CMS80F752x 用户手册

增强型闪存1T 8051单片机

Rev. 0.5.6

请注意以下有关CMS知识产权政策

＊中微半导体（深圳）股份有限公司（以下简称本公司）已申请了专利，享有绝对的合法权益。与本公司MCU或其他产品有关的专利权并未被同意授权使用，任何经由不当手段侵害本公司专利权的公司、组织或个人，本公司将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨本公司因侵权行为所受的损失、或侵权者所得的不法利益。

＊中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

＊本公司保留对规格书中产品在可靠性、功能和设计方面的改进作进一步说明的权利。然而本公司对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，本公司不保证和不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本公司的产品不授权适用于救生、维生器件或系统中作为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考官方网站www.mcu.com.cn。

目录

1. 中央处理器（CPU）.....10
1.1 复位向量（0000H）.....10
1.2 累加器（ACC）.....10
1.3 B 寄存器（B）.....10
1.4 堆栈指针寄存器（SP）.....10
1.5 数据指针寄存器（DPTR0）.....11
1.6 数据指针选择寄存器（DPS）.....11
1.7 程序状态寄存器（PSW）.....12
1.8 程序计数器（PC）.....12
1.9 时序存取寄存器（TA）.....12
2. 存储器和寄存器映射.....13
2.1 程序储存器 APROM.....13
2.2 非易失性数据存储器 Data FLASH.....14
2.3 通用数据存储器 RAM.....15
2.4 通用外部数据寄存器 XRAM.....17
2.5 特殊功能寄存器 SFR.....18
2.6 外部特殊功能寄存器 XSFR.....19
3. 复位.....23
3.1 上电复位.....24
3.2 外部复位.....25
3.3 POR 复位.....25
3.4 LVD 低压检测复位/中断.....26
3.5 看门狗复位.....27
3.6 软件复位.....28
3.7 复位控制寄存器（RSTCF）.....28
4. 时钟结构.....29
4.1 系统时钟结构.....29
4.2 相关寄存器.....30
4.2.1 外设时钟源选择寄存器 CKCON.....30
4.2.2 系统时钟监控寄存器 XTSCM.....31
4.2.3 模块时钟允许寄存器 PCGEN0.....32
4.2.4 模块时钟允许寄存器 PCGEN1.....33
4.3 系统时钟监控.....34
5. 电源管理.....35
5.1 电源管理寄存器.....35
5.1.1 电源管理寄存器 PCON.....35
5.1.2 LVD 低压检测中断控制寄存器 LVDCON.....36
5.1.3 LVD 低压检测中断配置寄存器 LVDEICFG.....36
5.1.4 LVD 低压检测中断电压选择寄存器 LVDS.....37
5.2 IDLE 空闲模式.....38
5.3 STOP 休眠模式.....38
5.3.1 休眠唤醒.....38
5.3.2 唤醒等待状态.....38
5.3.3 休眠唤醒时间.....38

5.3.4 休眠下复位操作 39
5.3.5 调试模式下的休眠功耗 39
5.3.6 休眠模式应用举例 39
6. 中断 40
6.1 中断概述 40
6.2 中断重映射 41
6.3 外部中断 41
6.3.1 INT0/INT1 中断 41
6.3.2 外部扩展中断 41
6.3.3 外部中断滤波 41
6.4 中断与休眠唤醒 42
6.5 中断寄存器 43
6.5.1 中断偏移基地址寄存器 IREMAP 43
6.5.2 中断屏蔽寄存器 44
6.5.3 中断优先级控制寄存器 47
6.5.4 中断标志位寄存器 50
6.5.5 中断标志位的清除操作 54
6.5.6 外部中断滤波控制寄存器 54
6.5.7 调试模式下特殊中断标志位 55
7. I/O 端口 56
7.1 端口结构 56
7.2 GPIO 功能 56
7.2.1 PORTx 数字功能控制寄存器 PxDIG 56
7.2.2 PORTx 方向寄存器 PxTRIS 57
7.2.3 PORTx 数据寄存器 Px 57
7.2.4 PORTx 上拉电阻寄存器 PxUP 57
7.2.5 PORTx 下拉电阻寄存器 PxRD 57
7.2.6 PORT x 开漏控制寄存器 PxOD 58
7.3 复用功能 59
7.3.1 端口复用功能表 59
7.3.2 端口复用功能配置 60
7.3.3 端口模拟功能配置 73
7.3.4 端口外部中断配置 73
7.3.5 复用功能应用注意 75
8. 看门狗定时器（WDT） 76
8.1 概述 76
8.1.1 看门狗控制寄存器 WDCON 76
8.1.2 看门狗溢出控制寄存器 WDKEY 77
9. 定时计数器 0/1（Timer0/1） 78
9.1 概述 78
9.2 相关寄存器 79
9.2.1 Timer0/1 模式寄存器 TMOD 79
9.2.2 Timer0/1 控制寄存器 TCON 80
9.2.3 Timer0 数据寄存器低位 TL0 81
9.2.4 Timer0 数据寄存器高位 TH0 81
9.2.5 Timer1 数据寄存器低位 TL1 81

9.2.6 Timer1 数据寄存器高位 TH1 81
9.3 Timer0/1 中断 82
9.3.1 中断屏蔽寄存器 IE 82
9.3.2 中断优先级控制寄存器 IP 83
9.3.3 Timer0/1、INT0/1 中断标志位寄存器 TCON 84
9.4 Timer0 工作模式 85
9.4.1 T0 -模式 0（13 位定时/计数模式） 85
9.4.2 T0 -模式 1（16 位定时/计数模式） 85
9.4.3 T0 -模式 2（8 位自动重装定时/计数模式） 86
9.4.4 T0 -模式 3（两个单独的 8 位定时/计数器） 86
9.5 Timer1 工作模式 87
9.5.1 T1 -模式 0（13 位定时/计数模式） 87
9.5.2 T1 -模式 1（16 位定时/计数模式） 87
9.5.3 T1 -模式 2（8 位自动重装定时/计数模式） 88
9.5.4 T1 -模式 3（停止计数） 88
10. 定时计数器 2（Timer2） 89
10.1 概述 89
10.2 相关寄存器 90
10.2.1 Timer2 控制寄存器 T2CON 90
10.2.2 Timer2 数据寄存器低位 TL2 90
10.2.3 Timer2 数据寄存器高位 TH2 91
10.2.4 Timer2 比较/捕获/自动重装寄存器低位 RLDL 91
10.2.5 Timer2 比较/捕获/自动重装寄存器高位 RLDH 91
10.2.6 Timer2 比较/捕获通道 1 寄存器低位 CCL1 91
10.2.7 Timer2 比较/捕获通道 1 寄存器高位 CCH1 91
10.2.8 Timer2 比较捕获控制寄存器 CCEN 92
10.3 Timer2 中断 93
10.3.1 中断相关寄存器 93
10.3.2 定时中断 96
10.3.3 外部触发中断 96
10.3.4 比较中断 96
10.3.5 捕获中断 96
10.4 Timer2 功能描述 97
10.4.1 定时模式 97
10.4.2 重装模式 97
10.4.3 门控定时模式 98
10.4.4 事件计数模式 98
10.4.5 比较模式 98
10.4.6 捕获模式 100
11. 定时器 5（Timer5） 103
11.1 相关寄存器 103
11.1.1 T5CON 寄存器 103
11.1.2 TL5 数据寄存器低位 103
11.1.3 TH5 数据寄存器高位 104
11.2 休眠与定时唤醒 105
11.2.1 中断屏蔽寄存器 EIE2 105
11.2.2 外设中断标志位寄存器 EIF2 106
11.2.3 中断优先级控制寄存器 EIP2 106

11.3 功能描述 107
12. 波特率定时器（BRT） 108
12.1 概述 108
12.2 相关寄存器 108
12.2.1 BRT 定时器数据加载低 8 位寄存器 BRTDL 108
12.2.2 BRT 定时器数据加载高 2 位寄存器 BRTDH 108
12.3 功能描述 108
13. PWM 模块 109
13.1 概述 109
13.2 特性 109
13.3 端口配置 109
13.4 功能描述 110
13.4.1 功能框图 110
13.4.2 边沿对齐 111
13.4.3 互补模式 112
13.4.4 同步模式 113
13.5 PWM 相关寄存器 114
13.5.1 PWM 控制寄存器 PWMCON 114
13.5.2 PWM 输出使能控制寄存器 PWMOE 114
13.5.3 PWM 时钟分频控制寄存器 PWMnDIV(n=0-2) 115
13.5.4 PWM 数据加载使能控制寄存器 PWMLOADEN 115
13.5.5 PWM 输出极性控制寄存器 PWMPINV 115
13.5.6 PWM 计数器模式控制寄存器 PWMCNTM 116
13.5.7 PWM 计数器使能控制寄存器 PWMCNTE 116
13.5.8 PWM 计数器模式控制寄存器 PWMCNTCLR 116
13.5.9 PWM 周期数据寄存器低 8 位 PWMPnL (n=0-2) 117
13.5.10 PWM 周期数据寄存器高 8 位 PWMPnH (n=0-2) 117
13.5.11 PWM 比较数据寄存器低 8 位 PWMDnL (n=0-5) 117
13.5.12 PWM 比较数据寄存器高 8 位 PWMDnH (n=0-5) 117
13.5.13 PWM 死区使能控制寄存器 PWMDTE 118
13.5.14 PWM0/1 死区时间设置寄存器 PWM01DT 118
13.5.15 PWM2/3 死区时间设置寄存器 PWM23DT 118
13.5.16 PWM4/5 死区时间设置寄存器 PWM45DT 118
13.6 PWM 中断 119
13.6.1 中断屏蔽寄存器 EIE2 119
13.6.2 中断优先级控制寄存器 EIP2 120
13.6.3 PWM 零点中断屏蔽寄存器 PWMZIE 120
13.6.4 PWM 向下比较中断屏蔽寄存器 PWMDIE 121
13.6.5 PWM 零点中断标志寄存器 PWMZIF 121
13.6.6 PWM 向下比较中断标志寄存器 PWMDIF 121
14. 硬件 LED 矩阵驱动器 122
14.1 概述 122
14.2 特性 122
14.3 相关寄存器 123
14.3.1 LED 驱动模式选择寄存器 LEDMODE 123
14.3.2 LED 控制寄存器 LEDCON 123
14.3.3 LED 时钟选择寄存器 LEDCKS 124

14.3.4 COM 口有效时间选择寄存器 LEDCOMTIME 124
14.3.5 调光时间选择寄存器 LEDSEGTIME 124
14.3.6 COM 口使能控制寄存器 LEDCOMEN 125
14.3.7 SEG 口使能控制寄存器 LEDSEGEN0 125
14.3.8 SEG 口使能控制寄存器 LEDSEGEN1 125
14.3.9 SEG 口使能控制寄存器 LEDSEGEN2 125
14.3.10 SEG 口使能控制寄存器 LEDSEGEN3 126
14.3.11 COM0 对应 SEG 数据寄存器 LEDC0DATAn（n=0-3） 126
14.3.12 COM1 对应 SEG 数据寄存器 LEDC1DATAn（n=0-3） 126
14.3.13 COM2 对应 SEG 数据寄存器 LEDC2DATAn（n=0-3） 126
14.3.14 COM3 对应 SEG 数据寄存器 LEDC3DATAn（n=0-3） 127
14.3.15 COM4 对应 SEG 数据寄存器 LEDC4DATAn（n=0-3） 127
14.3.16 COM5 对应 SEG 数据寄存器 LEDC5DATAn（n=0-3） 127
14.3.17 COM6 对应 SEG 数据寄存器 LEDC6DATAn（n=0-3） 127
14.3.18 COM7 对应 SEG 数据寄存器 LEDC7DATAn（n=0-3） 128
14.3.19 LED 矩阵驱动端口电流控制寄存器 LEDDRV 128
14.3.20 LED COM 口驱动电流选择寄存器 LEDCOMDRV 128
14.3.21 管脚驱动使能寄存器 LEDSEGDR0 129
14.3.22 管脚驱动使能寄存器 LEDSEGDR1 130
14.3.23 管脚驱动使能寄存器 LEDSEGDR2 131
14.3.24 管脚驱动使能寄存器 LEDSEGDR3 132
14.4 LED 驱动器输出波形 133

15. 硬件 LED 点阵驱动器 134

15.1 概述 134
15.2 特性 134
15.3 功能描述 135
15.3.1 点阵 LED 周期选择 140
15.4 相关寄存器 141
15.4.1 LED 驱动模式选择寄存器 LEDMODE 141
15.4.2 LED 点阵驱动控制器 LEDCON1 141
15.4.3 LED 时钟选择寄存器 LEDCKS 142
15.4.4 LED 端口使能控制寄存器 LEDIOEN 142
15.4.5 LED 点阵驱动第一段周期配置寄存器 SCAN1W 142
15.4.6 LED 点阵驱动第二段周期配置寄存器 SCAN2W 143
15.4.7 LED 点阵驱动显示数据寄存器 LEDnDATA (n=0-7) 143
15.4.8 LED 点阵驱动周期选择寄存器 LEDnSEL (n=0-7) 144
15.4.9 LED 点阵驱动管脚电流控制寄存器 LEDDRV 144
15.4.10 管脚驱动使能寄存器 LEDSEGDR0 145
15.4.11 管脚驱动使能寄存器 LEDSEGDR3 145
15.4.12 LED 点阵引脚映射寄存器 LEDRESEQ 146
15.5 LED 点阵驱动中断 147
15.5.1 LED 点阵驱动状态寄存器 LEDSTATUS 147
15.5.2 中断优先级控制寄存器 EIP3 147

16. SPI 模块 148

16.1 概述 148
16.2 特性 148
16.3 SPI 端口配置 149
16.4 SPI 硬件描述 150

16.5 SPI 相关寄存器 151
16.5.1 SPI 控制寄存器 SPCR 151
16.5.2 SPI 数据寄存器 SPDR 151
16.5.3 SPI 从器件选择控制寄存器 SSCR 152
16.5.4 SPI 状态寄存器 SPSR 152
16.6 SPI 主控模式 153
16.6.1 写冲突错误 154
16.7 SPI 从动模式 155
16.7.1 被寻址错误 155
16.7.2 写冲突错误 156
16.8 SPI 时钟控制逻辑 157
16.8.1 SPI 时钟相位与极性控制 157
16.8.2 SPI 传输格式 157
16.8.3 CPHA=0 传输格式 157
16.8.4 CPHA=1 传输格式 158
16.9 SPI 数据传输 159
16.9.1 SPI 传输启动 159
16.9.2 SPI 传输结束 159
16.10 SPI 时序图 160
16.10.1 主控模式传输 160
16.10.2 从动模式传输 160
16.11 SPI 中断 161
16.11.1 中断屏蔽寄存器 EIE2 161
16.11.2 中断优先级控制寄存器 EIP2 162
16.11.3 外设中断标志位寄存器 EIF2 163

17. I²C 串行接口控制器（I²C） 164

17.1 概述 164
17.2 特性 164
17.3 寄存器映射 164
17.4 I2C 传输速度模式说明 165
17.5 寄存器说明 166
17.5.1 I²C 控制寄存器（I2CCON） 166
17.5.2 I²C 清零寄存器（I2CCLR） 167
17.5.3 I²C 状态寄存器（I2CSTAT） 168
17.5.4 I²C 数据寄存器（I2CDAT） 169
17.5.5 I²C 时钟控制寄存器（I2CCLK） 169
17.5.6 I²C 从机地址寄存器（I2CADR） 169
17.5.7 I²C 扩展地址复位寄存器（I2CXAR） 169

18. UARTn 模块 170

18.1 概述 170
18.2 UARTn 端口配置 170
18.3 UARTn 波特率 171
18.3.1 波特率时钟源 171
18.3.2 波特率计算 172
18.3.3 波特率误差 172
18.4 UARTn 模式 173
18.4.1 模式 1-8 位异步模式（可变波特率） 173
18.4.2 模式 3-9 位异步模式（可变波特率） 174

18.5 UARTn 寄存器 175
18.5.1 UARTn 缓冲寄存器 SBUFn 175
18.5.2 UART 控制寄存器 SCONn 176
18.6 UARTn 中断 177
18.6.1 中断屏蔽寄存器 IE 177
18.6.2 中断优先级控制寄存器 IP 178
19. 模数转换器（ADC） 179
19.1 概述 179
19.2 ADC 配置 180
19.2.1 端口配置 180
19.2.2 通道选择 180
19.2.3 ADC 参考电压 180
19.2.4 转换时钟 180
19.3 ADC 工作原理 181
19.3.1 启动转换 181
19.3.2 完成转换 181
19.3.3 终止转换 181
19.3.4 A/D 转换步骤 182
19.3.5 转换过程中进入休眠 182
19.4 相关寄存器 183
19.4.1 AD 控制寄存器 ADCON0 183
19.4.2 AD 控制寄存器 ADCON1 184
19.4.3 AD 数据寄存器高位 ADRESH 184
19.4.4 AD 数据寄存器低位 ADRESL 185
19.4.5 AD 参考电压控制寄存器 ADCLDO 185
19.5 ADC 中断 186
19.5.1 中断屏蔽寄存器 EIE2 186
19.5.2 中断优先级控制寄存器 EIP2 187
19.5.3 外设中断标志位寄存器 EIF2 188
20. 循环冗余校验单元（CRC） 189
20.1 概述 189
20.2 相关寄存器 189
20.2.1 CRC 数据输入寄存器 CRCIN 189
20.2.2 CRC 运算结果低 8 位数据寄存器 CRCDL 189
20.2.3 CRC 运算结果高 8 位数据寄存器 CRCDH 189
20.3 功能描述 190
21. 触摸模块 191
21.1 触摸模块使用注意事项 191
22. 存储器管理控制器（MMC/FMC） 192
22.1 概述 192
22.2 相关寄存器 193
22.2.1 存储器保护锁定寄存器 MLOCK 193
22.2.2 存储器状态寄存器 MSTATUS 193
22.2.3 存储器数据寄存器 MDATA 194
22.2.4 存储器地址寄存器 MADRL 194
22.2.5 存储器地址寄存器 MADRH 194

22.2.6 存储器区域控制寄存器 MREGION 195
22.2.7 存储器模式控制寄存器 MMODE 195
22.3 功能描述 196
22.3.1 读操作（单字节读取） 196
22.3.2 多字节读操作（128 字节，读取到缓存） 197
22.3.3 程序区读操作（MOVC 指令） 197
22.3.4 单字节写操作 197
22.3.5 多字节写操作（1~128 字节，先写入缓存） 198
22.3.6 擦除操作（扇区擦除） 199
22.3.7 存储器 CRC 校验 200
22.3.8 Information 信息 200
22.3.9 唯一 ID（UID） 200
23. 用户配置 201
24. 在线编程与调试 203
24.1 在线编程模式 203
24.2 在线调试模式 204
25. 指令说明 205
25.1 符号说明 205
25.2 指令一览表 206
26. 版本修订说明 209

1. 中央处理器（CPU）

该系列是 8 位 8051 框架结构的微控制器。CPU 是微控制器内部的核心部件，由运算器、控制器和专用寄存器组等组成。运算器模块主要实现数据的算术逻辑运算、位变量处理和数据传送操作；控制器模块主要对指令进行译码，然后发出各种控制信号；专用寄存器组主要用于表示当前要执行指令的内存地址、存放操作数和指示指令执行后的状态等。专用寄存器组主要包括累加器 ACC、通用寄存器 B、堆栈指针 SP、数据指针 DPTR(数据指针为单 DPTR)、程序状态寄存器 PSW、程序计数器 PC 等。

1.1 复位向量（0000H）

微控制器具有一个字长的系统复位向量（0000H），发生复位后，程序将从 0000H 处重新开始执行，系统寄存器也都将恢复为默认值。下面一段程序演示了如何定义 FLASH 中的复位向量。

例：定义复位向量

```
        ORG     0000H       ;系统复位向量
        LJMP    START
        ORG     0010H       ;用户程序起始
START:
        …                   ;用户程序
        …
        END                 ;程序结束
```

1.2 累加器（ACC）

ALU 是 8Bit 宽的算术逻辑单元，MCU 所有的数学、逻辑运算均通过它来完成。它可以对数据进行加、减、移位及逻辑运算；ALU 也控制状态位（PSW 状态寄存器中），用来表示运算结果的状态。

ACC 寄存器是一个 8Bit 的寄存器，ALU 的运算结果可以存放在此。

1.3 B 寄存器（B）

B 寄存器在使用乘法和除法指令时使用。如不使用乘除法指令，也可作为通用寄存器使用。

1.4 堆栈指针寄存器（SP）

SP 寄存器指向堆栈的地址，复位后默认值为 0x07，意味着堆栈的区域从 RAM 地址的 08H 开始。该 SP 的值可以修改，如果将堆栈区域设置为 0xC0 开始，则系统复位后需要将 SP 的值设置为 0xBF。

影响 SP 的操作有：指令 PUSH、LCALL、ACALL、POP、RET、RETI 以及进入中断。

PUSH 指令占用堆栈中一个字节，LCALL，ACALL 及中断占用堆栈中两个字节，POP 指令释放一个字节，RET/RETI 指令释放两个字节。

使用 PUSH 指令会将被操作的寄存器的当前值自动保存到 RAM 中。

1.5 数据指针寄存器（DPTR0）

数据指针主要用在 MOVX，MOVC 指令中，其作用是定位 XRAM 与 ROM 的地址。芯片内部有 1 个数据指针寄存器 DPTR0。每组指针包括 2 个 8 位寄存器：DPTR0={DPH0,DPL0}。

例如操作 XRAM 的汇编代码如下：

```
MOV     DPTR,#0001H
MOV     A,#5AH
MOVX    @DPTR,A          ;将 A 中的数据写入 XRAM 地址 0001H 中
```

1.6 数据指针选择寄存器（DPS）

数据指针选择寄存器 DPS

0x86	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPS	ID1	ID0	--	AU	--	--	--	--
读写	R	R/W	R	R/W	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6　ID<1:0>:　自减/自加功能选择。
- 00=　DPTR0加1；
- 01=　DPTR0减1
- 其他　禁止设置。

Bit5　--　保留，须均为0。

Bit4　AU:　自加/减使能位；
- 1=　允许MOVX @DPTR或者MOVC @DPTR指令运行后，执行自减/自加的操作（由ID1-ID0决定）。
- 0=　DPTR相关指令不影响DPTR本身。

Bit3~Bit0　--　保留，须均为0。

1.7 程序状态寄存器（PSW）

程序状态寄存器 PSW

0xD0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	F0	RS1	RS0	OV	--	P
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
复位值	0	0	0	0	0	0	0	0

Bit7 CY: 进位标志位；
1= 有进位；
0= 无进位。
Bit6 AC: 辅助进位标志位（半进位标志位）；
1= 有进位；
0= 无进位。
Bit5 F0: 通用标志位。
Bit4~Bit3 RS<1:0> 工作寄存器BANK选择位
00= 选择Bank0
01= 选择Bank1
10= 选择Bank2
11= 选择Bank3
Bit2 OV: 溢出标志位；
1= 算术或逻辑运算有溢出；
0= 算术或逻辑运算无溢出。
Bit1 -- 保留，须为0。
Bit0 P: 校验位；
1= 结果的最高位发生了进位。
0= 结果的最高位没有发生进位。

1.8 程序计数器（PC）

程序计数器（PC）控制程序内存 FLASH 中的指令执行顺序，它可以寻址整个 FLASH 的范围，取得指令码后，程序计数器（PC）会自动加一，指向下一个指令码的地址。但如果执行跳转、条件跳转、子程序调用、初始化复位、中断、中断返回、子程序返回等操作时，PC 会加载与指令相关的地址而不是下一条指令的地址。

当遇到条件跳转指令且符合跳转条件时，当前指令执行过程中读取的下一条指令将会被丢弃，且会插入一个空指令操作周期，随后才能取得正确的指令。反之，就会顺序执行下一条指令。

1.9 时序存取寄存器（TA）

时序存取寄存器 TA

0x96	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TA	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TA<7:0>: 时序存取控制位。
某些被保护的寄存器必须在对TA进行如下操作之前才能写入。

```
MOV    TA, #0AAH
MOV    TA, #055H
```

中间不能插入其他任何指令，再次修改时需要重新执行此序列。
被保护的寄存器: WDCON，IREMAP,WDKEY

2. 存储器和寄存器映射

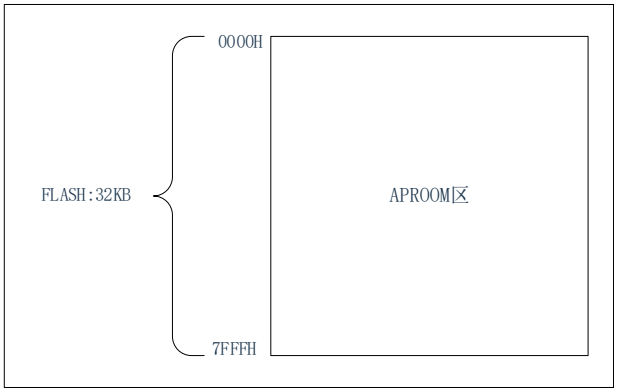
该系列微控制具有如下几种存储器：

- 最大 32KB 的 FLASH 程序存储器（APROM 区）。
- 最大 1KB 的非易失性数据存储器（Data FLASH）。
- 最大 256B 的通用内部数据存储器（RAM）。
- 最大 2KB 的通用外部数据存储器（XRAM）。
- 特殊功能寄存器 SFR。
- 外部特殊功能寄存器 XSFR。

2.1 程序储存器 APROM

程序存储器 APROM 用来存放源程序和表格数据，且以程序计数器 PC 作地址指针。PC 为 16 位程序计数器，因此可以寻址到的地址空间为 32KB。

FLASH 空间分配结构框图如下图所示：

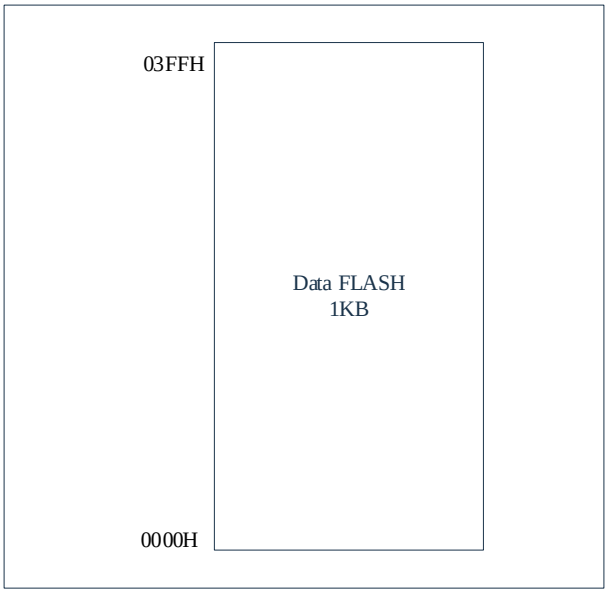


芯片复位后，CPU 从 0000H 开始执行。每个中断在程序存储器中都分配有固定的地址，中断使 CPU 跳到该地址开始执行服务程序。

例如外部中断 1，被分配地址为 0013H，如果使用外部中断 1，它的服务程序必须从 0013H 位置开始。如果没有用到该中断，其服务地址作为普通程序存储地址使用。

2.2 非易失性数据存储器 Data FLASH

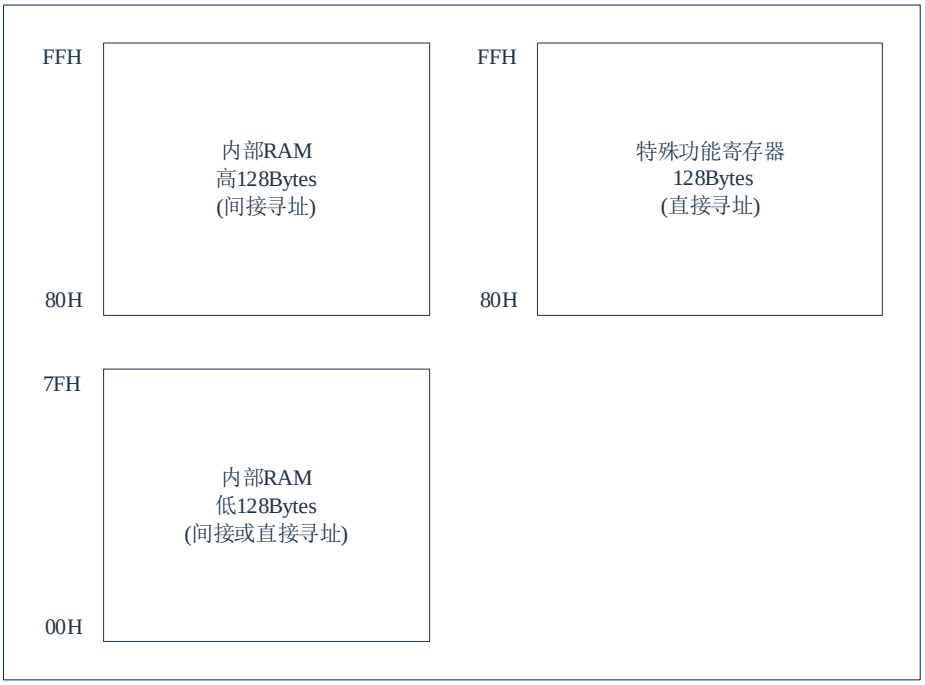
非易失性数据存储器 Data FLASH 可用于存放常量数据、校准数据、防护安全相关信息等重要数据。存储在该区域的数据具有在芯片断电或者突然性、意外性断电时，数据不会丢失的特性。Data FLASH 空间分配结构框图如下图所示：



Data FLASH 存储器的读、写、擦除操作通过 FLASH 控制接口实现。

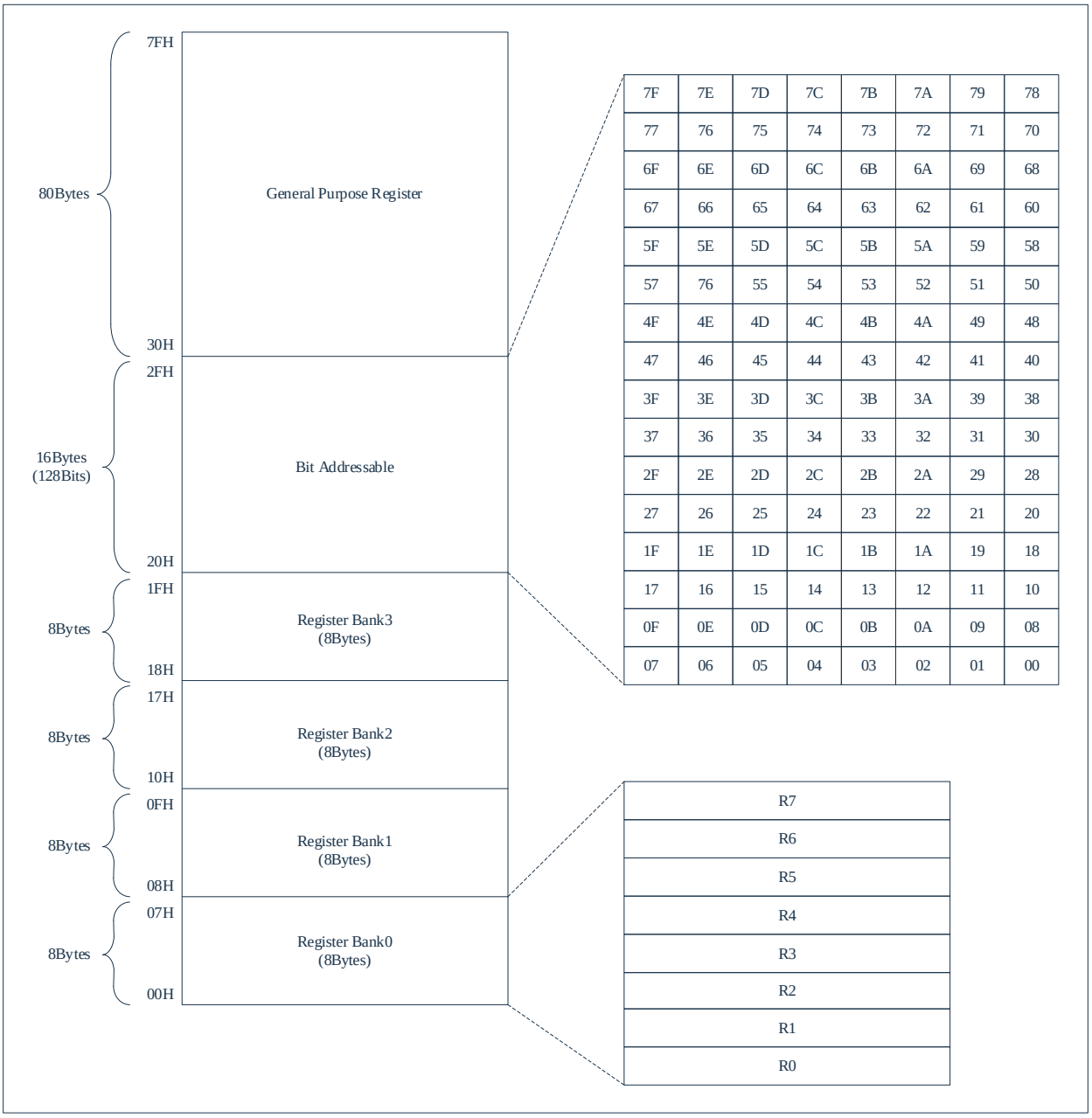
2.3 通用数据存储器 RAM

内部数据存储器分为 3 个部分：低 128Bytes、高 128Bytes、特殊功能寄存器 SFR。RAM 空间分配结构框图如下图所示：



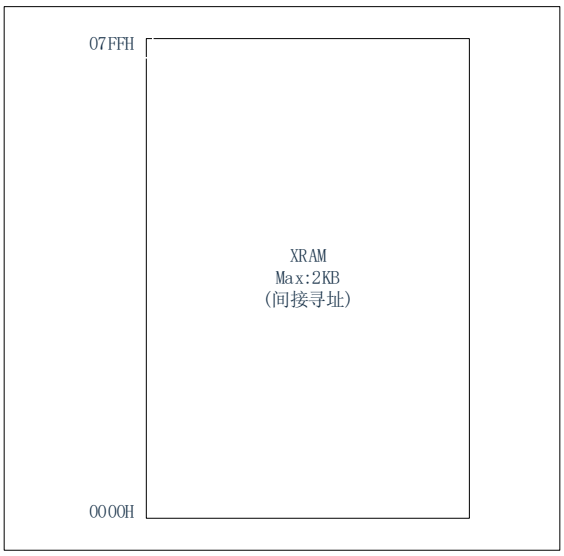
上图所示的高 128Bytes 和 SFR 占用相同的区域（80H~FFH），但它们本身却是独立的。直接寻址高于 7FH 的存储空间（SFR）和间接寻址高于 7FH（高 128Bytes）的存储空间进入到不同的存储空间。

上图所示的低 128Bytes 空间寄存器分配如下图所示。最低的 32 字节（00H~1FH）组成了 4 个寄存器组，每组 8 个存储单元，以 R0~R7 作为单元编号，用于保存操作数及中间结果等。复位后，默认选择 0 组，如果选择其他寄存器组，需通过改变程序状态来决定。寄存器组后边的 16Bytes（20H~2FH）组成了可位寻址的存储空间，该区域的 RAM 单元既可以按字节操作，也可以对单元中的每一位直接位操作。剩余的 80 个存储单元（30H~7FH），用户可设置堆栈区和存储中间数据。


7FH
30H
80Bytes
General Purpose Register
2FH
20H
16Bytes
(128Bits)
Bit Addressable
1FH
18H
8Bytes
Register Bank3
(8Bytes)
17H
10H
8Bytes
Register Bank2
(8Bytes)
0FH
08H
8Bytes
Register Bank1
(8Bytes)
07H
00H
8Bytes
Register Bank0
(8Bytes)
7F 7E 7D 7C 7B 7A 79 78
77 76 75 74 73 72 71 70
6F 6E 6D 6C 6B 6A 69 68
67 66 65 64 63 62 61 60
5F 5E 5D 5C 5B 5A 59 58
57 76 55 54 53 52 51 50
4F 4E 4D 4C 4B 4A 49 48
47 46 45 44 43 42 41 40
3F 3E 3D 3C 3B 3A 39 38
37 36 35 34 33 32 31 30
2F 2E 2D 2C 2B 2A 29 28
27 26 25 24 23 22 21 20
1F 1E 1D 1C 1B 1A 19 18
17 16 15 14 13 12 11 10
0F 0E 0D 0C 0B 0A 09 08
07 06 05 04 03 02 01 00
R7
R6
R5
R4
R3
R2
R1
R0

2.4 通用外部数据寄存器 XRAM

芯片内部有最大 2K XRAM 区域，该区域与 FLASH/RAM 没有联系，XRAM 空间分配结构框图如下图所示：



XRAM/XSFR 空间访问通过 DPTR0 数据指针操作，例如通过 MOVX 间接寻址操作，汇编代码如下：

```
MOV     R0,#01H
MOV     A,#5AH
MOVX    @R0,A          ;将 A 中的数据写入 XRAM 地址 01H 中，高 8 位地址由 DPH0 决定
```

在 Keil51 中将 Target-->Memory Model 设置为 Large 后，C 编译器将采用 XRAM 作为变量地址。一般用 DPTR 进行 XRAM/XSFR 的操作。

2.5 特殊功能寄存器 SFR

特殊功能寄存器是指有特殊用途的寄存器集合，本质上是一些具有特殊功能的片内 RAM 单元，离散地分布在地址范围 80H~FFH 内。用户可以通过直接寻址指令对它们进行字节存取，地址低四位为 0000 或 1000 的可进行位寻址，如 P0、TCON、P1。

寄存器表如下：

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	-	MREGION	MMODE	MDATA	MADRL	MADRH	MSTATUS	MLOCK
0xF0	B	I2CCON	I2CCLR	I2CSTAT	I2CDAT	I2CCLK	I2CADR	I2CXAR
0xE8	-	-	SCON1	SBUF1	SPCR	SPSR	SPDR	SSCR
0xE0	ACC	-	-	-	-	-	-	-
0xD8	P5	-	-	-	ADRESL	ADRESH	ADCON1	ADCON0
0xD0	PSW	-	-	TL5	TH5	T5CON	--	--
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2	CCEN	T2IE
0xC0	-	-	CCL1	CCH1	-	-	-	-
0xB8	IP	EIP1	EIP2	EIP3	-	-	-	-
0xB0	-	-	EIF2	-	EI0IF	-	-	-
0xA8	IE	-	EIE2	-	EI0IE	-	-	-
0xA0	P2	IREMAP	-	-	-	-	-	-
0x98	SCON0	SBUF0	P0TRIS	P1TRIS	P2TRIS	-	-	P5TRIS
0x90	P1	-	PCGEN0	PCGEN1	-	RSTCF	TA	WDCON
0x88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	WDKEY
0x80	P0	SP	DPL0	DPH0	-	-	DPS	PCON

2.6 外部特殊功能寄存器 XSFR

XSFR 是寻址空间与 XRAM 共用的特殊寄存器，主要包括：端口控制寄存器、其他功能控制寄存器。其寻址范围如下图所示：

外部特殊功能寄存器列表如下：

地址	寄存器	寄存器描述
F060H	P0DIG	P0 口模拟控制寄存器
F061H	P1DIG	P1 口模拟控制寄存器
F062H	P2DIG	P2 口模拟控制寄存器
--	--	--
F065H	P5DIG	P5 口模拟控制寄存器
--	--	--
F068H	P0UP	P0 口上拉电阻控制寄存器
F069H	P1UP	P1 口上拉电阻控制寄存器
F06AH	P2UP	P2 口上拉电阻控制寄存器
--	--	--
F06DH	P5UP	P5 口上拉电阻控制寄存器
--	--	--
F070H	P0OD	P0 口开漏输出控制寄存器
F071H	P1OD	P1 口开漏输出控制寄存器
F072H	P2OD	P2 口开漏输出控制寄存器
--	--	--
F075H	P5OD	P5 口开漏输出控制寄存器
--	--	--
F078H	P0RD	P0 口下拉电阻控制寄存器
F079H	P1RD	P1 口下拉电阻控制寄存器
F07AH	P2RD	P2 口下拉电阻控制寄存器
--	--	--
F07DH	P5RD	P5 口下拉电阻控制寄存器
--	--	--
F0C2H	PS_T0	Timer0 外部时钟输入端口分配寄存器
F0C3H	PS_T0G	Timer0 门控输入端口分配寄存器
F0C4H	PS_T1	Timer1 外部时钟输入端口分配寄存器
F0C5H	PS_T1G	Timer1 门控输入端口分配寄存器
F0C6H	PS_T2	Timer2 外部事件或门控输入端口分配寄存器
F0C7H	PS_T2EX	Timer2 下降沿自动重装输入端口分配寄存器
F0C8H	PS_EINT0	外部中断扩展输入 EINT0 通道 GPIO 选择寄存器
F0C9H	PS_EINT1	外部中断扩展输入 EINT1 通道 GPIO 选择寄存器
F0CAH	PS_EINT2	外部中断扩展输入 EINT2 通道 GPIO 选择寄存器
F0CBH	PS_EINT3	外部中断扩展输入 EINT3 通道 GPIO 选择寄存器
--	--	--
F0D0H	PS_TXD1	UART1 数据输出端口分配寄存器
F0D1H	PS_RXD1	UART1 数据输入端口分配寄存器
F0D2H	PS_TXD0	UART0 数据输出端口分配寄存器
F0D3H	PS_RXD0	UART0 数据输入端口分配寄存器
F0D4H	PS_SCL	IIC 时钟输入端口分配寄存器
F0D5H	PS_SDA	IIC 数据输入端口分配寄存器
F0D6H	PS_PG0	EPWM 输出通道 0 端口选择寄存器
F0D7H	PS_PG1	EPWM 输出通道 1 端口选择寄存器
F0D8H	PS_PG2	EPWM 输出通道 2 端口选择寄存器

地址	寄存器	寄存器描述
F0D9H	PS_PG3	EPWM 输出通道 3 端口选择寄存器
F0DAH	PS_PG4	EPWM 输出通道 4 端口选择寄存器
F0DBH	PS_PG5	EPWM 输出通道 5 端口选择寄存器
F0DCH	PS_CC0	Timer2 输出通道 0 端口选择寄存器
F0DDH	PS_CC1	Timer2 输出通道 1 端口选择寄存器
--	--	--
F0E0H	EI0CFGL	外部扩展中断组中断控制寄存器
--	--	--
F0E4H	INTFC	外部中断滤波控制寄存器
--	--	--
F0F0H	PS_SCLKOI	SPI 的时钟输入端口选择寄存器
F0F1H	PS_MOSI	SPI 的主控输出从控输入端口选择寄存器
F0F2H	PS_MISO	SPI 的主控输入从控输出端口选择寄存器
F0F3H	PS_NSS0OI	SPI 的片选端口 0 选择寄存器
F0F4H	PS_NSS1OI	SPI 的片选端口 1 选择寄存器
F0F5H	PS_NSS2OI	SPI 的片选端口 2 选择寄存器
F0F6H	PS_NSS3OI	SPI 的片选端口 3 选择寄存器
F0F7H	PS_CLO	时钟输出口选择寄存器
--	--	--
F120H	PWMCON	PWM 控制寄存器
F121H	PWMOE	PWM 输出使能寄存器
F122H	PWMPINV	PWM 输出极性选择寄存器
--	--	--
F126H	PWMCNTE	PWM 计数启动控制寄存器
F127H	PWMCNTM	PWM 计数模式选择寄存器
F128H	PWMCNTCLR	PWM 计数器清零控制寄存器
F129H	PWMLOADEN	PWM 加载使能控制寄存器
F12AH	PWM0DIV	PWM0 时钟分频控制寄存器
F12BH	PWM1DIV	PWM1 时钟分频控制寄存器
F12CH	PWM2DIV	PWM2 时钟分频控制寄存器
--	--	--
F130H	PWMP0L	PWM01 周期数据寄存器低 8 位
F131H	PWMP0H	PWM01 周期数据寄存器高 8 位
F134H	PWMP1L	PWM23 周期数据寄存器低 8 位
F135H	PWMP1H	PWM23 周期数据寄存器高 8 位
F138H	PWMP2L	PWM45 周期数据寄存器低 8 位
F139H	PWMP2H	PWM45 周期数据寄存器高 8 位
--	--	--
F140H	PWMD0L	PWM0 比较数据寄存器低 8 位
F141H	PWMD0H	PWM0 比较数据寄存器高 8 位
F142H	PWMD1L	PWM1 比较数据寄存器低 8 位
F143H	PWMD1H	PWM1 比较数据寄存器高 8 位

地址	寄存器	寄存器描述
F144H	PWMD2L	PWM2 比较数据寄存器低 8 位
F145H	PWMD2H	PWM2 比较数据寄存器高 8 位
F146H	PWMD3L	PWM3 比较数据寄存器低 8 位
F147H	PWMD3H	PWM3 比较数据寄存器高 8 位
F148H	PWMD4L	PWM4 比较数据寄存器低 8 位
F149H	PWMD4H	PWM4 比较数据寄存器高 8 位
F14AH	PWMD5L	PWM5 比较数据寄存器低 8 位
F14BH	PWMD5H	PWM5 比较数据寄存器高 8 位
--	--	--
F160H	PWMDTE	PWM 死区使能控制寄存器
F161H	PWM01DT	PWM0/1 死区延时数据寄存器
F162H	PWM23DT	PWM2/3 死区延时数据寄存器
F163H	PWM45DT	PWM4/5 死区延时数据寄存器
--	--	--
F169H	PWMZIE	PWM 零点中断屏蔽寄存器
--	--	--
F16BH	PWMDIE	PWM 向下比较中断屏蔽寄存器
--	--	--
F16DH	PWMZIF	PWM 零点中断标志寄存器
--	--	--
F16FH	PWMDIF	PWM 向下比较中断标志寄存器
--	--	--
--	--	--
--	--	--
--	--	--
F5C1H	BRTDL	BRT 定时器数据加载值低 8 位
F5C2H	BRTDH	BRT 定时器数据加载值高 8 位
--	--	--
F690H	LVDCON	电源监控寄存器
F691H	LVDEICFG	LVD 中断控制寄存器
F692H	LVDS	LVD 电压选择寄存器
F693H	ADCLDO	AD 参考电压控制寄存器
--	--	--
F697H	XT_SCM	LSE/HSE 时钟停振检测控制寄存器
--	--	--
--	--	--
F708H	CRCIN	CRC 模块数据输入寄存器
F709H	CRCDL	CRC 运算结果低 8 位数据寄存器
F70AH	CRCDH	CRC 运算结果高 8 位数据寄存器
--	--	--
F740H	LEDC0DATA0/LED0DATA	COM0 对应 SEG 数据寄存器
F741H	LEDC0DATA1/LED1DATA	
F742H	LEDC0DATA2	
F743H	LEDC0DATA3	
F744H	LEDC1DATA0/LED2DATA	COM1 对应 SEG 数据寄存器
F745H	LEDC1DATA1/LED3DATA	
F746H	LEDC1DATA2	
F747H	LEDC1DATA3	

地址	寄存器	寄存器描述
F748H	LEDC2DATA0/LED4DATA	COM2 对应 SEG 数据寄存器
F749H	LEDC2DATA1/LED5DATA	
F74AH	LEDC2DATA2	
F74BH	LEDC2DATA3	
F74CH	LEDC3DATA0/LED6DATA	COM3 对应 SEG 数据寄存器
F74DH	LEDC3DATA1/LED7DATA	
F74EH	LEDC3DATA2	
F74FH	LEDC3DATA3	
F750H	LEDC4DATA0/LED0SEL	COM4 对应 SEG 数据寄存器
F751H	LEDC4DATA1/LED1SEL	
F752H	LEDC4DATA2	
F753H	LEDC4DATA3	
F754H	LEDC5DATA0/LED2SEL	COM5 对应 SEG 数据寄存器
F755H	LEDC5DATA1/LED3SEL	
F756H	LEDC5DATA2	
F757H	LEDC5DATA3	
F758H	LEDC6DATA0/LED4SEL	COM6 对应 SEG 数据寄存器
F759H	LEDC6DATA1/LED5SEL	
F75AH	LEDC6DATA2	
F75BH	LEDC6DATA3	
F75CH	LEDC7DATA0/LED6SEL	COM7 对应 SEG 数据寄存器
F75DH	LEDC7DATA1/LED7SEL	
F75EH	LEDC7DATA2	
F75FH	LEDC7DATA3	
F760H	LEDCOMEN/LEDIOEN	COM 口使能控制寄存器
F761H	LEDSEGEN0/SCAN1W	SEG 口使能控制寄存器
F762H	LEDSEGEN1/SCAN2W	SEG 口使能控制寄存器
F763H	LEDSEGEN2	SEG 口使能控制寄存器
F764H	LEDSEGEN3	SEG 口使能控制寄存器
F765H	LEDCON/LEDCON1	LED 控制寄存器
F766H	LEDCKS	LED 时钟选择寄存器
F768H	LEDCOMTIME	COM 口有效时间选择寄存器
F769H	LEDMODE	LED 驱动模式选择寄存器
F76AH	LEDSTATUS	LED 点阵扫描中断寄存器
F76BH	LEDDRV	LED 矩阵驱动端口电流控制寄存器
F76CH	LEDRESEQ	LED 点阵引脚映射寄存器
F76DH	LEDSEGTIME	调光时间选择寄存器
0xF76E	LEDCOMDRV	LED COM 口驱动电流选择寄存器
0xF76F	LEDSEGDR0	LED 管脚驱动使能控制寄存器 0
0xF770	LEDSEGDR1	LED 管脚驱动使能控制寄存器 1
0xF771	LEDSEGDR2	LED 管脚驱动使能控制寄存器 2
0xF772	LEDSEGDR3	LED 管脚驱动使能控制寄存器 3

3. 复位

复位时间（Reset Time）是指从芯片复位到芯片开始执行指令的时间，其默认设计值约为 22ms。该时间包括振荡器起振时间，配置时间。芯片硬件复位后都会存在这个复位时间。

