高速 1T 8051 内核 Flash MCU ，4 Kbytes SRAM ，64 Kbytes Flash，0~4 Kbytes LDROM，6 Kbytes 独立 EEPROM， 12 位 ADC，14 路 16 位 PWM，5 个定时器，乘除法器，UART，3 路 USCI，CRC 校验模块

# 1 总体描述

SC95F7675/7673/7672/7671 是一系列增强型的高速 1T 8051 内核工业级的 Flash 微控制器，指令系统完全兼容传统 8051 产品系列。

SC95F767X 具有超高速 1T8051 CPU 内核，运行频率高达 32MHz，在相同工作频率下，其执行速度约为其它 1T 8051 的 2 倍；IC 内部集成硬件乘除法器、硬件 CRC 及双 DPTR 数据指针，用来加速数据运算及移动的速度。硬件乘除法器、硬件 CRC 不占用 CPU 周期，运算由硬件实现，速度比软件实现速度更快；双 DPTR 数据指针，可用来加速数据存储及移动。

SC95F767X 系列具有高性能和可靠性，具有宽工作电压 2.0V~5.5V，超宽工作温度-40℃~105℃，并具备良好的 ESD 性能及 EFT 抗干扰能力。采用业界领先的 eFlash 制程， Flash 写入>10 万次，常温下可保存 100 年。

SC95F767X 系列内建低功耗 WDT 看门狗定时器，有 4 级可选电压 LVR 低电压复位功能及系统时钟监控功能，具备运行和掉电模式下的低功耗能力。正常工作模式：5V 下典型约 4.5mA@32M。

SC95F767X 系列还集成有超级丰富的硬件资源：64 Kbytes Flash ROM、SRAM ：内部 256 bytes+外部 4 Kbytes+ PWM RAM、6 Kbytes EEPROM、最多 30 个 GP I/O（全部可分级控制）、13 个 IO 可外部中断、5 个 16 位定时器、共 14 路 16bit PWM：8 路带死区互补的多功能 PWM，6 路由 Timer 输出的 PWM、1 个 UART，3 个 UART/SPI/TWI 三选一通信口 USCI、内部±1%高精度高频 32/16/8/4MHz 振荡器和±4%精度低频 32kHz 振荡器、可外接 2~16MHz 高频晶体振荡器、13 路 12 位高精度 ADC。

SC95F767X 开发调试非常方便，具有 ISP（In System Programing）、ICP（In Circuit Programing）和 IAP(In Application Programing)功能。允许芯片在线或带电的情况下，直接在电路板上对程序存储器进行调试及升级。

SC95F767X 具有非常优异的抗干扰性能，非常适合应用于各种使用场合的主控控制，如大小智能家电和智能家居、物联网、无线通讯、游戏机等工业控制和消费应用领域。

Preliminary

# 2 主要功能

**工作条件：**
- 工作电压：2.0V~5.5V
- 工作温度：-40 ~ +105℃

**CPU**
- 超高速 1T 8051 内核， 指令集全兼容 8051，执行速度约为其它 1T 8051 的 2 倍
- 双数据指针（DPTRs）

**Flash ROM：**
- 64 Kbytes Flash ROM
- 可重复写入 10 万次
- 可通过 Code Option 设置项将 APROM 区域可 IAP 操作的范围设为 0K/1K/2K/全部 APROM

**LDROM**
- 用于存放用户的 BootLoader 引导代码（boot code）
- 可通过 Code Option 设置项将 LDROM 设为 0K/1K/2K/4K

**EEPROM**
- 独立的 6 Kbytes EEPROM
- 10 万次写入，常温下 100 年以上保存寿命

**SRAM**
- 256 bytes 片内直接存取 RAM
- 额外 4 Kbytes 片内间接存取 RAM
- PWM 专用 RAM

**烧写和仿真：**
- 2 线 JTAG 烧写和仿真接口

**系统时钟（$f_{SYS}$）：**
- 内建高频 32MHz 振荡器（$f_{HRC}$）
  - 作为系统时钟源时，$f_{SYS}$ 可通过编程器选择设定为 32/16/8/4MHz@2.0~5.5V
  - 频率误差：跨越(2.0V~5.5V) 及 (-40 ~ 105℃)应用环境，不超过 ±1%
- 内置高频晶体振荡器电路
  - 可外接 2~16MHz 振荡器
  - 作为系统时钟源时，$f_{SYS}$ 可通过编程器选择使用外接晶振 /1 /2 /4 /8 这四种分频中的一种
  - 内建系统时钟监控电路，当用户选择系统时钟源为晶体振荡并且晶体振荡电路发生停振时，系统时钟源将被自动切换到内建的 HRC，并保持此状态直至下次复位时再重置

**内建低频 32kHz LRC 振荡器：**
- 可作为 Base Timer 的时钟源，可唤醒 STOP
- 可作为 WDT 的时钟源
- 频率误差：跨越 (4.0V ~ 5.5V) 及 (-20 ~ 85℃) 应用环境，频率误差不超过 ±4%

**低电压复位（LVR）：**
- 复位电压有 4 级可选：分别是：4.3V、3.7V、2.9V、1.9V
- 缺省值为用户烧写 Code Option 所选值

**中断（INT）：**
- Timer0~4，INT0~2，ADC，PWM，UART，USCI0~2，Base Timer 共 15 个中断源
- 外部中断有 3 个中断向量，共 13 个中断口，全部可设上升沿、下降沿、双沿中断
- 两级中断优先级可设

**数字外围：**
- GPIO 最大 30 个双向可独立控制的 I/O 口
  - 可独立设定上拉电阻
  - 全部 GPIO 源驱动能力分四级控制
  - 全部 IO 具有大灌电流驱动能力（50mA）
- 内置 WDT，可选时钟分频比
- 5 个定时器 Timer0~4
  - Time2、Timer3 和 Timer4 可实现 Capture 功能
  - Time2、Timer3 和 Timer4 可各提供两路常规 PWM
- 6 路 16 位常规 PWM
  - 由 Timer2、Timer3 和 Timer4 各提供两路
- 8 路 16 位多功能 PWM
  - 共用周期、占空比单独可调
  - 带死区、可互补 PWM 输出
  - 20PIN 及以上管脚封装的型号支持 PWM 故障检测机制(FLT)
- 1 个独立 UART 通信口
- 3 个 UART/SPI/TWI 三选一通讯口 USCI
  - 其中，当 USCI0 被设置为 SPI0 时，其信号口所对应管脚的驱动能力将增强
- 集成 16 ×16 位硬件乘除法器（MDU）
- 内建 CRC 校验模块

**模拟外围：**
- 13 路 12 位 ADC
  - 内建基准的 2.048V/ 1.024V/ 2.4V 参考电压
  - ADC 的参考电压有 4 种选择，分别是 $V_{DD}$ 和内部 2.048V/ 1.024V/ 2.4V
  - 内部一路 ADC 可直接测量 $V_{DD}$ 电压
  - 可设 ADC 转换完成中断

**省电模式：**
- IDLE Mode，可由任何中断唤醒
- STOP Mode，由 INT0~2、BaseTimer 唤醒

Preliminary

# 产品命名规则

| 名称 | SC | 95 | F | 7 | 6 | 7 | 5 | X | P | 32 | R |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 序号 | ① | ② | ③ | ④ | ⑤ | ⑥ | ⑦ | ⑧ | ⑨ | ⑩ | ⑪ |

| 序号 | 含义 |
|---|---|
| ① | Sinone Chip 缩写 |
| ② | 产品系列名称 |
| ③ | 产品类型（F：Flash MCU） |
| ④ | 系列号：7：GP 系列，8：TK 系列 |
| ⑤ | ROM Size：1 为 2K，2 为 4K，3 为 8K，4 为 16K，5 为 32K, 6 为 64K, 7 为 128K… |
| ⑥ | 子系列编号：0~9，A~Z |
| ⑦ | 引脚数：0：8pin， 1：16pin，2：20pin，3：28pin，5：32pin，6：44pin，7：48pin，8：64pin，9：100pin |
| ⑧ | 版本号：（缺省、B、C、D） |
| ⑨ | 封装形式：（D：DIP；M：SOP；X：TSSOP；F：QFP；P：LQFP；Q：QFN；K：SKDIP） |
| ⑩ | 引脚数 |
| ⑪ | 包装方式：（U：管装；R：盘装；T：卷带） |

Preliminary

# 订购信息

| 产品编号 | 封装 | 包装 |
|---|---|---|
| SC95F7671M16U | SOP16 | 管装 |
| SC95F7672M20U | SOP20 | 管装 |
| SC95F7672X20U | TSSOP20 | 管装 |
| SC95F7672Q20R | QFN20 | 盘装 |
| SC95F7673M28U | SOP28 | 管装 |
| SC95F7673X28U | TSSOP28 | 管装 |
| SC95F7673Q28R | QFN28 | 盘装 |
| SC95F7675P32R | LQFP32 | 盘装 |
| SC95F7675Q32R | QFN32 | 盘装 |

Preliminary

# 目录

1 总体描述 ..... 1

2 主要功能 ..... 2

产品命名规则 ..... 3

订购信息 ..... 4

目录 ..... 5

3 管脚定义 ..... 10

3.1 管脚配置 ..... 10

3.2 管脚定义 ..... 14

4 内部框图 ..... 16

5 FLASH ROM 和 SRAM ..... 17

5.1 APROM 和 LDROM ..... 17

5.2 6K bytes 独立 EEPROM ..... 18

5.3 96 bits Unique ID 区域 ..... 18
5.3.1 Unique ID 读取 C 语言例程 ..... 19

5.4 User ID 区域 ..... 19

5.5 编程 ..... 19
5.5.1 JTAG 专用模式 ..... 20
5.5.2 常规模式（JTAG 专用口无效） ..... 20

5.6 IAP (Application Programming) 操作 ..... 20
5.6.1 IAP 操作相关寄存器 ..... 22
5.6.2 IAP 操作流程 ..... 23
5.6.3 IAP 操作 C 语言例程 ..... 23

5.7 BootLoader 功能 ..... 24
5.7.1 BootLoader 模式操作相关寄存器 ..... 24

5.8 安全加密 ..... 27

5.9 Customer Option 区域（用户烧写设置） ..... 27
5.9.1 Option 相关 SFR 操作说明 ..... 29

5.10 SRAM ..... 30
5.10.1 内部 256 bytes SRAM ..... 30

Preliminary

5.10.2 外部 4096 bytes SRAM ..... 31
5.10.3 外部 PWM SRAM ..... 31

**6 特殊功能寄存器(SFR) ..... 32**

**6.1 SFR 映像 ..... 32**

**6.2 SFR 说明 ..... 33**
6.2.1 特殊功能寄存器 ..... 33
6.2.2 PWM0 占空比调节寄存器（读/写） ..... 34
6.2.3 PWM2~4 占空比调节寄存器（读/写） ..... 35
6.2.4 8051 CPU 内核常用特殊功能寄存器介绍 ..... 35

**7 电源、复位和时钟 ..... 38**

**7.1 电源电路 ..... 38**

**7.2 上电复位过程 ..... 38**
7.2.1 复位阶段 ..... 38
7.2.2 调入信息阶段 ..... 38
7.2.3 正常操作阶段 ..... 38

**7.3 复位方式 ..... 38**
7.3.1 外部 RST 复位 ..... 39
7.3.2 低电压复位 LVR ..... 39
7.3.3 上电复位 POR ..... 39
7.3.4 看门狗复位 WDT ..... 39
7.3.5 软件复位 ..... 40

**7.4 高频系统时钟电路 ..... 40**

**7.5 低频振荡器及低频时钟定时器 ..... 42**

**7.6 STOP 模式和 IDLE 模式 ..... 43**

**8 中央处理单元 CPU 及指令系统 ..... 44**

**8.1 CPU ..... 44**

**8.2 寻址方式 ..... 44**
8.2.1 立即寻址 ..... 44
8.2.2 直接寻址 ..... 44
8.2.3 间接寻址 ..... 44
8.2.4 寄存器寻址 ..... 44
8.2.5 相对寻址 ..... 44
8.2.6 变址寻址 ..... 45
8.2.7 位寻址 ..... 45

**9 INTERRUPT 中断 ..... 46**

**9.1 中断源、向量 ..... 46**

**9.2 中断结构图 ..... 48**

Preliminary

9.3 中断优先级 ..... 49

9.4 中断处理流程 ..... 49

9.5 中断相关 SFR 寄存器 ..... 49

10 定时器 TIMER0 、TIMER1 ..... 54

10.1 T0 和 T1 相关特殊功能寄存器 ..... 54

10.2 T0 工作模式 ..... 56

10.3 T1 工作模式 ..... 57

11 定时器 TIMER2/3/4 ..... 59

11.1 T2/3/4 相关特殊功能寄存器 ..... 59

11.2 定时器 Timer2 ..... 59

11.3 定时器 Timer3 ..... 61

11.4 定时器 Timer4 ..... 62

11.5 Timer2/3/4 工作模式 ..... 64
11.5.1 Timer2/3/4 工作模式说明 ..... 64

12 常规脉冲宽度调制计数器 PWM2/3/4 ..... 68

12.1 PWM2/3/4 相关寄存器 ..... 68

12.2 PWM2/3/4 占空比变化特性 ..... 70

12.3 PWM2/3/4 周期变化特性 ..... 70

13 多功能脉冲宽度调制计数器 PWM0 ..... 70

13.1 PWM0 结构框图 ..... 70

13.2 PWM0 通用配置寄存器 ..... 71
13.2.1 PWM0 通用配置寄存器 ..... 71
13.2.2 PWM0 故障检测功能设置 ..... 75

13.3 PWM0 独立模式 ..... 76
13.3.1 PWM0 独立模式框图 ..... 76
13.3.2 PWM0 独立模式占空比配置 ..... 77

13.4 PWM0 互补模式 ..... 77
13.4.1 PWM0 互补模式框图 ..... 77
13.4.2 PWM0 互补模式占空比配置 ..... 78
13.4.3 PWM0 互补模式死区时间设置 ..... 78
13.4.4 PWM0 死区输出波形 ..... 78

Preliminary

13.5 PWM 波形及用法 ..... 79

14 GP I/O ..... 81

14.1 GPIO 结构图 ..... 81

14.2 I/O 端口相关寄存器 ..... 82

15 UART0 ..... 85

15.1 串口通信的波特率 ..... 86

16 SPI/TWI/UART 三选一串行接口 USCI ..... 87

16.1 SPI ..... 88

16.1.1 SPI0/1/2 ..... 88
16.1.1.1 SPI0/1/2 操作相关寄存器 ..... 88
16.1.1.2 SPI0/1/2 信号描述 ..... 90
16.1.1.3 SPI0/1/2 工作模式 ..... 90

16.2 TWI ..... 93
16.2.1 信号描述 ..... 95
16.2.2 从机工作模式 ..... 95
16.2.3 从机模式操作步骤 ..... 97
16.2.4 主机工作模式 ..... 97
16.2.5 主机模式操作步骤 ..... 99

