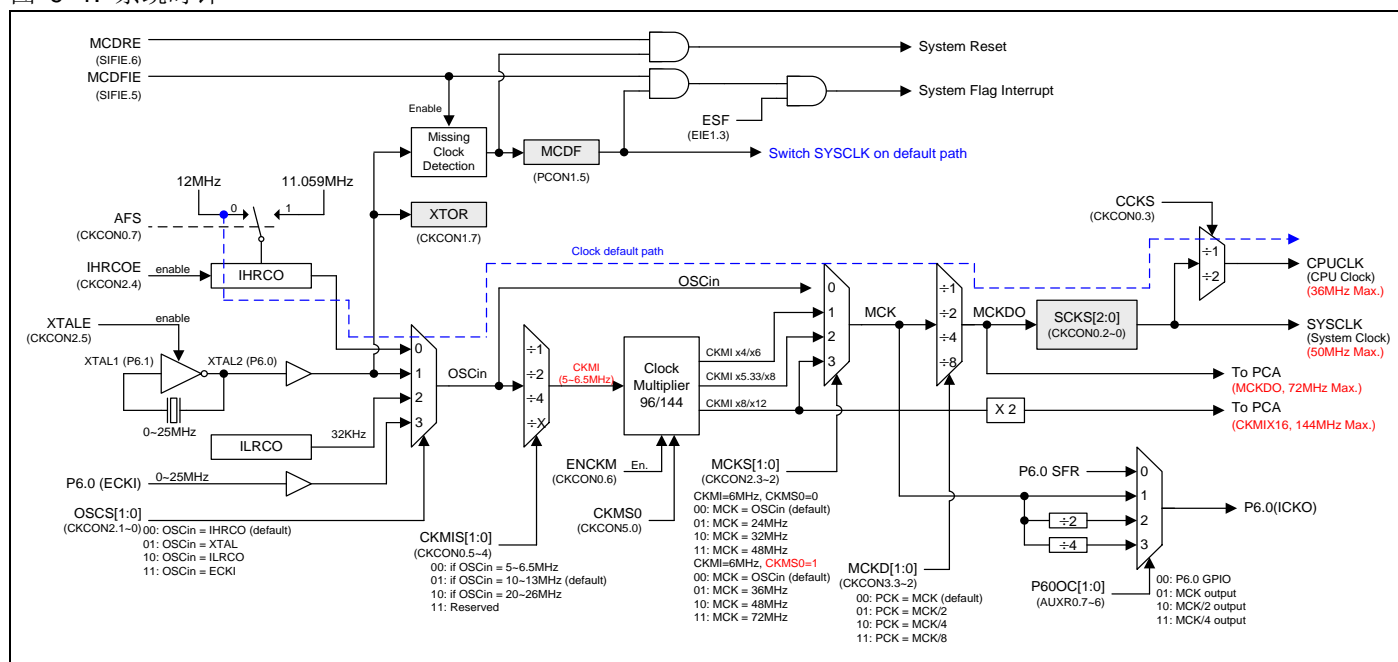
如果需要更高频的应用，当CPUCLK > 6MHz时HSE(DCON0.7)需要置位。此外，当CPUCLK>25MHz时HSE1也需要置位。

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36MHz)。

时钟模块也提供两个时钟源给高速PCA应用。

- MCKDO：最大可达72MHz
- CKMIX16：最大可达144MHz

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振荡器(IHRCO)，外部晶振，内部慢频RC振荡器(ILRCO)和外部时钟输入。如图 9-1所示MGEQ1C064系统时钟结构。MGEQ1C064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)

MGEQ1C064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“37.6 CKM特性”。

### 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MCDFIE启用。如果MCDFIE被清零，则时钟丢失监测模块处于非活动状态。MCDP( 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MCDP。一旦时钟丢失发生，软件必须在再次切换OSCin到XTAL之前清除MCDP。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MCDRE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IHRCO稳定时间是200us，要快于晶振。这就可用来在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 XTALE (CKCON2.5) 为“0”，禁止外部晶振电路
- MCU 进入掉电模式
- .....
- MCU 唤醒
- 设置 XTALE (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 &amp; 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/x12)。

0: 禁止X8/x12 时钟倍频器。

1: 使能X8/x12 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 = 0000-0000

7	6	5	4	3	2	1	0
XTOR	0	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 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]	
		CKMS0 = 0	CKMS0 = 1	CKMS0 = 0	CKMS0 = 1
0 0	OSCin	12MHz		11.059MHz	
0 1	CKMI x4 / x6	24MHz	36MHz	22.118MHz	33.177MHz
1 0	CKMI x5.33 / x8	32MHz	48MHz	29.491MHz	44.236MHz
1 1	CKMI x8 / x12	48MHz	72MHz	44.236MHz	66.354MHz

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 ≤ 36MHz, 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 频率	如MCK = 12MHz	如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

#### CKCON5: 时钟控制寄存器5

SFR 页 = 仅 P 页

SFR 地址 = 0x43

复位值 = 0000-0000

7	6	5	4	3	2	1	0
--	--	--	--	--	--	--	CKMS0
W	W	W	W	W	W	W	R/W

Bit 7~1: 保留位。软件在写入CKCON5时，必须向这些位写入"0"。

Bit 0: CKMS0, CKM模式选择0。

0: 选择CKM运行在16X模式。(96MHz)

1: 选择CKM运行在24X模式。(144MHz)

#### 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
P60OC1	P60OC0	P60FD	PBKF	BM1F	BM0F	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 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]	P6.0 功能	I/O 模式
00	P6.0	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

请参考章节“9 系统时钟”以获取更详细的时钟信息。对于 P6.0 引脚的时钟输出功能，建议将 {P6M1.0, P6M0.0} 设置为 01，即将 P6.0 配置为推挽输出模式。

Bit 5: 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 &amp; 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	W	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( $F_{CPUCLK} \leq 6\text{MHz}$ )这样减慢内部电路从而降低功耗。

1: 使能MCU全速运行( $F_{CPUCLK} > 6\text{MHz}$ )。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 5: HSE1, 超高速运行使能。

0: 无功能。

1: 使能MCU超高速运行。( $F_{CPUCLK} > 25\text{MHz}$ )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。MGEQ1C064的WDT结构框图如图 10-1所示。

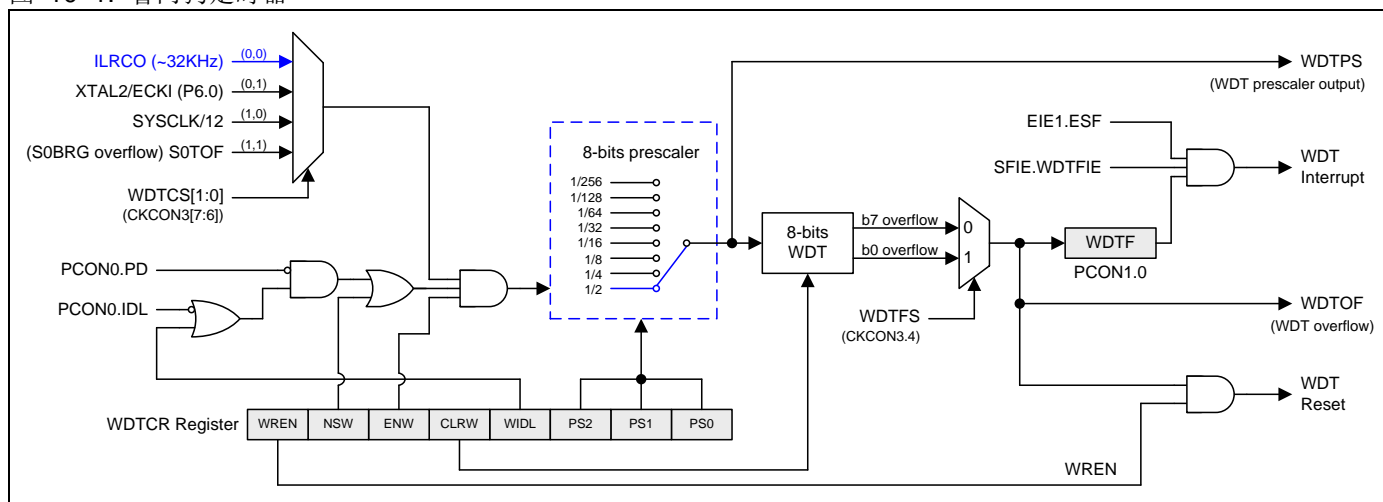
WDT时钟源有四种选择。必须在启用WDT之前配置时钟源。默认的WDT时钟源是32 KHz ILRCO。WDT溢出将设置WDTF (PCON1.0)，可以通过使能WDTFIE (SFIE .0)和ESF (EIE 1.3)来配置WDTF以产生中断。当置位WREN (WDTCR.7)时，溢出也会触发系统复位。为了防止WDT溢出，软件需要在WDT溢出之前向CLRW位(WDTCR.4)写入“1”来清零WDT。

使用 WDT 复位 (WREN = 1) 时，请在设置 WREN = 1 之前设置分频器。

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寄存器”和“33 P页访问”。

图 10-1. 看门狗定时器



### 10.2. 空闲模式的 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 &amp; P

SFR 地址 = 0xE1

POR = XXX0-XXXX (0000-0111)

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]	分频值	WDT 时间 (WDT时钟= ILRCO)	WDT 时间 (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]	分频值	WDT时间 (时钟源 = 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7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 &amp; 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 (CLR W) 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。

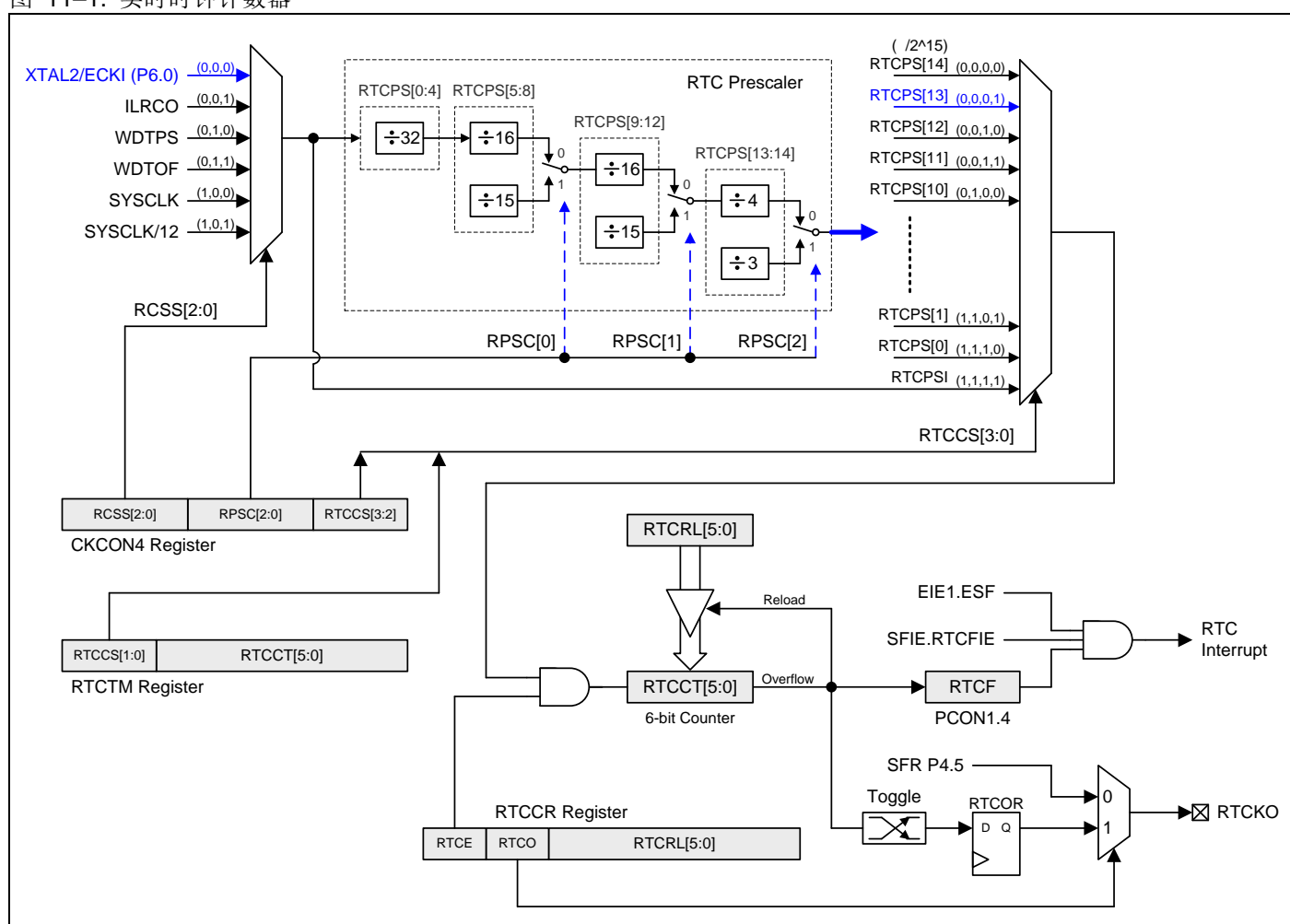
## 11. 实时时钟(RTC)/系统时钟

MGEQ1C064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MGEQ1C064的RTC结构。

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 $SYSCLK/2^{21}$ 。ILRCO为RTC模块提供内部时钟源。WDTPS和WDTOF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 = 0~7 &amp; P

SFR 地址 = 0xBE/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 = 0~F &amp; P

SFR 地址 = 0xB6/0x55

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	最小周期
0 0 0 0	RTCPS[14] ( $/2^{15}$ )	1S ~ 64S 当 P6.0 = 32768Hz	1S
0 0 0 1	RTCPS[13] ( $/2^{14}$ )	0.5S ~ 32S 当 P6.0 = 32768Hz	0.5S (默认)
0 0 1 0	RTCPS[13] ( $/2^{13}$ )	0.25S ~ 16S 当 P6.0 = 32768Hz	0.25S
.....	.....	.....	.....
1 0 1 0	RTCPS[4] ( $/2^5$ )	976us ~ 62.46ms 当 P6.0 = 32768Hz	976 us
1 0 1 1	RTCPS[3] ( $/2^4$ )	488us ~ 31.2ms 当 P6.0 = 32768Hz	488 us
1 1 0 0	RTCPS[2] ( $/2^3$ )	244us ~ 15.6ms 当 P6.0 = 32768Hz	244 us
1 1 0 1	RTCPS[1] ( $/2^2$ )	122us ~ 7.8ms 当 P6.0 = 32768Hz	122 us
1 1 1 0	RTCPS[0] ( $/2^1$ )	61us ~ 3.9ms 当 P6.0 = 32768Hz	61 us
1 1 1 1	RTCPSI ( $/2^0$ )	30.5us ~ 1.952ms 当 P6.0 = 32768Hz	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 Clock 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 &amp; 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

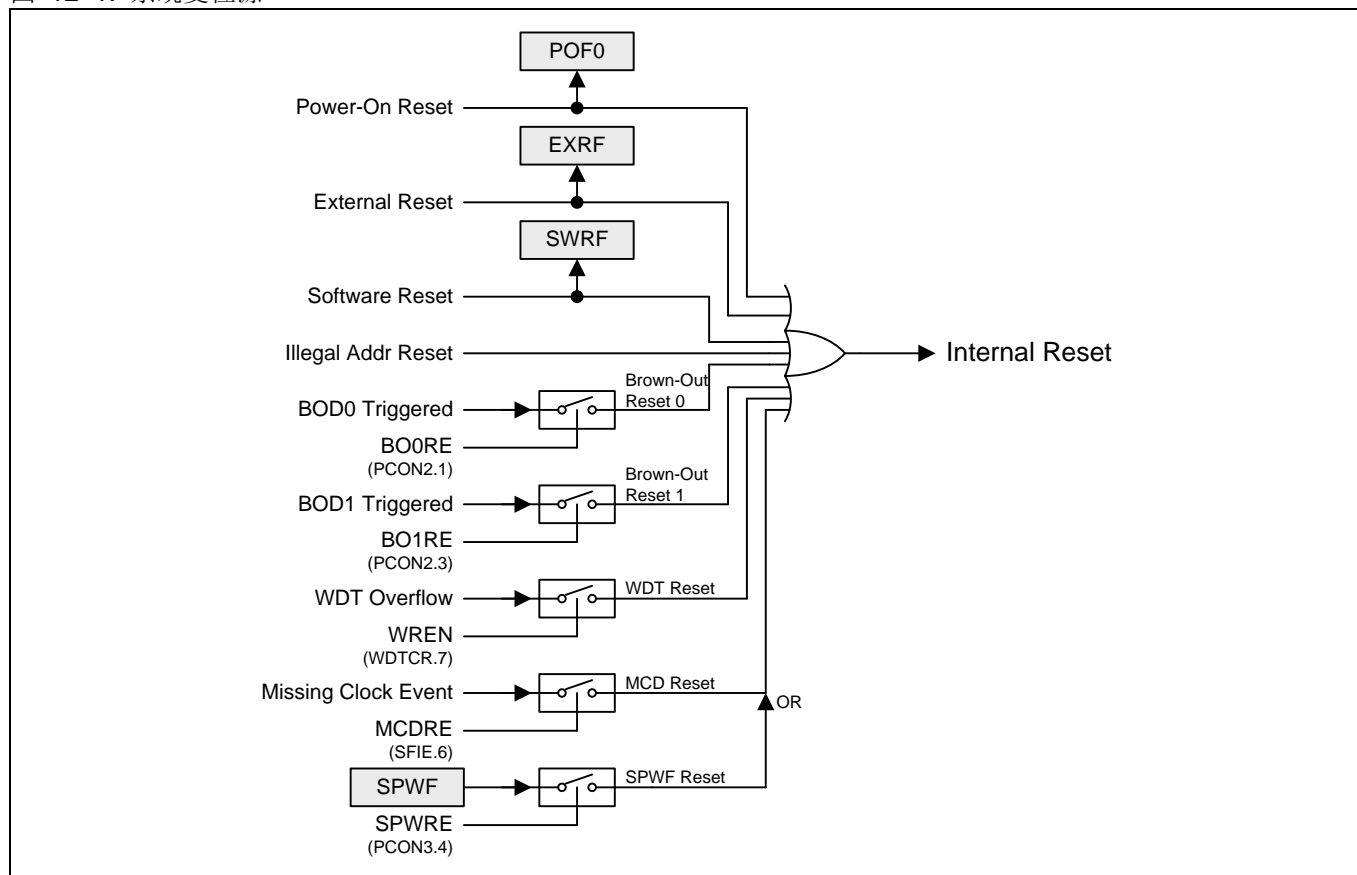
在复位期间，所有I/O寄存器都被设置为它们的初始值，程序根据OR选项从复位向量0000H或ISP起始地址开始执行。**MGEQ1C064**有8个复位源:上电复位、外部复位、软件复位、非法地址复位、低压监测复位0、低压监测复位1、WDT复位和时钟丢失监测复位。图 12-1显示了**MGEQ1C064**的系统复位源。

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 **MGEQ1C064** 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 $V_{POR}$ (POR开始电压)电压之前将保持复位状态。VDD电压降到 $V_{POR}$ 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 $V_{POR}$ 之下。

### PCON0: 电源控制寄存器0

SFR 页 = 0~F & P

SFR 地址 = 0x87

POR = 0001-0000

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也能有软件置位。

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”。它可以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

保持复位引脚 RST至少 24个振荡周期的高电平, 将产生一个复位信号。为了确保可靠的上电复位, 从RST引脚硬件复位是必要的。

### 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 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SRST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

SFR 地址 = 0xE7

POR = 0000-XXXX

7	6	5	4	3	2	1	0
ISPEN	SWBS	SRST	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: SRST, 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 &amp; 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**

MGEQ1C064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, 3.7V, 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。

**PCON1: 电源控制寄存器1**

SFR 页 = 0~F &amp; 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(4.2V/3.7/2.4/2.0)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(1.7V)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 &amp; 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MGEQ1C064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. 堆栈指针警告复位

### 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 ≥ 1111-1111, 将置位SPWF。

如果SPHB == 1111-0000, 当SP ≥ 1111-0000, 将置位SPWF。.

### 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 ≥ SPHB由硬件置位。当SP < SPHB 时，写“1”清除SPWF。

## 13. 电源管理

MGEQ1C064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WDTCCR寄存器的访问来操作这些电源事件。

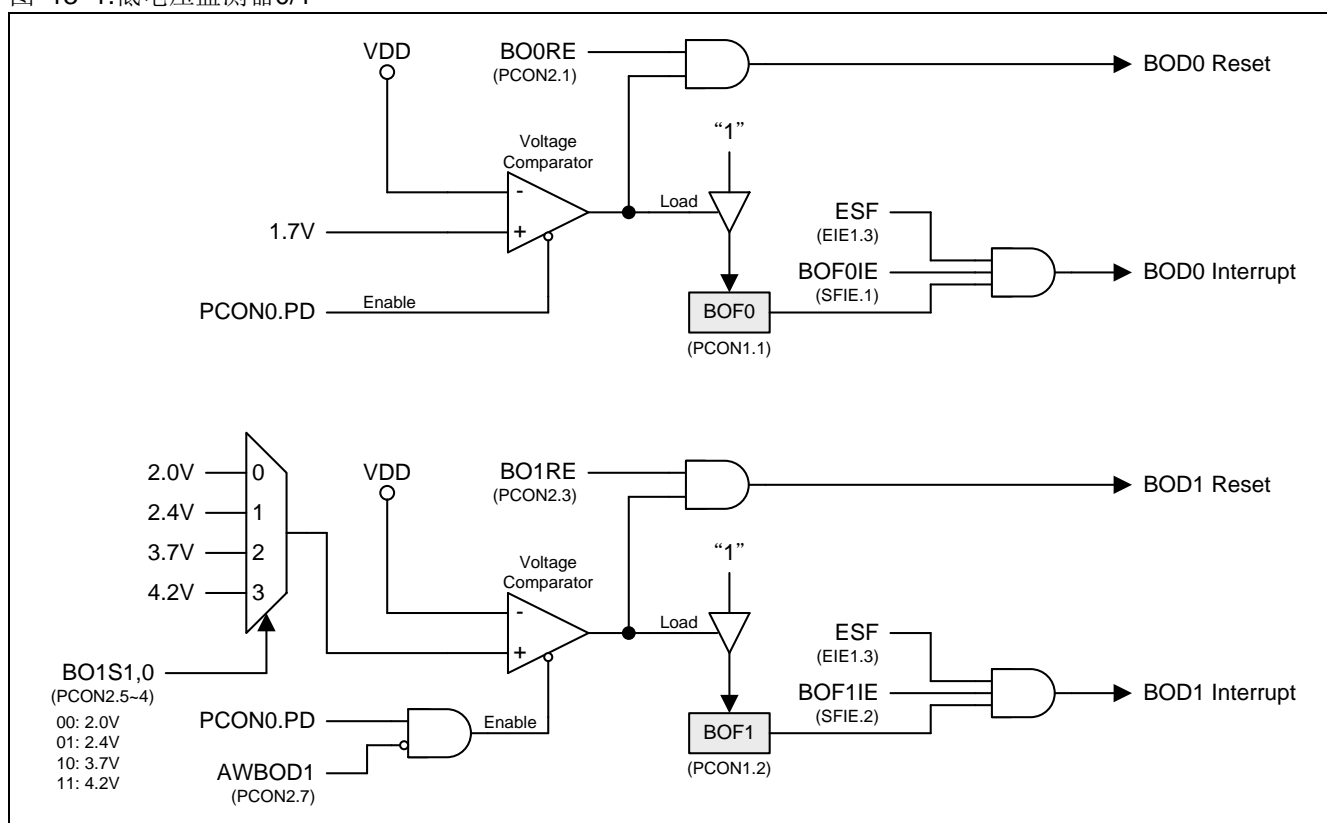
### 13.1. 低电压监测器

MGEQ1C064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/3.7V/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。使用者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”)使用者可以使MCU的速度最低到250Hz。

### 13.2.3. RTC 模式

MGEQ1C064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MGEQ1C064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MGEQ1C064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“13.1低电压监测器”和章节“15 中断”。

### 13.2.6. Idle 模式

可以通过软件的方式置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、定时器4、DMA、SPI、KBI、ADC、AC0、AC1、AC2、S0~S3、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”。

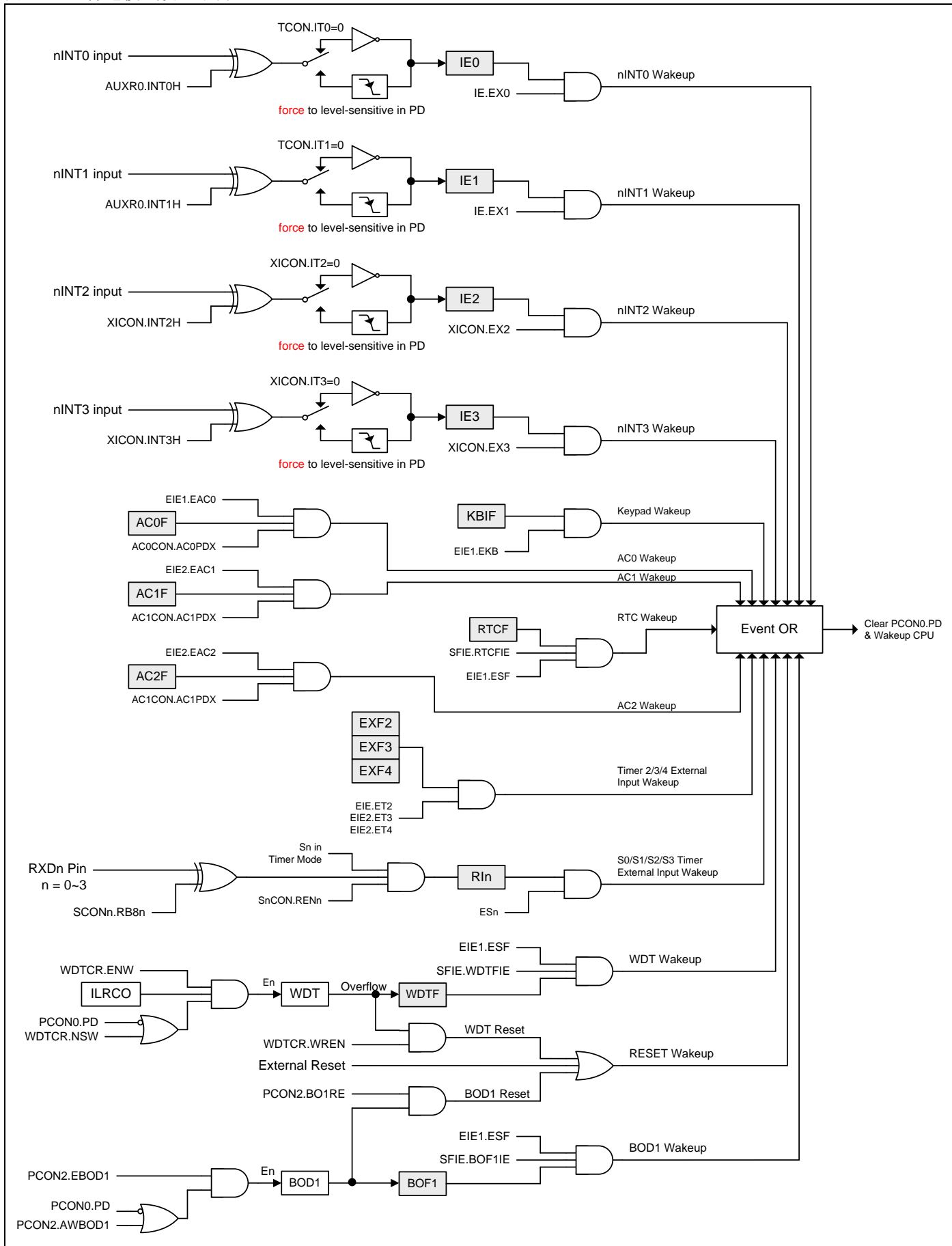
### 13.2.7. 掉电模式

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 MGEQ1C064 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。复位脚有上升沿电平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高电平以保证系统完全复位，复位脚变低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

MGEQ1C064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 &amp; P

SFR 地址 = 0x87

POR = 0001-0000

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 &amp; 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	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: 保留位, 写寄存器时, 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	3.7V
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	0	0	SPWRE	0	0	0	0
R/W	W	W	R/W	W	W	W	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 6~5: 保留位, 写寄存器时, 此位必须写“0”。

Bit 3~0: 保留位, 写寄存器时, 此位必须写“0”。

## 14. I/O 口配置(GPIO)

**MGEQ1C064** 有下列I/O端口：P0.0~P0.7, P1.0~P1.7, P2.0~P2.7, P3.0~P3.7, P4.0~P4.7和P6.0~P6.7。若启用外部晶振作为系统时钟或 RTC 输入,则端口 6.0和端口 6.1被配置为XTAL2和XTAL1。若禁用外部复位功能,则P4.7功能有效。可用 I/O 引脚的实际数量取决于封装类型。见表 14.1。

表 14.1.可用 I/O 引脚数量

封装类型	I/O 引脚	引脚输量
48-pin	P0.0~P0.7, P1.0~P1.7, P2.0~P2.7, P3.0~P3.7, P4.0~P4.6, P4.7(RST), P6.0(ECKI/XTAL2), P6.1(XTAL1), P6.2~P6.3	44 或 43 (选择 RST) 或 42 (选择 RST & ECKI) 或 41 (选择 RST & XTAL)

### 14.1. IO 结构

**MGEQ1C064**输入输出口分成两个配置类型。第一类仅仅是端口3有四种模式，这四种模式有：准双向口(标准8051的I/O端口)、推挽输出、集电极开漏输出和输入(高阻抗输入)。缺省值是弱上拉的准双向口模式。

其它口属于第二类，这些口有四种模式分别是仅模拟输入、上拉电阻的集电极开漏输出、集电极开漏输出和推挽输出。默认设置是仅模拟输入，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。

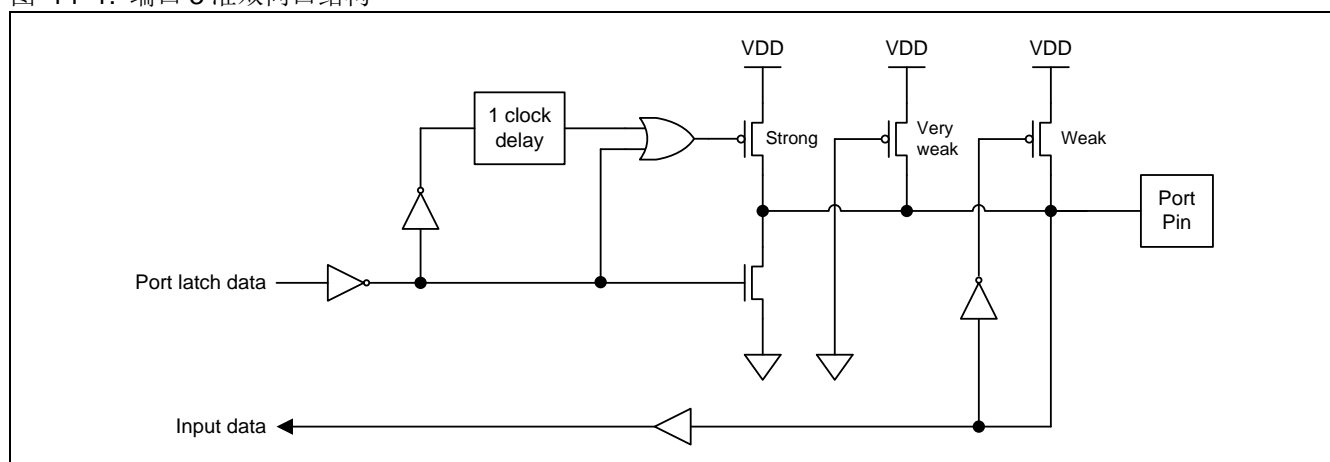
#### 14.1.1. 端口 3 准双向口结构

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 准双向口结构

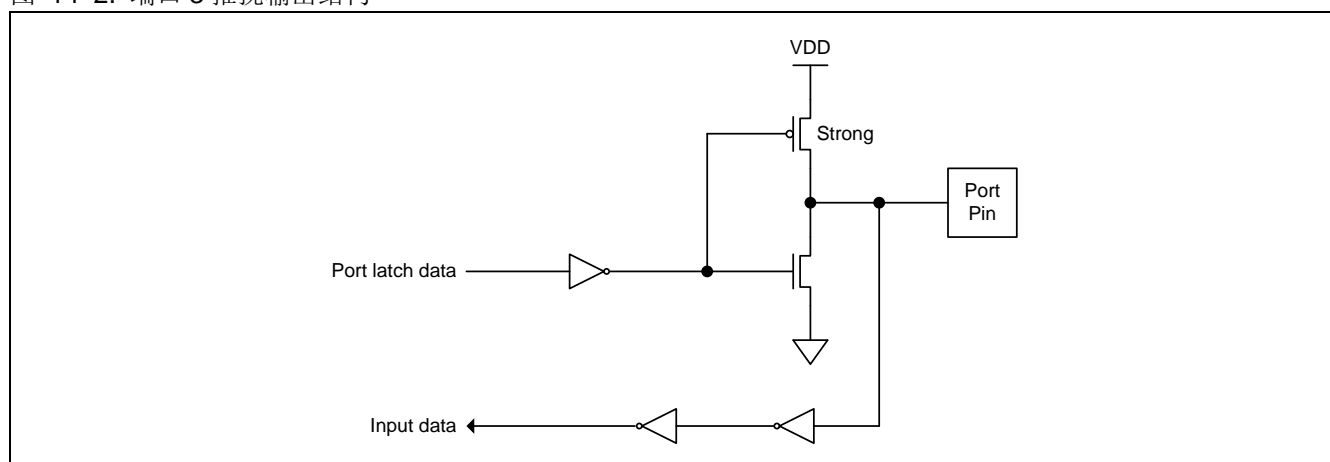


### 14.1.2. 端口 3 推挽输出结构

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。

推挽输出结构见图 14-2.

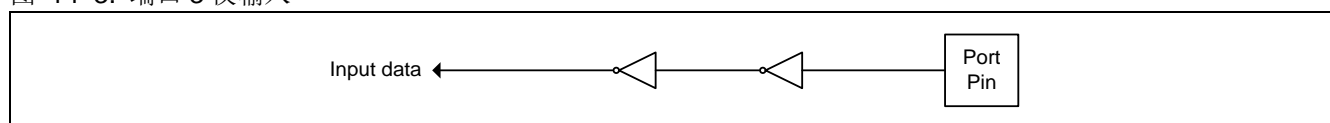
图 14-2. 端口 3 推挽输出结构



### 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，如下 图 14-3.

图 14-3. 端口 3 仅输入

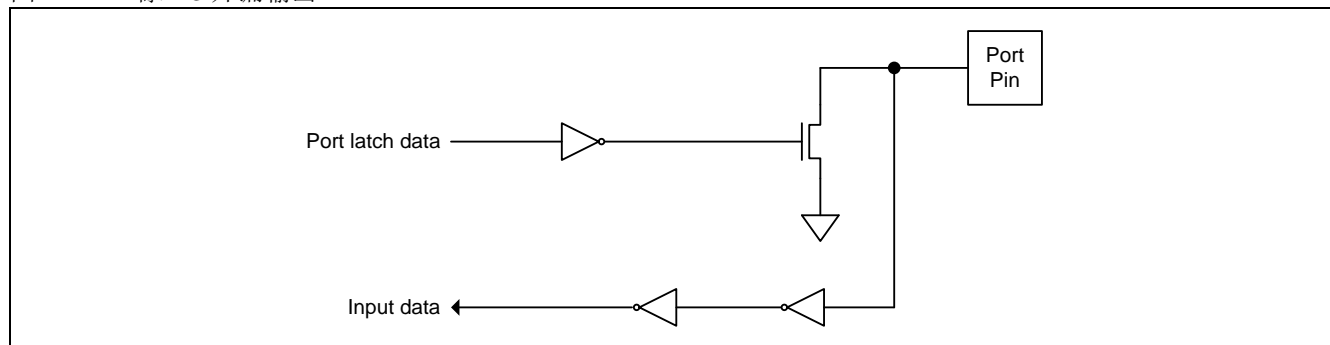


#### 14.1.4. 端口 3 开漏输出结构

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

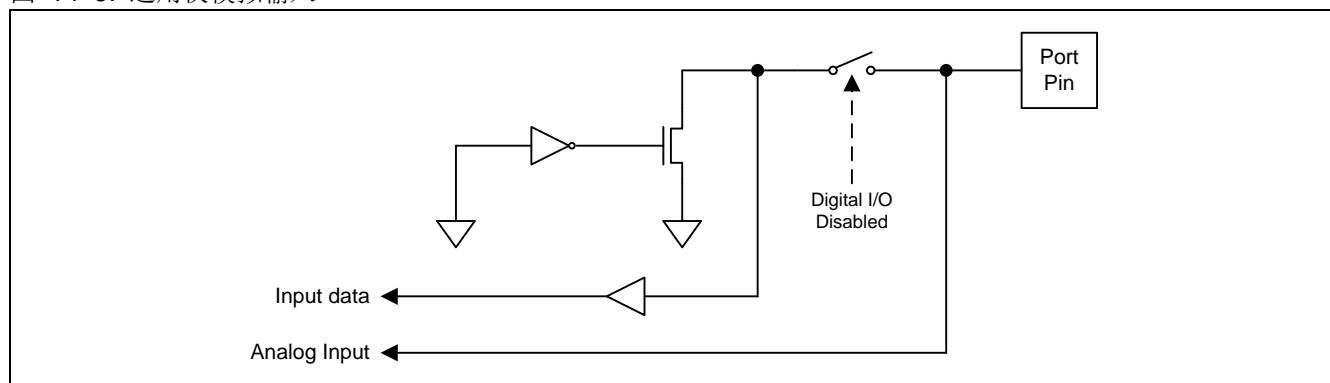


#### 14.1.5. 用仅模拟输入结构

在通用端口引脚上仅模拟输入结构是默认设置。作为ADC或模拟比较器输入应用，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关联的结构。

仅模拟输入端口结构如图 14-5.

图 14-5. 通用仅模拟输入

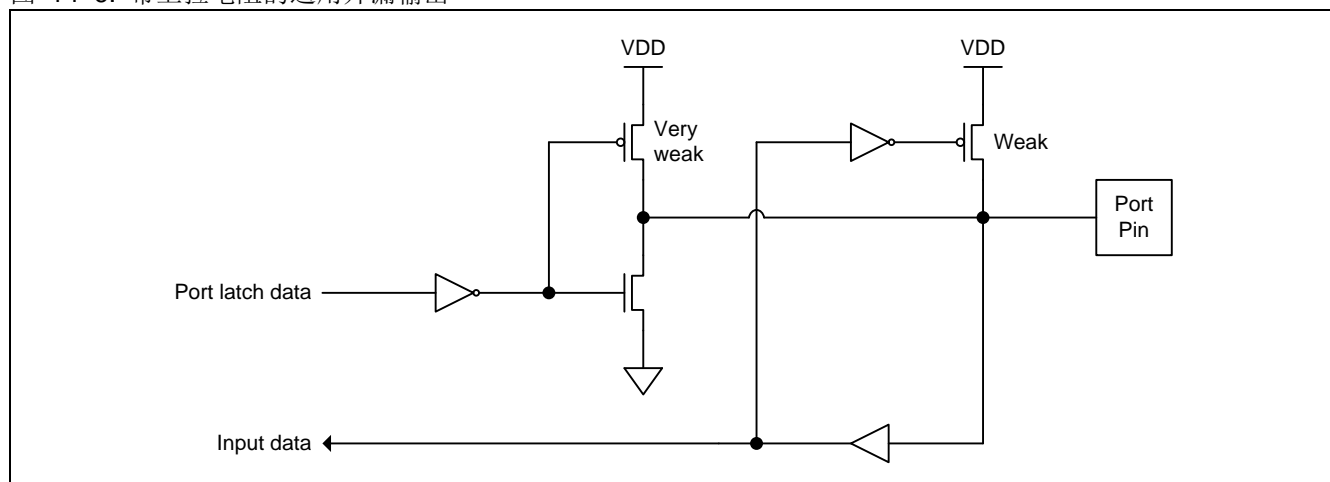


### 14.1.6. 带上拉电阻的通用开漏输出

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

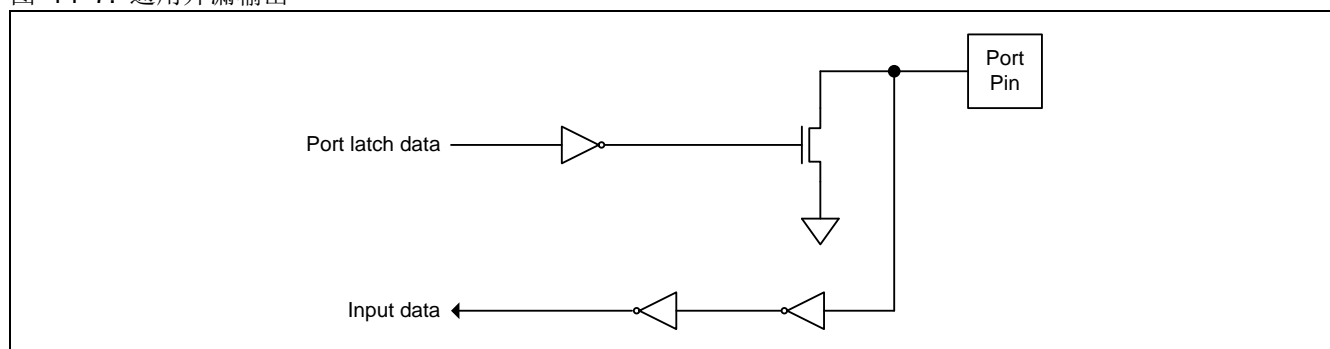


### 14.1.7. 通用开漏输出结构

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 14-7.

图 14-7. 通用开漏输出



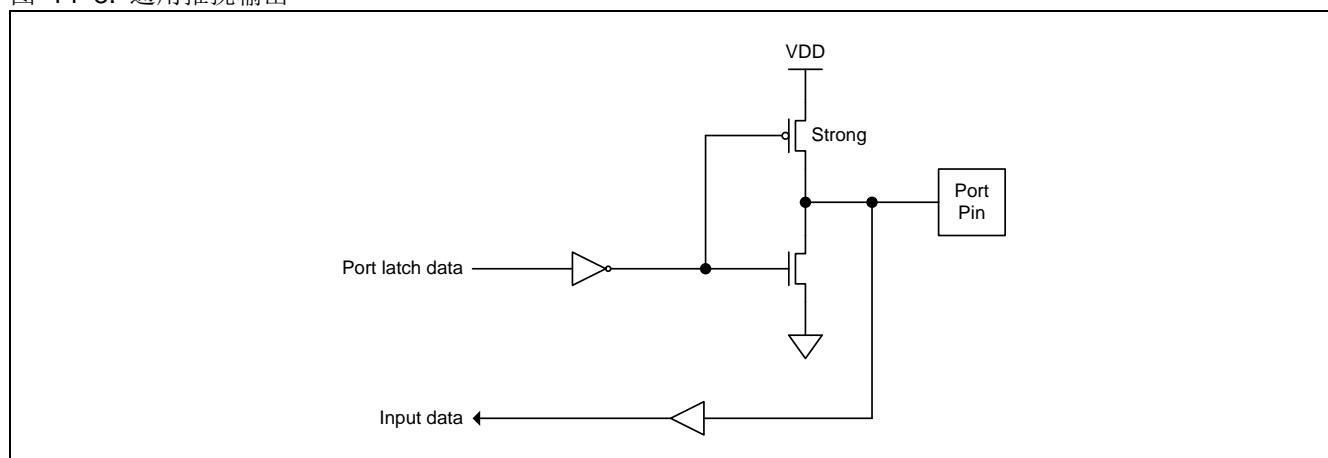
### 14.1.8. 通用端口的数字输入结构

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. 端口引脚输出驱动力选择

MGEQ1C064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“14.2.9 端口输出驱动力控制寄存器。”。

### 14.1.11. 端口引脚输出快速驱动选择

MGEQ1C064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“14.2.10 端口输出快速驱动控制寄存器”。

## 14.2. I/O 端口寄存器

MGEQ1C064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4种工作模式，如表 14.2。两个模式寄存器用于选择每个端口3引脚的输出类型。仅端口3支持准双向模式且系统复位之后它们为准双向模式。

表 14.2.端口3配置设定

P3M0.y	P3M1.y	端口模式
0	0	准双向(默认)
0	1	推挽输出
1	0	仅输入 (高阻抗输入)
1	1	集电极开漏输出

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输入。

表 14.3. 通用端口配置设定

PxM0.y	PxM1.y	端口模式
0	1	仅模拟输入(模拟)
1	1	带上拉电阻的集电极开漏
0	0	集电极开漏输出/ 通用数字输入(端口引脚设置为“1”)
1	0	推挽输出

这里x=0, 1, 2, 4, 5, 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 = 1111-1111

7	6	5	4	3	2	1	0
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
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	P0M0.6	P0M0.5	P0M0.4	P0M0.3	P0M0.2	P0M0.1	P0M0.0
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	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0
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 = 1111-1111

7	6	5	4	3	2	1	0
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
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
P2M0.7	P2M0.6	P2M0.5	P2M0.4	P2M0.3	P2M0.2	P2M0.1	P2M0.0
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
P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0
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 = 1111-1111

7	6	5	4	3	2	1	0
P3.7	P3.6	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
P3M0.7	P3M0.6	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
P3M1.7	P3M1.6	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

## 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 = 1111-1111

7	6	5	4	3	2	1	0
P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	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	P4M0.6	P4M0.5	P4M0.4	P4M0.3	P4M0.2	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	P4M1.6	P4M1.5	P4M1.4	P4M1.3	P4M1.2	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 5 寄存器 ((48 引脚封装不支持))****P5: 端口5寄存器**

SFR 页 = 仅 0 页

SFR 地址 = 0xF8

复位值 = 1111-1111

7	6	5	4	3	2	1	0
P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0:端口5输出数据通过CPU置位/清零

**P5M0: 端口5模式寄存器 0**

SFR 页 = 仅 0 页

SFR 地址 = 0xB5

复位值 = 0000-0000

7	6	5	4	3	2	1	0
P5M0.7	P5M0.6	P5M0.5	P5M0.4	P5M0.3	P5M0.2	P5M0.1	P5M0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**P5M1: 端口5模式寄存器 1**

SFR 页 = 仅 5 页

SFR 地址 = 0x92

复位值 = 1111-1111

7	6	5	4	3	2	1	0
P5M1.7	P5M1.6	P5M1.5	P5M1.4	P5M1.3	P5M1.2	P5M1.1	P5M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**14.2.7. 端口 6 寄存器****P6: 端口6寄存器**

SFR 页 = 仅 1 页

SFR 地址 = 0xF8

复位值 = 1111-1111

7	6	5	4	3	2	1	0
P6.7	P6.6	P6.5	P6.4	P6.3	P6.2	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
P6M0.7	P6M0.6	P6M0.5	P6M0.4	P6M0.3	P6M0.2	P6M0.1	P6M0.0
R/W	R/W	R/W	R/W	R/W	R/W	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
P6M1.7	P6M1.6	P6M1.5	P6M1.4	P6M1.3	P6M1.2	P6M1.1	P6M1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**14.2.8. 端口 7 寄存器 (48 引脚封装不支持)****P7: 端口7寄存器**

SFR 页 = 仅 2 页

SFR 地址 = 0xF8

复位值 = 1111-1111

7	6	5	4	3	2	1	0
1	1	1	1	1	P7.2	P7.1	P7.0
W	W	W	W	W	R/W	R/W	R/W

Bit 7~0: 端口7 输出数据通过CPU置位/清零.

**P7M0: 端口7模式寄存器 0**

SFR 页 = 仅 2 页

SFR 地址 = 0xB5

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	P7M0.2	P7M0.1	P7M0.0
W	W	W	W	W	R/W	R/W	R/W

**P7M1: 端口7模式寄存器 1**

SFR 页 = 仅 6 页

SFR 地址 = 0x92

复位值 = 1111-1111

7	6	5	4	3	2	1	0
1	1	1	1	1	P7M1.2	P7M1.1	P7M1.0
W	W	W	W	W	R/W	R/W	R/W

**14.2.9. 端口输出驱动力控制寄存器**

MGEQ1C064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	P7DC0	P6DC1	P6DC0	P5DC1	P5DC0	P4DC1	P4DC0
W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留位。软件在写入PDRVC1时，必须向这些位写入”0”。

Bit 6: P7DC0, 端口7低4位输出驱动力控制。

0: P7.2 ~ P7.0输出选择为高驱动力。

1: P7.2 ~ P7.0输出选择为低驱动力。

Bit 5: P6DC1, 端口6高4位输出驱动力控制。

0: P6.7 ~ P6.4输出选择为高驱动力。

1: P6.7 ~ P6.4输出选择为低驱动力。

Bit 4: P6DC0, 端口6低4位输出驱动力控制。

0: P6.3 ~ P6.2输出选择为高驱动力。

1: P6.3 ~ P6.2输出选择为低驱动力。

Bit 3: P5DC1, 端口5高4位输出驱动力控制。

0: P5.7 ~ P5.4输出选择为高驱动力。

1: P5.7 ~ P5.4输出选择为低驱动力。

Bit 2: P5DC0, 端口5低4位输出驱动力控制。

0: P5.3 ~ P5.0输出选择为高驱动力。

1: P5.3 ~ P5.0输出选择为低驱动力。

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。

## 14.2.10. 端口输出快速驱动控制寄存器

**MGEQ1C064** 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
P3FDC.7	P3FDC.6	P3FDC.5	P3FDC.4	P3FDC.3	P3FDC.2	P3FDC.1	P3FDC.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	R/W	R/W

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
P2FDC.7	P2FDC.6	P2FDC.5	P2FDC.4	P2FDC.3	P2FDC.2	P2FDC.1	P2FDC.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
0	P4FDC.6	P4FDC.5	P4FDC.4	P4FDC.3	P4FDC.2	P4FDC.1	P4FDC.0
W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留位。软件在写入P4FDC时, 必须向这些位写入"0"。

Bit 6~0: 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	P0FDC.6	P0FDC.5	P0FDC.4	P0FDC.3	P0FDC.2	P0FDC.1	P0FDC.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。

**P6FDC: 端口6快速驱动控制寄存器**

SFR 页 = 仅 C 页

SFR 地址 = 0x92

复位值 = 0000-0000

7	6	5	4	3	2	1	0
P6FDC.7	P6FDC.6	P6FDC.5	P6FDC.4	P6FDC.3	P6FDC.2	0	0
R/W	R/W	R/W	R/W	R/W	R/W	W	W

Bit 7~2: 端口6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。

Bit 1~0: 保留位。软件在写入P6FDC时, 必须向这些位写入"0"。

**P5FDC: 端口5快速驱动控制寄存器**

SFR 页 = 仅 D 页

SFR 地址 = 0x92

复位值 = 0000-0000

7	6	5	4	3	2	1	0
P5FDC.7	P5FDC.6	P5FDC.5	P5FDC.4	P5FDC.3	P5FDC.2	P5FDC.1	P5FDC.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 端口5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。

**P7FDC: 端口7快速驱动控制寄存器**

SFR 页 = 仅 E 页

SFR 地址 = 0x92

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	0	0	0	0	P7FDC.2	P7FDC.1	P7FDC.0
W	W	W	W	W	R/W	R/W	R/W

Bit 7~3: 保留位。软件在写入P7FDC时, 必须向这些位写入"0"。

Bit 2~0: 端口7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。

## 15. 中断

MGEQ1C064有24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。

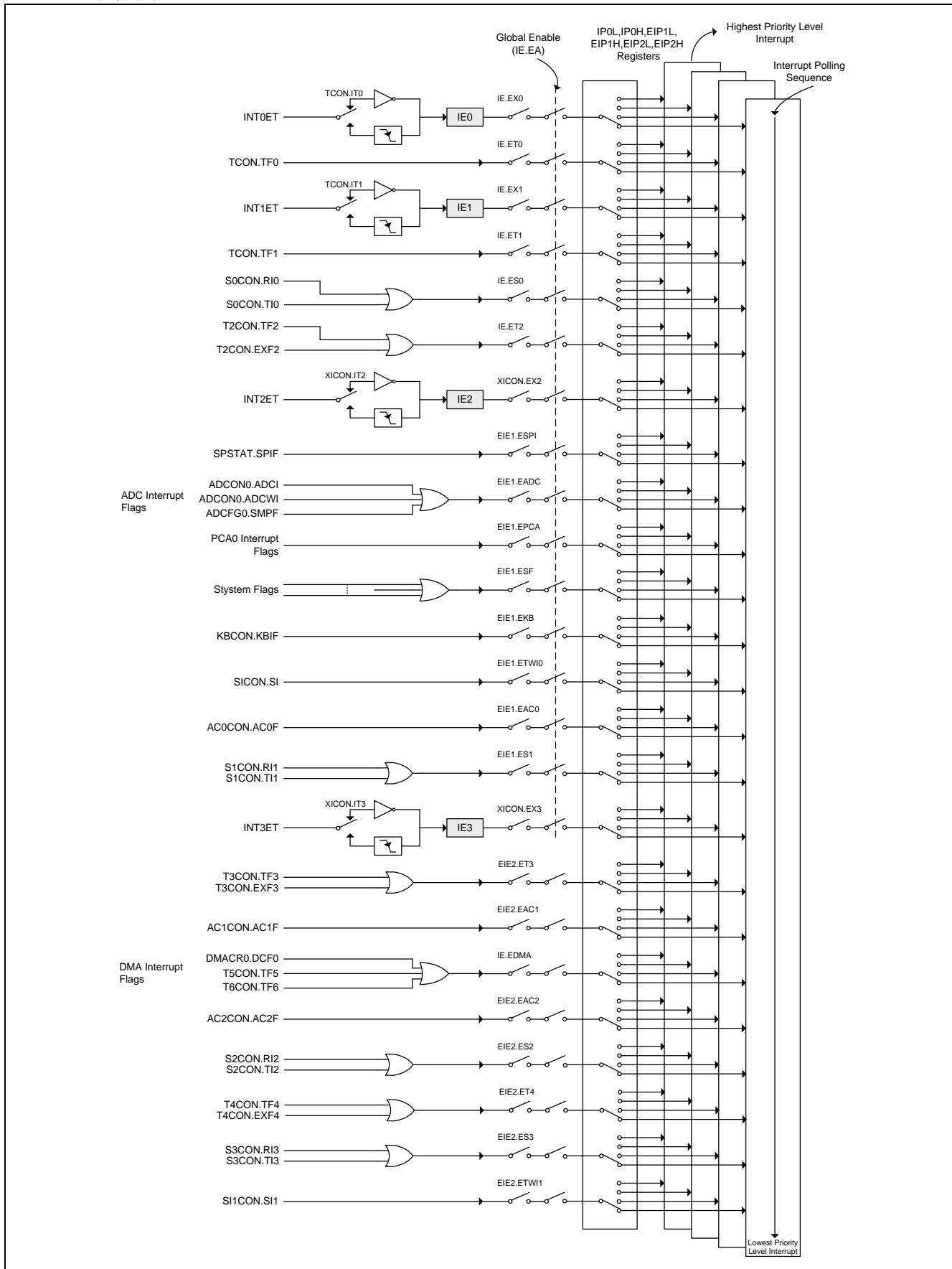
表 15.1. 中断源

序号	中断名称	使能位	请求位	优先级位	优先级	向量地址
#0	外部中断 0, nINT0	EX0	IE0	[ PX0H, PX0L ]	(最高)	0003H
#1	定时器0	ET0	TF0	[ PT0H, PT0L ]	...	000Bh
#2	外部中断 1, nINT1	EX1	IE1	[ PX1H, PX1L ]	...	0013H
#3	定时器1	ET1	TF1	[ PT1H, PT1L ]	...	001BH
#4	串口0	ES0	RI0, TI0	[ PS0H, PS0L ]	...	0023H
#5	定时器2	ET2	TF2, EXF2 (TF2L)	[ PT2H, PT2L ]	...	002Bh
#6	外部中断 2, nINT2	EX2	IE2	[ PX2H, PX2L ]	...	0033H
#7	SPI	ESPI	SPIF	[ PSPIH, PSPIL ]	...	003BH
#8	ADC	EADC	ADCI, ADCWI, SMPF	[ PADCH, PADCL ]	...	0043H
#9	PCA0	EPCA	CF, CCFn (n=0~7)	[ PPCAH, PPCAL ]	...	004Bh
#10	系统标志	ESF	(Note 1)	[ PSFH, PSFL ]	...	0053H
#11	键盘中断	EKB	KBIF	[ PKBH, PKBL ]	...	005BH
#12	TWI0/I2C0	ETWI0	SI	[ PTWI0H, PTWI0L ]	...	0063H
#13	模拟比较器 0	EAC0	AC0F	[ PAC0H, PAC0L ]	...	006BH
#14	串口1	ES1	RI1, TI1	[ PS1H, PS1L ]	...	0073H
#15	外部中断 3, nINT3	EX3	IE3	[ PX3H, PX3L ]	...	007BH
#16	定时器3	ET3	TF3, EXF3 (TF3L)	[ PT3H, PT3L ]	...	0083H
#17	模拟比较器 1	EAC1	AC1F	[ PAC1H, PAC1L ]	...	008BH
#18	DMA	EDMA	(Note 2)	[ PDMAH, PDMAL ]	...	0093H
#19	模拟比较器 2	EAC2	AC2F	[ PAC2H, PAC2L ]	...	009BH
#20	串口2	ES2	RI2, TI2	[ PS2H, PS2L ]	...	00A3H
#21	定时器4	ET4	TF4, EXF4 (TF4L)	[ PT4H, PT4L ]	...	00ABH
#22	串口3	ES3	RI3, TI3	[ PS3H, PS3L ]	...	00B3H
#23	TWI1	ETWI1	SI1	[ PTWI1H, PTWI1L ]	(最低)	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MCFD; 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

序号	中断名称	请求位	位的位置
#0	外部中断 0,nINT0	IE0	TCON.1
#1	定时器0	TF0	TCON.5
#2	外部中断 1,nINT1	IE1	TCON.3
#3	定时器1	TF1	TCON.7
#4	串口0	RI0, TI0	S0CON.0 S0CON.1
#5	定时器2	TF2, EXF2, (TF2L)	T2CON.7 T2CON.6 T2CON.5
#6	外部中断 2,nINT2	IE2	XICON.1
#7	SPI	SPIF	SPSTAT.7
#8	ADC	ADCI, ADCWI, SMPF	ADCON0.4 ADCON0.6 ADCFG0.2
#9	PCA0	CF, CCFn (n=0~5), CCFn (n=6~7)	CCON.7 CCON.5~0 PCAPWMn.3
#10	系统标志位	WDTF, BOF0, BOF1, SPWF, RTCF, MCDF, STAF, STOF, BM0F, BM1F, (TI0)	PCON1.0 PCON1.1 PCON1.2 PCON1.3 PCON1.4 PCON1.5 AUXR2.7 AUXR2.6 AUXR0.2 AUXR0.3 S0CON.1
#11	键盘中断	KBIF	KBCON.0
#12	TWI0/I2C0	SI	SICON.3
#13	模拟比较器 0	AC0F	AC0CON.4
#14	串口1	RI1, TI1	S1CON.0 S1CON.1
#15	外部中断 3,nINT3	IE3	XICON.5
#16	定时器3	TF3, EXF3, (TF3L)	T3CON.7 T3CON.6 T3CON.5
#17	模拟比较器 1	AC1F	AC1CON.4
#18	DMA	DCF0 TF5 TF6	DMACR0.0 T5CON.7 T6CON.7
#19	模拟比较器 2	AC2F	AC2CON.4
#20	串口2	RI2, TI2	S2CON.0 S2CON.1
#21	定时器4	TF4, EXF4, (TF4L)	T4CON.7 T4CON.6 T4CON.5
#22	串口3	RI3, TI3	S3CON.0 S3CON.1
#23	TWI1/I2C1	SI1	S1CON.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。当中断设置成边缘触发模式时, 进入中断服务程序硬件将清除外部中断所产生的标志位, 否则由外部请求源控制这个标志位。

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。

定时器4中断在TF4、EXF4和TF4L的报告标志位上与定时器3完全兼容。

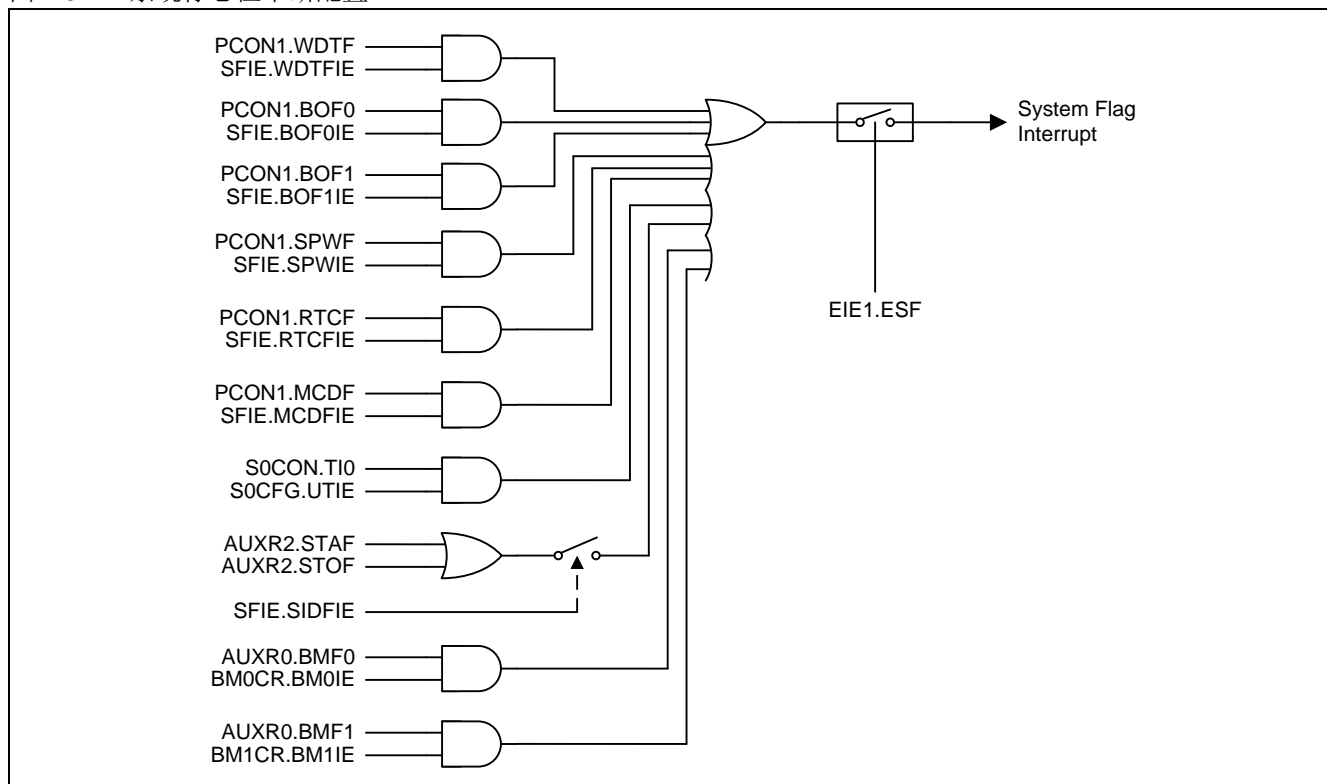
SPI中断由寄存器 SPSTAT里的SPIF位产生，SPI引擎完成一个SPI 传送后置该标志位。该标志位在执行中断服务程序后不会被硬件清除。

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。

PCA0中断由寄存器 CCON里的CF、CCF5、CCF4、CCF3、CCF2、CCF1 和CCF0位，PCAPWM6/7里的CCF6,CCF7逻辑或产生。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、BMF0、BMF1、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。OBM0和OBM1置位BMF0和BMF1。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. 系统标志位中断配置



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，AC2中断由AC12CON寄存器的位AC2F来产生。两者的功能和AC0一样。

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

序号	中断名称	使能位	位的位置
#0	外部中断 0,nINT0	EX0	IE.0
#1	定时器0	ET0	IE.1
#2	外部中断 1,nINT1	EX1	IE.2
#3	定时器1	ET1	IE.3
#4	串口0	ES0	IE.4
#5	定时器2	ET2	IE.5
#6	外部中断 2,nINT2	EX2	XICON.2
#7	SPI	ESPI	EIE1.0
#8	ADC	EADC	EIE1.1
#9	PCA	EPCA	EIE1.2
#10	系统标志位	ESF	EIE1.3
#11	键盘中断	EKB	EIE1.5
#12	TWI0/I2C0	ETWI0	EIE1.6
#13	模拟比较器 0, AC0	EAC0	EIE1.7
#14	串口1	ES1	EIE1.4
#15	外部中断 3,nINT3	EX3	XICON.6
#16	定时器3	ET3	EIE2.0
#17	模拟比较器 1, AC1	EAC1	EIE2.1
#18	DMA	EDMA	IE.6
#19	模拟比较器 2, AC2	EAC2	EIE2.7
#20	串口2	ES2	EIE2.2
#21	定时器4	ET4	EIE2.3
#22	串口3	ES3	EIE2.4
#23	TWI1/I2C1	ETWI1	EIE2.6

MGEQ1C064有24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)。IPOL、IPOH、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。

请注意中断号 19~23，分别是模拟比较器2、串口2、定时器4、串口3 和 TWI1，这5个中断不支持优先级 2~4。当系统需要使用中断优先级时，不建议使用中断19~23，以免误操作。

## 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：

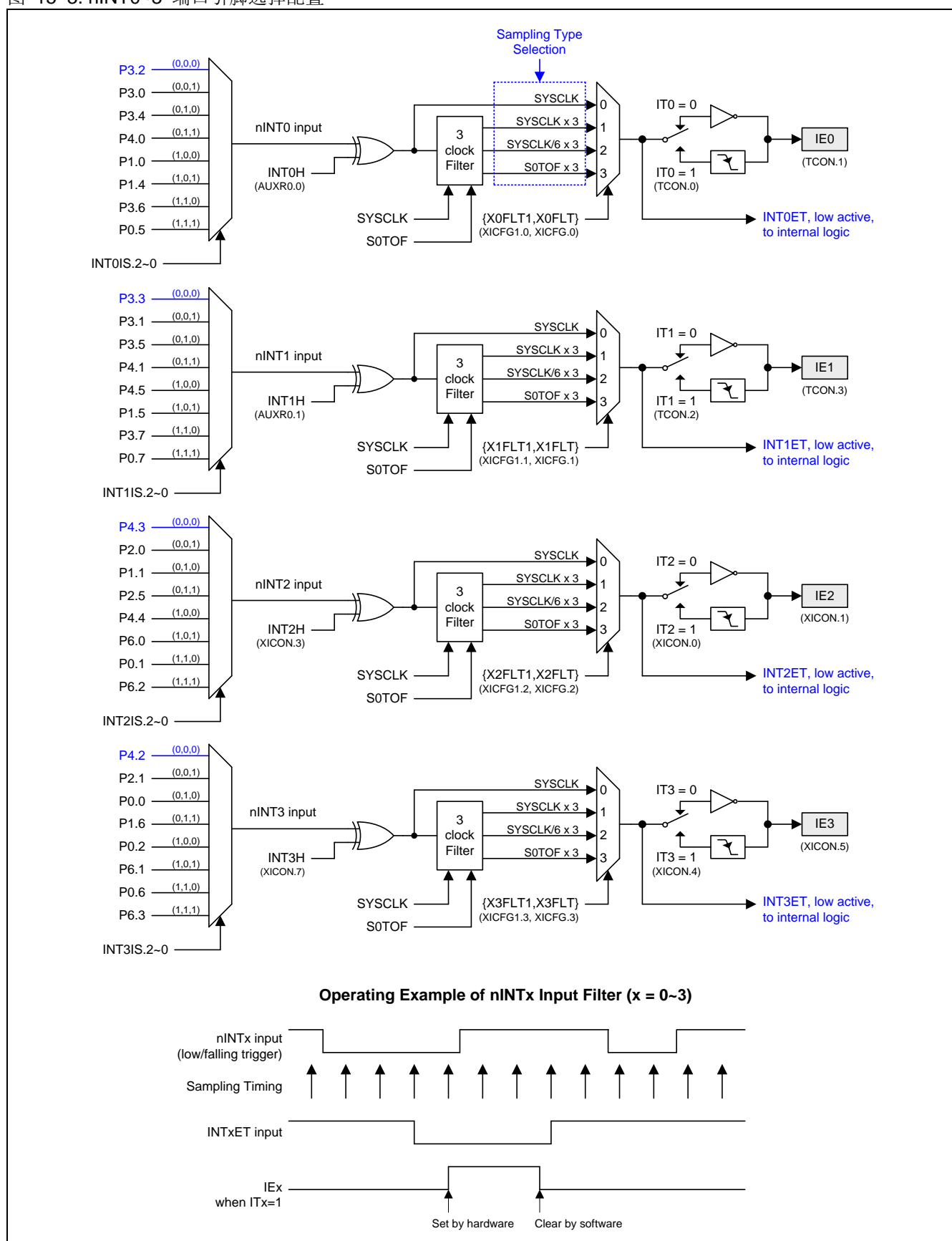
- 进行中已经有一个同级或更高级优先级的中断。
- 行中当前周期(轮询周期)不是指令执行结束周期。
- 指令进行是RETI 或IE、IP0L、IP0H、EIE1、EIP1L、EIP1H、EIE2、EIP2L、EIP2H和XICON寄存器的写操作。

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)

The MGEQ1C064 提供灵活的 nINT0、nINT1、nINT2 和 nINT3 输入源选择去共享端口引脚输入。

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 = 0X00-0000

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
P60OC1	P60OC0	P60FD	PBKF	BM1F	BM0F	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: PBKF, PWM终止标志。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 终止事件出现。仅由软件清零。

1: PWM 终止事件出现或软件触发一个PWM 终止。

Bit 3: BM1F, OBM1 标志。该位由OBM1切换源使能置位(BM1SE)。

0: 没有OBM1切换事件发生。仅由软件清零。

1: 有OBM1切换事件发生

Bit 2: BM0F, OBM0 标志。该位由OBM0切换源使能置位(BMOSE)

0: 没有OBM0切换事件发生。仅由软件清零。

1: 有OBM0切换事件发生。

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: 外部中断控制寄存器**

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 7: 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 6: 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: 侦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: 侦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。

#### IPOL: 中断优先级 0 低寄存器

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	EPCA	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, AUXR0的位{BM1F、BM0F}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, AUXR0的位{BM1F、BM0F}置位, 或TI0与UTIE一起置位时使能中断。

Bit 2: EPCA, PCA0中断使能。

0: 禁止PCA0中断。

1: 使能PCA0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	PTWIO0L	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: PTWIO0L, TWIO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, PCA0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, PCA0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
EAC2	ETWI1	0	ES3	ET4	ES2	EAC1	ET3
R/W	R/W	W	R/W	R/W	R/W	R/W	R/W

Bit 7: EAC2, 使能模拟比较器 2 (AC2) 中断。

0: 禁止AC2 中断。

1: 使能AC2 中断。

Bit 6: ETWI1, 使能TWI1 中断。

0: 禁止TWI1 中断。

1: 使能TWI1 中断。

Bit 5: 保留位。软件在写入EIE2时, 必须向这些位写入” 0”。

Bit 4: ES3, 使能串口 3 (UART3) 中断。

0: 禁止 串口3 中断。

1: 使能串口3 中断。

Bit 3: ET4, 定时器4中断使能。

0: 禁止定时器4 中断。

1: 使能定时器4 中断。

Bit 2: ES2, 使能串口 2 (UART2) 中断。

0: 禁止串口2 中断。

1: 使能串口2 中断。

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 低寄存器**

SFR 页 = 0~F

SFR 地址 = 0xA6

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PAC2L	PTWI1L	0	PS3L	PT4L	PS2L	PAC1L	PT3L
R/W	R/W	W	R/W	R/W	R/W	R/W	R/W

Bit 7: PAC2L, AC2中断优先级低位

Bit 6: PTWI1L, TWI1 中断优先级低位

Bit 5: 保留位。软件在写入EIP2L时, 必须向这些位写入” 0”。

Bit 4: PS3L, UART3 中断优先级低位

Bit 3: PT4L, Timer 4中断优先级低位.

Bit 2: PS2L, UART2 中断优先级低位

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

Bit 0: PT3L, 定时器3 中断优先级低位

**EIP2H: 扩展中断优先级 2 高寄存器**

SFR 页 = 0~F

SFR 地址 = 0xA7

复位值 = 0000-0000

7	6	5	4	3	2	1	0
PAC2H	PTWI1H	0	PS3H	PT4H	PS2H	PAC1H	PT3H
R/W	R/W	W	R/W	R/W	R/W	R/W	R/W

Bit 7: PAC2H, AC2 中断优先级高位

Bit 6: PTWI1H, TWI1 中断优先级高位

Bit 5: 保留位。软件在写入EIP2H时, 必须向这些位写入” 0”。

Bit 4: PS3H, UART3 中断优先级高位

Bit 3: PT4H, Timer 4 中断优先级高位

Bit 2: PS2H, UART2 中断优先级高位

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	P3.7
1 1 1	P0.7

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

INTOIS.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	P3.6
1 1 1	P0.5

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

X3FLT1, X3FLT	nINT3输入滤波模式
0 0	禁止
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

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

Bit 1: X1FLT, nINT1 Filter mode control. It selects nINT1 input filter mode with X1FLT1 (XICFG1.1)

X1FLT1, X1FLT	nINT1输入滤波模式
0 0	禁止
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	禁止
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	INTOIS.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.1~0一起构成nINT1输入端口引脚选择位。

Bit 6: INTOIS2, 与INTOIS.1~0一起构成nINT0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	P4.3
0 0 1	P2.0
0 1 0	P1.1
0 1 1	P2.5
1 0 0	P4.4
1 0 1	P6.0
1 1 0	P0.1
1 1 1	P6.2

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	nINT3输入引脚选择
0 0 0	P4.2
0 0 1	P2.1
0 1 0	P0.0
0 1 1	P1.6
1 0 0	P0.2
1 0 1	P6.1
1 1 0	P0.6
1 1 1	P6.3

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	0	COPLK	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	W	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/计数器

**MGEQ1C064** 有5个16位定时器/计数器：定时器0，定时器1，定时器2，定时器3和定时器4。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、T3MOD.T3X12和T4MOD.T4X12可以设置定时器0/1/2/3/4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，T3或T4。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 and 定时器 1

#### 16.1.1. 定时器 0/1 Mode 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

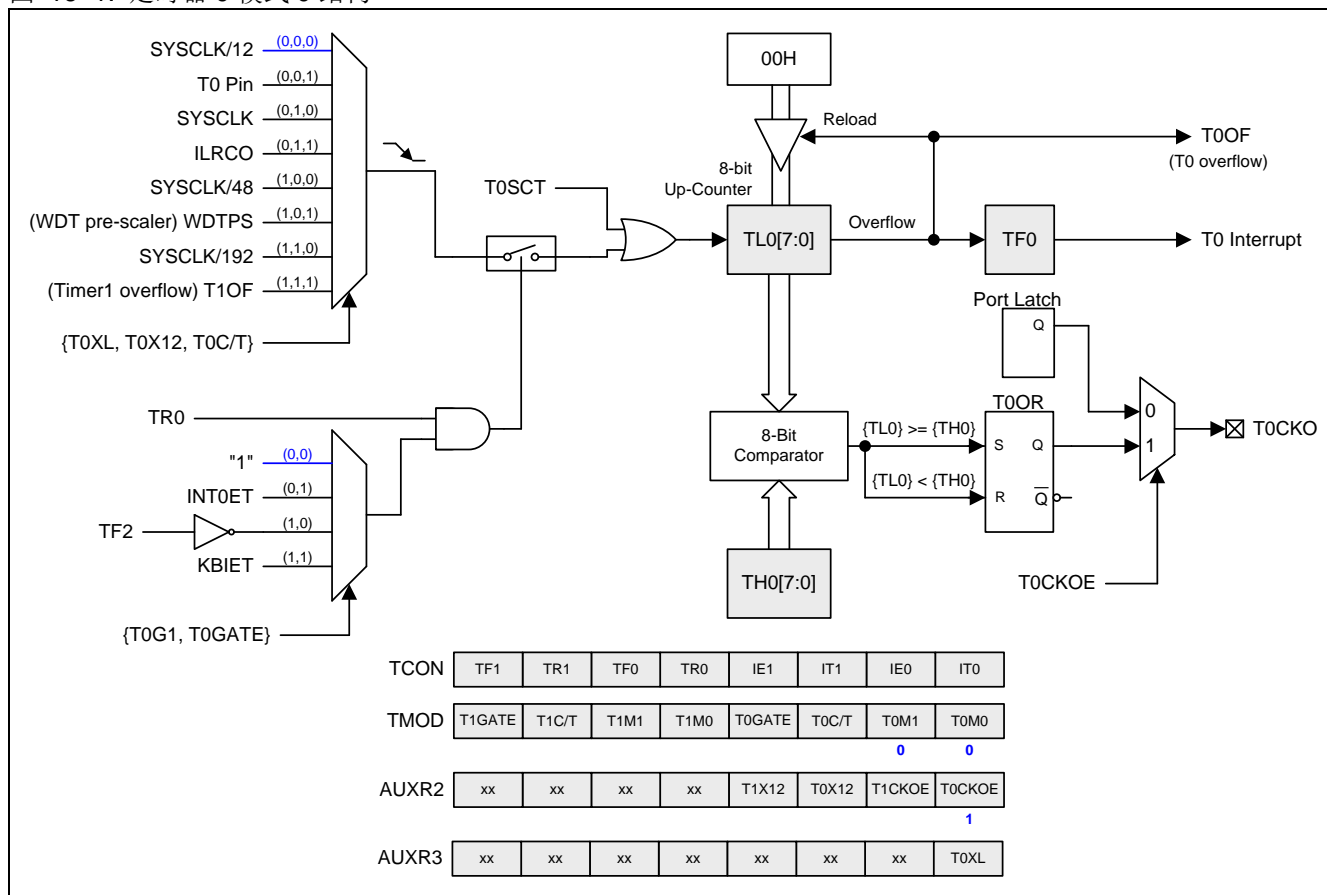
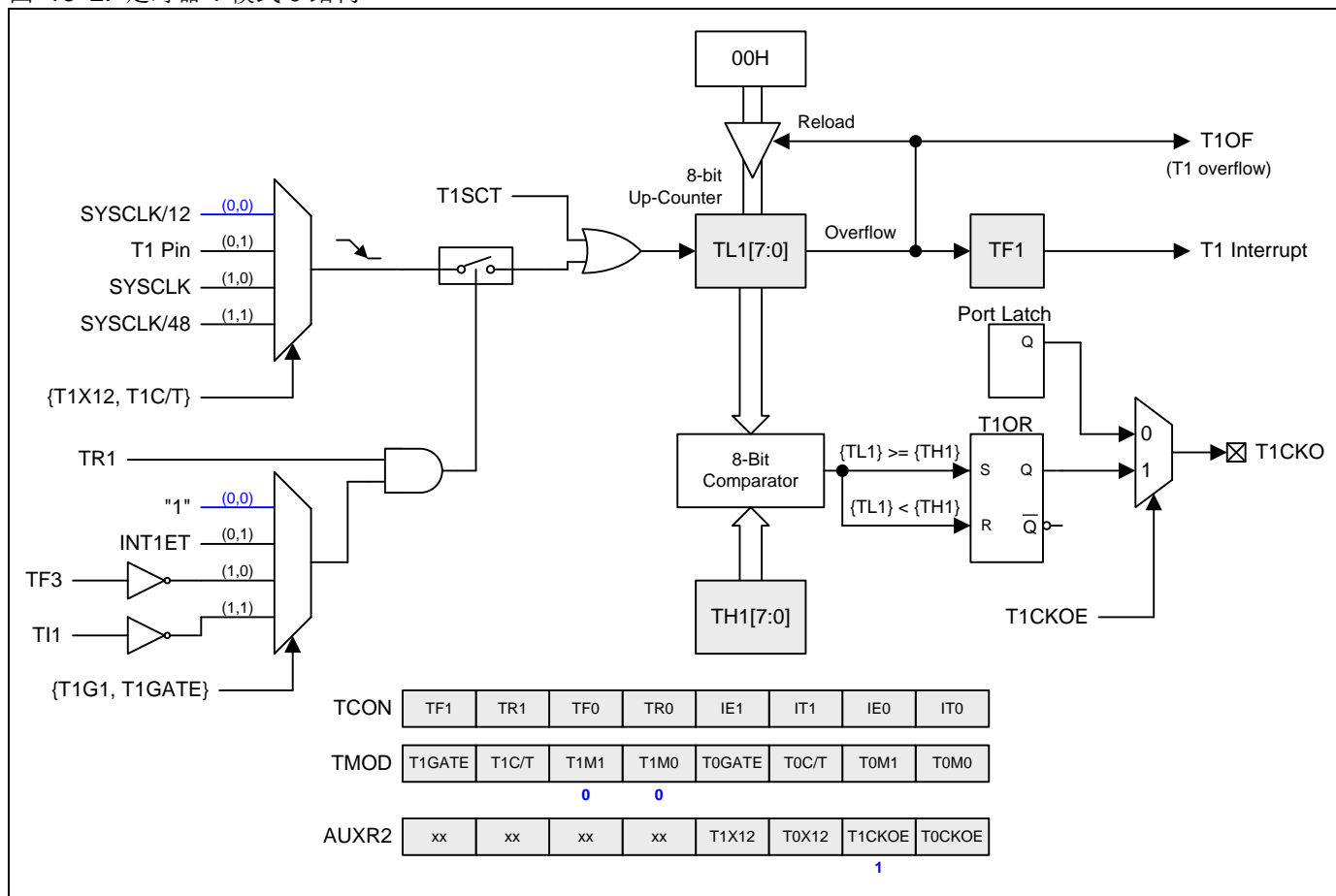


图 16-2. 定时器 1 模式 0 结构



## 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

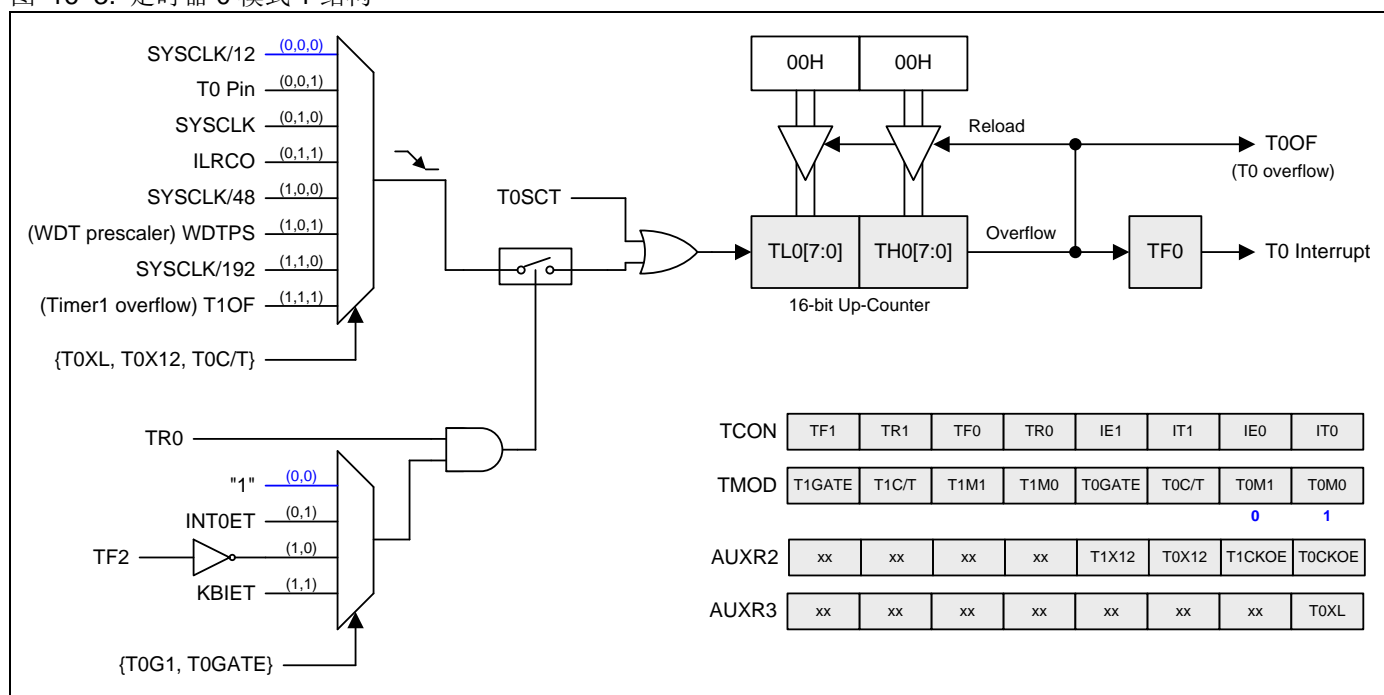
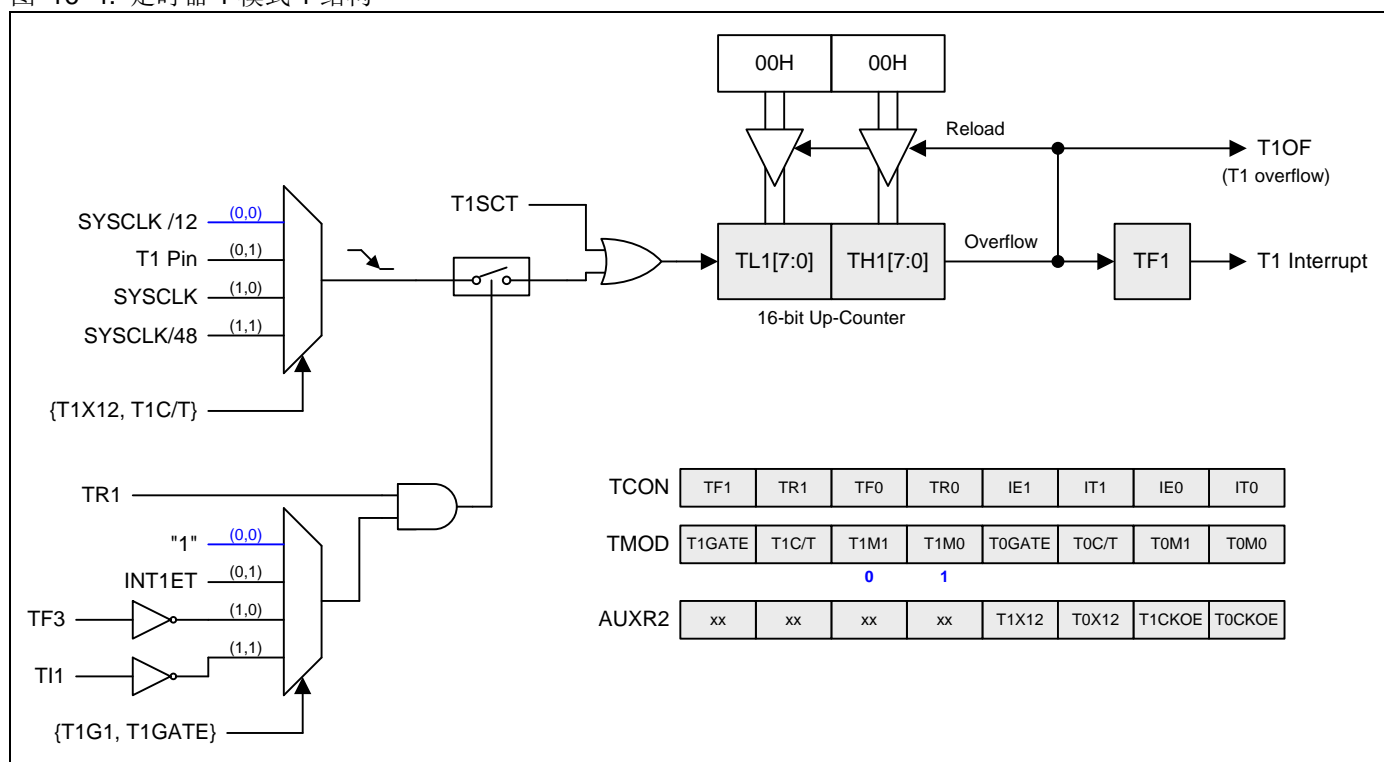


