

HYM8F028C6

8 位 **FLASH** 微控制器用户手册

西安华羿微电子股份有限公司

2016 年 5 月 18 日

1. 概述.....	7
1.1. 简介.....	7
1.2. 开发工具.....	7
1.2.1. 编译器.....	7
1.2.2. 仿真器.....	7
1.2.3. 程序烧录器.....	7
1.3. 特性.....	8
2. 引脚分配.....	11
3. 中央处理器结构.....	14
3.1. 性能.....	14
3.2. 编程和调试支持.....	14
3.3. 指令集.....	16
4. 存储器.....	23
4.1. 程序存储器 (Program Memory).....	23
4.2. 数据存储器 (Data Memory).....	23
4.3. EEPROM.....	24
4.4. 特殊功能寄存器分布.....	24
4.5. 特殊功能寄存器地址表.....	24
4.6. 编译器兼容的 SFR.....	26
4.6.1. 累加器 (Accumulator) - ACC.....	26
4.6.2. B 寄存器 - B.....	26
4.6.3. 程序状态寄存器 - PSW.....	26
4.6.4. 堆栈指针寄存器 - SP.....	27
4.6.5. 数据指针 - DPH, DPL.....	27
4.6.6. 数据指针 1 - DPH1, DPL1.....	28
4.6.7. 数据指针选择寄存器 - DPS.....	28
4.6.8. 数据指针控制寄存器 - DPC.....	28
4.6.9. 程序存储器页选寄存器 - PAGESEL.....	29
4.6.10. 数据存储器页选寄存器 - D_PAGESEL.....	29
5. 系统时钟.....	31
5.1. 特性.....	31
5.2. 系统时钟寄存器.....	31
6. 中断.....	33
6.1. 中断向量表.....	33
6.2. 中断控制寄存器.....	34
7. MDU.....	42
7.1. MDU 结构图.....	42
7.2. MDU 操作.....	43
7.3. MDU 寄存器.....	44
7.3.1. 乘除寄存器 - MD0, MD1, MD2, MD3, MD4, MD5.....	44
7.3.2. 运算控制寄存器 - ARCON.....	45

7.3.3. 例程.....	45
8. 输入输出端口.....	49
9. 定时/计数器.....	60
9.1. 定时/计数器 0.....	60
9.1.1. 模式 0 和模式 1.....	60
9.1.2. 模式 2.....	60
9.1.3. 模式 3.....	61
9.2. 定时/计数器 1.....	61
9.2.1. 模式 0 和模式 1.....	61
9.2.2. 模式 2.....	61
9.2.3. 模式 3.....	62
9.2.4. 定时器 0 和定时器 1 寄存器.....	62
9.3. 定时/计数器 2.....	65
9.3.1. 定时模式.....	65
9.3.2. 事件计数模式.....	65
9.3.3. 门控定时模式.....	66
9.3.4. 定时器 2 重载.....	66
9.3.5. 比较功能.....	66
9.3.6. 捕捉功能.....	68
9.3.7. 定时器 2 寄存器.....	69
9.4. 独立脉冲宽度调制器 PWM0.....	72
9.4.1. PWM0 计数时钟.....	72
9.4.2. PWM0 工作模式.....	72
9.4.3. PWM0 计数模式.....	72
9.4.4. PWM0 更新方式.....	74
9.4.5. PWM0 中断.....	75
9.4.6. PWM0 控制寄存器.....	76
9.5. 独立脉冲宽度调制器 PWM1、PWM2.....	79
9.5.1. PWM1 控制寄存器.....	79
9.5.2. PWM2 控制寄存器.....	81
9.6. 看门狗 (Watch-dog)	83
9.6.1. 看门狗寄存器.....	84
10. 变频马达控制功能.....	86
10.1. 脉冲宽度调制定时器 (MPWM)	86
10.1.1. MPWM 计数时钟.....	86
10.1.2. MPWM 工作模式.....	87
10.1.3. MPWM 计数模式.....	87
10.1.4. MPWM 更新方式.....	88
10.1.5. MPWM 中断.....	88
10.2. 消隐.....	89
10.3. 死区插入电路.....	90
10.4. 极性电路.....	90
10.5. MPWM 的控制寄存器.....	90
11. 线性模拟电路.....	100

11.1. 结构图.....	100
11.2. 高速比较器.....	101
11.3. 可编程增益放大 PGA.....	102
11.3.1. 工作模式.....	102
11.4. 数模转换器 DAC.....	109
12. 模数 (AD) 转换器.....	110
12.1. 自动扫描.....	110
12.2. 双通道采样.....	110
12.3. ADC 控制.....	110
12.4. ADC 寄存器.....	111
13. 异步串行通讯口 (UART0、UART1).....	115
13.1. UART0.....	115
13.1.1. UART0 结构图.....	115
13.1.2. 功能.....	115
13.1.3. UART0 寄存器.....	119
13.2. UART1.....	121
13.2.1. UART1 结构图.....	121
13.2.2. 功能.....	122
13.2.3. 本地互连网络 (LIN)	123
13.2.4. UART1 寄存器.....	126
14. 同步串行通讯口 (SPI)	129
14.1. SPI 寄存器.....	129
15. EEPROM 和 FLASH 读写, 程序加密.....	134
15.1. EEPROM 编程寄存器.....	134
15.2. FLASH 擦写.....	135
15.3. 程序加密.....	135
15.4. EEPROM 擦写例程.....	135
15.5. FLASH 擦写例程.....	136
16. RESET.....	137
16.1. 复位说明.....	137
16.2. 上电复位.....	137
16.3. 掉电复位 (LVR)	137
16.4. 外部复位.....	138
16.5. 看门狗复位.....	138
16.6. 软件复位.....	138
16.7. 软件复位寄存器.....	139
17. 电源模式.....	140
17.1. 空闲模式.....	140
17.2. 停止模式.....	140
17.3. 电源模式控制寄存器.....	141
18. 电气特性.....	142
18.1. 极限值.....	142

18.2. 推荐操作参数.....	142
18.3. 直流特性.....	143
18.4. 交流特性.....	144
18.5. 模数 (AD) 转换器特性.....	144
18.6. (LVR) 低电压复位特性.....	145
18.7. 模拟比较器特性.....	145
18.8. 可编程增益放大器 (PGA) 特性.....	146
18.9. DAC 特性.....	146
18.10. 内部 RC 振荡器特性.....	147
18.11. PLL 特性.....	147
18.12. UART 特性.....	147
18.13. 内部 Flash Rom 特性.....	148
19. 封装尺寸.....	149

1. 概述

1.1. 简介

HYM8F028C6 内核是 8051XC2，8051XC2 与 MCS-51TM 指令集完全兼容可以使用标准 803x/805x 的汇编器和编译器进行软件开发该系列 MCU，8051XC2 还包含片内调试硬件和与 MCU 直接接口的模拟和数字子系统在一个集成电路内提供了完整的数据采集或控制系统解决方案。

HYM8F028C6 内核除了具有标准 8051 的组织结构和外设以外另有增加的定制外设和功能，大大增强了它的处理能力。

主要应用：

- 电动自行车
- 抽油烟机变频马达控制
- 变频电风扇
- 工业设备的泵控制
- 洗衣机、干衣机
- 冰箱压缩泵、风机
- 空调压缩泵、风机
- 空气净化器风机

1.2. 开发工具

1.2.1. 编译器

HYM8F028C6 是基于 8051 内核的 MCU。同时，存储器大小 $\leq 64K$ 字节。开发人员可以使用所有第三方各种标准 8051 编译器。如 KEIL，SDCC 等。

1.2.2. 仿真器

仿真器支持在线目标调试。可以直接下载软件到目标系统和调试前下载的软件。

1.2.3. 程序烧录器

程序烧录器支持批量脱机烧写。

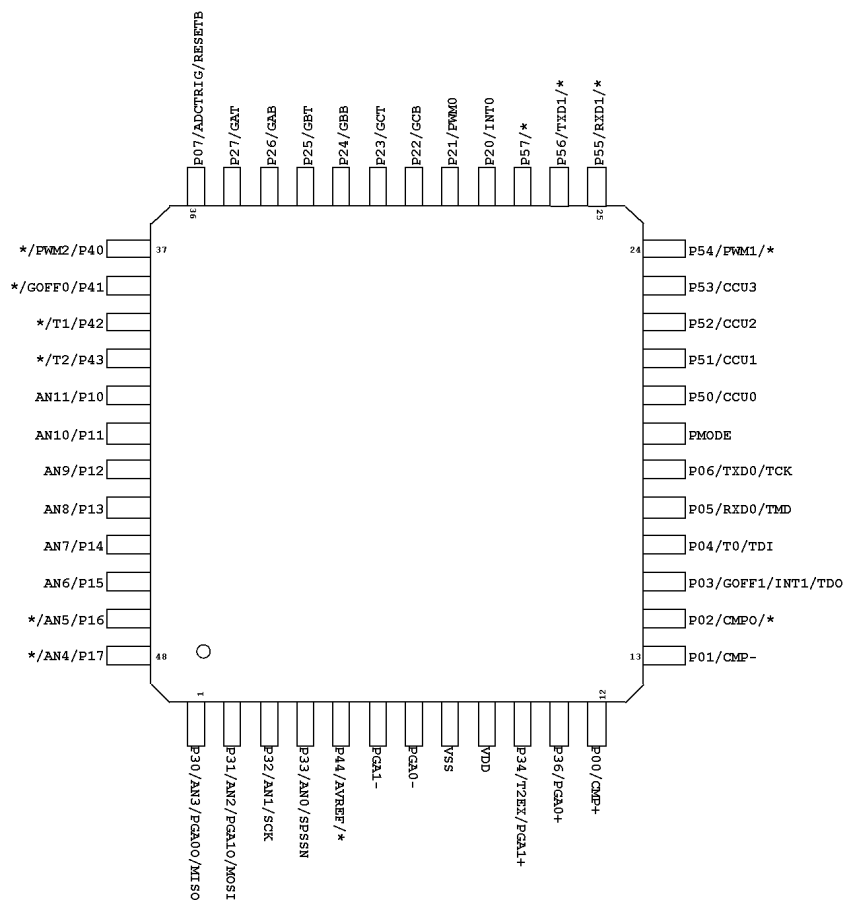
1.3. 特性

- **CPU**
 - 8 位 CISC 内核 (兼容 8051)
 - 1 时钟指令执行周期
 - 时钟最高达到: 60MHz (运行 RAM 中程序)
20MHz (运行 Flash 中程序)
- **存储器**
 - 32k 字节 单电源 Flash
 - Flash 可用户程序编程、可禁止整个芯片擦除、禁止 Flash 块擦除、禁止写入
 - 片上调试 (OCDS)
 - 20,000 次擦除
 - 256 字节内部 RAM
 - 3072 字节外部 RAM (可执行用户程序)
 - 2048 字节 (共 16 个扇区, 每个扇区 128 个字节) EEPROM
- **乘除功能**
 - 无符号 16 x 16-bit (1 个时钟周期)
 - 符号 16 x 16-bit (1 个时钟周期)
 - 无符号 32/16 (8 个时钟周期)
 - 32-bit 左右移位、规范化 (1 个时钟周期)
- **通用输入输出端口 (GPIO)**
 - 43 个通用 I/O 端口: P0[7:0]、P1[7:0]、P2[7:0]、P3[6,4:0]、P4[4:0]、P5[7:0]
 - 输入电平: 斯密特触发器 CMOS、TTL 兼容
 - P0 有上拉电阻、下拉电阻、开漏选择
 - P1、P2、P3、P4 只有上拉电阻选择
 - 输出驱动电流: $I_{Source} = 12\text{ mA}$ ($0.8V_{DD}$); $I_{Sink} = 12\text{ mA}$ ($0.1V_{DD}$)
- **通用定时器、计数器**
 - 2 个兼容 8051 的 16 位定时器 (T0、T1)
 - 一个 16 位定时器 (T2), 带 CCU 功能。
 - 一路独立 16 位 PWM, 可设置占空比和周期, 锯齿波调制 (边沿对齐模式)、三角波调制 (中心对齐模式)。
 - 两个 8 位 PWM, 可设置占空比和周期, 锯齿波调制 (边沿对齐模式)、三角波调制 (中心对齐模式)。
- **看门狗定时器 (WDT)**
 - 15Bit x 1ch
 - 系统时钟, PLL 输出时钟或内部 RC 振荡器 500kHz。
- **三路互补输出 PWM 定时器**
 - 输入时钟频率最高: 60MHz (500k PLL)
 - 16 位分辨率的 PWM 信号, 可以设置周期、占空比。
 - 可进行 0% - 100% 占空比输出
 - 三角波调制 (中心对齐模式)
 - 死区时间设定
 - 启动 AD 转换功能

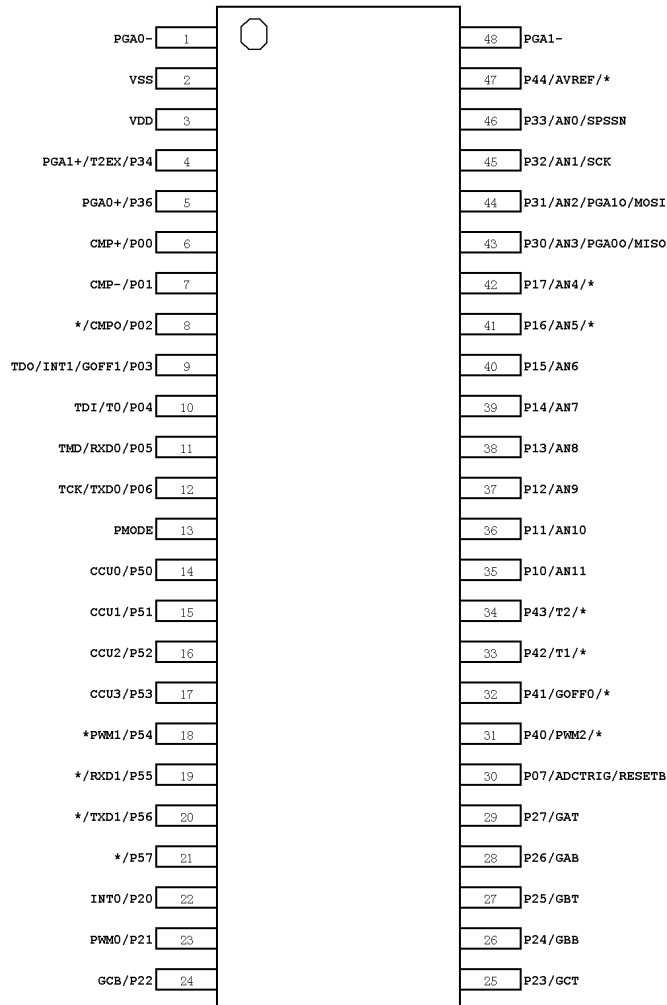
- PWM 一个周期启动同步消隐延时时间设定
- 模拟比较器输出可强制停止 PWM, 并把相应的 IO 口设置为高阻
- 互补输出每个端口可正反向设置
- 互补输出每个端口可设置高端 PWM 输出、低端 PWM 输出、高低端同时 PWM 输出
- 占空比带 Buffer, Buffer 可用户程序更新。
- 在 PWM 比较中可产生中断, 可启动 AD 转换。
- **UART**
- UART0:
 - 同步模式, 固定波特率
 - 8-bit 传送模式, 可变波特率
 - 9-bit 传送模式, 可变或固定波特率
- UART1:
 - 8 位或 9 位传送模式, 可变波特率, 带接收缓存
 - LIN
- **12 Bit A/D Converter**
- 12 输入通道外部引脚输入, 两路 PGA 输入, 内部基准
- AD 基准: VDD、片外 AD_VREF
- 转换速度 1 μ s
- 可设置 4 个通道自动扫描转换把结果存入缓存, 并设置完成标志。
- 双通道同时采样。
- PWM 采样信号启动, 可软件启动
- 输出结果左、右对齐功能。
- 模拟外设
- 高速模拟比较器
 - 1 个
 - 输入负端可接外部输入电压、内部产生的基准电压 (DAC 输出作为基准电压)
 - 输入正端可接外部输入电压、可编程增益放大器输出端。
 - 输出上升沿信号 (有消隐延时) 可以停止 PWM 并设置 PWM 互补输出端口为高阻、可产生中断
- 2 个可编程增益放大器
 - 单端输出可连接到 ADC 输入或高速比较器的正输入端。
 - 输入信号范围: GND ~ AVDD-1.2V
 - 增益: $\times 1$ 、 $\times 2$ 、 $\times 4$ 、 $\times 6$ 、 $\times 8$ 、 $\times 10$ 、 $\times 12$ 、 $\times 14$ 、 $\times 16$
- 5 位 DA 输出提供给比较器作为基准输入。
- 上电复位 (**Power On Reset**)
- 复位响应电压 1.8 V Typical
- 低电压复位 (**Low Voltage Reset**)
- 多档低电压检测: 2.5V - 4.5V 按 0.25V 分档
- 中断源
- 外部中断 (EINT0~1)
- Timer(0/1/2)
- PWM0
- WDT

- UART
- ADC
- MPWM
- 比较器
- GPIO
- SPI
- 内部 **RC** 振荡器
- 内部 RC 振荡器频率： 内部 500kHz 通过 PLL 达到 60MHz±2%
($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$) CPU:
60MHz (运行 RAM 中程序)
20MHz (运行 Flash 中程序)
外围: 60MHz
- 低功耗模式 (**Power Down Mode**)
- STOP, IDLE mode
- 片内 **OCDS**
- 片内调试电路提供全速非侵入式的在系统调试不需仿真器
- 支持断点单步观察点堆栈监视器可以观察/修改存储器和寄存器
- PGM 引脚用于 OCD 模式和正常运行模式控制
- 廉价的开发工具
- 工作电压范围: 2.7 V $\sim$ 5.5V
- 工作温度范围: - 40 $^{\circ}\text{C} \sim$ + 125 $^{\circ}\text{C}$
- 振荡器类型
- 内部 RC 振荡器 (500kHz)
- 封装
- LQFP48 (7×7)、SSOP48

2. 引脚分配



* 可产生GPIO中断



* 可产生GPIO中断

V _{DD} 、V _{SS}	电源
RESETB	复位，内部弱上拉
PMODE	编程使能，内部弱上拉，低有效，建议外部接 1k 上拉。 当 PMODE 接地时，mcu 处于调试模式，此时 P03, P04, P05, P06, P07 不能用于普通 I/O 口；同时 P50, P51, P52 也不能用于普通 I/O 口。注意，此时 P07, P50, P51, P52 不能接地，否则，程序不能下载。只有 PMODE 为低，才可下载程序。 只有当 PMODE 没有接地（拉高或悬空）时，MCU 处于运行模式，此时 P03, P04, P05, P06, P07, P50, P51, P52 可以用于普通 I/O 口模式。
P0(7:0)、P1(7:0)	数字端口，P0 可选择上拉电阻、下拉电阻、开漏输出；其他端口只能选择上拉电阻。 其中 P02、P16、P17、P32、P33、P34、P40、P41、P54、P57 可产生 GPIO 中断。
P2(7:0)、P3(6、4:0)	
P4(4:0)、P5[7:0]	
AN0-AN11	AD 输入
RXD0、TXD0、RXD1、TXD1	UART0 和 UART1 端口
T0、T1、T2EX	Timer0~Timer2 外部时钟输入端口
CCU0、CCU1、CCU2、CCU3	CCU 输入、输出端口
CMP+	模拟比较器正输入
CMP-	模拟比较器负输入
CMPO	模拟比较器输出
PGA0+、PGA0-	可编程增益放大器 0 正反输入
PGA0_O	PGA0 输出
PGA1+、PGA1-	可编程增益放大器 1 正反输入
PGA1_O	PGA1 输出
GAT、GBT、GCT	三相高边驱动输出端口
GAB、GBB、GCB	三相低边驱动输出端口
GOFF0, GOFF1	使三相高低边驱动输出端口高阻，高电平有效。
INT0、INT1	外部中断
PWM0、PWM1、PWM2	独立 PWM 输出
MISO、MOSI、SCK、SS0	SPI 端口
TDI、TMS、TCK、TDO	JTAG 端口
ADC_VREF	ADC 基准输入

3. 中央处理器结构

MCU 系统控制器的内核是 8051XC2，微控制器 8051XC2 与 MCS-51™ 指令集完全兼容可以使用标准 803x/805x 的汇编器和编译器进行软件开发该系列 MCU，8051XC2 还包含片内调试硬件和与 MCU 直接接口的模拟和数字子系统在一个集成电路内提供了完整的数据采集或控制系统解决方案。

8051XC2 微控制器内核除了具有标准 8051 的组织结构和外设以外另有增加的定制外设和功能，大大增强了它的处理能力。

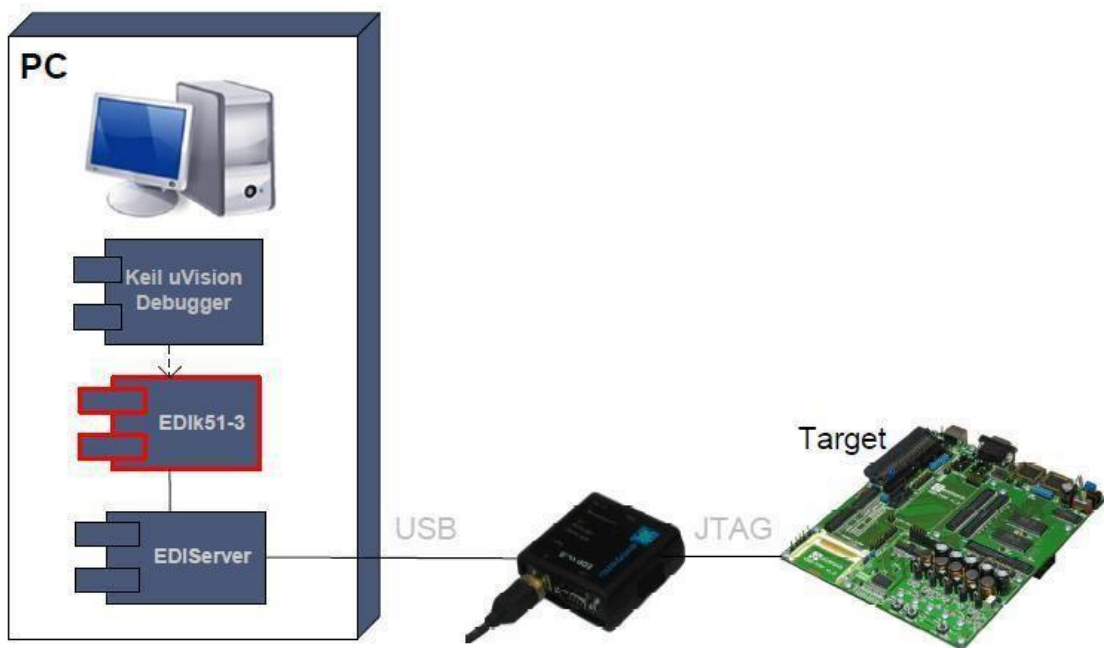
3.1. 性能

8051XC2 采用流水线结构，与标准的 8051 结构相比指令执行速度有很大的提高。8051XC2 内核 70% 的指令的执行时间为 1 或 2 个系统时钟周期，没有执行时间超过 8 个系统时钟周期的指令。8051XC2CIP-51 工作在最大系统时钟频率 60 MHz 时，它的峰值速度达到 60MIPS。8051XC2 共有 111 条指令下表列出了指令条数与执行时所需的系统时钟周期数的关系。

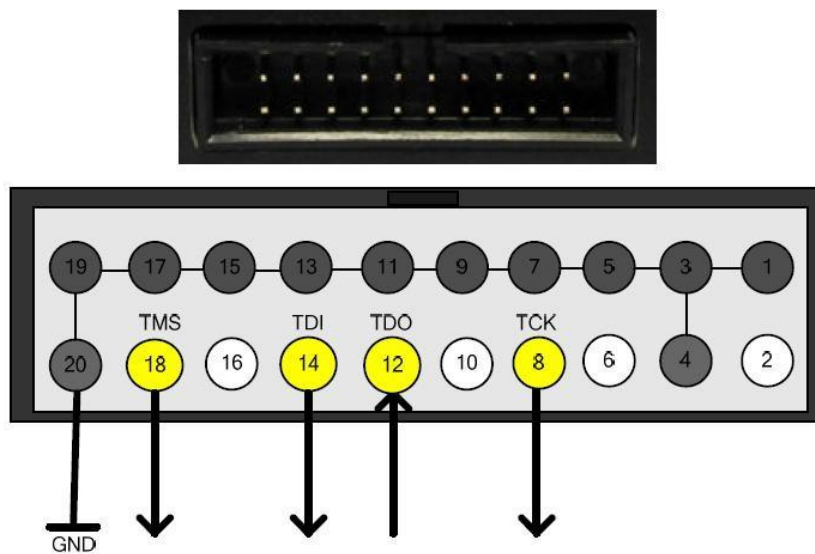
3.2. 编程和调试支持

本 MCU 提供了 JTAG 串行接口，通过该接口能对 FLASH 程序存储器进行在线编程，并可与片内调试电路通信。应用程序可以使用 MOV_C 和 MOV_X 指令对可编程 FLASH 进行读写，每次只能读写一个字节。这一特性允许将程序存储器用于非易失性数据存储，以及在软件控制下更新程序代码。

片内 JTAG 调试电路支持全速在线调试，单步，全速，查看资源断点等操作。



调试系统连接图



调试器 JTAG 连接器引脚分配

3.3. 指令集

所有 R8051XC2 指令都是二进制码兼容，并执行与工业标准 8051 相同的功能，下面表格给出了 R8051XC2 微控制器内核的指令周期。

数据寻址模式助记符

符号	说明
Rn	寄存器 R0-R7
direct	内部 128 字节 RAM 或任何的特殊功能寄存器(SFR)
@Ri	通过 R0 或 R1 间接访问的内部 RAM 或外部 RAM
#data	指令中 8 位立即操作数常量
#data 16	指令中 16 位立即操作数常量
bit	内部 RAM 中 128 个可位寻址的位或任何可位寻址的特殊寄存器(SFR)中的位, 包括 I/O 脚和状态寄存器中状态位。
A	累加器

程序寻址模式助记符

addr16	用于 LCALL or LJMP 的目标地址，能访问 64k 字节程序存储空间中的任何地方。
addr11	用于 ACALL or AJMP 的目标地址,能访问 2k 字节程序存储空间内的地方，此 2k 程序空间是指这两个指令目标地址开始的空间。
rel	SJMP 和所有条件转移指令(包括一个 8 位偏移量)寻址范围为相对于这些指令的 +127/-128。

下面表格显示指令的十六进制代码，字节和机器周期，每个指令需要被执行的周期数。注意周期数指无程序存储器的等待状态时的值。

算术运算指令

助记符	说明 on	操作码	字节数	周期数
ADD A,Rn	累加器加寄存器	0x28-0x2F	1	1
ADD A,direct	累加器加直接寻址单元	0x25	2	2
ADD A,@Ri	累加器加寄存器间接寻址单元	0x26-0x27	1	2
ADD A,#data	累加器加立即数	0x24	2	2
ADDC A,Rn	累加器加寄存器和进位标志	0x38-0x3F	1	1
ADDC A,direct	累加器加直接寻址单元和进位标志	0x35	2	2
ADDC A,@Ri	累加器加寄存器间接寻址单元和进位标志	0x36-0x37	1	2
ADDC A,#data	累加器加立即数和进位标志	0x34	2	2
SUBB A,Rn	累加器减寄存器和进位标志	0x98-0x9F	1	1
SUBB A,direct	累加器减直接寻址单元和进位标志	0x95	2	2
SUBB A,@Ri	累加器减寄存器间接寻址单元和进位标志	0x96-0x97	1	2
SUBB A,#data	累加器减立即数和进位标志	0x94	2	2

INCA	累加器加 1	0x04	1	1
INC Rn	寄存器加 1	0x08-0x0F	1	1
INC direct	直接寻址单元加 1	0x05	2	2
INC @Ri	寄存器间接寻址单元加 1	0x06-0x07	1	2
INC DPTR	数据指针加 1	0xA3	1	1
DECA	累加器减 1	0x14	1	1
DEC Rn	寄存器减 1	0x18-0x1F	1	1
DEC direct	直接寻址单元减 1	0x15	2	2
DEC @Ri	寄存器间接寻址单元减 1	0x16-0x17	1	2
MUL AB	累加器乘寄存器 B	0xA4	1	4
DIV	累加器除寄存器 B	0x84	1	4
DA A	累加器十进制调整	0xD4	1	1

逻辑运算

助记符	说明	操作码	字节数	周期数
ANL A,Rn	累加器与寄存器	0x58-0x5F	1	1
ANL A,direct	累加器与直接寻址单元	0x55	2	2
ANL A,@Ri	累加器与寄存器间接寻址单元	0x56-0x57	1	2
ANL A,#data	累加器与立即数	0x54	2	2
ANL direct,A	直接寻址单元与累加器	0x52	2	2
ANL direct,#data	直接寻址单元与立即数	0x53	3	3
ORL A,Rn	累加器或寄存器	0x48-0x4F	1	1
ORL A,direct	累加器或直接寻址单元	0x45	2	2
ORL A,@Ri	累加器或寄存器间接寻址单元	0x46-0x47	1	2
ORL A,#data	累加器或立即数	0x44	2	2
ORL direct,A	直接寻址单元或累加器	0x42	2	2
ORL direct,#data	直接寻址单元或立即数 location	0x43	3	3
XRL A,Rn	累加器异或寄存器	0x68-0x6F	1	1
XRL A,direct	累加器异或直接寻址单元	0x65	2	2
XRL A,@Ri	累加器异或寄存器间接寻址单元	0x66-0x67	1	2
XRL A,#data	累加器异或立即数	0x64	2	2
XRL direct,A	直接寻址单元异或累加器	0x62	2	2
XRL direct,#data	直接寻址单元异或立即数	0x63	3	3
CLR A	清累加器	0xE4	1	1
CPL A	累加器取反	0xF4	1	1
RL A	累加器循环左移	0x23	1	1
RLC A	累加器带进位循环左移	0x33	1	1
RR A	累加器循环右移	0x03	1	1
RRC A	累加器带进位循环右移	0x13	1	1
SWAP A	累加器高 4 位与低 4 位互换	0xC4	1	1

数据传送

助记符	说明 on	操作码	字节数	周期数
MOV A,Rn	寄存器送累加器	0xE8-0xEF	1	1
MOV A,direct	直接寻址单元送累加器	0xE5	2	2
MOV A,@Ri	寄存器间接寻址单元送累加器	0xE6-0xE7	1	2
MOV A,#data	立即数送累加器	0x74	2	2
MOV Rn,A	累加器送寄存器	0xF8-0xFF	1	1
MOV Rn,direct	直接寻址单元送寄存器	0xA8-0xAF	2	2
MOV Rn,#data	立即数送寄存器	0x78-0x7F	2	2
MOV direct,A	累加器送直接寻址单元	0xF5	2	2
MOV direct,Rn	寄存器送直接寻址单元	0x88-0x8F	2	2

MOV direct1,direct2	送直接寻址单元	0x85	3	3
MOV direct,@Ri	寄存器间接寻址单元送直接寻址单元	0x86-0x87	2	2
MOV direct,#data	立即数送直接寻址单元	0x75	3	3
MOV @Ri,A	累加器送寄存器间接寻址单元	0xF6-0xF7	1	1
MOV @Ri,direct	直接寻址单元送寄存器间接寻址单元	0xA6-0xA7	2	2
MOV @Ri,#data	立即数送寄存器间接寻址单元	0x76-0x77	2	2
MOV DPTR,#data16	16 位立即数送数据指针	0x90	3	3
MOVC A,@A+DPTR	查表数据送累加器 (DPTR 为基址)	0x93	1	3
MOVC A,@A+PC	查表数据送累加器 (PC 为基址)	0x83	1	3
MOVX A,@Ri	¹⁾ 外部 RAM 单元送累加器 (8 位地址)	0xE2-0xE3	1	3-10
MOVX A,@DPTR	¹⁾ 外部 RAM 单元送累加器 (16 位地址)	0xE0	1	3-10
MOVX @Ri,A	¹⁾ 累加器送外部 RAM 单元 (8 位地址)	0xF2-0xF3	1	3-12
MOVX @DPTR,A	¹⁾ 累加器送外部 RAM 单元 (16 位地址)	0xF0	1	3-12
PUSH direct	直接寻址单元进栈	0xC0	2	2
POP direct	直接寻址单元出栈	0xD0	2	2
XCH A,Rn	累加器和寄存器互换	0xC8-0xCF	1	1
XCH A,direct	累加器和直接寻址单元互换	0xC5	2	2
XCH A,@Ri	累加器和寄存器间接寻址单元互换	0xC6-0xC7	1	2
XCHD A,@Ri	间接寻址寄存器低 4 位和累加器互换	0xD6-0xD7	1	2

注意：¹⁾ MOVX 指令执行两种操作取决于“PMW”位 (PCON.4) 状态 (PCON.4)。

跳转指令

助记符	说明	操作码	字节数	周期数
ACALL addr11	2KB 范围内绝对子程序调用	xxx10001b	2	2
LCALL addr16	长子程序调用	0x12	3	3
RET	子程序返回	0x22	1	4
RETI	中断返回	0x32	1	4
AJMP addr11	绝对跳转	xxx00001b	2	2
LJMP addr16	长跳转	0x02	3	3
SJMP rel	短相对寻址跳转	0x80	2	2
JMP @A+DPTR	数据指针间接相对寻址跳转	0x73	1	3
JZ rel	累加器为零跳转	0x60	2	3
JNZ rel	累加器非零跳转	0x70	2	3
JC rel	进位为 1 跳转	0x40	2	3
JNC	进位为 0 跳转	0x50	2	3
JB bit,rel	直接位寻址为 1 跳转	0x20	3	4
JNB bit,rel	直接位寻址为 0 跳转	0x30	3	4
JBC bit,rel	间接位寻址为 1 跳转并且清	0x10	3	4
CJNE A,direct,rel	累加器与直接寻址数据不相等跳转	0xB5	3	4
CJNE A,#data,rel	累加器与立即数不相等跳转	0xB4	3	4
CJNE Rn,#data,rel	寄存器与立即数不相等跳转	0xB8-0xBF	3	4
CJNE @Ri,#data,rel	寄存器间接寻址单元与立即数不相等跳转	B6-B7	3	5
DJNZ Rn,rel	寄存器减 1 不为零跳转	D8-DF	2	3

DJNZ direct,rel	直接寻址单元减 1 不为零跳转	D	3	4
NOP	空操作	0	1	1

布尔操作

助记符	说明	操作码	字节数	周期数
CLR C	清进位标志	0xC3	1	1
CLR bit	清位寻址位标志	0xC2	2	2
SETB C	进位置 1	0xD3	1	1
SETB bit	位寻址位置 1	0xD2	2	2
CPL C	进位取反	0xB3	1	2
CPL bit	位寻址位取反	0xB2	2	2
ANL C,bit	进位与位寻址位	0x82	2	2
ANL C,/bit	进位与位寻址位的反码	0xB0	2	2
ORL C,bit	进位或位寻址位	0x72	2	2
ORL C,/bit	进位或位寻址位反码 carry	0xA0	2	2
MOV C,bit	位寻址位送进位	0xA2	2	2
MOV bit,C	进位送位寻址位	0x92	2	2

每个指令的持续时间可以使用下面的公式来计算。

如果字节大于 1 或周期=1，那么

$$\text{执行时间} = \text{周期数} + (\text{字节数} + R) \cdot P + X \cdot D$$

其他情况

$$\text{执行时间} = \text{周期数} + (2 + R) \cdot P + X \cdot D$$

- 其中：
- 字节数为指令的字节数（见上表）
 - 周期数为无等待状态的周期数量（见上表）
 - 如果是 MOVC 指令，则 $R = 1$ ，否则 $R = 0$
 - 如果是 MOVX 指令，则 $X = 1$ ，否则 $X = 0$
 - P = 程序存储器的等待状态数（=CKCON[6:4]）
 - D = 数据存储器的等待状态数（=CKCON[2:0]）。