16.3 UART ..... 100

17 模数转换 ADC ..... 102

17.1 ADC 相关寄存器 ..... 102

17.2 ADC 转换步骤 ..... 104

18 硬件 CRC 模块 ..... 105

18.1 CRC 校验操作相关寄存器 ..... 105

19 乘除法器 ..... 106

20 电气特性 ..... 108

20.1 极限参数 ..... 108

20.2 推荐工作条件 ..... 108

20.3 Flash ROM 参数 ..... 108

20.4 直流电气特性 ..... 108

Preliminary

20.5 交流电气特性 ..... 110

20.6 ADC 电气特性 ..... 110

21 封装信息 ..... 112

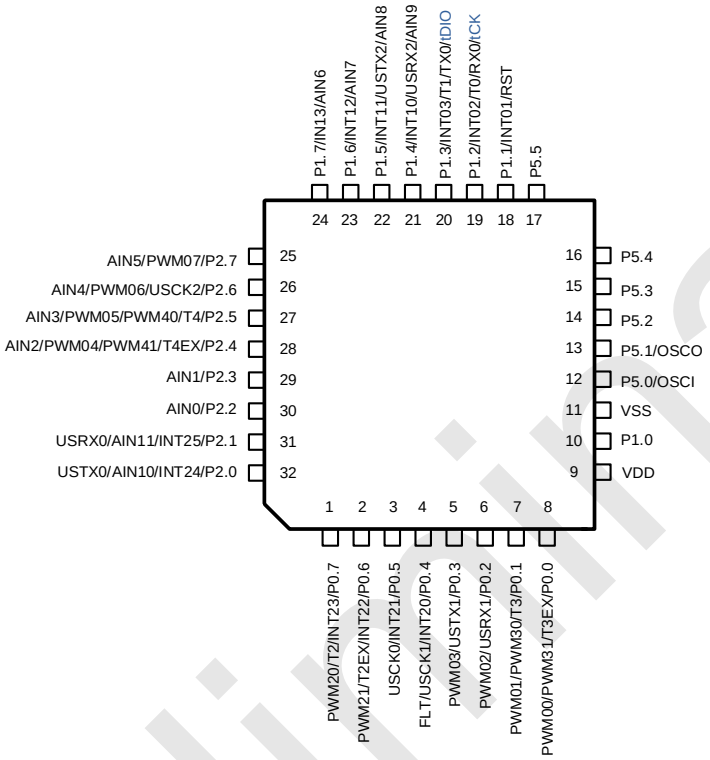
22 规格更改记录 ..... 122

声明 ..... 123

Preliminary

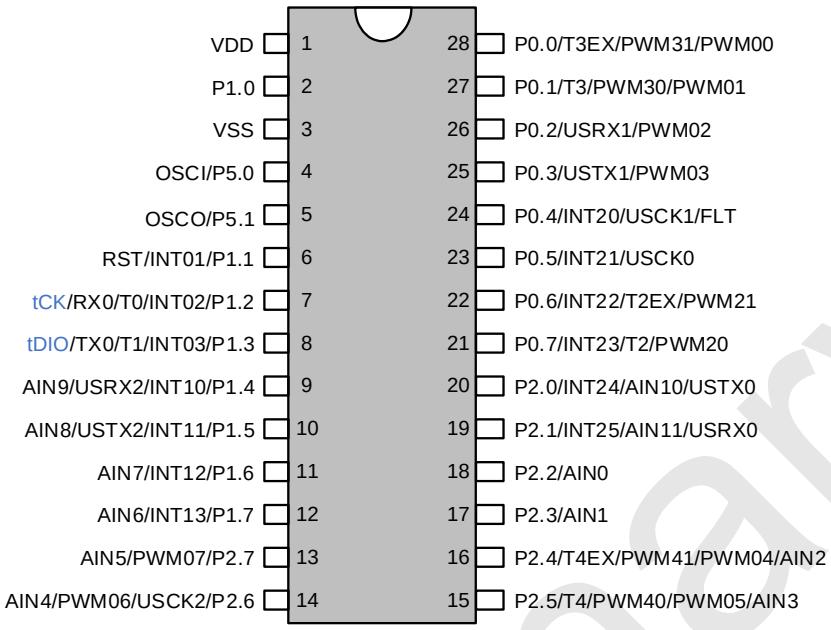
# 3 管脚定义

## 3.1 管脚配置

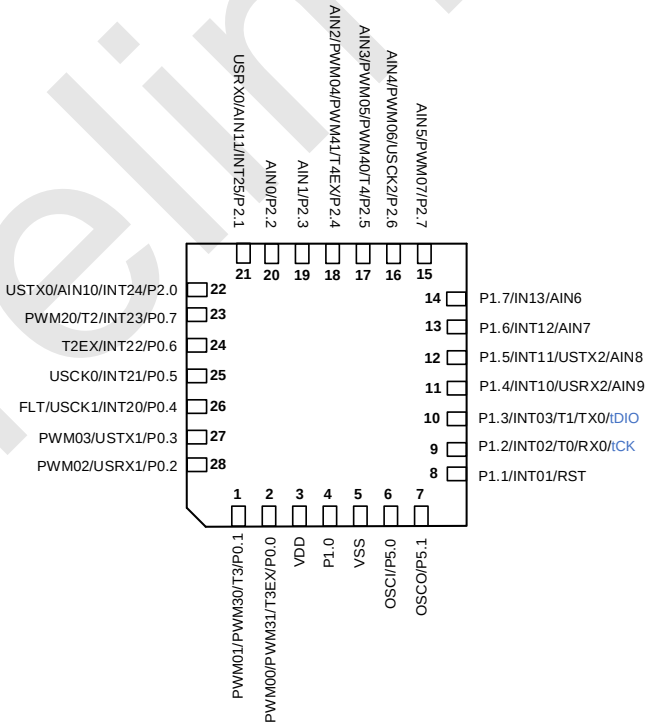



SC95F7675 管脚配置图
适用于 LQFP32、QFN32 封装

Preliminary

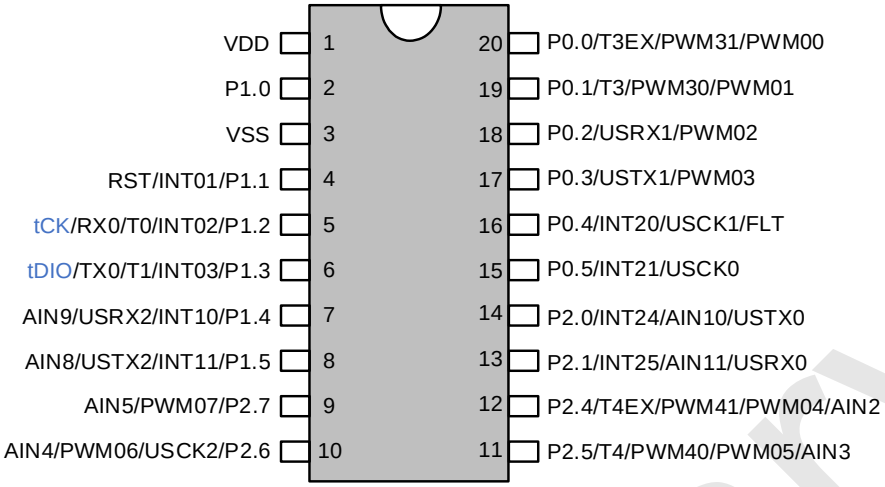



SC95F7673 管脚配置图
适用于 SOP28、TSSOP28 封装

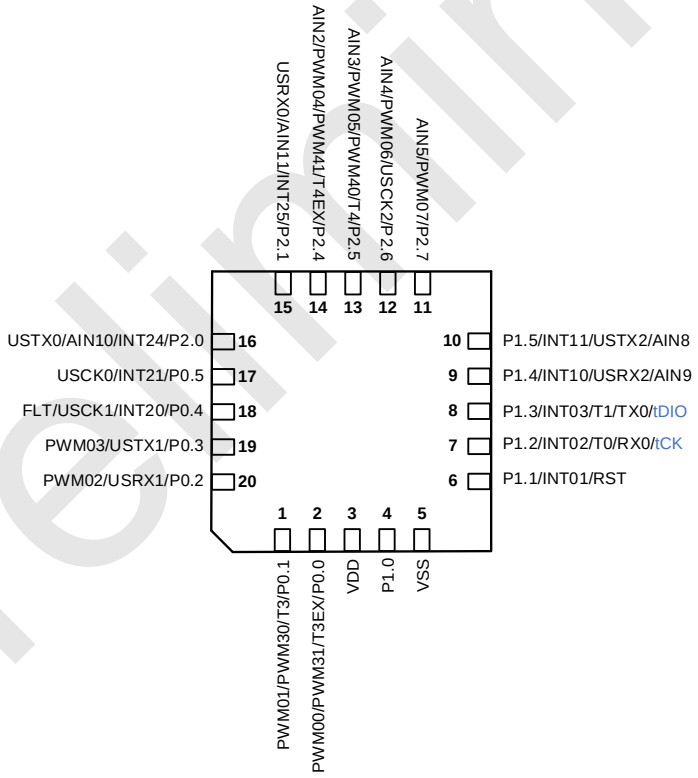



SC95F7673 管脚配置图
适用于 QFN28 封装

Preliminary

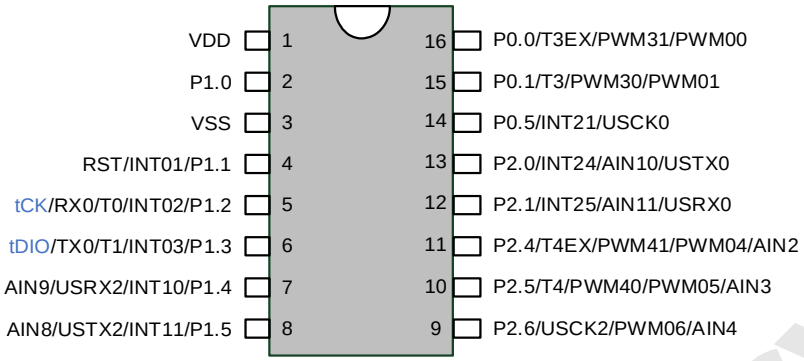



SC95F7672 管脚配置图
适用于 SOP20、TSSOP20 封装




SC95F7672 管脚配置图
适用于 QFN20 封装

Preliminary




SC95F7671 管脚配置图
适用于 SOP16 封装

Preliminary

## 3.2 管脚定义

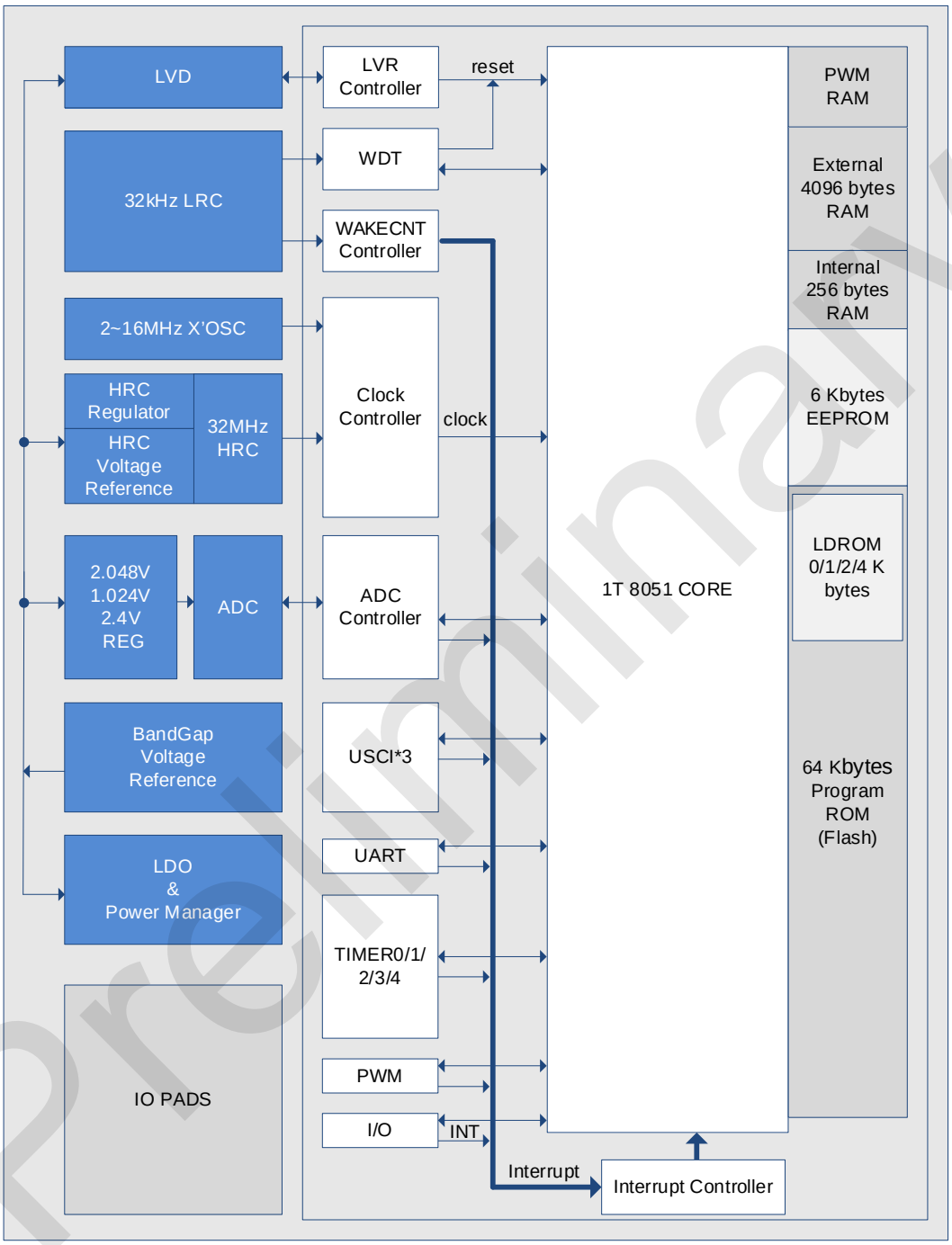
| 管脚编号 | | | | | | 管脚名称 | 类型 | 功能说明 |
|---|---|---|---|---|---|---|---|---|
| LQFP32 /QFN32 | SOP28 /TSSOP 28 | SOP20 /TSSOP 20 | SOP 16 | QFN 28 | QFN 20 | | | |
| 1 | 21 | - | - | 23 | - | P0.7/INT23/T2/PWM20 | I/O | P0.7: GPIO P0.7<br>INT23: 外部中断 2 的输入 3<br>T2: 计数器 2 外部输入<br>PWM20: PWM20 输出口 |
| 2 | 22 | - | - | 24 | - | P0.6/INT22/T2EX/PWM21 | I/O | P0.6: GPIO P0.6<br>INT22: 外部中断 2 的输入 2<br>T2EX: 定时器 2 外部捕获信号输入<br>PWM21: PWM21 输出口 |
| 3 | 23 | 15 | 14 | 25 | 17 | P0.5/INT21/USCK0 | I/O | P0.5: GPIO P0.5<br>INT21: 外部中断 2 的输入 1<br>USCK0: USCI0 的 SCK |
| 4 | 24 | 16 | - | 26 | 18 | P0.4/INT20/USCK1/FLT | I/O | P0.4: GPIO P0.4<br>INT20: 外部中断 2 的输入 0<br>USCK1: USCI1 的 SCK<br>FLT: PWM 故障检测输入脚 |
| 5 | 25 | 17 | - | 27 | 19 | P0.3/USTX1/PWM03 | I/O | P0.3: GPIO P0.3<br>USTX1: USCI1 的 MOSI/SDA/TX<br>PWM03: PWM03 输出口 |
| 6 | 26 | 18 | - | 28 | 20 | P0.2/USRX1/PWM02 | I/O | P0.2: GPIO P0.2<br>USRX1: USCI1 的 MISO/RX<br>PWM02: PWM02 输出口 |
| 7 | 27 | 19 | 15 | 1 | 1 | P0.1/T3/PWM30/PWM01 | I/O | P0.1: GPIO P0.1<br>T3: 计数器 3 外部输入<br>PWM30: PWM30 输出口<br>PWM01: PWM01 输出口 |
| 8 | 28 | 20 | 16 | 2 | 2 | P0.0/T3EX/PWM31/PWM00 | I/O | P0.0: GPIO P0.0<br>T3EX: 定时器 3 外部捕获信号输入<br>PWM31: PWM31 输出口<br>PWM00: PWM00 输出口 |
| 9 | 1 | 1 | 1 | 3 | 3 | VDD | Power | 电源 |
| 10 | 2 | 2 | 2 | 4 | 4 | P1.0 | I/O | P1.0: GPIO P1.0 |
| 11 | 3 | 3 | 3 | 5 | 5 | VSS | Power | 接地 |
| 12 | 4 | - | - | 6 | - | P5.0/OSCI | I/O | P5.0: GPIO P5.0<br>OSCI: 外接晶振输入脚 |
| 13 | 5 | - | - | 7 | - | P5.1/OSCO | I/O | P5.1: GPIO P5.1<br>OSCO: 外接晶振输出脚 |
| 14 | - | - | - | - | - | P5.2 | I/O | P5.2: GPIO P5.2 |
| 15 | - | - | - | - | - | P5.3 | I/O | P5.3: GPIO P5.3 |
| 16 | - | - | - | - | - | P5.4 | I/O | P5.4: GPIO P5.4 |
| 17 | - | - | - | - | - | P5.5 | I/O | P5.5: GPIO P5.5 |
| 18 | 6 | 4 | 4 | 8 | 6 | P1.1/INT01/RST | I/O | P1.1: GPIO P1.1<br>INT01: 外部中断 0 的输入 1RST: 复位管脚 |
| 19 | 7 | 5 | 5 | 9 | 7 | P1.2/INT02/T0/RX0/tCK | I/O | P1.2: GPIO P1.2<br>INT02: 外部中断 0 的输入 2<br>T0: 计数器 0 外部输入<br>RX0: UART0 接收口之一<br>tCK: 烧录和仿真口时钟线 |
| 20 | 8 | 6 | 6 | 10 | 8 | P1.3/INT03/T1/TX0/tDIO | I/O | P1.3: GPIO P1.3<br>INT03: 外部中断 0 的输入 3<br>T1: 计数器 1 外部输入<br>TX0: UART0 发送口之一<br>tDIO: 烧录和仿真口数据线 |
| 21 | 9 | 7 | 7 | 11 | 9 | P1.4/INT10/USRX2/AIN9 | I/O | P1.4: GPIO P1.4<br>INT10: 外部中断 1 的输入 0<br>USRX2: USCI2 的 MISO/RX<br>AIN9: ADC 输入通道 9 |
| 22 | 10 | 8 | 8 | 12 | 10 | P1.5/INT11/USTX2/AIN8 | I/O | P1.5: GPIO P1.5<br>INT11: 外部中断 1 的输入 1<br>USTX2: USCI2 的 MOSI/SDA/TX |

Preliminary

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | AIN8: ADC 输入通道 8 |
| **23** | **11** | **-** | **-** | **13** | **-** | **P1.6/INT12/AIN7** | I/O | P1.6: GPIO P1.6<br>INT12: 外部中断 1 的输入 2<br>AIN7: ADC 输入通道 7 |
| **24** | **12** | **-** | **-** | **14** | **-** | **P1.7/IN13/AIN6** | I/O | P1.7: GPIO P1.7<br>INT13: 外部中断 1 的输入 3<br>AIN6: ADC 输入通道 6 |
| **25** | **13** | **9** | **-** | **15** | **11** | **P2.7/PWM07/AIN5** | I/O | P2.7: GPIO P2.7<br>PWM07: PWM07 输出口<br>AIN5: ADC 输入通道 5 |
| **26** | **14** | **10** | **9** | **16** | **12** | **P2.6/USCK2/PWM06/AIN4** | I/O | P2.6: GPIO P2.6<br>USCK2: USCI2 的 SCK<br>PWM06: PWM06 输出口<br>AIN4: ADC 输入通道 4 |
| **27** | **15** | **11** | **10** | **17** | **13** | **P2.5/T4/PWM40/PWM05/AIN3** | I/O | P2.5: GPIO P2.5<br>T4: 计数器 4 外部输入<br>PWM40: PWM40 输出口<br>PWM05: PWM05 输出口<br>AIN3: ADC 输入通道 3 |
| **28** | **16** | **12** | **11** | **18** | **14** | **P2.4/T4EX/PWM41/PWM04/AIN2** | I/O | P2.4: GPIO P2.4<br>T4EX: 定时器 4 外部捕获信号输入<br>PWM41: PWM41 输出口<br>PWM04: PWM04 输出口<br>AIN2: ADC 输入通道 2 |
| **29** | **17** | **-** | **-** | **19** | **-** | **P2.3/AIN1** | I/O | P2.3: GPIO P2.3<br>AIN1: ADC 输入通道 1 |
| **30** | **18** | **-** | **-** | **20** | **-** | **P2.2/AIN0** | I/O | P2.2: GPIO P2.2<br>AIN0: ADC 输入通道 0 |
| **31** | **19** | **13** | **12** | **21** | **15** | **P2.1/INT25/AIN11/USRX0** | I/O | P2.1: GPIO P2.1<br>INT25: 外部中断 2 的输入 5<br>AIN11: ADC 输入通道 11<br>USRX0: USCI0 的 MISO/RX |
| **32** | **20** | **14** | **13** | **22** | **16** | **P2.0/INT24/AIN10/USTX0** | I/O | P2.0: GPIO P2.0<br>INT24: 外部中断 2 的输入 4<br>AIN10: ADC 输入通道 10<br>USTX0: USCI0 的 MOSI/SDA/TX |

Preliminary

# 4 内部框图

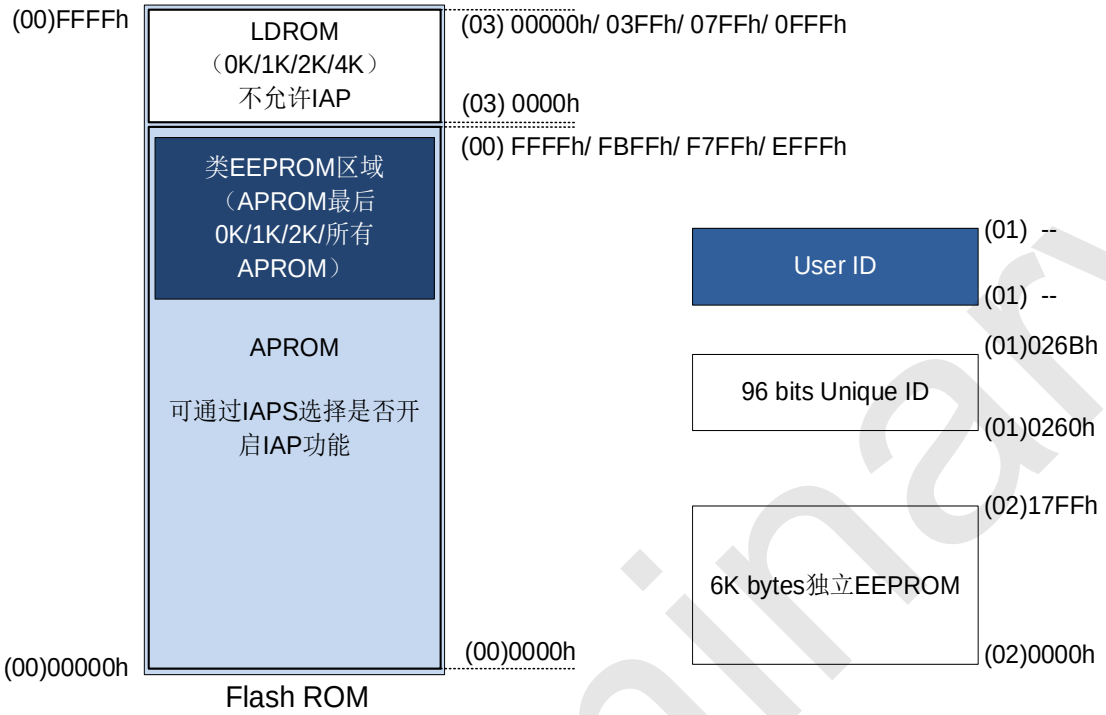



SC95F767X BLOCK DIAGRAM

Preliminary

# 5 Flash ROM 和 SRAM

SC95F767X 的 ROM 分为五个区域：APROM / LDROM / EEPROM / User ID / Unique ID，结构如下图所示：




## 5.1 APROM 和 LDROM

APROM 和 LDROM 是通过 LDSIZE[1:0]将 ROM 划分的硬件上独立的两块，由 IAPADE 寄存器设定的拓展地址“00”和“03”来区分，可通过专用烧写器 SCLINK PRO 来进行编程及擦除：

- APROM 区域拓展地址为“00”，区域大小 60~64K bytes 可选，支持 IAP (In Application Programming)，可通过 Customer Option 项设置其允许 IAP 操作的范围为 0K / 1K / 2K 或 整个 APROM 区域；
- LDROM 区域拓展地址为“03”，区域大小 0~4K bytes 可选，不允许对 LDROM 进行 IAP 操作。

SC95F767X 有 64 Kbytes 的 Flash ROM，地址为(00)00000H~(00)FFFFH，括号里的“00”为拓展地址，由 IAPADE 寄存器设定。可通过赛元提供的专用 ICP 烧写器(SC LINK PRO)来进行编程及擦除。此 64 Kbytes Flash ROM 特性如下：

- 分为 128 个扇区（sector），每个 sector 为 512 bytes；
- 可反复写入 10 万次；
- 25℃环境下数据可保存 100 年以上；
- ICP 模式下支持查空（BLANK）、编程（PROGRAM）、校验（VERIFY）、擦除（ERASE）和读取（READ）功能，其中 READ 功能仅对未开启安全加密功能的 IC 有效；
- 安全加密：可选择是否开启 APROM（即 64 Kbytes Flash ROM）和 LDROM 安全加密功能；
- 支持 IAP (In Application Programming)。

APROM 和 LDROM 共有 128 个扇区（sector），每个 sector 为 512 bytes，可反复写入 10 万次，25℃环境下数据可保存 100 年以上。

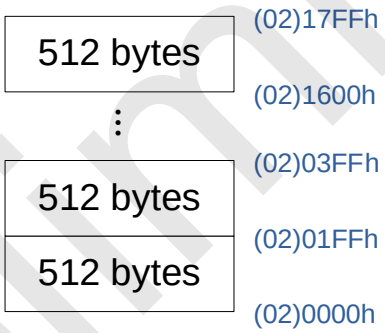
Preliminary

SC95F767X APROM Sector 分区示意

## 5.2 6K bytes 独立 EEPROM

6K bytes 独立 EEPROM 区域地址为（02）000H~17FFH，其中“（02）”为拓展地址，由 IAPADE 寄存器设定。独立 EEPROM 可反复写入 10 万次，常温下数据保存时间为 100 年以上。独立 EEPROM 支持查空、编程、校验、擦除和读取功能。

EEPROM 共有 12 个扇区（sector），每个 sector 为 512 bytes：




SC95F767X EEPROM Sector 分区示意

注意：EEPROM 擦写次数为 10 万次，用户擦写不要超过 EEPROM 的额定烧写次数，否则会出现异常！

## 5.3 96 bits Unique ID 区域

SC95F767X 提供了一个独立的 Unique ID 区域，可出厂前会预烧一个 96 bits 的唯一码，用以确保该芯片的唯一性。用户获得序列号的唯一方式是通过 IAP 指令读取相对地址(01)0260H~(01)026BH 来获取。地址(01)0260H~(01)026BH 括号里的“01”表示拓展地址，由 IAPADE 寄存器设定。具体操作方法如下：

**IAPADE (F4H) IAP 写入扩展地址寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPADER[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | IAPADER[7:0] | IAP 扩展地址： |



Preliminary

| | | |
|---|---|---|
| | | 0x00：MOVC 和 IAP 烧写都针对 APROM 进行<br>0x01：针对 **Unique ID** 区域进行读操作，不可进行写操作<br>0x02：MOVC 和 IAP 烧写都针对独立 EEPROM 进行<br>0x03：仅在 LDROM 程序操作时生效。此时允许运行在 LDROM 区域的程序对 LDROM 程序区进行 MOVC 操作。（**注意：只可以 MOVC，不可以 IAP，此项仅在 LDROM 程序操作时生效，APROM 程序操作该项无效**）<br>其它：保留 |

### 5.3.1 Unique ID 读取 C 语言例程

```
#include ”intrins.h”

unsigned char UniqueID [12];//存放 UniqueID

unsigned char code * POINT =0x0260;

unsigned char i;


EA = 0;                 //关闭总中断

IAPADE = 0X01;                  //拓展地址 0x01，选择 Unique ID 区域

for(i=0;i<12;i++)

{

    UniqueID [i]= *( POINT+i);      //读取 UniqueID 的值

}

IAPADE = 0X00;                  //拓展地址 0x00，返回 Code 区域

EA = 1;                 //开启总中断
```

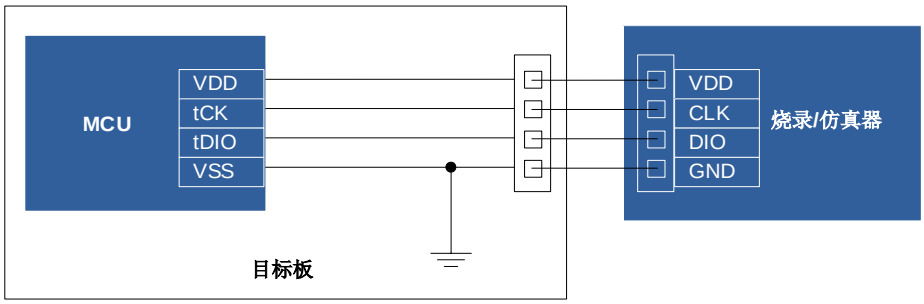
## 5.4 User ID 区域

User ID 区域，扩展地址为（01），出厂时写入用户定制 ID，用户可对其进行读操作，但禁止对 User ID 区域进行写操作。

## 5.5 编程

SC95F767X 的 APROM 和 LDROM 可通过 tDIO、tCK、VDD、VSS 进行编程，具体连接关系如下：

Preliminary




ICP 模式 编程连接示意图

tDIO、tCK 是 2 线 JTAG 烧写和仿真的信号线，用户在烧录时可通过 Customer Option 项配置这两端口的模式：

### 5.5.1 JTAG 专用模式

tDIO、tCK 为烧写仿真专用口，与之复用的其它功能不可用。此模式一般用于在线调试阶段，方便用户仿真调试；JTAG 专用模式生效后，芯片无需重新上下电即可直接进入烧录或仿真模式。

### 5.5.2 常规模式（JTAG 专用口无效）

JTAG 功能不可用，与之复用的其它功能可正常使用。此模式可防止烧录口占用 MCU 管脚，方便用户最大化利用 MCU 资源。

**注意：当 JTAG 专用口无效的配置设定成功后，芯片必须彻底下电再重新上电后才能进入烧录或仿真模式，这样就会影响到带电模式下的烧录和仿真。赛元建议用户在量产烧录时选择 JTAG 专用口无效的配置，在研发调试阶段选择 JTAG 模式。**

相关 Customer Option 寄存器如下：

**OP_CTM1 (C2H@FFH) Code Option 寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | VREFS[1:0] | | OP_BL | DISJTG | IAPS[1:0] | | LDSIZE[1:0] | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | n | n | n | n | n | n | n | n |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 4 | **DISJTG** | IO/JTAG 口切换控制<br>0：JTAG 模式使能，P1.1、P1.3 只能作为 tCK/tDIO 使用。研发调试阶段推荐设置。<br>1：常规模式（Normal），JTAG 功能无效。量产烧录阶段推荐设置。 |

## 5.6 IAP (Application Programming) 操作

SC95F767X 的 APROM 中的 IAP 区域（0K、1K、2K 或所有 APROM 范围可选）及 6K bytes EEPROM 内都可进行 IAP（Application Programming）操作，用户可以通过 IAP 操作实现远程程序更新，也可以通过 IAP 读操

Preliminary

作获取 Unique ID 区域或 User ID 区域信息。进行 IAP 写数据操作前，用户必须对目标地址所属的 Sector 进行扇区擦除操作，一个 Sector 为 512 bytes，Flash ROM 从(00)0000H~(00)FFFFH 共分为 128 个 Sector，括号里的“00”为拓展地址，由 IAPADE 寄存器设定：

SC95F767X 64 Kbytes Flash ROM Sector 分区示意

**注意：**

1. **IAP 的擦/写过程中，CPU 保持程序计数器，IAP 擦/写完成后，程序计数器才继续执行之后的指令；**

2. **APROM 区域内的 IAP 操作有一定的风险，需要用户在软件中做相应的安全处理措施，如果操作不当可能会造成用户程序被改写！除非用户必需此功能(比如用于远程程序更新等)，不建议用户使用；**

3. **EEPROM 擦写次数为 10 万次，用户擦写不要超过 EEPROM 的额定烧写次数，否则会出现异常！**

用户可以通过 Customer Option 在编程时选择 APROM 的 IAP 区域范围，也可以通过 IAPS 控制位在程序中设定 APROM 的 IAP 区域，相关寄存器如下：

**OP_CTM1 (C2H@FFH) Code Option 寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | VREFS[1:0] | | OP_BL | DISJTG | IAPS[1:0] | | LDSIZE[1:0] | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | n | n | n | n | n | n | n | n |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3~2 | **IAPS[1:0]** | **IAP 空间范围选择**<br>00：全部 Flash ROM 均不允许 IAP 操作<br>01：最后 1K Flash ROM 允许 IAP 操作<br>10：最后 2K Flash ROM 允许 IAP 操作<br>11：全部 Flash ROM 允许 IAP 操作<br>**注意：**<br>**1. BootLoader 模式下以上设置项无效，BootLoader 程序可对整个 APROM 区域进行 IAP 操作。**<br>**2. LDROM 在任何情况下都不允许 IAP 操作** |

Preliminary

### 5.6.1 IAP 操作相关寄存器

**IAP 操作相关寄存器说明：**

| 符号 | 地址 | 说明 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Reset 值 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| IAPKEY | F1H | 数据保护寄存器 | IAPKEY[7:0] | | | | | | | | 00000000b |
| IAPADL | F2H | IAP 写入地址低位寄存器 | IAPADR[7:0] | | | | | | | | 00000000b |
| IAPADH | F3H | IAP 写入地址高位寄存器 | IAPADR[15:8] | | | | | | | | 00000000b |
| IAPADE | F4H | IAP 写入扩展地址寄存器 | IAPADER[7:0] | | | | | | | | 00000000b |
| IAPDAT | F5H | IAP 数据寄存器 | IAPDAT[7:0] | | | | | | | | 00000000b |
| IAPCTL | F6H | IAP 控制寄存器 | BTLD | - | SERASE | PRG | - | - | CMD[1:0] | | 0x00xx00b |

**IAPKEY (F1H) 数据保护寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPKEY[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **IAPKEY[7:0]** | 打开 IAP 功能及操作时限设置<br>写入一个大于等于 0x40 的值 n，代表：<br>1. 打开 IAP 功能；<br>2. n 个系统时钟后如果接收不到 IAP 写入命令，则 IAP 功能被重新关闭。 |

**IAPADL (F2H) IAP 写入地址低位寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPADR[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **IAPADR[7:0]** | IAP 写入地址的低 8 位 |

**IAPADH (F3H) IAP 写入地址高位寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPADR[15:8] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **IAPADR[15:8]** | IAP 写入地址的高 8 位 |

Preliminary

**IAPADE (F4H) IAP 写入扩展地址寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPADER[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **IAPADER[7:0]** | IAP 扩展地址：<br>0x00：MOVC 和 IAP 烧写都针对 Code 进行<br>0x01：针对 **Unique ID** 区域进行读操作，不可进行写操作<br>0x02：MOVC 和 IAP 烧写都针对独立 EEPROM 进行<br>0x03：**仅在 LDROM 程序操作时生效。此时允许运行在 LDROM 区域的程序对 LDROM 程序区进行 MOVC 操作。注意：对 LDROM 的操作权限仅开放了 MOVC 操作，禁止对 LDROM 进行 IAP 操作，否则将导致不可预测的异常！**<br>其它：保留 |

**IAPDAT (F5H) IAP 数据寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPDAT[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 读写 | 说明 |
|---|---|---|---|
| 7~0 | **IAPDAT[7:0]** | 读/写 | IAP 写入的数据 |

**IAPCTL (F6H) IAP 控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | BTLD | - | SERASE | PRG | - | - | CMD[1:0] | |
| 读/写 | 读/写 | - | 读/写 | 读/写 | - | - | 读/写 | 读/写 |
| 上电初始值 | 0 | x | 0 | 0 | x | x | 0 | 0 |

### 5.6.2 IAP 操作流程

SC95F767X 的 IAP 操作流程请参考《SC95FXX7X IAP 操作库资料包》

### 5.6.3 IAP 操作 C 语言例程

以下例程共用的头文件如下：

```
#include ”intrins.h”

unsigned int IAP_Add;

unsigned char IAP_Data;

unsigned char code * POINT =0x0000;
```

Preliminary

**IAP 操作：读数据**

```
EA = 0;                    //关闭总中断

IAPADE = 0X00;             //拓展地址为 0x00，选择 Flash ROM

IAP_Data = *( POINT+IAP_Add);   //读取 IAP_Add 的值到 IAP_Data

EA = 1;                    //开启总中断
```

**IAP 操作：擦写**

用户如需进行 IAP 擦写操作，请参考《SC95FXX7X IAP 操作库资料包》

## 5.7 BootLoader 功能

LDROM 用来存放 IC 的 BootLoader 引导代码（boot code）。LDROM 在 ICP 模式下支持查空（BLANK）、编程（PROGRAM）、校验（VERIFY）、擦除（ERASE）和读取（READ）功能，用户也可以通过烧写器将自己开发的引导代码下载到 LDROM 中。

用户可以通过 LDROM 实现 ISP（In System Programing）功能：ISP 执行时，IC 运行的是 LDROM 区的引导代码，引导代码执行时会通过串口接收新的程序代码，再将接收到的代码通过 IAP 命令编程到用户代码区域。整个过程不需要将芯片从系统板上拆下来，也不需要烧写器。

LDROM 地址范围有四种：

- (03)0000H~(03)0000H（无 LDROM）
- (03)0000H~(03)03FFH（1K）
- (03)0000H~(03)07FFH（2K）
- (03)0000H~(03)0FFFH（4K）

其中：上述地址括号里的“03”表示拓展地址，由 LDSIZE[1:0]设定。

### 5.7.1 BootLoader 模式操作相关寄存器

**OP_CTM1 (C2H@FFH) Code Option 寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | VREFS[1:0] | | OP_BL | DISJTG | IAPS[1:0] | | LDSIZE[1:0] | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | n | n | n | n | n | n | n | n |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 5 | **OP_BL** | 程序运行区域控制位<br>0：芯片复位后进入 APROM<br>1：芯片复位后进入 LDROM；<br><br>① APROM 的 MOVC 及 IAP 操作限制如下：<br>操作 / 能否操作<br>对 LDROM MOVC / x<br>对 APROM MOVC / √ |



Preliminary

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>对 LDROM IAP</td><td>×</td></tr>
<tr><td>对 APROM IAP</td><td>√</td></tr>
</table>
② LDROM 的 MOVC 及 IAP 操作限制如下：
<table>
<tr><th>操作</th><th>能否操作</th></tr>
<tr><td>对 LDROM MOVC</td><td>√</td></tr>
<tr><td>对 APROM MOVC</td><td>√</td></tr>
<tr><td>对 LDROM IAP</td><td>×</td></tr>
<tr><td>对整个 APROM IAP，不受 IAPRANGE 限制</td><td>√</td></tr>
</table>
</td></tr>
<tr><td>1~0</td><td>LDSIZE[1:0]</td><td>LDROM 空间范围选择<br>00: 无 LDROM，APROM 地址为 00000H~FFFFH<br>01: Flash ROM 最后 1K APROM 区域为 LDROM，APROM 地址为 00000H~FBFFH<br>10: Flash ROM 最后 2K APROM 区域为 LDROM，APROM 地址为 00000H~F7FFH<br>11: Flash ROM 最后 4K APROM 区域为 LDROM，APROM 地址为 00000H~EFFFH<br>注意：LDROM 在任何情况下都不允许 IAP 操作</td></tr>
</table>

**IAPKEY (F1H) 数据保护寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPKEY[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **IAPKEY[7:0]** | 打开 IAP 功能及操作时限设置<br>写入一个大于等于 0x40 的值 n，代表：<br>1. 打开 IAP 功能；<br>2. n 个系统时钟后如果接收不到 IAP 写入命令，则 IAP 功能被重新关闭。 |

**IAPADL (F2H) IAP 写入地址低位寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPADR[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **IAPADR[7:0]** | IAP 写入地址的低 8 位 |

**IAPADH (F3H) IAP 写入地址高位寄存器(读/写)**

Preliminary

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPADR[15:8] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **IAPADR[15:8]** | IAP 写入地址的高 8 位 |

**IAPADE (F4H) IAP 写入扩展地址寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPADER[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **IAPADER[7:0]** | IAP 扩展地址：<br>0x00：MOVC 和 IAP 烧写都针对 APROM 进行<br>0x01：针对 Unique ID 区域进行读操作，不可进行写操作<br>0x02：MOVC 和 IAP 烧写都针对独立 EEPROM 进行<br>0x03：仅在 LDROM 程序操作时生效。此时允许运行在 LDROM 区域的程序对 LDROM 程序区进行 MOVC 操作。**注意：对 LDROM 的操作权限仅开放了 MOVC 操作，禁止对 LDROM 进行 IAP 操作，否则将导致不可预测的异常！**<br>其它：保留 |

**IAPDAT (F5H) IAP 数据寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IAPDAT[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **IAPDAT[7:0]** | IAP 写入的数据 |