图 16-4. 定时器 1 模式 1 结构



## 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

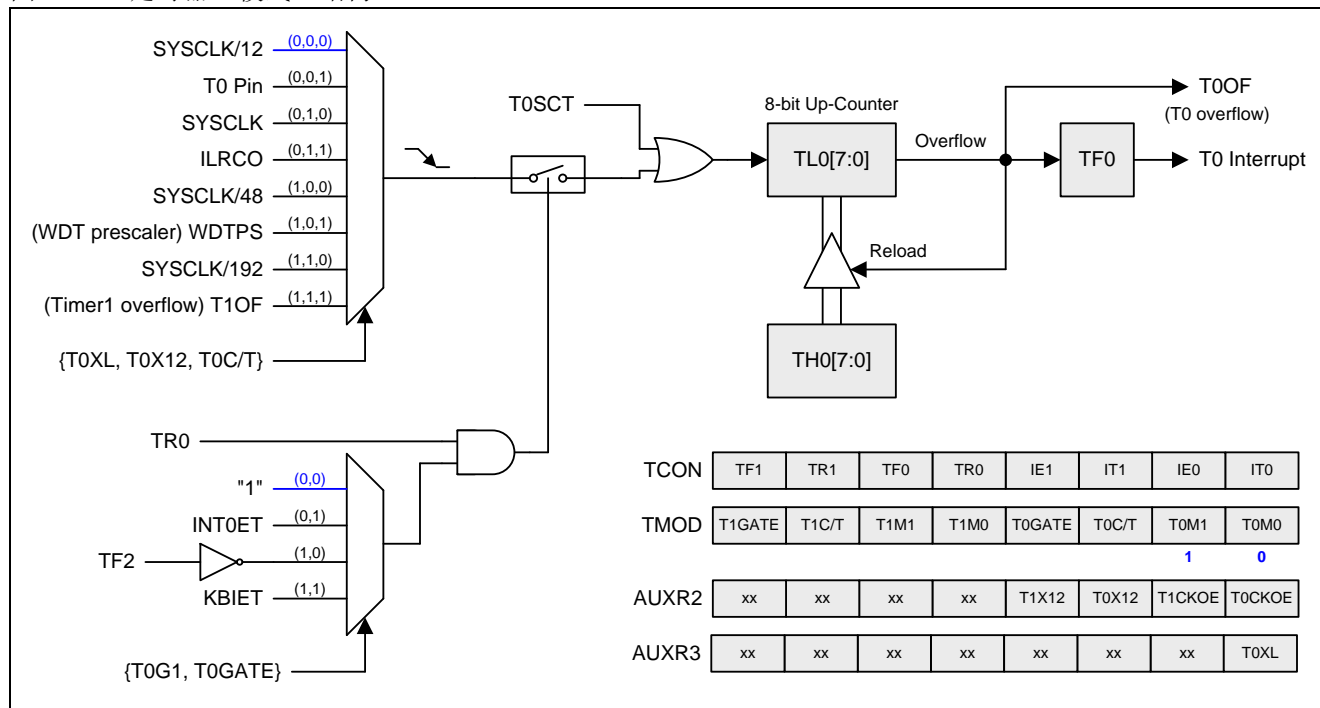
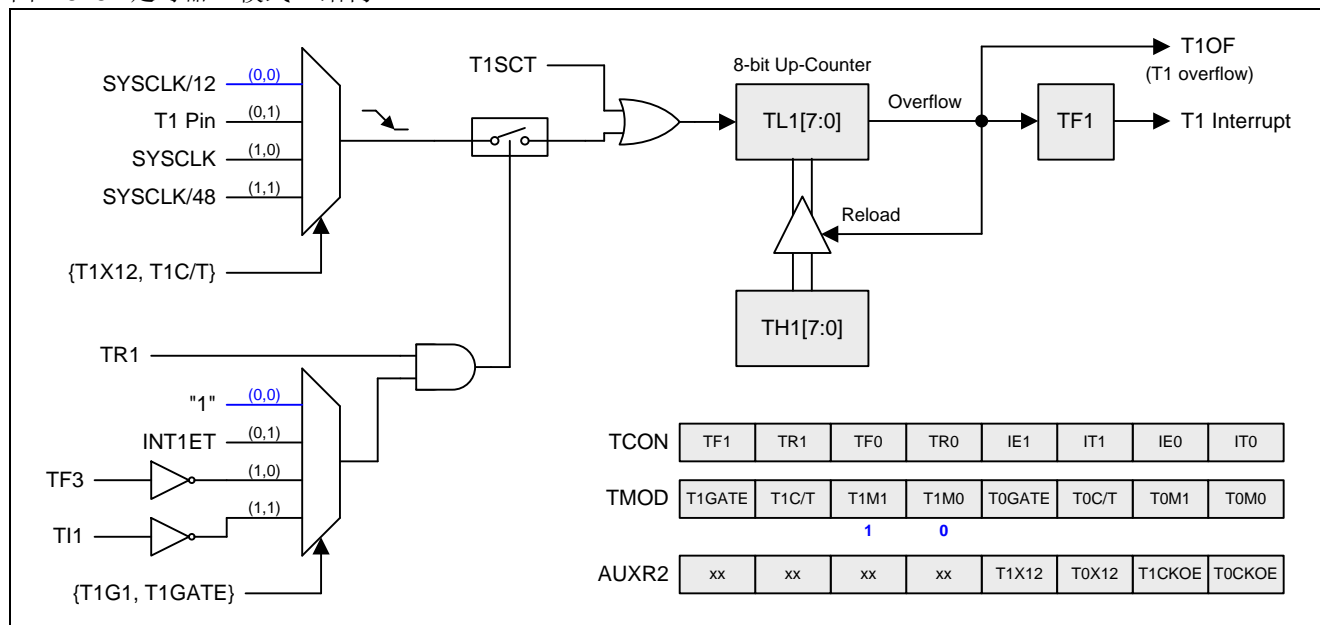


图 16-6. 定时器 1 模式 2 结构



### 16.1.4. 定时器 0/1 模式 3

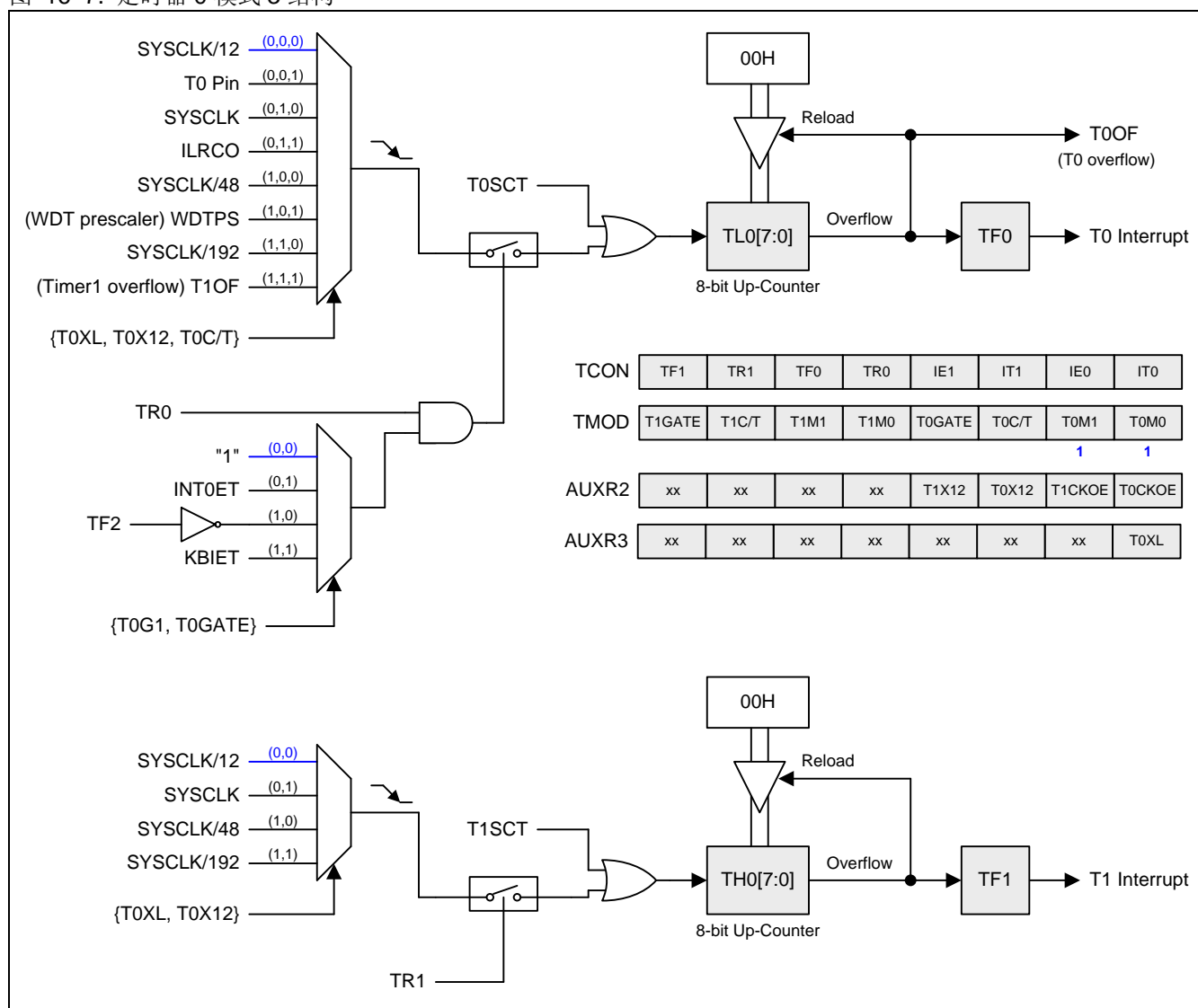
当定时器0 和定时器1 处于模式 3 时，定时器1 的一些寄存器将与定时器0一起使用。同时定时器1在模式3下被挂起，即使设置TR1=1。

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(不能作为外部事件计数器)，定时器1的TR1将被用于控制TH0。TH0溢出将置位TF1并触发定时器1中断。

为了保证TL1不发生故障，请在定时器0设置为模式3之前将定时器1设置为模式3。

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。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

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

图 16-9. 定时器 0 时钟输出公式

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

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

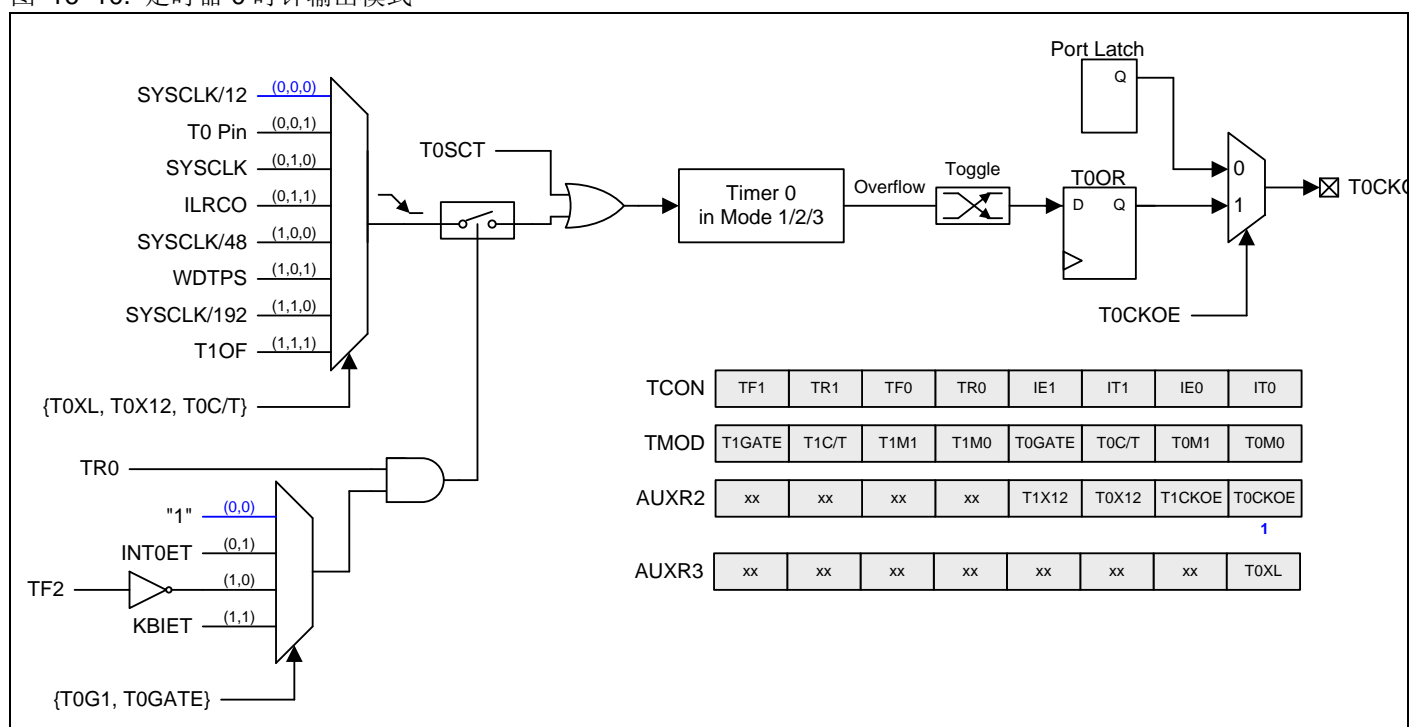


图 16-11. 定时器 0 时钟输出控制

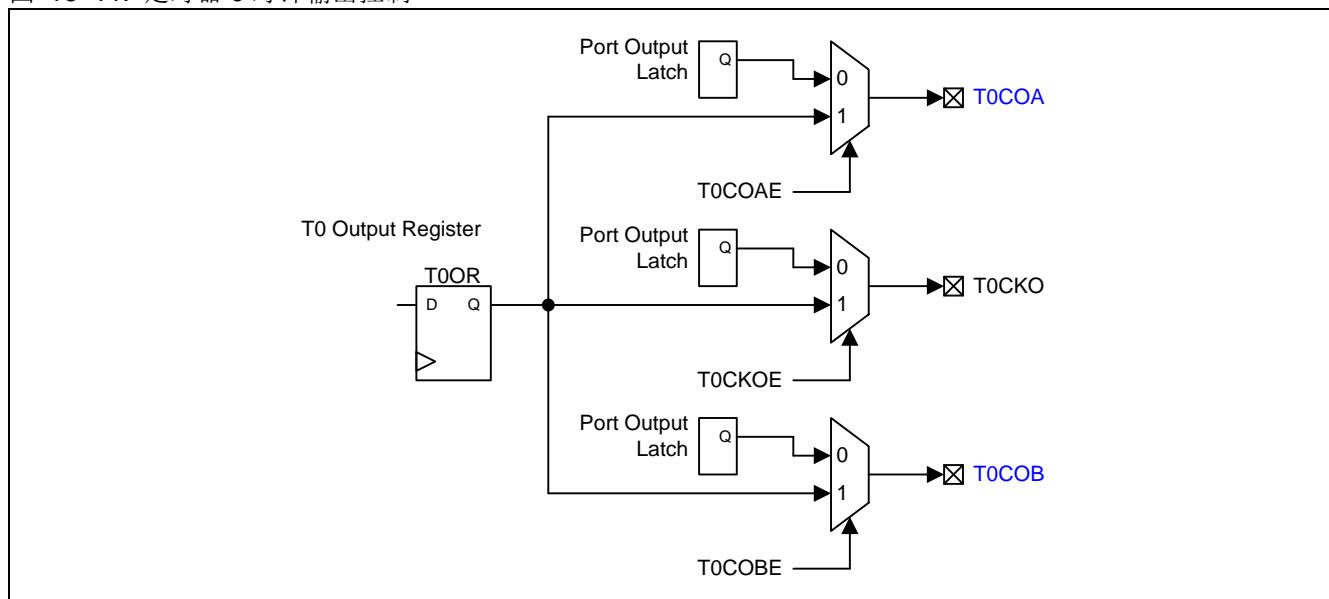
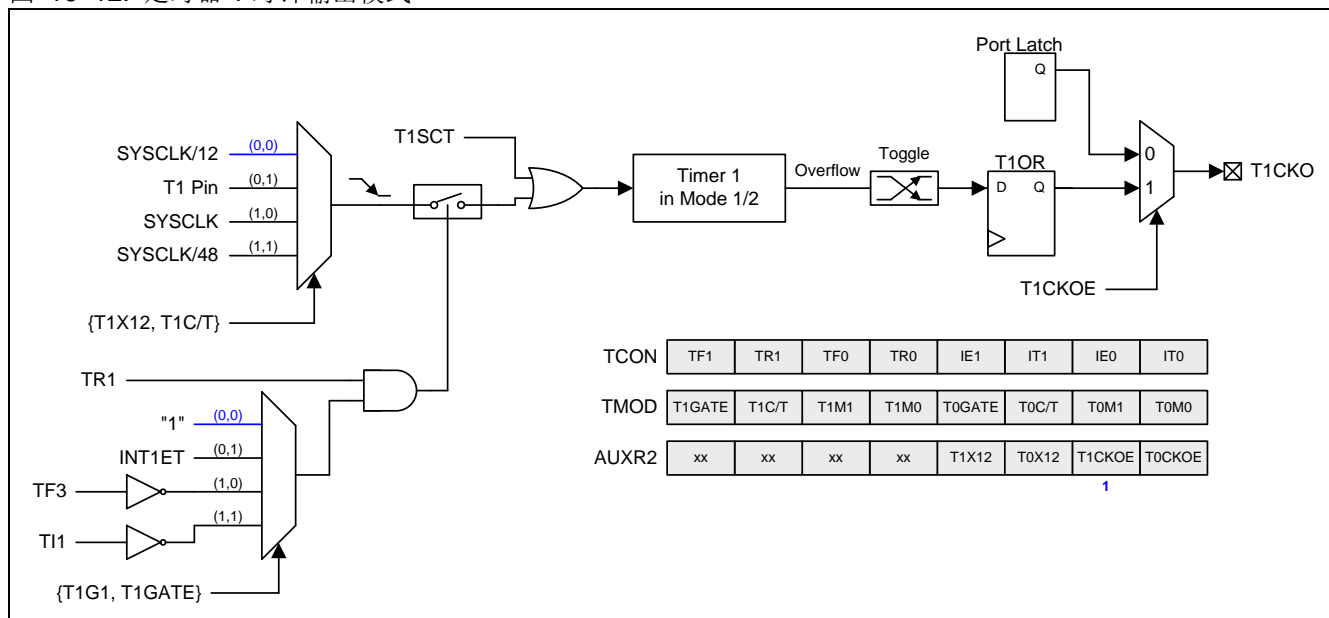


图 16-12. 定时器 1 时钟输出模式



### 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。
- AUXR2 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-----&gt;|←-----Timer0-----&gt;|

Bit 7: T1GATE, 定时器1门控制位

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

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: 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/计数器1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激活
1 0	TF2激活
1 1	KBI激活

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: 工作模式选择

TOM1	TOM0	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	0	COPLK	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	W	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时钟选择
0 0	SYSClk/12
0 1	T1 Pin
1 0	SYSClk
1 1	SYSClk/48

Bit 2: TOX12, 与TOC/T和TOXL一起控制定时器0时钟源选择

TOXL, TOX12, TOC/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

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时钟输出在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时钟输出在T0CKO端口引脚。

### 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	TOXL
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: TOXL是定时器0预分频控制位。TOXL功能定义请参考TOX12 (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	AC1FLT1	AC0OE	AC0FLT1
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	P3.6
1 1	P3.7

**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	C0FDC1	C0FDC0	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

**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
0	0	T0COBE	T0COAE	0	0	0	0
W	W	R/W	R/W	W	W	W	W

Bit 5: T0CKOB(=T0OR) 输出使能.

0: 禁止T0CKOB 输出

1: 使能P4.3输出T0CKOB.

Bit 4: T0CKOA(=T0OR) 输出使能

0: 禁止T0CKOA 输出

1: 使能P4.0输出T0CKOA

## 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(向上或向下计数)、8位PWM、波特率发生器和可编程时钟输出。

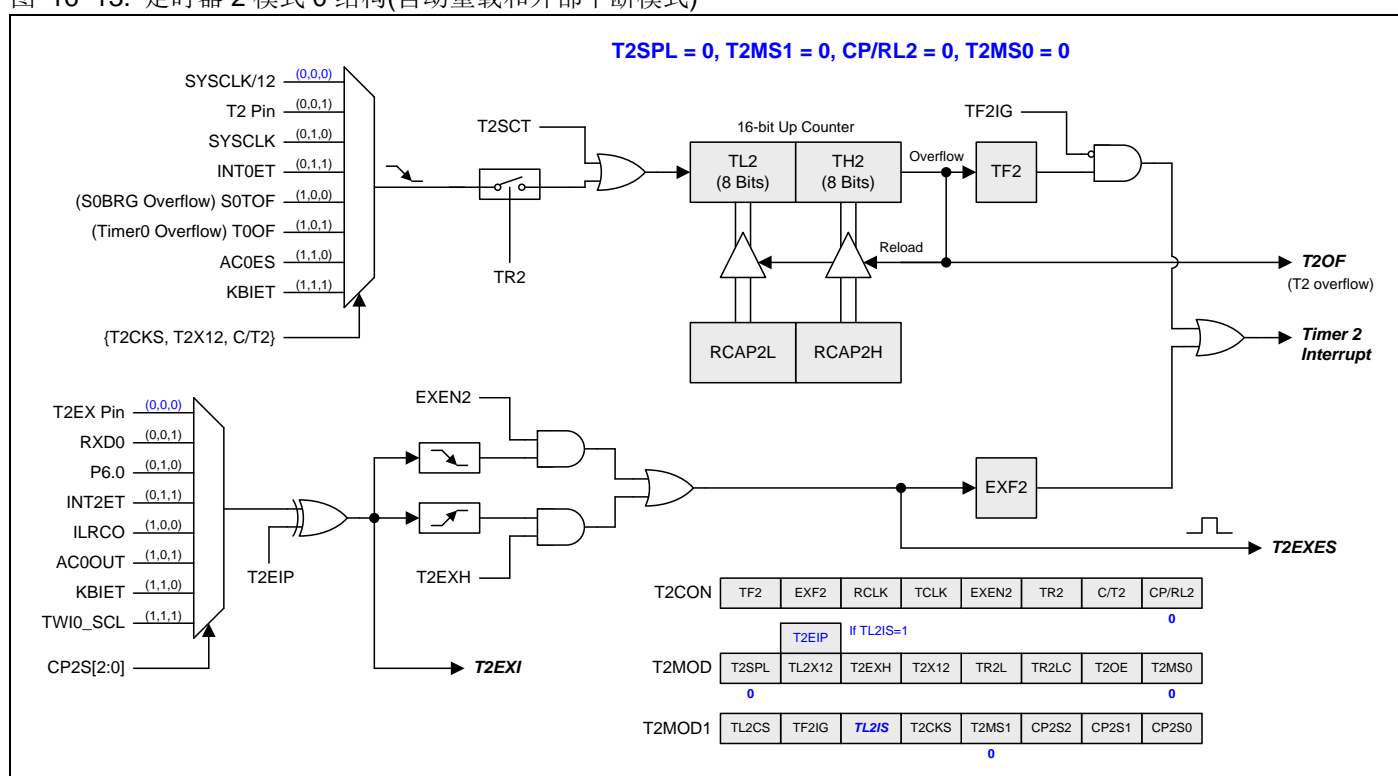
### 16.2.1. 定时器 2 模式 0 (自动重载和外部中断)

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

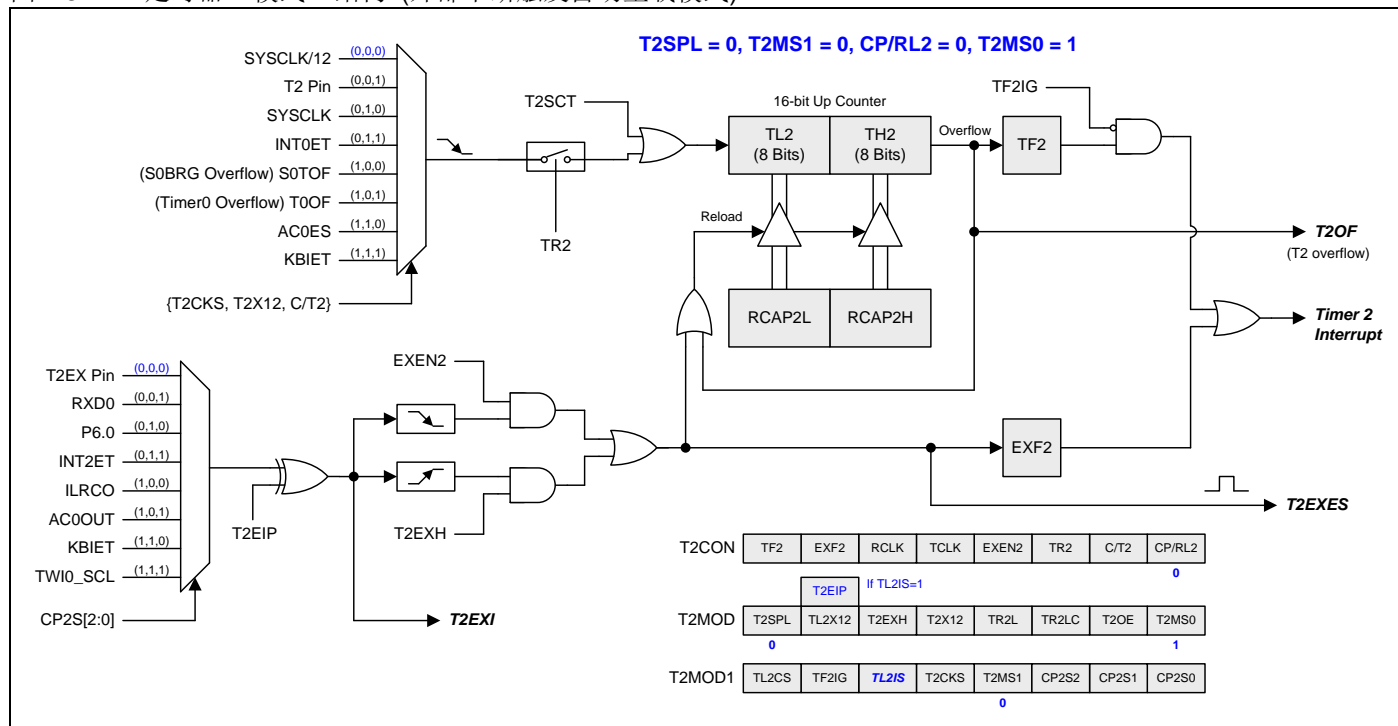
图 16-13. 定时器 2 模式 0 结构(自动重载和外部中断模式)



## 16.2.2. 定时器 2 模式 1 (外部中断触发外部中断)

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一个, 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。

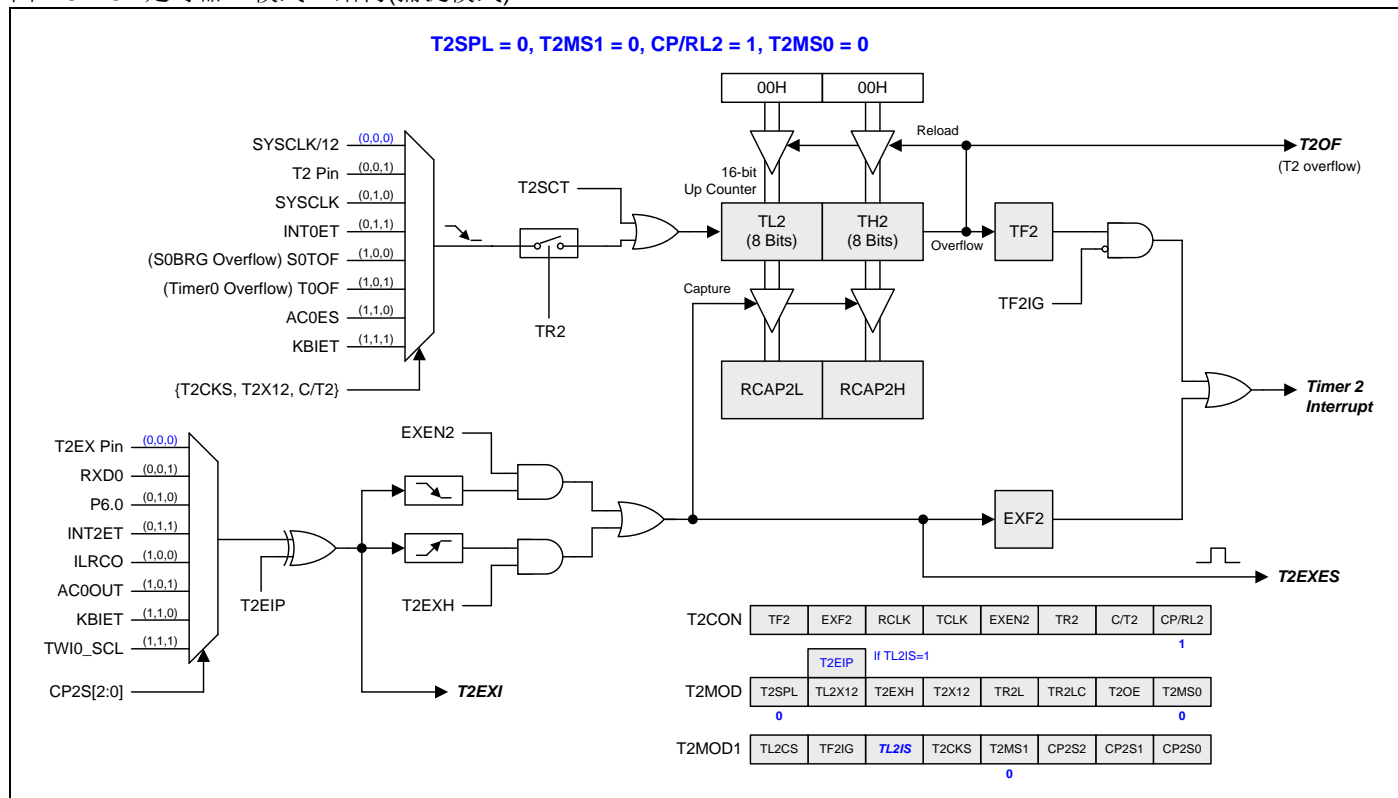
图 16-14. 定时器 2 模式 1 结构 (外部中断触发自动重载模式)



## 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)

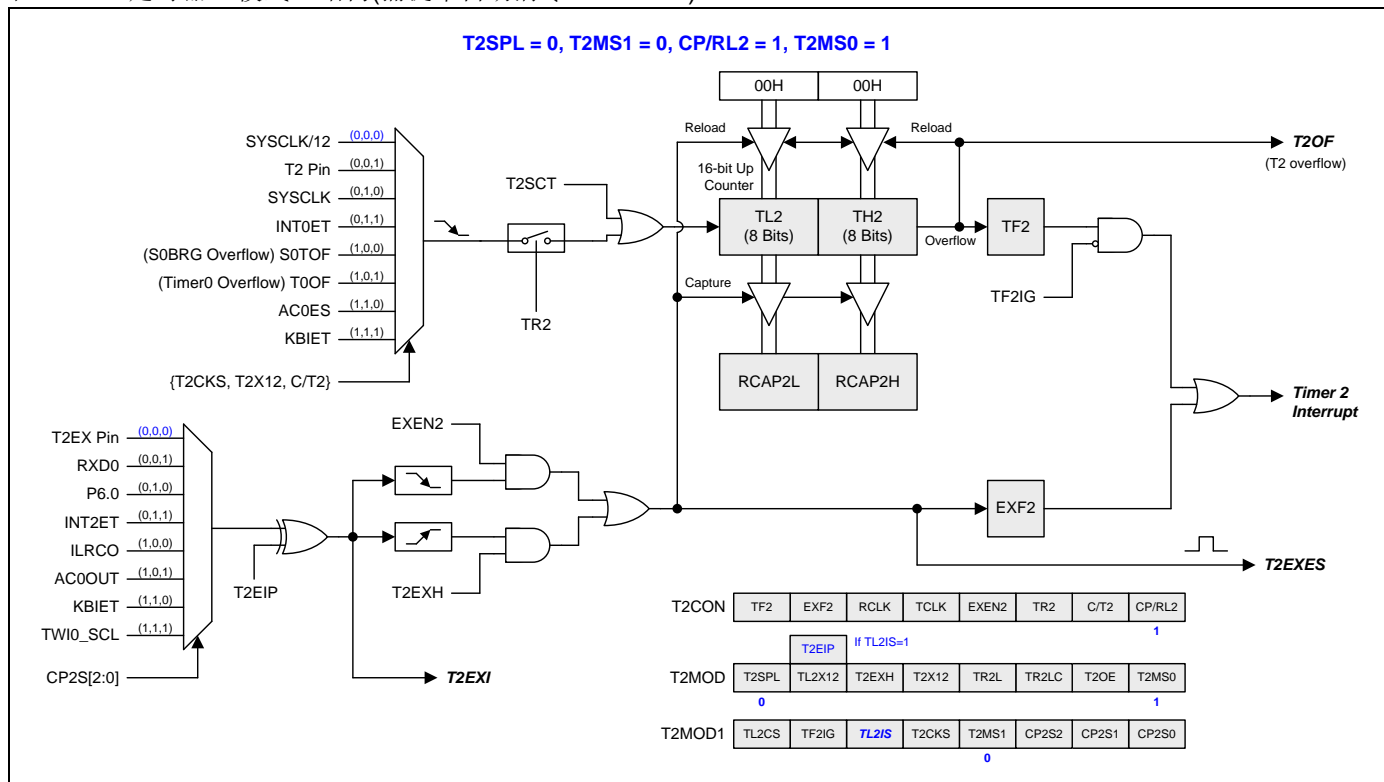


### 16.2.4. 定时器 2 模式 3 (捕捉带自动清零)

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&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。

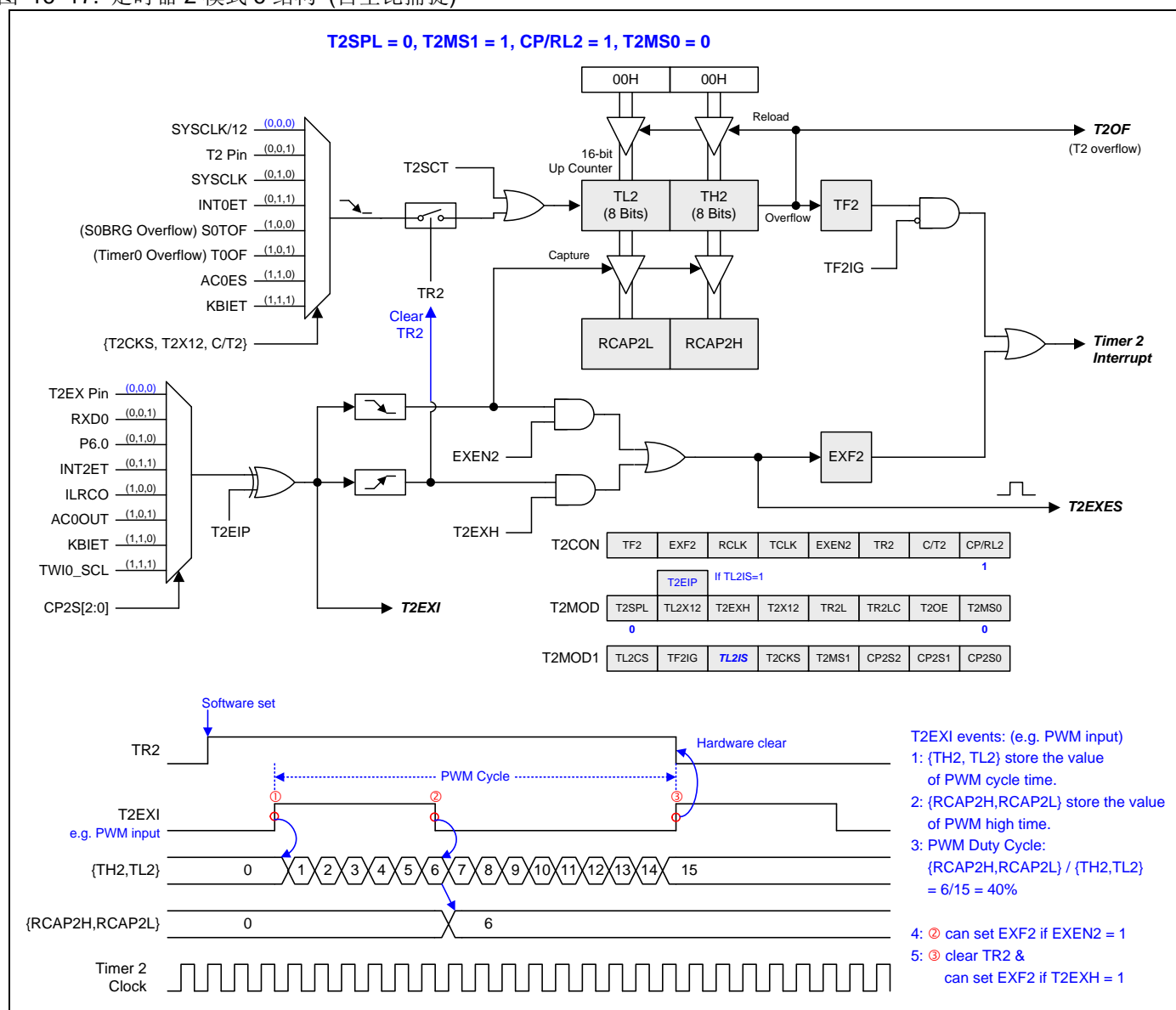
但是如果你想获得周期时间，那么就要通过清除EXEN2去阻止第二边沿。

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)

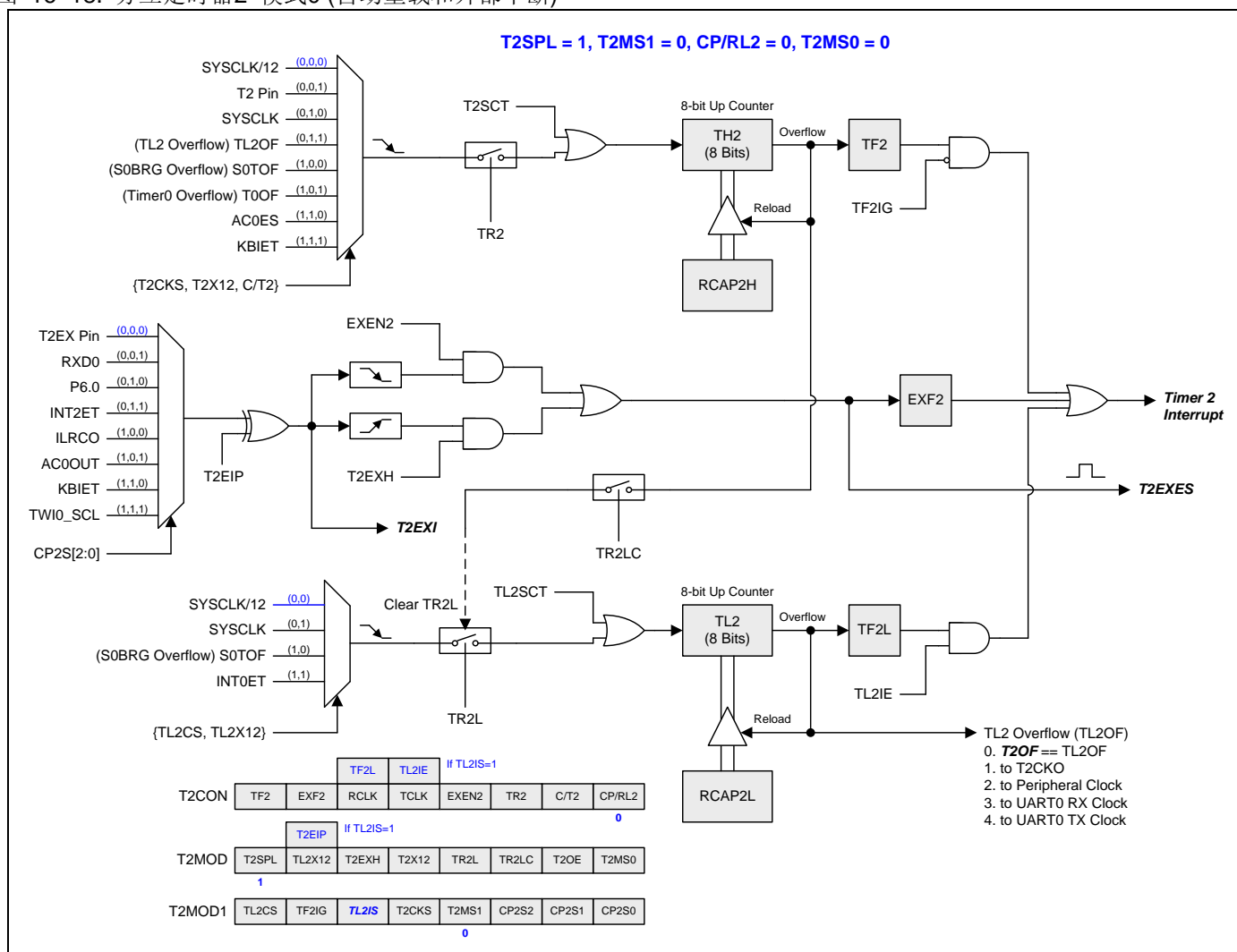
本模式中T2SPLIT置位，定时器2 变为两个8位定时器(TH2和TL2)。两个8位定时器都是向上计数如图 16-18所示。TH2保存RCAP2H的重载值和保持16位模式一样的8个时钟源输入选择。8位定时器功能跟16位模式的定时器2模式0相似。TL2保存4个时钟输入选择的RCAP2L重载值。T2CON的位TR2控制着TH2的运行。T2MOD的位TR2L控制着TL2的运行。当TR2LC置位时TH2溢出会停止TR2L的运行。

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。

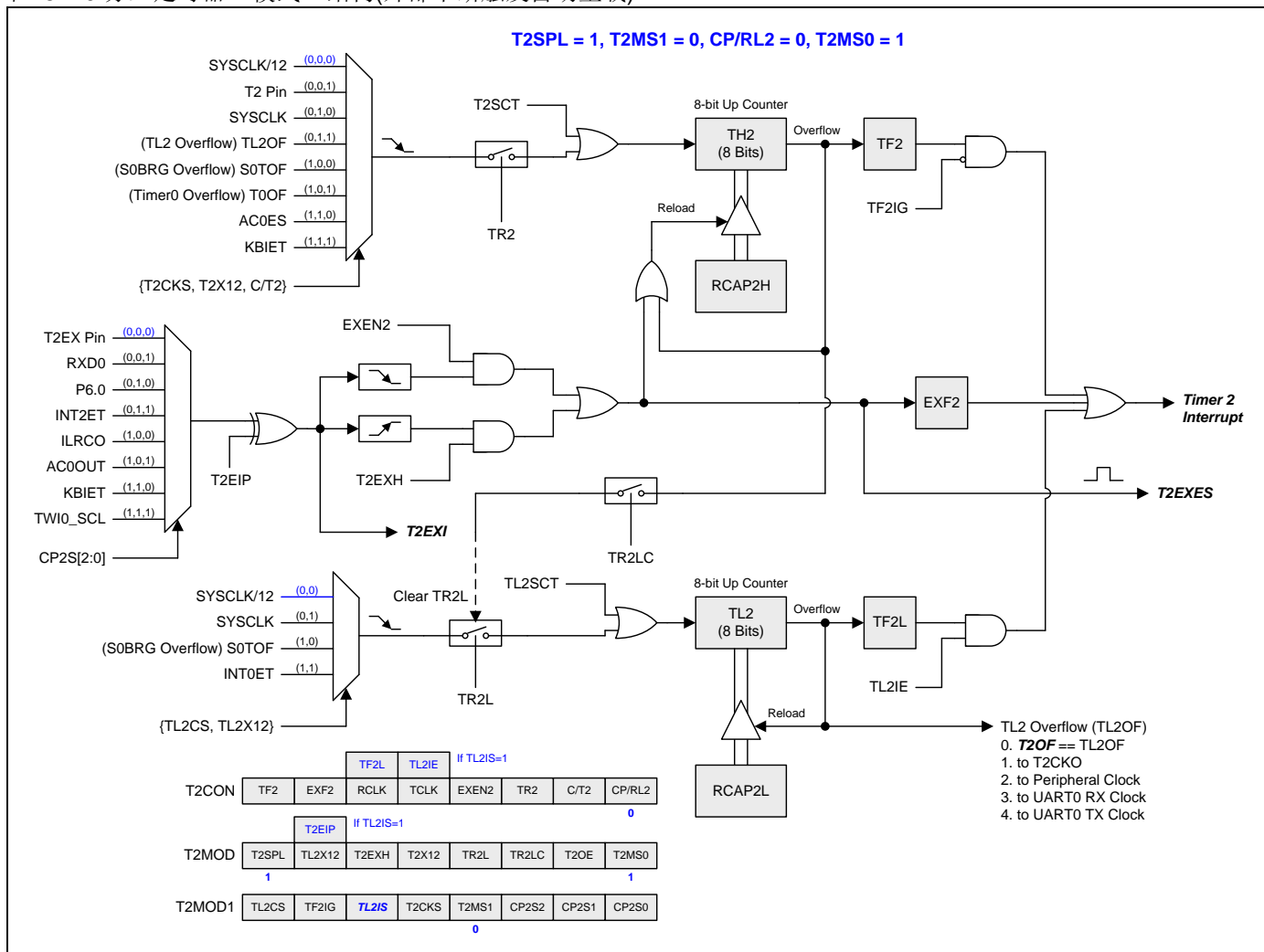
图 16-18. 分立定时器2 模式0 (自动重载和外部中断)



### 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。

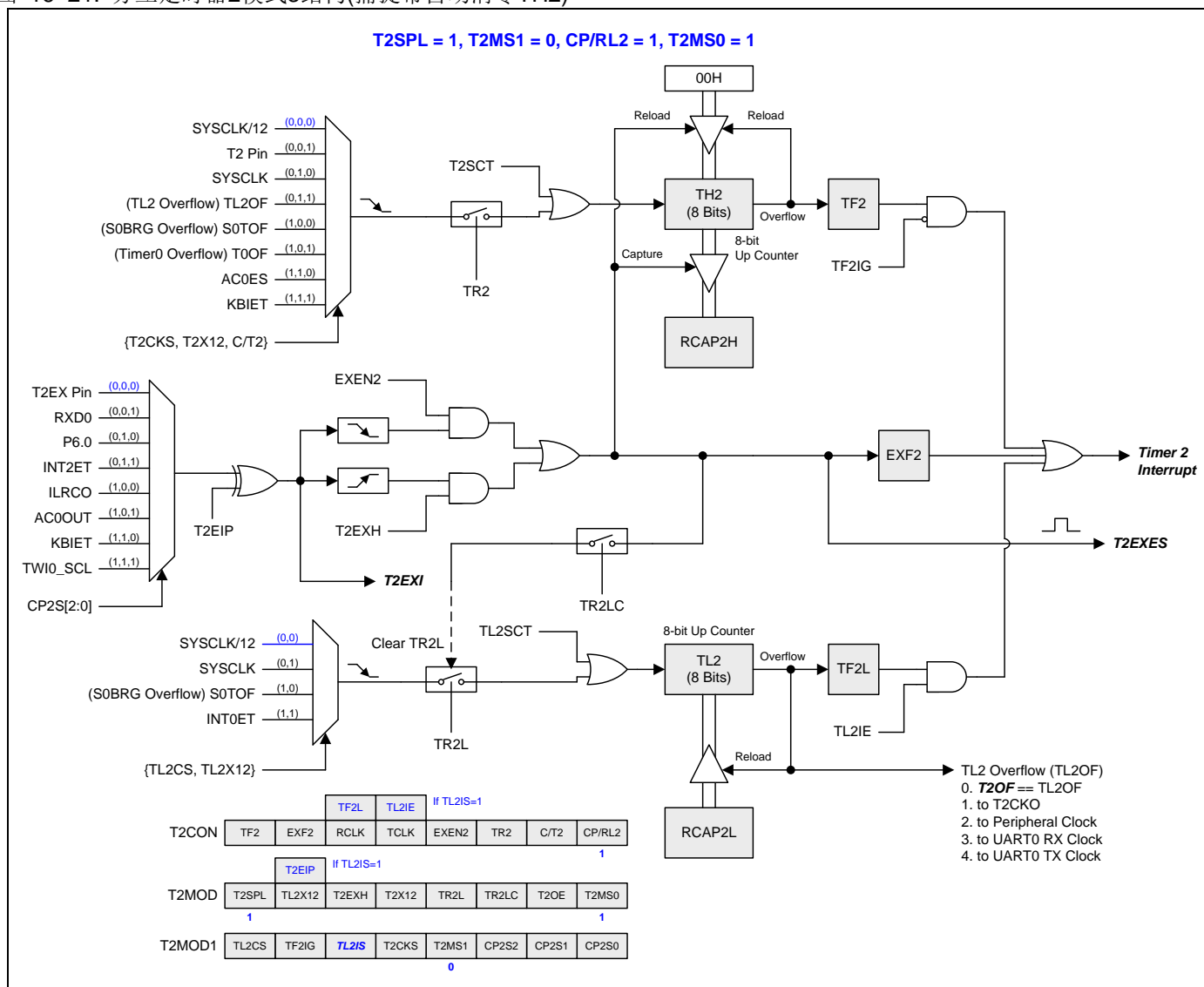
图 16-20. 分立定时器 2 模式 2 结构 (捕捉)



### 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。

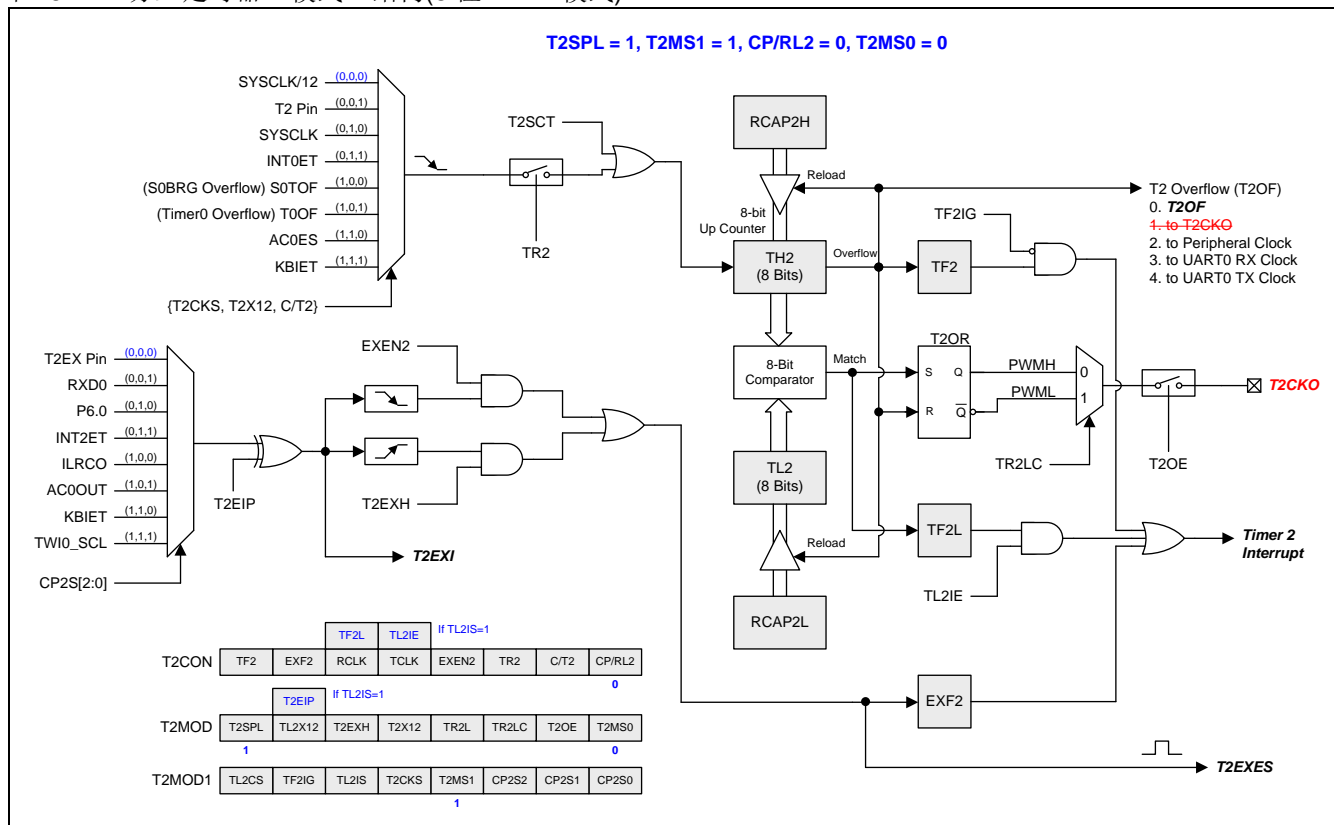
图 16-21. 分立定时器2模式3结构(捕捉带自动清零TH2)



## 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。这两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. 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 16-23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 18-6.)。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“18.8.4 模式1 & 3”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 16-24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 16-23. 定时器2波特率发生器模式

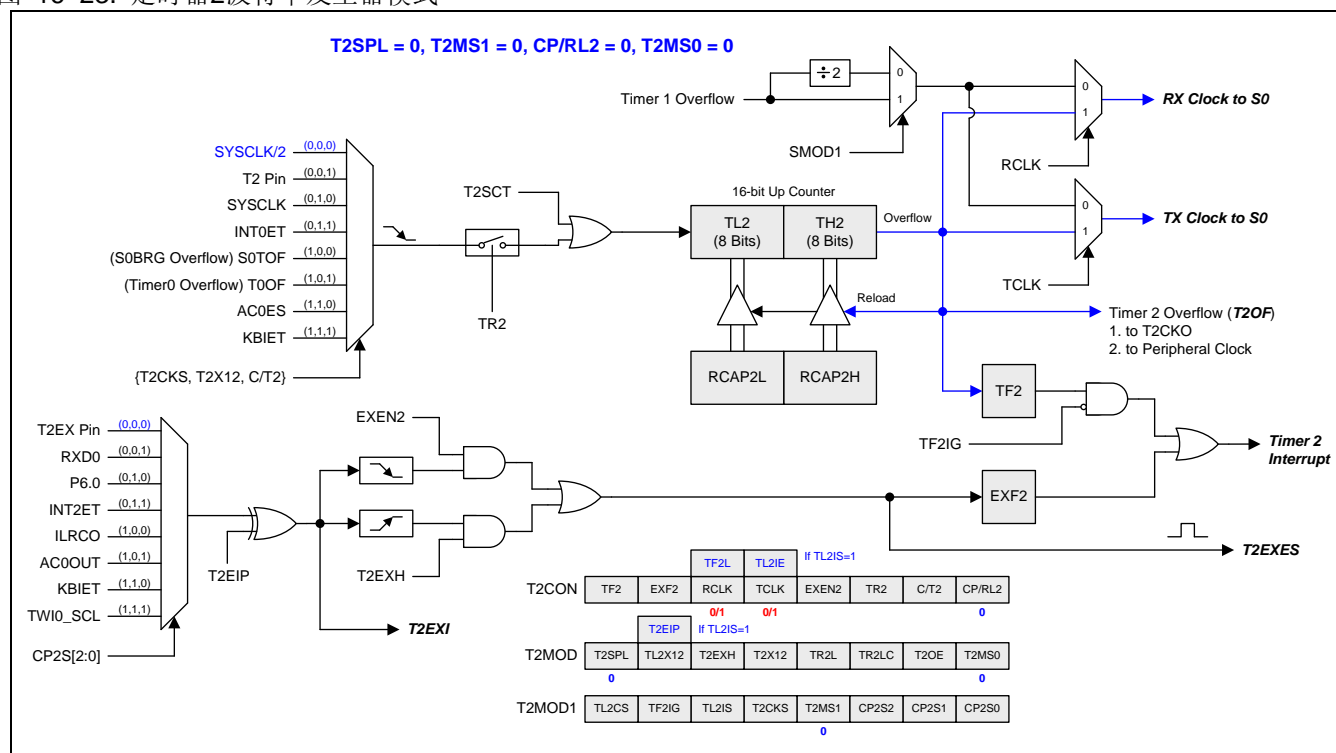
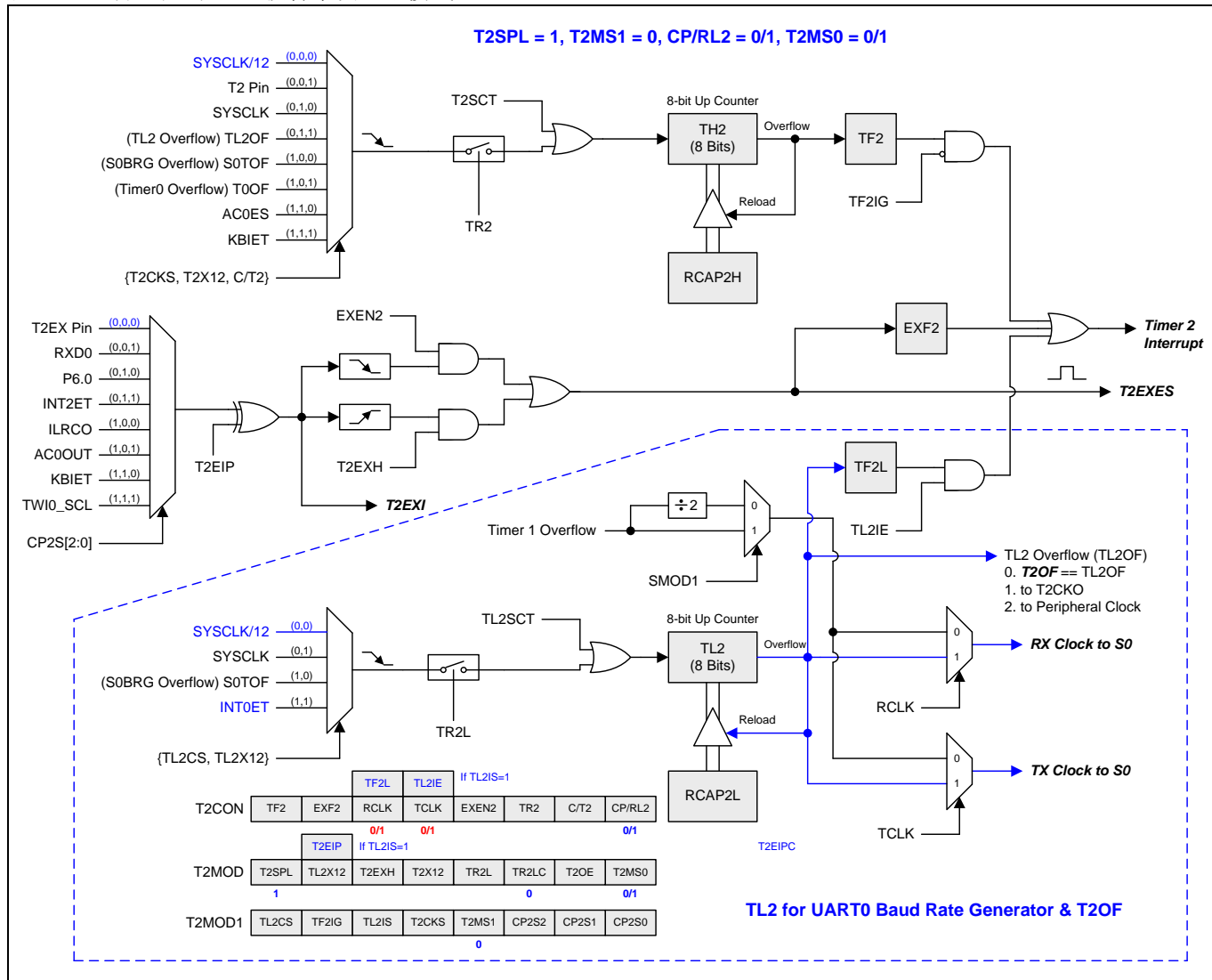


图 16-24. 分立定时器 2 波特率发生器模式



## 16.2.12. 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 16-25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 16-26所示。

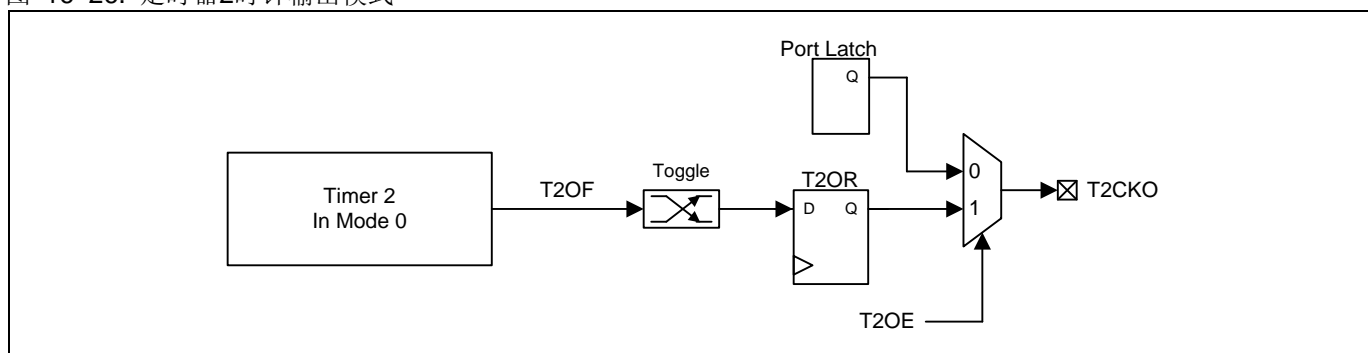
图 16-25. 定时器2时钟输出公式

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

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 16-26. 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的溢出速率来决定且不产生中断。

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 16-27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 16-28所示。

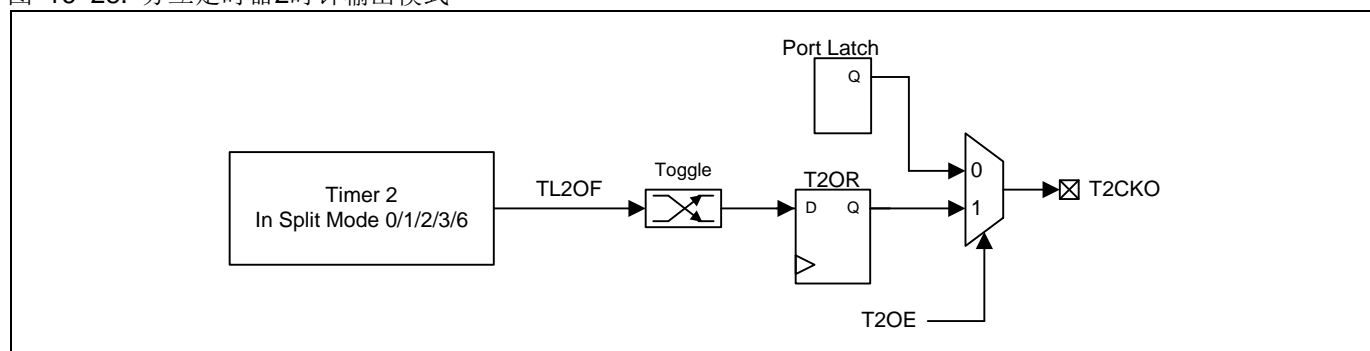
图 16-27. 分立定时器 2 时钟输出公式

$$\text{Split T2 Clock-out Frequency} = \frac{\text{TL2 Clock Frequency}}{2 \times (256 - \text{RCAP2L})}$$

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 16-28. 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

- 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。

## 16.2.13. 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	INTOET

**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
Others	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。

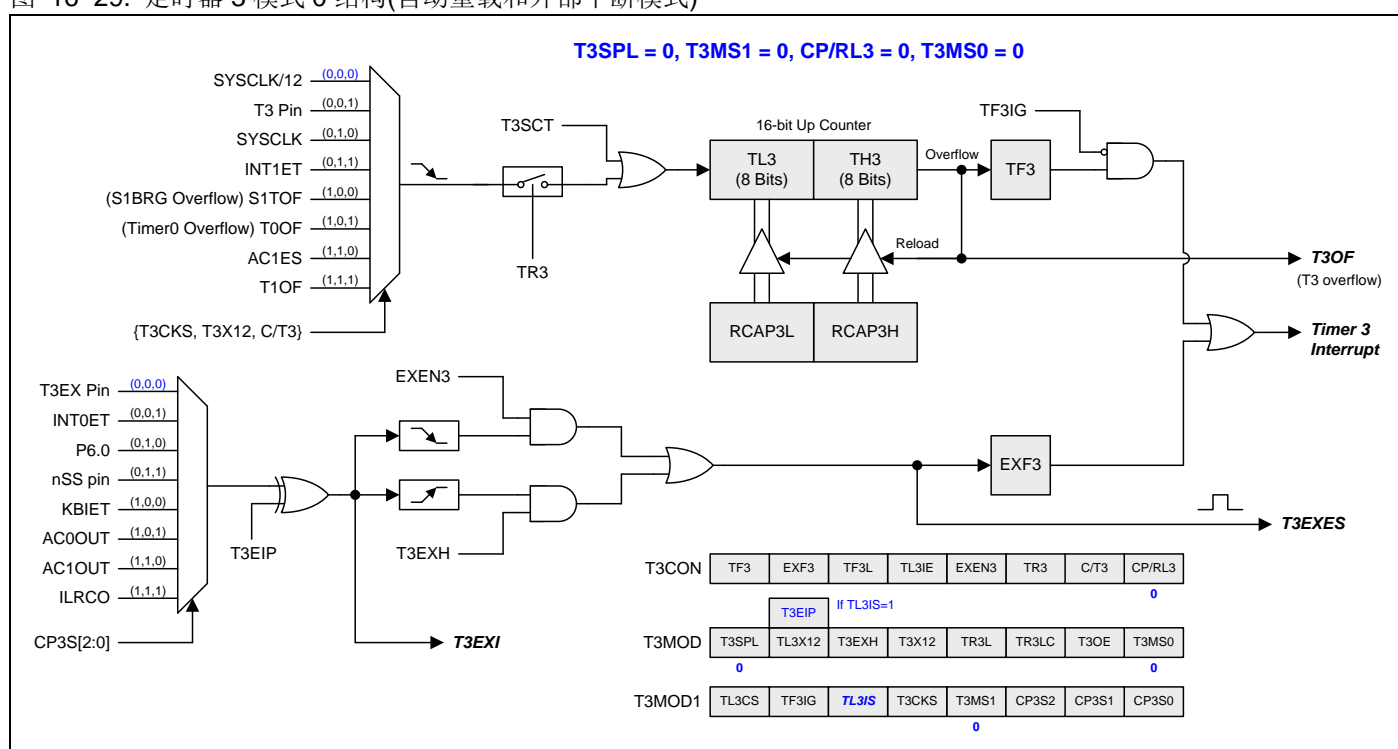
### 16.3.1. 定时器 3 模式 0 (自动重载和外部中断)

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 16-29所示

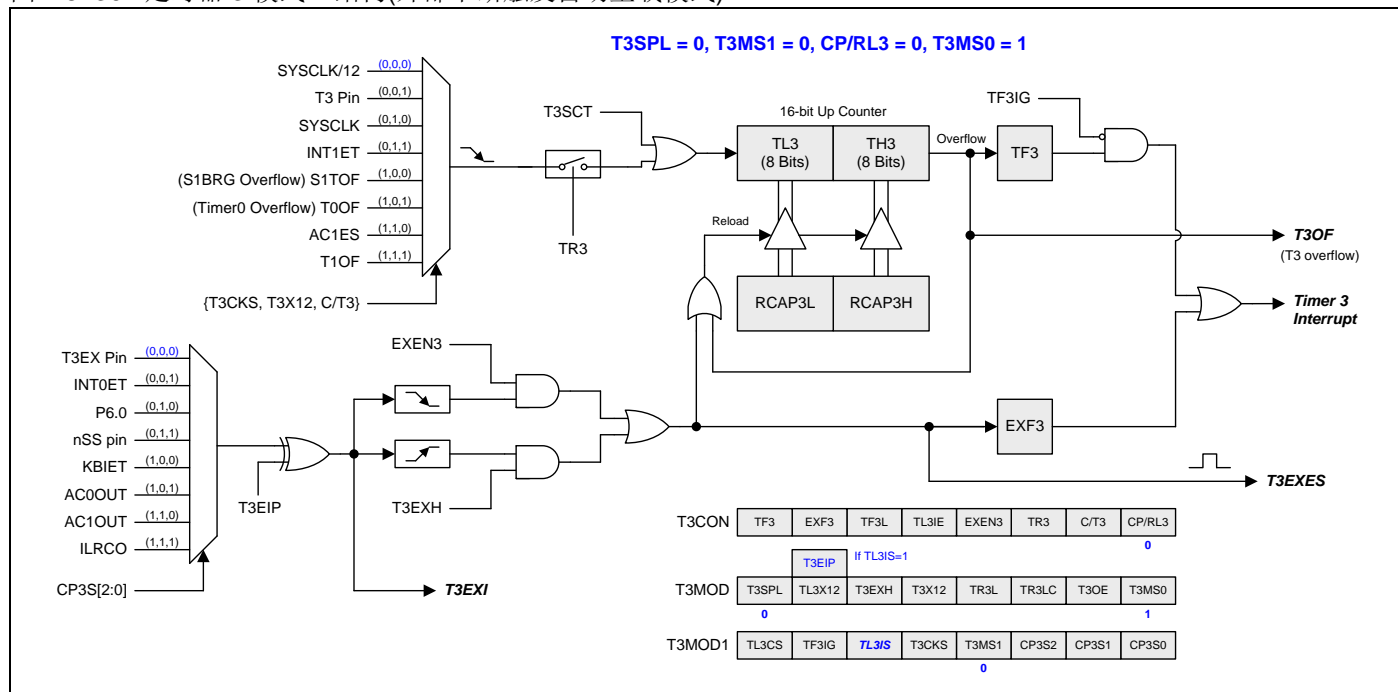
图 16-29. 定时器 3 模式 0 结构(自动重载和外部中断模式)



## 16.3.2. 定时器 3 模式 1(外部中断触发自动重载)

如图 16-30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。

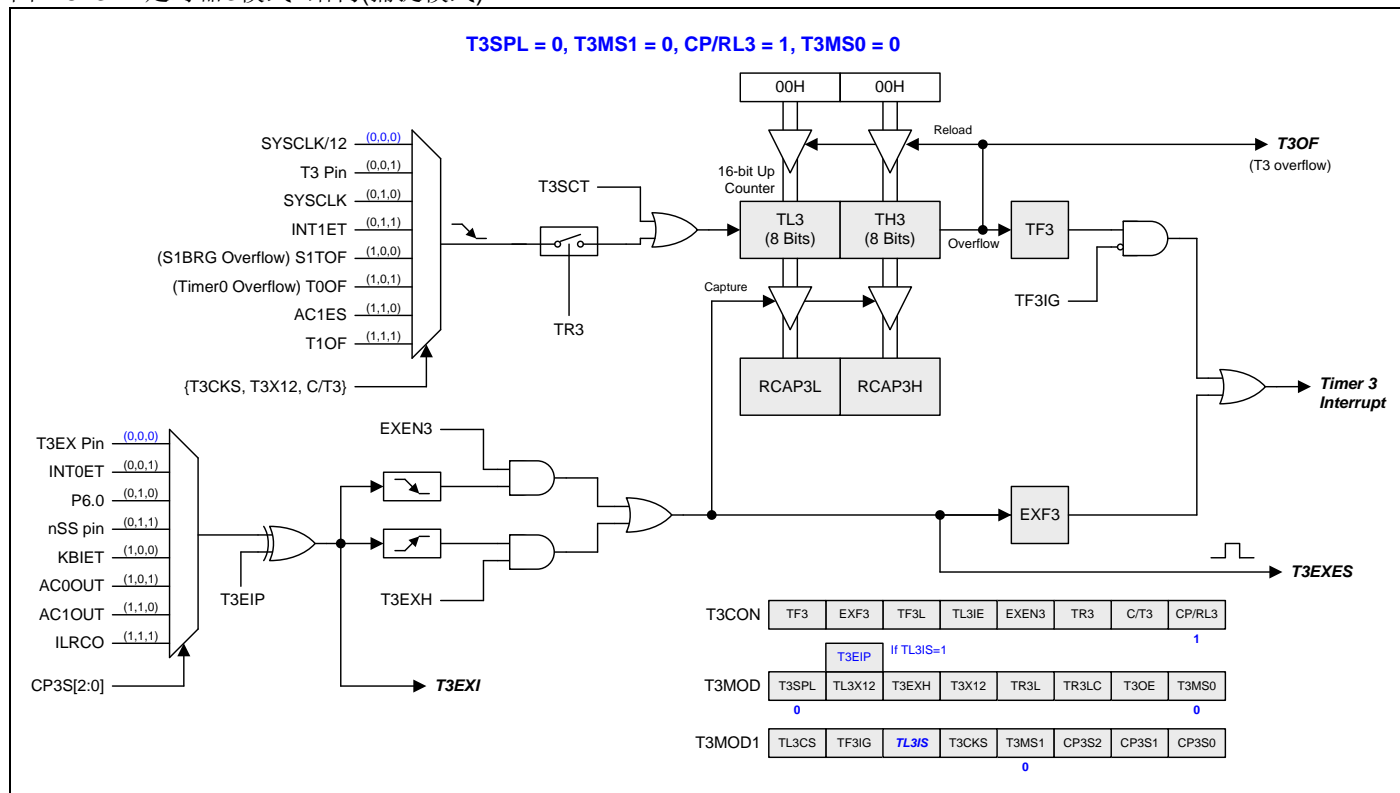
图 16-30. 定时器 3 模式 1 结构(外部中断触发自动重载模式)



## 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 16-31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 16-31. 定时器3模式2结构(捕捉模式)

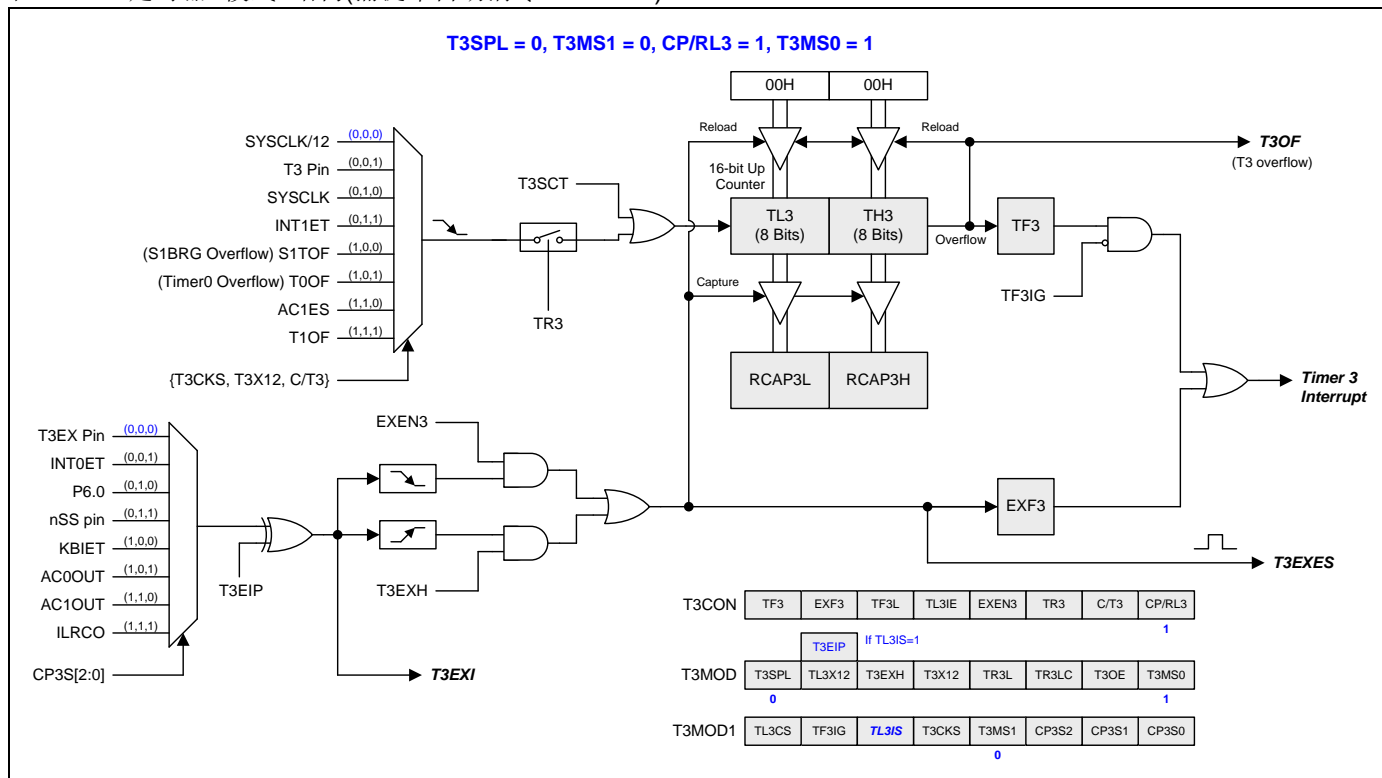


### 16.3.4. 定时器3 模式3(捕捉带自动清零)

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 16-32所示

图 16-32. 定时器3模式3结构(捕捉带自动清零TL3&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。

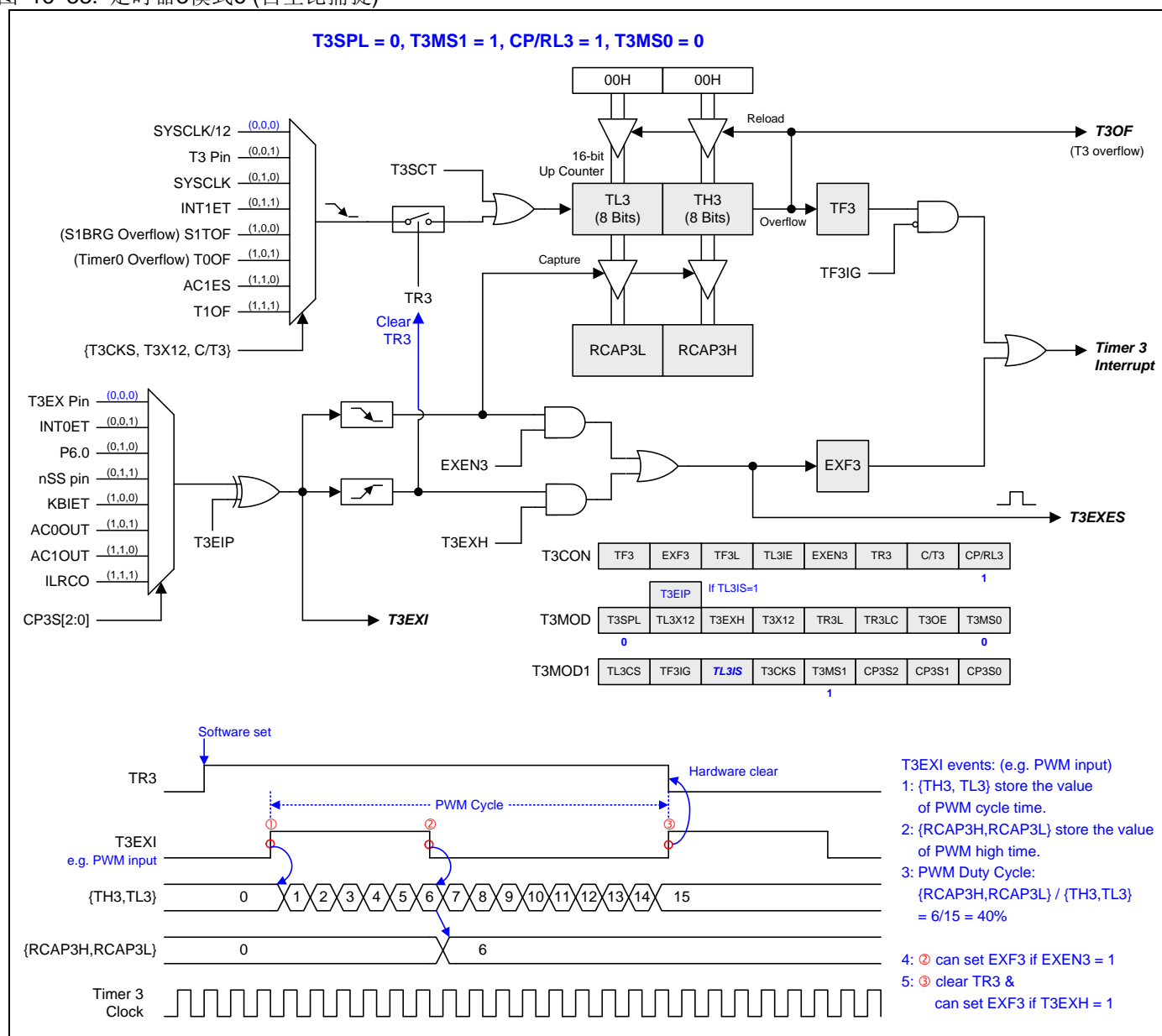
但是如果你想获得周期时间，那么就要通过清除EXEN3去阻止第二边沿。

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 16-33所示

图 16-33. 定时器3模式6 (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)

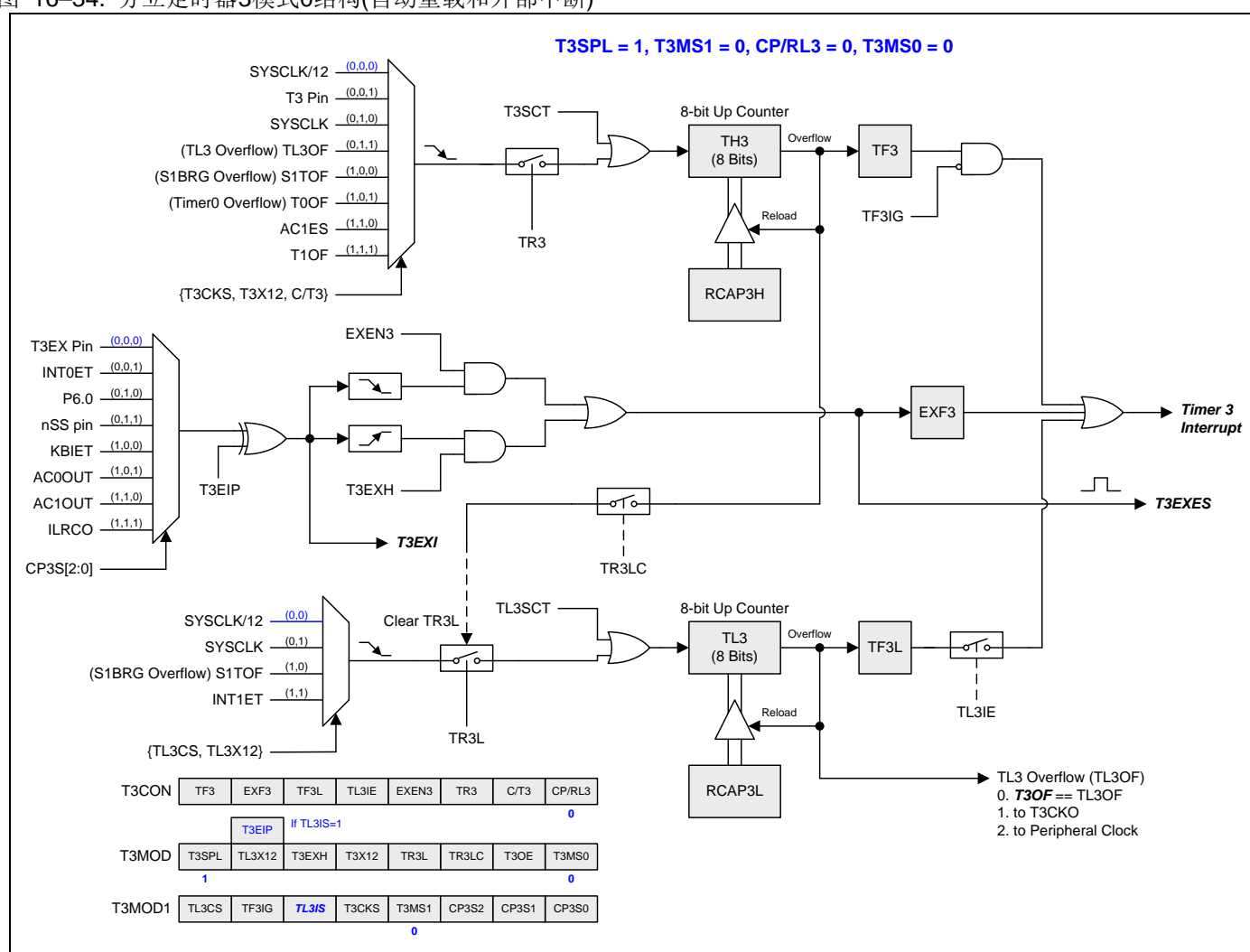
本模式中T3SPLIT置位, 定时器3 变为两个8位定时器 (TH3和TL3)。两个8位定时器都是向上计数如图 16-34所示。TH3 保存RCAP3H的重载值和保持16位模式一样的8个时钟源输入选择。8位定时器功能跟16位模式的定时器3模式0相似。TL3 保存4个时钟输入选择的RCAP3L重载值。T3CON的位TR3控制着TH3的运行。T3MOD的位TR3L控制着TL3的运行。当TR3LC置位时TH3溢出会停止TR3L的运行。