在程序存储器写入模式 (PMW) 时，MOVX 指令的执行时间如下：

$$\text{执行时间} = \text{周期数} + (2 + X) \cdot P$$

操作码十六进制排序表

操作码	助记符	操作码	助记符
00 H	NOP	80 H	SJMP rel
01 H	AJMP addr11	81 H	AJMP addr11
02 H	LJMP addr16	82 H	ANL C,bit
03 H	RR A	83 H	MOVCA,@A+PC
04 H	INC A	84 H	DIV AB
05 H	INC direct	85 H	MOV direct,direct
06 H	INC @R0	86 H	MOV direct,@R0

07 H	INC @R1	87 H	MOV direct,@R1
08 H	INC R0	88 H	MOV direct,R0
09 H	INC R1	89 H	MOV direct,R1
0A H	INC R2	8A H	MOV direct,R2
0B H	INC R3	8B H	MOV direct,R3
0C H	INC R4	8C H	MOV direct,R4
0D H	INC R5	8D H	MOV direct,R5
0E H	INC R6	8E H	MOV direct,R6
0F H	INC R7	8F H	MOV direct,R7
10 H	JBC bit,rel	90 H	MOV DPTR,#data16
11 H	ACALL addr11	91 H	ACALL addr11
12 H	LCALL addr16	92 H	MOV bit,C
13 H	RRC A	93 H	MOVCA,@A+DPTR
14 H	DEC A	94 H	SUBB A,#data
15 H	DEC direct	95 H	SUBB A,direct
16 H	DEC @R0	96 H	SUBB A,@R0
17 H	DEC @R1	97 H	SUBB A,@R1
18 H	DEC R0	98 H	SUBB A,R0
19 H	DEC R1	99 H	SUBB A,R1
1A H	DEC R2	9A H	SUBB A,R2
1B H	DEC R3	9B H	SUBB A,R3
1C H	DEC R4	9C H	SUBB A,R4
1D H	DEC R5	9D H	SUBB A,R5
1E H	DEC R6	9E H	SUBB A,R6
1F H	DEC R7	9F H	SUBB A,R7
20 H	JB bit,rel	A0 H	ORL C,/bit
21 H	AJMP addr11	A1 H	AJMP addr11
22 H	RET	A2 H	MOV C,bit
23 H	RL A	A3 H	INC DPTR
24 H	ADD A,#data	A4 H	MUL AB
25 H	ADD A,direct	A5 H	-
26 H	ADD A,@R0	A6 H	MOV @R0,direct
27 H	ADD A,@R1	A7 H	MOV @R1,direct
28 H	ADD A,R0	A8 H	MOV R0,direct
29 H	ADD A,R1	A9 H	MOV R1,direct
2A H	ADD A,R2	AA H	MOV R2,direct
2B H	ADD A,R3	AB H	MOV R3,direct
2C H	ADD A,R4	AC H	MOV R4,direct
2D H	ADD A,R5	AD H	MOV R5,direct
2E H	ADD A,R6	AE H	MOV R6,direct
2F H	ADD A,R7	AF H	MOV R7,direct
30 H	JNB bit,rel	B0 H	ANL C,/bit
31 H	ACALL addr11	B1 H	ACALL addr11
32 H	RETI	B2 H	CPL bit
33 H	RLC A	B3 H	CPL C
34 H	ADDC A,#data	B4 H	CJNE A,#data,rel
35 H	ADDC A,direct	B5 H	CJNE A,direct,rel
36 H	ADDC A,@R0	B6 H	CJNE @R0,#data,rel
37 H	ADDC A,@R1	B7 H	CJNE @R1,#data,rel
38 H	ADDC A,R0	B8 H	CJNE R0,#data,rel
39 H	ADDC A,R1	B9 H	CJNE R1,#data,rel
3A H	ADDC A,R2	BA H	CJNE R2,#data,rel
3B H	ADDC A,R3	BB H	CJNE R3,#data,rel
3C H	ADDC A,R4	BC H	CJNE R4,#data,rel

3D H	ADDC A,R5	BD H	CJNF R5,#data,rel
3E H	ADDC A,R6	BE H	CJNE R6,#data,rel
3F H	ADDC A,R7	BF H	CJNE R7,#data,rel
40 H	JC rel	C0 H	PUSH direct
41 H	AJMP addr11	C1 H	AJMP addr11
42 H	ORL direct,A	C2 H	CLR bit
43 H	ORL direct,#data	C3 H	CLR C
44 H	ORL A,#data	C4 H	SWAP A
45 H	ORL A,direct	C5 H	XCH A,direct
46 H	ORL A,@R0	C6 H	XCH A,@R0
47 H	ORL A,@R1	C7 H	XCH A,@R1
48 H	ORL A,R0	C8 H	XCH A,R0
49 H	ORL A,R1	C9 H	XCH A,R1
4A H	ORL A,R2	CA H	XCH A,R2
4B H	ORL A,R3	CB H	XCH A,R3
4C H	ORL A,R4	CC H	XCH A,R4
4D H	ORL A,R5	CD H	XCH A,R5
4E H	ORL A,R6	CE H	XCH A,R6
4F H	ORL A,R7	CF H	XCH A,R7
50 H	JNC rel	D0 H	POP direct
51 H	ACALL addr11	D1 H	ACALL addr11
52 H	ANL direct,A	D2 H	SETB bit
53 H	ANL direct,#data	D3 H	SETB C
54 H	ANL A,#data	D4 H	DA A
55 H	ANL A,direct	D5 H	DJNZ direct,rel
56 H	ANL A,@R0	D6 H	XCHD A,@R0
57 H	ANL A,@R1	D7 H	XCHD A,@R1
58 H	ANL A,R0	D8 H	DJNZ R0,rel
59 H	ANL A,R1	D9 H	DJNZ R1,rel
5A H	ANL A,R2	DA H	DJNZ R2,rel
5B H	ANL A,R3	DB H	DJNZ R3,rel
5C H	ANL A,R4	DC H	DJNZ R4,rel
5D H	ANL A,R5	DD H	DJNZ R5,rel
5E H	ANL A,R6	DE H	DJNZ R6,rel
5F H	ANL A,R7	DF H	DJNZ R7,rel
60 H	JZ rel	E0 H	MOVX A,@DPTR
61 H	AJMP addr11	E1 H	AJMP addr11
62 H	XRL direct,A	E2 H	MOVX A,@R0
63 H	XRL direct,#data	E3 H	MOVX A,@R1
64 H	XRL A,#data	E4 H	CLR A
65 H	XRL A,direct	E5 H	MOV A,direct
66 H	XRL A,@R0	E6 H	MOV A,@R0
67 H	XRL A,@R1	E7 H	MOV A,@R1
68 H	XRL A,R0	E8 H	MOV A,R0
69 H	XRL A,R1	E9 H	MOV A,R1
6A H	XRL A,R2	EA H	MOV A,R2
6B H	XRL A,R3	EB H	MOV A,R3
6C H	XRL A,R4	EC H	MOV A,R4
6D H	XRL A,R5	ED H	MOV A,R5
6E H	XRL A,R6	EE H	MOV A,R6
6F H	XRL A,R7	EF H	MOV A,R7
70 H	JNZ rel	F0 H	MOVX @DPTR,A
71 H	ACALL addr11	F1 H	ACALL addr11
72 H	ORL C,bit	F2 H	MOVX @R0,A

73 H	IMP @A+DPTR	F3 H	MOVX @R1,A
74 H	MOV A,#data	F4 H	CPL A
75 H	MOV direct,#data	F5 H	MOV direct,A
76 H	MOV @R0,#data	F6 H	MOV @R0,A
77 H	MOV @R1,#data	F7 H	MOV @R1,A
78 H	MOV R0.#data	F8 H	MOV R0,A
79 H	MOV R1.#data	F9 H	MOV R1,A
7A H	MOV R2.#data	FA H	MOV R2,A
7B H	MOV R3.#data	FB H	MOV R3,A
7C H	MOV R4.#data	FC H	MOV R4,A
7D H	MOV R5.#data	FD H	MOV R5,A
7E H	MOV R6.#data	FE H	MOV R6,A
7F H	MOV R7.#data	FF H	MOV R7,A

读-修改-写指令

从 SFR 或内部 RAM 中的读取字节、修改并重新回写指令称为“读-修改-写”指令 (RMW 指令)。当操作 I/O 端口 (P0-P4)，或一个端口位，这些指令读取的是输出锁存内容，而不是管脚。

RMW 指令

助记符 c	说明	操作码	字节数	周期数
ANL direct,A	直接寻址单元与累加器	0x52	2	3
ANL direct,#data	直接寻址单元与立即数	0x53	3	4
ORL direct,A	直接寻址单元或累加器	0x42	2	3
ORL direct,#data	直接寻址单元或立即数	0x43	3	4
XRL direct,A	直接寻址单元异或累加器	0x62	2	3
XRL direct,#data	直接寻址单元异或立即数	0x63	3	4
JBC bit, rel	位寻址位为 1 跳转并清	0x10	3	4
CPL bit	位寻址位取反	0xB2	2	3
INC direct	直接寻址单元加 1	0x05	2	3
INC @Ri	寄存器间接寻址单元加 1	0x06-0x07	1	3
DEC direct	直接寻址单元减 1	0x15	2	3
DEC @Ri	寄存器间接寻址单元减 1	0x16-0x17	1	3
DJNZ direct,rel	直接寻址单元减 1 为非零跳转	0xD5	3	4
MOV bit,C	进位送位寻址位	0x92	2	3
CLR bit	清位寻址位	0xC2	2	3
SETB bit	位寻址位置 1	0xD2	2	3

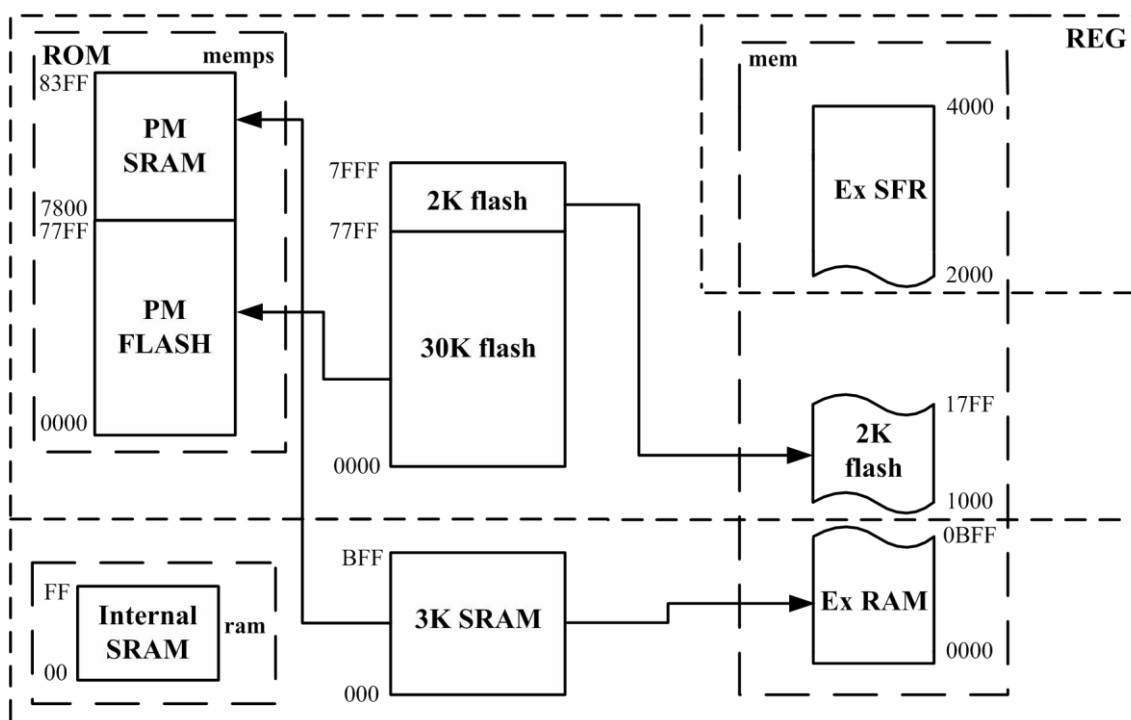
4. 存储器

存储器的组织结构与标准的 8051 内存组织类似，它分为两个独立的内存空间：程序内存和数据内存。程序和数据内存共享同一地址空间，但需经不同的指令类型存取。有 32k 字节可编程闪存，可用户程序编程。256 字节 IRAM，3072 字节 Extern RAM。

4.1. 程序存储器 (Program Memory)

程序存储器有 30k 字节可编程闪存空间，可存储非易失性程序或数据，可用户程序编程。地址 0000H~77FFH 存储空间为程序区；地址 7800H~83FFH 存储空间与地址 0000H~0BFFH XRAM 存储空间重合，MCON.3 为 1 时，XRAM 映射到程序区，可运行程序。

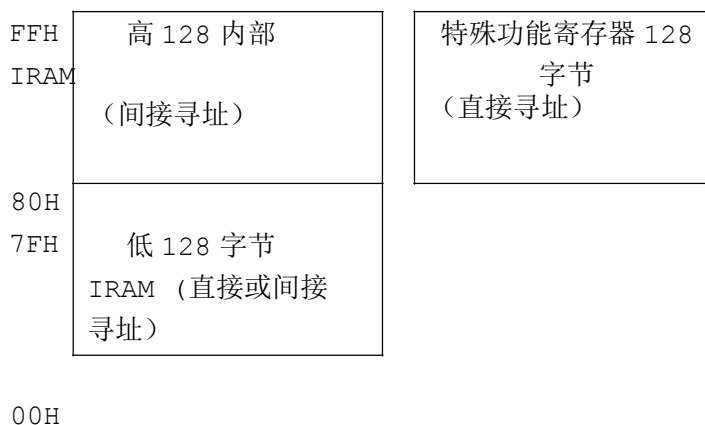
2k EEPROM 数据区映射到地址 1000H~17FFH XRAM 存储区。



4.2. 数据存储器 (Data Memory)

有 256 字节内部数据 RAM, 从 0 到 127 有 128 字节内部 RAM 被映射到数据内存空间用于通用寄存器及高速暂存器。不论直接或间接寻址均可用于存取。从 128 到 255 字节内部 RAM 可通过间接寻址存取。有四个通用寄存器组，每组包括 4 个 8 位寄存器，地址从 0 到 31。从 32 到 47 单元的 16 字节，可按字节寻址或以直接寻址模式寻址 128 位。高 128 字节区作为特殊功能寄存器 (SFR)，使用直接寻址的指令将存取 SFR 空间。

3072 字节 XRAM 可通过间接寻址读写。



4.3. EEPROM

EEPROM 擦写次数在 2 万次以上。可分为若干个扇区，每个扇区包含 128 字节，数据存储器的擦除操作是按扇区进行的。

EEPROM 可用于保存一些需要在应用过程中修改并且掉电不丢失的参数数据。在用户程序中，可以对 EEPROM 进行字节读 / 字节编程 / 扇区擦除操作。

4.4. 特殊功能寄存器分布

4.5. 特殊功能寄存器地址表

ADDR (HEX)	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	ADDR (HEX)
80	P0	SP	DPL	DPH	DPL1	DPH1	WDTREL	PCON	87
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	MCON	8F
90	P1	DACI	DPS	DPC	PAGESEL	D_PAGESEL	PWM_INT_FLG	DSAB	97
98	S0CON	S0BUF	IEN2	S1CON	S1BUF	S1RELL	PWM_INT_EN	PWM_CTR_L2	9F
A0	P2	LINRCBL	LINRCBH	LINCON	SBFTIG	T0CKP	T1CKP	PWM_CTR_L1	A7



A8	IEN0	IP0	SORELL	PROT_EN	PROT_CTR L1	PROT_CTR L2	PROT_CTR L3	TIME_BL ANK	AF
B0	P3	RST_CTRL	CLK_CTRL	POL_CTRL	LEV_BUF	OUTPUT_M	OUTPUT_U	TIME_DE	B7

						ODE	PCTRL	ADZONE	
B8	IP/IEN1	IP1	S0RELH	S1RELH	GPIOIFR0	GPIOIFR1	GPIOIFR2	IRCON2	BF
C0	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3	C7
C8	T2CON	AEN	CRCL	CRCH	TL2	TH2	GPIOIFLG0	GPIOIFLG1	CF
D0	PSW	IEN4	SLCT_H	SLCT_L	ADC_D3L	ADC_D3H	ADC_D2L	ADC_D2H	D7
D8	ADCON	ADC_MOD	ADC_CFG	ADC_TRIG_EN	ADC_D1L	ADC_D1H	ADC_D0L	ADC_D0H	DF
E0	ACC	SPSTA	SPCON	SPDAT	SPSNN	PWM0_INT_EN	PWM0_INT_FLG	CMP_CTRL	E7
E8	P4	MD0	MD1	MD2	MD3	MD4	MD5	ARCON	EF
F0	B	PWM_DC1_L	PWM_DC1_H	PWM_DC2_L	PWM_DC2_H	PWM_DC3_L	PWM_DC3_H	SRST	F7
F8	P5	PWM_CYC_L	PWM_CYC_H	PWM_CM1_L	PWM_CM1_H	PWM_CM2_L	PWM_CM2_H	PGA_GAIN	FF
2000	P0IEN	P1IEN	P2IEN	P3IEN	P4IEN	P5IEN			2007
2010	P0PUEN	P1PUEN	P2PUEN	P3PUEN	P4PUEN	P5PUEN			2017
2020	P0ODEN								2027
2030	P0PDEN								2030
2040	P0FUN0	P0FUN1	P1FUN	P2FUN	P3FUN0	P3FUN1	P4FUN	P5FUN	2047
2050	GPIO_DS	GPIO_SR							2057
2100	PWM0_CYC_H	PWM0_CYC_L	PWM0_DC_H	PWM0_DC_L	PWM0_CTRL_L1	PWM0_CTRL_L2	PWM0_INT_EN	PWM0_INT_FLG	2107
2200	PWM1_CYC	PWM1_DC	PWM1_CTRL_L1	PWM1_CTRL_L2	PWM1_INT_EN	PWM1_INT_FLG			2207
2300	PWM2_CYC	PWM2_DC	PWM2_CTRL_L1	PWM2_CTRL_L2	PWM2_INT_EN	PWM2_INT_FLG			2307
3000	RSTB_LVR_CTRL								3007
ADDR (HEX)	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	ADDR (HEX)

4.6. 编译器兼容的 SFR

4.6.1. 累加器 (Accumulator) - ACC

地址 - E0H 初始值 - 00H

该寄存器是算术操作累加器。

4.6.2. B 寄存器 - B

地址 - F0H

初始值 - 00H

该寄存器为某些算术操作的第二累加器。

4.6.3. 程序状态寄存器 - PSW

地址 - D0H

初始值 - 00H

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

7 CY 进位标志 当最后一次算术操作产生进位（加法）或借位（减法）时，该位被设置。其它算术操作将其清为逻辑 0。

6 AC 辅助进位标志 当最后一次算术操作向高半字节有进位（加法）或借位（减法）时，该位被设置。其它算术操作将其清为逻辑 0。

5 F0 用户标志 0
这是一个可位寻址、受软件控制的通用标志位。

4 RS1: RS0 寄存器区选择

3 这些位在寄存器访问时用于选择寄存器区。

		00: 寄存器区 0, 地址 0x00-0x07
		01: 寄存器区 1, 地址 0x08-0x0F
		10: 寄存器区 2, 地址 0x10-0x17
		11: 寄存器区 3, 地址 0x18-0x1F
-	2	OV
		溢出标志 在下列情况下, 该位设置为 1:
		一个 ADD、ADDC 或 SUBB 指令导致符号改变溢出。
		一个 MUL 指令导致溢出 (结果大于 255)。
		一个 DIV 指令导致出现除以 0 的情况。
		在所有其它情况下, OV 位被 ADD、ADDC、SUBB、MUL 和 DIV 指令清 0。
1	F1	
		用户标志 1
		这是一个可位寻址、受软件控制的通用标志位。
0	P	
		奇偶标志
		若累加器中八个位的和为奇数, 则该位设置为逻辑 1, 为偶数时则清零。

4.6.4. 堆栈指针寄存器 - SP

地址 - 81H

初始值 - 07H

堆栈指针保持栈顶位置。在每次执行 PUSH 操作前, 堆栈指针增加。SP 寄存器复位后的默认值为 0x07。

4.6.5. 数据指针 - DPH, DPL

DPL:

地址 - 82H

初始值 - 00H

DPH:

地址 - 83H

初始值 - 00H

4.6.6. 数据指针 1 – DPH1, DPL1

DPL1:

地址 – 84H

初始值 – 00H

DPH1:

地址 – 85H

初始值 – 00H

4.6.7. 数据指针选择寄存器 – DPS

DPS:

地址 – 92H

初始值 – 00H

R8051XC2 包含多达八个数据指针寄存器，取决于配置。这些寄存器可以作为 16 位间接寻址的地址。DPS 寄存器用于选择有效数据指针寄存器。

IP 核心配置中，如果只选一个数据指针寄存器 (DPTR)，然后 DPS 寄存器分配不在 SFR 的内存空间。然而，还有一种可能性，选择两个数据指针或八数据指针来分配。

7	6	5	4	3	2	1	0
-	-	-	-	BSE	DPSEL2	DPSEL1	DPSEL0
R	R	R	R	R/W	R/W	R/W	R/W

3 BSE 数据指针区开关使能

1 – 支持数据指针区，DPSEL2:DPSEL1 变成只读，由 PSW 寄存器中 RS1:RS0 取代。四组两字节数据指针建立于内部数据存储器 (IRAM) R0~R7 寄存器区。DPSEL0 位维持读写，并且由软件或 DPTR 自动开关正常控制。

2:0 数据指针寄存器选择，指定活动数据指针寄存器的数量。

000 只有一个 DPTR; DPSEL 设置，
xx1 两个到八个 DPTRs 选位)

4.6.8. 数据指针控制寄存器 – DPC

DPS:

地址 – 93H

初始值 – 00H、08H、10H、18H、20H、28H、30H、38H

R8051XC2 包含一个可选的数据指针相关运算单元。它提供了自增(1)或递减(2)，并且在有效的数据寄存器之间自动开关。DPC 寄存器控制这个功能。每个数据指针寄存器都有对应的 DPC 寄存器提供高度灵活的数据传输操作。DPC 内容指向 DPS 选择的数据指针寄存器。

当 IP 核配置为两个数据指针寄存器时，DPC 中位 5 核位 4 是不需要关心的。当只有一个数据指针寄存器，DPC 中位 5、位 4 和位 3 是不需要关心的。

7	6	5	4	3	2	1	0
-	-	NDPS2	NDPS1	NDPS0	AMS	AMD	AME
R	R	R/W	R/W	R/W	R/W	R/W	R/W

5:3 NDPSn 下一个数据指针选择

每个 MOVX @ DPTR 指令后，此字段的内容加载到 DPS 寄存器。请注意，此功能总是启用。因此每个 DPS 寄存器此字段包含一个不同的值指向其自身以便自动切换时不会出现被复位初始值代替。

2 AMS 自动增减大小，DPS.0 = 1;
0 - 每个 MOVX @DPTR 指令执行后，当前的 DPTR 自动递增或递减 1。
1 - 每个 MOVX @DPTR 指令执行后，当前的 DPTR 自动递增或递减 2。

1 AMD 自动增或减，DPS.0 = 1;
0 - 每个 MOVX @DPTR 指令执行后，当前的 DPTR 自动递增。
1 - 每个 MOVX @DPTR 指令执行后，当前的 DPTR 自动递减。

0 AME 自动增减使能
0 - 每个 MOVX @DPTR 指令执行后，当前的 DPTR 自动增减禁止。
1 - 每个 MOVX @DPTR 指令执行后，当前的 DPTR 自动增减使能。

4.6.9. 程序存储器页选寄存器 - PAGESEL

DPS:

地址 - 94H

初始值 - 01H

此寄存器为程序存储器扩展提供了一个额外的地址空间。请注意，复位后不写此寄存器，此寄存器初始值是 1，提供生成正常的地址（最大逻辑地址的 8000H 的物理空间地址）。对此寄存器写 0，会引发逻辑地址 8000H~FFFFH 之间的程序存储区覆盖逻辑地址 0000H~7FFFH 的公共程序存储区。

此寄存器用于程序存储器地址总线大于 16 位时，扩展程序存储器。

4.6.10. 数据存储器页选寄存器 - D_PAGESEL

DPS:

地址 - 95H

初始值 - 01H

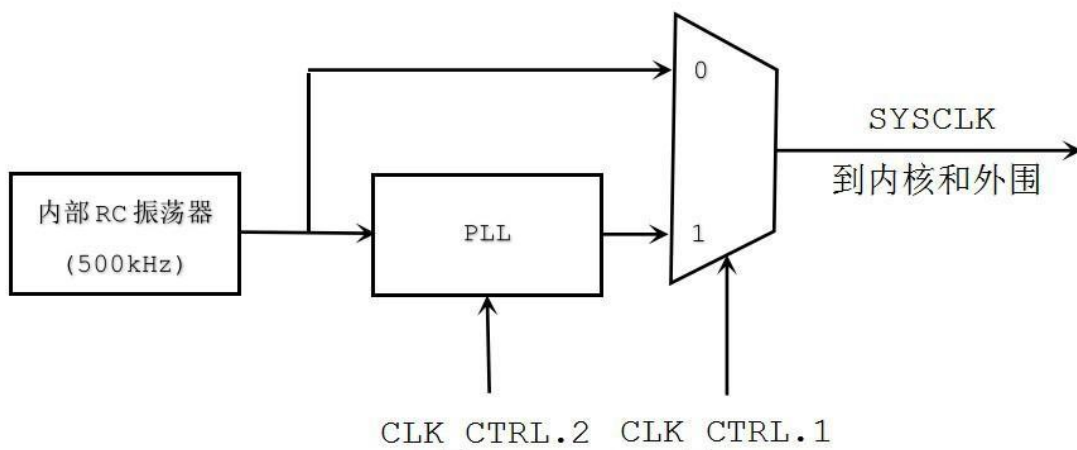
此寄存器为数据存储器扩展提供了一个额外的地址空间。请注意,复位后不写此寄存器,此寄存器初始值是 1,提供生成正常的地址(最大逻辑地址的 8000H 的物理空间地址)。对此寄存器写 0,会引发逻辑地址 8000H~ FFFFH 之间的数据存储区覆盖逻辑地址 0000H~7FFFH 的公共数据存储区。

此寄存器用于数据存储器地址总线大于 16 位时,扩展数据存储器。

5. 系统时钟

5.1. 特性

1. 内部 RC 振荡或外部晶体振荡器作为精确可调频率锁相环 (PLL) 时钟源，产生高频时钟提供 CPU 和外围设备。最高 60MHz。
2. 内部 RC 振荡或外部晶体振荡器作为时钟源供 CPU 和外围设备。
3. RESET 后 MCU 使用内部 RC 振荡器作为时钟源



5.2. 系统时钟寄存器

CKCON 存储器读写时钟控制寄存器: **8EH** 初始值: **71H**

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

6:4 程序存储器写等待状态控制数。

2:0 外部数据存储器延展周期数

**CKCON 使用需要小心，无特别需求使用默认值。CKCON 设置为 0，CPU 以最快速度 运行。

CLK_CTRL 系统时钟控制寄存器: **B2H** 初始值 - **FCH**

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

7:6	P41 时钟输出配置。 00 - RC 时钟输出。 01 - 输出 0。 10 - PLL 时钟 4 分频输出。 11 - 输出 0 (P41)。
-----	---

5	ADC 时钟使能。 0 - 禁止。 1 - 使能。
---	---------------------------------

4	GPIO 时钟使能。 0 - 禁止。 1 - 使能。
---	----------------------------------

3	PWM 时钟使能。 0 - 禁止。 1 - 使能。
---	---------------------------------

2	锁相环 (PLL) 使能。 0 - 禁止。 1 - 使能。
---	-------------------------------------

1	芯片主时钟选择。 0 - 内部 RC 低频时钟或外部低频晶振时钟。 1 - 内部 RC 或外部晶振经 PLL 产生高频时钟。 设置后读出是 1，确认高频时钟启动。设置前先使能 PLL。
---	--

6. 中断

6.1. 中断向量表

中断源	中断向量	优先级	请求标志	位寻址	硬件清	使能标志
RESET	0x0000	最高	无			
INT0	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IEN0.0)
TIMER0	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IEN0.1)
INT1	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IEN0.2)
TIMER1	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IEN0.3)
UART0	0x0023	4	RI0 (S0CON.0) TI0 (S0CON.1)	Y	N	ES0 (IEN0.4)
TIMER2	0x002B	5	EXF2 (IRCON.7) TF2 (IRCON.6)	Y	Y	ET2 (IEN0.5) EXEN2 (IEN1.7)
ADC	0x0043	8	IADC (ADC_CFG.0)	N	N	IEN1.0
SPI	0x004B	9	ISPI (IRCON.1)	Y	Y	IEN1.1
COM0	0x0053	10	ICC0 (IRCON.2)	Y	Y	IEN1.2
COM1	0x005B	11	ICC1 (IRCON.3)	Y	Y	IEN1.3
COM2	0x0063	12	ICC2 (IRCON.4)	Y	Y	IEN1.4
COM3	0x006B	13	ICC3 (IRCON.5)	Y	Y	IEN1.5
UART1	0x0083	16	TI1 (S1CON.1) RI1 (S1CON.0)	N	N	ES1 (IEN2.0)
MPWM	0x008B	17	CM2_INT_FLG (PWM_INT_FLG.6) CM1_INT_FLG (PWM_INT_FLG.5) DC3_INT_FLG (PWM_INT_FLG.4) DC2_INT_FLG (PWM_INT_FLG.3) DC1_INT_FLG (PWM_INT_FLG.2) TOP_INT_FLG (PWM_INT_FLG.1) BOT_INT_FLG (PWM_INT_FLG.0)	N	N	IEN2.1 CM2_INT_EN (PWM_INT_EN.6) CM1_INT_EN (PWM_INT_EN.5) DC3_INT_EN (PWM_INT_EN.4) DC2_INT_EN (PWM_INT_EN.3) DC1_INT_EN (PWM_INT_EN.2) TOP_INT_EN (PWM_INT_EN.1) BOT_INT_EN (PWM_INT_EN.0)
MPWM_HIG HZ	0x0093	18	HIGHZ_INT_FLG (PROT_CTRL2.4)	N	N	IEN2.2 HIGHZ_INT_EN (P

						ROT_CTRL1.7)
PWM0	0x009B	19	DC_INT_FLG(PWM0_FLG.2) TOP_INT_FLG(PWM0_FLG.1) BOT_INT_FLG(PWM0_FLG.0)	N	N	IEN2.3 DC_INT_EN(PWM0_INT.2) TOP_INT_EN(PWM0_INT.1) BOT_INT_EN(PWM0_INT.0)
GPIO	0x00A3	20	IGPIO(IRCON2.4)	Y	Y	IEN2.4
CMP	0x00AB	21	IRCON2.4	Y	Y	IEN2.5

6.2. 中断控制寄存器

IEN0 中断使能寄存器: **A8H** 初始值: **00H**

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

7 EA 全局中断使能
0 - 禁止
1 - 使能

6 WDT 看门狗定时器刷新标志
启动一个看门狗定时器刷新。在 SWDT (IEN1.6) 设置前, 为防止意外的看门狗定时器刷新, 必须先设置此位。在设置此位的下一条指令后, WDT 位由硬件清零。依次设置 WDT 和 SWDT, 看门狗被刷新。

5 ET2 定时器 2 中断使能
0 - 禁止
1 - 使能

4 ES0 UART0 中断使能
0 - 禁止
1 - 使能

3 ET1 定时器 1 中断使能
0 - 禁止
1 - 使能

2 EX1 外部中断 1 中断使能

0 - 禁止
1 - 使能

1 ET0 定时器 0 中断使能
0 - 禁止
1 - 使能

0 EX0 外部中断 0 中断使能
0 - 禁止
1 - 使能

IEEN1 中断使能寄存器 1: B8H 初始值: 00H

7	6	5	4	3	2	1	0
EXEN2	SWDT	CC3_IN T	CC2_IN T	CC1_IN T	CC0_INT	SPI_IN T	ADC_IN T
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 EXEN2 定时器 2 外部重载中断使能
0 - 禁止
1 - 使能

6 SWDT 看门狗启动/刷新标志
当设置 WDT (IE.6) 后, 立即设置此位, 刷新看门狗定时器。此位写 1 后硬件自动清 0。

5 CC3_INT 0 - 禁止
1 - 使能
CCU3 比较中断可以设定为上下沿触发 (具体取决于 T2CON 中的 I3FR 位)。上下电平变化的脉冲宽度必须大于 1 个时钟周期。
与此中断对应的标志位 (IRCON.5)。当 CRC 寄存器设置为比较模式时, 定时器 2 中的值与 CRC 的值相等产生中断。
CC3_INT 标志在被调用中断服务程序时, 由硬件自动清除。

4 CC2_INT 0 - 禁止
1 - 使能
CCU2 比较中断可以设定为上下沿触发 (具体取决于 T2CON 中的 I3FR 位)。上下电平变化的脉冲宽度必须大于 1 个时钟周期。
与此中断对应的标志位 (IRCON.4)。当 CRC 寄存器设置为比较模式时, 定时器 2 中的值与 CRC 的值相等产生中断。
CC2_INT 标志在被调用中断服务程序时, 由硬件自动清除。

3 CC1_INT 0 - 禁止
1 - 使能

CCU1 比较中断可以设定为上下沿触发 (具体取决于 T2CON 中的 I3FR 位)。上下电平变化的脉冲宽度必须大于 1 个时钟周期。

与此中断对应的标志位 (IRCON.3)。当 CRC 寄存器设置为比较模式时, 定时器 2 中的值与 CRC 的值相等产生中断。

CC1_INT 标志在被调用中断服务程序时, 由硬件自动清除。

-
- 2 CC0_INT 0 - 禁止
 1 - 使能

CCU0 比较中断可以设定为上下沿触发 (具体取决于 T2CON 中的 I3FR 位)。上下电平变化的脉冲宽度必须大于 1 个时钟周期。

与此中断对应的标志位 (IRCON.2)。当 CRC 寄存器设置为比较模式时, 定时器 2 中的值与 CRC 的值相等产生中断。

CC0_INT 标志在被调用中断服务程序时, 由硬件自动清除。

-
- 1 SPI_INT SPI 中断使能
 0 - 禁止
 1 - 使能

-
- 0 ADC_INT ADC 中断使能
 0 - 禁止
 1 - 使能
-

IEN2 中断使能寄存器 2: 9AH

初始值: 00H

7	6	5	4	3	2	1	0
—	—	CMP_IN T	GPIO_I NT	PWM0_I NT	MPWM_HIG HZ_INT	MPWM_I NT	ES1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

-
- 5 CMP_INT 比较器中断使能, 上升沿产生中断。
 0 - 禁止
 1 - 使能

-
- 4 GPIO_INT 输入端口中断使能
 0 - 禁止
 1 - 使能

-
- 3 PWM0_INT PWM0 中断使能
 0 - 禁止
 1 - 使能

-
- 2 MPWM_HIGHZ_INT MPWM 保护中断使能
 0 - 禁止
 1 - 使能

1 MPWM_INT MPWM 中断使能
0 - 禁止
1 - 使能

0 ES1 UART1 中断使能
0 - 禁止
1 - 使能

GPIOIFR0 GPIO 中断使能寄存器 0: BCH 初始值: 00H

7	6	5	4	3	2	1	0
P40IFR		P17IFR		P16IFR		P02IFR	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:6 P40IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