**IAPCTL (F6H) IAP 控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | BTLD | - | SERASE | PRG | - | - | CMD[1:0] | |
| 读/写 | 读/写 | - | 读/写 | 读/写 | - | - | 读/写 | 读/写 |
| 上电初始值 | 0 | x | 0 | 0 | x | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **BTLD** | BootLoader 控制位<br>0：软件复位后程序从主程序区（main program）开始运行； |

Preliminary

| | | |
|---|---|---|
| | | 1：软件复位后程序从 BootLoader 区开始运行 |

**PCON (87h) 电源管理控制寄存器(只写、*不可读 *)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SMOD | - | - | - | RST | - | STOP | IDL |
| 读/写 | 只写 | - | - | - | 只写 | - | 只写 | 只写 |
| 上电初始值 | 0 | x | x | x | n | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3 | **RST** | 软件 reset 控制位：<br>写状态：<br>0：程序正常运行；<br>1：此位被写“1”后 CPU 立刻 reset |

**Bootloader 操作注意事项：**

1. **对 Code 区进行写数据操作前，用户必须对目标地址所属的 Sector 进行扇区擦除操作；**
2. **具体操作方法参考赛元提供的说明文档《赛元硬件 Boot loader 功能实现应用指南》。**

## 5.8 安全加密

用户可通过烧录上位机界面的“加密”设置项选择是否开启 SC95F767X 的 ROM 安全加密功能：

1. 关闭安全加密功能后，用户可以通过烧写器读取 APROM（即 64 Kbytes Flash ROM）和 LDROM 的数据，方便开发调试；
2. 开启安全加密功能，APROM 及 LDROM 的数据将无法被外界读出。当用户通过烧写器对一颗已开启了加密功能的 SC95F767X 执行烧录改写操作时，无论改写的目标是 APROM 或 LDROM，烧写器均会先会强制擦除 APROM 和 LDROM，再执行写入操作。推荐用户在量产烧录时选择开启安全加密功能；
3. 解除安全加密的唯一方式是关闭安全加密功能，并执行编程操作；
4. 安全加密不影响 IAP 功能；

具体操作方法见烧录工具使用手册，安全加密及读取章节。

## 5.9 Customer Option 区域（用户烧写设置）

| 符号 | OPINX 地址 | 说明 | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|---|---|---|---|---|---|---|---|---|---|---|
| OP_CTM0 | C1H@FFH | Code Option 寄存器 0 | ENWDT | ENXTL | SCLK[1:0] | | DISRST | DISLVR | LVRS[1:0] | |
| OP_CTM1 | C2H@FFH | Code Option 寄存器 1 | VREFS[1:0] | | OP_BL | DISJTG | IAPS[1:0] | | LDSIZE[1:0] | |
| OP_HRCR | 83H@FFH | 系统时钟改变寄存器 | OP_HRCR[7:0] | | | | | | | |

| IFB 地址 | 符号 | 读/写 | 说明 |
|---|---|---|---|
| OP_CTM0[7] | ENWDT | 读/写 | WDT 开关<br>0：WDT 无效<br>1：WDT 有效（但 IC 在执行 IAP 过程中 WDT 停止计数） |
| OP_CTM0[6] | ENXTL | 读/写 | 外部晶振选择开关 |



Preliminary

<table>
<tr><td></td><td></td><td></td><td>0：外部晶振关闭，P5.0、P5.1 有效；<br>1：外部晶振打开，P5.0、P5.1 无效。</td></tr>
<tr><td>OP_CTM0[5~4]</td><td>SCLKS[1:0]</td><td>读/写</td><td>系统时钟频率选择：<br>00：系统时钟频率为高频振荡器频率除以 1；<br>01：系统时钟频率为高频振荡器频率除以 2；<br>10：系统时钟频率为高频振荡器频率除以 4；<br>11：系统时钟频率为高频振荡器频率除以 8。</td></tr>
<tr><td>OP_CTM0[3]</td><td>DISRST</td><td>只读</td><td>IO/RST 复位切换控制<br>0 ：P1.1 当复位脚使用<br>1 ：P1.1 当正常的 I/O 管脚使用</td></tr>
<tr><td>OP_CTM0[2]</td><td>DISLVR</td><td>读/写</td><td>LVR 开关<br>0：LVR 有效<br>1：LVR 无效</td></tr>
<tr><td>OP_CTM0[1~0]</td><td>LVRS [1:0]</td><td>读/写</td><td>LVR 电压选择控制<br>11：4.3 V 复位<br>10：3.7V 复位<br>01：2.9V 复位<br>00：1.9V 复位</td></tr>
<tr><td>OP_CTM1[7~6]</td><td>VREFS[1:0]</td><td>读/写</td><td>参考电压选择 (初始值从 Code Option 调入，用户可修改设置)<br>00：设定 ADC 的 VREF 为 VDD；<br>01：设定 ADC 的 VREF 为内部准确的 2.048V；<br>10：设定 ADC 的 VREF 为内部准确的 1.024V；<br>11：设定 ADC 的 VREF 为内部准确的 2.4V。</td></tr>
<tr><td>OP_CTM1[5]</td><td>OP_BL</td><td>读/写</td><td>程序运行区域控制位<br>0：芯片复位后进入 APROM<br>1：芯片复位后进入 LDROM；<br><br>APROM 的 MOVC 及 IAP 操作限制如下：
<table>
<tr><th>操作</th><th>能否操作</th></tr>
<tr><td>对 LDROM MOVC</td><td>x</td></tr>
<tr><td>对 APROM MOVC</td><td>√</td></tr>
<tr><td>对 LDROM IAP</td><td>×</td></tr>
<tr><td>对 APROM IAP</td><td>√</td></tr>
</table>
LDROM 的 MOVC 及 IAP 操作限制如下：
<table>
<tr><th>操作</th><th>能否操作</th></tr>
<tr><td>对 LDROM MOVC</td><td>√</td></tr>
<tr><td>对 APROM MOVC</td><td>√</td></tr>
<tr><td>对 LDROM IAP</td><td>×</td></tr>
<tr><td>对整个 APROM IAP，不受 IAPRANGE 限制</td><td>√</td></tr>
</table></td></tr>
<tr><td>OP_CTM1[4]</td><td>DISJTG</td><td>读/写</td><td>IO/JTAG 口切换控制<br>0 ：JTAG 模式使能，P1.2、P1.3 只能作为 tCK/tDIO 使用<br>1 ：常规模式（Normal），JTAG 功能无效</td></tr>
<tr><td>OP_CTM1[3~2]</td><td>IAPS[1:0]</td><td>读/写</td><td>IAP 空间范围选择<br>00: Code 区域禁止 IAP 操作<br>01: 最后 1K Code 区域允许 IAP 操作<br>10: 最后 2K Code 区域允许 IAP 操作<br>11: 全部 Code 区域允许 IAP 操作<br>注意：<br>1. BootLoader 模式下以上设置项无效，BootLoader 程序可对整个 APROM 区域进行 IAP 操作。<br>2. LDROM 在任何情况下都不允许 IAP 操作</td></tr>
<tr><td>OP_CTM1[1:0]</td><td>LDSIZE[1:0]</td><td>只读</td><td>LDROM 空间范围选择<br>00: 无 LDROM，APROM 地址为 0000H~FFFFH<br>01: Flash ROM 最后 1K APROM 区域为 LDROM，APROM 地址为 0000H~FBFFH<br>10: Flash ROM 最后 2K APROM 区域为 LDROM，APROM 地址为 0000H~F7FFH</td></tr>
</table>

Preliminary

<table>
<tr><td></td><td></td><td></td><td>11: Flash ROM 最后 4K APROM 区域为 LDROM，APROM 地址为 0000H~EFFFH<br>**注意：LDROM 在任何情况下都不允许 IAP 操作**</td></tr>
<tr><td>OP_HRCR</td><td>OP_HRCR[7:0]</td><td>读/写</td><td>**HRC 频率改变寄存器**<br><br>用户可通过修改此寄存器的值实现高频振荡器频率 $f_{HRC}$ 的改变，进而改变 IC 的系统时钟频率 $f_{SYS}$：<br><br>1. OP_HRCR[7:0]上电后的初始值 OP_HRCR[s]是一个固定值，以确保 $f_{HRC}$ 为 32MHz，每颗 IC 的 OP_HRCR[s]都可能会有差异。<br><br>2. 初始值为 OP_HRCR[s] 时 IC 的系统时钟频率 $f_{SYS}$ 可通过 Option 项设置为准确的 32/16/8/4MHz，OP_HRCR [7:0]每改变 1 则 $f_{SYS}$ 频率改变约 0.18%。<br><br>OP_HRCR [7:0]和 HRC 输出频率的关系如下：
<table>
<tr><td>OP_HRCR [7:0]值</td><td>$f_{SYS}$ 实际输出频率(32M 为例)</td></tr>
<tr><td>OP_HRCR [s]-n</td><td>32000*(1-0.18%*n)kHz</td></tr>
<tr><td>...</td><td>....</td></tr>
<tr><td>OP_HRCR [s]-2</td><td>32000*(1-0.18%*2) = 31 884.8kHz</td></tr>
<tr><td>OP_HRCR [s]-1</td><td>32000*(1-0.18%*1) = 31 942.4kHz</td></tr>
<tr><td>OP_HRCR [s]</td><td>32000kHz</td></tr>
<tr><td>OP_HRCR [s]+1</td><td>32000*(1+0.18%*1) = 32 057.6kHz</td></tr>
<tr><td>OP_HRCR [s]+2</td><td>32000*(1+0.18%*2) = 32 115.2kHz</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>OP_HRCR [s]+n</td><td>32000*(1+0.18%*n)kHz</td></tr>
</table>
注意：<br>1. IC 每次上电后 OP_HRCR[7:0]的值都是 HRC 工作在最接近 32/16/8/4MHz 的 HRC；用户可借助 EEPROM 在每次上电后修正 HRC 的值以让 HRC 工作在用户需要的频率；<br>2. 为保证 IC 工作可靠，IC 最高工作频率请勿超过 32MHz 的 10%，即 35.2MHz；<br>3. 请用户确认 HRC 频率的改变不会影响其它功能。</td></tr>
</table>

### 5.9.1 Option 相关 SFR 操作说明

Option 相关 SFR 的读写操作由 OPINX 和 OPREG 两个寄存器进行控制，各 Option SFR 的具体位置由 OPINX 确定，各 Option SFR 的写入值由 OPREG 确定：

| 符号 | 地址 | 说明 | | 上电初始值 |
|---|---|---|---|---|
| OPINX | FEH | Option 指针 | OPINX[7:0] | 00000000b |
| OPREG | FFH | Option 寄存器 | OPREG[7:0] | nnnnnnnnb |

操作 Option 相关 SFR 时 OPINX 寄存器存放相关 OPTION 寄存器的地址，OPREG 寄存器存放对应的值。例如：要将 OPINX_HRCR 配置为 0x01，具体操作方法如下：

C 语言例程：

```
OPINX = 0x83;        //将 OPINX_HRCR 的地址写入 OPINX 寄存器
OPREG = 0x01;        //对 OPREG 寄存器写入 0x01（待写入 OPINX_HRCR 寄存器的值）
```

汇编例程：

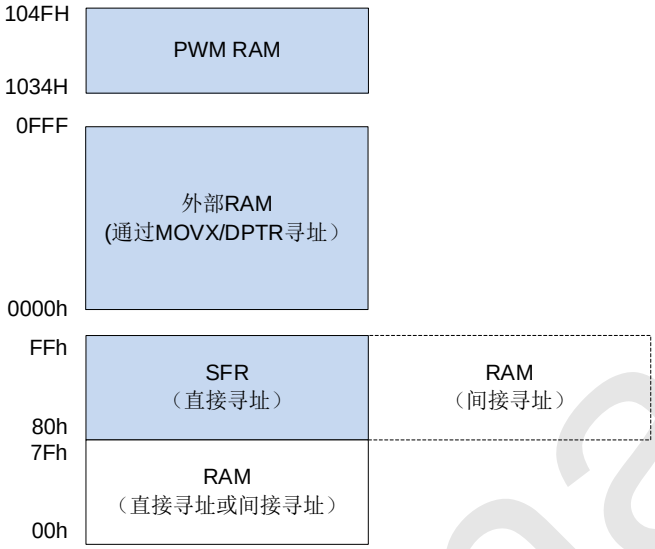
```
MOV OPINX,#83H       ; 将 OPINX_HRCR 的地址写入 OPINX 寄存器
MOV OPREG,#01H       ; 对 OPREG 寄存器写入 0x01（待写入 OPINX_HRCR 寄存器的值）
```

**注意：禁止向 OPINX 寄存器写入 Customer Option 区域 SFR 地址之外的数值！否则会造成系统运行异常！**

Preliminary

## 5.10 SRAM

SC95F767X 的 SRAM 结构如下：



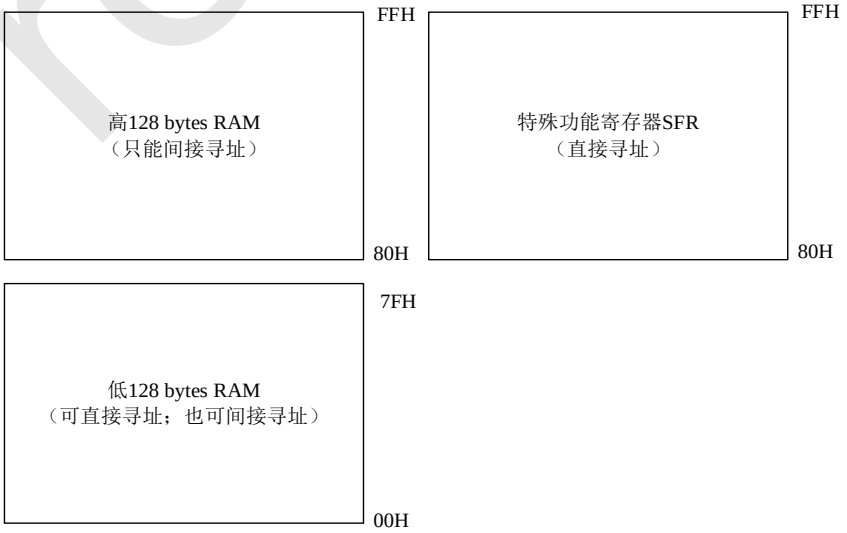

SC95F767X 单片机内部集成了 4.25 Kbytes 的 SRAM，分为内部 256 bytes RAM 和外部 4096bytes RAM。内部 RAM 的地址范围为 00H~FFH，其中高 128 bytes（地址 80H~FFH）只能间接寻址，低 128 bytes（地址 00H~7FH）可直接寻址也可间接寻址。

特殊功能寄存器 SFR 的地址也是 80H~FFH。但 SFR 同内部高 128 bytes SRAM 的区别是：SFR 寄存器是直接寻址，而内部高 128 bytes SRAM 只能是间接寻址。

外部 RAM 的地址为 0000H~0FFFH，但需通过 MOVX 指令来寻址。

### 5.10.1 内部 256 bytes SRAM

内部低 128 bytes SRAM 区可分为三部分：①工作寄存器组 0~3，地址 00H~1FH，程序状态字寄存器 PSW 中的 RS0、RS1 组合决定了当前使用的工作寄存器，使用工作寄存器组 0~3 可加快运算的速度；②位寻址区 20H~2FH，此区域用户可以用作普通 RAM 也可用作按位寻址 RAM；按位寻址时，位的地址为 00H~7FH，（此地址按位编地址，不同于通用 SRAM 按字节编地址），程序中可由指令区分；③用户 RAM 和堆栈区，SC95F767X 复位过后，8 位的堆栈指针指向堆栈区，用户一般会在初始化程序时设置初值，建议设置在 E0H~FFH 的单元区间。

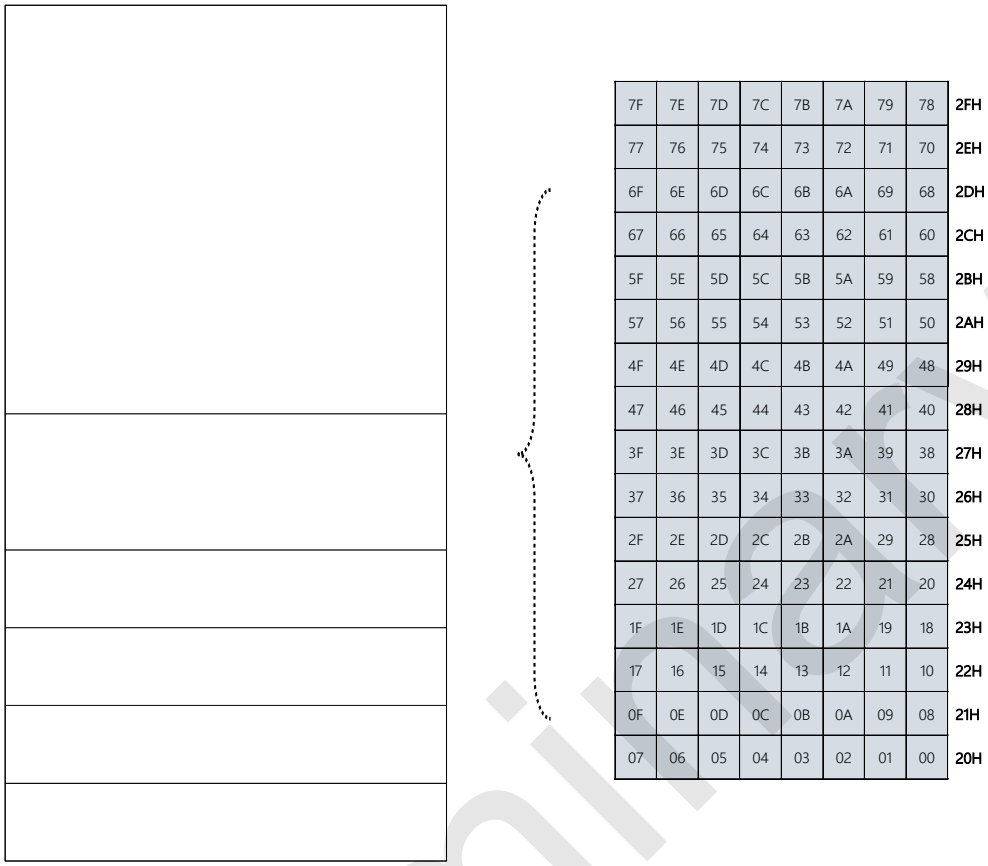



内部 256 bytes RAM 结构图

内部低 128 bytes RAM 结构如下：



Preliminary




SRAM 结构图

### 5.10.2 外部 4096 bytes SRAM

可通过 MOVX @DPTR，A 来访问外部 4096 字节 RAM；也可以使用 MOVX A, @Ri 或 MOVX @Ri, A 配合 EXADH 寄存器来访问外部 4096 字节 RAM：EXADH 寄存器存放外部 SRAM 的高位地址， Ri 寄存器存放外部 SRAM 的低 8 位地址。

**EXADH (F7H) 外部 SRAM 操作地址高位(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | EXADH [3:0] | | | |
| 上电初始值 | x | x | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3~0 | **EXADH [3:0]** | 外部 SRAM 操作地址的高位 |
| 7~4 | **-** | 保留 |

### 5.10.3 外部 PWM SRAM

PWM 占空比调节寄存器占用 1034H~104FH，**可读写。**

Preliminary

# 6 特殊功能寄存器(SFR)

## 6.1 SFR 映像

SC95F767X 系列有一些特殊功能寄存器，我们称为 SFR。这些 SFR 寄存器的地址位于 80H~FFH，有些可以位寻址，有些不能位寻址。能够进行位寻址操作的寄存器的地址末位数都是“0”或“8”，这些寄存器在需要改变单个位的数值时非常方便。所有的 SFR 特殊功能寄存器都必须使用直接寻址方式寻址。

SC95F767X 的特殊功能寄存器名称及地址如下表：

| | 0/8 | 1/9 | 2/A | 3/B | 4/C | 5/D | 6/E | 7/F |
|---|---|---|---|---|---|---|---|---|
| F8h | - | - | - | BTMCON | CRCINX | CRCREG | OPINX | OPREG |
| F0h | B | IAPKEY | IAPADL | IAPADH | IAPADE | IAPDAT | IAPCTL | EXADH |
| E8h | - | EXA0 | EXA1 | EXA2 | EXA3 | EXBL | EXBH | OPERCON |
| E0h | ACC | - | - | - | - | - | - | - |
| D8h | P5 | P5CON | P5PH | - | - | - | - | - |
| D0h | PSW | PWMCFG | PWMCON0 | PWMCON1 | PWMPDL | PWMPDH | PWMDFR | PWMFLT |
| C8h | TXCON | TXMOD | RCAPXL | RCAPXH | TLX | THX | TXINX | WDTCON |
| C0h | - | - | - | - | US2CON0 | US2CON1 | US2CON2 | US2CON3 |
| B8h | IP | IP1 | IP2 | INT0R | INT1F | INT1R | INT2F | INT2R |
| B0h | - | - | - | - | INT0F | ADCCFG2 | - | - |
| A8h | IE | IE1 | IE2 | ADCCFG0 | ADCCFG1 | ADCCON | ADCVL | ADCVH |
| A0h | P2 | P2CON | P2PH | - | US1CON0 | US1CON1 | US1CON2 | US1CON3 |
| 98h | SCON | SBUF | P0CON | P0PH | - | US0CON1 | US0CON2 | US0CON3 |
| 90h | P1 | P1CON | P1PH | - | - | US0CON0 | IOHCON0 | IOHCON1 |
| 88h | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | TMCON | OTCON |
| 80h | P0 | SP | DPL | DPH | DPL1 | DPH1 | DPS | PCON |
| | 可位寻址 | 不可位寻址 | | | | | | |

说明：

1. SFR 寄存器中空的部分代表没有此寄存器 RAM，不建议用户使用。
2. SFR 中的 F1H~FFH 为系统配置使用的特殊功能寄存器，用户使用可能会导致系统异常，用户在初始化系统时，不能对这些寄存器进行清零或其它操作。

Preliminary

## 6.2 SFR 说明

### 6.2.1 特殊功能寄存器

特殊功能寄存器 SFR 的具体解释说明如下：

| 符号 | 地址 | 说明 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 上电初始值 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| P0 | 80H | P0 口数据寄存器 | P07 | P06 | P05 | P04 | P03 | P02 | P01 | P00 | 00000000b |
| SP | 81H | 堆栈指针 | SP[7:0] | | | | | | | | 00000111b |
| DPL | 82H | DPTR 数据指针低位 | DPL[7:0] | | | | | | | | 00000000b |
| DPH | 83H | DPTR 数据指针高位 | DPH[7:0] | | | | | | | | 00000000b |
| DPL1 | 84H | DPTR1 数据指针低位 | DPL1[7:0] | | | | | | | | 00000000b |
| DPH1 | 85H | DPTR1 数据指针高位 | DPH1[7:0] | | | | | | | | 00000000b |
| DPS | 86H | DPTR 选择寄存器 | ID1 | ID0 | TSL | AU1 | AU0 | - | - | SEL | 00000xx0b |
| PCON | 87H | 电源管理控制寄存器 | SMOD | - | - | - | RST | - | STOP | IDL | 0xxx0x00b |
| TCON | 88H | 定时器控制寄存器 | TF1 | TR1 | TF0 | TR0 | IE1 | - | IE0 | - | 00000x0xb |
| TMOD | 89H | 定时器工作模式寄存器 | - | C/T1 | M11 | M01 | - | C/T0 | M10 | M00 | x000x000b |
| TL0 | 8AH | 定时器 0 低 8 位 | TL0[7:0] | | | | | | | | 00000000b |
| TL1 | 8BH | 定时器 1 低 8 位 | TL1[7:0] | | | | | | | | 00000000b |
| TH0 | 8CH | 定时器 0 高 8 位 | TH0[7:0] | | | | | | | | 00000000b |
| TH1 | 8DH | 定时器 1 高 8 位 | TH1[7:0] | | | | | | | | 00000000b |
| TMCON | 8EH | 定时器频率控制寄存器 | USMD2[1:0] | | - | - | - | - | T1FD | T0FD | 00xxxx00b |
| OTCON | 8FH | 输出控制寄存器 | USMD1[1:0] | | USMD0[1:0] | | - | - | - | - | 0000xxxxb |
| P1 | 90H | P1 口数据寄存器 | P17 | P16 | P15 | P14 | P13 | P12 | P11 | P10 | 00000000b |
| P1CON | 91H | P1 口输入/输出控制寄存器 | P1C7 | P1C6 | P1C5 | P1C4 | P1C3 | P1C2 | P1C1 | P1C0 | 00000000b |
| P1PH | 92H | P1 口上拉电阻控制寄存器 | P1H7 | P1H6 | P1H5 | P1H4 | P1H3 | P1H2 | P1H1 | P1H0 | 00000000b |
| US0CON0 | 95H | USCI0 控制寄存器 0 | US0CON0[7:0] | | | | | | | | 00000000b |
| IOHCON0 | 96H | IOH 设置寄存器 0 | P1H[1:0] | | P1L[1:0] | | P0H[1:0] | | P0L[1:0] | | 00000000b |
| IOHCON1 | 97H | IOH 设置寄存器 1 | P5H[1:0] | | P5L[1:0] | | P2H[1:0] | | P2L[1:0] | | 00000000b |
| SCON | 98H | 串口控制寄存器 | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI | 00000000b |
| SBUF | 99H | 串口数据缓存寄存器 | SBUF[7:0] | | | | | | | | 00000000b |
| P0CON | 9AH | P0 口输入/输出控制寄存器 | P0C7 | P0C6 | P0C5 | P0C4 | P0C3 | P0C2 | P0C1 | P0C0 | 00000000b |
| P0PH | 9BH | P0 口上拉电阻控制寄存器 | P0H7 | P0H6 | P0H5 | P0H4 | P0H3 | P0H2 | P0H1 | P0H0 | 00000000b |
| US0CON1 | 9DH | USCI0 控制寄存器 1 | US0CON1[7:0] | | | | | | | | 00000000b |
| US0CON2 | 9EH | USCI0 控制寄存器 2 | US0CON2[7:0] | | | | | | | | 00000000b |
| US0CON3 | 9FH | USCI0 控制寄存器 3 | US0CON3[7:0] | | | | | | | | 00000000b |
| P2 | A0H | P2 口数据寄存器 | P27 | P26 | P25 | P24 | P23 | P22 | P21 | P20 | 00000000b |
| P2CON | A1H | P2 口输入/输出控制寄存器 | P2C7 | P2C6 | P2C5 | P2C4 | P2C3 | P2C2 | P2C1 | P2C0 | 00000000b |
| P2PH | A2H | P2 口上拉电阻控制寄存器 | P2H7 | P2H6 | P2H5 | P2H4 | P2H3 | P2H2 | P2H1 | P2H0 | 00000000b |
| US1CON0 | A4H | USCI1 控制寄存器 0 | US1CON0[7:0] | | | | | | | | 00000000b |
| US1CON1 | A5H | USCI1 控制寄存器 1 | US1CON1[7:0] | | | | | | | | 00000000b |
| US1CON2 | A6H | USCI1 控制寄存器 2 | US1CON2[7:0] | | | | | | | | 00000000b |
| US1CON3 | A7H | USCI1 控制寄存器 3 | US1CON3[7:0] | | | | | | | | 00000000b |
| IE | A8H | 中断使能寄存器 | EA | EADC | ET2 | EUART | ET1 | EINT1 | ET0 | EINT0 | 00000000b |
| IE1 | A9H | 中断使能寄存器 1 | ET4 | ET3 | - | - | EINT2 | EBTM | EPWM | EUSCI0 | 00xx0000b |
| IE2 | AAH | 中断使能寄存器 2 | - | - | - | - | - | - | EUSCI2 | EUSCI1 | xxxxxx00b |
| ADCCFG0 | ABH | ADC 设置寄存器 0 | EAIN7 | EAIN6 | EAIN5 | EAIN4 | EAIN3 | EAIN2 | EAIN1 | EAIN0 | 00000000b |
| ADCCFG1 | ACH | ADC 设置寄存器 1 | - | - | - | - | EAIN11 | EAIN10 | EAIN9 | EAIN8 | xxxx0000b |
| ADCCON | ADH | ADC 控制寄存器 | ADCEN | ADCS | EOC/ ADCIF | ADCIS[4:0] | | | | | 00000000b |
| ADCVL | AEH | ADC 结果寄存器 | ADCV[3:0] | | | | - | - | - | - | 1111xxxxb |
| ADCVH | AFH | ADC 结果寄存器 | ADCV[11:4] | | | | | | | | 11111111b |
| INT0F | B4H | INT0 下降沿中断控制寄存器 | - | - | - | - | INT0F3 | INT0F2 | INT0F1 | - | xxxx000xb |
| ADCCFG2 | B5H | ADC 设置寄存器 2 | - | - | - | LOWSP[2:0] | | | - | - | xxx000xxb |
| IP | B8H | 中断优先级控制寄存器 | - | IPADC | IPT2 | IPUART | IPT1 | IPINT1 | IPT0 | IPINT0 | x0000000b |
| IP1 | B9H | 中断优先级控制寄存器 1 | IPT4 | IPT3 | - | - | IPINT2 | IPBTM | IPPWM | IPUSCI0 | 00xx0000b |
| IP2 | BAH | 中断优先级控制寄存器 2 | - | - | - | - | - | - | IPUSCI2 | IPUSCI1 | xxxxxx00b |
| INT0R | BBH | INT0 上升沿中断控制寄存器 | - | - | - | - | INT0R3 | INT0R2 | INT0R1 | - | xxxx000xb |