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。

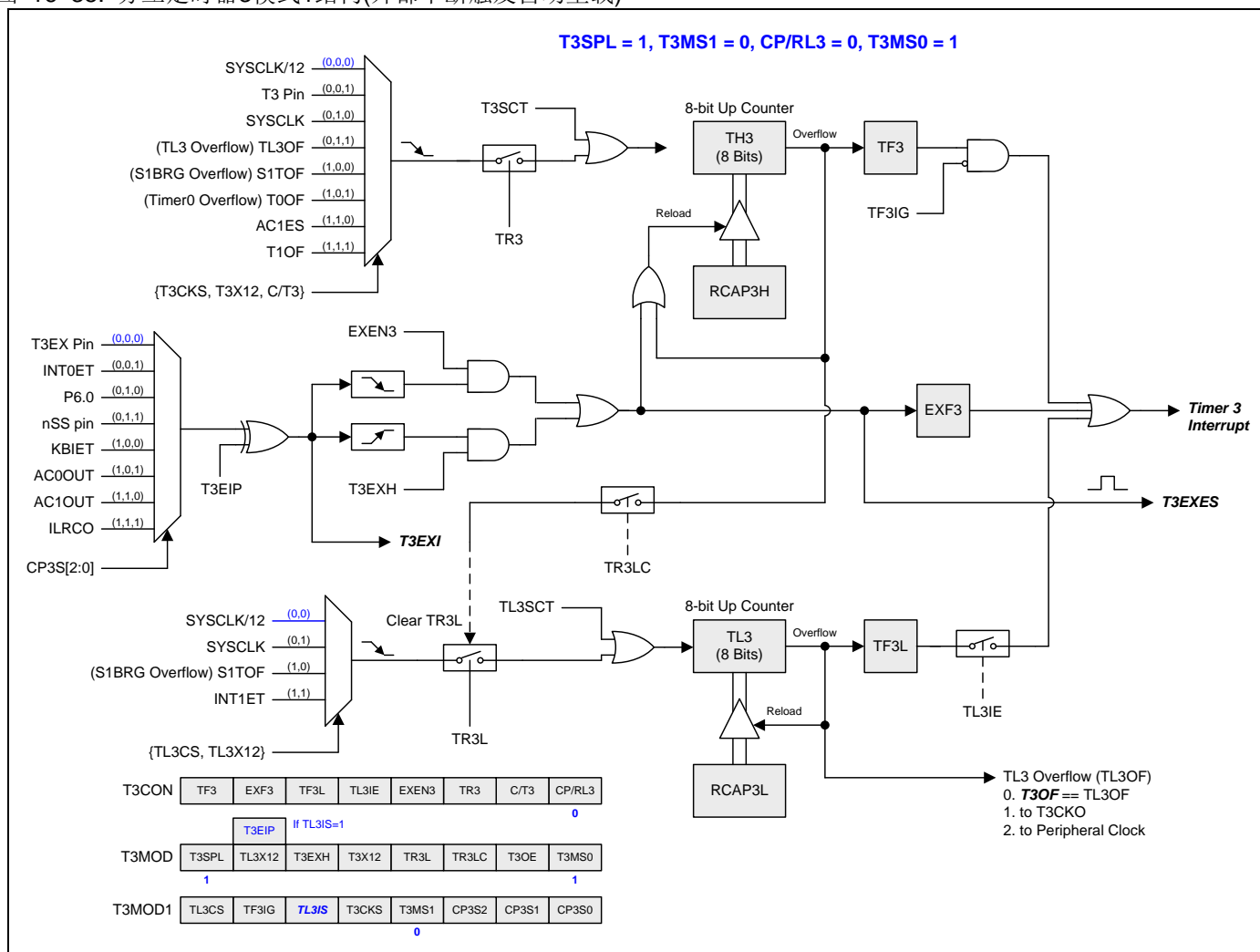
图 16-34. 分立定时器3模式0结构(自动重载和外部中断)



### 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 16-35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。

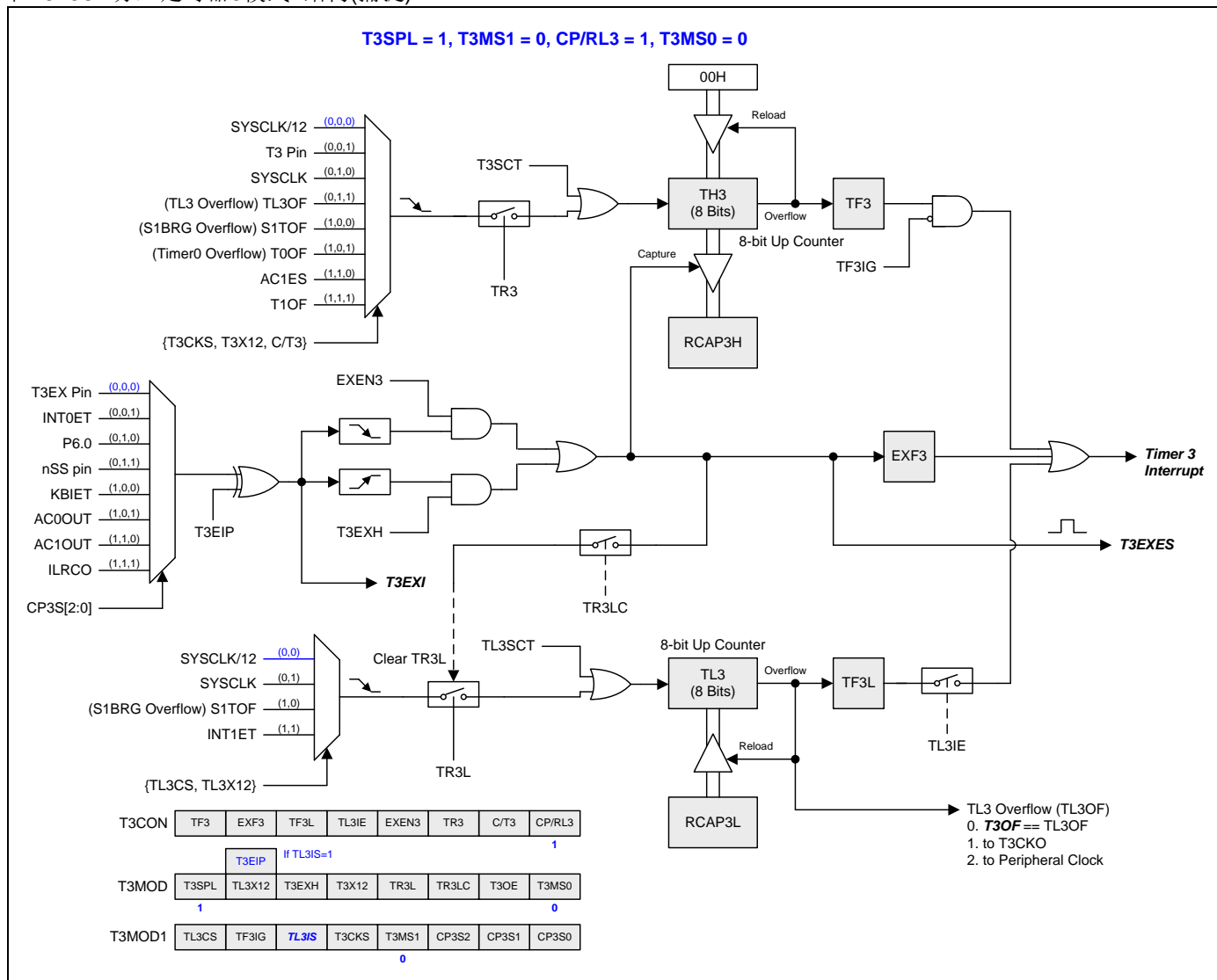
图 16-35. 分立定时器3模式1结构(外部中断触发自动重载)



### 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分为两个8位定时器如图 16-36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。

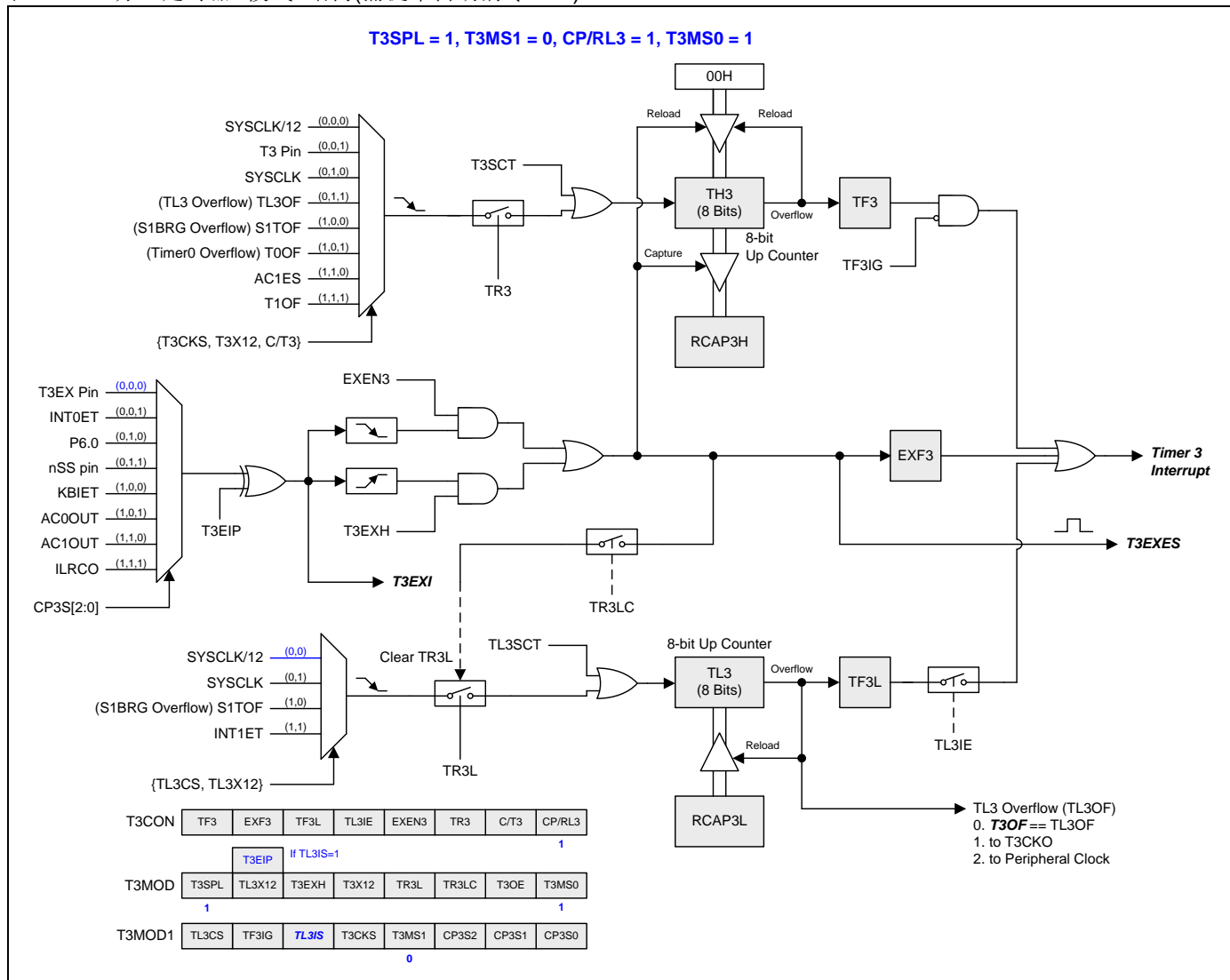
图 16-36. 分立定时器3模式2结构(捕捉)



### 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分列为两个8位定时器如图 16-37.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。

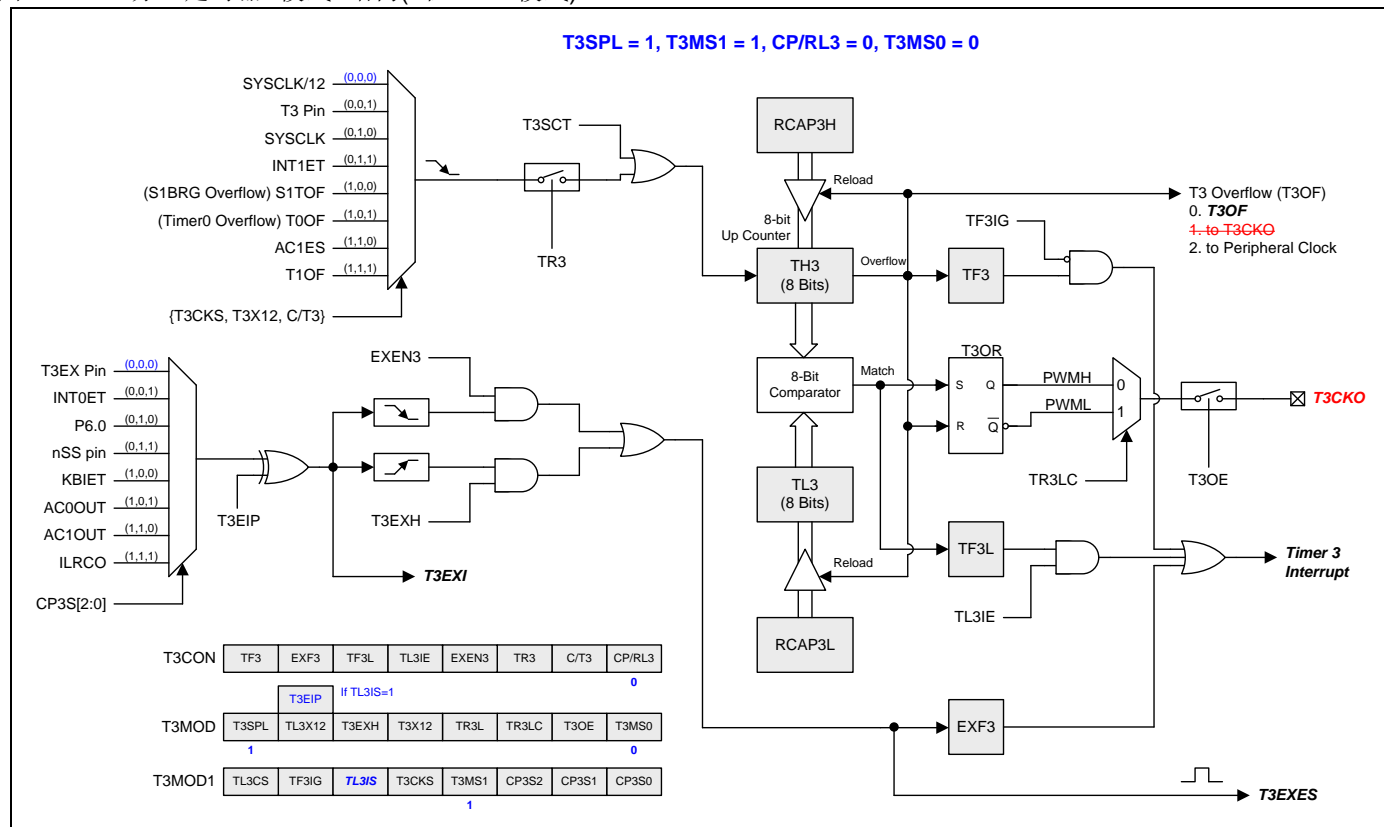
图 16-37. 分立定时器3模式3结构(捕捉带自动清零TH3)



## 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 16-38所示。TH3和RCAP3H相结合为一个8位的自动重载计数器。这两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 16-38. 分立定时器3模式4结构(8位PWM模式)



### 16.3.11. 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 16-39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 16-40所示。

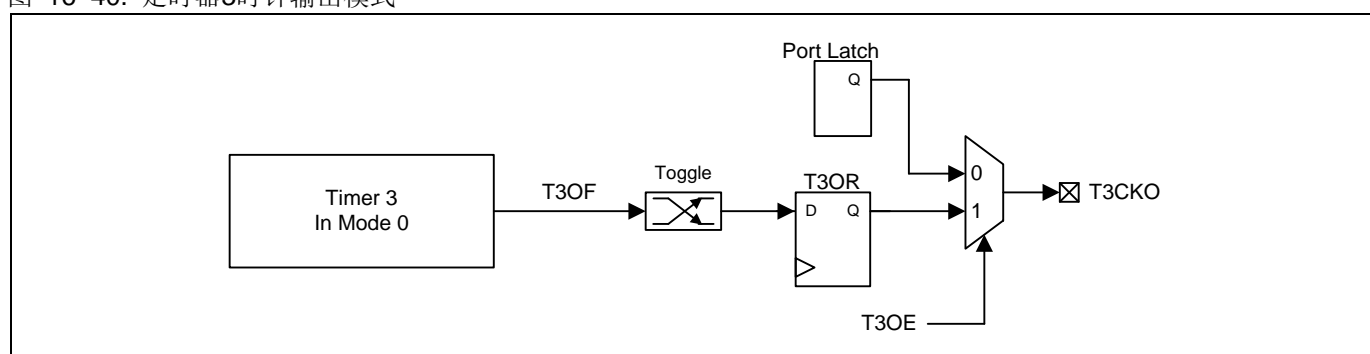
图 16-39. 定时器3时钟输出公式

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

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 16-40. 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 16-41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 16-42所示。

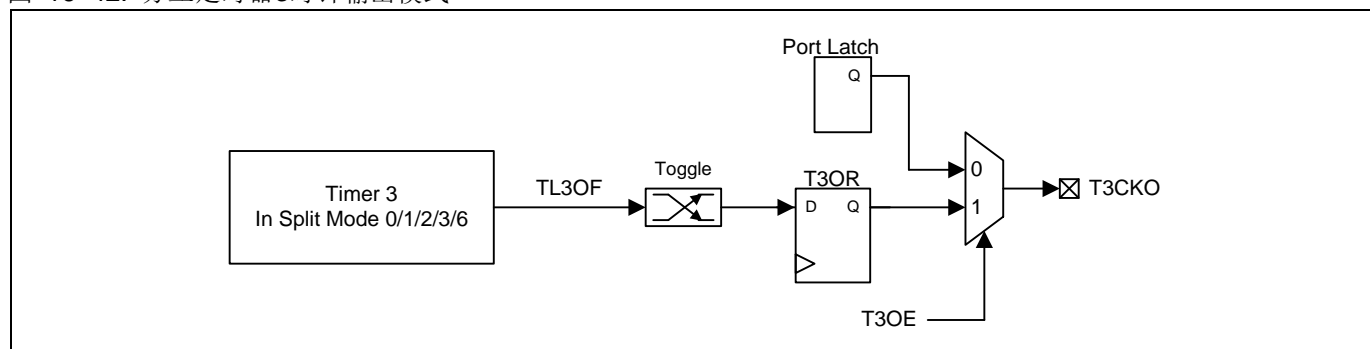
图 16-41. 分立定时器3时钟输出公式

$$\text{Split T3 Clock-out Frequency} = \frac{\text{TL3 Clock Frequency}}{3 \times (256 - \text{RCAP3L})}$$

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 16-42. 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。

## 16.3.12. 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引脚	T3引脚
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
其它	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	T3CKS	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 = 1100-0000

7	6	5	4	3	2	1	0
POE7	POE6	C0PPS2	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	W	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	P4.6	P0.0
0 1	P3.3	P3.2
1 0	P2.1	P2.0
1 1	P6.6	P6.5

## 16.4. 定时器 4

定时器4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T4CKS, T4X12和C/T4选择。定时器4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T4CON、T4MOD和T4MOD1寄存器中的位来选择的。

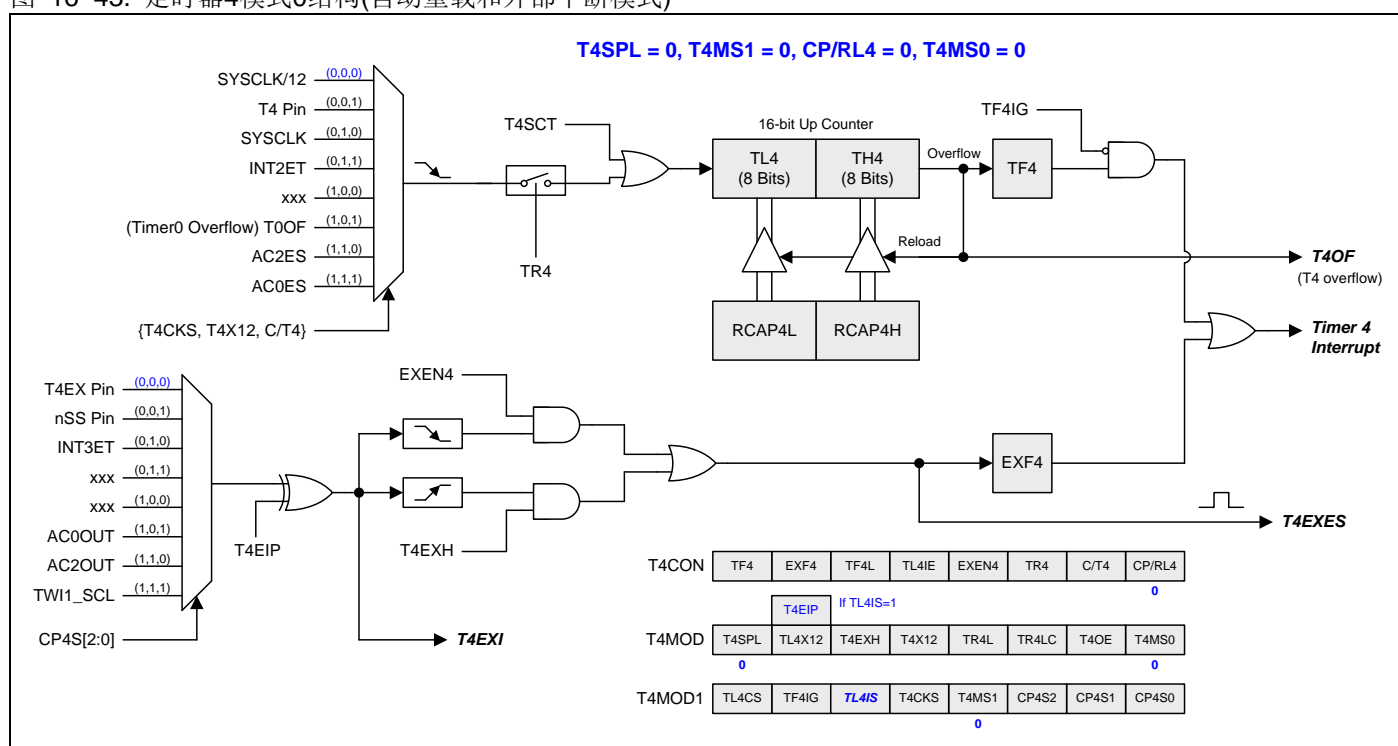
### 16.4.1. 定时器 4 模式 0(自动重载和外部中断)

在这个模式中，定时器4提供一个16位的自动重载定时器/计数器。TF4是定时器4的溢出标志，是一个中断会被TF4IG阻断的定时器4中断源。EXEN4使能T4EXI引脚的下降沿置位EXF4，EXF4可作为一个外部中断与TF4共享定时器4中断。T4EXI由8个外部触发输入中选出一个。T4EXH的功能与EXEN4一样，只是T4EXH使能T4EXI引脚的上升沿置位EXF4。

本模块中的定时器4溢出事件(T4OF)可以作为时钟输入或事件源输出到其它外设。

定时器4模式0如图 16-43所示

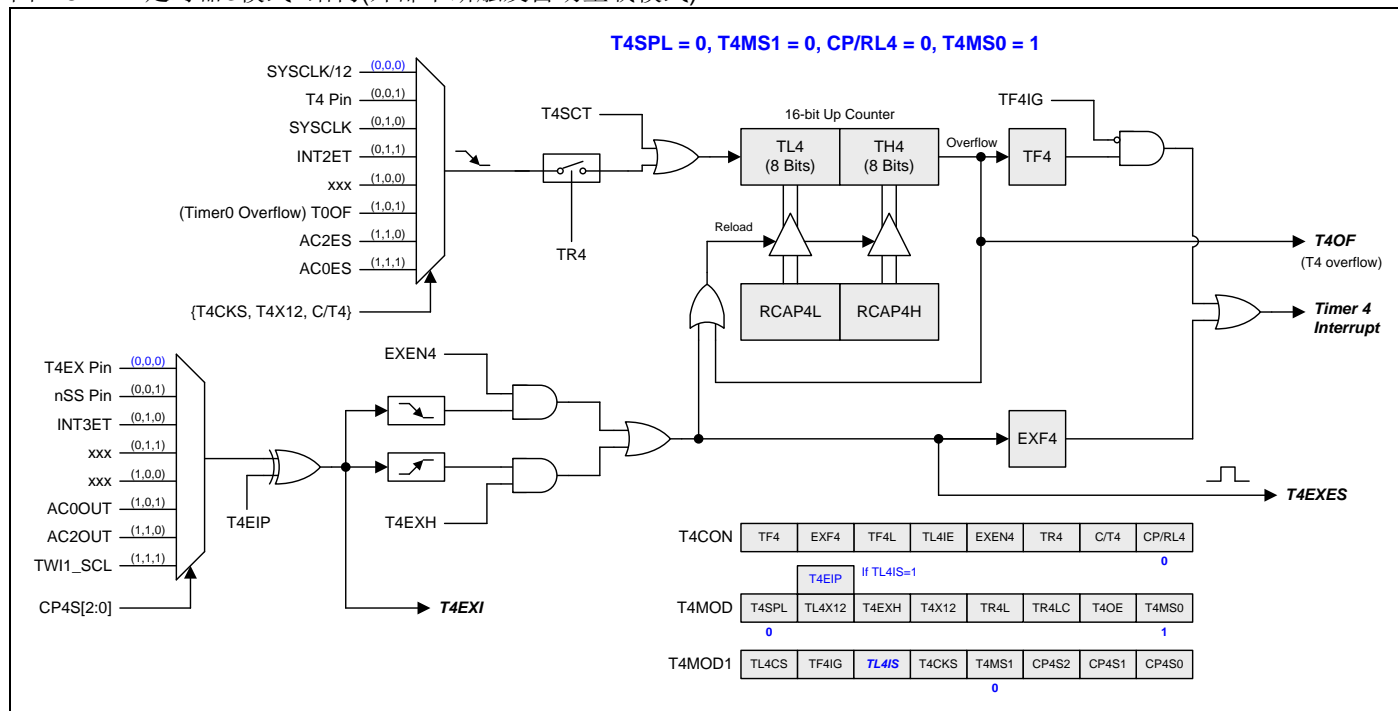
图 16-43. 定时器4模式0结构(自动重载和外部中断模式)



## 16.4.2. 定时器 4 模式 1(外部中断触发自动重载)

如图 16-44所示, 定时器4模式1, 使能定时器4自动向上计数。本模式有T4CON寄存器的EXEN4决定的两种选择。如果EXEN4=0, 定时器4向上计数到0xFFFFH且直到溢出而置位TF4(溢出标志)。这时定时器4寄存器被重载入RCAP4L和RCAP4H的16位数据。RCAP4L和RCAP4H的值由软件预设。如果EXEN4=1, 16位重载会被一个T4溢出或一个T4EXI的下降沿触发, T4EXI由8个外部触发输入中选出一个。此触发同时置位EXF4。如果定时器4中断使能, 无论TF4或EXF4置位则产生中断。T4EXH的功能与EXEN4一样, 只是T4EXH使能T4EXI引脚的上升沿置位EXF4。

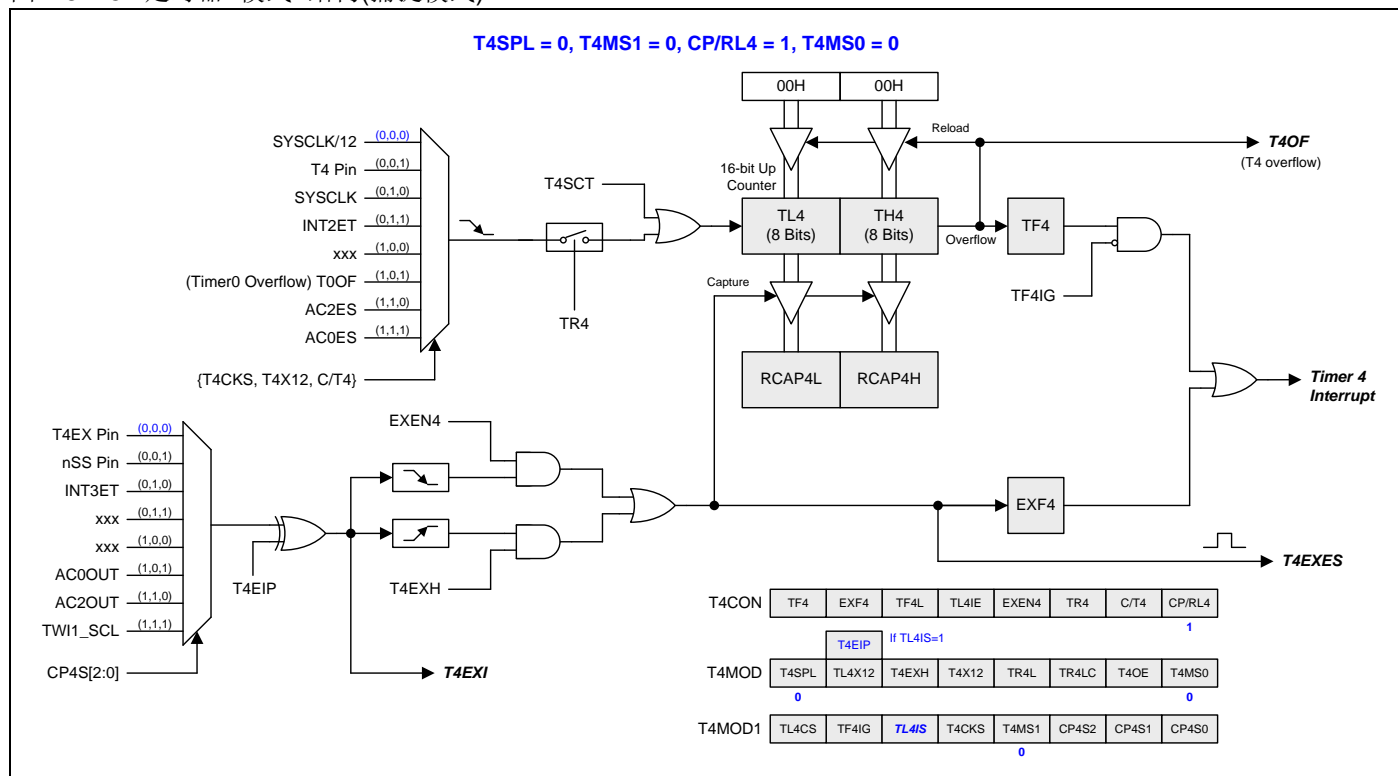
图 16-44. 定时器3模式1结构(外部中断触发自动重载模式)



## 16.4.3. 定时器 4 模式 2(捕捉)

由T4CON寄存器的位EXEN4做二种选择的捕捉模式如图 16-45所示。如果EXEN4=0, 定时器4是一个16位定时器或计数器, 向上溢出而置位TF4(定时器4溢出标志)。TF4常用来产生中断(IE寄存器使能定时器4中断相关位)。模式2增加了捕捉功能, 如果EXEN4=1, T4EXI引脚下降沿(由8个外部触发输入中选出)把定时器4寄存器(TH4和TL4)各自捕捉到寄存器(RCAP4H和RCAP4L)。另外, T4EXI引脚跳变使T4CON寄存器的EXF4置位, 且EXF4跟TF4共享中断位置。T4EXH的功能与EXEN4一样, 只是T4EXH使能T4EXI引脚的上升沿置位EXF4。

图 16-45. 定时器4模式2结构(捕捉模式)

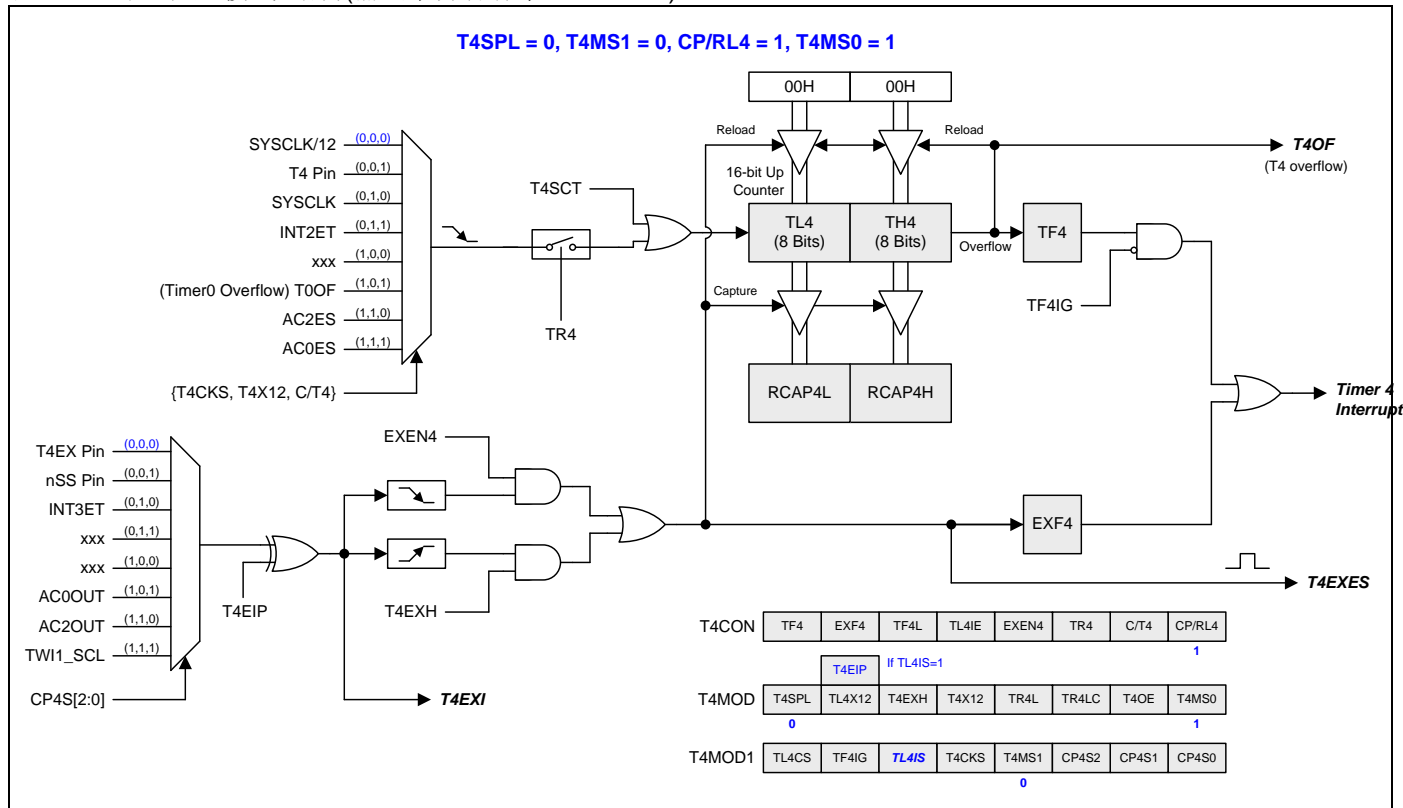


### 16.4.4. 定时器 4 模式 3(捕捉带自动清零)

定时器4模式3与定时器4模式2的功能相似。有一点不同的就是T4EXES、EXF4置位，不但定时器4会被捕捉而且TL4和TH4也会被清零。

定时器4模式3如图 16-46所示

图 16-46. 定时器4模式3结构(捕捉带自动清零TL4 & TH4)



### 16.4.5. 定时器 4 模式 6 (占空比捕捉)

定时器4 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TH4, TL4清零。然后通过置位TR4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T4EX 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，T4EXI的第一个触发边缘必须是上升沿。即使你设置了T4EXH，第一个边沿也会被忽略以触发EXF4。

其次，如果只想计算脉冲宽度，那么可以设置EXEN4通过第二个边来触发EXF4。在本例中，当第二个边缘触发计时器将其值捕捉到RCAP4H: RCAP4L中时，它还触发用于中断的EXF4，以标识它已经完成。

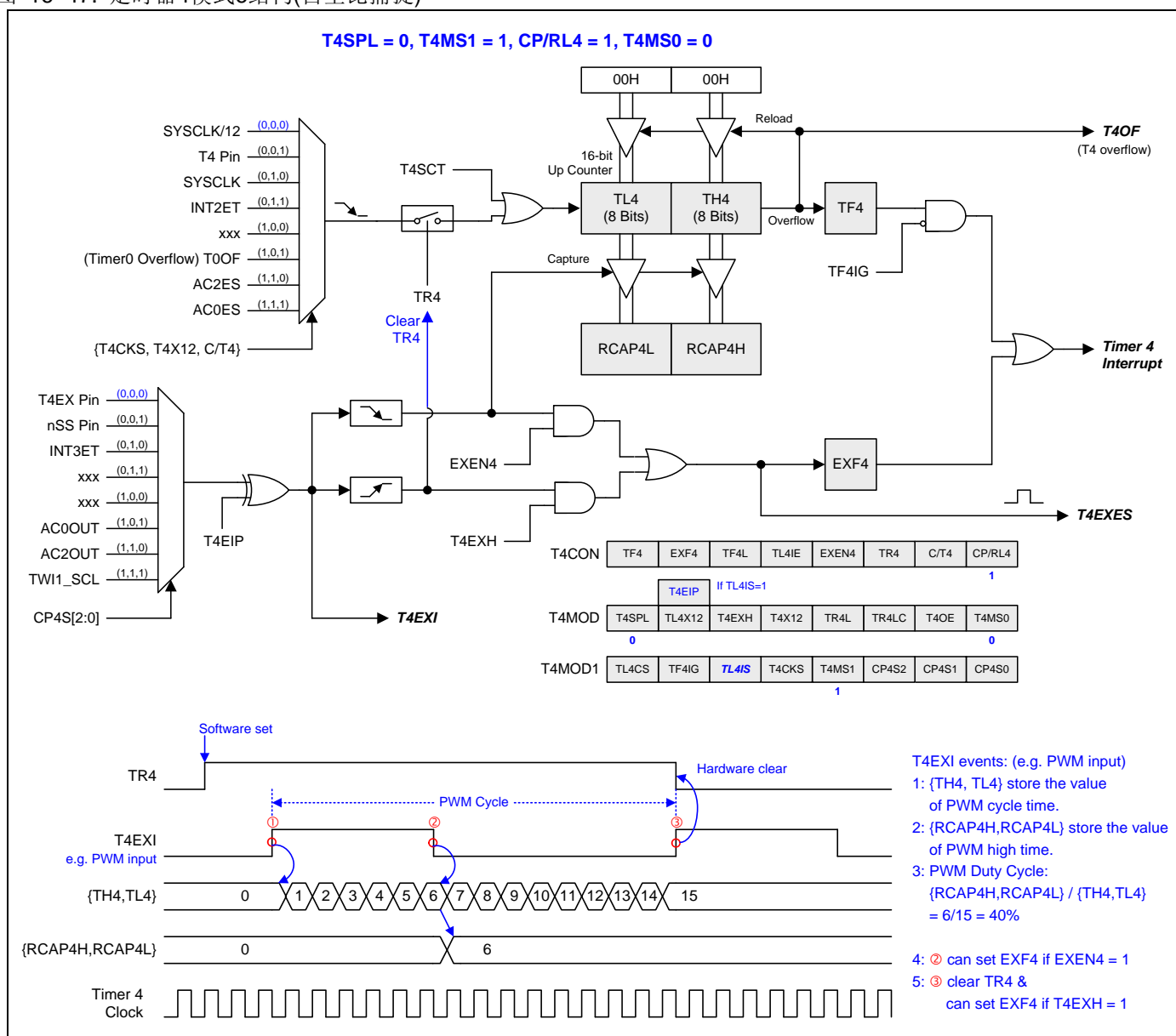
但是如果你想获得周期时间，那么就要通过清除EXEN4去阻止第二边沿。

当第三个边沿到来，它将自动清除TR4来停止计数器。

使用TH4: TL4(第三条边)、RCAP4H: RCAP4L(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 16-47所示

图 16-47. 定时器4模式6结构(占空比捕捉)



### 16.4.6. 分立定时器 4 模式 0 (自动重载和外部中断)

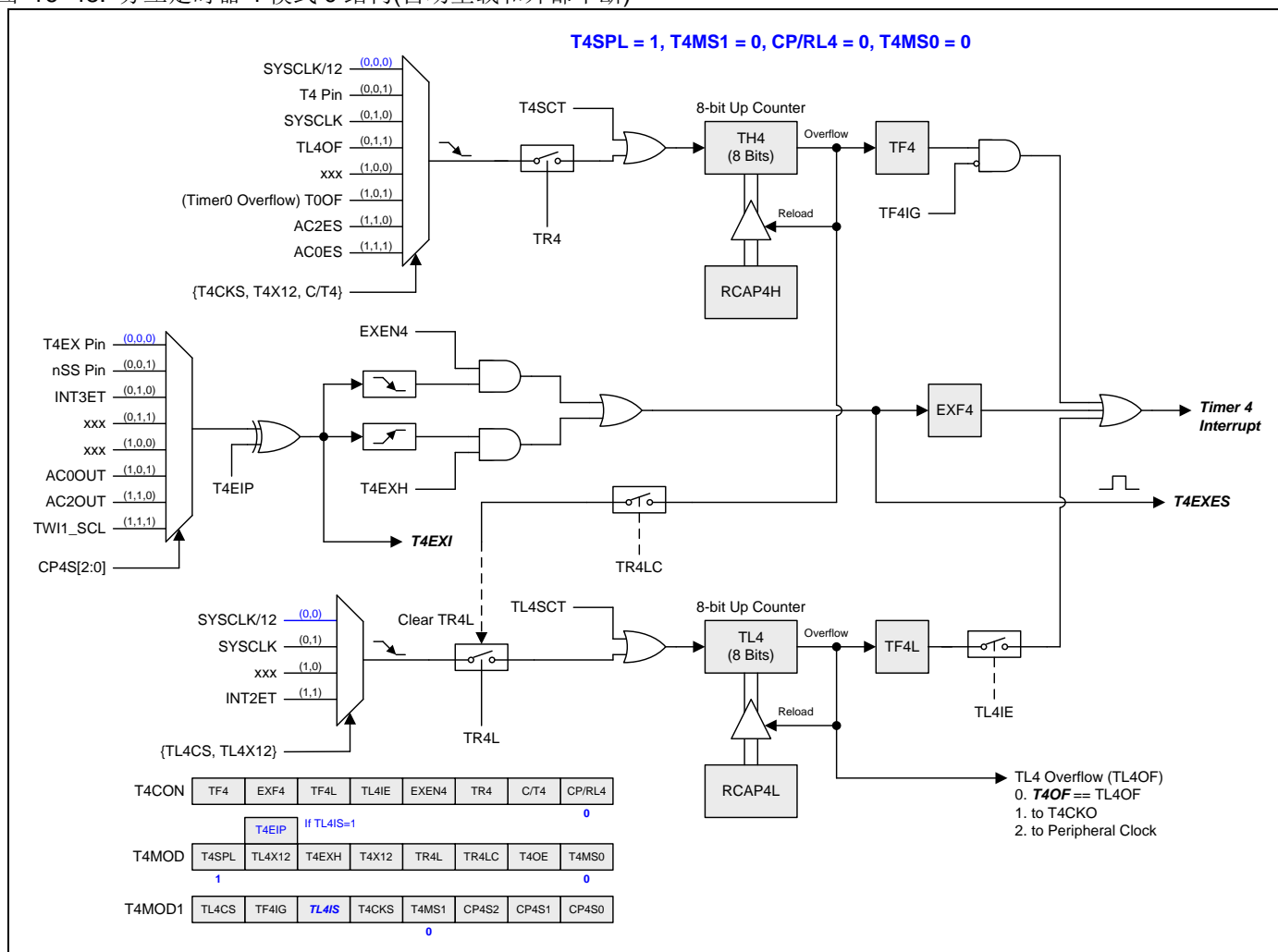
本模式中T4SPLIT置位,定时器4变为两个8位定时器(TH4和TL4)。两个8位定时器都是向上计数如图16-48所示。TH4保存RCAP4H的重载值和保持16位模式一样的8个时钟源输入选择。8位定时器功能跟16位模式的定时器4模式0相似。TL4保存4个时钟输入选择的RCAP4L重载值。T4CON的位TR4控制着TH4的运行。T4MOD的位TR4L控制着TL4的运行。当TR4LC置位时TH4溢出会停止TR4L的运行。

分立模式有4个中断标志EXF4、TF4和TF4L。EXF4与16位模式一样的功能用来侦测T4EXI引脚的跳变。TF4IG控制TF4在TH4从0xFF到0x00溢出时是否置位。TL4从0xFF到0x00溢出时TF4L置位,TL4IE使能中断。EXF4、TF4和TF4L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4溢出事件(T4OF)将被分立模式中的TL4溢出事件(TL4OF)取代。

如果T4MOD1的位TL4IS=0,位T4MOD.6是TL4X12的功能。如果TL4IS=1,位T4MOD.6是T4EIP的功能。

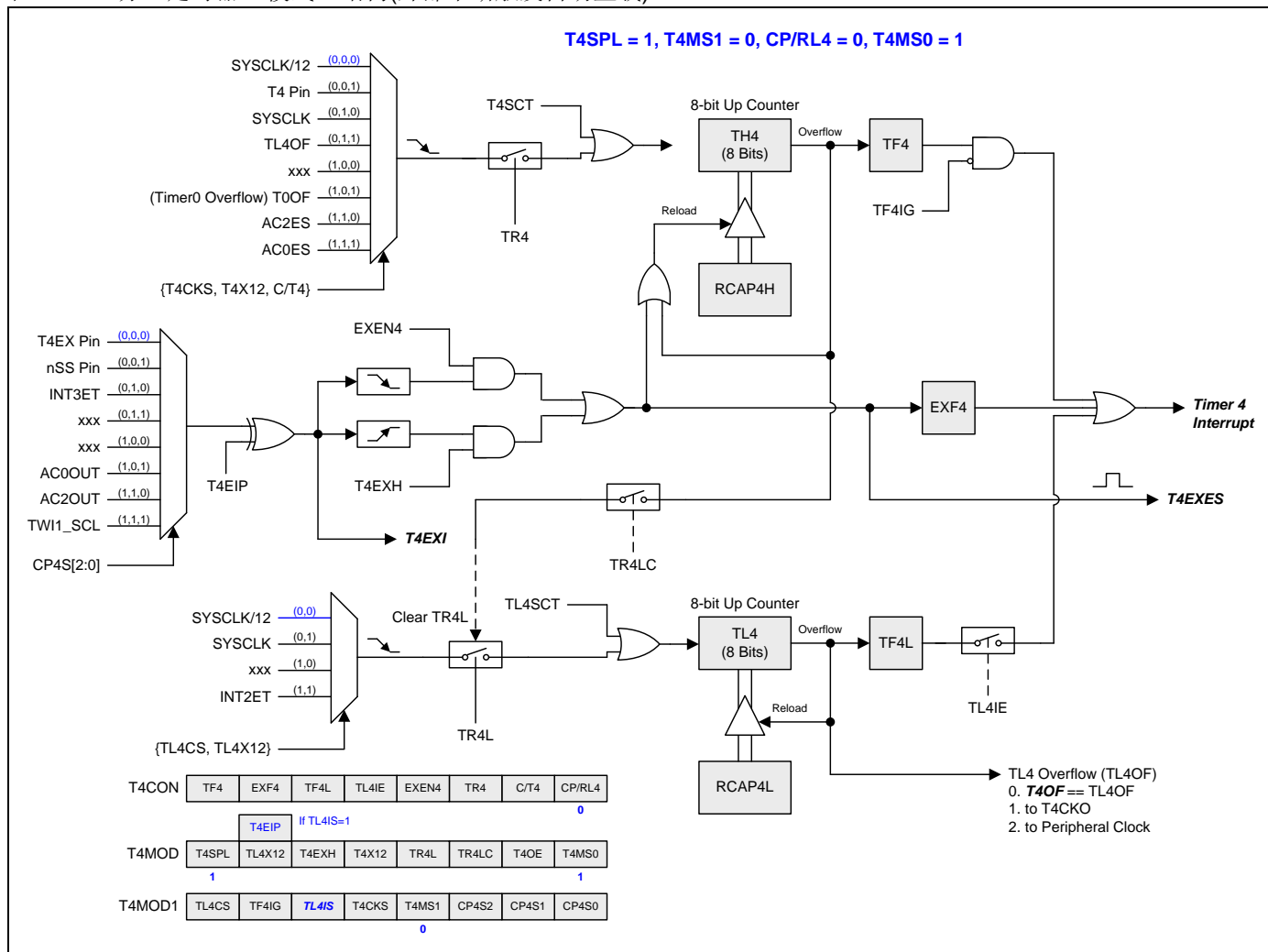
图 16-48. 分立定时器 4 模式 0 结构(自动重载和外部中断)



### 16.4.7. 分立定时器 4 模式 1(外部中断触发自动重载)

本模式中T4SPLIT置位，定时器4分立为两个8位定时器如图 16-49所示。跟定时器4模式1相似的功能且保持与分立定时器4模式0一样的中断方式。

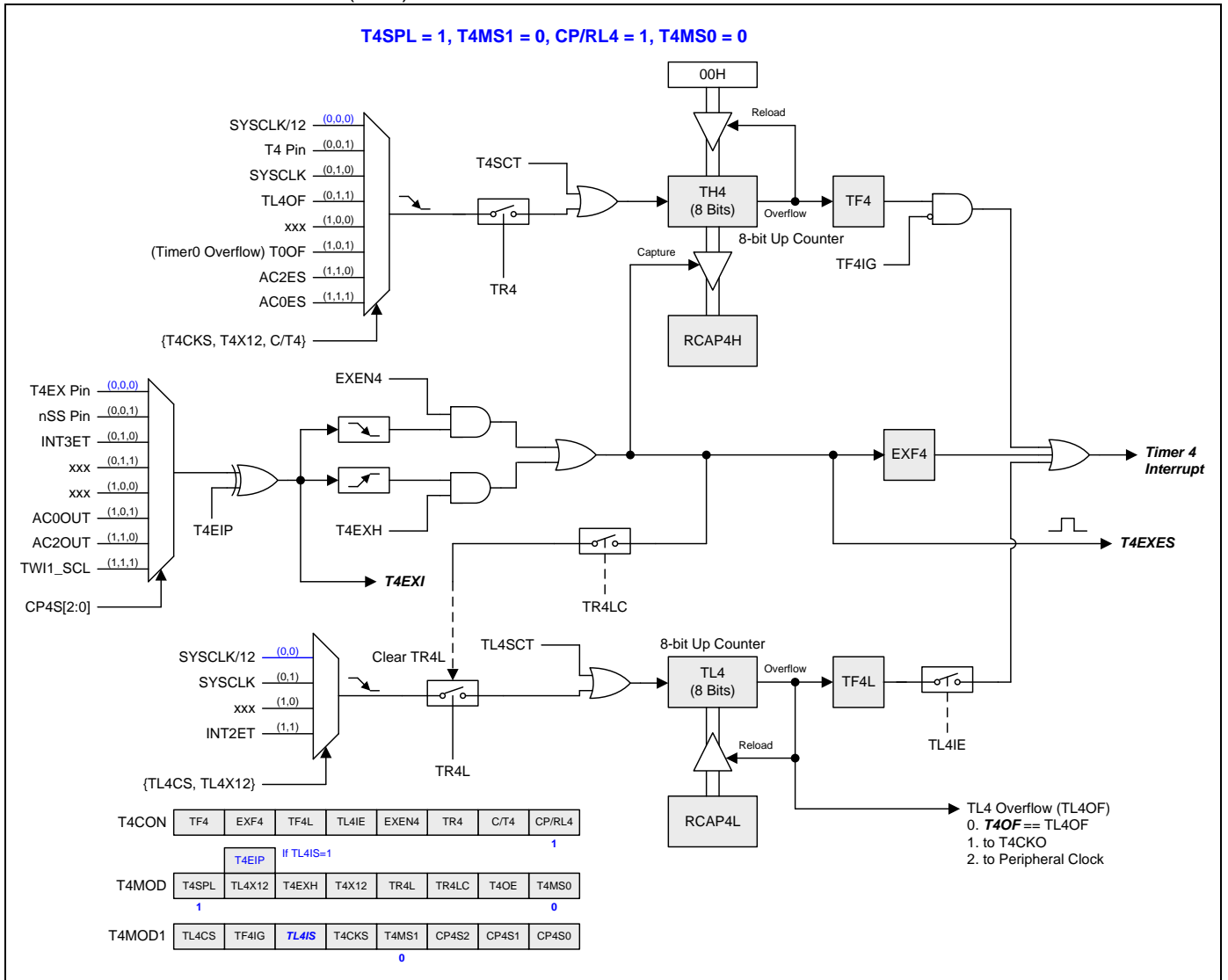
图 16-49. 分立定时器 4 模式 1 结构(外部中断触发自动重载)



16.4.8. 分立定时器 4 模式 2(捕捉)

本模式中T4SPLIT置位，定时器4分立为两个8位定时器如图 16-50所示。跟定时器4模式2相似的功能且保持与分立定时器4模式0一样的中断方式。

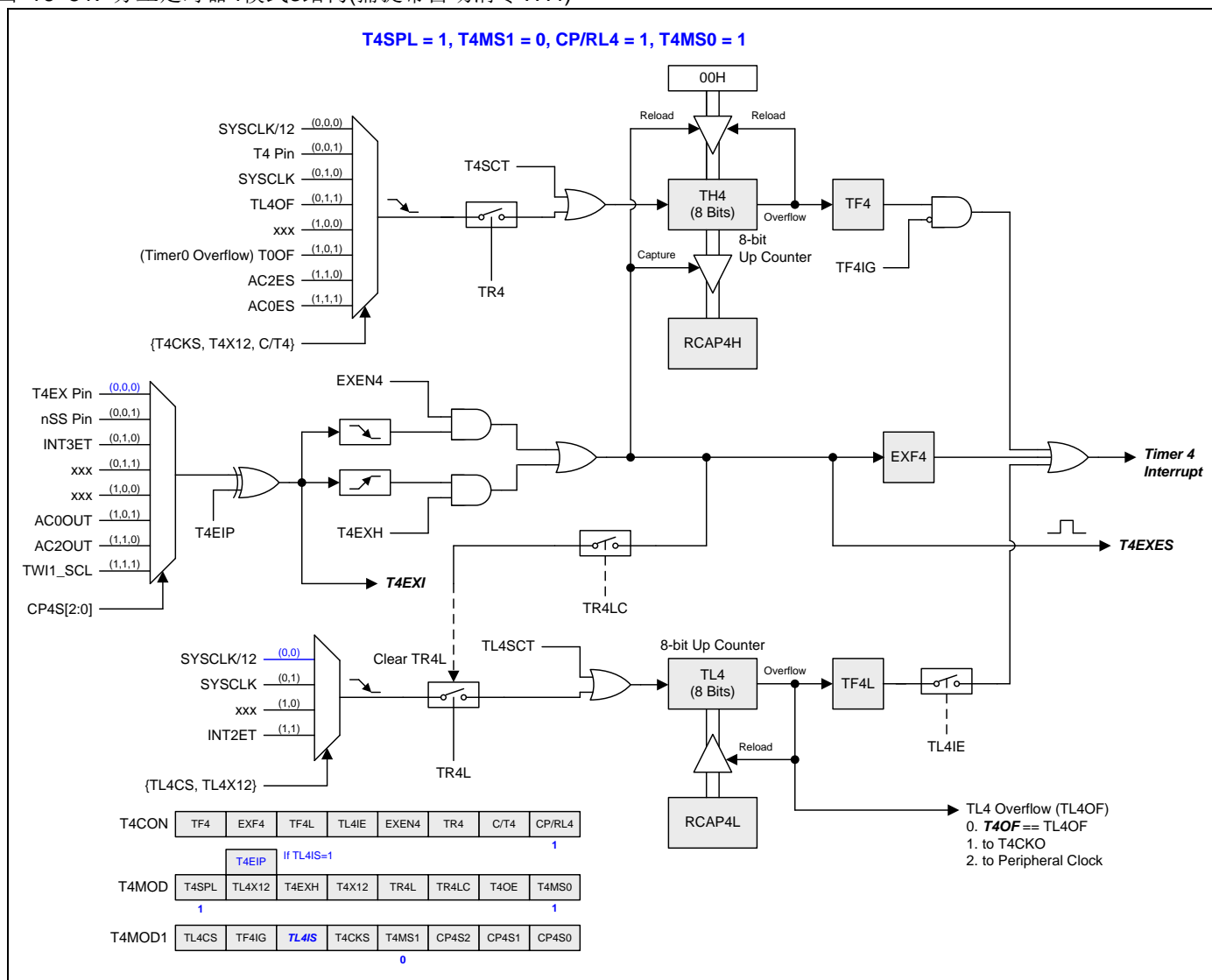
图 16-50. 分立定时器4模式2结构(捕捉)



### 16.4.9. 分立定时器 4 模式 3 (捕捉带自动清零)

本模式中T4SPLIT置位，定时器4分为两个8位定时器如图 16-51所示。跟定时器4模式3相似的功能且保持与分立定时器4模式0一样的中断方式。

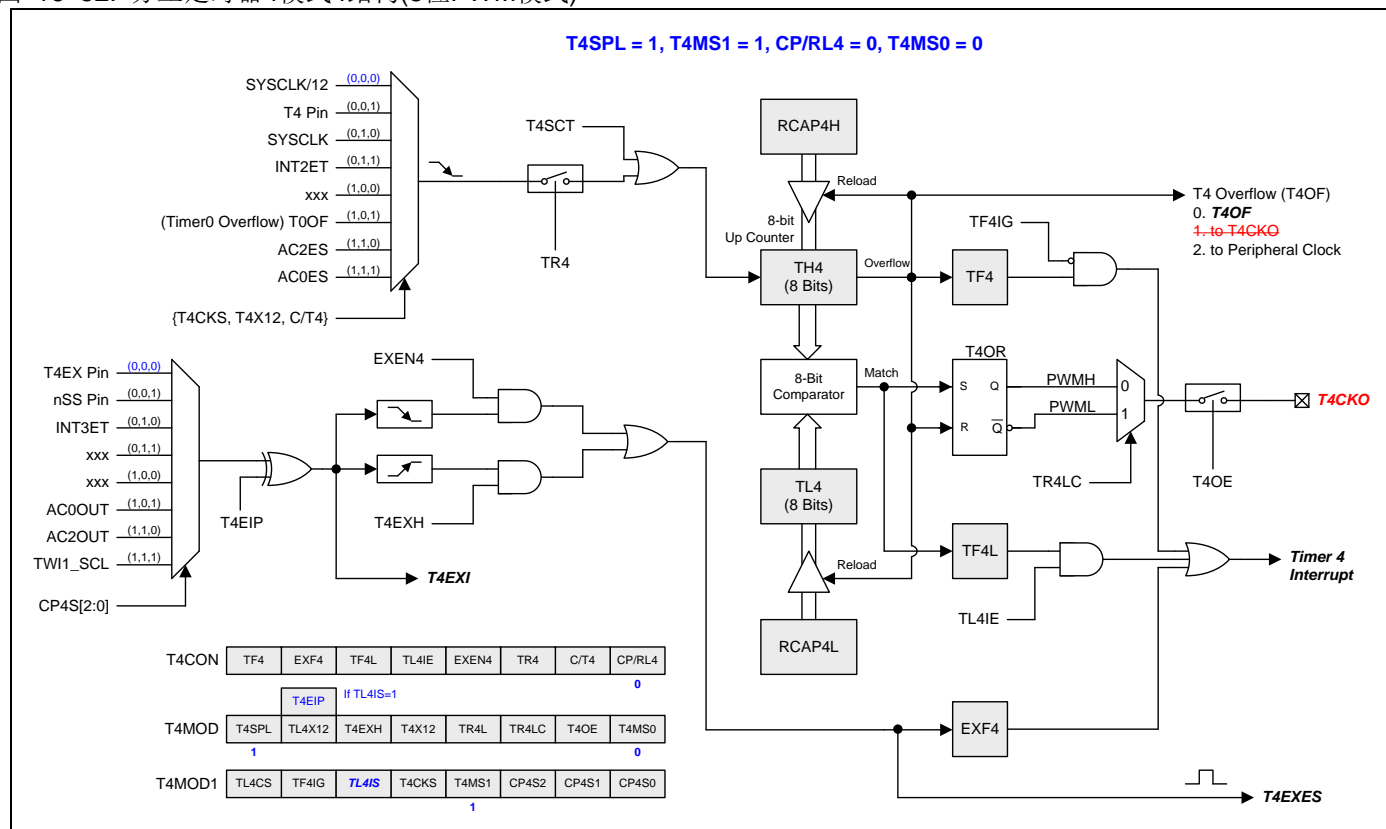
图 16-51. 分立定时器4模式3结构(捕捉带自动清零TH4)



## 16.4.10. 分立定时器4模式4(8位PWM模式)

本模式，定时器4是一个8位的PWM模式如图 16-52所示。TH4和RCAP4H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。TL4是PWM比较寄存器用来生成PWM波形。RCAP4L是PWM缓冲寄存器且在此寄存器中软件更新PWM数据。每次TH4溢出事件置位TF4且RCAP4L值载入到TL4。PWM信号输出到T4CKO功能引脚且输出的开关由T4MOD寄存器的位T4OE决定。

图 16-52. 分立定时器4模式4结构(8位PWM模式)



### 16.4.11. 定时器 4 可编程时钟输出

定时器4有一个时钟输出模式(当CP/RL4=0并且T4OE=1)。在这个模式,定时器4运行为一个占空比为50%的可编程时钟输出。产生的时钟从T4CKO输出。输入时钟(SYSCLK/2或SYSCLK)使16位定时器(TH4, TL4)加一。定时器从载入值到溢出重复计数。一旦溢出, (RCAP4H, RCAP4L)的值被载入到(TH4, TL4)同时计数。图 16-53给出了定时器4时钟输出频率计算公式。定时器4的时钟输出结构如图 16-54所示。

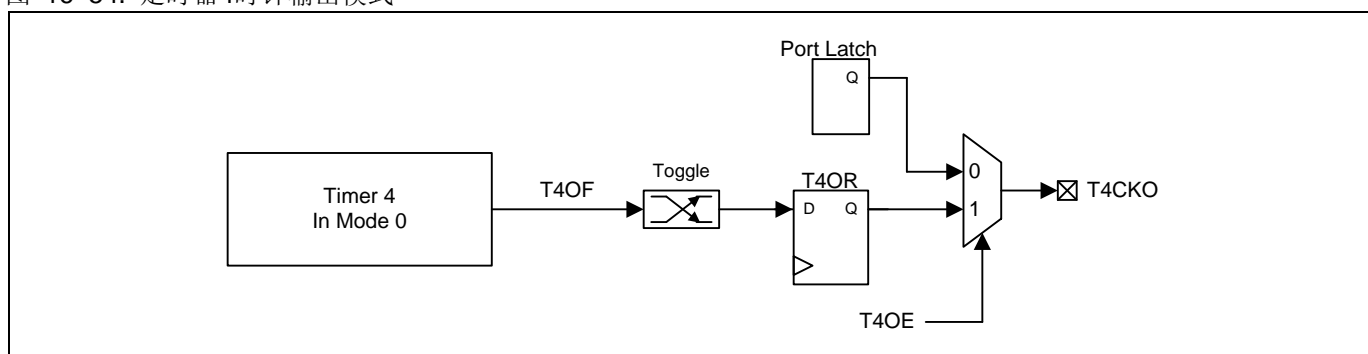
图 16-53. 定时器4时钟输出公式

$$\text{T4 Clock-out Frequency} = \frac{\text{T4 Clock Frequency}}{2 \times (65536 - (\text{RCAP4H}, \text{RCAP4L}))}$$

注意:

- (1) 定时器4溢出标志TF4,在定时器4溢出时置位可产生中断。但是TF4中断会被T4MOD1寄存器的位TF4IG锁住。
- (2) 当SYSCLK=12MHz及SYSCLK/2作为定时器4时钟源,定时器4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4时钟源,定时器3可编程输出频率范围从91.5Hz到6MHz。

图 16-54. 定时器4时钟输出模式



#### 定时器 4 时钟输出模式如何编程

- 选择定时器4时钟源
- 从公式计算出16位自动加载值并输入到RCAP4H和RCAP4L寄存器.
- 在TH4和TL4寄存器输入一个跟自动加载值相同的初始值
- T4MOD寄存器的T4OE置位.
- T4CON寄存器的TR4置位启动定时器4.

在时钟输出模式,定时器4翻转产生TF4中断。中断可以被TF4IG阻止。

如果定时器4在分立模式，时钟输出功能由TL4溢出产生且输出时钟频率为TL4溢出率的二分之一。当TL4溢出时RCAP4L是TL4重载值。TL4有4种时钟源选择。在使能分立定时器4时钟输出功能之前，软件必须结束TL4时钟源配置。图 16-56给出了TL4时钟输出频率公式。分立定时器4的时钟输出结构如图 16-56所示。

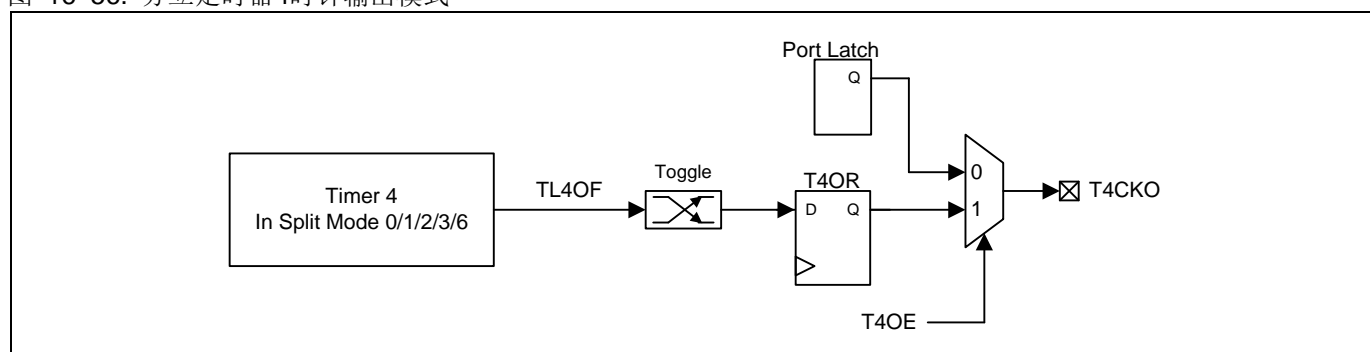
图 16-55. 分立定时器4时钟输出公式

$$\text{Split T4 Clock-out Frequency} = \frac{\text{TL4 Clock Frequency}}{3 \times (256 - \text{RCAP4L})}$$

注意:

- (1) TL4溢出标志TF4L,在TL4溢出时置位产生中断。但是TF4L中断由T4CON寄存器的位TL4IE使能。
- (2) 当SYSCLK=12MHz及SYSCLK/12作为TL4时钟源,TL4可编程输出频率范围从1.95KHz到500KHz。
- (3) 当SYSCLK=12MHz及SYSCLK作为TL4时钟源,TL4可编程输出频率范围从23.44KHz到6MHz。

图 16-56. 分立定时器4时钟输出模式



#### 分立定时器 4 时钟输出模式如何编程

- 选择TL4时钟源.
- 从公式计算出8位自动加载值并输入到RCAP4L寄存器.
- 在TL4寄存器输入一个跟自动加载值相同的初始值.
- T4MOD寄存器的T4OE置位.
- T4MOD寄存器的TR4L置位启动定时器4.

在时钟输出模式，TL4翻转不会产生中断。这和TL4用作波特率发生器时相似。可同时使用TL4作为一个波特率发生器和时钟发生器。注意，在分立定时器4模式下波特率和时钟输出都由TL4的溢出速率来决定。TF4L中断由T4CON寄存器的TL4IE位使能。

## 16.4.12. 定时器 4 寄存器

**T4CON: 定时器4控制寄存器**

SFR 页 = 仅 2 页

SFR 地址 = 0xC8

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TF4	EXF4	TF4L	TL4IE	EXEN4	TR4	C/T4	CP/RL4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TF4, 定时器4溢出标志

0: TF4必须软件清零。

1: 定时器4溢出TF4置位。

Bit 6: EXF4, 定时器4外部标志。

0: EXF4必须软件清零。

1: 当EXEN4=1且在T4EX上有负跳变导致重载或捕获, 或者T4EXH=1并且在T4EX上有一个正跳变, 将置位定时器外部标志。当定时器4中断使能时, EXF4=1时将引起CPU进入定时器4中断向量程序。当MCU在掉电模式下定时器4中断使能时, EXF4被设置成具有唤醒MCU能力的电平触发。

Bit 5: TF4L, 在定时器4分立模式中TL4溢出标志。

0: TF4L必须软件清零。

1: 在定时器4分立模式中TL4溢出TF4L置位。

Bit 4: TL4IE, TF4L中断使能。

0: 禁止TF4L中断。

1: 使能共享定时器4中断入口的TF4L中断。

Bit 3: EXEN4, 定时器4外部使能位在定时器4外部输入引脚的负跳变。

0: 定时器4忽略定时器4外部输入引脚的负跳变事件。

1: 在定时器4外部输入的负跳变时捕获或加载并作为结果。如果定时器4配置为没有捕捉或重载的模式0, 定时器4外部输入保持外部信号侦测并产生 EXF4标志响应定时器4中断。

Bit 2: TR4, 定时器4运行控制位。如果在定时器4分立模式中, 仅控制TH4。

0: 定时器/计数器4停止运行。

1: 定时器/计数器4开启运行。

Bit 1: C/T4, 定时器4时钟或计数器输入源选择位。和T4X12与T4CKS一起决定定时器4的输入来源。如下定义:

T4CKS, T4X12, C/T4	定时器 4 时钟源	分立模式下 TH4 时钟选择
0 0 0	SYSClk/12	SYSClk/12
0 0 1	T4 引脚	T4 引脚
0 1 0	SYSClk	SYSClk
0 1 1	INT2ET	TL4OF
1 0 0	--	--
1 0 1	T0OF	T0OF
1 1 0	AC2ES	AC2ES
1 1 1	AC0ES	AC0ES

Bit 0: CP/RL4, 定时器4模式控制位。参考T4MOD.T4MS0 的功能定义描述

**T4MOD: 定时器4 模式寄存器**

SFR 页 = 仅 2 页

SFR 地址 = 0xC9

复位值 = 0000-0000

7	6	5	4	3	2	1	0
T4SPL	TL4X12/ T4EIP	T4EXH	T4X12	TR4L	TR4LC	T4OE	T4MS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: T4SPL, 定时器4分立模式控制。

0: 禁止定时器4的分立模式。

1: 使能定时器4的分立模式。

**必须清零TL4IS (T4MOD1.5)才能访问TL4X12位。**

Bit 6: TL4X12, 定时器4分立模式下TL4时钟控制位。

TL4CS, TL4X12	TL4 时钟选择
0 0	SYSCCLK/12
0 1	SYSCCLK
1 0	--
1 1	INT2ET

**必须置位TL4IS (T4MOD1.5)才能访问T4EIP位。**

Bit 6: T4EIP, T4EXI 输入信号反相控制位。

0: T4EXI输入信号不反相

1: T4EXI输入信号反相。

Bit 5: T4EXH, 定时器4外部T4EX 引脚的正跳变使能标志。

0: 定时器4忽略T4EX引脚的正跳变事件。

1: 允许在T4EX的正跳变时捕获或加载并作为结果且置位EXF4。

Bit 4: T4X12, 定时器4时钟源选择。参考C/T4的功能定义描述。

Bit 3: TR4L, 在定时器4分立模式中, TL4运行控制位。

0: 停止TL4。

1: 使能TL4。

Bit 2: TR4LC, TR4L清除控制位。

0: 禁止硬件事件清零TR4L。

1: 使能TH4溢出(定时器4在模式0/1)或者捕获输入(定时器4在模式2/3)时自动清零TR4L。

Bit 1: T4OE, 定时器4时钟输出使能位。

0: 禁止定时器4时钟输出。

1: 使能定时器4时钟输出。

Bit 0: T4MS0, 定时器4模式选择位0。

T4MS1, CP/RL4, T4MS0	定时器4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4捕捉带自动清零
1 0 0	模式4: 8位PWM (T4SPL = 1)
1 1 0	模式6: 占空比捕捉
其它	保留

**T4MOD1: 定时器4 模式寄存器 1**

SFR 页 = 仅 3 页

SFR 地址 = 0x93

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL4CS	TF4IG	TL4IS	T4CK2	T4MS1	CP4S2	CP4S1	CP4S0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TL4CS, 在定时器4分立模式下的TL4 时钟选择选择。参考T4MOD.TL4X12的功能描述。

Bit 6: TF4IG, TF4中断忽略。

0: 使能TF4中断。默认是使能的。

1: 禁止TF4中断。

Bit 5: TL4IS, TL4X12/T4EIP访问控制

0: 使能T4MOD.6访问TL4X12

1: 使能T4MOD.6访问T4EIP

Bit 4: T4CKS, 定时器4时钟输入选择。参考C/T4的功能描述。

Bit 3: T4MS1, 定时器4模式选择位1。参考T4MOD. T4MS0的功能描述。

Bit 2~0: CP4S.2~0, 此3位定义定时器4的捕捉源选择。

CP4S.2~0	定时器4捕捉源选择
0 0 0	T4EX引脚
0 0 1	nSS引脚
0 1 0	INT3ET
0 1 1	--
1 0 0	--
1 0 1	AC0OUT
1 1 0	AC2OUT
1 1 1	TWI1_SCL

#### TL4: 定时器 4 低字节寄存器

SFR 页 = 仅 2 页

SFR 地址 = 0xCC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL4.7	TL4.6	TL4.5	TL4.4	TL4.3	TL4.2	TL4.1	TL4.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

#### TH4: 定时器 4高字节寄存器

SFR 页 = 仅 2 页

SFR 地址 = 0xCD

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TH4.7	TH4.6	TH4.5	TH4.4	TH4.3	TH4.2	TH4.1	TH4.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

#### RCAP4L: 定时器 4 捕捉低字节寄存器

SFR 页 = 仅 2 页

SFR 地址 = 0xCA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
RCAP4L.7	RCAP4L.6	RCAP4L.5	RCAP4L.4	RCAP4L.3	RCAP4L.2	RCAP4L.1	RCAP4L.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

#### RCAP4H: 定时器 4 捕捉高字节寄存器

SFR 页 = 仅 2 页

SFR 地址 = 0xCB

复位值 = 0000-0000

7	6	5	4	3	2	1	0
RCAP4H.7	RCAP4H.6	RCAP4H.5	RCAP4H.4	RCAP4H.3	RCAP4H.2	RCAP4H.1	RCAP4H.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**AUXR13: 辅助寄存器 13**

SFR 页 = 仅 A 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	S3PS0	0	S2PS0	0	T4FCS	T4PS1	T4PS0
W	R/W	W	R/W	W	R/W	R/W	R/W

Bit 1~0: T4PS1~0, 定时器4 端口引脚选择 [1:0].