芯片可用如下几种硬件复位方式：

- 上电复位
- 外部复位
- POR 复位
- 低电压复位
- 看门狗溢出复位

芯片支持软件复位方式：

- 写 SWRST 复位(软件复位会立即将 PC 清零，程序从复位向量开始运行)

上述任意一种复位发生时，所有的系统寄存器将恢复默认状态，程序停止运行，同时程序计数器 PC 清零，复位结束后程序从复位向量 0000H 开始运行。

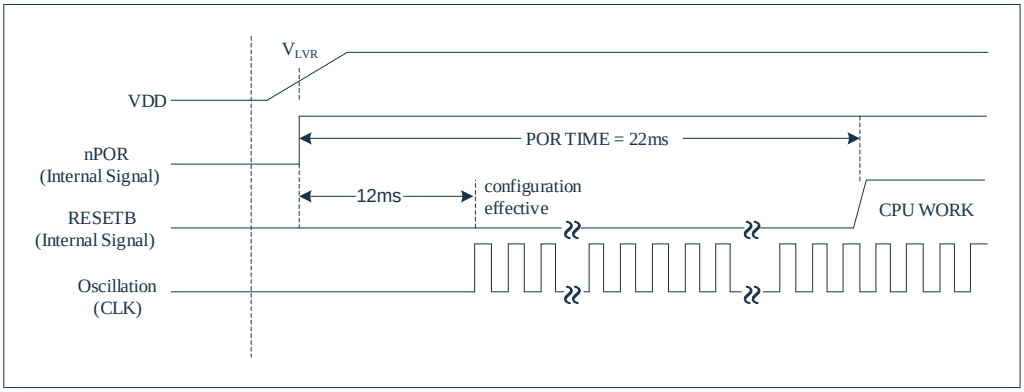
任何一种复位情况都需要一定的响应时间，系统提供完善的复位流程以保证复位动作的顺利进行。

3.1 上电复位

上电复位与 LVR 操作密切相关。系统上电的过程呈逐渐上升的曲线形式，需要一定时间才能达到正常电平值。下面给出上电复位的正常时序：

- 上电：系统检测到电源电压上升并等待其稳定；
- 系统初始化：所有的系统寄存器被置为初始值；
- 振荡器开始工作：振荡器开始提供系统时钟；
- 执行程序：上电结束，程序开始运行。

稳定时间（Stabilization Time）默认为 22ms。上电复位时序图如下图所示：



上电过程中相关配置默认状态为：LVD 低压检测使能打开、LVD 检测电压为 2.5V、LVD 模式为复位模式，所以 VDD 电压大于 2.5V 以上时系统才进行上电配置过程。LVD 相关配置可修改为其他状态，但在修改配置生效前，系统按照默认配置运行。

系统是否为上电复位可通过 PORF（RSTCF.6）标志位来判断。可置 PORF 标志位为 1 的复位类型有：上电复位、LVR 复位、外部复位。

3.2 外部复位

外部复位指来自外部端口（NRST）的复位信号，经施密特触发器输入后复位芯片。如果在操作电压范围及稳定振荡情况下 NRST 脚保持低电平约 4*T_{LSI} 以上，就会请求复位。内部状态被初始化复位状态变为“1”后，需要 22ms 的稳定时间，内部 RESETB 信号才变为“0”，程序从向量地址 0000H 处开始执行。

稳定时间（Stabilization Time）内芯片重新进行配置过程，与上电复位配置过程相同。外部复位引脚 NRST 及其上拉电阻使能，通过 CONFIG 配置。

系统是否为外部复位可通过 EXTIF（RSTCF.5）标志位来判断。

3.3 POR 复位

芯片内部有一个上电即独立运行的低功耗 POR 电路，该电路一直开启，且不受芯片其他任何模式和电路控制，包括空闲模式/休眠模式。POR 电路一直检测 VDD 的电压，低于约 2.1V 时，系统产生复位。

3.4 LVD 低压检测复位/中断

芯片内部集成了 LVD 低压检测复位/中断功能，当系统电压 VDD 跌至低于设定电压时，可以使系统产生复位或者中断。系统产生复位还是中断由 CONFIG 寄存器 LVDCFG 的 bit[1:0]决定。

(a) 复位模式（LVDCFG[1:0]=0x03，默认为复位模式）

选择复位模式后，电压比较结果会产生内部复位信号。

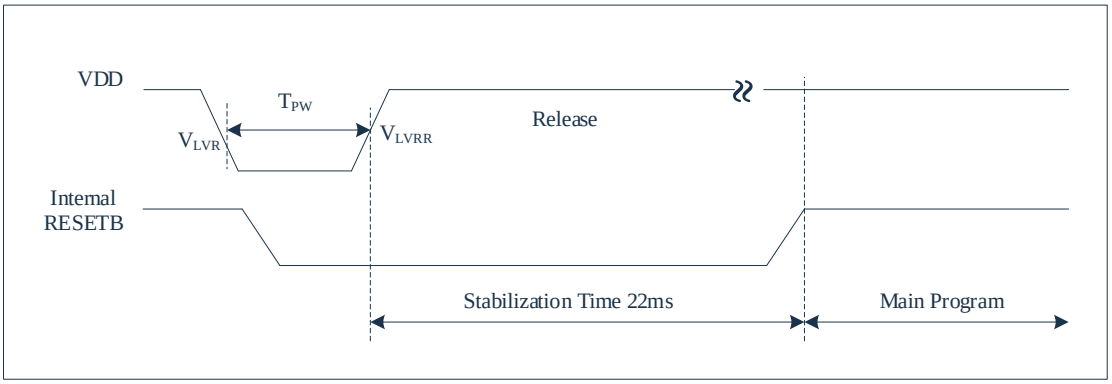
此时电源电压检测点通过 CONFIG 寄存器 LVDCFG 的 bit[4:2]选择 2.5、2.7、3.0、3.3、3.7、4.0、4.3V 共 7 种检测电压。

(b) 中断模式（LVDCFG[1:0]=0x01）

选择中断模式后，电压比较结果会产生内部中断信号。

此时电源电压检测点通过 LVDS 寄存器的 bit[2:0]选择 2.5、2.7、3.0、3.3、3.7、4.0、4.3V 共 7 种检测电压。

复位时序图如下图所示：



注 1：V_{LVRR} 为上电时的复位释放电压，$V_{LVRR} > V_{LVR}$，迟滞电压约为 50mV。

注 2：T_{PW} 为从 VDD 低于 V_{LVR} 到回升超过 V_{LVRR} 为止的各种情况下保证正确上电复位所需要的时间（并不指小于该时间时芯片不会产生有效复位）。T_{PW}>32us。

注 3：稳定时间（Stabilization Time）内芯片重新进行配置过程，与上电复位配置过程相同。

3.5 看门狗复位

看门狗复位是系统的一种保护设置。在正常状态下，由程序将看门狗定时器清零。若出错，系统处于未知状态，看门狗定时器溢出，此时系统复位。看门狗复位后，系统重启进入正常状态。

WDT 的计数器不可被寻址，在上电后 WDT 溢出复位功能默认开启，可以通过寄存器关闭该功能；另外也可以在 CONFIG 中设置在上电复位后强制开启溢出复位功能。设置 WDT 寄存器时建议先将 WDT 计数器清除，以便准确控制 WDT 的溢出时间。

看门狗复位的时序如下：

1) 看门狗定时器状态：系统检测看门狗定时器是否溢出，若溢出，则系统复位；
2) 初始化：所有的系统寄存器被置为默认状态；
3) 程序：复位完成，程序开始从 0000H 运行。

WDT 的时钟源由 LSI 提供，WDT 计数器的计时基本周期为 TLSI。WDT 溢出后复位 CPU 与所有的寄存器。在调试态下 WDT 复位不会重新进行上电复位配置，1 个 Tsys 后程序立即从 0000H 开始执行。非调试态下会重新进行上电复位配置。看门狗的溢出时间可由程序设置，在 WDCON 寄存器 WDS2-WTS0 两位可选择溢出时间。看门狗溢出时间如下表所示：

WTS[2:0]	Watchdog Interval	Number of clocks	OVT@F_{LSI}=32KHz
000	2^8	256	8ms
001	2^9	512	16ms
010	2^{10}	1024	32ms
011	2^{11}	2048	64ms
100（默认）	2^{12}	4096	128ms
101	2^{14}	16384	512ms
110	2^{15}	32768	1s
111	2^{17}	131072	4s

3.6 软件复位

芯片内部可实现程序软件复位，软件复位可将程序流程重新定位到复位地址 0000H，然后再次运行程序。用户可写软件复位控制位 RSTCF [7]（SWRST=1）实现自定义的软件复位。软件复位不会重新进行上电复位配置。

3.7 复位控制寄存器（RSTCF）

0x95	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RSTCF	SWRST	PORF	EXTIF	—	--	WDTRF	--	LVRRF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	0	0	0	0	0	0

Bit7 SWRST: 软件复位控制位；
1: 执行系统软件复位（复位后需写0清掉，不需要TA写时序，写1需要TA写时序）。
0: --
Bit6 PORF: 上电复位标志位；
1: 系统为上电复位（写0清掉，不需要TA写时序）。
0: --
Bit5 EXTIF: 外部复位标志位；
1= 系统为外部复位（写0清除，不需要TA写时序）；
0= --
Bit4 - 保留。
Bit3 - 保留，须为0。
Bit2 WDTRF: WDT复位标志位；
1= 系统由WDT复位（写0清除）；
0= 系统不由WDT复位。
Bit1 - 保留，必须为0。
Bit0 LVRRF LVD复位标注位；
1: 系统为LVD复位（写0清除，不需要TA写时序）；
0: --

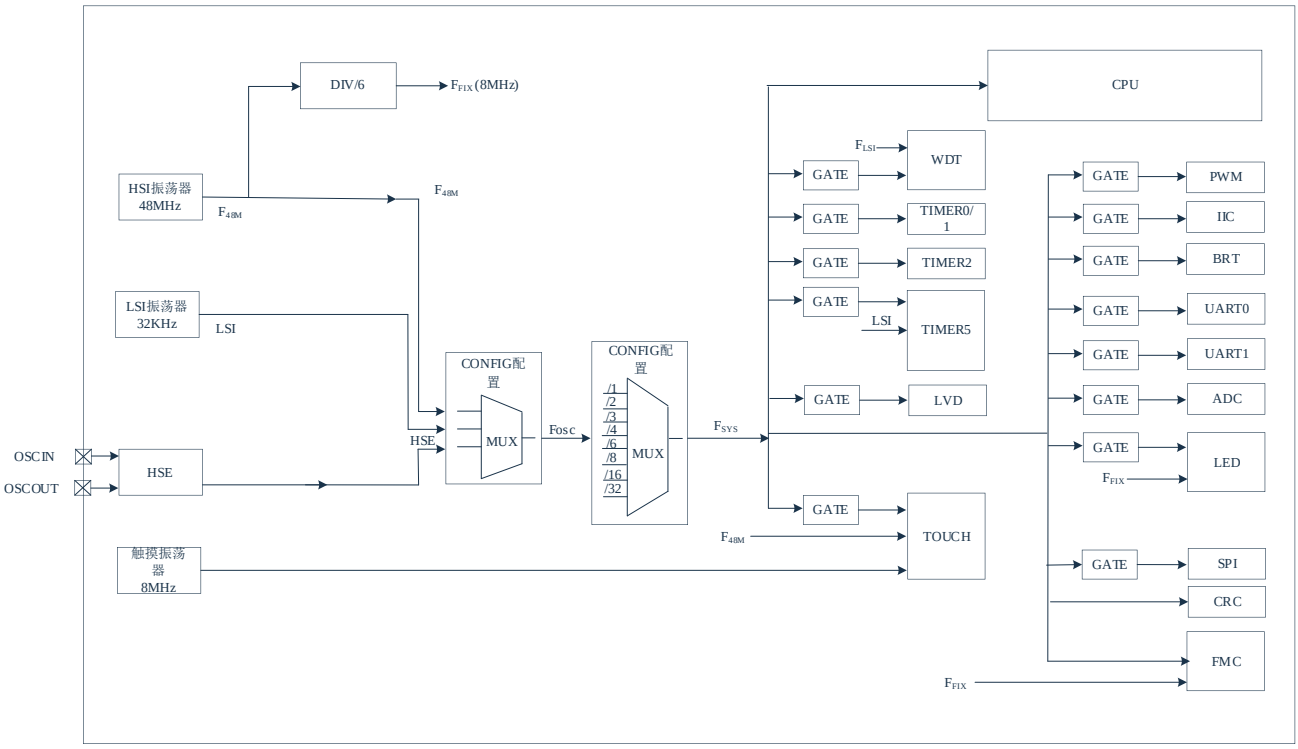
4. 时钟结构

系统时钟的时钟源具有 3 种类型，可通过系统配置寄存器或用户寄存器的设置进行时钟源、时钟分频选择。系统时钟源如下：

- 内部高速振荡 HSI（48MHz）。
- 外部高速振荡 HSE（8MHz/16MHz）。
- 内部低速振荡 LSI（32KHz）。

4.1 系统时钟结构

系统时钟结构框图如下图所示：



4.2 相关寄存器

4.2.1 外设时钟源选择寄存器 CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	--	--	--	U1M	U0M	T2M	T1M	T0M
R/W	R/W	R/W	R/W	W	W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit5 -- 保留，必须为0

Bit4 U1M: UART1 的定时器时钟选择
0= BRT 的溢出时钟；
1= Timer1 的溢出时钟。

Bit3 U0M: UART0 的定时器时钟选择
0= BRT 的溢出时钟；
1= Timer1 的溢出时钟。

Bit2 T2M: Timer2 的时钟源选择位；
0= 由 T2PS 选择；
1= Fsys。

Bit1 T1M Timer1 的时钟源选择位；
0= Fsys/12；
1= Fsys/4。

Bit0 T0M Timer0 的时钟源选择位；
0= Fsys/12；
1= Fsys/4。

4.2.2 系统时钟监控寄存器 XTSCM

F697H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XTSCM	SCMEN	SCMIE	--	--	--	--	SCMIF	SCMSTA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
复位值	1	0	0	0	0	0	0	0

Bit7 SCMEN: 停振检测模块使能；
1= 使能；
0= 禁止。
Bit6 SCMIE: 停振检测中断使能位；
1= 使能；
0= 禁止。
Bit5~Bit2 -- 保留，须均为 0。
Bit1 SCMIF: 停振中断标志位；
1= 表示停振；
0= 软件清 0，清 0 后会自动切换到 HSE 主频的情况（只能软件清 0）。
Bit0 SCMSTA: 停振状态位，只读；
1= 表示停振；
0= 停振恢复。

说明：

1) SCMIF 和 SCMSTA 都能反映 HSE 做系统时钟的状态。两者最大的区别在于当 HSE 停振后，SCMSTA 会一直处于高电平状态，直到 HSE 恢复；SCMIF 也能反映 HSE 停振，但是它可产生中断（需开中断使能），也可通过寄存器清除 SCMIF，清除后主频会切回 HSE（若此时还是处于停振状态，那么将再次触发中断）。
2) 停振后，主频会由 HSE 切到 HSI，若 HSE 恢复后，SCMSTA 会自动清零，主频也会由 HSI 自动切回 HSE。
3) 在系统复位期间（进行 CONFIG 配置）若发生过停振（或系统复位期间一直停振未恢复），SCMIF 也会置 1。在正常运行阶段，若 SCMIF 为 1 的状态时进行写软件复位，软件复位后 SCMIF 任然为 1。
4) SCMSTA=1 时禁止软件关闭 SCMEN。

4.2.3 模块时钟允许寄存器 PCGEN0

92H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCGEN0	UART1EN	UART0EN	SPIEN	IICEN	WDTEN	TIMER5EN	TIMER2EN	TIMER01EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
复位值	0	0	0	0	0	0	0	0

注：操作模块相关寄存器前需使能模块时钟。

Bit7 UART1EN: UART1 时钟使能控制；
1= 使能 UART1 时钟；
0= 不使能 UART1 时钟。

Bit6 UART0EN: UART0 时钟使能控制；
1= 使能 UART0 时钟；
0= 不使能 UART0 时钟。

Bit5 SPIEN: SPI 时钟使能控制；
1= 使能 SPI 时钟；
0= 不使能 SPI 时钟。

Bit4 IICEN: IIC 时钟使能控制；
1= 使能 IIC 时钟；
0= 不使能 IIC 时钟。

Bit3 WDTEN: WDT 时钟使能控制；
1= 使能 WDT 时钟；
0= 不使能 WDT 时钟。

Bit2 TIMER5EN: TIMER5 时钟使能控制；
1= 使能 TIMER5 时钟；
0= 不使能 TIMER5 时钟。

Bit1 TIMER2EN: TIMER2 时钟使能控制；
1= 使能 TIMER2 时钟；
0= 不使能 TIMER2 时钟。

Bit0 TIMER01EN: TIMER01 时钟使能控制；
1= 使能 TIMER01 时钟；
0= 不使能 TIMER01 时钟。

4.2.4 模块时钟允许寄存器 PCGEN1

93H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCGEN1	-	TOUCHEN	LEDEN	ADCEN	--	PWMEN	LVDEN	BRTEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

注：操作模块相关寄存器前需使能模块时钟。

Bit7 -: 保留。

Bit6 TOUCHEN: TOUCH 时钟使能控制；
1= 使能 TOUCH 时钟；
0= 不使能 TOUCH 时钟。

Bit5 LEDEN: LED 时钟使能控制；
1= 使能 LED 时钟；
0= 不使能 LED 时钟。

Bit4 ADCEN: ADC 时钟使能控制；
1= 使能 ADC 时钟；
0= 不使能 ADC 时钟。

Bit3 --

Bit2 PWMEN: PWM 时钟使能控制；
1= 使能 PWM 时钟；
0= 不使能 PWM 时钟。

Bit1 LVDEN: LVD 时钟使能控制；
1= 使能 LVD 时钟；
0= 不使能 LVD 时钟。

Bit0 BRTEN: BRT 时钟使能控制；
1= 使能 BRT 时钟；
0= 不使能 BRT 时钟。

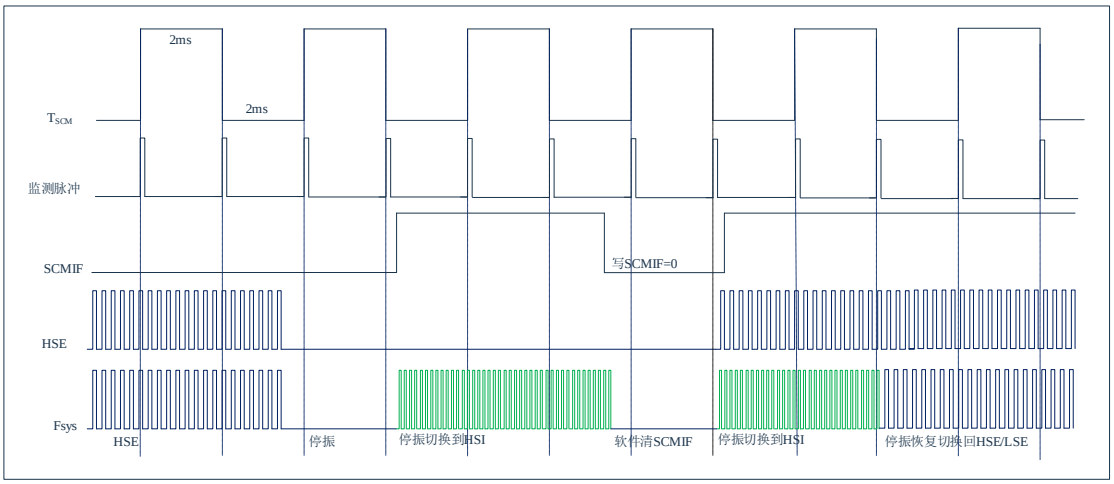
4.3 系统时钟监控

系统时钟监控（SCM：system clock monitoring）是为防止因晶振停振影响到系统不工作的情况而设计的一种监控保护电路。当使用 HSE 做系统时钟时，一旦检测到 HSE 时钟停止，系统会强制启动 HSI 时钟源，待 HSI 稳定后系统以 8MHz 主频运行，之后若 HSE 时钟恢复且稳定，系统时钟将自动从 HSI 切回 HSE。

SCM 模块每 4ms 对系统时钟 HSE 监测一次，监测周期 T_{SCM} 的占空比为 1:1。在 T_{SCM} 为高电平期间 SCM 对 HSE 进行停振监测，T_{SCM} 为低电平期间对监测结果进行处理，如果检测到 HSE 停振，则将系统时钟切换到 HSI，同时停振中断标志位 SCMIF 置 1。若清除 SCMIF，即使 HSE 已停振，系统时钟也将自动切换回 HSE。

系统上电运行时，若选择 HSE 作为系统主时钟，则默认开启系统时钟监控。系统复位完成后，需要先查询中断状态位是否处于停振，并清除时钟监控中断标志位，之后可写寄存器关闭或再次打开时钟监控功能。

系统时钟监控结构框图如下图所示：



5. 电源管理

低功耗模式分为 2 类：

- IDLE：空闲模式
- STOP：休眠模式

用户利用 C 语言进行程序开发时，强烈建议使用 IDLE 和 STOP 宏指令来控制系统模式，不要直接设置 IDLE 和 STOP 位。宏指令如下：

进入空闲模式：IDLE();

进入休眠模式：STOP();

5.1 电源管理寄存器

5.1.1 电源管理寄存器 PCON

0x87	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	--	--	--	--	--	--	STOP	IDLE
R/W	R	R	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit2 -- 保留，须均为 0。

Bit1 STOP: 休眠状态控制位；

0= 未进入休眠状态；

1= 进入休眠状态（退出 STOP 模式自动清零）。

Bit0 IDLE: 空闲状态控制位；

0= 未进入空闲状态；

1= 进入空闲状态（退出 IDLE 模式自动清零）。

进入休眠模式的汇编代码：

```
MOV     PCON,#02H
NOP     ;//后边必须跟至少 6 个 NOP 指令
NOP
NOP
NOP
NOP
NOP
```

5.1.2 LVD 低压检测中断控制寄存器 LVDCON

该 MCU 自带电源检测功能。配制用户 config 选择 LVD 或者 LVR 功能，同时设置好电压监测点，当电源电压下降到低于/上升到高于 LVD 设定值时，将会产生中断，提醒用户。

如果休眠前 LVD 模块使能，进入休眠后硬件不会关闭该模块电路。

0xF690	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDCON	--	--	--	--	LVDOUTB_FILT	LVDOUTB	LVDINTE	LVDINTF
R/W	--	--	--	--	R	R	R/W	R/W
复位值	0	0	0	0	--	--	0	0

Bit7~Bit4 -- 保留，须均为 0。
Bit3 LVDOUTB_FILT 电源电压检测状态滤波后的输出
Bit2 LVDOUTB 电源电压监测状态输出
0= 电源电压低于监测电压；
1= 电源电压高于监测电压。
Bit1 LVDINTE: LVD 中断使能位；
0= LVD 中断禁止；
1= LVD 中断使能。
Bit0 LVDINTF: LVD 中断标志位（LVDEICFG[1:0]选择边沿）；
0= 未产生中断；
1= 产生中断。

5.1.3 LVD 低压检测中断配置寄存器 LVDEICFG

0xF691	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDEICFG	LVDSNUS1	LVDSNUS0	LVDSCKS1	LVDSCKS0	--	--	INTEDGE1	INTEDGE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~6 LVDSNUS<1:0>: LVD 滤波时间选择；
00= 1 个滤波时钟；
01= 2 个滤波时钟；
10= 3 个滤波时钟
11= 4 个滤波时钟
Bit5~4 LVDSCKS<1:0>: 滤波时钟选择；
00= 滤波时钟为系统时钟；
01= 滤波时钟为系统时钟 4 分频。
10= 滤波时钟为系统时钟 16 分频。
11= 滤波时钟为系统时钟 64 分频。
Bit3~2 保留
Bit1~0 INTEDGE<1:0>: LVD 中断边沿选择
00= 边沿检测功能关闭
01= 上升沿产生中断
10= 下降沿产生中断
11= 双沿产生中断

5.1.4 LVD 低压检测中断电压选择寄存器 LVDS

F692H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDS	LVDSEN3	LVDSEN2	LVDSEN1	LVDSEN0	--	LVDS2	LVDS1	LVDS0
R/W	W	W	W	W	--	R/W	R/W	R/W
复位值	0	0	0	0	--	1	1	1

Bit[7:4] LVDSEN<3:0> LVDS[2:0]访问使能
0101= 在写LVDS[2:0]时，需同LVDSEN[3:0]同时写入0101才能写入
其他= 无效
Bit3 —— 保留
Bit[2:0] LVDS<2:0>: LVD低压检测中断电压点选择

000=	4.3V	100=	3.3V
001=	4.0V	101=	3.0V
010=	3.7V	110=	2.7V
011=	3.0V	111=	2.5V

5.2 IDLE 空闲模式

在此种模式下，只有 CPU 时钟源被关闭。因此，在这种状态下，外设功能（如定时器、PWM 和 I^2C）和时钟发生器仍然正常工作。

系统进入空闲模式后，可由任意中断唤醒，唤醒后进入中断处理程序，中断返回后，继续执行休眠操作后指令。

如果在中断服务程序中进入空闲模式，则只能由优先级较高的中断唤醒系统。

5.3 STOP 休眠模式

在此种模式下，TIMER5（选择 LSI 作为模块计数时钟源）模块外其他所有的电路被关闭，系统处于低功耗模式，数字电路均不工作。

5.3.1 休眠唤醒

进入休眠模式后，可开启休眠唤醒功能唤醒休眠模式。有以下几种方式唤醒休眠模式：

1) INT0/1 中断

采用 INT0/1 中断唤醒休眠模式，必须在进入休眠前将总中断使能与 INT0/1 中断使能打开，才能唤醒系统。INT0，INT1 中断相关寄存器包括 IE，IP，TCON，IO 复用映射寄存器，INT0/1 中断唤醒只能下降沿中断唤醒休眠。

2) EINT0-EINT3 中断

采用外部中断唤醒，必须在进入休眠前将总中断使能与端口中断使能打开，才能唤醒系统。外部中断唤醒可选择上升沿、下降沿、双沿中断唤醒休眠，中断唤醒边沿由外部中断控制寄存器 EI0CFGL 设置。

3) Timer5 唤醒

由 Timer5 定时唤醒，必须在进入休眠前将定时唤醒功能打开，同时设置好休眠状态到唤醒的时间。定时唤醒电路的时钟源由 LSI（低功耗振荡器）提供，开启定时唤醒功能则 LSI 在休眠状态下自动打开。

5.3.2 唤醒等待状态

无论是 INT0/1 中断、EI00-EI03 中断，还是 T5 定时唤醒，在中断产生或定时时间到后，都需要等待一段时间才能唤醒系统，执行程序的下一条指令。中断产生或定时时间到后，系统振荡器启动，但振荡频率还未稳定，CPU 未工作，PC 仍停止在休眠状态，系统需要等待一段时间才将时钟提供给 CPU。唤醒 CPU 等待的时间在烧写 CONFIG 中设置，等待时间可设置为 50us~1s。唤醒等待时间过后，MCU 认为系统时钟已经稳定，才将时钟提供给 CPU，程序继续执行。

如果内部唤醒定时器和外部中断唤醒功能都为打开状态，系统进入休眠模式后，任意一种唤醒方式都能唤醒 CPU。如果内部定时器先唤醒振荡器，之后有外部中断输入，那么在唤醒等待时间过后，程序先执行中断处理程序然后再继续执行休眠操作后的指令。

5.3.3 休眠唤醒时间

利用外部中断唤醒系统总的唤醒时间为：

电源管理器稳定时间（250us）+唤醒等待时间

利用定时唤醒系统总的唤醒时间为：

电源管理器稳定时间（250us）+唤醒定时器的计时+唤醒等待时间

(上述给出时间的条件为 Fsys>1MHz)

5.3.4 休眠下复位操作

在休眠模式下，也可通过掉电复位、外部复位重新启动系统。

掉电复位：不需要任何其他条件，VDD 降低至 0V 后重新上电至工作电压，进入上电复位状态。

外部复位：需要开启外部复位功能，相关端口配置成专用复位口，在休眠时复位口保持>4*T_{LSI} 的低电平，系统产生复位，释放复位口，则系统重新启动。

5.3.5 调试模式下的休眠功耗

调试模式下的休眠状态并不能反映实际芯片休眠状态。

在调试模式下，系统进入休眠状态后，相关的电源管理电路，振荡器并没有关闭，而是继续开启。调试模式下也可以进行唤醒操作，其唤醒方式与正常模式相同。

故在此状态下，测试得到的休眠电流并不是真正的休眠功耗。建议在调试模式下将休眠唤醒功能开发完成后关闭调试模式，然后重新启动系统，此时的测得电流即为实际休眠功耗。

5.3.6 休眠模式应用举例

系统在进入休眠模式之前，若用户需要获得较小的休眠电流，请先确认所有 I/O 的状态，若用户方案中存在悬空的 I/O 口，把所有悬空口都设置为输出口，确保每一个输入口都有一个固定的状态，以避免 I/O 为输入状态时，口线电平处于不定态而增大休眠电流；关断 ADC 模块及其他外设以减小休眠电流。

例：采用 timer5 定时唤醒时，进入休眠的处理程序（汇编程序）

```
SLEEP_MODE:   MOV          P0DIG, #0FFh
              MOV          P0TRIS,#0FFh
              MOV          P0,#0FFh
              MOV          P1DIG, #0FFh
              MOV          P1TRIS,#0FFh
              MOV          P1,#0FFh
              MOV          P2DIG, #0FFh
              MOV          P2TRIS,#0FFh
              MOV          P2,#0FFh
              MOV          P5DIG, #0FFh
              MOV          P5TRIS,#0FFh
              MOV          P5,#0FFh
              MOV          T5CON,#0E0h
              MOV          EIE2,#004h
              MOV          IE,#080h
              关闭其它功能的操作指令
              MOV          PCON,#02H      ; 执行可功能唤醒的休眠操作，
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP          ;执行休眠操作的指令后必须跟 6 条 NOP 指令
              唤醒后的其他操作指令
```

6. 中断

6.1 中断概述

芯片具有多个中断源及中断向量，默认中断向量偏移地址为 0x0000(中断偏移地址通过寄存器 IREMAP 设置)：

中断源	中断描述	中断向量	同级优先序列
INT0	外部中断 0	0-0x0003	1
Timer0	定时器 0 中断	1-0x000B	2
INT1	外部中断 1	2-0x0013	3
Timer1	定时器 1 中断	3-0x001B	4
UART0	TI0 or RI0	4-0x0023	5
Timer2	定时器 2 中断	5-0x002B	6
UART1	TI1 or RI1	6-0x0033	7
EXT0IF<3:0>	扩展的外部中断组 0	7-0x003B	8
--	--	8-0x0043	9
--	--	9-0x004B	10
--	--	10-0x0053	11
--	--	11-0x005B	12
XTDET	停振检测中断	12-0x0063	13
--	--	13-0x006B	14
--	--	14-0x0073	15
--	--	15-0x007B	16
--	--	16-0x0083	17
T5	定时器 5 中断	17-0x008B	18
PWM	PWM 中断	18-0x0093	19
ADC	ADC 中断	19-0x009B	20
--	--	20-0x00A3	21
I^2C	I^2C 中断	21-0x00AB	22
SPI	SPI 中断	22-0x00B3	23
--	--	23-0x00BB	24
--	--	24-0x00C3	25
LED	LED 中断	25-0x00CB	26
LVD	LVD 掉电中断	26-0x00D3	27
TOUCH	触摸中断	27-0x00DB	28
--	--	28-0x00E3	29

芯片规定两个中断优先级，可实现两级中断嵌套。当一个中断已经响应，若有高级别中断发出请求，后者可以中断前者，实现中断嵌套。

6.2 中断重映射

系统默认中断向量偏移基地址为 0x0000，但某些应用场景下需要将偏移基地址设置为其他值，此功能可通过修改中断偏移地址寄存器 IREMAP 来实现。

IREMAP 寄存器为设置为 0x01 时，0X01<<9 =0X0200，即中断偏移基地址为 0X0200，则其他所有中断地址需加上偏移基地址才是实际的中断向量地址。比如 INT0 的向量地址重映射之后应为：0x0200+0x0003=0x0203

由上述描述可知，中断向量偏移基地址最小的偏移单元为 0.5KB。

中断重映射使用实例：

在程序空间最后 2KB 区域实现 ISP 功能，上电后，程序跳转至最后 2KB 起始地址，之后设置 IRMAP 为 0x3C。即中断偏移的基地址为 0x7800。在此 2KB 区域实现完成相应的 ISP 功能或者跳过 ISP 后，关闭所有的中断使能，跳转至主程序区域，然后将 IREMAP 设置为 0X00，中断偏移的基地址即为 0x0000

6.3 外部中断

6.3.1 INT0/INT1 中断

芯片支持 INT0、INT1 外部中断，INT0/INT1 可选择下降沿或低电平触发中断，相关控制寄存器为 TCON。INT0 与 INT1 占用两个中断向量。

6.3.2 外部扩展中断

外部扩展中断 EINT 包含 4 个输入通道 EINT0-EINT3，每个通道有所有 GPIO（PS_EINT0，PS_EINT1，PS_EINT2，PS_EINT3），且每个通道支持下降沿/上升沿/双沿中断，边沿触发类型通过 EI0CFGL 寄存器配置。

例如将 EINT0 口配置为下降沿中断：

```
……;                    //将相关端口设置为输入模式
PS_EINT0   =0x00;       //EINT0 通道选择 EINT0 管脚输入
EI0CFGL=0x02;           //将 EINT0 设置为下降沿触发中断
```

EINT0-EINT3 占用一个中断向量 0x003B；

当中断产生后，进入中断服务程序可先判断是哪个端口触发中断，然后再进行相应的处理。

6.3.3 外部中断滤波

外部中断可选择滤波参数，包括采样时钟，采样个数。所有外部中断口的滤波参数统一在 INTFC 寄存器中设置。

6.4 中断与休眠唤醒

系统在进入休眠模式（STOP 可唤醒模式）后，每个外部中断可设置唤醒系统。

INT0/INT1 中断唤醒系统需要打开相应的中断使能以及总中断使能，唤醒方式为下降沿唤醒（INT0/INT1 唤醒方式和中断触发方式选择位 IT0/IT1 无关）。

外部扩展中断 EINT 唤醒系统，建议在进入休眠模式前设置好相应的端口中断触发边沿方式（可选择上升沿/下降沿/双沿唤醒），以及打开相应的中断使能及总中断使能。

系统由外部中断唤醒后，首先进入中断服务程序处理中断唤醒任务，退出中断服务程序后，系统继续执行休眠操作后的指令。

6.5 中断寄存器

6.5.1 中断偏移基地址寄存器 IREMAP

0xA1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IREMAP	-	-	MAP05	MAP04	MAP03	MAP02	MAP01	MAP00
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

注：IREMAP 为受保护寄存器。

Bit7-6 - 保留

Bit5-0 MAP<5:0> 中断偏移地址

0X00= 中断偏移地址为 0X0000;

0X01= 中断偏移地址为(0X01<<9)，0X0200;

0X02= 中断偏移地址为(0X02<<9)，0X0400;

…… ……

0X20= 中断偏移地址为(0X20<<9)，0X4000;

…… ……

修改 IREMAP 需要的指令序列（中间不能插入其他任何指令）：

```
CLR     EA
MOV     TA,#0AAH
MOV     TA,#055H
ORL     IREMAP,#01H
SETB    EA
```

6.5.2 中断屏蔽寄存器

6.5.2.1 中断屏蔽寄存器 IE

中断屏蔽寄存器 IE 是可读写的寄存器，可位操作。当有中断条件产生时，无论对应的中断允许位或全局允许位 EA 的状态如何，中断标志位都将置 1。用户软件应在允许一个中断之前，确保先将相应的中断标志位清零。

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 EA: 全局中断允许位；
1= 允许所有未被屏蔽的中断；
0= 禁止所有中断。
Bit6 ES1: UART1中断允许位；
1= 允许UART1中断；
0= 禁止UART1中断。
Bit5 ET2: TIMER2总中断允许位；
1= 允许TIMER2所有中断；
0= 禁止TIMER2所有中断。
Bit4 ES0: UART0中断允许位；
1= 允许UART0中断；
0= 禁止UART0中断。
Bit3 ET1: TIMER1中断允许位；
1= 允许TIMER1中断；
0= 禁止TIMER1中断。
Bit2 EX1: 外部中断1中断允许位；
1= 允许外部中断1中断；
0= 禁止外部中断1中断。
Bit1 ET0: TIMER0中断允许位；
1= 允许TIMER0中断；
0= 禁止TIMER 0中断。
Bit0 EX0: 外部中断0中断允许位；
1= 允许外部中断0中断；
0= 禁止外部中断0中断。

6.5.2.2 中断屏蔽寄存器 EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	--	ADCIE	PWMIE	ET5	--	--
R/W	R/W	R/W	R	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI中断使能位；
1= 允许SPI中断；
0= 禁止SPI中断。
Bit6 I2CIE: I^2C中断使能位；
1= 允许I^2C中断；
0= 禁止I^2C中断。
Bit5 --: 保留，须为0；
Bit4 ADCIE: ADC中断使能位；
1= 允许ADC中断；
0= 禁止ADC中断。
Bit3 PWMIE: PWM总中断使能位；
1= 允许PWM所有中断；
0= 禁止PWM所有中断。
Bit2 ET5: Timer5中断使能位；
1= 允许Timer5中断；
0= 禁止Timer5中断。
Bit1~0 -- 保留，须为0。

6.5.2.3 Timer2 中断屏蔽寄存器 T2IE

0xCF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IE	T2OVIE	T2EXIE	--	--	--	--	T2C1IE	T2C0IE
R/W	R/W	R/W	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 T2OVIE: Timer2溢出中断允许位；
1= 允许中断；
0= 禁止中断。
Bit6 T2EXIE: Timer2外部加载中断允许位；
1= 允许中断；
0= 禁止中断。
Bit5~Bit2 -- 保留，须均为0。
Bit1 T2C1IE: Timer2比较/捕获通道1中断允许位；
1= 允许中断；
0= 禁止中断。
Bit0 T2C0IE: Timer2比较/捕获通道0中断允许位；
1= 允许中断；
0= 禁止中断。

若开启 Timer2 的中断，还需要开启 Timer2 的总中断使能位 ET2=1（IE.5=1）

6.5.2.4 外部中断扩展组 0 控制寄存器 EI0IE

0xAC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EI0IE	-	-	-	-	EI0IE3	EI0IE2	EI0IE1	EI0IE0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit4 -- 保留，须均为0。
Bit3~Bit0 EIiIE: EI0i口中断允许位(i=0-3)；
1= 允许中断；
0= 禁止中断。

6.5.3 中断优先级控制寄存器

6.5.3.1 中断优先级控制寄存器 IP

中断优先级控制寄存器 IP 是可读写的寄存器，可位操作。

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 -- 保留，须为0。
Bit6 PS1: UART1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit5 PT2: TIMER2中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit4 PS0: UART0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit3 PT1: TIMER1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit2 PX1: 外部中断1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit1 PT0: TIMER0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit0 PX0: 外部中断0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

6.5.3.2 中断优先级控制寄存器 EIP1

0xB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	--	--	--	--	--	--	--	PEI0
R/W	R	R	R	R	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0

Bit7-Bit1 -- 保留，须为0。
Bit0 PEI0: 外部扩展中断组0优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

6.5.3.3 中断优先级控制寄存器 EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	--	PADC	PPWM	PT5	--	--
R/W	R/W	R/W	R	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit6 PI2C: I2C中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit5 -- 保留，须为0。
Bit4 PADC: ADC中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit3 PPWM: PWM中断优先级控制位
1= 设置为高级中断；
0= 设置为低级中断。
Bit2 PT5: TIMER5中断优先级控制位
1= 设置为高级中断；
0= 设置为低级中断。
Bit1~0 -- 保留，须为0。

6.5.3.4 中断优先级控制寄存器 EIP3

0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	PXTDET	--	--	PTOUCH	PLVD	PLED	--	--
R/W	R/W	R	R	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 PXTDET 晶振停振中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit6-Bit5 -- 保留，须为0。
Bit4 PTOUCH TOUCH中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit3 PLVD: LVD中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit2 PLED: LED中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit1-Bit0 -- 保留，须为0。

6.5.4 中断标志位寄存器

6.5.4.1 Timer0/1、INT0/1 中断标志位寄存器 TCON

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 TF1: Timer1 计数器溢出中断标志位；
1= Timer1计数器溢出，进入中断服务程序时硬件自动清零，也可软件清零；
0= Timer1 计数器无溢出。

Bit6 TR1: Timer1运行控制位；
1= Timer1 启动；
0= Timer1 关闭。

Bit5 TF0: Timer0计数器溢出中断标志位；
1= Timer0计数器溢出，进入中断服务程序时硬件自动清零，也可软件清零；
0= Timer0计数器无溢出。

Bit4 TR0: Timer0运行控制位；
1= Timer0启动。
0= Timer0关闭。

Bit3 IE1: 外部中断1标志位；
1= 外部中断1产生中断，进入中断服务程序时硬件自动清零，也可软件清零；
0= 外部中断1没有产生中断。

Bit2 IT1: 外部中断1触发方式控制位；
1= 下降沿触发；
0= 低电平触发。

Bit1 IE0: 外部中断0标志位；
1= 外部中断0产生中断，进入中断服务程序时硬件自动清零，也可软件清零；
0= 外部中断0没有产生中断。

Bit0 IT0: 外部中断0触发方式控制位；
1= 下降沿触发；
0= 低电平触发。

6.5.4.2 Timer2 中断标志位寄存器 T2IF

0xC9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IF	TF2	T2EXIF	--	--	--	--	T2C1IF	T2C0IF
R/W	R/W	R/W	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 TF2: Timer2 计数器溢出中断标志位；
1= Timer2 计数器溢出，需软件清零；
0= Timer2 计数器无溢出。
Bit6 T2EXIF: Timer2 外部加载标志位；
1= Timer2 的T2EX口产生下降沿，需软件清零；
0= --
Bit5~Bit2 -- 保留，须均为0。
Bit1 T2C1IF: Timer2 比较/捕获通道1标志位；
1= Timer2 比较通道1 {CCH1:CCL1}={TH2:TL2}或捕获通道1产生了捕获操作，需软件清零。
0= --
Bit0 T2C0IF: Timer2 比较/捕获通道0标志位；
1= Timer2 比较通道0{RLDH:RLDL}={TH2:TL2}或捕获通道0产生了捕获操作，需软件清零。
0= --

6.5.4.3 外设中断标志位寄存器 EIF2

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	TF5	--	--
R/W	R	R	R	R/W	R	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI总中断指示位，只读；
1= SPI产生中断，（清除具体的中断标志位后，此位自动清除）；
0= SPI未产生中断。
Bit6 I2CIF: I^2C总中断指示位，只读；
1= I^2C产生中断，（清除具体的中断标志位后，此位自动清除）；
0= I^2C未产生中断。
Bit5 -- 保留，须为0。
Bit4 ADCIF: ADC中断标志位；
1= ADC转换完成，需软件清零；
0= ADC转换未完成。
Bit3 PWMIF: PWM总中断指示位，只读；
1= PWM产生中断，（清除具体的中断标志位后，此位自动清除）；
0= PWM未产生中断。
Bit2 TF5: Timer5定时器溢出中断标志位；
1= Timer5定时器溢出，需软件清零；
0= Timer5定时器无溢出。
Bit1~0 -- 保留，须为0

6.5.4.4 SPI 中断标志位寄存器 SPSR

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	SPISIF	WCOL	--	--	--	--	--	SSCEN
R/W	R	R	--	--	--	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 SPISIF: SPI传输完毕中断标志位，只读；
1= SPI传输完毕（先读SPSR，再读/写SPDR后清零）；
0= SPI未传输完毕。
Bit6 WCOL: SPI写冲突中断标志位，只读；
1= SPI传输未完成时产生写SPDR操作冲突（先读SPSR，再读/写SPDR后清零）；
0= 无写冲突。
Bit5~Bit1 -- 保留，须均为0。
Bit0 SSCEN: SPI主控模式NSS输出控制位。
1= SPI处于空闲状态时，NSS输出高电平；
0= NSS输出寄存器SSCR的内容。

6.5.4.5 UART 控制寄存器 SCONn

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCONn	UnSM0	UnSM1	UnSM2	UnREN	UnTB8	UnRB8	TIn	RIn
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 SCON0 地址 0x98；寄存器 SCON1 地址 0xEA。

Bit7~Bit2 UnSM0、UnSM1、UnSM2、UnREN、UnTB8、UnRB8n，详见UARTn功能说明

Bit1 TIn: 发送中断标志位（需要软件清零）；
1= 说明发送缓冲器已空，可以发送一下帧数据。
0= --

Bit0 RIn: 接收中断标志位（需要软件清零）；
1= 说明接收缓冲器已满，读取后可以接收下一帧数据。
0= --

TIn 与 RIn 占用同一中断向量，需要查询来判断是接收中断还是发送中断。

6.5.4.6 外部扩展中断组 0 标志位寄存器 EI0IF

0xB4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EI0IF	-	-	-	-	EINT3IF	EINT2IF	EINT1IF	EINT0IF
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7-Bit4 -- 保留，须为0。

Bit3-Bit0 EINTiIF: 外部扩展中断组0标志位(i=0-3)；
1= EINTi产生中断，需软件清零；
0= EINTi未产生中断。

6.5.5 中断标志位的清除操作

中断标志位的清除操作分为以下几种：

- 硬件自动清除（需进入中断服务程序）
- 软件清除

1) 硬件自动清除的标志位

支持硬件自动清除的位有 INT0、INT1、T0、T1 产生的中断标志位。硬件自动清除标志的条件为：打开总中断使能位 EA=1，且打开相应的中断使能位，在产生中断后系统进入相应中断服务程序，标志位自动清除。如中断使能关闭，则这些标志位也可使用软件进行清除操作。

2) 软件清除的标志位

系统中存在只能用软件清除的标志位。这些标志位在进入中断服务程序后不会自动清除，需要软件写 0 清除。否则在退出中断服务程序后又会再次进入中断服务程序。

软件清除操作需要注意：当多个中断标志位在同一个寄存器中，这些标志位产生的时刻相互无关联时，不建议用读-修改-写操作来进行。比如 PWMDIF 中断标志位寄存器，其中包含 PG0-PG5 通道的向下比较中断，这些中断标志位互不相关。当 PG0 产生向下比较中断时，PWMDIF 的值为 0x01，进入中断服务程序后进行读-修改-写操作清除该位

PWMDIF &= 0xFE；

该操作具体实现为先将 PWMDIF 的值读回到 CPU，然后再进行运算，最后送回到 PWMDIF。如果 PG1 的中断标志位 PWMDIF[1]在 CPU 读之后置 1，而 PWMDIF[1]读取时为 0，则进行运算后，送回到 PWMDIF[1]也为 0，此时将会清除 PG1 已经产生的向上中断标志位 PWMDIF[1]。

清除上述类型的中断标志位，建议直接写 0 操作，其他不相关的标志位写 1：PWMDIF = 0xFE。该操作对不相关的中断标志写 1 无实际影响。

6.5.6 外部中断滤波控制寄存器

6.5.6.1 中断屏蔽寄存器 INTFC

F0E4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTFC	-	-	-	-	INTSNUS1	INTSNUS0	INTSCKS1	INTSCKS0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit4 - 保留，必须为0；
Bit3~Bit2 INTSNUS<1:0>: 外部中断滤波采样个数选择（对INT0/INT1/ EINT0-EINT3同时生效）；
00= 1；
01= 2；
10= 3；
11= 4。
Bit1~Bit0 INTSCKS<1:0>: 外部中断滤波采样时钟选择（对INT0/INT1/EINT0-EINT3同时生效）；
00= $F_{SYS}/1$；
01= $F_{SYS}/4$；
10= $F_{SYS}/16$；
11= $F_{SYS}/64$。

6.5.7 调试模式下特殊中断标志位

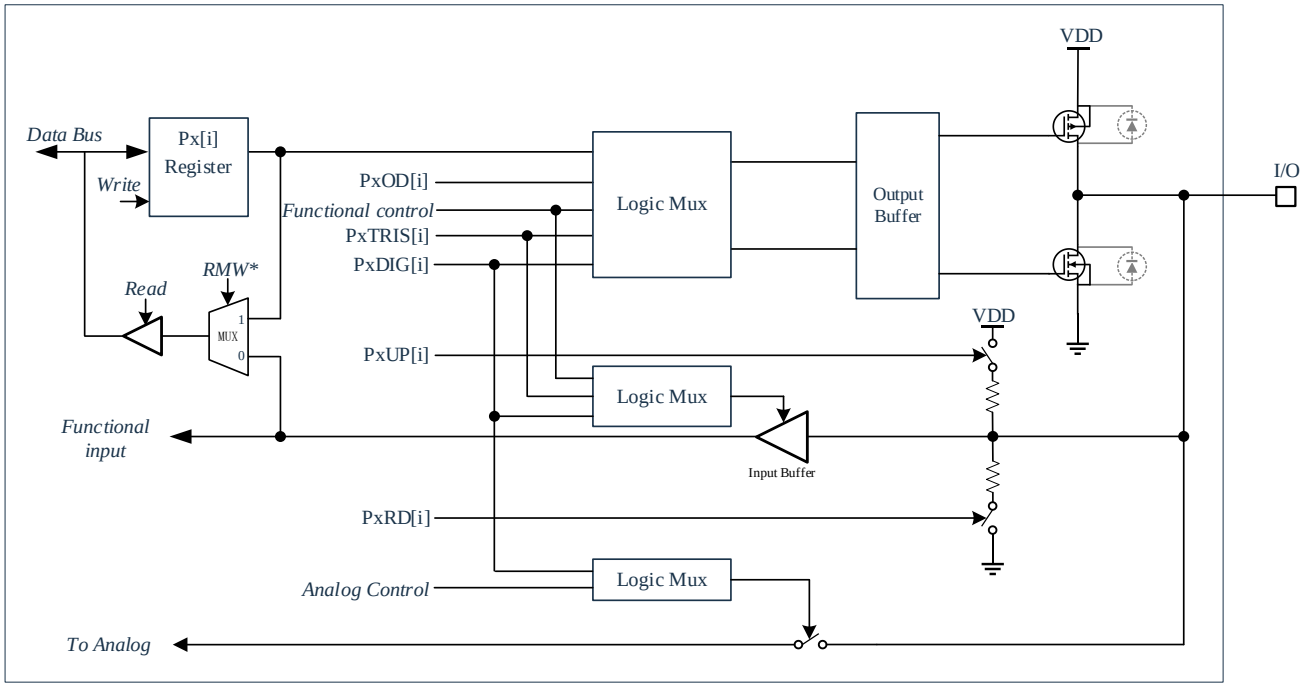
系统中有标志位并不是写 0 到该标志位清零，而需要读/写其他寄存器来清除标志位。