5:4 P17IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

3:2 P16IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

1:0 P02IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

GPIOIFR1 GPIO 中断使能寄存器 1: BDH 初始值: 00H

7	6	5	4	3	2	1	0
P44IFR		P43IFR		P42IFR		P41IFR	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:6 P44IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断

11 - 上升和下降沿中断

5:4 P43IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

3:2 P42IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

1:0 P41IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

GPIOIFR2 GPIO 中断使能寄存器 2: BEH 初始值: 00H

7	6	5	4	3	2	1	0
P57IFR		P56IFR		P55IFR		P54IFR	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:6 P57IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

5:4 P56IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

3:2 P55IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

1:0 P54IFR 00 - 禁止
01 - 上升沿中断
10 - 下降沿中断
11 - 上升和下降沿中断

GPIOIFLG0 GPIO 中断请求控制寄存器 0: CEH 初始值: 00H

7	6	5	4	3	2	1	0
P44IFLG	P43IFLG	P42IFLG	P41IFLG	P40IFLG	P17IFLG	P16IFLG	P02IFLG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:0 1 - 有中断请求, 软件清零

GPIOIFLG1 GPIO 中断请求控制寄存器 1: CFH 初始值: 00H

7	6	5	4	3	2	1	0
-	-	-	-	P57IFLG	P56IFLG	P55IFLG	P54IFLG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:0 1 - 有中断请求, 软件清零

IP 中断优先级寄存器: B8H 初始值: 00H

7	6	5	4	3	2	1	0
EXEN2	SWDT	PT2	PS	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 EXEN2 定时器 2 外部输入中断、重载使能。
EXEN2=0, T2EX (定时器 2 外部输入) 中断 2 禁止, T2 重载由 T2 溢出设定。
EXEN2=1 并且 EAL=1, T2EX 中断使能, T2 重载由 T2EX 设定。

6 SWDT Watchdog 定时器“启动/刷新”标志。看门狗定时器“激活/刷新”标志。
WDT (IEN0.6) 写 1, Watchdog 定时器 清零, 同时 SWDT (IP.6) 由硬件立即清零。

5 PT2 定时器 2 中断优先级。
1: 定时器 2 中断优先级高。
0: 定时器 2 中断优先级低。

4 PS UART0 中断优先级。
1: UART0 中断优先级高。
0: UART0 中断优先级低。

3 PT1 定时器 1 溢出中断优先级。
1: 定时器 1 溢出中断优先级高。
0: 定时器 1 溢出中断优先级低。

2 PX1 外部中断 1 优先级。

1: 外部中断 1 优先级高。

0: 外部中断 1 优先级低。

1 PT0 定时器 0 溢出中断优先级。
1: 定时器 0 溢出中断优先级高。
0: 定时器 0 溢出中断优先级低。

0 PX0 外部中断 0 优先级。
1: 外部中断 0 优先级高。
0: 外部中断 0 优先级低。

IP0 中断优先级寄存器 0: A9H 初始值: 00H

7	6	5	4	3	2	1	0
—	WDTS	每一位与 IP1 相应的位一起指定对应中断优先组的优先级					
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

6 WDTS 看门狗定时器中断请求标志 当看门
狗定时器溢出时由硬件置位。

IP1 中断优先级寄存器 0: B9H 初始值: 00H

7	6	5	4	3	2	1	0
—	—	每一位与 IP0 相应的位一起指定对应中断优先组的优先级					
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

中断优先组

组	中断位	中断
0	IP1.0:IP0.0	INT0, UART1
1	IP1.1:IP0.1	TIMER0, INT_MPWM
2	IP1.2:IP0.2	INT1, INT_MPWM_HIGHZ
3	IP1.3:IP0.3	TIMER1, INT_PWM0
4	IP1.4:IP0.4	UART0, INT_GPIO
5	IP1.5:IP0.5	TIMER2, INT_CMP

优先级表

IP1.x	IP0.x	优先级
0	0	0 级 (最低)
0	1	1 级
1	0	2 级
1	1	3 级 (最高)

**IRCON 中断请求控制寄存器：C0H**

初始值：00H

7	6	5	4	3	2	1	0
EXF2	TF2	ICC3	ICC2	ICC1	ICC0	ISPI	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 EXF2 定时器 2 外部重载标志

6 TF2 定时器 2 溢出标志

5 ICC3 CCU3 中断标志 此标志在被调用中断服务程序时，由
硬件自动清除。

4 ICC2 CCU2 中断标志 此标志在被调用中断服务程序时，由
硬件自动清除。

3 ICC1 CCU1 中断标志 此标志在被调用中断服务程序时，由
硬件自动清除。

2 ICC0 CCU0 中断标志 此标志在被调用中断服务程序时，由
硬件自动清除。

1 ISPI SPI 中断标志 此标志在被调用中断服务程序时，由
硬件自动清除。

IRCON2 中断请求控制寄存器：BFH

初始值：00H

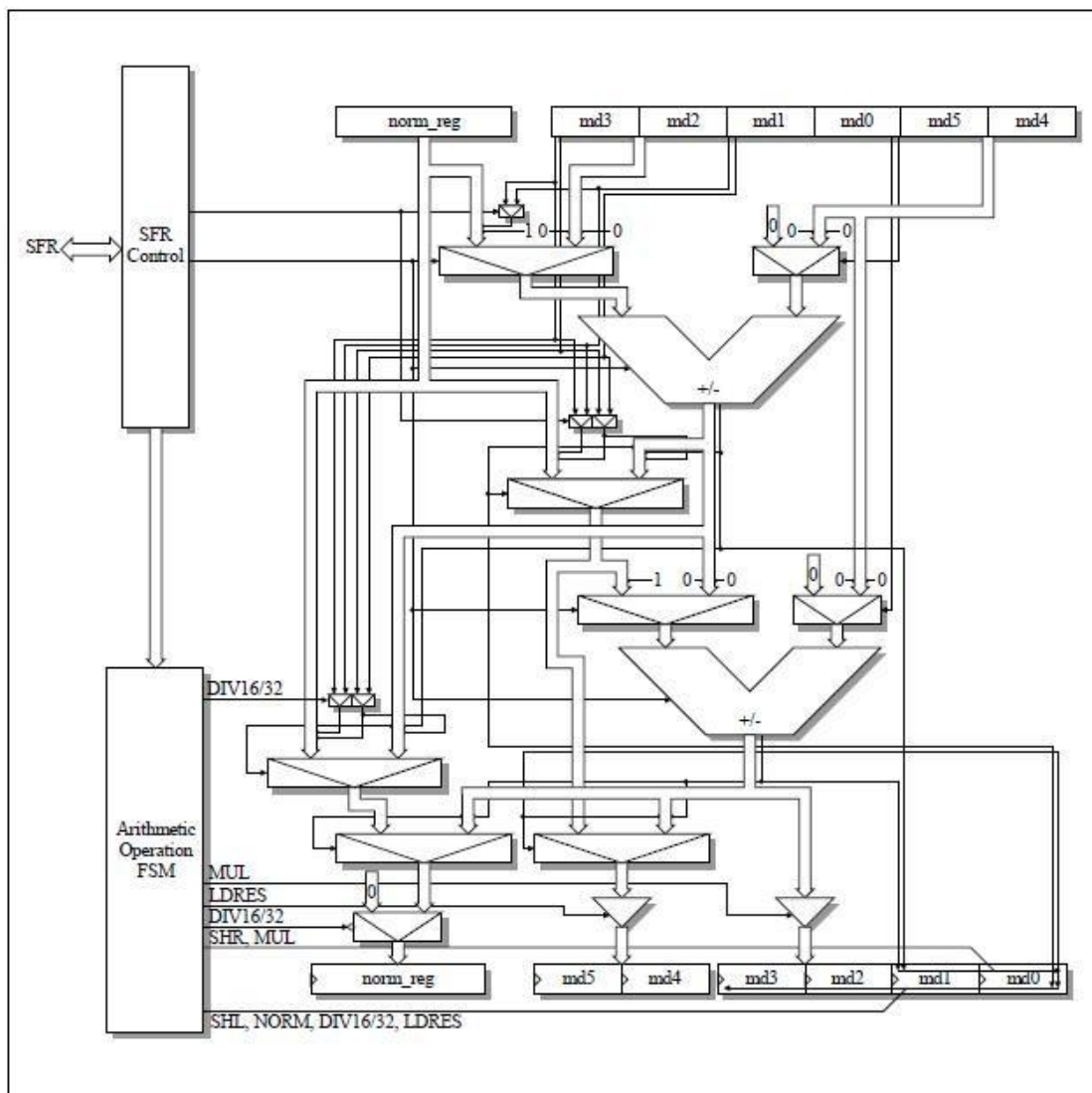
7	6	5	4	3	2	1	0
—	—	—	CMPFLG	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

4 CMPFLG 比较器中断标志

7. MDU

MDU 乘法除法器，是一个运算协处理器，支持 R8051XC2 提供 32 位、16 位乘法，除法，移位和规格化等无符号整数运算。MDU 由 7 个特殊功能寄存器实现独立于 CPU 的运算操作。操作数和结果存储在 MD0~MD5 寄存器中，该模块由 ARCON 特殊寄存器控制。任何 MDU 的计算重写它的操作数。

7.1. MDU 结构图



7.2. MDU 操作

MDU 操作分为三个阶段：

a) 加载 MDX 寄存器

MDU 计算的类型，由写入“MDx”寄存器的顺序决定。

MDU 寄存器写入顺序

操作	32 bit/16 bit	符号 16 bit×16 bit	无符号 16 bit × 16 bit	shifting normalizing
先写	md0 被除数[7:0]	md0 被乘数[7:0]	md0 被乘数[7:0]	md0 [7:0]
	md1 被除数[15:8]	md1 被乘数[15:8]	md4 乘数[7:0]	md1 [15:8]
	md2 被除数[23:16]		md1 被乘数[15:8]	md2 [23:16]
	md3 被除数[31:24]			md3 [31:24]
	md4 除数[7:0]	md4 乘数[7:0]		
最后写	md5 除数[15:8]	md5 乘数[15:8]	md5 乘数[15:8]	arcon

最先写入“md0”启动各类运算的数据传送动作，接着按上表顺序写入确定 MDU 运算，最后写入启动运算。SFR 控制检测上述某些序列，并将控制传递给算术运算 FSM。

b) 执行计算， 在执行运算时，MDU 与 CPU 并行工作。

运算	时钟周期数
除法 32bit/16bit	8 clock cycles
符号 16bit×16bit	1 clock cycles
无符号 16bit × 16bit	1 clock cycles
移位	1 clock cycles
规格化	1 clock cycles

MDU 运算时间

c) 从 MDX 寄存器读运算结果

MDU 寄存器读顺序

操作	32 bit/16 bit	符号 16 bit×16 bit	16 bit x 16 bit	shifting normalizing
先读	md0 商[7:0]	md0 乘积[7:0]	md0 乘积[7:0]	md0 [7:0]
	md1 商[15:8]	md1 乘积[15:8]	md1 乘积[15:8]	md1 [15:8]
	md2 商[23:16]	md2 乘积[23:16]	md2 乘积[23:16]	md2 [23:16]
	md3 商[31:24]			
	md4 余数[7:0]			
最后读	md5 余数[15:8]	md3 乘积[31:24]	md3 乘积[31:24]	md3 [31:24]

d) 规格化 (Normalizing)

存储于 MD0...MD3 寄存器的 32 位整数左移操作删除前导零。整个操作完成后,MD3 寄存器最高有效位为“1”,ARCON 寄存器后 5 位 (ARCON.4:ARCON.0) 包含左移操作次数。

e) 移位 (Shift)

在移位操作中,存储在 MD0...MD3 寄存器的 32 位整数被左或右移动指定的位数。ARCON 寄存器中 SLR(ARCON.5) 位) 定义移动方向。ARCON 寄存器后 5 位 (ARCON.4:ARCON.0) 包含移动次数。在移动操作中,零被填入移动的最后一位。

f) MDEF (ARCON.7) 标志

MDEF (ARCON.7) 标志指出不正确的操作 (当一个算术运算已被重新启动或被新的操作中断)。错误检测从第一次写入 MD0 寄存器自动启动到最终读 MD3 寄存器 (乘法或移动/规格化) 或 MD5 (除法) 停止。

错误标志被设置时:

- 在 MDU 操作重新启动或计算中断期间,写访问 MDx 寄存器 (任何 MD0...MD5) 和 ARCON 寄存器
- 在 MDU 操作重新启动或计算中断期间,错误标志机制启用时,读访问一个 MDx 寄存器。在这种情况下,错误标志设置但计算不中断。
读 ARCON 寄存器,错误标志被复位。

g) MDOV (ARCON.6) 标志

MDOV 溢出标志被设置时,出现下列情形:

- 除以零
- 乘法的结果大于 0000 FFFFH
- 如果 MD3 最高位是 1 (MD3.7=1) 时,启动规格化。

任何不符合上述条件的 MDU 操作,清除溢出标志。注意,溢出标志是由硬件专门控制,不能被写入。

7.3. MDU 寄存器

7.3.1. 乘除寄存器 – MD0, MD1, MD2, MD3, MD4, MD5

名称	地址	初始值
MD0	E9H	0
MD1	EAH	0
MD2	EBH	0
MD3	ECH	0
MD4	EDH	0
MD5	EEH	0

7.3.2. 运算控制寄存器 - ARCON

ARCON 寄存器:EEH

初始值: 00H

7	6	5	4	3	2	1	0
MDEF	MDOV	SLR	SC4	SC3	SC2	SC1	SC0
R	R	R/W	R/W	R/W	R/W	R/W	R/W

7 MDEF 指出不正确的操作（当一个算术运算已被重新启动或被新的操作中中断）。

6 MDOV MDU 运算出现溢出。

5 SLR 移位方向
0 : 左移
1 : 右移

4 SC4:SC0 移位次数
3 当这些位都为 0，执行规格化操作。规格化完成后，这些位储存移位的次
2 数。其他值执行移位操作，这些位指出移位次数，SC4 是最高位。
1
0

7.3.3. 例程

32 位数据除 16 位数据

```
MD0 = T0; // 被除数[7:0]
MD1 = T1; // 被除数[15:8]
MD2 = T2; // 被除数[23:16]
MD3 = T3; // 被除数[31:24]
```

```
MD4 = T4; // 除数[7:0]
MD5 = T5; // 除数[15:8]
```

```
_nop(); // 等待 8 个时钟周期
_nop();
_nop();
_nop();
```

```
_nop_();
_nop_();
_nop_();
_nop_();

if ((ARCON&0xC0)!=0)
{
    // 出错处理
}

T0 = MD0; // 商[7:0]
T1 = MD1; // 商[15:8]
T2 = MD2; // 商[23:16]
T3 = MD3; // 商[31:24]

T4 = MD4; // 余数[7:0]
T5 = MD5; // 余数[15:8]
```

有符号 16 位数据乘 16 位数据

```
MD0 = T0; // 被乘数[7:0]
MD1 = T1; // 被乘数[15:8]

MD4 = T4; // 乘数[7:0]
MD5 = T5; // 乘数[15:8]

_nop_(); // 等待 1 个时钟周期

if ((ARCON&0xC0)!=0)
{
    // 出错处理
}

T0 = MD0; // 积[7:0]
T1 = MD1; // 积[15:8]
T2 = MD2; // 积[23:16]
T3 = MD3; // 积[31:24]
```

无符号 16 位数据乘 16 位数据

```
MD0 = T0; // 被乘数[7:0]
MD4 = T4; // 乘数[15:8]
```

```
MD1 = T1; // 被乘数[7:0]
MD5 = T5; // 乘数[15:8]

_nop_(); // 等待 1 个时钟周期

if ((ARCON&0xC0)!=0)
{
    // 出错处理
}

T0 = MD0; // 积[7:0]
T1 = MD1; // 积[15:8]
T2 = MD2; // 积[23:16]
T3 = MD3; // 积[31:24]
```

移位操作(数据规范化)

```
MD0 = T0; // 被移位数[7:0]
MD1 = T1; // 被移位数[15:8]
MD2 = T2; // 被移位数[7:0]
MD3 = T3; // 被移位数[15:8]
ARCON = 0; // 执行规范化操作

_nop_(); // 等待 1 个时钟周期

if ((ARCON&0xC0)!=0)
{
    // 出错处理
}

T0 = MD0; // 积[7:0]
T1 = MD1; // 积[15:8]
T2 = MD2; // 积[23:16]
T3 = MD3; // 积[31:24]
```

移位操作

```
MD0 = T0; // 被移位数[7:0]
MD1 = T1; // 被移位数[15:8]
MD2 = T2; // 被移位数[7:0]
MD3 = T3; // 被移位数[15:8]
ARCON = 1; // 执行左移一位操作
```

```
_nop();    // 等待 1 个时钟周期

if ((ARCON&0xC0)!=0)
{
    // 出错处理
}

T0 = MD0;  // 积[7:0]
T1 = MD1;  // 积[15:8]
T2 = MD2;  // 积[23:16]
T3 = MD3;  // 积[31:24]
```

8. 输入输出端口

P0 输入输出数据寄存器：80H

7	6	5	4	3	2	1	0
P07	P06	P05	P04	P03	P02	P01	P00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1 输入输出数据寄存器：90H

7	6	5	4	3	2	1	0
P17	P16	P15	P14	P13	P12	P11	P10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2 输入输出数据寄存器：A0H

7	6	5	4	3	2	1	0
P27	P26	P25	P24	P23	P22	P21	P20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3 输入输出数据寄存器：B0H

7	6	5	4	3	2	1	0
—	P36	—	P34	P33	P32	P31	P30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P4 输入输出数据寄存器：E8H

7	6	5	4	3	2	1	0
—	—	—	P44	P43	P42	P41	P40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P5 输入输出数据寄存器：F8H

7	6	5	4	3	2	1	0
P57	P56	P55	P54	P53	P52	P51	P50
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P0IEN P0 数据方向选择寄存器：2000H 初始值：FFH

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

7:0 数据方向选择
0 - 输出
1 - 输入

P1IEN P1 数据方向选择寄存器: 2001H 初始值: FFH

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

7:0 数据方向选择

0 - 输出

1 - 输入

P2IEN P2 数据方向选择寄存器: 2002H 初始值: FFH

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

7:0 数据方向选择

0 - 输出

1 - 输入

P3IEN P3 数据方向选择寄存器: 2003H 初始值: FFH

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

7:0 数据方向选择

0 - 输出

1 - 输入

P4IEN P4 数据方向选择寄存器: 2004H 初始值: FFH

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

4:0 数据方向选择

0 - 输出

1 - 输入

P5IEN P5 数据方向选择寄存器: 2005H 初始值: FFH

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

7:0 数据方向选择
0 - 输出
1 - 输入

P0ODEN P0 开漏使能寄存器: 2020H 初始值: 00H

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

7:0 输出开漏使能
0 - 禁止
1 - 使能

P0PUEN P0 上拉电阻使能寄存器: 2010H 初始值: 00H

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

7:0 输入上拉电阻使能
0 - 禁止
1 - 使能
P07 关闭上拉电阻时, 注意必须把 P0FUNC0.7 设置成 1 以避免引起 MCU 复位。

P1PUEN P1 上拉电阻使能寄存器: 2011H 初始值: 00H

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

7:0 输入上拉电阻使能
0 - 禁止
1 - 使能

P2PUEN P2 上拉电阻使能寄存器: 2012H 初始值: 00H

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:0 输入上拉电阻使能
0 - 禁止
1 - 使能

P3PUEN P3 上拉电阻使能寄存器: 2013H 初始值: 00H

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

7:0 输入上拉电阻使能
0 - 禁止
1 - 使能

P4PUEN P4 上拉电阻使能寄存器: 2014H 初始值: 00H

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

4:0 输入上拉电阻使能
0 - 禁止
1 - 使能

P5PUEN P5 上拉电阻使能寄存器: 2015H 初始值: 00H

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

3:0 输入上拉电阻使能
0 - 禁止
1 - 使能

P0PDEN P0 下拉电阻使能寄存器: 2030H 初始值: 00H

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

7:0 输入下拉电阻使能

0 - 禁止

1 - 使能

P0FUN0 P0 特殊功能选择寄存器 0: 2040H 初始值: **FFH**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P01FUN	P00FUN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

1 P01 功能选择

0 - CMP- (模拟)

1 - P01

0 P00 功能选择

0 - CMP+ (模拟)

1 - P00

P0FUN1 P0 特殊功能选择寄存器 1: 2041H 初始值: **FFH**

7	6	5	4	3	2	1	0
P07FUN	P06FUN	P05FUN	P04FUN	P03FUN		P02FUN	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 P07 功能选择 (只有在 LVR_CTRL[7] (P07_RSTB_EN) 配置为 0, 即 P07 脚不用做 RESE 时, 本寄存器才起作用。P07_RSTB_EN 的优先级高于本寄存器的配置)。此功能只在 PMODE 引脚悬空时有; PMODE 接地, 此引脚为复位功能。此脚不能接地。
0 - ADC_PIN_TRIG
1 - P07

6 P06 功能选择

0 - TXD0

1 - P06

5 P05 功能选择

0 - RXD0

1 - P05

4 P04 功能选择

0 - T0

1 - P04

3:2 P03 功能选择

00 - INT1
01 - GOFF1
1x - P03

1:0 P02 功能选择
00 - DAC 输出
01 - CMPO (比较器模拟输出)
1x - P02

P1FUN P1 特殊功能选择寄存器: 2042H

初始值: **FFH**

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

7 P17 功能选择
0 - AN4 (ADC CH4 输入)
1 - P17

6 P16 功能选择
0 - AN5 (ADC CH5 输入); 或 AEN.5 (VCMEN=1), 则输出模拟 VCM 电压。
1 - P16

5 P15 功能选择
0 - AN6 (ADC CH6 输入)
1 - P15

4 P14 功能选择
0 - AN7 (ADC CH7 输入)
1 - P14

3 P13 功能选择
0 - AN8 (ADC CH8 输入)
1 - P13

2 P12 功能选择
0 - AN9 (ADC CH9 输入)
1 - P12

1 P11 功能选择
0 - AN10 (ADC CH10 输入)
1 - P11

0 P10 功能选择

0 - AN11 (ADC CH11 输入)

1 - P10

P2FUN P2 特殊功能选择寄存器: 2043H

初始值: **FFH**

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

7 P27 功能选择

0 - GAT 输出

1 - P27

6 P26 功能选择

0 - GAB 输出

1 - P26

5 P25 功能选择

0 - GBT 输出

1 - P25

4 P24 功能选择

0 - GBB 输出

1 - P24

3 P23 功能选择

0 - GCT 输出

1 - P23

2 P22 功能选择

0 - GCB 输出

1 - P22

1 P21 功能选择

0 - PWM0 输出

1 - P21

0 P20 功能选择

0 - INT0

1 - P20

P3FUN0 P3 特殊功能选择寄存器 0: 2044H

初始值: **FFH**

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

7:6 P33 功能选择
0x - AN0 (ADC CH0 输入)
10 - SS0 (SPI)
11 - P33

5:4 P32 功能选择
0x - AN1 (ADC CH1 输入)
10 - SCK (SPI)
11 - P32

3:2 P31 功能选择
00 - PGA10
01 - AN2 (ADC CH2 输入)
10 - MOSI (SPI)
11 - P31MOSI (SPI)

1:0 P30 功能选择
00 - PGA00
01 - AN3 (ADC CH3 输入)
10 - MISO (SPI)
11 - P30

P3FUN1 P3 特殊功能选择寄存器 1: 2045H 初始值: FFH

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

6 P36 功能选择
0 - PGA0+ (模拟输入)
1 - P36

5:4 P34 功能选择
0x - PGA1+ (模拟输入)
10 - T2EX
11 - P34

P4FUN P4 特殊功能选择寄存器: 2046H 初始值: FFH

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

-	-	-					
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

4 P44 功能选择
0 - ADC_VREF
1 - P44

3 P43 功能选择
0 - T2
1 - P43

2 P42 功能选择
0 - T1
1 - P42

1 P41 功能选择
0 - GOFF0
1 - P41

0 P40 功能选择
0 - PWM2
1 - P40

P5FUN P5 特殊功能选择寄存器: 2047H 初始值: FFH

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

7 P57 功能选择
0 - CKOUT
1 - P57

6 P56 功能选择
0 - TXD1
1 - P56

5 P55 功能选择
0 - RXD1
1 - P55

4 P54 功能选择

0 - PWM1

1 - P54

3 P53 功能选择

0 - CCU3

1 - P53

2 P52 功能选择

0 - CCU2

1 - P52

1 P51 功能选择

0 - CCU1

1 - P51

0 P50 功能选择

0 - CCU0

1 - P50

GPIO_DS GPIO 驱动强弱选择寄存器: 205H 初始值: FFH

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

5 P50-P57 摆动速率选择

0 - 强

1 - 弱

4 P40-P44 摆动速率选择

0 - 强

1 - 弱

3 P30-P34、P36 摆动速率选择

0 - 强

1 - 弱

2 P20-P27 摆动速率选择

0 - 强

1 - 弱

1 P10-P17 摆动速率选择

0 - 强

1 - 弱

0 P00-P07 摆动速率选择
 0 - 强
 1 - 弱

GPIO_SR GPIO 摆动速率选择寄存器: 2051H 初始值: FFH

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

5 P50-P57 摆动速率选择
 0 - 快
 1 - 慢

4 P40-P44 摆动速率选择
 0 - 快
 1 - 慢

3 P30-P34、P36 摆动速率选择
 0 - 快
 1 - 慢

2 P20-P27 摆动速率选择
 0 - 快
 1 - 慢

1 P10-P17 摆动速率选择
 0 - 快
 1 - 慢

0 P00-P07 摆动速率选择
 0 - 快
 1 - 慢

9. 定时/计数器

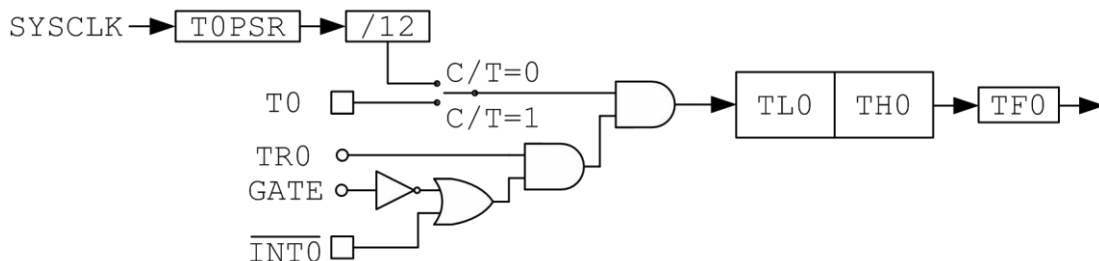
定时/计数器 0 和定时/计数器 1 是两个 16 位定时器，与 8051 的定时器完全兼容。两个 16 位定时器 T0 和 T1 都具有计数方式和定时方式两种工作方式。

定时/计数器 0, 有 4 种工作模式，定时/计数器 1 有三种工作模式。

定时器的计数频率=60M/T0PSR/12。

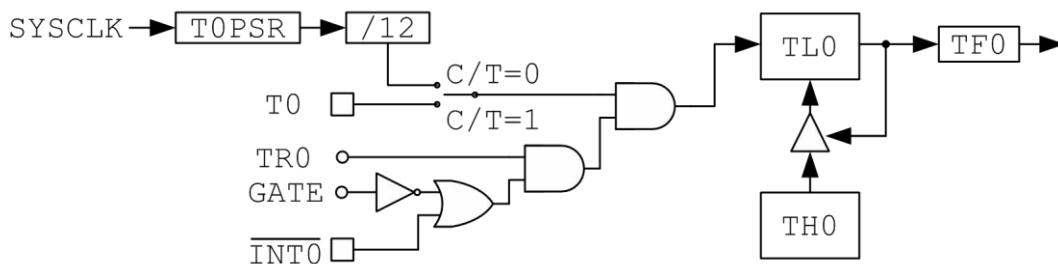
9.1. 定时/计数器 0

9.1.1. 模式 0 和模式 1



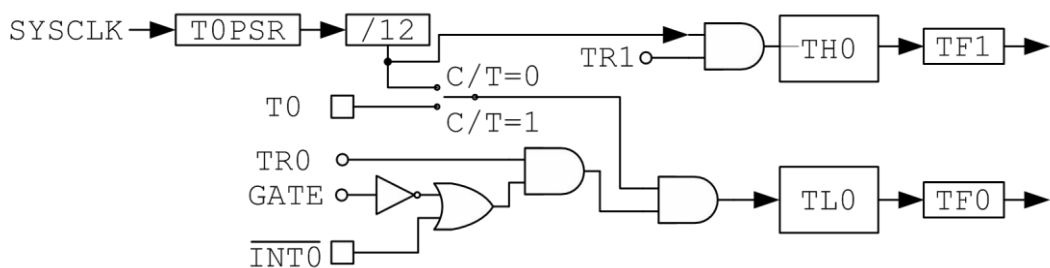
定时器 0 处于模式 0 和模式 1

9.1.2. 模式 2



定时器 0 处于模式 2

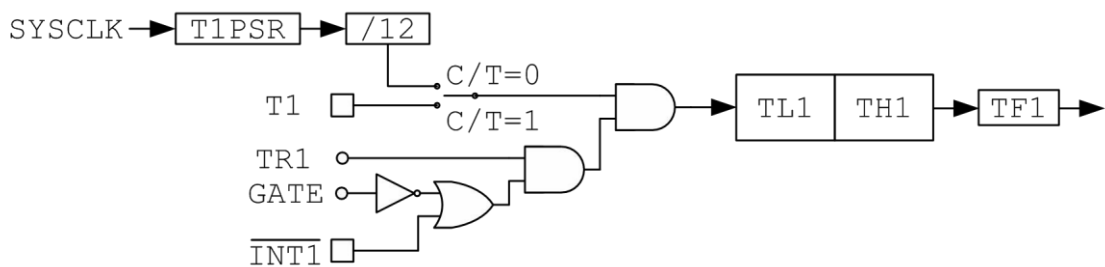
9.1.3. 模式 3



定时器 0 处于模式 3

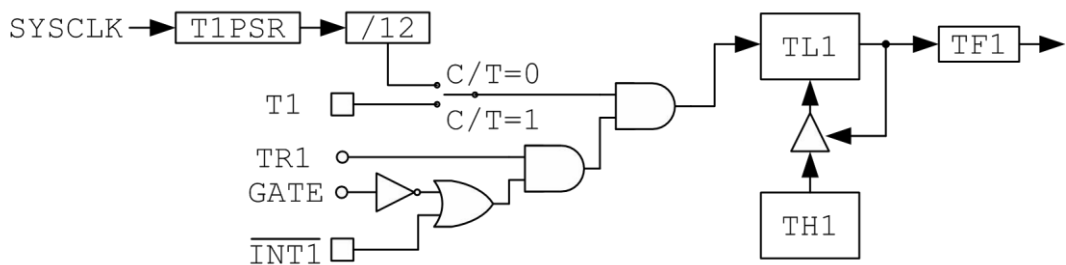
9.2. 定时/计数器 1

9.2.1. 模式 0 和模式 1



定时器 1 处于模式 0 和模式 1

9.2.2. 模式 2



定时器 1 处于模式 2

9.2.3. 模式 3

此模式定时器 1 停止工作。

9.2.4. 定时器 0 和定时器 1 寄存器

TCON 定时器控制寄存器：88H

初始值：00H

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

-
- 7 TF1 定时器 1 溢出标志。
当定时器 1 溢出时由硬件设置为 1。该标志可通过软件清除，当 CPU 转至定时器 1 中断服务程序时，该标志被自动清除。
-
- 6 TR1 定时器 1 运行控制。
通过将此位设置为 1 允许定时器 1 工作。
-
- 5 TF0 定时器 0 溢出标志。
当定时器 0 溢出时由硬件设置为 1。该标志可通过软件清除。当 CPU 转至定时器 0 中断服务程序时，该标志被自动清除。
-
- 4 TR0 定时器 0 运行控制。
通过将此位设置为 1 允许定时器 0 工作。
-
- 3 IE1 外部中断 1 请求标志。
当检测到一个由 IT1 (TCON.2) 定义的中断类型（边沿/电平）时，该标志由硬件设置。该标志可通过软件清除，但当在边缘触发模式下 CPU 转至外部中断 1 服务程序时，该标志被自动清除。
-
- 2 IT1 外部中断 1 触发方式。
此位选择配置 /INT1 中断是下降沿触发还是电平触发。0：/INT1 为电平触发。1：/INT1 为下降沿触发。
-
- 1 IE0 外部中断 0 请求标志。
当检测到一个由 IT0 (TCON.0) 定义的中断类型（边沿/电平）时，该标志由硬件设置。该标志可通过软件清除，但当在边缘触发模式下 CPU 转至外部中断 0 服务程序时，该标志被自动清除。
-
- 0 IT0 外部中断 0 触发方式。
此位选择配置 /INT0 中断是下降沿触发还是电平触发。0：/INT0 为电平触发。1：/INT0 为下降沿触发。

TMOD 定时器模式寄存器: 89H 初始值: 00H

7	6	5	4	3	2	1	0
GATE1	C/T1	M1	M0	GATE0	C/T0	M1	M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

-
- 7 GATE1 定时器 1 门控制。
0: 当 TR1 = 1 时定时器 1 允许, 与 INT1 的逻辑电平无关。
1: 仅当 TR1 = 1 并且 INT1 为高, 定时器 1 在 T1 输入脚信号的下降沿加 1。
-
- 6 C/T1 计数器/定时器 1 选择。
0: 定时器: 定时器 1 由 CKCON 寄存器中的 T1M 位定义的时钟加 1。
1: 计数器: 定时器 1 由外部引脚 (T1) 的负变跳加 1。
-
- 5 T1M[1:0] 定时器 1 模式选择。这些位选择定时器 1 的操作模式。
4 00: 模式 0, 13 位计数器/定时器
 01: 模式 1, 16 位计数器/定时器
 10: 模式 2, 8 位自动重载计数器/定时器
 11: 模式 3, 定时器 1 停止运行
-
- 3 GATE0 定时器 0 门控制。
0: 当 TR0 = 1 时定时器 0 被允许, 与 INT0 的逻辑电平无关。
1: 仅当 TR0 = 1 并且 INT0 为高, 定时器 0 在 T0 输入脚信号的下降沿加 1。
-
- 2 C/T0 计数器/定时器 0 选择。
0: 定时器: 定时器 0 由 CKCON 寄存器中的 T0M 位定义的时钟加 1。
1: 计数器: 定时器 0 由外部引脚 (T0) 的负变跳加 1。
-
- 1 T0M[1:0] 定时器 0 模式选择。这些位选择定时器 0 的操作模式。
0 00: 模式 0, 13 位计数器/定时器
 01: 模式 1, 16 位计数器/定时器
 10: 模式 2, 8 位自动重载计数器/定时器
 11: 模式 3, 两个 8 位计数器/定时器
-

TL0 定时器 0 低 8 位数据寄存器: 8AH 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH0 定时器 0 高 8 位数据寄存器: 8BH 初始值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TL1 定时器 1 低 8 位数据寄存器: 8CH 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH1 定时器 1 高 8 位数据寄存器: 8DH 初始值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T0PSR 定时器 0 输入时钟预分频寄存器: A5H 初始值: 00H

7	6	5	4	3	2	1	0
					T0CKP		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

2:0 T0CKP Timer0 输入时钟预分频

000: Fosc/1

001: Fosc/2

010: Fosc/4

011: Fosc/8

100: Fosc/16

101: Fosc/32

110: Fosc/64

111: Fosc/128

T1PSR 定时器 1 输入时钟预分频寄存器: A6H 初始值: 00H

7	6	5	4	3	2	1	0
					T1CKP		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