T4PS1~0	T4/T4CKO	T4EX
0 0	P7.0	P7.1
0 1	P3.7	P3.6
1 0	P7.2	P5.0
1 1	P3.6	P3.7

## 16.5. 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。

### 16.5.1. 所有定时器运行的全局使能

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
TR4LE	TR3LE	TR2LE	TR4E	TR3E	TR2E	TR1E	TR0E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TR4LE, 当定时器4在分立模式下时，这个位上写“1”设置TR4L使能(TR4L=1)来控制TL4。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: TR3LE, 当定时器3在分立模式下时，这个位上写“1”设置TR3L使能(TR3L=1)来控制TL3。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: TR2LE, 当定时器2在分立模式下时，这个位上写“1”设置TR2L使能(TR2L=1)来控制TL2。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 4: TR4E, 这个位上写“1”设置TR4使能(TR4=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 3: 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 2: 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 1: TR1E, 这个位上写“1”设置TR1使能(TR1=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: TR0E, 这个位上写“1”设置TR0使能(TR0=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。

### 16.5.2. 所有定时器重载的全局使能

#### 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
TL4RLC	TL3RLC	TL2RLC	T4RLC	T3RLC	T2RLC	T1RLC	T0RLC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: TL4RLC, 当定时器4在分立模式下时，这个位上写“1”强制TL4重载条件激活。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: TL3RLC, 当定时器3在分立模式下时，这个位上写“1”强制TL3重载条件激活。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: TL2RLC, 当定时器2在分立模式下时，这个位上写“1”强制TL2重载条件激活。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 4: T4RLC, 当定时器4在非分立模式下时，这个位上写“1”强制TH4和TL4重载。或者在分立模式下强制TH4重载。如果定时器在占空比捕捉模式下强制重载无效。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 3: T3RLC**, 当定时器3在非分立模式下时, 这个位上写“1”强制TH3和TL3重载。或者在分立模式下强制TH3重载。如果定时器在占空比捕捉模式下强制重载无效。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 2: T2RLC**, 当定时器2在非分立模式下时, 这个位上写“1”强制TH2和TL2重载。或者在分立模式下强制TH2重载。如果定时器在占空比捕捉模式下强制重载无效。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 1: T1RLC**, 定时器1模式2下, 这个位上写“1”强制TL1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: T0RLC**, 定时器0模式2下, 这个位上写“1”强制TL0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。

### 16.5.3. 所有定时器停止的全局使能

#### TSPC0: 定时器 Stop 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
TL4SC	TL3SC	TL2SC	T4SC	T3SC	T2SC	T1SC	T0SC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Bit 7: TL4SC**, 当定时器4在分立模式下时, 这个位上写“1”设置TR4L禁止( $TR4L=0$ )。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: TL3SC**, 当定时器3在分立模式下时, 这个位上写“1”设置TR3L禁止( $TR3L=0$ )。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: TL2SC**, 当定时器2在分立模式下时, 这个位上写“1”设置TR2L禁止( $TR2L=0$ )。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 4: T4SC**, 这个位上写“1”设置TR4禁止( $TR4=0$ )。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 3: 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 2: 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 1: T1SC**, 这个位上写“1”设置TR1禁止( $TR1=0$ )。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: T0SC**, 这个位上写“1”设置TR0禁止( $TR0=0$ )。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. 可编程计数器阵列(PCAO)

MGEQ1C064带有一个可编程计数器阵列(PCAO)，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. PCA 概述

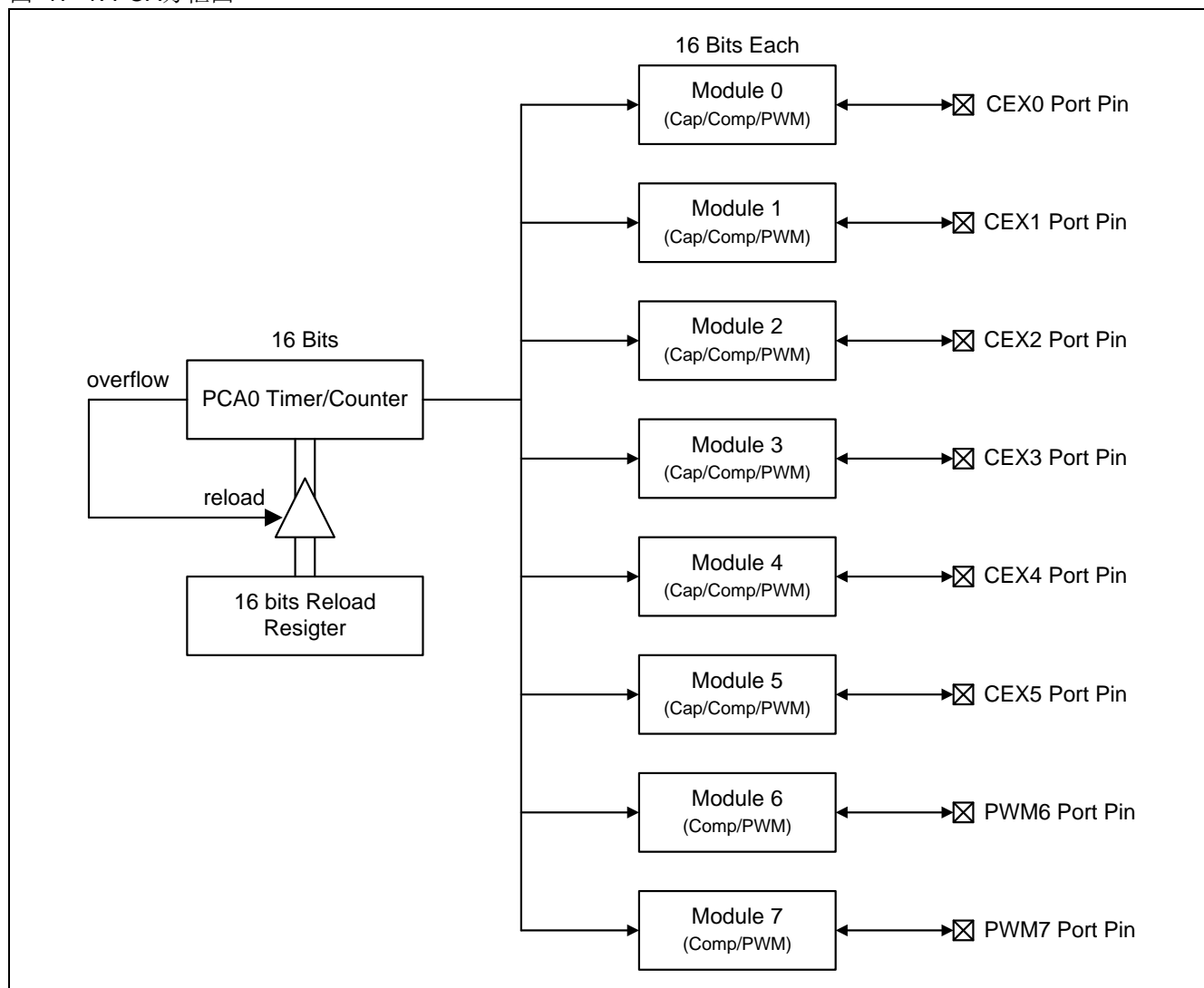
PCAO由一个专用定时/计数器作为一个6组比较/捕获/PWM模块和2组比较/PWM模块的时间基准。PCAO的功能方框图如图 17-1所示。需要注意的是PCAO定时器和模块都是16位的。如果一个外部事件同一个模块关联，那么该功能就和相应的端口引脚共享。若某模块没有使用端口引脚，这个引脚还可以用作标准I/O。

模块0~5中的任意一组都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)

模块6~7除了上升和/或下降沿捕获之外支持上面的模式。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PCA定时器和模块。

图 17-1. PCA方框图





Bit 6: BME4, PCA0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: PCA0模块4/5禁止缓冲模式。

1: PCA0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: PCA0模块2/3禁止缓冲模式。

1: PCA0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: CPS2~0, PCA计数器时钟源选择位。

CPS2	CPS1	CPS0	PCA时钟源
0	0	0	内部时钟, (系统时钟)/12
0	0	1	内部时钟, (系统时钟)/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, (系统时钟)/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  $\geq$  SYSCLK x2。

Bit 0: ECF, 使能PCA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PCA运行控制位和PCA定时器与每个模块的标志。要运行PCA, CR位(CCON.6)必须软件置位。要关闭PCA, 可以清除该位。PCA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。PCA中断系统如图 17-3.所示。

#### CCON: PCA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, PCA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, PCA计数器运行控制位。

0: 软件清零关闭PCA计数器。

1: 软件置位开启PCA计数器。

Bit 5: CCF5, PCA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, PCA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, PCA模块3中断标志。

0: 只能软件清零。

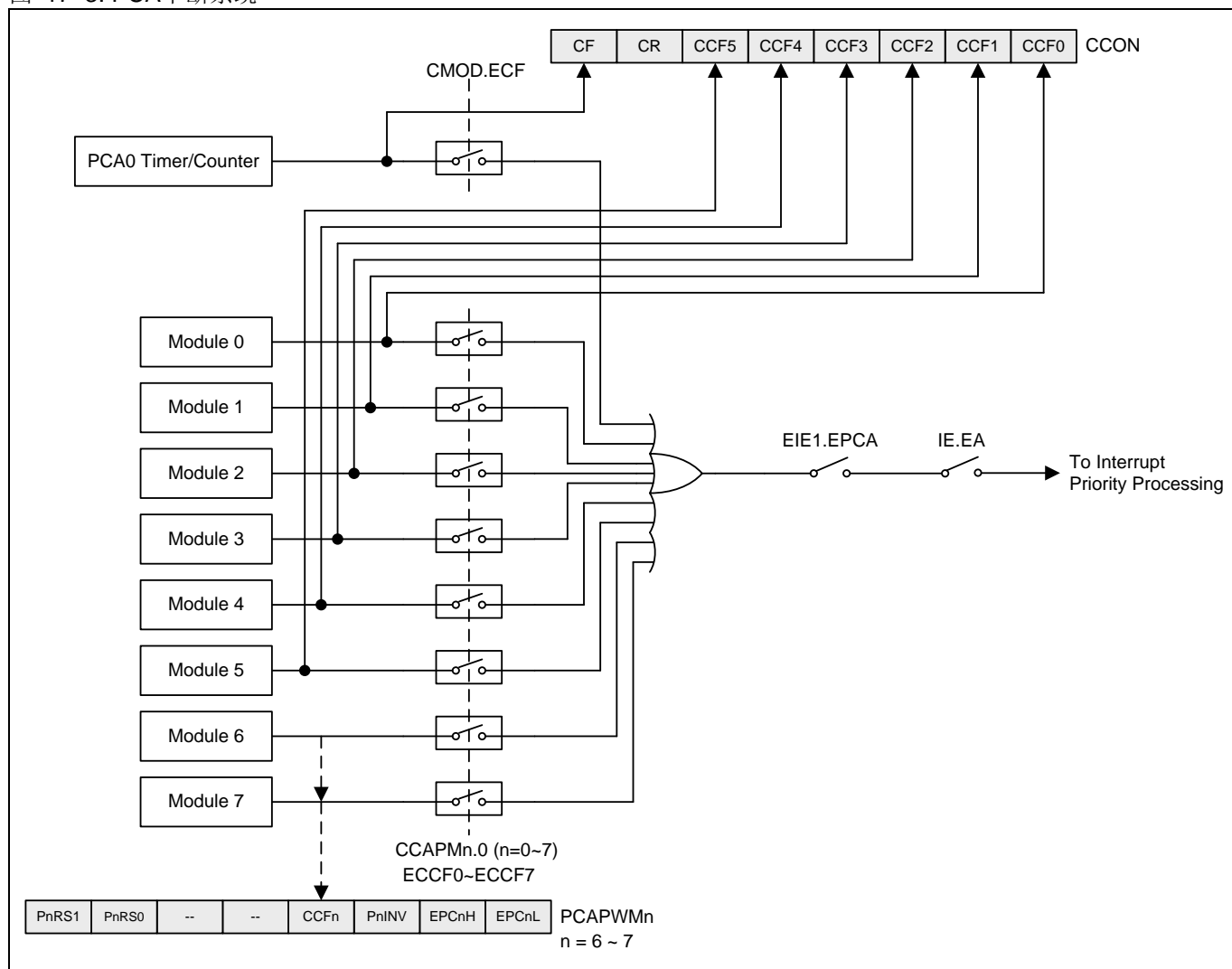
1: 当发生一个匹配或捕获事件时, 硬件置位。

**Bit 2: CCF2, PCA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, PCA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, PCA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. PCA中断系统



**PCAPWMn: PWM模式辅助寄存器, n=0~7**

SFR 页 = 仅 0 页 for n= 0~1 (n=2~5 所有页)

SFR 页 = 仅 1 页 for n= 6~7

SFR 地址 = 0xF2~0xF7

复位值 = 0000-0000

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

Bit 3: CCFn, PCA0模块6和7中断标志仅CCF6和CCF7有效。

0: 只能软件清零。

1: 当发生一个匹配时, 硬件置位。

**CH: PCA基准定时器高字节**

SFR 页 = 0 ~ F

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: PCA基准定时器低字节**

SFR 页 = 0 ~ F

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: PCA CH重载寄存器**

SFR 页 = 0 ~ F

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: PCA CL重载寄存器**

SFR 页 = 0 ~ F

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/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CCAPMn(n=0、1、2、3、4、5、6或7)，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: PCA模块比较/捕捉寄存器, n=0~5

SFR 页 = 仅 0 页 for n= 0~1 (n=2~5 所有页)

SFR 地址 = 0xDA~0xDF

复位值 = 0000-0000

7	6	5	4	3	2	1	0
DTE <sub>n</sub>	ECOM <sub>n</sub>	CAPP <sub>n</sub>	CAPN <sub>n</sub>	MAT <sub>n</sub>	TOG <sub>n</sub>	PWM <sub>n</sub>	ECCF <sub>n</sub>
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

0: PWM<sub>n</sub>输出禁止死区时间控制。

1: PWM<sub>n</sub>输出使能死区时间控制。

Bit 6: ECOM<sub>n</sub>, 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: CAPP<sub>n</sub>, 正跳变捕捉使能。模块6和7不支持捕捉模式。

0: 禁止PCA捕捉功能在CEX<sub>n</sub>引脚上正跳变侦测。

1: 使能PCA捕捉功能在CEX<sub>n</sub>引脚上正跳变侦测。

Bit 4: CAPN<sub>n</sub>, 负跳变捕捉使能。模块6和7不支持捕捉模式。

0: 禁止PCA捕捉功能在CEX<sub>n</sub>引脚上负跳变侦测。

1: 使能PCA捕捉功能在CEX<sub>n</sub>引脚上负跳变侦测。

Bit 3: MAT<sub>n</sub>, 匹配控制。

0: 禁止数字比较器匹配事件去置位CCF<sub>n</sub>。

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<sub>n</sub>置位。

Bit 2: TOG<sub>n</sub>, 切换控制。

0: 禁止数字比较器匹配事件去切换CEX<sub>n</sub>。

1: PCA计数器同相应模块的比较/捕获寄存器匹配时使CEX<sub>n</sub>引脚切换。

Bit 1: PWM<sub>n</sub>, PWM控制。

0: 禁止PCA模块中的PWM模式。

1: 使能PWM功能并使CEX<sub>n</sub>引脚用作脉宽调制输出。

Bit 0: ECCF<sub>n</sub>, 使能CCF<sub>n</sub>中断。

0: 禁止CCON寄存器中的比较/捕获标志位CCF<sub>n</sub>产生中断。

1: 使能CCON寄存器中的比较/捕获标志位CCF<sub>n</sub>产生中断。

*注意: CAPN<sub>n</sub> (CCAPMn.4)位和CAPP<sub>n</sub> (CCAPMn.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(CCAPnH, CCAPnL)。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PCAPWM<sub>n</sub>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17.4.6 和 17.4.7。

**CCAPMn: PCA模块比较/捕捉寄存器, n=6~7**

SFR 页 = 仅 1 页 for n= 6~7

SFR 地址 = 0xDB, 0xDA

复位值 = 0000-0000

7	6	5	4	3	2	1	0
BME6	ECOMn	--	CAPNn	MATn	TOGn	PWMn	ECCFn
R/W	R/W	W	R/W	R/W	R/W	R/W	R/W

Bit 7: BME6(此位仅在CCAPM6), PCA模块6/7缓冲模式使能。仅在捕捉模式, PWM模式或COPM模式下的PCA模块6和7有效。

0: PCA 模块6/7禁止缓冲模式。

1: PCA 模块6/7使能缓冲模式。

Bit 6: ECOMn, 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: 保留位。模块6和7不支持捕捉模式。

Bit 4: CAPNn, 负跳变捕捉使能。模块6和7不支持捕捉模式。CAPN6和CAPN7应用在其他PCA模式, 详情请参考“[表 17.1. PCA模块模式](#)”。

Bit 3: MATn, 匹配控制。

0: 禁止数字比较器匹配事件去置位CCFn。

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CCFn置位。

Bit 2: TOGn, 切换控制。

0: 禁止数字比较器匹配事件去切换CEXn。

1: PCA计数器同相应模块的比较/捕获寄存器匹配时使CEXn引脚切换。

Bit 1: PWMn, PWM控制。

0: 禁止PCA模块中的PWM模式。

1: 使能PWM功能并使CEXn引脚用作脉宽调制输出。

Bit 0: ECCFn, 使能CCFn中断。

0: 禁止CCON寄存器中的比较/捕获标志位CCFn产生中断。

1: 使能CCON寄存器中的比较/捕获标志位CCFn产生中断。

每个模块都有一对8位比较/捕获寄存器(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PCAPWMn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[17.4.6](#) 和 [17.4.7](#)

**CCAPnH: PCA模块n捕捉高寄存器高字节, n=0~7**

SFR 页 = 仅 0 页 for n= 0~1 (n=2~5 所有页)

SFR 页 = 仅 1 页 for n= 6~7

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: PCA模块n捕捉高寄存器低字节, n=0~7**

SFR 页 = 仅 0 页 for n= 0~1 (n=2~5 所有页)

SFR 页 = 仅 1 页 for n= 6~7

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=0~7**

SFR 页 = 仅 0 页 for n= 0~1 (n=2~5 所有页)

SFR 页 = 仅 1 页 for n= 6~7

SFR 地址 = 0xF2~0xF7

复位值 = 0000-0000

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

Bit 7~6: PnRS1~0, PWMn分别率设置位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-1111-1111-1111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 5~4: 保留位, 写寄存器时, 此位必须写“0”。

Bit 3: CCFn, 仅在模块6和7中CCF6和CCF7是有效的中断标志。

0: 必需软件清零。

1: 当发生一个匹配或捕获事件时, 硬件置位。

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. PCA 操作模式**

不同PCA功能对应的CCAPMn寄存器设置如表 17.1所示。

表 17.1. PCA模块模式

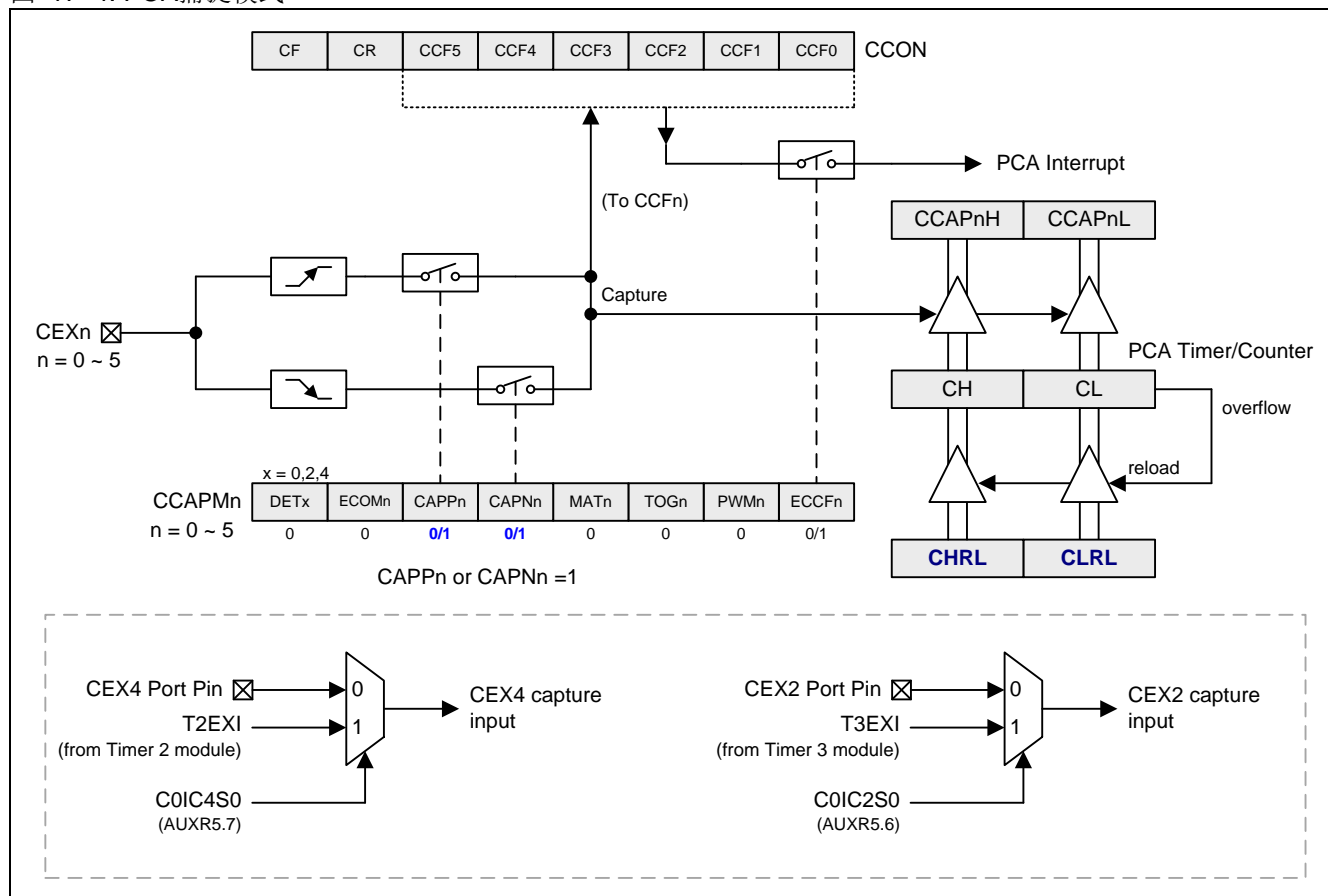
ECOMn	CAPPn	CAPn	MATn	TOGn	PWMn	ECCFn	模块功能
0	0	0	0	0	0	0	无操作
X	1	0	0	0	0	X	CEXn 引脚正跳变触发16位捕捉
X	0	1	0	0	0	X	CEXn 引脚负跳变触发16位捕捉
X	1	1	0	0	0	X	CEXn 引脚正负跳变触发16位捕捉
1	0	0	1	0	0	X	16位软件定时器 (比较)
1	0	0	1	1	0	X	16位高速输出(HSO)
1	0	0	0/1	0	1	X	脉宽调制器 (PWM)
1	0	0	0	1	1	X	PWM匹配事件的比较输出(COPM)
1	0	1	0	0	1	X	FIFO数据模式

注意: PCA模块6和模块7不支持捕捉模式。

## 17.4.1. 捕捉模式

要让某一PCA模块工作在捕获模式，CAPN和CAPP任何一位或两位必须置位。外部CEX输入会在每次跳变时采样。当有效跳变发生时，PCA硬件会将PCA计数器寄存器值(CH和CL)载入到模块的捕捉寄存器(CCAPnH和CCAPnL)。若模块的CCFn和ECCFn标志同时置位，会产生一个中断。

图 17-4. PCA捕捉模式



## 17.4.2. 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(CCAPnH, CCAPnL, n= 1、3、5)送到偶数模块捕捉数据寄存器(通道0、2、4)。这不影响模块0/2/4的捕捉操作。BME0使能通道0/1的缓冲操作。BME2和BME4控制模块2/3和模块4/5。

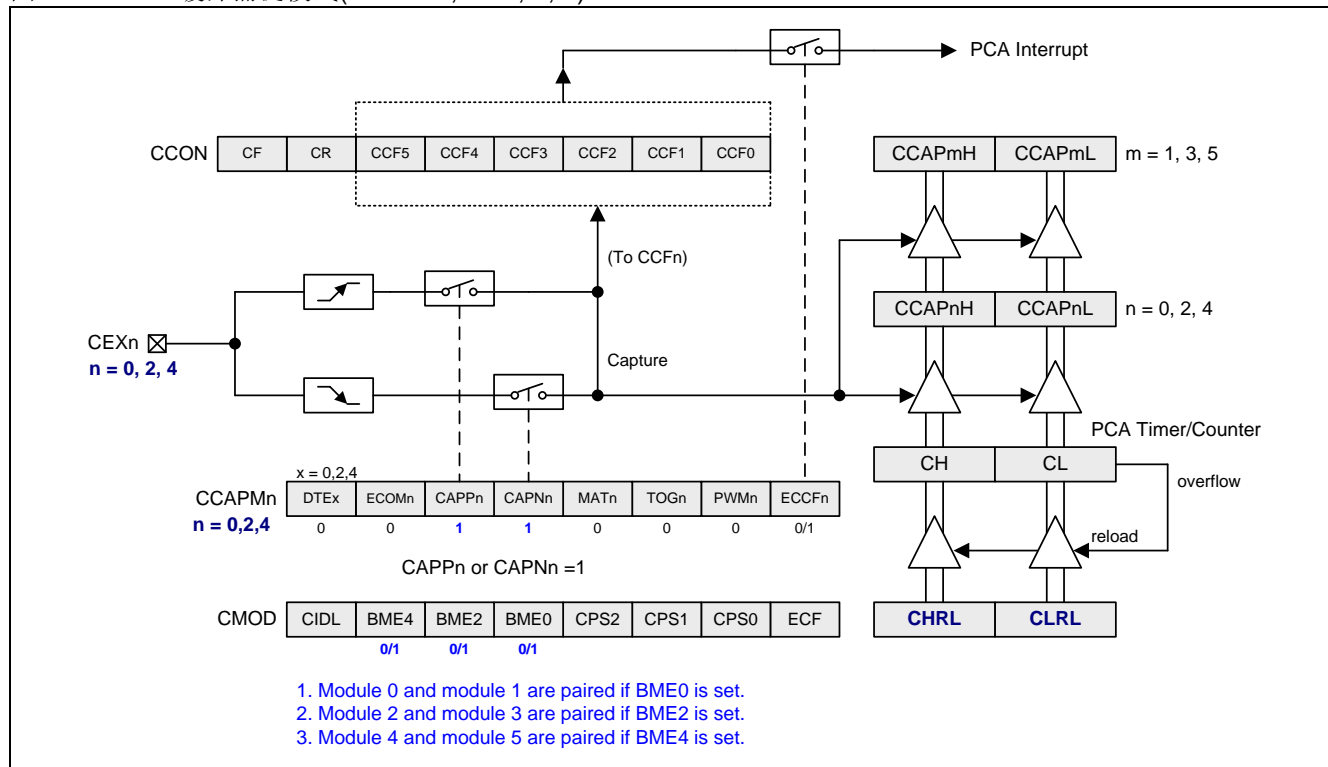
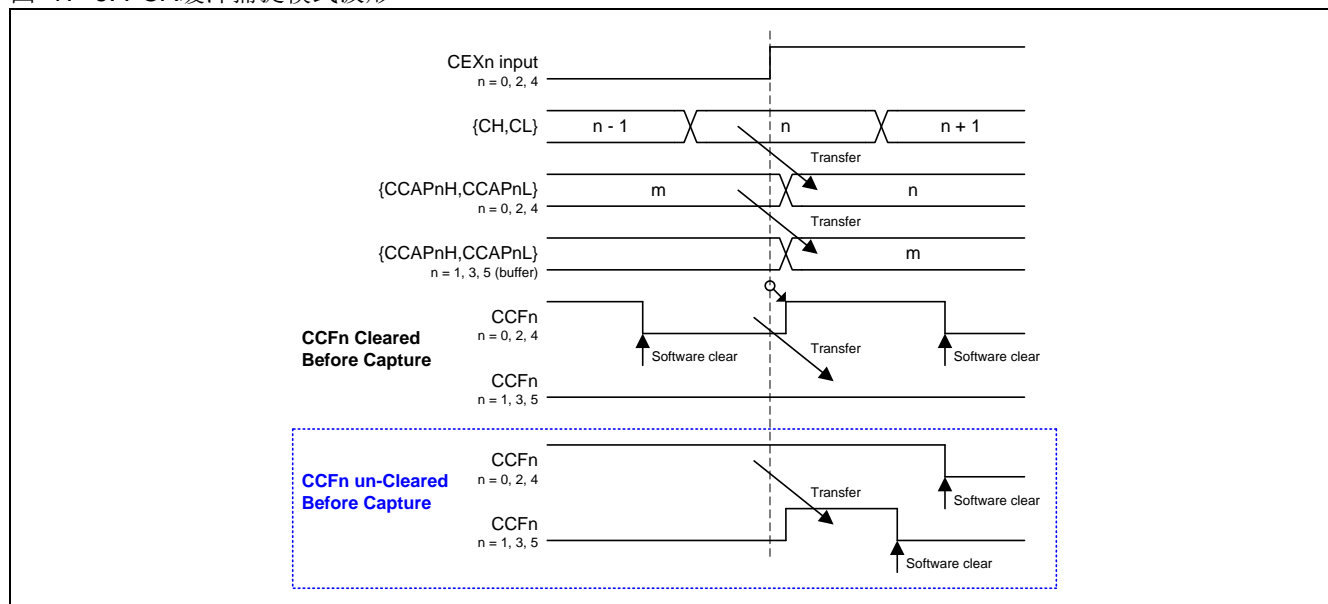
图 17-5. PCA缓冲捕捉模式(BME<sub>n</sub>=1, n= 0, 2, 4)

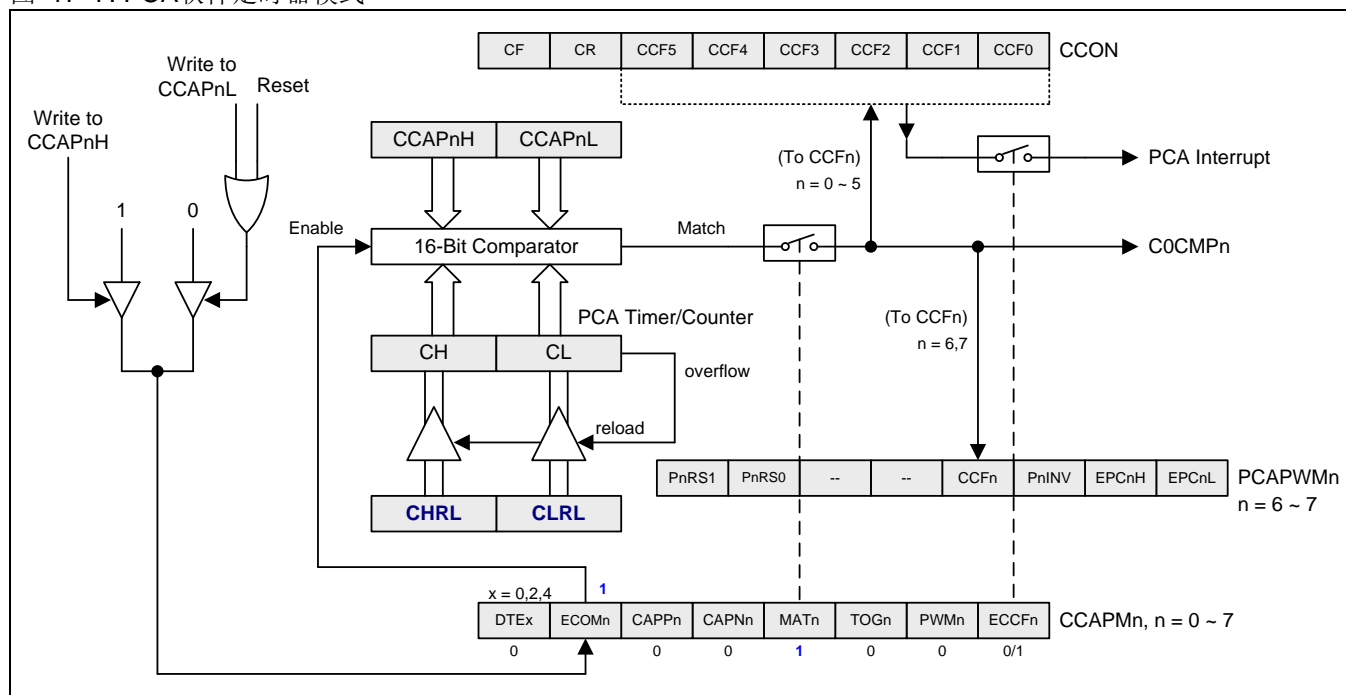
图 17-6. PCA缓冲捕捉模式波形



### 17.4.3. 16 位软件定时器模式(比较模式)

PCA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。PCA定时器与模块的捕获寄存器进行比较，若相等且当CCF<sub>n</sub>和ECCF<sub>n</sub>位同时设置时会产生一个中断信号。

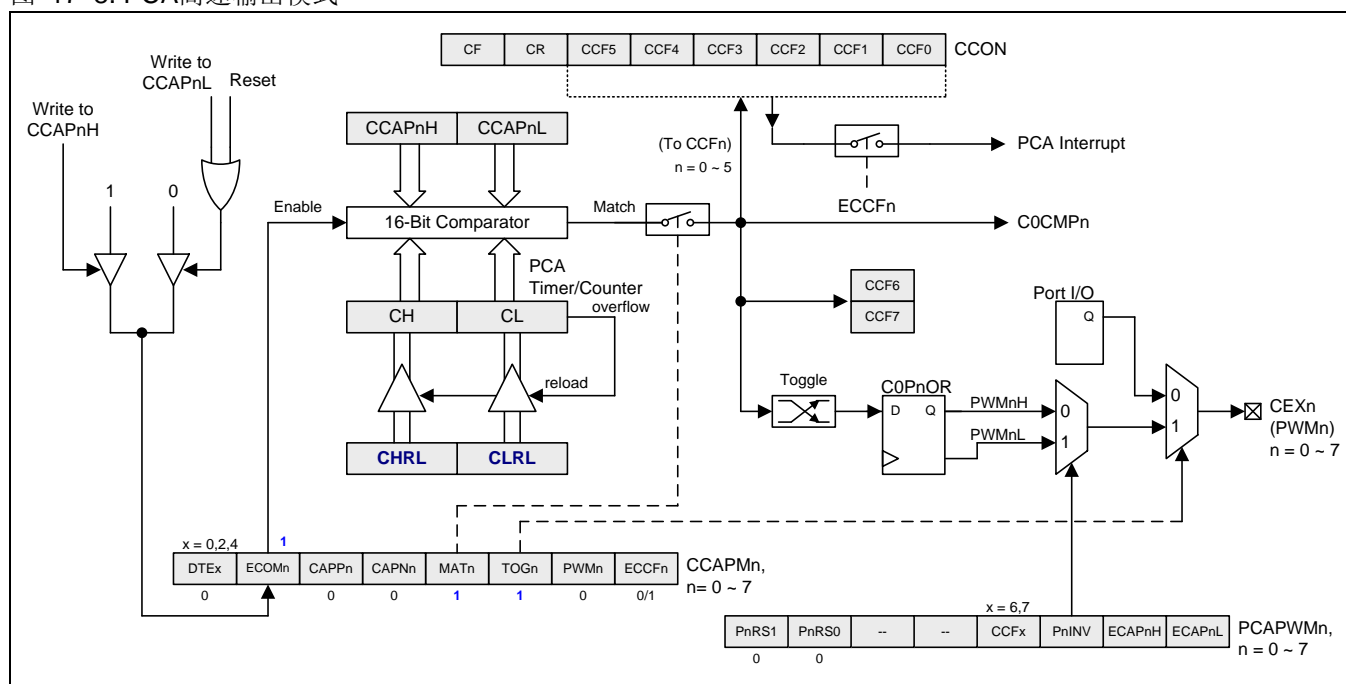
图 17-7. PCA软件定时器模式



### 17.4.4. 高速输出模式(比较输出模式)

这种模式下，每当PCA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. PCA高速输出模式



### 17.4.5. 缓冲 8 位 PWM 模式

所有PCA模块都可用作PWM输出。输出频率取决于PCA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PCA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：

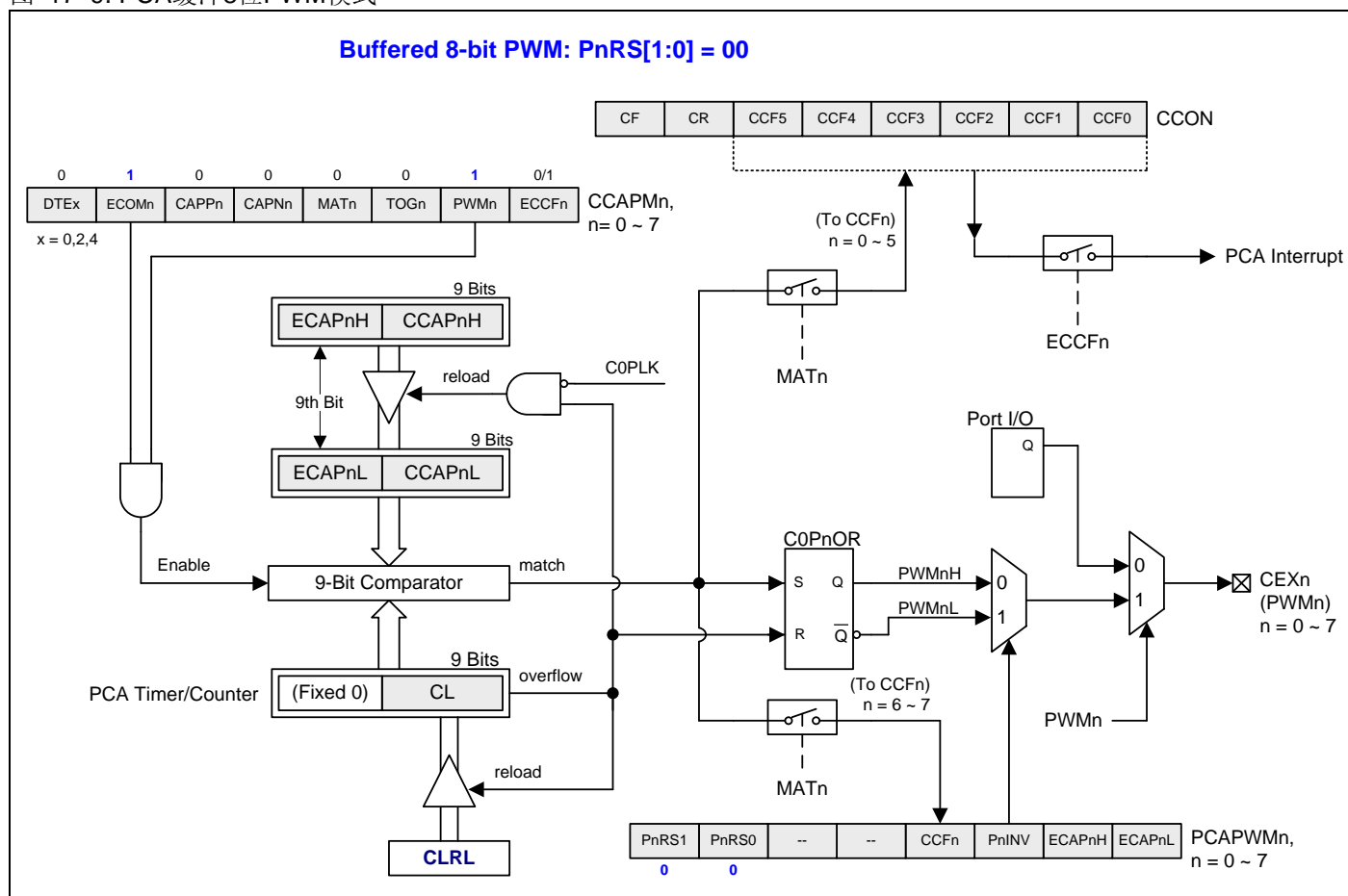
$$Duty\ Cycle = 1 - \{ ECAPnH, [CCAPnH] \} / 256.$$

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，

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

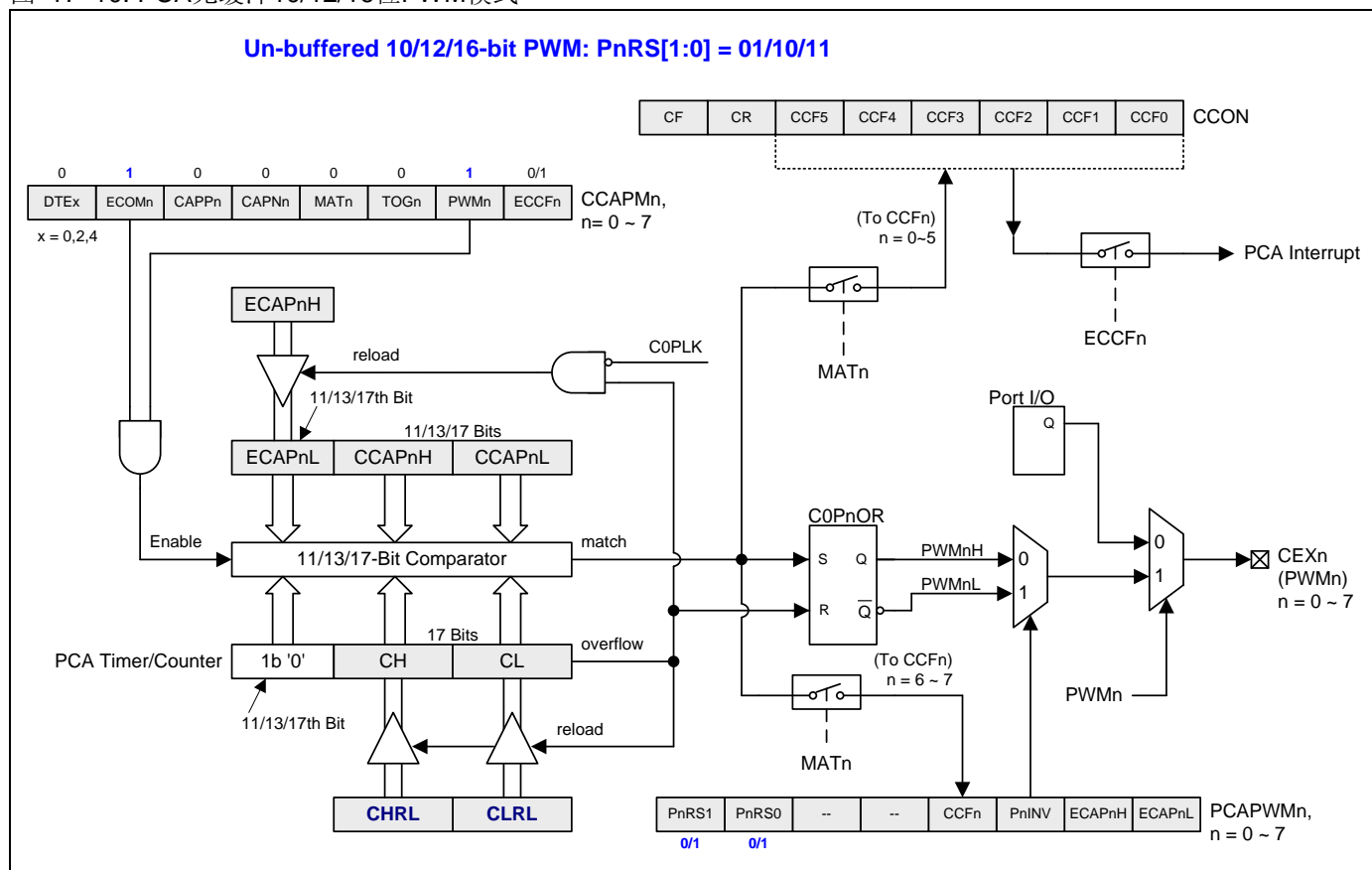
图 17-9. PCA缓冲8位PWM模式



## 17.4.6. 无缓冲 10/12/16 位 PWM 模式

PCA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。

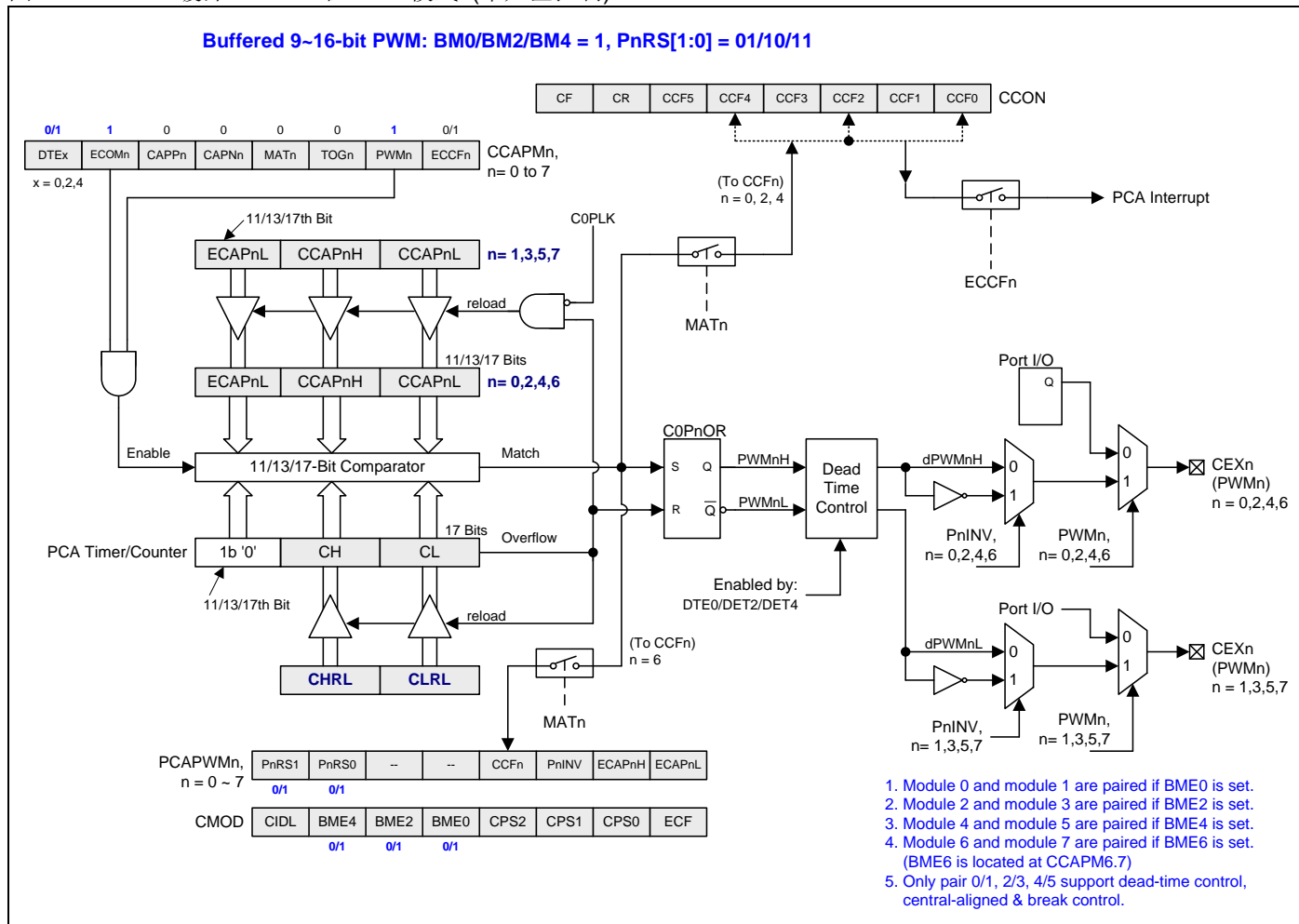
图 17-10. PCA无缓冲10/12/16位PWM模式



## 17.4.7. 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。

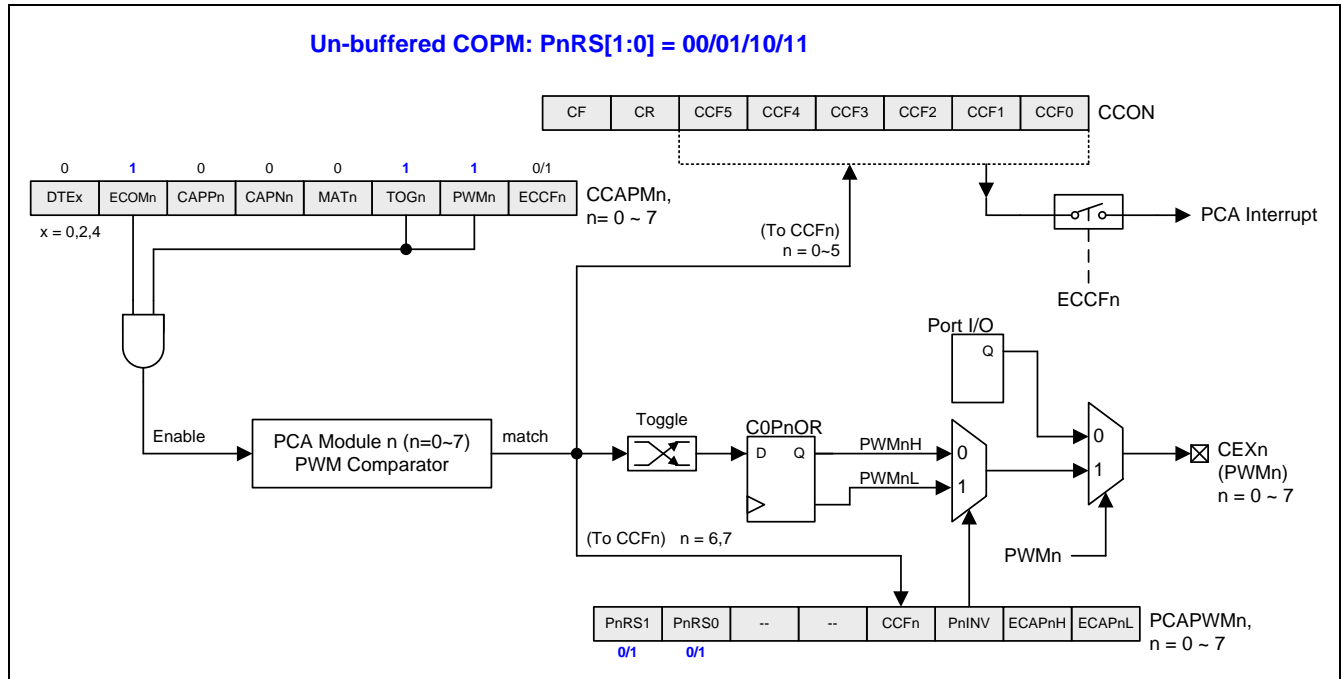
图 17-11. PCA缓冲10/12/16位PWM模式 (带死区控制)



## 17.4.8. COPM 模式

PWM匹配模式下的比较输出与高速输出模式相似，但它使用PCA PWM比较器而不是固定的16位比较器。它为应用程序提供了更多的灵活性。例如，如果PCA比较器使用8位PWM，可以比高速输出模式输出更高的频率。

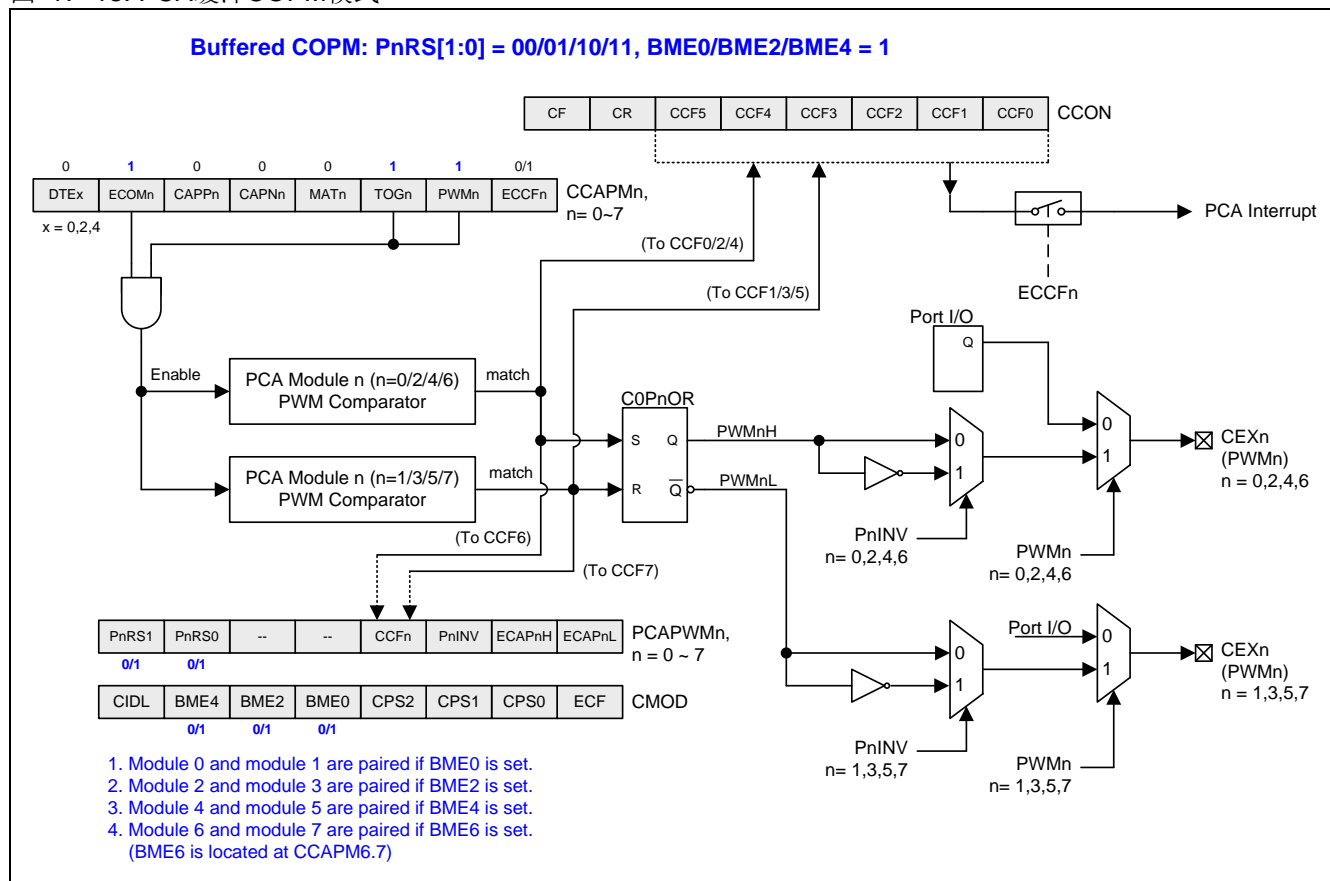
图 17-12. PCA COPM模式



## 17.4.9. 缓冲 COPM 模式

如果应用程序需要对PWM信号进行任何相位控制，则需要设置PCA模块为缓冲COPM模式。一组PCA模块( $n=0&1 / 2&3 / 4&5$ )可以对PWM信号的一个周期的两个边缘的时间延迟进行编程。这意味着你可以设置波形的开始和结束点。当2个或3个相关PWM信号可以设置彼此之间的相移时，这是很有用的。

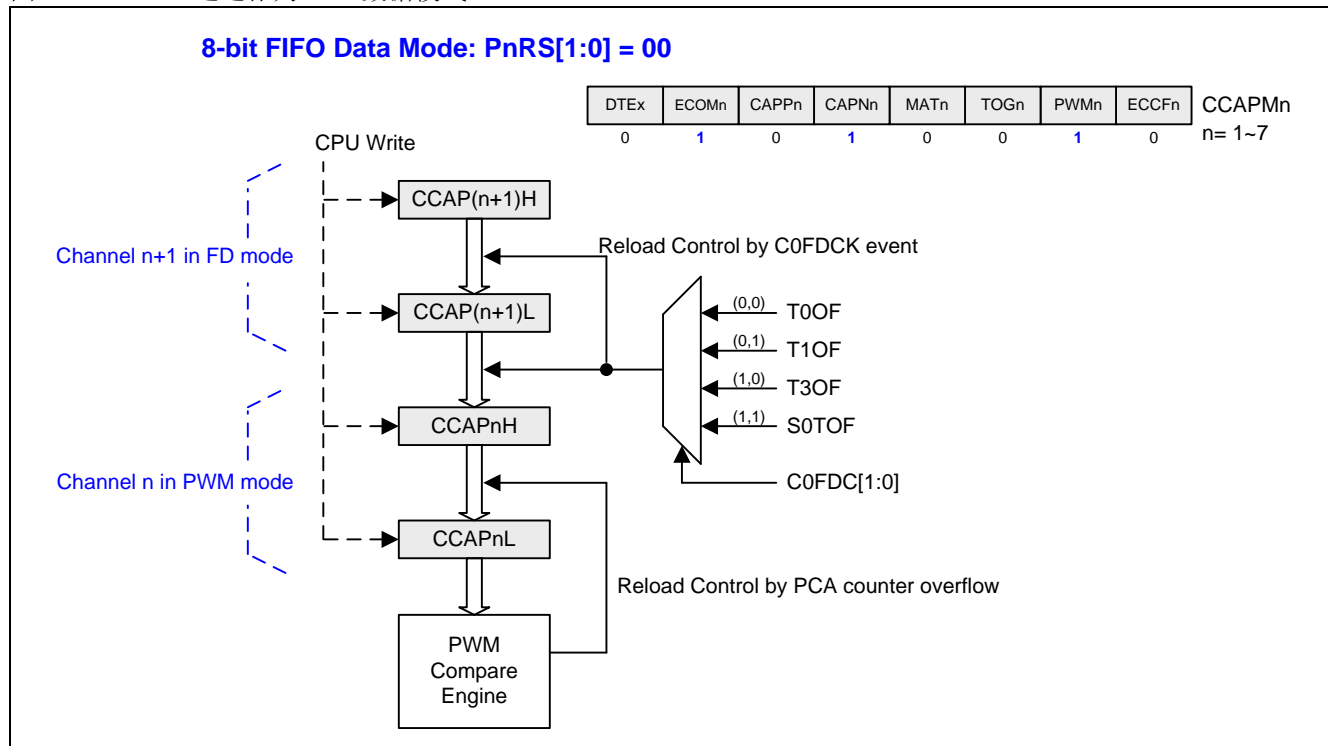
图 17-13. PCA缓冲COPM模式



### 17.4.10. FIFO 数据模式

在这个模式用户能设置CCAPnL、CCAPnH、CCAP(n+1)L和CCAP(n+1)H作为一个缓冲链。在所有这些缓冲设置之后，通过T0OF、T1OF、T3OF或S0TOF顺序触发来改变占空比。这个功能使能，CPU可以离开让它自己运行而获得更多时间做其它的操作。比如，功率变换器从轻载到重载开始升高电压，将更好用于设置比目标开启周期更大的占空比，然后降低占空比一步一步靠近目标占空比。在缓冲中设置所有占空比并且让它自己完成。

图 17-14. PCA通道作为FIFO数据模式



通道FIFO数据模式由C0FDCK移动。

C0FDCK 源选择，更新PCA FIFO数据模式的时钟选择

C0FDC1~0	C0FDCK
00	T0OF
01	T1OF
10	T3OF
11	S0TOF

#### 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
--	--	T1G0	T0G1	C0FDC1	C0FDC0	S1PS1	S1PS0
W	W	R/W	R/W	R/W	R/W	R/W	R/W

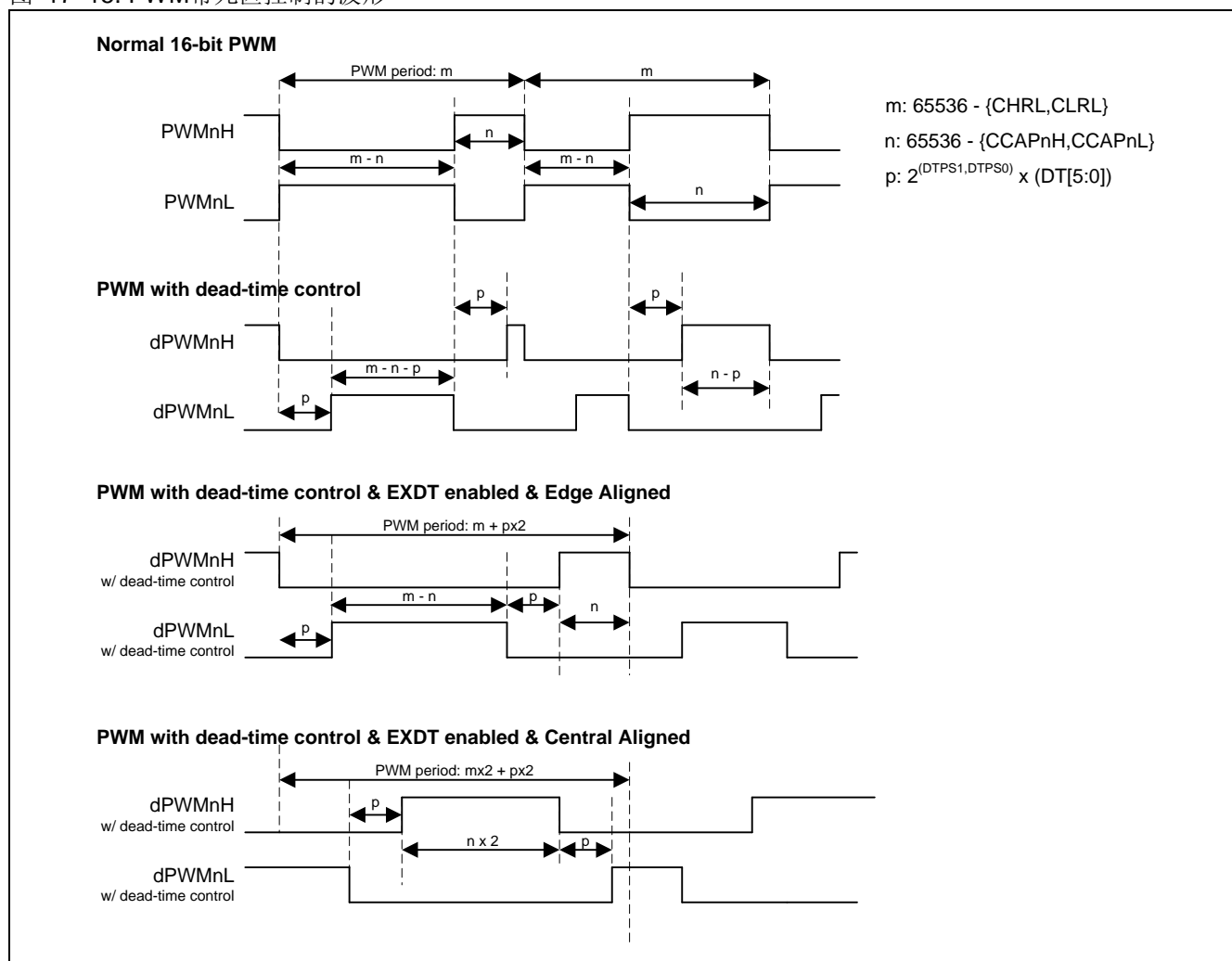
Bit 3~2: C0FDC1~0, C0FDCK选择 [1:0].

C0FDC1~0	C0FDCK
00	T0OF
01	T1OF
10	T3OF
11	S0TOF

## 17.4.11. 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可以分配到每个通道且每个PWM通道可以同时运行在不同的分辨率和不同的相位延时。

图 17-15. PWM带死区控制的波形

**CCAPMn: PCA模块比较/捕捉寄存器, n=0~5**

SFR 页 = 仅 0 页 for n= 0~1 (n=2~5 所有页)

SFR 地址 = 0xDA~0xDF

复位值 = 0000-0000

7	6	5	4	3	2	1	0
DTE <sub>n</sub>	ECOM <sub>n</sub>	CAPP <sub>n</sub>	CAPN <sub>n</sub>	MAT <sub>n</sub>	TOG <sub>n</sub>	PWM <sub>n</sub>	ECCF <sub>n</sub>
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

0: PWM<sub>n</sub>输出禁止死区时间控制。

1: PWM<sub>n</sub>输出使能死区时间控制。

**PDTBCRA: 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
<b>DTPS1</b>	<b>DTPS0</b>	<b>DT5</b>	<b>DT4</b>	<b>DT3</b>	<b>DT2</b>	<b>DT1</b>	<b>DT0</b>
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

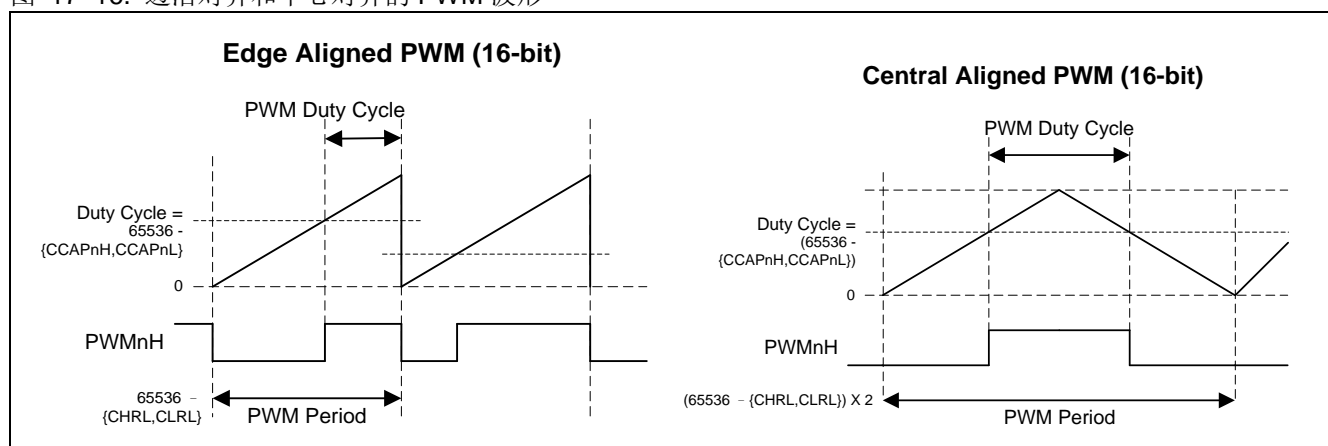
7	6	5	4	3	2	1	0
<b>PCAE</b>	<b>EXDT</b>	<b>PBKM</b>	<b>PBKE1.1</b>	<b>PBKE1.0</b>	<b>PBKE0.2</b>	<b>PBKE0.1</b>	<b>PBKE0.0</b>
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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PWMO0~5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。仅支持CHRL和CLRL设定的8/10/12/16位分辨率。

图 17-16. 边沿对齐和中心对齐的 PWM 波形



Bit 6: EXDT, PWM时间的扩展死区时间。此功能使能将使非PWM通道功能改变为PWM通道。比如捕捉模式, 软件定时器模式和高速输出模式。

0: 禁止  $M + 2P$ 。1: 在使能的PWM通道使能  $M + 2P$ 。

Bit 5: PBKM, PWM中止模式选择。

0: 锁存模式。

1: 逐周期模式。

图 17-17. PWM 中止控制的锁存模式波形

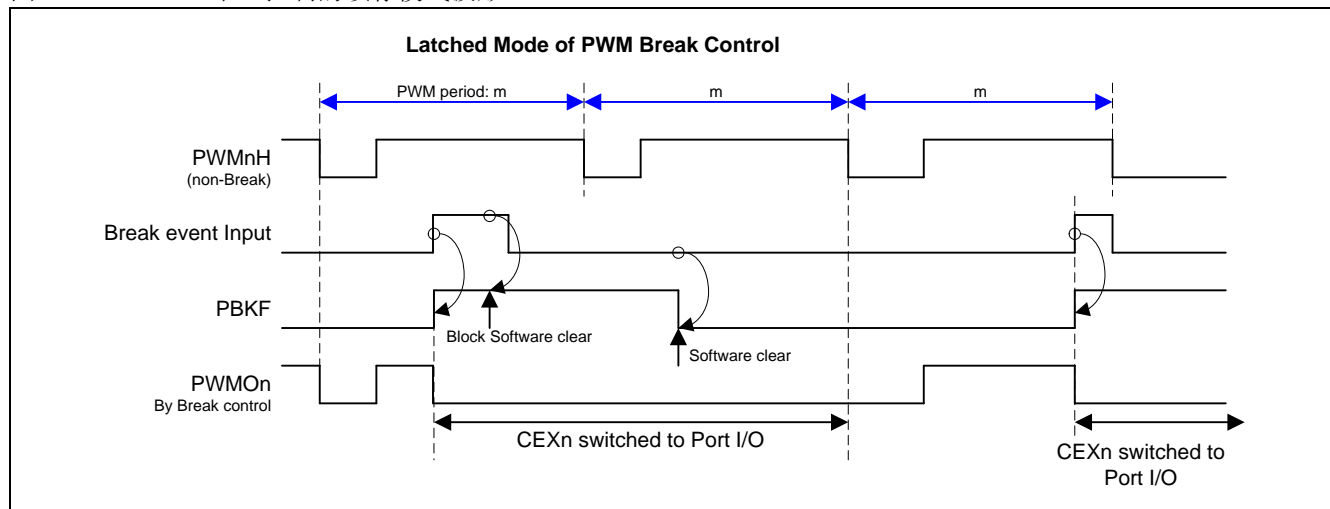
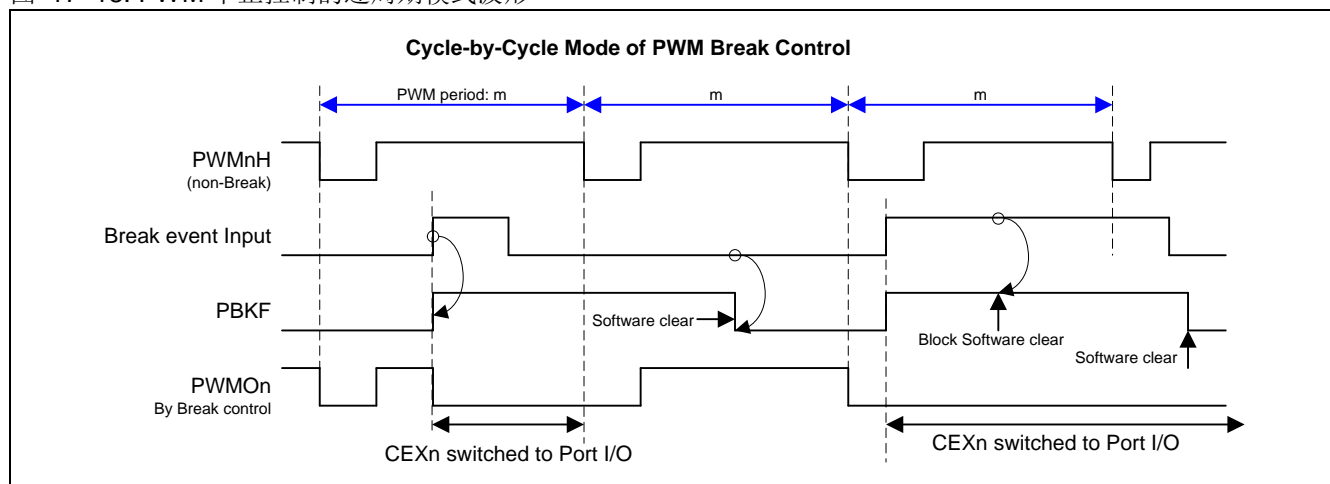


图 17-18. PWM 中止控制的逐周期模式波形



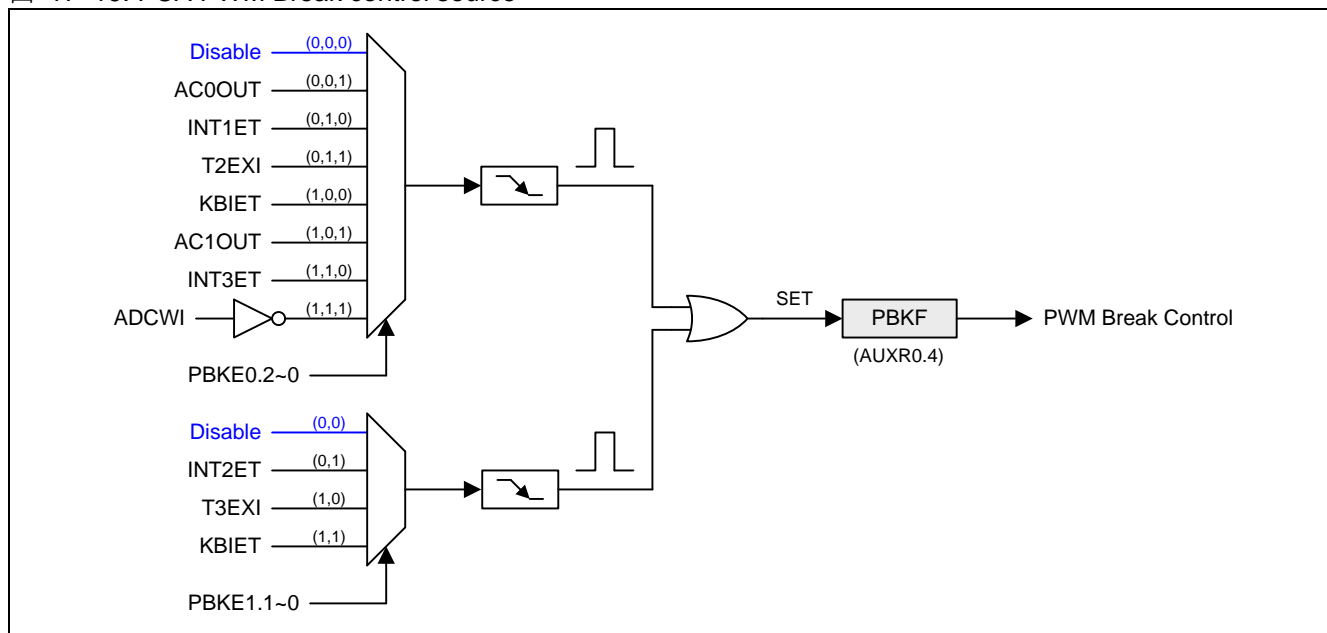
Bit 4~3: PBKE1.1~0, PWM中止使能1选择。此功能仅在CEXn输出模式有效(n=0~5)。

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	KBIET, KBI匹配激活

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 17-19. PCA PWM Break control source

**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
P60OC1	P60OC0	P60FD	PBKF	BM1F	BM0F	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。

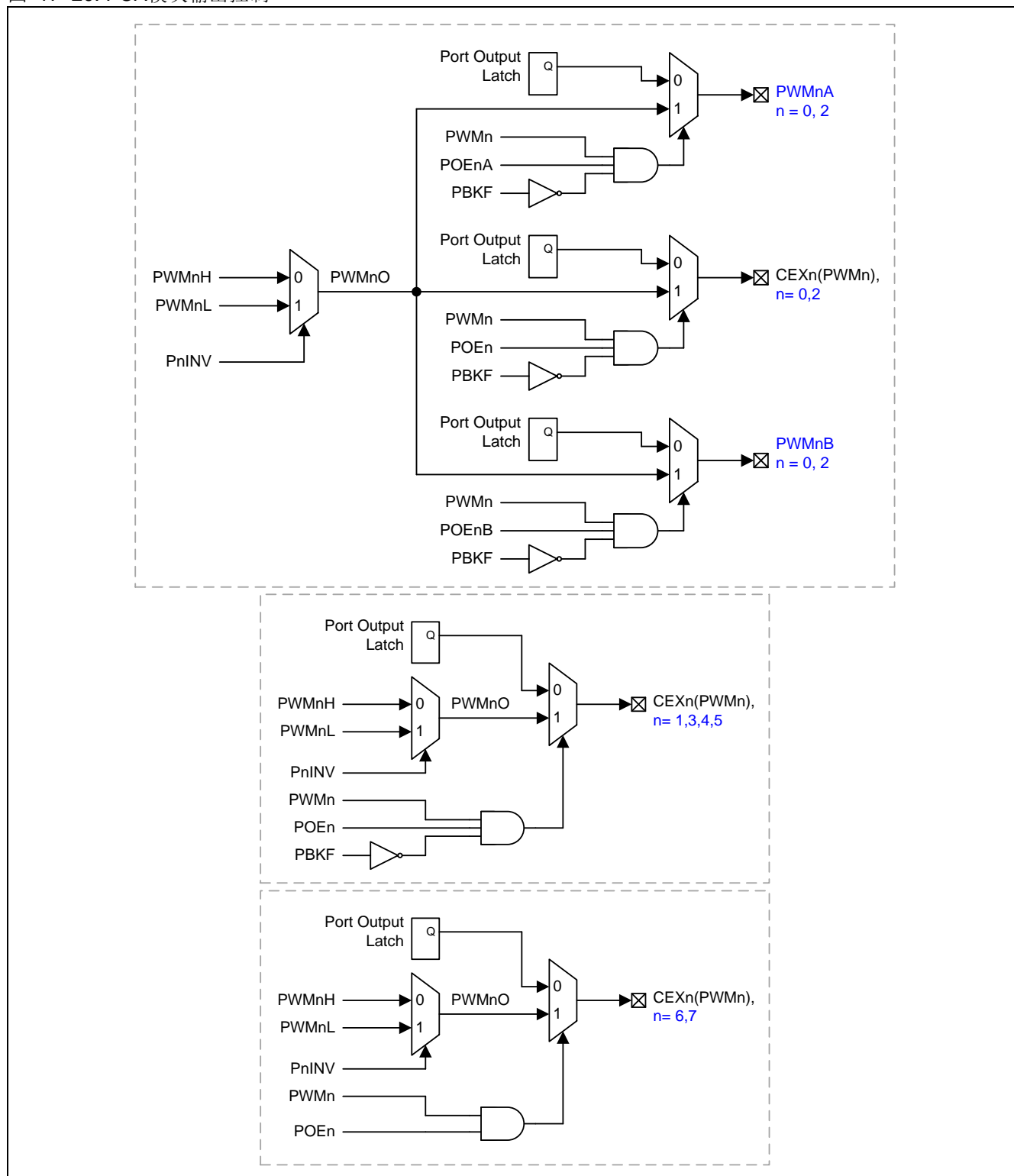
## 17.4.12. PCA 模块输出控制

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、3、4、5、6、7)可编程为一般I/O端口或PCA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。

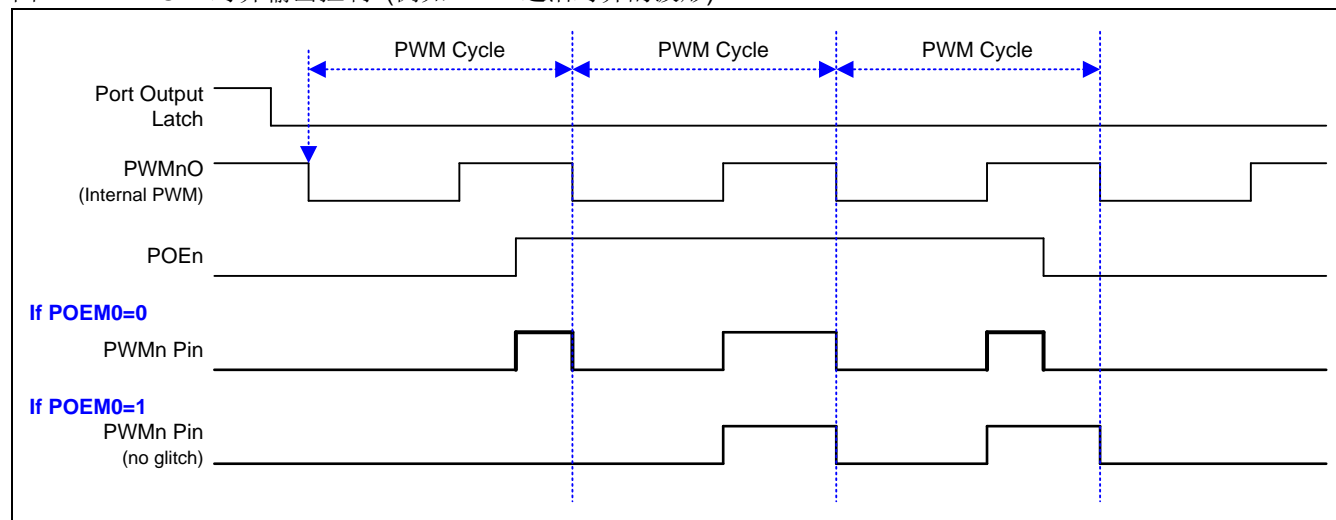
另外，PCA模块0和2在不同的端口引脚有2个“克隆”信号。这三个相同的PWM信号可以被POEnA或POEnB或PWMn屏蔽，用于需要相位控制的应用程序。

图 17-20. PCA模块输出控制



MGEQ1C064 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 17-21 所示。

图 17-21. 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~F

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, PCA 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, PCA PWM2第3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, PCA PWM2第2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, PCA 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, PCA 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, PCA PWM0第3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, PCA PWM0第3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, PCA 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。

**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	0	C0PLK	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: C0PLK, PCA0 PWM/COPM缓冲寄存器更新控制。

0: 使能缓冲PWM/COPM模式的缓冲器由PCA0基准定时器溢出自动更新

1: 禁止缓冲PWM/COPM模式下由PCA0基准定时器溢出对缓冲器自动更新

**AUXR7: 辅助寄存器 7**

SFR 页 = 仅 4 页

SFR 地址 = 0xA4

复位值 = 1100-0000

7	6	5	4	3	2	1	0
POE5	POE4	C0CKOE	SPI0M0	BM1OE1	BM1OE0	BM0OE1	BM0OE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 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 6: 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。

**Bit 5: C0CKOE, PCA时钟输出使能。**

0: 禁止PCA时钟输出。

1: 使能PCA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 = 1100-0000

7	6	5	4	3	2	1	0
POE7	POE6	C0PPS2	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	W	W	R/W	R/W	R/W

Bit 7: POE7 PCA PWM7主通道(PWM7O)输出控制。

0: 禁止PWM7O在端口引脚输出。

1: 使能PWM7O在端口引脚输出。默认是使能。

Bit 6: POE6, PCA PWM6主通道(PWM6O)输出控制。

0: 禁止PWM6O在端口引脚输出。

1: 使能PWM6O在端口引脚输出。默认是使能。

Bit 5: C0PPS2, {PWM6, PWM7} 引脚选择 2.

C0PPS2	PWM6	PWM7
0	P6.2	P6.3
1	P6.0	P6.1

**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
I2C1CF	PAA1	I2C1PS1	I2C1PS0	0	POEM0	COM0	COOFS
R/W	R/W	R/W	R/W	W	R/W	R/W	R/W

Bit 2: POEM0, PCA0 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

**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
C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	C0PS1	C0PS0	ECIPS0	C0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: C0IC4S0, PCA0 输入通道4输入端口引脚选择

C0IC4S0	CEX4 input
0	CEX4 端口Pin
1	T2EXI

Bit 6: C0IC2S0, PCA0输入通道2输入端口引脚选择.

C1IC2S0	CEX2 input
0	CEX2 端口Pin
1	T3EXI

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	P3.6	P3.7
1	P6.0	P6.1

Bit 3: C0PS1, PCA0 端口引脚选择1.

C0PS1	CEX1	CEX3	CEX5
0	P2.3	P2.5	P2.7
1	P3.3	P3.4	P3.5

Bit 2: C0PS0, PCA0端口引脚选择0.

C0PS0	CEX0	CEX2	CEX4
0	P2.2	P2.4	P2.6
1	P4.0	P4.1	P3.2

Bit 1: ECIPS0, PCA0 ECI 端口引脚选择0.

ECIPS0	ECI
0	P2.1
1	P4.6

Bit 0: C0COPS, PCA0 时钟 (C0CKO) 端口引脚选择

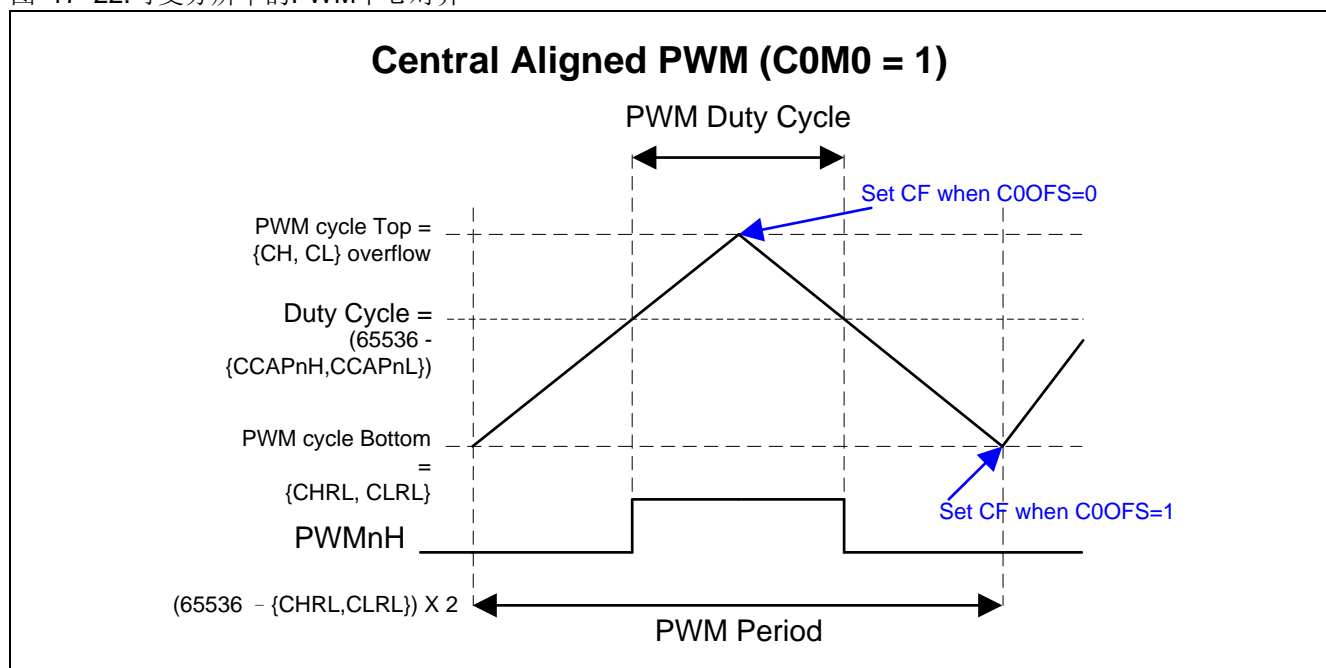
C0COPS	C0CKO
0	P4.7
1	P7.2

### 17.4.13. 可变分辨率的 PWM 中心对齐

在章节“17.4.11 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。

如果需要应用可变分辨率的PWM中心对齐，要同时置位PCAE及C0M0使能此功能。当可变分辨率的中心对齐模式使能，PCA在其它非PWM模块中能支持所有比较模式。请注意当C0M0 = 1使用PWM中心对齐，基准定时器需要使用16位0xFFFF减去{CCAPnH, CCAPnL}的值避免预期外的错误。

图 17-22.可变分辨率的PWM中心对齐



#### 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
I2C1CF	PAA1	I2C1PS1	I2C1PS0	0	POEM0	C0M0	C0OFS
R/W	R/W	R/W	R/W	W	R/W	R/W	R/W

Bit 2: POEM0, PCA0 POEn控制 0.

0: POEn功能在CPU写入之后立即激活

1: POEn功能与PWM周期对齐

Bit 1: C0M0, PCA模式控制0.