在调试态下，断点执行，单步运行或停止操作后，仿真器会从系统中读出所有的寄存器的值到仿真软件，仿真器的读/写操作与正常模式下的读/写完全一样。

所以在调试过程中，会出现暂停后，应该出现置 1 的中断标志位，但在观察窗口中显示为 0。在调试模式下需要注意。

7. I/O 端口

7.1 端口结构



RMW: 读-修改-写指令，即执行RMW指令时读取的是Px[i]寄存器的值，其他指令读取引脚电平。*

7.2 GPIO 功能

芯片有 4 组 I/O 端口：PORT0、PORT1、PORT2、PORT5。

PORTx 是双向端口。其状态通过 Px、PxDIG、PxTRIS、PxUP、PxOD 寄存器配置。

PORTx 作输出口时，写 Px 寄存器将会写入端口锁存器，所有写操作都是读－修改－写操作。因此，写一个端口就意味着先读该端口的引脚电平，然后修改读到的值，最后再将改好的值写入端口数据锁存器。

当将 PORTx 引脚用作模拟输入时，I/O 引脚读为 0。

7.2.1 PORTx 数字功能控制寄存器 PxDIG

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxDIG	PxDIG7	PxDIG6	PxDIG5	PxDIG4	PxDIG3	PxDIG2	PxDIG1	PxDIG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

寄存器 P0DIG 地址：F060H；寄存器 P1DIG 地址：F061H；寄存器 P2DIG 地址：F062H；寄存器 P5DIG 地址：F065H。

P0DIG 的复位值为 0x2C，P1DIG/P2DIG 复位值为 0x00；P5DIG 的复位值为 0x18。

Bit7~Bit0　　PxDIG<7:0>:　数字功能控制位；

1=　引脚被配置为数字功能；

0=　引脚被配置为模拟功能(关闭GPIO端口的输入/输出功能)。

注：P02.P03，P53，P54 默认为数字输入。

7.2.2 PORTx 方向寄存器 PxTRIS

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxTRIS	PxTRIS7	PxTRIS6	PxTRIS5	PxTRIS4	PxTRIS3	PxTRIS2	PxTRIS1	PxTRIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0TRIS 地址：0x9A；寄存器 P1TRIS 地址：0x9B；寄存器 P2TRIS 地址：0x9C；寄存器 P5TRIS 地址：0x9F。

Bit7~Bit0 PxTRIS<7:0>: 三态控制位；
1= 引脚被配置为输出；
0= 引脚被配置为输入（三态）。

7.2.3 PORTx 数据寄存器 Px

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Px	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

寄存器 P0 地址：0x80；寄存器 P1 地址：0x90；寄存器 P2 地址：0xA0；寄存器 P3 地址：0xB0。

Bit7~Bit0 Px<7:0>: Px I/O 引脚位；
1= 端口引脚电平>V_{IH}（正向阈值电压）；
0= 端口引脚电平<V_{IL}（负向阈值电压）。

7.2.4 PORTx 上拉电阻寄存器 PxUP

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxUP	PxUP7	PxUP6	PxUP5	PxUP4	PxUP3	PxUP2	PxUP1	PxUP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0UP 地址：F068H；寄存器 P1UP 地址：F069H；寄存器 P2UP 地址：F06AH；寄存 P5UP 地址：F06DH。

Bit7~Bit0 PxUP<7:0>: 上拉电阻控制位(独立控制，与IO其他配置及复用功能无关)；
1= 引脚上拉电阻打开；
0= 引脚上拉电阻关闭。

7.2.5 PORTx 下拉电阻寄存器 PxRD

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxRD	PxRD7	PxRD6	PxRD5	PxRD4	PxRD3	PxRD2	PxRD1	PxRD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0RD 地址：0F078H；寄存器 P1RD 地址：0F079H。寄存器 P2RD 地址：0F07AH；寄存器 P5RD 地址：0F07DH。

Bit7~Bit0 PxRD<7:0>: 下拉电阻控制位(独立控制，与IO其他配置及复用功能无关)；
1= 引脚下拉电阻打开；
0= 引脚下拉电阻关闭。

7.2.6 PORT x 开漏控制寄存器 PxOD

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxOD	PxOD7	PxOD6	PxOD5	PxOD4	PxOD3	PxOD2	PxOD1	PxOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0OD 地址：F070H；寄存器 P1OD 地址：F071H；寄存器 P2OD 地址：F072H；寄存器 P5OD 地址：F075H。

Bit7~Bit0 PxOD<7:0>: 开漏控制位(独立控制，与IO其他配置及复用功能无关)；

1= 引脚被配置为开漏状态（输出为开漏输出）；

0= 引脚被配置为正常状态（输出为推挽输出）。

7.3 复用功能

7.3.1 端口复用功能表

管脚多种功能共享，每个 I/O 端口可灵活配置为指定的模拟功能或者数字功能。

设置 PxDIG 寄存器为 0，将对应口设置为模拟功能口（模拟通道 ANn/TKn 选择详见对应章节）。设置 PxDIG 寄存器为 1，将对应口设置为数字功能口。

数字功能口用于外部输入时，可通过复用功能配置为 RXD1/RXD0/SCL/SDA 等，或者通用 IO 设置为输入（PxTRIS 设置为 0）生效，同时若设置的功能具有多个引脚可分配的，需要将对应 PS_xx 设置到相应引脚上。

数字功能口用于复用功能时，通过对应功能寄存器 PS_xx 设置，设置 PxTRIS 寄存器，根据应用需求可通过 PxUP 寄存器配置上拉电阻，通过 PxOD 寄存器配置开漏功能。

数字功能口用于通用 IO 时，通过 PxTRIS 寄存器设置 IO 的输入/输出，PxUP 寄存器配置上拉电阻，通过 PxOD 寄存器配置开漏功能。

注：管脚用于数字功能时，复用功能的优先级高于通用 IO 的功能。即管脚用于通用 IO 功能时，须禁止对应口的所用复用功能选择。

功能配置如下表所示：

	AD	TK	LED矩阵	LED点阵	HSE/LSE	INT0	INT1	EINT0	EINT1	EINT2	EINT3	T0	T1	T2	IIC	UART0	UART1	SPI	T2_O	PWM	CLKO	SWE	DEBUG
P00	AN0	TK0	COM0	LED0				EINT0	EINT1	EINT2	EINT3				SCL		TXD1/RXD1	SCLK	CC0	PG0			
P01	AN1	TK1	COM1	LED1				EINT0	EINT1	EINT2	EINT3				SDA		TXD1/RXD1	MOSI	CC1	PG1			
P02	AN2	TK2	COM2	LED2				EINT0	EINT1	EINT2	EINT3	T0G	T1G	T2EX			TXD1/RXD1	MISO		PG2			DSDA2
P03	AN3	TK3	COM3	LED3				EINT0	EINT1	EINT2	EINT3	T0	T1	T2			TXD1/RXD1	NSS(NSSO0)		PG3			DSCK2
P04	AN4	TK4	COM4/SEG0	LED4				EINT0	EINT1	EINT2	EINT3					TXD0/RXD0	TXD1/RXD1	NSS(NSSO1)		PG4			
P05	AN5	TK5	COM5/SEG1	LED5				EINT0	EINT1	EINT2	EINT3					TXD0/RXD0	TXD1/RXD1	NSS(NSSO2)		PG5			
P06	AN6	TK6	COM6/SEG2	LED6				EINT0	EINT1	EINT2	EINT3						TXD1/RXD1	NSS(NSSO3)					
P07	AN7	TK7	COM7/SEG3	LED7				EINT0	EINT1	EINT2	EINT3						TXD1/RXD1				CLO		
P10	AN8	TK8	SEG4	LED8				EINT0	EINT1	EINT2	EINT3				SCL		TXD1/RXD1	SCLK		PG0	CLO		
P11	AN9	TK9	SEG5					EINT0	EINT1	EINT2	EINT3				SDA		TXD1/RXD1	MOSI		PG1			
P12	AN10	TK10	SEG6					EINT0	EINT1	EINT2	EINT3	T0G	T1G	T2EX			TXD1/RXD1	MISO		PG2			
P13	AN11	TK11	SEG7					EINT0	EINT1	EINT2	EINT3	T0	T1	T2			TXD1/RXD1	NSS(NSSO0)		PG3			
P14	AN12	TK12	SEG8					EINT0	EINT1	EINT2	EINT3			CAP0		TXD0/RXD0	TXD1/RXD1	NSS(NSSO1)	CAP0	PG4			
P15	AN13	TK13	SEG9					EINT0	EINT1	EINT2	EINT3			CAP1		TXD0/RXD0	TXD1/RXD1	NSS(NSSO2)	CAP1	PG5			
P16	AN14	TK14	SEG10					EINT0	EINT1	EINT2	EINT3						TXD1/RXD1	NSS(NSSO3)					
P17	AN15	TK15	SEG11					EINT0	EINT1	EINT2	EINT3						TXD1/RXD1	-			CLO		
P20	AN16	TK16	SEG12					EINT0	EINT1	EINT2	EINT3				SCL		TXD1/RXD1	SCLK	CC0	PG0			
P21	AN17	TK17	SEG13					EINT0	EINT1	EINT2	EINT3				SDA		TXD1/RXD1	MOSI	CC1	PG1			
P22	AN18	TK18	SEG14					EINT0	EINT1	EINT2	EINT3	T0G	T1G	T2EX			TXD1/RXD1	MISO		PG2			
P23	AN19	TK19	SEG15					EINT0	EINT1	EINT2	EINT3	T0	T1	T2			TXD1/RXD1	NSS(NSSO0)		PG3			
P24	AN20	TK20	SEG16					EINT0	EINT1	EINT2	EINT3					TXD0/RXD0	TXD1/RXD1	NSS(NSSO1)		PG4			
P25	AN21	TK21	SEG17					EINT0	EINT1	EINT2	EINT3					TXD0/RXD0	TXD1/RXD1	NSS(NSSO2)		PG5			
P26	AN22	TK22	SEG18			INT0		EINT0	EINT1	EINT2	EINT3						TXD1/RXD1	NSS(NSSO3)					
P27	AN23	TK23	SEG19				INT1	EINT0	EINT1	EINT2	EINT3						TXD1/RXD1	-					
P50	AN39	TK39	SEG20					EINT0	EINT1	EINT2	EINT3				SCL		TXD1/RXD1	SCLK		PG0			
P51	AN40	TK40	SEG21		OSCIN			EINT0	EINT1	EINT2	EINT3				SDA		TXD1/RXD1	MOSI		PG1			
P52	AN41	TK41	SEG22		OSCOUT			EINT0	EINT1	EINT2	EINT3						TXD1/RXD1	MISO		PG2			
P53	AN42	TK42	SEG23					EINT0	EINT1	EINT2	EINT3						TXD1/RXD1	NSS(NSSO0)		PG3		SWE	DSDA1
P54	AN43	TK43	SEG24					EINT0	EINT1	EINT2	EINT3					TXD0/RXD0	TXD1/RXD1	NSS(NSSO1)		PG4			DSCK1
P55	AN44	TK44	SEG25					EINT0	EINT1	EINT2	EINT3					TXD0/RXD0	TXD1/RXD1	NSS(NSSO2)		PG5			

7.3.2 端口复用功能配置

所有的 GPIO 均可复用成特定功能，如 GPIO 中断输入，INT0/1 中断输入，EPWM 输出，SPI，IIC，UART 等功能。管脚复用详细功能见管脚封装或复用功能列表。任何一个 GPIO 配置成复用功能需按照特有的配置方式进行。详见复用功能寄存器列表。

		寄存器	优先级	功能配置方法		
0		PS_EINT0		PxDIG=1	PxTRIS=0	PS_EINT0=0xmn
1		PS_EINT1		PxDIG=1	PxTRIS=0	PS_EINT1=0xmn
2		PS_EINT2		PxDIG=1	PxTRIS=0	PS_EINT2=0xmn
3		PS_EINT3		PxDIG=1	PxTRIS=0	PS_EINT3=0xmn
4		PS_T0		PxDIG=1	PxTRIS=0	PS_T0=0xn
5		PS_T0G		PxDIG=1	PxTRIS=0	PS_T0G=0xn
6		PS_T1		PxDIG=1	PxTRIS=0	PS_T1=0xn
7		PS_T1G		PxDIG=1	PxTRIS=0	PS_T1G=0xn
8		PS_T2		PxDIG=1	PxTRIS=0	PS_T2=0xn
9		PS_T2EX		PxDIG=1	PxTRIS=0	PS_T2EX=0xn
10		PS_RXD0		PxDIG=1	PxTRIS=0	PS_RXD0=0xn
11		PS_RXD1		PxDIG=1	PxTRIS=0	PS_RXD1=0xmn
12		PS_SCL	高	PxDIG=1	PxTRIS=1	PS_SCL=0xn
13		PS_SDA		PxDIG=1	PxTRIS=1	PS_SDA=0xn
14		PS_TXD0		PxDIG=1	PxTRIS=1	PS_TXD0=0xn
15		PS_TXD1		PxDIG=1	PxTRIS=1	PS_TXD1=0xmn
16		PS_SCLKOI		PxDIG=1	主机 PxTRIS=1 从机 PxTRIS=0	PS_SCLKOI=0xn
17		PS_MOSI		PxDIG=1	主机 PxTRIS=1 从机 PxTRIS=0	PS_MOSI=0xn
18		PS_MISO		PxDIG=1	主机 PxTRIS=0 从机 PxTRIS=1	PS_MISO=0xn
19		PS_NSS0OI		PxDIG=1	主机 PxTRIS=1 从机 PxTRIS=0	PS_NSS0OI=0xn
20	输出	PS_NSS1OI		PxDIG=1	主机 PxTRIS=1 从机 PxTRIS=0	PS_NSS1OI=0xn
21		PS_NSS2OI		PxDIG=1	主机 PxTRIS=1 从机 PxTRIS=0	PS_NSS2OI=0xn
22		PS_NSS3OI		PxDIG=1	主机 PxTRIS=1 从机 PxTRIS=0	PS_NSS3OI=0xn
23		PS_CC0		PxDIG=1	PxTRIS=1	PS_CC0=0xn
24		PS_CC1		PxDIG=1	PxTRIS=1	PS_CC1=0xn
25		PS_PG0		PxDIG=1	PxTRIS=1	PS_PG0=0xn
26		PS_PG1		PxDIG=1	PxTRIS=1	PS_PG1=0xn
27		PS_PG2		PxDIG=1	PxTRIS=1	PS_PG2=0xn
28		PS_PG3		PxDIG=1	PxTRIS=1	PS_PG3=0xn
29		PS_PG4		PxDIG=1	PxTRIS=1	PS_PG4=0xn
30		PS_PG5		PxDIG=1	PxTRIS=1	PS_PG5=0xn
31		PS_CLO	低	PxDIG=1	PxTRIS=1	PS_CLO=0xn

7.3.2.1 PS_EINT0 外部中断扩展输入 EINT0 通道端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_EINT0	PS_EINT07	PS_EINT06	PS_EINT05	PS_EINT04	PS_EINT03	PS_EINT02	PS_EINT01	PS_EINT00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit0 PS_EINT0mn<7:0>: 外部中断扩展输入EINT0通道GPIO选择（m=0,1,2,5；n=0~7）

0x00= 分配到P00；
0x01= 分配到P01；
… …
0x07= 分配到P07；
0x10= 分配到P10；
0x11= 分配到P11；
… …
0x17= 分配到P17；
… …
0x55= 分配到P55；
其他： 保留

7.3.2.2 PS_EINT1 外部中断扩展输入 EINT1 通道端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_EINT1	PS_EINT17	PS_EINT16	PS_EINT15	PS_EINT14	PS_EINT13	PS_EINT12	PS_EINT11	PS_EINT10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit0 PS_EINT1mn<7:0>: 外部中断扩展输入EINT1通道GPIO选择（m=0,1,2,5；n=0~7）

0x00= 分配到P00；
0x01= 分配到P01；
… …
0x07= 分配到P07；
0x10= 分配到P10；
0x11= 分配到P11；
… …
0x17= 分配到P17；
… …
0x55= 分配到P55；
其他： 保留

7.3.2.3 PS_EINT2 外部中断扩展输入 EINT2 通道端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_EINT2	PS_EINT27	PS_EINT26	PS_EINT25	PS_EINT24	PS_EINT23	PS_EINT22	PS_EINT21	PS_EINT20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit0 PS_EINT2mn<7:0>: 外部中断扩展输入EINT2通道GPIO选择（m=0,1,2,5；n=0~7）

0x00= 分配到P00；
0x01= 分配到P01；
… …
0x07= 分配到P07；
0x10= 分配到P10；
0x11= 分配到P11；
… …
0x17= 分配到P17；
… …
0x55= 分配到P55；
其他： 保留

7.3.2.4 PS_EINT3 外部中断扩展输入 EINT3 通道端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_EINT3	PS_EINT37	PS_EINT36	PS_EINT35	PS_EINT34	PS_EINT33	PS_EINT32	PS_EINT31	PS_EINT30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit0 PS_EINT3mn<7:0>: 外部中断扩展输入EINT3通道GPIO选择（m=0,1,2,5；n=0~7）

0x00= 分配到P00；
0x01= 分配到P01；
… …
0x07= 分配到P07；
0x10= 分配到P10；
0x11= 分配到P11；
… …
0x17= 分配到P17；
… …
0x55= 分配到P55；
其他： 保留

7.3.2.5 PS_T0 TIMER0 外部时钟输入端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_T0						PS_T02	PS_T01	PS_T00
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_T0<2:0>: TIMER0外部时钟输入端口选择
0x00= 选择P03；
0x01= 选择P13；
0x02= 选择P23；
其他： 保留

7.3.2.6 PS_T0G TIMER0 门控输入端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_T0G						PS_T0G2	PS_T0G1	PS_T0G0
R/W	~~R~~	~~R~~	~~R~~	~~R~~	~~R~~	R/W	R/W	R/W
复位值	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_T0G<2:0>: TIMER0门控输入端口选择
0x00= 选择P02；
0x01= 选择P12；
0x02= 选择P22；
其他： 保留

7.3.2.7 PS_T1 TIMER1 外部时钟输入端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_T1						PS_T12	PS_T11	PS_T10
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_T1<2:0>: TIMER1外部时钟输入端口选择
0x00= 选择P03；
0x01= 选择P13；
0x02= 选择P23；
其他： 保留

7.3.2.8 PS_T1G TIMER1 门控输入端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_T1G						PS_T1G2	PS_T1G1	PS_T1G0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_T1G<2:0>: TIMER0门控输入端口选择
0x00= 选择P02；
0x01= 选择P12；
0x02= 选择P22；
其他： 保留

7.3.2.9 PS_T2 TIMER2 外部事件或门控输入端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_T2						PS_T22	PS_T21	PS_T20
R/W	~~R~~	~~R~~	~~R~~	~~R~~	~~R~~	R/W	R/W	R/W
复位值	~~0~~	~~0~~	~~0~~	~~0~~	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_T2<2:0>: TIMER2外部事件或门控输入端口选择
0x00= 选择P03；
0x01= 选择P13；
0x02= 选择P23；
其他： 保留

7.3.2.10 PS_T2EX TIMER2 下降沿自动重装输入端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_T2EX						PS_T2EX2	PS_T2EX1	PS_T2EX0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_T2EX<2:0>: TIMER2下降沿自动重装输入端口选择
0x00= 选择P02；
0x01= 选择P12；
0x02= 选择P22；
其他： 保留

7.3.2.11 PS_RXD0 UART0 RXD 输入端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_RXD0	--	--	--	--	PS_RXD03	PS_RXD02	PS_RXD01	PS_RXD00
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	1	1	1	1

Bit7~Bit3 -- 保留，须为0

Bit2~Bit0 PS_RXD0<3:0>: UART0 RXD输入端口选择

0x00= 选择P04；
0x01= 选择P05；
0x02= 选择P14；
0x03= 选择P15；
0x04= 选择P24；
0x05= 选择P25；
0x06= 选择P54；
0x07= 选择P55；
其他： 保留

7.3.2.12 PS_RXD1 UART1 RXD 输入端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_RXD1	--	PS_RXD16	PS_RXD15	PS_RXD14	PS_RXD13	PS_RXD12	PS_RXD11	PS_RXD10
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	1	1	1	1	1	1

Bit7 --: 保留，须为0

Bit7~Bit0 PS_RXD1mn<6:0>: UART1 RXD输入端口选择（m=0,1,2,5；n=0~7）

0x00= 分配到P00；
0x01= 分配到P01；
… …
0x07= 分配到P07；
0x10= 分配到P10；
0x11= 分配到P11；
… …
0x17= 分配到P17；
… …
0x55= 分配到P55；
其他： 保留

7.3.2.13 PS_SCL IIC 通信 SCL 端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_SCL	--	--	--	--	--	PS_SCL2	PS_SCL1	PS_SCL0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 - 保留，须为0
Bit2~Bit0 PS_SCLmn<2:0>: IIC通信的SCL端口选择
0x00= 分配到P00；
0x01= 分配到P10；
0x02= 分配到P20；
0x03= 分配到P50；
其他： 保留

7.3.2.14 PS_SDA IIC 通信 SDA 端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_SDA	--	--	--	--	--	PS_SDA2	PS_SDA1	PS_SDA0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 - 保留，须为0
Bit2~Bit0 PS_SCLmn<2:0>: IIC通信的SCL端口选择
0x00= 分配到P01；
0x01= 分配到P11；
0x02= 分配到P21；
0x03= 分配到P51；
其他： 保留

7.3.2.15 PS_TXD0 UART0 TXD 输出端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_TXD0						PS_TXD02	PS_TXD01	PS_TXD00
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_TXD0<3:0>: UART0 TXD输出端口选择
0x00= 选择P04；
0x01= 选择P05；
0x02= 选择P14；
0x03= 选择P15；
0x04= 选择P24；
0x05= 选择P25；
0x06= 选择P54；
0x07= 选择P55；
其他： 保留

7.3.2.16 PS_TXD1 UART1 TXD 输出端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_TXD1	--	PS_TXD16	PS_TXD15	PS_TXD14	PS_TXD13	PS_TXD12	PS_TXD11	PS_TXD10
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	1	1	1	1	1	1

Bit7 -- 保留，须为0

Bit6~Bit0 PS_TXD1mn<6:0>: UART1 TXD输入端口选择（m=0,1,2,5；n=0~7）

0x00= 分配到P00；
0x01= 分配到P01；
… …
0x07= 分配到P07；
0x10= 分配到P10；
0x11= 分配到P11；
… …
0x17= 分配到P17；
… …
0x55= 分配到P55；
其他： 保留

7.3.2.17 PS_SCLKOI SPI SCLK 输入/输出端口选择寄存器

F0F0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_SCLKOI						PS_SCLKOI 2	PS_SCLKOI 1	PS_SCLKOI 0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0

Bit2~Bit0 PS_SCLKOI<2:0>: SPI SCLK输入/输出端口选择

0x00= 选择P00；
0x01= 选择P10；
0x02= 选择P20；
0x03= 选择P50；
其他： 保留

7.3.2.18 PS_MOSI SPI 主机模式数据输出/从机模式数据输入端口选择寄存器

F0F1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_MOSI						PS_MOSI2	PS_MOSI1	PS_MOSI0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0

Bit2~Bit0 PS_MOSI<2:0>: SPI主机模式数据输出/从机模式数据输入端口选择

0x00= 选择P01；
0x01= 选择P11；
0x02= 选择P21；
0x03= 选择P51；
其他： 保留

7.3.2.19 PS_MISO SPI 主机模式数据输入/从机模式数据输出端口选择寄存器

F0F2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_MISO						PS_MISO2	PS_MISO1	PS_MISO0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_MISO<2:0>: SPI主机模式数据输入/从机模式数据输出端口选择
0x00= 选择P02；
0x01= 选择P12；
0x02= 选择P22；
0x03= 选择P52；
其他： 保留

7.3.2.20 PS_NSS0OI SPI 从机 0 片选端口选择寄存器

F0F3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_NSS0OI						PS_NSS0OI2	PS_NSS0OI1	PS_NSS0OI0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_ NSS0OI<2:0>: SPI从机0片选端口选择
0x00= 选择P03；
0x01= 选择P13；
0x02= 选择P23；
0x03= 选择P53；
其他： 保留

7.3.2.21 PS_NSS1OI SPI 从机 1 片选端口选择寄存器

F0F4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_NSS1OI						PS_NSS1OI2	PS_NSS1OI1	PS_NSS1OI0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_ NSS1OI<2:0>: SPI从机1片选端口选择
0x00= 选择P04；
0x01= 选择P14；
0x02= 选择P24；
0x03= 选择P54；
其他： 保留

7.3.2.22 PS_NSS2OI SPI 从机 2 片选端口选择寄存器

F0F5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_NSS2OI						PS_NSS2OI2	PS_NSS2OI1	PS_NSS2OI0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_ NSS2OI<2:0>: SPI从机2片选端口选择
0x00= 选择P05；
0x01= 选择P15；
0x02= 选择P25；
0x03= 选择P55；
其他： 保留

7.3.2.23 PS_NSS3OI SPI 从机 3 片选端口选择寄存器

F0F6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_NSS3OI						PS_NSS3OI2	PS_NSS3OI1	PS_NSS3OI0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_ NSS3OI<2:0>: SPI从机3片选端口选择
0x00= 选择P06；
0x01= 选择P16；
0x02= 选择P26；
其他： 保留

7.3.2.24 PS_CC0 TIMER2 通道 0 端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_CC0						PS_CC02	PS_CC01	PS_CC00
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_CC0<2:0>: TIMER2通道0端口选择
0x00= 选择P00；
0x01= 选择P20；
其他： 保留

7.3.2.25 PS_CC1 TIMER2 通道 1 端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_CC1						PS_CC12	PS_CC11	PS_CC10
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_CC1<2:0>: TIMER2通道1端口选择
0x00= 选择P01；
0x01= 选择P21；
其他： 保留

7.3.2.26 PS_PG0 PWM 通道 0 端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_PG0						PS_PG02	PS_PG01	PS_PG00
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_PG0<2:0>: PWM通道0端口选择
0x00= 选择P00；
0x01= 选择P10；
0x02= 选择P20；
0x03= 选择P50；
其他： 保留

7.3.2.27 PS_PG1 PWM 通道 1 端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_PG1						PS_PG12	PS_PG11	PS_PG10
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_PG1<2:0>: PWM通道1端口选择
0x00= 选择P01；
0x01= 选择P11；
0x02= 选择P21；
0x03= 选择P51；
其他： 保留

7.3.2.28 PS_PG2 PWM 通道 2 端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_PG2						PS_PG22	PS_PG21	PS_PG20
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_PG2<2:0>: PWM通道2端口选择
0x00= 选择P02；
0x01= 选择P12；
0x02= 选择P22；
0x03= 选择P52；
其他： 保留

7.3.2.29 PS_PG3 PWM 通道 3 端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_PG3						PS_PG32	PS_PG31	PS_PG30
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_PG3<2:0>: PWM通道3端口选择
0x00= 选择P03；
0x01= 选择P13；
0x02= 选择P23；
0x03= 选择P53；
其他： 保留

7.3.2.30 PS_PG4 PWM 通道 4 端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_PG4						PS_PG42	PS_PG41	PS_PG40
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_PG4<2:0>: PWM通道4端口选择
0x00= 选择P04；
0x01= 选择P14；
0x02= 选择P24；
0x03= 选择P54；
其他： 保留

7.3.2.31 PS_PG5 PWM 通道 5 端口选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_PG5						PS_PG52	PS_PG51	PS_PG50
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_PG5<2:0>: PWM通道5端口选择
0x00= 选择P05；
0x01= 选择P15；
0x02= 选择P25；
0x03= 选择P55；
其他： 保留

7.3.2.32 PS_CLO CLK 输出通道端口选择寄存器

F0F7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_CLO						PS_CLO	PS_CLO	PS_CLO
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit3 -- 保留，须为0
Bit2~Bit0 PS_CLO<2:0>: CLK输出通道端口选择
0x00= 选择P07；
0x01= 选择P17；
0x02= 选择P10；
其他： 保留

7.3.3 端口模拟功能配置

模拟功能配置如下表所示：

模拟功能	端口模式	通道使能控制	备注
ADC 通道	高阻态	ADCON1 控制	配置端口模式为高阻态 PxDIG=0，同时配置通道使能
TK 通道	高组态		配置端口模式为高阻态 PxDIG=0，同时配置通道使能

7.3.4 端口外部中断配置

端口的外部中断有：EINT0,EINT1,EINT2,EINT3。所有外部中断支持入口任意映射。

外部中断配置在 EI0IE、EI0CFGL、EI0IF 寄存器中进行配置。

如配置 EINT0 下降沿触发中断，配置如下：

```
P0DIG=0x01;          //设置 EINT0 为数字功能
P0TRIS=0x00;         //设置 EINT0 为输入
PS_EINT0 =0x00       //选择 P00
EI0CFGL= 0x02;       //配置 EINT0 为下降沿触发中断
EI0IE = 0x01;        //允许 EINT0 的中断功能
EA = 1;              //全局中断使能
```

使用外部扩展中断时，需要将端口配置成数字功能，且将方向置为输入口。或者复用功能为输入端口（如 RXD0 等）。

7.3.4.1 外部扩展中断组 0 控制寄存器低位 EI0CFGL

F0E0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EI0CFGL	EI3CFGL1	EI3CFGL0	EI2CFGL1	EI2CFGL0	EI1CFGL1	EI1CFGL0	EI0CFGL1	EI0CFGL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6 EI3CFGL<7:6>: 外部扩展中断EINT3控制位；
00= 外部中断输入禁止；
01= 上升沿触发中断；
10= 下降沿触发中断；
11= 上升沿或下降沿均触发中断。

Bit5~Bit4 EI2CFGL<5:4>: 外部扩展中断EINT2控制位；
00= 外部中断输入禁止；
01= 上升沿触发中断；
10= 下降沿触发中断；
11= 上升沿或下降沿均触发中断。

Bit3~Bit2 EI1CFGL<3:2>: 外部扩展中断EINT1控制位；
00= 外部中断输入禁止；
01= 上升沿触发中断；
10= 下降沿触发中断；
11= 上升沿或下降沿均触发中断。

Bit1~Bit0 EI0CFGL<1:0>: 外部扩展中断EINT0控制位；
00= 外部中断输入禁止；
01= 上升沿触发中断；
10= 下降沿触发中断；
11= 上升沿或下降沿均触发中断。

7.3.5 复用功能应用注意

1) 端口功能配置具有优先级限制，CONFIG 配置功能>触摸配置功能>模拟功能>复用功能>通用 IO。

例如将 P50 口配置为 NRST，同时配置为 PG0 功能。则 NRST 将有效，PG0 无效。

2) 若同一端口同时设置多个复用功能时，按照

SCL>SDA>TXD0>TXD1>PGx>CCx>CLO 的优先级从高到低选择。

3) 设置端口为 ADC 通道使能，需要软件设置端口模式为高阻态。

4) 输入功能与复用功能的结构相对独立。

例如将 P00 口配置为 RXD1，同时配置 P00 的 EINT0 中断触发方式为上升沿触发、中断使能，当 P00 输入从低电平变到高电平时，将触发 P00 的 GPIO 中断。

5）建议配置为 PGx，CCx 功能时，关闭上拉和开漏功能；配置为 SCL，SDA 功能时，开启上拉和开漏功能。

8. 看门狗定时器（WDT）

8.1 概述

看门狗定时器（Watch Dog Timer）是一个溢出时间可选、由 LSI 提供时钟源的片内定时器。

看门狗溢出复位是系统的一种保护设置，当系统运行到一个未知状态时，可通过看门狗来使系统复位，从而避免系统进入到无限期的死循环。看门狗溢出复位详见复位章节。

1) 看门狗功能关闭（WDKEY==AAH）

看门狗功能关闭，计数器值为 0 且未启动。

2) 看门狗溢出复位功能开启（WDKEY=其他值）

当 WDT 计数器记到设定溢出值时，产生 WDT 溢出复位，复位后系统重新配置。如果要避免产生 WDT 复位，则在产生 WDT 溢出复位前，可通过写寄存器 WDKEY=99H 清除看门狗计数器，计数器清除后，计数器重新从 0 开始计数，直到下一次计数器溢出。

3) 看门狗溢出复位功能强制开启（CONFIG 中设置）

此时复位功能强制开启（与 WDKEY 的值无关），仍可通过写 WDKEY=99H 清除 WDT 计数器。

WDT 的键控寄存器复位值为 00H，上电复位完成后 WDT 功能开启，如需关闭 WDT 功能，则需要设置相关寄存器 WDCON 与 WDKEY。操作 WDT 相关寄存器需要打开 WDT 时钟门控（PCGEN0[3]）。

修改 WDT 溢出时间建议在清除 WDT 计数器后立即修改 WTS<2:0>,或关闭 WDT 后再修改溢出时间，避免在修改 WTS<2:0>时可能引起的复位动作。

8.1.1 看门狗控制寄存器 WDCON

0x97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	--	WTS2	WTS1	WTS0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	0	0	0	0	0	0

Bit7 -- 保留，必须为0。

Bit6~Bit4 WTS<2:0>: WDT溢出时间选择位

000= 2^8* T_{LSI}；

001= 2^9* T_{LSI}；

010= 2^{10}* T_{LSI}；

011= 2^{11}* T_{LSI}；

100= 2^{12}* T_{LSI}；

101= 2^{14}* T_{LSI}；

110= 2^{15}* T_{LSI}；

111= 2^{17}* T_{LSI}。

Bit3~Bit0 -- 保留，必须为0。

8.1.2 看门狗溢出控制寄存器 WDKEY

0x8F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDKEY	WDKEY7	WDKEY6	WDKEY5	WDKEY4	WDKEY3	WDKEY2	WDKEY1	WDKEY0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 WDKEY<7:0> WDT 键控制位；
写 AAH= WDT 关闭；
写 99H= WDT 清除计数器。
其他值= WDT 启动计数器；

注：
1. 如果 CONFIG 中 WDT 配置为：ENABLE，则 WDT 强制启动计数器。且 WDT 的溢出复位功能强制打开。
2. 如果 CONFIG 中 WDT 配置为：SOFTWARE CONTROL，则可以使用 WDKEY 控制计数器和复位功能。

修改 WDCON 需要的指令序列（中间不能插入其他任何指令）：

```
CLR     EA
NOP
MOV     TA,#0AAH
MOV     TA,#055H
ORL     WDCON,#50H   //设置 WDT 溢出时间 WTS=101，即 256ms。
SETB    EA
```

修改 WDKEY 需要的指令序列（中间不能插入其他任何指令）：

```
CLR     EA
NOP
MOV     TA,#0AAH
MOV     TA,#055H
MOV     WDKEY,#AAH    //清除 WDT 计数器
SETB    EA
```

9. 定时计数器 0/1（Timer0/1）

定时器 0 与定时器 1 的类型和结构相似，是两个 16 位定时器。定时器 1 有三种工作模式，定时器 0 有四种工作模式。它们提供基本的定时和事件计数操作。

在“定时器模式”时，定时寄存器在定时器时钟使能时，每 12 个或 4 个系统周期递增一次。

在“计数器模式”时，定时器 0 的定时寄存器每当检测到相应的输入引脚上 T0 的下降沿就会递增；定时器 1 的定时寄存器每当检测到相应的输入引脚上（T1）的下降沿就会递增。

9.1 概述

定时器 0 和定时器 1 与标准的 8051 定时器完全兼容。

每个定时器由两个 8 位寄存器：{TH0（0x8C）:TL0（0x8A）}和{TH1（0x8D）:TL1（0x8B）}组成。定时器 0、1 在四种相同的模式下工作。Timer0 和 Timer1 模式如下所述。

模式	M1	M0	功能描述
0	0	0	THx[7:0]，TLx[4:0]组成 13 位定时/计数器
1	0	1	THx[7:0]，TLx[7:0]组成 16 位定时/计数器
2	1	0	TLx[7:0]组成 8 位自动重装定时/计数器，从 THx 中重装
3	1	1	TL0，TH0 为两个 8 位定时/计数器，Timer1 停止计数

寄存器 THx 和 TLx 为特殊功能寄存器，具有储存实际定时器值的功能。THx 和 TLx 可以通过模式选项级联成 13 位或 16 位寄存器。每收到一个内部时钟脉冲或外部定时器引脚发生状态跳变时，寄存器的值加 1。定时器将从预置寄存器所载入的值开始计数直到定时器计满溢出，此时会产生一个内部中断信号。如果选为定时器的自动重载模式，定时器的值将重置为预载寄存器的初始值并继续计数，否则定时器的值将重置为零。注意，为了得到定时/计数器的最大计算范围，预置寄存器必须先被清零。

9.2 相关寄存器

9.2.1 Timer0/1 模式寄存器 TMOD

0x89	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMOD	GATE1	CT1	T1M1	T1M0	GATE0	CT0	T0M1	T0M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 GATE1: 定时器1门控制位；
1= 使能；
0= 禁止。
Bit6 CT1: 定时器1定时/计数选择位；
1= 计数；
0= 定时。
Bit5~Bit4 T1M<1:0>: 定时器1模式选择位；
00= 模式0，13位定时/计数器；
01= 模式1，16位定时/计数器；
10= 模式2，8位自动重装定时/计数器；
11= 模式3，停止计数。
Bit3 GATE0: 定时器0门控制位；
1= 使能；
0= 禁止。
Bit2 CT0: 定时器0定时/计数选择位；
1= 计数；
0= 定时。
Bit1~ Bit0 T0M<1:0>: 定时器0模式选择位；
00= 模式0，13位定时/计数器；
01= 模式1，16位定时/计数器；
10= 模式2，8位自动重装定时/计数器；
11= 模式3，两个独立的8位定时/计数器。

9.2.2 Timer0/1 控制寄存器 TCON

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 TF1: Timer1计数器溢出中断标志位；
1= Timer1计数器溢出，进入中断服务程序硬件自动清零；
0= Timer1计数器无溢出。

Bit6 TR1: Timer1运行控制位；
1= Timer1启动；
0= Timer1关闭。

Bit5 TF0: Timer0计数器溢出中断标志位；
1= Timer0计数器溢出，进入中断服务程序硬件自动清零；
0= Timer0计数器无溢出。

Bit4 TR0: Timer0运行控制位；
1= Timer0启动；
0= Timer0关闭。

Bit3 IE1: 外部中断1标志位；
1= 外部中断1产生中断，进入中断服务程序硬件自动清零；
0= 外部中断1没有产生中断。

Bit2 IT1: 外部中断1触发方式控制位；
1= 下降沿触发；
0= 低电平触发。

Bit1 IE0: 外部中断0标志位；
1= 外部中断0产生中断，进入中断服务程序硬件自动清零；
0= 外部中断0没有产生中断。

Bit0 IT0: 外部中断0触发方式控制位；
1= 下降沿触发；
0= 低电平触发。

9.2.3 Timer0 数据寄存器低位 TL0

0x8A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit0　　TL0<7:0>:　定时器0低位数据寄存器（同时作为计数器低位）。

9.2.4 Timer0 数据寄存器高位 TH0

0x8C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　TH0<7:0>:　定时器0高位数据寄存器（同时作为计数器高位）。

9.2.5 Timer1 数据寄存器低位 TL1

0x8B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　TL1<7:0>:　定时器1低位数据寄存器（同时作为计数器低位）。

9.2.6 Timer1 数据寄存器高位 TH1

0x8D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　TH1<7:0>:　定时器1高位数据寄存器（同时作为计数器高位）。

9.3 Timer0/1 中断

Timer0/1 可以通过 IE 寄存器使能或关闭中断，还可以通过 IP 寄存器设置高/低优先级，其中断相关位如下。

9.3.1 中断屏蔽寄存器 IE

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 EA: 全局中断允许位；
1= 允许所有未被屏蔽的中断；
0= 禁止所有中断。

Bit6 ES1: UART1中断允许位；
1= 允许UART1中断；
0= 禁止UART1中断。

Bit5 ET2: TIMER2总中断允许位；
1= 允许TIMER2 所有中断；
0= 禁止TIMER2 所有中断。

Bit4 ES0: UART0中断允许位；
1= 允许UART0中断；
0= 禁止UART0中断。

Bit3 ET1: TIMER1中断允许位；
1= 允许TIMER1中断；
0= 禁止TIMER1中断。

Bit2 EX1: 外部中断1中断允许位；
1= 允许外部中断1中断；
0= 禁止外部中断1中断。

Bit1 ET0: TIMER0中断允许位；
1= 允许TIMER0中断；
0= 禁止TIMER0中断。

Bit0 EX0: 外部中断0中断允许位；
1= 允许外部中断0中断；
0= 禁止外部中断0中断。

9.3.2 中断优先级控制寄存器 IP

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 -- 保留，须为0。

Bit6 PS1: UART1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

Bit5 PT2: TIMER2中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

Bit4 PS0: UART0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

Bit3 PT1: TIMER1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

Bit2 PX1: 外部中断1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

Bit1 PT0: TIMER0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

Bit0 PX0: 外部中断0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

9.3.3 Timer0/1、INT0/1 中断标志位寄存器 TCON

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 TF1: Timer1计数器溢出中断标志位；
1= Timer1计数器溢出，进入中断服务程序时硬件自动清零，也可软件清零；
0= Timer1计数器无溢出。
Bit6 TR1: Timer1运行控制位；
1= Timer1启动；
0= Timer1关闭。
Bit5 TF0: Timer0计数器溢出中断标志位；
1= Timer0计数器溢出，进入中断服务程序时硬件自动清零，也可软件清零；
0= Timer0计数器无溢出。
Bit4 TR0: Timer0运行控制位；
1= Timer0启动；
0= Timer0关闭。
Bit3 IE1: 外部中断1标志位；
1= 外部中断1产生中断，进入中断服务程序时硬件自动清零，也可软件清零；
0= 外部中断1没有产生中断。
Bit2 IT1: 外部中断1触发方式控制位；
1= 下降沿触发；
0= 低电平触发。
Bit1 IE0: 外部中断0标志位；
1= 外部中断0产生中断，进入中断服务程序时硬件自动清零，也可软件清零；
0= 外部中断0没有产生中断。
Bit0 IT0: 外部中断0触发方式控制位；
1= 下降沿触发；
0= 低电平触发。

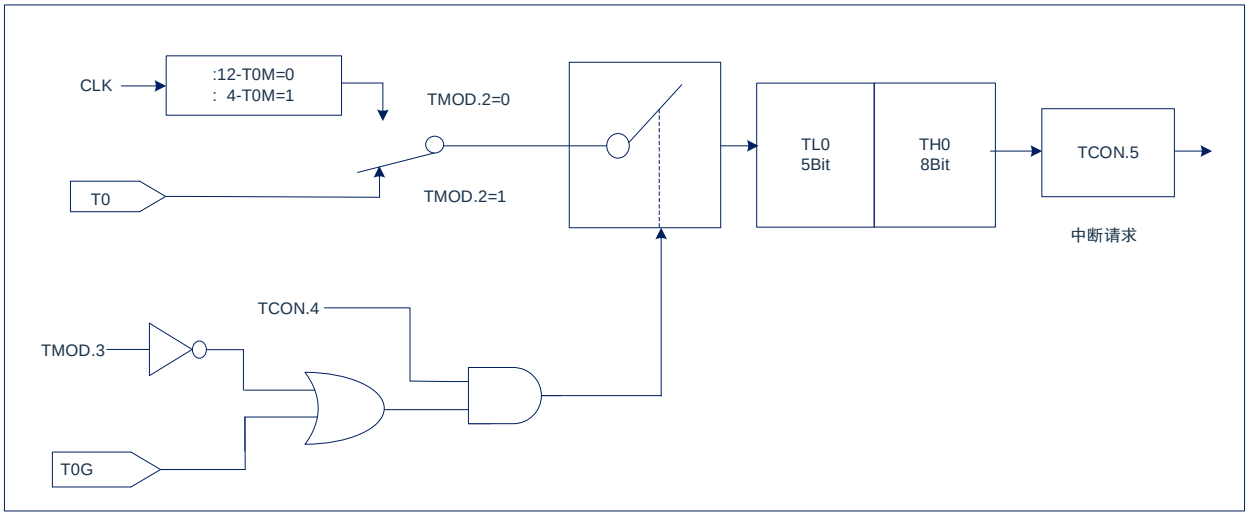
产生中断的标志位可以通过软件清零，与通过硬件清零的结果相同。也就是说，可以通过软件来产生中断（不建议通过写标志位来产生中断）或者取消待处理的中断。

TF0，TF1 标志位在没有开启中断的情况下，可以通过写 0 清除。

9.4 Timer0 工作模式

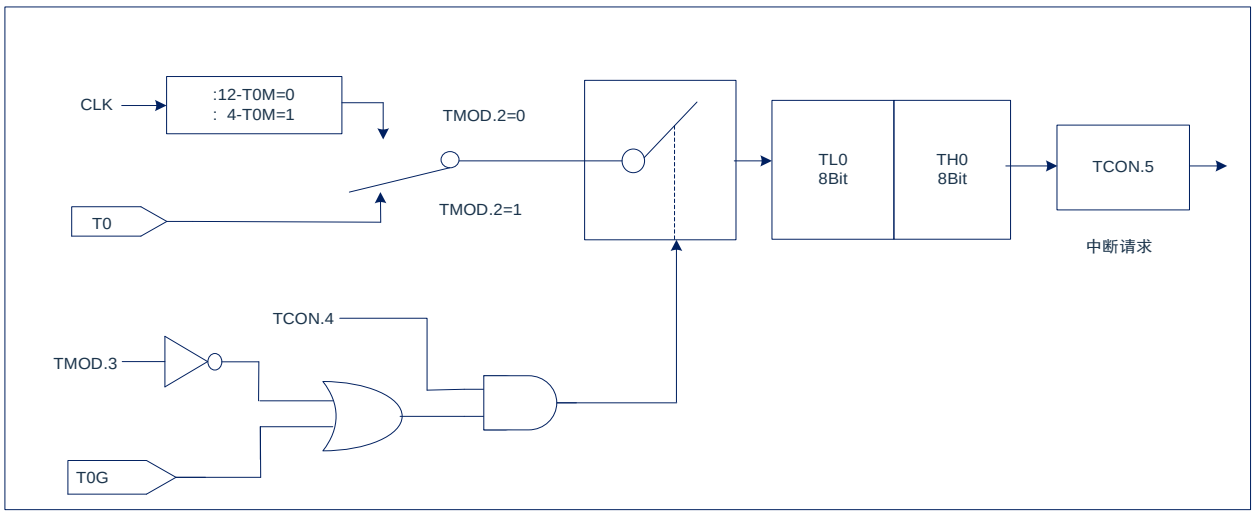
9.4.1 T0 -模式 0（13 位定时/计数模式）

在这种模式下，定时器 0 是 13 位的寄存器。当计数器所有的位由 1 向 0 翻转后，定时器 0 中断标志 TF0 置 1。当 TCON.4=1 且 TMOD.3=0 或 TCON.4=1，TMOD.3=1，T0G=1 时，计数输入使能到定时器 0。（设置 TMOD.3=1 允许定时器 0 由外部引脚 T0G 控制，以便脉宽测量）。13 位的寄存器由 TH0 和 TL0 低 5 位组成。TL0 高 3 位应忽略。Timer0 模式 0 结构框图如下图所示：



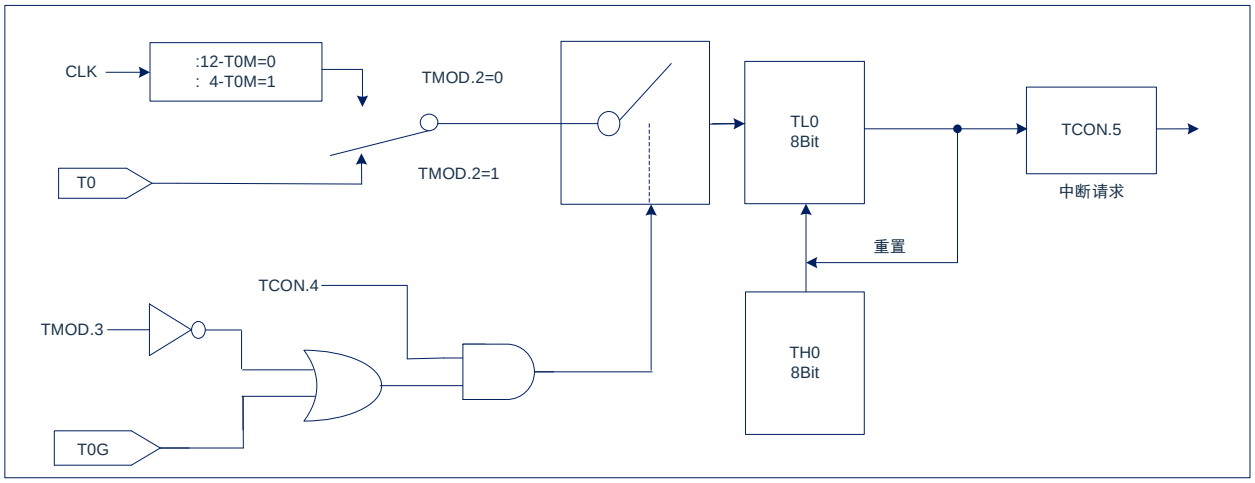
9.4.2 T0 -模式 1（16 位定时/计数模式）

模式 1 与模式 0 相同，只是模式 1 下定时器 0 数据寄存器 16 位全部运行。Timer0 模式 1 结构框图如下图所示：



9.4.3 T0 -模式 2（8 位自动重装定时/计数模式）

模式 2 下定时器寄存器是一个配有自动重装模式的 8 位计数器（TL0），如下图所示。来自 TL0 的溢出不仅使 TF0 置 1，而且还将 TH0 的内容由软件重装到 TL0。重装过程中 TH0 的值保持不变。Timer0 模式 2 结构框图如下图所示：