2:0 T1CKP Timer1 输入时钟预分频

000: Fosc/1

001: Fosc/2

010: Fosc/4

011: Fosc/8

100: Fosc/16

101: Fosc/32

110: Fosc/64

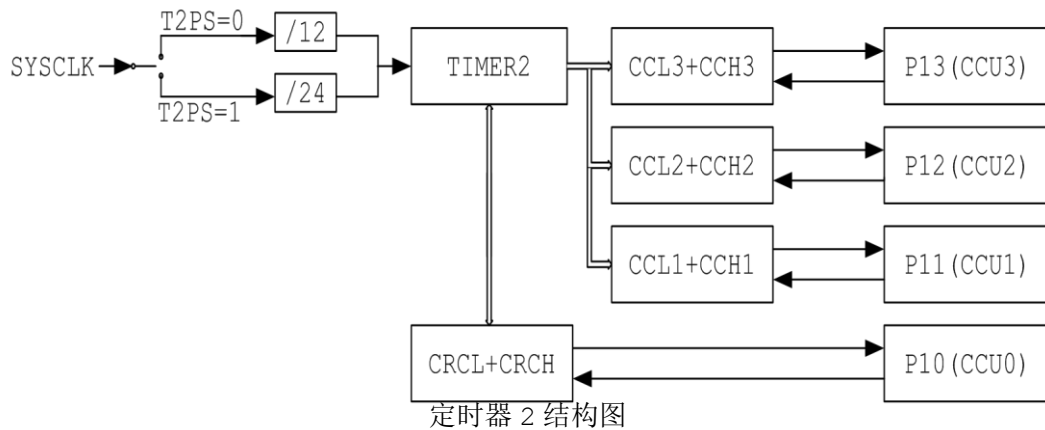
111: $F_{osc}/128$

IP 中断优先级寄存器: B8H

- [PT1 \(IP.3\) 定时器 1 溢出中断优先级。](#)
- [PX1 \(IP.2\) 外部中断 1 优先级。](#)
- [PT0 \(IP.1\) 定时器 0 溢出中断优先级。](#)
- [PX0 \(IP.0\) 外部中断 0 优先级。](#)

9.3. 定时/计数器 2

定时器 2 可以作为定时器、计数器或门控定时器。 定时器 2 包括比较/捕捉单元。



9.3.1. 定时模式

这种模式通过设置 T2CON 中标志位“T2I0=1”和“T2I1=0”建立。在此模式下，计数频率由 T2CON 中的 T2PS 位决定。

9.3.2. 事件计数模式

这种模式通过设置 T2CON 中标志位“T2I0=0”和“T2I1=1”建立。在此模式下，外部信号“T2”从 1 变到 0，定时器 2 加 1。“T2”输入在每个时钟的上升沿采样，定时器 2 是在检测到下一个周期信号翻转后递增。最大计数速率是时钟频率的一半。

9.3.3. 门控定时模式

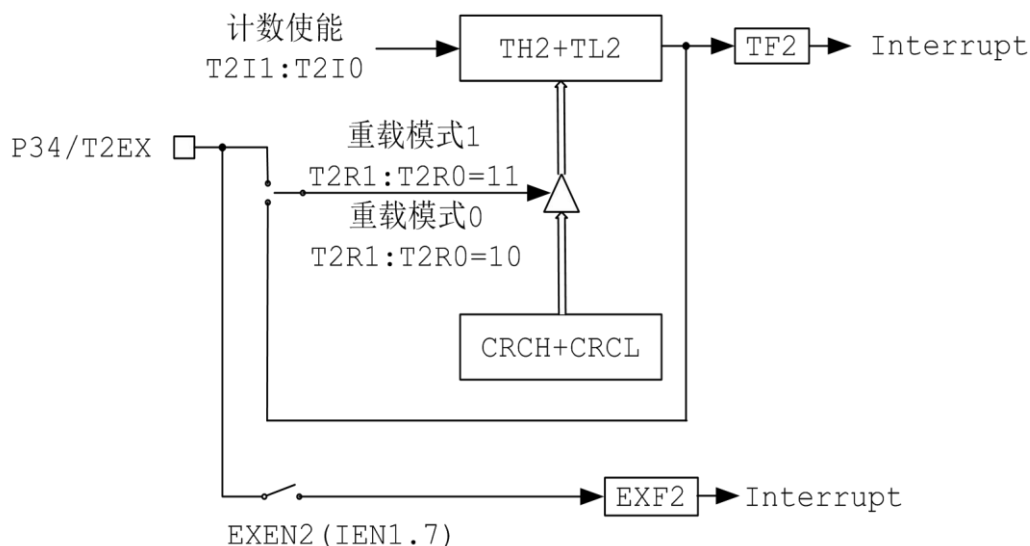
此模式通过设置 T2CON 中“T2I0=1”和“T2I1=1”建立。在这种模式下，定时器 2 每 12 或 24 个时钟（取决于“T2PS”标志）递增。另一个，它是由外部信号“T2”选通，当“T2=0”时，定时器 2 停止。

9.3.4. 定时器 2 重载

一个 16 位重载“CRC”寄存器（2.4.22）可以在两种模式下执行：

模式 0：重载信号是定时器 2 溢出（自动重装） 模式 1：重载信号

是相应的输入脚 T2EX “所产生的负跳变”。



定时器 2 重载模式

9.3.5. 比较功能

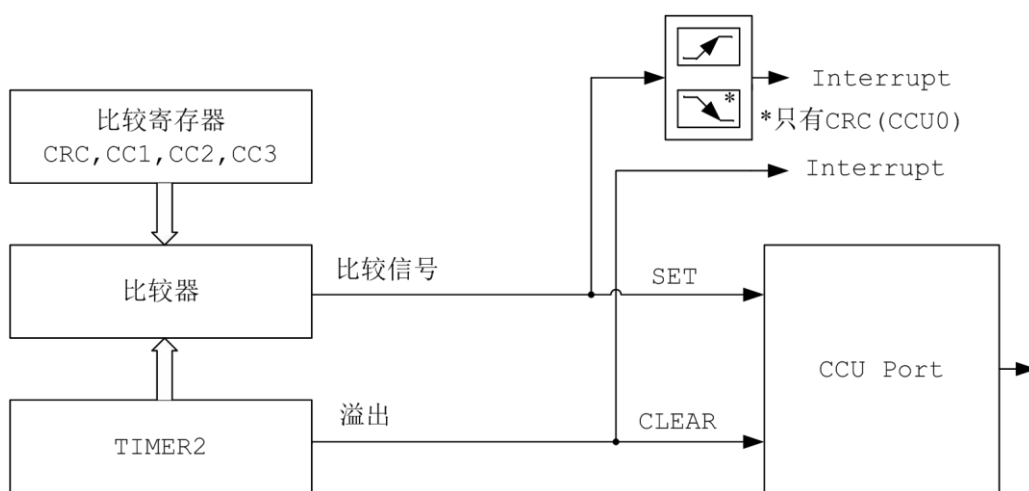
比较 / 捕捉单元由 CRC、CC1、CC2、CC3 四个寄存器组成。这些寄存器可以配置为比较工作模式。在这种模式下，寄存器的值与定时器 2 的值比较。比较器的输出驱动 P1 端口的低四位：

- P1.0 比较输出与寄存器 CRC 对应。
- P1.1 比较输出与寄存器 CC1 对应。
- P1.2 比较输出与寄存器 CC2 对应。
- P1.3 比较输出与寄存器 CC3 对应。

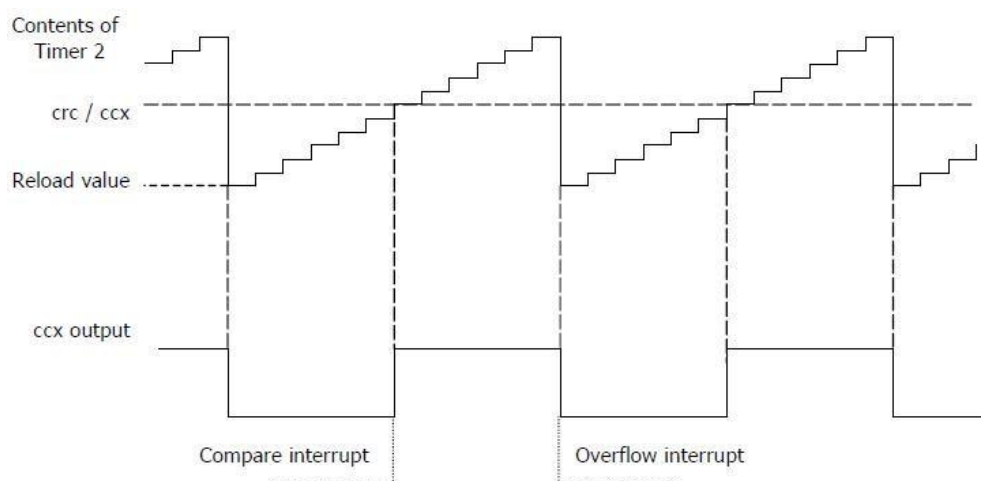
T2CON 寄存器中的 T2CM 选择两种比较模式。

9.3.5.1. 比较模式 0

“T2CM=0”设定比较模式 0，在模式 0 中，当定时器 2 的值等于的比较寄存器的值，比较器的输出电平从低变化到高；定时器 2 溢出恢复输出电平到低。在这种模式下，写对应的端口 (P1.0、P1.1、P1.2、P1.3) 不影响端口的状态，因为从内部总线与写入寄存器输入线断开。此模式可用于产生 PWM 输出。下图说明了功能比较模式 0。



定时器比较模式 0

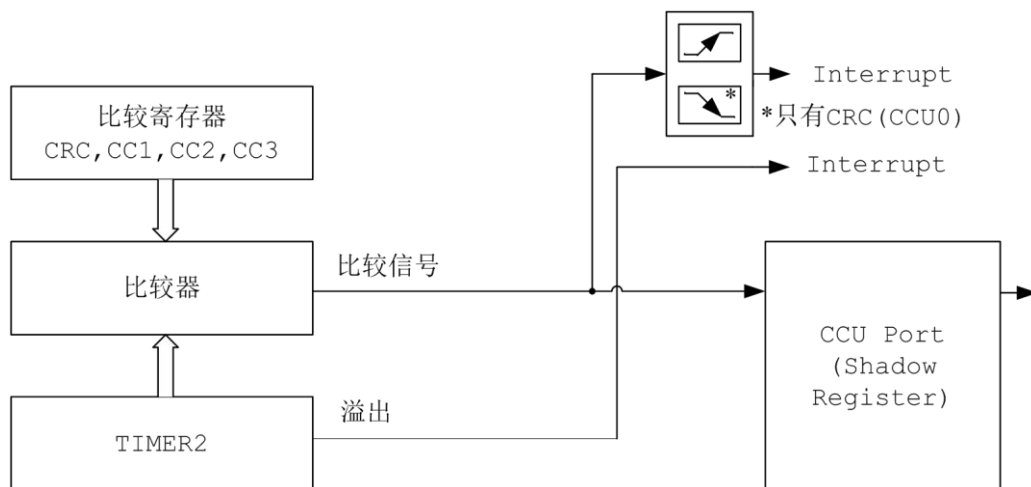


定时器比较模式操作时序

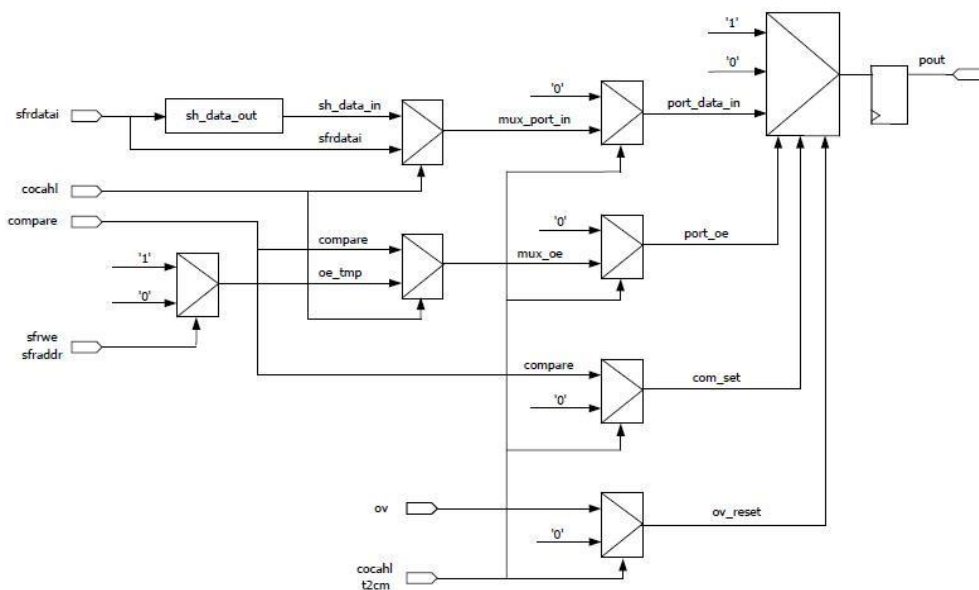
9.3.5.2. 比较模式 1

“T2CM=1”设置比较模式 1。在比较模式 1，定时器 2 溢出不会引起输出变化。在这种模式下的输出信号翻转可以由软件控制。下图显示了在一个寄存器端口配置功能图比较模式 1。

在比较模式 1，输出值第一次写入“影子寄存器(与 CCU 口共用地址)”，当比较信号有效，这个值被传送到输出寄存器。



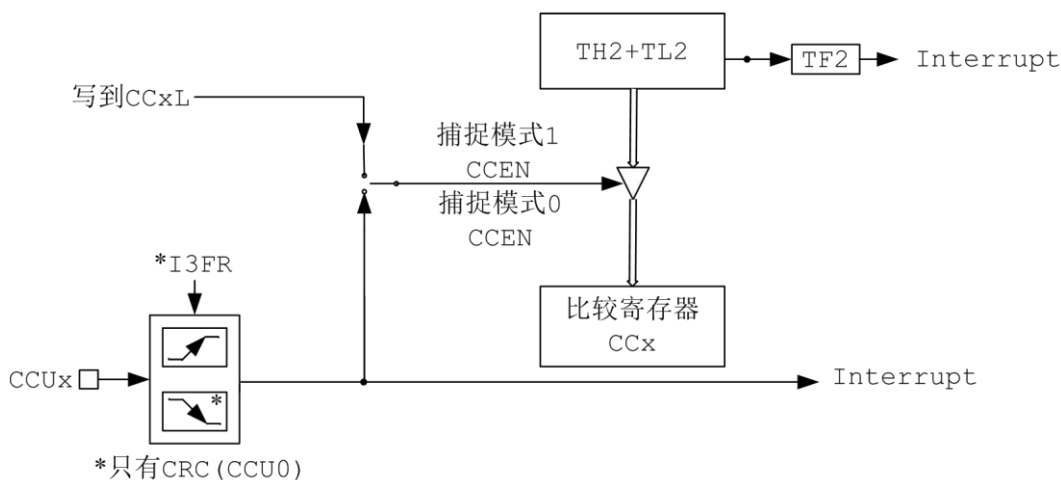
定时器 2 处于比较模式 1



CCU 口结构图

9.3.6. 捕捉功能

每四个 16 位寄存器可以配置 CCU 工作于捕捉模式。在这种模式下，外部事件(模式 0)或软件写操作(模式 1)把定时器/计数器的内容保存到 CCU 寄存器。



定时器 2 处于捕捉模式

9.3.6.1. 捕捉模式 0

在模式 0，捕捉定时器 2 内容由以下条件执行：

- 在输入端口 CC1 检测到上升沿跳变 (CC1 工作于捕捉模式)
 - 在输入端口 CC2 检测到上升沿跳变 (CC2 工作于捕捉模式)
 - 在输入端口 CC3 检测到上升沿跳变 (CC3 工作于捕捉模式)
 - 根据 I3FR 设置，在输入端口 CC0 检测到上升或下降沿跳变 (CC0 工作于捕捉模式)
- 定时器 2 的内容将被锁定到相应的捕捉寄存器。在这种模式下，无中断请求产生。

9.3.6.2. 捕捉模式 1

在模式 1，任何写入专用捕捉寄存器的低位字节引起定时器 2 内容捕捉。这个功能与写入捕捉寄存器的值无关。定时器 2 的内容将被锁定到专用捕捉寄存器。在这种模式下，无中断请求产生。

9.3.7. 定时器 2 寄存器

T2CON 定时器 2 控制寄存器：C8H

初始值：00H

7	6	5	4	3	2	1	0
T2PS	I3FR	-	T2R1	T2R0	T2CM	T2I1	T2I0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 T2PS

定时器 2 预分频选择。

0：定时器 2 时钟频率是主振的 12 分频。

1：定时器 2 时钟频率是主振的 24 分频。

6 I3FR

外部中断 3 有效边沿选择。也用于 CC0 比较/捕捉信号。

0 - 下降沿
1 - 上升沿

4 T2R1:T2R0 定时器 2 重载模式选择。
3 0X - 重载禁止
10 - 模式 0
11 - 模式 1

2 T2CM 定时器 2 比较模式选择。
0 - 模式 0
1 - 模式 1

1 T2I1:T2I0 定时器 2 输入选择。
0 00 定时器 2 停止
01 输入系统时钟的 12 分频或 24 分频
10 定时器 2 在 T2 输入的下降沿增 1
11 T2 输入频率的 12 分频或 24 分频

TL2 定时器 2 低 8 位数据寄存器: CCH 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH2 定时器 2 高 8 位数据寄存器: CDH 初始值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CCEN 比较/捕捉使能寄存器: C1H 初始值: 00H

7	6	5	4	3	2	1	0
COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 COCAH3:COCAL3 CC3 寄存器比较/捕捉模式。
6 00: 比较/捕捉禁止。
01: CC3 上升沿产生捕捉。
10: 比较使能。
11: 写 CCL3 寄存器产生捕捉。

5 COCAH2:COCAL2 CC2 寄存器比较/捕捉模式。
4 00: 比较/捕捉禁止。
01: CC2 上升沿产生捕捉。
10: 比较使能。

11: 写 CCL2 寄存器产生捕捉。

3 COCAH1:COCAL1

CC1 寄存器比较/捕捉模式。

2

00: 比较/捕捉禁止。

01: CC1 上升沿产生捕捉。

10: 比较使能。

11: 写 CCL1 寄存器产生捕捉。

1 COCAH0:COCAL0

CRC 寄存器比较/捕捉模式。

0

00: 比较/捕捉禁止。

01: CC0 上升沿产生捕捉。

10: 比较使能。

11: 写 CRCL 寄存器产生捕捉。

CCL1 比较/捕捉寄存器 CC1 低 8 位: C2H 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CCH1 比较/捕捉寄存器 CC1 高 8 位: C3H 初始值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CCL2 比较/捕捉寄存器 CC2 低 8 位: C4H 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CCH2 比较/捕捉寄存器 CC2 高 8 位: C5H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CCL3 比较/捕捉寄存器 CC3 低 8 位: C6H 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CCH3 比较/捕捉寄存器 CC3 高 8 位: C7H 初始值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-----	-----	-----	-----	-----	-----	-----	-----

CRCL 定时器 2 重载数据寄存器低 8 位: **CAH** 初始值: **00H**

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CRCH 定时器 2 重载数据寄存器高 8 位: **CBH** 初始值: **00H**

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IE 中断使能寄存器: **A8H**

– [ET2 \(IE.5\) 定时器 2 中断使能](#)

9.4. 独立脉冲宽度调制器 PWM0

16 位 PWM 定时器，可设置周期、占空比，可产生周期匹配中断和占空比匹配中断，可连续工作或单次触发工作。脉冲调制信号通过 PWM0 端口输出。

可触发 AD 启动：PWM0 占空比中断触发，PWM0 顶部中断触发，PWM0 底部中断触发。

9.4.1.PWM0 计数时钟

PWM0 的计数时钟分两种，一种是 TIMER2 溢出（T2OV），一种是系统主时钟；计数时钟的选择可以根据寄存器 PWM0_CTRL2.5 进行选择。

9.4.2.PWM0 工作模式

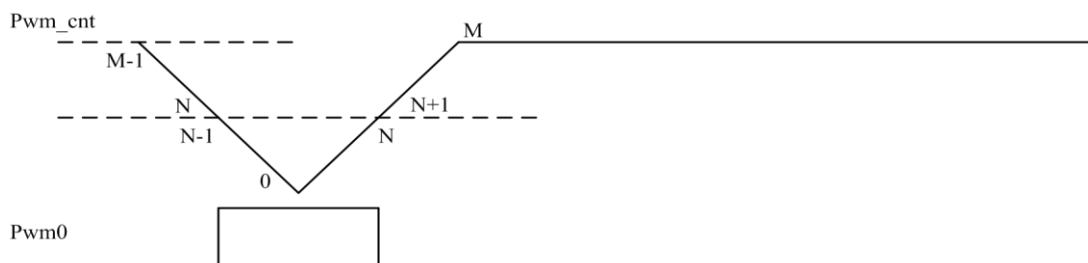
PWM0 的工作模式有单次模式和连续模式，根据 PWM0_CTRL2.4 进行设置。 单次模式 PWM0 只输出一个 PWM 周期；连续模式 PWM0 连续输出 PWM 波形。

9.4.3.PWM0 计数模式

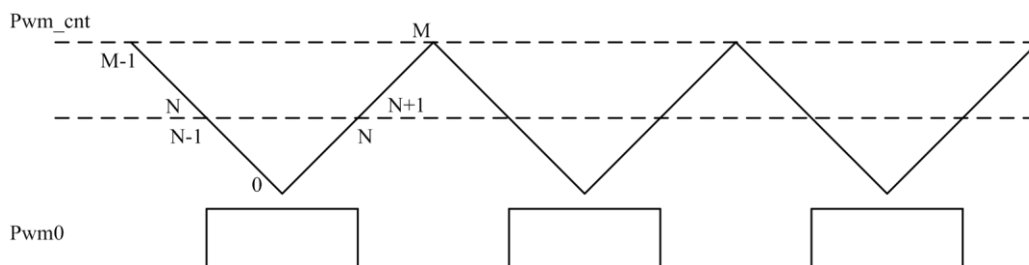
PWM0 包含三角波计数和锯齿波计数两种方式，可根据 PWM0_CTRL2.3 进行设置。

9.4.3.1. 三角波计数方式：（假设周期设置为 **M**，占空比设置为 **N**）。

单次模式（PWM0_EN 使能后）：



连续模式（PWM0_EN 使能后）：



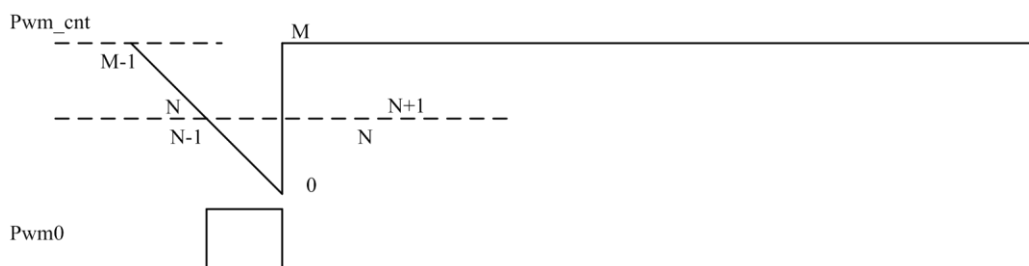
综上，三角波计数时的周期占空比 $= \frac{2 \cdot N}{2 \cdot M} = \frac{N}{M}$ ；下降部分(M-1~0)和上升部分(1~M)，占空比均为 N/M。

当 N=0 时，PWM 输出为 0；占空比为 0%。

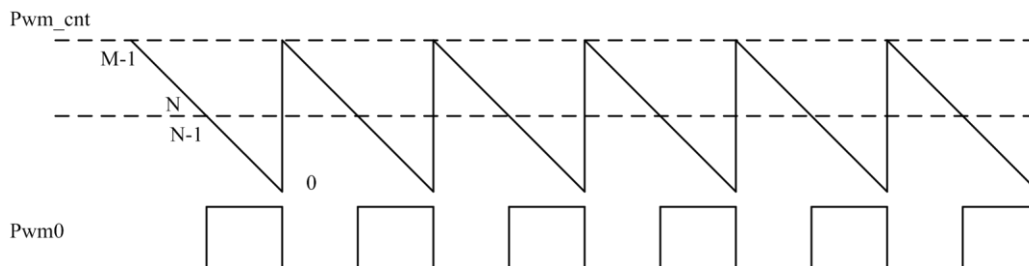
当 N>= M 时，PWM 输出为 1；占空比为 100%。

9.4.3.2. 锯齿波计数方式：（假设周期设置为 **M**，占空比设置为 **N**）

单次模式（PWM0_EN 使能后）：



连续模式（PWM0_EN 使能后）：



综上，锯齿波计数时的周期占空比= N/M ; (同三角波计数时的下降部分($M-1 \sim 0$));

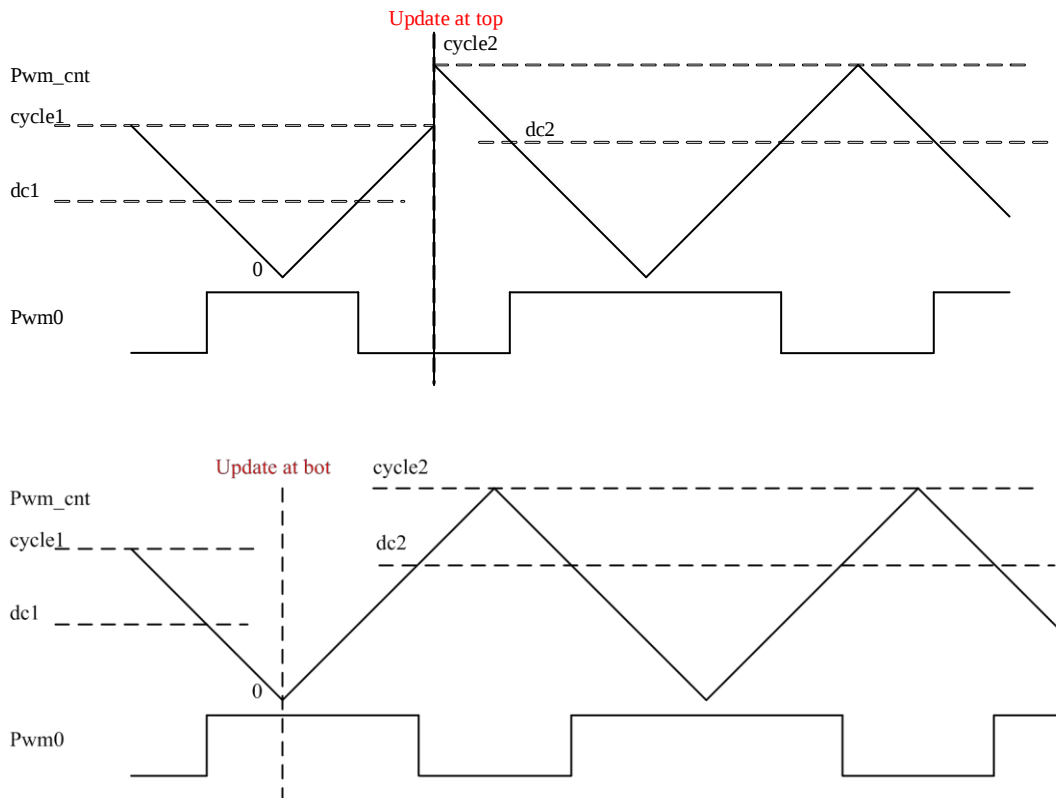
当 $N=0$ 时，pwm 输出为 0；占空比为 0%。

当 $N \geq M$ 时，pwm 输出为 1；占空比为 100%。

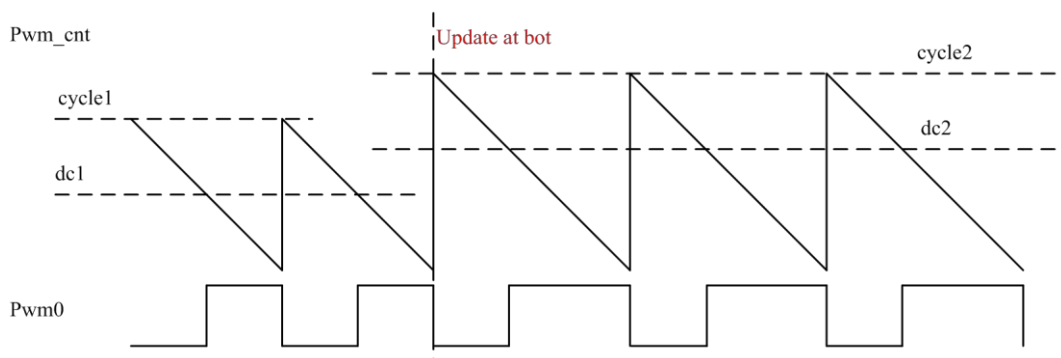
9.4.4.PWM0 更新方式

PWM0 周期和占空比的更新方式有 2 种：计数顶部(Top)更新或者计数底部(Bottom)更新。

当三角波计数方式时，可以底部更新，也可以顶部更新；如下图：

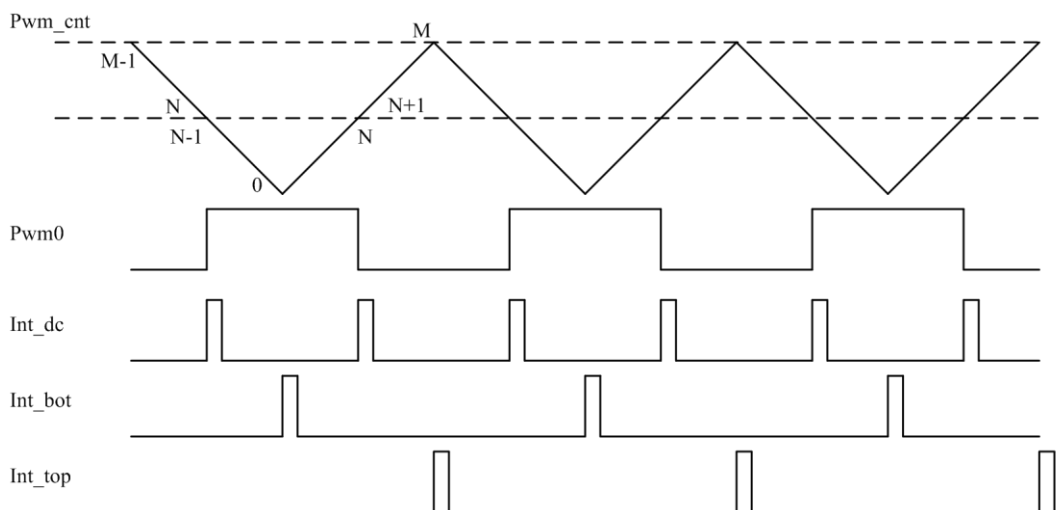


当锯齿波计数方式时，只能底部更新；如下图：



9.4.5.PWM0 中断

PWM0 的中断根据计数方式的不同也有不同方式。对于三角波计数方式：有计数顶部(Top)，计数底部(Bottom)和占空比(DC)三种不同的中断。如下图：



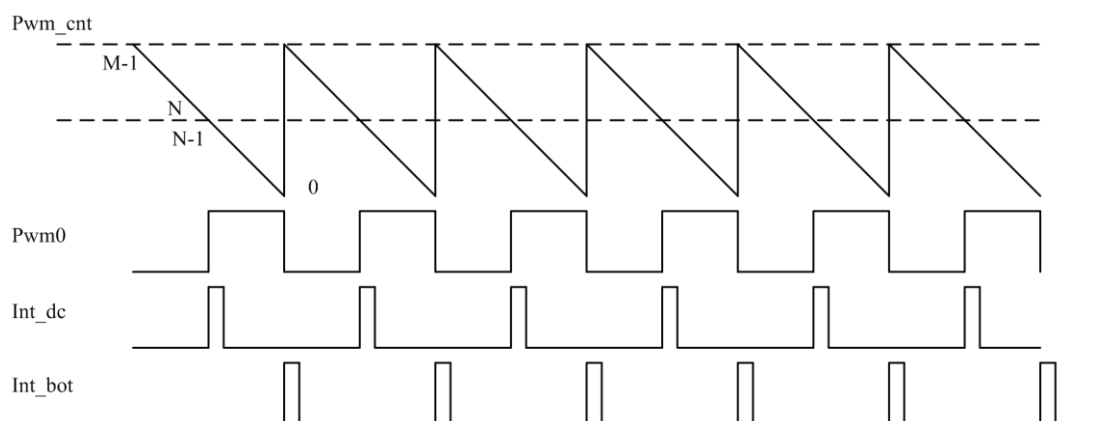
注： M 不可设置为 0

当 $N > M$ 时，没有占空比中断；

当 $N = M$ 时，则占空比中断和计数顶部中断在相同位置；

当 $N = 0$ 时，则占空比中断和计数底部中断在相同位置；

对于锯齿波计数方式：只有计数底部(Bottom)和占空比(DC)两种中断，没有计数顶部(Top)中断。如下图：



注：M 不可设置为 0

当 $N \geq M$ 时，没有占空比中断。

当 $N=0$ 时，则占空比中断和计数底部中断在相同位置。

最终 PWM0 的中断为计数顶部(Top)，计数底部(Bottom)和占空比(DC)三中断的相或。所有中断均有相应的使能控制。硬件置位，软件清 0。

所有的中断产生条件中除了中断允许外，还必须 $PWM0_EN=1$ 。对于计数暂停功能，

正好在中断位置暂停，中断标志若清了，PWM 还在暂停处，则还会产生同样的中断。

9.4.6. PWM0 控制寄存器

PWM0_CYC_H PWM0 周期设置寄存器高 8 位: 2100H 初值: FFH

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM0_CYC_L PWM0 周期设置寄存器低 8 位: 2101H 初值: FFH

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

周期不能设置为 0000H。

PWM0_DC_H PWM0 占空比设置寄存器高 8 位: 2102H 初值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM0_DC_L PWM0 占空比设置寄存器低 8 位: 2103H 初值: 00H

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM0 占空比设置。注意：当设置占空比 = 0 时，PWM0 输出为 0；当设置占空比 $\geq$ 周期时，PWM0 输出 1。

PWM0_CTRL1 PWM0 控制寄存器 1: 2104H 初值: 00H

7	6	5	4	3	2	1	0
PWM0_EN	PWM0_PAUSE	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 PWM0_EN PWM0 使能控制
1 - 使能，开始 PWM 计数。
0 - 停止 PWM，计数器清 0。
对于单次触发，PWM 只计一个周期；对于连续 PWM，该位为 1 时，连续 PWM 计数。

6 PWM0_PAUSE PWM 暂停/继续控制。
1 - 暂停 PWM，计数器保持原有值。
0 - 继续 PWM，计数器继续计数。

PWM0_CTRL2 PWM0 控制寄存器 2: 2105H 初值: 00H

7	6	5	4	3	2	1	0
-	-	CLK_SEL	WORK_MODE	CNT_MODE	UPDATE_AT_TOP	UPDATE_AT_BOTTOM	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

5 CLKSEL PWM0 时钟源选择；(必须在 PWM0_EN=0 时候配置)
0 - 主时钟，根据 CLK_CTRL 选择。
1 - 定时器 2 溢出作为时钟。

4 WORK_MODE 工作模式选择：(必须在 PWM0_EN=0 时候配置)
1 - 单次触发模式。
0 - 连续工作模式。

3 CNT_MODE 计数模式：(必须在 PWM0_EN=0 时候配置)
0 - 锯齿波模式。
1 - 三角波模式 (与 MPWM 相同)。

2 UPDATE_AT_TOP 计数顶部 (TOP: 计数值=当前周期设置值) 时，更新 (周期和占空比) 使能。
1 - 使能。

0 - 禁止。

1 UPDATE_AT_BOT 计数低部 (BOT: 计数值=0) 时, 更新 (周期和占空比) 使能;

1 - 使能。

0 - 禁止。

PWM0_INT_EN PWM0 中断控制寄存器: E5H 初值: 00H

7	6	5	4	3	2	1	0
-	-	-	-	-	DC_INT _EN	TOP_INT _EN	BOT_IN T_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