Preliminary

| 符号 | 地址 | 说明 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 上电初始值 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| INT1F | BCH | INT1 下降沿中断控制寄存器 | - | - | - | - | INT1F3 | INT1F2 | INT1F1 | INT1F0 | xxxx0000b |
| INT1R | BDH | INT1 上升沿中断控制寄存器 | - | - | - | - | INT1R3 | INT1R2 | INT1R1 | INT1R0 | xxxx0000b |
| INT2F | BEH | INT2 下降沿中断控制寄存器 | - | - | INT2F5 | INT2F4 | INT2F3 | INT2F2 | INT2F1 | INT2F0 | xx000000b |
| INT2R | BFH | INT2 上升沿中断控制寄存器 | - | - | INT2R5 | INT2R4 | INT2R3 | INT2R2 | INT2R1 | INT2R0 | xx000000b |
| US2CON0 | C4H | USCI2 控制寄存器 0 | US2CON0[7:0] | | | | | | | | 00000000b |
| US2CON1 | C5H | USCI2 控制寄存器 1 | US2CON1[7:0] | | | | | | | | 00000000b |
| US2CON2 | C6H | USCI2 控制寄存器 2 | US2CON2[7:0] | | | | | | | | 00000000b |
| US2CON3 | C7H | USCI2 控制寄存器 3 | US2CON3[7:0] | | | | | | | | 00000000b |
| TXCON | C8H | 定时器 2/3/4 控制寄存器 | TFX | EXFX | RCLK | TCLK | EXENX | TRX | C/TX | CP/RLX | 00000000b |
| TXMOD | C9H | 定时器 2/3/4 工作模式寄存器 | TXFD | - | EPWMN1 | EPWMN0 | INVN1 | INVN0 | TXOE | DCEN | 0x000000b |
| RCAPXL | CAH | 定时器 2/3/4 重载低 8 位 | RCAPXL[7:0] | | | | | | | | 00000000b |
| RCAPXH | CBH | 定时器 2/3/4 重载高 8 位 | RCAPXH[7:0] | | | | | | | | 00000000b |
| TLX | CCH | 定时器 2/3/4 低 8 位 | TLX[7:0] | | | | | | | | 00000000b |
| THX | CDH | 定时器 2/3/4 高 8 位 | THX[7:0] | | | | | | | | 00000000b |
| TXINX | CEH | 定时器控制寄存器指针 | - | - | - | - | - | TXINX[2:0] | | | xxxxx010b |
| WDTCON | CFH | WDT 控制寄存器 | - | - | - | CLRWDT | - | WDTCKS[2:0] | | | xxx0x000b |
| PSW | D0H | 程序状态字寄存器 | CY | AC | F0 | RS1 | RS0 | OV | F1 | P | 00000000b |
| PWMCFG | D1H | PWM0 设置寄存器 | INV7 | INV6 | INV5 | INV4 | INV3 | INV2 | INV1 | INV0 | 00000000b |
| PWMCON0 | D2H | PWM0 控制寄存器 0 | ENPWM | PWMIF | PWMCK[1:0] | | - | - | PWMMD[1:0] | | 0000xx00b |
| PWMCON1 | D3H | PWM0 控制寄存器 1 | ENPWM7 | ENPWM6 | ENPWM5 | ENPWM4 | ENPWM3 | ENPWM2 | ENPWM1 | ENPWM0 | 00000000b |
| PWMPDL | D4H | PWM0 周期寄存器低 8 位 | PWMPDL[7:0] | | | | | | | | 00000000b |
| PWMPDH | D5H | PWM0 周期寄存器高 8 位 | PWMPDH[7:0] | | | | | | | | 00000000b |
| PWMDFR | D6H | PWM0 死区设置寄存器 | PDF1[3:0] | | | | PDR1[3:0] | | | | 00000000b |
| PWMFLT | D7H | PWM0 故障检测设置寄存器 | FLTEN1 | FLTSTA1 | FLTMD1 | FLTLV1 | - | - | FLTDT1[1:0] | | 0000xx00b |
| P5 | D8H | P5 口数据寄存器 | - | - | P55 | P54 | P53 | P52 | P51 | P50 | xx000000b |
| P5CON | D9H | P5 口输入/输出控制寄存器 | - | - | P5C5 | P5C4 | P5C3 | P5C2 | P5C1 | P5C0 | xx000000b |
| P5PH | DAH | P5 口上拉电阻控制寄存器 | - | - | P5H5 | P5H4 | P5H3 | P5H2 | P5H1 | P5H0 | xx000000b |
| ACC | E0H | 累加器 | ACC[7:0] | | | | | | | | 00000000b |
| EXA0 | E9H | 扩展累加器 0 | EXA[7:0] | | | | | | | | 00000000b |
| EXA1 | EAH | 扩展累加器 1 | EXA[15:8] | | | | | | | | 00000000b |
| EXA2 | EBH | 扩展累加器 2 | EXA[23:16] | | | | | | | | 00000000b |
| EXA3 | ECH | 扩展累加器 3 | EXA[31:24] | | | | | | | | 00000000b |
| EXBL | EDH | 扩展 B 寄存器 L | EXB [7:0] | | | | | | | | 00000000b |
| EXBH | EEH | 扩展 B 寄存器 H | EXB [15:8] | | | | | | | | 00000000b |
| OPERCON | EFH | 运算控制寄存器 | OPERS | MD | - | - | - | - | CRCRST | CRCSTA | 00xxxx00b |
| B | F0H | B 寄存器 | B[7:0] | | | | | | | | 00000000b |
| IAPKEY | F1H | IAP 保护寄存器 | IAPKEY[7:0] | | | | | | | | 00000000b |
| IAPADL | F2H | IAP 写入地址低位寄存器 | IAPADR[7:0] | | | | | | | | 00000000b |
| IAPADH | F3H | IAP 写入地址高位寄存器 | IAPADR[15:8] | | | | | | | | 00000000b |
| IAPADE | F4H | IAP 写入扩展地址寄存器 | IAPADER[7:0] | | | | | | | | 00000000b |
| IAPDAT | F5H | IAP 数据寄存器 | IAPDAT[7:0] | | | | | | | | 00000000b |
| IAPCTL | F6H | IAP 控制寄存器 | BTLD | - | SERASE | PRG | - | - | CMD[1:0] | | 0x00xx00b |
| EXADH | F7H | 外部 SRAM 操作地址高位 | - | - | - | - | EXADH [3:0] | | | | xxxx0000b |
| BTMCON | FBH | 低频定时器控制寄存器 | ENBTM | BTMIF | - | - | BTMFS[3:0] | | | | 00xx0000b |
| CRCINX | FCH | CRC 指针 | CRCINX[7:0] | | | | | | | | 00000000b |
| CRCREG | FDH | CRC 寄存器 | CRCREG[7:0] | | | | | | | | nnnnnnnnb |
| OPINX | FEH | Option 指针 | OPINX[7:0] | | | | | | | | 00000000b |
| OPREG | FFH | Option 寄存器 | OPREG[7:0] | | | | | | | | nnnnnnnnb |

### 6.2.2 PWM0 占空比调节寄存器（读/写）

| 地址 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 上电初始值 |
|---|---|---|---|---|---|---|---|---|---|
| 1040H | PDT00[15:8] | | | | | | | | 00000000b |
| 1041H | PDT00[7:0] | | | | | | | | 00000000b |
| 1042H | PDT01[15:8] | | | | | | | | 00000000b |
| 1043H | PDT01[7:0] | | | | | | | | 00000000b |
| 1044H | PDT02[15:8] | | | | | | | | 00000000b |
| 1045H | PDT02[7:0] | | | | | | | | 00000000b |

Preliminary

| | | |
|---|---|---|
| 1046H | PDT03[15:8] | 00000000b |
| 1047H | PDT03[7:0] | 00000000b |
| 1048H | PDT04[15:8] | 00000000b |
| 1049H | PDT04[7:0] | 00000000b |
| 104AH | PDT05[15:8] | 00000000b |
| 104BH | PDT05[7:0] | 00000000b |
| 104CH | PDT06[15:8] | 00000000b |
| 104DH | PDT06[7:0] | 00000000b |
| 104EH | PDT07[15:8] | 00000000b |
| 104FH | PDT07[7:0] | 00000000b |

### 6.2.3 PWM2~4 占空比调节寄存器（读/写）

| 地址 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 上电初始值 |
|---|---|---|---|---|---|---|---|---|---|
| 1034H | PDT20[15:8] | | | | | | | | 00000000b |
| 1035H | PDT20[7:0] | | | | | | | | 00000000b |
| 1036H | PDT21[15:8] | | | | | | | | 00000000b |
| 1037H | PDT21[7:0] | | | | | | | | 00000000b |
| 1038H | PDT30[15:8] | | | | | | | | 00000000b |
| 1039H | PDT30[7:0] | | | | | | | | 00000000b |
| 103AH | PDT31[15:8] | | | | | | | | 00000000b |
| 103BH | PDT31[7:0] | | | | | | | | 00000000b |
| 103CH | PDT40[15:8] | | | | | | | | 00000000b |
| 103DH | PDT40[7:0] | | | | | | | | 00000000b |
| 103EH | PDT41[15:8] | | | | | | | | 00000000b |
| 103FH | PDT41[7:0] | | | | | | | | 00000000b |

### 6.2.4 8051 CPU 内核常用特殊功能寄存器介绍

**程序计数器 PC**

程序计数器 PC 不属于 SFR 寄存器。PC 有 16 位，是用来控制指令执行顺序的寄存器。单片机上电或者复位后，PC 值为 0000H，也即是说单片机程序从 0000H 地址开始执行程序。

**累加器 ACC (E0H)**

累加器 ACC 是 8051 内核单片机的最常用的寄存器之一，指令系统中采用 A 作为助记符。常用来存放参加计算或者逻辑运算的操作数及结果。

**B 寄存器 (F0H)**

B 寄存器在乘除法运算中必须与累加器 A 配合使用。乘法指令 MUL A，B 把累加器 A 和寄存器 B 中的 8 位无符号数相乘，所得的 16 位乘积的低位字节放在 A 中，高位字节放在 B 中。除法指令 DIV A，B 是用 A 除以 B，整数商放在 A 中，余数放在 B 中。寄存器 B 还可以作为通用的暂存寄存器使用。

**堆栈指针 SP (81H)**

堆栈指针是一个 8 位的专用寄存器，它指示出堆栈顶部在通用 RAM 中的位置。单片机复位后，SP 初始值为 07H，即堆栈会从 08H 开始向上增加。08H~1FH 为工作寄存器组 1~3。

**PSW (D0H) 程序状态字寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | CY | AC | F0 | RS1 | RS0 | OV | F1 | P |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **CY** | 标志位<br>1：加法运算最高位有进位，或者减法运算最高位有借位时 |



Preliminary

| | | |
|---|---|---|
| | | 0：加法运算最高位无进位，或者减法运算最高位无借位时 |
| 6 | **AC** | 进位辅助标志位（可在 BCD 码加减法运算时方便调整）<br>1：加法运算时在 bit3 位有进位，或减法运算在 bit3 位有借位时<br>0：无借位、进位 |
| 5 | **F0** | 用户标志位 |
| 4~3 | **RS1、RS0** | 工作寄存器组选择位:<br>RS1 \| RS0 \| 当前使用的工作寄存器组 0~3<br>0 \| 0 \| 组 0 （00H~07H）<br>0 \| 1 \| 组 1 （08H~0FH）<br>1 \| 0 \| 组 2 （10H~17H）<br>1 \| 1 \| 组 3 （18H~1FH） |
| 2 | **OV** | 溢出标志位 |
| 1 | **F1** | F1 标志<br>用户自定义标志 |
| 0 | **P** | 奇偶标志位。此标志位为累加器 ACC 中 1 的个数的奇偶值。<br>1：ACC 中 1 的个数为奇数<br>0：ACC 中 1 的个数为偶数（包括 0 个） |

**数据指针**

SC95F767X 有两个数据指针 DPTR0 和 DPTR1，数据指针 DPTR0/DPTR1 是 16 位的专用寄存器，由低 8 位 DPL/DPL1 和高 8 位 DPH/DPH1 组成。DPTR0/DPTR1 是可以直接进行 16 位操作的寄存器，也可以分别对 DPL 和 DPH 按字节进行操作，数据指针 DPTR0/DPTR1 的选择和工作状态由数据指针选择寄存器 DPS 进行设定。

**DPS（86H）数据指针选择寄存器（读/写）**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ID1 | ID0 | TSL | AU1 | AU0 | - | - | SEL |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | - | - | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | x | x | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **ID1** | DPTR1 加减控制位<br>0: 当 AU1=1 时，每当 MOVC/MOVX @DPTR 执行后，当前 DPTR1 自动加 1<br>1: 当 AU1=1 时，每当 MOVC/MOVX @DPTR 执行后，当前 DPTR1 自动减 1 |
| 6 | **ID0** | DPTR0 加减控制位<br>0: 当 AU0=1 时，每当 MOVC/MOVX @DPTR 执行后，当前 DPTR0 自动加 1<br>1: 当 AU0=1 时，每当 MOVC/MOVX @DPTR 执行后，当前 DPTR0 自动减 1 |
| 5 | **TSL** | SEL 翻转控制位<br>0: 每当 MOVC/MOVX @DPTR 执行后，DPS.0 (SEL) 不翻转<br>1: 每当 MOVC/MOVX @DPTR 执行后，DPS.0 (SEL) 翻转一次 |

Preliminary

| | | |
|---|---|---|
| 4 | **AU1** | DPTR1 自动加减控制位<br>0: 无<br>1: 每当 MOVC/MOVX @DPTR 执行后，当前 DPTR1 自加或自减 1( 依赖 ID1 ) |
| 3 | **AU0** | DPTR0 自动加减控制位<br>0: 无<br>1: 每当 MOVC/MOVX @DPTR 执行后，当前 DPTR0 自加或自减 1( 依赖 ID0 ) |
| 0 | **SEL** | DPTR0、DPTR1 选择位<br>0: MOVC/MOVX @DPTR 对象为 DPTR0<br>1: MOVC/MOVX @DPTR 对象为 DPTR1 |
| 2~1 | **-** | 保留 |

Preliminary

# 7 电源、复位和时钟

## 7.1 电源电路

SC95F767X 电源核心包括了 BG、LDO、POR、LVR 等电路，可实现在 2.0~5.5V 范围内可靠工作。此外，IC 内建经调校过的精准 2.048V/ 1.024V/ 2.4V 电压, 可用作 ADC 内部参考电压。用户可在 18 模数转换 ADC 查找具体设置内容。

## 7.2 上电复位过程

SC95F767X 上电后，在客户端软件执行前，会经过以下的过程：

- 复位阶段
- 调入信息阶段
- 正常操作阶段

### 7.2.1 复位阶段

是指 SC95F767X 会一直处于复位的情况，直到供应给 SC95F767X 的电压高过某一电压，内部才开始有效的 Clock。复位阶段的时间长短和外部电源的上升速度有关，外部电源达到内建 POR 电压后，复位阶段才会完成。

### 7.2.2 调入信息阶段

在 SC95F767X 内部有一个预热计数器。在复位阶段期间，此预热计数器一直被清为 0，直到电压过了 POR 电压后，内部 RC 振荡器开始起振，该预热计数器开始计数。当内部的预热计数器计数到一定数目后，每隔一定数量个 HRC clock 就会从 Flash ROM 中的 IFB（包含 Code Option）读出一个 byte 数据存放到内部系统寄存器中。直到预热完成后，该复位信号才会结束。

### 7.2.3 正常操作阶段

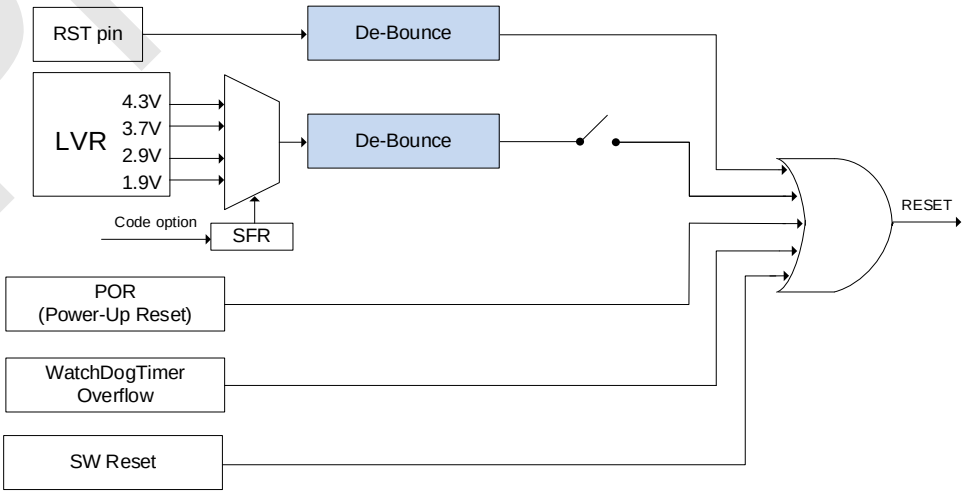
结束调入信息阶段后，SC95F767X 开始从 Flash 中读取指令代码即进入正常操作阶段。这时的 LVR 电压值是用户写入 Code Option 的设置值。

## 7.3 复位方式

SC95F767X 有 5 种复位方式，前四种为硬件复位：

1. 外部 RST 复位
2. 低电压复位 LVR
3. 上电复位 POR
4. 看门狗 WDT 复位
5. 软件复位

SC95F767X 的复位部分电路结构图如下：

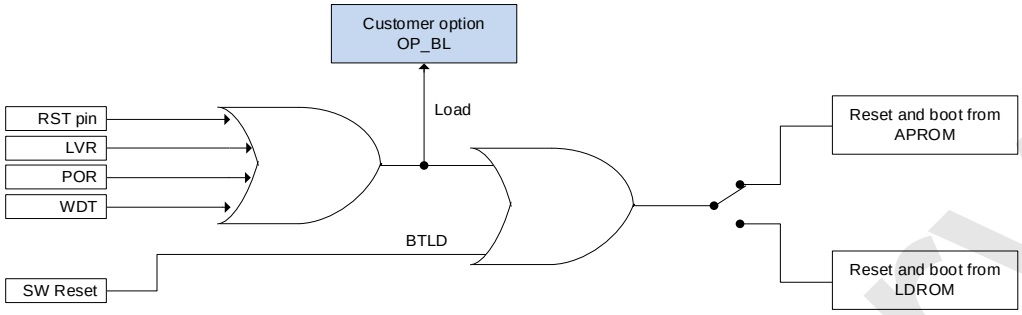



SC95F767X 复位电路图

Preliminary

复位后的启动区域：

1. 外部 RST 复位、低电压复位 LVR、上电复位 POR、看门狗 WDT 这四种硬件复位后，芯片从用户 OP_BL 设定的启动区域(APROM/LDROM)启动。
2. 软件复位后，芯片根据 BTLD（IAPCTL.7）设定的启动区域(APROM/LDROM)启动。




SC95F767X 复位后启动区域切换示意图

### 7.3.1 外部 RST 复位

外部 RST 复位就是从外部 RST 给 SC95F767X 一定宽度的复位脉冲信号，来实现 SC95F767X 的复位。用户在烧录程序前可通过烧录上位机软件配置 Customer Option 项将 P1.1 管脚配置为 RST（复位脚）使用。

### 7.3.2 低电压复位 LVR

SC95F767X 内建了一个低电压复位电路。而复位的门限电压有 4 种选择：4.3V、3.7V、2.9V、1.9V，缺省值是用户写入的 Option 值。当 VDD 电压小于低电压复位的门限电压，且持续时间大于 $T_{LVR}$ 时，会产生复位。其中，$T_{LVR}$ 是 LVR 的消抖时间，约 30μs。

**OP_CTM0 (C1H@FFH) Customer Option 寄存器 0(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ENWDT | ENXTL | SCLKS[1:0] | | DISRST | DISLVR | LVRS[1:0] | |
| 读/写 | 读/写 | 读/写 | 读/写 | | 读/写 | 读/写 | 读/写 | |
| 上电初始值 | n | n | n | | n | n | n | |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 2 | **DISLVR** | LVR 使能设置<br>0：LVR 正常使用<br>1：LVR 无效 |
| 1~0 | **LVRS [1:0]** | LVR 电压选择控制<br>11：4.3 V 复位<br>10：3.7V 复位<br>01：2.9V 复位<br>00：1.9V 复位 |

### 7.3.3 上电复位 POR

SC95F767X 内部有上电复位电路，当电源电压 $V_{DD}$ 达到 POR 复位电压时，系统自动复位。

### 7.3.4 看门狗复位 WDT

SC95F767X 有一个 WDT，其时钟源为内部的 32kHz 振荡器。用户可以通过编程器的 Code Option 选择是否开启看门狗复位功能。

**OP_CTM0 (C1H@FFH) Customer Option 寄存器 0(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ENWDT | ENXTL | SCLKS[1:0] | | DISRST | DISLVR | LVRS[1:0] | |

Preliminary

| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
|---|---|---|---|---|---|---|
| 上电初始值 | n | n | n | n | n | n |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **ENWDT** | WDT 开关(此位由系统将用户 Code Option 所设的值调入)<br>1: WDT 开始工作<br>0: WDT 关闭 |

**WDTCON (CFH) 看门狗控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | CLRWDT | - | WDTCKS[2:0] | | |
| 读/写 | - | - | - | 读/写 | - | 读/写 | | |
| 上电初始值 | x | x | x | 0 | x | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 4 | **CLRWDT** | WDT 清“0”位（写 1 有效）<br>1 ：WDT 计数器从 0 开始计数<br>此位由系统硬件自动置 0 |
| 2~0 | **WDTCKS [2:0]** | 看门狗时钟选择 |
| 7~5,3 | **-** | 保留 |

| WDTCKS[2:0] | WDT 溢出时间 |
|---|---|
| 000 | 500ms |
| 001 | 250ms |
| 010 | 125ms |
| 011 | 62.5ms |
| 100 | 31.5ms |
| 101 | 15.75ms |
| 110 | 7.88ms |
| 111 | 3.94ms |

### 7.3.5 软件复位

**PCON (87h) 电源管理控制寄存器(只写、*不可读 *)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SMOD | - | - | - | RST | - | STOP | IDL |
| 读/写 | 只写 | - | - | - | 只写 | - | 只写 | 只写 |
| 上电初始值 | 0 | x | x | x | n | x | 0 | 0 |

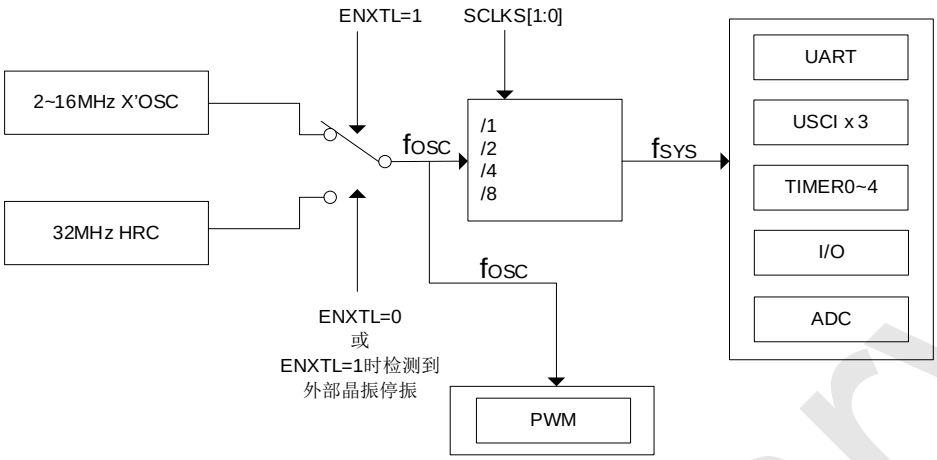
| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3 | **RST** | 软件 reset 控制位：<br>写状态：<br>0：程序正常运行；<br>1：此位被写“1”后 CPU 立刻 reset |

## 7.4 高频系统时钟电路

SC95F767X 内建了一个振荡频率可调的高精度 HRC 及一个晶体振荡电路，用户可以根据选择其中之一作为系统时钟。HRC 出厂时被精确地调校至 32MHz@5V/25℃，用户可以通过编程器的 Code Option 将系统时钟设置为 32/16/8/4MHz 使用。调校过程是过滤掉制程上的偏差对精度所造成的影响。此 HRC 受工作的环境温度和工作电压影响会有一定的漂移：跨越（2.0V~5.5V）及(-40℃~105℃)应用环境，不超过±1%。

为了增强系统的可靠性，SC95F767X 内建有一个系统时钟监控电路，当用户选择系统时钟源为晶体振荡并且晶体振荡电路发生停振时，系统时钟源将被自动切换到内建的 HRC，并保持此状态直至下次复位时再重置。

Preliminary




SC95F767X 内部时钟关系

**OP_CTM0 (C1h@FFH) Customer Option 寄存器 0(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ENWDT | ENXTL | SCLKS[1:0] | | DISRST | DISLVR | LVRS[1:0] | |
| 读/写 | 读/写 | 读/写 | 读/写 | | 读/写 | 读/写 | 读/写 | |
| 上电初始值 | n | n | n | | n | n | n | |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 6 | ENXTL | 外部高频晶振选择开关<br>0：外部高频晶振关闭，P5.0、P5.1 有效；<br>1：外部高频晶振打开，P5.0、P5.1 无效。 |
| 5~4 | SCLKS[1:0] | 系统时钟频率选择：<br>00：系统时钟频率为高频振荡器频率除以 1；<br>01：系统时钟频率为高频振荡器频率除以 2；<br>10：系统时钟频率为高频振荡器频率除以 4；<br>11：系统时钟频率为高频振荡器频率除以 8。 |

SC95F767X 有一个特殊的功能：用户可修改 SFR 的值实现 HRC 频率在一定范围的调整。用户可以通过配置 OP_HRCR 寄存器实现，该寄存器的配置方法可参考章节：5.9.1 Option 相关 SFR 操作说明。

**OP_HRCR (83H@FFH) 系统时钟改变寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | OP_HRCR[7:0] | | | | | | | |
| 读/写 | 读/写 | | | | | | | |
| 上电初始值 | n | n | n | n | n | n | n | n |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | OP_HRCR[7:0] | **HRC 频率改变寄存器**<br>用户可通过修改此寄存器的值实现高频振荡器频率 $f_{HRC}$ 的改变，进而改变 IC 的系统时钟频率 $f_{SYS}$：<br>1. OP_HRCR[7:0]上电后的初始值 OP_HRCR[s]是一个固定值，以确保 $f_{HRC}$ 为 32MHz，每颗 IC 的 OP_HRCR[s]都可能会有差异。 |

Preliminary

2. 初始值为 OP_HRCR[s] 时 IC 的系统时钟频率 $f_{SYS}$ 可通过 Option 项设置为准确的 32/16/8/4MHz，OP_HRCR [7:0]每改变 1 则 $f_{SYS}$ 频率改变约 0.18%。

OP_HRCR [7:0]和 HRC 输出频率的关系如下：

| OP_HRCR [7:0]值 | $f_{SYS}$ 实际输出频率(32M 为例) |
|---|---|
| OP_HRCR [s]-n | 32000*(1-0.18%*n)kHz |
| ... | .... |
| OP_HRCR [s]-2 | 32000*(1-0.18%*2) = 31 884.8kHz |
| OP_HRCR [s]-1 | 32000*(1-0.18%*1) = 31 942.4kHz |
| OP_HRCR [s] | 32000kHz |
| OP_HRCR [s]+1 | 32000*(1+0.18%*1) = 32 057.6kHz |
| OP_HRCR [s]+2 | 32000*(1+0.18%*2) = 32 115.2kHz |
| ... | ... |
| OP_HRCR [s]+n | 32000*(1+0.18%*n)kHz |