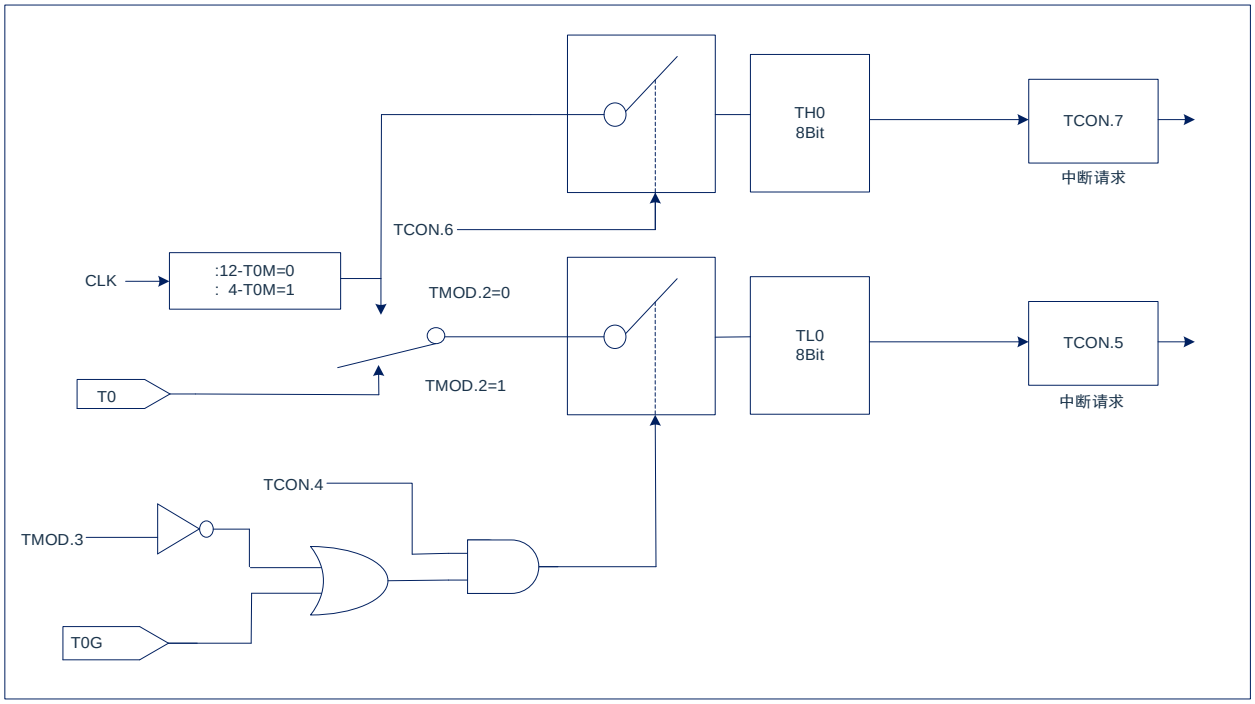
9.4.4 T0 -模式 3（两个单独的 8 位定时/计数器）

模式 3 下的定时器 0 将 TL0 和 TH0 设置为两个独立的计数器。定时器 0 模式 3 的逻辑如下图所示。

TL0 可工作为定时器或计数器，并使用定时器 0 的控制位：如 CT0、TR0、GATE0 和 TF0。

TH0 只能工作为定时器，且使用定时器 1 的 TR1 和 TF1 标志并控制定时器 1 的中断。

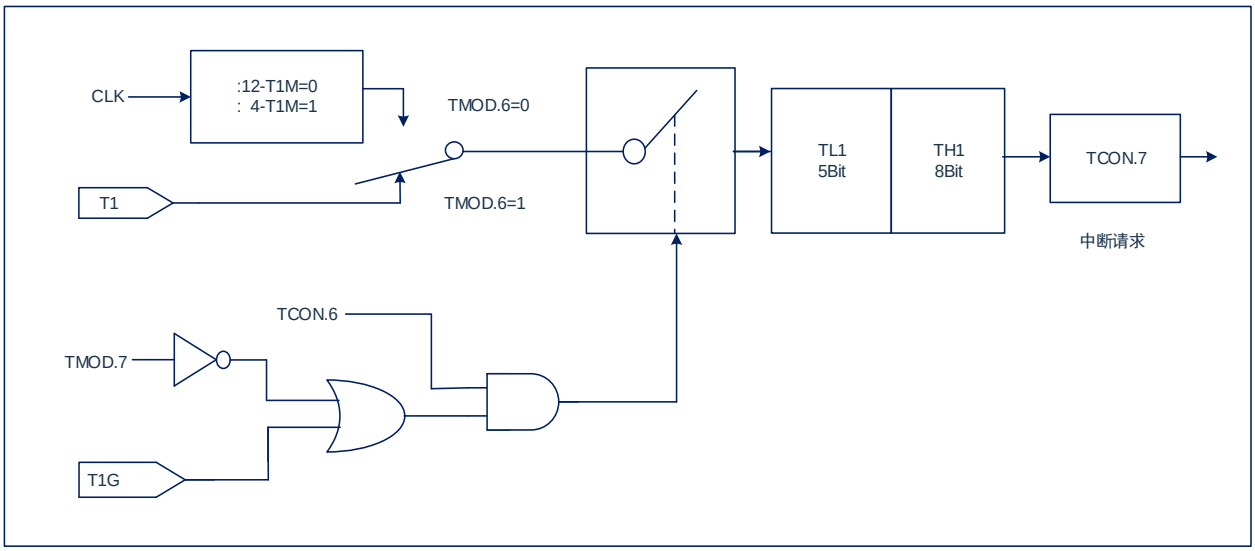
需要使用两个 8 位定时器/计数器时可以使用模式 3。当定时器 0 处于模式 3 时，定时器 1 可以通过切换到自己的模式 3 而将其关闭，或者仍然可以被串行通道用作波特率发生器，或在不需要定时器 1 中断的任何应用中。Timer0 模式 3 结构框图如下图所示：



9.5 Timer1 工作模式

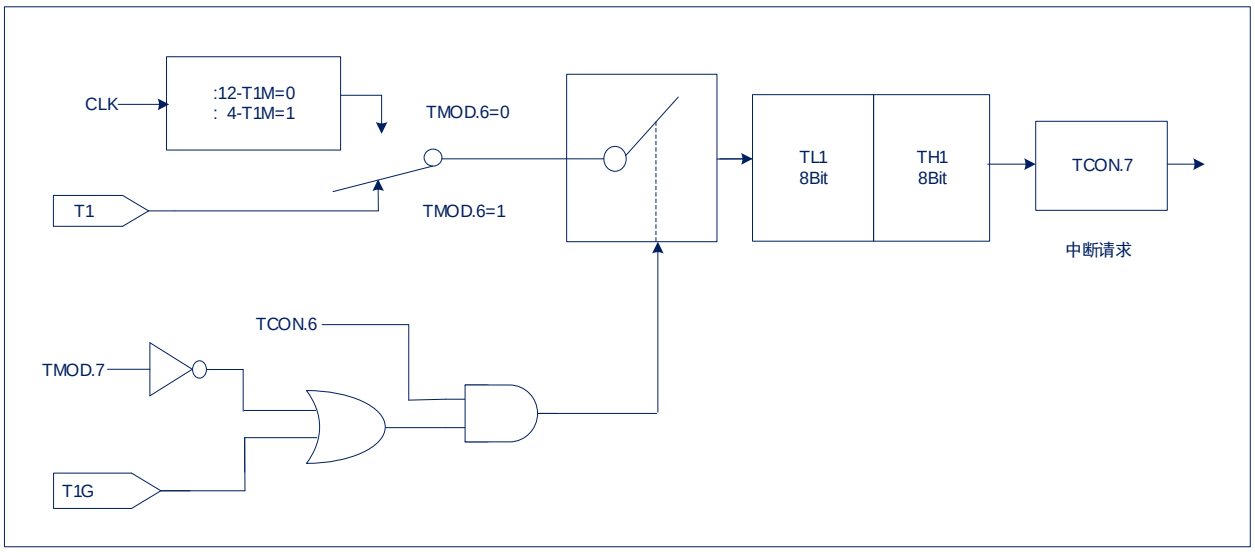
9.5.1 T1 -模式 0（13 位定时/计数模式）

在这种模式下，定时器 1 是 13 位的寄存器。当计数器所有的位由 1 向 0 翻转后，定时器 1 中断标志 TF1 置 1。当 TCON.6=1 且 TMOD.7=0 或当 TCON.6=1，TMOD.7=1 且 T1G=1 时，计数输入使能到定时器 1。（设置 TMOD.7=1 允许定时器 1 由外部引脚 T1G 控制，以便脉宽测量）。13 位的寄存器由 TH1 8 位和 TL1 低 5 位组成。TL1 高三位应忽略。Timer1 模式 0 结构框图如下图所示：



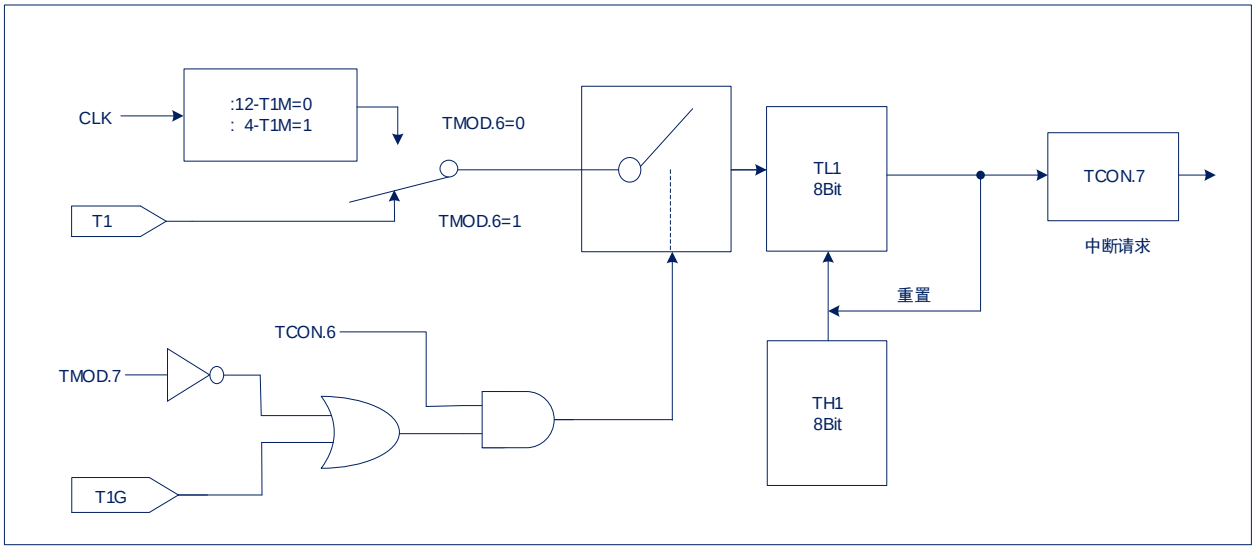
9.5.2 T1 -模式 1（16 位定时/计数模式）

模式 1 与模式 0 相同，只是模式 1 下定时器 1 寄存器 16 位全部运行。Timer1 模式 1 结构框图如下图所示：



9.5.3 T1 -模式 2（8 位自动重装定时/计数模式）

模式 2 下定时器 1 寄存器是一个配有自动重装模式的 8 位计数器（TL1），如下图所示。来自 TL1 的溢出不仅使 TF1 置 1，而且还将 TH1 的内容由软件重装到 TL1。重装过程中 TH1 的值保持不变。Timer1 模式 2 结构框图如下图所示：



9.5.4 T1 -模式 3（停止计数）

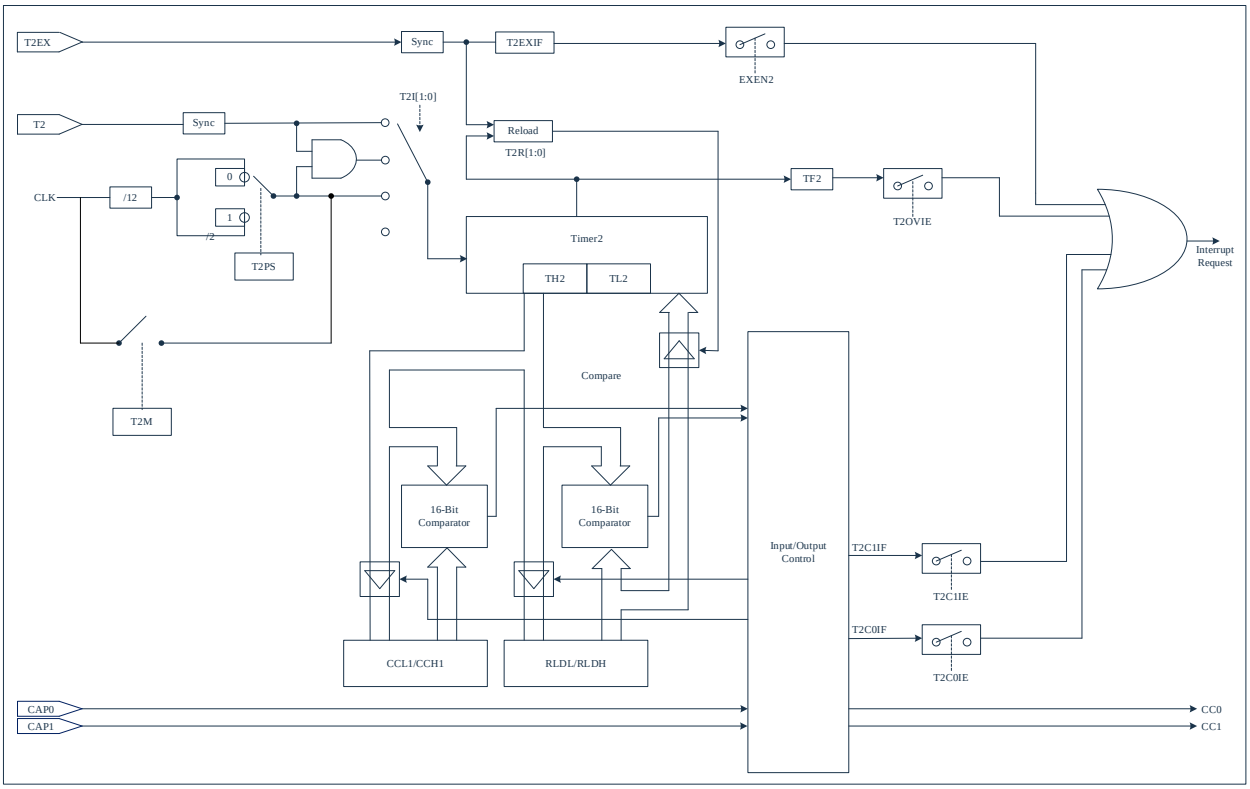
模式 3 中的定时器 1 停止计数，其效果与设置 TR1=0 相同。

10. 定时计数器 2（Timer2）

具有额外的比较/捕获/重装功能的定时器 2 是最核心的外设单元之一。它可以用于各种数字信号的生成和事件捕获，比如脉冲发生、脉冲宽度调制、脉宽测量等。

10.1 概述

具有附加比较/捕获/重装寄存器功能的定时器 2 的结构框图如下图所示：



10.2 相关寄存器

10.2.1 Timer2 控制寄存器 T2CON

0xC8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	T2PS	I3FR	CAPES	T2R1	T2R0	T2CM	T2I1	T2I0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 T2PS: Timer2 时钟预分频选择位（T2M=0时生效，详见CKCON描述）；
1= Fsys/24；
0= Fsys/12。

Bit6 I3FR: 捕获通道0输入单边沿选择与比较中断时刻选择位；
捕获通道0模式：
1= 上升沿捕获到RLDL/RLDH寄存器；
0= 下降沿捕获到RLDL/RLDH寄存器。
比较通道0模式：
1= TL2/TH2与RLDL/RLDH从不相等到相等时刻产生中断；
0= TL2/TH2与RLDL/RLDH从相等到不相等时刻产生中断；

Bit5 CAPES: 捕获通道1输入单边沿选择。
0= 上升沿捕获到CCL1/CCH1寄存器；
1= 下降沿捕获到CCL1/CCH1寄存器。

Bit4~Bit3 T2R<1:0>: Timer2 加载模式选择位；
0x= 重装禁止；
10= 加载模式1：Timer2溢出时自动重装；
11= 加载模式2：在T2EX引脚下降沿重装。

Bit2 T2CM: 比较模式选择；
1= 比较模式1；
0= 比较模式0。

Bit1~Bit0 T2I<1:0>: Timer2时钟输入选择位；
00= Timer2停止；
01= 系统时钟的分频（由T2PS控制分频选择）；
10= 外部引脚T2作事件输入（事件计数模式）；
11= 外部引脚T2作门控输入（门控定时模式）。

10.2.2 Timer2 数据寄存器低位 TL2

0xCC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL2	TL27	TL26	TL25	TL24	TL23	TL22	TL21	TL20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TL2<7:0>: 定时器2低位数据寄存器（同时作为计数器低位）。

10.2.3 Timer2 数据寄存器高位 TH2

0xCD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH2	TH27	TH26	TH25	TH24	TH23	TH22	TH21	TH20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　TH2<7:0>:　定时器2高位数据寄存器（同时作为计数器高位）。

10.2.4 Timer2 比较/捕获/自动重装寄存器低位 RLDL

0xCA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RLDL	RLDL7	RLDL6	RLDL5	RLDL4	RLDL3	RLDL2	RLDL1	RLDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　RLDL<7:0>:　定时器2比较/捕获/自动重装寄存器低位。

10.2.5 Timer2 比较/捕获/自动重装寄存器高位 RLDH

0xCB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RLDH	RLDH7	RLDH6	RLDH5	RLDH4	RLDH3	RLDH2	RLDH1	RLDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　RLDH<7:0>:　定时器2比较/捕获/自动重装寄存器高位。

10.2.6 Timer2 比较/捕获通道 1 寄存器低位 CCL1

0xC2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCL1	CCL17	CCL16	CCL15	CCL14	CCL13	CCL12	CCL11	CCL10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　CCL1<7:0>:　定时器2比较/捕获通道1寄存器低位。

10.2.7 Timer2 比较/捕获通道 1 寄存器高位 CCH1

0xC3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCH1	CCH17	CCH16	CCH15	CCH14	CCH13	CCH12	CCH11	CCH10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　CCH1<7:0>:　定时器2比较/捕获通道1寄存器高位。

10.2.8 Timer2 比较捕获控制寄存器 CCEN

0xCE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCEN	--	--	--	--	CMH1	CML1	CMH0	CML0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit4 保留 须均为0。

Bit3~Bit2 CMH1-CML1: 捕获/比较模式控制位；

00= 捕获/比较禁止；

01= 捕获操作在通道1的上升沿或下降沿触发(CAPES选择)；

10= 比较模式使能；

11= 捕获操作在写CCL1时触发或通道1的双沿触发。

Bit1~Bit0 CMH0-CML0: 捕获/比较模式控制位；

00= 捕获/比较禁止；

01= 捕获操作在通道0的上升沿或下降沿触发(I3FR选择)；

10= 比较模式使能；

11= 捕获操作在写RLDL时触发或通道0的双沿触发。

10.3 Timer2 中断

定时器 2 可以通过寄存器 IE 使能或关闭总中断，还可以通过 IP 寄存器设置高/低优先级。Timer2 具有 4 种中断类型：

- 定时溢出中断。
- 外部引脚 T2EX 下降沿中断。
- 比较中断。
- 捕获中断。

设置 Timer2 中断需配置全局中断允许位（EA=1）、Timer2 总中断允许位（ET2=1）、及 Timer2 相应中断类型使能位（T2IE）。Timer2 的 4 种类型中断均共用一个中断向量，进入中断服务程序后，需要判断相关标志位才能确定是哪一种类型产生了中断。

10.3.1 中断相关寄存器

10.3.1.1 中断屏蔽寄存器 IE

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 EA: 全局中断允许位；
1= 允许所有未被屏蔽的中断；
0= 禁止所有中断。
Bit6 ES1: UART1中断允许位；
1= 允许UART1中断；
0= 禁止UART1中断。
Bit5 ET2: TIMER2总中断允许位；
1= 允许TIMER2所有中断；
0= 禁止TIMER2所有中断。
Bit4 ES0: UART0中断允许位；
1= 允许UART0中断；
0= 禁止UART0中断。
Bit3 ET1: TIMER1中断允许位；
1= 允许TIMER1中断；
0= 禁止TIMER1中断。
Bit2 EX1: 外部中断1中断允许位；
1= 允许外部中断1中断；
0= 禁止外部中断1中断。
Bit1 ET0: TIMER0中断允许位；
1= 允许TIMER0中断；
0= 禁止TIMER0中断。
Bit0 EX0: 外部中断0中断允许位；
1= 允许外部中断0中断；
0= 禁止外部中断0中断。

10.3.1.2 Timer2 中断屏蔽寄存器 T2IE

0xCF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IE	T2OVIE	T2EXIE	--	--	--	--	T2C1IE	T2C0IE
R/W	R/W	R/W	--	--	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 T2OVIE: Timer2溢出中断允许位；
1= 允许中断；
0= 禁止中断。
Bit6 T2EXIE: Timer2外部加载中断允许位；
1= 允许中断；
0= 禁止中断。
Bit5~Bit2 -- 保留，须为0。
Bit1 T2C1IE: Timer2比较通道1中断允许位；
1= 允许中断；
0= 禁止中断。
Bit0 T2C0IE: Timer2比较通道0中断允许位；
1= 允许中断；
0= 禁止中断。

若开启 Timer2 的中断，还需要开启 Timer2 的总中断使能位 ET2=1（IE.5=1）。

10.3.1.3 中断优先级控制寄存器 IP

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 -- 保留，须为0。
Bit6 PS1: UART1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit5 PT2: TIMER2中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit4 PS0: UART0 中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit3 PT1: TIMER1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit2 PX1: 外部中断1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit1 PT0: TIMER0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit0 PX0: 外部中断0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

10.3.1.4 Timer2 中断标志位寄存器 T2IF

0xC9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IF	TF2	T2EXIF	--	--	--	--	T2C1IF	T2C0IF
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 TF2: Timer2 计数器溢出中断标志位；
1= Timer2 计数器溢出，需软件清零；
0= Timer2 计数器无溢出。
Bit6 T2EXIF: Timer2 外部加载标志位；
1= Timer2 的T2EX口产生下降沿，需软件清零；
0= --
Bit5~Bit2 -- 保留，须为0。
Bit1 T2C1IF: Timer2 比较/捕获通道1标志位；
1= Timer2 比较通道1 {CCH1:CCL1}={TH2:TL2}或捕获通道1产生了捕获操作，需软件清零。
0= --
Bit0 T2C0IF: Timer2 比较/捕获通道0标志位；
1= Timer2 比较通道0{RLDH:RLDL}={TH2:TL2}或捕获通道0产生了捕获操作，需软件清零。
0= --

10.3.2 定时中断

定时中断使能位由寄存器 T2IE[7]设置，中断标志位通过寄存器 T2IF[7]查看。当 Timer2 定时器溢出时，定时器溢出中断标志位 TF2 将会置 1。

10.3.3 外部触发中断

外部引脚 T2EX 下降沿触发中断使能位由寄存器 T2IE[6]设置，中断标志位通过寄存器 T2IF[6]查看。当 T2EX 引脚下降沿时，外部加载中断标志位 T2EXIF 将会置 1。

10.3.4 比较中断

2 个比较通道均支持比较中断。比较中断使能位由寄存器 T2IE[1:0]设置，中断标志位通过寄存器 T2IF[1:0]查看。

比较通道 0 可选择比较中断产生的时刻，产生中断则会将比较通道 0 的中断标志 T2C0IF 置 1。

I3FR = 0 时，TL2/TH2 与 RLDL/RLDH 从不相等到相等时刻产生中断；

I3FR = 1 时，TL2/TH2 与 RLDL/RLDH 从相等到不相等时刻产生中断；

比较通道 1 不能选择中断产生时刻，固定为 TL2/TH2 与 CCxL/CCxH 从不相等到相等时刻产生中断。如果产生中断则会将相应的比较通道中断标志 T2CxIF 置 1。

10.3.5 捕获中断

2 个捕获通道均支持外部捕获中断。捕获中断使能位由寄存器 T2IE[1:0]设置，中断标志位通过寄存器 T2IF[1:0]查看。产生捕获操作时会将相应捕获通道的中断标志 T2CxIF 置 1。

注意写操作捕获方式不会产生中断。

10.4 Timer2 功能描述

定时器 2 是一个时钟源来自系统时钟、16 位的向上计数定时器。Timer2 可以配置如下功能模式：

- 定时模式。
- 重装模式。
- 门控定时模式。
- 事件计数模式。
- 比较模式。
- 捕获模式。

设置定时器 2 的不同模式，可以用于各种数字信号的生成和事件捕获，比如脉冲发生、脉冲宽度调制、脉宽测量等。

10.4.1 定时模式

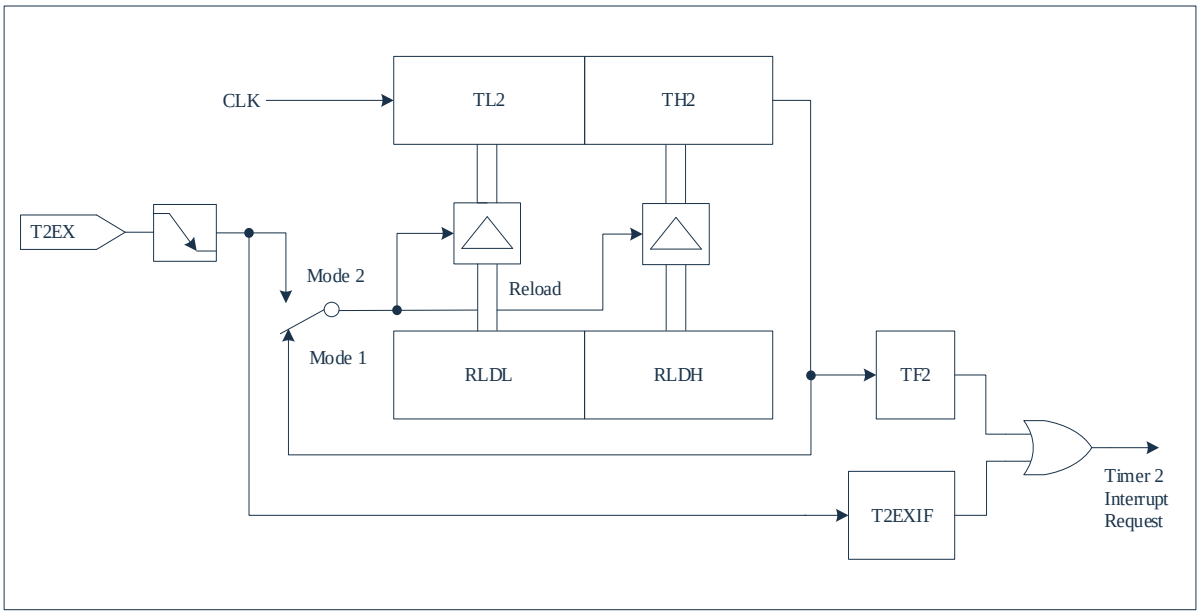
用作定时器功能时，时钟源来自系统时钟。预分频器提供系统时钟或 1/12 或 1/24 的系统频率选择，预分频器的值由寄存器 T2CON 的 T2PS 位和 CKCON 的 T2M 位选择。因此，16 位定时器寄存器（由 TH2 和 TL2 组成）每个时钟周期或每 12 个时钟周期或每 24 个时钟周期递增。

10.4.2 重装模式

定时器 2 的重装模式由寄存器 T2CON 的 T2R0 和 T2R1 位选择，重装结构框图如下图所示。

在加载模式 1 中：当 Timer2 计数器由全 1 向 0 翻转后（计数器溢出）时，不仅溢出中断标志位 TF2 置 1，而且 Timer2 寄存器自动加载来自 RLDL/RLDH 寄存器的 16 位值，从而覆盖计数值 0x0000，所需的 RLDL/RLDH 值可由软件预设。

在加载模式 2 中：来自 RLDL/RLDH 寄存器的 16 位重装操作由相应 T2EX 输入引脚的下降沿触发。当检测到 T2EX 的下降沿时，外部加载中断标志位 T2EXIF 置 1，同时 Timer2 自动加载 RLDL/RLDH 寄存器的 16 位值作为计数初始值。



10.4.3 门控定时模式

Timer2 用作门控定时器功能时，外部输入引脚 T2 作为定时器 2 的门控输入。如果 T2 引脚为高电平，则内部时钟输入被选通到定时器。T2 引脚为低电平则会终止计数。该功能常被用来测量脉冲宽度。

10.4.4 事件计数模式

Timer2 用作事件计数功能时，定时计数器在外部输入引脚 T2 的下降沿计数值加 1。在每个系统时钟周期对外部输入信号进行采样，当采样输入在一个周期内显示高电平，在下一个周期内显示为低电平时，计数增加。在接下来的周期再检测到 T2 引脚高到低的变化，则新的计数值更新到定时器数据寄存器中。

10.4.5 比较模式

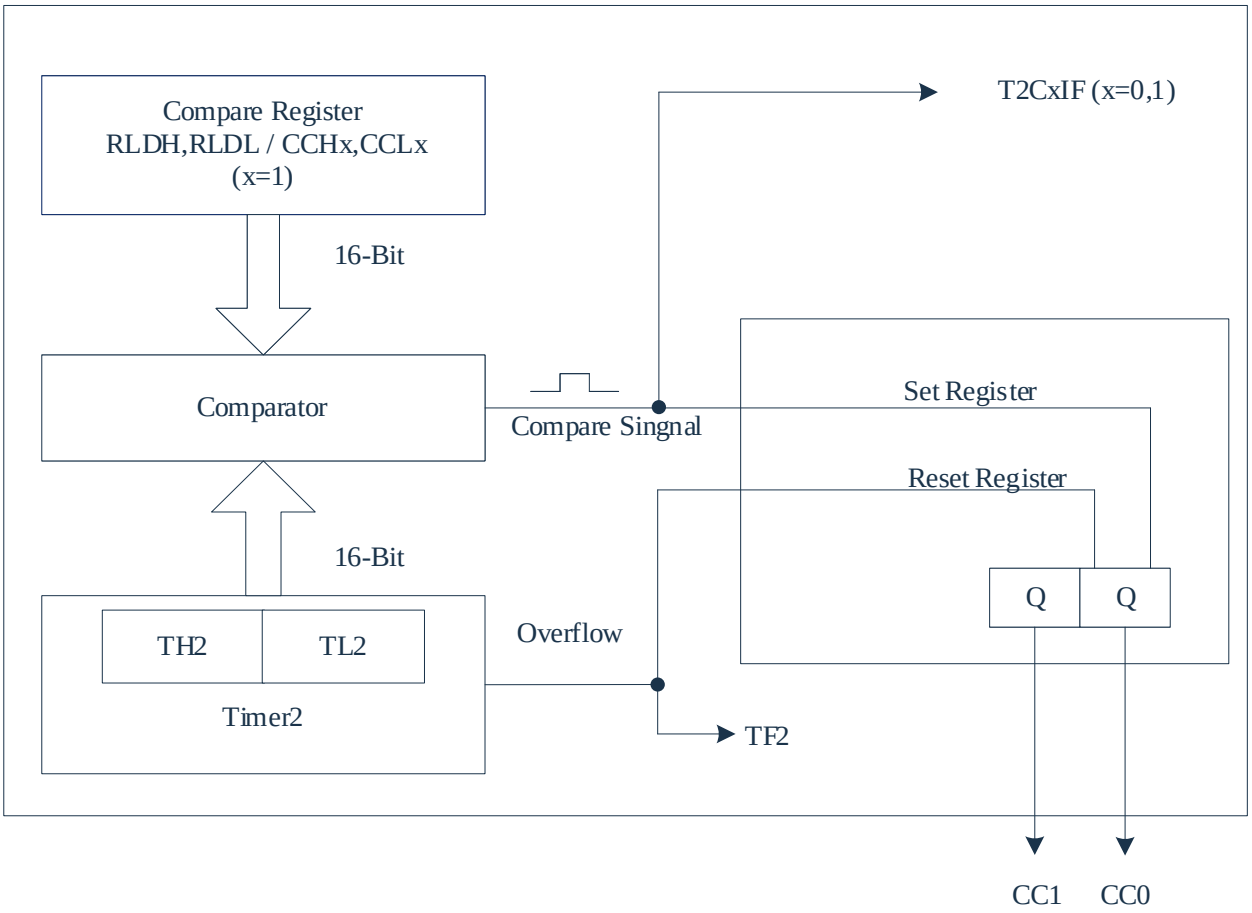
比较功能包含两种模式：比较模式 0 和比较模式 1，由特殊功能寄存器 T2CON 中的 T2CM 位选择。这两种比较模式可产生周期性信号且可改变占空比控制方式，常被用于脉冲宽度调制（PWM）以及需要产生连续方波的控制场合，涵盖了较为广泛的应用。

比较功能的输出通道为 CC0、CC1。分别对应 16 位比较寄存器{RLDH，RLDL}、{CCH1，CCL1}与数据寄存器{TH2，TL2}的比较结果输出信号。

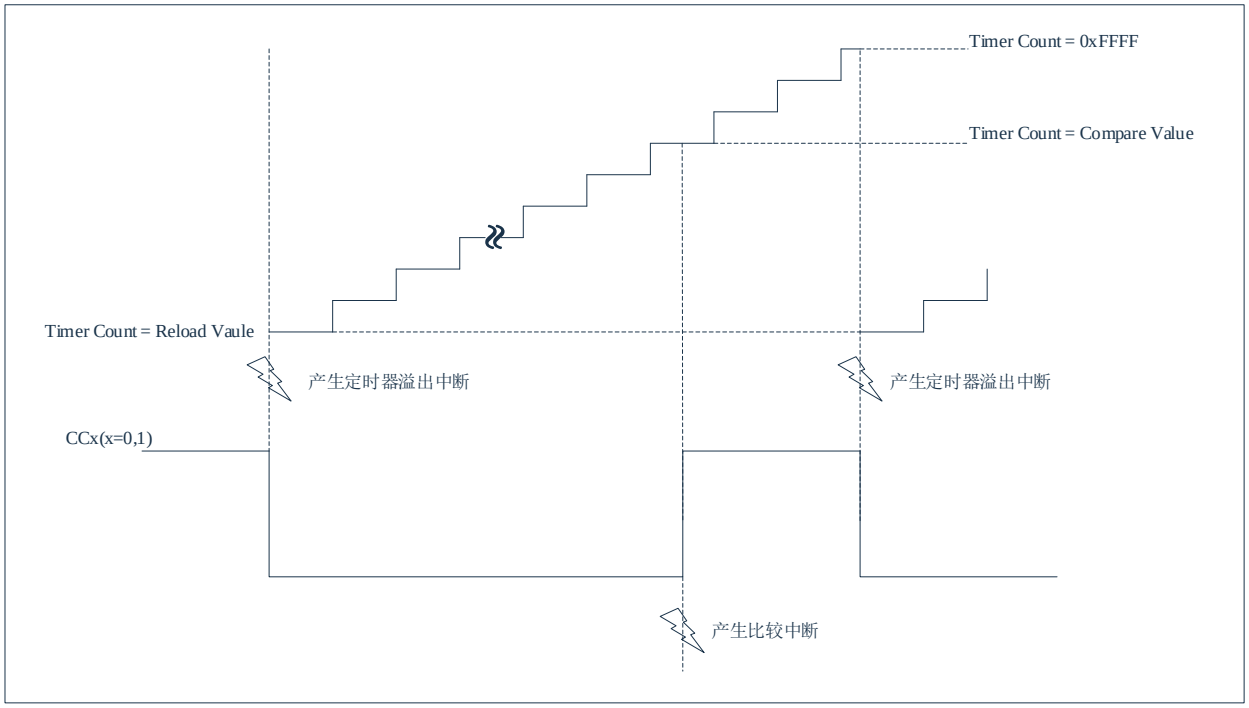
存储在比较寄存器中的 16 位存储值与定时器的计数值进行比较，如果数据寄存器中的计数值与存储值相匹配，则在相应的端口引脚上会产生输出信号的跳变，并产生中断标志位。

10.4.5.1 比较模式 0

在模式 0 中，当定时器的计数值和比较寄存器相等后，比较输出信号从低电平变为高电平。当定时器计数值溢出时，比较输出信号变为低电平。比较输出通道由两个事件直接控制：定时器溢出和比较操作。比较模式 0 的结构框图如下图所示：



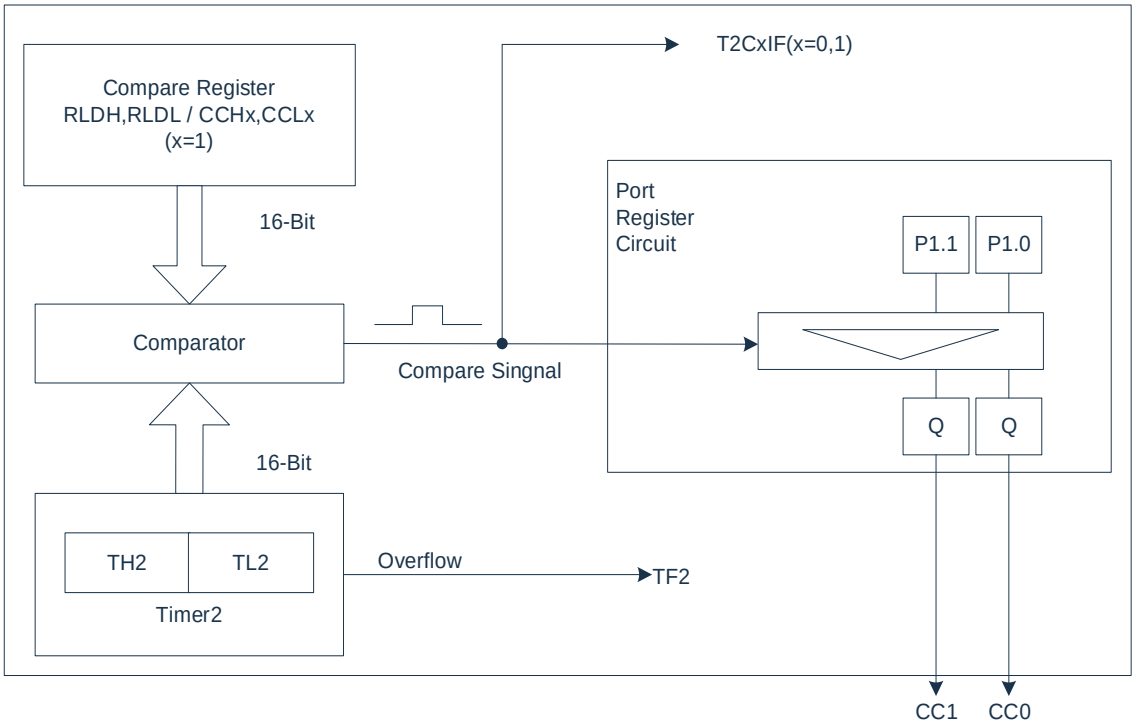
比较模式 0 的输出结构框图如下图所示：



10.4.5.2 比较模式 1

在比较模式 1 中，通常使用在输出信号与恒定信号周期无关、软件自适应地确定输出信号跳变的场合。

如果模式 1 使能，软件将写入到 CCx 端口的相应的输出寄存器，在下一个比较匹配发生之前，新的值不会出现在输出引脚上。当定时器 2 计数器与存储的比较值相匹配时，用户可以选择输出信号是进行新的改变还是保持其旧值这两种方式其一。比较模式 1 的结构框图如下图所示：



10.4.6 捕获模式

2 个作为捕获功能的 16 位寄存器{RLDH，RLDL}、{CCH1，CCL1}中的每一组寄存器都可用于锁存{TH2，TL2}的当前 16 位值。该功能提供了两种不同的捕获模式。

在模式 0 中，一个外部事件可将定时器 2 的内容锁存到捕获寄存器中。

在模式 1 中，捕获操作发生在写入 16 位捕获寄存器的低位字节（RLDL/CCL1）时。这种模式允许软件在运行时读取{TH2，TL2}的内容。

捕获通道 0、1 分别选择捕获输入引脚 CAP0(P14)、CAP1(P15)作为输入源信号。P14、P15 作为 Timer2 捕获通道使用时，需配置为数字输入，且配置 CCEN 寄存器的对应的捕获方式。

10.4.6.1 捕获模式 0

在捕获模式 0 中，捕获通道 0~1（CAP0~CAP1）上的正跳变、负跳变或者正负跳变都将产生捕获事件。捕获事件发生时，定时器的计数值锁存在对应捕获寄存器中。

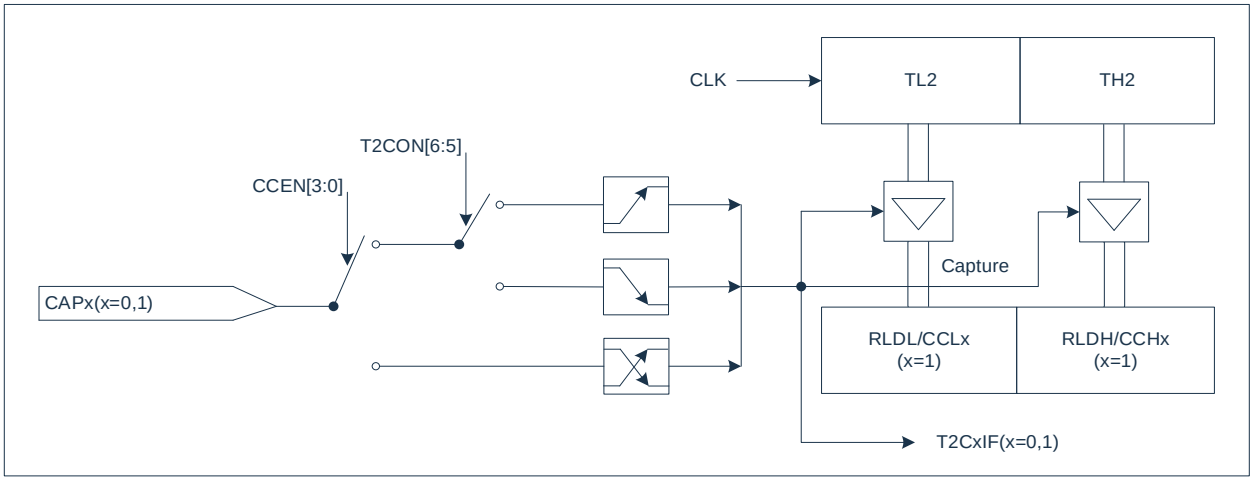
捕获通道 0 上的是正跳变触发捕获操作还是负跳变触发捕获操作取决于 T2CON 的 I3FR 位。I3FR=0，负跳变触发捕获；I3FR=1，正跳变触发捕获。

捕获通道 1 上的是正跳变触发捕获操作还是负跳变触发捕获操作取决于 T2CON 的 CAPES 位。CAPES=0，正跳变触发捕获；CAPES=1，负跳变触发捕获。

捕获通道 0~1 同时支持双跳变的捕获操作。将 CCEN 寄存器相应的工作模式控制位选择为 11，则该通道支持双跳变的捕获操作。需要注意的是，该工作模式下也支持捕获模式 1，即写操作可产生捕获动作。

捕获模式 0 下，捕获通道 0~1 的外部捕获事件均可产生中断。

捕获模式 0 的结构框图如下图所示：

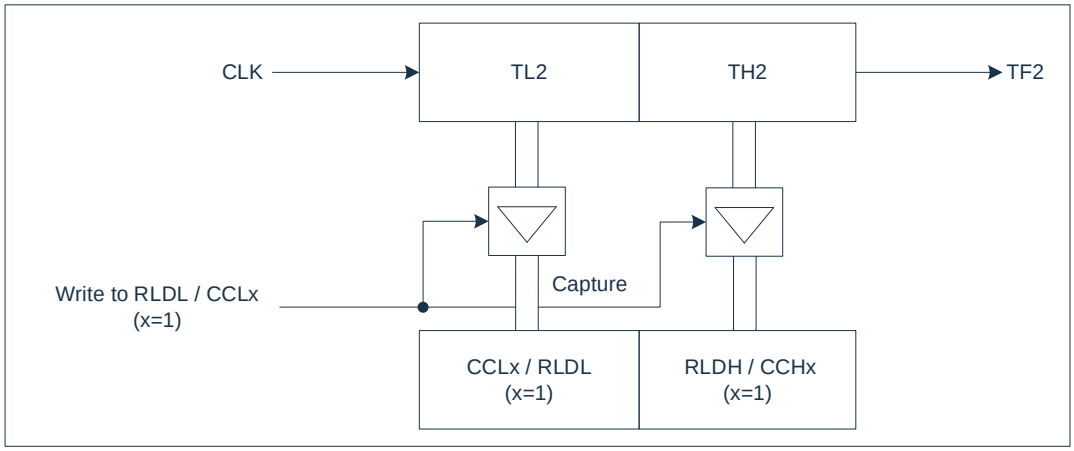


10.4.6.2 捕获模式 1

在捕获模式 1 中，捕获操作事件为写捕获寄存器的低位字节指令的执行。写寄存器信号（例如写 RLDL）启动捕获操作，且写入的值与此功能无关。在写指令执行后，定时器 2 的内容将被锁存到相应的捕获寄存器中。

捕获模式 1 下，捕获通道 0~1 的捕获事件不会产生中断请求标志。

捕获模式 1 的结构框图如下图所示：



11. 定时器 5（Timer5）

Timer5 是一个 16 位定时器，时钟源可选择来自系统时钟、内部低速时钟 LSI。

选择 LSI 作为计数时钟源，则可用于休眠模式下定时唤醒系统。在系统进入休眠前先配置定时唤醒时间，并使能定时唤醒功能。当芯片进入休眠模式后，计数值等于设置值时，芯片进入唤醒等待状态。

11.1 相关寄存器

11.1.1 T5CON 寄存器

D5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T5CON	T5E	T5WUE	T5CKS1	T5CKS0	--	T5PSC2	T5PSC1	T5PSC0
读写	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 T5E 定时器使能位；
1= 启动；
0= 关闭。
Bit6 T5WUE 定时唤醒功能使能位；
1= 定时唤醒功能打开；
0= 定时唤醒功能关闭。
Bit5~Bit4 T5CKS<1:0> 时钟源选择位
（时钟切换需要先关闭定时器使能位，再进行时钟选择，最后启动定时器使能）
00= 系统时钟；
01= 系统时钟；
10= LSI；
11= 禁止配置
Bit3 -- 保留。
Bit2~Bit0 T5PSC<2:0>: 计数器的时钟分频系数；
000= 1；
001= 2；
010= 4；
011= 8；
100= 16；
101= 32；
110= 64；
111= 128；

11.1.2 TL5 数据寄存器低位

D3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL5	TL57	TL56	TL55	TL54	TL53	TL52	TL51	TL50
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TL5<7:0>: 定时器5数据寄存器低8位。

11.1.3 TH5 数据寄存器高位

D4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH5	TH57	TH56	TH55	TH54	TH53	TH52	TH51	TH50
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　TH5<7:0>:　定时器5数据寄存器高8位。

11.2 休眠与定时唤醒

当定时器 5 的计数值等于定时值{TL5/T5H}时，定时器的中断标志位 TF5 置 1。若全局中断使能（EA=1），并且定时器中断使能（ET5=1）,CPU 将执行中断服务程序。

采用 Timer5 定时唤醒休眠模式，需在休眠之前打开 T5E、T5WUE，并且设置后休眠状态到唤醒的时间{ T5H[7:0]，T5L[7:0]}。

如果休眠之前打开全局中断使能和 Timer5 中断使能，休眠唤醒后，将首先执行中断服务程序，中断返回之后再执行休眠指令的下一条指令。如果休眠之前未开启相应中断。休眠唤醒后，则执行休眠指令的下一条指令。

11.2.1 中断屏蔽寄存器 EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	--	ADCIE	PWMIE	ET5	--	--
R/W	R/W	R/W	R	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI中断使能位；
1= 允许SPI中断；
0= 禁止SPI中断。
Bit6 I2CIE: I^2C中断使能位；
1= 允许I^2C中断；
0= 禁止I^2C中断。
Bit5 -- 保留，须为0。
Bit4 ADCIE: ADC中断使能位；
1= 允许ADC中断；
0= 禁止ADC中断。
Bit3 PWMIE: PWM总中断使能位；
1= 允许PWM所有中断；
0= 禁止PWM所有中断。
Bit2 ET5: Timer5中断使能位；
1= 允许Timer5中断；
0= 禁止Timer5中断。
Bit1~Bit0 -- 保留，须为0。

11.2.2 外设中断标志位寄存器 EIF2

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	TF5	--	--
R/W	R	R	R	R/W	R	R/W	RW	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI总中断指示位，只读；
1= SPI产生中断，（清除具体的中断标志位后，此位自动清除）；
0= SPI未产生中断。
Bit6 I2CIF: I^2C总中断指示位，只读；
1= I^2C产生中断，（清除具体的中断标志位后，此位自动清除）；
0= I^2C未产生中断。
Bit5 -- 保留，须为0。
Bit4 ADCIF: ADC中断标志位；
1= ADC转换完成，需软件清零；
0= ADC转换未完成。
Bit3 PWMIF: PWM总中断指示位，只读；
1= PWM产生中断，（清除具体的中断标志位后，此位自动清除）；
0= PWM未产生中断。
Bit2 TF5: Timer5定时器溢出中断标志位；
1= Timer5定时器溢出，需软件清零；
0= Timer5定时器无溢出。
Bit1~Bit0 -- 保留，须为0。

11.2.3 中断优先级控制寄存器 EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	--	PADC	PPWM	PT5	--	--
R/W	R/W	R/W	R	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit6 PI2C: I^2C中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit5 -- 保留，须为0。
Bit4 PADC: ADC中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit3 PPWM: PWM中断优先级控制位
1= 设置为高级中断；
0= 设置为低级中断。
Bit2 PT5: TIMER5中断优先级控制位
1= 设置为高级中断；
0= 设置为低级中断。
Bit1~Bit0 -- 保留，须为0。