2 DC_INT_EN PWM0 占空比匹配 (计数器值=占空比设置值) 中断使能。

1-使能。

0-禁止。

1 TOP_INT_EN PWM0 顶部 (计数器值=周期值) 中断使能控制

1-使能

0-禁止

0 BOT_INT_EN PWM0 低部 (计数器值=0) 中断允许控制

1-使能

0-禁止

注意: 使用上面中断时必须打开 IEN2.3

PWM0_INT_FLG PWM0 中断标志寄存器: E6H 初值: 00H

7	6	5	4	3	2	1	0
-	-	-	-	-	DC_INT _FLG	TOP_INT _FLG	BOT_IN T_FLG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

2 DC_INT_FLG PWM0 占空比匹配中断标志, 硬件置位, 软件清 0。

1 TOP_INT_FLG PWM0 顶部中断标志, 硬件置位, 软件清 0。对于锯齿波计数方式, 无顶部中断, 因此该位无效, 一直为 0。

0 BOT_INT_FLG PWM0 底部中断标志 (只有中心对齐模式才有效), 硬件置位, 软件清 0。

9.5. 独立脉冲宽度调制器 PWM1、PWM2

8 位 PWM 定时器，可设置周期、占空比，可产生周期匹配中断和占空比匹配中断，可连续工作或单次触发工作。脉冲调制信号通过 PWM1、PWM2 端口输出。

9.5.1. PWM1 控制寄存器

PWM1_CYC PWM1 周期设置寄存器: 2200H

初值: FFH

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

周期不能设置为 00H。

PWM1_DC PWM1 占空比设置寄存器: 2201H

初值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM1 占空比设置。注意：当设置占空比 = 0 时，PWM1 输出为 0；当设置占空比 $\geq$ 周期时，PWM1 输出 1。

PWM1_CTRL1 PWM1 控制寄存器 1: 2202H

初值: 00H

7	6	5	4	3	2	1	0
PWM1_EN	PWM1_PAUSE	-	-	PREDIV3	PREDIV2	PREDIV1	PREDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 PWM1_EN PWM 使能控制
1 - 使能，开始 PWM 计数。
0 - 停止 PWM，计数器清 0。
对于单次触发，PWM 只计一个周期；对于连续 PWM，该位为 1 时，连续 PWM 计数。

6 PWM1_PAUSE PWM 暂停/继续控制。
1 - 暂停 PWM，计数器保持原有值。
0 - 继续 PWM，计数器继续计数。

3:0 PREDIV PWM 计数 60MHz 时钟预分频控制。
1~12, 2^{PREDIV} 分频。
其他值没有预分频，用 60MHz 时钟。

PWM1_CTRL2 PWM1 控制寄存器 2: 2203H

初值: 00H

7	6	5	4	3	2	1	0
-	-	-	WORK_MODE	CNT_MODE	UPDATE_AT_TOP	UPDATE_AT_BOTTOM	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

-
- 4 WORK_MODE 工作模式选择: (必须在 PWM_EN=0 时候配置)
 1 - 单次触发模式。
 0 - 连续工作模式。
-
- 3 CNT_MODE 计数模式: (必须在 PWM_EN=0 时候配置)
 0 - 锯齿波模式。
 1 - 三角波模式 (与 MPWM 相同)。
-
- 2 UPDATE_AT_TOP 计数顶部 (TOP: 计数值=当前周期设置值) 时, 更新 (周期和占空比) 使能。
 1 - 使能。
 0 - 禁止。
-
- 1 UPDATE_AT_BOTTOM 计数低部 (BOT: 计数值=0) 时, 更新 (周期和占空比) 使能;
 1 - 使能。
 0 - 禁止。
-

PWM1_INT_EN PWM1 中断控制寄存器: 2204H

初值: 00H

7	6	5	4	3	2	1	0
-	-	-	-	-	DC_INT_EN	TOP_INT_EN	BOTTOM_INT_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

-
- 2 DC_INT_EN PWM1 占空比匹配 (计数器值=占空比设置值) 中断使能。
 1-使能。
 0-禁止。
-
- 1 TOP_INT_EN PWM1 顶部 (计数器值=周期值) 中断使能控制
 1-使能
 0-禁止
-
- 0 BOTTOM_INT_EN PWM1 低部 (计数器值=0) 中断允许控制
 1-使能
 0-禁止
-

注意：使用上面中断时必须打开 IEN2.3

PWM1_INT_FLG PWM1 中断标志寄存器：2205H 初值：00H

7	6	5	4	3	2	1	0
-	-	-	-	-	DC_INT_FLG	TOP_INT_FLG	BOT_INT_FLG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

2 DC_INT_FLG PWM1 占空比匹配中断标志，硬件置位，软件清 0。

1 TOP_INT_FLG PWM1 顶部中断标志，硬件置位，软件清 0。对于锯齿波计数方式，无顶部中断，因此该位无效，一直为 0。

0 BOT_INT_FLG PWM1 底部中断标志 (只有中心对齐模式才有效)，硬件置位，软件清 0。

9.5.2. PWM2 控制寄存器

PWM2_CYC PWM2 周期设置寄存器：2300H 初值：FFH

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

周期不能设置为 00H。

PWM2_DC PWM2 占空比设置寄存器：2301H 初值：00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM1 占空比设置。注意：当设置占空比 = 0 时，PWM1 输出为 0；当设置占空比 ≥ 周期时，PWM1 输出 1。

PWM2_CTRL1 PWM2 控制寄存器 1：2302H 初值：00H

7	6	5	4	3	2	1	0
PWM2_EN	PWM2_PA USE	-	-	PREDIV3	PREDIV2	PREDIV1	PREDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 PWM2_EN PWM 使能控制
1 - 使能，开始 PWM 计数。
0 - 停止 PWM，计数器清 0。

对于单次触发，PWM 只计一个周期；对于连续 PWM，该位为 1 时，连续 PWM 计数。

-
- 6 PWM2_PAUSE PWM 暂停/继续控制。
- 1 - 暂停 PWM，计数器保持原有值。
 - 0 - 继续 PWM，计数器继续计数。
-

- 3:0 PREDIV PWM 计数 60MHz 时钟预分频控制。
- 1~12, 2^{PREDIV} 分频。
- 其他值没有预分频，用 60MHz 时钟。
-

PWM2_CTRL2 PWM2 控制寄存器 2: 2303H

初值: 00H

7	6	5	4	3	2	1	0
-	-	-	WORK_MODE	CNT_MODE	UPDATE_AT_TOP	UPDATE_AT_BOTTOM	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

-
- 4 WORK_MODE 工作模式选择：（必须在 PWM_EN=0 时候配置）
- 1 - 单次触发模式。
 - 0 - 连续工作模式。
-
- 3 CNT_MODE 计数模式：（必须在 PWM_EN=0 时候配置）
- 0 - 锯齿波模式。
 - 1 - 三角波模式（与 MPWM 相同）。
-
- 2 UPDATE_AT_TOP 计数顶部 (TOP: 计数值=当前周期设置值) 时，更新 (周期和占空比) 使能。
- 1 - 使能。
 - 0 - 禁止。
-
- 1 UPDATE_AT_BOTTOM 计数低部 (BOT: 计数值=0) 时，更新 (周期和占空比) 使能；
- 1 - 使能。
 - 0 - 禁止。
-

PWM2_INT_EN PWM2 中断控制寄存器: 2304H

初值: 00H

7	6	5	4	3	2	1	0
-	-	-	-	-	DC_INT_EN	TOP_INT_EN	BOTTOM_INT_EN

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-----	-----	-----	-----	-----	-----	-----	-----

2 DC_INT_EN PWM2 占空比匹配（计数器值=占空比设置值）中断使能。
1-使能。
0-禁止。

1 TOP_INT_EN PWM2 顶部（计数器值=周期值）中断使能控制
1-使能
0-禁止

0 BOT_INT_EN PWM2 低部（计数器值=0）中断允许控制
1-使能
0-禁止

注意：使用上面中断时必须打开 IEN2.3

PWM2_INT_FLG PWM2 中断标志寄存器：2205H 初值：00H

7	6	5	4	3	2	1	0
-	-	-	-	-	DC_INT _FLG	TOP_INT _FLG	BOT_IN T_FLG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

2 DC_INT_FLG PWM2 占空比匹配中断标志，硬件置位，软件清 0。

1 TOP_INT_FLG PWM2 顶部中断标志，硬件置位，软件清 0。对于锯齿波计数方式，无顶部中断，因此该位无效，一直为 0。

0 BOT_INT_FLG PWM2 底部中断标志（只有中心对齐模式才有效），硬件置位，软件清 0。

9.6. 看门狗 (Watch-dog)

看门狗定时器由 15 位计数器（非 SFR）、重载寄存器 (WDTR_EL)、2/16 的预分频器和控制逻辑组成。看门狗定时器的计数速率取决于 WDTR_EL 寄存器最高位。当 WDTR_EL 最高位为 1，看门狗定时器每递增 $1232 = 384$ 个时钟周期，这使得整个周期为 $12 \times 32 \times 256 \times 128 = 12582912$ 个时钟周期长。当 WDTR_EL 为 0，看门狗定时器每递增 $12 \times 2 = 24$ 个时钟周期，这使得整个周期为 $12 \times 2 \times 256 \times 128 = 786432$ 个时钟周期长。当 WDT_TM 测试模式的输入设置为 1，看门狗定时器的输入时钟分频（所有分频 1/12、1/8、1/16）被忽略，缩短看门狗溢出所需的时间。

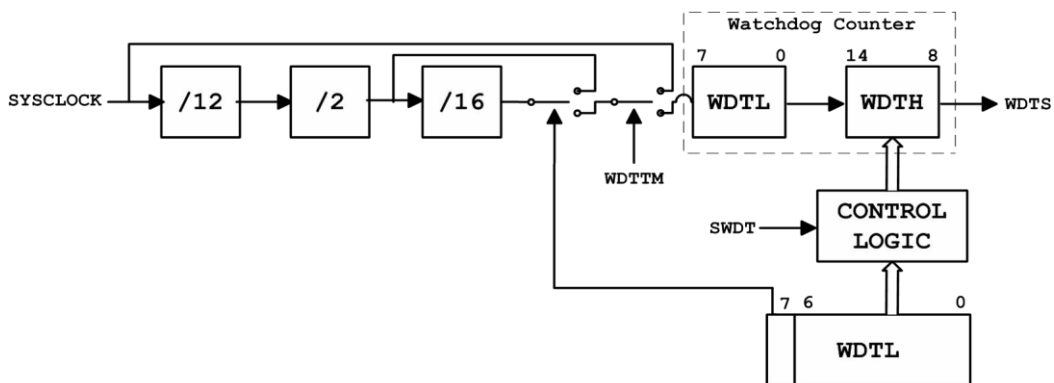
有两种方法来启动看门狗。一种方法，称为硬件自动启动，是基于对 SWD 输入信号电平在外部有效复位信号。当 SWD 高直到复位信号变低，在复位后，看门狗定时器自动默认启动（所有的寄存器设置为零）。启动看门狗定时器，另一种是由软件。设置 SWDT 标志（IEN1.6, IEN1 寄存器的第六位）启动看门狗定时器。设置 SWDT 标志只启动看门狗定时器，不重载看门狗定时器。一旦开始，看门狗定时器只能被 SWD 和复位引脚硬件复位强迫停止。当看门狗计数器计从 7FFFH 溢出到 0000H，产生内部复位。WDTS 标志（IP0.6, IP0 寄存器第六位）被硬件设置，外部复位信号消失后清除。WDTS 信号不复位看门狗，它一直维持。WDTS 标志是为了让软件判断复位是由外部引脚复位或看门狗超时复位。WDTS 标志也可以由软件清除。

看门狗开启后，必须定期刷新防止复位，首先设置 WDTWDT (IEN0.6)，然后设置 SWDT (IEN1.6)。WDTEEL 值越小，超时复位时间越长；WDTREL 值 7FH，超时复位时间最短。刷新看门狗 C 程序如下：

```
sfr IEN0 = 0xA8;
sfr IEN1 = 0xB8;
sbit WDT = IEN0^6;
sbit SWDT = IEN1^6;

#define CLRWDT() { WDT = 1; SWDT = 1; }
```

如果单片机处于停止 (STOP) 模式，WDT 超时事件将唤醒单片机并使其继续执行 STOP 后面的指令。通过将配置位 WDTE 清 0/ 置 1，可关闭 / 打开 WDT。



9.6.1. 看门狗寄存器

标识符	功能	地址	初始值
WDTREL	看门狗控制寄存器	86H	00H
IE	中断使能寄存器	A8H	00H
IEN1	中断使能寄存器 1	B8H	00H

WDTREL 看门狗控制寄存器：86H

7	6	5	4	3	2	1	0
WDTPS	WDTRV						

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-----	-----	-----	-----	-----	-----	-----	-----

-
- | | | |
|---|-------|--|
| 7 | WDTPS | 看门狗定时器输入时钟分频选择
1 - 看门狗定时器输入时钟被 16 分频
0 - 无分频 |
|---|-------|--|
-
- | | | |
|---|-------|---|
| 6 | WDTRV | 看门狗定时器重载值
看门狗最高 7 位的重载值。当 WDT (IE.6)=1 并且 WDT (IEN1.6)=1，
一个刷新出现重载值装载到看门狗定时器。 |
|---|-------|---|
-

IEN0 中断使能寄存器 0: A8H

- [WDT \(IEN0.6\) 看门狗定时器刷新标志](#)

IEN1 中断使能寄存器 1: B8H

- [SWDT \(IP.6\) Watchdog 定时器“启动/刷新”标志。](#)

PCON 电源控制寄存器: 87H

- [WDTTM \(PCON.6\) 看门狗测试模式标志](#)

10. 变频马达控制功能

此功能内建一个 16 位电机控制 PWM 模块。PWM 模块由时基模块、波形发生模块、死区和极性控制模块，故障检测以及输出控制模块组成。

时基模块中通过一个 16 位计数器对 PWM 时钟计数，用该计数器的值与周期寄存器比较以产生周期匹配信号与周期重启信号、与占空比寄存器比较以产生占空比匹配信号，与 PWM_CM 寄存器比较以产生周期中任意点匹配信号，可产生 6 路原始 PWM 信号，该原始 PWM 信号经过死区和极性控制模块后产生带死区和极性的 PWM 信号，最后的输出控制模块和故障检测模块决定是否将对应的信号输出至 PWM 引脚上。

保护模块可以使能或禁止，可以使能比较器保护、引脚 G0FF0、引脚 G0FF1 输入有效沿触发六路 PWM 输出引脚高阻。

10.1. 脉冲宽度调制定时器 (MPWM)

1. 三路互补输出 PWM 定时器
2. 输入时钟频率最高：60MHz
3. 16 位分辨率的 PWM 信号，可以设置周期、占空比。
4. 可进行 0% - 100% 输出
5. 三角波调制 (中心对齐模式)
6. 死区时间设定
7. 启动 AD 转换功能，MPWM 周期计数器值和 PWM_CMx 值匹配时中断触发 ADC，MPWM 占空比中断触发，MPWM 顶部中断触发，MPWM 底部中断触发。
8. PWM 一个周期启动同步消隐延时时间设定
9. 模拟比较器输出高电平可强制 IO 口 (GAT、GAB、GBT、GBB、GCT、GCB) 为高阻
10. 外部高电平信号加到 G0FF 端口，可强制 IO 口 (GAT、GAB、GBT、GBB、GCT、GCB) 为高阻。
11. 互补输出每个端口可正反向设置
12. 互补输出每个端口可设置高端 PWM 输出、低端 PWM 输出、高低端同时 PWM 输出
13. Duty 带 Buffer, Buffer 可用用户程序更新。Duty Buffer 内容在 PWM 开始中点 (中心对齐模式) 写入 Duty 比较器。
14. 在 Duty 中点产生中断；Duty 结束产生中断；这些信号可屏蔽，可启动 AD 转换。
15. 可工作于连续输出模式或单次触发输出模式

10.1.1. MPWM 计数时钟

MPWM 的计数时钟分两种，一种是 TIMER2 溢出 (T2OV)，一种是系统主时钟；计数时钟的选择可以根据寄存器 PWM_CTRL2.5 进行选择。

10.1.2. MPWM 工作模式

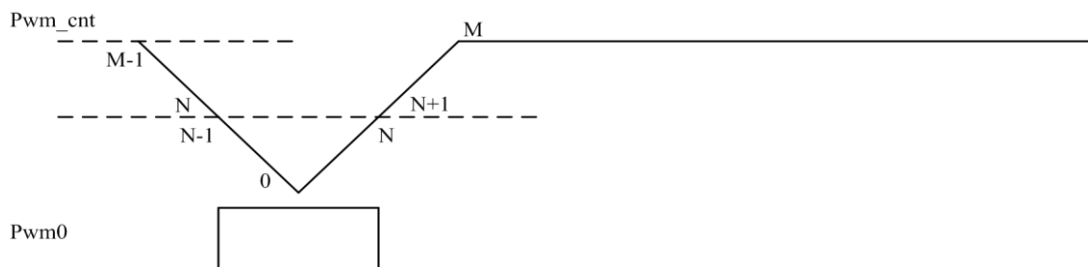
MPWM 的工作模式有单次模式和连续模式，根据 PWM_CTRL2.4 进行设置。单次模式 MPWM 只输出一个 PWM 周期；连续模式 MPWM 连续输出 PWM 波形。

10.1.3. MPWM 计数模式

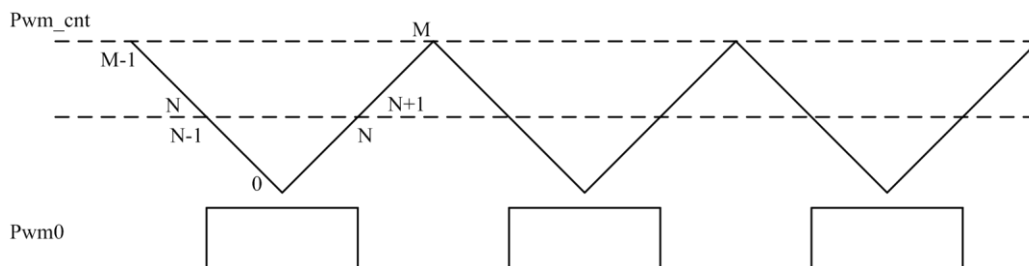
MPWM 只有三角波计数方式。

10.1.3.1. 三角波计数方式：（假设周期设置为 **M**，占空比设置为 **N**）。

单次模式（PWM_EN 使能后）：



连续模式（PWM_EN 使能后）：



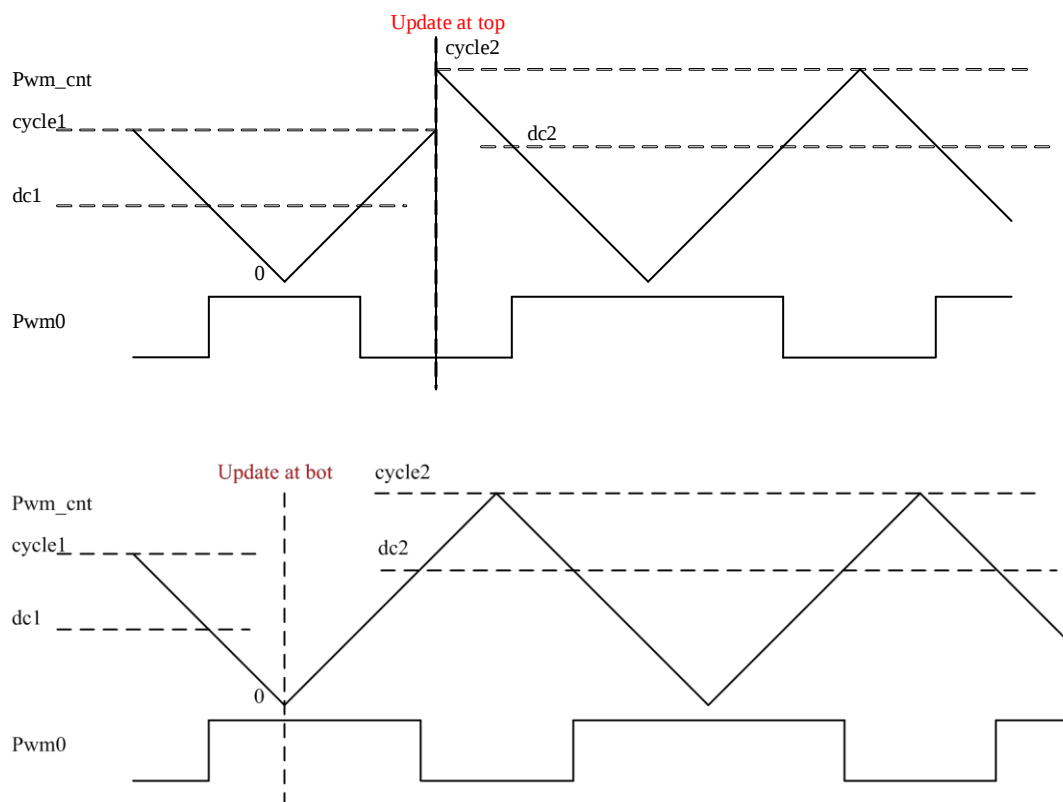
综上，三角波计数时的周期占空比 $= \frac{2 \cdot N}{2 \cdot M} = \frac{N}{M}$ ；下降部分(M-1~0)和上升部分(1~M)，占空比均为 N/M 。

当 $N=0$ 时，PWM 输出为 0；占空比为 0%。

当 $N \geq M$ 时，PWM 输出为 1；占空比为 100%。

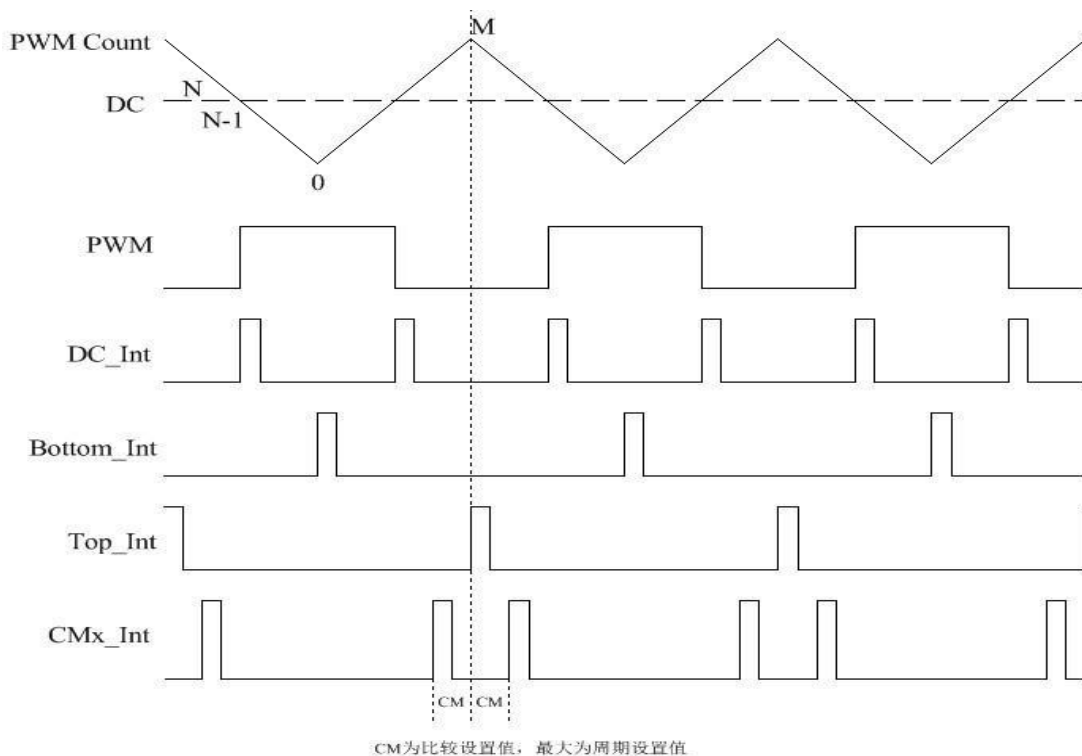
10.1.4. MPWM 更新方式

MPWM 周期和占空比的更新方式有 2 种：计数顶部(Top)更新或者计数底部(Bottom)更新。如下图：



10.1.5. MPWM 中断

MPWM 的中断有计数顶部(Top)，计数底部(Bottom)、占空比(DC)和比较中断。如下图：



注：M 不可设置为 0

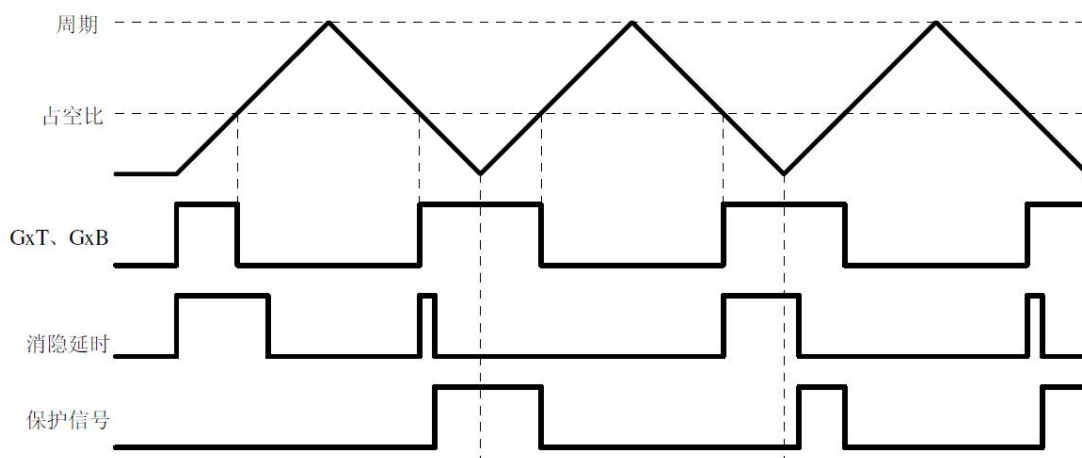
当 $N > M$ 时，没有占空比中断；

当 $N = M$ 时，则占空比中断和计数顶部中断在相同位置；

当 $N = 0$ 时，则占空比中断和计数底部中断在相同位置；

CM 比较值最大为周期设置值，在上计数和下计数各产生一次中断。

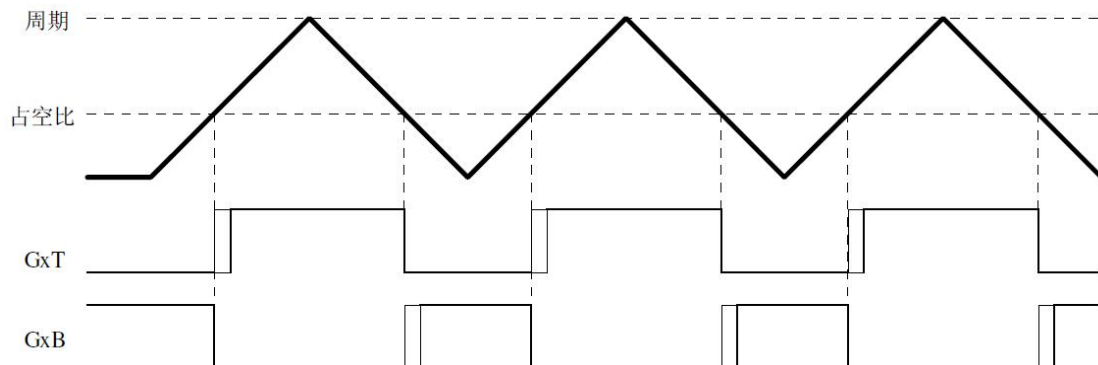
10.2. 消隐



10.3. 死区插入电路

死区插入设计的目的：死区插入电路确保外部门驱动电路晶体管对在转态时不会瞬间导通（上下臂 MOS 都开启），产生短路电流。为了消除这种危险，设计了一段死区时间，确保输出转态的过程中，两个晶体管都处于不导通的状态。死区时间要控制在 $0.3\mu\text{s} \sim 5\mu\text{s}$ 左右。

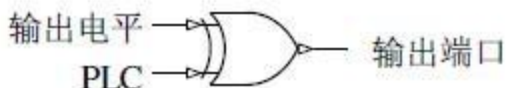
在上升沿插入死区，下降沿不需要。



10.4. 极性电路

利用 PLC 寄存器可决定 GAT、GAB、GBT、GBB、GCT、GCB 输出有效电平，以支持 外部门驱动电路多种接口电平选择。

GAT、GAB、GBT、GBB、GCT、GCB 输出引脚复位默认值是高阻。



10.5. MPWM 的控制寄存器

PWM_CYC_H PWM 周期设置寄存器高 8 位：F9H

初始值：FFH

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM_CYC_L PWM 周期设置寄存器低 8 位：F9H

初始值：FFH

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

设置周期顺序是先高字节后低字节，也可只配置低字节。

PWM_DC1_H PWM 通道 1 占空比设置寄存器高 8 位: F2H 初始值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM_DC1_L PWM 通道 1 占空比设置寄存器低 8 位: F1H 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

设置 PWM 通道 1 占空比。注意：当设置占空比 = 0 时，PWM 输出为 0；当设置占空比 $\geq$ 周期时，PWM 输出。

PWM_DC2_H PWM 通道 2 占空比设置寄存器高 8 位: F4H 初始值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM_DC2_L PWM 通道 2 占空比设置寄存器低 8 位: F3H 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

设置 PWM 通道 2 占空比。注意：当设置占空比 = 0 时，PWM 输出为 0；当设置占空比 $\geq$ 周期时，PWM 输出。

PWM_DC3_H PWM 通道 3 占空比设置寄存器高 8 位: F6H 初始值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM_DC3_L PWM 通道 3 占空比设置寄存器低 8 位: F5H 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

设置 PWM 通道 3 占空比。注意：当设置占空比 = 0 时，PWM 输出为 0；当设置占空比 $\geq$ 周期时，PWM 输出。

PWM_CM1_H PWM 比较 1 设置寄存器高 8 位: FCH 初始值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM_CM1_L PWM 比较 1 设置寄存器低 8 位: FBH 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM 比较值 1 设置, 周期计数器值与比较值匹配, 最大为周期设置值, 在周期上计数和下计数期间各产生一次中断, 可触发 ADC。详见寄存器 PWM_CTRL2[7:6] (TRIG_ADC_CTRL) 说明。

PWM_CM2_H PWM 比较 2 设置寄存器高 8 位: FEH 初始值: 00H

7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM_CM2_L PWM 比较 2 设置寄存器低 8 位: FDH 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM 比较值 2 设置, 周期计数器值与比较值匹配, 最大为周期设置值, 在周期上计数和下计数期间各产生一次中断, 可触发 ADC。详见寄存器 PWM_CTRL2[7:6] (TRIG_ADC_CTRL) 说明。

PWM_CTRL1 PWM 控制寄存器 1: A7H 初始值: 00H

7	6	5	4	3	2	1	0
PWM_EN	PWM_PAUSE	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 PWM_EN

PWM 使能控制

0 停止 PWM, 计数器清零

1 使能 PWM, 开始 PWM 计数

6 PWM_PAUSE

0 PWM 继续工作, 计数器继续计数

1 暂停 PWM 计数, 计数器保持原有值, PWM 输出保持先前状态

PWM_CTRL2 PWM 控制寄存器 2: 9FH 初始值: 00H

7	6	5	4	3	2	1	0
TRIG_ADC_CTRL	CLK_SEL	WORK_MO	—	UPDATE	UPDATE	—	—

			DE		_AT_TO P	_AT_BO T	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:6 TRIG_ADC_CTRL 00-不触发 ADC
01-周期计数器值和 PWM_CM1 值匹配时触发 ADC
10-周期计数器值和 PWM_CM2 值匹配时触发 ADC
11-PWM_CM1, PWM_CM2 匹配时均触发 ADC
(注: 触发 adc 的控制和 cm1, cm2 的中断无关;)

5 CLK_SEL MPWM 时钟源选择; (必须在 PWM_EN=0 时候配置)
0-主时钟, 根据 CLK_CTRL 控制
1-定时器 2 溢出作为时钟

4 WORK_MODE 工作模式选择: (必须在 PWM_EN=0 时候配置)
1-单次触发
0-连续工作模式

2 UPDATE_AT_TOP PWM 计数顶部 (TOP: 计数器值=周期设置值) 时, 更新 (周期或者占空比) 使能
1-使能
0-禁止

1 UPDATE_AT_BOT PWM 计数底部 (Bot: 计数器值=0) 时, 更新 (周期或者占空比) 使能。
1-使能
0-禁止

PWM_INT_EN PWM 中断使能寄存器: 9EH 初始值: 00H

7	6	5	4	3	2	1	0
-	CM2_IN T_EN	CM1_IN T_EN	DC3_IN T_EN	DC2_IN T_EN	DC1_IN T_EN	TOP_IN T_EN	BOT_IN T_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

6 CM2_INT_EN PWM 比较值 2 匹配中断使能;
1-使能
0-禁止

5 CM1_INT_EN PWM 比较值 1 匹配中断使能;
1-使能
0-禁止

4 DC3_INT_EN PWM 通道 3 占空比匹配中断使能;

1-使能
0-禁止

3 DC2_INT_EN PWM 通道 2 占空比匹配中断使能;
1-使能
0-禁止

2 DC1_INT_EN PWM 通道 1 占空比匹配中断使能;
1-使能
0-禁止

1 TOP_INT_EN 顶部 中断允许控制（中心对齐模式）。
1-使能
0-禁止

0 BOT_INT_EN 底部中断允许控制（中心对齐模式）。
1-使能
0-禁止

PWM_INT_FLG PWM 中断状态寄存器: 96H 初始值: 00H

7	6	5	4	3	2	1	0
—	CM2_IN T_FLG	CM1_IN T_FLG	DC3_IN T_FLG	DC2_IN T_FLG	DC1_IN T_FLG	TOP_IN T_FLG	BOT_IN T_FLG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

6 CM2_INT_FLG PWM 比较值 2 匹配中断标志; 硬件置位, 软件清 0。

5 CM1_INT_FLG PWM 比较值 1 匹配中断标志; 硬件置位, 软件清 0。

4 DC3_INT_FLG PWM 通道 3 占空比匹配中断标志; 硬件置位, 软件清 0。

3 DC2_INT_FLG PWM 通道 2 占空比匹配中断标志; 硬件置位, 软件清 0。

2 DC1_INT_FLG PWM 通道 1 占空比匹配中断标志; 硬件置位, 软件清 0。

1 TOP_INT_FLG 顶部/周期中断标志; 硬件置位, 软件清 0。

0 BOT_INT_FLG 底部中断标志; 硬件置位, 软件清 0。

POL_CTRL PWM 输出极性控制寄存器: B3H 初始值: 00H

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

—	—	GAT_POL	GAB_POL	GBT_POL	GBB_POL	GCT_POL	GCB_POL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

5:0 GAT、GAB、GBT、GBB、GCT、GCB 输出极性控制。
0 - 正极性
1 - 负极性

LEV_BUF PWM 输出电平控制寄存器: B4H 初始值: 00H

7	6	5	4	3	2	1	0
—	—	GAT_LEV	GAB_LEV	GBT_LEV	GBB_LEV	GCT_LEV	GCB_LEV
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

5:0 GAT、GAB、GBT、GBB、GCT、GCB 输出有效电平控制。

OUTPUT_MODE PWM 输出模式控制寄存器: B5H 初始值: 00H

7	6	5	4	3	2	1	0
PWM_SE L	CO_EN	GAT_MD	GAB_MD	GBT_MD	GBB_MD	GCT_MD	GCB_MD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 PWM_SEL 0 - PWM 输出时, 通道 PWM1-PWM3 均用到。
1 - PWM 输出时, 只用 PWM1。