注意：
1．IC 每次上电后 OP_HRCR[7:0]的值都是 HRC 工作在最接近 32/16/8/4MHz 的 HRC；用户可借助 EEPROM 在每次上电后修正 HRC 的值以让 HRC 工作在用户需要的频率；
2．为保证 IC 工作可靠，IC 最高工作频率请勿超过 32MHz 的 10%，即 35.2MHz；
3．请用户确认 HRC 频率的改变不会影响其它功能。

## 7.5 低频振荡器及低频时钟定时器

SC95F767X 内建一个频率为 32kHz 的 RC 振荡电路，作为低频时钟定时器 Base Timer 和 WDT 的时钟源。开启 Base Timer 或使能 WDT 均可启动 32kHz 低频振荡器。

低频时钟定时器 Base Timer 可以把 CPU 从 STOP mode 唤醒，并且产生中断。

**BTMCON (FBH) 低频定时器控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ENBTM | BTMIF | - | - | BTMFS[3:0] | | | |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | | | |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **ENBTM** | 低频 Base Timer 启动控制<br>0： Base Timer 不启动<br>1： Base Timer 启动 |
| 6 | **BTMIF** | Base Timer 中断申请标志<br>当 CPU 接受 Base Timer 的中断后，此标志位会被硬件自动清除。 |
| 3~0 | **BTMFS [3:0]** | 低频时钟中断频率选择<br>0000：每 15.625ms 产生一个中断<br>0001：每 31.25ms 产生一个中断<br>0010：每 62.5ms 产生一个中断<br>0011：每 125ms 产生一个中断<br>0100：每 0.25 s 产生一个 中断<br>0101：每 0.5 s 产生一个 中断<br>0110：每 1.0 s 产生一个 中断<br>0111：每 2.0 s 产生一个 中断<br>1000：每 4.0 s 产生一个 中断 |

Preliminary

| | | |
|---|---|---|
| | | 1001：每 8.0 s 产生一个 中断<br>1010：每 16.0 s 产生一个 中断<br>1011：每 32.0 s 产生一个 中断<br>1100~1111：保留 |
| 5~4 | - | 保留 |

## 7.6 STOP 模式和 IDLE 模式

SC95F767X 提供了一个特殊功能寄存器 PCON，配置该寄存器的 bit0 和 bit1 可控制 MCU 进入不同的工作模式。

对 PCON.1 写入 1，内部的高频系统时钟就会停止, 进到 STOP 模式 , 达到省电功能。在 STOP 模式下， 用户可以通过外部中断 INT0~2，低频时钟中断把 SC95F767X 唤醒，也可以通过外部复位将 STOP 唤醒。

对 PCON.0 写入 1，程序停止运行，进入 IDLE 模式，但外部设备及时钟继续运行，进入 IDLE 模式前所有 CPU 状态都被保存。IDLE 模式可由任何中断唤醒。

**PCON (87H) 电源管理控制寄存器(只写、*不可读 *)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SMOD | - | - | - | RST | - | STOP | IDL |
| 读/写 | 只写 | - | - | - | 只写 | - | 只写 | 只写 |
| 上电初始值 | 0 | x | x | x | n | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 1 | **STOP** | STOP 模式控制<br>0： 正常操作模式<br>1： 节能模式， 高频振荡器停止工作，低频振荡器及 WDT 可根据设定选择工作与否。 |
| 0 | **IDL** | IDLE 模式控制<br>0： 正常操作模式<br>1： 节能模式， 程序停止运行，但外部设备及时钟继续运行，进入 IDLE 模式前所有 CPU 状态都被保存。 |

**注意：配置 MCU 进入 STOP 或 IDLE 模式时，对 PCON 寄存器进行配置操作的语句后面要加上至少 12 个 NOP 指令，不能直接跟其它指令，否则在唤醒后无法正常执行后续的指令！**

例如：设置 MCU 进入 STOP 模式：

C 语言例程：

```
#include"intrins.h"

PCON |= 0x02;        //PCON 的 bit1 STOP 位写 1，配置 MCU 进入 STOP 模式
_nop_();             //至少需要 12 个_nop_()
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
……
```

汇编例程：

```
ORL PCON,#02H        ; PCON 的 bit1 STOP 位写 1，配置 MCU 进入 STOP 模式
```

Preliminary

```
NOP                    ; 至少需要 12 个 NOP 指令
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
……
```

# 8 中央处理单元 CPU 及指令系统

## 8.1 CPU

SC95F767X 所用的 CPU 是一个 高速的 1T 标准 8051 内核，其指令完全兼容传统 8051 内核单片机。

## 8.2 寻址方式

SC95F767X 的 1T 8051 CPU 指令的寻址方式有：①立即寻址②直接寻址③间接寻址④寄存器寻址⑤相对寻址⑥变址寻址⑦位寻址

### 8.2.1 立即寻址

立即寻址也称为立即数寻址，它是在指令操作数中直接给出参加运算的操作数，指令举例如下：

MOV A, #50H （这条指令是将立即数 50H 送到累加器 A 中）

### 8.2.2 直接寻址

在直接寻址方式中，指令操作数域给出的是参加运算操作数的地址。直接寻址方式只能用来表示特殊功能寄存器、内部数据寄存器和位地址空间。其中特殊功能寄存器和位地址空间只能用直接寻址方式访问。举例如下：

ANL 50H, #91H （表示 50H 单元中的数与立即数 91H 相“与”，结果存放在 50H 单元中。其中 50H 为直接地址，表示内部数据寄存器 RAM 中的一个单元。）

### 8.2.3 间接寻址

间接寻址采用 R0 或 R1 前添加“@”符号来表示。假设 R1 中的数据是 40H，内部数据存储器 40H 单元的数据为 55H，则指令为

MOV A, @R1 （ 把数据 55H 传送至累加器 A）。

### 8.2.4 寄存器寻址

寄存器寻址是对选定的工作寄存器 R7~R0、累加器 A、通用寄存器 B、地址寄存器和进位 C 中的数进行操作。其中寄存器 R7~R0 由指令码的低 3 位表示，ACC、B、DPTR 及进位位 C 隐含在指令码中。因此，寄存器寻址也包含一种隐含寻址方式。寄存器工作区的选择由程序状态字寄存器 PSW 中的 RS1、RS0 来决定。指令操作数指定的寄存器均指当前工作区的寄存器。

INC R0 是指（R0）+1→R0

### 8.2.5 相对寻址

相对寻址是将程序计数器 PC 中的当前值与指令第二字节给出的数相加，其结果作为转移指令的转移地址。转移地址也成为转移目的地址，PC 中的当前值成为基地址，指令第二字节给出的数成为偏移量。由于目的地址是相对于 PC 中的基地址而言，所以这种寻址方式成为相对寻址。偏移量为带符号的数，所能表示的范围为+127~-128。这种寻址方式主要用于转移指令。

JC ＄+50H

表示若进位位 C 为 0，则程序计数器 PC 中的内容不改变，即不转移。若进位位 C 为 1，则以 PC 中的当前值及基



Preliminary

地址，加上偏移量 50H 后所得到的结果作为该转移指令的目的地址。

### 8.2.6 变址寻址

在变址寻址方式中，指令操作数制定一个存放变址基址的变址寄存器。变址寻址时，偏移量与变址基值相加，其结果作为操作数的地址。变址寄存器有程序计数器 PC 和地址寄存器 DPTR。

MOVC A, @A+DPTR

表示累加器 A 为偏移量寄存器，其内容与地址寄存器 DPTR 中的内容相加，其结果作为操作数的地址，取出该单元中的数送入累加器 A 中。

### 8.2.7 位寻址

位寻址是指对一些可进行位操作的内部数据存储器 RAM 和特殊功能寄存器进行位操作时的寻址方式。在进行位操作时，借助于进位位 C 作为位操作累加器，指令操作数直接给出该位的地址，然后根据操作码的性质对该位进行位操作。位地址与字节直接寻址中的字节地址编码方式完全一样，主要由操作指令的性质加以区分，使用时应特别注意。

MOV C, 20H (将地址为 20H 的位操作寄存器值送入进位位 C 中。)

Preliminary

# 9 INTERRUPT 中断

SC95F767X 单片机提供 15 个中断源：Timer0~4，INT0~2，ADC，PWM，UART，USCI0~2，Base Timer。这 15 个中断源分为 2 个中断优先级，并可以单独分别设置为高优先级或者低优先级。三个外部中断可以分别设定其中每个中断源的触发条件为上升、下降或上下沿，每个中断分别有独立的优先级设置位、中断标志、中断向量和使能位，总的中断使能位 EA 可以实现所有中断的打开或者关闭。

## 9.1 中断源、向量

SC95F767X 的中断源、中断向量、及相关控制位列表如下：

| 中断源 | 中断发生时间 | 中断标志 | 中断使能控制 | 中断优先权控制 | 中断向量 | 查询优先级 | 中断号（C51） | 标志清除方式 | 能否唤醒 STOP |
|---|---|---|---|---|---|---|---|---|---|
| INT0 | 外部中断 0 条件符合 | IE0 | EINT0 | IPINT0 | 0003H | 1 (高) | 0 | H/W Auto | 能 |
| Timer0 | Timer0 溢出 | TF0 | ET0 | IPT0 | 000BH | 2 | 1 | H/W Auto | 不能 |
| INT1 | 外部中断 1 条件符合 | IE1 | EINT1 | IPINT1 | 0013H | 3 | 2 | H/W Auto | 能 |
| Timer1 | Timer1 溢出 | TF1 | ET1 | IPT1 | 001BH | 4 | 3 | H/W Auto | 不能 |
| UART | 接收或发送完成 | RI/TI | EUART | IPUART | 0023H | 5 | 4 | 必须用户清除 | 不能 |
| Timer2 | Timer2 溢出 | TF2 | ET2 | IPT2 | 002BH | 6 | 5 | 必须用户清除 | 不能 |
| ADC | ADC 转换完成 | ADCIF | EADC | IPADC | 0033H | 7 | 6 | 必须用户清除 | 不能 |
| USCI0 | 接收或发送完成 | SPIF0/TWIF0 | EUSCI0 | IPSPI | 003BH | 8 | 7 | 必须用户清除 | 不能 |
| PWM | PWM 溢出 | PWMIF | EPWM | IPPWM | 0043H | 9 | 8 | 必须用户清除 | 不能 |
| BTM | Base timer 溢出 | BTMIF | EBTM | IPBTM | 004BH | 10 | 9 | H/W Auto | 能 |
| INT2 | 外部中断 2 条件符合 | - | EINT2 | IPINT2 | 0053H | 11 | 10 | - | 能 |
| Timer3 | Timer3 溢出 | TFX | ET3 | IPT3 | 006BH | 14 | 13 | 必须用户清除 | 不能 |
| Timer4 | Timer4 溢出 | TFX | ET4 | IPT4 | 0073H | 15 | 14 | 必须用户清除 | 不能 |
| USCI1 | 接收或发送完成 | SPIF1/TWIF1 | EUSCI1 | IPSPI1 | 007BH | 16 | 15 | 必须用户清除 | 不能 |
| USCI2 | 接收或发送完成 | SPIF2/TWIF2 | EUSCI2 | IPSPI2 | 0083H | 17 | 16 | 必须用户清除 | 不能 |

在 EA=1 及各中断使能控制为 1 的情况下，各中断发生情况如下：

定时器中断：Timer0 和 Timer1 溢出时会产生中断并将中断标志 TF0 和 TF1 置为“1”，当单片机执行该定时器中断时，中断标志 TF0 和 TF1 会被硬件自动清“0”。Timer2 溢出时会产生中断并将中断标志 TF2 置为“1”，在 Timer2 中断发生后，硬件并不会自动清除 TF2 位，此 bit 必须由使用者的软件负责清除。

ADC 中断：ADC 中断的发生时间为 ADC 转换完成时，其中断标志就是 ADC 转换结束标志 EOC/ADCIF（ADCCON.5）。当使用者设定 ADCS 开始转换后， EOC 会被硬件自动清除为 “0” ；当转换完成后，EOC 会被硬件自动置为 “1” 。使用者在 ADC 中断发生之后，进入中断服务程序时，必须用软件去清除它。

USCI 中断：当 USCI 接收或发送一帧数据完成时 SPIF/TWIF 位会被硬件自动置“1”， USCI 中断产生。当单片机执行该 USCI 中断时，中断标志 SPIF/TWIF 必须由使用者的软件负责清除。

PWM 中断：当 PWM 计数器溢出时(也就是说：计数器数到超过 PWMPRD 时)，PWMIF 位(PWM Interrupt

Preliminary

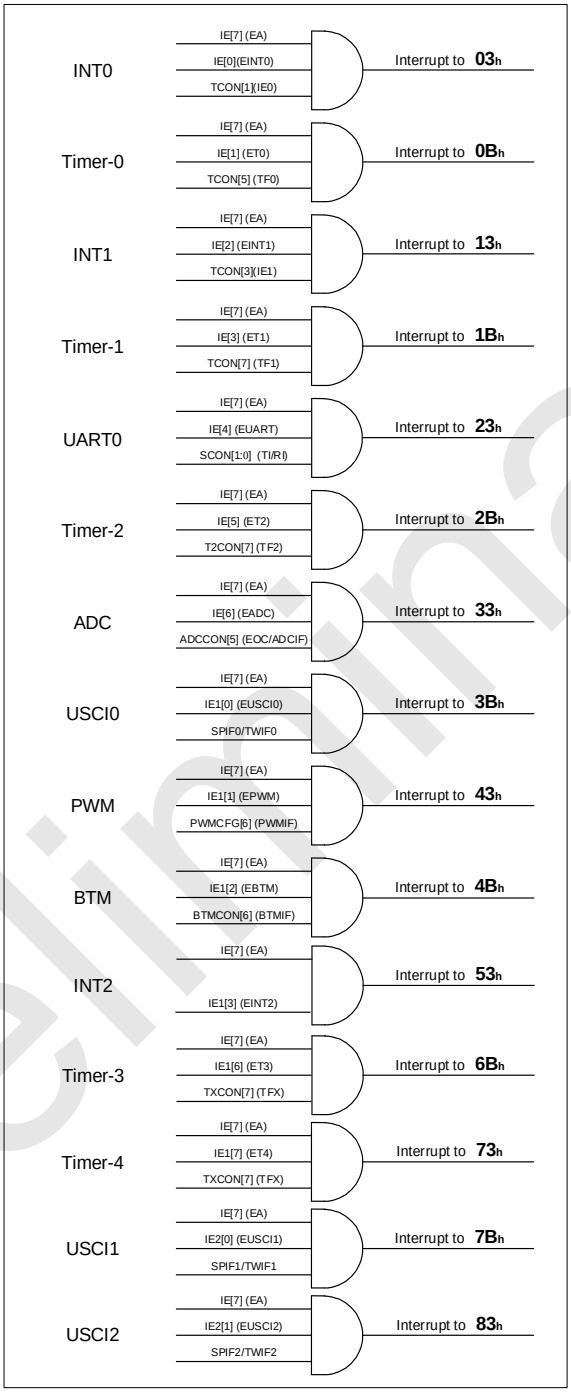
Flag)会被硬件自动置“1”，PWM 中断产生。在 PWM 中断发生后，硬件并不会自动清除此位，此位必须由使用者的软件负责清除。

外部中断 INT0~2：当外部中断口有中断条件发生时，外部中断就发生了。INT0 有三个外部中断源，INT1 有四个外部中断源，INT2 有六个外部中断源，用户可以根据需要设成上沿、下沿或者双沿中断，可通过设置 SFR（INTxF 和 INTxR）来实现。用户可通过 IP 寄存器来设置每个中断的优先级级别。外部中断 INT0~2 还可以唤醒单片机的 STOP。

Preliminary

## 9.2 中断结构图

SC95F767X 的中断结构如下图所示：




SC95F767X 中断结构和向量

Preliminary

## 9.3 中断优先级

SC95F767X 单片机的中断具有两个中断优先级，这些中断源的请求可编程为高优先级中断或者低优先级中断，即可实现两级中断服务程序的嵌套。一个正在执行的低优先级中断能被高优先级中断请求所中断，但不能被另一个同一优先级的中断请求所中断，一直执行到结束，遇到返回指令 RETI，返回主程序后再执行一条指令才能响应新的中断请求。

也就是说：

① 低优先级中断可被高优先级中断请求所中断，反之不能；

② 任何一种中断，在响应过程中，不能被同一优先级的中断请求所中断。

中断查询顺序：SC95F767X 单片机的同一优先级中断，如果同时来几个中断，则中断响应的优先顺序同 C51 中的中断查询号相同，即查询号小的会优先响应，查询号大的会慢响应。

## 9.4 中断处理流程

当一个中断产生并且被 CPU 响应，则主程序运行被中断，将执行下述操作

① 当前正在执行的指令执行完；

② PC 值被压入堆栈，保护现场；

③ 中断向量地址载入程序计数器 PC；

④ 执行相应的中断服务程序；

⑤ 中断服务程序结束并 RETI；

⑥ 将 PC 值退栈，并返回执行中断前的程序。

在此过程中，系统不会立即执行其它同一优先级的中断，但会保留所发生的中断请求，在当前中断处理结束后，转去执行新的中断请求。

## 9.5 中断相关 SFR 寄存器

**IE (A8H) 中断使能寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | EA | EADC | ET2 | EUART | ET1 | EINT1 | ET0 | EINT0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **EA** | 中断使能的总控制<br>0: 关闭所有的中断<br>1: 打开所有的中断 |
| 6 | **EADC** | ADC 中断使能控制<br>0：关闭 ADC 中断<br>1：允许 ADC 转换完成时产生中断 |
| 5 | **ET2** | Timer2 中断使能控制<br>0：关闭 TIMER2 中断<br>1：允许 TIMER2 中断 |
| 4 | **EUART** | UART 中断使能控制<br>0：关闭 UART 中断<br>1：允许 UART 中断 |
| 3 | **ET1** | Timer1 中断使能控制<br>0：关闭 TIMER1 中断<br>1：允许 TIMER1 中断 |
| 2 | **EINT1** | 外部中断 1 使能控制<br>0：关闭 INT1 中断 |

Preliminary

| | | |
|---|---|---|
| | | 1：打开 INT1 中断 |
| 1 | **ET0** | Timer0 中断使能控制<br>0：关闭 TIMER0 中断<br>1：允许 TIMER0 中断 |
| 0 | **EINT0** | 外部中断 0 使能控制<br>0：关闭 INT0 中断<br>1：打开 INT0 中断 |

**IP (B8H) 中断优先级控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | IPADC | IPT2 | IPUART | IPT1 | IPINT1 | IPT0 | IPINT0 |
| 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 6 | **IPADC** | ADC 中断优先权选择<br>0：ADC 中断优先权为低<br>1：ADC 中断优先权为高 |
| 5 | **IPT2** | Timer2 中断优先权选择<br>0：Timer2 中断优先权为低<br>1：Timer2 中断优先权为高 |
| 4 | **IPUART** | UART 中断优先权选择<br>0：UART 中断优先权为低<br>1：UART 中断优先权为高 |
| 3 | **IPT1** | Timer1 中断优先权选择<br>0：Timer1 中断优先权为低<br>1：Timer1 中断优先权为高 |
| 2 | **IPINT1** | INT1 计数器中断优先权选择<br>0：INT1 中断优先权为低<br>1：INT1 中断优先权为高 |
| 1 | **IPT0** | Timer0 中断优先权选择<br>0：Timer0 中断优先权为低<br>1：Timer0 中断优先权为高 |
| 0 | **IPINT0** | INT0 计数器中断优先权选择<br>0：INT0 中断优先权为低<br>1：INT0 中断优先权为高 |
| 7 | **-** | 保留 |

**IE1 (A9H) 中断控制寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ET4 | ET3 | - | - | EINT2 | EBTM | EPWM | EUSCI |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **ET4** | Timer4 中断使能控制<br>0：关闭 Timer4 中断<br>1：允许 Timer4 中断 |
| 6 | **ET3** | Timer3 中断使能控制<br>0：关闭 Timer3 中断 |

Preliminary

| | | |
|---|---|---|
| | | 1：允许 Timer3 中断 |
| 3 | **EINT2** | 外部中断 2 使能控制<br>0：关闭 INT2 中断<br>1：打开 INT2 中断 |
| 2 | **EBTM** | Base Timer 中断使能控制<br>0： 关闭 Base Timer 中断<br>1： 允许 Base Timer 中断 |
| 1 | **EPWM** | PWM 中断使能控制<br>0：关闭 PWM 中断<br>1：允许 PWM 计数溢出（数到 PWMPRD）时产生中断 |
| 0 | **EUSCI** | 三合一串口中断使能控制<br>0： 关闭串口中断<br>1： 允许串口中断 |
| 5~4 | **-** | 保留 |

**IP1 (B9H) 中断优先级控制寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IPT4 | IPT3 | - | - | IPINT2 | IPBTM | IPPWM | IPUSCI0 |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **IPT4** | Timer4 中断优先权选择<br>0：Timer4 中断优先权为低<br>1：Timer4 中断优先权为高 |
| 6 | **IPT3** | Timer3 中断优先权选择<br>0：Timer3 中断优先权为低<br>1：Timer3 中断优先权为高 |
| 3 | **IPINT2** | INT2 计数器中断优先权选择<br>0：INT2 中断优先权为低<br>1：INT2 中断优先权为高 |
| 2 | **IPBTM** | Base Timer 中断优先权选择<br>0：Base Timer 中断优先权为低<br>1：Base Timer 中断优先权为高 |
| 1 | **IPPWM** | PWM 中断使能选择<br>0： PWM 中断优先权为低<br>1： PWM 中断优先权为高 |
| 0 | **IPUSCI0** | 三合一串口中断优先权选择<br>0： USCI0 中断优先权为低<br>1： USCI0 中断优先权为高 |
| 5~4 | **-** | 保留 |

**IE2 (AAH) 中断控制寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | - | - | EUSCI2 | EUSCI1 |
| 读/写 | - | - | - | - | - | - | 读/写 | 读/写 |
| 上电初始值 | x | x | x | x | x | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 1 | **EUSCI2** | 三合一串口 USCI2 中断使能控制<br>0： 关闭串口中断<br>1： 允许串口中断 |

Preliminary

| | | |
|---|---|---|
| 0 | **EUSCI1** | 三合一串口 USCI1 中断使能控制<br>0： 关闭串口中断<br>1： 允许串口中断 |
| 7~2 | **-** | 保留 |

**IP2 (BAH) 中断优先级控制寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | - | - | IPUSCI2 | IPUSCI1 |
| 读/写 | - | - | - | - | - | - | 读/写 | 读/写 |
| 上电初始值 | x | x | x | x | x | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 1 | **IPUSCI2** | 三合一串口 USCI2 中断优先权选择<br>0： USCI2 中断优先权为低<br>1： USCI2 中断优先权为高 |
| 0 | **IPUSCI1** | 三合一串口 USCI1 中断优先权选择<br>0： USCI1 中断优先权为低<br>1： USCI1 中断优先权为高 |
| 7~2 | **-** | 保留 |

**TCON (88H) 定时器控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TF1 | TR1 | TF0 | TR0 | IE1 | - | IE0 | - |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | - | 读/写 | - |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | x | 0 | x |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3 | **IE1** | INT1 溢出中断请求标志。INT1 产生溢出，发生中断时，硬件将 IE1 置为“1”，申请中断，CPU 响应时，硬件清“0”。 |
| 1 | **IE0** | INT0 溢出中断请求标志。INT0 产生溢出，发生中断时，硬件将 IE0 置为“1”，申请中断，CPU 响应时，硬件清“0”。 |
| 2,0 | **-** | 保留 |

**INT0F (BAH) INT0 下降沿中断控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | INT0F3 | INT0F2 | INT0F1 | - |
| 读/写 | - | - | - | - | 读/写 | 读/写 | 读/写 | - |
| 上电初始值 | x | x | x | x | 0 | 0 | 0 | x |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3~1 | **INT0Fn (n=1~3)** | INT0 下降沿中断控制<br>0 : INT0n 下降沿中断关闭<br>1: INT0n 下降沿中断使能 |
| 7~4,0 | **-** | 保留 |

**INT0R (BBH) INT0 上升沿中断控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | INT0R3 | INT0R2 | INT0R1 | - |
| 读/写 | - | - | - | - | 读/写 | 读/写 | 读/写 | - |
| 上电初始值 | x | x | x | x | 0 | 0 | 0 | x |

Preliminary

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3~1 | **INT0Rn (n=1~3)** | INT0 上升沿中断控制<br>0 : INT0n 上升沿中断关闭<br>1: INT0n 上升沿中断使能 |
| 7~4,0 | **-** | 保留 |

**INT1F (BCH) INT1 下降沿中断控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | INT1F3 | INT1F2 | INT1F1 | INT1F0 |
| 读/写 | - | - | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | x | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3~0 | **INT1Fn (n=0~3)** | INT1 下降沿中断控制<br>0 : INT1n 下降沿中断关闭<br>1: INT1n 下降沿中断使能 |
| 7~4 | **-** | 保留 |

**INT1R (BDH) INT1 上升沿中断控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | INT1R3 | INT1R2 | INT1R1 | INT1R0 |
| 读/写 | - | - | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | x | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3~0 | **INT1Rn (n=0~3)** | INT1 上升沿中断控制<br>0 : INT1n 上升沿中断关闭<br>1: INT1n 上升沿中断使能 |
| 7~4 | **-** | 保留 |

**INT2F (C6H) INT2 下降沿中断控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | INT2F5 | INT2F4 | INT2F3 | INT2F2 | INT2F1 | INT2F0 |
| 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | x | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 5~0 | **INT2Fn (n=0~5)** | INT2 下降沿中断控制<br>0 : INT2n 下降沿中断关闭<br>1: INT2n 下降沿中断使能 |
| 7~6 | **-** | 保留 |

**INT2R (C7H) INT2 上升沿中断控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | INT2R5 | INT2R4 | INT2R3 | INT2R2 | INT2R1 | INT2R0 |
| 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | x | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 5~0 | **INT2Rn (n=0~5)** | INT2 上升沿中断控制<br>0 : INT2n 上升沿中断关闭 |

Preliminary

| | | |
|---|---|---|
| | | 1: INT2n 上升沿中断使能 |
| 7~6 | - | 保留 |

# 10 定时器 TIMER0 、TIMER1

SC95F767X 单片机内部的五个 16 位定时器/计数器，其中 Timer0/1 有独立的寄存器组，Timer2~4 共用寄存器组。本章节主要为 Timer0~1 的功能介绍，Timer2~4 详见下一章节。它们具有计数方式和定时方式两种工作模式。特殊功能寄存器 TMOD 中有一个控制位 C/Tx 来选择 T0 和 T1 是定时器还是计数器。它们本质上都是一个加法计数器，只是计数的来源不同。定时器的来源为系统时钟或者其分频时钟，但计数器的来源为外部管脚的输入脉冲。只有在 TRx=1 的时候，T0 和 T1 才会被打开计数。

计数器模式下，P1.2/T0 和 P1.3/T1 管脚上的每一个脉冲，T0 和 T1 的计数值分别增加 1。

定时器模式下，可通过特殊功能寄存器 TMCON 来选择 T0 和 T1 的计数来源是 $f_{SYS}/12$ 或 $f_{SYS}$ ($f_{SYS}$ 为分频后的系统时钟)。

定时器/计数器 T0 有 4 种工作模式，定时器/计数器 T1 有 3 种工作模式(模式三不存在)：

① 模式 0：13 位定时器/计数器模式

② 模式 1：16 位定时器/计数器模式

③ 模式 2： 8 位自动重载模式

④ 模式 3：两个 8 位定时器/计数器模式

在上述模式中，T0 和 T1 的模式 0、1、2 都相同，模式 3 不同。

## 10.1 T0 和 T1 相关特殊功能寄存器

| 符号 | 地址 | 说明 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Reset 值 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TCON | 88H | 定时器控制寄存器 | TF1 | TR1 | TF0 | TR0 | IE1 | - | IE0 | - | 00000x0xb |
| TMOD | 89H | 定时器工作模式寄存器 | - | C/T1 | M11 | M01 | - | C/T0 | M10 | M00 | x000x000b |
| TL0 | 8AH | 定时器 0 低 8 位 | TL0[7:0] | | | | | | | | 00000000b |
| TL1 | 8BH | 定时器 1 低 8 位 | TL1[7:0] | | | | | | | | 00000000b |
| TH0 | 8CH | 定时器 0 高 8 位 | TH0[7:0] | | | | | | | | 00000000b |
| TH1 | 8DH | 定时器 1 高 8 位 | TH1[7:0] | | | | | | | | 00000000b |
| TMCON | 8EH | 定时器频率控制寄存器 | USMD2[1:0] | | - | - | - | - | T1FD | T0FD | xxxxxx00b |