0: PWM中心对齐不支持可变分辨率。

1: 使能PCA支持可变分辨率的PWM中心对齐。使能此功能，PCAE (PWMCR.7)也需要置位。

Bit 0: C0OFS, 当C0M0使能PCA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。

## 18. 串口 0 (UART0)

**MGEQ1C064**支持一个全双工的串口，意思是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。

### 18.1. 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 18.11 串口0增强功能。

表 18.1. 串口0模式选择

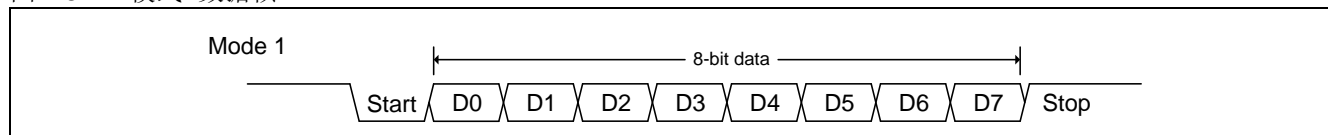
SM30,SM00,SM10	S0RCK	S0TCK	MODE	功能	波特率时基	注意
000	0	0	0	移位寄存器	SYSClk/12 or SYSClk/4 (URM0X3=1)	
001	0	0	1	8位UART	定时器1或定时器2溢出	当SMOD1 & SMOD2 =1，计数器不能为 满额-1 或 满额-2 (如. 254, 255, 65534, 65535)
010	0	0	2	9位UART	SYSClk/64, /32, /16, or /8	
011	0	0	3	9位UART	定时器1或定时器2溢出	当SMOD1 & SMOD2 =1，计数器不能为 满额-1 或 满额-2 (如. 254, 255, 65534, 65535)
100	0	0	4	SPI主机	SYSClk/12 or SYSClk/4 (URM0X3=1)	
000	0	1	增强	移位寄存器	S0BRG溢出	S0BRT 不能为255
001	0/1	0/1	增强	8位UART	TX或RX可选S0BRG 溢出	SMOD1 & SMOD2 不能同时为1
010	0	1	增强	9位UART	TX: S0BRG 溢出 RX: SYSClk/64, /32 or /16	SMOD1 & SMOD2 不能同时为1
010	1	0	增强	9位 UART	TX: SYSClk/64, /32 or /16 RX: S0BRG 溢出	SMOD1 & SMOD2 不能同时为1
010	1	1	增强	纯定时器	仅定时器功能	
011	0/1	0/1	增强	9位 UART	TX或RX可选S0BRG 溢出	SMOD1 & SMOD2 不能同时为1
100	0	1	增强	SPI主机	S0BRG 溢出	S0BRT 不能为255
101	1	1	增强	LIN总线	S0BRG 溢出和自动波特率	SMOD1 & SMOD2 不能同时为1
其它				保留		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18.11 串口0增强功能的详细描述

**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S0CFG寄存器的URM0X3位选择为系统时钟频率的1/12或1/4。**MGEQ1C064**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 18-4和图 18-5所示模式0的时钟极性波形。

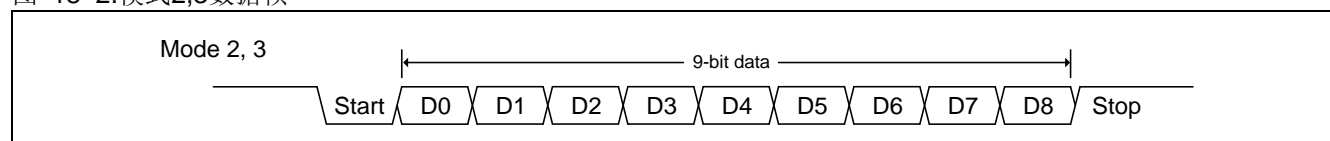
**模式 1:** 10 位通过TXD0 传送或通过 RXD0 接收，数据帧包括一个起始位(0)，8 个数据位(低位优先)，和一个停止位(1)(如图 18-1所示)。在接收时，停止位进入到专用寄存器(S0CON)的RB80。波特率是可变的。

图 18-1. 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 18-2.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 18-2.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 时启动接收。在其它模式，在 REN0=1 时，收到起始位时启动接收。

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。

## 18.2. 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S0CFG寄存器中的URM0X3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 18-3。串口0模式0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。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 18-4。

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。

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)脚并移到寄存器中。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 18-5。

图 18-3. 串口0模式0

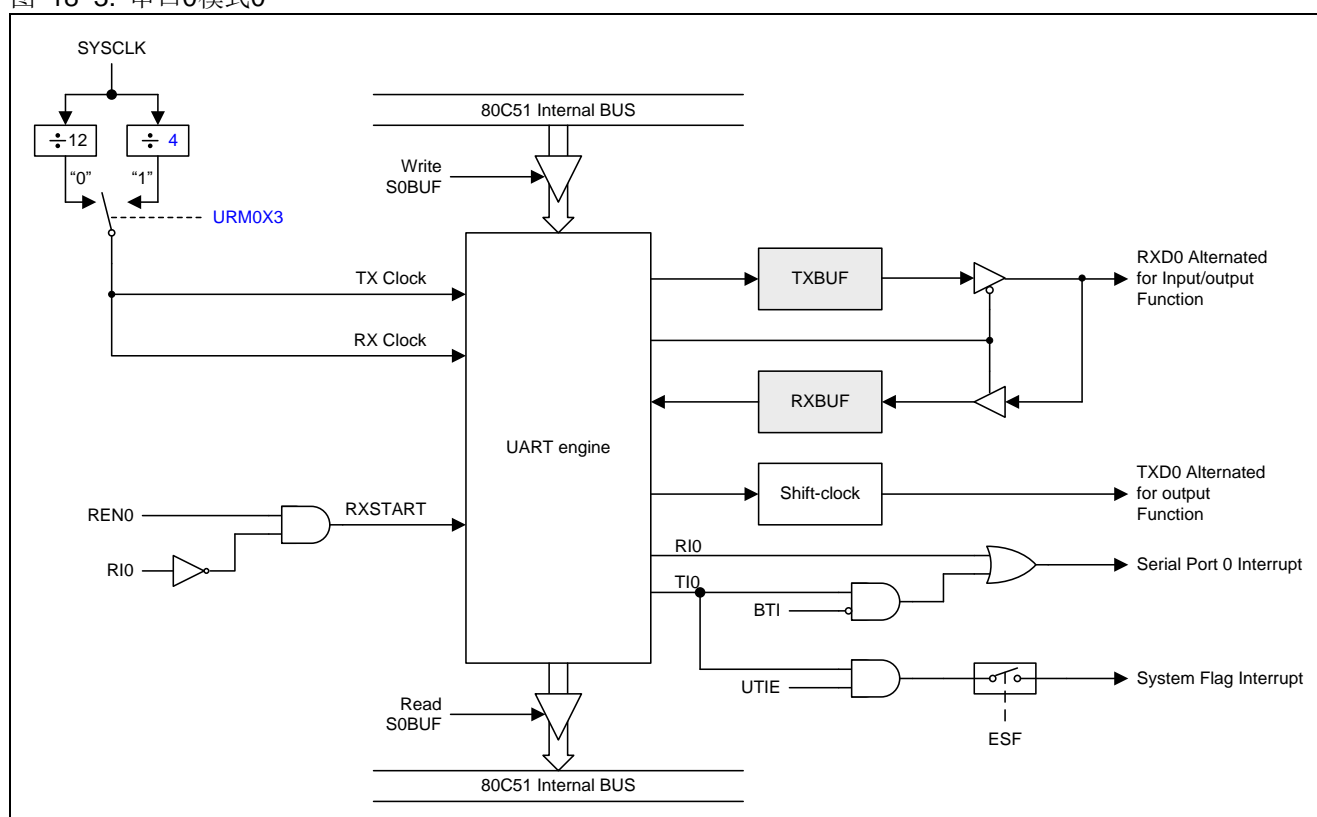


图 18-4. 模式 0 发送波形

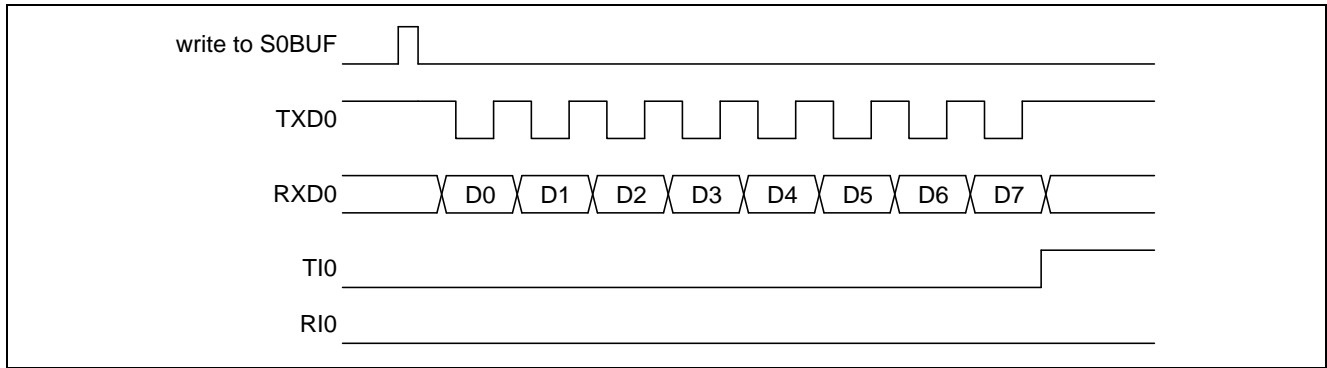
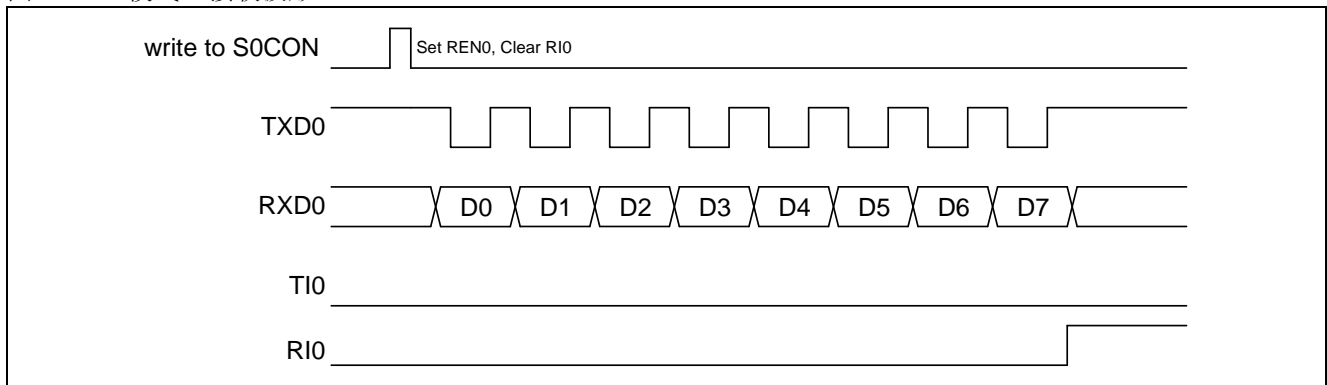


图 18-5. 模式 0 接收波形



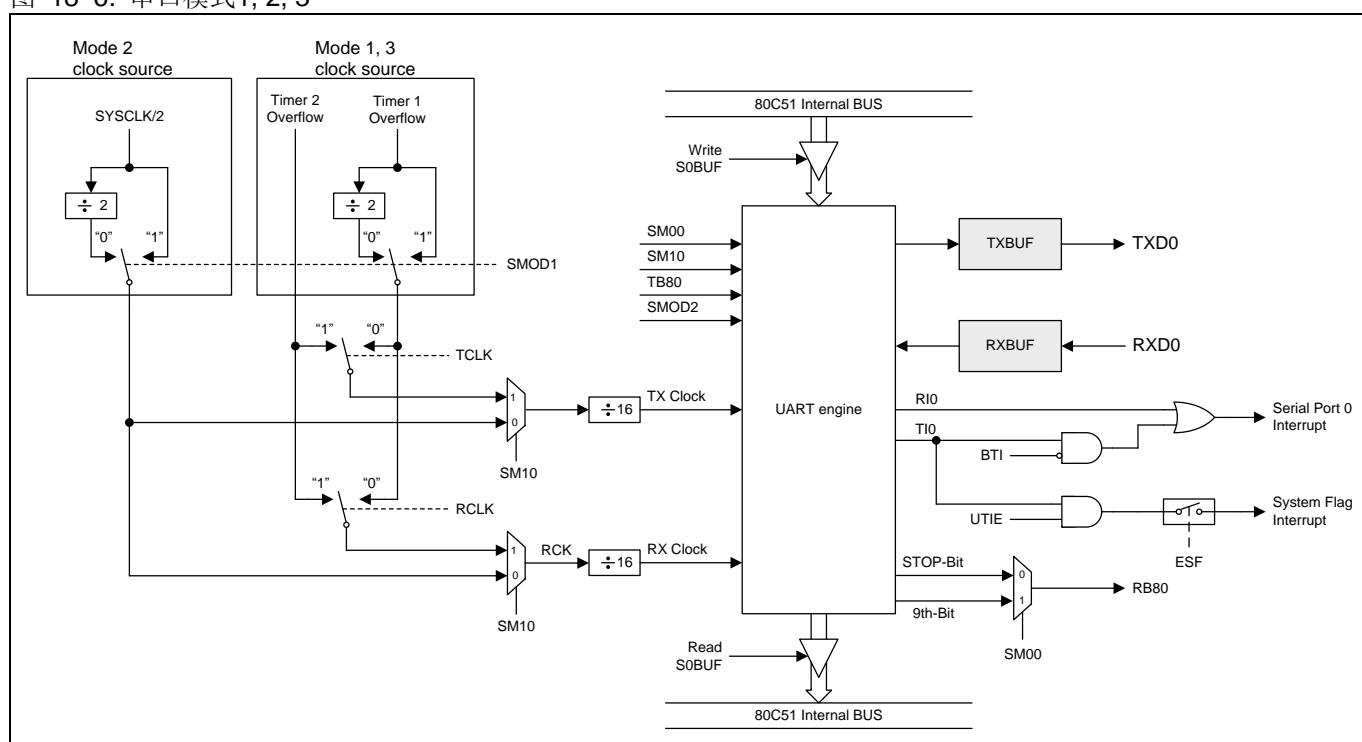
### 18.3. 串口0模式1

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18-1所示并且模式1的简化功能框图如图18-6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18-1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 18-6. 串口模式1, 2, 3



## 18.4. 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(TB8)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 18-2所示，模式2和模式3的串口0功能框图如图 18-5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 18-2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。

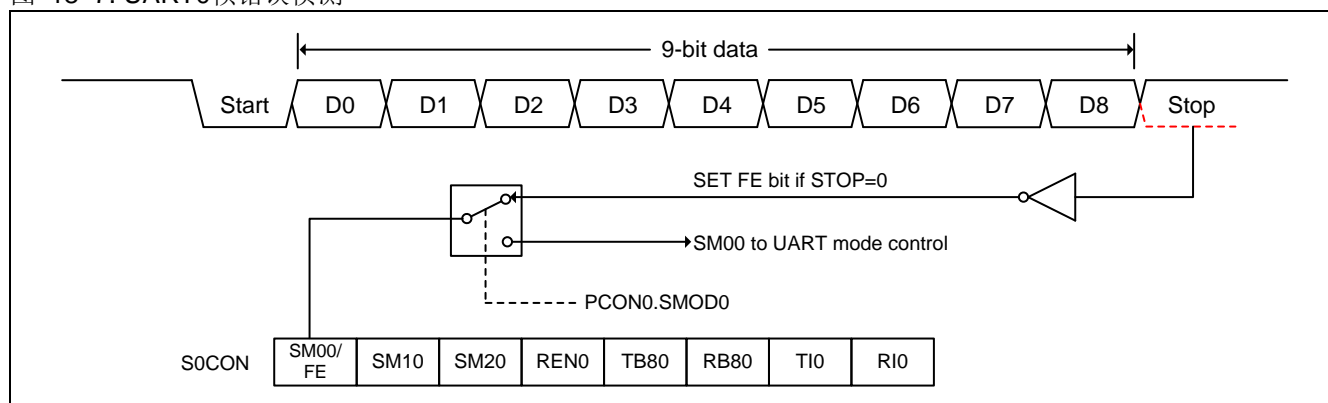
当串口0控制器在RCK采样时钟下检测到在RXD0有负跳变的起始位时接收开始。在RXD0引脚上的数据将被串口0的位侦测器采样。当收数据接收完后，硬件置位RI0表示接收结束并把第9位加载到S0CON寄存器的RB80。

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RENO=1时启动接收。在其它模式，在 RENO=1 时，收到有负跳变的起始位时启动接收。

## 18.5. 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 18-7。。

图 18-7. UART0帧错误侦测



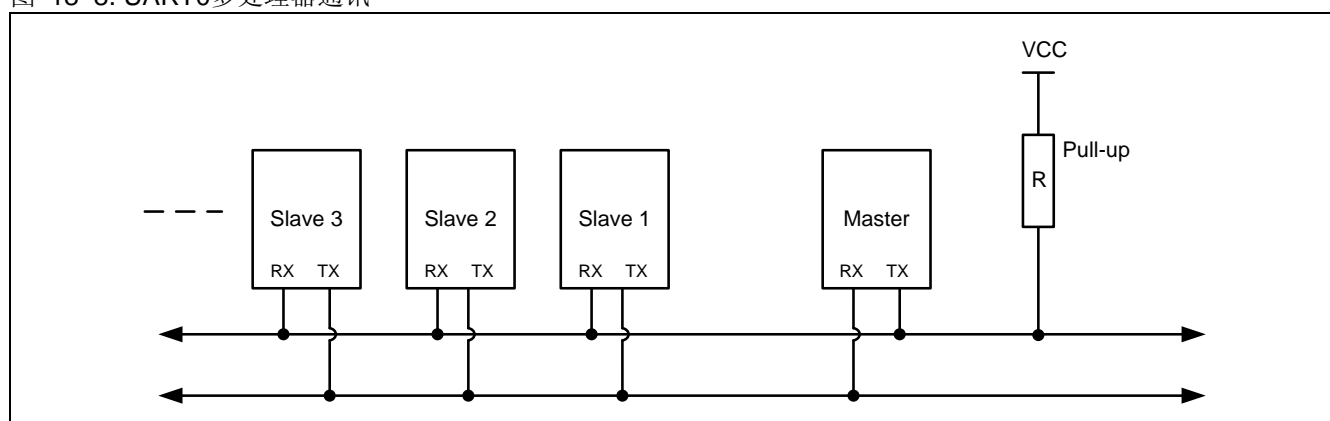
## 18.6. 多处理器通讯

模式2和3在用作多处理器通讯时有特殊的规定如图 18-8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 18-8. UART0多处理器通讯

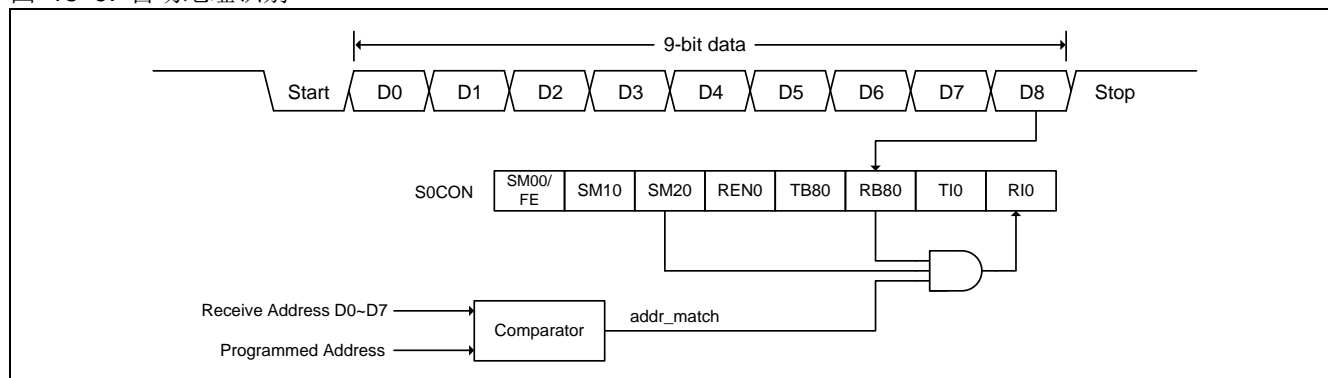


## 18.7. 自动地址识别

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SOCON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 18-9。

图 18-9. 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：

<b>从机 0</b>	<b>从机 1</b>
SADDR = 1100 0000	SADDR = 1100 0000
SADEN = 1111 1101	SADEN = 1111 1110
Given = 1100 00X <b>0</b>	Given = 1100 00 <b>0</b> 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：

<b>从机 0</b>	<b>从机1</b>	<b>从机2</b>
SADDR = 1100 0000	SADDR = 1110 0000	SADDR = 1110 0000
SADEN = 1111 1001	SADEN = 1111 1010	SADEN = 1111 1100
Given = 1100 0XX <b>0</b>	Given = 1110 0X <b>0</b> X	Given = 1110 0 <b>0</b> 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。

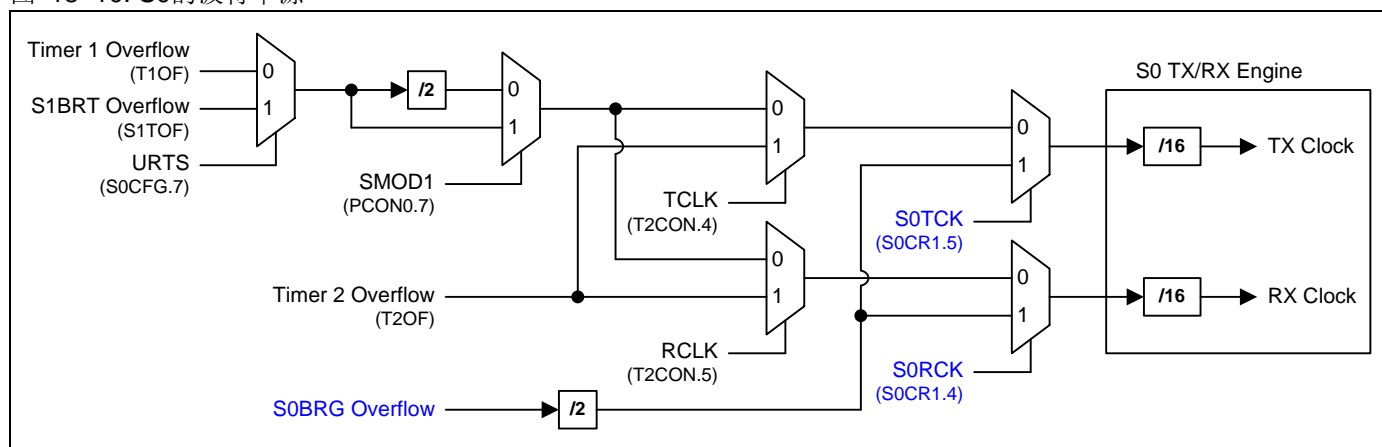
## 18.8. 波特率设置

位T2X12 (T2MOD.4)、T1X12 (AUXR2.3)、URM0X3 (S0CFG.5)和SMOD2 (S0CFG.6)提供一个的波特率选项设置,如下所列:

### 18.8.1. 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 UART1(模式 1 或模式 3)也是这样运行, 这两个串行口(UART)将有同样的波特率。来源 S0BRG,请参考 18.11.1 S0 波特率发生器(S0BRG)

图 18-10. S0的波特率源



### 18.8.2. 模式 0 波特率

$$\text{Mode 0 Baud Rate} = \frac{F_{\text{SYSCLK}}}{n} \quad ; n=12, \text{ if } \text{URM0X3}=0 \\ ; n=4, \text{ if } \text{URM0X3}=1$$

注意:

如果URM0X3=0, 波特率公式跟标准8051一样。

## 18.8.3. 模式 2 波特率

$$\text{Mode 2 Baud Rate} = \frac{2^{\text{SMOD1}} \times 2^{(\text{SMOD2} \times 2)}}{64} \times F_{\text{SYSCLK}}$$

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 18.2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 18.2. 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 <b>X2</b>	增强型功能	± 2%
1		1	双倍波特率 <b>X4</b>	增强型功能	± 1%

注意: 当定时器1在双倍波特率x4(SMOD1=1和SMOD2=1)模式时, TH1不能等于254和255。

表 18.3. S0模式2波特率 @ F<sub>SYSCLK</sub>=11.0592MHz

波特率	SMOD2	SMOD1	误差
172,800	0	0	0.0%
345,600	0	1	0.0%
691,200	1	0	0.0%
1,382,400	1	1	0.0%

表 18.4. S0模式2波特率 @ F<sub>SYSCLK</sub>=12.00MHz

波特率	SMOD2	SMOD1	误差
187,500	0	0	0.0%
375,000	0	1	0.0%
750,000	1	0	0.0%
1,500,000	1	1	0.0%

## 18.8.4. 模式 1 &amp; 3 波特率

## 18.8.4.1. 使用定时器 1 作为波特率发生器

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}} \times 2^{(\text{SMOD2} \times 2)}}{32} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{TH1})} \quad ; \text{T1X12}=0$$

; T1 CLK Source = SYSCLK/12

$$\text{or} = \frac{2^{\text{SMOD1}} \times 2^{(\text{SMOD2} \times 2)}}{32} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{TH1})} \quad ; \text{T1X12}=1$$

; T1 CLK Source = SYSCLK

注意:

如果SMOD2=0, T1X12=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 18.5定义了定时器1波特率发生器由SMOD2因数决定的波特率设置。

表 18.5. SMOD2在模式1和3使用定时器1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 <b>X2</b>	增强型功能	± 2%
1	1	双倍波特率 <b>X4</b>	增强型功能	± 1%

注意: 当定时器1在双倍波特率x4(SMOD1=1和SMOD2=1)模式时, TH1不能等于254和255。

表 18.6 ~ 表 18.21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<sub>SYSCLK</sub> = 48MHz时, 频率最大是6MHz。

表 18.6. 定时器1产生的常用波特率 @ F<sub>SYSCLK</sub>=11.0592MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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 18.7. 定时器1产生的高波特率 @ F<sub>SYSCLK</sub> = 11.0592MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD=0	SMOD=1	误差
230.4K	--	255	0.0%	250	244	0.0%
460.8K	--	--	--	253	250	0.0%
691.2K	--	--	--	254	252	0.0%
921.6K	--	--	--	--	253	0.0%
1.3824M	--	--	--	255	254	0.0%
2.7648M	--	--	--	--	255	0.0%

表 18.8. 定时器1产生的常用波特率@ F<sub>SYSClk</sub>=22.1184MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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%
460800	--	--	--	--	253	0.0%

表 18.9. 定时器1产生的高波特率@ F<sub>SYSClk</sub>=22.1184MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD=0	SMOD=1	误差
460.8K	--	255	0.0%	250	244	0.0%
691.2K	--	--	--	252	248	0.0%
921.6K	--	--	--	253	250	0.0%
1.3824M	--	--	--	254	252	0.0%
1.8432M	--	--	--	--	253	0.0%
2.7648M	--	--	--	255	254	0.0%
5.5296M	--	--	--	--	255	0.0%

表 18.10. 定时器1产生的常用波特率@ F<sub>SYSClk</sub>=12.0MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD=0	SMOD=1	误差	SMOD=0	SMOD=1	误差
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	--	--	--	--	--	--

表 18.11. 定时器1产生的高波特率@ F<sub>sysclk</sub>=12.0MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD=0	SMOD=1	误差
115.2K	--	--	--	243	230	0.16%
230.4K	--	--	--	--	243	0.16%
460.8K	--	--	--	--	--	--

表 18.12. 定时器1产生的常用波特率@ F<sub>sysclk</sub>=24.0MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD=0	SMOD=1	误差	SMOD=0	SMOD=1	误差
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 18.13. 定时器1产生的高波特率@ F<sub>sysclk</sub>=24.0MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD=0	SMOD=1	误差
230.4K	--	--	--	243	230	0.16%
460.8K	--	--	--	--	243	0.16%
691.2K	--	--	--	--	--	--
921.6K	--	--	--	--	--	--

表 18.14. 定时器1产生的常用波特率@ F<sub>sysclk</sub>=29.4912MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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%
230.4K	--	--	--	252	248	0.0%
460.8K	--	--	--	254	252	0.0%
921.6K	--	--	--	255	254	0.0%
1.8432M	--	--	--	--	255	0.0%

表 18.15. 定时器1产生的高波特率@ F<sub>sysclk</sub>=29.4912MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
1.8432M	--	--	--	254	252	0.0%
2.7648M	--	--	--	--	--	--
3.6864M	--	--	--	--	254	--

表 18.16. 定时器1产生的常用波特率@ F<sub>sysclk</sub>=44.2368MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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%
230.4K	--	255	0.0%	250	244	0.0%
460.8K	--	--	--	253	250	0.0%
921.6K	--	--	--	--	253	0.0%
2.7648M	--	--	--	--	255	0.0%

表 18.17. 定时器1产生的高波特率@ F<sub>sysclk</sub>=44.2368MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
2.7648M	--	--	--	254	252	0.0%
3.6864M	--	--	--	--	253	0.0%
5.5296M	--	--	--	--	254	0.0%

表 18.18. 定时器1产生的常用波特率@ F<sub>sysclk</sub>=32MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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 18.19. 定时器1产生的高波特率@ F<sub>sysclk</sub>=32MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
38400	--	--	--	152	48	0.16%
57600	--	--	--	--	117	-0.08%
115200	--	--	--	--	187	0.64%
230.4K	--	--	--	--	221	-0.79%

表 18.20. 定时器1产生的常用波特率@ F<sub>sysclk</sub>=48.0MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=0			T1X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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%
230.4K	--	--	--	--	243	0.16%

表 18.21. 定时器1产生的高波特率@ F<sub>sysclk</sub>=48.0MHz

波特率	TH1,重载值					
	T1X12=0 & SMOD2=1			T1X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
230.4K	--	--	--	230	204	0.16%
460.8K	--	--	--	243	230	0.16%
921.6K	--	--	--	--	243	0.16%

## 18.8.4.2. 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:

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD2} \times (\text{SMOD1}+1)} \times F_{\text{SYSCLK}}}{32 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))} \quad ; \text{T2X12}=0$$

$$\text{OR} = \frac{2^{\text{SMOD2} \times (\text{SMOD1}+1)} \times F_{\text{SYSCLK}}}{16 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))} \quad ; \text{T2X12}=1$$

; T2 CLK Source = SYSCLK/12  
; T2 CLK Source = 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 18.22 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 18.22. SMOD2 在模式 1 和 3 使用定时器 2 的应用标准

SMOD2	SMOD1	波特率	备注	推荐的最大接收误差 (%)
0	X	缺省波特率	标准功能	± 3%
1	0	双倍波特率	增强型功能	± 3%
1	1	双倍波特率X2	增强型功能	± 2%

注意: 当定时器2在双倍波特率x2(SMOD1=1和SMOD2=1)模式时, RCAP2H和RCAP2L不能等于65534和65535。

表 18.23 ~ 表 18.38 列出了定时器2中各种常用的波特率和怎样获得

表 18.23. 定时器2产生的常用波特率@ F<sub>SYSCLK</sub>=11.0592MHz

波特率	[RCAP2H, RCAP2L], 重载值					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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	65533	0.0%

表 18.24. 定时器2产生的高波特率@ F<sub>SYSCLK</sub>=11.0592MHz

波特率	[RCAP2H, RCAP2L], 重载值					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD=0	SMOD=1	误差
230.4K	65533	65530	0.0%	65530	65524	0.0%
460.8K	--	65533	0.0%	65533	65530	0.0%
691.2K	65535	65534	0.0%	65534	65532	0.0%
921.6K	--	--	--	--	65533	0.0%
1.3824M	--	65535	0.0%	65535	65534	0.0%
2.7648M	--	--	--	--	65535	0.0%

表 18.25. 定时器2产生的常用波特率@ F<sub>sysclk</sub>=22.1184MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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%
460800	--	--	--	65533	65533	0.0%

表 18.26. 定时器2产生的高波特率@ F<sub>sysclk</sub>=22.1184MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD=0	SMOD=1	误差
460.8K	65533	65530	0.0%	65530	65524	0.0%
691.2K	65534	65532	0.0%	65532	65528	0.0%
921.6K	--	65533	0.0%	65533	65530	0.0%
1.3824M	65535	65534	0.0%	65534	65532	0.0%
1.8432M	--	--	--	--	65533	0.0%
2.7648M	--	65535	0.0%	65535	65534	0.0%
5.5296M	--	--	--	--	65535	0.0%

表 18.27. 定时器2产生的常用波特率@ F<sub>sysclk</sub>=12.0MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD=0	SMOD=1	误差	SMOD=0	SMOD=1	误差
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	--	--	--	--	--	--

表 18.28. 定时器2产生的高波特率@ F<sub>sysclk</sub>=12.0MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD=0	SMOD=1	误差
115.2K	--	65523	0.16%	65523	65510	0.16%
230.4K	--	--	--	--	65523	0.16%

表 18.29. 定时器2产生的常用波特率@ F<sub>sysclk</sub>=24.0MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD=0	SMOD=1	误差	SMOD=0	SMOD=1	误差
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%

表 18.30. 定时器2产生的高波特率@ F<sub>sysclk</sub>=24.0MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD=0	SMOD=1	误差
230.4K	--	65523	0.16%	65523	65510	0.16%
460.8K	--	--	--	--	65523	0.16%

表 18.31. 定时器2产生的常用波特率@ F<sub>sysclk</sub>=29.4912MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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%
230.4K	65532	65532	0.0%	65528	65528	0.0%
460.8K	65534	65534	0.0%	65532	65532	0.0%
921.6K	65535	65535	0.0%	65534	65534	0.0%

表 18.32. 定时器2产生的高波特率@ F<sub>sysclk</sub>=29.4912MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
921.6K	65534	65532	0.0%	65532	65528	0.0%

表 18.33. 定时器2产生的常用波特率@ F<sub>sysclk</sub>=44.2368MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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%
230.4K	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	0.0%	65534	65534	0.0%
2.7648M	--	--	--	65535	65535	0.0%

表 18.34. 定时器2产生的高波特率@ F<sub>sysclk</sub>=44.2368MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
2.7648M	65535	65534	0.0%	65534	65532	0.0%
5.5296M	--	65535	0.0%	65535	65534	0.0%
11.0592M	--	--	--	--	65535	0.0%

表 18.35. 定时器2产生的常用波特率@ F<sub>sysclk</sub>=32MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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	--	--	--	65519	65519	2.12%

表 18.36. 定时器2产生的高波特率@ F<sub>sysclk</sub>=32MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
115200	65519	65502	2.12%	65501	65467	0.64%
230.4K	--	--	--	--	65501	-0.79%

表 18.37. 定时器2产生的常用波特率@ F<sub>sysclk</sub>=48.0MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=0			T2X12=1 & SMOD2=0		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
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%
230.4K	--	--	--	65523	65523	0.16%

表 18.38. 定时器2产生的高波特率@ F<sub>sysclk</sub>=48.0MHz

波特率	[RCAP2H, RCAP2L],重载值					
	T2X12=0 & SMOD2=1			T2X12=1 & SMOD2=1		
	SMOD1=0	SMOD1=1	误差	SMOD1=0	SMOD1=1	误差
230.4K	65523	65510	0.16%	65510	65484	0.16%
460.8K	--	65522	0.16%	65523	65510	0.16%
691.2K	--	--	--	--	--	--
921.6K	--	--	--	--	65523	0.16%

### 18.8.4.3. 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

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD2} \times (\text{SMOD1}+1)} \times F_{\text{SYSCLK}}}{16 \times 12 \times (256 - \text{RCAP2L})} \quad ; \text{T2X12=0} \\ ; \text{T2 CLK Source} = \text{SYSCLK}/12$$

$$\text{or} = \frac{2^{\text{SMOD2} \times (\text{SMOD1}+1)} \times F_{\text{SYSCLK}}}{16 \times 1 \times (256 - \text{RCAP2L})} \quad ; \text{T2X12=1} \\ ; \text{T2 CLK Source} = \text{SYSCLK}$$

注:

表 18.39 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 18.39. SMOD2 在分割定时器 2 模式 1&amp;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	± 3%
1	1	双倍波特率 X2	增强功能	± 2%

注: 当定时器2在双倍波特率X2模式下(SMOD1=1 & SMOD2=1), RPAC2L不能等于254&255.

### 18.8.4.4. 使用串口 0 波特率定时器作为波特率发生器(S0BRG)

MGEQ1C064 的 S0 中嵌入了一个专用的波特率发生器(S0BRG)，其详细功能在“18.11.2 独立波特率发生器 S0BRG 用于 S0”一节中进行了描述。当使用 S0BRG 作为 S0 的波特率发生器时，波特率计算公式如下:

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{(\text{SMOD2})}}{32} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{S0BRT})} \quad ; \text{S0TX12=0, SMOD1=0}$$

$$\text{or} = \frac{2^{(\text{SMOD2})}}{32} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{S0BRT})} \quad ; \text{S0TX12=1, SMOD1=0}$$

注:

表 18.40 在S0BRG波特率发生器中使用SMOD2和SMOD1因子定义波特率设置。

表 18.40. SMOD2 在 S0BRG 模式 1&amp;3 下的应用条件

SMOD2	SMOD1	波特率	备注	推荐最大接收误差(%)
0	0	默认波特率	标准功能	± 3%
1	0	双倍波特率	增强功能	± 3%

#### 18.8.4.5. 使用串口 1(S1)波特率定时器作为波特率发生器(S1BRG)

MGEQ1C064第二串口UART (S1)有一个独立的波特率发生器。串口0可以置位URTS (S0CFG.7)来选择S1BRT作为模式1和3的定时器源。详细的描述见章节“19.7 S1 波特率发生器用于S0”。

### 18.9. 串口 0 模式 4 (SPI 主机)

MGEQ1C064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 18.1. 串口0模式选择”

URM0X3 也可控制SPI 的传输速度。如果URM0X3 = 0，则SPI的时钟频率是SYSCLK/12。如果URM0X3 = 1，则SPI的时钟频率是SYSCLK/4

MGEQ1C064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 18-11所示。他支持多从机通讯架构见图 18-12。

图 18-11. 串口 0 模式 4，单主机和单从机架构(n = 0)

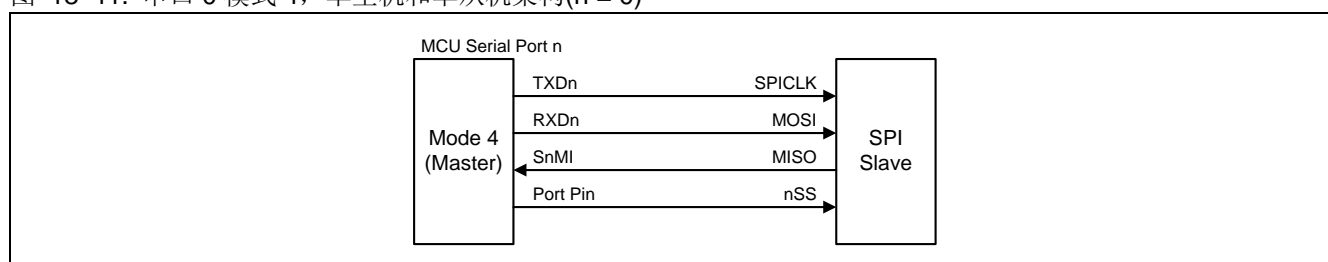
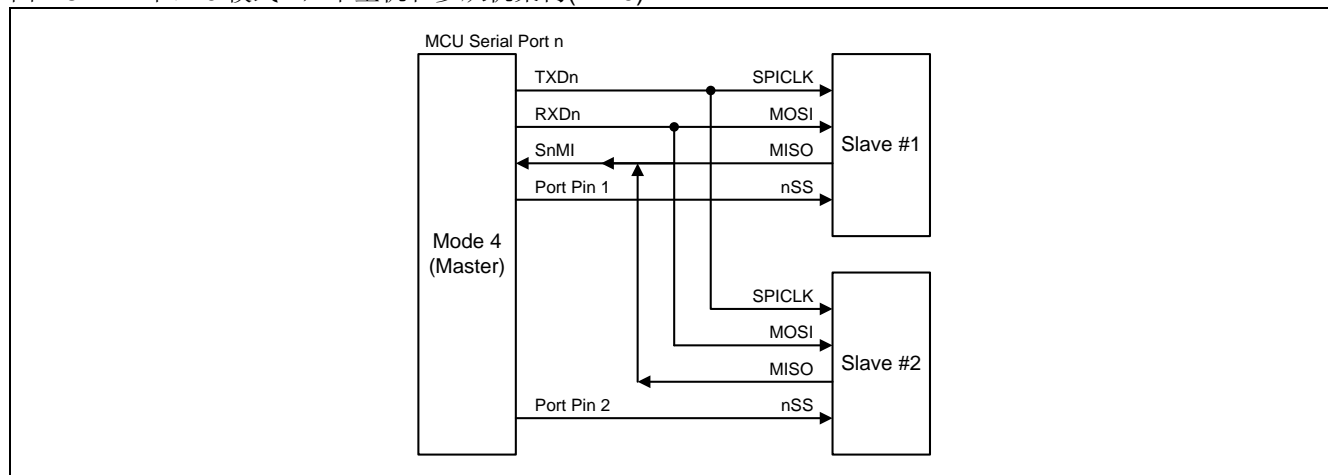


图 18-12. 串口 0 模式 4，单主机和多从机架构(n = 0)



SPI主机能满足笙泉MA82/84系列MCU(由CPOL、CPHA和DORD选择)的全功能SPI模块的传输。在CPOL和CPHA条件下，**MGEQ1C064**很容易初始化SPI的时钟极性去适合他们使用。串行口模式4的4个SPI工作模式如表 18.41所示。

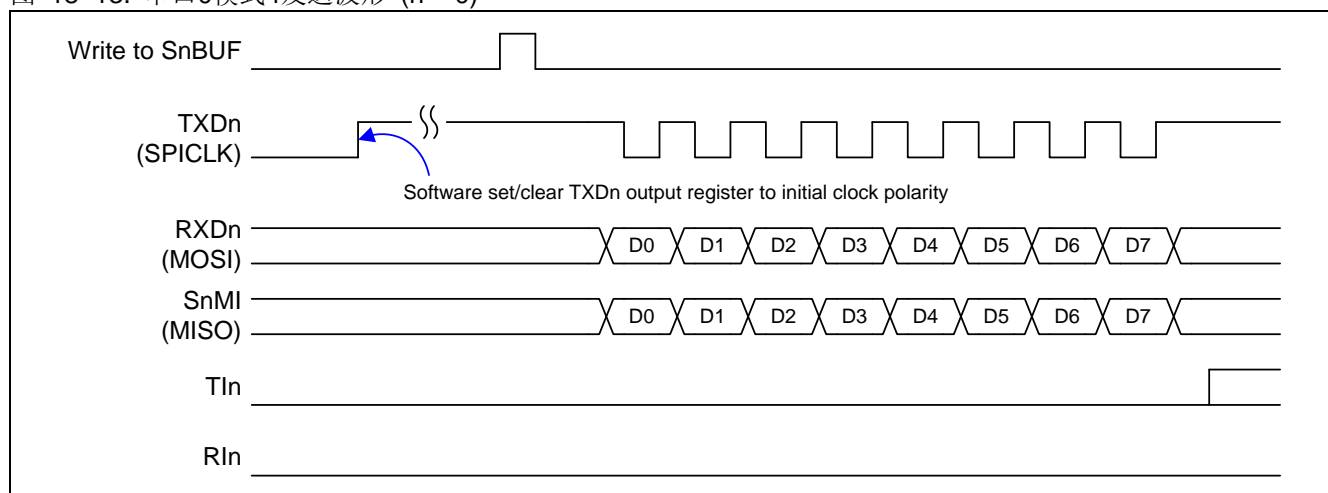
表 18.41.串口模式 4 SPI 模式映射配置

SPI 模式	CPOL	CPHA	TXD0的配置
0	0	0	清除TXD0输出寄存器 为“0”
1	0	1	清除TXD0输出寄存器 为“0”
2	1	0	设置TXD0输出寄存器 为“1”
3	1	1	设置TXD0输出寄存器 为“1”

SPI系列传输的位序控制(DORD)，**MGEQ1C064**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)。

由任何指令把S0BUF作为目的寄存器的使用初始化发送。“写到S0BUF”信号触发UART引擎开始发送。S0BUF 的数据作为MOSI串行数据被移位到RXD0引脚。SPI移位时钟在TXD0引脚上作为SPICLK输出。8个移位时钟上升沿之后，硬件生效TIO表示发送结束。同时S0MI引脚也被采样且移位到移位寄存器。然后，“读取S0BUF”会获得SPI的移入数据。模式0的发送波形如图 18-13所示。模式4中RI0不会生效。

图 18-13. 串口0模式4发送波形 (n = 0)



## 18.10. 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	TIO	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	<b>SPI 主机</b>	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。

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; 在模式0, SM20可以为0。

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。

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: RI0, 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。

$$\begin{array}{r}
 \text{SADDR} = 1100\ 0000 \\
 \text{SADEN} = 1111\ 1101 \\
 \hline
 \text{Given} = 1100\ 00x0 \longrightarrow \text{给定的从机地址将被核对, 除了位 1 被视为“忽略”}
 \end{array}$$

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“忽略”。在系统复位后，SADDR和SADEN都被初始化为0。这样生成了所有的“忽略”要求地址和所有的“忽略”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	URM0X3	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“18.8.4 模式1 &amp; 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: URM0X3**, 串口0模式0和模式4波特率选择。

串口1模式0和模式4波特率选择:

- 0: 清零选择SYSCLK/12作串口0模式0和模式4波特率。  
1: 置位选择SYSCLK/4作串口0模式0和模式4波特率。

串口0模式2:

- 0: 清零选择SYSCLK/32或/64作串口0波特率。  
1: 置位选择SYSCLK/96或/192作串口0波特率。

**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。

如果串口0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”。

如果串口0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TIO。

- 0: 保留TIO 作为一个串口0中断源。  
1: 阻止TIO 作为一个串口0中断源。

**Bit 1: UTIE**, 在系统标志中断里使能S0 TIO

- 0: 禁止在系统标志中断里中断向量共享给TIO。  
1: 设置TIO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	0	C0PLK	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	W	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 at AUXR10.3)

S0PS1~0	RXD0	TXD0
00	P3.0	P3.1
01	P4.4	P4.5
10	P3.1	P3.0
11	P7.0	P7.1

**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 &amp; S3MI 端口引脚选择.

SnMIPS	S0MI	S1MI	S2MI	S3MI
0	P6.2	P6.3	P7.0	P7.1
1	P5.7	P6.7	P6.6	P6.5

Bit 0: S0COPS, S0BRG 时钟输出 (S0CKO) 端口引脚选择.

S0COPS	S0CKO
0	P4.7
1	P4.4

**18.11. 串口 0 增强功能**

如果SMOD3 (S0CFG.0)置位, 特殊功能寄存器地址0xB9被当作S0CR1访问。S0CR1控制串口0的增强型功能包括:

- 使能S0嵌入的波特率发生器, S0BRG。
- 使能S0TX或RX由S0BRG选择波特率时间基准。
- 使能S0BRG成为一个通用定时器。
- 使能S0进入LIN总线模式。

**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操作。

1: 设置使能S0BRG操作。

Bit 6: S0TX12, S0BRG时钟源选择。

0: 清零选择SYSCLK/12作为S0BRG时钟源。

1: 设置选择SYSCLK作为S0BRG时钟源。

Bit 5: S0TCK, S0控制位选择S0BRG溢出作为UART0发送时钟。

0: 促使定时器1或定时器2溢出用作发送时钟。

1: 促使S0用S0BRG溢出作发送时钟且操作模式控制。

Bit 4: S0RCK, S0控制位选择S0BRG溢出作为UART0接收时钟。

0: 促使定时器1或定时器2溢出用作接收时钟。

1: 促使S0用S0BRG溢出作接收时钟且操作模式控制。

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, 自动重发使能。

0: 禁止自动重发。

1: 使能自动重发。

Bit 1~0: 保留位。软件在写入S0CR1时, 必须向这些位写入"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。

**S0BRC: 串口0 波特率计数器寄存器**

SFR 页 = 仅 0 页

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。

**18.11.1. S0 波特率发生器(S0BRG)**

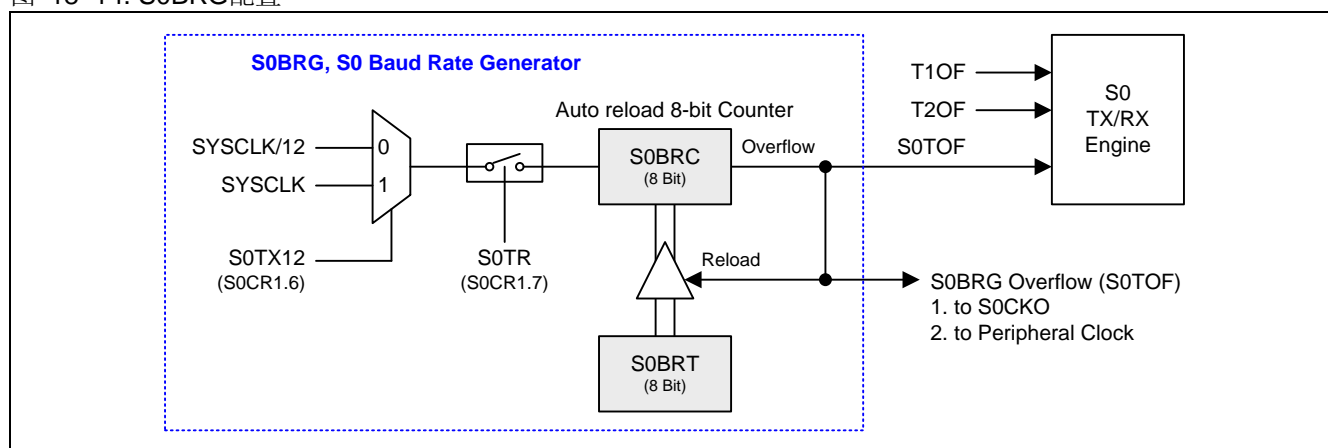
MGEQ1C064有一个嵌入的波特率发生器来产生串口0操作的时钟。它由一个8位的向上计数器(S0BRC)和一个8位的重载寄存器(S0BRT)构成。S0BRC溢出(S0TOF)是UART0串行引擎在所有操作模式下的时间基准且触发S0BRT值重载到S0BRC保证连续计数。

如果S0TR= 0, 软件写S0BRT数据将同时修改S0BRC。在S0TR使能开启S1BRC计数之后, 当S1BRT写入时不影响S1BRC。修改S1BRC不会影响S1BRT的值。

波特率发生器也提供时钟输出的时间基准, S0CKO, 为S0BRC溢出率的二分频(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波特率发生器结构如图 18-14所示。

图 18-14. S0BRG配置



### 18.11.2. 独立波特率发生器 S0BRG 用于 S0

为了给S0更大的灵活性，可以选择波特率发生器S0BRG作为S0波特率源。  
串口0波特率选择配置请参考“图 18-10. S0的波特率源”

### 18.11.3. S0 LIN 总线寄存器

#### S0CFG1: 串口0配置寄存器1

SFR 页 = 仅 0 页

SFR 地址 = 0x9D

复位值 = 0000-00xx

7	6	5	4	3	2	1	0
SBF0	TXER0	S0SB16	ATBR0	TXRX0	SYNC0	--	--
R/W	R/W	R/W	R/W	R/W	R/W	W	W

Bit 7: SBF0, S0同步中止标志。

0: 必须软件清零。

1: LIN总线侦测到中止事件的结束硬件置位。在主机模式，与TI0标志相连置位。在从机模式，与RI0标志相连置位。

Bit 6: TXER0, S0的LIN发送错误。

0: 必须软件清零。

1: 在TX模式，LIN总线侦测到发送错误硬件置位。

Bit 5: S0SB16, S0同步中止16位使能。

0: 在主机模式选择13位同步中止发送。

1: 在主机模式选择16位同步中止发送。

Bit 4: ATBR0, S0自动波特率。

0: 在同步域结束硬件自动清零。

1: 在同步域之前，软件置位在从机RX模式执行LIN总线同步域自动波特率调整。

Bit 3: TXRX0, S0 LIN总线的TX/RX选择。

0: 选择LIN总线接口引擎为RX功能。

1: 选择LIN总线接口引擎为TX功能。

Bit 2: SYNC, S0同步中止控制位。

0: 当主机模式同步中止发送或从机模式接收到自动清零。

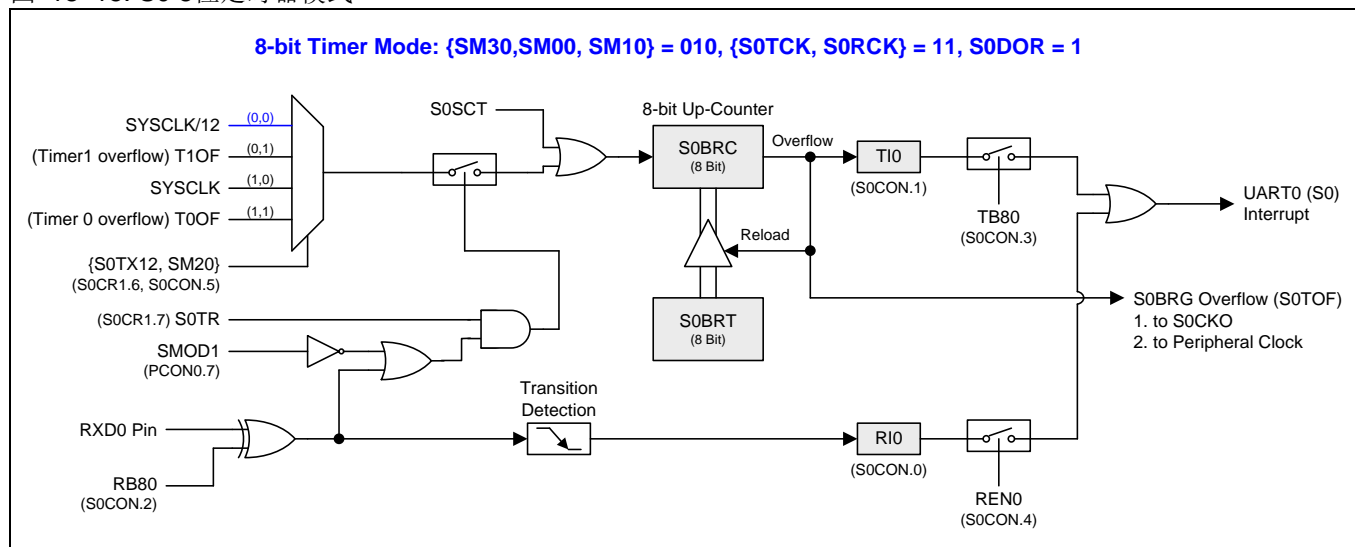
1: 软件置位。如果主机模式置位，下一个写S0BUF将发送一个同步中止到LIN总线。如果从机模式置位，LIN接口引擎将等待接收一个同步中止。

Bit 1~0: 保留位。软件在写入 S0CFG1 时，必须向这些位写入” 0”。

### 18.11.4. S0 当做 16 位定时器模式

S0的8位定时器模式如图 18-15所示

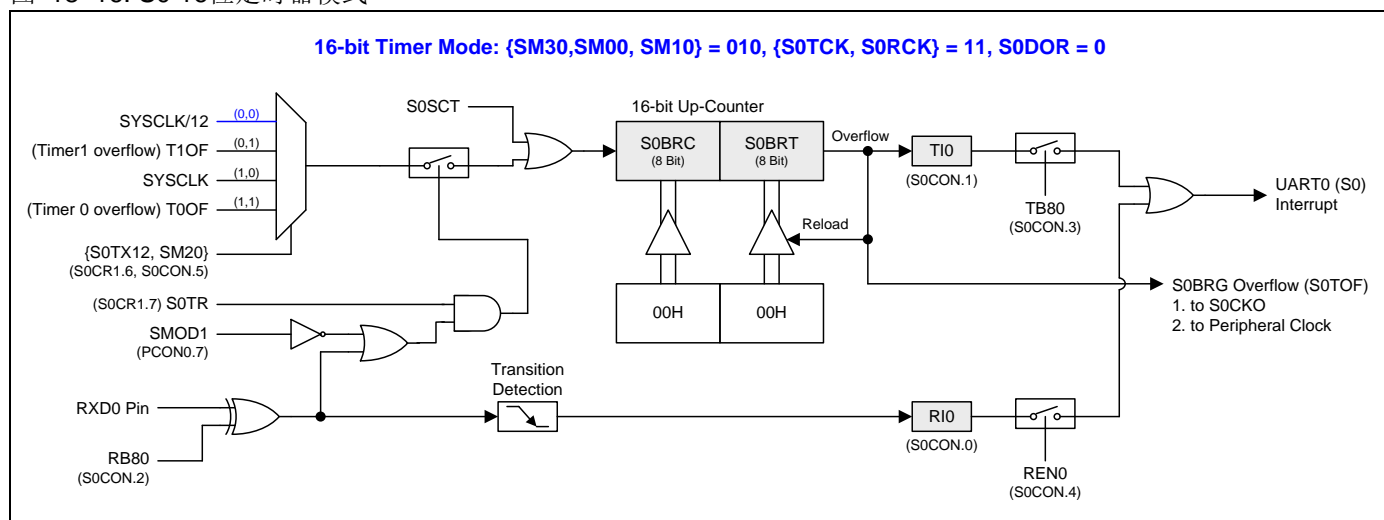
图 18-15. S0 8位定时器模式



### 18.11.5. S0 当做 16 位定时器模式

S0的16位定时器模式如图 18-16所示

图 18-16. S0 16位定时器模式



### 18.11.6. S0BRG 可编程时钟输出

S0BRG有一个时钟输出模式如图 18-17和图 18-18所示

图 18-17. S0BRG时钟输出(S0BRG为8位定时器模式)

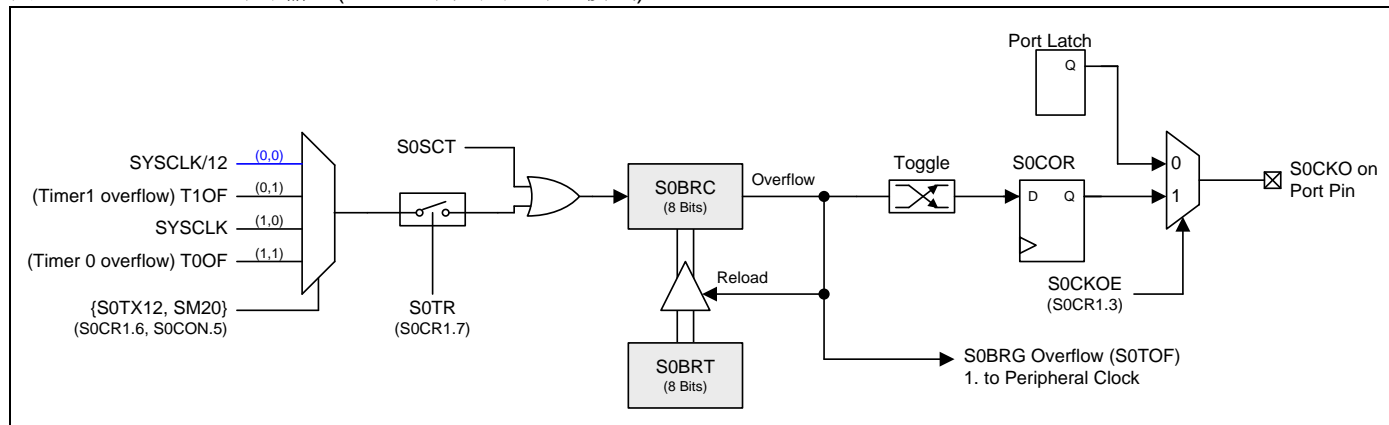
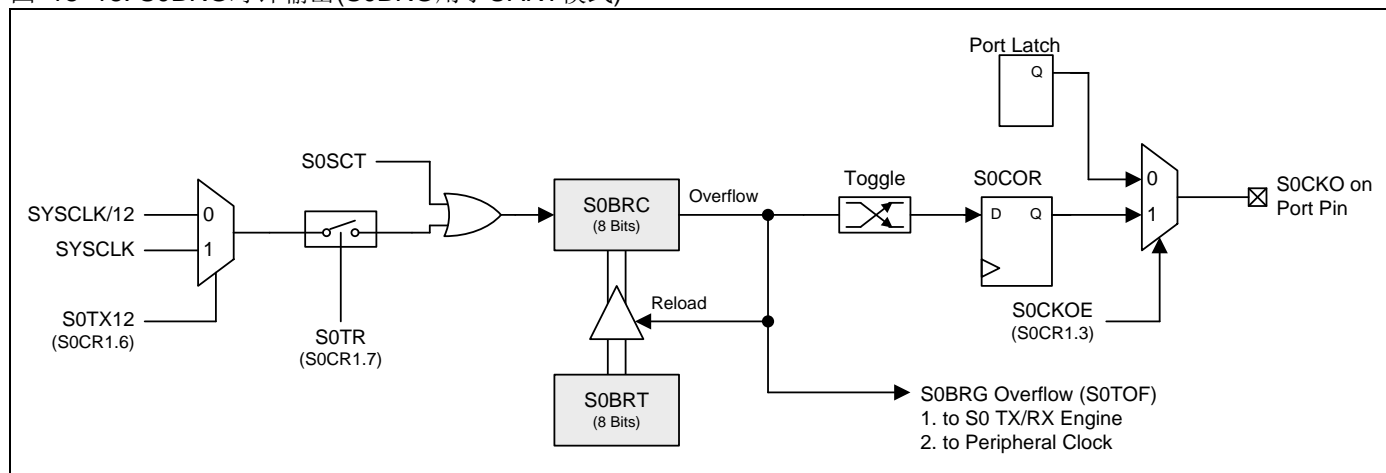


图 18-18. S0BRG时钟输出(S0BRG用于UART模式)



#### 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. 串口 1 (UART1)

MGEQ1C064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S1CKO切换时钟源和外设时钟。
- (5) S1 + S1BRG 在端口改变侦测下可以被配置成一个8位自动重载定时器。
- (6) 模式0和模式4，UART1的S1TX12与UART0的URM0X3一样的功能。

MGEQ1C064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。

### 19.1. 串口 1 波特率发生器 (S1BRG)

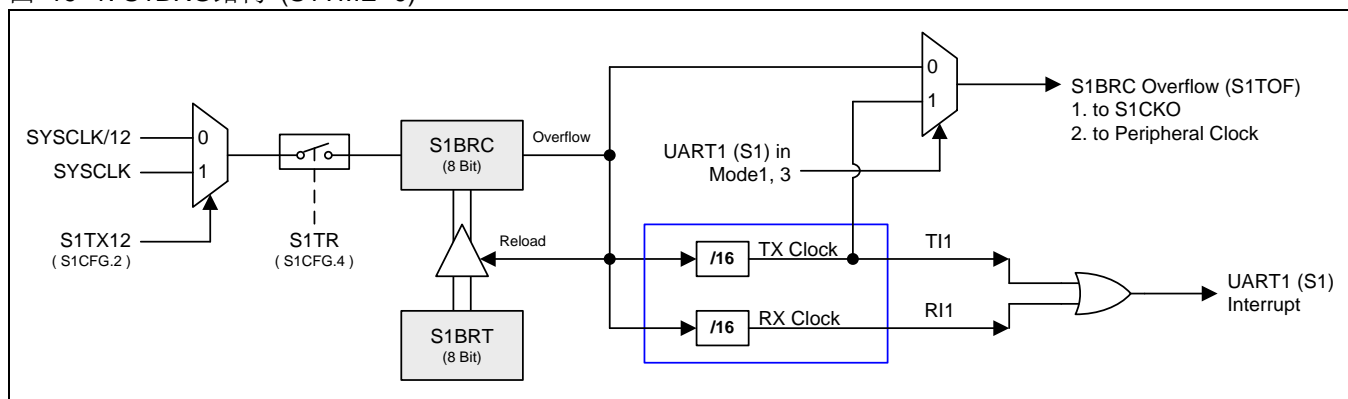
MGEQ1C064在模式1和模式3有一个嵌入式波特率发生器给串口1提供UART时钟。它由一个8位的向上计数器(S1BRC)和一个自动装载寄存器(S1BRT)构成。S1BRC的溢出(S1TOF)是UART1在模式1和模式3串行引擎的时间基准并且触发S1BRT的值重载到S1BRC继续计数。

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溢出率的二分频(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 19-1

图 19-1. S1BRG结构 (S1TME=0)



## 19.2. 串口 1 波特率设定

### 19.2.1. 模式 0 波特率

S1模式0波特率公式: (n=1)

$$\text{Sn Mode 0 Baud Rate} = \frac{F_{\text{SYSCLK}}}{12} \quad ; \text{SnTX12}=0$$

$$\text{or} = \frac{F_{\text{SYSCLK}}}{4} \quad ; \text{SnTX12}=1$$

### 19.2.2. 模式 2 波特率

S1模式2波特率公式: (n=1)

$$\text{Sn Mode 2 Baud Rate} = \frac{2^{\text{SnMOD1}}}{64} \times F_{\text{SYSCLK}} \quad ; \text{SnM0X3}=0$$

$$\text{or} = \frac{2^{\text{SnMOD1}}}{192} \times F_{\text{SYSCLK}} \quad ; \text{SnM0X3}=1$$

表 19.1. S1模式2波特率@ F<sub>SYSCLK</sub>=11.0592MHz

波特率	S1M0X3	S1MOD1	误差
172800	0	0	0.0%
345600	0	1	0.0%
57600	1	0	0.0%
115200	1	1	0.0%

表 19.2. S1模式2波特率@ F<sub>SYSCLK</sub>=12.00MHz

波特率	S1M0X3	S1MOD1	误差
187500	0	0	0.0%
375000	0	1	0.0%
62500	1	0	0.0%
125000	1	1	0.0%

## 19.2.3. 模式 1 &amp; 3 波特率

S1模式1, 3波特率公式: (n=1)

$$\text{Sn Mode 1, 3 Baud Rate} = \frac{2^{\text{SnMOD1}}}{32} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{SnBRT})} ; \text{SnTX12=0}$$

$$\text{or} = \frac{2^{\text{SnMOD1}}}{32} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{SnBRT})} ; \text{SnTX12=1}$$

表 19.3 ~ 表 19.6列出S1BRG(串行端口1波特率产生器)各种通用的波特率和怎样获得

表 19.3. S1BRG产生的常用波特率@ F<sub>SYSCLK</sub>=11.0592MHz

波特率	S1BRT, S1BRG重载值					
	S1TX12=0			S1TX12=1		
	S1MOD1=0	S1MOD1=1	误差	S1MOD1=0	S1MOD1=1	误差
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 19.4. S1BRG产生的常用波特率@ F<sub>SYSCLK</sub>=22.1184MHz

波特率	S1BRT, S1BRG重载值					
	S1TX12=0			S1TX12=1		
	S1MOD1=0	S1MOD1=1	误差	S1MOD1=0	S1MOD1=1	误差
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%
460800	--	--	--	--	253	0.0%

表 19.5. S1BRG产生的常用波特率@ F<sub>SYSClk</sub>=12.0MHz

波特率	S1BRT, S1BRG重载值					
	S1TX12=0			S1TX12=1		
	S1MOD=0	S1MOD=1	误差	S1MOD=0	S1MOD=1	误差
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	--	--	--	--	--	--

表 19.6. S1BRG产生的常用波特率@ F<sub>SYSClk</sub>=24.0MHz

波特率	S1BRT, S1BRG重载值					
	S1TX12=0			S1TX12=1		
	S1MOD=0	S1MOD=1	误差	S1MOD=0	S1MOD=1	误差
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 19.7. S1BRG产生的常用波特率@ F<sub>SYSClk</sub>=29.4912MHz

波特率	S1BRT, S1BRG重载值					
	S1TX12=0			S1TX12=1		
	S1MOD1=0	S1MOD1=1	误差	S1MOD1=0	S1MOD1=1	误差
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%
230.4K	--	--	--	252	248	0.0%
460.8K	--	--	--	254	252	0.0%
921.6K	--	--	--	255	254	0.0%
1.8432M	--	--	--	--	255	0.0%

表 19.8. S1BRG产生的常用波特率@  $F_{\text{SYSCLK}}=44.2368\text{MHz}$ 

波特率	S1BRT, S1BRG重载值					
	S1TX12=0			S1TX12=1		
	S1MOD1=0	S1MOD1=1	误差	S1MOD1=0	S1MOD1=1	误差
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%
230.4K	--	255	0.0%	250	244	0.0%
460.8K	--	--	--	253	250	0.0%
921.6K	--	--	--	--	253	0.0%
1.8432M	--	--	--	--	--	--
2.7648M	--	--	--	--	255	0.0%

表 19.9. S1BRG产生的常用波特率@  $F_{\text{SYSCLK}}=32.0\text{MHz}$ 

波特率	S1BRT, S1BRG重载值					
	S1TX12=0			S1TX12=1		
	S1MOD1=0	S1MOD1=1	误差	S1MOD1=0	S1MOD1=1	误差
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 19.10. S1BRG产生的常用波特率@ F<sub>sysclk</sub>=48.0MHz

波特率	S1BRT, S1BRG重载值					
	S1TX12=0			S1TX12=1		
	S1MOD1=0	S1MOD1=1	误差	S1MOD1=0	S1MOD1=1	误差
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%
230.4K	--	--	--	--	243	0.16%

### 19.3. 串口 1 模式 4 (SPI 主机)

MGEQ1C064的串口1嵌入了一个额外的模式4来支持SPI主机引擎。模式4由SM31、SM01和SM11选择。MGEQ1C064的串口1模式定义如表 19.11所示。

表 19.11. 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	SYSCCLK/12 或SYSCCLK/4
0	0	1	1	8-位UART	可变
0	1	0	2	9-位UART	SYSCCLK/64, /32或/192, /96
0	1	1	3	9-位UART	可变
1	0	0	4	SPI主机	SYSCCLK/12 或SYSCCLK/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

S1M0X3也控制SPI 的传输速度。如果S1M0X3= 1, 则SPI的时钟频率是SYSCCLK/4。否则, SPI的时钟频率是SYSCCLK/12。

MGEQ1C064的SPI主机使用TXD1作为SPICLK, RXD1作为 MOSI, 以及S1MI作为MISO。而nSS由MCU软件选择在其它端口引脚。SPI 连接如图 19-2所示。他支持多从机通讯架构见图 19-3。

图 19-2. 串口模式4, 单主机和单从机架构(n = 1)

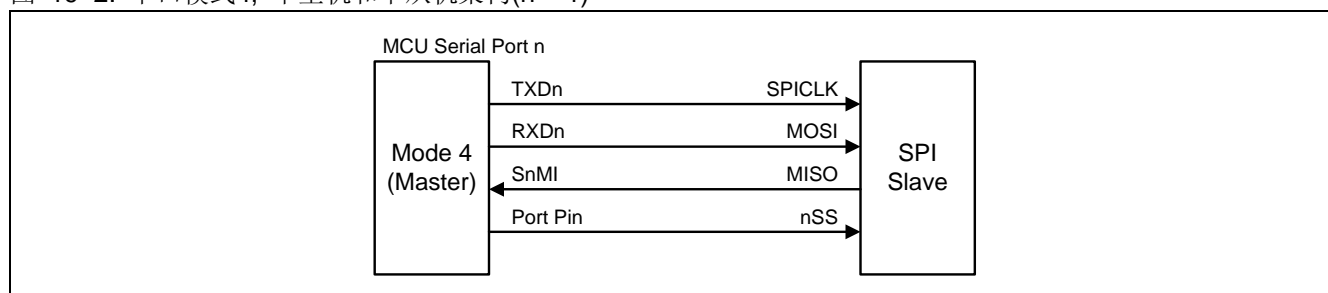
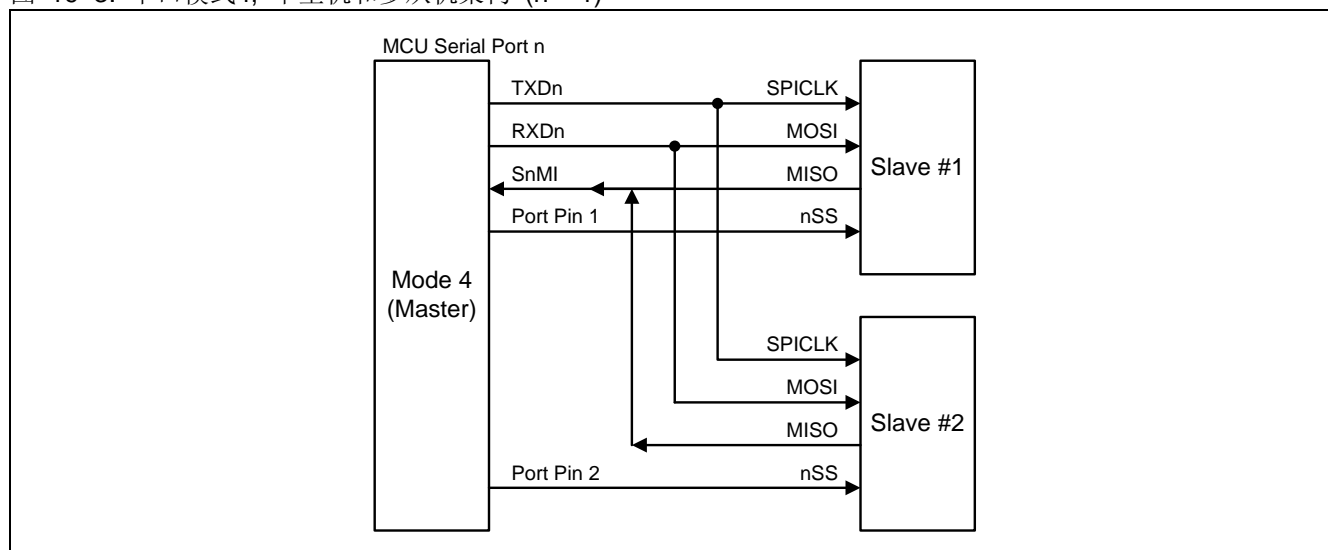


图 19-3. 串口模式4, 单主机和多从机架构 (n = 1)



SPI主机能满足笙泉MA82/84系列MCU(由CPOL、CPHA和DORD选择)的全功能SPI模块的传输。在CPOL和CPHA条件下，**MGEQ1C064**很容易初始化SPI的时钟极性去适合他们使用。串行口模式4的4个SPI工作模式如表 19.12所示。

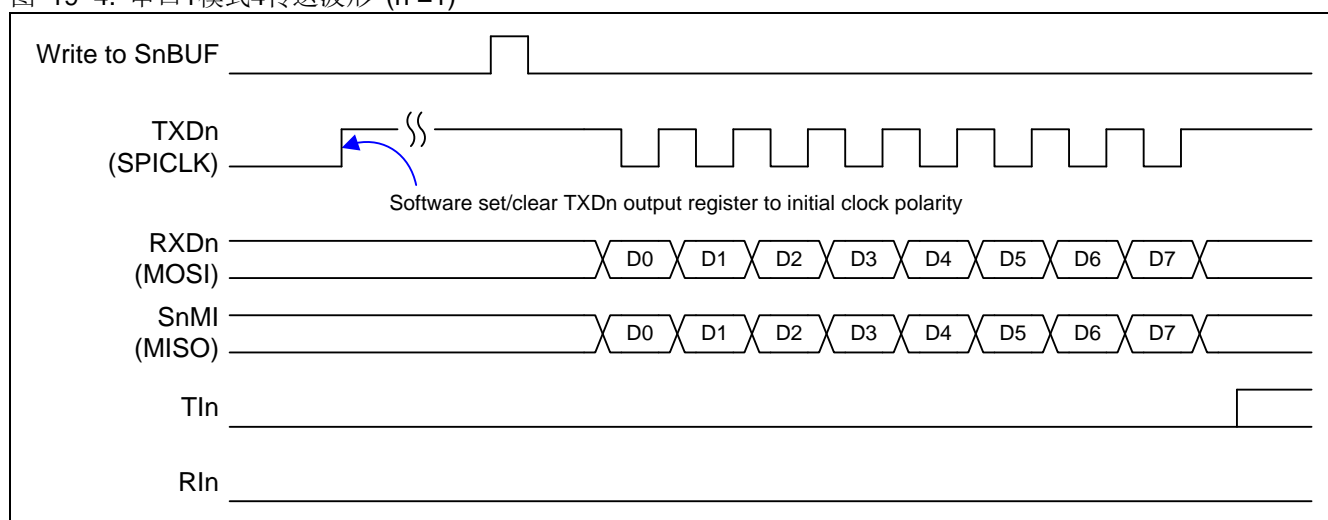
表 19.12.串口1模式4的SPI模式配置

SPI Mode	CPOL	CPHA	TXD1的配置
0	0	0	清除TXD1输出寄存器为“0”
1	0	1	清除TXD1输出寄存器为“0”
2	1	0	设置TXD1输出寄存器为“1”
3	1	1	设置TXD1输出寄存器为“1”

SPI系列传输的位序控制(DORD)，**MGEQ1C064**提供了位序控制(S1DOR)。S0DOR的默认值是“1”且位序控制为低位在先(LSB)。

通过指令使用S1BUF作为目标寄存器初始化传送。“写数据到S1BUF”触发UART引擎开始传送。S1BUF的数据被移位到作为MOSI 串口数据的RXD1引脚。SPI 移位时钟由作为SPICLK 输出的TXD1引脚输出。在8个移位时钟的上升沿之后，T11被硬件声明传送结束。同时S0MI 引脚的内容被采样并且移位到移位寄存器。然后“读取S1BUF”能获取SPI 的移入数据。模式4传送波形如图 19-4所示。RI1在模式4不被声明。

图 19-4. 串口1模式4传送波形 (n=1)



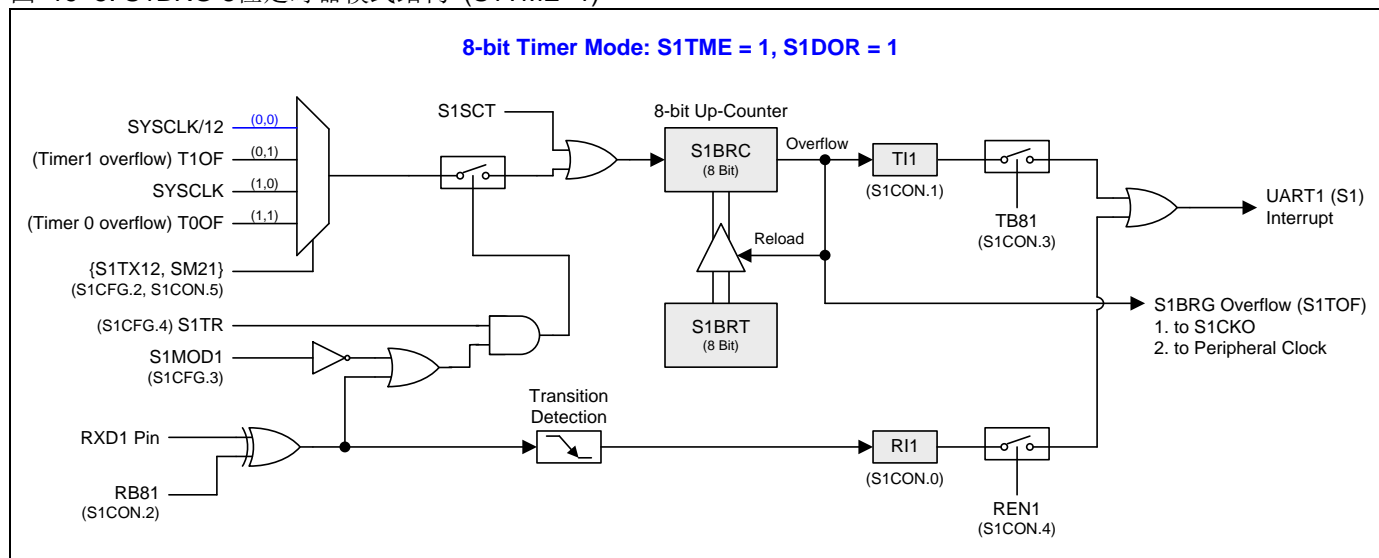
## 19.4. S1BRG 8 位定时器模式

如果UART1不用或软件搁置，MGEQ1C064设定S1TME=1串行口1波特率发生器(S1BRG)提供纯定时器操作模式。此定时器操作在一个8位自动重载定时器并且提供溢出标志TI1(S1CON.1)置位。RI1(S1CON.0)服务于RXD1端口引脚的端口变化检测器。在此模式下TI1和RI1保持UART1中断源的中断能力并且有一个独立中断使能控制(TB81和REN1)。RB81选择RXD1端口输入的RI1检测电平。如果RB81=0，RI1由REN1=1并且检测到RXD1引脚下降沿置位；否则RI1将检测RXD1端口引脚的上升沿。在MCU掉电模式，RI1被强制成电平触发操作并且如果UART1中断使能可以唤醒CPU。

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 19-5所示。

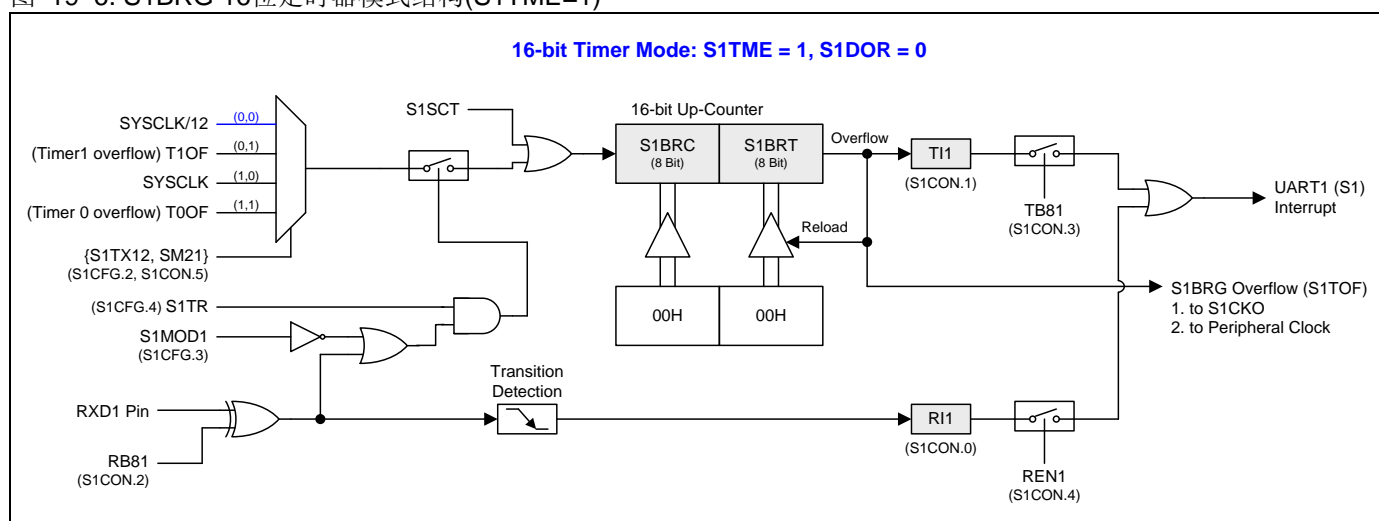
图 19-5. S1BRG 8位定时器模式结构 (S1TME=1)



## 19.5. S1BRG 16 位定时器模式

S1BRG 16位定时器模式结构如图 19-6所示。

图 19-6. S1BRG 16位定时器模式结构(S1TME=1)

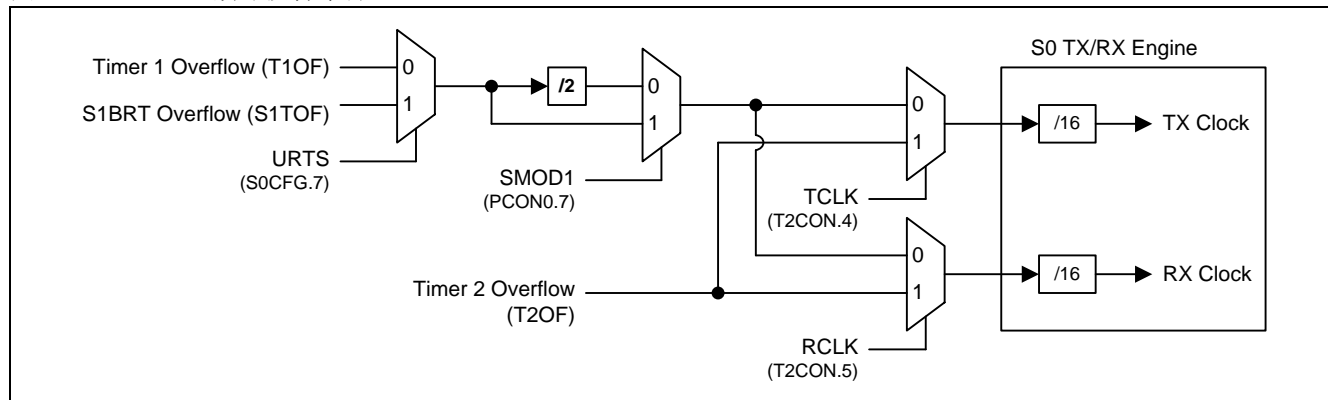




## 19.7. S1 波特率发生器用于 S0

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 19-9. 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：

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}} \times 2^{(\text{SMOD2} \times 2)}}{32} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{S1BRT})} ; \text{S1TX12}=0$$

$$\text{or} = \frac{2^{\text{SMOD1}} \times 2^{(\text{SMOD2} \times 2)}}{32} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{S1BRT})} ; \text{S1TX12}=1$$

表 19.13. S1BRG产生的常用波特率for S0 @ F<sub>sysclk</sub>=11.0592MHz

波特率	S1BRT, S1BRG 重载值					
	S1TX12=0 & SMOD2=0			S1TX12=1 & SMOD2=0		
	SMOD1=0 (PCON0.7)	SMOD1=1 (PCON0.7)	误差	SMOD1=0 (PCON0.7)	SMOD1=1 (PCON0.7)	误差
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%

## 19.8. 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 and 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, /32或/192, /96
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	保留	保留
1	1	1	7	保留	保留

Bit 5: 串口1模式位2。

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地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地地址或者是一个广播地址; 在模式0, SM21可以为0。在 S1BRG 纯定时器模式下, 设置 SM21 选择定时器 1 溢出作为时钟源来级联定时器 1 和 S1BRG 以扩大定时器计数器位。

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。

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位个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 波特率计数器寄存器**

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	S1M0X3	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	SYSCCLK/12或SYSCCLK/4
0	0	1	1	8-位UART	可变
0	1	0	2	9-位UART	SYSCCLK/64, /32或/192,/96
0	1	1	3	9-位UART	可变
1	0	0	4	SPI 主机	SYSCCLK/12或 SYSCCLK/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 6: S1M0X3, 串口1模式0、模式2和模式4波特率选择。

串口1模式0和模式4:

0: 清零选择SYSCCLK/12作串口1模式0和模式4波特率。

1: 置位选择SYSCCLK/4作串口1模式0和模式4波特率。

串口1模式2:

0: 清零选择SYSCCLK/32或/64作串口1波特率。

1: 置位选择SYSCCLK/96或/192作串口1波特率。

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16位重载定时器/计数器模式。

1: 设置S1BRG为8位定时器/计数器模式。

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。

Bit 2: S1TX12, UART1波特率发生器时钟源选择。

0: 清零选择SYSCCLK/12作S1BRG的时钟源。

1: 置位选择SYSCLK作S1BRG的时钟源。

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	COFDC1	COFDC0	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	P7.1	P7.0
1 0	P2.0	P2.1
1 1	P3.4	P3.5

#### 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	S2MI	S3MI
0	P6.2	P6.3	P7.0	P7.1
1	P5.7	P6.7	P6.6	P6.5

#### AUXR8: 辅助寄存器 8

SFR 页 = 仅 5 页

SFR 地址 = 0xA4

复位值 = 1100-0000

7	6	5	4	3	2	1	0
POE7	POE6	COPPS2	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	W	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	P4.6
1	P4.5

## 20. 串口 2 (UART2)

**MGEQ1C064** 装备有第三UART (以后就称作UART2)，有5种运行模式，模式0~模式4，除了下面的不同点与第二个UART (UART1)一样：

- (1) S1BRG可以共享波特率给UART0。UART2没有这个功能。
- (2) 所有UART1 的SRFs位于SFR页面1和页面2，但是所有UART2的SFRs位于SFR页面 3。

UART2其它所有功能可以参考UART1功能定义。所有UARTs都可以在相同或不同的模式和通信速度下同时运行。

### 20.1. 串口 2 波特率发生器 (S2BRG)

**MGEQ1C064** 在模式1和模式3有一个嵌入式波特率发生器给串口2提供UART时钟。它由一个8位的向上计数器(S2BRC)和一个自动装载寄存器(S2BRT)构成。S2BRC的溢出(S2TOF)是UART2在模式1和模式3串行引擎的时间基准并且触发S2BRT的值重载到S2BRC继续计数。

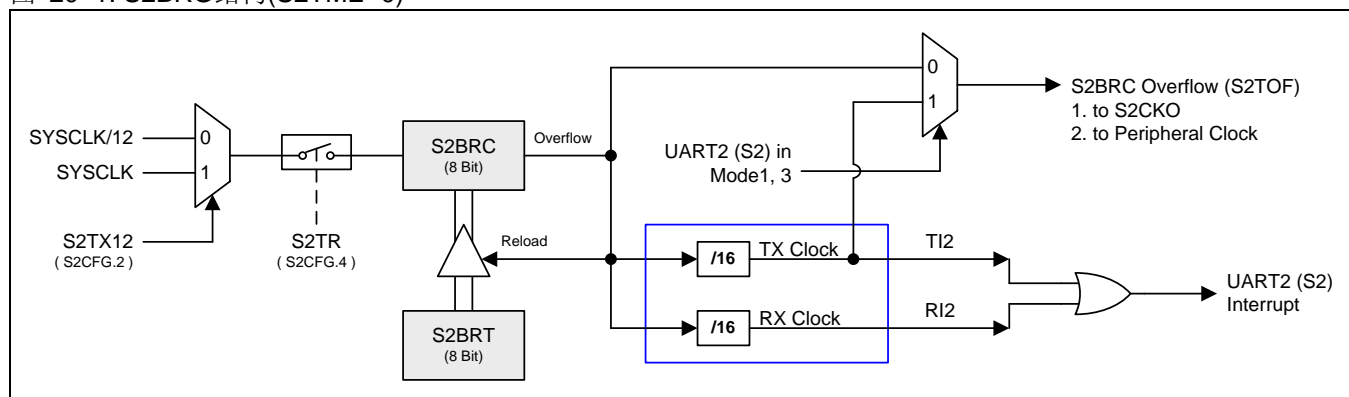
如果S2TR=0, 软件写S2BRT数据将同时修改S2BRC。在S2TR使能开启S2BRC计数之后, 当S2BRT写入时不影响S2BRC。修改S2BRC不会影响S2BRT的值。

这里有一个S2BRC溢出率的二分频(S2TOF/2)的额外时钟输出(S2CKO)。S2TOF也供给其它外设时钟输入源切换。无论S2运行还是待运行，S2BRG总是服务于这些外设的时间基准功能。

串口2波特率发生器结构见

图 20-1..