6 CO_EN 互补输出使能。
0 - 非互补输出。
1 - GAT、GAB 互补, GBT、GBB 互补, GCT、GCB 互补输出。

5 GAT_MD GAT 电平输出或 PWM 调制输出控制。
0 - 电平输出 (电平值由 LEV_BUF.5 控制)。
1 - PWM 输出。
若 CO_EN=1, 则 GAT 和 GAB 互补。
若 CO_EN=0, 则 GAT 和 GAB 有相同的波形。

4 GAB_MD GAB 电平输出或 PWM 调制输出控制。
0 - 电平输出 (电平值由 LEV_BUF.4 控制)。
1 - PWM 输出。
若 CO_EN=1, 则 GAT 和 GAB 互补。
若 CO_EN=0, 则 GAT 和 GAB 有相同的波形。

3 GBT_MD GBT 电平输出或 PWM 调制输出控制。
0 - 电平输出 (电平值由 LEV_BUF.3 控制)。
1 - PWM 输出。

若 CO_EN=1, 则 GBT 和 GBB 互补。
若 CO_EN=0, 则 GBT 和 GBB 有相同的波形。

2 GBB_MD GBB 电平输出或 PWM 调制输出控制。
0 - 电平输出 (电平值由 LEV_BUF.2 控制)。
1 - PWM 输出。
若 CO_EN=1, 则 GBT 和 GBB 互补。
若 CO_EN=0, 则 GBT 和 GBB 有相同的波形。

1 GCT_MD GCT 电平输出或 PWM 调制输出控制。
0 - 电平输出 (电平值由 LEV_BUF.1 控制)。
1 - PWM 输出。
若 CO_EN=1, 则 GCT 和 GCB 互补。
若 CO_EN=0, 则 GCT 和 GCB 有相同的波形。

0 GCB_MD GCB 电平输出或 PWM 调制输出控制。
0 - 电平输出 (电平值由 LEV_BUF.0 控制)。
1 - PWM 输出。
若 CO_EN=1, 则 GCT 和 GCB 互补。
若 CO_EN=0, 则 GCT 和 GCB 有相同的波形。

OUTPUT_UPCTRL PWM 输出更新控制寄存器: B6H 初始值: 00H

7	6	5	4	3	2	1	0
SFTW_T RIG_UP DATE	—	—	—	—	UPDATE_CTRL		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 SFTW_TRIG_UPDATE 软件触发更新。0->1, 触发一次更新。硬件清 0; 该位读为 0。

2:0 当寄存器“LEV_BUF, 输出模式”配置有变化时 (不包含 PWM_SEL 寄存器), GAT~GCB 输出更新控制:
001-MPWM 顶部 (Top) 更新;
010-MPWM 底部 (Bottom) 更新;
011-实时更新;
110-定时器 2 中断更新;
111-保留; (不更新)

PROT_EN 保护模式使能寄存器: ABH 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

保护模式使能控制。

A5H - 保护模式使能。

5AH - 保护模式禁止。寄存器“GOFF_CTRL_EN、CMP_CTRL_EN”无效，无 GOFF 和 CMP 的保护机制。

其他值 - 保持原有状态。

PROT_CTRL1 MPWM 保护控制寄存器 1: ACH 初始值: 00H

7	6	5	4	3	2	1	0
HIGHZ_ INT_EN	CLEAR_ HIGHZ_ CTRL	SFTW_ TRIG_H Z_EN	GOFF1_ EN	GOFF0_ EN	CMP_EN	GOFF1_ TRIG_C TRL	GOFF0_ TRIG_C TRL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 HIGHZ_INT_EN GAT~GCB 输出高阻中断使能；
1-使能，输出为高阻时，产生中断；
0-禁止，输出为高阻时，不产生中断；

6 CLEAR_HIGHZ_CTRL 保护释放控制，即从保护模式恢复正常输出的控制。
1 - 当比较器或者 GOFF 无效沿产生时，即恢复正常的输出。
0 - 由保护模式进入高阻后，由用户软件设置是否恢复正常输出，即由 SFTW_TRIG_OUTPUT 寄存器控制。

5 SFTW_TRIG_HZ_EN 软件触发高阻输出使能。
1 - 允许软件触发输出高阻；SFTW_TRIG_HIGHZ 寄存器有效。
0 - 禁止软件触发高阻；SFTW_TRIG_HIGHZ 寄存器无效。

4 GOFF1_EN GOFF1 引脚保护使能。
1 - 当 GOFF1 有效沿时，停止 MPWM，6 路 MPWM 输出高阻。
0 - GOFF1 值不影响 MPWM。

3 GOFF0_EN GOFF0 引脚保护使能。
1 - 当 GOFF0 有效沿时，停止 MPWM，6 路 MPWM 输出高阻。
0 - GOFF0 值不影响 MPWM。

2 CMP_EN 比较器输出保护使能。
1 - 比较器有上升时，停止 MPWM，6 路 MPWM 输出高阻。

0 - MPWM 的输出不受比较器输出影响。

1 GOFF1_TRIG_CTRL GOFF1 触发有效电平控制。
0-上升沿有效。
1-下降沿有效。

0 GOFF0_TRIG_CTRL GOFF0 触发有效电平控制。
0 - 上升沿有效。
1 - 下降沿有效。

PROT_CTRL2 MPWM 保护控制寄存器 2: ADH 初始值: 00H

7	6	5	4	3	2	1	0
SFTW_T RIG_OUT TPUT	SFTW_T RIG_HIGH HZ	-	HIGHZ_I NT_FLG	HIGHZ_ FLG	GOFF1_ VAL	GOFF0_ VAL	COMP_V AL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 SFTW_TRIG_OUTPUT 0->1, 触发 MPWM 正常输出。硬件清 0, 读这位总是 0

6 SFTW_TRIG_HIGHZ 0->1, 触发 GAT、GAB、GBT、GBB、GCT、GCB 输出高阻。
硬件清 0, 读总是 0。

4 HIGHZ_INT_FLG 由 GOFF、比较器或者软件引起的 GAT、GAB、GBT、GBB、
GCT、GCB 输出高阻保护中断指示。
硬件置位, 软件清 0。

3 HIGHZ_FLG 当前 GAT、GAB、GBT、GBB、GCT、GCB 高阻指示。
1 - 高阻。
0 - 正常输出。

2 GOFF1_VAL GOFF1 值指示。

1 GOFF0_VAL GOFF0 值指示。

0 COMP_VAL 比较器输出指示。

PROT_CTRL3 MPWM 保护控制寄存器 3: AEH 初始值: 00H

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

7:5 PROT_SIG_FIL 保护信号 (goff0, goff1) 滤波延时控制
0~7 - 滤波延迟为 0~7 主时钟周期;

4:2 COMPO_SIG_FIL 比较器输出滤波设置;
0~7 - 滤波延迟为 0~7 主时钟周期;

1:0 DT_BT_CTRL 死区和消隐时间扩展控制

00-
死区时间=TIME_DEADZONE×2
消隐时间 = TIME_BLANK×2;

01-
死区时间=TIME_DEADZONE×4
消隐时间 = TIME_BLANK×4;

10-
死区时间=TIME_DEADZONE×8
消隐时间 = TIME_BLANK×8;

11-
死区时间=TIME_DEADZONE×16
消隐时间 = TIME_BLANK×16;

TIME_DEADZONE 死区时间寄存器: **B7H** 初始值: **00H**

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

死区时间 =TIME_DEADZONE*DT_BT_CTRL/60M;

死区时间设置, 范围 0.03μs ~ 68μs。上升沿插入死区, 下降沿不用。

0 - 无死区。

其他值 - 死区时间和 DT_BT_CTRL 配置有关。

TIME_BLANK 消隐时间寄存器: **AFH** 初始值: **00H**

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

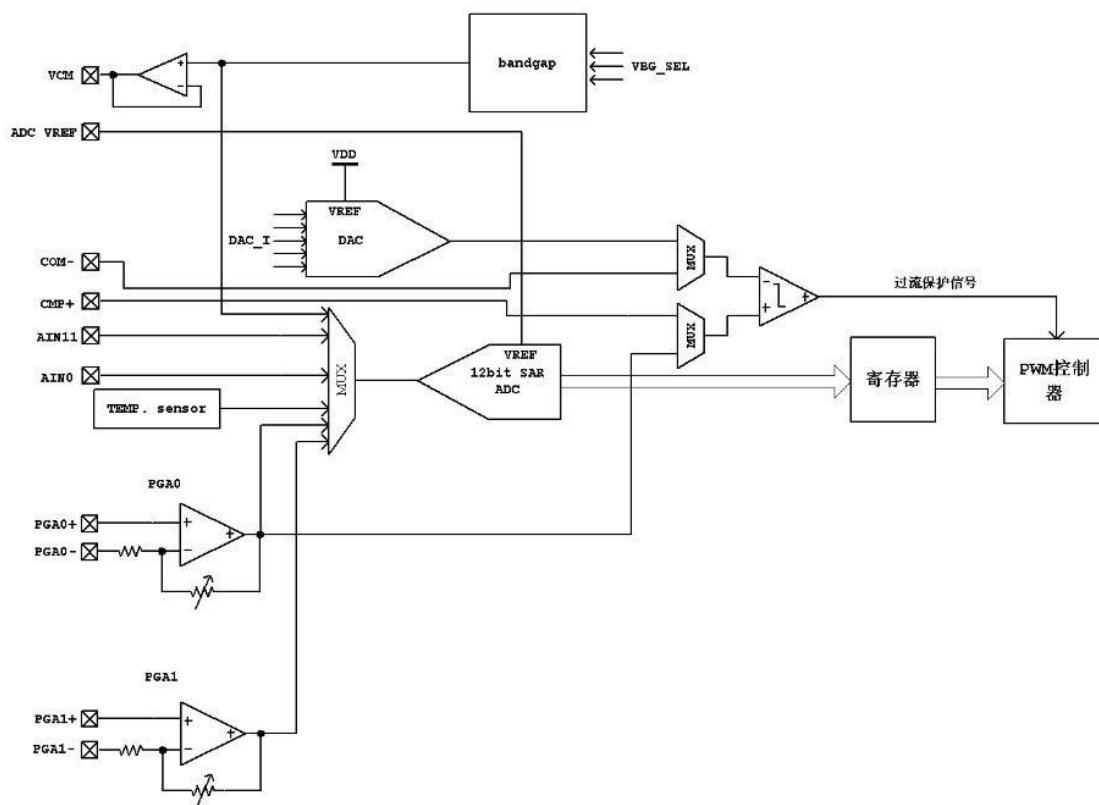
消隐延时控制, 范围 0~5us (保护信号有效延时), 消隐计数器时钟分频控制由寄存器 DT_BT_CTRL 控制。

0 - 无消隐时间。

其他值 - 消隐延时和 DT_BT_CTRL 配置有关。

11. 线性模拟电路

11.1. 结构图



AEN 模拟模块控制寄存器: C9H

初始值 - 10H

7	6	5	4	3	2	1	0
-	DAC_EN	VCMEN	-	-	PGA1_EN	PGA0_EN	CMP_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

6 DAC 使能
1-使能
0-禁止

5 VCMEN 输出使能，控制 VCM 在 P16 输出。
1-使能
0-禁止

2 可编程增益放大器 PGA1 使能
1-使能
0-禁止

1 可编程增益放大器 PGA0 使能
1-使能
0-禁止

0 比较器使能
1-使能
0-禁止

11.2. 高速比较器

特性

1. 输入负端可接外部输入电压、内部 DAC 输出作为基准电压）。
2. 输入正端可接外部输入电压、可编程增益放大器输出端。
3. 比较器输出端信号产生可屏蔽中断。
4. 比较器输出信号可停止 MPWM 工作同时使 MPWM 输出端口为高阻状态。
5. 看比较器输出上升沿可触发 ADC。

AEN.CMP_EN(C9H.0) 使能比较器。

CMP_CTRL.CMPP_SEL(E7H.7) 选择比较器正输入端连接到外部引脚 P00, CMP_CTRL.CMPN_SEL(E7H.6) 选择比较器负输入端连接到外部引脚 P01。

POFUN0.2(2040H.2) 选择比较器模拟输出到 P02, POFUN0.1(2040H.1) 选择比较器输入负端到 P01, POFUN0.0(2040H.0) 选择比较器输入正端到 P00。

CMP_CTRL 寄存器选择比较器正输入端连接到 PGA0 或 P00, 负输入连接到 DAC 或 P01。

CMP_CTRL 模拟控制寄存器: E7H

初始值: 00H

7	6	5	4	3	2	1	0
CMPP_SEL	CMPN_SEL		CMPFIL_CTRL				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7 CMPP_SEL 比较器正输入端连接控制
0-PGA0 输出
1-外部引脚 P00

6 CMPN_SEL 比较器负输入端连接控制
0-DAC 输出
1-外部引脚 P01

4:0 CMPFIL_CTRL 比较器输出滤波控制

1~20 滤波时间: 100ns, 200ns...2000ns。
其他值无滤波。

11.3. 可编程增益放大 PGA

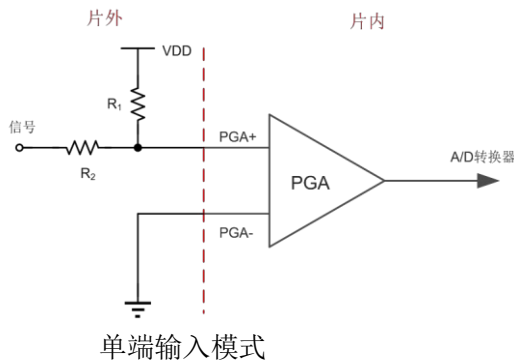
特性

1. 单端输出可连接到外部均值电路、ADC 输入或内部高速比较器的正输入端。
2. 增益: X1,X2,X4,X6,X8,X10,X12,X14,X16

11.3.1. 工作模式

通过外部配置, 可编程增益放大器兼容单端输入模式和差分输入模式两种。

11.3.1.1. 单端输入



可编程增益放大器单端输入模式如图 1 所示。配置方式: PGA 反向输入端接地, 同向输入端通过电阻连接到采样端和电源。通过改变 R_1 和 R_2 阻值, 可以调节 PGA 直流偏置及整个信号链的增益 (1)。

$$gain_{total} = \frac{R_1}{R_1 + R_2} \cdot gain_{PGA} \quad (1)$$

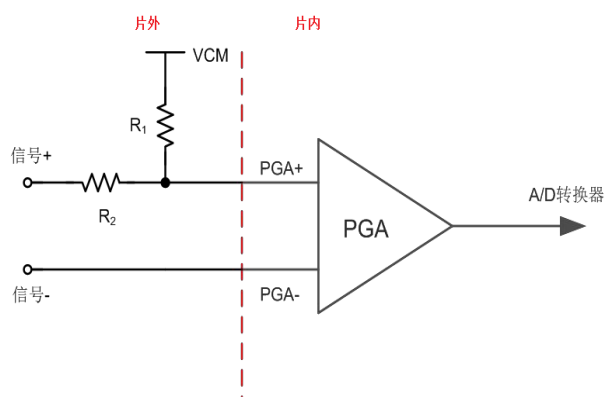
单端输入模式 PGA 增益设置如表 1。

1000	X1
0000	X2
0001	X4

0010	X 6
0011	X 8
0100	X 10
0101	X 12
0110	X 14
0111	X 16

单端输入模式 PGA 增益设置

11.3.1.2. 全差分输入



差分输入配置方式

可编程增益放大器差分输入模式如图 2 所示，其可以抑制采样信号的共模干扰。配置方式：PGA 同向输入端通过电阻可设置为差分输入正端，反向输入端可直接设置为差分输入负端，通过改变 VCM 来调节 PGA 直流偏置。在差分模式下，PGA 增益设置如表 2，其中 R_1 和 R_2 需满足公式（1）。

$$\frac{R_1}{R_2} = gain_{PGA} \quad (2)$$

差分输入模式 PGA 增益设置如表 2。

0000	X1
0001	X3
0010	X5
0011	X 7
0100	X 9
0101	X 11
0110	X 13
0111	X 15

差分输入模式 PGA 增益设置

11.3.1.3. PGA 应用方案

方案一

- 外部电源电压提供 ADC reference;
- 片内集成两路 PGA (Programmable Gain Amplifier)，其工作在单端输入模式，片内集成 PGA 不仅提高了集成度，还大大降低了干扰，提高 I/O 的利用效率；
- 过流保护信号链由片外滤波器，片内 PGA0，DAC 和高速比较器构成，其中 PGA0 增益和 DAC 输出电压可数字编程；（在过流保护信号链中，PGA0 增益推荐使用单位增益($\times 1$)，PGA0 和高速比较器延时之和的典型值为 200nS）。PGA0 的 $SR \geq 15V/\mu s$ ，过流信号经过 PGA0 能跟踪信号的变化，还可滤除信号中叠加的毛刺。
- 平均电流检测信号链由片外滤波器，片内 PGA1 和 ADC 构成，其中 PGA1 的增益可数字编程 ($\times 1$ 、 $\times 2$ 、 $\times 4$ 、 $\times 6$ 、 $\times 8$ 、 $\times 10$ 、 $\times 12$ 、 $\times 14$ 、 $\times 16$)。

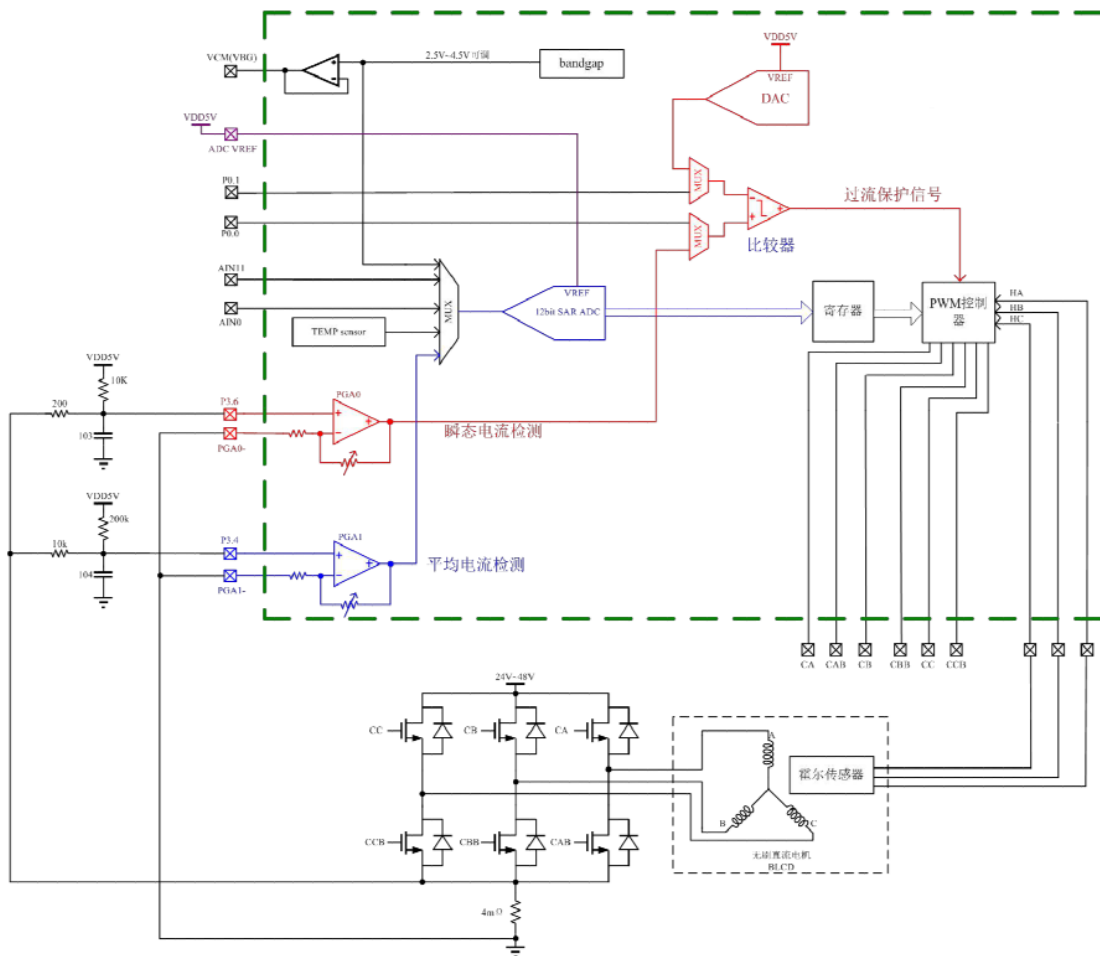


图 1、第一种应用方案工作原理图

方案二

相比于第一种应用方案，第二种应该方案在 PGA 和 ADC 之间插入一低通滤波器，进行滤波。

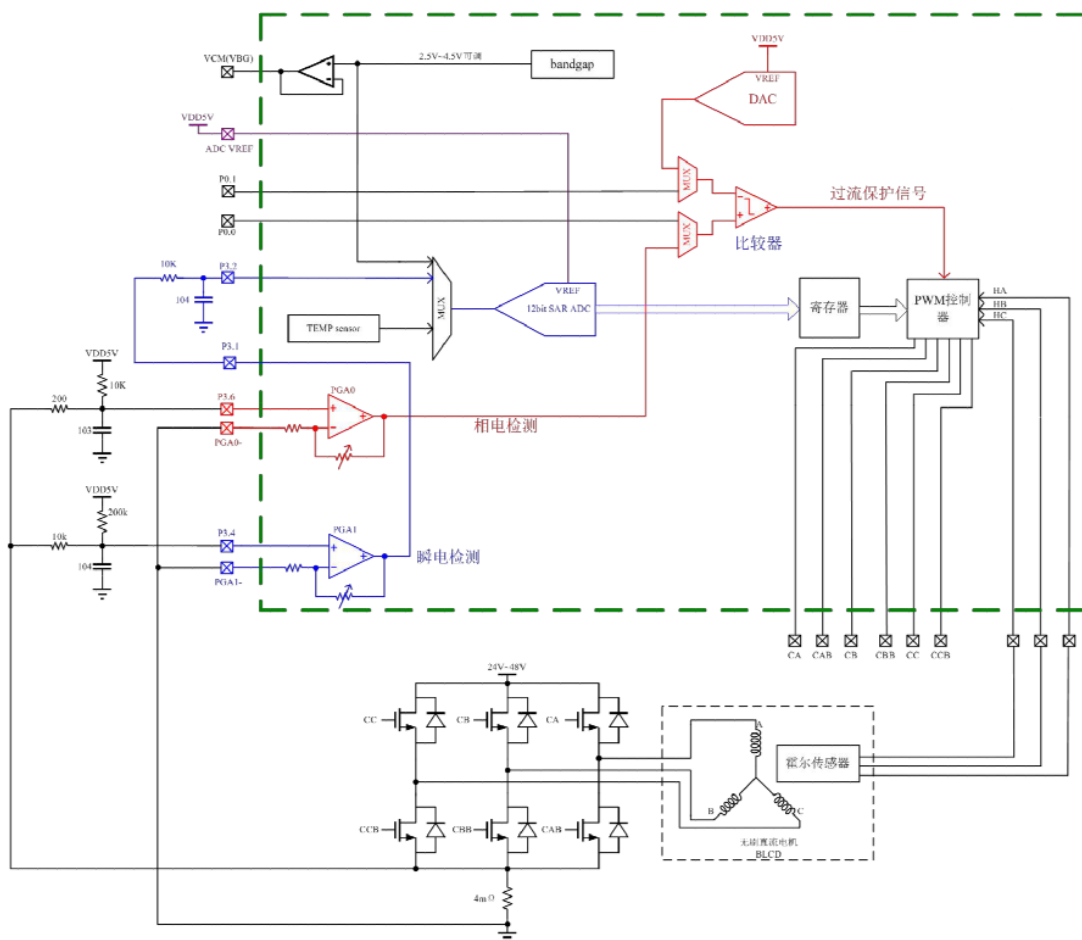


图 2、第二种应用方案工作原理图

此外，通过外部配置，可编程增益放大器兼容单端输入模式和差分输入模式两种。

● 单端输入

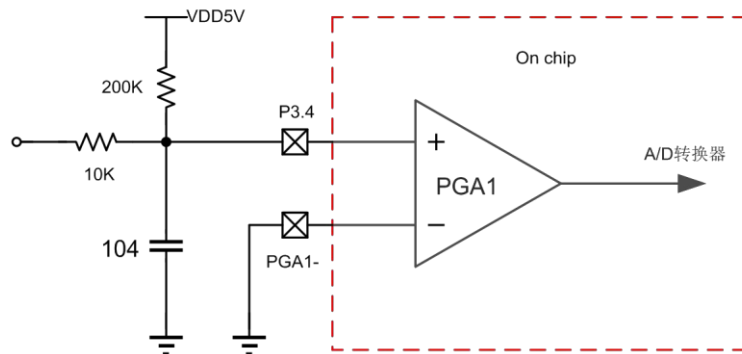


图 3、单端输入模式

可编程增益放大器单端输入模式如图 3 所示。配置方式：PGA 反向输入端接地，同向输入端通过电阻连接到采样端和电源。通过改变两个电阻阻值，可以调节 PGA 直流偏置及整个信号链的增益（1）。

$$gain_{total} = \frac{200K}{200K + 10K} \bullet gain_{PGA} \quad (1)$$

单端输入模式 PGA 增益设置如表 1。

1000	X1
0000	X2
0001	X4
0010	X6
0011	X 8
0100	X 10
0101	X 12
0110	X 14
0111	X 16

单端输入模式 PGA 增益设置

● 全差分输入

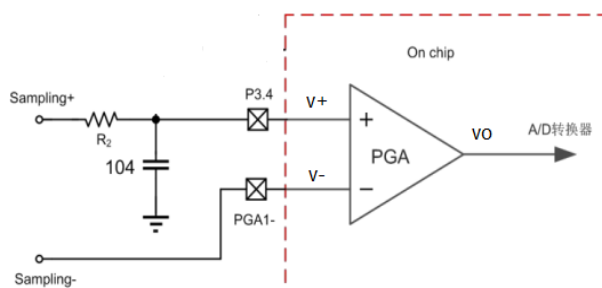


图 4、差分输入配置方式

可编程增益放大器差分输入模式如图 4 所示，在差分模 式下，PGA 增益设置如表 2 其中 $gain_{PGA}$ 需满足公式

$$VO = (V+ - V-) \bullet gain_{PGA} + V-$$

$$gain_{PGA} \bullet V+ < V_{CC}$$

差分输入模式 PGA 增益设置如表 2。

0000	X1
0001	X3
0010	X5

0011	X 7
0100	X 9
0101	X 11
0110	X 13
0111	X 15

差分输入模式 PGA 增益设置

AEN.PGA1_PD 使能 PGA1，AEN.PGA0_PD 使能 PGA0，PGA 增益由 PGA_GAIN 寄存器控制。

PGA_GAIN 可编程增益放大器增益设置寄存器：**FFH** 初始值：**00H**

7	6	5	4	3	2	1	0
PGA1_GAIN				PGA0_GAIN			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:4 PGA1 增益设置

1xxx-×1
0000-×2
0001-×4
0010-×6
0011-×8
0100-×10
0101-×12
0110-×14
0111-×16

3:0 PGA0 增益设置

1xxx-×1
0000-×2
0001-×4
0010-×6
0011-×8
0100-×10
0101-×12
0110-×14
0111-×16

ADC_SLCTH.PGA0 选择 PGA0 输出到 AD 输入，ADC_SLCTH.PGA1 选择 PGA1 输出到 AD 输入。P3FUN0 和 P3FUNC1 选择 PGA0 和 PGA1 输入输出引到 I/O 端口。

11.4. 数模转换器 DAC

特性

工作电压范围：2.7V ~ 5.5V

5-bit 分辨率

可选基准源

输出可接到高速比较器的负输入端或 ADC 基准

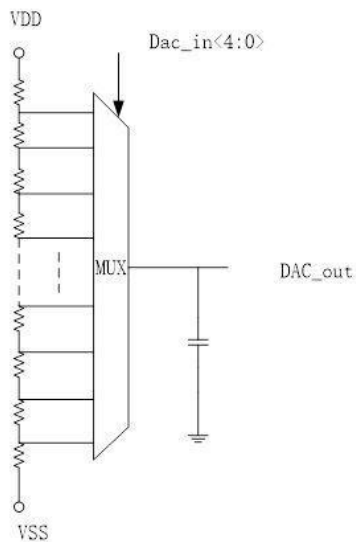
DAC_I DAC 输出电压设置寄存器：91H 初始值：20H

7	6	5	4	3	2	1	0
—	—	—	DAC_I				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

4:0

DAC 数字输入值设置。输出电压 = $\frac{V_{DD} \cdot (DAC_I + 1)}{32}$ 。

P0FUN1.0:P0FUN1.0 选择 P02 DAC 输出。AEN.6 使能 DAC。



12. 模数 (AD) 转换器

1. 外部 12 输入通道
2. 片内温度传感器输入
3. 模拟运放输出可作为 AD 输入
4. 片内基准作为 AD 输入
5. AD 基准: VDD 和片外 ADC_VREF
6. 最快转换速度 1 μ s
7. 4 个数据缓存, 可启动自动扫描把结果依次存入缓存, 自动扫描通道可配置。
8. 对两个 PGA 输出同时进行 AD 转换。
9. AD 启动方式: 软件触发, PWM0 占空比中断触发, PWM0 顶部中断触发, PWM0 底部中断触发, MPWM 周期计数器值和 PWM_CMx 值匹配时中断触发 ADC, MPWM 占空比中断触发, MPWM 顶部中断触发, MPWM 底部中断触发, 比较器输出上升沿触发, P07 触发上升下降沿触发 ADC。
10. 输出结果左、右对齐功能。

12.1. 自动扫描

ADC 的通道可配置为自动扫描序列, 并设置自动扫描模式。当产生 PWM 中断启动 ADC、软件启动 ADC 或其他源启动时, 跟据设置的扫描序列自动扫描, 每完成一次序列扫描, 设置完成标志, 停止扫描, 等待下一次 ADC 启动。从 ADC 得到的数据依据轮询通道个数和顺序放入 ADC0_DATA、ADC1_DATA、ADC2_DATA、ADC3_DATA 中。

12.2. 双通道采样

在模式选择寄存器中置 MODE (ADC_MOD.2) 位为 1, 并选择 DC_SET (ADC_MOD.1:0) 的数据。此时不考虑通道选择寄存器中的数据。依据双通道模式工作, 也可通过配置寄存器选择连续采样、数据存放模式和轮询停止。从 ADC 中得到的数据依据 DC_SET, 放入 ADC0_DATA、ADC1_DATA 寄存器中。

单通道采样模式时, 从 ADC 中得到的数据放入 ADC0_DATA 寄存器中。

12.3. ADC 控制

ADC_MOD.ADC_EN ADC 使能, 打开 ADC 电源, 初始化 ADC。

ADC_CFG.S_C 控制 ADC 转换模式。为 0 选择单次模式, 每个 AD 启动触发, 只执行一次 AD 转换, AD 转换完成设置完成标志 (ADC_CFG.1); 为 1 连续模式, 触发启动 ADC 后, ADC 连续转换选择的输入通道存入 ADC 结果寄存器, 连续转换可以软件设置 ADC_CFG.STOP 停止。触发启动后, AD 转换根据设置的采样模式处理转换结果。

ADC 采样模式：单通道、双通道。双通道有四种组合，分别是 CH0 和 CH1、CH0 和 PGA1OUT、CH1 和 PGA0OUT、PGA0OUT 和 PGA1OUT。

ADC 转换启动可以由软件、PWM0 占空比中断触发，PWM0 顶部中断触发，PWM0 底部中断触发，MPWM 周期计数器值和 PWM_CMx 值匹配时中断触发 ADC，MPWM 占空比中断触发，MPWM 顶部中断触发，MPWM 底部中断触发，比较器输出上升沿触发，P07 触发上升下降沿触发 ADC。

还可进行 4 输入通道自动扫描转换，有四组寄存器存放扫描转换的结果。

ADC 输入通道选择：

0 - 11 外部引脚输入

PGAOUT 片内温

度传感器 片内

基准电压源

ADC 结果可以选择左右对齐格式。

ADC 使用中注意要设置 ANA_CFG 模拟控制寄存器 (DAH) 中 ADC_VCM_CTRL (7:5) 位控制 ADC VCM，用户使用 ADC 时，把它设置到接近 $\frac{1}{2}AVDD$ 。

12.4. ADC 寄存器

ADC_MOD ADC 模式寄存器：D9H 初始值：00H

7	6	5	4	3	2	1	0
-	-	-	-	ADCEN	MODE	DC_SET	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

3 ADC_EN ADC 使能控制
0-关闭 ADC
1-使能 ADC

2 MODE ADC 工作模式
0-单通道模式
1-双通道模式

1:0 DC_SET 当 MODE=1 时，双通道的选择：采样顺序按以下排列。
00-PGA0 和 PGA1
01-PGA0 和 CH0
10-PGA1 和 CH0
11-CH0 和 CH1

ADC_CFG ADC 配置寄存器：DAH 初始值：00H

7	6	5	4	3	2	1	0
ADC_VCM_CTRL			STOP	P_MODE	LEFT	S_C	IADC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:5 ADC_VCM_CTRL ADC 内部共模电压调节位，控制 VCM 电平。（根据外部参加电压的不同而进行调节）
 芯片 5V 供电电压变化时，可调整 $V_{CM} = V_{DD}/2$ ；
 不参与 AD 转换计算，为了提高 AD 精度。

000 - 1.4V
 001 - 1.6V
 010 - 1.8V
 011 - 2.0V
 100 - 2.2V
 101 - 2.4V
 110 - 2.6V
 111 - 2.8V

4 STOP 1-停止连续转换
 读时，一直为 0

3 P_MODE 外部引脚 P07 触发边沿选择
 0-上升沿触发
 1-下降沿触发

2 LEFT ADC 数据左右对齐控制
 0-右对齐
 1-左对齐

1 S_C 0-每个 AD 转换启动触发，只进行一次转换
 1-AD 转换启动触发后，自动连续进行转换，STOP 可以停止

0 IADC AD 转换结束中断标志，硬件置位，软件清 0。

ADC_TRIGEN ADC 触发模式寄存器：DBH 初始值：00H

7	6	5	4	3	2	1	0
-	-	SOFT_T RIG	MPWM_T GEN	PWM0_T GEN	TIMER1 _TGEN	CMP_T GEN	PIN_TGE N
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

5 SOFT_TRIG 软件触发 ADC 启动，1-启动 ADC 转换，硬件自动清零

4 MPWM_TGEN MPWM 触发使能
 1-使能
 0-禁止
 MPWM 底部中断触发，MPWM 顶部中断触发，MPWM 占空比匹配中断触发，MPWM_CM1、MPWM_CM2 匹配中断触发。

3 PWM0_TGEN PWM0 触发使能

1-使能

0-禁止

在中心对齐模式，有 PWM0 底部中断触发，PWM0 顶部中断触发，PWM0 占空比匹配中断触发。

在边沿对齐模式，有 PWM0 占空比中断触发，PWM0 周期中断触发。

1 CMP_TGEN

比较器触发使能

1-使能

0-禁止

比较器输出上升沿触发 AD。

0 PIN_TGEN

P07 触发使能

1-使能

0-禁止

ADC_CFG.3 可选择上升或下降沿触发 AD。

ADC_SLCTH ADC 通道选择寄存器 1: D2H

初始值: 00H

7	6	5	4	3	2	1	0
PGA0	PGA1	CH0	CH1	CH2	CH3	CH4	CH5
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:0

1 - 使能

0 - 禁止

PGA0_SEL 优先级最高

SLCT_H 和 SLCT_L 为 ADC 通道选择：最多可同时选择 4 个通道，优先级次序是 SLCT_H.7、SLCT_H.6...SLCT_L.1、SLCT_L.0，优先级最高 SLCT_H.7 (PGA0) 最高，SLCT_L.0 (BGAP) 最低。