各寄存器的解释说明如下：

**TCON (88H) 定时器控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TF1 | TR1 | TF0 | TR0 | IE1 | - | IE0 | - |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | - | 读/写 | - |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | x | 0 | x |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **TF1** | T1 溢出中断请求标志。T1 产生溢出，发生中断时，硬件将 TF1 置为“1”，申请中断，CPU 响应时，硬件清“0”。 |
| 6 | **TR1** | 定时器 T1 的运行控制位。此位由软件置 1 和清 0。当 TR1=1 时，允许 T1 开始计数。TR1=0 时禁止 T1 计数。 |
| 5 | **TF0** | T0 溢出中断请求标志。T0 产生溢出，发生中断时，硬件将 TF0 置为“1”，申请中断，CPU 响应时，硬件清“0”。 |
| 4 | **TR0** | 定时器 T0 的运行控制位。此位由软件置位和清 0。当 TR0=1 时，允许 T0 开始计数。TR0=0 时禁止 T0 计数。 |
| 2,0 | - | 保留 |

**TMOD (89H) 定时器工作模式寄存器(读/写)**



Preliminary

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | C/T1 | M11 | M01 | - | C/T0 | M10 | M00 |
| 读/写 | - | 读/写 | 读/写 | 读/写 | - | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | 0 | 0 | 0 | x | 0 | 0 | 0 |
| | T1 | | | | T0 | | | |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 6 | **C/T1** | TMOD[6]控制定时器 1<br>0：定时器，T1 计数来源于 $f_{SYS}$ 分频<br>1：计数器，T1 计数来源于外部管脚 T1/P1.3 |
| 5~4 | **M11,M01** | 定时器/计数器 1 模式选择<br>00： 13 位定时器/计数器，TL1 高 3 位无效<br>01： 16 位定时器/计数器，TL1 和 TH1 全有效<br>10： 8 位自动重载定时器，溢出时将 TH1 存放的值自动重装入 TL1<br>11： 定时器/计数器 1 无效(停止计数) |
| 2 | **C/T0** | TMOD[2]控制定时器 0<br>0：定时器，T0 计数来源于 $f_{SYS}$ 分频<br>1：计数器，T0 计数来源于外部管脚 T0/P1.2 |
| 1~0 | **M10,M00** | 定时器/计数器 0 模式选择<br>00： 13 位定时器/计数器，TL0 高 3 位无效<br>01： 16 位定时器/计数器，TL0 和 TH0 全有效<br>10： 8 位自动重载定时器，溢出时将 TH0 存放的值自动重装入 TL0<br>11： 定时器 0 此时作为双 8 位定时器/计数器。TL0 作为一个 8 位定时器/计数器，通过标准定时器 0 的控制位控制；TH0 仅作为一个 8 位定时器，由定时器 1 的控制位控制。 |
| 7,3 | **-** | 保留 |

TMOD 寄存器中 TMOD[0]~TMOD[2]是设置 T0 的工作模式；TMOD[4]~TMOD[6]是设置 T1 的工作模式。

定时器和计数器 Tx 功能由特殊功能寄存器 TMOD 的控制位 C/Tx 来选择，M0x 和 M1x 都是用来选择 Tx 的工作模式。TRx 作为 T0 和 T1 的开关控制，只有 TRx=1 时 T0 和 T1 才打开。

**TMCON (8EH) 定时器频率控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | USMD2[1:0] | | - | - | - | - | T1FD | T0FD |
| 读/写 | 读/写 | 读/写 | - | - | - | - | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | x | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 1 | **T1FD** | T1 输入频率选择控制<br>0：T1 频率源自于 $f_{SYS}/12$<br>1：T1 频率源自于 $f_{SYS}$ |
| 0 | **T0FD** | T0 输入频率选择控制<br>0：T0 频率源自于 $f_{SYS}/12$<br>1：T0 频率源自于 $f_{SYS}$ |

**IE (A8H) 中断使能寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | EA | EADC | ET2 | - | ET1 | EINT1 | ET0 | EINT0 |
| 读/写 | 读/写 | 读/写 | 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | x | 0 | 0 | 0 | 0 |

Preliminary

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3 | **ET1** | Timer1 中断使能控制<br>0： 关闭 TIMER1 中断<br>1： 允许 TIMER1 中断 |
| 1 | **ET0** | Timer0 中断使能控制<br>0： 关闭 TIMER0 中断<br>1： 允许 TIMER0 中断 |

**IP (B8H) 中断优先级控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | IPADC | IPT2 | - | IPT1 | IPINT1 | IPT0 | IPINT0 |
| 读/写 | - | 读/写 | 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | 0 | 0 | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3 | **IPT1** | Timer1 中断优先权<br>0：设定 Timer 1 的中断优先权是 “低”<br>1：设定 Timer 1 的中断优先权是 “高” |
| 1 | **IPT0** | Timer0 中断优先权<br>0：设定 Timer 0 的中断优先权是 “低”<br>1：设定 Timer 0 的中断优先权是 “高” |

## 10.2 T0 工作模式

通过对寄存器 TMOD 中的 M10、M00(TMOD[1]、TMOD[0])的设置，定时器/计数器 0 可实现 4 种不同的工作模式 。

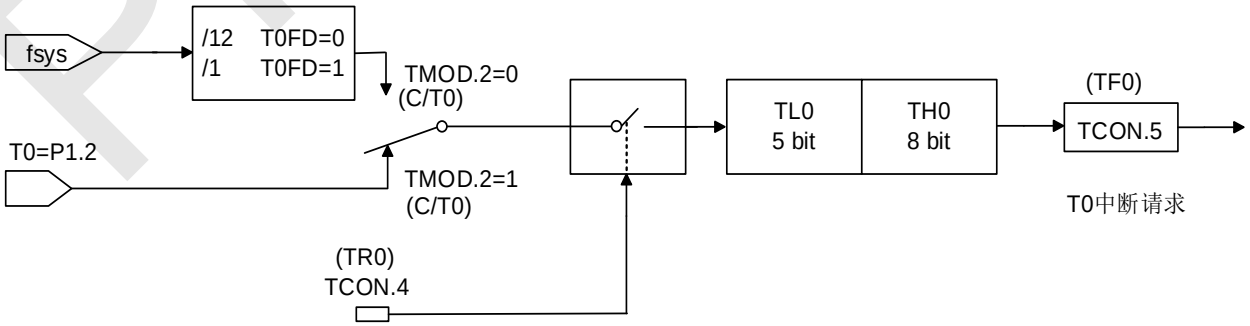
**工作模式 0: 13 位计数器/定时器。**

TH0 寄存器存放 13 位计数器/定时器的高 8 位(TH0.7~TH0.0), TL0 存放低 5 位(TL0.4~TL0.0)。TL0 的高三位(TL0.7~TL0.5)是不确定值，读取时应被忽略掉。当 13 位定时器/计数器递增溢出时，系统会将定时器溢出标志 TF0 置 1。如果定时器 0 中断被允许，将会产生一个中断。

C/T0 位选择计数器/定时器的时钟输入源。如果 C/T0=1，定时器 0 输入脚 T0(P1.2)的电平从高到低的变化，会使定时器 0 数据寄存器加 1。如果 C/T0=0，选择系统时钟的分频为定时器 0 的时钟源。

当 TR0 置 1 打开定时器 T0。TR0 置 1 并不强行复位定时器，意味着如果 TR0 置 1，定时器寄存器将从上次 TR0 清 0 时的值开始计数。所以，在允许定时器之前，应该设定定时器寄存器的初始值。

当作为定时器应用时，可配置 T0FD 来选择时钟源的分频比例。



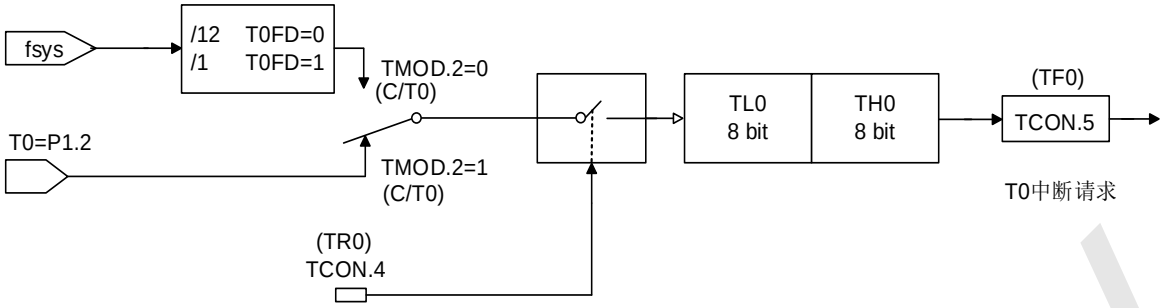

定时器/计数器工作模式 0: 13 位定时器/计数器

**工作模式 1: 16 位计数器/定时器**

除了使用 16 位(TL0 的 8 位数据全部有效)计数器/定时器之外，模式 1 和模式 0 的运行方式相同。打开和配置

Preliminary

计数器/定时器方式也相同。



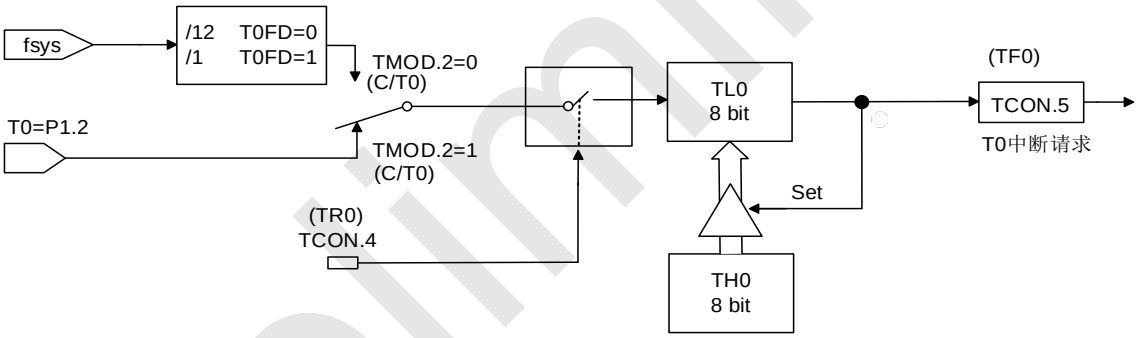

定时器/计数器工作模式 0: 16 位定时器/计数器

**工作模式 2: 8 位自动重载计数器/定时器**

在工作模式 2 中, 定时器 0 是 8 位自动重载计数器/定时器。TL0 存放计数值，TH0 存放重载值。当在 TL0 中的计数器溢出至 0x00 时，定时器溢出标志 TF0 被置 1，寄存器 TH0 的值被重载入寄存器 TL0 中。如果定时器中断使能，当 TF0 置 1 时将产生一个中断，但在 TH0 中的重载值不会改变。在允许定时器正确计数开始之前，TL0 必须初始化为所需要的值。

除了自动重载功能外，工作模式 2 中的计数器/定时器的使能和配置方式同模式 0 和 1 是相同的。

当作为定时器应用时，可配置寄存器 TMCON.0(T0FD)来选择定时器时钟源被系统时钟 $f_{SYS}$ 分频的比例。




定时器/计数器工作模式 2: 自动重载的 8 位定时器/计数器

**工作模式 3: 两个 8 位计数器/定时器(仅限于定时器 0)**

在工作模式 3 中，定时器 0 用作两个独立的 8 位计数器/定时器，分别由 TL0 和 TH0 控制。TL0 通过定时器 0 的控制位(在 TCON 中)和状态位(在 TMOD 中)：TR0、C/T0、TF0 控制。定时器 0 可通过 T0 的 TMOD.2(C/T0)来选择是定时器模式还是计数器模式。

TH0 通过定时器 1 的控制 TCON 来设置相关的控制，但 TH0 仅被限定为定时器模式，无法通过 TMOD.2(C/T0)来设定为计数器模式。TH0 由定时器控制位 TR1 的控制使能，需设定 TR1=1。当发生溢出及产生中断时，TF1 会置 1，并按 T1 发生中断来进行相应的处理。

在 T0 被设为工作模式 3 时，TH0 定时器占用了 T1 的中断资源及 TCON 中寄存器，T1 的 16 位计数器会停止计数，相当于“TR1=0”。当采用 TH0 定时器工作时，需设置 TR1=1。

## 10.3 T1 工作模式

通过对寄存器 TMOD 中的 M11、M01(TMOD[5]、TMOD[4])的设置，定时器/计数器 1 可实现 3 种不同的工作模式 。

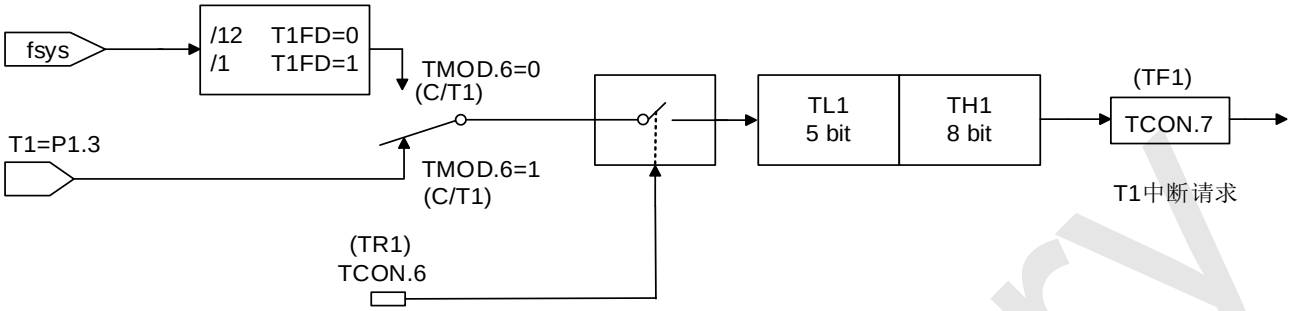
**工作模式 0: 13 位计数器/定时器。**

TH1 寄存器存放 13 位计数器/定时器的高 8 位(TH1.7~TH1.0)；TL1 存放低 5 位(TL1.4~TL1.0)。TL1 的高三位(TL1.7~TL1.5)是不确定值，读取时应被忽略掉。当 13 位定时器计数器递增溢出时，系统会将定时器溢出标志 TF1 置 1。如果定时器 1 中断被允许，将会产生一个中断。C/T1 位选择计数器/定时器的时钟源。

如果 C/T1=1，定时器 1 输入脚 T1(P1.3)的电平从高到低的变化，会使定时器 1 数据寄存器加 1。如果 C/T1=0，选择系统时钟的分频为定时器 1 的时钟源。

TR1 置 1 打开定时器。TR1 置 1 并不强行复位定时器，意味着如果 TR1 置 1，定时器寄存器将从上次 TR1 清 0 时的值开始计数。所以，在允许定时器之前，应该设定定时器寄存器的初始值。

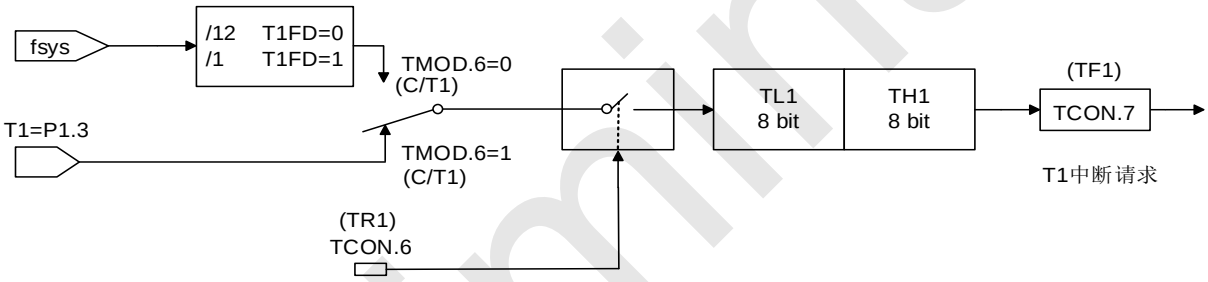
当作为定时器应用时，可配置 T1FD 来选择时钟源的分频比例。




定时器/计数器工作模式 0: 13 位定时器/计数器

**工作模式 1: 16 位计数器/定时器**

除了使用 16 位(TL1 的 8 位数据全部有效)计数器/定时器之外，模式 1 和模式 0 的运行方式相同。打开和配置计数器/定时器方式也相同。



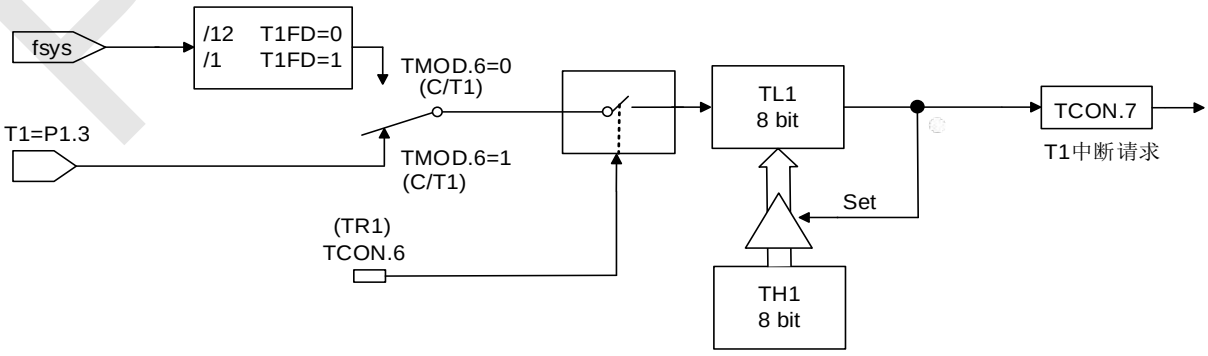

定时器/计数器工作模式 0: 16 位定时器/计数器

**工作模式 2: 8 位自动重载计数器/计数器**

在工作模式 2 中，定时器 1 是 8 位自动重载计数器/定时器。TL1 存放计数值，TH1 存放重载值。当在 TL1 中的计数器溢出至 0x00 时，定时器溢出标志 TF1 被置 1，寄存器 TH1 的值被重载入寄存器 TL1 中。如果定时器中断使能，当 TF1 置 1 时将产生一个中断，但在 TH1 中的重载值不会改变。在允许定时器正确计数开始之前，TL1 必须初始化为所需要的值。

除了自动重载功能外，工作模式 2 中的计数器/定时器的使能和配置方式同方式 0 和 1 是相同的。

当作为定时器应用时，可配置寄存器 TMCON.4(T1FD)来选择定时器时钟源被系统时钟 $f_{SYS}$ 分频的比例。




定时器/计数器工作模式 2: 自动重载的 8 位定时器/计数器

Preliminary

# 11 定时器 Timer2/3/4

SC95F767X 单片机内部的 Timer2/3/4 是三个独立的 Timer，其中 Timer2 有 4 种工作模式，Timer3 和 Timer4 有 3 种工作模式。

Timer2/3/4 的控制寄存器共用同一组地址（C8H-CDH），用户可通过 TXINX[2:0]将 TimerX 寄存器组(TXCON / TXMOD / RCAPXL / RCAPXH / TLX / THX)指向 Timer2/3/4，从而实现一组寄存器配置三个独立 Timer 的功能。

**注意：只有在 TXINX[2:0]配置成功后 TimerX 寄存器组才会指向用户指定的 Timer2/3/4，此时操作 TimeX 寄存器组才是对相应 Timer 的有效操作。**

## 11.1 T2/3/4 相关特殊功能寄存器

| 符号 | 地址 | 说明 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Reset 值 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TXINX | CEH | 定时器 2/3/4 控制寄存器指针 | - | - | - | - | - | TXINX[2:0] | | | xxxxx010b |
| TXCON | C8H | 定时器 2/3/4 控制寄存器 | TFX | EXFX | RCLKX | TCLKX | EXENX | TRX | C/TX | CP/RLX | 00000000b |
| TXMOD | C9H | 定时器 2/3/4 工作模式寄存器 | TXFD | - | EPWMN1 | EPWMN0 | INVN1 | INVN0 | TXOE | DCXEN | 0xxxxx00b |
| RCAPXL | CAH | 定时器 2/3/4 重载低 8 位 | RCAPXL[7:0] | | | | | | | | 00000000b |
| RCAPXH | CBH | 定时器 2/3/4 重载高 8 位 | RCAPXH[7:0] | | | | | | | | 00000000b |
| TLX | CCH | 定时器 2/3/4 低 8 位 | TLX[7:0] | | | | | | | | 00000000b |
| THX | CDH | 定时器 2/3/4 高 8 位 | THX[7:0] | | | | | | | | 00000000b |

**TXINX (CEH) 定时器 2/3/4 控制寄存器指针(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | - | TXINX[2:0] | | |
| 读/写 | - | - | - | - | - | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | x | x | x | x | 0 | 1 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 2~0 | **TXINX[2:0]** | 定时器 2/3/4 控制寄存器指针<br>010：TimerX 寄存器组：TXCON / TXMOD / RCAPXL / RCAPXH / TLX / THX 指向 T2<br>011：TimerX 寄存器组指向 T3<br>100：TimerX 寄存器组指向 T4<br>其他：保留 |
| 7~3 | **-** | 保留 |

## 11.2 定时器 Timer2

SC95F767X 单片机内部的 Timer2 具有计数方式和定时方式两种工作模式。特殊功能寄存器 TXCON 中有一个控制位 C/TX 来选择 T2 是定时器还是计数器。它们本质上都是一个加法计数器，只是计数的来源不同。定时器的来源为系统时钟或者其分频时钟，但计数器的来源为外部管脚的输入脉冲。TRX 是 T2/T3/T4 在定时器/计数器模式计数的开关控制，只有在 TRX=1 的时候，T2 才会被打开计数。

计数器模式下，T2 管脚上的每一个脉冲，T2 的计数值分别增加 1。

定时器模式下，可通过特殊功能寄存器 TXMOD.7(TXFD)来选择 T2 的计数来源是 $f_{SYS}/12$ 或 $f_{SYS}$。

定时器/计数器 T2 有 4 种工作模式：

① 模式 0：16 位捕获模式

② 模式 1：16 位自动重载定时器模式

③ 模式 2： 波特率发生器模式

④ 模式 3：可编程时钟输出模式

TXINX[2:0] = 010，TimerX 寄存器组指向 Timer2，各寄存器的解释说明如下：

Preliminary

**TXCON (C8H) 定时器 2 控制寄存器(读/写) ( TXINX[2:0] = 010)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TFX | EXFX | RCLKX | TCLKX | EXENX | TRX | C/TX | CP/RLX |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **TFX** | 定时器 2 溢出标志位<br>0：无溢出(必须由软件清 0)<br>1：溢出(如果 RCLKX = 0 和 TCLKX = 0，由硬件设 1) |
| 6 | **EXFX** | T2EX 引脚外部事件输入(下降沿)被检测到的标志位<br>0：无外部事件输入(必须由软件清 0)<br>1：检测到外部输入(如果 EXENX = 1，由硬件设 1) |
| 5 | **RCLKX** | UART0 接收时钟控制位<br>0：定时器 1 产生接收波特率<br>1：定时器 2 产生接收波特率 |
| 4 | **TCLKX** | UART0 发送时钟控制位<br>0：定时器 1 产生发送波特率<br>1：定时器 2 产生发送波特率 |
| 3 | **EXENX** | T2EX 引脚上的外部事件输入(下降沿)用作重载/捕获触发器允许/禁止控制:<br>0：忽略 T2EX 引脚上的事件<br>1：当定时器 2 不做为 UART0 时钟时，检测到 T2EX 引脚上一个下降沿，产生一个捕获或重载 |
| 2 | **TRX** | 定时器 2 开始/停止控制位<br>0：停止定时器 2/停止 PWM2 计数器<br>1：开始定时器 2/开启 PWM2 计数器 |
| 1 | **C/TX** | 定时器 2 定时器/计数器方式选定位<br>0：定时器方式，T2 引脚用作 I/O 端口<br>1：计数器方式 |
| 0 | **CP/RLX** | 捕获/重载方式选定位<br>0：16 位带重载功能的定时器/计数器<br>1：16 位带捕获功能的定时器/计数器，TXEX 为定时器 2 外部捕获信号输入口 |

**TXMOD (C9H) 定时器 2 工作模式寄存器(读/写) ( TXINX[2:0] = 010)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TXFD | - | EPWM21 | EPWM20 | INV21 | INV20 | TXOE | DCXEN |
| 读/写 | 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | x | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **TXFD** | T2 输入频率选择控制<br>0：T2 频率源自于 $f_{SYS}/12$<br>1：T2 频率源自于 $f_{SYS}$ |
| 1 | **TXOE** | 定时器 2 输出允许位<br>0：设置 T2 作为时钟输入或 I/O 端口<br>1：设置 T2 作为时钟输出 |
| 0 | **DCXEN** | 递减计数允许位<br>0：禁止定时器 2 作为递增/递减计数器，定时器 2 仅作为递增计数器<br>1：允许定时器 2 作为递增/递减计数器，T2EX 用来选择计数方向。 |
| 6 | **-** | 保留 |

Preliminary

**IE (A8H) 中断使能寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | EA | EADC | ET2 | EUART | ET1 | EINT1 | ET0 | EINT0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 5 | **ET2** | Timer2 中断使能控制<br>0： 关闭 Timer2 中断<br>1： 允许 Timer2 中断 |

**IP (B8H) 中断优先级控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | IPADC | IPT2 | IPUART | IPT1 | IPINT1 | IPT0 | IPINT0 |
| 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 5 | **IPT2** | Timer2 中断优先权<br>0：设定 Timer2 的中断优先权是 “低”<br>1：设定 Timer2 的中断优先权是 “高” |

## 11.3 定时器 Timer3

SC95F767X 单片机内部的 Timer3 作为定时器本质上都是一个加法计数器，定时器的时钟来源为系统时钟或者其分频时钟。TRX 是 T3 计数的开关控制，只有在 TRX=1 的时候，T3 才会被打开计数。

定时器模式下，可通过特殊功能寄存器 TXMOD.7(TXFD)来选择 T3 的计数来源是 $f_{SYS}/12$ 或 $f_{SYS}$。

TXINX[2:0] = 011，TimerX 寄存器组指向 Timer3，各寄存器的解释说明如下：

**TXCON (C8H) 定时器 3 控制寄存器(读/写) ( TXINX[2:0] = 011)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TFX | EXFX | - | - | EXENX | TRX | C/TX | CP/RLX |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **TFX** | 定时器 3 溢出标志位<br>0：无溢出(必须由软件清 0)<br>1：溢出(由硬件设 1) |
| 6 | **EXFX** | T3EX 引脚外部事件输入(下降沿)被检测到的标志位<br>0：无外部事件输入(必须由软件清 0)<br>1：检测到外部输入(如果 EXENX = 1，由硬件设 1) |
| 3 | **EXENX** | T3EX 引脚上的外部事件输入(下降沿)用作重载/捕获触发器允许/禁止控制:<br>0：忽略 T3EX 引脚上的事件<br>1：检测到 T3EX 引脚上一个下降沿，产生一个捕获或重载 |
| 2 | **TRX** | 定时器 3 开始/停止控制位<br>0：停止定时器 3/停止 PWM3 计数器<br>1：开始定时器 3/开启 PWM3 计数器 |
| 1 | **C/TX** | 定时器 3 定时器/计数器方式选定位<br>0：定时器方式，T3 引脚用作 I/O 端口<br>1：计数器方式 |

Preliminary

| | | |
|---|---|---|
| 0 | **CP/RLX** | 捕获/重载方式选定位<br>0：16 位带重载功能的定时器/计数器<br>1：16 位带捕获功能的定时器/计数器，TXEX 为定时器 3 外部捕获信号输入口 |
| 5~4 | **-** | 保留 |