11.3 功能描述

Timer5 内部有一个 16 位向上的计数器，启动计数器后，当计数器的值等于{T5H,T5L}后产生中断标志位，同时计数器清零后重新计数。

唤醒定时器原理为：在系统进入休眠模式后，CPU 与外围所有电路停止工作，而低功耗振荡器 LSI 保持工作，为 T5 计数器提供时钟。当计数器产生中断标志位后将唤醒系统，MCU 进入唤醒等待状态。

注 1：定时器 5 可选择系统时钟、LSI 作为计数时钟源。若使用过程中需要切换时钟源，需要先关闭定时器的使能位（T5E=0），再设置需要切换的时钟源（修改 T5CKS），最后再开启定时器使能位（T5E=0）。

注 2：定时器 5 选择 LSI 作为计数时钟源时，无论是否开启定时器唤醒功能，休眠下 LSI 都将保持工作。即芯片进入休眠时，计数器继续保持计数。

注 3：在休眠模式下，若不需要定时器 5 唤醒系统，注意软件关闭定时器 5 以降低功耗。若需要定时器 5 唤醒系统，则在芯片进去休眠模式之前，需要先配置好定时器唤醒功能(T5WUE=1)。当计数器的值等于{T5L/T5H}后将唤醒系统，MCU 进入唤醒等待状态，同时计数器清零后重新计数。

注 4：若在定时器 5 计数过程中，修改数据寄存器的值{T5L/T5H}。会在计数器计数到原设定的数据寄存器的值之后，再重新加载修改后的数据寄存器的值。

12. 波特率定时器（BRT）

12.1 概述

芯片内部有 1 个 10 位波特率定时器 BRT，主要为 UART 模块提供时钟。

12.2 相关寄存器

12.2.1 BRT 定时器数据加载低 8 位寄存器 BRTDL

F5C1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTDL	BRTDL7	BRTDL6	BRTDL5	BRTDL4	BRTDL3	BRTDL2	BRTDL1	BRTDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　BRTDL<7:0>:　BRT定时器加载值低8位；

12.2.2 BRT 定时器数据加载高 2 位寄存器 BRTDH

F5C2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTDH	BRTEN	--	--	--	--	--	BRTDH1	BRTDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7　　BRTEN:　BRT定时器使能位；
1=　使能；
0=　禁止。
Bit6~Bit2　　--　保留；
Bit1~Bit0　　BRTDH<1:0>:　BRT定时器加载值高2位；

12.3 功能描述

BRT 内部有 1 个 10 位递增计数器，其时钟来源于预分频电路，预分频的时钟由定时器预分频选择位 BRTCKDIV 决定，计数器的初值为 0。

开启定时器使能位 BRTEN=1 时，计数器开始工作。当 10 位计数器的值等于{BRTDH，BRTDL}时，BRT 计数器发生溢出。溢出后自动从 0 开始，重新进行计数。

BRT 计数器的溢出信号专门提供给 UART 模块作为波特率的时钟源，溢出时不会产生中断，也没有相应中断结构。BRT 在调试模式下，其时钟不会停止，如果 UART 模块已经开始发送或接收数据，即使芯片进入暂停状态，UART 也会完成整个发送或接收的过程。

BRT 定时器溢出率：

$$BRTov=\frac{Fsys}{\{BRTDH,BRTDL+1\}}$$

13. PWM 模块

13.1 概述

PWM 模块支持 6 路 PWM 发生器，可以配置 6 路 PWM 输出（PG0-PG5），也可以配置成 3 对同步 PWM 输出，或 3 对分别带有死区编程发生器的互补 PWM 输出，其中 PG0-PG1，PG2-PG3，PG4-PG5 分别为一对。

每一对 PWM 共用一个 16 位周期寄存器，每一路 PWM 具有独立的 16 位的占空比寄存器（比较数据寄存器），用以配置 PWM 输出的周期和调节占空比。每一对 PWM 具有独立的时钟分频控制寄存器，8 种分频时钟可选。

每一对 PWM 可配置成边沿对齐计数模式。每一对 PWM 也可设置单次模式（产生一个 PWM 信号周期）或者自动装载（连续输出 PWM 波形）输出，每一路 PWM 的输出极性可通过输出极性控制器设置。

PWM 支持中断功能。6 路 PWM 发生器总共提供 9 个中断标志，包括 3 个零点中断和 6 个向下比较中断，这些中断共用一个中断向量入口。

13.2 特性

PWM 模块具有如下特性：

- 6 路 16 位 PWM 控制模式。
  - 3 对互补 PWM 对输出：（PG0-PG1）、（PG2-PG3）、（PG4-PG5），可插入可编程死区时间；
  - 3 对同步 PWM 对输出：（PG0-PG1）、（PG2-PG3）、（PG4-PG5），每对 PWM 对引脚同步。
- 支持组控制，PG0，PG2，PG4 输出同步，PG1，PG3，PG5 输出同步。
- 支持边沿对齐模式。
- 支持单次模式或者自动装载模式。
- 每路 PWM 有独立的极性控制。

13.3 端口配置

使用 PWM 模块前需要先将相应端口配置成 PWM 通道，PWM 通道在复用功能分配表上用 PG0~PG5 来标注，分别对应 PWM 通道 0~5。

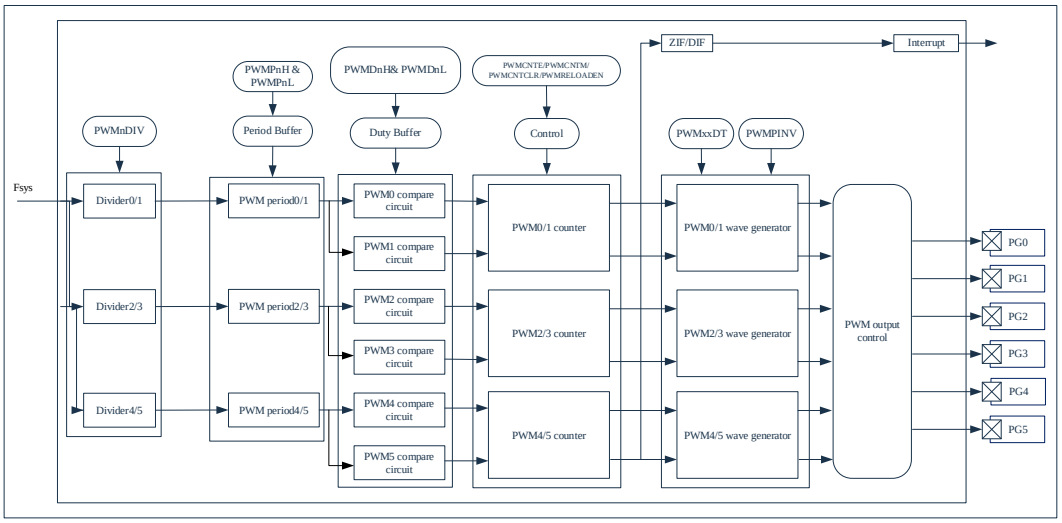
PWM 通道的分配由相应的端口配置寄存器来控制，例如：

```
PS_PG0=0x00;    //P00 配置为 PG0 口
PS_PG1=0x00;    //P01 配置为 PG1 口
PS_PG2=0x00;    //P02 配置为 PG2 口
PS_PG3=0x00;    //P03 配置为 PG3 口
PS_PG4=0x00;    //P04 配置为 PG4 口
PS_PG5=0x00;    //P05 配置为 PG5 口
```

13.4 功能描述

13.4.1 功能框图

PWM 由时钟控制模块、PWM 计数器模块、输出比较单元、波形发生器和输出控制器组成，其结构框图如下图所示：



13.4.2 边沿对齐

边沿对齐模式下，16 位 PWM 计数器 CNTn 在每个周期开始向下计数，并与锁存在 PWMDnH/PWMDnL 寄存器中的值 CMPn 进行比较，当 CNTn=CMPn 时 PGn 输出高电平，PWMnDIF 置 1。CNTn 继续向下计数至 0，此时 PGn 将输出低电平，PWMnZIF 置 1。在 CNTn 计数到零点时，若 PWMnCNTM=1，CMPn 和 PERIODn 将重新加载。

边沿对齐相关参数如下：

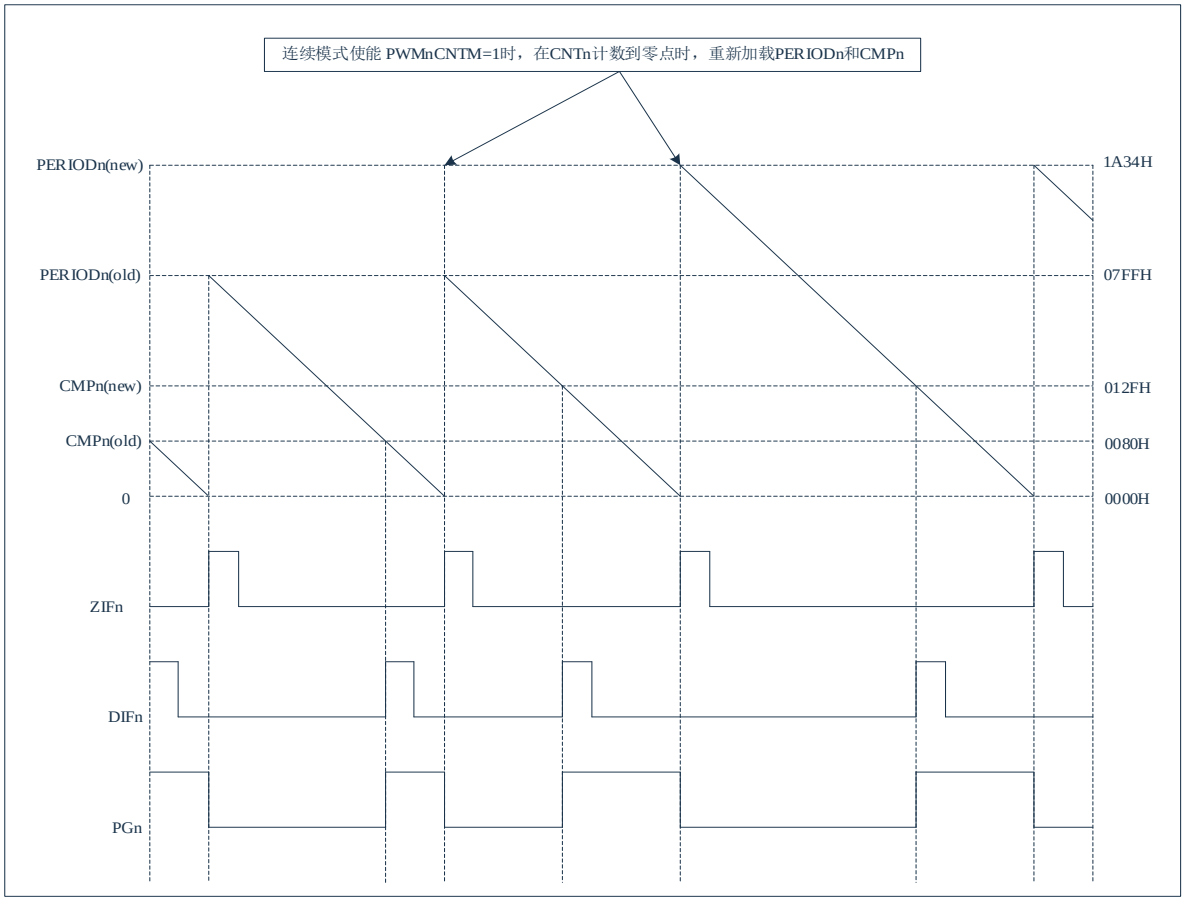
高电平时间=（*CMPn+1*）×*Tpwm*(CMPn≥1)

周期=（*PERIODn+1*）×*Tpwm*

占空比=(CMPn+1) / (PERIODn+1)　(CMPn≥1)

CMPn=0 时，占空比为 0%。

边沿对齐时序如下图所示：



13.4.3 互补模式

6 路 PWM 可设置为 3 组互补 PWM 对。在互补模式下，PG1，PG3，PG5 的周期、占空比分别由 PG0，PG2，PG4 相关寄存器决定，即除了对应的输出使能控制位（PWMnOE）和极性控制，PG1、PG3、PG5 输出波形不再受自己的寄存器控制。

在互补模式下，每组互补 PWM 对均支持插入死区延时，插入的死区时间如下：

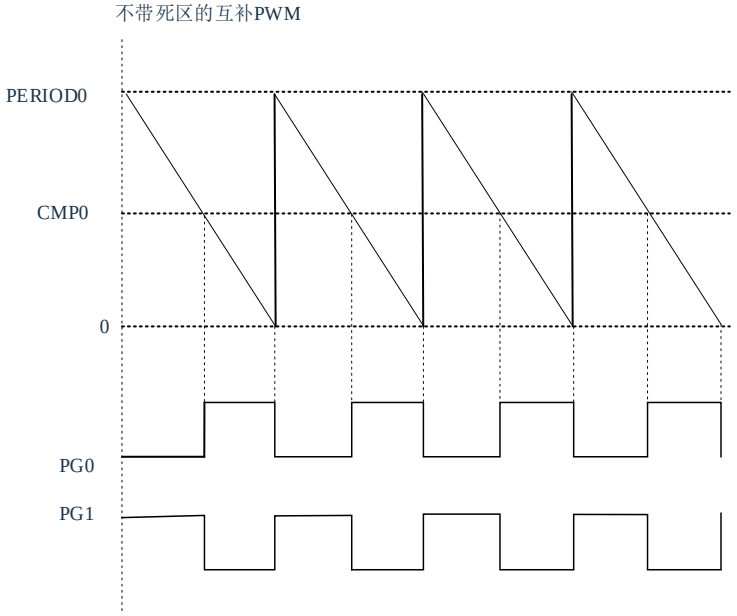
PWM0/1 死区时间：（PWM01DT+1）*T_{PWM0}；

PWM2/3 死区时间：（PWM23DT+1）*T_{PWM2}；

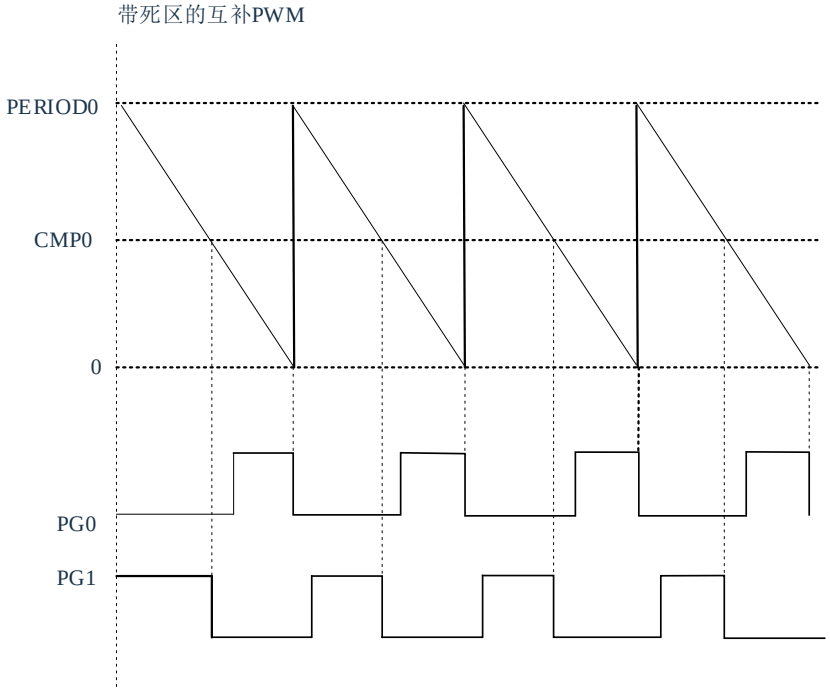
PWM4/5 死区时间：（PWM45DT+1）*T_{PWM4}；

T_{PWM0}/T_{PWM2}/T_{PWM4} 分别为 PG0/PG2/PG4 的时钟源周期。

以 PG0/PG1 为例，互补模式下不带死区的波形图如下图所示：



以 PG0/PG1 为例，互补模式下带死区的波形图如下图所示：



13.4.4 同步模式

6 路 PWM 可设置为 3 组同步 PWM 对。在同步模式下，PG1，PG3，PG5 的周期、占空比分别由 PG0，PG2，PG4 相关寄存器决定，即除了对应的输出使能控制位（PWMnOE）和极性控制，PG1、PG3、PG5 输出波形不再受自己的寄存器控制，PG1 输出波形同 PG0，PG3 输出波形同 PG2，PG5 输出波形同 PG4。

13.5 PWM 相关寄存器

13.5.1 PWM 控制寄存器 PWMCON

F120H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON	--	PWMRUN	PWMMODE1	PWMMODE0	GROUPEN	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 -- 保留，须为0。
Bit6 PWMRUN: PWM时钟预分频，时钟分频使能位；
1= 禁止（PWMmnDIV均被清0）；
0= 使能。
Bit5~Bit4 PWMMODE<1:0>: PWM的模式控制位；
00= 独立模式；
01= 互补模式；
10= 同步模式；
11= 保留。
Bit3 GROUPEN: PWM成组功能使能位；
1= PG0控制PG2，PG4；PG1控制PG3，PG5；
0= 所有PWM通道信号相互独立。
Bit2~Bit0 -- 保留，须为0。

13.5.2 PWM 输出使能控制寄存器 PWMOE

F121H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMOE	--	--	PWM5OE	PWM4OE	PWM3OE	PWM2OE	PWM1OE	PWM0OE
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6 -- 保留，须均为0。
Bit5 PWM5OE: PWM通道5的输出使能位；
1= 使能；
0= 禁止。
Bit4 PWM4OE: PWM通道4的输出使能位；
1= 使能；
0= 禁止。
Bit3 PWM3OE: PWM通道3的输出使能位；
1= 使能；
0= 禁止。
Bit2 PWM2OE: PWM通道2的输出使能位；
1= 使能；
0= 禁止。
Bit1 PWM1OE: PWM通道1的输出使能位；
1= 使能；
0= 禁止。
Bit0 PWM0OE: PWM通道0的输出使能位；
1= 使能；
0= 禁止。

13.5.3 PWM 时钟分频控制寄存器 PWMnDIV(n=0-2)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMnDIV	--	--	--	--	--	PWMnDIV2	PWMnDIV1	PWMnDIV0
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMnDIV(n=0-2)地址：F12AH，F12BH，F12CH。

(n=0,对应 PWM 通道 0,1；n=1,对应 PWM 通道 2,3；n=2,对应 PWM 通道 4,5)

Bit7~Bit3 -- 保留，须均为0。

Bit2~Bit0 PWMnDIV<2:0>: PWM通道n时钟分频控制位;

000= Fsys；
001= Fsys/2；
010= Fsys/4；
011= Fsys/8；
100= Fsys/16；
101= Fsys/32；
110= Fsys/64；
111= Fsys/128；

13.5.4 PWM 数据加载使能控制寄存器 PWMLOADEN

F129H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMLOADEN	--	--	--	--	--	PWM2LE	PWM1LE	PWM0LE
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3 -- 保留，须均为0。

Bit2~Bit0 PWMnLE: PWM通道n的数据加载使能位(n=0-2)（加载完成后硬件清零）；

(n=0,对应 PWM 通道 0,1；n=1,对应 PWM 通道 2,3；n=2,对应 PWM 通道 4,5)

1= 使能加载周期，占空比数据(PERIODn、CMPn)。

0= 写0无效。

13.5.5 PWM 输出极性控制寄存器 PWMPINV

F122H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPINV	--	--	PWM5PINV	PWM4PINV	PWM3PINV	PWM2PINV	PWM1PINV	PWM0PINV
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6 -- 保留，须均为0。

Bit5~Bit0 PWMnPINV: PWM通道n输出极性控制位（n=0-5）；

1= 反向输出；

0= 正常输出。

13.5.6 PWM 计数器模式控制寄存器 PWMCNTM

F127H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTM	--	--	--	--	--	PWM2CNTM	PWM1CNTM	PWM0CNTM
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3 -- 保留，须均为0。

Bit2~Bit0 PWMnCNTM: PWM通道n计数器模式控制位（n=0-2）；
(n=0,对应PWM通道0,1；n=1,对应PWM通道2,3；n=2,对应PWM通道4,5)
1= 自动加载模式；
0= One-shot模式。

13.5.7 PWM 计数器使能控制寄存器 PWMCNTE

F126H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTE	--	--	--	--	--	PWM2CNTE	PWM1CNTE	PWM0CNTE
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3 -- 保留，须均为0。

Bit2~Bit0 PWMnCNTE: PWM通道n计数器使能控制位(n=0-2)；
(n=0,对应PWM通道0,1；n=1,对应PWM通道2,3；n=2,对应PWM通道4,5)
1= PWMn计数器开启（PWMn开始输出）；
0= PWMn计数器停止（软件写0则计数器停止并清掉计数器值）。
（单次模式完成该位硬件清0）

13.5.8 PWM 计数器模式控制寄存器 PWMCNTCLR

F128H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTCLR	--	--	--	--	--	PWM2CNTCLR	PWM1CNTCLR	PWM0CNTCLR
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3 -- 保留，须均为0。

Bit2~Bit0 PWMnCNTCLR: PWM通道n计数器清零控制位(n=0-2)（硬件自动清零）；
(n=0,对应PWM通道0,1；n=1,对应PWM通道2,3；n=2,对应PWM通道4,5)
1= PWMn计数器清零；
0= 写0无效。

13.5.9 PWM 周期数据寄存器低 8 位 PWMPnL (n=0-2)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPnL	PWMPnL7	PWMPnL6	PWMPnL5	PWMPnL4	PWMPnL3	PWMPnL2	PWMPnL1	PWMPnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMPnL (n=0-2)地址：F130H，F134H，F138H。

(n=0,对应 PWM 通道 0,1；n=1,对应 PWM 通道 2,3；n=2,对应 PWM 通道 4,5)

Bit7~Bit0　　PWMPnL<7:0>:　PWM通道n周期数据寄存器低8位。

13.5.10 PWM 周期数据寄存器高 8 位 PWMPnH (n=0-2)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPnH	PWMPnH7	PWMPnH6	PWMPnH5	PWMPnH4	PWMPnH3	PWMPnH2	PWMPnH1	PWMPnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMPnH (n=0-2)地址：F131H，F135H，F139H。

(n=0,对应 PWM 通道 0,1；n=1,对应 PWM 通道 2,3；n=2,对应 PWM 通道 4,5)

Bit7~Bit0　　PWMPnH<7:0>:　PWM通道n周期数据寄存器高8位。

13.5.11 PWM 比较数据寄存器低 8 位 PWMDnL (n=0-5)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDnL	PWMDnL7	PWMDnL6	PWMDnL5	PWMDnL4	PWMDnL3	PWMDnL2	PWMDnL1	PWMDnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMDnL (n=0-5)地址：F140H，F142H，F144H，F146H，F148H，F14AH。

Bit7~Bit0　　PWMDnL<7:0>:　PWM通道n比较数据（占空比数据）寄存器低8位。

13.5.12 PWM 比较数据寄存器高 8 位 PWMDnH (n=0-5)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDnH	PWMDnH7	PWMDnH6	PWMDnH5	PWMDnH4	PWMDnH3	PWMDnH2	PWMDnH1	PWMDnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMDnH (n=0-5)地址：F141H，F143H，F145H，F147H，F149H，F14BH。

Bit7~Bit0　　PWMDnH<7:0>:　PWM通道n比较数据（占空比数据）寄存器高8位。

13.5.13 PWM 死区使能控制寄存器 PWMDTE

F160H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDTE	--	--	--	--	--	PWM45DTE	PWM23DTE	PWM01DTE
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3 -- 保留，须均为0。
Bit2 PWM45DTE: PWM4/5通道死区延时使能位；
1= 使能；
0= 禁止。
Bit1 PWM23DTE: PWM2/3通道死区延时使能位；
1= 使能；
0= 禁止。
Bit0 PWM01DTE: PWM0/1通道死区延时使能位；
1= 使能；
0= 禁止。

13.5.14 PWM0/1 死区时间设置寄存器 PWM01DT

F161H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM01DT	PWM01 DT7	PWM01 DT6	PWM01 DT5	PWM01 DT4	PWM01 DT3	PWM01 DT2	PWM01 DT1	PWM01 DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM01DT<7:0>: PWM通道0/1死区延时数据寄存器。

13.5.15 PWM2/3 死区时间设置寄存器 PWM23DT

F162H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM23DT	PWM23 DT7	PWM23 DT6	PWM23 DT5	PWM23 DT4	PWM23 DT3	PWM23 DT2	PWM23 DT1	PWM23 DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM23DT<7:0>: PWM通道2/3死区延时数据寄存器。

13.5.16 PWM4/5 死区时间设置寄存器 PWM45DT

F163H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM45DT	PWM45 DT7	PWM45 DT6	PWM45 DT5	PWM45 DT4	PWM45 DT3	PWM45 DT2	PWM45 DT1	PWM45 DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM45DT<7:0>: PWM通道4/5死区延时数据寄存器。

13.6 PWM 中断

PWM 总共具有 9 个中断标志，包含 3 个零点中断标志和 6 个向下比较中断标志，中断标志位的产生与对应中断使能位是否开启无关。开启 PWM 任何一种类型的中断均需打开全局中断使能位（EA=1）、PWM 总中断使能位 PWMIE，才能成功配置 PWM 中断功能。PWM 的所有中断共用一个中断向量入口，故进入中断服务程序后用户可通过中断标志位判断是哪种类型中断产生。

PWM 的中断使能、优先级可通过如下相关寄存器位设置。

13.6.1 中断屏蔽寄存器 EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	--	ADCIE	PWMIE	ET5	--	--
R/W	R/W	R/W	--	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI中断使能位；
1= 允许SPI中断；
0= 禁止SPI中断。
Bit6 I2CIE: I^2C中断使能位；
1= 允许I^2C中断；
0= 禁止I^2C中断。
Bit5 -- 保留，须为0。
Bit4 ADCIE: ADC中断使能位；
1= 允许ADC中断；
0= 禁止ADC中断。
Bit3 PWMIE: PWM总中断使能位；
1= 允许PWM所有中断；
0= 禁止PWM所有中断。
Bit2 ET5: Timer5中断使能位；
1= 允许Timer5中断；
0= 禁止Timer5中断。
Bit1~Bit0 -- 保留，须为0。

13.6.2 中断优先级控制寄存器 EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	--	PADC	PPWM	PT5	--	--
R/W	R/W	R/W	--	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit6 PI2C: I2C中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit5 -- 保留，须为0。
Bit4 PADC: ADC中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit3 PPWM: PWM中断优先级控制位
1= 设置为高级中断；
0= 设置为低级中断。
Bit2 PT5: TIMER5中断优先级控制位
1= 设置为高级中断；
0= 设置为低级中断。
Bit1~Bit0 -- 保留，须为0。

13.6.3 PWM 零点中断屏蔽寄存器 PWMZIE

F169H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMZIE	--	--	--	--	--	PWM2ZIE	PWM1ZIE	PWM0ZIE
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3 -- 保留，须均为0。
Bit2~Bit0 PWMnZIE: PWM通道n零点中断屏蔽位(n=0-2)；
(n=0,对应PWM通道0,1；n=1,对应PWM通道2,3；n=2,对应PWM通道4,5)
1= 使能中断；
0= 禁止中断。

13.6.4 PWM 向下比较中断屏蔽寄存器 PWMDIE

F16BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDIE	--	--	PWM5DIE	PWM4DIE	PWM3DIE	PWM2DIE	PWM1DIE	PWM0DIE
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6 -- 保留，须均为0。

Bit5~Bit0 PWMnDIE: PWM通道n向下比较中断屏蔽位 (n=0-5)；

1= 使能中断；

0= 禁止中断。

13.6.5 PWM 零点中断标志寄存器 PWMZIF

F16DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMZIF	--	--	--	--	--	PWM2ZIF	PWM1ZIF	PWM0ZIF
R/W	R	R	R	R	R	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3 -- 保留，须均为0。

Bit2~Bit0 PWMnZIF: PWM通道n零点中断标志位(n=0-2)；

(n=0,对应PWM通道0,1；n=1,对应PWM通道2,3；n=2,对应PWM通道4,5)

1= 产生中断（软件清零）；

0= 未产生中断。

13.6.6 PWM 向下比较中断标志寄存器 PWMDIF

F16FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDIF	--	--	PWM5DIF	PWM4DIF	PWM3DIF	PWM2DIF	PWM1DIF	PWM0DIF
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6 -- 保留，须均为0。

Bit5~Bit0 PWMnDIF: PWM通道n向下比较中断标志位 (n=0-5)；

1= 产生中断（软件清零）；

0= 未产生中断。

14. 硬件 LED 矩阵驱动器

14.1 概述

该芯片内集成硬件 LED 矩阵驱动电路，可方便用户实现 LED 的显示驱动。

14.2 特性

硬件 LED 矩阵驱动器具有如下特性：

- 1/4、1/5、1/6、1/8 四种 DUTY 可选。
- 时钟源固定为 F_{FIX} (8MHz)。
- 最多支持 8 个 COM 口、26 个 SEG 口。
- COM 口共阴、共阳两种驱动方式可选。
- COM 口有效时间可通过 8bit 寄存器设置。
- COM 口电流 55 mA、220mA 两档可选。
- SEG 口电流 16 档可选，最大电流可达 50mA。
- LED 计数时钟源周期可选。
- 死区时间可配置。
- 支持调光模式，调光时间可选。

14.3 相关寄存器

14.3.1 LED 驱动模式选择寄存器 LEDMODE

F769H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDMODE	LEDMODE7	LEDMODE6	LEDMODE5	LEDMODE4	LEDMODE3	LEDMODE2	LEDMODE1	LEDMODE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7-0 LEDMODE LED驱动模式选择寄存器；
0x55= LED矩阵驱动模式有效，相关寄存器生效；
0xaa= LED点阵驱动模式有效，相关寄存器生效；
其它值= 无效。

14.3.2 LED 控制寄存器 LEDCON

F765H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCON	LED_EN	DUTY1	DUTY0	CC_CA	--	LED_FIXCLK_EN	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	W
复位值	0	0	0	0	0	0	0	0

Bit7 LED_EN: LED使能控制位；
1= LED使能；（LED_FIXCLK_EN=1有效）
0= LED禁止。
Bit6~Bit5 DUTY<1:0>: LED的占空比选择位；
11= 1/4DUTY；
10= 1/5DUTY；
01= 1/6DUTY；
00= 1/8DUTY；
Bit4 CC_CA: LED驱动模式选择位；
1= 共阳驱动模式；
0= 共阴驱动模式。
Bit3 - 保留。
Bit2 LED_FIXCLK_EN: LED运行时钟使能控制
1= 使能；
0= 禁止。
Bit1~Bit0 - 保留。

COM 选择说明表

DUTY	ICOM0	ICOM1	ICOM2	ICOM3	ICOM4	ICOM5	ICOM6	ICOM7	有效 SEG 口
11	LED_C0	LED_C1	LED_C2	LED_C3	-	-	-	-	LED_S0-LED_S25
10	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	-	-	-	LED_S1-LED_S25
01	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	LED_C5	-	-	LED_S2-LED_S25
00	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	LED_C5	LED_C6	LED_C7	LED_S4-LED_S25

注：上表中 ICOM0-ICOM7 为 LED 内部 COM 驱动输出信号。

LED_C0-LED_C7，LED_S0-LED_S25 为 LED 内部驱动信号最终映射到的管脚名称。

14.3.3 LED 时钟选择寄存器 LEDCKS

F766H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCKS	-	-	DTSEL1	DTSEL0	-	-	CLK1	CLK0
R/W	W	W	R/W	R/W	W	W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6 -- 保留
Bit5~Bit4 DTSEL<1:0>: 矩阵LED死区时间选择
00: 8*Tfix
01: 16*Tfix
10: 32*Tfix
11: 64*Tfix
Bit3~Bit2 --: 保留
Bit1~Bit0 CLK<1:0>: LED计数时钟（TLED_CLK）。
00: 32*Tfix
10: 64*Tfix
10: 128*Tfix
11: 256*Tfix

14.3.4 COM 口有效时间选择寄存器 LEDCOMTIME

F768H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCOMTIME	COMT7	COMT6	COMT5	COMT4	COMT3	COMT2	COMT1	COMT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 COMT<7:0>: COM口有效时间设置。
注：禁止设置成0x00；
COM时间 = (COMT<7:0> + 1) * T_{LED_CLK}。

14.3.5 调光时间选择寄存器 LEDSEGTIME

F76DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGTIME	SEGT7	SEGT6	SEGT5	SEGT4	SEGT3	SEGT2	SEGT1	SEGT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 SEGT<7:0>: SEG口有效时间设置。
注：禁止设置成0x00；
调光有效时间 = (SEGT<7:0> + 1) * TLED_CLK 。

备注：扫描一个 COM 的有效显示占空比为：(SEGT<7:0> + 1) / (COMT<7:0> + 1) ；SEGT[7:0]≤COMT[7:0]。

14.3.6 COM 口使能控制寄存器 LEDCOMEN

F760H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCOMEN	COMEN7	COMEN6	COMEN5	COMEN4	COMEN3	COMEN2	COMEN1	COMEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 COMEN<7:0>: LED_C7-LED_C0端口使能控制位；
1= 使能；
0= 禁止。

14.3.7 SEG 口使能控制寄存器 LEDSEGEN0

F761H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGEN0	SEGEN7	SEGEN6	SEGEN5	SEGEN4	SEGEN3	SEGEN2	SEGEN1	SEGEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 SEGEN<7:0>: LED_S7-LED_S0端口使能控制位；
1= 使能；
0= 禁止。

14.3.8 SEG 口使能控制寄存器 LEDSEGEN1

F762H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGEN1	SEGEN15	SEGEN14	SEGEN13	SEGEN12	SEGEN11	SEGEN10	SEGEN9	SEGEN8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 SEGEN<15:8>: LED_S15-LED_S8端口使能控制位；
1= 使能；
0= 禁止。

14.3.9 SEG 口使能控制寄存器 LEDSEGEN2

F763H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGEN2	SEGEN23	SEGEN22	SEGEN21	SEGEN20	SEGEN19	SEGEN18	SEGEN17	SEGEN16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 SEGEN<23:16>: LED_S23-LED_S16端口使能控制位；
1= 使能；
0= 禁止。

14.3.10 SEG 口使能控制寄存器 LEDSEGEN3

F764H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGEN3	-	-	-	-	-	-	SEGEN25	SEGEN24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit2 - 保留，须为0。
Bit1~Bit0 SEGEN<25:24>: LED_S25-LED_S24端口使能控制位；
1= 使能；
0= 禁止。

14.3.11 COM0 对应 SEG 数据寄存器 LEDC0DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC0DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

LEDC0DATA0 地址：F740H；LEDC0DATA1 地址：F741H；LEDC0DATA2 地址：F742H；LEDC0DATA3 地址：F743H。(n=3 时，bit7-bit2 值写入无效)。

Bit7~Bit0 SEG<8n+7:8n>: COM0端口有效时，SEG[8n+7]-SEG[8n]端口数据输出；
1= 高电平；
0= 低电平。

14.3.12 COM1 对应 SEG 数据寄存器 LEDC1DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC1DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

LEDC1DATA0 地址：F744H；LEDC1DATA1 地址：F745H；LEDC1DATA2 地址：F746H；LEDC1DATA3 地址：F747H。(n=3 时，bit7-bit2 值写入无效)。

Bit7~Bit0 SEG<8n+7:8n>: COM1端口有效时，SEG[8n+7]-SEG[8n]端口数据输出；
1= 高电平；
0= 低电平。

14.3.13 COM2 对应 SEG 数据寄存器 LEDC2DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC2DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

LEDC2DATA0 地址：F748H；LEDC2DATA1 地址：F749H；LEDC2DATA2 地址：F74AH；LEDC2DATA3 地址：F74BH。(n=3 时，bit7-bit2 值写入无效)。

Bit7~Bit0 SEG<8n+7:8n>: COM2端口有效时，SEG[8n+7]-SEG[8n]端口数据输出；
1= 高电平；
0= 低电平。

14.3.14 COM3 对应 SEG 数据寄存器 LEDC3DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC3DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

LEDC3DATA0 地址：F74CH；LEDC3DATA1 地址：F74DH；LEDC3DATA2 地址：F74EH；LEDC3DATA3 地址：F74FH；(n=3 时，bit7-bit2 值写入无效)。

Bit7~Bit0　　SEG<8n+7:8n>:　COM3端口有效时，SEG[8n+7]-SEG[8n]端口数据输出；
1=　高电平；
0=　低电平。

14.3.15 COM4 对应 SEG 数据寄存器 LEDC4DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC4DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

LEDC4DATA0 地址：F750H；LEDC4DATA1 地址：F751H；LEDC4DATA2 地址：F752H；LEDC4DATA3 地址：F753H。(n=3 时，bit7-bit2 值写入无效)。

Bit7~Bit0　　SEG<8n+7:8n>:　COM4端口有效时，SEG[8n+7]-SEG[8n]端口数据输出；
1=　高电平；
0=　低电平。

14.3.16 COM5 对应 SEG 数据寄存器 LEDC5DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC5DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

LEDC5DATA0 地址：F754H；LEDC5DATA1 地址：F755H；LEDC5DATA2 地址：F756H；LEDC5DATA3 地址：F757H。(n=3 时，bit7-bit2 值写入无效)。

Bit7~Bit0　　SEG<8n+7:8n>:　COM5端口有效时，SEG[8n+7]-SEG[8n]端口数据输出；
1=　高电平；
0=　低电平。

14.3.17 COM6 对应 SEG 数据寄存器 LEDC6DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC6DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

LEDC6DATA0 地址：F758H；LEDC6DATA1 地址：F759H；LEDC6DATA2 地址：F75AH；LEDC6DATA3 地址：F75BH；(n=3 时，bit7-bit2 值写入无效)。

Bit7~Bit0　　SEG<8n+7:8n>:　COM6端口有效时，SEG[8n+7]-SEG[8n]端口数据输出；
1=　高电平；
0=　低电平。

14.3.18 COM7 对应 SEG 数据寄存器 LEDC7DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC7DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

LEDC7DATA0 地址：F75CH；LEDC7DATA1 地址：F75DH；LEDC7DATA0 地址：F75EH；LEDC7DATA3 地址：F75FH。(n=3 时，bit7-bit2 值写入无效)。

Bit7~Bit0　　SEG<8n+7:8n>:　COM7端口有效时，SEG[8n+7]-SEG[8n]端口数据输出；
1=　高电平；
0=　低电平。

14.3.19 LED 矩阵驱动端口电流控制寄存器 LEDDRV

F76BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDDRV	--	--	--	--	SDRC3	SDRC2	SDRC1	SDRC0
R/W	W	W	W	W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit4　　--　保留，须均为0。
Bit3~Bit0　　SDRC<3:0>:　所有的SEG口拉电流驱动选择控制位；
相关管脚驱动使能才生效，否则电流默认为最大值。

0000=	0mA；	1000=	26.9mA；
0001=	4.9mA；	1001=	31.8mA；
0010=	7.4mA；	1010=	34.3mA；
0011=	12.2mA；	1011=	39.2mA；
0100=	14.7mA；	1100=	41.6mA；
0101=	19.5mA；	1101=	46.1mA；
0110=	22.1mA；	1110=	49mA；
0111=	26.9mA；	1111=	50mA。

14.3.20 LED COM 口驱动电流选择寄存器 LEDCOMDRV

F76EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCOMDRV	CDRC7	CDRC6	CDRC5	CDRC4	CDRC3	CDRC2	CDRC1	CDRC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit[7:0]　　CDRC<7:0>:　对应COM7-COM0的IO端口驱动电流选择;
(只与该寄存器位的值有关，与LED模块使能和COM口使能无关，电流默认较小值)
1=　220mA
0=　55mA

14.3.21 管脚驱动使能寄存器 LEDSEGDR0

F76FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGDR0	LEDSEGDR0_7	LEDSEGDR0_6	LEDSEGDR0_5	LEDSEGDR0_4	LEDSEGDR0_3	LEDSEGDR0_2	LEDSEGDR0_1	LEDSEGDR0_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 LEDSEGDR0_7:SEG7管脚驱动使能位；

1= SEG7（P13）管脚驱动使能；SEG7（P13）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG7（P13）管脚驱动使能禁止,SEG7（P13）管脚的拉电流驱动为默认值。

Bit6 LEDSEGDR0 _6:SEG6管脚驱动使能位；

1= SEG6（P12）管脚驱动使能；SEG6（P12）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG6（P12）管脚驱动使能禁止,SEG6（P12）管脚的拉电流驱动为默认值。

Bit5 LEDSEGDR0 _5:SEG5管脚驱动使能位；

1= SEG5（P11）管脚驱动使能；SEG5（P11）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG5（P11）管脚驱动使能禁止,SEG5（P11）管脚的拉电流驱动为默认值。

Bit4 LEDSEGDR0 _4:SEG4管脚驱动使能位；

1= SEG4（P10）管脚驱动使能；SEG4（P10）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG4（P10）管脚驱动使能禁止,SEG4（P10）管脚的拉电流驱动为默认值。

Bit3 LEDSEGDR0 _3:SEG3管脚驱动使能位；

1= SEG3（P07）管脚驱动使能；SEG3（P07）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG3（P07）管脚驱动使能禁止,SEG3（P07）管脚的拉电流驱动为默认值。

Bit2 LEDSEGDR0_2:SEG2管脚驱动使能位；

1= SEG2（P06）管脚驱动使能；SEG2（P06）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG2（P06）管脚驱动使能禁止,SEG2（P06）管脚的拉电流驱动为默认值。

Bit1 LEDSEGDR0 _1:SEG1管脚驱动使能位；

1= SEG1（P05）管脚驱动使能；SEG1（P05）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG1（P05）管脚驱动使能禁止,SEG1（P05）管脚的拉电流驱动为默认值。

Bit0 LEDSEGDR0 _0:SEG0管脚驱动使能位；

1= SEG0（P04）管脚驱动使能；SEG0（P04）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG0（P04）管脚驱动使能禁止,SEG0（P04）管脚的拉电流驱动为默认值。

14.3.22 管脚驱动使能寄存器 LEDSEGDR1

F770H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGDR1	LEDSEGDR1 _7	LEDSEGDR1 _6	LEDSEGDR1 _5	LEDSEGDR1 _4	LEDSEGDR1_3	LEDSEGDR1_2	LEDSEGDR1 _1	LEDSEGDR1 _0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 LEDSEGDR1 _7:SEG15管脚驱动使能位；

1= SEG15（P23）管脚驱动使能；SEG15（P23）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG15（P23）管脚驱动使能禁止, SEG15（P23）管脚的拉电流驱动为默认值。

Bit6 LEDSEGDR1 _6:SEG14管脚驱动使能位；

1= SEG14（P22）管脚驱动使能；SEG14（P22）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG14（P22）管脚驱动使能禁止, SEG14（P22）管脚的拉电流驱动为默认值。

Bit5 LEDSEGDR1 _5:SEG13管脚驱动使能位；

1= SEG13（P21）管脚驱动使能；SEG13（P21）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG13（P21）管脚驱动使能禁止,SEG13（P21）管脚的拉电流驱动为默认值。

Bit4 LEDSEGDR1 _4:SEG12管脚驱动使能位；

1= SEG12（P20）管脚驱动使能；SEG12（P20）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG12（P20）管脚驱动使能禁止,SEG12（P20）管脚的拉电流驱动为默认值。

Bit3 LEDSEGDR1 _3:SEG11管脚驱动使能位；

1= SEG11（P17）管脚驱动使能；SEG11（P17）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG11（P17）管脚驱动使能禁止,SEG11（P17）管脚的拉电流驱动为默认值。

Bit2 LEDSEGDR1_2:SEG10管脚驱动使能位；

1= SEG10（P16）管脚驱动使能；SEG10（P16）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG10（P16）管脚驱动使能禁止,SEG10（P16）管脚的拉电流驱动为默认值。

Bit1 LEDSEGDR1 _1:SEG9管脚驱动使能位；

1= SEG9（P15）管脚驱动使能；SEG9（P15）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG9（P15）管脚驱动使能禁止, SEG9（P15）管脚的拉电流驱动为默认值。

Bit0 LEDSEGDR1 _0:SEG8管脚驱动使能位；

1= SEG8（P14）管脚驱动使能；SEG8（P14）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG8（P14）管脚驱动使能禁止, SEG8（P14）管脚的拉电流驱动为默认值。

14.3.23 管脚驱动使能寄存器 LEDSEGDR2

F771H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGDR2	LEDSEGDR2 _7	LEDSEGDR2 _6	LEDSEGDR2 _5	LEDSEGDR2 _4	LEDSEGDR2_3	LEDSEGDR2_2	LEDSEGDR2 _1	LEDSEGDR2 _0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 LEDSEGDR2 _7:SEG23管脚驱动使能位；
1= SEG23（P53）管脚驱动使能；SEG23（P53）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；
0= SEG23（P53）管脚驱动使能禁止, SEG23（P53）管脚的拉电流驱动为默认值。

Bit6 LEDSEGDR2 _6:SEG22管脚驱动使能位；
1= SEG22（P52）管脚驱动使能；SEG22（P52）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；
0= SEG22（P52）管脚驱动使能禁止, SEG22（P52）管脚的拉电流驱动为默认值。

Bit5 LEDSEGDR2 _5:SEG21管脚驱动使能位；
1= SEG21（P51）管脚驱动使能；SEG21（P51）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；
0= SEG21（P51）管脚驱动使能禁止,SEG21（P51）管脚的拉电流驱动为默认值。

Bit4 LEDSEGDR2 _4:SEG20管脚驱动使能位；
1= SEG20（P50）管脚驱动使能；SEG20（P50）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；
0= SEG20（P50）管脚驱动使能禁止,SEG20（P50）管脚的拉电流驱动为默认值。

Bit3 LEDSEGDR2 _3:SEG19管脚驱动使能位；
1= SEG19（P27）管脚驱动使能；SEG19（P27）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；
0= SEG19（P27）管脚驱动使能禁止,SEG19（P27）管脚的拉电流驱动为默认值。

Bit2 LEDSEGDR2_2:SEG18管脚驱动使能位；
1= SEG18（P26）管脚驱动使能；SEG18（P26）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；
0= SEG18（P26）管脚驱动使能禁止,SEG18（P26）管脚的拉电流驱动为默认值。

Bit1 LEDSEGDR2 _1:SEG17管脚驱动使能位；
1= SEG17（P25）管脚驱动使能；SEG17（P25）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；
0= SEG17（P25）管脚驱动使能禁止, SEG17（P25）管脚的拉电流驱动为默认值。

Bit0 LEDSEGDR2 _0:SEG16管脚驱动使能位；
1= SEG16（P24）管脚驱动使能；SEG16（P24）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；
0= SEG16（P24）管脚驱动使能禁止, SEG16（P24）管脚的拉电流驱动为默认值。

14.3.24 管脚驱动使能寄存器 LEDSEGDR3

F772H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGDR 3	--	LEDSEGDR 3 _6	LEDSEGDR 3 _5	LEDSEGDR 3 _4	LEDSEGDR 3_3	LEDSEGDR 3_2	LEDSEGDR 3 _1	LEDSEGDR 3 _0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 -- 保留，须为0

Bit6 LEDSEGDR3 _6:P03管脚驱动使能位；

1= P03管脚驱动使能；P03管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P03管脚驱动使能禁止,P03管脚的拉电流驱动为默认值。

Bit5 LEDSEGDR3 _5:P02管脚驱动使能位；

1= P02管脚驱动使能；P02管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P02管脚驱动使能禁止,P02管脚的拉电流驱动为默认值。

Bit4 LEDSEGDR3 _4:P01管脚驱动使能位；

1= P01管脚驱动使能；P01管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P01管脚驱动使能禁止,P01管脚的拉电流驱动为默认值。

Bit3 LEDSEGDR3 _3:P00管脚驱动使能位；

1= P00管脚驱动使能；P00管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P00管脚驱动使能禁止,P00管脚的拉电流驱动为默认值。

Bit2 保留，须为0

Bit1 LEDSEGDR3 _1:SEG25管脚驱动使能位；

1= SEG25（P55）管脚驱动使能；SEG25（P55）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG25（P55）管脚驱动使能禁止, SEG25（P55）管脚的拉电流驱动为默认值。

Bit0 LEDSEGDR3 _0:SEG24管脚驱动使能位；

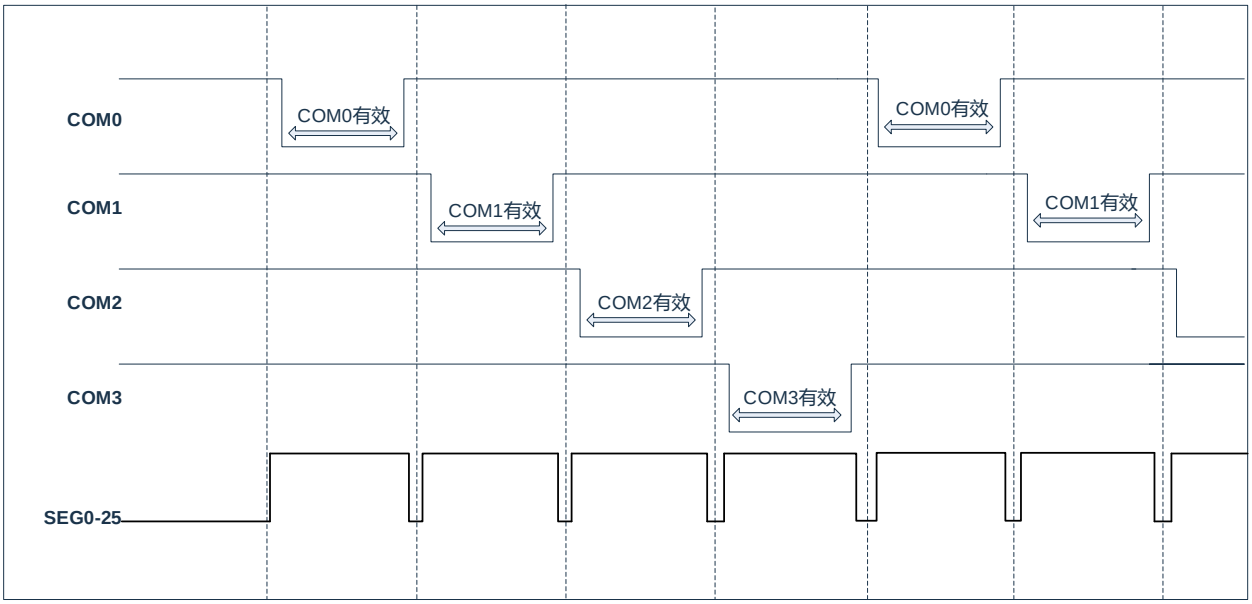
1= SEG24（P54）管脚驱动使能；SEG24（P54）管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= SEG24（P54）管脚驱动使能禁止, SEG24（P54）管脚的拉电流驱动为默认值。