图 20-1. S2BRG结构(S2TME=0)



## 20.2. 串口 2 波特率设定

### 20.2.1. 模式 0 波特率

S2模式0波特率公式: (n=2)

$$\text{Sn Mode 0 Baud Rate} = \frac{F_{\text{SYSCLK}}}{12} \quad ; \text{SnTX12}=0$$

$$\text{or} = \frac{F_{\text{SYSCLK}}}{4} \quad ; \text{SnTX12}=1$$

### 20.2.2. 模式 2 波特率

S2模式2波特率公式: (n=2)

$$\text{Sn Mode 2 Baud Rate} = \frac{2^{\text{SnMOD1}}}{64} \times F_{\text{SYSCLK}} \quad ; \text{SnM0X3}=0$$

$$\text{or} = \frac{2^{\text{SnMOD1}}}{192} \times F_{\text{SYSCLK}} \quad ; \text{SnM0X3}=1$$

表 20.1. S2模式2波特率@ F<sub>SYSCLK</sub>=11.0592MHz

波特率	S2M0X3	S2MOD1	误差
172800	0	0	0.0%
345600	0	1	0.0%
57600	1	0	0.0%
115200	1	1	0.0%

表 20.2. S2模式2波特率@ F<sub>SYSCLK</sub>=12.00MHz

波特率	S2M0X3	S2MOD1	误差
187500	0	0	0.0%
375000	0	1	0.0%
62500	1	0	0.0%
125000	1	1	0.0%

### 20.2.3. 模式 1 & 3 波特率

S2 Mode 1, 3 波特率 equation: (n=2)

$$\text{Sn Mode 1, 3 Baud Rate} = \frac{2^{\text{SnMOD1}}}{32} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{SnBRT})} ; \text{SnTX12}=0$$

$$\text{or} = \frac{2^{\text{SnMOD1}}}{32} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{SnBRT})} ; \text{SnTX12}=1$$

### 20.3. 串口 2 模式 4(SPI 主机)

MGEQ1C064的串口2模式4(SPI主机)与串口1模式4(SPI主机)完全兼容。详细功能请参考串口1说明。

### 20.4. S2BRG 8 位定时器模式

如果UART2不用或软件搁置，MGEQ1C064设定S2TME=1串行口2波特率发生器(S2BRG)提供纯定时器操作模式。此定时器操作在一个8位自动重载定时器并且提供溢出标志TI2(S2CON.1)置位。RI2(S2CON.0)服务于RXD2端口引脚的端口变化检测器。在此模式下TI2和RI2保持UART2中断源的中断能力并且有一个独立中断使能控制(TB82和REN2)。RB82选择RXD2端口输入的RI2检测电平。如果RB82=0，RI2由REN2=1并且检测到RXD2引脚下降沿置位；否则RI2将检测RXD2端口引脚的上升沿。在MCU掉电模式，RI2被强制成电平触发操作并且如果UART2中断使能可以唤醒CPU。

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S2BRC溢出，它可以触发端口引脚输出。S2CKOE=1使能S2CKO输出在端口引脚并且屏蔽RI2中断。

S2BRG 8位定时器模式结构如图 20-2所示。

图 20-2. S2BRG 8位定时器模式结构(S2TME=1)

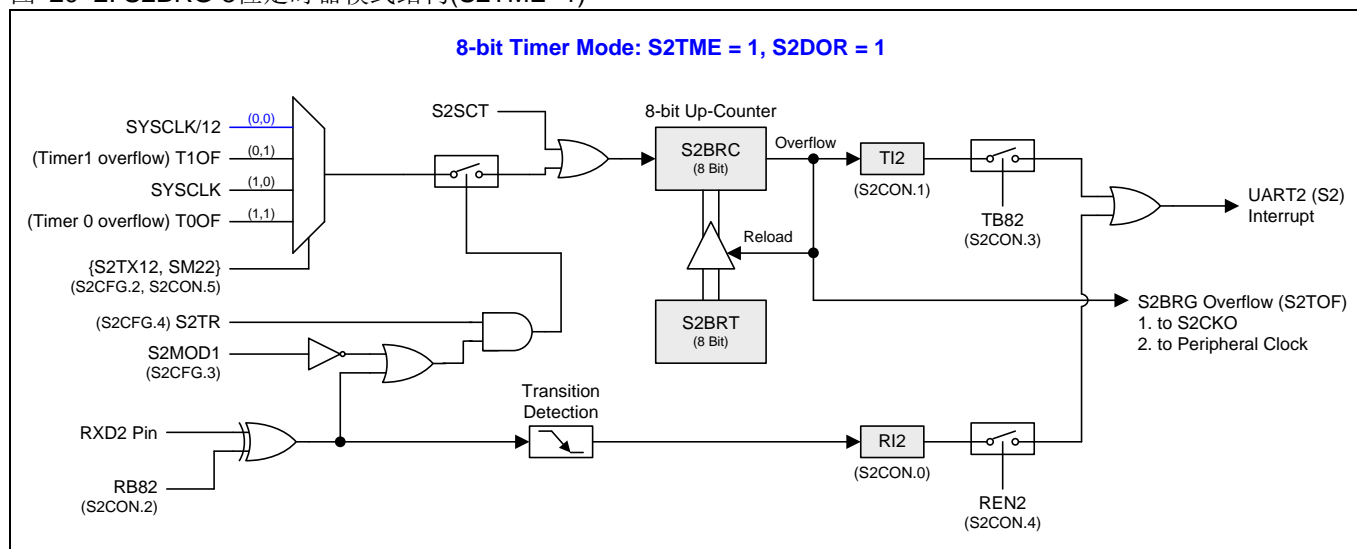
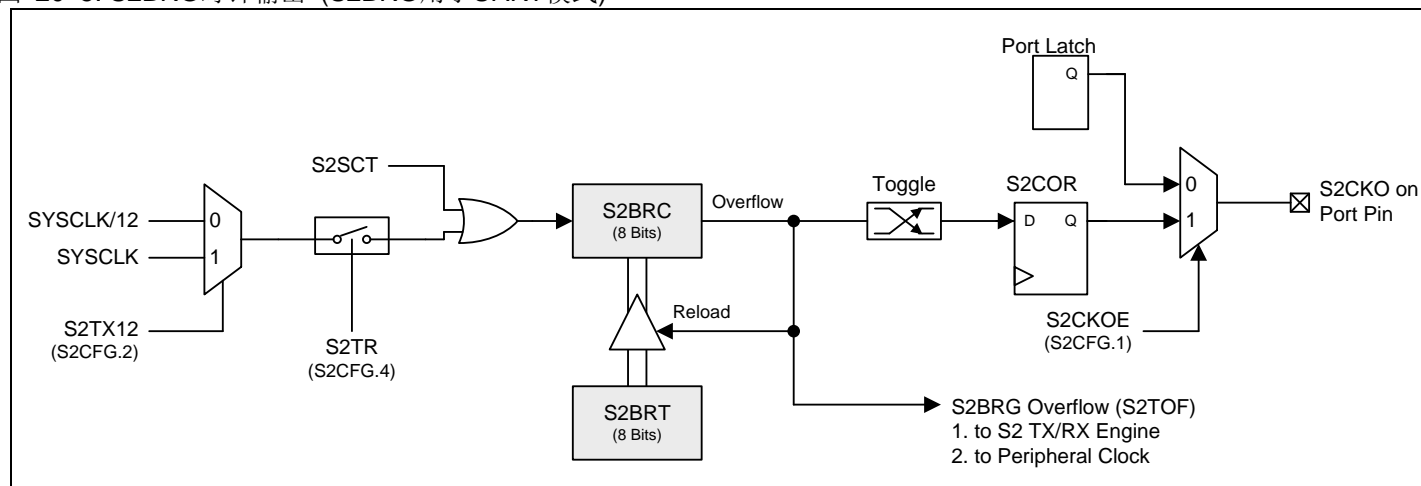




图 20-5. S2BRG时钟输出 (S2BRG用于UART模式)



### 在时钟输出模式如何编程8位S2BRG

- 设置S2CFG.S2TX12和S2CON.SM22选择S2BRG 时钟源.
- 由公式计算8位重装载值并且存入S2BRT和S2BRC寄存器
- 置位S2CFG的S2CKOE位
- 置位S2TR 去启动S2BRC定时器

## 20.7. 串口 2 寄存器

下面的特殊功能寄存器与UART2的操作有关:

### S2CON: 串口2控制寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0x98

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SM02	SM12	SM22	REN2	TB82	RB82	TI2	RI2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SM02, 串口2模式位0

Bit 6: SM12, 串口2模式位1.

SM32	SM02	SM12	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, /32或/192, /96
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	保留	保留
1	1	1	7	保留	保留

Bit 5: 串口2模式位2。

0: 禁止SM22功能。

1: 在模式2和3时使能地址自动识别, 如果SM22=1那么RI2将不能设置, 除非接收到的第9位数据(RB82)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SM22=1那么RI2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; 在模式0, SM22可以为0。

Bit 4: REN2, 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: TB82, 在模式2和3时第9位数据被传送, 需要通过软件置位或清零。

Bit 2: RB82, 在模式2和3时收到的第9位数据。在模式1, 如果SM22=0, RB82是收到数据的停止位。在模式0, RB82没有使用。

Bit 1: TI2, 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: RI2, 接收中断标志。

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(除留意SM22外), 在接收停止位的中间时刻由硬件置位。

### S2BUF: 串口2缓冲寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0x99

复位值 = XXXX-XXXX

7	6	5	4	3	2	1	0
S2BUF.7	S2BUF.6	S2BUF.5	S2BUF.4	S2BUF.3	S2BUF.2	S2BUF.1	S2BUF.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。

### S2BRT: 串口2波特率定时器重载寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0x9A

复位值 = 0000-0000

7	6	5	4	3	2	1	0
S2BRT.7	S2BRT.6	S2BRT.5	S2BRT.4	S2BRT.3	S2BRT.2	S2BRT.1	S2BRT.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。

### S2BRC: 串口2波特率计数器寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0x9B

复位值 = 0000-0000

7	6	5	4	3	2	1	0
S2BRC.7	S2BRC.6	S2BRC.5	S2BRC.4	S2BRC.3	S2BRC.2	S2BRC.1	S2BRC.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S2CFG1.S2TME = 0, 软件写数据到S2BRT同时数据存入S2BRT和S2BRC。

### S2CFG: 串口2配置寄存器

SFR 页 = 仅 3 页

SFR 地址 = 0x9C

复位值 = 0010-0000

7	6	5	4	3	2	1	0
SM32	S2MOX3	S2DOR	S2TR	S2MOD1	S2TX12	S2CKOE	S2TME
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SM32, 串口2模式控制位3.

SM32	SM02	SM12	模式	描述	波特率
0	0	0	0	移位寄存器	SYSCLK/12或SYSCL/4
0	0	1	1	8-位UART	可变
0	1	0	2	9-位UART	SYSCLK/64, /32或/192, /96
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	保留	保留
1	1	1	7	保留	保留

Bit 6: S2M0X3, 串口2模式0、模式2和模式4波特率选择。

串口2模式0和模式4:

0: 清零选择SYSCLK/12作串口2模式0和模式4波特率。

1: 置位选择SYSCLK/4作串口2模式0和模式4波特率。

串口2模式2:

0: 清零选择SYSCLK/32或/64作串口2波特率。

1: 置位选择SYSCLK/96或/192作串口2波特率。

Bit 5: S2DOR, 串口2所有操作模式的数据位序控制。

如果S2TME = 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S2DOR为“1”。

如果S2TME = 1:

0: 设置S2BRG为16位重载定时器/计数器模式。

1: 设置S2BRG为8位定时器/计数器模式。

Bit 4: S2TR, UART2波特率发生器控制位。

0: 清零关闭S2BRG。

1: 置位开启S2BRG。

Bit 3: S2MOD1, UART2双倍波特率选择使能位。

0: 禁止UART2双倍波特率功能。

1: 使能UART2双倍波特率功能。

Bit 2: S2TX12, UART2波特率发生器时钟源选择。

0: 清零选择SYSCLK/12作S2BRG的时钟源。

1: 置位选择SYSCLK作S2BRG的时钟源。

Bit 1: S2CKOE, 串口2波特率发生器时钟输出使能。

0: 禁止S2CKO 在端口引脚输出。

1: 使能S2CKO 在端口引脚输出。

Bit 0: S2TME, 串口2波特率(BRG)定时器模式使能。

0: 保持S2BRG服务串口2 (UART2)。

1: 禁止串口2功能并且S2BRG作为一个8位自动装载的定时器。这个模式下, 这是一个RXD2 端口引脚变化检测器的额外功能。

### AUXR13: 辅助寄存器 13

SFR 页 = 仅 A 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	S3PS0	0	S2PS0	0	T4FCS	T4PS1	T4PS0
W	R/W	W	R/W	W	R/W	R/W	R/W

Bit 4: S2PS0, 串口2引脚选择0.

S2PS0	RXD2	TXD2
0	P3.2	P3.3
1	P6.7	P5.7

**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 &amp; S3MI端口引脚选择.

SnMIPS	S0MI	S1MI	S2MI	S3MI
0	P6.2	P6.3	P7.0	P7.1
1	P5.7	P6.7	P6.6	P6.5

## 21. 串口 3 (UART3)

MGEQ1C064装备有第四UART (以后就称作UART2)，有5种运行模式，模式0~模式4，除了下面的不同点与第二个UART (UART1)一样：

- (1) S1BRG可以共享波特率给UART0。UART3没有这个功能。
- (2) 所有UART1 的SRFs位于SFR页面1和页面2.但是所有UART3的SFRs位于SFR页面 4.

UART3其它所有功能可以参考UART1功能定义。所有UARTs都可以在相同或不同的模式和通信速度下同时运行。

### 21.1. 串口 3 波特率发生器 (S3BRG)

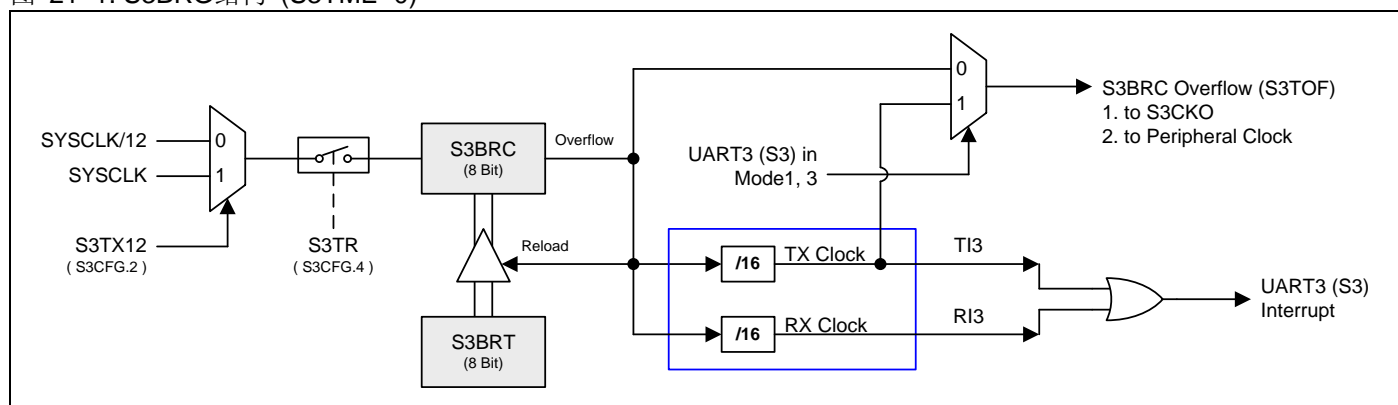
MGEQ1C064在模式1和模式3有一个嵌入式波特率发生器给串口3提供UART时钟。它由一个8位的向上计数器(S3BRC)和一个自动装载寄存器(S3BRT)构成。S3BRC的溢出(S3TOF)是UART3在模式1和模式3串行引擎的时间基准并且触发S3BRT的值重载到S3BRC继续计数。

如果S3TR=0, 软件写S3BRT数据将同时修改S3BRC。在S3TR使能开启S3BRC计数之后, 当S3BRT写入时不影响S3BRC。修改S3BRC不会影响S3BRT的值。

这里有一个S3BRC溢出率的二分频(S3TOF/2)的额外时钟输出(S3CKO)。S3TOF也供给其它外设时钟输入源切换。无论S3运行还是待运行, S3BRG总是服务于这些外设的时间基准功能。

串口3波特率发生器结构见图 21-1

图 21-1. S3BRG结构 (S3TME=0)



## 21.2. 串口 3 波特率设定

### 21.2.1. 模式 0 波特率

S3模式0波特率公式: (n=3)

$$\text{Sn Mode 0 Baud Rate} = \frac{F_{\text{SYSCLK}}}{12} \quad ; \text{SnTX12}=0$$

$$\text{or} = \frac{F_{\text{SYSCLK}}}{4} \quad ; \text{SnTX12}=1$$

### 21.2.2. 模式 2 波特率

S3模式2波特率公式: (n=3)

$$\text{Sn Mode 2 Baud Rate} = \frac{2^{\text{SnMOD1}}}{64} \times F_{\text{SYSCLK}} \quad ; \text{SnM0X3}=0$$

$$\text{or} = \frac{2^{\text{SnMOD1}}}{192} \times F_{\text{SYSCLK}} \quad ; \text{SnM0X3}=1$$

表 21.1. S3模式2波特率@ F<sub>SYSCLK</sub>=11.0592MHz

波特率	S3M0X3	S3MOD1	误差
172800	0	0	0.0%
345600	0	1	0.0%
57600	1	0	0.0%
115200	1	1	0.0%

表 21.2. S3模式2波特率@ F<sub>SYSCLK</sub>=12.00MHz

波特率	S3M0X3	S3MOD1	误差
187500	0	0	0.0%
375000	0	1	0.0%
62500	1	0	0.0%
125000	1	1	0.0%

### 21.2.3. 模式 1 & 3 波特率

S3模式1, 3 波特率公式: (n=3)

$$\text{Sn Mode 1, 3 Baud Rate} = \frac{2^{\text{SnMOD1}}}{32} \times \frac{F_{\text{SYSCLK}}}{12 \times (256 - \text{SnBRT})} \quad ; \text{SnTX12}=0$$

$$\text{or} = \frac{2^{\text{SnMOD1}}}{32} \times \frac{F_{\text{SYSCLK}}}{1 \times (256 - \text{SnBRT})} \quad ; \text{SnTX12}=1$$

### 21.2.4. 串口 3 模式 4 (SPI 主机)

MGEQ1C064的串口3模式4(SPI主机)与串口1模式4(SPI主机)完全兼容。详细功能请参考串口1说明。

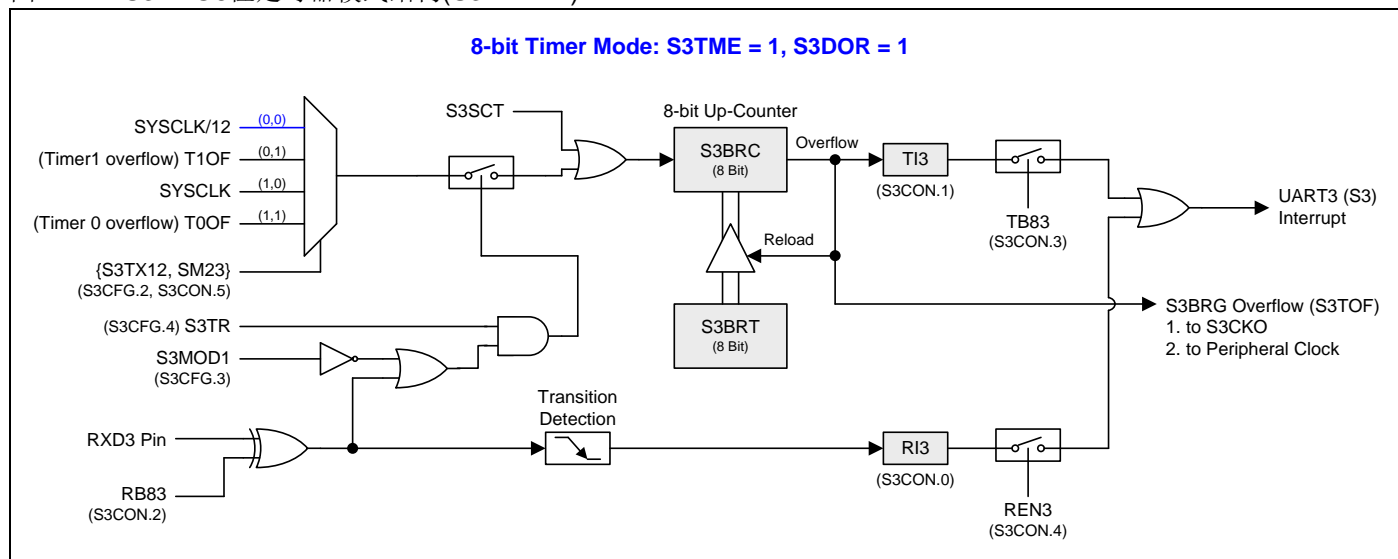
### 21.3. S3BRG 8 位定时器模式

如果UART3不用或软件搁置，MGEQ1C064设定S3TME=1串行口3波特率发生器(S3BRG)提供纯定时器操作模式。此定时器操作在一个8位自动重载定时器并且提供溢出标志TI3(S3CON.1)置位。RI3(S3CON.0)服务于RXD3端口引脚的端口变化检测器。在此模式下TI3和RI3保持UART3中断源的中断能力并且有一个独立中断使能控制(TB83和REN3)。RB83选择RXD3端口输入的RI3检测电平。如果RB83=0，RI3由REN3=1并且检测到RXD3引脚下降沿置位；否则RI3将检测RXD3端口引脚的上升沿。在MCU掉电模式，RI3被强制成电平触发操作并且如果UART3中断使能可以唤醒CPU。

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S3BRC溢出，它可以触发端口引脚输出。S3CKOE=1使能S3CKO输出在端口引脚并且屏蔽RI3中断。

S3BRG 8位定时器模式结构如图 21-2所示。

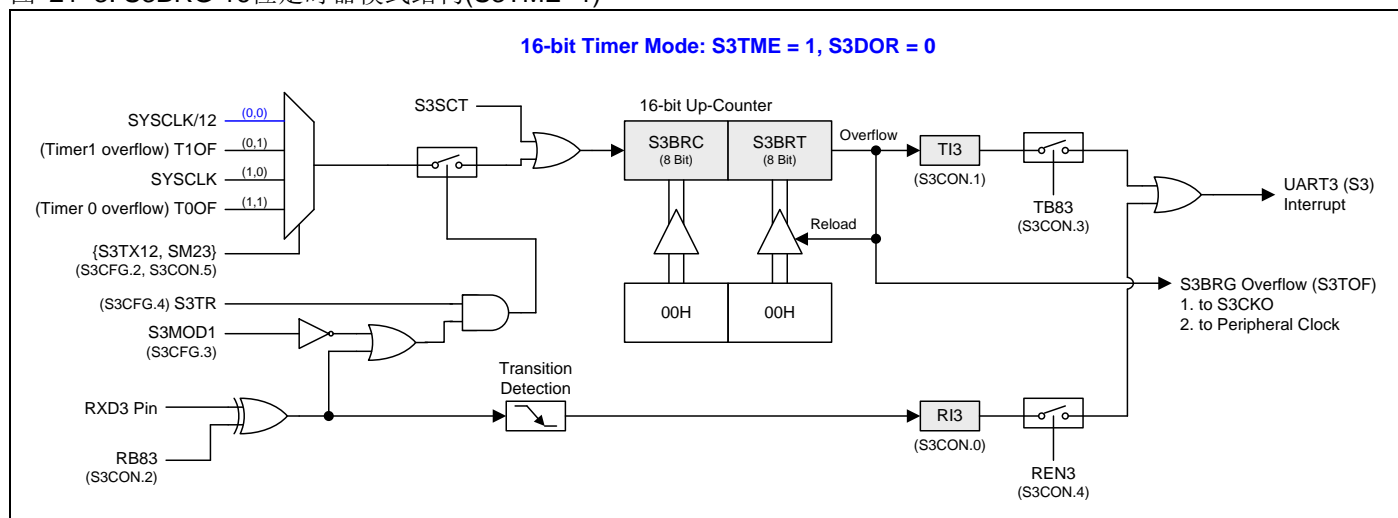
图 21-2. S3BRG8位定时器模式结构(S3TME=1)



### 21.4. S3BRG 16 位定时器模式

S3BRG 16位定时器模式结构如图 21-3所示。

图 21-3. S3BRG 16位定时器模式结构(S3TME=1)



### 21.5. S3BRT 可编程时钟输出

当S3BRC溢出时，溢出标志S3TOF为S3CKO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(S3BRC)时钟源。定时器重复计数到被载入值溢出。一旦溢出，S3BRC载入S3BRT的值继续计数。



## 21.6. 串口3寄存器

下面的特殊功能寄存器与UART3的操作有关:

### S3CON: 串口3控制寄存器

SFR 页 = 仅 4 页

SFR 地址 = 0x98

复位值 = 0000-0000

7	6	5	4	3	2	1	0
SM03	SM13	SM23	REN3	TB83	RB83	TI3	RI3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SM03, 串口3模式位0.

Bit 6: SM13, 串口3模式位1.

SM33	SM03	SM13	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, /32或/192, /96
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	保留	保留
1	1	1	7	保留	保留

Bit 5: 串口3模式位2.

0: 禁止SM23功能。

1: 在模式2和3时使能地址自动识别, 如果SM23=1那么RI3将不能设置, 除非接收到的第9位数据(RB83)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SM23=1那么RI3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地地址或者是一个广播地址; 在模式0, SM23可以为0。

Bit 4: REN3, 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: TB83, 在模式2和3时第9位数据被传送, 需要通过软件置位或清零。

Bit 2: RB83, 在模式2和3时收到的第9位数据。在模式1, 如果SM23=0, RB83是收到数据的停止位。在模式0, RB83没有使用。

Bit 1: TI3, 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: RI3, 接收中断标志。

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(除留意SM23外), 在接收停止位的中间时刻由硬件置位。

### S3BUF: 串口3缓冲寄存器

SFR 页 = 仅 4 页

SFR 地址 = 0x99

复位值 = XXXX-XXXX

7	6	5	4	3	2	1	0
S3BUF.7	S3BUF.6	S3BUF.5	S3BUF.4	S3BUF.3	S3BUF.2	S3BUF.1	S3BUF.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。

**S3BRT: 串口3 波特率 定时器重载寄存器**

SFR 页 = 仅 4 页

SFR 地址 = 0x9A

复位值 = 0000-0000

7	6	5	4	3	2	1	0
S3BRT.7	S3BRT.6	S3BRT.5	S3BRT.4	S3BRT.3	S3BRT.2	S3BRT.1	S3BRT.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。

**S3BRC: 串口3 波特率计数器寄存器**

SFR 页 = 仅 4 页

SFR 地址 = 0x9B

复位值 = 0000-0000

7	6	5	4	3	2	1	0
S3BRC.7	S3BRC.6	S3BRC.5	S3BRC.4	S3BRC.3	S3BRC.2	S3BRC.1	S3BRC.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S3CFG1.S3TME = 0，软件写数据到S3BRT同时数据存入S3BRT和S3BRC。

**S3CFG: 串口3配置寄存器**

SFR 页 = 仅 4 页

SFR 地址 = 0x9C

复位值 = 0010-0000

7	6	5	4	3	2	1	0
SM33	S3MOX3	S3DOR	S3TR	S3MOD1	S3TX12	S3CKOE	S3TME
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: SM33, 串口3模式控制位3。

SM33	SM03	SM13	模式	描述	波特率
0	0	0	0	移位寄存器	SYSCCLK/12或SYSCCLK/4
0	0	1	1	8-位UART	可变
0	1	0	2	9-位UART	SYSCCLK/64, /32或/192,/96
0	1	1	3	9-位UART	可变
1	0	0	4	SPI 主机	SYSCCLK/12或 SYSCCLK/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 6: S3MOX3, 串口3模式0、模式2和模式4波特率选择。

串口3模式0和模式4:

0: 清零选择SYSCCLK/12作串口3模式0和模式4波特率。

1: 置位选择SYSCCLK/4作串口3模式0和模式4波特率。

串口32模式2:

0: 清零选择SYSCCLK/32或/64作串口3波特率。

1: 置位选择SYSCCLK/96或/192作串口3波特率。

Bit 5: S3DOR, 串口3所有操作模式的数据位序控制。

如果S3TME = 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S3DOR为“1”。

如果S3TME = 1:

0: 设置S3BRG为16位重载定时器/计数器模式。

1: 设置S3BRG为8位定时器/计数器模式。

Bit 4: S3TR, UART3波特率发生器控制位。

0: 清零关闭S3BRG。

1: 置位开启S3BRG。

Bit 3: S3MOD1, UART3双倍波特率选择使能位。

0: 禁止UART3双倍波特率功能。

1: 使能UART3双倍波特率功能。

Bit 2: S3TX12, UART3波特率发生器时钟源选择。

0: 清零选择SYSCLK/12作S3BRG的时钟源。

1: 置位选择SYSCLK作S3BRG的时钟源。

Bit 1: S3CKOE, 串口3波特率发生器时钟输出使能。

0: 禁止S3CKO 在端口引脚输出。

1: 使能S3CKO 在端口引脚输出。

Bit 0: S3TME, 串口3波特率(BRG)定时器模式使能。

0: 保持S3BRG服务串口3 (UART3)。

1: 禁止串口3功能并且S3BRG作为一个8位自动装载的定时器。这个模式下, 这是一个RXD3 端口引脚变化检测器的额外功能。

### AUXR13: 辅助寄存器 13

SFR 页 = 仅 A 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	S3PS0	0	S2PS0	0	T4FCS	T4PS1	T4PS0
W	R/W	W	R/W	W	R/W	R/W	R/W

Bit 6: S3PS0, 串口3引脚选择0.

S3PS0	RXD3	TXD3
0	P3.6	P3.7
1	P6.5	P6.6

### 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

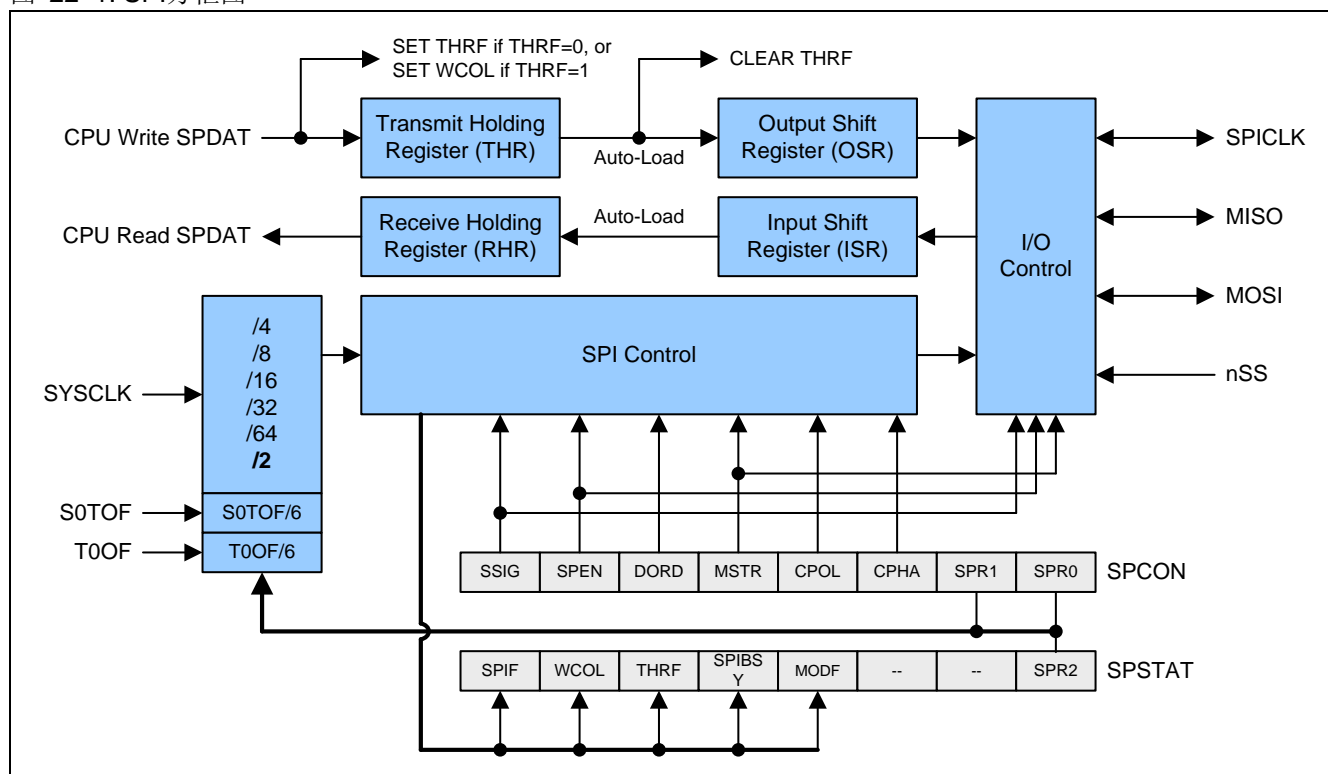
Bit 1: SnMIPS, S0MI, S1MI, S2MI & S3MI端口引脚选择..

SnMIPS	S0MI	S1MI	S2MI	S3MI
0	P6.2	P6.3	P7.0	P7.1
1	P5.7	P6.7	P6.6	P6.5

## 22. 串行外设接口(SPI)

MGEQ1C064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 22-1. 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。

- /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“22.2.3 nSS引脚模式改变”)。

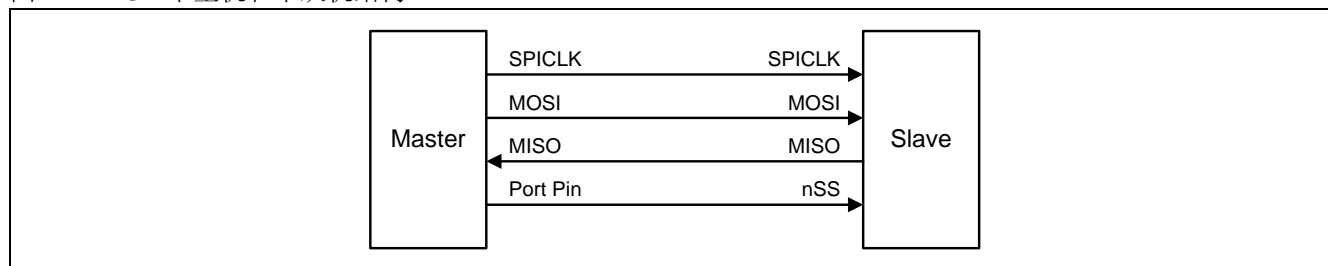
## 22.1. 典型 SPI 配置

### 22.1.1. 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 22-2. SPI单主机和单从机结构



### 22.1.2. 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“22.2.3 nSS引脚模式改变”)。

图 22-3. SPI双设备结构，既是主机也是从机

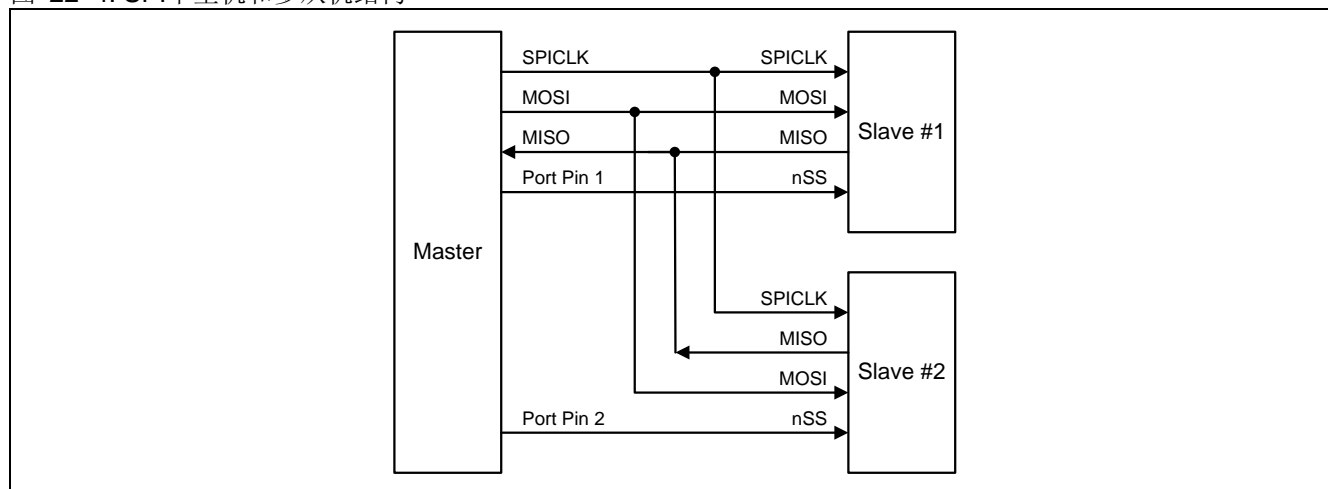


### 22.1.3. 单主机和多从机

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 22-4. SPI单主机和多从机结构



## 22.2. SPI 配置

表 22.1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和引脚状态。

表 22.1. SPI主机和从机选择

SPEN (SPCON.6)	SSIG (SPCON.7)	nSS -引脚	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”

### 22.2.1. 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。

### 22.2.2. 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。

### 22.2.3. 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。

#### 22.2.4. 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。

#### 22.2.5. 写冲突

MGEQ1C064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。

#### 22.2.6. 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 22.2所示。

表 22.2. 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/6	可变	可变
1	1	1	T0OF/6	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。

## 22.3. 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 22.3. 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 22-5. SPI在CPHA=0时从机传送格式

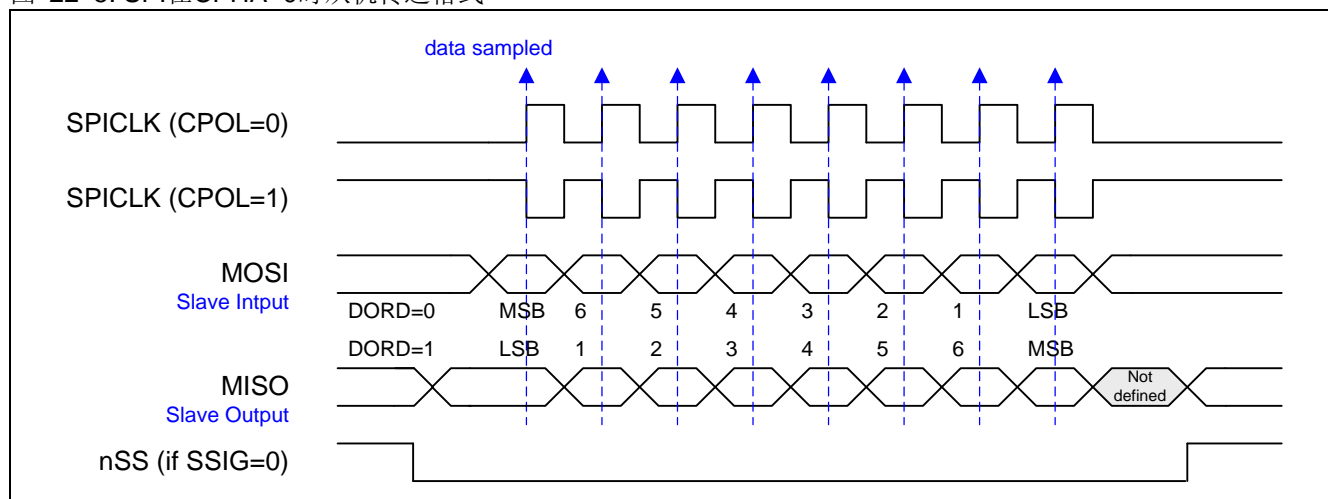


图 22-6. SPI在CPHA=1时从机传送格式

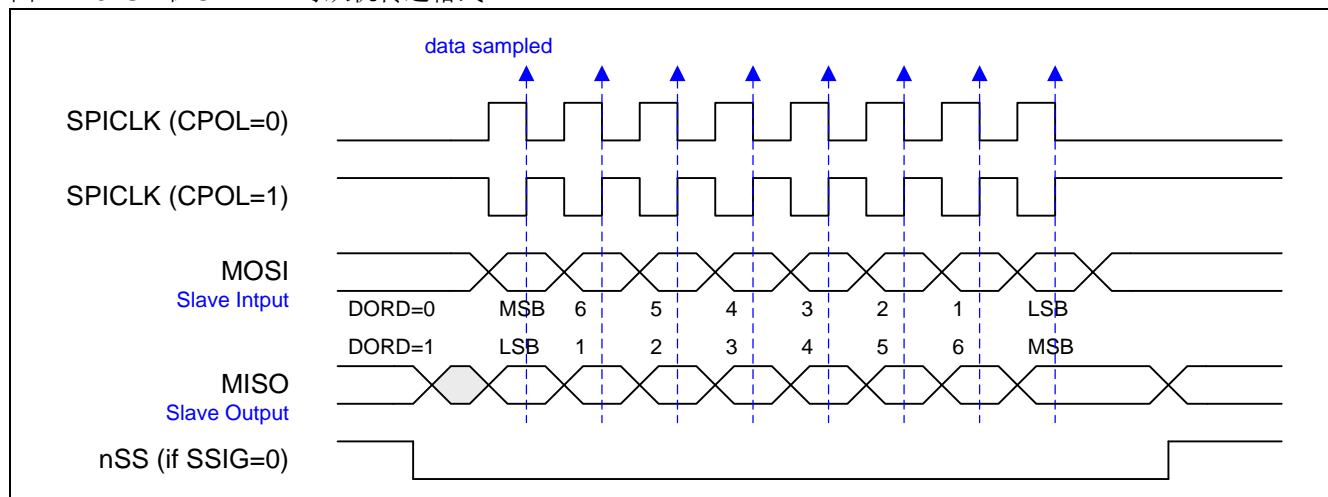


图 22-7. SPI在CPHA=0时主机传送格式

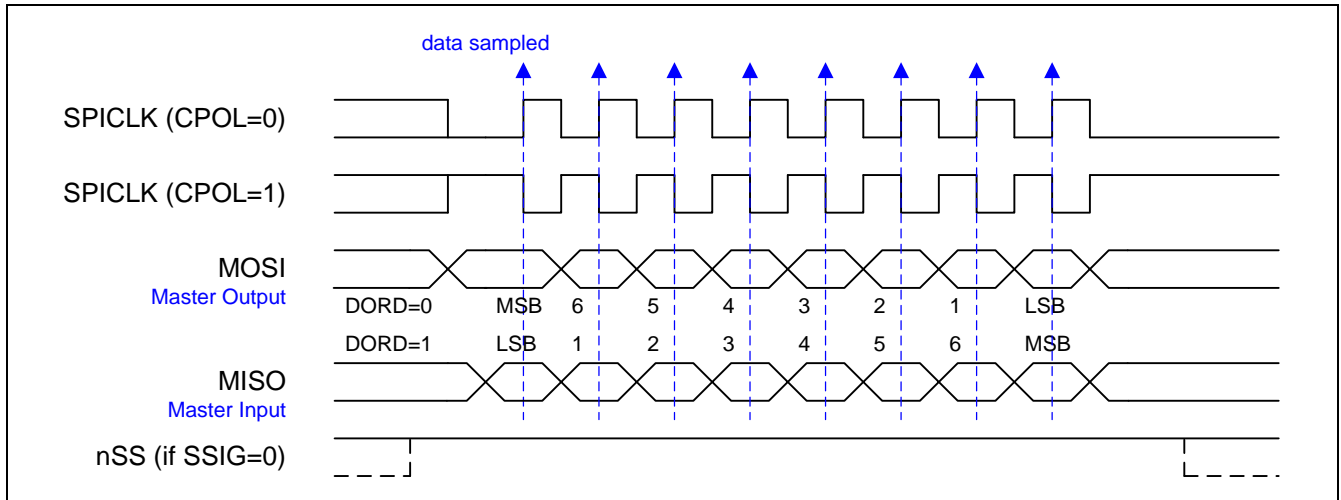
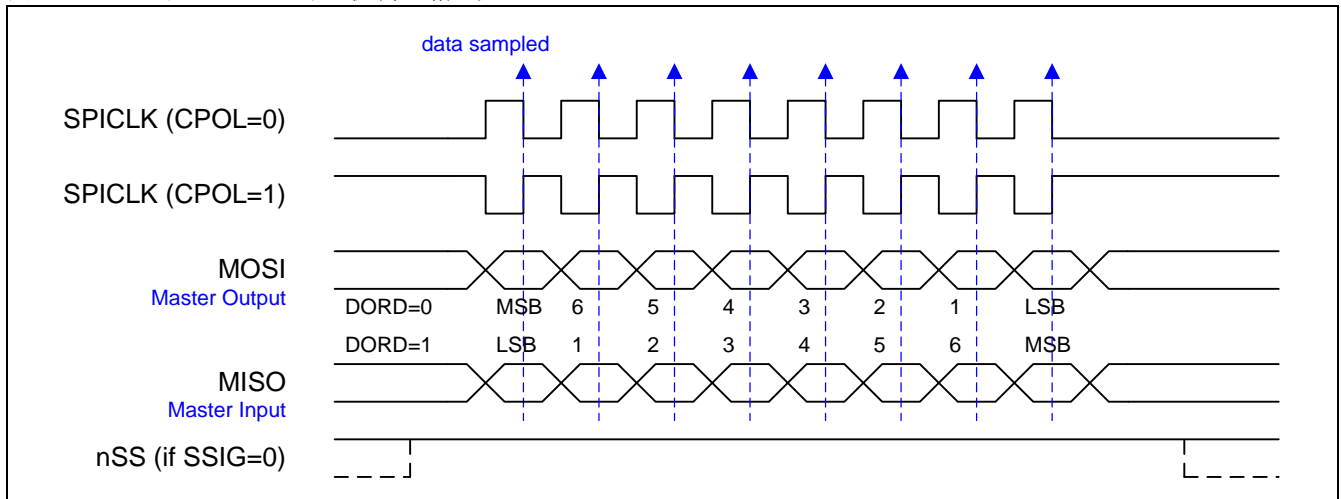


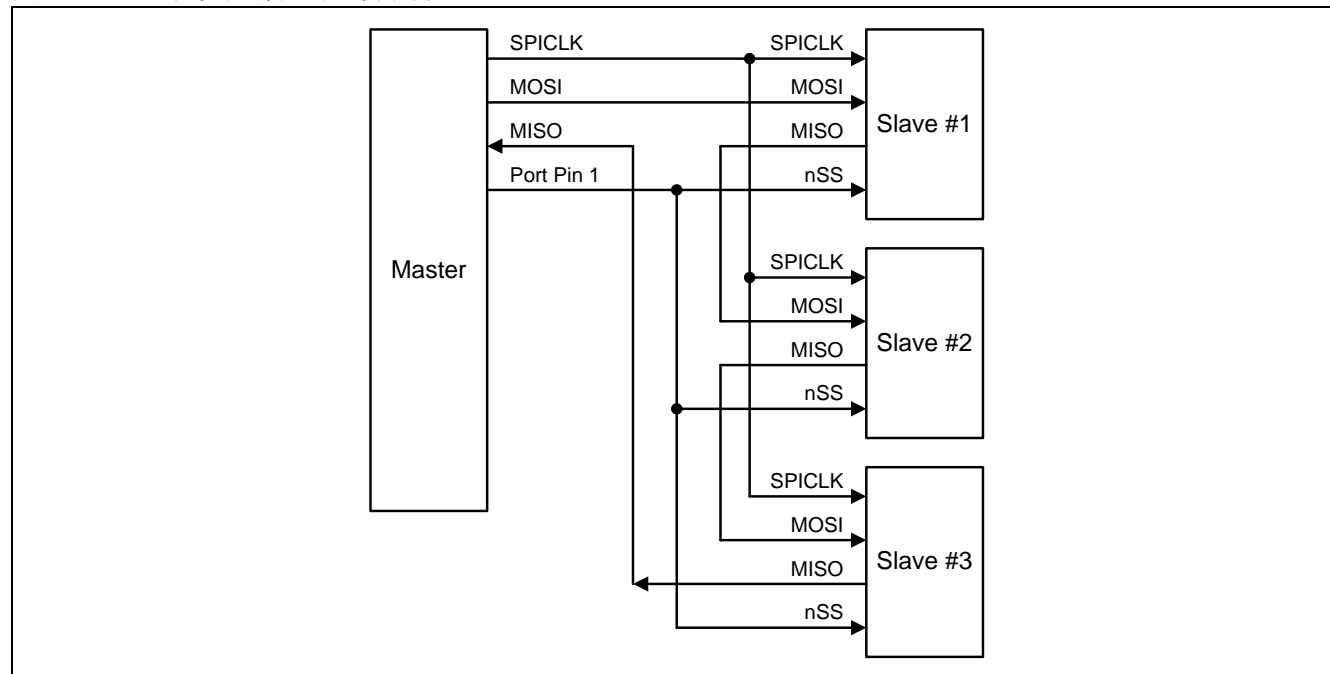
图 22-8. SPI在CPHA=1时主机传送格式



## 22.4. 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 22-9. SPI从机在菊花链连接结构



### 22.4.1. 菊花链配置

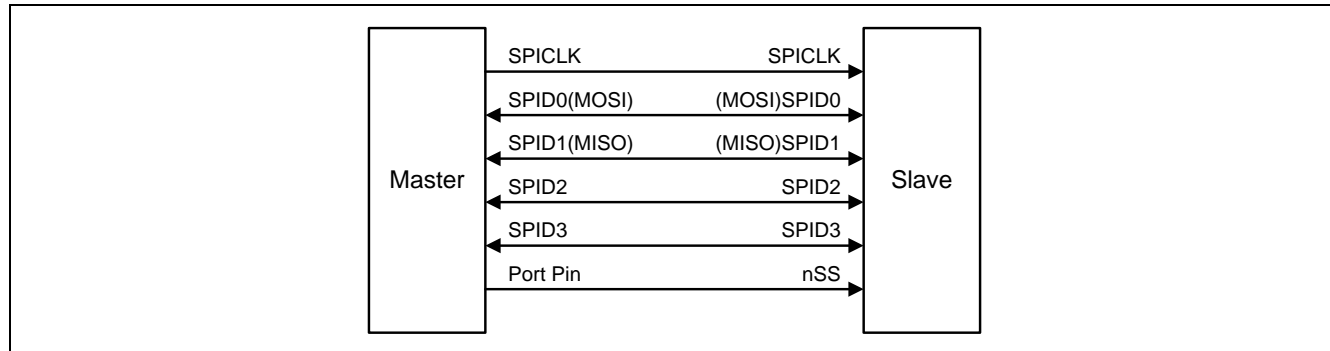
#### 如何配置 SPI 从机为菊花链

- 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

## 22.5. QPI 模式连接

MGEQ1C064中的SPI模块提供四线数据传输模式来支持QPI协议。在此QPI模式下，所有SPI操作功能：主机/从机，模式0/1/1/2/3，MSB/LSB配置，nSS控制，均通过软件编程实现。但是四数据线只执行双向、半双工传输。这四个数据线被定义为SPID0(MOSI)、SPID1(MISO)、SPID2和SPID3。

图 22-10. QPI单主机&单从机结构



### 22.5.1. QPI 模式配置

#### 如何配置 QPI 模式

- 配置SPCON来选择SPI0工作模式
- 如果SPI0是QPI主机模式，定义{SPR2, SPR1, SPR0}去选择SPICLK 速度
- 置位QPIEN (SPSTAT.1)来使能QPI模式
- 编程QDOE (SPSTAT.0)去控制QPI数据总线输入或输出
- 写SPDAT 触发SPI0在QPI模式里工作

## 22.6. 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~F

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	<b>SYSClk/2</b>	<b>6 MHz</b>	<b>24 MHz</b>
1	1	0	<b>S0TOF/6</b>	可变	可变
1	1	1	<b>T0OF/6</b>	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”

**SPSTAT: SPI状态寄存器**

SFR 页 = 0~F

SFR 地址 = 0x84

复位值 = 0000-XXX0

7	6	5	4	3	2	1	0
SPIF	WCOL	THRF	SPIBSY	MODF	0	QPIEN	SPR2 /QDOE
R/W	R/W	R	R	R/W	W	R/W	R/W

Bit 7: SPIF, SPI传输完成标志。

0: 软件写“1”清零SPIF。

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”清零WCOL。

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“22.2.5 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。

Bit 2: 保留位, 写寄存器时, 此位必须写“0”。

Bit 1: QPIEN, QPI模式使能

0: 禁止QPI模式

1: 使能QPI模式。

如果QPIEN=0,

Bit 0: SPR2, SPI时钟率选择位2(与SPR1和SPR0相配合)

如果QPIEN=1,

Bit 0: QDOE, QPI数据输出使能

0: 禁止QPI 数据输出

1: 使能QPI数据输出

QPIEN	PSTAT.2	SPCON.1	SPCON.0	CPU控制功能
0	SPR2	SPR1	SPR0	SPI时钟选择
1	QDOE	0	0	QPI/OPI 控制

**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。

**AUXR7: 辅助寄存器 7**

SFR 页 = 仅 4 页

SFR 地址 = 0xA4

复位值 = 1100-0000

7	6	5	4	3	2	1	0
POE5	POE4	C0CKOE	SPI0M0	BM1OE1	BM1OE0	BM0OE1	BM0OE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 4: SPI0M0, SPI0模式控制位0。控制SPI 的菊花链连接。

0: 禁止这个模式控制。

1: 使能这个模式控制。

**AUXR10: 辅助寄存器 10**

SFR 页 = 仅 7 页

SFR 地址 = 0xA4

复位值 = 1100-0000

7	6	5	4	3	2	1	0
AC1HC0	AC0HC0	SPIPS1	SPIPS0	S0PS1	SPFACE	TWICF	PAA
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[1:0].

Pin Options	SPIPS1~0	nSS	MOSI (SPID0)	MISO (SPID1)	SPICLK	SPID2	SPID3
0	0 0	P1.4	P1.5	P1.6	P1.7	P5.3	P5.4
1	0 1	P4.3	P4.2	P4.1	P4.0	P3.6	P3.7
2	1 0	P4.0	P4.1	P4.2	P4.3	P6.5	P5.7
3	1 1	P4.3	P4.1	P4.2	P4.0	P6.5	P5.7

Bit 2: SPFACE, SPIF自动清零使能位

0: 禁止, SPIF只能软件清零

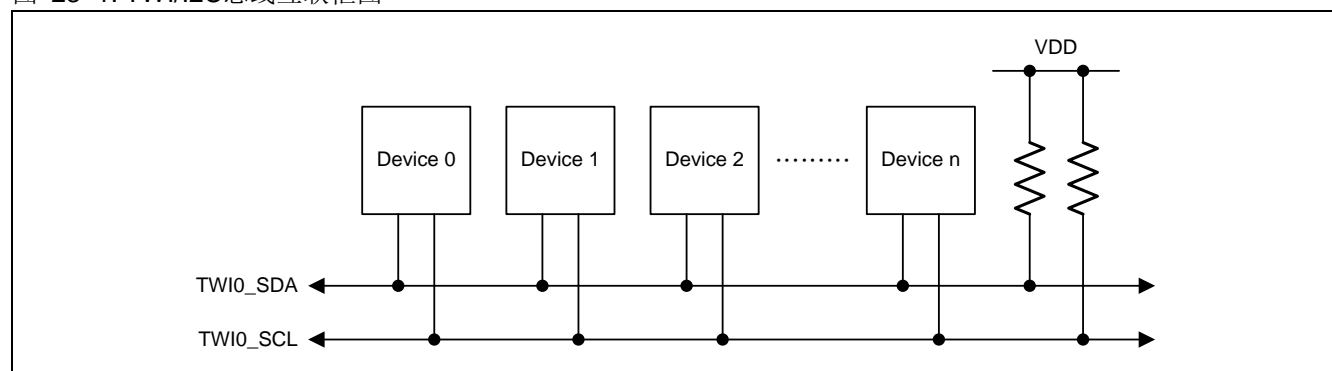
1: 使能. SPIF会被CPU读/写SPDAT操作清零

## 23. 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。**MGEQ1C064**内嵌两个TWI/I2C, TWI0/I2C0 和 TWI1/I2C1。两者都支持多从机地址识别。

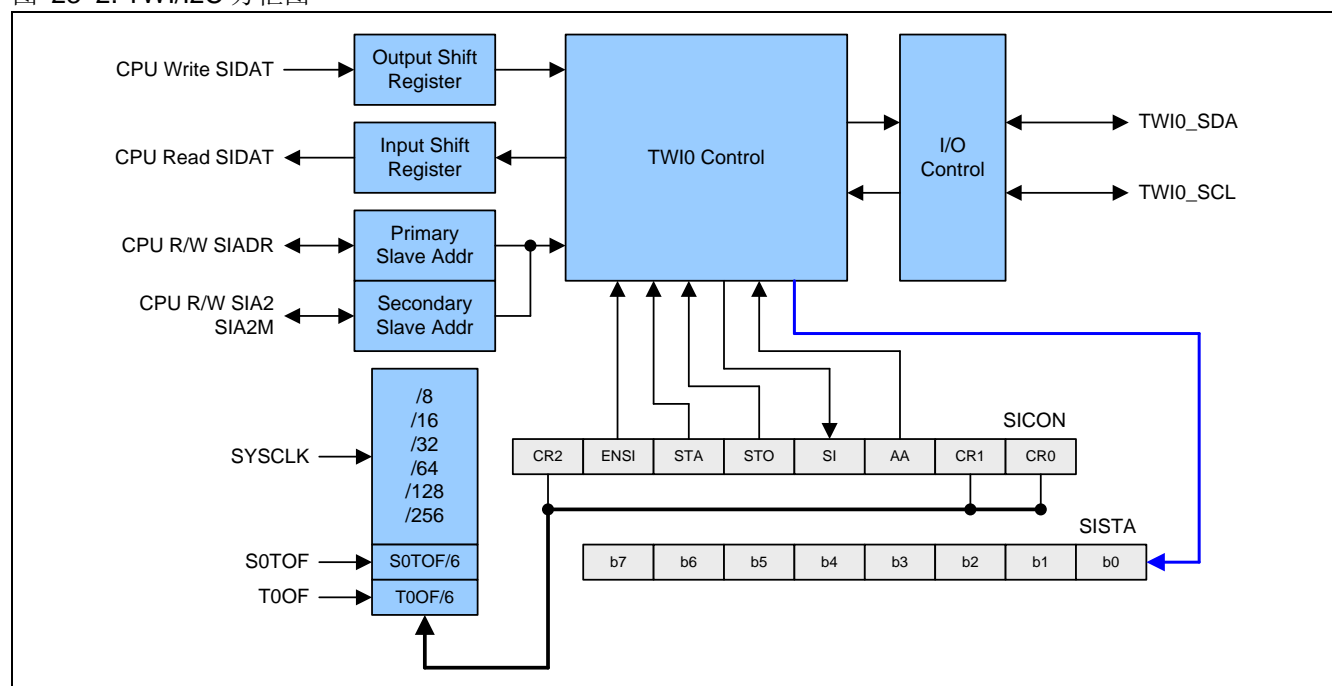
TWI/ I2C协议允许系统设计人员只用两根双向传输线来连接多达128个不同的设备, 一根用于时钟(TWI0\_SCL), 一根用于数据(TWI0\_SDA)。双线串行接口(TWI)由TWI0\_SDA(串行数据)、TWI0\_SCL(串行时钟)控制产生和同步, 仲裁逻辑以及起始/停止(START/STOP)。唯一需要的外部硬件就是在TWI/ I2C的每根传输线上添加一个上拉电阻。所有连接到总线的设备都有各自的地址, 而且TWI/ I2C协议解决了总线仲裁的问题。

图 23-1. TWI/I2C总线互联框图



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 23-2. TWI/I2C 方框图



## 23.1. 操作模式

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。

### 23.1.1. 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。

### 23.1.2. 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。

### 23.1.3. 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。

### 23.1.4. 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。

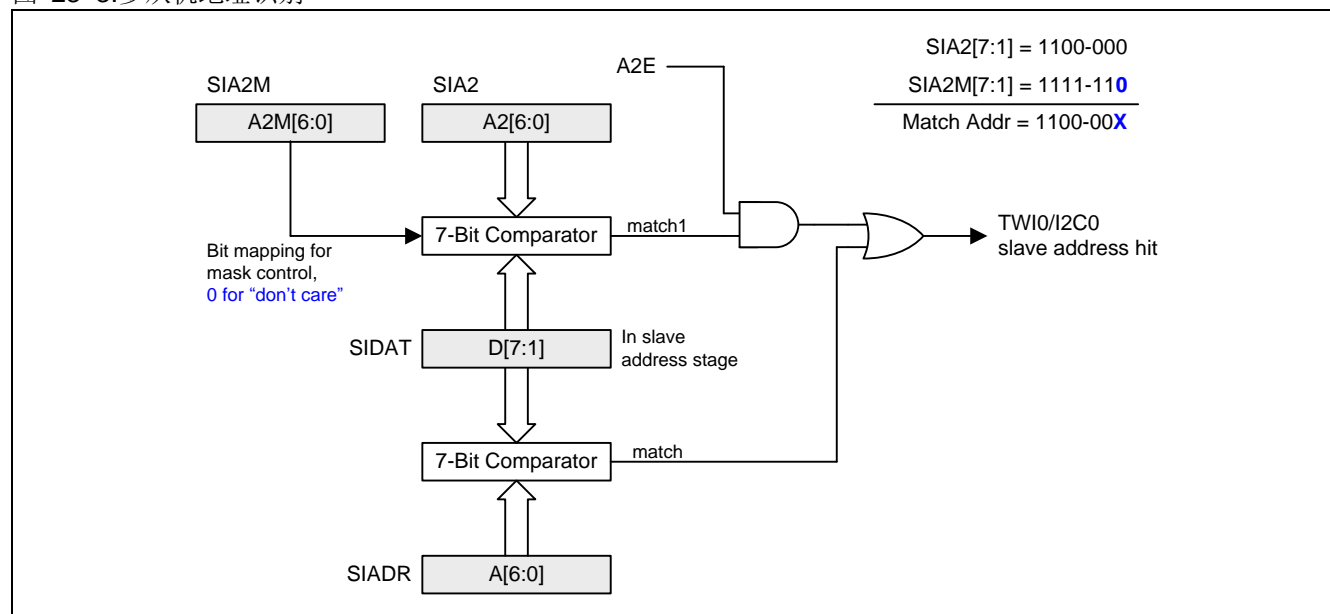
SIADR与SICON初始化后，TWI/ I2C会等待直到其从机地址被寻址并且数据方向为“0” (W)，TWI/ I2C将工作于从机接收模式。在接收到其从机地址与“W”位后，串行中断标志(SI)置位，并且可以从SISTA读出一个可用的状态编码。这些状态编码可以用作指示一个中断服务程序，在这些状态编码下对应的操作将在随后的工作流程图详细叙述。当TWI/ I2C处于主机模式时，如果仲裁失败可能进入从机接收模式(参考状态68H和78H)。

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。

## 23.1.5. 多从机地址识别

在MGEQ1C064中SIADR定义TWI/I2C的首从地址。MGEQ1C064还提供带有掩码功能的从地址，该掩码功能在SIA2和SIA2M上实现。在从地址掩码SIA2M[7:1]的位元位置上的1，使能所接收的从地址与这些位元的从地址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 23-3.多从机地址识别



## 23.2. 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:

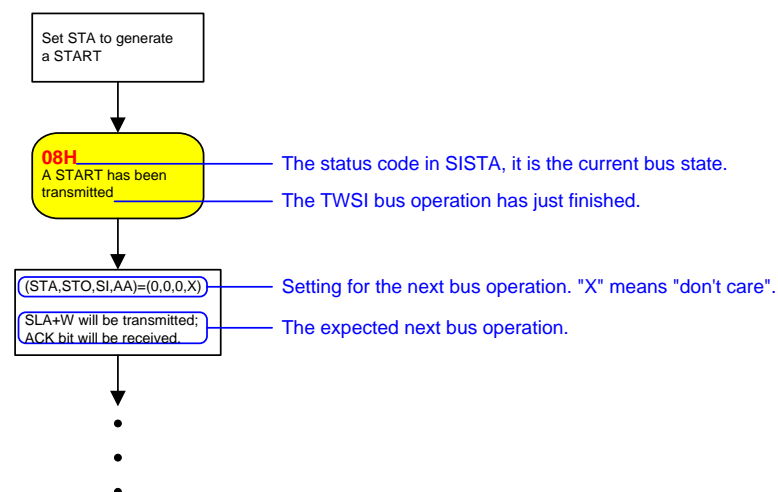
这个状态编码表明在一个TWI/ I2C串行传输过程中发生总线错误。当一个START或STOP信号在一帧的非法位置发送时，总线错误就会发生。例如：在传输一个字节地址、数据时，或者在应答位。总线错误也会在外界干扰扰乱内部TWI/ I2C信号时发生。当总线错误发生时，SI被置位，STO标志必须置位并且SI必须软件清零用来从总线错误中恢复。这会使TWI/ I2C进入“未被寻址”的从机状态(已定义的状态)并且清除STO标志(SICON的其它位不受影响)。TWI/ I2C0\_SDA与TWI/ I2C0\_SCL线将被释放(不会发送STOP信号)。

## 23.3. 使用 TWI/I2C

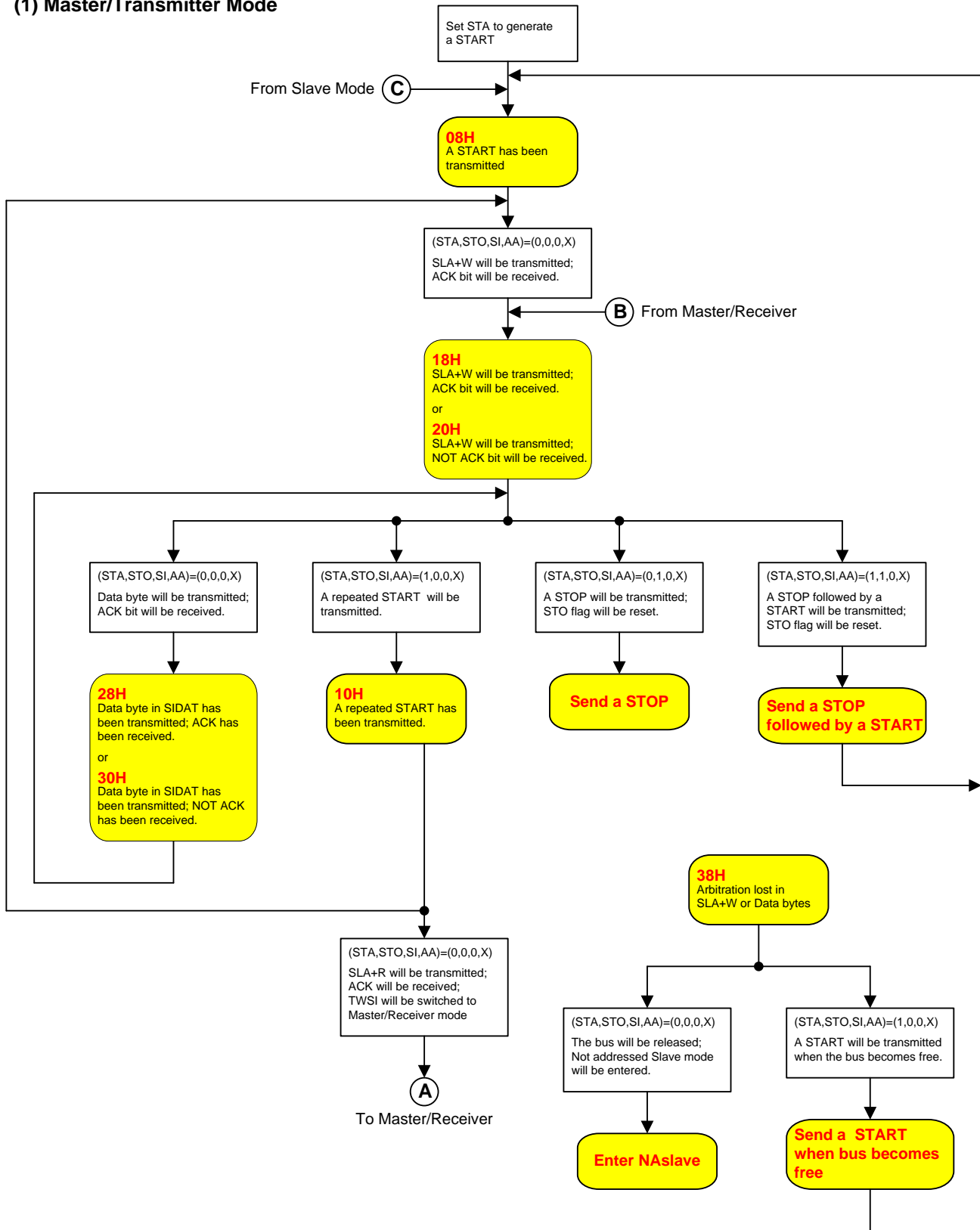
TWI/ I2C是面向字节并且基于中断的。中断会在所有总线事件后发生，例如接收到一字节数据或发送START信号。因为TWI/ I2C是基于中断的，应用软件可以自由的一个TWI/ I2C字节发送的过程中处理其它工作。注意，TWI/ I2C中断标志位(EIE1.6)与EA位允许应用程序选择在SI标志出现时是否产生中断请求。当SI标志出现时，表明TWI/ I2C已经完成一个操作并且等待程序响应。此时状态寄存器SISTA保存的状态编码表明TWI/ I2C总线的当前状态。用户程序可以通过对STA, STO和AA位(在SICON中)进行适当的编程来决定接下来TWI/ I2C总线将如何运行。

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。

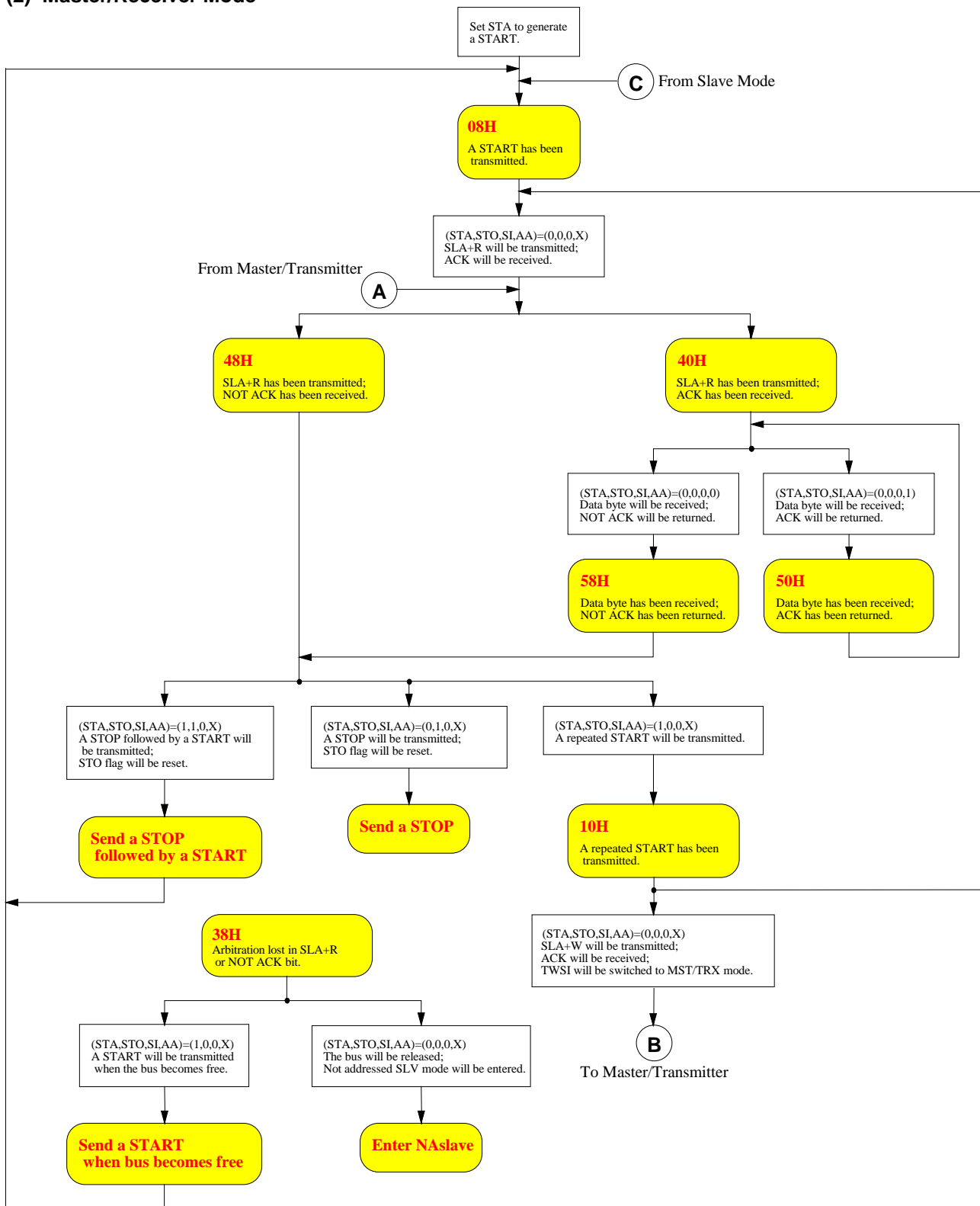
下面的图指出如何读流程图



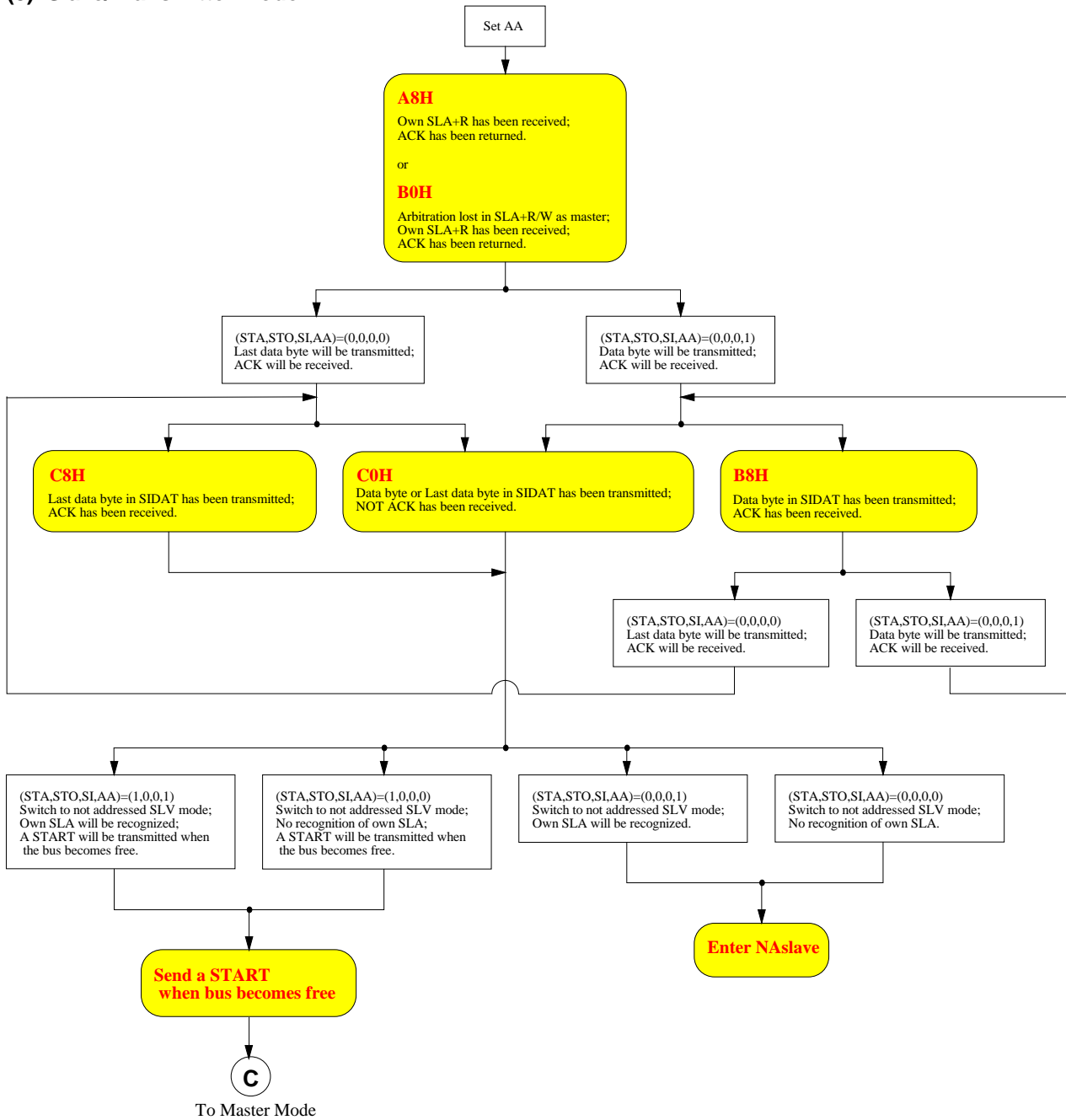
(1) Master/Transmitter Mode



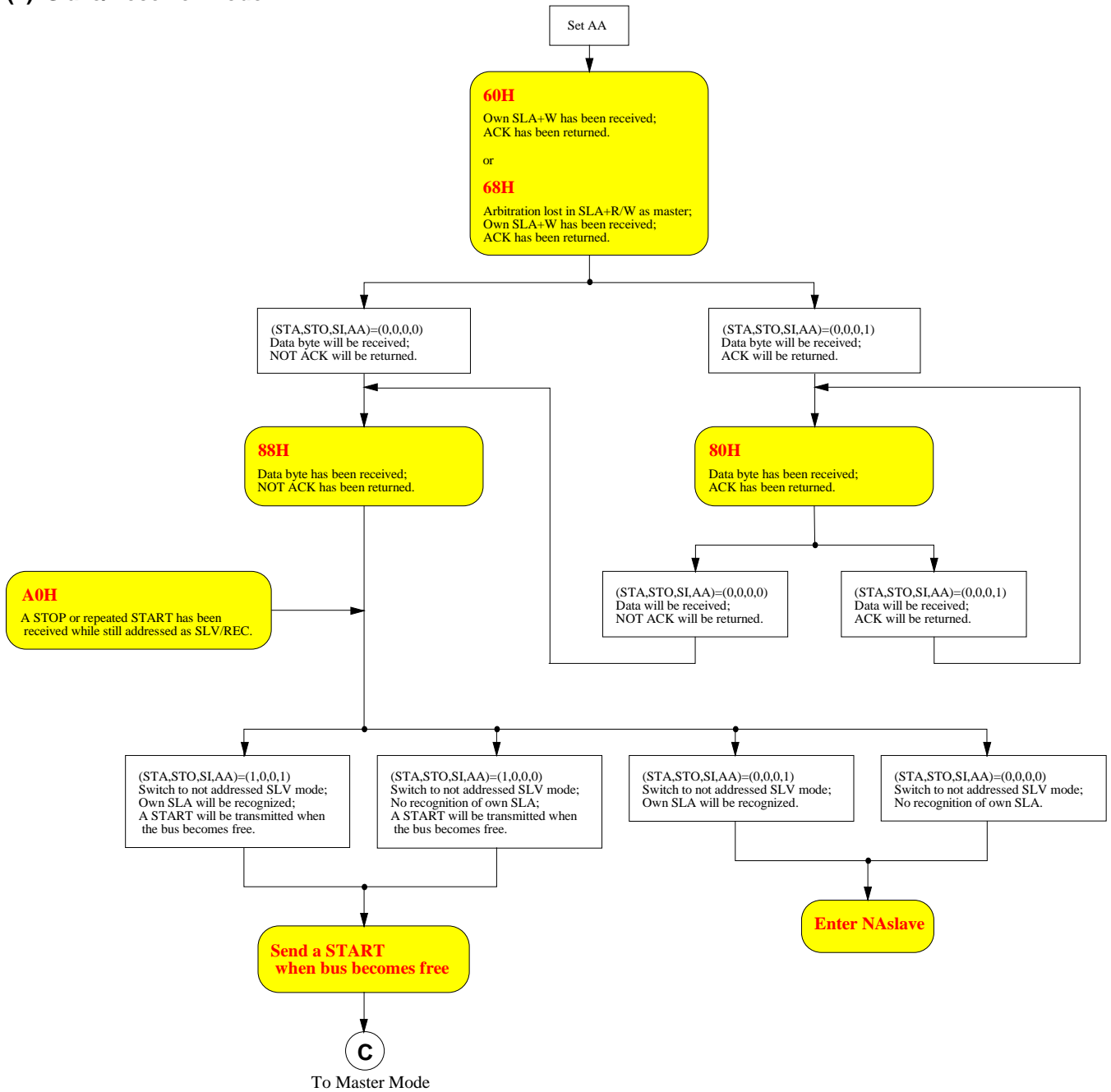
(2) Master/Receiver Mode



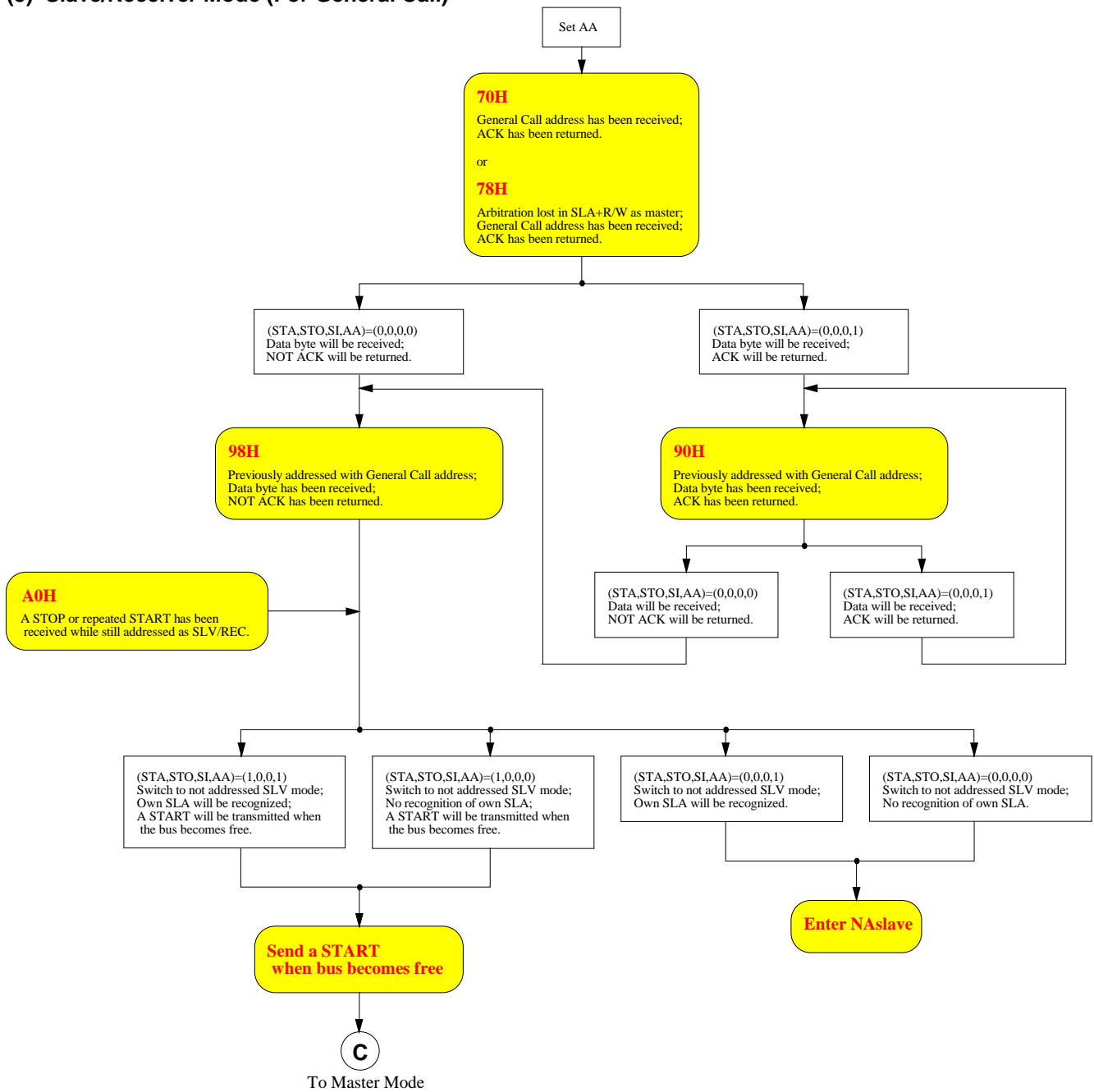
(3) Slave/Transmitter Mode



(4) Slave/Receiver Mode



(5) Slave/Receiver Mode (For General Call)



## 23.4. 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

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，寄存的高七位必须被用于本机的从机地址，并且当最低位(GC)置位时，广播地址(00H)会被识别，否则忽略。在START状态后，最高位与从TWI0/I2C0总线上收到的首位相对应。

### 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 23.1 给出不同的时钟速率设置：

表 23.1. TWI0/I2C0 串行时钟速率

CR2	CR1	CR0	TWI0/ I2C0时钟选择	TWI0/ I2C0时钟率@ SYSCLK=12MHz
0	0	0	SYSCLK/8	1.5 MHz <sup>注1</sup>
0	0	1	SYSCLK/16	750 KHz
0	1	0	SYSCLK/32	375 KHz
0	1	1	SYSCLK/64	187.5 KHz
1	0	0	SYSCLK/128	93.75 KHz
1	0	1	SYSCLK/256	46.875 KHz
1	1	0	S0TOF/6	可变
1	1	1	T0OF/6	可变

注1:

1. 最大TWI0/I2C0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。

**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 第二地址寄存器**

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的第二从机地址.

Bit 0: A2E,第二从机地址识别使能控制

0: 禁止第二从机地址识别

1: 使能第二从机地址识别

**SIA2M: TWI0/I2C0 第二地址掩码寄存器**

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	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：

$$\begin{array}{rcl}
 \text{SIA2}[7:1] & = & 1100\ 000 \\
 \text{SIA2M}[7:1] & = & 1111\ 110 \\
 \hline
 \text{2}^{\text{nd}}\ \text{ADR}[7:1] & = & 1100\ 00x \quad \longrightarrow \quad \text{第二从机地址将被检测，除了位1被“忽略”外。}
 \end{array}$$

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	P2.5	P2.3

**AUXR10: 辅助寄存器 10**

SFR 页 = 仅 7 页

SFR 地址 = 0xA4

复位值 = 1100-0000

7	6	5	4	3	2	1	0
AC1HC0	AC0HC0	SPIPS1	SPIPS0	S0PS1	SPFACE	TWICF	PAA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1: TWICF, TWI0/I2C0串行时钟输入滤波

0: 禁止TWICF功能

1: 使能TWICF 功能

Bit 0: PAA, 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

## 23.5. 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

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，寄存的高七位必须被用于本机的从机地址，并且当最低位(GC1)置位时，广播地址(00H)会被识别，否则忽略。在START状态后，最高位与从TWI1/I2C1总线上收到的首位相对应。

### 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。当在总线上检测到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 23.2 给出不同的时钟速率设置：

表 23.2. TWI1/I2C1串行时钟速率

CR21	CR11	CR01	TWI1/I2C1时钟选择	TWI1/I2C1时钟率@ SYSCLK=12MHz
0	0	0	SYSCLK/8	1.5 MHz <sup>注1</sup>
0	0	1	SYSCLK/16	750 KHz
0	1	0	SYSCLK/32	375 KHz
0	1	1	SYSCLK/64	187.5 KHz
1	0	0	SYSCLK/128	93.75 KHz
1	0	1	SYSCLK/256	46.875 KHz
1	1	0	S0TOF/6	可变
1	1	1	T0OF/6	可变

注1

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。

**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第二地址寄存器**

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的第二从机地址.

Bit 0: A2E1,第二从机地址识别使能控制

0: 禁止地址从机地址识别

1: 使能地址从机地址识别

**SI1A2M: TWI1/I2C1第二地址掩码寄存器**

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	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：

$$\begin{array}{rcl}
 \text{SI1A2}[7:1] & = & 1100\ 000 \\
 \text{SI1A2M}[7:1] & = & 1111\ 110 \\
 \hline
 \text{2}^{\text{nd}}\ \text{ADR}[7:1] & = & 1100\ 00x \quad \longrightarrow \quad \text{第二从机地址将被检测，除了位1被“忽略”外。}
 \end{array}$$

Bit 0: 保留位，写寄存器时，此位必须写“1”。

**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
I2C1CF	PAA1	I2C1PS1	I2C1PS0	0	POEM0	COM0	COOFS
R/W	R/W	R/W	R/W	W	R/W	R/W	R/W

Bit 7: I2C1CF, TWI1/I2C1串行时钟输入滤波

0: 禁止 I2C1CF 功能.