**TXMOD (C9H) 定时器 3 工作模式寄存器(读/写) ( TXINX[2:0] = 011)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TXFD | - | EPWM31 | EPWM30 | INV31 | INV30 | TXOE | DCXEN |
| 读/写 | 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | x | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **TXFD** | T3 输入频率选择控制<br>0：T3 频率源自于 $f_{SYS}/12$<br>1：T3 频率源自于 $f_{SYS}$ |
| 1 | **TXOE** | 定时器 3 输出允许位<br>0：设置 T3 作为时钟输入或 I/O 端口<br>1：设置 T3 作为时钟输出 |
| 0 | **DCXEN** | 递减计数允许位<br>0：禁止定时器 3 作为递增/递减计数器，定时器 3 仅作为递增计数器<br>1：允许定时器 3 作为递增/递减计数器，T3EX 用来选择计数方向。 |
| 6 | **-** | 保留 |

**IE1 (A9H) 中断使能寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ET4 | ET3 | - | - | EINT2 | EBTM | EPWM | EUSCI0 |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 6 | **ET3** | Timer3 中断使能控制<br>0：关闭 Timer3 中断<br>1：允许 Timer3 中断 |

**IP1 (B9H) 中断优先级控制寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IPT4 | IPT3 | - | - | IPINT2 | IPBTM | IPPWM | IPUSCI0 |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 6 | **IPT3** | Timer3 中断优先权选择<br>0：Timer3 中断优先权为低<br>1：Timer3 中断优先权为高 |
| 5~4 | **-** | 保留 |

## 11.4 定时器 Timer4

SC95F767X 单片机内部的 Timer4 作为定时器本质上都是一个加法计数器，定时器的时钟来源为系统时钟或者其分频时钟。TRX 是 T4 计数的开关控制，只有在 TRX=1 的时候，T4 才会被打开计数。

Preliminary

定时器模式下，可通过特殊功能寄存器 TXMOD.7(TXFD)来选择 T4 的计数来源是 $f_{SYS}/12$ 或 $f_{SYS}$。
TXINX[2:0] = 100，TimerX 寄存器组指向 Timer4，各寄存器的解释说明如下：

**TXCON (C8H) 定时器 4 控制寄存器(读/写) ( TXINX[2:0] = 100)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TFX | EXFX | - | - | EXENX | TRX | C/TX | CP/RLX |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **TFX** | 定时器 4 溢出标志位<br>0：无溢出(必须由软件清 0)<br>1：溢出(由硬件设 1) |
| 6 | **EXFX** | T4EX 引脚外部事件输入(下降沿)被检测到的标志位<br>0：无外部事件输入(必须由软件清 0)<br>1：检测到外部输入(如果 EXENX = 1，由硬件设 1) |
| 3 | **EXENX** | T4EX 引脚上的外部事件输入(下降沿)用作重载/捕获触发器允许/禁止控制:<br>0：忽略 T4EX 引脚上的事件<br>1：检测到 T4EX 引脚上一个下降沿，产生一个捕获或重载 |
| 2 | **TRX** | 定时器 4 开始/停止控制位<br>0：停止定时器 4/停止 PWM4 计数器<br>1：开始定时器 4/开启 PWM4 计数器 |
| 1 | **C/TX** | 定时器 4 定时器/计数器方式选定位<br>0：定时器方式，T4 引脚用作 I/O 端口<br>1：计数器方式 |
| 0 | **CP/RLX** | 捕获/重载方式选定位<br>0：16 位带重载功能的定时器/计数器<br>1：16 位带捕获功能的定时器/计数器，TXEX 为定时器 4 外部捕获信号输入口 |
| 5~4 | **-** | 保留 |

**TXMOD (C9H) 定时器 4 工作模式寄存器(读/写) ( TXINX[2:0] = 100)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TXFD | - | EPWM41 | EPWM40 | INV41 | INV40 | TXOE | DCXEN |
| 读/写 | 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | x | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **TXFD** | T4 输入频率选择控制<br>0：T4 频率源自于 $f_{SYS}/12$<br>1：T4 频率源自于 $f_{SYS}$ |
| 1 | **TXOE** | 定时器 4 输出允许位<br>0：设置 T4 作为时钟输入或 I/O 端口<br>1：设置 T4 作为时钟输出 |
| 0 | **DCXEN** | 递减计数允许位<br>0：禁止定时器 4 作为递增/递减计数器，定时器 4 仅作为递增计数器<br>1：允许定时器 4 作为递增/递减计数器，T4EX 用来选择计数方向。 |
| 6 | **-** | 保留 |

**IE1 (A9H) 中断使能寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ET4 | ET3 | - | - | EINT2 | EBTM | EPWM | EUSCI0 |

Preliminary

| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
|---|---|---|---|---|---|---|---|---|
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **ET4** | Timer4 中断使能控制<br>0：关闭 Timer4 中断<br>1：允许 Timer4 中断 |

**IP1 (B9H) 中断优先级控制寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IPT4 | IPT3 | - | - | IPINT2 | IPBTM | IPPWM | IPUSCI0 |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **IPT4** | Timer4 中断优先权选择<br>0：Timer4 中断优先权为低<br>1：Timer4 中断优先权为高 |
| 5~4 | **-** | 保留 |

## 11.5 Timer2/3/4 工作模式

定时器 Timer2/3/4 的工作模式如下：

① 模式 0：16 位捕获

② 模式 1：16 位自动重载定时器

③ 模式 2： 波特率发生器，**仅 Timer2 支持该模式**

④ 模式 3：可编程时钟输出

⑤ 模式 4：PWM 输出模式

以上工作模式与配置方式如下表：

| C/TX | TXOE | DCXEN | TRX | CP/RLX | EXENX | 工作模式 | |
|---|---|---|---|---|---|---|---|
| X | 0 | X | 1 | 1 | 1 | 模式 0 | 16 位捕获 |
| X | 0 | 0 | 1 | 0 | 0 | 模式 1 | 16 位自动重载定时/计数器，普通自动重载 |
| X | 0 | 0 | 1 | 0 | 1 | | 16 位自动重载定时/计数器，带 TnEX 触发重载 |
| X | 0 | 1 | 1 | 0 | X | | 16 位自动重载定时/计数器，递增或递减重载 |
| X | 0 | X | 1 | X | X | 模式 2 | UART0 波特率发生器，仅 Timer2 支持该模式 |
| 0 | 1 | X | 1 | X | X | 模式 3 | 可编程时钟输出 |
| X | X | X | 0 | X | 1 | X | 定时器停止，TnEX(n=2~4)通路仍旧允许设置捕获/重载产生中断 |

### 11.5.1 Timer2/3/4 工作模式说明

**工作模式 0: 16 位捕获**

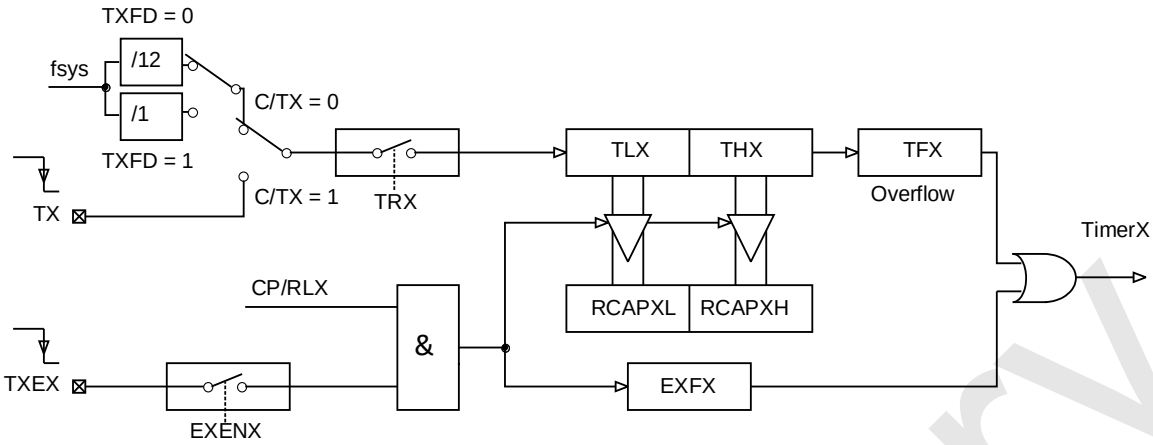
配置 CP/RLX =1，将定时器 n (n=2~4)设置为 16 位捕获模式。

在捕获方式中， TXCON 的 EXENX 位有两个选项：

如果 EXENX = 0，定时器 n 作为 16 位定时器或计数器，如果 ETn 被允许的话，定时器 n 能设置 TFX 溢出产生一个中断。

如果 EXENX = 1，定时器 n 执行相同操作，但是在外部输入 TnEX 上的下降沿也能引起在 THX 和 TLX 中的当前值分别被捕获到 RCAPXH 和 RCAPXL 中，此外，在 TnEX 上的下降沿也能引起在 TXCON 中的 EXFX 被设置。如果 ETn 被允许，EXFX 位也像 TFX 一样也产生一个中断。

Preliminary




模式 0: 16 位捕获

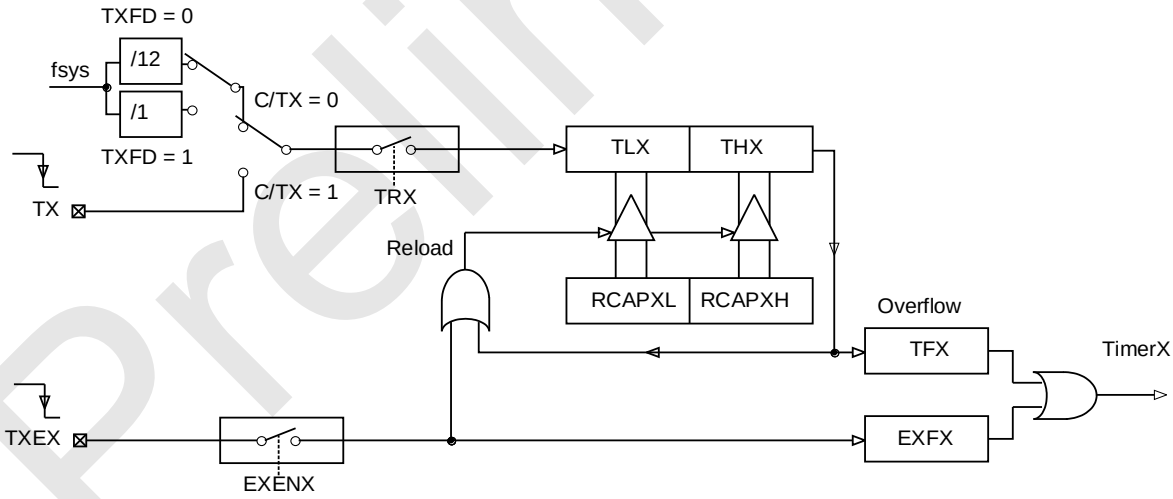
**工作模式 1: 16 位自动重载定时器**

在 16 位自动重载方式下，定時器 n (n=2~4)可以被选为递增计数或递减计数。这个功能通过 TnMOD 中的 DCEN 位(递减计数允许)选择。系统复位后，DCEN 位复位值为 0，定时器 n 默认递增计数。当 DCEN 置 1 时，定时器 n 递增计数或递减计数取决于 TnEX 引脚上的电平。

当 DCEN = 0，通过在 TXCON 中的 EXENX 位选择两个选项。

如果 EXENX = 0，定时器 n 递增到 0xFFFFH，在溢出后置起 TFX 位，同时定时器自动将用户软件写好的寄存器 RCAPXH 和 RCAPXL 的 16 位值装入 THX 和 TLX 寄存器。

如果 EXENX = 1，溢出或在外部输入 TnEX 上的下降沿都能触发一个 16 位重载。TnEX 上有下降沿产生时，EXFX 位置起。如果 ETn 被使能，TFX 和 EXFX 位都能产生一个中断。




模式 1: 16 位自动重载 DCEN = 0

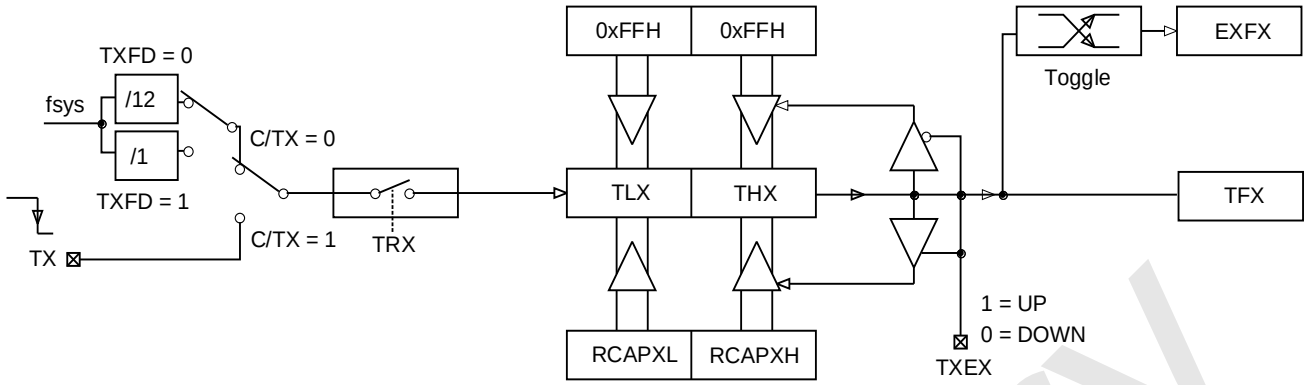
设置 DCEN 位允许定时器 n 递增计数或递减计数。当 DCEN = 1 时，TnEX 引脚控制计数的方向，而 EXENX 控制无效。

TnEX 置 1 可使定时器 n 递增计数。定时器向 0xFFFFH 溢出，然后设置 TFX 位。溢出也能分别引起 RCAPXH 和 RCAPXL 上的 16 位值重载入定时器寄存器。

TnEX 置 0 可使定时器 n 递减计数。当 THX 和 TLX 的值等于 RCAPXH 和 RCAPXL 的值时，定时器溢出。置起 TFX 位，同时 0xFFFFH 重载入定时器寄存器。

无论定时器 n 溢出与否，EXFX 位都被用作结果的第 17 位。在此工作方式下，EXFX 不作为中断标志。

Preliminary




模式 1: 16 位自动重载 DCEN = 1

**工作模式 2: 波特率发生器，仅 Timer2 支持**

通设置 TXCON 寄存器中的 TCLK 和/或 RCLK 选择定时器 2 作为波特率发生器。接收器和发送器的波特率可以不同。如果定时器 2 作为接收器或发送器，则定时器 1 相应的作为另一种的波特率发生器

设置 TXCON 寄存器中的 TCLK 和/或 RCLK 使定时器 2 进入波特率发生器方式，该方式与自动重载方式相似

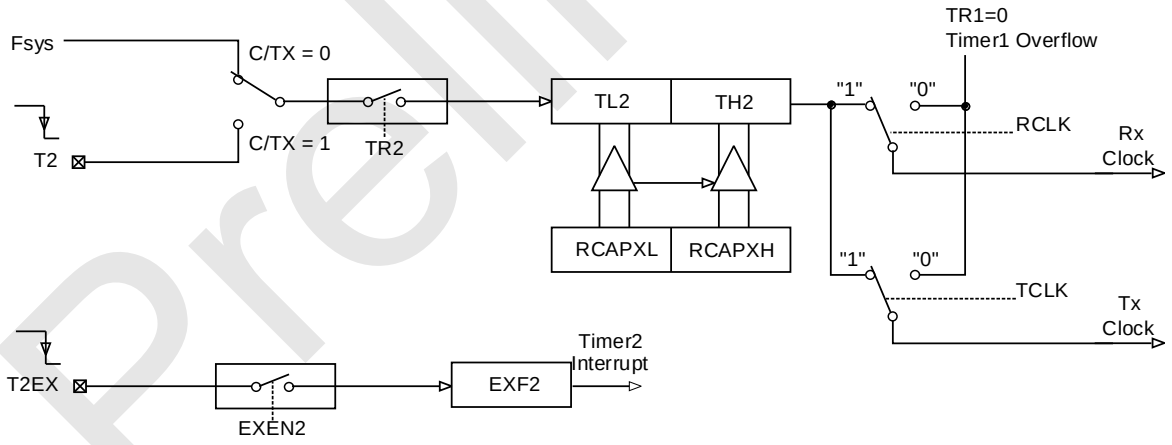
定时器 2 的溢出会使 RCAPXH 和 RCAPXL 寄存器中的值重载入定时器 2 计数，但不会产生中断

如果 EXENX 被置 1，在 T2EX 脚上的下降沿会置起 EXFX，但不会引起重载。因此当定时器 2 作为波特率发送器时，T2EX 可作为一个额外的外部中断

在 UART0 方式 1 和 3 中的波特率由定时器 2 的溢出率根据下列方程式决定：

$$\text{BaudRate} = \frac{\text{fsys}}{[\text{RCAPXH,RCAPXL}]};$$ (注意：[RCAPXH,RCAPXL] 必须大于 0x0010)

定时器 2 作为波特率发生器的原理图如下：




模式 2：波特率发生器

**工作模式 3: 可编程时钟输出**

在这种方式中，定时器 n (n=2~4)可以编程为输出 50%的占空比时钟周期：当 C/$\overline{\text{Tn}}$ = 0；TnOE = 1，使能定时器 n 作为时钟发生器

在这种方式中，Tn 输出占空比为 50%的时钟

$$\text{Colck Out Frequency} = \frac{\text{fn}}{(65536-[\text{RCAPXH,RCAPXL}])\times 4};$$

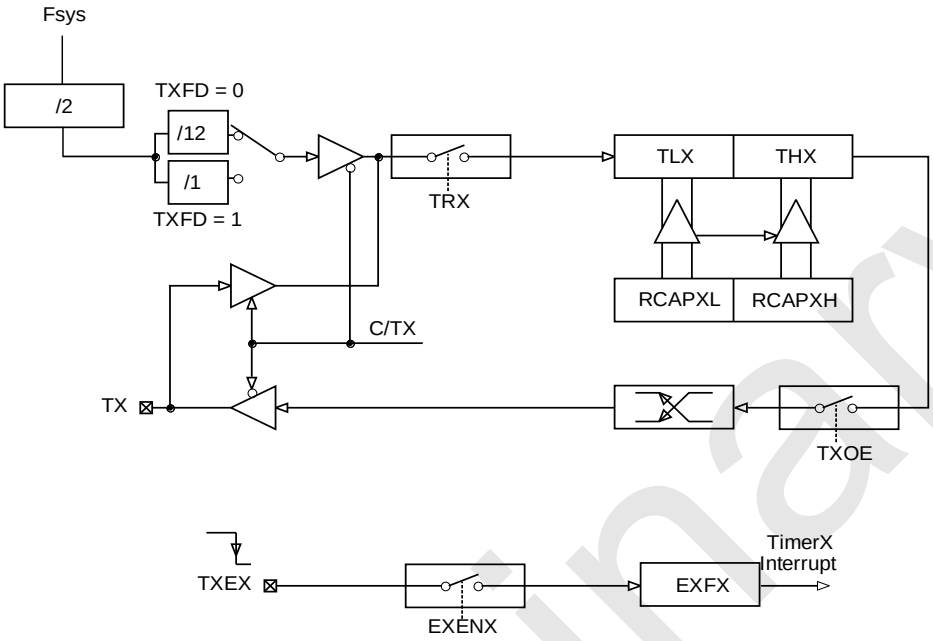
其中， fn 为定时器 n 时钟频率：

$$\text{fn} = \frac{\text{fsys}}{12}; \qquad \text{TXFD} = 0$$

Preliminary

fn = fsys;　　TXFD = 1

定时器 n 溢出不产生中断，Tn 端口作时钟输出。




模式 3：可编程时钟输出

**注意：**

1. TFX 和 EXFX 都能引起定时器 n (n=2~4)的中断请求，两者有相同的向量地址；
2. 当事件发生时或其它任何时间都能由软件设置 TFX 和 EXFX 为 1，只有软件以及硬件复位才能使之清 0；
3. 当 EA = 1 且 ETn = 1 时，设置 TFX 或 EXFX 为 1 能引起定时器 n 中断；
4. 当定时器 2 作为波特率发生器时，写入 THX/TLX 或 RCAPXH/RCAPXL 会影响波特率的准确性，引起通信出错。

Preliminary

# 12 常规脉冲宽度调制计数器 PWM2/3/4

SC95F767X 最多提供 14 路 PWM，这 14 路 PWM 分为两类：

1. 多功能 PWM：共 8 路，只有一组，即 PWM0，输出信号口为：PWM00~07；
2. 常规 PWM：共 6 路，分为三组：PWM2、PWM3、PWM4。**注意：这三组 PWM 的周期寄存器分别与 Timer2，Timer3，Timer4 的 TLX 和 THX 共用，因此一旦用户使用了 PWM2、PWM3、PWM4 资源，就不能再更改 Timer2，Timer3，Timer4 的定时/计数值，否则会导致 PWM 周期输出异常！**

## 12.1 PWM2/3/4 相关寄存器

PWM2/3/4 相关寄存器如下：

**TXINX (CEH) 定时器 2/3/4 控制寄存器指针(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | - | TXINX[2:0] | | |
| 读/写 | - | - | - | - | - | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | x | x | x | x | 0 | 1 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 2~0 | **TXINX[2:0]** | 定时器 2/3/4 控制寄存器指针<br>010：TimerX 寄存器组：TXCON / TXMOD / RCAPXL / RCAPXH / TLX / THX 指向 PWM2<br>011：TimerX 寄存器组指向 PWM3<br>100：TimerX 寄存器组指向 PWM4<br>其他：保留 |
| 7~3 | **-** | 保留 |

**TXCON (C8H) 定时器 n 控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TFX | EXFX | RCLKX | TCLKX | EXENX | TRX | C/TX | CP/RLX |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 2 | **TRX** | 定时器 n 开始/停止控制位<br>0：停止定时器 n/停止 PWMn 计数器<br>1：开始定时器 n/开启 PWMn 计数器 |

当 EPWMn0 或 EPWMn1 置 1 时 Timer 即可开启 PWM 模式，此时 Tn 和 TnEX（n=2~4）无效，PWMxy（x=2~4，y=0~1）可输出 PWM 波形。

**TXMOD (C9H) 定时器 n 工作模式寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TXFD | - | EPWMn1 | EPWMn0 | INVn1 | INVn0 | TXOE | DCXEN |
| 读/写 | 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | x | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 5 | **ENPWMn1** | PWMn1 波形输出选择<br>0：PWMn1 输出被关闭<br>1：PWMn1 所在的 I/O 作为 PWM 波形输出口 |
| 4 | **ENPWMn0** | PWMn0 波形输出选择 |

Preliminary

| | | |
|---|---|---|
| | | 0：PWMn0 输出被关闭<br>1：PWMn0 所在的 I/O 作为 PWM 波形输出口 |
| 3 | **INVn1** | PWMn1 波形输出反向控制<br>1：PWMn1 波形输出反向<br>0：PWMn1 波形输出不反向 |
| 2 | **INVn0** | PWMn0 波形输出反向控制<br>1：PWMn0 波形输出反向<br>0：PWMn0 波形输出不反向 |

THX 和 TLX 计数器从 0 开始向上计数，当计数值与占空比设置项 PDTxy [15:0]的值匹配时 PWM 输出波形切换高低电平，接着 THX 和 TLX 计数器继续向上计数到自动重载值 PWMPDX，然后重新从 0 开始计数并生成计数上溢事件，一个 PWM 周期结束。如果定时器中断已使能，此时会产生定时中断。

Timer 输出的 PWM 周期 $T_{PWM}$ 计算公式如下：

$$\mathrm{Tpwm} = \frac{\mathrm{PWMPDX}[15:0] + 1}{\mathrm{fsys}}$$

占空比 duty 计算公式：

$$\mathrm{duty} = \frac{\mathrm{PDTxy}\ [15:0]}{\mathrm{PWMPDX}[15:0] + 1}$$

PWM 周期通过以下寄存器设定：

**RCAPXH (CBH)**

**PWMn 周期寄存器高 8 位(读/写)**

**注意：PWM2/3/4 的周期寄存器与 Timer2，Timer3，Timer4 复用，因此，用户一旦使用了 PWM2、PWM3、PWM4 资源，就不能再更改 Timer2，Timer3，Timer4 的定时/计数值，否则会导致 PWM 周期输出异常！**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | PWMPDHX[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**RCAPXL (CAH)**

**PWMn 周期寄存器低 8 位(读/写) ( TXINX[2:0] = 010)**

**注意：PWM2/3/4 的周期寄存器与 Timer2，Timer3，Timer4 复用，因此，用户一旦使用了 PWM2、PWM3、PWM4 资源，就不能再更改 Timer2，Timer3，Timer4 的定时/计数值，否则会导致 PWM 周期输出异常！**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | PWMPDLX[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **PWMPDX[15:0]** | PWMn 周期设置<br>此数值代表 PWMn 的输出波形的 (周期 – 1); 也就是说 PWMn 输出的周期值为（PWMPDX[15:0] + 1 ）* PWM 时钟； |

PWM 的 duty 通过以下寄存器设定：

**PWM2~4 占空比调节寄存器(读/写)**

| 地址 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 上电初始值 |
|---|---|---|---|---|---|---|---|---|---|
| 1034H | PDT20[15:8] | | | | | | | | 00000000b |
| 1035H | PDT20[7:0] | | | | | | | | 00000000b |
| 1036H | PDT21[15:8] | | | | | | | | 00000000b |
| 1037H | PDT21[7:0] | | | | | | | | 00000000b |
| 1038H | PDT30[15:8] | | | | | | | | 00000000b |
| 1039H | PDT30[7:0] | | | | | | | | 00000000b |
| 103AH | PDT31[15:8] | | | | | | | | 00000000b |
| 103BH | PDT31[7:0] | | | | | | | | 00000000b |

Preliminary

| | | |
|---|---|---|
| 103CH | PDT40[15:8] | 00000000b |
| 103DH | PDT40[7:0] | 00000000b |
| 103EH | PDT41[15:8] | 00000000b |
| 103FH | PDT41[7:0] | 00000000b |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **PDTxy[15:0]**<br>**（x=2~4，y=0~1）** | PWMxy 波形占空比长度设置<br>PWMxy 的波形的高电平宽度 为：<br>(PDTxy[15:0] + 1 )个 PWM 时钟 |

## 12.2 PWM2/3/4 占空比变化特性

当 PWM2/3/4 输出波形时，若需改变占空比，可通过改变高电平设置寄存器 PDTxy（x=2~4，y=0~1）的值实现。但需要注意:更改 PDTxy 的值，占空比不会立即改变，而是等待本周期结束，在下个周期改变。

## 12.3 PWM2/3/4 周期变化特性

当 PWM2/3/4 输出波形时，若需改变周期，可通过改变周期设置寄存器组 TLX 和 THX 的值实现。更改周期寄存器的值，PWM 输出周期变化情况如下：

定义当前周期计数值为 Tn，写入周期寄存器时，定时器计到的值为 Tm，待更新的周期计数值为 Tx，则：

Tm ≤ Tx：周期按照 Tx 实时改变；

Tm ＞ Tx：此时周期变化会分为两个阶段。第一个阶段，写入周期寄存器之后，周期计数器会从当前计数值累加至溢出清零。第二个阶段，周期按照 Tx 改变。

# 13 多功能脉冲宽度调制计数器 PWM0

SC95F767X 提供 14 路 PWM，这 14 路 PWM 分为两类：

1. 多功能 PWM：PWM0，共 8 路：PWM00~07；
2. 常规 PWM：共 6 路，分为三组：PWM2、PWM3、PWM4。**注意：这三组 PWM 的周期寄存器分别与 Timer2，Timer3，Timer4 的 TLX 和 THX 共用，因此一旦用户使用了 PWM2、PWM3、PWM4 资源，就不能再更改 Timer2，Timer3，Timer4 的定时/计数值，否则会导致 PWM 周期输出异常！**