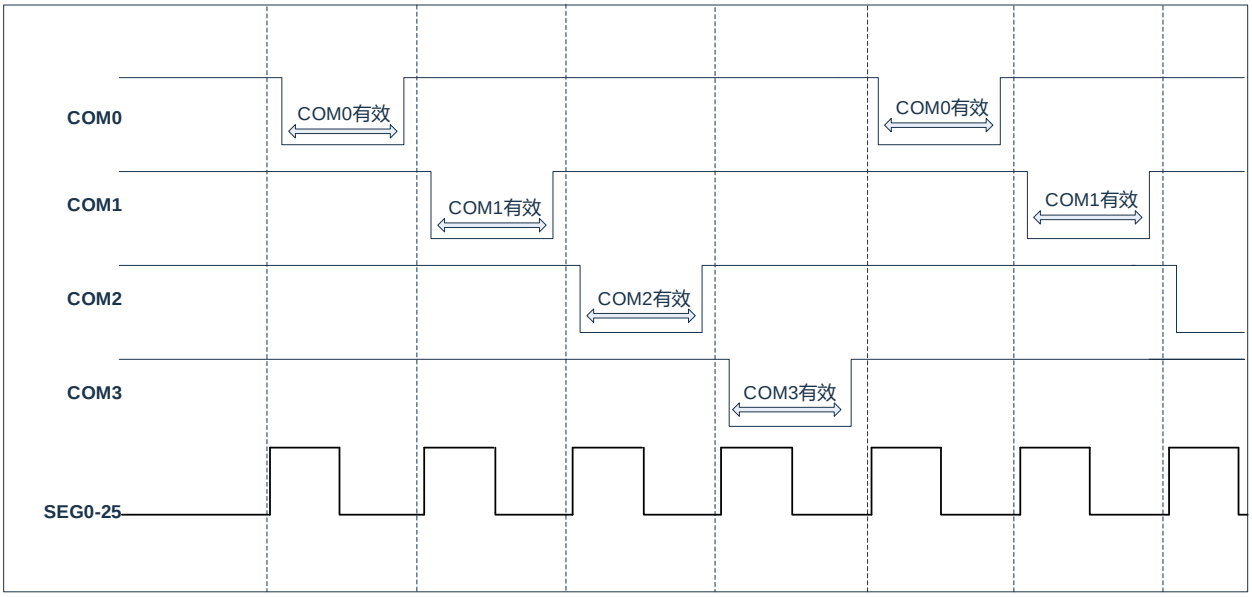
14.4 LED 驱动器输出波形

根据 LED 驱动器相关配置寄存器，可设置对应的 LED 驱动器输出波形。

LED 配置 1/4DUTY、共阴驱动模式、seg 数据全为 1、seg 输出占空比为 1 的波形如下图所示：



LED 配置 1/4DUTY、共阴驱动模式、seg 数据全为 1、seg 输出占空比为 1/2 的波形如下图所示：



15. 硬件 LED 点阵驱动器

15.1 概述

LED 点阵驱动是通过配置 LED0~LED8 口，从而驱动多个 LED 灯，方便用户进行 LED 点阵驱动。

15.2 特性

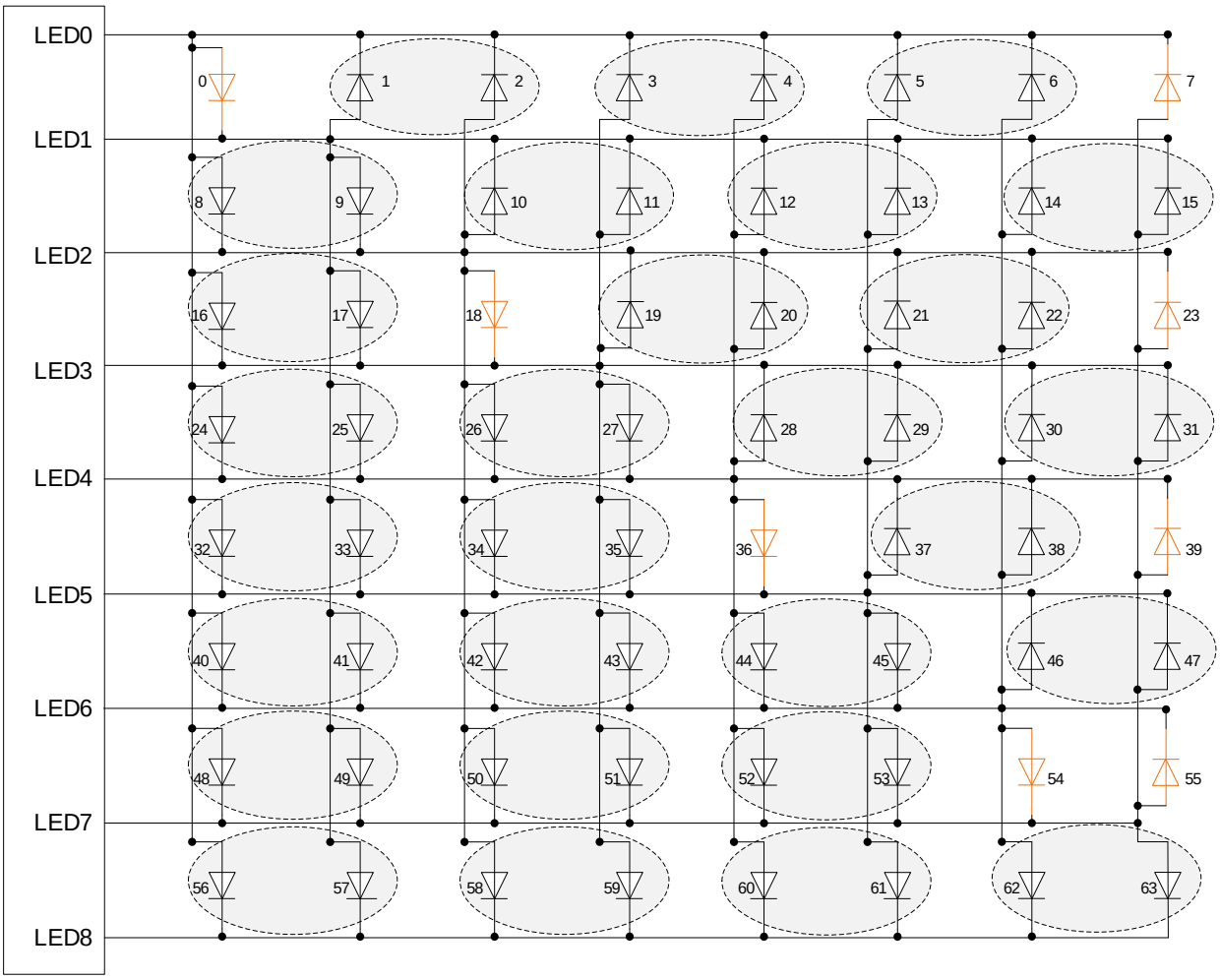
LED 点阵驱动模式功能特点：

- 支持最大 64 灯 LED 驱动，可配置选择点阵 4x4、5x5、6x6、6x7、7x7、7x8、8x8。
- 每个灯支持两种导通时间选择，每种导通时间为 8 位定时设置。
- 每个灯显示数据单独可选。
- LED 点阵驱动支持循环扫描模式和中断扫描模式。
- 时钟源固定为 F_{FIX} (8MHz)。
- LED 计数时钟源周期可选。
- LED 点阵驱动模式 LED0-LED8 管脚使能可选。
- LED 点阵驱动管脚电流 16 档可选，最大电流可达 50mA。
- LED0-LED7 的起始位置可选，其他口顺序循环。
- 64 灯点阵地址具有唯一性，见下面点阵描述，用于输入开关灯信息。

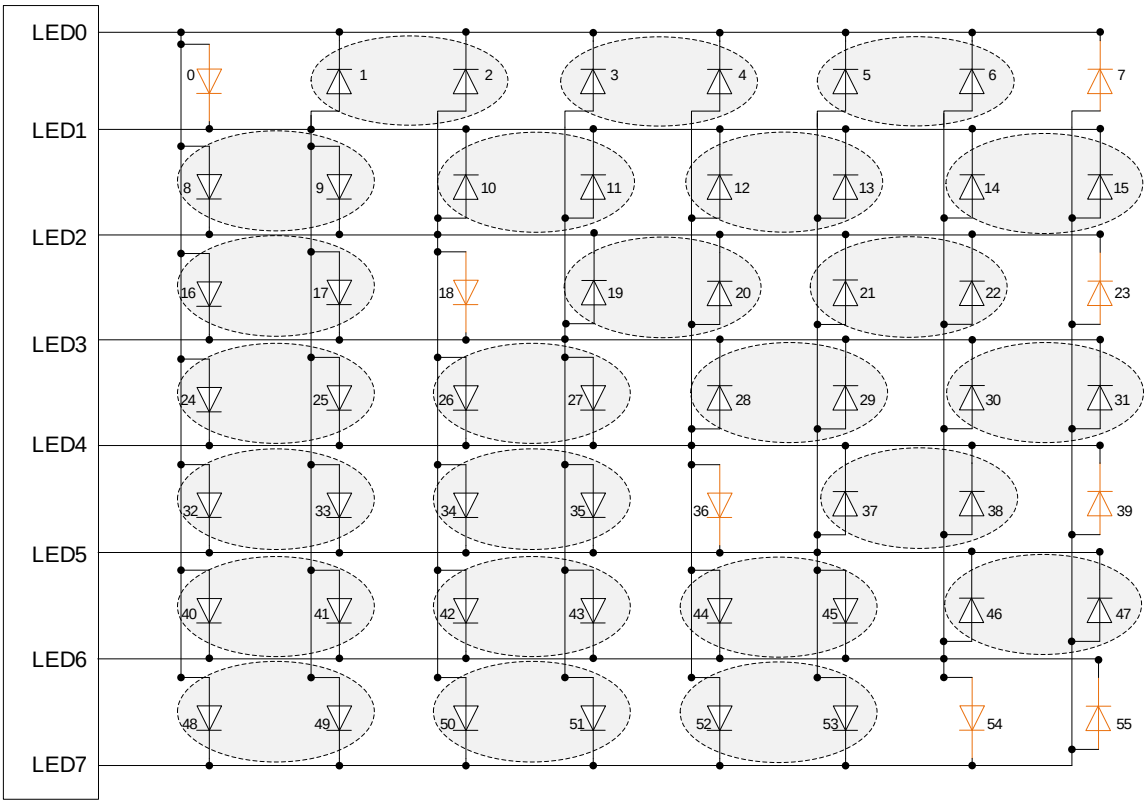
15.3 功能描述

LED点阵是通过8*8点阵双灯模式扫描，即一次点两个灯（共阴极），对应LED0~LED8口，最多可配置驱动8x8=64个灯。配置对应地址的灯亮情况（1表示亮灯，0表示不亮灯），硬件将解析亮灯地址和当前扫描地址，并自动完成相应IO口的输出控制。可配置4x4、5x5、6x6、6x7、7x7、7x8、8x8，不同大小点阵（对应灯地址不变）。

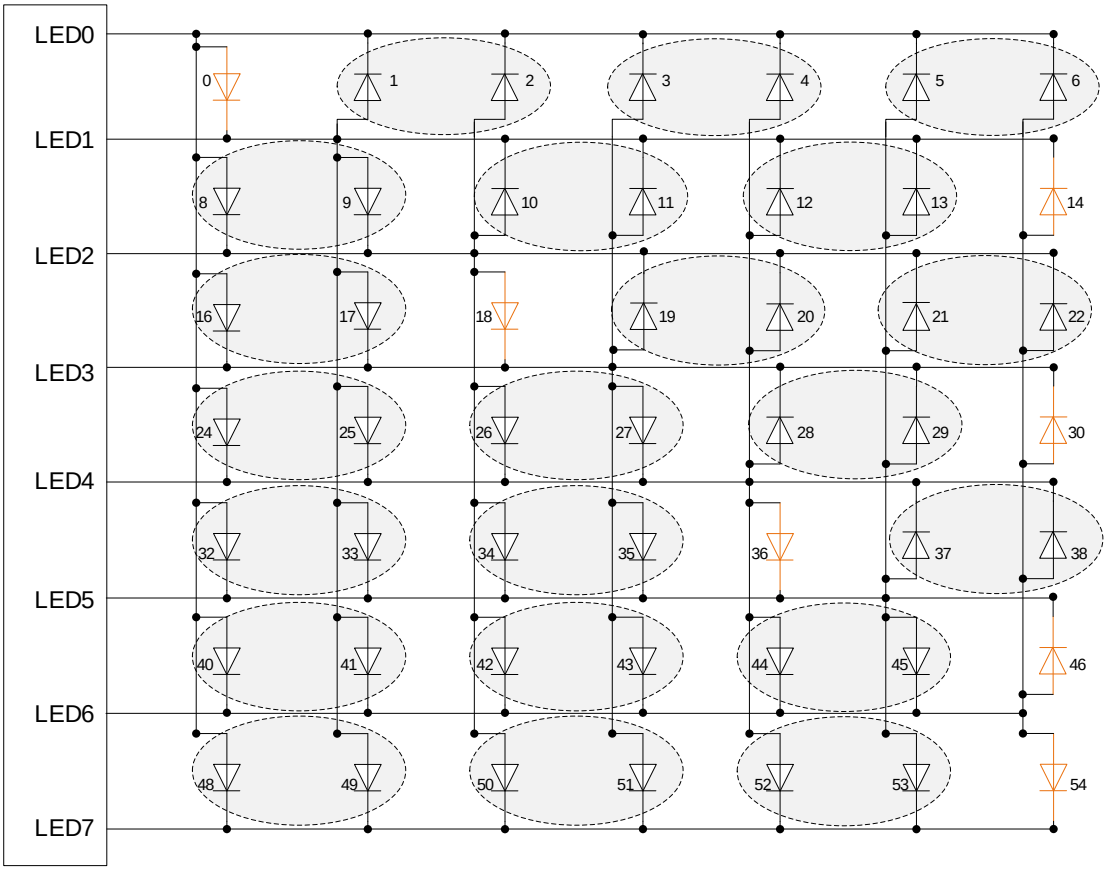
8*8点阵如下图所示（图中黄色灯为单独扫描灯）：



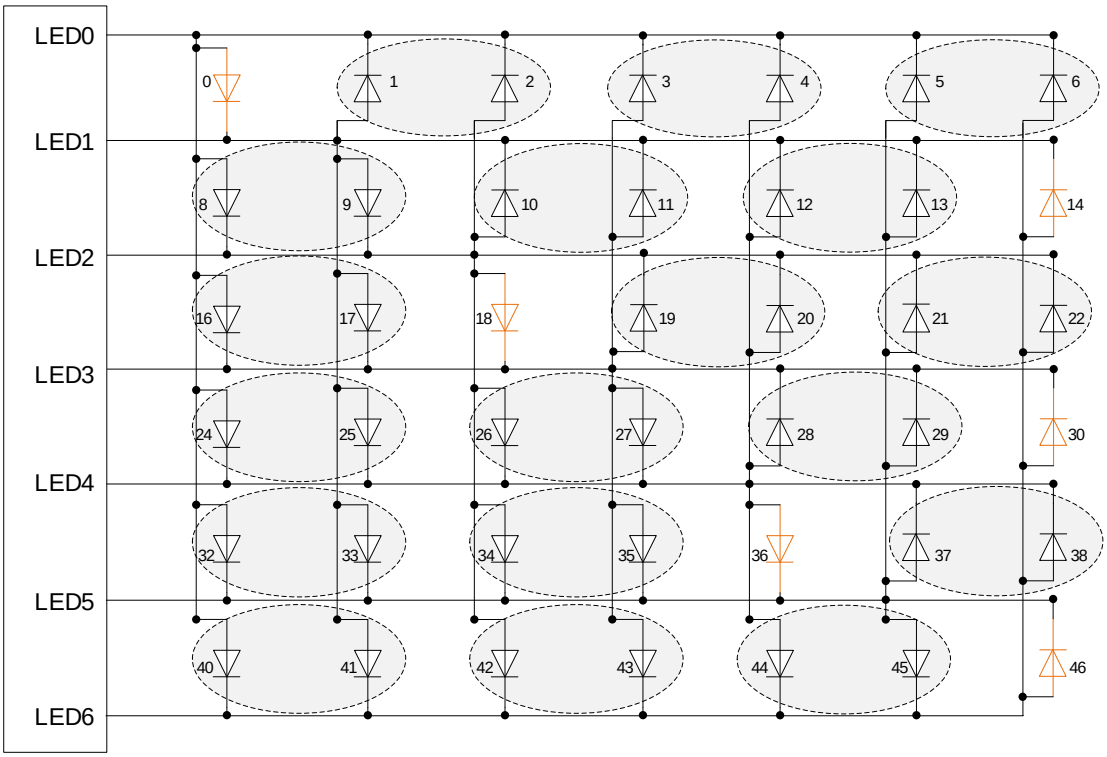
7*8点阵如下图所示：



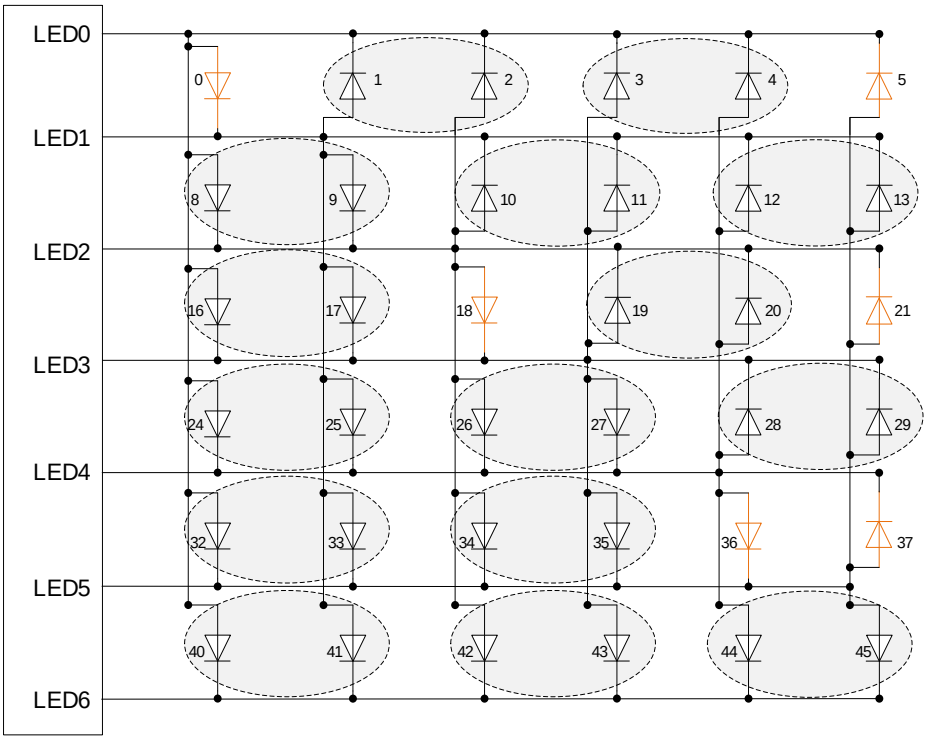
7*7点阵如下图所示：



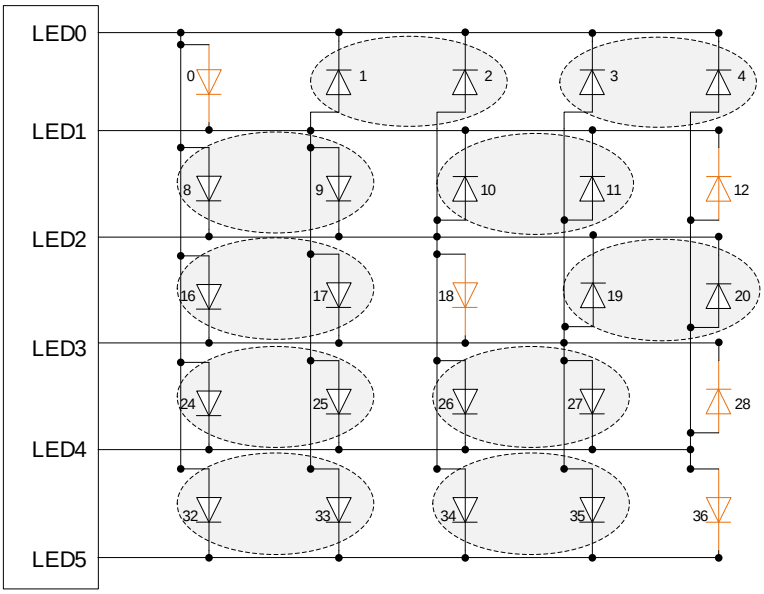
6*7点阵如下图所示：



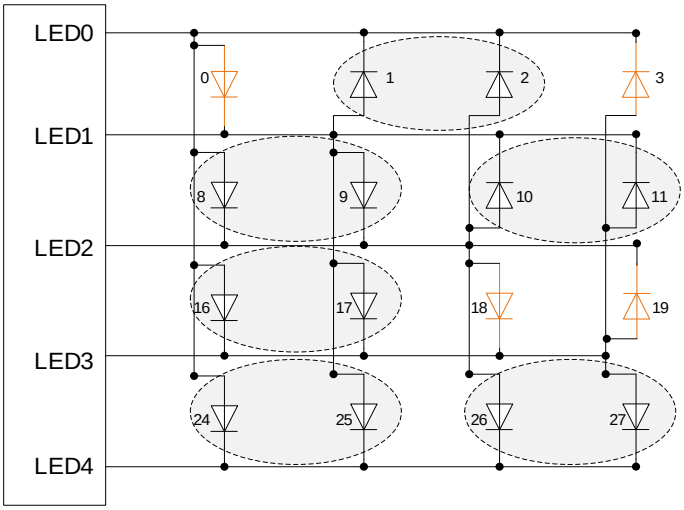
6*6点阵如下图所示：



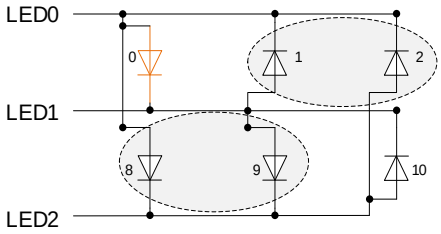
5*5点阵如下图所示：



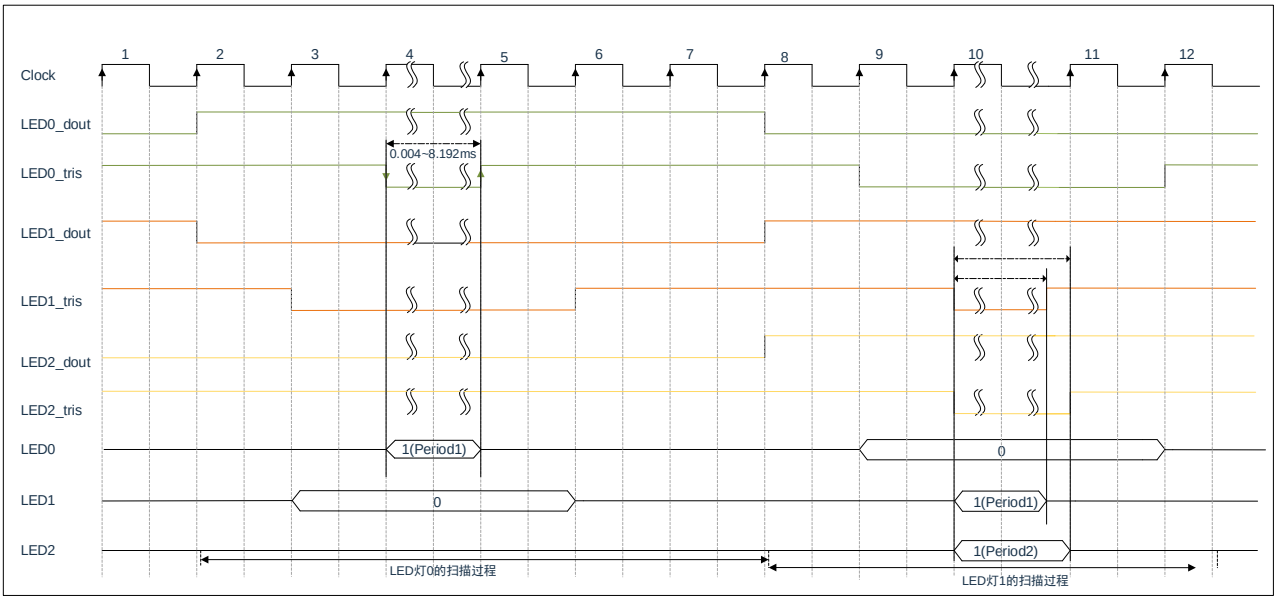
4*4点阵如下图所示：



以点亮灯 0、1、2 为例，灯 0、1 选择第一段周期，灯 2 选择第二段周期，且 SCAN1W[7:0]＜SCAN2W[7:0]，详细数字输出接口控制时序见下图：



两灯示意图



数字输出时序控制图

15.3.1 点阵 LED 周期选择

点阵 LED 每个灯有两段周期可以选择 Period1/ Period2：

- 第一段周期 Period1 = ({SCAN1W[7:0]}+1) * T_{LEDCLK}，SCAN1W 为第一段周期配置寄存器；
- 第二段周期 Period2 = ({SCAN2W[7:0]}+1) * T_{LEDCLK}，SCAN2W 为第二段周期配置寄存器；
- 周期选择寄存器为 LEDnSEL(n=0-7)，Led0-Led63 对应 8*8 点阵的 64 个 LED 坐标位置见下表，对应值为 1 的灯选择 Period2，为 0 的灯选择 Period1；

注：Period1/Period2 即是灯 Led0-Led63 对应的显示时长（如上图：LED0/1/2 输出 1 的时间）；

点阵驱动模式对应 LED 显示周期配置：

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LED0SEL0	Led7	Led6	Led5	Led4	Led3	Led2	Led1	Led0
LED1SEL1	Led15	Led14	Led13	Led12	Led11	Led10	Led9	Led8
LED2SEL2	Led23	Led22	Led21	Led20	Led19	Led18	Led17	Led16
LED3SEL3	Led31	Led30	Led29	Led28	Led27	Led26	Led25	Led24
LED4SEL4	Led39	Led38	Led37	Led36	Led35	Led34	Led33	Led32
LED5SEL5	Led47	Led46	Led45	Led44	Led43	Led42	Led41	Led40
LED6SEL6	Led55	Led54	Led53	Led52	Led51	Led50	Led49	Led48
LED7SEL7	Led63	Led62	Led61	Led60	Led59	Led58	Led57	Led56

15.4 相关寄存器

15.4.1 LED 驱动模式选择寄存器 LEDMODE

F769H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDMODE	LEDMODE7	LEDMODE6	LEDMODE5	LEDMODE4	LEDMODE3	LEDMODE2	LEDMODE1	LEDMODE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7-0 LEDMODE LED驱动模式选择寄存器；
0x55= LED矩阵驱动模式有效，相关寄存器生效；
0xaa= LED点阵驱动模式有效，相关寄存器生效；
其它值= 无效。

15.4.2 LED 点阵驱动控制器 LEDCON1

F765H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCON1	SCAN_START	DUTY2	DUTY1	DUTY0	SCAN_MODE	LED_FIXCLK_EN	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	W
复位值	0	0	0	0	0	0	0	0

Bit7 SCAN_START: LED点阵驱动扫描开启位；
1= LED扫描开启；（LED_FIXCLK_EN=1有效）
0= LED扫描停止。
Bit6~Bit4 DUTY<2:0>: LED点阵选择；
000= 无效；
001= 4*4点阵；
010= 5*5点阵；
011= 6*6点阵；
100= 6*7点阵；
101= 7*7点阵；
110= 7*8点阵；
111= 8*8点阵。
Bit3 SCAN_MODE: LED扫描模式配置；
1= LED点阵驱动循环扫描模式；
0= LED点阵驱动中断扫描模式。（当选择中断扫描模式，完成一次扫描，硬件清除SCAN_START位，中断标志位置1,软件重写SCAN_START为1后，重新进行扫描。）
Bit2 LED_FIXCLK_EN: LED运行时钟使能控制
1= 使能
0= 禁止
Bit1~Bit0 - 保留。

15.4.3 LED 时钟选择寄存器 LEDCKS

F766H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCKS	-	-	DTSEL1	DTSEL0	-	-	CLK1	CLK0
R/W	W	W	R/W	R/W	W	W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6 -- 保留

Bit5~Bit4 DTSEL<1:0>: 点阵LED死区时间选择

00: $12*T_{fix}$

01: $24*T_{fix}$

10: $48*T_{fix}$

11: $96*T_{fix}$

Bit3~Bit2 -- 保留

Bit1~Bit0 CLK<1:0>: LED计数时钟源周期（T_{LED_CLK}）(步进时间)。

00: $32*T_{fix}$

01: $64*T_{fix}$

10: $128*T_{fix}$

11: $256*T_{fix}$

15.4.4 LED 端口使能控制寄存器 LEDIOEN

F760H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDIOEN	IOEN7	IOEN6	IOEN5	IOEN4	IOEN3	IOEN2	IOEN1	IOEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 IOEN<7:0>: LED7-LED0端口使能控制位(LED8端口在LEDRESEQ寄存器中设置)；

1= 使能(需要与LED点阵模式选择的端口匹配才能生效)；

0= 禁止。

15.4.5 LED 点阵驱动第一段周期配置寄存器 SCAN1W

F761H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCAN1W	SCAN1W7	SCAN1W6	SCAN1W5	SCAN1W4	SCAN1W3	SCAN1W2	SCAN1W1	SCAN1W0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

（LEDMODE=0xaa,该寄存器生效。）

Bit7~Bit0 SCAN1W<7:0>: LED点阵驱动模式，第一段灯点亮周期配置寄存器。

Period1 = ({SCAN1W[7:0]}+1) * T_{LEDCLK}

15.4.6 LED 点阵驱动第二段周期配置寄存器 SCAN2W

F762H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCAN2W	SCAN2W7	SCAN2W6	SCAN2W5	SCAN2W4	SCAN2W3	SCAN2W2	SCAN2W1	SCAN2W0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

（LEDMODE=0xaa,该寄存器生效。）

Bit7~Bit0　　SCAN2W<7:0>:　LED点阵驱动模式，第二段灯点亮周期配置寄存器。

Period2 = ({SCAN2W [7:0]}+1) * T_{LEDCLK}

15.4.7 LED 点阵驱动显示数据寄存器 LEDnDATA (n=0-7)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDnDATA	LEDnDATA7	LEDnDATA6	LEDnDATA5	LEDnDATA4	LEDnDATA3	LEDnDATA2	LEDnDATA1	LEDnDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

LED0DATA 地址：F740H; LED1DATA 地址：F741H; LED2DATA 地址：F744H;　LED3DATA 地址：F745H;

LED4DATA 地址：F748H; LED5DATA 地址：F749H; LED6DATA 地址：F74CH;　LED7DATA 地址：F74DH;

Bit7~Bit0　　LEDnDATA<7:0>:　LED点阵驱动模式，点阵显示数据配置寄存器;
1=　LED点阵驱动模式，点阵上对应坐标的LED点亮;
0=　LED点阵驱动模式，点阵上对应坐标的LED不亮。

点阵驱动模式对应 LED 显示数据配置：

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LED0DATA	Led7	Led6	Led5	Led4	Led3	Led2	Led1	Led0
LED1DATA	Led15	Led14	Led13	Led12	Led11	Led10	Led9	Led8
LED2DATA	Led23	Led22	Led21	Led20	Led19	Led18	Led17	Led16
LED3DATA	Led31	Led30	Led29	Led28	Led27	Led26	Led25	Led24
LED4DATA	Led39	Led38	Led37	Led36	Led35	Led34	Led33	Led32
LED5DATA	Led47	Led46	Led45	Led44	Led43	Led42	Led41	Led40
LED6DATA	Led55	Led54	Led53	Led52	Led51	Led50	Led49	Led48
LED7DATA	Led63	Led62	Led61	Led60	Led59	Led58	Led57	Led56

注：Led0-Led63 对应 8*8 点阵的 64 个 LED 坐标位置。

15.4.8 LED 点阵驱动周期选择寄存器 LEDnSEL (n=0-7)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDnSEL	LEDnSEL7	LEDnSEL6	LEDnSEL5	LEDnSEL4	LEDnSEL3	LEDnSEL2	LEDnSEL1	LEDnSEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

LED0SEL 地址：F750H; LED1SEL 地址：F751H; LED2SEL 地址：F754H; LED3SEL 地址：F755H;

LED4SEL 地址：F758H; LED5SEL 地址：F759H; LED6SEL 地址：F75CH; LED7SEL 地址：F75DH;

Bit7~Bit0 LEDnSEL<7:0>: LED点阵驱动模式，点阵显示周期配置寄存器;
1= LED点阵驱动模式，点阵上对应坐标的LED显示周期选择第二段周期;
0= LED点阵驱动模式，点阵上对应坐标的LED显示周期选择第一段周期。

15.4.9 LED 点阵驱动管脚电流控制寄存器 LEDDRV

F76BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDDRV	--	--	--	--	SDRC3	SDRC2	SDRC1	SDRC0
R/W	W	W	W	W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit4 -- 保留，须均为0。
Bit3~Bit0 SDRC<3:0>: 点阵LED管脚拉电流驱动选择控制位；
相关管脚驱动使能才生效，否则电流默认为最大值。

0000=	0mA；	1000=	26.9mA；
0001=	4.9mA；	1001=	31.8mA；
0010=	7.4mA；	1010=	34.3mA；
0011=	12.2mA；	1011=	39.2mA；
0100=	14.7mA；	1100=	41.6mA；
0101=	19.5mA；	1101=	46.1mA；
0110=	22.1mA；	1110=	49mA；
0111=	26.9mA；	1111=	50mA。

15.4.10 管脚驱动使能寄存器 LEDSEGDR0

F76FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGDR0	-			LEDSEGDR0_4	LEDSEGDR0_3	LEDSEGDR0_2	LEDSEGDR0_1	LEDSEGDR0_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7:5 保留

Bit4 LEDSEGDR0 _4:P10管脚驱动使能位；

1= P10管脚驱动使能； P10管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P10管脚驱动使能禁止, P10管脚的拉电流驱动为默认值。

Bit3 LEDSEGDR0 _3:P07管脚驱动使能位；

1= P07管脚驱动使能； P07管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P07管脚驱动使能禁止,P07管脚的拉电流驱动为默认值。

Bit2 LEDSEGDR0_2: P06管脚驱动使能位；

1= P06管脚驱动使能；P06管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P06管脚驱动使能禁止,P06管脚的拉电流驱动为默认值。

Bit1 LEDSEGDR0 _1:P05管脚驱动使能位；

1= P05管脚驱动使能；P05管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P05管脚驱动使能禁止,P05管脚的拉电流驱动为默认值。

Bit0 LEDSEGDR0 _0:P04管脚驱动使能位；

1= P04管脚驱动使能；P04管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P04管脚驱动使能禁止,P04管脚的拉电流驱动为默认值。

15.4.11 管脚驱动使能寄存器 LEDSEGDR3

F772H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGDR 3	--	LEDSEGDR 3 _6	LEDSEGDR 3 _5	LEDSEGDR 3 _4	LEDSEGDR 3_3	-	-	-
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 -- 保留，须为0

Bit6 LEDSEGDR3 _6:P03管脚驱动使能位；

1= P03管脚驱动使能；P03管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P03管脚驱动使能禁止,P03管脚的拉电流驱动为默认值。

Bit5 LEDSEGDR3 _5:P02管脚驱动使能位；

1= P02管脚驱动使能；P02管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P02管脚驱动使能禁止,P02管脚的拉电流驱动为默认值。

Bit4 LEDSEGDR3 _4:P01管脚驱动使能位；

1= P01管脚驱动使能；P01管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P01管脚驱动使能禁止,P01管脚的拉电流驱动为默认值。

Bit3 LEDSEGDR3 _3:P00管脚驱动使能位；

1= P00管脚驱动使能；P00管脚的拉电流驱动由LEDDRV[3:0]寄存器配置；

0= P00管脚驱动使能禁止,P00管脚的拉电流驱动为默认值。

Bit2:0 -- 保留，须为0

15.4.12 LED 点阵引脚映射寄存器 LEDRESEQ

F76CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDRESEQ	-	-	-	-	LED8_EN	LREMAP2	LREMAP1	LREMAP0
R/W	W	W	W	W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit4	-	保留，必须为0；
Bit3	LED8_EN	LED8端口使能位
	1=	允许LED8端口功能（需要LED点阵模式选择为8X8才能生效）；
	0=	LED8端口为GPIO或其他功能
Bit2~Bit0	LEDSEQ<2:0>:	LED 点阵起始端口选择；
	000=	LED0 开始，LED0-LED1-……-LED6- LED7
	001=	LED1 开始，LED1-LED2-……-LED7- LED0
	…	…
	111=	LED7 开始，LED7-LED0-……-LED5- LED6

15.5 LED 点阵驱动中断

15.5.1 LED 点阵驱动状态寄存器 LEDSTATUS

F76AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSTATUS	--	--	--	--	--	--	LEDIE	LEDIF
R/W	W	W	W	W	W	W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit2 -- 保留，须为0。
Bit1 LEDIE: LED点阵驱动模式中断使能位；
1= LED点阵驱动模式中断使能；
0= LED点阵驱动模式中断禁止。
Bit0 LEDIF: LED点阵驱动模式中断标志位
1= LED点阵驱动扫描完成；
0= 软件清0。

15.5.2 中断优先级控制寄存器 EIP3

0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	PXTDET	--	--	PTOUCH	PLVD	PLED	--	--
R/W	R/W	W	W	R/W	R/W	R/W	W	W
复位值	0	0	0	0	0	0	0	0

Bit7 PXTDET 晶振停振中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit6-Bit5 -- 保留，须均为0。
Bit4 PTOUCH TOUCH中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit3 PLVD: LVD中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit2 PLED: LED中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit1-Bit0 -- 保留，须均为0。

16. SPI 模块

16.1 概述

此 SPI 是一个完全可配置的 SPI 主机/从机设备，允许用户配置串行时钟信号 SCLK 的极性和相位。串行时钟线（SCLK）与两个独立串行数据线上信息的移位和采样同步，SPI 数据同时发送和接收。SPI 允许 MCU 与串行外围设备进行通信，它还能够在多主机系统中进行处理器间通信，是一种可以在各种工艺技术中实现的技术独立的设计。

SPI 系统具有足够的灵活性，可以与多家制造商的许多标准产品外设直接连接。为了适应大多数可用的同步串行外设，时钟控制逻辑允许选择时钟极性和相位。系统可以配置为主机设备或从机设备，当 SPI 被配置为主机设备时，软件为串行时钟在八种不同的比特率中选择其一，速率高达系统时钟除以 4（Fsys/4）。

SPI 从机片选用以寻址 SPI 从机设备来交换串行数据。当 SPI 作主机设备时，SPI 自动驱动由从机选择控制寄存器 SSCR 选择。SPI 控制器包含逻辑错误检测以支持处理器间通信，如写入冲突检测器能指示何时在传送过程中将数据写入串行移位寄存器。

16.2 特性

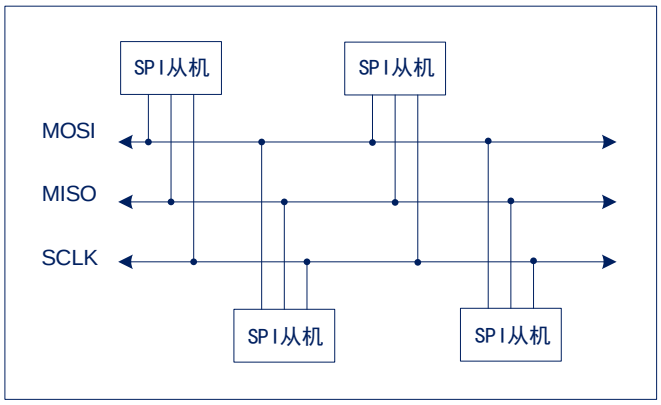
- 全双工同步串行数据传输。
- 支持主机/从机模式。
- 支持多主机系统。
- 系统错误检测。
- 产生中断。
- 支持速度高达系统时钟的 1/4（F_{SYS}≤24MHz）。
- 比特率产生系统时钟的 1/4、1/8、1/16、1/32、1/64、1/128、1/256、1/512。
- 支持四种传输格式。
- 简单接口允许轻松的连接到微控制器。

16.3 SPI 端口配置

使用 SPI 功能需要将相关端口配置成 SPI 通道，且通过通信输入端口寄存器选择对应端口输入。例如配置 P00、P01、P02、P03 为 SPI 通信口。配置代码如下：

```
PS_NSS0OI= 0x00;        //选择 P03 配置成 NSS 通道
PS_SCLKOI= 0x00;        //选择 P00 配置成 SCLK 通道
PS_MOSI = 0x00;       //选择 P01 配置成 MOSI 通道
PS_MISO = 0x00;       //选择 P02 配置成 MISO 通道
```

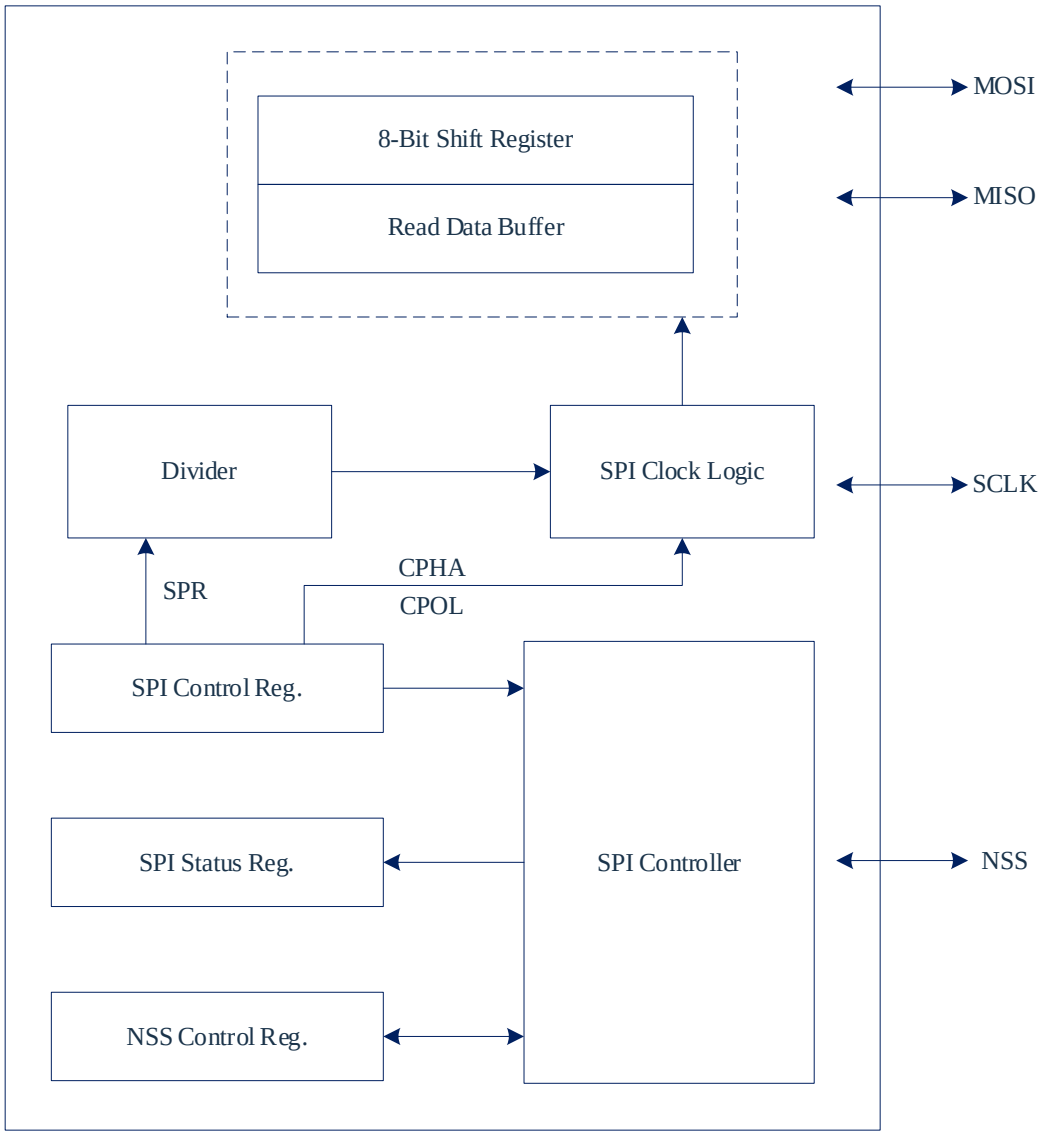
配置为 SCLK、MOSI、MISO 和 NSS 口，其上拉电阻与开漏输出的开关强制关闭。多机 SPI 通讯结构示意图如下图所示：



16.4 SPI 硬件描述

发生 SPI 传输时，当一个数据引脚移出一个 8 位字符的同时，另一个数据引脚上移入其他的 8 位字符。主机设备中的 8 位移位寄存器和从机设备中的另一个 8 位移位寄存器作为循环 16 位移位寄存器连接，当发生转移时，该分布式移位寄存器被移位 8 位，从而有效的交换了主机从机的字符。

SPI 系统中的中心元件是包含移位寄存器和读取数据缓冲区的模块。系统在发送方向上为单缓冲器，在接收方向上为双缓冲器。这意味着直到前一个数据传输完，新的数据才能写入移位器；然而，接收的数据被传送到并行读取数据缓冲器中，所以移位器可以自由地接收第二串行字符。只要在下一个串行字符准备好传输之前，第一个字符从读取数据缓存器中读出，就不会出现覆盖情况。SPI 控制框图如下图所示：



与 SPI 相关联的引脚有：NSS，SCLK，MOSI，MISO。

主机模式下的 NSS 输出引脚用于选择从机设备，从机模式下的 NSS 输入引脚用于使能传输。

在主机模式下，SCLK 引脚用作 SPI 时钟信号参考。当主机设备启动传输时，SCLK 引脚上会自动生成八个时钟周期。

当 SPI 被配置为从机设备时，SI 引脚是从机设备输入数据线，SO 是从机设备输出数据线。

当 SPI 被配置为主机设备时，MI 引脚是主机设备输入数据线，MO 是主机设备输出数据线。

16.5 SPI 相关寄存器

16.5.1 SPI 控制寄存器 SPCR

0xEC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR	--	SPEN	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	1	0	0

Bit7 -- 保留，须为0。
Bit6 SPEN: SPI模块使能位；
1= 使能；
0= 禁止。
Bit5 SPR2: SPI时钟频率选择位第[2]位。
Bit4 MSTR: SPI模式选择位；
1= 主动模式；
0= 从动模式。
Bit3 CPOL: SPI时钟极性选择位；
1= SCLK空闲时为高；
0= SCLK空闲时为低。
Bit2 CPHA: SPI时钟相位选择位。
Bit1~Bit0 SPR<1:0>: SPI时钟频率选择位[1:0]
（频率控制详见下面表格）

SPR2-SPR0 控制 SPI 时钟分频

SPR2	SPR1	SPR0	系统时钟分频
0	0	0	4
0	0	1	8
0	1	0	16
0	1	1	32
1	0	0	64
1	0	1	128
1	1	0	256
1	1	1	512

16.5.2 SPI 数据寄存器 SPDR

0xEE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPDR	SPIDATA7	SPIDATA6	SPIDATA5	SPIDATA4	SPIDATA3	SPIDATA2	SPIDATA1	SPIDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 SPIDATA: SPI发送或接收的数据。
写操作: 写将要发送的数据（发送顺序由高位依次到低位）。
读操作: 已经接收到的数据。

16.5.3 SPI 从器件选择控制寄存器 SSCR

从器件选择控制寄存器 SSCR 可以随时读取或写入，它用于配置在确认 SPI 主机传输时应驱动哪个从机选择输出。当 SPI 主机传输启动时，SSCR 寄存器的内容将自动分配给 NSS 引脚。

0xEF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSCR	--	--	--	--	--	--	--	NSSO0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit1 -- 保留，须均为1。
Bit0 NSSO0: SPI从器件选择控制位（主机片选输出NSS为NSSO0）。
0= SPI主机传输启动时，NSSO0输出0。
1= SPI主机传输启动时，NSSO0输出1。

16.5.4 SPI 状态寄存器 SPSR

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	SPISIF	WCOL	--	--	--	--	--	SSCEN
R/W	R	R	--	R	--	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 SPISIF: SPI传输完毕中断标志位，只读；
1= SPI传输完毕（先读SPSR，再读/写SPDR后清零）；
0= SPI未传输完毕。
Bit6 WCOL: SPI写冲突中断标志位，只读；
1= SPI传输未完成时产生写SPDR操作冲突（先读SPSR，再读/写SPDR后清零）；
0= 无写冲突。
Bit5~Bit1 -- 保留，须均为0。
Bit0 SSCEN: SPI主控模式NSS输出控制位。
1= SPI处于空闲状态时，NSS输出高电平；
0= NSS输出寄存器SSCR的内容。

SPI 状态寄存器（SPSR）包含指示传输完成或系统错误发生的标志。当相应的事件发生并通过软件按顺序清除时，所有标志都将自动设置。通过读取 SPSR，然后访问 SPDR，SPISIF 和 WCOL 将会自动清除。