ADC_SLCTL ADC 通道选择寄存器 2: D3H

初始值: 00H

7	6	5	4	3	2	1	0
CH6	CH7	CH8	CH9	CH10	CH11	TEMP. SENS	BGAP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:0

1 - 使能

0 - 禁止

BGAP 优先级最低

注：ADC 处于扫描模式时，只能最多选择四个通道，扫描优先级最高 ADC_SLCTH.7，依次 ADC_SLCTH.6……ADC_SLCTH.0、ADC_SLCTL.7……ADC_SLCTL.0。双通道扫描时，只能选择 2 通道或 4 通道。

ADC_DATA0H ADC 数据寄存器 0 高 8 位: DFH 初始值: 00H

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

ADC_DATA0L ADC 数据寄存器 0 低 8 位: DEH 初始值: 00H

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

ADC_DATA1H ADC 数据寄存器 1 高 8 位: DDH 初始值: 00H

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

ADC_DATA1L ADC 数据寄存器 1 低 8 位: DCH 初始值: 00H

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

ADC_DATA2H ADC 数据寄存器 2 高 8 位: D7H 初始值: 00H

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

ADC_DATA2L ADC 数据寄存器 2 低 8 位: D6H 初始值: 00H

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

ADC_DATA3H ADC 数据寄存器 3 高 8 位: D5H 初始值: 00H

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

ADC_DATA3L ADC 数据寄存器 3 低 8 位: D4H 初始值: 00H

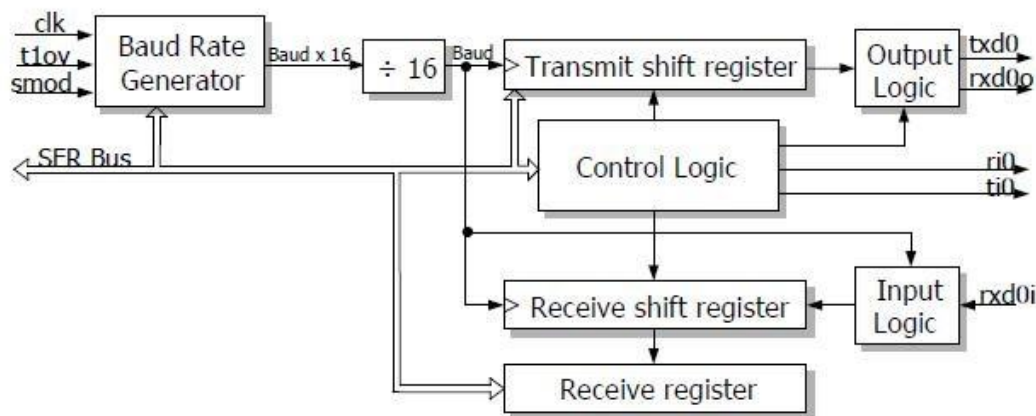
7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13. 异步串行通讯口 (UART0、UART1)

13.1. UART0

UART0 可编程的双工串行通信接口。它可作为通用异步接收和发送器, 也可作为同步移位寄存。其帧格式可有 8 位、10 位和 11 位, 并能设置各种波特率, 给使用者带来很大的灵活性。

13.1.1. UART0 结构图

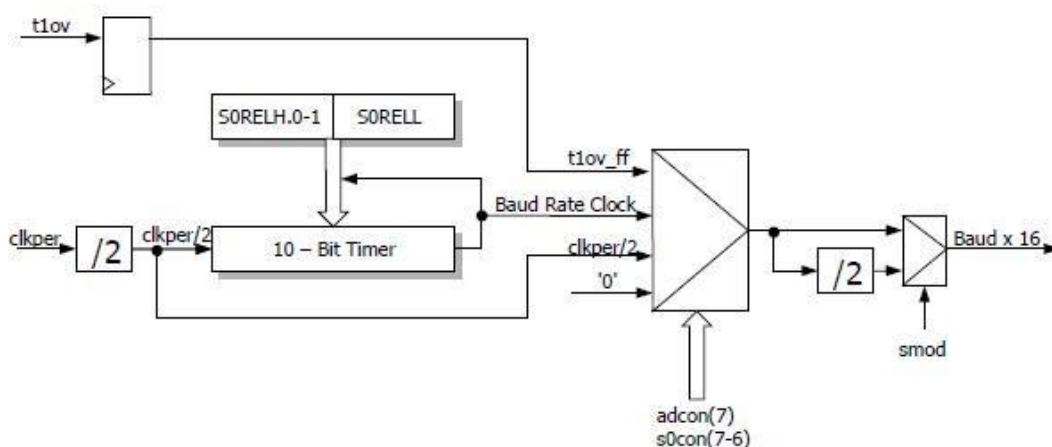


13.1.2. 功能

通过引脚 RXD0 和引脚 TXD0 与外界进行通信。串行口内有一个可直接寻址的专用寄存器 S0BUF, 在物理上 S0BUF 由二个 8 位寄存器组成一个是发送寄存器一个是接收寄存器。两者共用一个地址, 可同时发送、接收数据。CPU 写 S0BUF, 就是修改发送寄存器; 读 S0BUF, 就是读接收寄存器。接收寄存器是双级缓冲的, 以避免在接收下一帧数据之前, CPU 未能及时响应接收寄存器的中断。没有把上一帧数据读走, 而产生两帧数据重叠的问题。对于发送寄存器, 为了保持最大的传输速率, 一般不需要双缓冲, 因为发送时 CPU 是主动的, 不会产生重叠的问题。

根据实际需要, 串行口可设置四种工作方式。

13.1.2.1. 波特率发生器

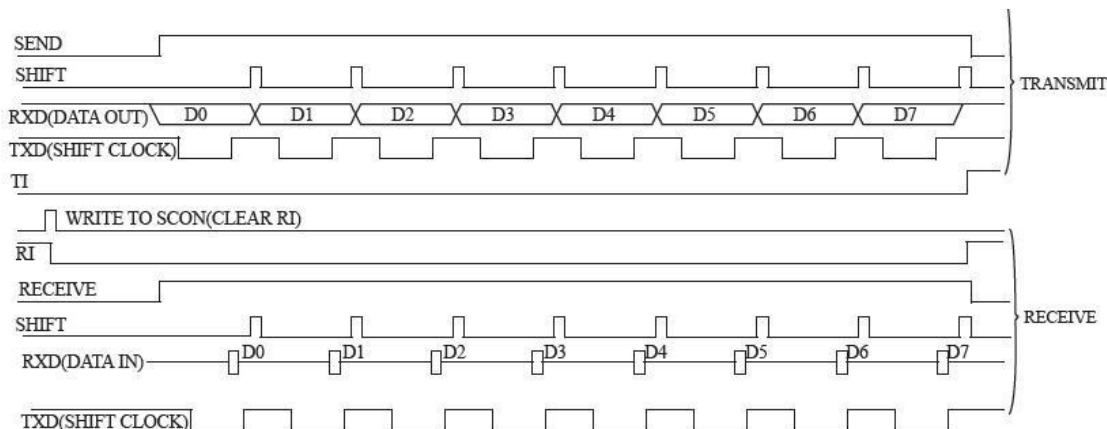


模式 0 和模式 2 的波特率是固定的，而模式 1 和模式 3 的波特率是可变的。由定时器 T1 的溢出率来决定。四种工作模式对应着三种波特率。由于输入的移位时钟的来源不同，所以，各种模式的波特率计算公式也不同。

13.1.2.2. 工作模式

13.1.2.2.1. 模式 0

模式 0 为同步移位寄存器发送 / 接收方式，常用于扩展 I/O 口。串行数据通过 RXD0 输入或输出，而 TXD 用于输出移位时钟，作为外接部件的同步信号。在这种方式下，收 / 发的数据为 8 位，低位在前无起始位、奇偶位及停止位。波特率固定为系统频率的 1/12。

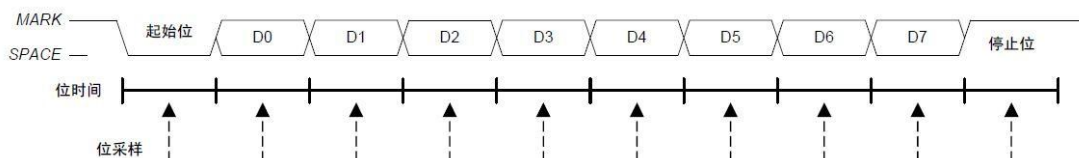


模式 0 时序

13.1.2.2.2. 模式 1

模式 1 真正用于串行发送或接收。为 10 位通用异步接口。TXD0 与 RXD0 分别用于发送与接收数据，收发一帧数据的格式为 1 位起始位、8 位数据位（低位在前）、1 位停止位，共 10 位。在接收时，停止位进入 SOCON 的 RB8，此方式的传送波特率可调。

模式 1 发送时，数据从引脚 TXD0 端输出，当执行数据写入发送级冲器 S0BUF 的命令时，就启动了发送器开始发送。发送时的定时信号，也就是发送移位时钟，是由定时器 T1 送来的溢出信号经过 16 分频或 32 分频（取决 PCON 寄存器中的 SMOD 值）而取得的，时钟就是内部产生的发送波特率。可见方式 1 波特率是可变的。



模式 1 接收时，数据从引脚 RXD0 端输入。接收是在 SOCON 寄存器中 REN0 位=1 的前提下，并检测到起始位（RXD0 上检测到“1”→“0”的跳变，即起始位）而开始的。接收时，定时信号有两种：一种是接收移位时钟（RX 时钟），它的频率和传送波特率相同，也是由定时器 T1 溢出信号经过 16 或 32 分频而得到的；另一种是位检测器采样脉冲，它的频率是 RX 时钟的 16 倍，即在一位数据的期间有 16 位检测器采样脉冲，为完成检测，以 16 倍于波特率的速率对 RXD0 进行采样。为了接收准确无误在正式接收数据之前，还必须判定这个“1”→“0”跳变是否由干扰引起。为此，在这位中间（即 1 位时间分成 16 等份，在第 7, 8 及 9 等份）连续对 RXD0 采样三次，取其中两次相同的值进行判断所检测的值。这样能较好地消除干扰的影响。当确认是真正的起始位“0”后，就开始接收一帧数据。当一帧数据接收完毕后，必须同时满足以下两个条件，这次接收才真正有效。

(1) RI0=0，即上一帧数据接收完成时，RI=1 发出的中断请求已被响应，S0BUF 中的数据已被取走。由软件使 RI0=0，以便提供“接收 S0BUF 已空”的信息。

(2) SM20=0 或收到的停止位为 1（方式 1 时停止位进入 RB80），则将接收到的数据装入串行口的 S0BUF 和 RB80（方式 1 RB80 装入停止位），并置位 RI0；如果不满足，接收到的数据不能装入 S0BUF，这意味着该帧信息将会丢失。

值得注意的是：在整个接收过程中，保证 REN0=1 是一个先决条件，只有当 REN0=1，才能对 RXD0 进行检测。

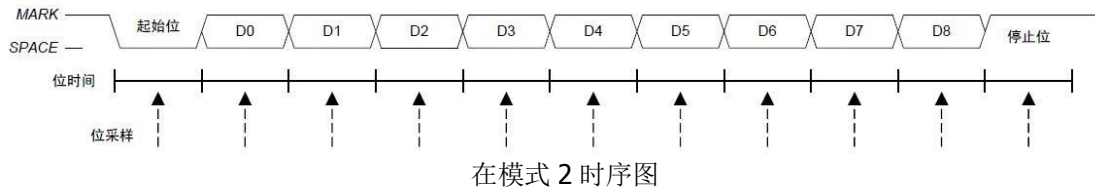
13.1.2.2.3. 模式 2

串行口工作在模式 2 为每帧 11 位异步通信格式，由 TXD0 和 RXD0 发送与接收。每帧 11 位，1 位起始位，8 位数据位（低位在前），1 位可编程的第 9 数据位和 1 位停止位。发送时，第 9 数据位（TB80）可以设置为 1 或 0，也可将奇偶位装入 TB80，从而进行奇偶校验；接收时，第 9 数据位进入 SOCON 的 RB80。

发送前，先根据通信协议由软件设置 TB8（如作奇偶校验位或地址 / 数据标志位），然后将要发送的数据写入 S0BUF 即能启动发送过程，串行口能自动把 TB80 取出，并装入到第 9 位数据的位置，再逐一发送出去。发送完毕，使 TI0=1。

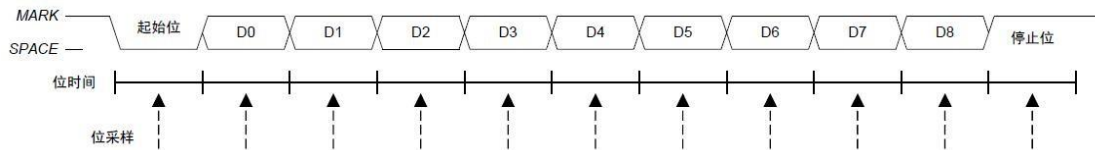
接收时，使 S0CON 中的 REN0=1 允许接收，当检测到 RXD0 端有“1”到“0”的跳变（起始位）时，开始接收 9 位数据，送入移位寄存器(9 位)。当满足 RI0=0 且 SM20=0 或接收到的第 9 位为 1 时，前 8 位数据送入 S0BUF，附加的第 9 位数据送入 S0CON 中的 RB80，置 RI0 为 1；否则，这次接收无效，也不置位 RI。

模式 2 波特率固定到系统频率的 32 分频或系统频率的 64 分频，由 PCON 寄存器中的 SMOD 设置。



13.1.2.2.4. 模式 3

模式 3 与模式 2 数据收发操作是完全一样的，所不同的只是波特率。在模式 3，波特率由内部波特率发生器或定时器 1 溢出速率决定。



在模式 3，UART0 数据位是 9 位，并且波特率可编程。根据 ADCON 寄存器中 BD 位 (ADCON. 7)，使用定时器 1 溢出或者 SORELH、SORELL 寄存器设定的波特率。此外波特率可以使用 PCON 寄存器中的 SMOD 位提高一倍。

通过写数据到 S0BUF 寄存器启动发送过程，在 TXD0 引脚输出数据。发送的第一位是起始位 0，然后处理 9 位数据，其中第 9 位数据取自 S0CON 寄存器中的 TB80 位。最后，一个停止位 1。

RXD0 引脚输入的数据。UART0 引脚 RXD0 下降沿启动数据检测。接收 8 位数据存于 S0BUF 寄存器中，第 9 位存于 S0CON 寄存器中的 RB80 位。期间 S0BUF 和 RB80 保持不变，直到一帧数据接收完成。

13.1.2.3. UART0 多机通讯

模式 2 和模式 3 的 9 位数据收发功能可用于多机通讯。当 S0CON 寄存器中 SM20 位设置并且 RB80 为 1，产生接收中断。

要使用多机通信功能，从机处理器 SM20 位设置为 1，主机处理器发送 9 位从机地址(其中第 9 位设置为 1)，可引起所有从机产生接收中断。从机处理器用接收到的 8 位数据和它们的网络地址进行比较。如果有一个匹配，被寻址的从机将清除它的 SM20 标志准备接收数据信息(RB80=0)。

13.1.3. UART0 寄存器

ADCON UART0 波特率调试配置寄存器（内部调试用）：D8H 初始值：00H

7	6	5	4	3	2	1	0
BD	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R

7 BD UART0 波特率选择 (在模式 1 和模式 3)
 1 - 使用内部波特率发生器。
 0 - 使用定时器 1 溢出信号。

S0CON UART0 控制寄存器：98H 初始值：00H

7	6	5	4	3	2	1	0
SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7:6 SM0:SM1 UART0 模式选择

SM0	SM1	模式	说明	波特率	
0	0	0	移位寄存器模式	Fclk/12	
0	1	1	8 位 UART	可变（详细在下说明）	
1	0	2	9 位 UART	与 PCON 中 SMOD 位有关	
				SMOD	波特率
				0	Fclk/64
				1	Fclk/32
1	1	3	9 位 UART	可变（详细在下说明）	

模式 1 和模式 3 的波特率：

BD (ADCON.7) = 0:

$$\text{波特率} = \frac{2^{\text{SMOD}} \times \text{Fclk}}{32} \times (\text{定时器 1 溢出速率})$$

BD (ADCON.7) = 1:

$$\text{波特率} = \frac{2^{\text{SMOD}} \times \text{Fclk}}{64 \times (2^{10} - \text{S0REL})}$$

Fclk — 系统时钟频率

S0REL — S0RE LH:S0RE LL

SMOD — PCON.7

5 SM20 多机通讯使能

主要用于模式 2 和模式 3。

0 — 禁止

1 — 允许多机通信

4	REN0	允许接收控制位 由软件置 1 或清 0，只有当 REN0=1 时，才允许接收，相当于串行接收的开关；若 REN0=0，则禁止接收。 在串行通信接收控制程序中，如果满足 RI0=0，置位 REN=1（允许接收）的条件，就会启动一次接收过程一帧数据就装入 S0BUF 中。
3	TB80	发送数据的第 9 位 在模式 2 或模式 3 中，根据发送数据的需要由软件置位或复位。在许多通信协议中可作奇偶校验位。也可在多机通信中作为发送地址帧或数据帧的标志。对于后者，TB80=1，说明发送该帧数据为地址；TB80=0，说明发送该帧的数据为数据字节。在模式 0 和模式 1 中，该位未用。
2	RB80	接收数据的第 9 位 在模式 2 或模式 3 中，接收到的第 9 位数据放在 RB80 位。它或是约定的奇 / 偶校验位，或是约定的地址 / 数据标志位。 在模式 2 和模式 3 多机通信中，若 SM20=1，. 如果 RB80=1，说明收到的数据为地址帧。 在模式 1 中，若 SM20=0（即不是多机通信情况），RB80 中存放的是已接收到的地址位。 在模式 0 中，该位未用。
1	TI0	发送中断标志 在一帧数据发送完时被置位。在模式 0，串行发送第 8 位结束时，或其它方式串行发送到停止位的开始时由硬件置位，可用软件查询。它同时也申请中断，TI0 置位意味着向 CPU 提供“发送缓冲器 S0BUF 已空”的信息，CPU 可以准备发送下一帧数据。串行口发送中断被响应后，TI0 不会自动清 0，必须由软件清 0。
0	RI0	接收中断标志 在接收到一帧有效数据后由硬件置位。在方式 0 中，第 8 位数据发送结束时，由硬件置位，在其它三种方式中，则在接收到停止位中间时由硬件置位。RI0=1，申请中断。表示一帧数据接收结束，并已装入接收 S0BUF 中，要求 CPU 响应中断，取走数据。RI0 也必须由软件清 0，解除中断申请，并准备接收下一帧数据。 串行发送中断标志 TI0 和接收中断标志 RI0 是同一个中断源，CPU 事先不知道是发送中断 TI0 还是接收中断 RI0 产生的中断请求，所以在全双工通信时，必须由软件来识别。

S0BUF UART0 数据缓冲寄存器：99H

初始值：00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

S0RELH UART0 重载寄存器高 8 位: BAH 初始值: D9H

7	6	5	4	3	2	1	0
D15	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

S0RELL UART0 重载寄存器低 8 位: AAH 初始值: 03H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[PCON 电源控制寄存器: 87H](#)

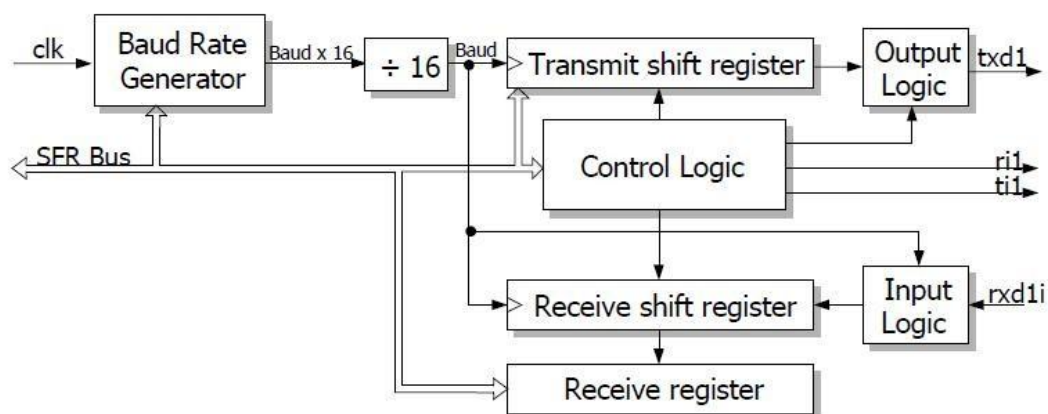
IEN0 中断使能寄存器 0: A8H

– [ES0 \(IEN0.4\) UART0 中断使能](#)

13.2. UART1

UART1 提供一个灵活全双工异步收发器的功能。它可以工作在两种模式。它提供接收缓存，即，第二帧数据接收完成前，不更新先前已接收存于缓存中数据。

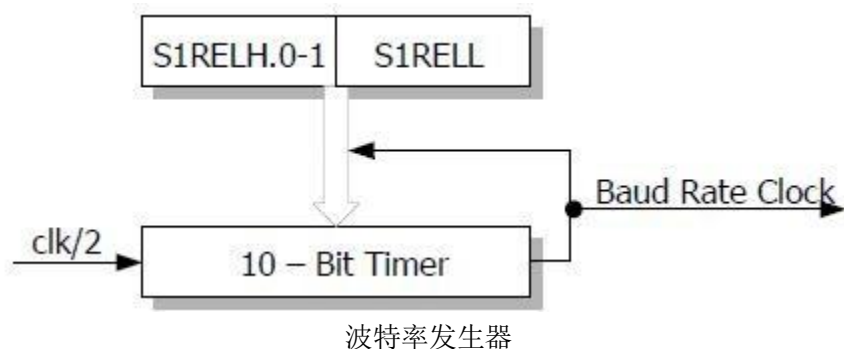
13.2.1. UART1 结构图



结构图

13.2.2. 功能

13.2.2.1. 波特率发生器



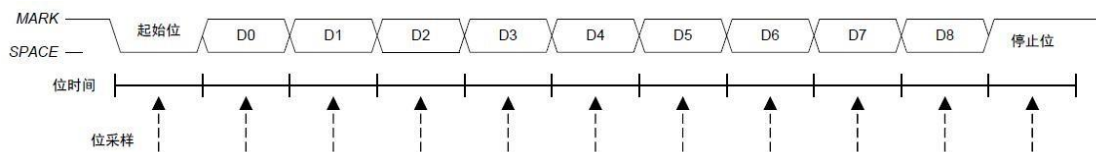
13.2.2.2. 工作模式

13.2.2.2.1. 模式 A

在模式 A, 使用 9 个数据位和可编程的波特率收发数据。S1RELH 和 S1RELL 被用于产生波特率。UART1 的波特率不能使用 SMOD (PCON. 7) 改变速率。

通过写数据到 S1BUF 寄存器启动发送过程, 在 TXD1 引脚输出数据。发送的第一位是起始位 0, 然后处理 9 位数据, 其中第 9 位数据取自 S1CON 寄存器中的 TB81 位。最后一个停止位 1。

RXD1 引脚输入的数据。UART1 引脚 RXD1 下降沿启动数据检测。接收 8 位数据存于 S1BUF 寄存器中, 第 9 位存于 S1CON 寄存器中的 RB81 位。期间 S1BUF 和 RB81 保持不变, 直到一帧数据接收完成。



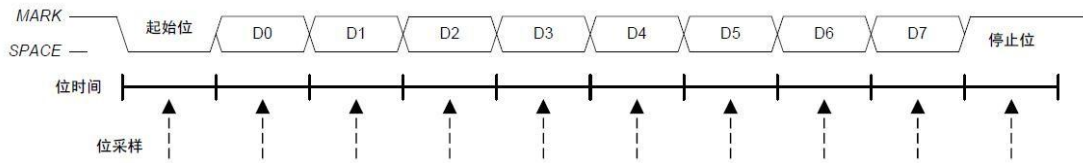
模式 A 时序

13.2.2.2.2. 模式 B

在 B 模式下, UART1 使用 8 个数据位和可编程的波特率收发数据。S1RELH 和 S1RELL 被用于产生波特率。UART1 的波特率不能使用 SMOD (PCON. 7) 改变速率。

通过写数据到 S1BUF 寄存器启动发送过程, 在 TXD1 引脚输出数据。发送的第一位是起始位 0, 然后处理 8 位数据, 最后一个停止位 1。

RXD1 引脚输入的数据。UART1 引脚 RXD1 下降沿启动数据检测。接收 8 位数据存于 S1BUF 寄存器中，停止位存于 S1CON 寄存器中的 RB81 位。期间 S1BUF 和 RB81 保持不变，直到一帧数据接收完成。



模式 B 时序

13.2.2.3. UART1 多机通讯

模式 A 的 9 位数据收发功能可用于多机通讯。当 S1CON 寄存器中 SM21 位设置并且 RB81 为 1，产生接收中断。

要使用多机通信功能，从机处理器 SM21 位设置为 1，主机处理器发送 9 位从机地址(其中第 9 位设置为 1)，可引起所有从机产生接收中断。从机处理器用接收到的 8 位数据和它们的网络地址进行比较。如果有一个匹配，被寻址的从机将清除它的 SM21 标志准备接收数据信息(RB81=0)。

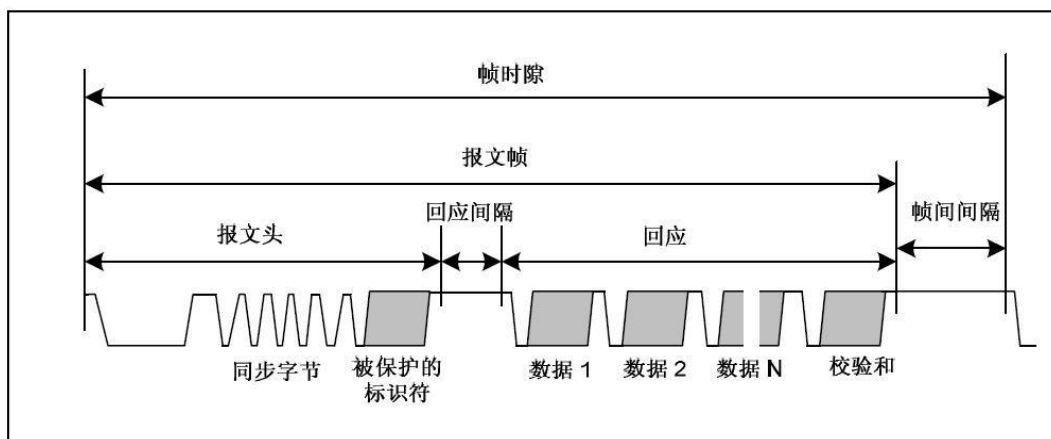
13.2.3. 本地互连网络 (LIN)

UART1 支持本地互连网络 (LIN) 协议的主机和从机操作。

13.2.3.1. LIN 协议

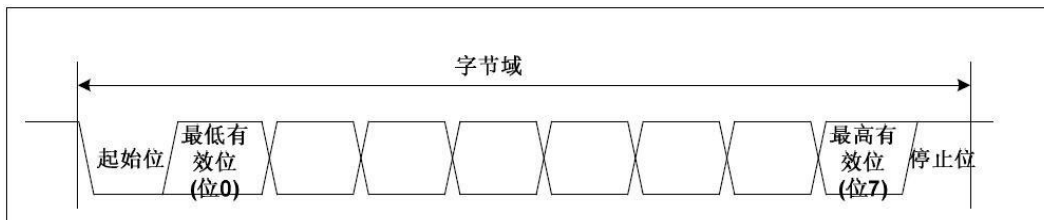
LIN 是一种汽车内部网络互连的整体通信概念。该通信协议基于 SCI (UART) 数据格式、单主控制器 / 多从设备模式、无需固定的时间基准即可实现各节点时钟同步。LIN 具有吸引力的一大特性是：从节点无需通过石英晶振或陶瓷晶振即可实现自同步，大大降低了硬件平台的成本。因此，需要计算并返回每一个报文帧的波特率。

LIN 的报文帧结构如图 10-7 所示。由以下部分组成： 报文头，由分隔域（13 个位时间，低电平）、同步字节 (55H) 和 ID 标识符域 组成
回应时间 数据字节（根据
UART 协议） 校验和



LIN 的帧结构

每个字节域串行发送，如下图所示。先发送 LSB：最后发送 MSB。起始位编码为值“0”，停止位编码为值“1”。



字节域结构

分割域用作新的一帧的起始标志。只有该域不遵循上图的字节域结构。分隔域始终由主机任务（主机模式）产生，它必须至少保持包括起始位在内的 13 位显性值，随后为分隔符，如下图所示。分隔符必须至少保持 1 位标称位时间。

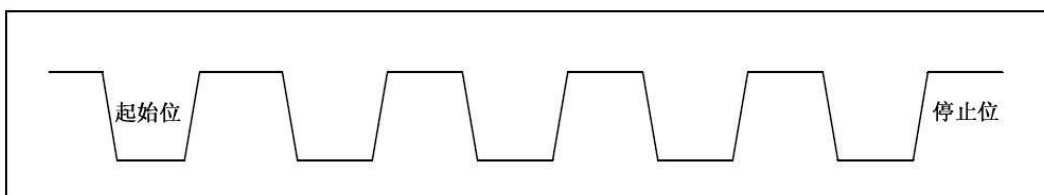
从节点将使用的分隔域检测阈值为 11 个标称位时间。



分割域 同步了节延确定时

间鉴准的特定序列，值为 55H。如下圈所示。

即使从机希望接收到字节域（假定字节域相互分开），它始终要检测分隔切同步序列。如果在检测阶段接收到字节域，分隔戈刃同步序列检测将中断正在进行的数据传送，开始新一帧数据的处理。



同步字节域

当主机发送了正确的 ID 标识符，相应的目标从机将接收和发送数据：

1. 从机等待同步分隔
2. 从机与同步字节同步
3. 从机侦听标识符 (ID)
4. 从机根据 ID 标识符来确定接收或发送数据，或不做任何操作
5. 发送数据时，从机发送 2, 4 或 8 个数据字节，随后是校验字节

13.2.3.2. LIN 报文头发送

LIN 报文头发送只适用于主模式。在 LIN 通信中，主机任务决定在何时、将哪帧传送到总线上；还决定由哪个从机来接收每帧数据。主机和从机之间所需的握手信息由主机的报文头提供。

报文头由分隔序列、同步序列及随后的标识符组成。这三个域中，只有分隔域不能按正常的 8 位 UART 数据格式传送。分隔域必须包含 13 位或更多的显性值以保证从机能正确同步。

在 LIN 通信中，开始传送被保护的标识符时要求从机已被同步。为了实现同步，每一帧以分隔域开始、随后紧跟同步字节。该序列是唯一的，它为任何从机任务提供了足够的信息来检测新帧的开始、并在标识符开始时使从机同步。

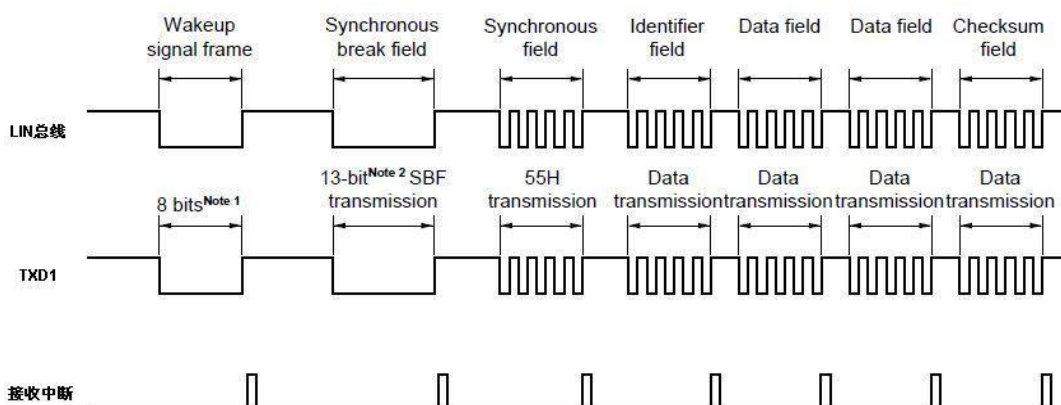
13.2.3.3. 和主机自动同步

讲入 LIN 通信，建立主从机连接，串行通信端（主机）的传输速度（波特率）按照下述步骤自动同步：

步骤 1：初始化接收接口用于接收，初始化定时器用于波特率测量

步骤 2：等待来自主机的 LIN 帧

步骤 3：和主机波特率同步 步骤 4：进入主机请求帧或从机回应帧 注：每个主机请求报文头或从机回应报文头 LIN 帧都要进行重新同步和波特率建立操作。



注意：

1. 唤醒信号帧通过设置 UART1 8 位模式发送 80H 完成。
2. 同步分割域由硬件产生，输出宽度由控制寄存器 LINCON (2:0) 设置。
3. 同步域自动进行波特率校准，校准的的波特率分频值存于寄存器 LINRCBH 和 LINRCBL。

13.2.4. UART1 寄存器

S1CON UART1 控制寄存器: 9BH 初始值: 00H

7	6	5	4	3	2	1	0
SM		SM21	REN1	TB81	RB81	TI1	RI1
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

-
- 7 SM UART1 模式选择
 0 — 9 位数据传送格式
 1 — 8 位数据传送格式
-
- 5 SM21 多机通讯使能
 0 — 禁止
 1 — 允许多机通信
-
- 4 REN1 允许接收控制位
 由软件置 1 或清 0，只有当 REN1=1 时，才允许接收，相当于串行接收的开关；若 REN1=0，则禁止接收。
 在串行通信接收控制程序中，如果满足 RI1=0，置位 REN1=1（允许接收）的条件，就会启动一次接收过程一帧数据就装入 S1BUF 中。
-
- 3 TB81 发送数据的第 9 位
 在模式 A 中，根据发送数据的需要由软件置位或复位。在许多通信协议中可作奇偶校验位。也可在多机通信中作为发送地址帧或数据帧的标志。
 对于后者，TB81=1，说明发送该帧数据为地址；TB81=0，说明发送该帧的数据为数据字节。
-
- 2 RB81 接收数据的第 9 位
 在模式 A 中，接收到的第 9 位数据放在 RB81 位。它或是约定的奇 / 偶校验位，或是约定的地址 / 数据标志位。
 在模式 B 多机通信中，若 SM21=1，如果 RB81=1，说明收到的数据为地址帧。
 在模式 B 中，若 SM21=0（即不是多机通信情况），RB81 中存放的是已接收到的停止位。
-
- 1 TI1 发送中断标志
 在一帧数据发送完时被置位。串行发送第 8 位或第 9 位结束时，由硬件置位，可用软件查询。TI1 不会自动清 0，必须由软件清 0。

0 RI1 接收中断标志

在接收到一帧有效数据后由硬件置位。在模式 A 和模式 B 中，在接收到停止位中间时由硬件置位。RI1 也必须由软件清 0。

S1BUF UART1 数据缓冲寄存器: 9CH 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

S1RELH UART1 重载寄存器高 8 位: 9DH 初始值: 00H

7	6	5	4	3	2	1	0
D15	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

S1RELL UART1 重载寄存器低 8 位: BBH 初始值: 03H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

UART1 波特率计算:

$$baudrate = \frac{Fclk}{32 \times (2^{10} - s1rel)}$$

s1rel 是 S1RELH:S1RELL 中的数值

IEN2 中断使能寄存器 2: 9AH

– [ES1 \(IEN2.0\) UART1 中断使能](#)

LINRCBL 同步域重新计算波特率寄存器低 8 位: A1H 初始值: 00H

7	6	5	4	3	2	1	0
D15	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LINRCBH 同步域重新计算波特率寄存器高 8 位: A2H 初始值: 03H