1: 使能 I2C1CF 功能.

Bit 6: PAA1, TWI1/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PAA功能

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

I2C1PS1~0	TWI1_SCL	TWI1_SDA
0 0	P4.2	P4.3
0 1	P6.0	P6.1
1 0	P6.5	P6.6
1 1	P6.3	P6.2

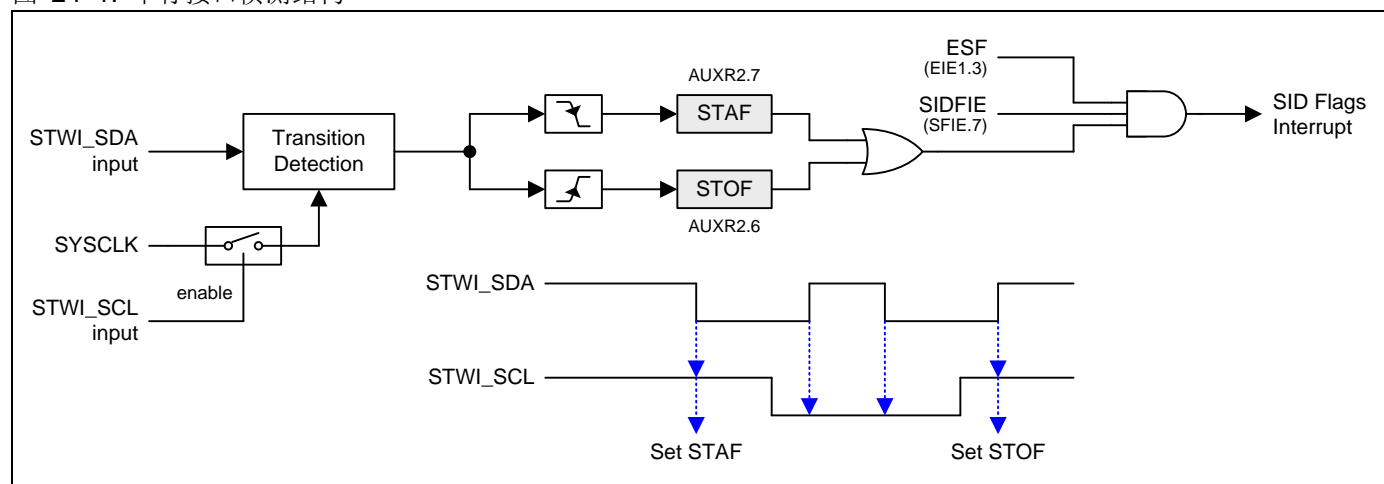
## 24. 串行接口侦测(STWI/SI2C)

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侦测到，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。

### 24.1. SID 结构

STAF和STOF侦测的结构，中断结构和事件侦测波形如图 24-1所示。

图 24-1. 串行接口侦测结构



### 24.2. 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	0	COPLK	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	W	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

7	6	5	4	3	2	1	0
SIDPS1	SIDPS0	T1G1	T0G1	C0FDC1	C0FDC0	S1PS1	S1PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

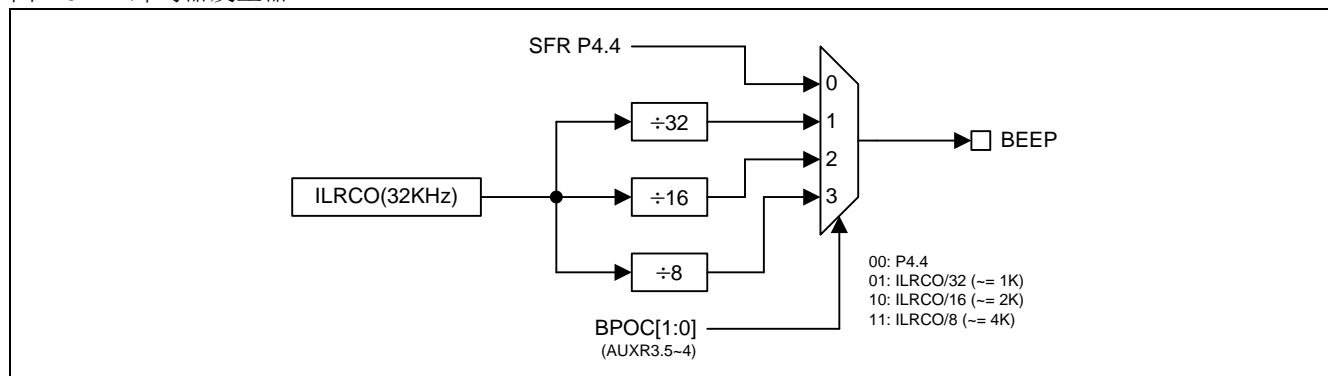
Bit 7~6: SID/STWI 端口引脚侦测 [1:0].

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

## 25. 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 25-1 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“37.5 ILRCO特性”。

图 25-1. 蜂鸣器发生器



### 25.1. 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。

## 26. 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。

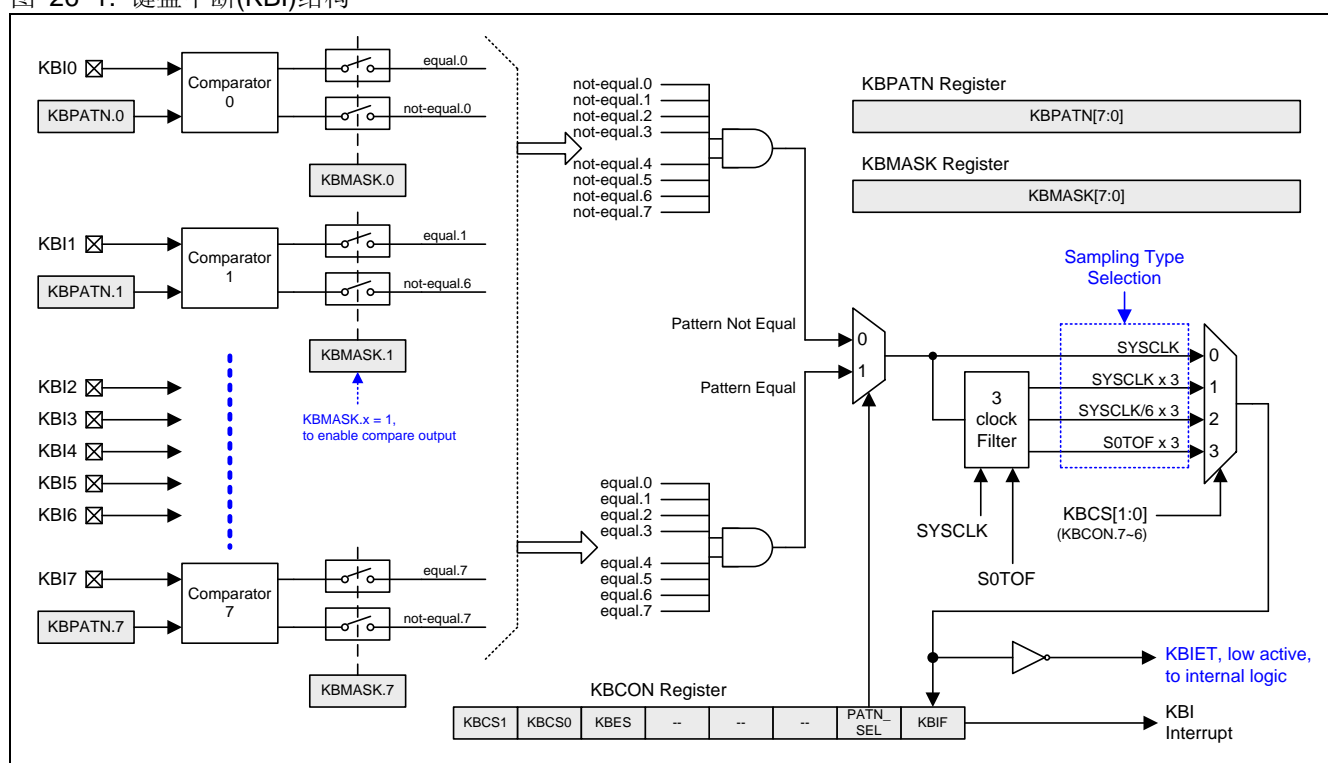
有3个特殊功能寄存器与此功能相关。键盘中断掩码寄存器(KBMASK)用来定义哪些KBI输入引脚可以产生中断。键盘样式寄存器(KBPATN)用来定义与键盘输入值进行比较的值，比较匹配时硬件置位键盘中断控制寄存器(KBCON)中的键盘中断标志(KBIF)，若EIE1中的EKBI中断允许且EA=1，则还会产生一个中断。键盘中断控制寄存器(KBCON)中的PATN\_SEL位用来定义比较是“相等”还是“不等”匹配。键盘输入可以分配给不同的端口引脚，详情请参考章节“4.3 功能复用”。

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。在“不相等”的模式下需要重写 KBPATN 以清除内部逻辑以允许下一次按键事件。

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。

### 26.1. KBI 结构

图 26-1. 键盘中断(KBI)结构



## 26.2. 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: 键盘控制寄存器**

SFR 页 = 0~F

SFR 地址 = 0xD6

复位值 = 0000-0000

7	6	5	4	3	2	1	0
KBCS1	KBCS0	KBES	--	0	0	PATN_SEL	KBIF
R/W	R/W	R/W	R	W	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 3 ~ 2: 保留位。软件在写入KBCON时, 必须向这些位写入"0"。

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	P0.7	P0.6	P0.5	P0.4
1 0	P2.7	P2.6	P2.5	P2.4
1 1	P6.3	P6.2	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	P0.3	P0.2	P0.1	P0.0
1 0	P2.3	P2.2	P2.1	P2.0
1 1	P4.3	P4.2	P4.1	P4.0

## 27. 通用逻辑(GPL-CRC)

MGEQ1C064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)。

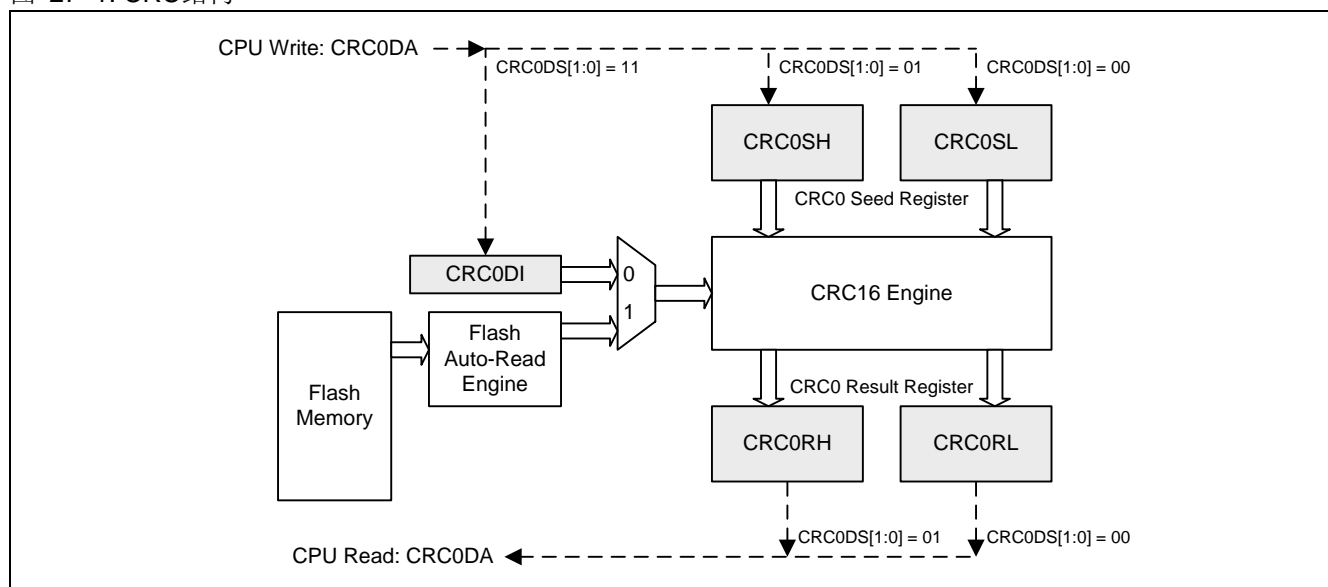
### 27.1. GPL-CRC 结构

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自动重载模式下，需要保持CRCDS1~0在“0x00”。并且如下步骤所示：

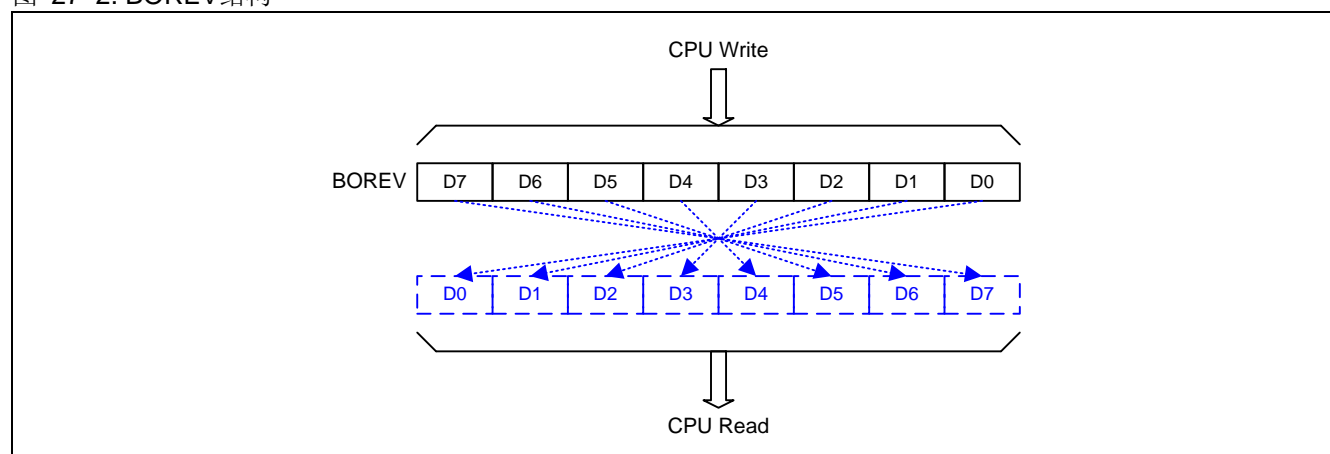
1. 设置重载区的起始地址IFADRH和IFADRL。
2. 结合IAPLB(7位)和9位1-1111-1111设置结束地址。
3. 设置IFMT寄存器(ISP/IAP Flash 模式)为0x80作为Flash自动重载模式。
4. 顺序将0x46h和0xB9h写入SCMD寄存器触发CRC计算。

图 27-1. CRC结构



## 27.2. GPL-BOREV 结构

图 27-2. BOREV结构



## 27.3. 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:

CRCDS1~0	CPU R/W	CRC0数据选择	描述
00	写	CRC0SL	CRC0数据种子低位寄存器。
01	写	CRC0SH	CRC0数据种子高位寄存器。
10	写	--	保留。
11	写	CRC0DI	CRC0数据输入寄存器。
00	读	CRC0RL	CRC0结果低位寄存器。
01	读	CRC0RH	CRC0结果高位寄存器。
10	读	--	保留。
11	读	--	保留。

### 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
--	--	CRCDS1	CRCDS0	--	--	--	DPS
W	W	R/W	R/W	W	W	W	R/W

Bit 5~4: CRCDS1~0. CRC0数据端口选择1~0.

**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:

如果0xA0写入到BOREV, 读回来将是0x05。

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IFMT时, 这些位必须写“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。
Others	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}。

## 28. 输出信号中止&调制 0/1 (OBM0/1)

输出信号中止与调制有两个主要功能，一是调制，二是事件中止。

当使用OBM作为调制器时，OBM可以将数据流BMnSW(也称为调制器信号)与载波信号BMnLCO和BMnHCO混合输出调制输出。

当BMnSW = “0” ， BMnLCO将被选择输出。

当BMnSW = “1” ， BMnHCO将被选择输出。

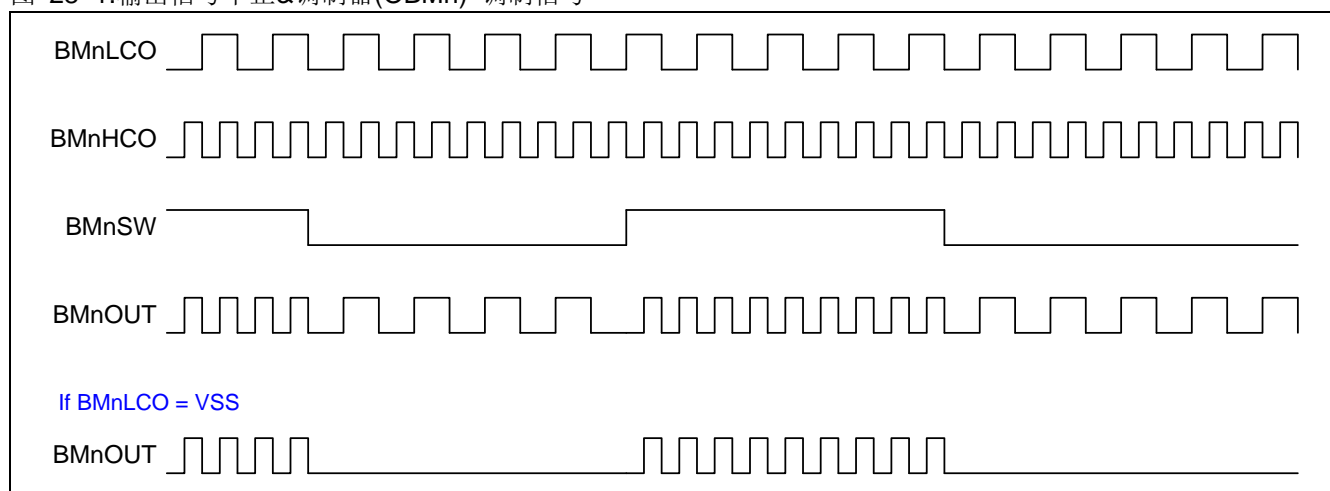
因此，可以将调制信号和载波信号作为目标信号调制到BMnOUT。

在调制器模式下，建议MnFLT[1:0] = 00禁用滤波功能，以保留调制信号的全部信息

当OBM作为事件中止器时，用户可以选择BMnLCO作为主控信号，BMnHCO作为替换信号。通过使用BMnSW作为中断事件触发器。该触发信号可以是BMnM1~0 = 00时的电平触发，也可以是BMnM1~0 = 01/10/11时的下降缘触发。

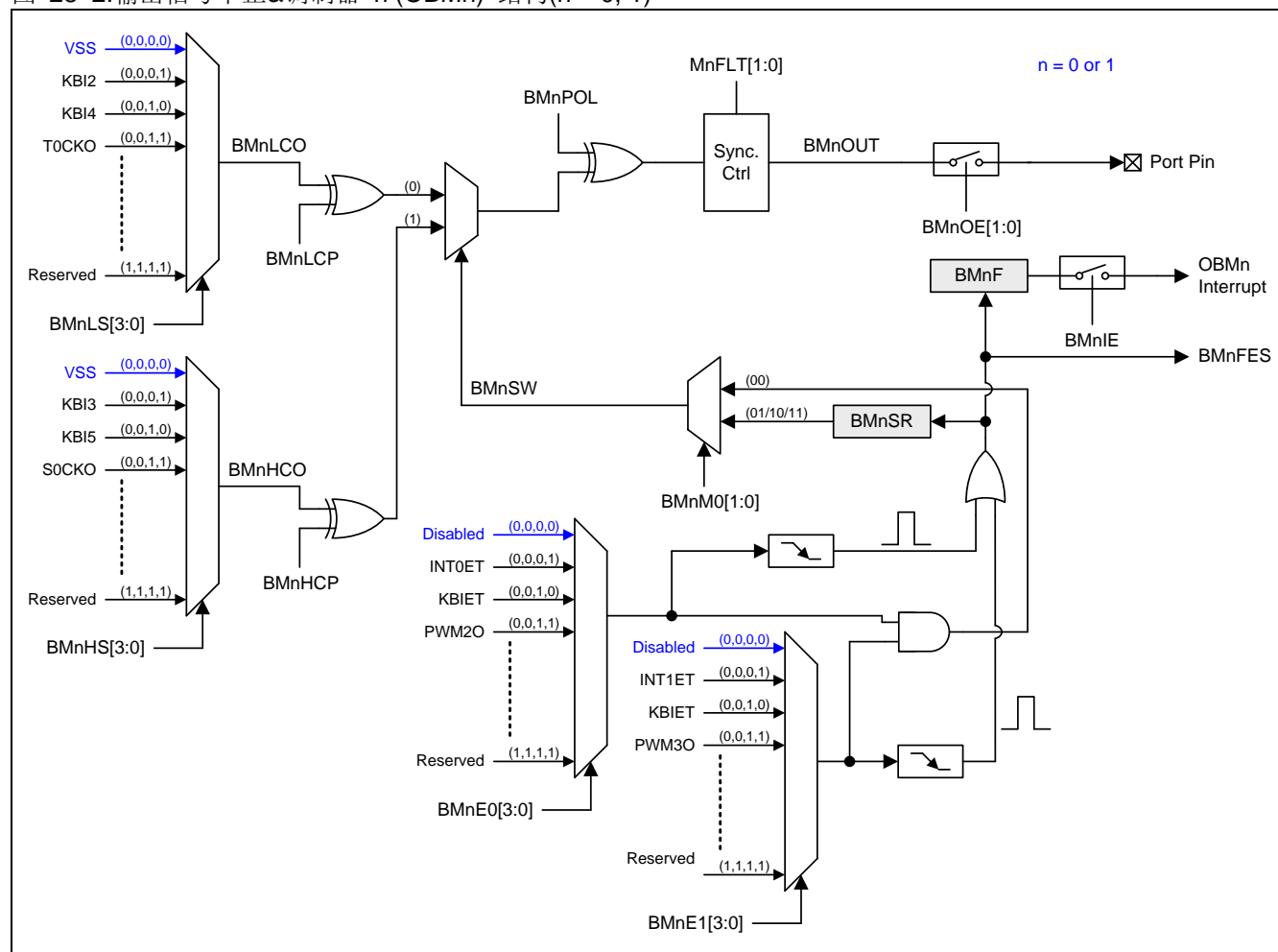
在事件中止器正常工作的情况下，通过OBM将主控信号传至端口引脚。但当中止事件到来时，它将切换到替换信号。边缘触发器甚至可以由软件设置BMnSR寄存器来保持。

图 28-1.输出信号中止&调制器(OBMn) 调制信号



### 28.1. OBM 结构

图 28-2. 输出信号中止&调制器 n (OBMn) 结构(n = 0, 1)



## 28.2. OBM0 寄存器

OBM0操作相关的特殊功能寄存器如下：

### BM0CR:输出信号中止&调制器0控制寄存器

SFR 页 = 仅 2 页

SFR 地址 = 0xBC

复位值 = 0100-0000

7	6	5	4	3	2	1	0
BM0OUT	BM0SW	BM0IE	BM0POL	MOFLT1	MOFLT0	BM0M1	BM0M0
R	R	R/W	R/W	R/W	R/W	R/W	R/W
7	6						
M0SRWE	BM0SR						
W	W						

Bit 7: BM0OUT, 仅读。软件读它得到OBM0的输出信号。

Bit 7: M0SRWE, BM0SR (Bit 6) 写使能, 仅写。

0: 禁止软件更新BM0SR的值。

1: 使能软件更新BM0SR的值。此位自动清零。因此, 必须同时写入这个位“1”和BM0SR的值

Bit 6: BM0SW, 仅读。软件读它获得OBM0输出MUX的开关信号

Bit 6: BM0SR, OBM0开关寄存器, 仅写。软件写这个位与M0SRWE=1 来更新BM0SR的内容

Bit 5: BM0IE, 使能BM0F (AUXR0.2)中断

0: 禁止BM0F中断

1: 使能BM0F中断

Bit 4: BM0POL, OBM0输出反相控制位

0: BM0OUT输出信号不反相

1: BM0OUT输出信号反相

Bit 3~2: M0FLT1~0, OBM0输出滤波控制

M0FLT1~0	OBM0输出滤波模式
00	禁止
01	SYSCLK x 3
10	SYSCLK/6 x 3
11	T3OF x 3

Bit 1~0: BM0M1~0, OBM0工作模式选择

BM0M1~0	OBM0模式选择
00	BM0SW由BM0E0 与 BM0E1 使能信号控制 (电平切换)
01	BM0SW 由BM0SR控制。BM0SR可以由软件更新, 也可以通过BM0E0事件或BM0E1事件的下降沿进行清除。
10	BM0SW由BM0SR控制。BM0SR可以通过软件进行更新, 也可以通过BM0E0事件或BM0E1事件的下降沿进行置位。
11	BM0SW由BM0SR控制。BM0SR可以通过软件进行更新, 也可以通过BM0E0事件或BM0E1事件的下降沿进行切换。

**BM0SE: 输出信号中止&调制器0切换源使能寄存器**

SFR 页 = 仅 3 页

SFR 地址 = 0xBC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
BM0E1.3	BM0E1.2	BM0E1.1	BM0E1.0	BM0E0.3	BM0E0.2	BM0E0.1	BM0E0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: OBM0 使能 1 控制位 3~0.

BM0E1[3:0]	OBM0控制选择
0 0 0 0	禁止
0 0 0 1	INT1ET
0 0 1 0	KBIET
0 0 1 1	PWM3O
0 1 0 0	T1CKO
0 1 0 1	T2CKO
0 1 1 0	T2EXI
0 1 1 1	PWM4O
1 0 0 0	BOD1ET
1 0 0 1	INT3ET
1 0 1 0	S1DO
1 0 1 1	AC1OUT
1 1 0 0	PWM6O
1 1 0 1	T4CKO
1 1 1 0	T4EXI
其它	禁止

Bit 3~0: OBM0 使能 0 控制位 3~0.

BM0E0[3:0]	OBM0控制选择
0 0 0 0	禁止
0 0 0 1	INT0ET
0 0 1 0	KBIET
0 0 1 1	PWM2O
0 1 0 0	T0CKO
0 1 0 1	S0DO
0 1 1 0	AC0OUT
0 1 1 1	PWM5O
1 0 0 0	INT2ET
1 0 0 1	RTCKO
1 0 1 0	T3CKO
1 0 1 1	T3EXI
1 1 0 0	--
1 1 0 1	AC2OUT
其它	禁止

**BM0LCR: 输出信号中止&调制器0低通道寄存器**

SFR 页 = 仅 4 页

SFR 地址 = 0xBC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
BM0LCP	0	0	0	BM0LS.3	BM0LS.2	BM0LS.1	BM0LS.0
R/W	W	W	W	R/W	R/W	R/W	R/W

Bit 7: BM0LCP, OBM0低通道输出反相控制位

0: BM0LCO 输出信号不反相

1: BM0LCO 输出信号反相

Bit 6~4: 保留位。软件在写入BM0LCR时，必须向这些位写入"0"。

Bit 3~0: BMOLS, OBM0低通道源选择位3~0.

BMOLS[3:0]	OBM0低通道源选择
0 0 0 0	VSS
0 0 0 1	KBI2
0 0 1 0	KBI4
0 0 1 1	T0CKO
0 1 0 0	INT0ET
0 1 0 1	PWM0O
0 1 1 0	PWM1O
0 1 1 1	PWM4O
1 0 0 0	T2CKO
1 0 0 1	T2EXI
1 0 1 0	INT2ET
1 0 1 1	RTCKO
1 1 0 0	PWM6O
其它	保留

**BM0HCR: 输出信号中止&调制器0高通道寄存器**

SFR 页 = 仅 5 页

SFR 地址 = 0xBC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
BM0HCP	0	0	0	BM0HS.3	BM0HS.2	BM0HS.1	BM0HS.0
R/W	W	W	W	R/W	R/W	R/W	R/W

Bit 7: BM0HCP, OBM0高通道输出反相控制位

0: BM0HCO输出信号不反相

1: BM0HCO输出信号反相

Bit 6~4: 保留位。软件在写入BM0HCR时, 必须向这些位写入"0"。

Bit 3~0: BM0HS, OBM0高通道源选择位3~0.

BM0HS[3:0]	高通道源选择
0 0 0 0	VSS
0 0 0 1	KBI3
0 0 1 0	KBI5
0 0 1 1	S0CKO
0 1 0 0	INT1ET
0 1 0 1	PWM2O
0 1 1 0	PWM3O
0 1 1 1	PWM5O
1 0 0 0	T3CKO
1 0 0 1	T3EXI
1 0 1 0	INT3ET
1 0 1 1	PWM7O
1 1 0 0	T4CKO
1 1 0 1	T4EXI
其它	禁止

**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
P60OC1	P60OC0	P60FD	PBKF	BM1F	BM0F	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 2: BM0F, OBM0标志位。此位由OBM0切换源使能 (BMOSE) 置位。

0: 没有OBM0切换事件发生。必须软件清零。

1: OBM0切换事件发生。

**AUXR7: 辅助寄存器 7**

SFR 页 = 仅 4 页

SFR 地址 = 0xA4

复位值 = 1100-0000

7	6	5	4	3	2	1	0
POE5	POE4	COCKOE	SPI0M0	BM1OE1	BM1OE0	BM0OE1	BM0OE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1~0: BM0OE1~0, BM0OUT输出端口引脚选择

BM0OE1~0	BM0OUT端口引脚
0 0	禁止
0 1	P4.2
1 0	P3.0
1 1	P5.0

## 28.3. OBM1 寄存器

OBM1操作相关的特殊功能寄存器如下：

### BM1CR: 输出信号中止&调制器1控制寄存器

SFR 页 = 仅 6 页

SFR 地址 = 0xBC

复位值 = 0100-0000

7	6	5	4	3	2	1	0
BM1OUT	BM1SW	BM1IE	BM1POL	M1FLT1	M1FLT0	BM1M1	BM1M0
R	R	R/W	R/W	R/W	R/W	R/W	R/W
7	6						
M1SRWE	BM1SR						
W	W						

Bit 7: BM1OUT, 仅读。软件读它得到OBM1的输出信号。

Bit 7: M1SRWE, BM1SR (Bit 6) 写使能, 仅写

0: 禁止软件更新BM1SR的值。

1: 使能软件更新BM1SR的值。此位自动清零。因此, 必须同时写入这个位“1”和BM1SR的值

Bit 6: BM1SW, 仅读。软件读它获得OBM1输出MUX的开关信号

Bit 6: BM1SR, OBM1开关寄存器, 仅写。软件写这个位与M1SRWE=1 来更新BM1SR的内容

Bit 5: BM1IE, 使能BM1F (AUXR0.3)中断

0: 禁止BM1F中断

1: 使能BM1F中断

Bit 4: BM1POL, OBM1输出反相控制位

0: BM1OUT输出信号不反相

1: BM1OUT输出信号反相

Bit 3~2: M1FLT1~0, OBM1输出滤波控制

M1FLT1~0	OBM1输出滤波模式
00	禁止
01	SYSCLK x 3
10	SYSCLK/6 x 3
11	T3OF x 3

Bit 1~0: BM1M1~0, OBM1 工作模式选择

BM1M1~0	OBM1模式选择
00	BM1SW由BM1E0 与 BM1E1 使能信号控制 (电平切换)
01	BM1SW 由BM1SR控制。BM1SR可以由软件更新, 也可以通过BM1E0事件或BM1E1事件的下降沿进行清除。
10	BM1SW由BM1SR控制。BM1SR可以通过软件进行更新, 也可以通过BM1E0事件或BM1E1事件的下降沿进行置位。
11	BM1SW由BM1SR控制。BM1SR可以通过软件进行更新, 也可以通过BM1E0事件或BM1E1事件的下降沿进行切换。

**BM1SE: 输出信号中止&调制器1切换源使能寄存器**

SFR 页 = 仅 7 页

SFR 地址 = 0xBC

复位值 = 0000-0000

7		6	5	4	3	2	1	0
BM1E1.3		BM1E1.2	BM1E1.1	BM1E1.0	BM1E0.3	BM1E0.2	BM1E0.1	BM1E0.0
R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~4: OBM1使能1控制位3~0.

BM1E1[3:0]	OBM1控制选择
0000	禁止
0001	INT1ET
0010	KBIET
0011	PWM3O
0100	T1CKO
0101	T2CKO
0110	T2EXI
0111	PWM4O
1000	BOD1ET
1001	INT3ET
1010	S1DO
1011	AC1OUT
1100	PWM6O
1101	T4CKO
1110	T4EXI
其它	禁止

Bit 3~0: OBM1使能0控制位3~0.

BM1E0[3:0]	OBM1控制选择
0000	禁止
0001	INT0ET
0010	KBIET
0011	PWM2O
0100	T0CKO
0101	S0DO
0110	AC0OUT
0111	PWM5O
1000	INT2ET
1001	RTCKO
1010	T3CKO
1011	T3EXI
1100	--
1101	AC2OUT
其它	禁止

**BM1LCR: 输出信号中止&调制器1低通道寄存器**

SFR 页 = 仅 8 页

SFR 地址 = 0xBC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
BM1LCP	0	0	0	BM1LS.3	BM1LS.2	BM1LS.1	BM1LS.0
R/W	W	W	W	R/W	R/W	R/W	R/W

Bit 7: BM1LCP, OBM1低通道输出反相控制位

0: BM1LCO 输出信号不反相

1: BM1LCO 输出信号反相

Bit 6~4: 保留位。软件在写入BM1LCR时，必须向这些位写入“0”。

Bit 3~0: BM1LS, OBM1低通道源选择位3~0.

BM1LS[3:0]	OBM1低通道源选择
0 0 0 0	VSS
0 0 0 1	KBI2
0 0 1 0	KBI4
0 0 1 1	T0CKO
0 1 0 0	INT0ET
0 1 0 1	PWM0O
0 1 1 0	PWM1O
0 1 1 1	PWM4O
1 0 0 0	T2CKO
1 0 0 1	T2EXI
1 0 1 0	INT2ET
1 0 1 1	RTCKO
1 1 0 0	PWM6O
其它	保留

**BM1HCR: 输出信号中止&调制器1高通道寄存器**

SFR 页 = 仅 9 页

SFR 地址 = 0xBC

复位值 = 0000-0000

7	6	5	4	3	2	1	0
BM1HCP	0	0	0	BM1HS.3	BM1HS.2	BM1HS.1	BM1HS.0
R/W	W	W	W	R/W	R/W	R/W	R/W

Bit 7: BM1HCP, OBM1高通道输出反相控制位

0: BM1HCO输出信号不反相

1: BM1HCO输出信号反相

Bit 6~4: 保留位。软件在写入BM1HCR时，必须向这些位写入"0"。

Bit 3~0: BM1HS, OBM1高通道源选择位3~0.

BM1HS[3:0]	OBM1高通道源选择
0 0 0 0	VSS
0 0 0 1	KBI3
0 0 1 0	KBI5
0 0 1 1	S0CKO
0 1 0 0	INT1ET
0 1 0 1	PWM2O
0 1 1 0	PWM3O
0 1 1 1	PWM5O
1 0 0 0	T3CKO
1 0 0 1	T3EXI
1 0 1 0	INT3ET
1 0 1 1	PWM7O
1 1 0 0	T4CKO
1 1 0 1	T4EXI
其它	保留

**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
P60OC1	P60OC0	P60FD	PBKF	BM1F	BM0F	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 2: BM1F, OBM1标志位。此位由OBM1切换源使能 (BM1SE) 置位。

0: 没有OBM1切换事件发生。必须软件清零。

1: OBM1切换事件发生。

**AUXR7: 辅助寄存器 7**

SFR 页 = 仅 4 页

SFR 地址 = 0xA4

复位值 = 1100-0000

7	6	5	4	3	2	1	0
POE5	POE4	COCKOE	SPI0M0	BM1OE1	BM1OE0	BM0OE1	BM0OE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 3~2: BM1OE1~0, BM1OUT输出端口引脚选择

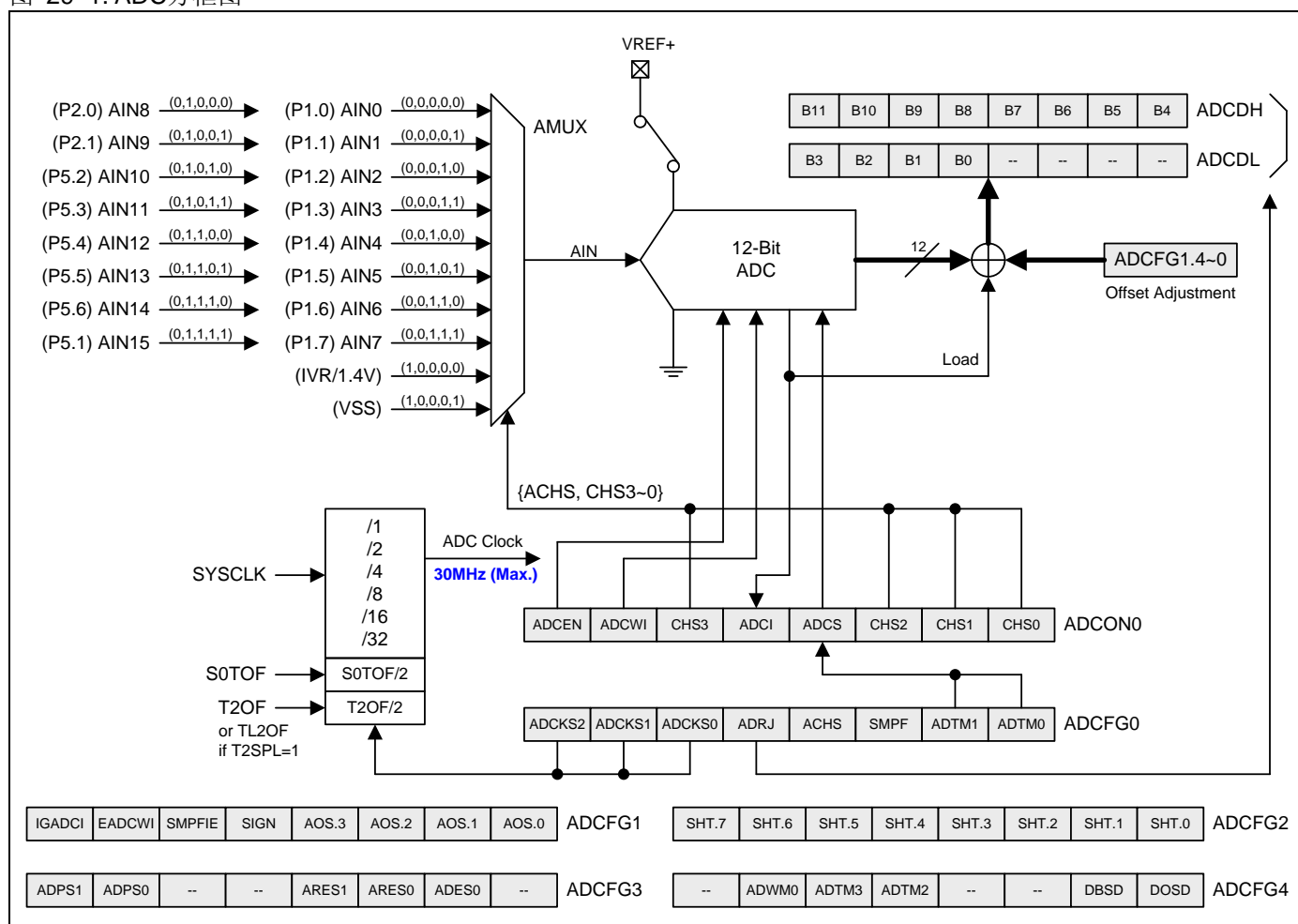
BM1OE1~0	BM1OUT端口引脚
00	禁止
01	P4.3
10	P3.1
11	P7.2

## 29. 12 位 ADC

MGEQ1C064的ADC子系统由一个模拟多路器(AMUX)和一个1.0AECM sps、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通道,如图 29-1。ADC运行在单一节点模式,并且可以配置测量AIN0~AIN15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子系统低功耗关闭。

### 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**1.0Msps**。ADC转换时钟由ADCFG0寄存器的ADCKS2~0位决定是系统时钟分频、S0BRG溢出或者定时器2的溢出。ADC转换时钟不能超过**36MHz**。

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：

$$\text{ADC Result} = \frac{V_{\text{IN}} \times 4096}{V_{\text{REF+ Voltage}}}$$

### 29.2.1. ADC 输入通道

模拟多路器(AMUX)选择输入给ADC，允许任何一个AIN7~0引脚成为被测量的单节点模式和一个内部电压参照(IVR, 1.4V)。通过ADCON0寄存器的**CHS3~0**位和ADCFG0寄存器的ACHS位选择进入ADC测量的通道(见图 29-1)。对被选择的引脚测量的是对地(GND)电压。

### 29.2.2. ADC 内部电压参考

默认的ADC参考电压是VDD。如果VDD不是固定在某个电压，那么使用以下步骤读取电压：

- 1) 将模拟多路复用器(AMUX)设置为IVR。
- 2) 通过ADC转换和存储IVR值。(提示：不同的VDD电压会得到不同的IVR回读值，但是IVR固定在1.4V。因此，这个读回值可以作为参考值。)
- 3) 使用IVR读取返回参考值来计算VDD值。现在VDD得到了一定的值，可以作为参考电压。
- 4) 使用参考电压转换输入电压。

### 29.2.3. 开启一次转换

在使用ADC功能之前，用户应：

- 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) 通过位CHS3、CHS2、CHS1和CHS0选择模拟输入通道。
- 4) 将所选引脚配置成仅模拟输入模式。
- 5) 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ADCDL。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。

### 29.2.4. 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。ADC的最大输入时钟是**36MHz**并且操作在最少**30**个ADC转换时钟的转换时间。用户可以通过ADCKS2~0 (ADCFG0.7~5)和SHT (ADCFG2.7~0)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1.0M的采样率:  
若SYSCLK =36MHz并且ADCKS = SYSCLK, SHT = 6  
这样转换速率= 36MHz / (30+6) =1.0Msps。  
(此案例, 交流 (AC)输入信号频率 $f_N$ 要低于500KHz确保测量的精度。)
- 2) 为了得到150K Hz的采样率:  
若SYSCLK =24MHz并且ADCKS = SYSCLK/4, SHT = 10  
这样转换速率= 24MHz/4/(30+10) =150K 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。)

### 29.2.5. ADC 中断

**MGEQ1C064**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

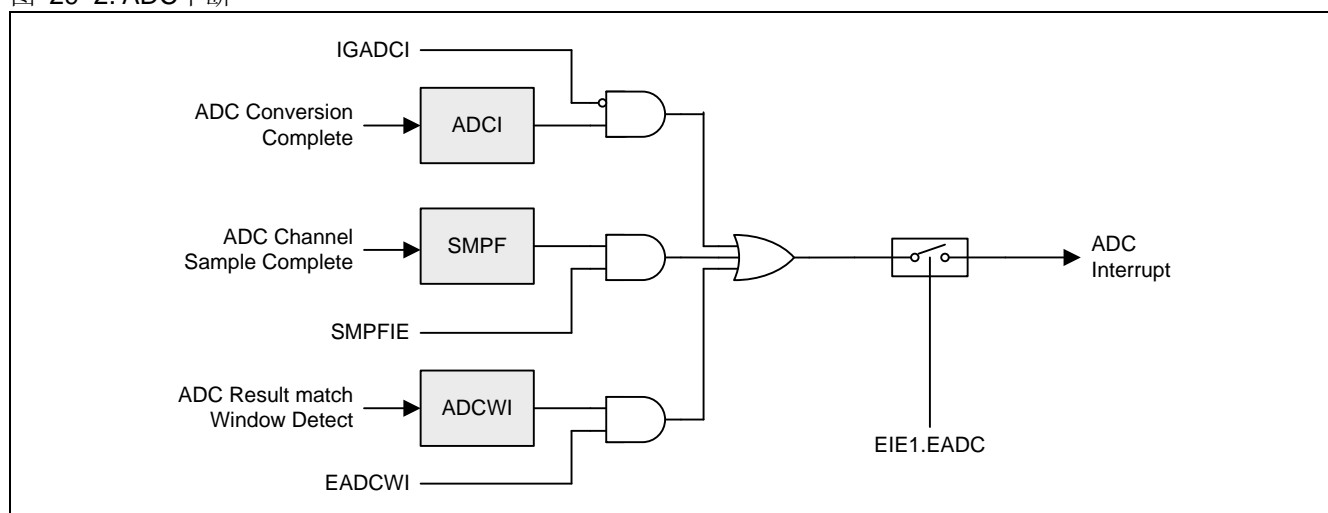
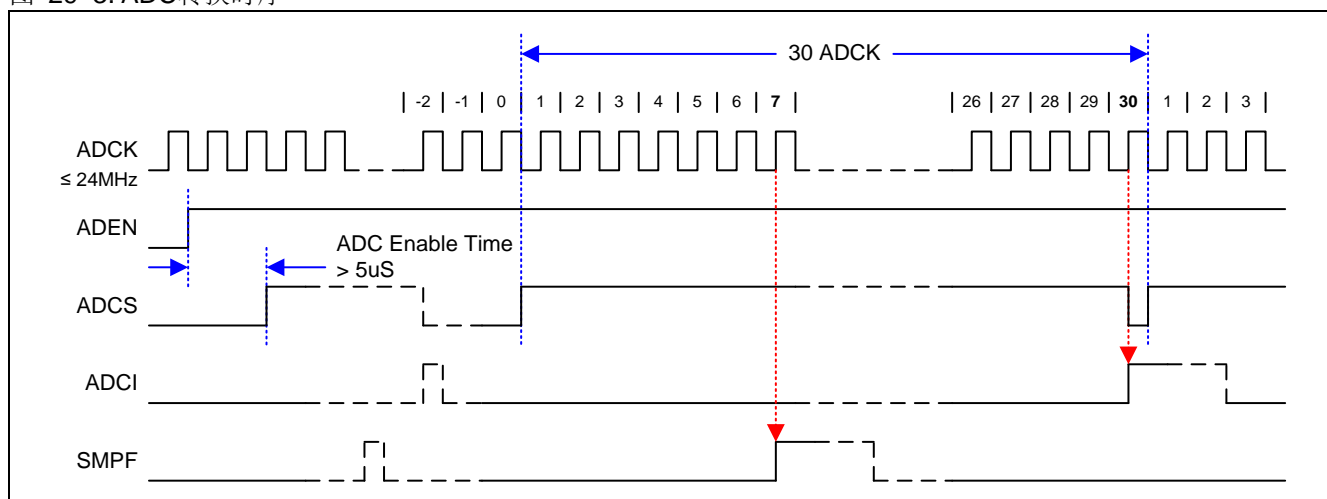


图 29-3. ADC转换时序

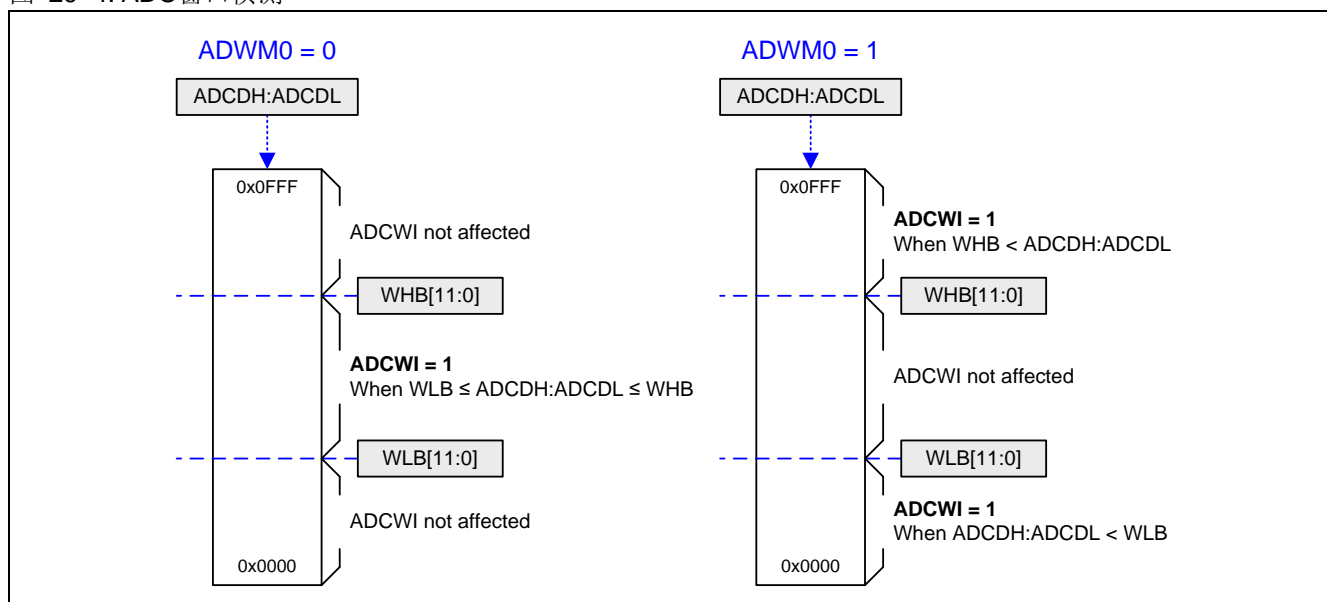


### 29.2.6. ADC 窗口侦测

**MGEQ1C064** 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窗口侦测



ADC窗口侦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  $\geq$  条件电压：  $ADWM0 = 0$ ，设置条件值在WLB和设置WHB = 0xFFFF。
- 2) 目标电压  $\leq$  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。

### 29.2.7. ADC 通道扫描模式

**MGEQ1C064**有16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通道扫描模式容易完成和保存通道切换时间。设置ADCFG5和ADCFG6的 ASCE.15 ~ 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通道。清零ASCE.15 ~ ASCE.0停止此模式禁止此功能。当ADC通道扫描模式使能，请不要手动写CHS3~0而改变通道，否则将会选中意外的通道。如果清零ADCWI (ADC窗口比较中断标志)，需要读修改写ADCON0来保护CHS3 ~ CHS0不被改变。请注意，当使用此模式，ACHS需要写“0”，保护被选中的内部ADC通道。

- 1) 置位ADCEN，打开ADC硬件。
- 2) 位ADCKS2、ADCKS1和ADCKS0配置ADC输入时钟。
- 3) 配置采样通道为仅模拟输入模式。
- 4) 用位ADRJ配置ADC结果分布。
- 5) 设置ADCFG5的位ASCE.7 ~ ASCE.0选择模拟输入通道。
- 6) 设置ADCFG6的位ASCE.15 ~ ASCE.8选择模拟输入通道。
- 7) 设置ADTM[3:0]选择ADC触发模式。

### 29.2.8. 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ADC DL然后ADC DH。
- 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控制器自动侦测传输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0xFFF，DMA传输不同的数据组合如下：

ADRJ	ARES[1:0] ADC数据分辨率	DBSD DMA传输ADC数据字节	数据传输次序	
			第1个数据	第2个数据
0	00 (12位)	0 (2-字节数据)	0xF0	0xFF
		1 (1-字节数据)	0xFF	X
	01 (10位)	0 (2-字节数据)	0xC0	0xFF
		1 (1-字节数据)	0xFF	X
	1x (8位)	0 (2-字节数据)	0x00	0xFF
		1 (1-字节数据)	0xFF	X
1	00 (12位)	0 (2-字节数据)	0xFF	0x0F
		1 (1-字节数据)	0xFF	X
	01 (10位)	0 (2-字节数据)	0xFF	0x03
		1 (1-字节数据)	0xFF	X
	1x (8位)	0 (2-字节数据)	0xFF	0x00
		1 (1-字节数据)	0xFF	X

### 29.2.9. I/O 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ADC I7~0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. 通用端口配置设定”所示。关于AIN引脚设置请参考章节“14.2 I/O端口寄存器”。

### 29.2.10. 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。

### 29.2.11. 如何提高 ADC 精度

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。MGEQ1C064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)

$$IVR \text{ Voltage} = \frac{IVR_{ADC\_PreStored\_Value} * 3300}{4096} (mV) \dots \dots \dots (1)$$

- 使用比例关系计算I/O引脚的电压

$$AIN \text{ Voltage} = \frac{IVR \text{ Voltage} * AIN_{ADC\_Value}}{IVR_{ADC\_Value}} (mV) \dots \dots \dots (2)$$

注: 请参考章节“31.3如何读取IVR (1.4V) ADC预设值”来读取IVR ADC预设值。

## 29.3. ADC 寄存器

### ADCON0: ADC控制寄存器0

SFR 页 =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 ~ CHS1, ADC模拟多路器输入通道选择位。

单节点模式::

ACHS	CHS3~0	通道选择
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)
0	1 0 0 0	AIN8 (P2.0)
0	1 0 0 1	AIN9 (P2.1)
0	1 0 1 0	AIN10 (P5.2)
0	1 0 1 1	AIN11 (P5.3)
0	1 1 0 0	AIN12 (P5.4)
0	1 1 0 1	AIN13 (P5.5)
0	1 1 1 0	AIN14 (P5.6)
0	1 1 1 1	AIN15 (P5.1)
1	0 0 0 0	Int. VREF (IVR/1.4V)
1	0 0 0 1	AVSS
	其它	保留

#### ADCFG0: ADC配置寄存器0

SFR 页 = 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

**ADC DL: 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

**ADC DL**

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 (单节点模式)	ADCDH:ADC DL (ADRJ = 0)	ADCDH:ADC DL (ADRJ = 1)
$VDD \times 4095/4096$	0xFFF0	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转换启动选择
0 0	ADCS置位
0 1	定时器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。这个寄存器的值将校正保存在{ADCDH, ADCDL}上的ADC转换结果，用来消除偏移量。软件通过设置ADCON0.AZEN可以动态收集ADC的偏移值，并且用这个值更新AD0ROC，用于ADC转换结果的自动修正。软件也可以将这个值存入到MGEQ1C064的IAP区域，用它作为一个ADC偏移量校正的常规参数。下表列举了ADC转换结果的AD0ROC校正值。

{Sign, AOS.[3:0]}	{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 = 0100-0000

7	6	5	4	3	2	1	0
ADPS1	ADPS0	0	0	ARES1	ARES0	ADES0	0
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3~2。

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: 保留位。软件在写入ADCFG3时，必须向这些位写入” 0”。

Bit 3~2: ARES1~0, AD数据长度选择位1~0.

ARES[1:0]	ADC 数据长度选择
0 0	12-位数据
0 1	10-位数据
1 0	8-位数据
1 1	保留(8-位数据)

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 0: 保留位。软件在写入ADCFG3时，必须向这些位写入” 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
0	ADWM0	ADTM3	ADTM2	0	0	DBSD	DOSD
W	R/W	R/W	R/W	W	W	R/W	R/W

Bit 7: 保留位。软件在写入ADCFG4时，必须向这些位写入”0”。

Bit 6: ADWM0, ADC窗口侦测器的模式选择。

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 T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	ADCS置位	软件
0 0 0 1	定时器0溢出(T0OF)	定时器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	定时器2
1 0 0 1	AC0ES	AC0
1 0 1 0	T3EXES	定时器3
1 0 1 1	AC1ES	AC1
1 1 0 0	PCA0 溢出(C0TOF)	PCA0计数器
1 1 0 1	C0CMP6 <sup>(注1)</sup>	PCA0 CH6 比较
1 1 1 0	C0CMP6 or C0CMP7 <sup>(注1)</sup>	PCA0 CH6/CH7 比较
1 1 1 1	BM0FES	OBM0

注1: C0CMPx: 参考 图 17-7 & 图 17-8

Bit 3~2: 保留位。软件在写入ADCFG4时，必须向这些位写入”0”。

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。

**ADCFG6: ADC配置寄存器6**

SFR 页 = 仅 6 页

SFR 地址 = 0xC3

复位值 = 0000-0000

7	6	5	4	3	2	1	0
ASCE.15	ASCE.14	ASCE.13	ASCE.12	ASCE.11	ASCE.10	ASCE.9	ASCE.8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: AIN15~AIN8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。

**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 = 仅 D 页

SFR 地址 = 0xC3

复位值 = 1111-1111

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: ADC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	0	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 (1.4V)。

1: 使能片内IVR (1.4V)。

Bit 6~5: 保留位。软件在写入PCON3时, 必须向这些位写入"0"。

Bit 3~0: 保留位。软件在写入PCON3时, 必须向这些位写入"0"。

## 30. 模拟比较器 0/1/2 (AC0/AC1/AC2)

MGEQ1C064内含三个模拟比较器模块。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。

- VIN+上的输入信号:
  - 对于AC0:有4个I/O输入，可以由AC0PIS[1:0]选择。
  - 对于AC1: 仅有一个I/O输入
  - 对于AC2: 仅有一个I/O输入
- VIN-上有3种参考电压可以使用:
  - VDD通过分压电阻分压成不同的电压，表 30.1 和 表 30.2 给出了详细的设置。
  - 从引脚AC0NI, AC1NI 或 AC2NI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。
  - IVR: 内部电压参考, 1.4V.
  - AC1/AC2 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。
- 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。
- 中断方式选择:AC0、AC1、AC2的中断可通过上升沿、下降沿、双边沿触发。

### 30.1. AC0/AC1/AC2 结构

图 30-1. 模拟比较器0方框图

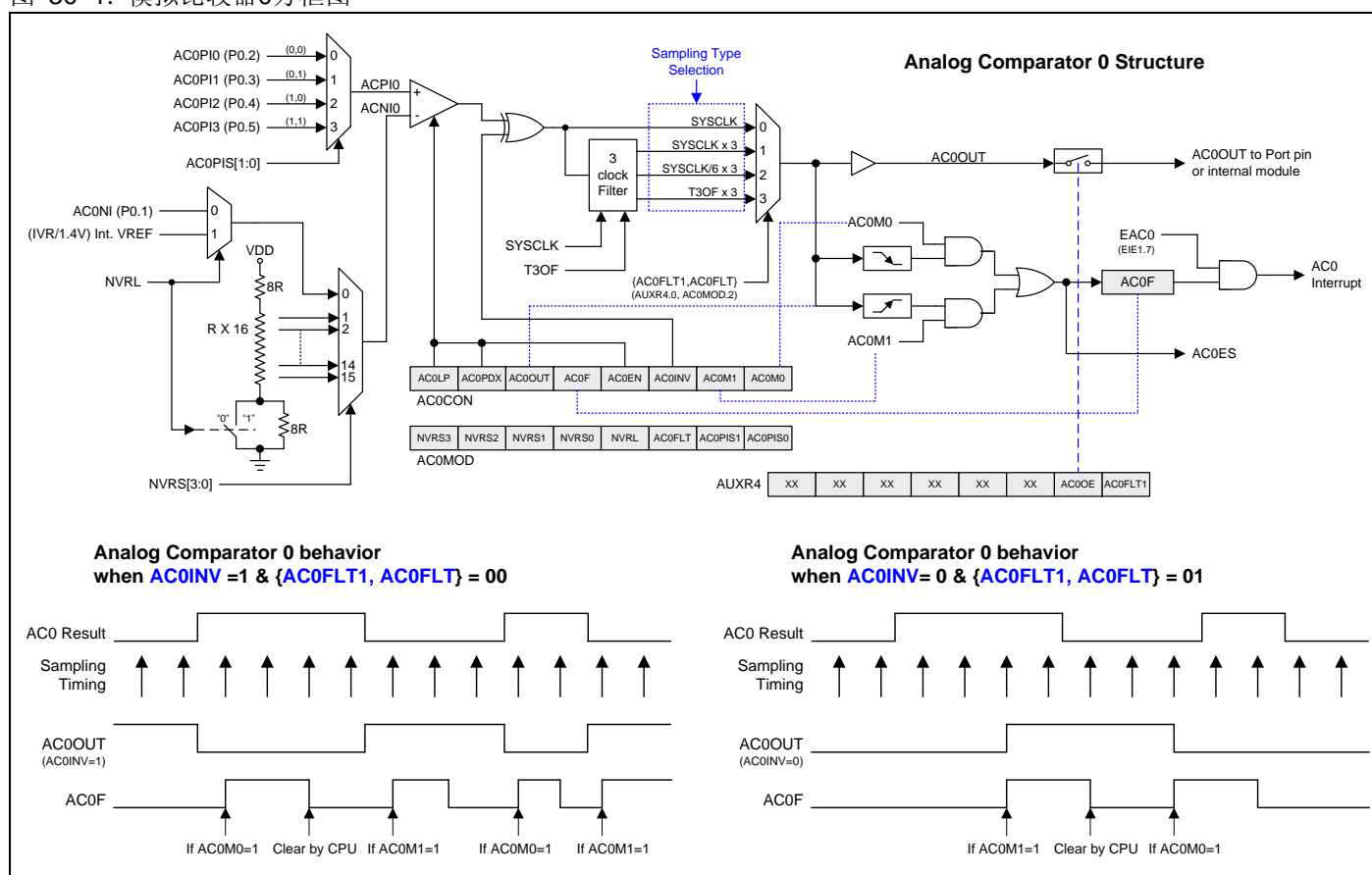


图 30-2. 模拟比较器1方框图

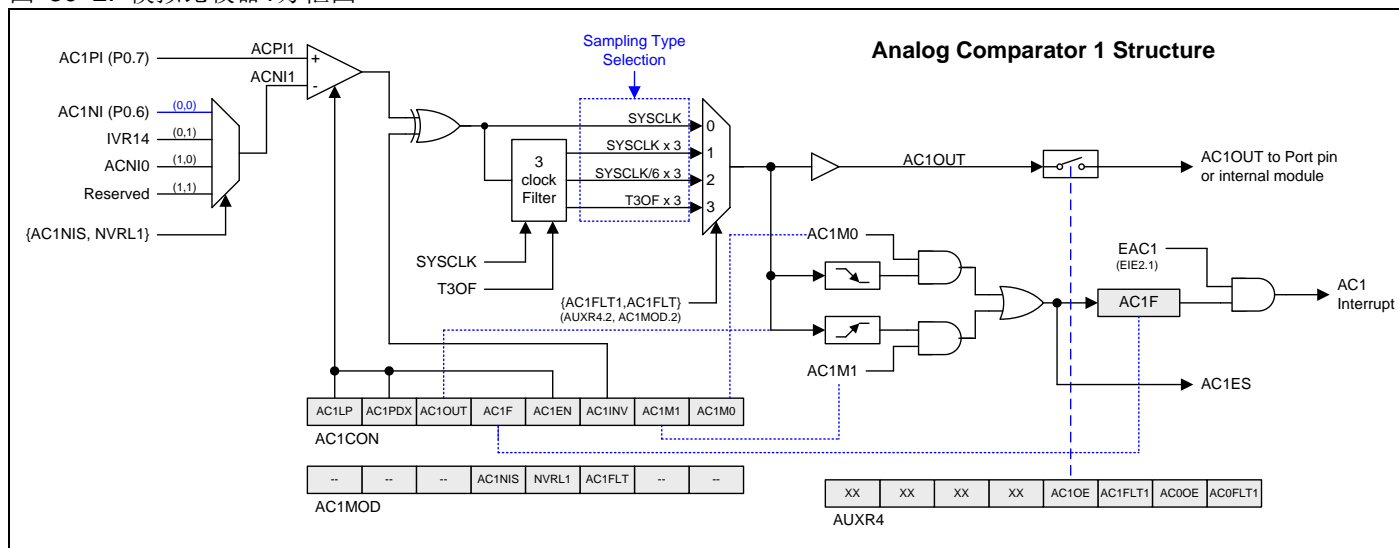
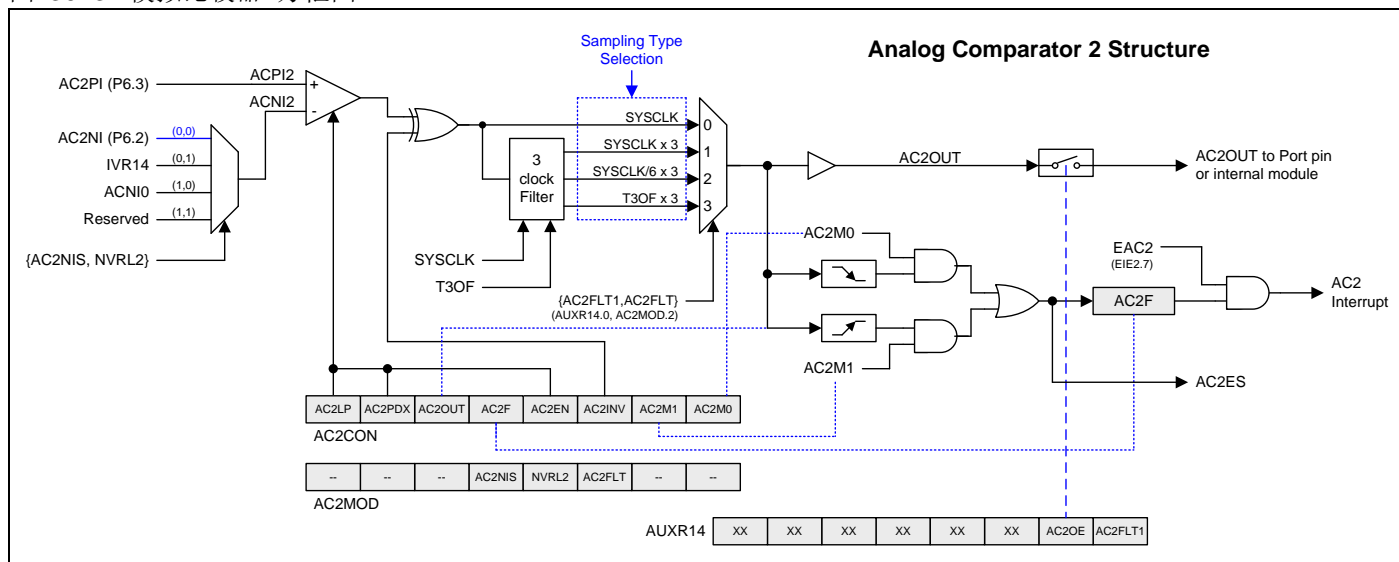


图 30-3. 模拟比较器2方框图



## 30.2. AC0/AC1/AC2 寄存器

### AC0CON: 模拟比较器 0 控制 & 状态寄存器

SFR 页 = 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中断模式

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	AC0FLT	AC0PIS1	AC0PIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~5: 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 NVRL = 0, 选择高区域

NVRS[3:0]	(V-) 输入	NVRS[3:0]	(V-) 输入
0000	AC0NI(P0.1)	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	23/32 VDD

表 30.2. NVRL = 1, 选择低区域

NVRS[3:0]	(V-)输入	NVRS[3:0]	(V-)输入
0000	<b>Int. VREF (1.4V)</b>	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	15/24 VDD

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: AC0FLT, 模拟比较器 0输出滤波控制。和AC0FLT1 (AUXR4.0)一起选择AC0OUT滤波模式。

AC0FLT1, AC0FLT	AC0OUT滤波模式
0 0	禁止
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	<b>T30F x 3</b>

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(P0.2)
0 1	AC0PI1(P0.3)
1 0	AC0PI2(P0.4)
1 1	AC0PI3(P0.5)

**AC1CON: 模拟比较器 1 控制&状态寄存器**

SFR 页 = 仅 1 页

SFR 地址 = 0x9E

复位值 = 00X0-0000

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 = xxx0-00xx

7	6	5	4	3	2	1	0
--	--	--	AC1NIS	NVRL1	AC1FLT	--	--
W	W	W	R/W	R/W	R/W	W	W

Bit 7~5: 保留位。软件在写入AC1MOD时, 必须向这些位写入"0"。

Bit 1~0: 保留位。软件在写入AC1MOD时, 必须向这些位写入"0"。

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	AC1NI 端口引脚	P0.6
0 1	IVR14	内部参考电压 1.4V
1 0	ACNI0	模拟比较器 0 负端输入
1 1	保留	--

Bit 2: AC1FLT, 模拟比较器 1 输出滤波控制。和AC1FLT1 (AUXR4.2)一起选择AC1OUT滤波模式。

AC1FLT1, AC1FLT	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

**AC2CON: 模拟比较器 2 控制&状态寄存器**

SFR 页 = 仅 2 页

SFR 地址 = 0x9E

复位值 = 00X0-0000

7	6	5	4	3	2	1	0
AC2LP	AC2PDX	AC2OUT	AC2F	AC2EN	AC2INV	AC2M1	AC2M0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit 7: AC2LP, 模拟比较器 2低功耗使能

0:禁止AC2 低功耗模式

1:使能 AC2 低功耗模式

Bit 6: AC2PDX, PD模式下模拟比较器2控制

0: PD模式下模拟比较器2关闭

1: PD模式下模拟比较器2继续工作

如果AC2EN, AC2PDX, EAC2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: AC2OUT, 这是比较器输出的只读位。

AC2 输入	AC2INV = 0	AC2INV = 1
ACPI2(+) > ACNI2(-)	AC2OUT = 1	AC2OUT = 0
ACPI2(+) < ACNI2(-)	AC2OUT = 0	AC2OUT = 1

Bit 4: AC2F. 模拟比较器 2 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AC2M[1:0]位所指定的条件并且AC2EN置位时置位。可以通过设置/清除EIE2的第7位来启用/禁用中断。

Bit 3: AC2EN. 模拟比较器 2 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AC2F。

1: 此位置位使能比较器。

Bit 2: AC2INV, 模拟比较器 2 输出反相使能位

0: AC2输出不反相

1: AC2输出反相

Bit 1~0: AC2M[1:0], 模拟比较器 2中断模式

AC2M[1:0]	AC2中断模式
0 0	保留
0 1	比较器2侦测输出下降沿
1 0	比较器2侦测输出上升沿
1 1	比较器2侦测输出突变

**AC2MOD: 模拟比较器 2 模式寄存器**

SFR 页 = 仅 2 页

SFR 地址 = 0x9F

复位值 = xxx0-00xx

7	6	5	4	3	2	1	0
--	--	--	AC2NIS	NVRL2	AC2FLT	--	--
W	W	W	R/W	R/W	R/W	W	W

Bit 7~5: 保留位。软件在写入AC2MOD时, 必须向这些位写入"0"。

Bit 1~0: 保留位。软件在写入AC2MOD时, 必须向这些位写入"0"。

Bit 4: AC2NIS &amp; NVRL2, 模拟比较器 2负端输入选择

AC1NIS, NVRL1	ACNI2 选择	输入源描述
0 0	AC2NI端口引脚	P6.2
0 1	IVR14	内部参考电压1.4V
1 0	ACNI0	模拟比较器 0负端输入
1 1	保留	--

Bit 2: AC2FLT, 模拟比较器 2 输出滤波控制。和AC2FLT1一起选择AC2OUT滤波模式。

AC2FLT1, AC2FLT	AC2OUT滤波模式
0 0	禁止
0 1	SYSCLK x 3
1 0	SYSCLK/6 x 3
1 1	T3OF x 3

#### AUXR10: 辅助寄存器 10

SFR 页 = 仅 7 页

SFR 地址 = 0xA4

复位值 = 1100-0000

7	6	5	4	3	2	1	0
AC1HC0	AC0HC0	SPIPS1	SPIPS0	S0PS1	SPFACE	TWICF	PAA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: AC1HC0, AC1滞环控制0.

0: 禁止在AC1上滞环输入

1: 使能在AC1上滞环输入。默认是使能

Bit 6: AC0HC0, AC0 滞环控制 0.

0: 禁止在AC0上滞环输入

1: 使能在AC0上滞环输入。默认是使能

#### 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 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P7.2上

Bit 2: AC1FLT1, AC1输出滤波控制1.

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P0.0上

Bit 0: AC0FLT1, AC0输出滤波控制1.

#### AUXR14: 辅助寄存器 14

SFR 页 = 仅 B 页

SFR 地址 = 0xA4

复位值 = 0000-0100

7	6	5	4	3	2	1	0
ADLPS1	ADLPS0	0	0	RWPS0	AC2HC0	AC2OE	AC2FLT1
R/W	R/W	W	W	R/W	R/W	R/W	R/W

Bit 2: AC2HC0, AC2 滞环控制 0.

0: 禁止在AC2上滞环输入

1: 使能在AC2上滞环输入。默认是使能

Bit 1: AC2OE, 使能AC2OUT输出在端口引脚上。

0: 禁止AC2OUT 输出在端口引脚上。

1: 使能AC2OUT输出在P6.4上

Bit 0: AC0FLT1, AC0输出滤波控制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	0	0	SPWRE	0	0	0	0
R/W	W	W	R/W	W	W	W	W

Bit 7: IVREN, 内部电压参考使能

0: 禁止内部 IVR (1.4V).

1: 使能内部IVR (1.4V).

Bit 6~5: 保留位。软件在写入PCON3时, 必须向这些位写入” 0”。

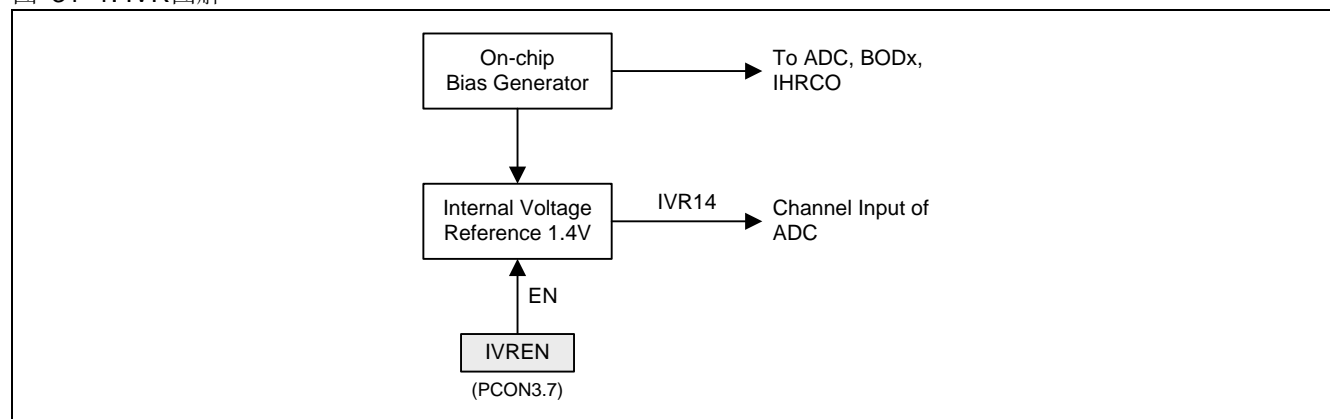
Bit 3~0: 保留位。软件在写入PCON3时, 必须向这些位写入” 0”。

## 31. 内部参考电压 (IVR, 1.4V)

可以将IVR作为AC0和ADC的参考电压。它典型的输出是1.4V，可以由IVREN禁用。

### 31.1. IVR (1.4V)结构

图 31-1. IVR图解



### 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	0	0	SPWRE	0	0	0	0
R/W	W	W	R/W	W	W	W	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 6~5: 保留位。软件在写入PCON3时，必须向这些位写入” 0”。

Bit 3~0: 保留位。软件在写入PCON3时，必须向这些位写入” 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“[29.2.11 如何提高ADC精准度](#)”来了解如何提高ADC的测量精度。

```
void Get_Prestored_IVR(void)
{
    ISPCR = ISP_ENABLE;
    BOREV = 0x22;
    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;
}
```

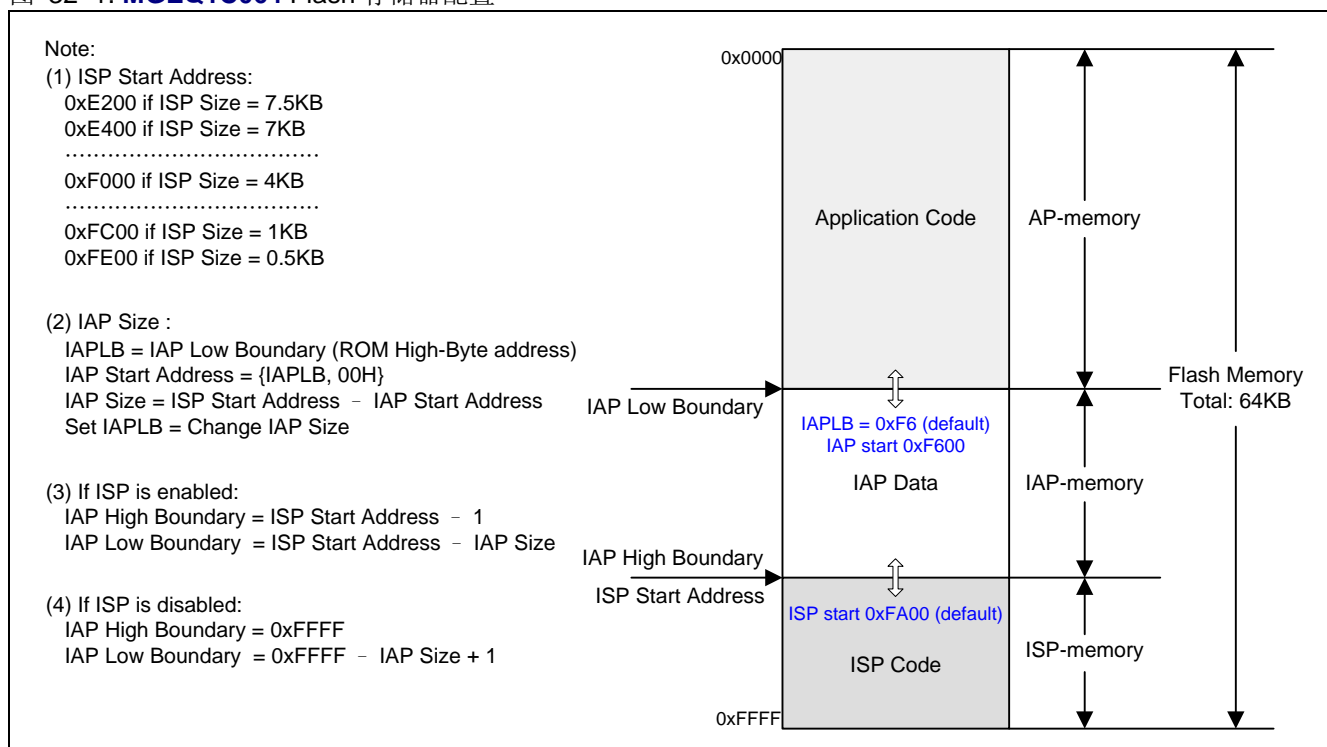
## 32. ISP 和 IAP

MGEQ1C064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. MGEQ1C064 Flash 存储器配置

MGEQ1C064总共有64K字节的Flash，MGEQ1C064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64K字节的存储空间。

图 32-1. MGEQ1C064 Flash 存储器配置



注意:

笙泉公司MGEQ1C064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。1.0K IAP大小可以通过应用程序软件来重新配置。

## 32.2. MGEQ1C064 Flash 在 SP/IAP 上的访问

**MGEQ1C064**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

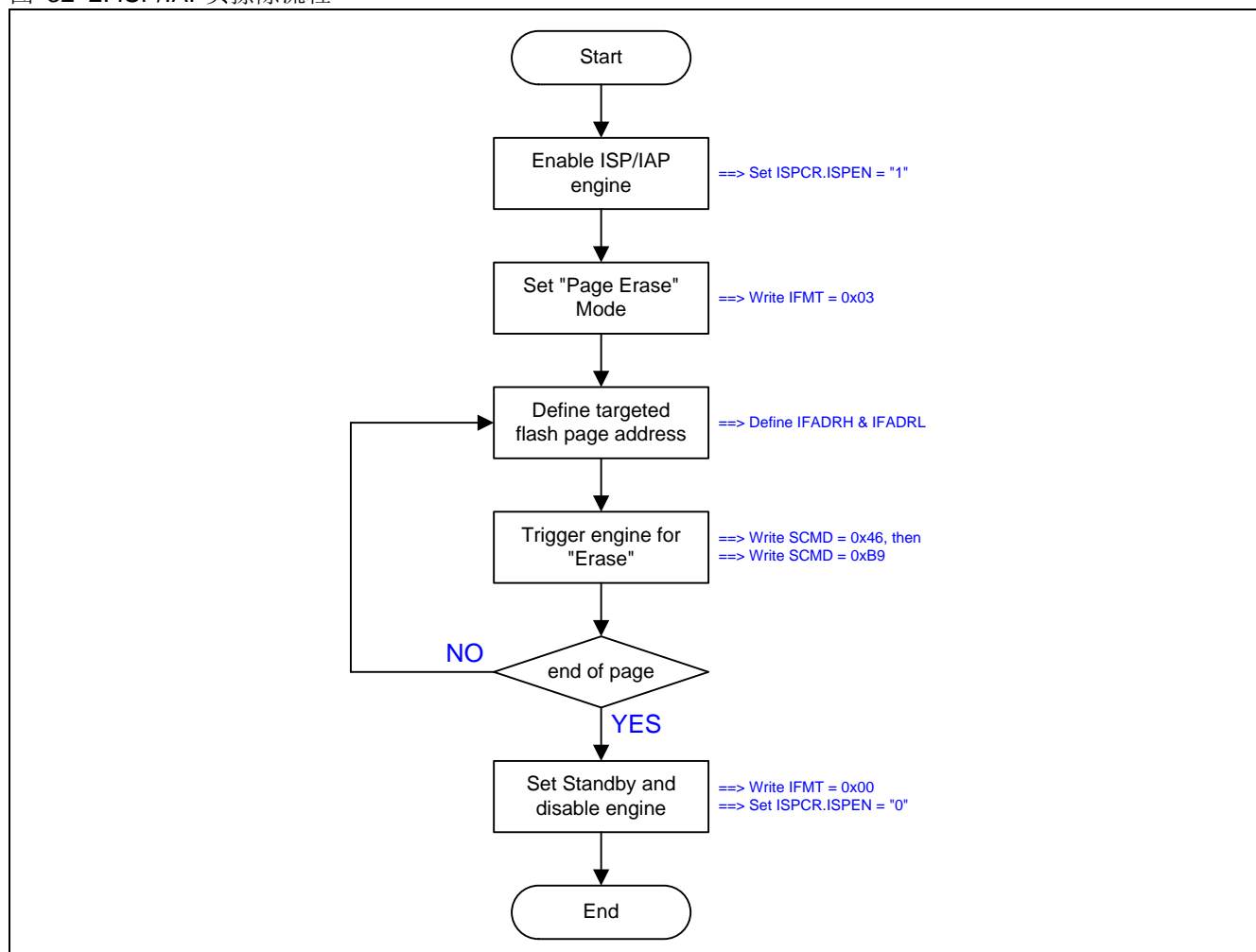
- 步骤1：在ISPCR 寄存器上置位ISPEN使能ISP/IAP流程。
- 步骤2：在IFMT寄存器上设置MS=0x01选择字节读取模式。
- 步骤3：填入字节地址到IFADRH和IFADRL寄存器。
- 步骤4：顺序地在SCMD 寄存器写入0x46h 然后0xB9h触发一个ISP处理。
- 步骤5：现在,Flash 数据在IFD 寄存器上。
- 步骤6：清零ISPEN和MS=0x00关闭ISP/IAP流程

**MGEQ1C064**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

MGEQ1C064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MGEQ1C064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。MGEQ1C064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, 使能ISP
MOV    IFMT,#03h       ; 选择页擦除模式
MOV    IFADRH,??       ; 页地址填到[IFADRH,IFADRL]
MOV    IFADRL,??       ;

MOV    SCMD,#46h       ; 触发ISP/IAP 处理
MOV    SCMD,#0B9h      ;

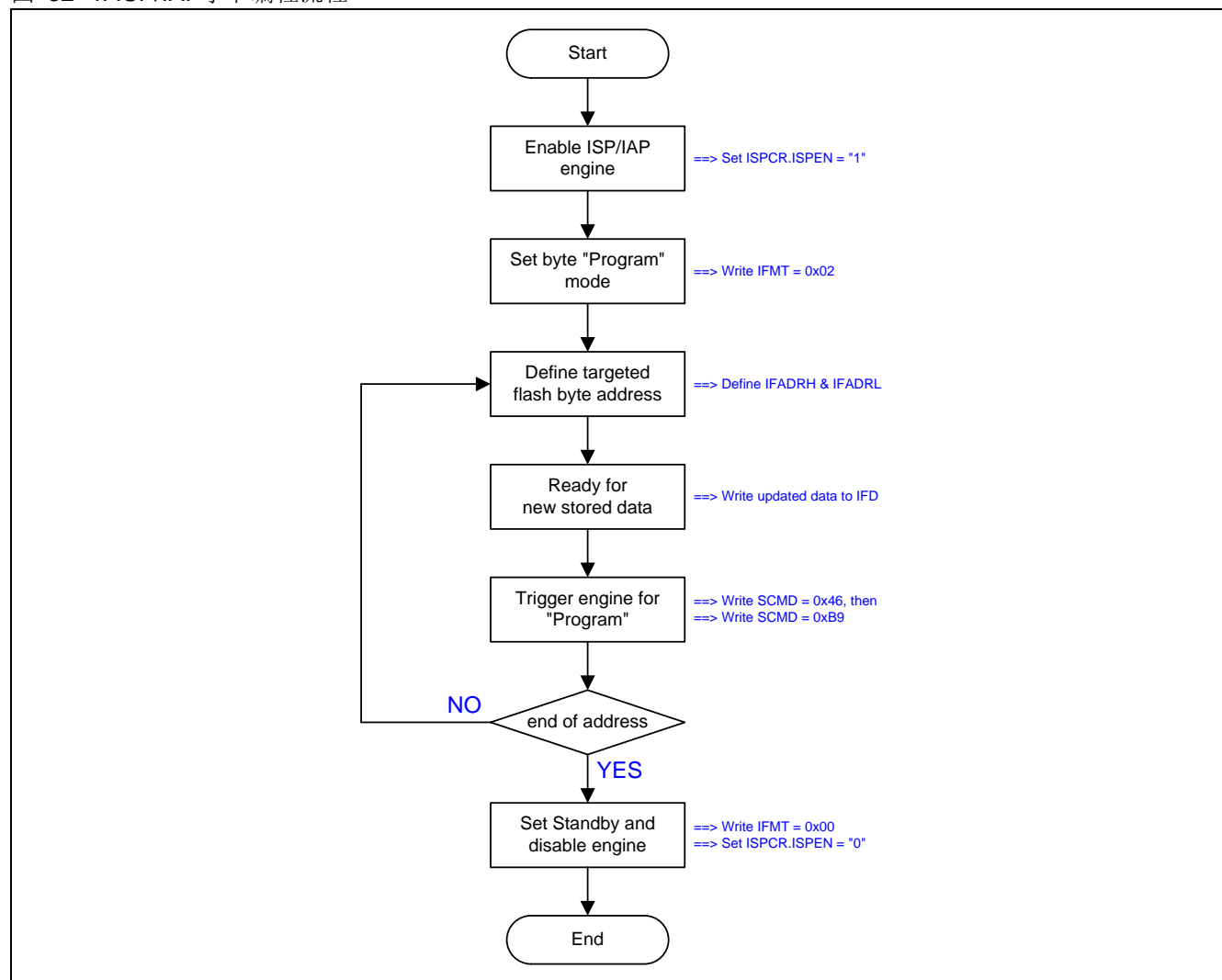
; 现在, MCU将停在这直ISP/IAP处理完成

MOV    IFMT,#00h       ;选择待命模式
MOV    ISPCR,#00000000b ;ISPCR.7= 0, 禁止ISP
  
```

### 32.2.2. ISP/IAP Flash 字节编程模式

MGEQ1C064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所示。

图 32-5. ISP/IAP字节编程的示例代码