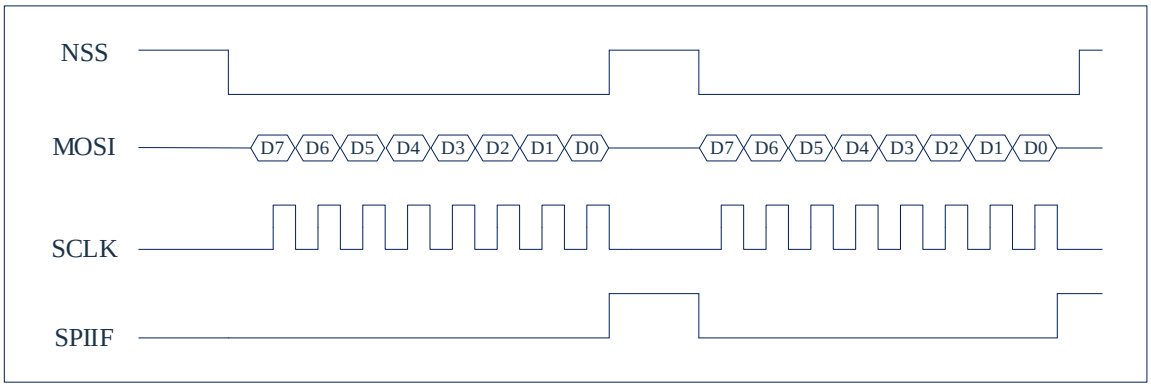
SSCEN 位是自动从机选择输出的使能位。当 SSCEN 置 1，传输正在进行时 NSS 线输出 SSCR 寄存器的内容，传输空闲时 NSS 为高电平。当 SSCEN 位清零时，NSS 线总是显示 SSCR 寄存器的内容。

16.6 SPI 主控模式

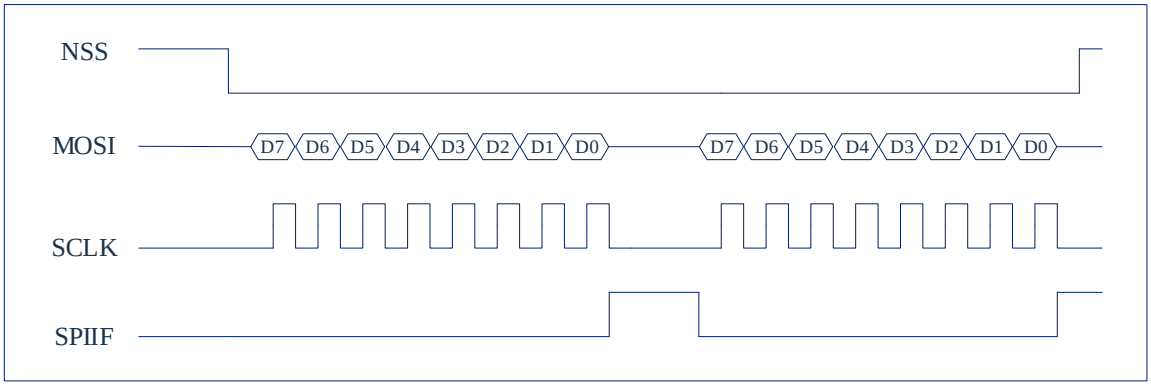
当 SPI 配置为主机模式时，通过写入 SPDR 寄存器启动传输。当新字节写入 SPDR 寄存器时，SPI 开始传输。串行时钟 SCLK 由 SPI 生成，在主机模式下 SPI 使能，SCLK 输出。

主机模式下的 SPI 可以通过 NSS 线选择 SPI 从机设备。NSS 线-从机选择输出线加载了 SSCR 寄存器的内容。SPSR 寄存器的 SSCEN 位在自动 NSS 线路控制和软件控制之间进行选择。将 SSCEN 位置于主机模式下，当 SSCEN 置 1，传输正在进行时 NSS 线输出 SSCR 寄存器的内容，传输空闲时 NSS 为高电平。当 SSCEN 位清零时，NSS 线由软件控制，并且始终显示 SSCR 寄存器的内容，无论传输正在进行还是 SPI 处于空闲状态。

当 SSCEN=1 时，配置 SPI 的时钟极性 CPOL=0、时钟相位 CPHA=0，从机选择线使用如下图所示：



当 SSCEN=0 时，配置 SPI 的时钟极性 CPOL=0、时钟相位 CPHA=0，从机选择线使用如下图所示：



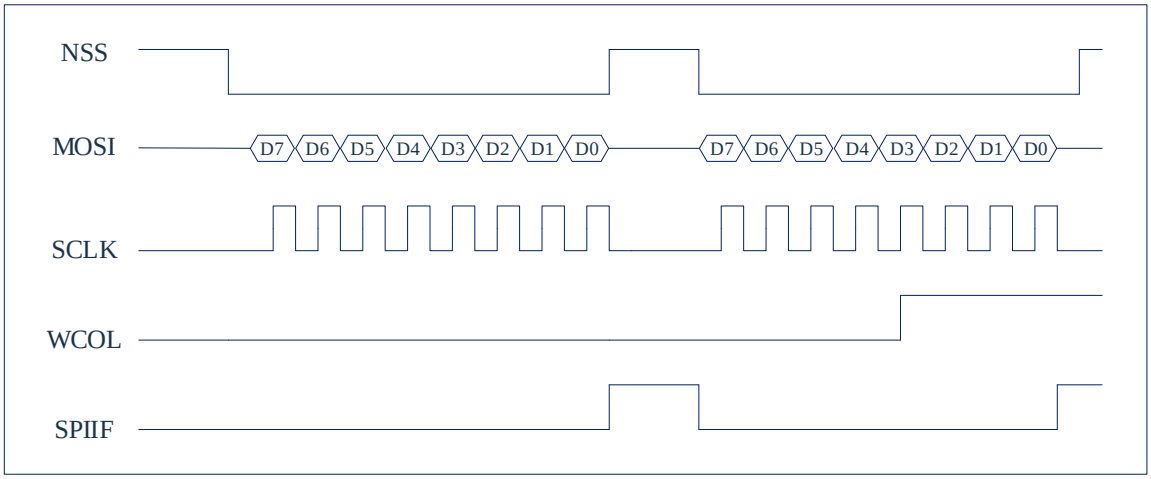
16.6.1 写冲突错误

如果在传输过程中写入了 SPI 数据寄存器，则会发生写冲突。传输继续不受干扰，导致错误的写入数据不会写入移位器。写冲突由 SPSR 寄存器中的 WCOL 标志指示。

当 WCOL 错误发生时，WCOL 标志由硬件自动置 1。要清除 WCOL 位，用户应执行以下步骤：

- 读取 SPSR 寄存器的内容；
- 访问 SPDR 寄存器（读或写）。

SPI 主控模式下，配置 SPI 的时钟极性 CPOL=0、时钟相位 CPHA=0 时写冲突错误如下图所示：



写冲突产生的具体条件为：在数据传输过程中，NSS 为低时，第一个数据开始发送时刻到第 8 个 SCLK 下降沿期间，如果在此期间写 SPDR，则会出现写冲突，WCOL 将置 1。

注意：开始发送数据时，在写 SPDR 之后，NSS 并不是立即变为低电平，需要等待最多一个 SPI 时钟后才开始为低。NSS 为低后，需要等待一个系统时钟才开始发送第一个数据，此时才进入真正的数据传输状态。在写 SPDR 到进入真正的数据传输状态期间，再次写入 SPDR 并不会产生写冲突。但该操作会更新准备发送的数据。如果有多次写 SPDR 的操作，发送的数据将会是最后一次写入 SPDR 的值。

由于 SPI 只有一个发送缓冲器，建议在写 SPDR 之前判断上一次的数据是否发送完毕，确定发送完成后再写 SPDR 寄存器，以防止产生写冲突。

16.7 SPI 从动模式

当配置为 SPI 从机设备时，SPI 传输由外部 SPI 主机模块通过使用 SPI 从机选择输入启动，并生成 SCLK 串行时钟。

在传输开始之前，需确定哪个 SPI 从机将用于交换数据。NSS 被使用（清零= 0），连接到 SCLK 线的时钟信号将使 SPI 从机设备转移到 MOSI 线的接收移位寄存器内容，并用发送器移位寄存器的内容驱动 MISO 线。当所有 8 位被移入/移出时，SPI 通过设置 IRQ 输出来产生中断请求。移位寄存器的内容驱动 MISO 线。

在 SPI 从机模式下，只能有一个传输错误-写冲突错误。

16.7.1 被寻址错误

在从机模式下，只有写冲突错误能被 SPI 检测到。

当 SPI 传输进行时执行 SPDR 寄存器写操作，会发生写冲突错误。

在从机模式下，当 CPHA 清零时，只要 NSS 从机选择线被驱动为低电平，即使已经传输了所有位，也可能会发生写冲突错误。这是因为没有明确指定传输开始，并且在全字节传输后 NSS 被驱动为低电平可能指示下一个字节传输的开始。

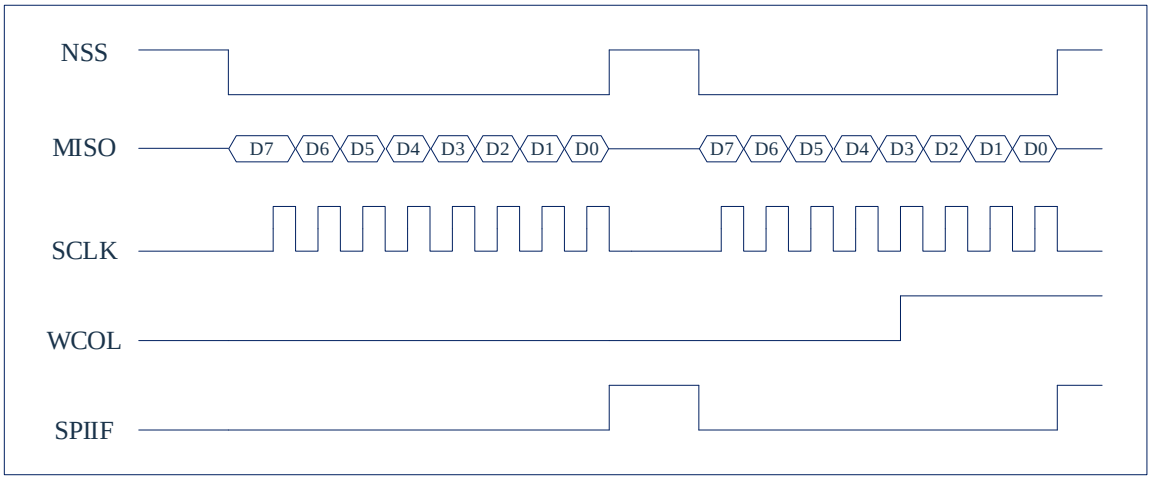
16.7.2 写冲突错误

如果在传输过程中写入 SPI 数据寄存器，则会发生写冲突。传输继续不受干扰，并且导致错误的写入数据不会写入移位器。写冲突由 SPSR 寄存器中的 WCOL 标志指示。

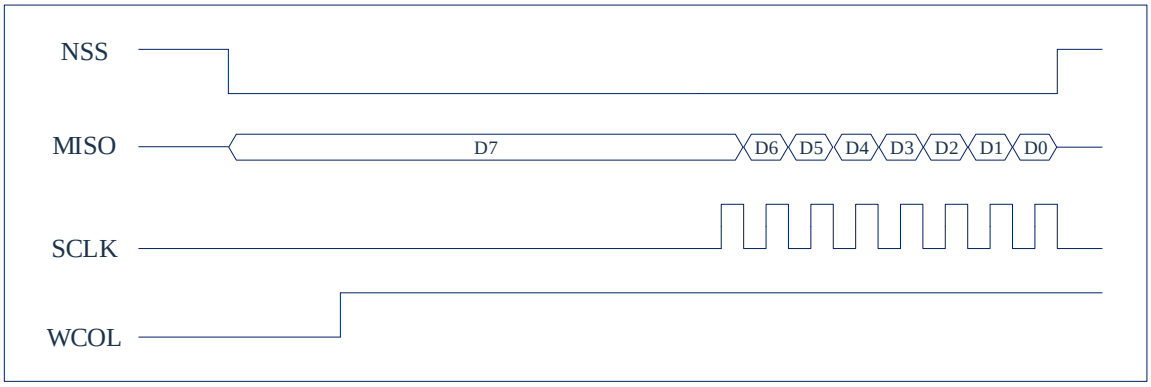
当 WCOL 错误发生时，WCOL 标志由硬件自动置 1。要清除 WCOL 位，用户应该执行以下顺序：

- 读取 SPSR 寄存器的内容；
- 访问 SPDR 寄存器（读或写）。

SPI 从动模式下传输过程中写冲突错误如下图所示：



以防 CPHA 被清除，WCOL 生成也可由任一 NSS 线清零时 SPDR 寄存器写入引起，此时 SPI 主机没有生成串行时钟 SCLK 也可以完成。这是因为没有明确指定传输开始，并且在全字节传输后 NSS 被驱动为低电平可能指示下一个字节传输的开始。当 NSS 传输线为低电平、时钟相位 CPHA = 0 时，写 SPDR 导致写冲突错误如下图所示：



另外，从动模式下在写 SPDR 之后，主机控制的 NSS 并不是立即变为低电平。当 NSS 为低后，需要等待 SCLK 第二个边沿才开始才进入真正的数据传输状态。

在写 SPDR 到开始发送第一个数据期间，再次写入 SPDR 并不会产生写冲突。但该操作会更新准备发送的数据。如果有多次写 SPDR 的操作，发送的数据将会是最后一次写入 SPDR 的值。

在开始发送第一个数据到 SCLK 第二个边沿期间，再次写入 SPDR 并不会产生写冲突，也不会更新正在发送的数据。即忽略该次写 SPDR 的操作。

由于 SPI 只有一个发送缓冲器，建议在写 SPDR 之前判断上一次的数据是否发送完毕，确定发送完成后再写 SPDR 寄存器，以防止产生写冲突。

16.8 SPI 时钟控制逻辑

16.8.1 SPI 时钟相位与极性控制

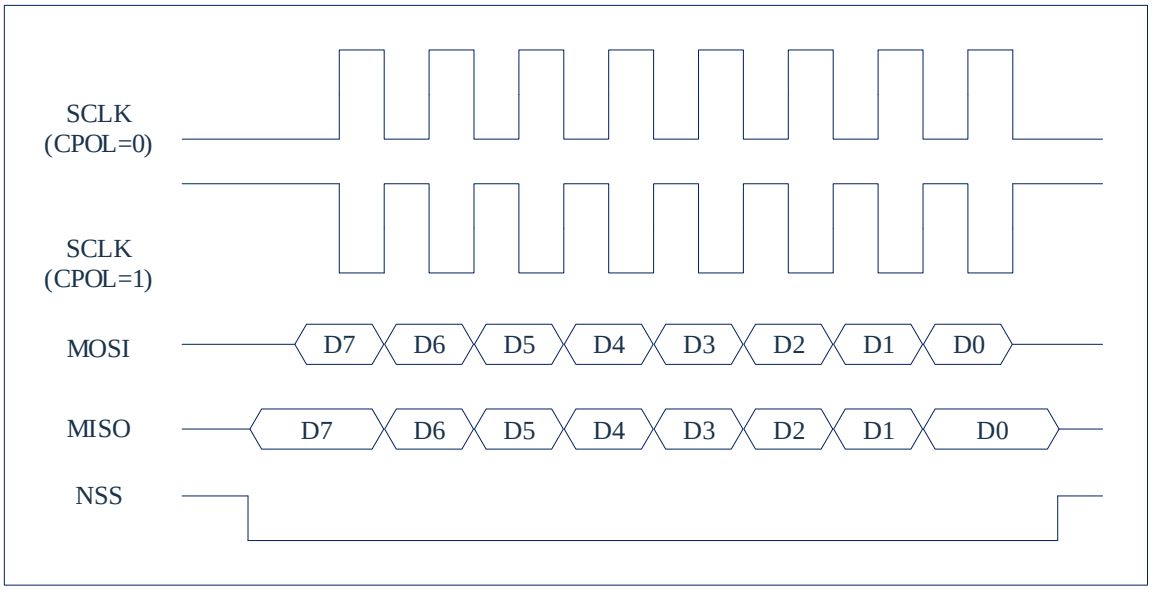
软件可以在 SPI 控制寄存器（SPCR）中选择使用两个控制位的（串行时钟 SCLK 的相位和极性）四种组合中的任何一种。时钟极性由 CPOL 控制位指定，在传输空闲时 CPOL 控制位选择高电平或低电平对传输格式没有显著影响。时钟相位（CPHA）控制位选择两种基本不同的传输格式中的一种。主机 SPI 设备和通信从机设备的时钟相位和极性应相同。在某些情况下，传输期间改变相位和极性，以允许主机设备与具有不同要求的外设从机进行通信。SPI 系统的灵活性允许与几乎所有现有的同步串行外设直接连接。

16.8.2 SPI 传输格式

在 SPI 传输期间，数据同时发送（串行移出）并接收（串行移入）。串行时钟线与两条串行数据线移位和采样同步。从机选择线允许单独选择从机 SPI 设备；未选择的从机设备不会干扰 SPI 总线活动。在 SPI 主机设备上，从机选择线可以有选择性的用于指示多主机总线竞争。

16.8.3 CPHA=0 传输格式

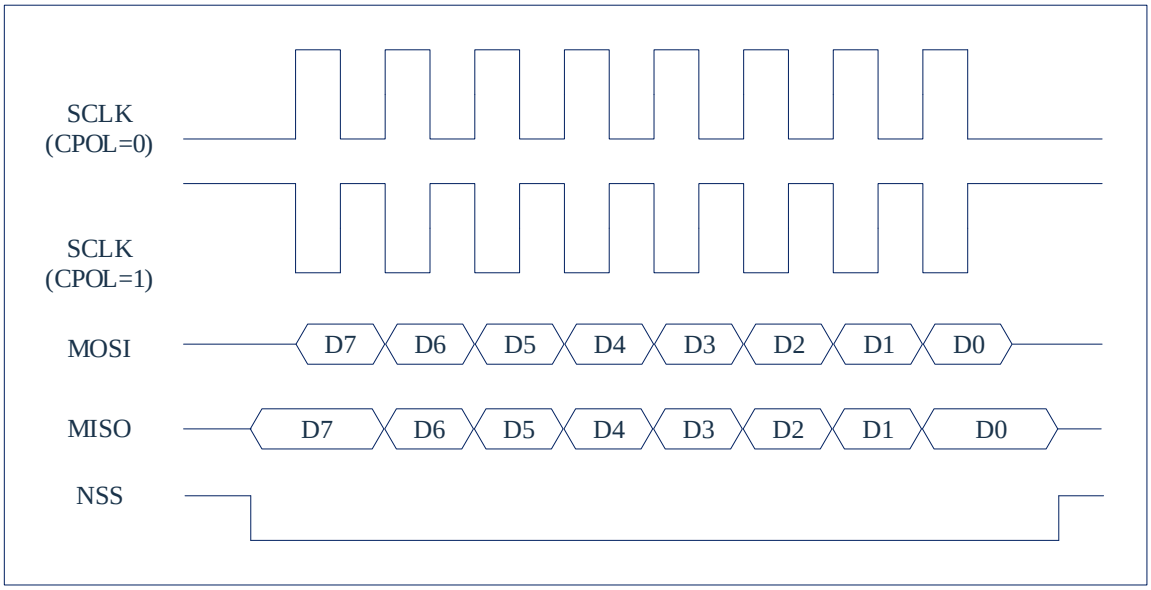
下图显示了 CPHA 为 0 的 SPI 传输的时序图。SCLK 显示两个波形：一个用于 CPOL 等于 0，另一个用于 CPOL 等于 1。通过 SCLK 该图可以描述为主机设备或从机设备时序图，主进/从出（MISO）和主出/从进（MOSI）引脚直接连接在主机和从机之间。MISO 信号是从机输出，MOSI 信号是主机输出。NSS 线是从机的从机选择输入；主机的 NSS 引脚未显示，但假定为无效。主机的 NSS 引脚必须为高电平。该时序图在功能上描述了如何进行传输；不应将其用作数据表参数信息的替代品。



当 CPHA=0 时，NSS 线必须在每个连续的串行字节之间解除置 1 并重新置 1。此外，如果当 NSS 处于低电平，从机将数据写入 SPI 数据寄存器（SPDR），则会产生写冲突错误。当 CPHA = 1 时，NSS 线可能在连续传输之间保持低电平（可以始终保持低电平）。在具有单个固定主机和驱动 MISO 数据线的单个从机的系统中，这种格式有时是优先选择的。

16.8.4 CPHA=1 传输格式

下图是 CPHA＝1 的 SPI 传输的时序图。SCLK 显示两个波形：一个用于 CPOL=0，另一个用于 CPOL=1。由于 SCLK、MISO 和 MOSI 引脚直接连接在主机和从机之间，所以该图可以解释为主机或从机时序图。MISO 信号是从机输出，MOSI 信号是主机输出。NSS 线是从机的从机选择输入；主机的 NSS 引脚未显示，但假定为无效。主机的 NSS 引脚必须为高电平，或必须重新配置为不影响 SPI 的通用输出。



16.9 SPI 数据传输

16.9.1 SPI 传输启动

所有 SPI 传输都由主 SPI 设备启动和控制。作为从机设备，SPI 根据选择的 CPHA 格式，将考虑传输开始于第一个 SCLK 边沿或 NSS 的下降沿。当 CPHA＝0 时，NSS 的下降沿表示传输的开始。当 CPHA＝1 时，SCLK 上的第一个边沿表示传输的开始。无论哪种 CPHA 模式，通过使 NSS 线为高电平可以中止传输，但会使 SPI 从机逻辑和计数器复位。选择的 SCLK 速率对从机操作没有影响，因为主机的时钟正在控制传输。

当 SPI 配置为主机时，通过写入 SPDR 的软件启动传输。

16.9.2 SPI 传输结束

当 SPIF 标志置 1 时，SPI 传输在技术上完成，但是根据 SPI 系统的配置，可能还有其他任务。由于 SPI 比特率不影响结束期的时间，因此在结束期间的讨论中只考虑最快的速率。当 SPI 被配置为主机时，SPIF 在第八个 SCLK 周期循环结束时置位。当 CPHA 等于 1 时，SCLK 在第八个 SCLK 周期的最后一半处于不活动状态。

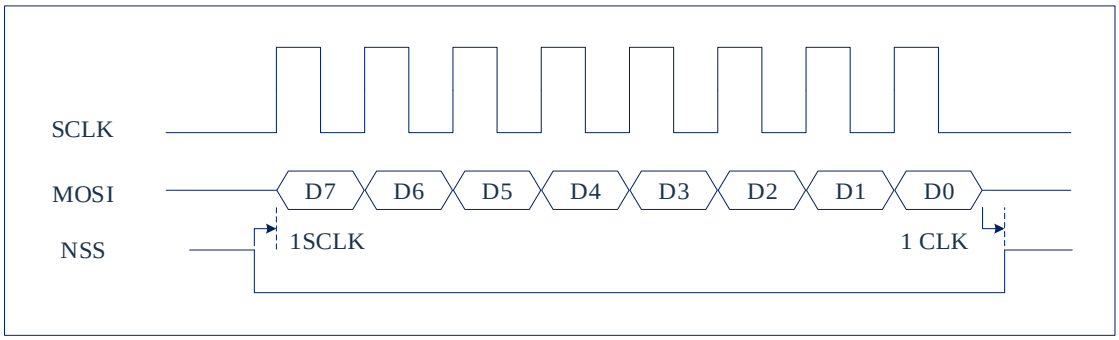
因为 SCLK 线可以与从机的 MCU 时钟异步，并且从机不能像访问 SCLK 周期那样访问主机尽可能多的信息，所以当 SPI 作为从机运行时，结束周期是不同的。例如，当 CPHA = 1 时，其中最后一个 SCLK 边沿在第八个 SCLK 周期的中间发生，从机无法知道上一个 SCLK 周期是何时结束。由于这些原因，从机认为在串行数据的最后一位被采样之后，传输完成，这对应于第八个 SCLK 周期的中间。

SPIF 标志设置在传输结束时，但是 NSS 线仍然为低电平时，从机不允许将新数据写入 SPDR。

16.10 SPI 时序图

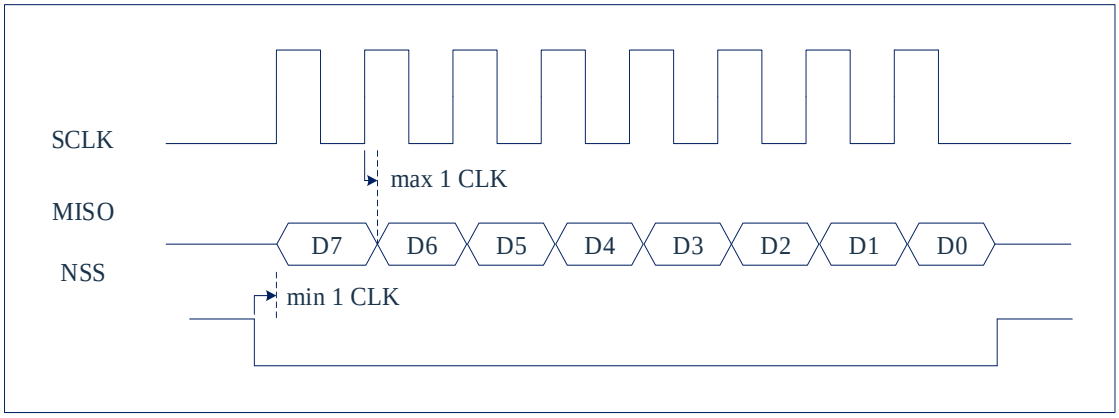
16.10.1 主控模式传输

当 SPI 的时钟极性 CPOL=0、时钟相位 CPHA=1 时，SPI 主控模式下 NSS 为低电平后一个系统时钟 CLK，MOSI 开始输出，MOSI 的数据在 SCLK 时钟的上升沿输出。主控模式时序图如下图所示：



16.10.2 从动模式传输

当 SPI 的时钟极性 CPOL=0、时钟相位 CPHA=1 时，MISO 上的数据在 NSS 线的下降沿之后开始输出。MSIO 数据输出与 NSS 的下降沿最大相差 1 个系统时钟 CLK。从动模式时序图如下图所示：



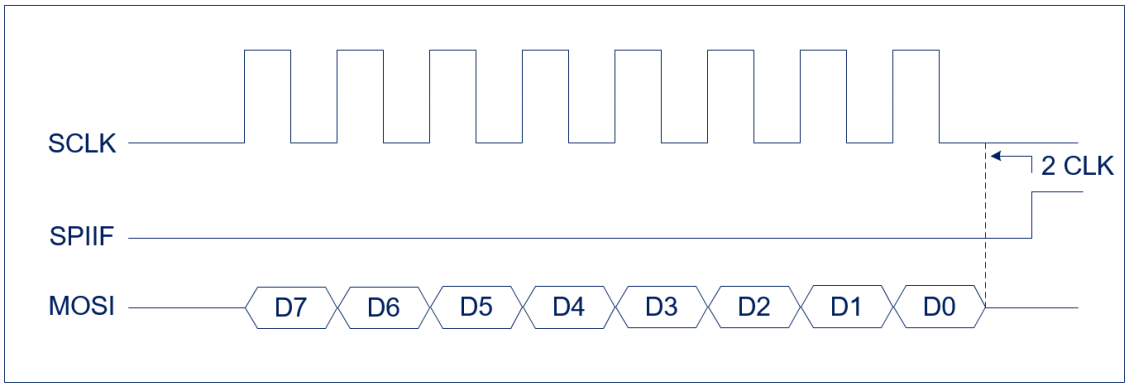
16.11 SPI 中断

SPI 的中断号为 22，其中断向量为 0x00B3。使能 SPI 中断必须将其使能位 SPIIE 置 1，且将总中断使能位 EA 置 1。

若 SPI 相关的中断使能均打开，SPI 总中断指示位 SPIIF=1 时，CPU 将进入中断服务程序。SPIIF 操作属性为只读，且与 SPIIE 的状态无关。

SPI 状态寄存器 SPSR 中传输完成标志 SPISIF、写冲突 WCOL 任意一个标志为 1 后，SPI 总中断指示位 SPIIF 将会置 1。仅当这 3 个标志位均为 0 时，SPIIF 自动清 0。

当 SPI 的时钟极性 CPOL=0、时钟相位 CPHA=1 时，SPI 主控模式下 SPIIF 在每帧数据的第 8 个 SCLK 时钟上升沿之后的 2 个 SCLK 时钟产生，时序图如下图所示：



16.11.1 中断屏蔽寄存器 EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	--	ADCIE	PWMIE	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI中断使能位；
1= 允许SPI中断；
0= 禁止SPI中断。
Bit6 I2CIE: I^2C中断使能位；
1= 允许I^2C中断；
0= 禁止I^2C中断。
Bit5 -- 保留，须为0。
Bit4 ADCIE ADC中断使能位；
1= 允许ADC中断；
0= 禁止ADC中断。
Bit3 PWMIE: PWM总中断使能位；
1= 允许PWM所有中断；
0= 禁止PWM所有中断。
Bit2~Bit0 -- 保留，须为0。

16.11.2 中断优先级控制寄存器 EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	--	PADC	PPWM	PT5	--	--
R/W	R/W	R/W	--	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit6 PI2C: I^2C中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit5 -- 保留，须为0。
Bit4 PADC: ADC中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit3 PPWM: PWM中断优先级控制位
1= 设置为高级中断；
0= 设置为低级中断。
Bit2 PT5: TIMER5中断优先级控制位
1= 设置为高级中断；
0= 设置为低级中断。
Bit1~Bit0 -- 保留，须为0。

16.11.3 外设中断标志位寄存器 EIF2

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	TF5	--	--
R/W	R	R	R	R/W	R	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI总中断指示位，只读；
1= SPI产生中断，（清除具体的中断标志位后，此位自动清除）；
0= SPI未产生中断。
Bit6 I2CIF: I^2C总中断指示位，只读；
1= I^2C产生中断，（清除具体的中断标志位后，此位自动清除）；
0= I^2C未产生中断。
Bit5 -- 保留，须为0。
Bit4 ADCIF: ADC中断标志位；
1= ADC转换完成，需软件清零；
0= ADC转换未完成。
Bit3 PWMIF: PWM总中断指示位，只读；
1= PWM产生中断，（清除具体的中断标志位后，此位自动清除）；
0= PWM未产生中断。
Bit2 TF5: Timer5定时器溢出中断标志位；
1= Timer5定时器溢出，需软件清零；
0= Timer5定时器无溢出。
Bit1~Bit0 -- 保留，须为0。

17. I²C 串行接口控制器（I²C）

17.1 概述

I²C 是一种两线双向串行总线，为设备之间的数据交换提供了一种简单有效的连接方式。I²C 是一个真正的多主机总线，包含了冲突检测和仲裁机制。冲突检测和仲裁机制用来在两个或多个主机同时尝试控制总线的情况下，防止数据损坏。

17.2 特性

- 支持主机/从机模式。
- 主从机之间双向数据传送。
- 多主机总线。
- 多主机间同时传输数据仲裁，避免总线上串行数据损坏。
- 总线采用串行同步时钟，可实现设备之间以不同的速率传输。
- 可编程的时钟可以用于多种速率控制。
- 支持 7 位/10 位从地址模式。
- 支持两种传输速度模式
  - 标准（高达 100Kb/s）；
  - 快速（高达 400Kb/s）；

17.3 寄存器映射

RO：只读；WO：只写；R/W：读写。

寄存器	地址	读/写	描述	复位值
I2CCON	0XF1	R/W	I²C控制寄存器	0x00
I2CCLR	0XF2	WO	I²C清零寄存器	0x00
I2CSTAT	0XF3	RO	I²C状态寄存器	0xF8
I2CDAT	0XF4	R/W	I²C数据寄存器	0x00
I2CCLK	0XF5	R/W	I²C时钟控制寄存器	0x00
I2CADR	0XF6	R/W	I²C从机地址寄存器	0x00
I2CXAR	0XF7	R/W	I²C扩展地址复位寄存器	0x00

17.4 I2C 传输速度模式说明

I2C 传输速度计算公式为：$SCL时钟 = \frac{PCLK}{(2^M \times (N+1) \times 10)}$ (M，N 见 I2C 寄存器 CLK)。

I2C 传输速度配置示例表

系统时钟 / 传输速度模式	48MHz		24MHz		16MHz		8MHz	
	M	N	M	N	M	N	M	N
标准模式(100Kb/s)	4	2	-	-	4	0	-	-
	3	5	3	2	3	1	3	0
	2	11	2	5	2	3	2	1
	-	-	1	11	1	7	1	3
	-	-	-	-	0	15	0	7
快速模式(400Kb/s)	2	2	-	-	2	0	-	-
	1	5	1	2	1	1	1	0
	0	11	0	5	0	3	0	1

注：1）做从机时，通信速率由主机设置决定。
2）快速模式时，建议外部上拉电阻≤4.7KΩ。

17.5 寄存器说明

17.5.1 I²C 控制寄存器（I2CCON）

0XF1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCON	I2CIE	I2CEN	STA	STO	SI	AA	XADRF	ADRF
R/W	R/W	R/W	R/W	R/W	R	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 I2CIE: 中断使能位；
0= 禁止；
1= 使能。

Bit6 I2CEN: I2C 接口使能位；
0= 禁止 I2C 接口；
1= 使能 I2C 接口（默认为从机模式）。

Bit5 STA: 启动标志位；
1= I2C 进入主机模式并发送启动信号；
- 当I2C已经处于主机模式，则发送重启动信号。
- 当I2C处于从机模式时，写1会结束当前传输并等待总线空闲时进入主机模式。

0= 不影响。
当启动位或重启动位发送完成时，该位自动清零。

Bit4 STO: 停止标志位；
在主机模式下写1时，会发送一个停止位。
在从机模式下写1时，I2C模块会当作接受到一个停止位
停止位
- 当同时将 STA 和 STO 置位时，I2C 模块会先发送一个停止位，接着发送一个启动位。
- 当停止位发送完成时，该位自动清零。

Bit3 SI: I2C中断标志位，只读；
当I2C发生总线状态改变时该位置位，可通过在SIC位写1清零。

Bit2 AA: 应答标志位；
0= 没有接收到 ACK 信号；
1= 在以下情况回复 ACK 信号。
- 从机地址匹配时
- 使能广播呼叫且接收到广播地址时

在主机或从机模式下接收到数据时可通过在 AAC 位写 1 清零该位

Bit1 XADRF: I²C 从机10位地址标志位，只读；
0= I²C地址不匹配；
1= 1²C 10位地址匹配。
当发送或接收新数据时该位清零

Bit0 ADRF: I²C 从机 7 位地址标志位，只读；
0= I²C 地址不匹配；
1= 1²C 7 位地址匹配。
当发送或接收新数据时该位清零

17.5.2 I²C 清零寄存器（I2CCLR）

0XF2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCLR	I2CIE	I2CEN	STAC	-	SIC	AAC	-	-
WO	WO	WO	WO	WO	WO	WO	WO	WO
复位值	0	0	0	0	0	0	0	0

Bit7 I2CIE： I²C中断禁止位；
0= 不影响；
1= 清零 I2CIE 位。
Bit6 I2CEN： I²C接口禁止位；
0= 不影响；
1= 清零 I2CEN 位。
Bit5 STAC： 启动标志清零位；
0= 不影响；
1= 清零 STA 位。
Bit4 - 保留。
Bit3 SIC： I2C中断标志位；
0= 不影响；
1= 清零 SI 位。
Bit2 AAC= I²C应答标志清零位；
0= 不影响；
1= 清零AA位。
Bit1~0 - 保留。

注：I2C 的操作需要清除相应的标志位才能进入下一状态。

17.5.3 I²C 状态寄存器（I2CSTAT）

0XF3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSTAT	Status7	Status6	Status5	Status4	Status3	Status2	Status1	Status0
RO	RO	RO	RO	RO	RO	RO	RO	RO
复位值	1	1	1	1	1	0	0	0

Bit7~Bit0　　Status:　I²C状态代码

00H:　总线错误（只在主机模式有效）
08H:　启动位发送完成
10H:　重启动位发送完成
18H:　地址+写位发送完成，接收到ACK
20H:　地址+写位发送完成，未接收到ACK
28H:　主机模式下数据发送完成，接收到ACK
30H:　主机模式下数据发送完成，未接收到ACK
38H:　在地址或数据传输过程中仲裁失败
40H:　地址+读位发送完成，接收到ACK
48H:　地址+读位发送完成，未接收到ACK
50H:　主机模式下接收到数据，回复ACK
58H:　主机模式下接收到数据，不回复ACK
60H:　从机模式下接收到地址+写位，回复ACK
68H:　主机仲裁失败，接收到从机地址+写位，回复ACK
70H:　接收到广播呼叫地址，回复ACK
78H:　主机仲裁失败，接收到广播呼叫地址，回复ACK
80H:　从机地址匹配后接收到数据，回复ACK
88H:　从机地址匹配后接收到数据，不回复ACK
90H:　从机接收广播呼叫地址后接收到数据，回复ACK
98H:　从机接收广播呼叫地址后接收到数据，不回复ACK
A0H:　从机模式下接收到停止信号或重启动信号
A8H:　从机模式下接收到地址+读位，回复ACK
B0H:　主机仲裁失败，接收到从机地址+读位，回复ACK
B8H:　从机模式下发送数据后，接收到ACK
C0H:　从机模式下发送数据后，未接收到ACK
C8H:　从机模式下发送完最后一个数据，接收到ACK
D0H:　从机模式下发送完最后一个数据，未接收到ACK
D8H:　未用
E0H:　主机模式下发送完第二个地址，接收到ACK
E8H:　主机模式下发送完第二个地址，未接收到ACK
F0H:　未用
F8H:　不确切的状态
其他：　保留

17.5.4 I²C 数据寄存器（I2CDAT）

0XF4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CDAT	Data7	Data6	Data5	Data4	Data3	Data2	Data1	Data0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　Data<7:0>　接收到的数据或将被发送的数据。

17.5.5 I²C 时钟控制寄存器（I2CCLK）

0XF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCLK	-	M2	M1	M0	N3	N2	N1	N0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7　　-　保留，须为0。
Bit6~Bit4　　M<2:0>:　采样时钟= $F_{SYS}/2^M$
Bit3~Bit0　　N<3:0>:　SCL时钟= F_{SYS} /（2^M×（N+1）×10）

17.5.6 I²C 从机地址寄存器（I2CADR）

0XF6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CADR	Address6	Address5	Address4	Address3	Address2	Address1	Address0	GC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit1　　Address<6:0>::　从机地址ADRS[6:0]。
Bit0　　GC:　广播呼叫地址识别使能控制；
1=　使能广播呼叫地址识别；
0=　禁止广播呼叫地址识别。

17.5.7 I²C 扩展地址复位寄存器（I2CXAR）

0XF7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CXAR	RST	GCF	-	-	-	XADR2	XADR1	XADR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7　　RST:　软件复位；
1=　产生软件复位；
0=　无效。
Bit6　　GCF:　I2C广播呼叫标志位，只读；
1=　广播呼叫地址匹配；
0=　未接收到广播呼叫。
Bit5~Bit3　　-　保留，须为0。
Bit2~Bit0　　XADR<2:0>:　扩展从机10位地址的高3位ADRS[9:7]。

18. UARTn 模块

18.1 概述

通用异步收发器（UART0 / UART1）提供一种灵活的方法与外部设备之间进行全双工数据交换。

UARTn 内部有两个物理上独立的接收、发送缓冲器 SBUFn，通过对 SBUFn 的读写指令来区别是对接收缓冲器还是发送缓冲器进行操作。写 SBUFn 时，数据加载到发送缓冲器；读 SBUFn 时，读取接收缓冲器中的内容。

UARTn 支持两种异步模式。模式 1 和模式 3。模式 3 具有多机通信功能，通过将 SCONn 寄存器中的 SMn2 位置 1 来使能该功能。主机处理器首先发送识别目标从机的地址字节。地址字节与数据字节不同，因为地址字节中第 9 位为 1，数据字节为 0。在 SMn2=1 时，从机不会被数据字节中断。地址字节将中断所有从机。寻址的从机将清除其 SMn2 位，并准备接收将要来的数据字节。未被寻址的从机使 SMn2 置 1 并忽略传入的数据。

18.2 UARTn 端口配置

使用 UARTn 模块前需要先将相应端口配置成 UARTn 的 TXDn 与 RXDn 通道。例如 UART0 的端口配置如下：

PS_TXD0 = 0x00;　　//P04 配置为 TXD0 口

PS_RXD0 = 0x01;　　//P05 配置为 RXD0 口

UARTn 的端口 RXDn 可通过 PS_RXDn 选择（RXDn 引脚只能选择其一），TXDn 端口可通过 PS_TXDn 选择（TXDn 引脚只能选择其一）。

使用时，建议先设置好工作模式，然后再将相应的口配置成串口。

18.3 UARTn 波特率

UARTn 仅支持模式 1 和模式 3，波特率由定时器 Timer1 或 BRT 设置。

18.3.1 波特率时钟源

UARTn 在模式 1 和模式 3 时，波特率时钟源选择如下：

1) UART0 波特率时钟源选择：

   CKCON [3]=1 时，选择 Timer1 作为 UART0 的波特率发生器；

   CKCON [3]=0 时，选择 BRT 作为 UART0 的波特率发生器；

2) UART1 波特率时钟源选择：

   CKCON [4]=1 时，选择 Timer1 作为 UART1 的波特率发生器；

   CKCON [4]=0 时，选择 BRT 作为 UART1 的波特率发生器；

18.3.2 波特率计算

UARTn 在模式 1 和模式 3 时，不同时钟源时波特率计算公式如下：

1) Timer1 工作在 8 位自动重装模式下波特率的公式：

$$BaudRate=\frac{Fsys}{16\times(4\times3^{1-T1M})\times(256-TH1)}$$

T1M 为定时器 1 时钟选择位。即 Timer1 在相应波特率下的 TH1 的值应设置为：

$$TH1=256-\frac{Fsys}{16\times(4\times3^{1-T1M})\times BaudRate}$$

2) BRT 作为波特率发生器时，波特率公式：

$$BaudRate=\frac{Fsys}{16\times(\{BRTDH,BRTDL\}+1)}$$

BRT 在相应波特率下{BRTDH,BRTDL}的值应设置为：

$$\{BRTDH,BRTDL\}=\frac{Fsys}{16\times BaudRate}-1$$

18.3.3 波特率误差

UARTn 在模式 1 和模式 3 时，选择不同的波特率时钟源，不同的波特率下误差如下：

表 1）为在可变波特率模式下，定时器 1 的 8 位自动重装模式下的部分波特率相关信息。

1) T1M=1

波特率	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	TH1	ActualRate	%(Error)	TH1	ActualRate	%(Error)	TH1	ActualRate	%(Error)	TH1	ActualRate	%(Error)
4800	230	4808	0.16	204	4808	0.16	178	4808	0.16	100	4808	0.16
9600	243	9615	0.16	230	9615	0.16	217	9615	0.16	178	9615	0.16
19200	--	--	--	243	19231	0.16	236	18750	2.34	217	19231	0.16
38400	-	-	-				246	37500	2.34	236	37500	2.34
115200	--	--	--									
250000	--	--	--									

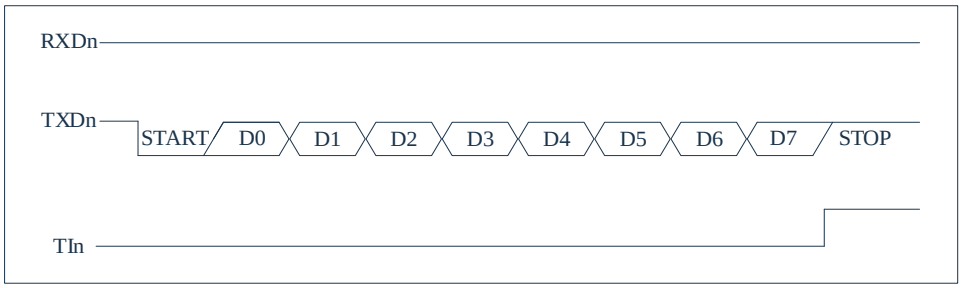
2) BRT 提供波特率时钟

波特率	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{BRTDH, BRTDL}	ActualRate	%(Error)	{BRTDH, BRTDL}	ActualRate	%(Error)	{BRTDH, BRTDL}	ActualRate	%(Error)	{BRTDH, BRTDL}	ActualRate	%(Error)
4800	103	4808	0.16	207	4808	0.16	312	4792	0.16	624	4800	0
9600	51	9615	0.16	103	9615	0.16	155	9615	0.16	312	9585	0.16
19200	25	19231	0.16	51	19231	0.16	77	19231	0.16	155	19231	0.16
38400	12	38462	0.16	25	38462	0.16	38	38462	0.16	77	38462	0.16
115200	--	--	--	8	111111	3.55	12	115385	0.16	25	115385	0.16
250000	--	--	--	--	--	--	--	--	--	11	250000	0
500000	--	--	--	--	--	--	--	--	--	5	500000	0
1000000	--	--	--	--	--	--	--	--	--	2	1000000	0

18.4 UARTn 模式

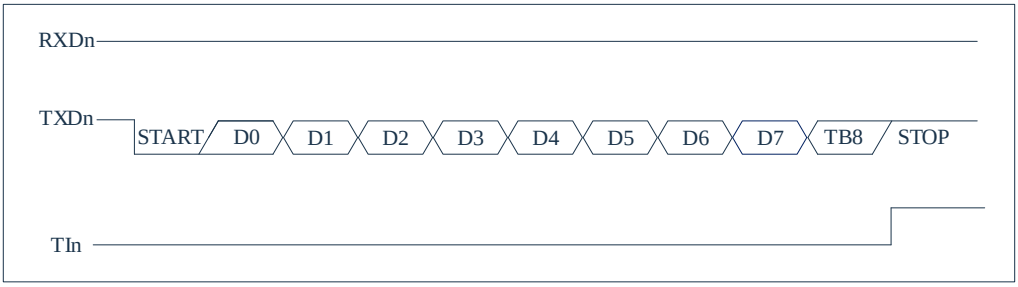
18.4.1 模式 1-8 位异步模式（可变波特率）

引脚 RXDn 用作输入，TXDn 用作串行输出。发送 10 位：起始位（始终为 0），8 位数据（LSB 优先）和停止位（始终为 1）。接收时，起始位同步传输，通过读取 SBUFn 可以获得 8 个数据位，停止位在 SCONn 中设置标志 TIn。波特率是可变的，取决于 TIMER1/BRT 模式。模式 1 时序图如下图所示：



18.4.2 模式 3-9 位异步模式（可变波特率）

模式 1 和模式 3 之间的唯一区别是模式 3 中的增加了第 9 位数据 TB8。当 UnREN=1 时，数据接收使能。波特率是可变的并且取决于 TIMER1/BRT 模式。模式 4 时序图如下图所示：



18.5 UARTn 寄存器

UARTn 具有与标准 8051 UART 相同的功能。其相关寄存器是：SBUFn、SCONn、PCON、IE、IP。UARTn 数据缓冲器（SBUFn）由 2 个独立的寄存器组成：发送和接收寄存器。写入 SBUFn 的数据将在 UARTn 输出寄存器中设置此数据并开始传输；读取 SBUFn 的数据将从 UARTn 接收寄存器中读取数据。SCON0 寄存器支持位寻址操作，SCON1 寄存器不支持位寻址操作，使用汇编语言时需注意。

18.5.1 UARTn 缓冲寄存器 SBUFn

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBUFn	BUFFERn7	BUFFERn6	BUFFERn5	BUFFERn4	BUFFERn3	BUFFERn2	BUFFERn1	BUFFERn0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 SBUF0 地址 0x99；寄存器 SBUF1 地址 0xEB。

Bit7~Bit0　　BUFFERn<7:0>:　缓冲数据寄存器。
写:　UARTn开始发送数据。
读:　读取接收到的数据。

18.5.2 UART 控制寄存器 SCONn

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCONn	UnSM0	UnSM1	UnSM2	UnREN	UnTB8	UnRB8	TIn	RIn
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 SCON0 地址 0x98；寄存器 SCON1 地址 0xEA。

Bit7~Bit6 UnSM0-UnSM1: 多机通信控制位；
00= 保留；
01= 8位异步模式，波特率可变；
10= 保留，禁止选择；
11= 9位异步模式，波特率可变。

Bit5 UnSM2: 多机通信控制位；
1= 使能；
0= 禁止。

Bit4 UnREN: 接收使能位；
1= 使能；
0= 禁止。

Bit3 UnTB8: 发送数据的第9位，主要用于9位异步模式的发送；
1= 第9位数据为1；
0= 第9位数据为0。

Bit2 UnRB8: 接收数据的第9位，主要用于9位异步模式的接收；
1= 接收到的第9位数据为1；
0= 接收到的第9位数据为0。

Bit1 Tln: 发送中断标志位（需要软件清零）；
1= 说明发送缓冲器已空，可以发送下一帧数据。
0= --

Bit0 RIn: 接收中断标志位（需要软件清零）；
1= 说明接收缓冲器已满，读取后可以接收下一帧数据。
0= --

UARTn 模式如下表：

SMn0	SMn1	模式	描述	波特率
0	0	0	-	-
0	1	1	8-Bit UART	由 BRT/Timer1 控制
1	0	2	-	0
1	1	3	9-Bit UART	由 BRT/Timer1 控制

18.6 UARTn 中断

UART0 的中断号为 4，其中断向量为 0x0023。

UART1 的中断号为 6，其中断向量为 0x0033。

使能 UARTn 中断必须将其使能位 ESn 置 1，且将总中断使能位 EA 置 1。若 UARTn 相关的中断使能均打开，TIn=1 或者 RIn=1 时，CPU 将进入相应的中断服务程序。TIn/RIn 与 ESn 的状态无关，且需要软件清零，详细描述参考寄存器 SCONn。

18.6.1 中断屏蔽寄存器 IE

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 EA: 全局中断允许位；
1= 允许所有未被屏蔽的中断；
0= 禁止所有中断。