7	6	5	4	3	2	1	0
–	–	–	–	–	–	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

同步域重新计算波特率寄存器，
 可变波特率 = $F_{clk} / (32 * (2^{10} - (LINRCBH : LINRCBL)))$ 。
 LINRCBH 只用低 2bit，高 6bit 不用。

LINCON LIN 控制寄存器：A3H 初始值：00H

7	6	5	4	3	2	1	0
SBFRP	SBFRLS	-	-	INTIS	SBFTWID		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- 7 SBFRP 此位置 1 指示正在进行分割域 (SBF) 接收处理中。
- 6 SBFRLS 此位置 1 指出分割域 (SBF) 宽度小于 11 位。当它由 LIN 置 1 后，通过软件写 0 清除。
- 3 INTIS 中断唤醒切换开关：
 1 - LIN 通信中 RDX1 (P15) 引脚用作 INT1，可进行中断唤醒。
 0 - 正常 INT1 中断
- 2:0 SBFTWID 分割域 (SBF) 传输输出宽度控制。
 000 - 13 位
 001 - 14 位
 010 - 15 位
 011 - 16 位
 100 - 17 位
 101 - 18 位
 110 - 19 位
 111 - 20 位

SBFTIG 分割域 (SBF) 触发控制寄存器：A4H 初始值：00H

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

- 1 SBFRT 分割域 (SBF) 接收触发，接收 SBF 后硬件自动清零。
- 0 SBFTT 分割域 (SBF) 发送触发器，SBF 发送后硬件自动清零。

14. 同步串行通讯口（SPI）

同步串行通讯接口（SPI）是一种提供高速全双工同步串行通信的总线。SPI 可作为主设备或子设备在 3 线或 4 线模式中运作，并支持单个 SPI 总线上的多种主设备或子设备。有如下特点：

- 全双工模式
- 三线同步传送
- 主从模式
- 七个 SPI 主波特率
- 从时钟达到系统时钟的 $1/8(F_{clkper}/8)$
- 串行时钟可进行极性和相位编程
- 主模式错误标志可产生中断
- 写冲突标志保护
- 8 位数据发送时，最高位在前，最低位在后
- 1 位从选择输出口控制外部从设备
- 特殊功能寄存器接口到主 CPU
- 非双向口，标准 SPI 引脚被连到外部三态缓冲器

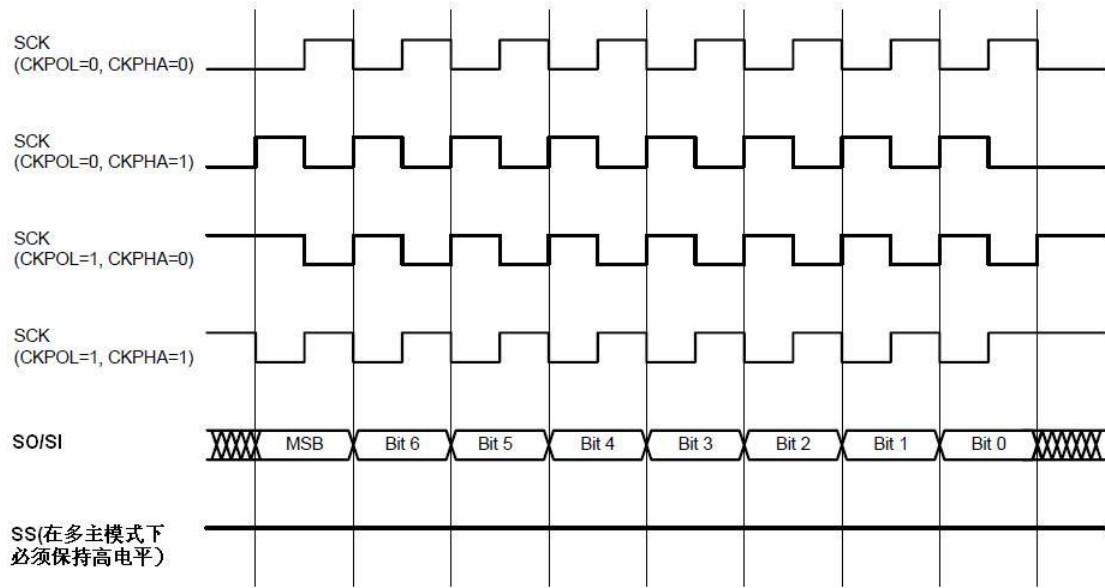
14.1. SPI 寄存器

SPCON SPI 控制寄存器：E2H 初始值：**14H**

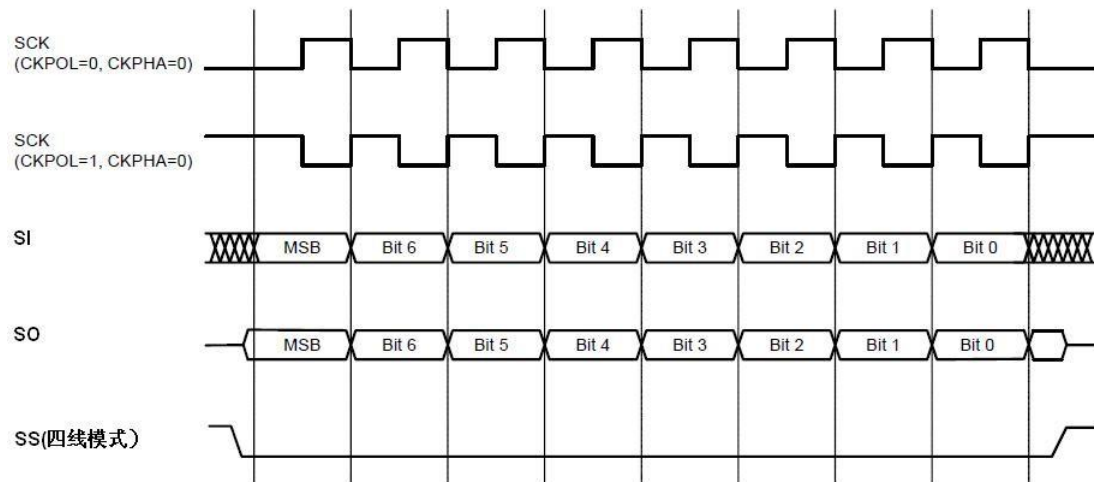
7	6	5	4	3	2	1	0
SPR2	SPEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	SPR2	通信速率设置，与 SP1 和 SPR0 一起使用，看下面 SP1 和 SPR0 说明。
6	SPEN	SPI 接口使能 0 - 禁止 1 - 使能
5	SSDIS	SS 禁止 在主从模式： 0 - 使能“SS”输入 1 - 禁止“SS”输入，SS 可以在从模式，如果 CPHA=0，这个位没有影响；当 SSDIS=1，没有“MODF”中断请求将产生
4	MSTR	主从设置 0 - SPI 配置为从模式 1 - SPI 配置为主模式
3	CPOL	时钟极性 0 - SCK 空闲时为低电平。SCK 的前时钟沿为上升沿而后沿为下降沿。 1 - SCK 空闲时为高电平。SCK 的前时钟沿为下降沿而后沿为上升沿。
2	CPHA	时钟相位 1 - 数据在 SCK 的前时钟沿被改变，并在后时钟沿采样。 0 - 数据在 SCK 的后时钟沿被改变，并在前时钟沿被采样。
1:0	SPR1:SPR0	通信速率设置 与 SPR2 一起使用，看下表：

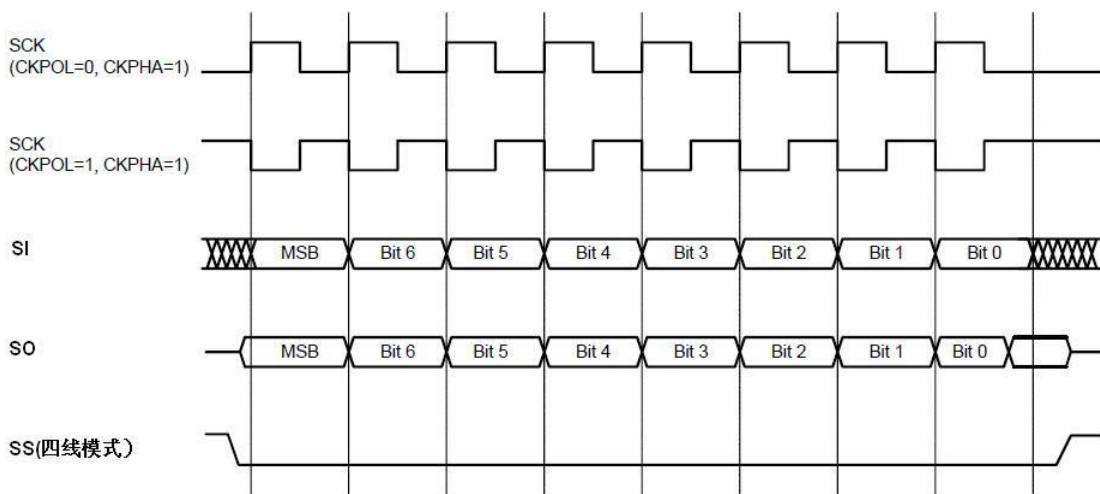
SPR2	SPR1	SPR0	SPI 速率
0	0	0	Fclk / 2
0	0	1	Fclk / 4
0	1	0	Fclk / 8
0	1	1	Fclk / 16
1	0	0	Fclk / 32
1	0	1	Fclk / 64
1	1	0	Fclk / 128
1	1	1	无主时钟 (当“CPOL” = ‘1’， “SCKO” 输出高电平，否则输出低电平)



主模式数据发送格式



从模式数据发送格式(CPHA=0)



从模式数据发送格式(CPHA=1)

SPSTA SPI 状态寄存器: E1H 初始值: 00H

7	6	5	4	3	2	1	0
SPIF	WCOL	SSERR	MODF	—	—	—	—
R	R	R	R	R	R	R	R

- 7 SPIF SPI 传输完成标志
当同步串行传输完成后，此位置 1。读 SPSTA，清此位。
- 6 WCOL 写入冲突标志
写数据到 SPIDAT 引起冲突 (发送数据寄存器未空)，由硬件置 1。读 SPSTA 和 SPDAT，清此位。
- 5 SSERR 从错误标志
在接收序列结束前，“SS”输入无效，硬件置 1。禁止 SPI 功能清此位 (清 SPCON 中的 SPEN 位)。
- 4 MODF 模式错误标志
当“SS”引脚的电平与 SPI 控制器实际的模式冲突时，由硬件置位。当“SS”引脚在适当的电平由硬件清除，也可通过读 SPSTA 清除此位。

SPDAT SPI 数据寄存器: E3H 初始值: 00H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SPSSN SPI 从选择寄存器: E4H **初始值: FFH**

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	SS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15. EEPROM 和 FLASH 读写, 程序加密

EEPROM 擦写次数在 2 万次以上。可分为若干个扇区, 每个扇区包含 128 字节, 共 2k 字节, 数据存储器的擦除操作是按扇区进行的。

EEPROM 可用于保存一些需要在应用过程中修改并且掉电不丢失的参数数据。在用户程序中, 可以对 EEPROM 进行字节读 / 字节编程 / 扇区擦除操作。

EEPROM 数据区映射到地址 1000H~17FFH XRAM 存储区。

当设置 PCON.4(PMW)为 1, 也可对程序 FLASH 区进行擦写操作。

15.1. EEPROM 编程寄存器

MCON EEPROM 编程控制寄存器: 8FH

初始值: 03H

7	6	5	4	3	2	1	0
-	-	-	-	XRAM_M ODE	FLH_ACK	FLH_OP	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

3 XRAM_MODE 为 1 时, 000H - BFFH XRAM 映射到程序空间 7800H~83FFH, 可移动程序代码到此区域高速运行。

2 FLH_ACK EEPROM 操作完成标志。

1:0 FLH_OP EEPROM 操作控制
 00-字节编程
 01-扇区擦除
 10-片擦除 (用户禁止)
 11-保持最后状态;
 除字节编程, FLH_OP 操作后自动清为 0。需要继续相同操作, 必须重置这两位。

设定扇区擦除后, 写擦除扇区中任意一个字节, 执行整个扇区擦除。

DSAB FLASH 编程模式寄存器: 97H

初始值: 00H

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

1 ER_DSAB 0-EEPROM 擦除使能
 1-EEPROM 擦除禁止

0	WR_DSAB	0-EEPROM 编程使能
		1-EEPROM 编程禁止

15.2. FLASH 擦写

当设置 PCON.4(PMW)为 1, 也可对程序 FLASH 区进行擦写操作。此时用访问外部存储器指令(MOVX), 对 FLASH 擦写, 操作方法同 EEPROM 擦写。

15.3. 程序加密

采用 EEPEOM 写字节方式, 把外部存储器地址 0x1800 单元中最高位写 0, 就完成程序加密, 整个 FLASH 不能读出正确的内容。加密后, 可以用擦除整个 FLASH 来重写新程序。

```
PCON = 0x08;
DSAB = 0x02;      // 编程使能, 擦除禁止
MCON = 0x00;      // 字节编程
ENCRYPTION = 0x00; // 写数据
DSAB = 0x03;      // 编程禁止, 擦除禁止
```

15.4. EEPEOM 擦写例程

```
#include "HYM8F028C6.H"

unsigned char xdata    EEPROM[2048] _at_ 0x1000;

main()
{
    DSAB = 0x01;      // 擦除使能, 编程禁止
    MCON = 0x01;      // EEPROM 片擦除
    EEPROM[0] = 0;     // 写扇区中任意一个字节, 执行整个扇区擦除
    EEPROM[128] = 0;
    EEPROM[256] = 0;

    DSAB = 0x02;      // 编程使能, 擦除禁止
    MCON = 0x00;      // 字节编程 EEPROM[0]
    = 0xA5;           // 写数据
    EEPROM[1] = 0x5A; // 写数据
```

```
    DSAB = 0x03;          // 编程禁止，擦除禁止
}
```

15.5. FLASH 擦写例程

```
#include "HYM8F028C6.H"

unsigned char xdata    XRAM[0x8000]    _at_ 0x0000;

main()
{
    DSAB = 0x01; // 编程禁止，擦除使能
    MCON = 0x01; // 扇区擦除
    PCON |= 0x10; // 设置 PMW=1，使能 FLASH 读写
    XRAM[0x4000] = 0xFF;
    Delay20ms();
    DSAB = 0x02; // 编程使能，擦除禁止
    MCON = 0x00; // 字节编程
    XRAM[0x4000] = 0xA5;

    DSAB = 0x03; // 编程禁止，擦除禁止
    PCON &= ~0x10; // PMW=0，使能 XRAM 读写
}
```

16. RESET

16.1. 复位说明

在复位，MCU 所有的寄存器进入初始状态。初始值可看 SFR 复位值。程序存储器，外部数据存储器，数据存储器芯片（内部数据存储器）和特殊功能寄存器接口控制输出设置无效。程序计数器复零。

16.2. 上电复位

在上电期间，设备处于复位状态，RST 引脚处于低电平，直至 VDD 上升至 VRST 以上时为止。设备从复位状态退出复位状态前会出现延时，延时随 VDD 上升时间的增加而减少（VDD 上升时间被定义为 VDD 从 0 V 上升到 VRST 的时间）。

由于所有复位都导致程序从同一个地址（0x0000）开始执行，软件可以通过读 SRSTREQ 标志来确定是否为上电引起的复位。在一次启动复位后，内部数据内存中的内容应被认为是未定义的。启动复位后，VDD 监视器被启用。



16.3. 掉电复位 (LVR)

当发生掉电或因电源波动导致 VDD 降到 V_{LVR} 以下时，电源监视器将 MCU 保持复位状态。当 VDD 又返回高于 V_{LVR} 的电平时，MCU 将退出复位状态。尽管内部数据内存的内容可能没有因掉电复位而发生改变，但无法确定 VDD 是否降到了数据保持所要求的最低电平以下。启动复位后，VDD 监视器被启用。但其设定状态（启用 / 禁用）不受任何其他复位源影响。例如，在 VDD 监视器通过代码被禁用后执行一次软件复位，复位后 VDD 监视器仍然为禁用状态。

LVR 电平有 7 种，用户可以根据使用的电源环境选用。

LVR_CTRL 低电压复位控制寄存器：3000H 初始值：00H

7	6	5	4	3	2	1	0
P07SEL	—	—	—	LVR_VTH			—
R/W	R	R	R	R	R	R/W	R/W

7 P07SEL P07 复位或 GPIO 选择，可以参考 P0FUN1 说明。
0 - GPIO
1 - 复位

3:1 LVR_VTH LVR 电压档位控制；
000-2.50V
001-2.75V
010-3.00V
011-3.25V
100-3.50V
101-3.75V
110-4.00V
111-4.25V

16.4. 外部复位

外部 RST 引脚提供了使用外部电路强制设备进入复位状态的手段。在 RST 引脚上加一个低电平有效信号产生复位；可能需要外部上拉及 / 或对 RST 去耦以避免过强噪声引起复位。

16.5. 看门狗复位

可编程看门狗 (Watchdog) 功能可用于在系统出现错误的情况下防止软件运行失控。在每次复位后，看门狗必须被软件启动。如系统故障阻止用户软件更新看门狗定时器，将产生一次复位。RST 引脚状态不受该复位影响。

16.6. 软件复位

详细说明看下面的软件复位寄存器解说。

16.7. 软件复位寄存器

SRST 软件复位寄存器: **F7H** 初始值: **00H**

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	SRSTREQ
R	R	R	R	R	R	R	R/W

- 0 SRSTREQ 软件复位请求, 通过写特殊寄存器 SRST 可产生软件复位。
 此位写“0”无影响。
 写一次“1”到此位也无影响。
 双写“1”值（在两个连续的指令）将产生一个内部软件复位。
 读此位可知复位源: 如果
 0-复位源不是软件复位（硬件、看门狗复位或调试器复位）
 1-复位源是软件复位（由双写“1”值到“SRSTREQ”位引起）。

RST_CTRL 复位控制寄存器: **B1H** 初始值: **00H**

7	6	5	4	3	2	1	0
—	—	—	—	—	PIN_RST	BOR_RST	POR_RST
R	R	R	R	R	R	R	R

- 2 外部引脚复位标志
 0-无复位
 1-复位

- 1 BOR 复位标志
 0-无复位
 1-复位

- 0 POR 复位标志
 0-无复位
 1-复位

17. 电源模式

有三种软件可编程的省电模式：空闲 (IDLE)、停止 (STOP)。这两种模式是标准 8051 结构的一部分。

空闲模式下，CPU 暂停而外围设备和时钟激活。停止模式下，CPU 暂停、所有中断处理程序和定时器均处于非激活状态，内部振荡器停止（模拟外围设备依然处于所选状态；外部振荡器不受影响）。由于时钟在空闲模式下运行，决定电耗的是系统时钟的频率和进出空闲状态前处于激活状态的外围设备数目。

停止模式耗电最少，因为大多数设备都被关闭，且没有时钟处于激活状态。

17.1. 空闲模式

设置空闲模式选择位 (PCON.0)，硬件让 CPU 暂停并在设置位的指令执行完成后立即进入空闲模式。所有内部寄存器和存储器均维持其原始数据。在空闲模式下，所有模拟和数字外围设备均可保持激活状态。

允许中断事件出现或出现复位时，空闲模式即告终止。中断将引起空闲模式选择位 (PCON.0) 被清除及 CPU 重新运行。中断将得到执行，而从中断中回复后要执行的下个指令将是紧随设置空闲模式选择位之后的指令。如空闲模式是因内部或外部复位而终止，则 MCU 执行正常的复位流程，并在 0x0000 位置开始程序执行。

附注：如写入空闲位之后的指令是一个单字节指令，且中断出现在设置空闲位的指令的执行阶段，未来出现中断时 CPU 可能无法从空闲模式中唤醒。因此，设置空闲位的指令后面应该紧随一个两字节或更多字节的指令，例如：

// 用 'C' 语言：

```
PCON |= 0x01;    // 设置IDLE 位
PCON = PCON;     // ...跟随一条3周期伪指令
```

//用汇编语言：

```
ORL PCON, #01h    ; 设置IDLE 位
MOV PCON, PCON    ; ...跟随一条3周期伪指令
```

如启用看门狗将最终引起内部看门狗复位，从而终止空闲模式。此特点保护系统在出现不慎写入 PCON 寄存器事件时免于被意外永久关机。如不需要此行为，而 WDT 最初被配置为允许此操作，可在进入空闲模式前通过软件禁用看门狗。这提供附加的省电机会，无限期允许系统保持空闲模式，等待外部刺激将其唤醒。参见看门狗了解使用及配置方面的更多资料。

17.2. 停止模式

设置停止模式选择位 (PCON.1)，在设置位的指令执行完成后立即让控制器内核进入

停止模式。在停止模式下，内部振荡器、CPU、Watchdog 及所有数字外围设备均停止；外部振荡器电路的状态不受影响。每一个模拟外围设备（包括外部振荡器电路）在进入停止模式前可能分别关闭。停止模式能通过外部复位终止。复位时，设备执行正常的复位流程，并在 0x0000 位置开始程序执行。INT0、INT1 也能退出 STOP 模式，并执行 STOP 的下一条指令。

17.3. 电源模式控制寄存器

PCON 电源模式控制寄存器：87H

初始值：08H

7	6	5	4	3	2	1	0
SMOD	WDTTM	ISR_TM	PMW	P2SEL	GF0	STOP	IDLE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	SMOD	UART0 波特率选择 具体用法可看 UART0 中波特率设置说明。
6	WDTTM	看门狗测试模式标志 1 - 看门狗输入时钟 12 预分频选择。 0 - 跳过预分频器。
5	IR_TM	中断测试。
4	PMW	程序存储器写使能 1 - 使能。 0 - 禁止。 当 PMW 为 0，MOVX 指令读写外部数据存储器空间 (XRAM)。 当 PMW 为 1，MOVX 指令读写程序存储器空间 (FLASH)。
3	P2SEL	高地址字节配置位。当执行 MOVX @Ri, operations; 指令时，高 8bits 地址选择控制。 0 - 高地址值为 P2 输出寄存器。 1 - 高地址值为 00H。
2	GF0	通用标志位
1	STOP	停止模式控制 设置此位进入停止模式，读此位总是为“0”。
0	IDLE	空闲模式控制 设置此位进入空闲模式，读此位总是为“0”。

18. 电气特性

18.1. 极限值

参 数		符号	极限值	单位	备 注
电源电压		V _{DD}	-0.3 ~ +6.5	V	
正常电压 引脚	输入电压	V _I	-0.3~V _{DD} +0.3	V	相对于 vss，任何引脚电压
	输出电压	V _o	-0.3 ~ V _{DD} +0.3	V	
	灌电流	I _{OH}	-6	mA	每个 I/O 端口最大的灌电流
	拉电流	I _{OL}	20	mA	每个 I/O 端口最大拉电流
总功率消耗		P _T	600	mW	
储存温度		T _{STG}	-65 ~ +150	℃	

18.2. 推荐操作参数

($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$, $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$)

参数	符号	测试条件	最小值	典型值	最大值	单位
工作电压	V_{DD}		2.7	-	5.5	V
工作温度	T_{OPR}	$V_{DD} = 2.7\text{V} \sim 5.5\text{V}$	-40	-	85	°C

18.3. 直流特性

($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$, $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$, $V_{SS} = 0\text{V}$, $f_{XIN} = 20\text{MHz}$)

参数	符号	测试条件	最小值	典型值	最大值	单位
输入高电压	V_{IH}	所有输入脚	$0.8V_{DD}$	—	V_{DD}	V
输入低电压	V_{IL}	所有输入脚	—	—	$0.2V_{DD}$	V
输出高电压	V_{OH}		$0.8V_{DD}$ (5V) $V_{DD}-0.8$ (3.3V)			V
输出低电压	V_{OL}				$0.1V_{DD}$ (5V) 0.5 (3.3V)	V
输入高漏电流	I_{IH}	所有输入端口	—	—	1	μA
输入低漏电流	I_{IL}	所有输入端口	-1	—	—	μA
上拉电阻	R_{PU}	$V_{DD}=5.0$, 所有输入口		20	100	$\text{k}\Omega$
下拉电阻	R_{PD}	$V_{DD}=5.0$, 所有输入口		20	100	$\text{k}\Omega$
供电电流	I_{DD1} (RUN)	$f_{PLL} = 60\text{MHz}$, $V_{DD} = 5\text{V} \pm 10\%$	—		20	mA
	I_{DD2} (IDLE)	$f_{PLL} = 60\text{MHz}$, $V_{DD} = 5\text{V} \pm 10\%$			10	mA
		$f_{PLL} = 60\text{MHz}$, $V_{DD} = 3\text{V} \pm 10\%$			8	mA
	I_{DD3}	STOP, $V_{DD} = 5\text{V} \pm 10\%$, $T_A = 25^{\circ}\text{C}$	—		560	μA

18.4. 交流特性

($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$, $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$)

参数	符号	测试条件	最小值	典型值	最大值	单位
RST 输入低脉宽	t_{RST}	输入, $V_{DD}=5\text{V}$	—	—	100	μs
系统复位延时时间(上电复位)	t_{RSTD}		—	64	—	ms
中断输入高、低脉宽	t_{IWH} , t_{IWL}	所有中断, $V_{DD}=5\text{V}$	6	—	—	Clock cycles
计数器外部输入高、低脉宽	t_{ECWH} , t_{ECWL}	EXn ($n=0, 1, 2$), $V_{DD}=5\text{V}$	6	—	—	
计数器外部输入翻转时间	t_{REC} , t_{FEC}	EXn ($n=0, 1, 2$), $V_{DD}=5\text{V}$	6	—	—	

18.5. 模数(AD)转换器特性

($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$, $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$)

参数	符号	测试条件	最小值	典型值	最大值	单位
分辨率	—		12	12	12	bit
ENOB			9.5	—	—	bit
积分线性误差	INL		—	—	1	LSB
微分线性误差	DNL		—	—	1	
零偏误差	ZOE				1	
满刻度误差	FSE				1	
转换时间	t_{CON}	15MHz 时钟			1	μs
电源电压范围	V_{DD}		2.7	—	5.5	V
模拟输入电压	V_{AN}		V_{SS}	—	AV_{REF}	V
模拟基准电压	AV_{REF}		2.0		V_{DD}	V
模拟输入漏电流	I_{AN}	$AV_{REF}=5\text{V}$	—	—	—	μA
ADC 工作电流	I_{ADC}	使能	—	3		mA
		禁止			0.1	μA

18.6. (LVR) 低电压复位特性

($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$, $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$)

参数	符号	测试条件	最小值	典型值	最大值	单位
检测电压	V_{LVR}			2.50		V
				2.75		
				3.00		
				3.25		
				3.50		
				3.75		
				4.00		
				4.25		
				4.50		
回差	ΔV	逻辑低电位				mV
		逻辑高电位				
最小脉宽	t_{LW}		100	—	—	μs
电流消耗	I_{LVR}	工作	—	—		μA
		停止	—	—	0.1	μA

18.7. 模拟比较器特性

($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$, $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$)

参数	符号	测试条件	最小值	典型值	最大值	单位
输入漏电流	I_L	$V_{DD}=5\text{V}, V_{IN}=1/2V_{DD}$	—	1	—	nA
电压偏移	V_{offset}	$V_{DD}=5\text{V}, V_{IN}=1/2V_{DD}$	—	—	10	mV
工作电流	I_{OP}	CMP1EN=H	0.125	0.15	0.25	mA
不工作电流	I_{PD}	CMP1EN=L	—	4	—	nA
响应时间	V_{RT}	CL=1pF, $V_{DD}=5\text{V}$	45	55	75	ns

18.8. 可编程增益放大器 (PGA) 特性

(TA= -40°C ~ +125°C, VDD= 2.7V ~ 5.5V)

参数	符号	测试条件	最小值	典型值	最大值	单位
输入电压范围	$V_{DIFFMAX}$		GND	—	$AV_{DD}-1.2V$	V
输入偏移电压	V_{IO}		—	—	10	mV
增益线性度	L_V	G=1	$1 - \frac{1}{2}LSB$	1	$1 + \frac{1}{2}LSB$	V/V
		G=2	$2 - \frac{1}{2}LSB$	2	$2 + \frac{1}{2}LSB$	
		G=4	$4 - \frac{1}{2}LSB$	4	$4 + \frac{1}{2}LSB$	
		G=8	$8 - \frac{1}{2}LSB$	8	$8 + \frac{1}{2}LSB$	
		G=10	$10 - \frac{1}{2}LSB$	10	$10 + \frac{1}{2}LSB$	
		G=12	$12 - \frac{1}{2}LSB$	12	$12 + \frac{1}{2}LSB$	
		G=16	$16 - \frac{1}{2}LSB$	16	$16 + \frac{1}{2}LSB$	
最大增益误差	E_G		—	1	2	%
增益带宽	GBW		0	—	4	MHz
电压上升速率	SR	@CL=20pF	15	—	—	V/ μs
工作电流	I_{OP}		—	—	500	μA
不工作电流	I_{PD}		—	—	0.1	μA

18.9. DAC 特性

(TA= -40°C ~ +125°C, VDD= 2.7V ~ 5.5V)

参数	符号	测试条件	最小值	典型值	最大值	单位
DAC 基准输入	V_{IN}		AV_{SS}	—	AV_{DD}	nA
DAC 步长	V_{STEP}		$0.7\% \frac{V_{IN}}{32}$	$\frac{V_{IN}}{32}$	$1.25\% \frac{V_{IN}}{32}$	mV
输出电压范围	V_{OUT}		$\frac{V_{IN}}{32}$	—	V_{IN}	V
积分线性误差	INL		—	—	1	LSB
微分线性误差	DNL		—	—	1	LSB
工作电流	I_{OP}		—	—	100	nA
不工作电流	I_{PD}		—	—	0.1	μA

18.10. 内部 RC 振荡器特性

($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$, $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$)

参数	符号	测试条件	最小值	典型值	最大值	单位
频率	f_{IRC}	$V_{DD} = 2.7\text{V} \sim 5.5\text{V}$	—	500	—	kHz
误差		$T_A = -40^{\circ}\text{C} \sim 125^{\circ}\text{C}$	—	—	± 2	%
占空比	TOD	—	45	50	55	%
稳定时间	T_{HFS}	—	—	—	50	μs
电流消耗	I_{IRC}	工作	—	—	20	μA
		停止	—	—	0.1	μA

内部 1.8V 工作电压

18.11. PLL 特性

($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$, $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$)

参数	符号	测试条件	最小值	典型值	最大值	单位
输入时钟频率	f_{IN}		—	500	—	kHz
输出时钟频率	f_{out}		—	60	—	MHz
VCO 中心频率	f_{vco}		—	60	—	MHz
分频比			—	120	—	
相位裕度			50	—	—	度
带宽			—	—	50	kHz
JITTER			—	100	—	ps
电流消耗	I_{OP}	工作	—	—	200	μA
		停止	—	—	0.1	μA

内部 1.8V 工作电压

18.12. UART 特性

($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$, $V_{DD} = 2.7\text{V} \sim 5.5\text{V}$, $f_{XIN} = 20\text{MHz}$)

参数	符号	最小值	典型值	最大值	单位
串口时钟周期时间	t_{sck}		$t_{CPU} \times 1$ 6	—	ns
输出到时钟上升沿数据设置	t_{s1}		$t_{CPU} \times 1$ 3	—	
时钟上升沿到输入数据有效	t_{s2}	—	—	—	
时钟上升沿后输出数据保持	t_{H1}		t_{CPU}	—	
时钟上升沿后输入数据保持	t_{H2}	0	—	—	

18.13. 内部 Flash Rom 特性

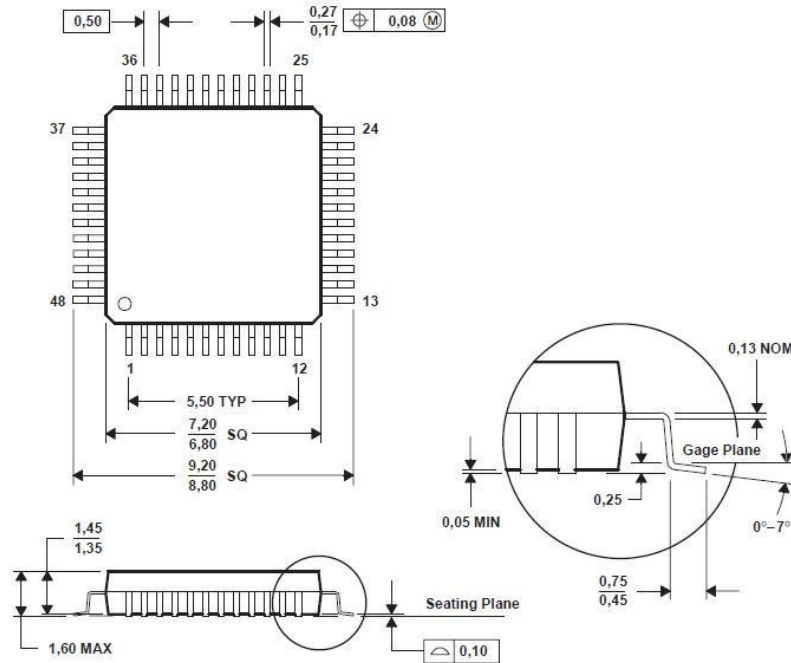
($T_A = -40^{\circ}\text{C} \sim +125^{\circ}\text{C}$, $V_{DD} = 1.8\text{V} \sim 5.5\text{V}$, $V_{SS} = 0\text{V}$)

参数	符号	测试条件	最小值	典型值	最大值	单位
扇区写时间	t_{FSW}			—	—	ms
扇区擦除时间	t_{FSE}		4	—	6	
锁定时间	t_{FHL}			—	—	
页缓冲器复位时间	t_{FBR}			—	—	μs
编程频率	f_{PGM}				60	MHz
擦写寿命	N_{FWE}		—	—	20000	次

19. 封装尺寸

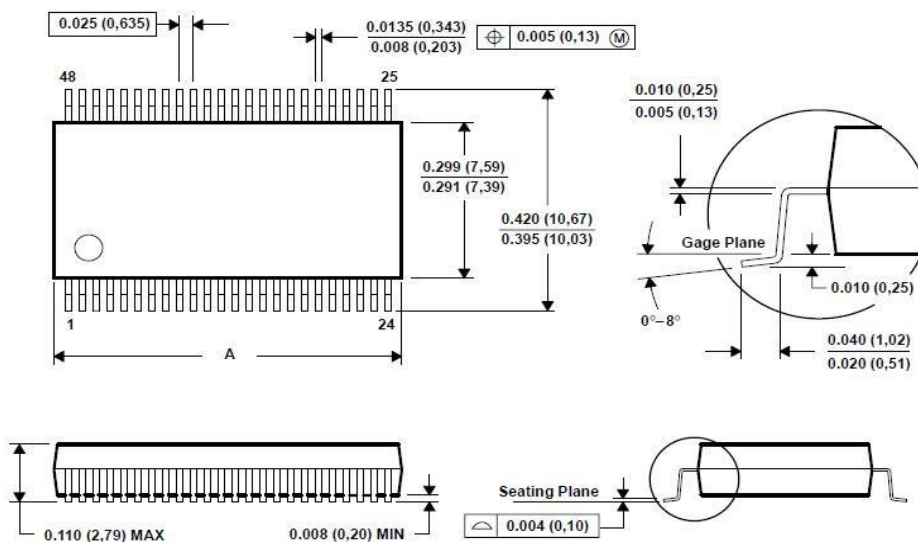
LQFP48 PLASTIC QUAD FLATPACK

Dimensions: mm



SSOP48 PLASTIC SMALL-OUTLINE PACKAGE

Dimensions: inch (mm)



	PINS		
DIM	28	48	56
A MAX	0.380 (9.65)	0.630 (16.00)	0.730 (18.54)
A MIN	0.370 (9.40)	0.620 (15.75)	0.720 (18.29)