```

MOV    ISPCR,#10000011b ;ISPCR.7=1, 使能ISP

MOV    IFMT,#02h      ; 选择字节编程模式

MOV    IFADRH,??      ; 字节地址填到[IFADRH,IFADRL]
MOV    IFADRL,??      ;
MOV    IFD,??         ; 编程数据填到IFD

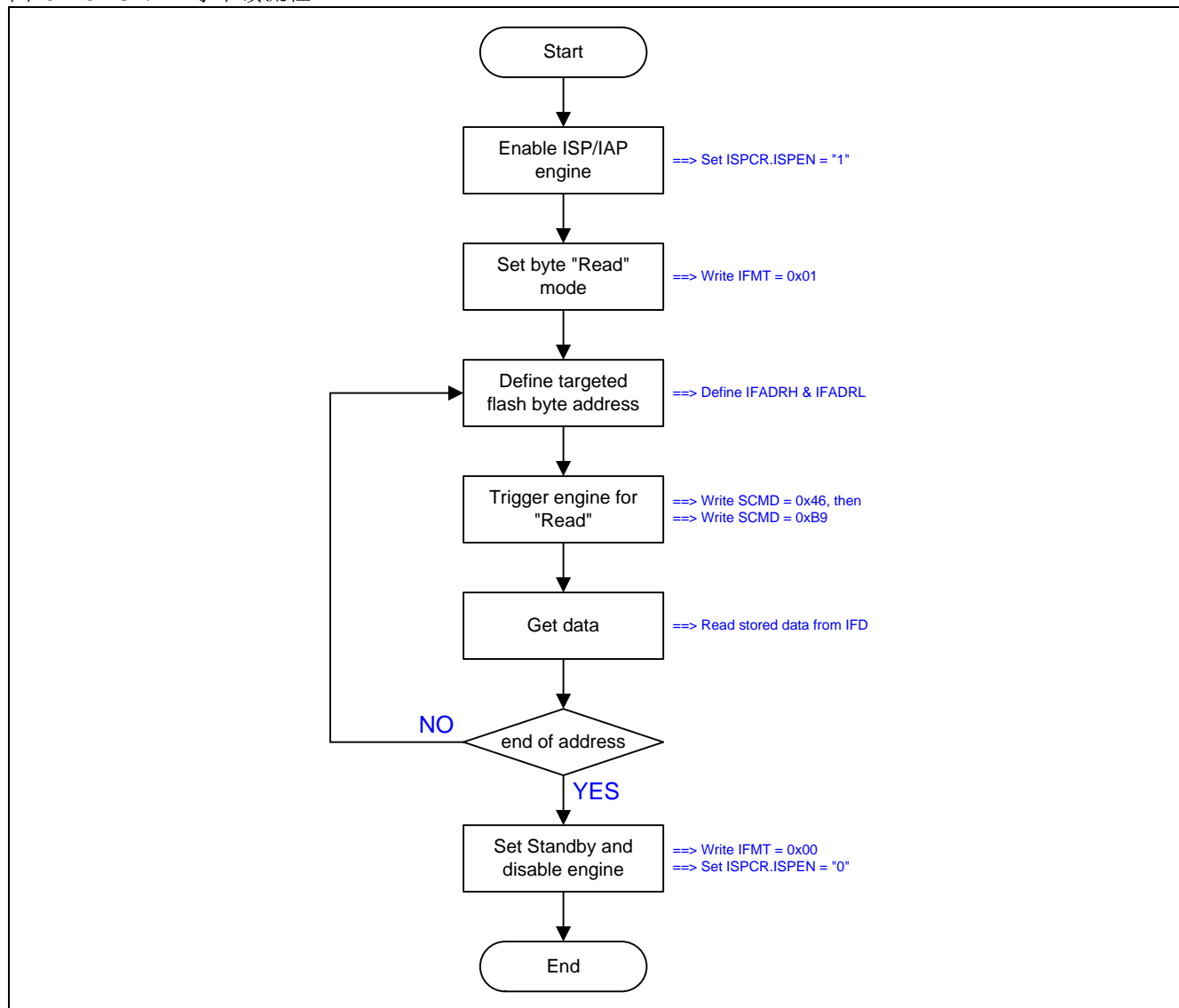
MOV    SCMD,#46h      ; 触发ISP/IAP处理
MOV    SCMD,#0B9h     ;
; 现在, MCU将停在这直ISP/IAP处理完成

MOV    IFMT,#00h      ; 选择待命模式
MOV    ISPCR,#00000000b ;ISPCR.7= 0, 禁止ISP
  
```

### 32.2.3. ISP/IAP Flash 读模式

MGEQ1C064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字节读流程



ISP/IAP字节读取操作的范例代码如图 32-7所示。

图 32-7. ISP/IAP字节读取的范例代码

```

MOV    ISPCR,#10000011b ;ISPCR.7=1, 使能ISP
MOV    IFMT,#01h       ; 选择字节读模式
MOV    IFADRH,??       ; 字节地址填写到[IFADRH,IFADRL]
MOV    IFADRL,??       ;
MOV    SCMD,#46h       ; 触发ISP/IAP处理
MOV    SCMD,#0B9h      ;
; 现在, MCU将停在这直ISP/IAP处理完成

MOV    A,IFD           ; 现在, 数据已经存在IFD里

MOV    IFMT,#00h       ; 选择待命模式
MOV    ISPCR,#00000000b ;ISPCR.7= 0, 禁止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,使用者必须先配置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,MGEQ1C064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**(MGEQ1C064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和空闲模式

MGEQ1C064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)

**MGEQ1C064**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. **MGEQ1C064**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. MGEQ1C064 IAP 存储边界/范围

如果ISP 存储空间被声明, 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 存储空间没有被声明, IAP存储空间范围由下列公式决定:

$$\begin{aligned} \text{IAP 高边界} &= \text{0xFFFF}. \\ \text{IAP 低边界} &= \text{0xFFFF} - \text{IAP} + 1. \end{aligned}$$

例如, 如果ISP存储空间是1K字节, 这样ISP的起始地址是**0xFC00**, 并且IAP存储空间是1K字节, 此时IAP存储空间的范围就在**0xF800 ~ 0xFBFF**。**MGEQ1C064**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寄存器](#)”。

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. IAP 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

**MGEQ1C064**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 = 0xE7

POR = 0000-0000

7	6	5	4	3	2	1	0
ISPEN	SWBS	SRST	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: SRST, 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: 保留位。软件在写入ISPCR时, 必须向这些位写入"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, IFMT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IFMT, 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。

$$IAP低边界 = IAPLB[7:0] \times 256$$

$$IAP高边界 = ISP起始地址 - 1。$$

如果IAPLB=0xC0且ISP起始地址是0xF000, 这样IAP存储区位于0xC000 ~ 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。

## 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,#1000000b ;ISPCR.7=1, enable ISP
;
;=====
; 1. Page Erase Mode (512 字节 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_erro ;compare with the wanted value
;          ...
ISP_error:
;          ...
;

```

### 33. P 页访问

**MGEQ1C064**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, IFMT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IFMT, 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。

$IAP低边界 = IAPLB[7:0] \times 256$

$IAP高边界 = ISP起始地址 - 1$ 。

例如, 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	Gain Define	Applied Crystal
0, 0	Low Gain	32.768KHz
0, 1	Medium Gain	2MHz ~ 25MHz
1, 0	Lower Gain	32.768KHz
1, 1	Reserved	Reserved

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]	
		CKMS0 = 0	CKMS0 = 1	CKMS0 = 0	CKMS0 = 1
0 0	OSCin	12MHz		11.059MHz	
0 1	CKMI x4 / x6	24MHz	36MHz	22.118MHz	33.177MHz
1 0	CKMI x5.33 / x8	32MHz	48MHz	29.491MHz	44.236MHz
1 1	CKMI x8 / x12	48MHz	72MHz	44.236MHz	66.354MHz

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CKKS选择CPUCLK和SYSCLK的正确范围不要超出限制。CPUCLK ≤ 36MHz, 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-7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

**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

**CKCON5: 时钟控制寄存器5**

SFR 页 = 仅 P 页

SFR 地址 = 0x43

复位值 = 0000-0000

7	6	5	4	3	2	1	0
--	--	--	--	--	--	--	CKMS0
W	W	W	W	W	W	W	R/W

Bit 7~1: 保留位。软件在写入CKCON5时，必须向这些位写入”0”。

Bit 0: CKMS0, CKM模式选择0。

0: 选择CKM运行在CKMI X8模式。(96MHz用于PCA)

1: 选择CKM运行在CKMI X1模式。(144MHz用于PCA)

**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: 保留位。软件在写入PCON2时，必须向这些位写入”0”。

Bit 5~4: BO1S[1:0]. 低电压监测器1监测电压选择, 这两个位的初始值从OR1.BO1S1O和OR1.BO1S0O加载

BO1S[1:0]	BOD1监测电压
0 0	2.0V
0 1	2.4V
1 0	3.7V
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。软件在写入PCON2时，必须向这些位写入” 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	0	0	SPWRE	0	0	0	0
R/W	W	W	R/W	W	W	W	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 6~5: 保留位。软件在写入PCON3时，必须向这些位写入” 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: 保留位。软件在写入PCON3时，必须向这些位写入” 0”。

### SPCON0: SFR页面控制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: 保留位。软件在写入SPCON时，必须向这些位写入” 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	W	W	W	W	R/W	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。软件在写入DCON0时, 必须向这些位写入"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
P60OC1	P60OC0	P60FD	PBKF	BM1F	BM0F	INT1H	INT0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~6: 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，可驱动内部RC振荡器输出至P6.0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，请参考章节“9 系统时钟”。P6.0作为时钟输出功能时，建议设置{P6M1.0, P6M0.0}为推挽输出模式。

Bit 5: 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。

Bit 4: PBKF, PWM终止标志。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 终止事件出现。仅由软件清零。

1: PWM 终止事件出现或软件触发一个PWM 终止。

Bit 3: BM1F, OBM1 标志。该位由OBM1切换源使能置位(BM1SE)。

0: 没有OBM1切换事件发生。仅由软件清零。

1: 有OBM1切换事件发生

Bit 2: BM0F, OBM0 标志。该位由OBM0切换源使能置位(BM0SE)。

0: 没有OBM0切换事件发生。仅由软件清零。

1: 有OBM0切换事件发生。

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
0	0	CRCDS1	CRCDS0	0	0	0	DPS
W	W	R/W	R/W	W	W	W	R/W

Bit 7~6: 保留位。软件在写入AUXR1时，必须向这些位写入” 0”。

Bit 5~4: CRCDS1~0。CRC0数据端口选择1~0。

Bit 3~1: 保留位。软件在写入AUXR1时，必须向这些位写入” 0”。

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	选择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	0	COPLK	T1X12	T0X12	T1CKOE	T0CKOE
R/W	R/W	W	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: 保留位。软件在写入AUXR2时, 必须向这些位写入”0”。

Bit 4: COPLK, PCA0 PWM/COPM缓冲寄存器更新控制。

0: 使能缓冲PWM/COPM模式的缓冲器由PCA0基准定时器溢出自动更新

1: 禁止缓冲PWM/COPM模式下由PCA0基准定时器溢出对缓冲器自动更新

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 2: 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: 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时钟输出在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时钟输出在T0CKO端口引脚。

### 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	P4.6
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	P7.0	P7.1

Bit 2~1: TWIP1~0, TWI0/I2C0 端口引脚选择[1:0].

TWIP1~0	TWI0_SCL	TWI0_SDA
0 0	P4.0	P4.1
0 1	P6.0	P6.1
1 0	P3.1	P3.0
1 1	P2.5	P2.3

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	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
0 0	P1.0	P1.1
0 1	P3.2	P3.3
1 0	P6.7	P5.7
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	P3.6
1 1	P3.7

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P7.2上

Bit 2: AC1FLT1, AC1输出滤波控制1.

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P0.0上

Bit 0: AC0FLT1, AC0输出滤波控制1.

**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
C0IC4S0	C0IC2S0	C0PPS1	C0PPS0	C0PS1	C0PS0	ECIPS0	C0COPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: C0IC4S0, PCA0 输入通道4输入选择

C0IC4S0	CEX4输入t
0	CEX4 端口引脚
1	T2EXI

Bit 6: C0IC2S0, PCA0 Input Channel 2 input 端口引脚选择.

C1IC2S0	CEX2输入
0	CEX2 端口引脚
1	T3EXI

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	P3.6	P3.7
1	P6.0	P6.1

Bit 3: C0PS1, PCA0 端口引脚选择1.

C0PS1	CEX1	CEX3	CEX5
0	P2.3	P2.5	P2.7
1	P3.3	P3.4	P3.5

Bit 2: C0PS0, PCA0 端口引脚选择0.

C0PS0	CEX0	CEX2	CEX4
0	P2.2	P2.4	P2.6
1	P4.0	P4.1	P3.2

Bit 1: ECIPS0, PCA0 ECI 端口引脚选择0.

ECIPS0	ECI
0	P2.1
1	P4.6

Bit 0: C0COPS, PCA0时钟输出(C0CKO) 端口引脚选择.

C0COPS	C0CKO
0	P4.7
1	P7.2

**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	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	P0.7	P0.6	P0.5	P0.4
1 0	P2.7	P2.6	P2.5	P2.4
1 1	P6.3	P6.2	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	P0.3	P0.2	P0.1	P0.0
1 0	P2.3	P2.2	P2.1	P2.0
1 1	P4.3	P4.2	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	S2MI	S3MI
0	P6.2	P6.3	P7.0	P7.1
1	P5.7	P6.7	P6.6	P6.5

Bit 0: SOCOPS, S0BRG时钟输出(S0CKO) 端口引脚选择.

SOCOPS	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 = 1100-0000

7	6	5	4	3	2	1	0
POE5	POE4	C0CKOE	SPI0M0	BM1OE1	BM1OE0	BM0OE1	BM0OE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: POE5, PCA0 PWM5主通道(PWM50)输出控制。

0: 禁止PWM50在端口引脚上输出。

1: 使能PWM50在端口引脚上输出。默认是使能的。

Bit 6: POE4, PCA0 PWM4主通道(PWM40)输出控制。

0: 禁止PWM40在端口引脚上输出。

1: 使能PWM40在端口引脚上输出。默认是使能的。

Bit 5: C0CKOE, PCA0时钟输出使能。

0: 禁止PCA0时钟输出。

1: PCA0基准定时器溢出率的二分之一时钟输出使能。

Bit 3~2: BM1OE1~0, BM1OUT输出端口引脚选择.

BM1OE1~0	BM1OUT 端口引脚
0 0	禁止
0 1	P4.3
1 0	P3.1
1 1	P7.2

Bit 1~0: BM0OE1~0, BM0OUT输出端口引脚选择.

BM0OE1~0	BM0OUT 端口引脚
0 0	禁止
0 1	P4.2
1 0	P3.0
1 1	P5.0

### AUXR8: 辅助寄存器 8

SFR 页 = 仅 5 页

SFR 地址 = 0xA4

复位值 = 1100-0000

7	6	5	4	3	2	1	0
POE7	POE6	C0PPS2	0	0	S1COPS	T3PS1	T3PS0
R/W	R/W	R/W	W	W	R/W	R/W	R/W

Bit 7: POE7, PCA0 PWM7主通道(PWM7O)输出控制。

0: 禁止PWM7O在端口引脚上输出。

1: 使能PWM7O在端口引脚上输出。默认是使能的。

Bit 6: POE6, PCA0 PWM6主通道(PWM6O)输出控制。

0: 禁止PWM6O在端口引脚上输出。

1: 使能PWM6O在端口引脚上输出。默认是使能的。

Bit 5: C0PPS2, {PWM6, PWM7} 端口引脚选择2.

C0PPS2	PWM6	PWM7
0	P6.2	P6.3
1	P6.0	P6.1

Bit 4~3: 保留位。软件在写入AUXR8时, 必须向这些位写入"0"。

Bit 2: S1COPS, S1BRG Clock输出(S1CKO) 端口引脚选择.

S1COPS	S1CKO
0	P4.6
1	P4.5

Bit 1~0: T3PS1~0, 定时器3 端口引脚选择[1:0].

T3PS1~0	T3/T3CKO	T3EX
0 0	P4.6	P0.0
0 1	P3.3	P3.2
1 0	P2.1	P2.0
1 1	P6.6	P6.5

### 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	C0FDC1	C0FDC0	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
0 0	禁止
0 1	INT1激活
1 0	TF3激活
1 1	TI1激活

Bit 4: T0G1, 定时器0门控源选择

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

Bit 3~2: C0FDC1~0, C0FDCK Selection [1:0].

C0FDC1~0	C0FDCK
0 0	T0OF
0 1	T1OF
1 0	T3OF
1 1	S0TOF

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	P7.1	P7.0
1 0	P2.0	P2.1
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 = 1100-0000

7	6	5	4	3	2	1	0
AC1HC0	AC0HC0	SPIPS1	SPIPS0	S0PS1	SPFACE	TWICF	PAA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: AC1HC0, AC1滞环控制0.

0: 禁止在AC1上滞环输入

1: 使能在AC1上滞环输入。默认是使能

Bit 6: AC0HC0, AC0 滞环控制 0.

0: 禁止在AC0上滞环输入

1: 使能在AC0上滞环输入。默认是使能

Bit 5~4: SPIPS1~0, SPI 端口引脚选择[1:0].

Pin Options	SPIPS1~0	nSS	MOSI (SPID0)	MISO (SPID1)	SPICLK	SPID2	SPID3
0	0 0	P1.4	P1.5	P1.6	P1.7	P5.3	P5.4
1	0 1	P4.3	P4.2	P4.1	P4.0	P3.6	P3.7
2	1 0	P4.0	P4.1	P4.2	P4.3	P6.5	P5.7
3	1 1	P4.3	P4.1	P4.2	P4.0	P6.5	P5.7

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: SPFACE, SPIF自动清零使能位

0: 禁止, SPIF只能软件清零

1: 使能. SPIF会被CPU读/写SPDAT操作清零

Bit 1: TWICF, TWI0/I2C0 串行时钟输入滤波

- 0: 禁止TWICF 功能.  
1: 使能TWICF 功能.

Bit 0: PAA, TWI0/I2C0 预回应.

- 0: 禁止PAA 功能.  
1: 在TWI0/I2C0 主机RX和从机TX/RX 使用DMA传送时使能PAA功能

#### 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
I2C1CF	PAA1	I2C1PS1	I2C1PS0	0	POEM0	C0M0	C0OFS
R/W	R/W	R/W	R/W	W	R/W	R/W	R/W

Bit 7: I2C1CF, TWI1/I2C1串行时钟输入滤波

- 0: 禁止 I2C1CF 功能.  
1: 使能 I2C1CF 功能.

Bit 6: PAA1, TWI1/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PAA功能

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

I2C1PS1~0	TWI1_SCL	TWI1_SDA
0 0	P4.2	P4.3
0 1	P6.0	P6.1
1 0	P6.5	P6.6
1 1	P6.3	P6.2

Bit 3: 保留位。软件在写入AUXR11时，必须向这些位写入“0”。

Bit 2: POEM0, PCA0 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 1: C0M0, PCA模式控制0.

- 0: PWM中心对齐不支持可变分辨率。  
1: 使能PCA支持可变分辨率的PWM中心对齐。使能此功能，PCA0 (PWMCR.7)也需要置位。

Bit 0: C0OFS, 当C0M0使能PCA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。

#### 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
--	--	T0COBE	T0COAE	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 5: T0CKOB(=T0OR) 输出使能.

- 0: 禁止T0CKOB 输出.  
1: 使能T0CKOB 在P4.3上输出.

Bit 4: T0CKOA(=T0OR) 输出使能.

- 0: 禁止T0CKOA 输出.  
1: 使能T0CKOA 在P4.0上输出.

**AUXR13: 辅助寄存器 13**

SFR 页 = 仅 A 页

SFR 地址 = 0xA4

复位值 = 0000-0000

7	6	5	4	3	2	1	0
0	S3PS0	0	S2PS0	0	T4FCS	T4PS1	T4PS0
W	R/W	W	R/W	W	R/W	R/W	R/W

Bit 6: S3PS0, 串口3 引脚选择0.

S3PS0	RXD3	TXD3
0	P3.6	P3.7
1	P6.5	P6.6

Bit 4: S2PS0, 串口2 引脚选择0.

S2PS0	RXD2	TXD2
0	P3.2	P3.3
1	P6.7	P5.7

Bit 2: T4FCS, 保留用于芯片测试

Bit 1~0: T4PS1~0, 定时器4 端口引脚选择[1:0].

T4PS1~0	T4/T4CKO	T4EX
0 0	P7.0	P7.1
0 1	P3.7	P3.6
1 0	P7.2	P5.0
1 1	P3.6	P3.7

**AUXR14: 辅助寄存器 14**

SFR 页 = 仅 B 页

SFR 地址 = 0xA4

复位值 = 0000-0100

7	6	5	4	3	2	1	0
ADLPS1	ADLPS0	0	0	RWPS0	AC2HC0	AC2OE	AC2FLT1
R/W	R/W	W	W	R/W	R/W	R/W	R/W

Bit 7~6: EMB AD总线引脚选择位(片外MOVX 时序).

ADLPS1~0	AD[7:0]
0 0	P0.7~P0.0
0 1	P2.7~P2.0
1 0	P7.2~P7.0, P5.7, P6.7~P6.4
1 1	保留.

Bit 3: RWPS0, EMB nRD/nWR 引脚选择位

RWPS0	nWR	nRD
0	P3.6	P3.7
1	P3.7	P3.6

Bit 2: AC2HC0, AC2 滞环控制 0.

0: 禁止在AC2上滞环输入

1: 使能在AC2上滞环输入。默认是使能

Bit 1: AC2OE, 使能AC2OUT输出在端口引脚上。

0: 禁止AC2OUT 输出在端口引脚上。

1: 使能AC2OUT输出在P6.4上

Bit 0: AC0FLT1, AC0输出滤波控制1.

**SFRPI: SFR页索引寄存器**

SFR 页 = 0~F

SFR 地址 = 0xAC

复位值 = xxxx-0000

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

Bit 7~4: 保留位。软件在写入SFRPI时，必须向这些位写入”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

## 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“Megawin 8051 ICE Adapter”（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。MGEQ1C064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：**0xFFFF**。下表列举了ISP空间选项。默认设定，MGEQ1C064 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	MGEQ1C064 ISP起始地址
7.5K字节	E200
7.0K字节	E400
6.5K字节	E600
6.0K字节	E800
5.5K字节	EA00
5.0K字节	EC00
4.5K字节	EE00
4.0K字节	F000
3.5K字节	F200
3.0K字节	F400
2.5K字节	F600
2.0K字节	F800
1.5K字节	FA00
1.0K字节	FC00
0.5K字节	FE00
无ISP空间	

### 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，启动外部复位从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，它被配置为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3.7V.
- , : 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、3.7V、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:**

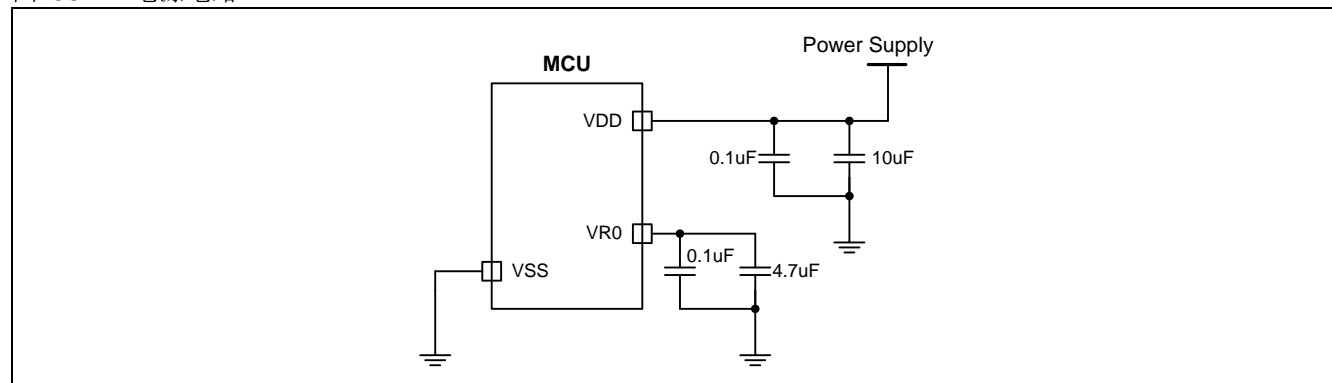
- : 使能。WDT 特殊寄存器，WDTCR的WREN、NSW、WIDL、PS2、PS1和PS0位，将被写保护。
- : 禁止。WDT 特殊寄存器，WDTCR的WREN、NSW、WIDL、PS2、PS1和PS0位，由软件自由写。

## 36. 应用说明

### 36.1. 电源电路

MG82F6D64/32 的工作电源变化可以从 2.4V 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



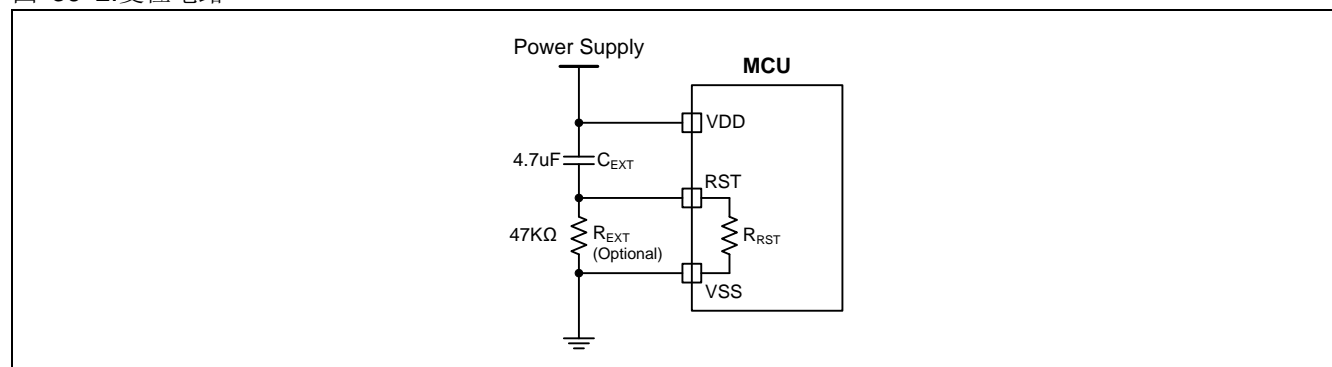
### 36.2. 复位电路

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 VDD(电源)的电容  $C_{EXT}$  和一个连接到 VSS(地)的电阻组成。

一般的， $R_{EXT}$  是可选的，因为 RST 引脚有一个内部下拉电阻 ( $R_{RST}$ )。这个对 VSS 的内部扩散电阻在仅使用一个外部对 VDD 的电容  $C_{EXT}$  时也可产生一个上电复位。

$R_{RST}$  的值见章节“37.2 DC 特性”。

图 36-2. 复位电路



### 36.3. XTAL 振荡电路

为了能成功起振 (最大到24MHz), 电容C1和C2是必须的, 如图 36-3所示。通常, C1和 C2使用相同的值。表 36.1列举了C1 & C2 在不同晶振下的值。

图 36-3. XTAL 振荡电路

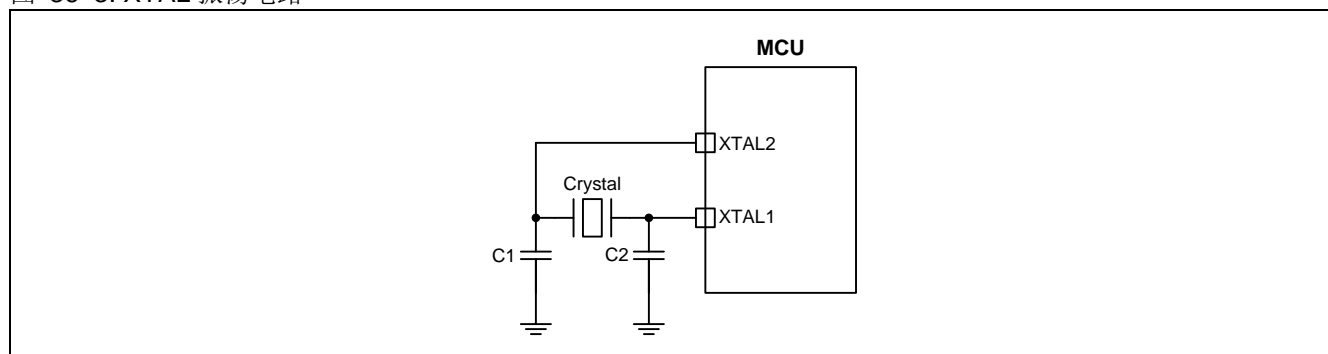


表 36.1. 振荡电路的电容 C1 & C2 参照表

晶振	C1, C2 电容 <sup>1</sup>
16MHz ~ 25MHz	10pF
6MHz ~ 16MHz	15pF
2MHz ~ 6MHz	33pF
32768Hz	7pF

注 1: 这些电容值仅供参考。负载电容值会根据晶体的特性有所不同。

### 36.4. ICP 和 OCD 接口电路

MGEQ1C064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。

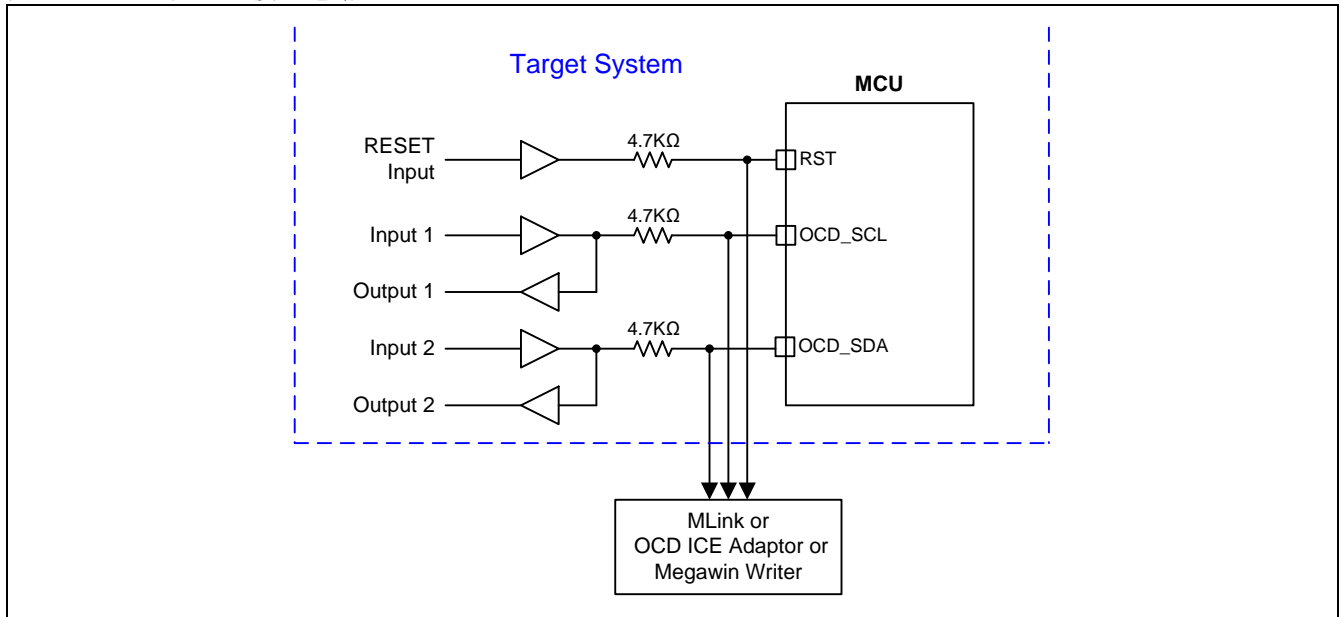
ICP接口允许的ICP\_SCL/ICP\_SDA 引脚与用户应用共享, 使得可以实现在芯片FLASH编程。这是可行的, 因为当芯片在Halt状态时执行ICP通信, 此时芯片上的外围设备和用户软件都是失效的。在halt状态, ICP接口能够安全的“借用” ICP\_SCL (P4.4)和 ICP\_SDA (P4.5)引脚。在大多应用中, 必须用外部电阻来隔开ICP电路和用户应用电路。一种典型的隔离方法如图 36-4所示。

*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, MGEQ1C064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OCD\_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”及“Megawin 8051 ICE Adapter”可以支持MGEQ1C064在线编程。“MLink”及“Megawin 8051 ICE Adapter”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及图 36-6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”及“Megawin 8051 ICE Adapter” 传送编程数据到目标MCU； RST 线用来暂停MCU； VCC和 GND 是6脚“Megawin 8051 ICE Adapter” 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”及“Megawin 8051 ICE Adapter”。

图 36-5.经ICP32 (MLink)的独立编程

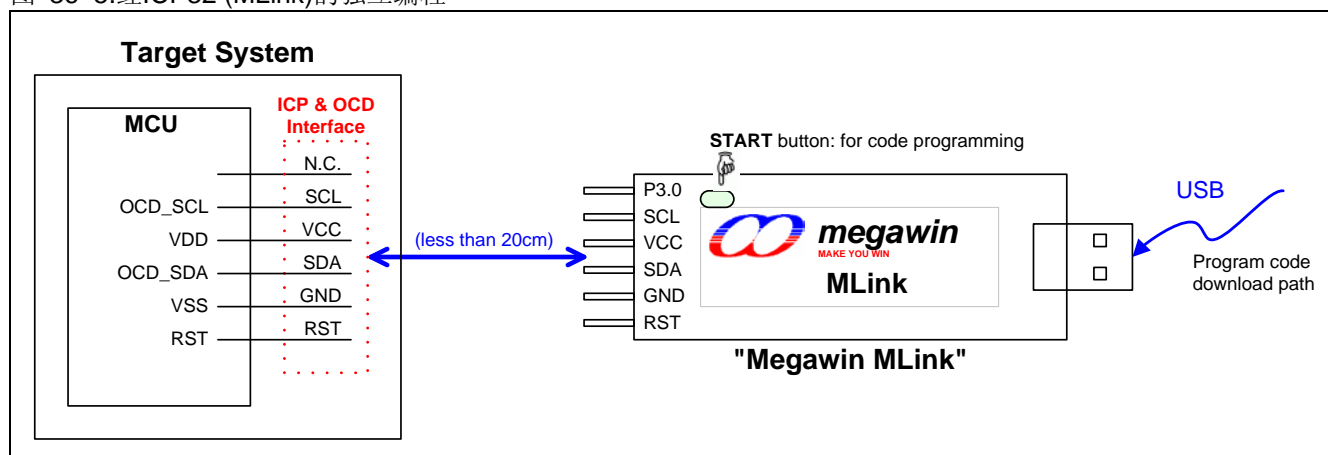
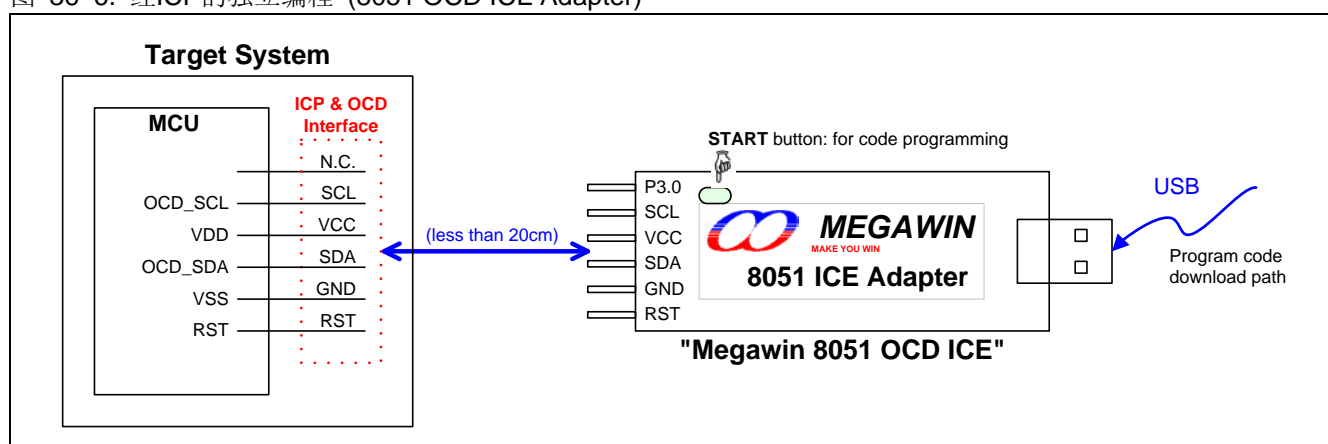


图 36-6. 经ICP的独立编程 (8051 OCD ICE Adapter)



## 36.6. 在线调试功能

**MGEQ1C064** 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”及“Megawin 8051 OCD ICE”给用户，如图 36-7及图 36-8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 36-7及图 36-8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 36-7. MLink ICE功能的系统框图

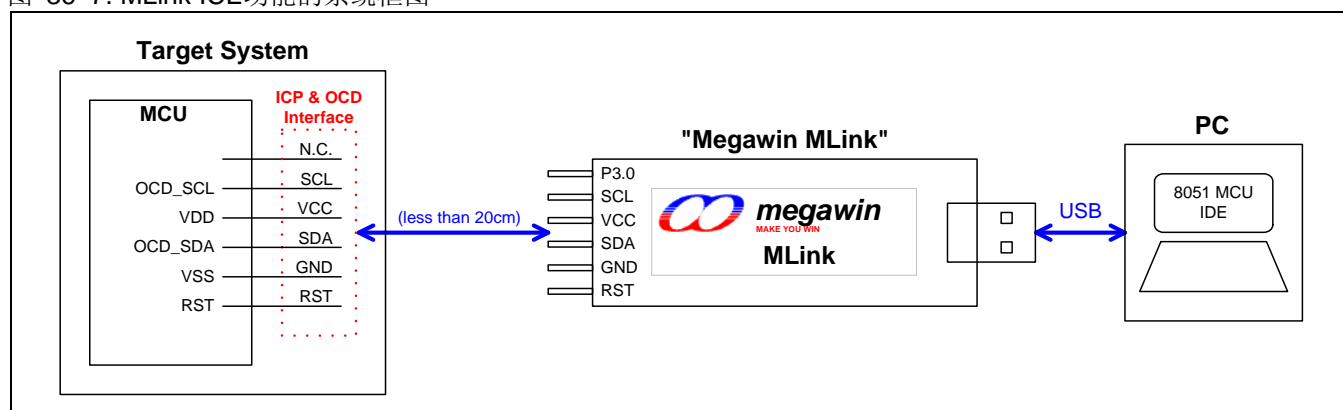
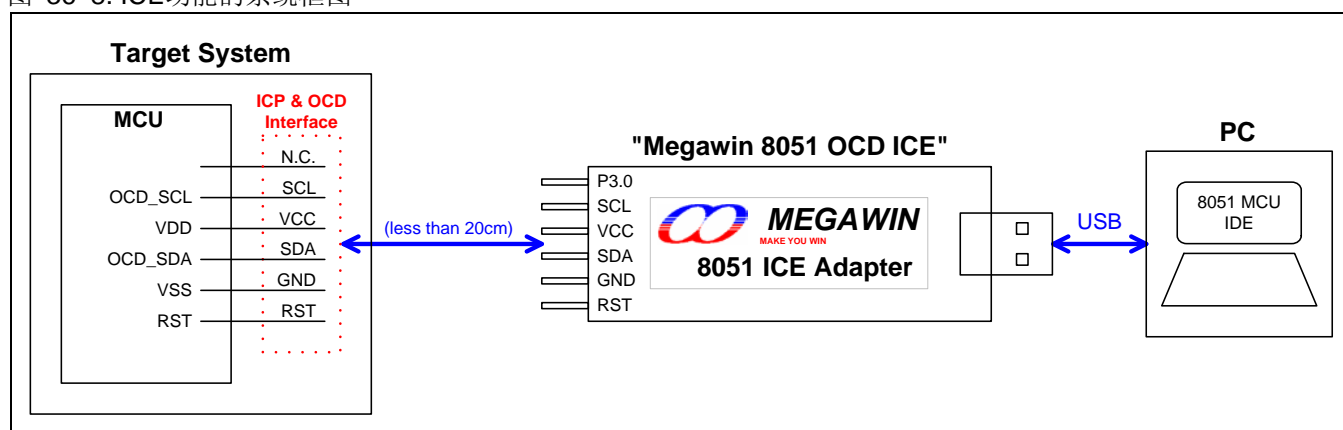


图 36-8. ICE功能的系统框图



*注意：有关OCD ICE的更多详细信息，请联系笙泉。*

## 37. 电气特性

### 37.1. 绝对最大额定值

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. DC 特性

VDD = 5.0V±10%, VSS = 0V, T<sub>A</sub> = 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	
<b>输入/输出特性</b>						
V <sub>IH1</sub>	输入高电平(所有I/O口)	除P6.0, P6.1之外	0.6			VDD
V <sub>IH2</sub>	输入高电平(RST, P6.0, P6.1)		0.75			VDD
V <sub>IL1</sub>	输入低电平(所有I/O口)	除P6.0, P6.1之外			0.15	VDD
V <sub>IL2</sub>	输入低电平(RST, P6.0, P6.1)				0.2	VDD
I <sub>IH</sub>	输入高漏电流(所有I/O口) <sup>(2) (3)</sup>	V <sub>PIN</sub> = VDD		0	±1	uA
I <sub>IL1</sub>	逻辑0输入电流(P3在准双向模式) <sup>(2) (3)</sup>	V <sub>PIN</sub> = 0.4V		20	30	uA
I <sub>IL2</sub>	逻辑0输入电流(所有仅输入或开漏端口) <sup>(2) (3)</sup>	V <sub>PIN</sub> = 0.4V		0	1	uA
I <sub>H2L</sub>	逻辑1到0输入转变电流(P3在准双向模式) <sup>(2) (3)</sup>	V <sub>PIN</sub> = 5.5V		160	390	uA
		V <sub>PIN</sub> = 5V		140	280	
		V <sub>PIN</sub> = 3.3V		55	110	
I <sub>OH1</sub>	输出高电流(P3在准双向模式) <sup>(2) (3)</sup>	VDD=5V; V <sub>PIN</sub> =2.4V	-250	-255		uA
		VDD=3.3V; V <sub>PIN</sub> =2.4V	-70	-72		uA
I <sub>OH2</sub>	输出高电流(所有推挽输出端口) <sup>(2) (3)</sup>	VDD=5V; V <sub>PIN</sub> =2.4V	-33	-34.5		mA
		VDD=3.3V; V <sub>PIN</sub> =2.4V	-9.5	-10.5		mA
I <sub>OH3</sub>	输出高电流(所有推挽输出端口在低驱动能力, 除RST引脚) <sup>(2) (3)</sup>	VDD=5V; V <sub>PIN</sub> =2.4V	-13	-13.4		mA
		VDD=3.3V; V <sub>PIN</sub> =2.4V	-4.0	-4.2		mA
I <sub>OL1</sub>	输出低电流(所有I/O口) <sup>(2) (3)</sup>	VDD=5V; V <sub>PIN</sub> =0.4V	24.0	25.0		mA
		VDD=3.3V; V <sub>PIN</sub> =0.4V	17.0	18.0		mA
I <sub>OL2</sub>	输出低电流(所有推挽输出端口在低驱动能力, 除RST引脚) <sup>(2) (3)</sup>	VDD=5V; V <sub>PIN</sub> =0.4V	3.2	3.3		mA
		VDD=3.3V; V <sub>PIN</sub> =0.4V	2.3	2.36		mA
R <sub>Very_Weak</sub>	内部非常弱上拉电阻	VDD=5V		275		Kohm
		VDD=3.3V		470		Kohm
R <sub>weak</sub>	内部弱上拉电阻	VDD=5V		11		Kohm
		VDD=3.3V		19.5		Kohm
R <sub>RST</sub>	内部复位下拉电阻	VDD=5V		123		Kohm
		VDD=4.5V		138		
		VDD=3.3V		203		
		VDD=2.7V		270		
<b>功耗</b>						
I <sub>OP1</sub>	正常模式工作电流t	CPUCLK=SYSCLK = 36MHz @ IHRCO with PLL	8.5	8.6	TBD	mA
I <sub>OP2</sub>		CPUCLK=SYSCLK = 32MHz @ IHRCO with PLL	7.5	8	TBD	mA
I <sub>OP3</sub>		CPUCLK=SYSCLK = 24MHz @ IHRCO with PLL	6.7	6.8	TBD	mA
I <sub>OP4</sub>		CPUCLK=SYSCLK = 12MHz @ IHRCO	4.7	4.8	TBD	mA
I <sub>OP5</sub>		CPUCLK=SYSCLK = 12MHz @ IHRCO, VDD = 5V with ADC 400K sps		6.5	TBD	mA
I <sub>OP6</sub>		CPUCLK=SYSCLK = 24MHz @ IHRCO with PLL, VDD = 5V with ADC 800K sps		8.5	TBD	mA
I <sub>OPS1</sub>	低速模式工作电流	CPUCLK=SYSCLK = 12MHz/128 @ IHRCO (SFR HSE=0)		0.45		mA
I <sub>IDLE1</sub>	空闲模式工作电流	SYSClk = 12MHz @ IHRCO		1.34		mA
I <sub>IDLE2</sub>		SYSClk = 12MHz/128 @ IHRCO (SFR HSE=0)		0.42		mA
I <sub>IDLE3</sub>		SYSClk = 32KHz @ ILRCO		58		uA
I <sub>SUB1</sub>	副频模式工作电流	SYSClk = 32KHz @ ILRCO, BOD1 禁止		51		uA
I <sub>SUB2</sub>		SYSClk = 32KHz/128 @ ILRCO, BOD1 禁止		40		uA

I <sub>WAT</sub>	Watch模式工作电流	WDT = 32KHz @ ILRCO在PD模式		7.8		uA
I <sub>MON1</sub>	Monitor模式工作电流	PD模式下BOD1使能		12.5		uA
I <sub>RTC1</sub>	RTC模式操作电流	RTC工作在PD模式, VDD = 5.0V		5.5		uA
I <sub>PD1</sub>	掉电模式电流			3.2		uA
<b>BOD0/BOD1 特性</b>						
V <sub>BOD0</sub>	BOD0监测电压	T <sub>A</sub> = -40° C to +105° C		1.7		V
V <sub>BOD10</sub>	BOD1监测电压为2.0V	T <sub>A</sub> = -40° C to +105° C		2.0		V
V <sub>BOD10</sub>	BOD1监测电压为2.4V	T <sub>A</sub> = -40° C to +105° C		2.37		V
V <sub>BOD11</sub>	BOD1监测电压为3.7V	T <sub>A</sub> = -40° C to +105° C		3.7		V
V <sub>BOD11</sub>	BOD1监测电压为4.2V	T <sub>A</sub> = -40° C to +105° C		4.2		V
I <sub>BOD1</sub>	BOD1功耗	T <sub>A</sub> = +25° C, VDD=5.0V		6.8		uA
		T <sub>A</sub> = +25° C, VDD=3.3V		5.2		
<b>工作环境</b>						
V <sub>PSR</sub>	上电边沿速率	T <sub>A</sub> = -40° C to +85° C	0.05			V/ms
V <sub>POR1</sub>	上电复位有效电压	T <sub>A</sub> = -40° C to +85° C			0.1	V
V <sub>OP1</sub>	CPU工作速度 0-36MHz	T <sub>A</sub> = -40° C to +105° C	2.7		5.5	V
V <sub>OP2</sub>	CPU工作速度0-24MHz	T <sub>A</sub> = -40° C to +105° C	2.4		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<sub>H2L</sub>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准双向口结构”。

(3) 所有电流流入芯片是正值，电流流出芯片为负值。

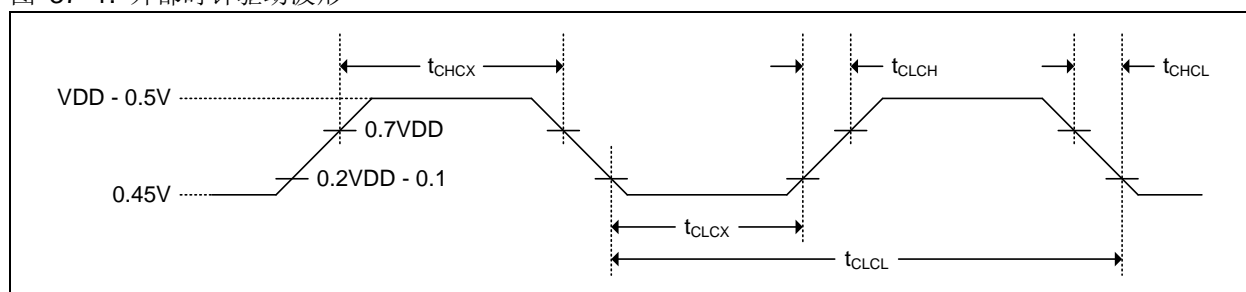
## 37.3. 外部时钟特性

VDD = 2.4V ~ 5.5V, VSS = 0V, T<sub>A</sub> = -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 <sub>CLCL</sub>	晶振频率	0.032	25	0	25	MHz
1/t <sub>CLCL</sub>	晶振频率 (VDD = 2.4V ~ 5.5V)	0.032	12	0	12	MHz
t <sub>CLCL</sub>	时钟周期	41.6		27.7		ns
t <sub>CHCX</sub>	高时间	0.4T	0.6T	0.4T	0.6T	t <sub>CLCL</sub>
t <sub>CLCX</sub>	低时间	0.4T	0.6T	0.4T	0.6T	t <sub>CLCL</sub>
t <sub>CLCH</sub>	上升时间		5		5	ns
t <sub>CHCL</sub>	下降时间		5		5	ns

注：数据为设计保证，非量产测试

图 37-1. 外部时钟驱动波形



## 37.4. IHRCO 特性

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
IHRCO 频率	T <sub>A</sub> = +25°C, AFS = 0		12		MHz
	T <sub>A</sub> = +25°C, AFS = 1		11.059		MHz
IHRCO 频率误差 (工厂校对)	T <sub>A</sub> = +25°C	-1.0		+1.0	%
	T <sub>A</sub> = -40°C to +105°C	-2.0		+2.0	%
	T <sub>A</sub> = -40°C to +105°C			32 <sup>(1)</sup>	us
IHRCO 启动时间	T <sub>A</sub> = +25°C, VDD=5.0V		350 <sup>(1)</sup>		uA

<sup>(1)</sup> 数据基于特性表征结果，非量产测试。

## 37.5. ILRCO 特性

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
ILRCO 频率	T <sub>A</sub> = +25°C		32		KHz
ILRCO 频率误差	T <sub>A</sub> = +25°C	-8		+8	%
	T <sub>A</sub> = -40°C to +105°C	-15		+15	%

注:数据基于特性表征结果，非量产测试。

## 37.6. 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	2.4		5.5	V
时钟输入范围	TA = -40° C to +105° C	4.5 <sup>(1)</sup>		6.5 <sup>(1)</sup>	MHz
CKM 启动时间	TA = -40° C to +105° C	30 <sup>(2)</sup>		100 <sup>(2)</sup>	us
CKM 功耗	TA = +25° C, VDD=5.0V, CKM = 96MHz		350		uA
	TA = +25° C, VDD=5.0V, CKM = 144MHz		450		

注:<sup>(1)</sup> 数据为设计保证, 非量产测试.

<sup>(2)</sup> 数据基于特性表征结果, 非量产测试.

## 37.7. 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	2.4		5.5	V
Flash 写 (擦除/编程)电压	TA = -40° C to +105° C	2.4		5.5	V
Flash 擦除/编程 周期	TA = -40° C to +105° C	20,000			次
Flash 数据保留期	TA = +25° C	100			年

注:数据为设计保证, 非量产测试.

## 37.8. ADC 特性

VDD=5.0V, T<sub>A</sub>= -40° C ~ +105° 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
<b>DC 精度 T<sub>A</sub>= 25° C</b>					
积分非线性	VDD = VREF+ ≥ 5V, 983K sps	-2.4		2.5	LSB
	VDD = VREF+ ≥ 2.4V, 533K sps	-2.3		3.0	
	VDD = VREF+ ≥ 5V, 400K sps	-2.1		1.7	
	VDD = 5.0V, VREF+ = 2.4V, 800K sps	-2.5		2.7	
差分非线性	VDD = VREF+ ≥ 5V, 983K sps	-1.0		1.4	LSB
	VDD = VREF+ ≥ 2.4V, 533K sps	-1.0		1.6	
	VDD = VREF+ ≥ 5V, 400K sps	-1.0		1.2	
	VDD = 5.0V, VREF+ = 2.4V, 800K sps	-1.0		2.1	
偏移误差	VDD= 2.4V~5.5V	-2		+2	LSB
<b>DC 精度 T<sub>A</sub>= -40° C ~ +105° C</b>					
积分非线性	VDD = VREF+ ≥ 5V, 983K sps	-2.8		4.0	LSB
	VDD = VREF+ ≥ 2.4V, 533K sps	-2.8		4.2	
	VDD = VREF+ ≥ 5V, 400K sps	-2.5		2.4	
	VDD = 5.0V, VREF+ = 2.4V, 800K sps	-3.4		4.0	
差分非线性	VDD = VREF+ ≥ 5V, 983K sps	-1.0		2.4	LSB
	VDD = VREF+ ≥ 2.4V, 533K sps	-1.0		1.8	
	VDD = VREF+ ≥ 5V, 400K sps	-1.0		1.6	
	VDD = 5.0V, VREF+ = 2.4V, 800K sps	-1.0		2.1	
偏移误差	VDD= 2.4V~5.5V	-4		+4	LSB
<b>转换率</b>					
SAR转换时钟			30		MHz
在SAR时钟里的转换时间			30		clocks
建议转换率	VDD = VREF+ ≥ 5V, 983K sps			983	K sps
	VDD = VREF+ ≥ 2.4V, 533K sps			533	
	VDD = 5.0V, VREF+ = 2.4V, 800K sps			800	
<b>模拟输入</b>					
ADC输入电压范围	单端 (AIN+ – GND)	0		VDD	V
C <sub>ADC</sub> 输入电容			4	6	pF
输入采样开关电阻	VDD = 5V		327		Ω
	VDD = 4.2V		369		Ω
	VDD = 3.3V		447		Ω
	VDD = 2.7V		541		Ω
	VDD = 2.4V		613		Ω
<b>通道切换稳定时间</b>					
原引脚与目标引脚电压在VDD或GN之间切换	CH0(VDD)→CH1(GND)		0		us
	CH0(GND)→CH1(VDD)		0		
原引脚电压=VDD切换到目标引脚带下拉电阻	CH0(VDD)→CH1(51K下拉)		3.63		
	CH0(VDD)→CH1(10K下拉)		0.55		
原引脚电压=GND切换到目标引脚带上拉电阻	CH0(GND)→CH1(51K上拉)		5.94		
	CH0(GND)→CH1(10K上拉)		0.63		
原引脚电压=VDD切换到电阻分压(VDD/2)	CH0(VDD)→CH1(VDD/2, 51K电阻分压)		2.11		
	CH0(VDD)→CH1(VDD/2, 10K电阻分压)		0.38		
原引脚电压=GND切换到电阻分压(VDD/2)	CH0(GND)→CH1(VDD/2, 51K电阻分压)		3.86		
	CH0(GND)→CH1(VDD/2, 10K电阻分压)		0.52		
<b>功耗</b>					
电源电流	ADPS<1:0>=00		1.65		mA
	ADPS<1:0>=01		1.63		
	ADPS<1:0>=10		1.60		
	ADPS<1:0>=11		1.56		

注:数据基于特性表征结果, 非量产测试.

## 37.9. IVR 特性

VDD=5.0V±10%, VSS=0V, T<sub>A</sub> = -40°C to +105°C, C<sub>LOAD</sub>=4.7µF/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	
<b>电源范围</b>					
电源电压		2.4	5.0	5.5	V
功耗	VDD = VREF+ = 5V		50	54	µA
<b>DC Accuracy</b>					
输出电压	-40°C ~ +85°C	1.37	1.4	1.43	V
电压变化从TA = 25°C到整个温度范围	VDD = 3.3V±10mV	-7		+8	mV

注:数据基于特性表征结果, 非量产测试.

## 37.10. 模拟比较器 AC0/AC1/AC2 特性

VDD=5.0V, T<sub>A</sub>= -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	
<b>电源范围</b>					
电源电压		2.4	5.0	5.5	V
工作电流	正常功耗状态		16		uA
	低功耗状态		1.1		uA
R-Ladder (32阶)	VDD= 5.0V		90		uA
R-Ladder (24阶)	VDD= 5.0V		120		uA
<b>DC 精度</b>					
输入电压范围	Rail to Rail	50		VDD-50	mV
输入偏移电压-AC0 (正常模式)	VDD= 5.0V		1.7	11	mV
	VDD= 3.0V		2.3	9	mV
输入偏移电压-AC1 (正常模式)	VDD= 5.0V		3	28	mV
	VDD= 3.0V		4	24	mV
输入偏移电压-AC2 (正常模式)	VDD= 5.0V		5	27	mV
	VDD= 3.0V		3.7	26	mV
输入偏移电压-AC0 (低功耗模式)	VDD= 5.0V		2	31	mV
	VDD= 3.0V		1.8	25	mV
输入偏移电压-AC1 (低功耗模式)	VDD= 5.0V		2	37	mV
	VDD= 3.0V		3.8	31	mV
输入偏移电压-AC2 (低功耗模式)	VDD= 5.0V		6	46	mV
	VDD= 3.0V		5	47	mV
输入共模电压		50		VDD-50	mV
比较器迟滞i-AC0 (正常模式)	VDD= 5.0V		10	30	mV
	VDD= 3.0V		11	33	
比较器迟滞i-AC1 (正常模式)	VDD= 5.0V		9	32	mV
	VDD= 3.0V		10	34	
比较器迟滞i-AC2 (正常模式)	VDD= 5.0V		8	30	mV
	VDD= 3.0V		9	32	
比较器迟滞i-AC0 (低功耗模式)	VDD= 5.0V		5	16	mV
	VDD= 3.0V		6	17	
比较器迟滞i-AC1 (低功耗模式)	VDD= 5.0V		3	16	mV
	VDD= 3.0V		3	18	
比较器迟滞i-AC2 (低功耗模式)	VDD= 5.0V		3	16	mV
	VDD= 3.0V		3	18	
响应时间(正常模式)	Rising(V <sub>DD</sub> = 5V, T <sub>a</sub> = 25° C, V <sub>OD</sub> =50mV, V <sub>CM</sub> < 50mV)	110		150	ns
	Falling(V <sub>DD</sub> = 5V, T <sub>a</sub> = 25° C, V <sub>OD</sub> =50mV, V <sub>CM</sub> < 50mV)	180		320	ns
	Rising(V <sub>DD</sub> ≥ 3V, V <sub>OD</sub> =100mV, 0.5V ≤ V <sub>CM</sub> ≤ VDD-0.5V)	80		150	ns

	Falling( $V_{DD} \geq 3V$ , $V_{OD}=100mV$ , $0.5V \leq V_{CM} \leq V_{DD}-0.5V$ )	85		140	ns
	Rising( $V_{DD} \geq 3V$ , $V_{OD}=100mV$ , $V_{CM} < 0.5V$ , $V_{CM} > 4V$ )	100		185	ns
	Falling( $V_{DD} \geq 3V$ , $V_{OD}=100mV$ , $V_{CM} < 0.5V$ , $V_{CM} > 4V$ )	110		170	ns
响应时间(低功耗模式)	Rising( $V_{DD} = 5V$ , $T_a = 25^\circ C$ , $V_{OD}=50mV$ , $V_{CM} < 50mV$ )	1.2		1.7	us
	Falling( $V_{DD} = 5V$ , $T_a = 25^\circ C$ , $V_{OD}=50mV$ , $V_{CM} < 50mV$ )	1.5		6.3	us
	Rising( $V_{DD} \geq 3V$ , $V_{OD}=100mV$ , $0.5V \leq V_{CM} \leq V_{DD}-0.5V$ )	0.48		2.1	us
	Falling( $V_{DD} \geq 3V$ , $V_{OD}=100mV$ , $0.5V \leq V_{CM} \leq V_{DD}-0.5V$ )	0.44		1.60	us
	Rising( $V_{DD} \geq 3V$ , $V_{OD}=100mV$ , $V_{CM} < 0.5V$ , $V_{CM} > 4V$ )	0.50		1.75	us
	Falling( $V_{DD} \geq 3V$ , $V_{OD}=100mV$ , $V_{CM} < 0.5V$ , $V_{CM} > 4V$ )	0.52		2.70	us
上电时间 (从掉电)	正常模式 ( $V_{DD} = 5V$ , SYSCLK = CPUCLK = 12MHz)		3		us
	低功耗模式 ( $V_{DD} = 5V$ , SYSCLK = CPUCLK = 12MHz)		75		us

注:数据基于特性表征结果, 非量产测试.

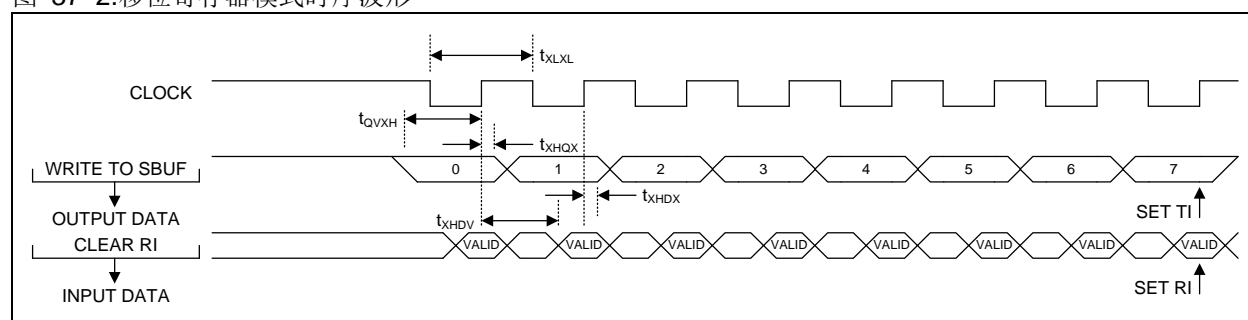
## 37.11. 串口时序特性

VDD = 5.0V±10%, VSS = 0V, T<sub>A</sub> = -40°C to +105°C, 除非另有说明

标号	参数	URM0X3 = 0		URM0X3 = 1		单位
		最小	最大	最小	最大	
t <sub>XLXL</sub>	串口时钟周期	12T		4T		T <sub>SYSCLK</sub>
t <sub>QVXH</sub>	设置输出数据到时钟上升沿	10T-20		2T-20		ns
t <sub>XHQX</sub>	上升沿后保持输出数据	T-10		T-10		ns
t <sub>XHDX</sub>	上升沿后保持输入数据	5		5		ns
t <sub>XHDV</sub>	时钟上升沿到输入数据有效		2T-10		2T-10	ns

注:数据为设计保证, 非量产测试.

图 37-2.移位寄存器模式时序波形



## 37.12. SPI 时序特性

VDD = 5.0V±10%, VSS = 0V, T<sub>A</sub> = -40°C to +105°C, 除非另有说明

标号	参数	最小	最大	单位
<b>主机模式时序</b>				
1/(t <sub>MCKH</sub> + t <sub>MCKL</sub> )	SPI时钟频率 @VDD = 3.0V ~ 5.5V		24	MHz
	SPI 时钟频率 @VDD = 2.4V ~ 3.0V		16	MHz
t <sub>MCKH</sub>	SPICLK 高时间	1T		T <sub>SYSCLK</sub>
t <sub>MCKL</sub>	SPICLK 低时间	1T		T <sub>SYSCLK</sub>
t <sub>MIS</sub>	MISO 有效到SPICLK 采样边沿	10		ns
t <sub>MIH</sub>	SPICLK转变边沿到MISO 变化	0		ns
t <sub>MOH</sub>	SPICLK转变边沿到MOSI 变化		10	ns
<b>从机模式时序</b>				
1/(t <sub>CKH</sub> + t <sub>CKL</sub> )	SPI 时钟频率 @VDD = 3.0V ~ 5.5V		16	MHz
	SPI 时钟频率 @VDD = 2.4V ~ 3.0V		12	MHz
t <sub>SE</sub>	nSS下降沿到第一个SPICLK边沿	2T		T <sub>SYSCLK</sub>
t <sub>SD</sub>	最后一个SPICLK边沿到nSS上升沿	2T		T <sub>SYSCLK</sub>
t <sub>SEZ</sub>	nSS下降沿到MISO有效		4T	T <sub>SYSCLK</sub>
t <sub>SDZ</sub>	nSS上升沿到MISO高阻		4T	T <sub>SYSCLK</sub>
t <sub>CKH</sub>	SPICLK高时间	2T		T <sub>SYSCLK</sub>
t <sub>CKL</sub>	SPICLK低时间	2T		T <sub>SYSCLK</sub>
t <sub>SIS</sub>	MOSI有效到SPICLK采样边沿	1T		T <sub>SYSCLK</sub>
t <sub>SIH</sub>	SPICLK采样边沿到MOSI变化	1T		T <sub>SYSCLK</sub>
t <sub>SOH</sub>	SPICLK移位边沿到MISO变化		2T	T <sub>SYSCLK</sub>
t <sub>SLH</sub>	最后的SPICLK边沿到MISO变化 (仅CPHA = 1)	1T	2T	T <sub>SYSCLK</sub>

注:数据为设计保证, 非量产测试.

图 37-3. CPHA=0时SPI主机传送波形

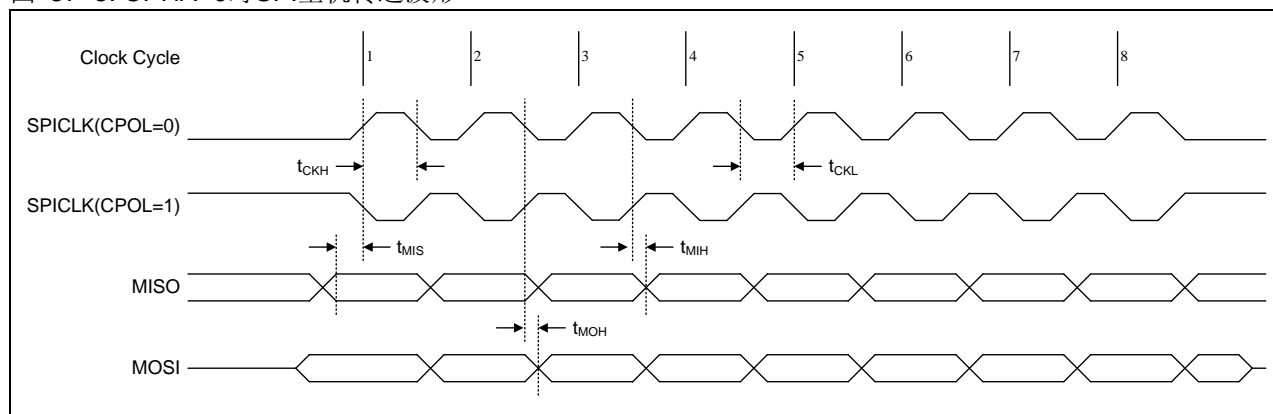


图 37-4. CPHA=1时SPI主机传送波形

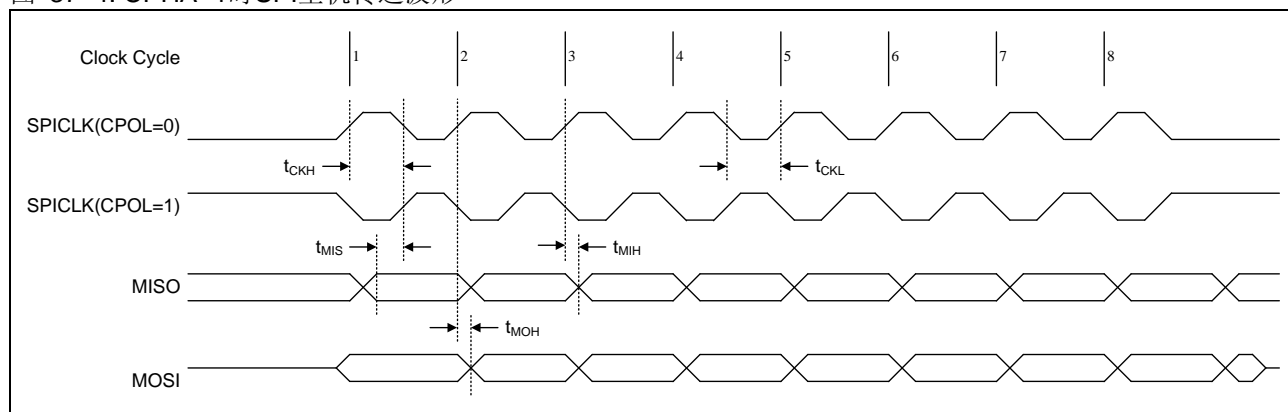


图 37-5. CPHA=0时SPI从机传送波形

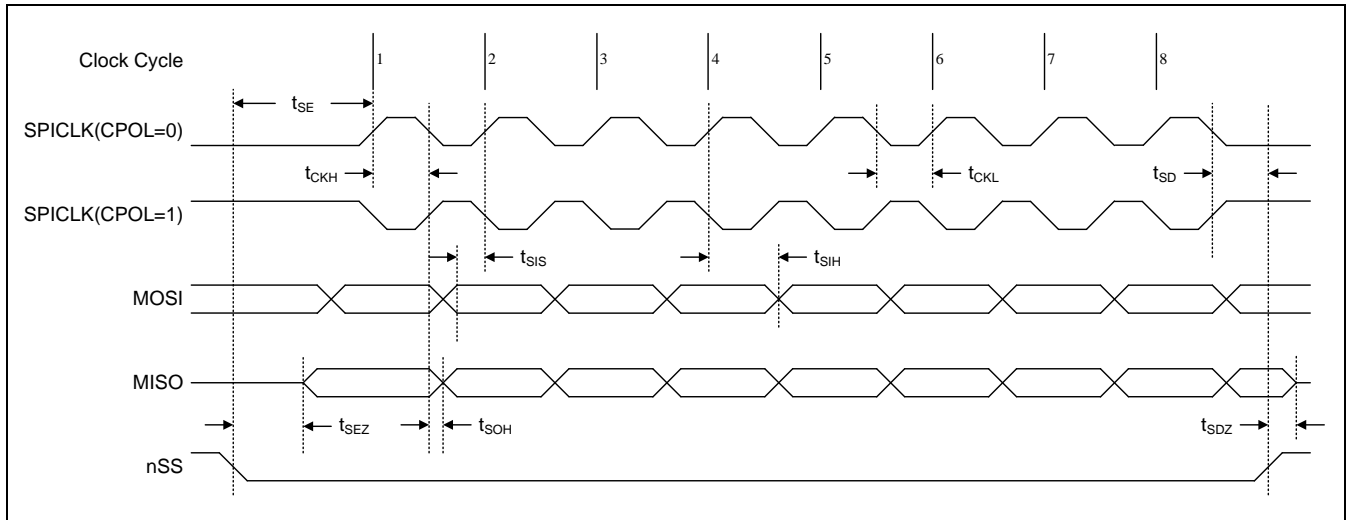
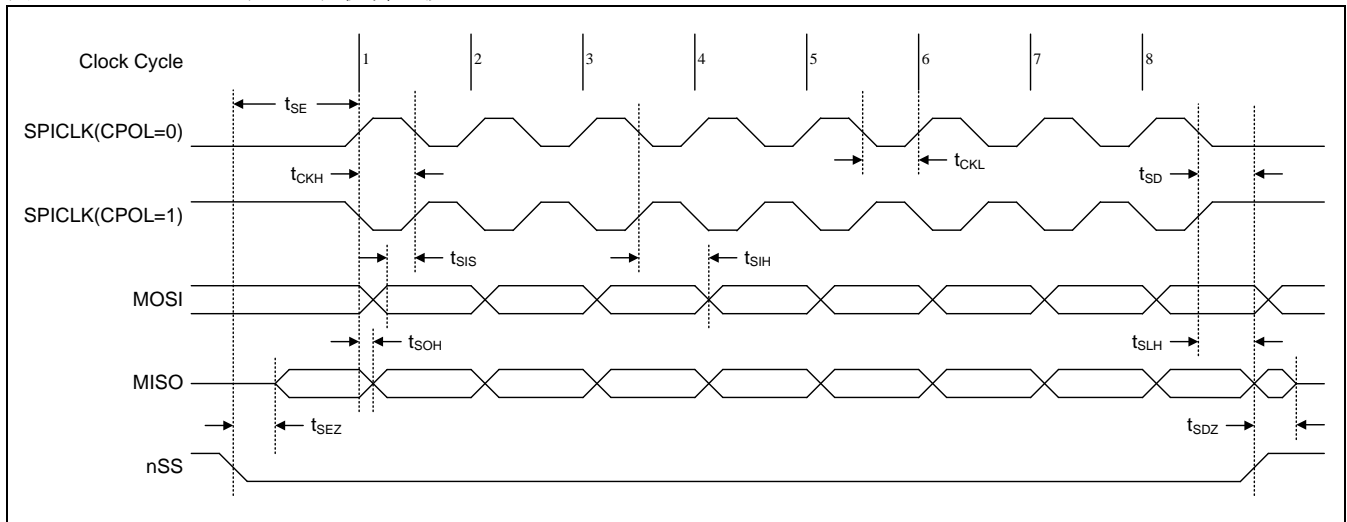


图 37-6. CPHA=1 时 SPI 从机传送波



## 37.13. EMB 时序特性

在运行条件下，所有输出负载电容为30pF。  $T_A = -40^{\circ}\text{C}$  to  $+105^{\circ}\text{C}$ ,  $V_{DD}=3.0\text{V}\sim 5.5\text{V}$ ,  $V_{SS}=0\text{V}$

T: 时钟周期

N: 读/写脉冲宽度扩展的时钟数,  $N = 0\text{T} \sim 7\text{T}$

L: 读/写脉冲设置/保持扩展的时钟数,  $L = 0\text{T} \sim 1\text{T}$

标号	参数	最小	最大	单位
1/t <sub>CLCL</sub>	CPUCLK 频率 (CPU MOVX 周期)		36	MHz
	SYSCLK频率 (DMA MOVX 周期)		48	
t <sub>RLRH</sub>	nRD 脉冲宽度	T+N-10		ns
t <sub>WLWH</sub>	nWR 脉冲宽度	T+N-10		ns
t <sub>DVRH</sub>	有效数据输入到nRD为高		10	ns
t <sub>RHDX</sub>	nRD之后数据保持	0		ns
t <sub>RHDZ</sub>	nRD之后数据浮空		10	ns
t <sub>WHQX</sub>	nWR之后数据保持	T+L-10		ns
t <sub>QVWH</sub>	数据有效到nWR 为高	2T+L+N-10		ns
t <sub>QVWX</sub>	数据有效到nWR 高到低转变	T+L-10		ns

注:数据为设计保证, 非量产测试.

**符号解释** 每个计时符号有5个字符。第一个字母总是“t”(代表时间)。其他字符, 取决于它们的位置, 表示信号的名称或该信号的逻辑状态。下面是所有字符及其代表的内容的列表。

A: 地址

Q: 输出数据

C: 时钟

R: nRD 信号

D: 输入数据

t: 时间

H: 逻辑高电平Logic level HIGH

V: 有效

L: 逻辑低电平或ALE

W: nWR 信号

X: 不再是有效的逻辑电平

Z: 高阻 (浮空)

例如:

t<sub>RLRH</sub> = nRD 脉冲宽度

图 37-7. EMB读周期(无地址相位)

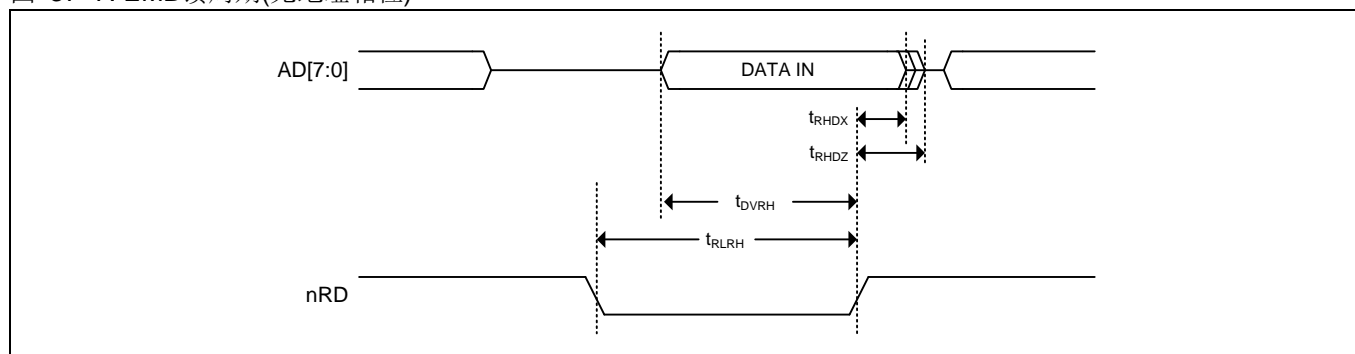
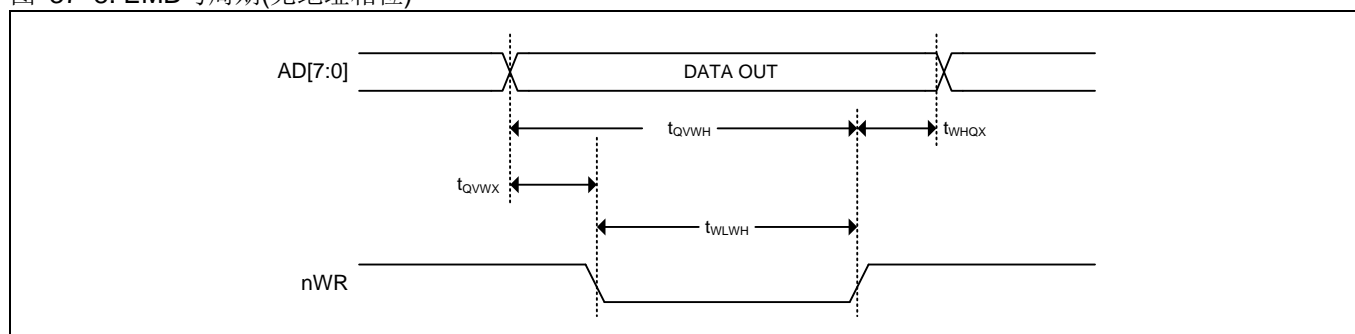


图 37-8. EMB写周期(无地址相位)



## 38. 指令集

表 38.1. 指令集

助记符	描述	字节	执行周期
<b>数据传送</b>			
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	1	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,#data	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	内置RAM(8位地址)的数据送入累加器中	1	3
MOVX A,@DPTR	寄存器Ri指向扩展RAM地址(8位地址)中的内容送到ACC中	1	3
MOVX @Ri,A	数据指针指向扩展RAM地址(16位地址)中的内容送到ACC中	1	3
MOVX @DPTR,A	累加器中的内容送到寄存器Ri指向的扩展RAM地址(8位地址)中	1	3
MOVX A,@Ri	累加器中的内容送到寄存器Ri指向的扩展RAM地址(16位地址)中	1	3~
MOVX A,@DPTR	外部RAM(16位地址)的数据送入累加器中	1	3~
MOVX @Ri,A	寄存器Ri指向片外RAM地址中的内容送到ACC中	1	3~
MOVX @DPTR,A	数据指针指向片外RAM地址(16位地址)中内容送到ACC中	1	3~
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
<b>算术运算</b>			
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
SUBB A,#data	累加器与立即数、连同借位位相减, 结果存在累加器中	2	2
INC A	累加器中的内容加1	1	2

助记符	描述	字节	执行周期
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
DAA	ACC十进制调整	1	4
<b>逻辑运算</b>			
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
<b>位逻辑运算</b>			
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
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

助记符	描述	字节	执行周期
MOV bit,C	直接地址位数据送入进位位	2	4
<b>位逻辑跳转</b>			
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
<b>程序跳转</b>			
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

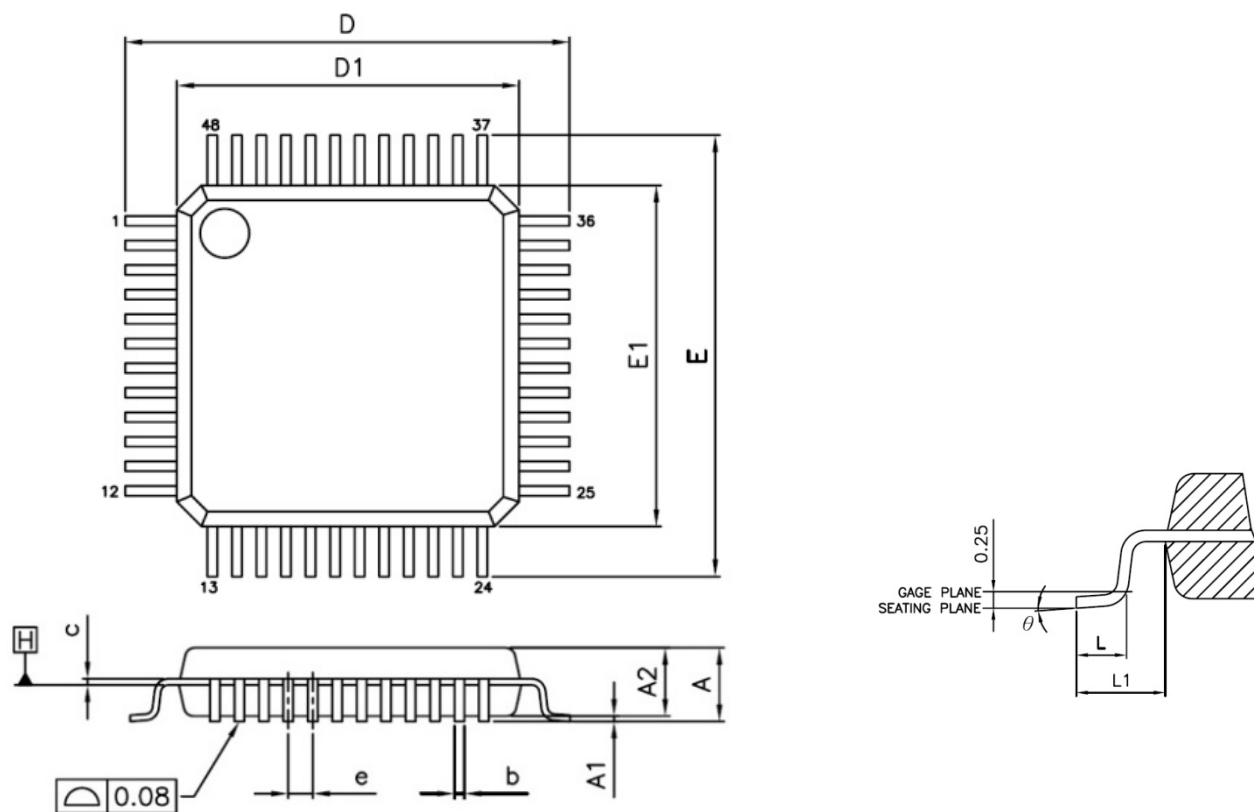
注 1:访问外部扩展RAM的周期时间是:

$$EMAI1 = 1: 3 + RW\_Stretch + 2 \times RWSH; (3\sim 12)$$

## 39. 封装尺寸

## 39.1. LQFP-48 (7mm X 7mm) 尺寸

图 39-1. LQFP-48 (7mm X 7mm) 封装尺寸



单位	mm			inch		
	最小	一般	最大	最小	一般	最大
A	---	---	1.60	---	---	0.062
A1	0.05	---	0.15	0.001	---	0.005
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.17	0.22	0.27	0.006	0.008	0.010
c	0.09	---	0.20	0.003	---	0.007
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	0.50 BSC			0.019 BSC		
L	0.45	0.60	0.75	0.017	0.023	0.029
L1	1.00 REF			0.039 REF		
θ	0°	3.5°	7°	0°	3.5°	7°

## 40. 版本历史

版本	描述	日期
V0.10	1. 初始版本	2023/07/07
V0.11	1. 增加 AEC-Q100 信息	2023/07/11
V0.12	1. 修改 VDD 工作范围到 2.4V – 5.5V 2. 增加 IHRCO/ILRCO 公差 3. 修改 ADC 特性	2023/08/14

## 41. 免责声明

在此，笙泉(Megawin)代表 “*Megawin Technology Co., Ltd.*”

**生命支援**—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。