Bit6 ES1: UART1中断允许位；
1= 允许UART1中断；
0= 禁止UART1中断。

Bit5 ET2: TIMER2总中断允许位；
1= 允许TIMER2所有中断；
0= 禁止TIMER2所有中断。

Bit4 ES0: UART0中断允许位；
1= 允许UART0中断；
0= 禁止UART0中断。

Bit3 ET1: TIMER1中断允许位；
1= 允许TIMER1中断；
0= 禁止TIMER1中断。

Bit2 EX1: 外部中断1中断允许位；
1= 允许外部中断1中断；
0= 禁止外部中断1中断。

Bit1 ET0: TIMER0中断允许位；
1= 允许TIMER0中断；
0= 禁止TIMER0中断。

Bit0 EX0: 外部中断0中断允许位；
1= 允许外部中断0中断；
0= 禁止外部中断0中断。

18.6.2 中断优先级控制寄存器 IP

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

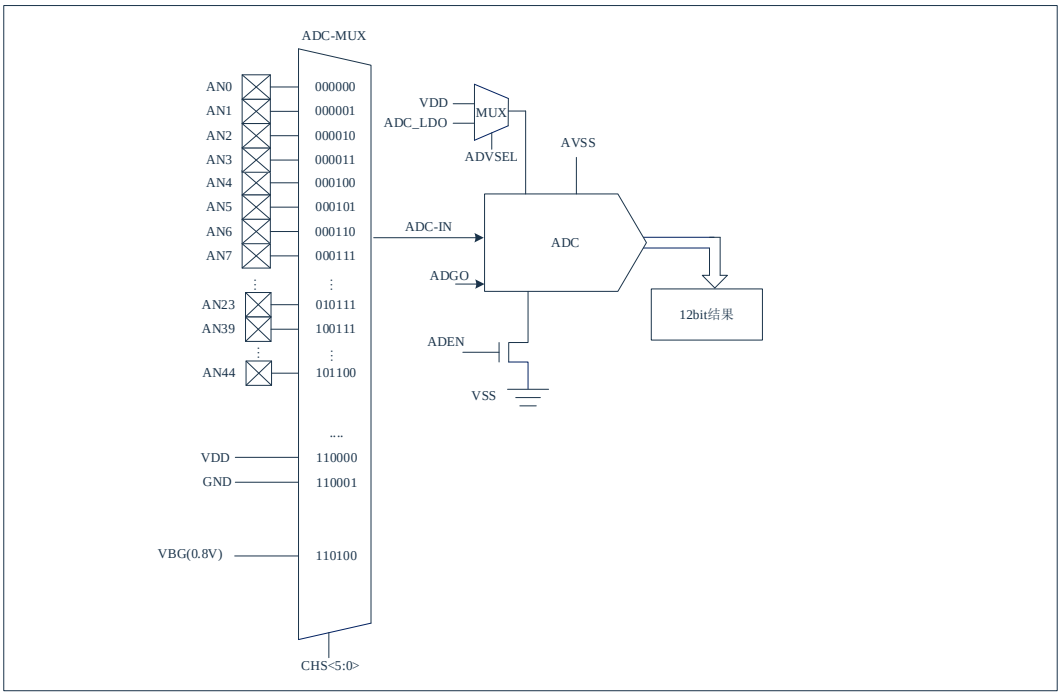
Bit7 -- 保留，须为0。
Bit6 PS1: UART1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit5 PT2: TIMER2中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit4 PS0: UART0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit3 PT1: TIMER1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit2 PX1: 外部中断1中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit1 PT0: TIMER0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit0 PX0: 外部中断0中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。

19. 模数转换器（ADC）

19.1 概述

模数转换器（ADC）可以将模拟输入信号转换为表示该信号的一个 12 位二进制数，ADC 结构框图如下图所示。

端口模拟输入信号和内部模拟信号经过多路选择器之后与模数转换器的输入相连。模数转换器采用逐次逼近法产生一个 12 位二进制结果，并将该结果保存在 ADC 结果寄存器（ADRESL 和 ADRESH）中，ADC 在转换完成之后可以产生一个中断。ADC 参考电压由控制信号 LDOEN 选择 VDD 或 ADC_LDO 提供。



19.2 ADC 配置

配置和使用 ADC 时，必须考虑如下因素：

- 端口配置。
- 通道选择。
- ADC 转换时钟源。
- 中断控制。

19.2.1 端口配置

ADC 既可以转换模拟信号，又可以转换数字信号。当转换模拟信号时，需将相应的端口配置为模拟端口。

注：对定义为数字输入的引脚施加模拟电压可能导致输入缓冲器出现过电流。

19.2.2 通道选择

由 ADCON1 寄存器的 ADCHS 位决定将哪个通道连接到模数转换器。

如果更改了通道，在下一次转换开始前需要一定的延迟。ADC 延时时间如下表所示：

延时时间	工作电压
500ns	2.5~4.5V
200ns	4.5~5.5V

19.2.3 ADC 参考电压

ADC 的参考电压默认由芯片的 VDD 提供，也可由内部 ADC-LDO 提供。ADC-LDO 输出电压：2.4V。

19.2.4 转换时钟

可以通过软件设置 ADCON0 寄存器的 ADCKS 位来选择转换的时钟源。

完成一位转换的时间定义为 T_{ADCK}。一个完整的 12 位转换需要 18.5 个 T_{ADCK} 周期（完成一次转换 ADGO 持续为高的时间）。必须符合相应的 T_{ADCK} 规范，才能获得正确的转换结果，下表为正确选择 ADC 时钟的示例。

Fsys	F_{ADCK} (T_A=25℃)	
	V_{REF}=V_{REFE}=AVDD (AVDD=VDD)	V_{REF}=V_{REFI}=2.4V
8MHz	Fsys/4	Fsys/16
16MHz	Fsys/8	Fsys/32
24MHz	Fsys/16	Fsys/64
48MHz	Fsys/32	Fsys/128

注：系统时钟频率的任何改变都会改变ADC时钟的频率，从而对ADC转换结果产生负面影响。

19.3 ADC 工作原理

19.3.1 启动转换

要使能 ADC 模块，必须先将 ADCON1 寄存器的 ADEN 位置 1，然后将 ADCON0 寄存器的 ADGO 位置 1 开始模数转换（ADEN 为 0 时无法将 ADGO 置 1）。

19.3.2 完成转换

当转换完成时，ADC 模块将：

- 清零 ADGO 位；
- 将 ADCIF 标志位置 1；
- 用转换的新结果更新 ADRESH:ADRESL 寄存器。

19.3.3 终止转换

如果必须要在转换完成前终止转换，尚未完成的模数转换结果不会更新到 ADRESH:ADRESL 寄存器。因此，ADRESH:ADRESL 寄存器将保持上次转换所得到的值。

注：器件复位将强制所有寄存器进入复位状态。因此，复位会关闭 ADC 模块并且终止任何待处理的转换。

19.3.4 A/D 转换步骤

使用 ADC 进行模数转换的配置步骤如下：

1) 端口配置：
   - 禁止引脚输出驱动器（见 PxTRIS 寄存器）；
   - 将引脚配置为模拟输入引脚。
2) 配置 ADC 中断（可选）：
   - 清零 ADC 中断标志位；
   - 允许 ADC 中断；
   - 允许外设中断；
   - 允许全局中断。
3) 配置 ADC 模块：
   - 选择 ADC 转换时钟；
   - 选择 ADC 输入通道；
   - 启动 ADC 模块。
4) 等待所需的采集时间。
5) 将 ADGO 置 1 启动转换。
6) 由如下方法之一等待 ADC 转换结束：
   - 查询 ADGO 位；
   - 等待 ADC 中断（允许中断）。
7) 读 ADC 结果。
8) 将 ADC 中断标志位清零（如果允许中断的话，需要进行此操作）。

注：如果用户尝试在使器件从休眠模式唤醒后恢复顺序代码执行，则必须禁止全局中断。

19.3.5 转换过程中进入休眠

系统进入休眠时，建议等待 ADC 正在进行的转换完成后，再进入休眠状态。

若在 ADC 正在进行的转换过程中进入休眠，则本次转换终止。唤醒后需重新进行转换操作。

19.4 相关寄存器

主要有 4 个寄存器与 AD 转换相关，分别是：

- AD 控制寄存器 ADCON0、ADCON1；
- AD 结果数据寄存器 ADRESH/L；

19.4.1 AD 控制寄存器 ADCON0

0xDF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	--	ADCKS2	ADCKS1	ADCKS0	--	--	ADGO	--
读写	W	R/W	R/W	R/W	W	W	R/W	W
复位值	0	0	0	0	0	0	0	0

Bit7 -- 保留，须为1。

Bit6~ Bit4 ADCKS<2:0>: ADC转换时钟选择位；（切换ADC时钟时，需要先关闭ADC使能）

000= Fsys/2； 100= Fsys/32；

001= Fsys/4； 101= Fsys/64；

010= Fsys/8； 110= Fsys/128；

011= Fsys/16； 111= Fsys/256。

Bit3~Bit2 -- 保留，须为0。

Bit1 ADGO: ADC转换启动位（对该位置1时ADEN必须为1，否则操作无效）；

1= 写入：开始ADC转换；

读取：ADC正在进行转换。

0= 写入：无效。

读取：ADC空闲/转换完毕；

在ADC的转换期间（ADGO=1），任何软件触发信号将被忽略。

Bit0 -- 保留，须为0。

19.4.2 AD 控制寄存器 ADCON1

0xDE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	ADEN	ADVSEL	ADCHS5	ADCHS4	ADCHS3	ADCHS2	ADCHS1	ADCHS0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 ADEN: ADC使能位；
1= 使能ADC；
0= 禁止ADC，不消耗工作电流。

Bit6 ADVSEL ADC参考电压选择
0:选择VDDREF（LDOAD模块必须关闭）
1:选择ADCLDO

Bit5~Bit0 ADCHS<5:0>: 模拟通道选择位，；

000000=	AN0(P00)；	010101=	AN21(P25)；
000001=	AN1(P01)；	010110=	AN22(P26)；
000010=	AN2(P02)；	010111=	AN23(P27)；
000011=	AN3(P03)；	011000-100110	保留，禁止选择
000100=	AN4(P04)；	100111=	AN39(P50)；
000101=	AN5(P05)；	101000=	AN40(P51)；
000110=	AN6(P06)；	101001=	AN41(P52)；
000111=	AN7(P07)；	101010=	AN42(P53)；
001000=	AN8(P10)	101011=	AN43(P54)；
001001=	AN9(P11)；	101100=	AN44(P55)；
001010=	AN10(P12)；	101101=	保留，禁止选择
001011=	AN11(P13)；	101110=	保留，禁止选择
001100=	AN12(P14)；	101111=	保留，禁止选择
001101=	AN13(P15)；	110000=	AN48(VDD)
001110=	AN14(P16)；	110001=	AN49(GND)
001111=	AN15(P17)；	110010=	保留，禁止选择
010000=	AN16(P20)；	110011=	保留，禁止选择
010001=	AN17(P21)；	110100=	AN52(vbg0.8)
010010=	AN18(P22)；	110101=	保留，禁止选择
010011=	AN19(P23)；	110110=	保留，禁止选择
010100=	AN20(P24)；	110111=	保留，禁止选择
		其他	保留，禁止选择

19.4.3 AD 数据寄存器高位 ADRESH

0xDD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESH	--	--	--	--	ADRES11	ADRES10	ADRES9	ADRES8
读写	--	--	--	--	R	R	R	R
复位值	--	--	--	--	X	X	X	X

Bit7~Bit4 未用。

Bit3~Bit0 ADRES<11:8>: ADC结果寄存器位。
12位转换结果的第11-8位。

19.4.4 AD 数据寄存器低位 ADRESL

0xDC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESL	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
读写	R	R	R	R	R	R	R	R
复位值	X	X	X	X	X	X	X	X

Bit7~Bit0　　ADRES<7:0>:　ADC结果寄存器位。
12位转换结果的第7-0位。

19.4.5 AD 参考电压控制寄存器 ADCLDO

F693H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCLDO	LDOEN	--	--	--	--	--	--	--
读写	R/W	--	--	--	--	--	--	--
复位值	0	0	0	0	0	0	0	0

Bit7　　LDOEN　ADC_LDO使能；
1=　LDO使能，参考电压选择2.4V；
0=　LDO禁止，参考电压为芯片电源电压。
Bit6~Bit0　　--　保留，须为0。

注意：当 ADC 选择 VDD 做参考电压时（ADCON1[6]=0），必须提前关闭 ADCLDO 使能(ADCLDO[7]=0)，并确保 ADC 选择 VDD 做参考期间 ADCLDO 使能一直保持关闭；

当 ADC 选择 ADCLDO 做参考电压时（ADCON1[6]=1），必须先配置 ADC 选择 ADCLDO 做参考电压（ADCON1[6]=1），再使能 ADCLDO(ADCLDO[7]=1)。

19.5 ADC 中断

ADC 模块允许在完成模数转换后产生一个中断。ADC 中断允许位是 EIE2 寄存器中的 ADCIE 位，ADC 中断标志位为 EIF2 寄存器中的 ADCIF 位。ADCIF 位必须用软件清零，每次转换结束后 ADCIF 位都会被置 1，与是否允许 ADC 中断无关。ADC 的中断使能、优先级可通过如下相关寄存器位设置。

19.5.1 中断屏蔽寄存器 EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	--	ADCIE	PWMIE	ET5	--	--
R/W	R/W	R/W	--	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI中断使能位；
1= 允许SPI中断；
0= 禁止SPI中断。
Bit6 I2CIE: I^2C中断使能位；
1= 允许I^2C中断；
0= 禁止I^2C中断。
Bit5 -- 保留，须为0。
Bit4 ADCIE: ADC中断使能位；
1= 允许ADC中断；
0= 禁止ADC中断。
Bit3 PWMIE: PWM总中断使能位；
1= 允许PWM所有中断；
0= 禁止PWM所有中断。
Bit2 ET5: Timer5中断使能位；
1= 允许Timer5中断；
0= 禁止Timer5中断。
Bit1~Bit0 -- 保留，须为0。

19.5.2 中断优先级控制寄存器 EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	--	PADC	PPWM	PT5	--	--
R/W	R/W	R/W	--	R/W	R/W	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit6 PI2C: I^2C中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit5 -- 保留，须为0。
Bit4 PADC: ADC中断优先级控制位；
1= 设置为高级中断；
0= 设置为低级中断。
Bit3 PPWM: PWM中断优先级控制位
1= 设置为高级中断；
0= 设置为低级中断。
Bit2 PT5: TIMER5中断优先级控制位
1= 设置为高级中断；
0= 设置为低级中断。
Bit1~Bit0 -- 保留，须为0。

19.5.3 外设中断标志位寄存器 EIF2

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	TF5	--	--
R/W	R	R	R	R/W	R	R/W	R	R
复位值	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI总中断指示位，只读；
1= SPI产生中断，（清除具体的中断标志位后，此位自动清除）；
0= SPI未产生中断。
Bit6 I2CIF: I^2C总中断指示位，只读；
1= I^2C产生中断，（清除具体的中断标志位后，此位自动清除）；
0= I^2C未产生中断。
Bit5 -- 保留，须为0。
Bit4 ADCIF: ADC中断标志位；
1= ADC转换完成，需软件清零；
0= ADC转换未完成。
Bit3 PWMIF: PWM总中断指示位，只读；
1= PWM产生中断，（清除具体的中断标志位后，此位自动清除）；
0= PWM未产生中断。
Bit2 TF5: Timer5定时器溢出中断标志位；
1= Timer5定时器溢出，需软件清零；
0= Timer5定时器无溢出。
Bit1~Bit0 -- 保留，须为0。

20. 循环冗余校验单元（CRC）

20.1 概述

为了保证运行过程中的安全，IEC61508 标准要求即使在 CPU 运行中也需要确认数据。此通用 CRC 模块能在 CPU 运行中作为外围功能进行 CRC 运算。通用 CRC 模块通过程序指定要确认的数据进行 CRC 校验，不限于代码闪存区而能用于多用途的检查。

CRC 生成多项式使用 CRC16-CCITT 的“$X^{16}+X^{12}+X^{5}+1$”。

20.2 相关寄存器

20.2.1 CRC 数据输入寄存器 CRCIN

F708H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CRCIN	CRCIN7	CRCIN6	CRCIN5	CRCIN4	CRCIN3	CRCIN2	CRCIN1	CRCIN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　CRCIN<7:0>　输入需要CRC运算的8位数据。

20.2.2 CRC 运算结果低 8 位数据寄存器 CRCDL

F709H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CRCDL	CRCD7	CRCD6	CRCD5	CRCD4	CRCD3	CRCD2	CRCD1	CRCD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　CRCD<7:0>　CRC运算结果低8位数据

20.2.3 CRC 运算结果高 8 位数据寄存器 CRCDH

F70AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CRCDH	CRCD15	CRCD14	CRCD13	CRCD12	CRCD11	CRCD10	CRCD9	CRCD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　CRCD<15:8>　CRC运算结果高8位数据

20.3 功能描述

在写 CRCIN 寄存器后经过 1 个系统时钟，将 CRC 的运算结果保存到 CRCDL/CRCDH 寄存器。如有需要，则在写入覆盖之前需读取前一次的运算数据，否则会被新的运算结果覆盖。

例如：发送数据“12345678H”，按照“12H”、“34H”、“56H”、“78H”的顺序给 CRCIN 寄存器写值，写完后从 CRCDL/CRCDH 寄存器读取为 CRCDL=0xF0,CRCDH=0x67，即数据“12345678H”的位序进行 CRC 运算的结果为 0x67F0。寄存器操作如下：

```
CRCIN=0x12;    //发送第一个数
CRCIN=0x34;    //发送第二个数
CRCIN=0x56;    //发送第三个数
CRCIN=0x78;    //发送第四个数
resl=CRCDL;        //读取 CRC 运算结果的低 8 位到变量 resl
resh=CRCDH;    //读取 CRC 运算结果的高 8 位到变量 resh
```

21. 触摸模块

触摸模块是为实现人体触摸接口而设计的集成电路，可替代机械式轻触按键，实现防水防尘、密封隔离、坚固美观的操作接口。

技术参数：

- 最多达 30 个触摸按键可选。
- 无需外部触摸电容。

21.1 触摸模块使用注意事项

- 触摸按键检测部分的地线应该单独连接成一个独立的地，再有一个点连接到整机的共地。
- 避免高压、大电流、高频操作的主板与触摸电路板上下重叠安置。如无法避免，应尽量远离高压大电流的期间区域或在主板上加屏蔽。
- 感应盘到触摸芯片的连线尽量短和细。
- 感应盘到触摸芯片的连线不要跨越强干扰、高频的信号线。

22. 存储器管理控制器（MMC/FMC）

22.1 概述

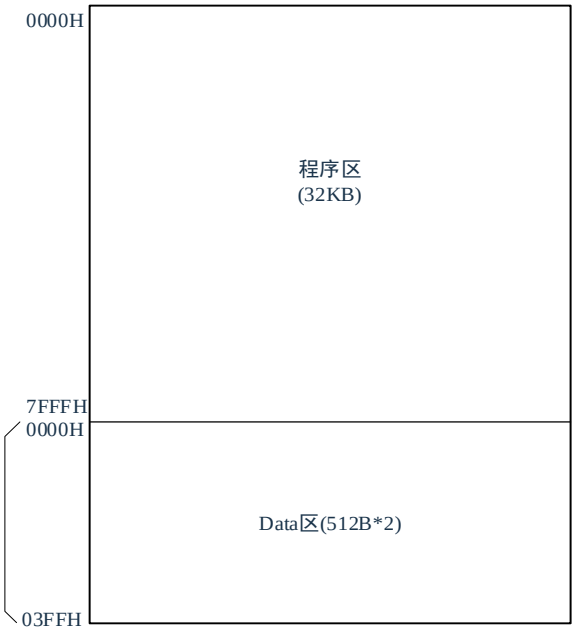
存储器包含程序存储器（APROM）、非易失数据存储器（Data）。程序存储器空间最大为 32KB。Data 空间为 1KB。

APROM 共有 64 扇区，每个扇区包含 512B。

Data 共有 2 个扇区，每个扇区包含 512 个字节。

Information 信息共 32 字节。

UID 共 16 个字节。



可通过相关特殊功能寄存器（SFR）对 FLASH 存储器进行存取操作以实现 IAP 功能。用于访问 FLASH 空间的 SFR 寄存器如下：

- MLOCK
- MSTATUS
- MDATA
- MADRL
- MADRH
- MREGION
- MMODE

MLOCK 寄存器用于使能存储器操作，MSTATUS 寄存器用于指示 FLASH 操作状态、以及设置操作开始控制位，MDATA 寄存器形成一个字节用于保存要写的 8 位数据，MADRL/MADRH 寄存器存放被访问的 MDATA 单元的地址， MREGION 寄存器用于存储器区域选择，MMODE 用于存储器操作模式选择。

通过存储器模块接口，可对存储器进行读取/写入/擦除/CRC 校验操作。存储器允许字节读写，写入时间由片上定时器控制，在写入新数据之前需确保该地址中的数据已被擦除。写入和擦除电压是由片上电荷泵产生，此电荷泵额定工作电压在器件的电压范围内，用于进行字节操作。

Flash 存储器擦除操作仅支持扇区擦除，不支持字节擦除。在修改某个地址的数据之前，建议先将其他数据保存后，再擦除当前扇区，最后进行数据写入操作。

22.2 相关寄存器

22.2.1 存储器保护锁定寄存器 MLOCK

0xFF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MLOCK	MLOCK7	MOCK6	MLOCK5	MLOCK4	MLOCK3	MLOCK2	MLOCK1	MLOCK0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　MLOCK<7:0>:　存储器操作使能位；
AAH=　允许存储器相关W/E/R/CRC操作；
其他=　不允许存储器相关W/E/R/CRC操作。

22.2.2 存储器状态寄存器 MSTATUS

0xFE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MSTATUS	MLOCKF	ERROR	-	RDINC	START	-	-	-
读写	R	R/W	R	R/W	R/W	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7　　MLOCKF:　存储器操作使能状态指示位（MLOCK使能时该位为1，否则为0）；
1=　使能，可通过寄存器操作FLASH；
0=　禁止，不可操作FLASH。
Bit6　　ERROR　操作出错标志位（写0清除）
1=　在编程操作开始前，检测编程地址中的数据不为“FFH”（未擦除），写入操作立即终止。（仅支持单字节写操作）
Bit5　　--　保留
Bit4　　RDINC:　单字节读模式地址自动加1使能位（仅在单字节读取操作完成后有效）；
1=　ADDRH/ADDRL自动加1
0=　ADDRH/ADDRL地址不变
Bit3　　START:　操作开始控制位；
1=　启动存储器W/E/R/CRC校验操作（操作完成后，可由硬件自动清零）；
0=　写：终止或不启动程序存储器W/E/R/CRC校验操作；
Bit2~Bit0　　--　保留，须为0。

22.2.3 存储器数据寄存器 MDATA

0xFB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MDATA	MDATA7	MDATA6	MDATA5	MDATA4	MDATA3	MDATA2	MDATA1	MDATA0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　MDATA<7:0>:　单字节写入：对程序存储器进行写入的数据。

22.2.4 存储器地址寄存器 MADRL

0xFC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADRL	MADRL7	MADRL6	MADRL5	MADRL4	MADRL3	MADRL2	MADRL1	MADRL0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0　　MADRL<7:0>:　单字节写入：指定存储器写入操作的地址低8位。
多字节写入：第7位为多字节写入地址的第7位，第6-0位为结束地址的低7位。
CRC校验：的结束地址低位（CRC校验模式下有效），第7位有效，忽略第6-0位。

22.2.5 存储器地址寄存器 MADRH

0xFD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADRH	--	MADRH6	MADRH5	MADRH4	MADRH3	MADRH2	MADRH1	MADRH0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7　　--　保留。
Bit6~Bit0　　MADRH<6:0>:　单字节写入：指定存储器写入操作的地址高8位。
多字节写入：多字节写入地址的高位。
存储器CRC操作的结束地址高8位（CRC校验模式下有效）。

APROM 的地址范围：0x0000-0x7FFF

DATA 区的地址范围：0x000-0x3FFH

注意：超出地址范围的操作无效。

22.2.6 存储器区域控制寄存器 MREGION

0xF9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MREGION	REGION7	REGION6	REGION5	REGION4	REGION3	REGION2	REGION1	REGION0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	0	1	0	1	0	1	0

Bit7~Bit0　　REGION<7:0>　Flash区域选择位；
55H=　选择APROM区；
AAH=　选择DATA区；
69H=　选择information（只能进行读操作，禁止写和擦除操作）；
66H=　选择UID（只能进行读操作，禁止写和擦除操作）；
其他=　无效，将退出所有正在进行的操作。

22.2.7 存储器模式控制寄存器 MMODE

0xFA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MMODE	MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	1	0	1	0	0	1

Bit7~Bit0　　MODE<7:0>　Flash模式选择位；
55H=　擦除操作(扇区擦除)；
AAH=　单字节写操作（单字节）
A9H=　多字节写操作（1~128个字节），将XRAM最后128个地址的内容写入多个地址中
6AH=　多字节读操作（1~128个字节），将多个地址的内容写入XRAM最后128个地址中
69H=　单字节读操作（可配置读取完完毕后{MADRH, MADRL}自动+1）
96H=　CRC校验；
其他=　保留，禁止使用。

22.3 功能描述

FLASH 存储器读/写/擦除操作时，CPU 处于暂停状态，操作完成时，CPU 继续运行指令。

注：FLASH 的读/写/擦除/CRC 校验，须保证 MLOCK、MREGION、MMODE 均为有效配置才能启动操作。在操作的过程中，若 MLOCK、MREGION、MMODE 任一寄存器无效，操作都将被禁止。

FLASH 的多字节操作需要缓存来支持，在多字节操作模式下，XRAM 的最后 128 个地址(0x0780~0x7FF)被用作缓存，为了防止读写冲突，此时不建议将其定义为变量区域。在此模式下，为了加快 FMC 的操作，写入，读取操作都通过缓存中转数据来完成。如果不使用多字节操作，则这些 XRAM 中的缓存(0x0780~0x7FF)可当作普通变量区域使用。

22.3.1 读操作（单字节读取）

存储器单字节读操作步骤如下：

1) 使能访问存储器寄存器：
   MLOCK=0xAA；
2) 设置将要访问的存储器地址对应区域：
   通过 MREGION 寄存器设置区域。
3) 设置要访问的存储器地址：
   通过 MADRL/MADRH 设置地址。
4) 设置单字节读命令：
   MMODE=0x69。
5) 启动写操作：
   MSTATUS[3]置 1。
6) 等待 6 个 NOP 指令后，判断写操作是否结束：
   写操作结束后 MSTATUS[3]硬件清 0。
7) 禁止存储器访问操作：
   MLOCK=0x00。

结束后，Flash 相应地址的内容加载到了 MDATA 中，且可配置操作结束后{MADRH,MADRL}的中内容加 1。

22.3.2 多字节读操作（128 字节，读取到缓存）

存储器多字节读操作步骤如下：

1) 使能访问存储器寄存器：

MLOCK=0xAA；

2) 设置将要访问的存储器地址对应区域：

通过 MREGION 寄存器设置区域。

3) 设置要访问的存储器地址起始地址以及结束地址：

通过 MADRL/MADRH 设置地址。例如：0x200(起始地址)+0x7F（偏移结束地址）=0x27F。

起始地址低 7 位固定为 0，MADRL[6:0]为结束的读取地址（即读取字节数-1）。

4) 设置多字节读命令：

MMODE=0x6A。

5) 启动写操作：

MSTATUS[3]置 1。

6) 等待 6 个 NOP 指令后，判断写操作是否结束：

写操作结束后 MSTATUS[3]硬件清 0。

7) 禁止存储器访问操作：

MLOCK=0x00。

结束后，Flash 地址 0x200 的数据写入到了 XRAM 的 0x780 中, 0x201 的数据写入到了 0x781 中，……，0x27F 的数据写入到了 0x7FF 中。

22.3.3 程序区读操作（MOVC 指令）

MOVC 指令仅支持读取程序区域。

22.3.4 单字节写操作

存储器写操作步骤如下：

1) 使能访问存储器寄存器：

MLOCK=0xAA；

2) 设置将要访问的存储器地址对应区域：

通过 MREGION 寄存器设置编程区域。

3) 设置要访问的存储器地址：

通过 MADRL/MADRH 设置地址。例如 0x200H。

4) 设置要写入的数据：

将数据写入 MDATA。例如 0x12。

5) 设置单字节写命令：

MMODE=0xAA。

6) 启动写操作：

MSTATUS[3]置 1。

7) 等待 6 个 NOP 指令后，判断写操作是否结束：

写操作结束后 MSTATUS[3]硬件清 0。

8) 禁止存储器访问操作：

MLOCK=0x00。

此时，地址 0x200 写入了 0x12。

22.3.5 多字节写操作（1~128 字节，先写入缓存）

存储器写操作步骤如下：

1) 首先将要写入的 1~128 个字节的数据放入 XRAM 最后 128 个地址中：

例如 XRAM 的 0x0780 地址放入 0x80，0x00781 放入 0x81，……， 0x0007FF 放入 0xFF。这 128 个字节为类似缓存的作用。

2) 使能访问存储器寄存器：

MLOCK=0xAA；

3) 设置将要访问的存储器地址对应区域：

通过 MREGION 寄存器设置编程区域。

4) 设置要访问的存储器地址起始地址以及结束地址：

通过 MADRL/MADRH 设置地址。例如：0x200(起始地址)+0x7F（偏移结束地址）=0x27F。

起始地址低 7 位固定为 0，MADRL[6:0]为结束的编程地址（即编程字节数-1）。

5) 设置多字节写命令：

MMODE=0xA9。

6) 启动写操作：

MSTATUS[3]置 1。

（如果缓存中有的字节为空数据 FFH，除了第一个字节其他将会跳过实际编程操作进入下一字节的编程）

7) 等待 6 个 NOP 指令后，判断写操作是否结束：

写操作结束后 MSTATUS[3]硬件清 0。

8) 禁止存储器访问操作：

MLOCK=0x00。

结束后，Flash 地址 0x200 写入了 0x80，地址 0x201 写入了 0x81，……，地址 0x27F 写入了 0x7F。

注意：a. 如果所有字节为非 0xFF 数据，1 个字节编程时间约为 25us，每多一个字节时间增加约 8us，128 字节编程需要的时间约为 1.05ms.

b. 如果所有字节均为 0xFF 数据，除第一个字节编程时间为 25us，其他 1 个字节编程时间约为 1us，128 字节编程需要的时间约为 140us.

22.3.6 擦除操作（扇区擦除）

存储器写操作步骤如下：

1) 使能访问存储器寄存器：

   MLOCK=0xAA；

2) 设置要擦除的存储器地址：

   通过 MADRL/MADRH 设置擦除扇区的地址。

3) 设置将要擦除的存储器地址对应区域：

   通过 MREGION 寄存器设置擦除区域。

4) 设置擦除命令：

   MMODE=0x55。

5) 启动擦除操作：

   MSTATUS[3]置 1。

6) 等待 6 个 NOP 指令后，判断等待擦除操作是否结束：

   擦除操作结束后 MSTATUS[3]硬件清 0。

7) 禁止存储器访问操作：

   MLOCK=0x00。

22.3.7 存储器 CRC 校验

存储器的 CRC 校验需要通用 CRC 模块配合使用，FMC 控制器产生启动、控制、结束信号，运算由通用 CRC 模块来完成。

CRC 校验命令由寄存器 MMODE 设置， MREGION 寄存器设置校验区域，起始地址固定为 0，结束地址通过寄存器 MADRL/MADRH 可自由配置（MADRL 第 6-0 位被忽略），即配置最小单元块为 128Byte。其结果保存在寄存器 CRCDL/CRCDH 中。在 CRC 校验过程中，CPU 停止工作，等待 CRC 计算完成后 CPU 再继续运行。该 CRC 校验按字节方式校验，顺序从 0 地址到结束地址。校验速度为 2 个主频时钟(2*T_{SYS})校验一个字节。

CRC 校验操作步骤如下：

1) 清除程序 CRC 之前校验结果：
   CRCIN=0x00；CRCDL/CRCDH=0x00。
2) 使能访问程序存储器寄存器：
   MLOCK=0xAA；
3) 设置程序 CRC 校验结束地址：
   通过 MADRL/MADRH 设置结束地址。
4) 选择 CRC 校验区域：
   通过 MREGION 寄存器设置校验区域。
5) 选择 CRC 校验命令：
   MMODE=0x96。
6) 启动 CRC 校验：
   MSTATUS[3]置 1。
7) 等待 6 个 NOP 指令后，判断程序 CRC 校验是否结束：
   CRC 校验结束后 MSTATUS[3]硬件清 0。
8) 读取程序 CRC 校验结果：
   CRCDL 存放程序 CRC 运算结果的低 8 位；
   CRCDH 存放程序 CRC 运算结果的高 8 位。
9) 禁止存储器访问操作：
   MLOCK=0x00。

22.3.8 Information 信息

用户不能通过程序修改，可通过烧写器更新。用于记录防窜码、wifi 序列、滚码等只读不写的信息。

Information 的地址为 0x480-0x49F 对应的 32 个字节。

22.3.9 唯一 ID（UID）

出厂时已经设定完成。烧写器及用户均不能修改。

每颗芯片拥有不同 128 位唯一身份识别号，即唯一 ID（Unique identification）。出厂时已经设置，用户不能修改。

UID 的地址为 0x680-0x68F 对应的 16 个字节。

23. 用户配置

系统配置寄存器（CONFIG）是 MCU 初始条件的 FLASH 选项，程序不能访问及操作。它包含了以下内容：

1.WDT（看门狗工作方式选择）

- ENABLE　　强制打开 WDT
- SOFTWARE CONTROL（默认）　　WDT 工作方式由 WDKEY 寄存器控制

2.PROTECT

- DISABLE（默认）　　FLASH 代码不加密
- ENABLE　　FLASH 代码加密，读出的代码为 00H。

3.FLASH_DATA_PROTECT

- DISABLE（默认）　　FLASH 数据区不加密
- ENABLE　　FLASH 数据区加密，加密后烧写仿真器读出来的值为 00H

4.LVD 低压检测使能

- 使能（默认）
- 禁止

5.LVD 复位检测电压设定

- 2.5V（默认）
- 2.7V
- 3.0V
- 3.3V
- 3.7V
- 4.0V
- 4.3V

6.LVD 模式选择

- 中断模式　　检测电压设定：见寄存器 LVDS[2:0]
- 复位模式（默认）　　检测电压设定：见用户配置 LVD 复位检测电压设定

7.DEBUGEN（调试模式使能）

- DISABLE（默认）　　调试模式禁止，DSCK，DSDA 引脚用做普通 IO 口
- ENABLE　　调试模式使能，DSCK，DSDA 引脚配置成调试口，引脚对应的其他功能关闭

8.DBGSEL（调试口选择，需要调试模式使能）

- TWI (默认双线调试)
- SWI（单线调试）

9.OSC（振荡方式）

- HSI（默认）　　48MHz
- HSE　　16MHz/8MHz（CONFIG 选择 HSE，P51、P52 禁止配置其他功能）
- LSI(32KHz)　　32KHz

10. SYS_PRESCALE（系统时钟预分频选择）

- Fosc/1（默认）
- Fosc/2
- Fosc/3
- Fosc/4
- Fosc/6
- Fosc/8
- Fosc/16
- Fosc/32

11. EXT_RESET（外部复位配置）

- DISABLE（默认）　　外部复位禁止
- ENABLE(OPEN PULLUP)　　外部复位使能且打开复位口内部上拉电阻

12. WAKE UP_WAIT TIME（休眠唤醒等待振荡器稳定的时间默认为 1.0s）
    - 50us
    - 100us
    - 500us
    - 1ms
    - 5ms
    - 10ms
    - 500ms
    - 1.0s（默认）
13. WTSTEN 程序存储器等待时间配置
    - 等待时间以 WTST 的配置为准
    - 等待时间 Fsys=48M，为 2T； Fsys=其他，为 1T（默认）
14. WTST 程序存储器等待时间选择位
    - 1*Tsys（Fsys=48M 时，禁止选择 1T）
    - 2*Tsys
15. WRITE_PROTECT 程序分区保护（可保护区间，所有默认区间为不保护）
    - 0000H-07FFH（保护/不保护）
    - 1000H-17FFH（保护/不保护）
    - 2000H-27FFH（保护/不保护）
    - 3000H-37FFH（保护/不保护）
    - 4000H-47FFH（保护/不保护）
    - 5000H-57FFH（保护/不保护）
    - 6000H-67FFH（保护/不保护）
    - 7000H-77FFH（保护/不保护）
    - 0800H-0FFFH（保护/不保护）
    - 1800H-1FFFH（保护/不保护）
    - 2800H-2FFFH（保护/不保护）
    - 3800H-3FFFH（保护/不保护）
    - 4800H-4FFFH（保护/不保护）
    - 5800H-5FFFH（保护/不保护）
    - 6800H-6FFFH（保护/不保护）
    - 7800H-7FFFH（保护/不保护）
16. WRITE_PROTECTDATA 分区保护（可保护区间，所有默认区间为不保护）
    - 0000H-01FFH（保护/不保护）
    - 0200H-03FFH（保护/不保护）

24. 在线编程与调试

24.1 在线编程模式

可在最终应用电路中对芯片进行串行编程。编程可以简单地通过以下 4/3 根线完成：

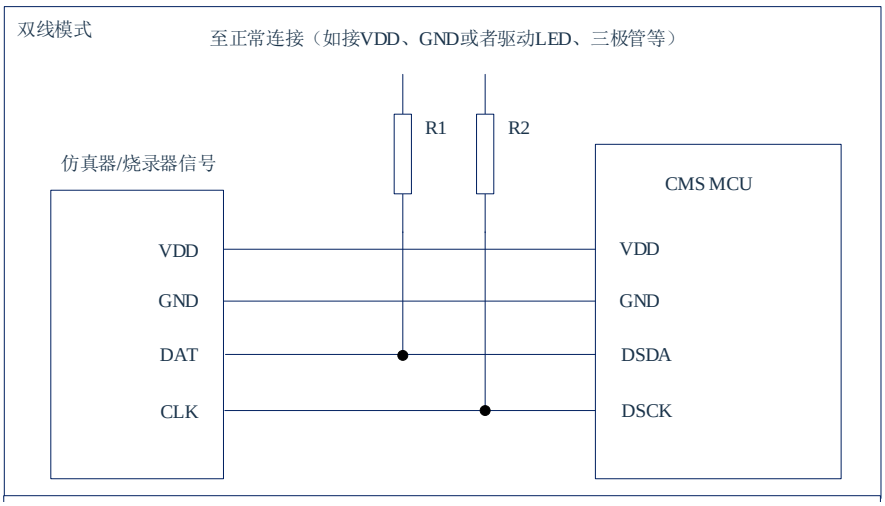
双线模式：

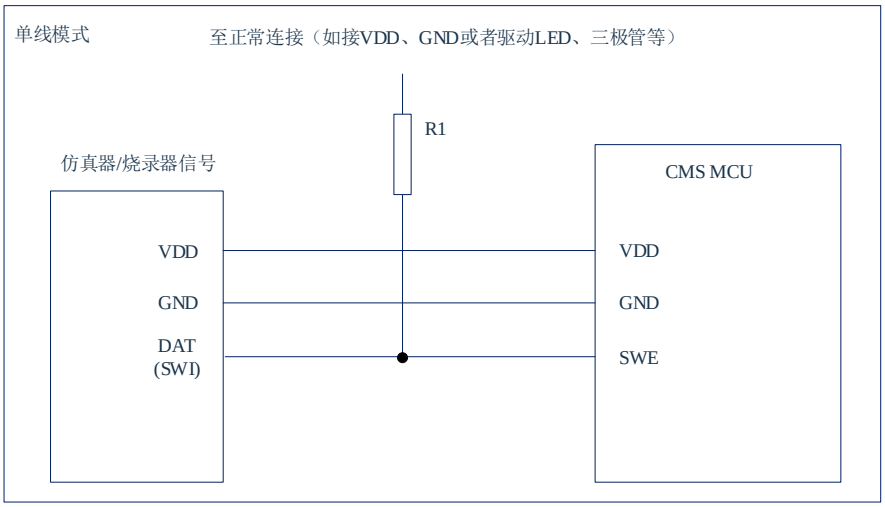
- 电源线
- 接地线
- 数据线
- 时钟线

单线模式：

- 电源线
- 接地线
- 数据时钟线

在线串行编程使得用户可使用未编程的器件制造电路板，仅在产品交付前才对芯片进行编程，从而可以将最新版本的固件或者定制固件烧写到芯片中。典型的在线串行编程连接方法如下图所示：





上图中，R1、R2 为电气隔离器件，常以电阻代替，其阻值如下：R1≥4.7K、R2≥4.7K。

注意在编程和调试时，DSDA 禁止连接下拉电阻。如果实际电路需要接下拉电阻，建议利用跳线结构，在编程/调试时断开下拉电阻，完成之后再接入下拉电阻。

24.2 在线调试模式

芯片支持 2 线（DSCK/DSDA）/单线（SWE）在线调试功能。如果使用在线调试功能，则需要将系统配置寄存器中的 DEBUG 设置为 ENABLE。使用调试模式时，需要注意以下几点：

- 调试状态下，DSCK/DSDA 作为专用调试口，严禁复用成触摸功能，不能实现其 GPIO 及复用功能。
- 调试状态下进入休眠模式/空闲模式，系统电源与振荡器不会停止工作，在该状态下可仿真休眠唤醒功能。如需关注功耗，则建议关闭调试功能后再测试芯片实际的休眠电流。
- 调试状态下暂停，其他功能外设继续运行，WDT，Timer0/1/2/5 计数器会停止。但是如果 Timer1 作为 UART0/1 的波特率产生器，则暂停状态下 Timer1 也会继续运行。暂停状态下继续运行的外设可能会产生中断，调试时需要注意。
- 调试状态下建议不要使用 WDT 复位以及软件复位功能，因为复位时芯片与调试器有可能失去连接。

25. 指令说明

汇编指令总共包括 5 类：算术运算、逻辑运算、数据传送运算、布尔操作和程序分支指令，这些指令全部都与标准 8051 兼容。

25.1 符号说明

符号	说明
Rn	工作寄存器 R0-R7
Direct	内部数据存储器 RAM 的单元地址（00H-FFH）或特殊功能寄存器 SFR 中的地址
@Ri	间接寻址寄存器（@R0 或 @R1）
#data	8 位二进制常数
#data16	在指令中的 16 位二进制常数
Bit	内部数据存储器 RAM 或特殊功能寄存器 SFR 中的位地址
Addr16	16位地址，地址范围0-64KB地址空间
Addr11	11位地址，地址范围0-2KB地址空间
Rel	相对地址
A	累加器

25.2 指令一览表

助记符		描述
运算类		
ADD	A,Rn	累加器加寄存器
ADD	A,direct	累加器加直接寻址单元
ADD	A,@Ri	累加器加间接寻址RAM
ADD	A,#data	累加器加立即数
ADDC	A,Rn	累加器加寄存器和进位标志
ADDC	A,direct	累加器加直接寻址单元和进位标志
ADDC	A,@Ri	累加器加间接寻址RAM和进位标志
ADDC	A,#data	累加器加立即数和进位标志
SUBB	A,Rn	累加器减寄存器和进位标志
SUBB	A,direct	累加器减直接寻址单元和进位标志
SUBB	A,@Ri	累加器减间接寻址RAM和进位标志
SUBB	A,#data	累加器减立即数和进位标志
INC	A	累加器加1
INC	Rn	寄存器加1
INC	direct	直接寻址单元加1
INC	@Ri	间接寻址RAM加1
INC	DPTR	数据指针加1
DEC	A	累加器减1
DEC	Rn	寄存器减1
DEC	direct	直接寻址单元减1
DEC	@Ri	间接寻址RAM减1
MUL	A,B	累加器乘寄存器B
DIV	A,B	累加器除以寄存器B
DA	A	十进制调整
逻辑运算类		
ANL	A,Rn	累加器与寄存器
ANL	A,direct	累加器与直接寻址单元
ANL	A,@Ri	累加器与间接寻址RAM
ANL	A,#data	累加器与立即数
ANL	direct,A	直接寻址单元与累加器
ANL	direct,#data	直接寻址单元与立即数
ORL	A,Rn	累加器或寄存器
ORL	A, direct	累加器或直接寻址单元
ORL	A,@Ri	累加器或间接寻址RAM
ORL	A, #data	累加器或立即数
ORL	direct,A	直接寻址单元或累加器
ORL	direct,#data	直接寻址单元或立即数
XRL	A,Rn	累加器异或寄存器
XRL	A,direct	累加器异或直接寻址单元
XRL	A,@Ri	累加器异或间接寻址RAM
XRL	A,#data	累加器异或立即数
XRL	direct,A	直接寻址单元异或累加器
XRL	direct,#data	直接寻址单元异或立即数
CLR	A	累加器清0
CPL	A	累加器取反
RL	A	累加器左循环移位

助记符	描述
RLC A	累加器连进位标志左循环移位
RR A	累加器右循环移位
RRC A	累加器连进位标志右循环移位
SWAP A	累加器高4位与低4位交换
数据传输类	
MOV A,Rn	寄存器传送到累加器
MOV A,direct	直接寻址单元传送到累加器
MOV A,@Ri	间接寻址RAM送累加器
MOV A,#data	立即数送累加器
MOV Rn,A	累加器送寄存器
MOV Rn,direct	直接寻址单元送寄存器
MOV Rn,#data	立即数送寄存器
MOV direct,A	累加器送直接寻址单元
MOV direct,Rn	寄存器送直接寻址单元
MOV direct1,direct2	直接地址单元传送到直接寻址单元
MOV direct,@Ri	间接寻址RAM送直接寻址单元
MOV direct,#data	立即数送直接寻址单元
MOV @Ri,A	累加器送间接寻址RAM
MOV @Ri,direct	直接寻址单元送间接寻址RAM
MOV @Ri,#data	立即数送间接寻址RAM
MOV DPTR,#data16	16位立即数送数据指针
MOVC A,@A+DPTR	查表数据送累加器（DPTR为基址）
MOVC A,@A+PC	查表数据送累加器（PC为基址）
MOVX A,@Ri	外部RAM单元送累加器（8位地址）
MOVX A,@DPTR	外部RAM单元送累加器（16位地址）
MOVX @Ri,A	累加器送外部RAM单元（8位地址）
MOVX @DPTR,A	累加器送外部RAM单元（16位地址）
PUSH direct	直接寻址单元压入栈顶
POP direct	栈顶弹出直接寻址单元
XCH A,Rn	累加器与寄存器交换
XCH A, direct	累加器与直接寻址单元RAM交换
XCH A,@Ri	累加器与间接寻址单元RAM交换
XCHD A,@Ri	累加器与间接寻址单元RAM交换低4位
布尔运算类	
CLR C	C清零
CLR bit	直接寻址位清零
SETB C	C 置位
SETB bit	直接寻址位置位
CPL C	C 取反
CPL bit	直接寻址位取反
ANL C,bit	C逻辑与直接寻址位
ANL C,/bit	C逻辑与直接寻址位的反
ORL C,bit	C逻辑或直接寻址位
ORL C,/bit	C逻辑或直接寻址位的反
MOV C,bit	直接寻址位送C
MOV bit,C	C送直接寻址位
程序跳转类	
ACALL addr11	2K地址范围内绝对调用
LCALL addr16	64K地址范围内长调用

助记符		描述
RET		子程序返回
RETI		中断返回
AJMP	addr11	2K地址范围内绝对转移
LJMP	addr16	64K地址范围内长转移
SJMP	rel	相对短转移
JMP	@A+DPTR	相对长转移
JZ	rel	累加器为0转移
JNZ	rel	累加器不为0转移
JC	rel	C为1转移
JNC	rel	C为0转移
JB	bit,rel	直接寻址位为1转移
JNB	bit,rel	直接寻址位为0转移
JBC	bit,rel	直接寻址位为1转移，并清该位
CJNE	A,direct,rel	累加器与直接寻址单元不等转移
CJNE	A,#data,rel	累加器与立即数不等转移
CJNE	Rn,#data,rel	寄存器与立即数不等转移
CJNE	@Ri,#data,rel	间接寻址单元RAM与立即数不等转移
DJNZ	Rn,rel	寄存器减1不为0转移
DJNZ	direct,rel	直接寻址单元减1不为0转移
NOP		空指令
读取—修改—写入指令（Read-Modify-Write）		
ANL		逻辑 (ANL direct, A 与 ANL direct, #data)
ORL		逻辑或(ORL direct, A 与ORL direct, #data)
XRL		逻辑异或 (XRL direct, A 与 XRL direct, #data)
JBC		直接寻址位为1转移，并清该位 (JBC bit, rel)
CPL		取反 (CPL bit)
INC		加1 (INC direct)
DEC		减1. (DEC direct)
DJNZ		减1不为0转移(DJNZ direct, rel)
MOV	bit,C	C送直接寻址位
CLR	bit	直接寻址位清零
SETB	bit	直接寻址位置位

26. 版本修订说明

版本号	时间	修改内容
V0.5.0	2025 年 3 月	初始版本
V0.5.1	2025 年 3 月	删除 SFR 表多余寄存器，调整 TIMER2 结构框图
V0.5.2	2025 年 4 月	删除 XSFR 表多余寄存器
V0.5.3	2025 年 4 月	删除冗余寄存器，部分内容修改
	2025 年 7 月	1) 修改 8.1.2 描述 2) 修改 1.7 描述 3) 修改 6.5.2.4 描述
V0.5.4	2025 年 7 月	1) 修改 18.1 描述 2) 修改 24.1 章节，新增单线描述 3) 修改第 3 章，消除上电复位和低压检测复位描述的歧义 4) 修改 19.4.1 描述 5) 修改 7.1 端口结构
V0.5.5	2025 年 10 月	1) 修改 4.1 结构框图 2) 修改 6.1 格式 3) 修改 17.2 描述。 4) 新增 17.4 I2C 传输速度模式说明
V0.5.6	2025 年 12 月	1) 修改 5.1.2 LVDCON 寄存器中断描述 2) 修改 19.2.4 描述