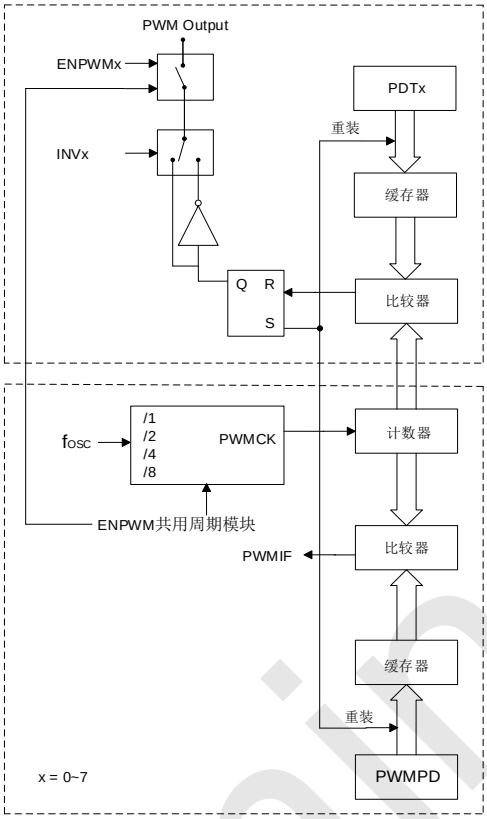
SC95F767X 的 PWM0 具有的功能如下：

1. 16 位 PWM 精度；
2. 输出波形可反向；
3. 类型：可设为中心对齐型或边沿对齐型；
4. 模式：可设为独立模式或互补模式：
   a) 独立模式下，8 路 PWM 周期相同，但每一路 PWM 输出波形的占空比单独可设置；
   b) 互补模式下可同时输出四组互补、带死区的 PWM 波形；
5. 提供 1 个 PWM 溢出的中断；
6. 支持故障检测机制。

SC95F767X 的 PWM 可支持周期及占空比的调整，寄存器 PWMCFG、PWMCON0 和 PWMCON1 控制 PWM 的状态及周期，各路 PWM 的打开及输出波形占空比可单独调整。

## 13.1 PWM0 结构框图

Preliminary




SC95F767X PWM 结构框图

## 13.2 PWM0 通用配置寄存器

### 13.2.1 PWM0 通用配置寄存器

用户可以通过配置 PWMMD[1:0]将 SC95F767X 的 PWM 输出模式设为独立模式或互补模式。独立模式下 8 路 PWM 周期相同，但每一路 PWM 输出波形的占空比单独可设置。互补模式下可同时输出四组互补、带死区的 PWM 波形。

SC95F767X 的 PWM 类型分为边沿对齐型和中心对齐型：

**边沿对齐型：**

PWM 计数器从 0 开始向上计数，当计数值与占空比设置项 PDTx [15:0]的值匹配时 PWM 输出波形切换高低电平，接着 PWM 计数器继续向上计数直至与周期设置项 PWMPD[15:0] +1 的值匹配（一个 PWM 周期结束），PWM 计数器清零，如果 PWM 中断已使能，此时会产生 PWM 中断。PWM 输出波形为左边沿对齐方式。

边沿对齐型周期 $T_{PWM}$ 计算公式：

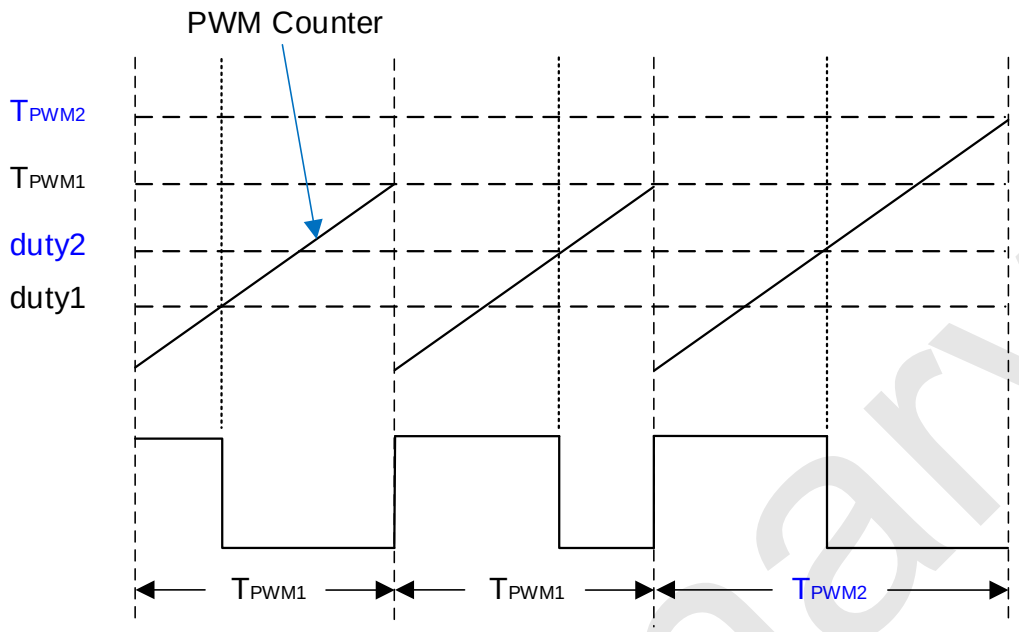
$$\mathrm{Tpwm} = \frac{\mathrm{PWMPD}[15:0] + 1}{\text{PWM 时钟频率}}$$

边沿对齐型占空比 duty 计算公式：

$$\mathrm{duty} = \frac{\mathrm{PDTx}\,[15:0]}{\mathrm{PWMPD}[15:0] + 1}$$

边沿对齐波形图如下：

Preliminary




边沿对齐的 PWM

**中心对齐型：**

PWM 计数器从 0 开始向上计数，当计数值与占空比设置项 PDTx [15:0]的值匹配时 PWM 输出波形切换高低电平，接着 PWM 计数器继续向上计数，当计数值与周期设置项 PWMPD[15:0] +1 的值匹配时（即 PWM 周期的中点）自动开始向下计数，当计数值与 PDTx [15:0]的值再次匹配时 PWM 输出波形再次切换高低电平，接着 PWM 计数器继续向下计数直至溢出（一个 PWM 周期结束），如果 PWM 中断已使能，此时会产生 PWM 中断。

中心对齐型周期 $T_{PWM}$ 计算公式：

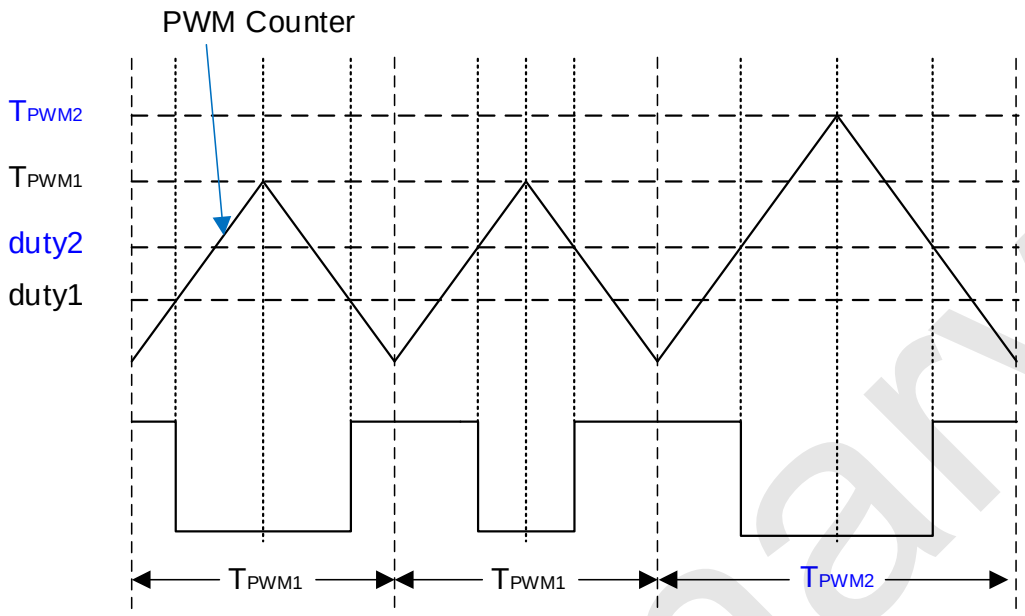
$$\mathrm{Tpwm} = 2 * \frac{\mathrm{PWMPD}[15:0] + 1}{\text{PWM 时钟频率}}$$

中心对齐型占空比 duty 计算公式：

$$\mathrm{duty} = \frac{\mathrm{PDTx}\ [15:0]}{\mathrm{PWMPD}[15:0] + 1}$$

中心对齐波形图如下：

Preliminary




中心对齐的 PWM

以上模式及类型可通过 PWMCON0 寄存器设置：

**PWMCON0 (D2H) PWM 控制寄存器 0(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ENPWM | PWMIF | PWMCK[1:0] | | - | - | PWMMD[1:0] | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | x | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **ENPWM** | PWM 模块开关控制(Enable PWM)<br>1：允许 Clock 进到 PWM 单元，PWM 处于工作状态，PWM 输出口的状态由寄存器 ENPWMx 控制（x=0~7）<br>0：PWM 单元停止工作，PWM 计数器清零，全部 PWM 输出口设置为 GPIO 状态 |
| 6 | **PWMIF** | PWM 中断请求标志位(PWM Interrupt Flag)<br>当 PWM 计数器溢出时(也就是说: 数到超过 PWMPD 时)， 此位会被硬件自动设定成 1。如果此时 IE1[1] (EPWM) 也是被设定成 1， PWM 的中断产生 。在 PWM 中断发生后，硬件并不会自动清除此位，此位必须由使用者的软件负责清除。 |
| 5~4 | **PWMCK[1:0]** | PWM 时钟档位选择(PWM Clock Source Selector)<br>00：fosc<br>01：fosc/2<br>10：fosc/4<br>11：fosc/8 |
| 1~0 | **PWMMD[1:0]** | PWM 工作模式设置<br>0x：独立模式<br>1x：互补模式<br>x0：边沿对齐模式<br>x1：中心对齐模式 |
| 3~2 | **-** | 保留 |

Preliminary

**PWMCFG (D1H) PWM 设置寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | INV7 | INV6 | INV5 | INV4 | INV3 | INV2 | INV1 | INV0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 6 | **INVx**<br>**（x=0~7）** | PWMx 波形输出反向控制<br>1：PWMx 波形输出反向<br>0：PWMx 波形输出不反向 |

**PWMCON1 (D3H) PWM 控制寄存器 1 (读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | EPWM7 | EPWM6 | EPWM5 | EPWM4 | EPWM3 | EPWM2 | EPWM1 | EPWM0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **ENPWMx**<br>**（x=0~7）** | PWMx 波形输出选择<br>0：PWMx 输出被关闭并作为 GPIO 口 [1]<br>1：当 ENPWM=1 时，PWMx 所在的 I/O 作为波形输出口 |

注：

1. 如果 ENPWM 置 1，PWM 模块被打开，但 ENPWMx=0，PWM 输出被关闭并作为 GPIO 口。此时 PWM 模块可以作为一个 16 位 Timer 使用，此时 EPWM(IE1.1)被置 1，PWM 仍然会产生中断。

**PWMPDL (D4H) 周期寄存器低 8 位(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | PWMPDL[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

PWMPDH (D5H) 周期寄存器高 8 位(读/写)

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | PWMPDH[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **PWMPD[15:0]** | PWM 共用的周期设置<br>此数值代表 PWM 输出波形的 (周期 – 1); 也就是说 PWM 输出的周期值为（PWMPD[15:0] + 1 ）* PWM 时钟; |

**IE1 (A9H) 中断使能寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ET4 | ET3 | - | - | EINT2 | EBTM | EPWM | EUSCI0 |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 1 | **EPWM** | PWM 中断使能控制<br>0：关闭 PWM 中断<br>1：允许 PWM 计数器溢出时产生中断 |

Preliminary

**IP1 (B9H) 中断优先权寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | IPT4 | IPT3 | - | - | IPINT2 | IPBTM | IPPWM | IPUSCI0 |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 1 | **IPPWM** | PWM 中断优先权选择<br>0：设定 PWM 的中断优先级是 “低”<br>1：设定 PWM 的中断优先级是 “高” |

## 13.2.2 PWM0 故障检测功能设置

SC95F767X 系列支持故障检测机制。故障检测功能常应用于电机系统的防护。故障检测功能常应用于电机系统的防护。当故障检测功能开启， FLTEN1 (PWMFLT.7)置 1，故障检测信号输入脚(FLT) 生效。当 FLT 管脚的信号满足故障条件，标志位 FLTSTA1 通过硬件置 1， PWM 输出停止，PWM 计数器仍保持计数，PWM 中断不受影响。故障检测模式分为锁存模式和立即模式；立即模式下，当 FLT 管脚上的故障信号满足失能条件，标志位 FLTSTA1 通过硬件清零，直到 PWM 计数器归零后 PWM 恢复输出；锁存模式下，当 FLT 管脚上的故障信号满足失能条件，标志位 FLTSTA1 状态保持不变，用户可通过软件清零，FLTSTA1 状态一旦清零，PWM 计数器恢复计数，直到 PWM 计数器归零后 PWM 恢复输出。故障检测模式分为锁存模式和立即模式具体配置方式如下：

**PWMFLT (D7H) PWM 故障检测设置寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | FLTEN1 | FLTSTA1 | FLTMD1 | FLTLV1 | - | - | FLTDT1[1:0] | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | x | x | 0 | 0 |

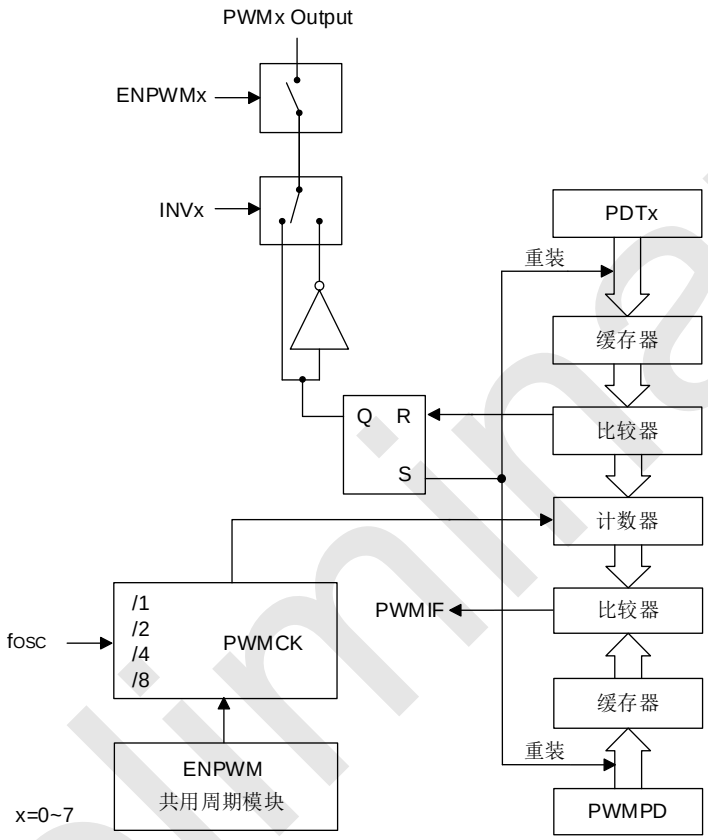
| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **FLTEN1** | PWM 故障检测功能控制位<br>0：故障检测功能关闭<br>1：故障检测功能开启 |
| 6 | **FLTSTA1** | PWM 故障检测状态标志位<br>0：PWM 处于正常输出状态；<br>1：故障检测有效，PWM 输出处于高阻状态，如果处于锁存模式，此位可软件清 0 |
| 5 | **FLTMD1** | PWM 故障检测模式设置位<br>0：锁存模式，当故障输入有效时，FLTSTA1 被置“1”，PWM 停止输出，当故障输入无效时 FLTSTA1 状态不变<br>1：立即模式：当故障输入有效时，FLTSTA1 被置“1”，PWM 停止输出，当故障输入无效时 FLTSTA1 状态立刻被清零，PWM 波形将在 PWM 计数器计数到 0 时恢复输出 |
| 4 | **FLTLV1** | PWM 故障检测电平选择位<br>0：故障检测低电平有效<br>1：故障检测高电平有效 |
| 1~0 | **FLTDT1[1:0]** | PWM 故障检测输入信号滤波时间设置<br>00：滤波时间为 0<br>01：滤波时间为 1us<br>10：滤波时间为 4us<br>11：滤波时间为 16us |
| 3~2 | **-** | 保留 |

Preliminary

## 13.3 PWM0 独立模式

独立模式下(PWMMD.1= 0)，8 路 PWM 通道的占空比均可独立设置。用户配置好 PWM 的输出状态及周期，再通过配置相应 PWM 通道的占空比寄存器即可按固定占空比输出 PWM 波形。

### 13.3.1 PWM0 独立模式框图




SC95F767X PWM 独立模式框图

Preliminary

### 13.3.2 PWM0 独立模式占空比配置

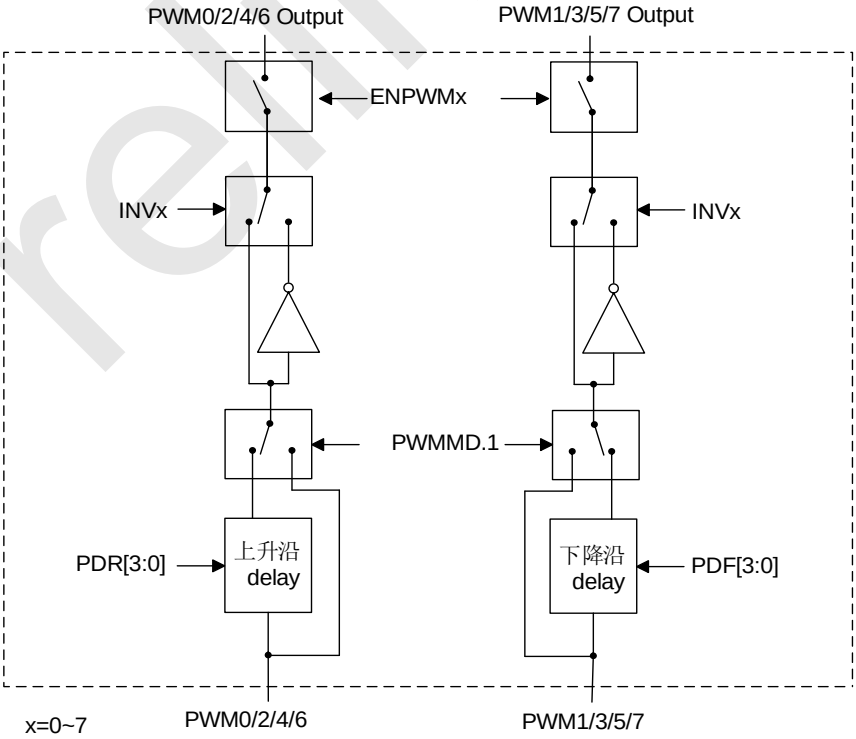
**PWM0 占空比调节寄存器 PDT0x (读/写)**

| 地址 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 上电初始值 |
|---|---|---|---|---|---|---|---|---|---|
| 1040H | PDT00[15:8] | | | | | | | | 00000000b |
| 1041H | PDT00[7:0] | | | | | | | | 00000000b |
| 1042H | PDT01[15:8] | | | | | | | | 00000000b |
| 1043H | PDT01[7:0] | | | | | | | | 00000000b |
| 1044H | PDT02[15:8] | | | | | | | | 00000000b |
| 1045H | PDT02[7:0] | | | | | | | | 00000000b |
| 1046H | PDT03[15:8] | | | | | | | | 00000000b |
| 1047H | PDT03[7:0] | | | | | | | | 00000000b |
| 1048H | PDT04[15:8] | | | | | | | | 00000000b |
| 1049H | PDT04[7:0] | | | | | | | | 00000000b |
| 104AH | PDT05[15:8] | | | | | | | | 00000000b |
| 104BH | PDT05[7:0] | | | | | | | | 00000000b |
| 104CH | PDT06[15:8] | | | | | | | | 00000000b |
| 104DH | PDT06[7:0] | | | | | | | | 00000000b |
| 104EH | PDT07[15:8] | | | | | | | | 00000000b |
| 104FH | PDT07[7:0] | | | | | | | | 00000000b |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 3~0 | **PDT0x [15:8]**<br>**（x=0~7）** | PWMx 波形占空比长度设置<br>PWMx 的波形的高电平宽度 是 (PDT0x [15:0])个 PWM 时钟 |

## 13.4 PWM0 互补模式

### 13.4.1 PWM0 互补模式框图




SC95F767X PWM 互补模式框图

Preliminary

### 13.4.2 PWM0 互补模式占空比配置

互补模式下(PWMMD[1:0] = 1x)，PWM0/PWM1, PWM2/PWM3, PWM4/PWM5 和 PWM6/PWM7 分为四组，分别通过 PDT00[15:0]、PDT02[15:0]、PDT04[15:0]和 PDT06[15:0]调节占空比；

**互补模式下寄存器 PDT01[15:0]、PDT03[15:0]、PDT05[15:0]和 PDT07[15:0]无效。**

**PWM0 占空比调节寄存器 PDT0x(读/写)**

| 地址 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 上电初始值 |
|---|---|---|---|---|---|---|---|---|---|
| 1040H | PDT00[15:8] | | | | | | | | 00000000b |
| 1041H | PDT00[7:0] | | | | | | | | 00000000b |
| 1042H | PDT01[15:8] | | | | | | | | 00000000b |
| 1043H | PDT01[7:0] | | | | | | | | 00000000b |
| 1044H | PDT02[15:8] | | | | | | | | 00000000b |
| 1045H | PDT02[7:0] | | | | | | | | 00000000b |
| 1046H | PDT03[15:8] | | | | | | | | 00000000b |
| 1047H | PDT03[7:0] | | | | | | | | 00000000b |
| 1048H | PDT04[15:8] | | | | | | | | 00000000b |
| 1049H | PDT04[7:0] | | | | | | | | 00000000b |
| 104AH | PDT05[15:8] | | | | | | | | 00000000b |
| 104BH | PDT05[7:0] | | | | | | | | 00000000b |
| 104CH | PDT06[15:8] | | | | | | | | 00000000b |
| 104DH | PDT06[7:0] | | | | | | | | 00000000b |
| 104EH | PDT07[15:8] | | | | | | | | 00000000b |
| 104FH | PDT07[7:0] | | | | | | | | 00000000b |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **PDT0x [7:0]**<br>**（x=0,2,4,6）** | PWMx 和 PWMy，y=x+1 口 PWM 波形占空比长度设置<br>Px 和 Py 管脚上的 PWM 波形的高电平宽度 是 (PDTx [15:0])个 PWM 时钟 |

### 13.4.3 PWM0 互补模式死区时间设置

当 SC95F767X 的 PWM0 工作在互补模式时，死区控制模块能够防止互补输出的两路 PWM 信号有效时区的互相交叠，以保证实际应用中 PWM 信号驱动的一对互补功率开关管不会同时导通。

**PWMDFR (D6H) PWM 死区时间设置寄存器(读/写)**

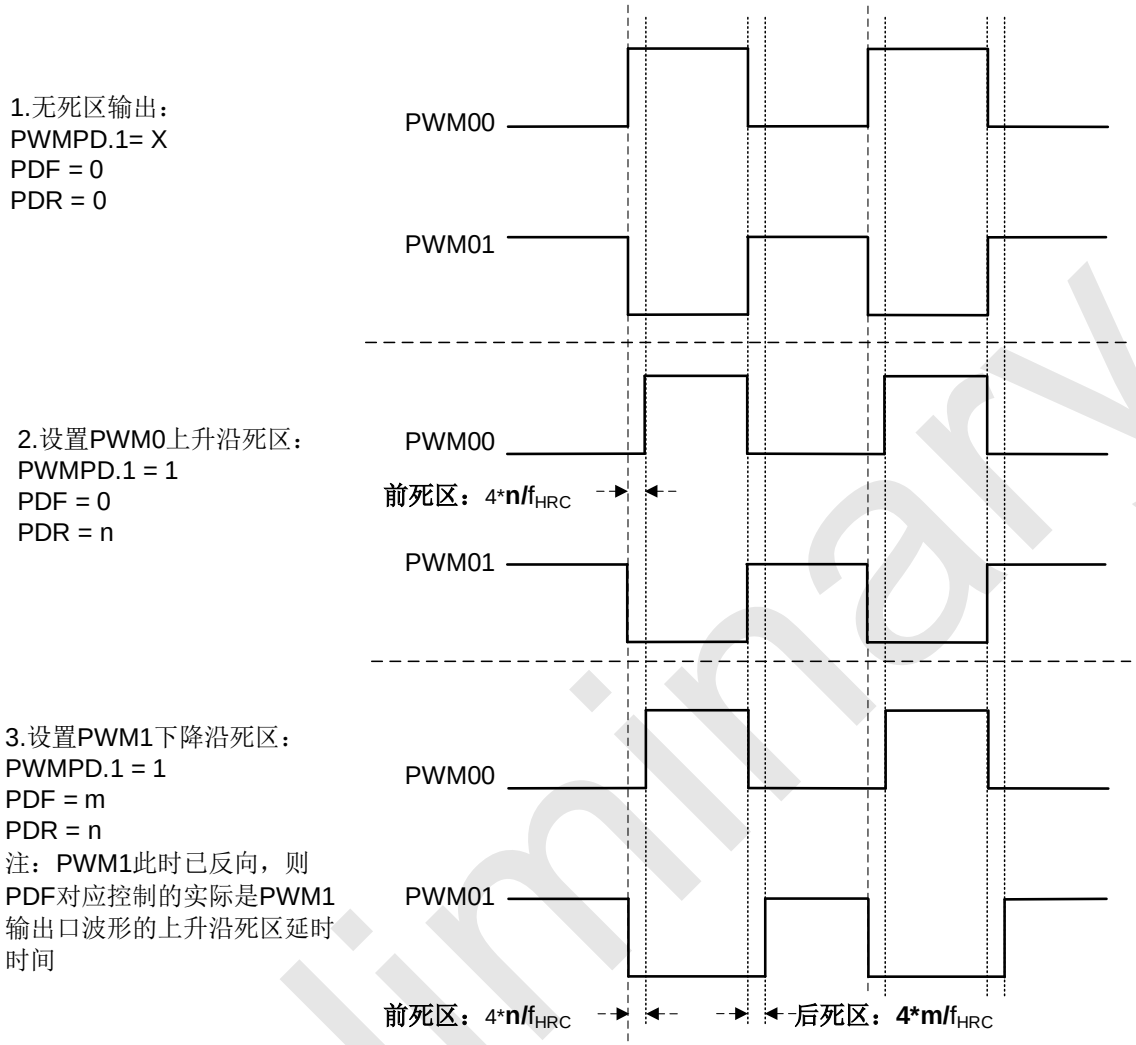
| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | **PDF[3:0]** | | | | **PDR[3:0]** | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~4 | **PDF[3:0]** | 互补模式：<br>PWM 下降沿死区时间= 4*PDF[3:0] / $f_{osc}$ |
| 3~0 | **PDR[3:0]** | 互补模式：<br>PWM 上升沿死区时间= 4*PDR[3:0] / $f_{osc}$ |

### 13.4.4 PWM0 死区输出波形

下图是以 PWM00 和 PWM01 在互补模式下的死区时间调整波形图，为了便于区分，PWM01 已反向（INV1=1）。

Preliminary




PWM0 死区输出波形

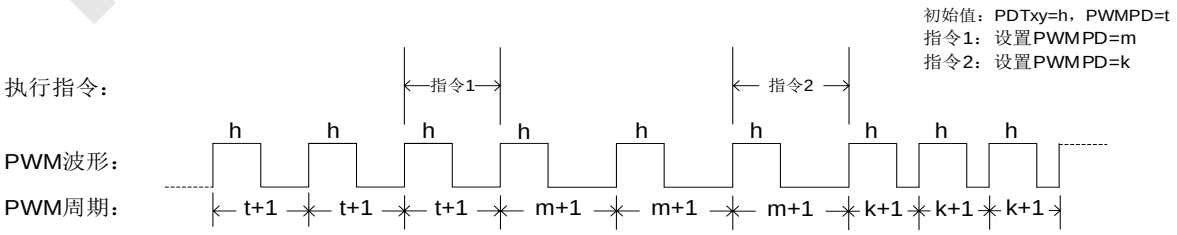
## 13.5 PWM 波形及用法

各 SFR 参数改变对 PWM0 波形影响如下所述：

① 占空比变化特性

当 PWMn 输出波形时，若需改变占空比，可通过改变高电平设置寄存器(PDTx)的值实现。但需要注意:更改 PDTx 的值，占空比不会立即改变，而是等待本周期结束，在下个周期改变。

② 周期变化特性




周期变化特性图

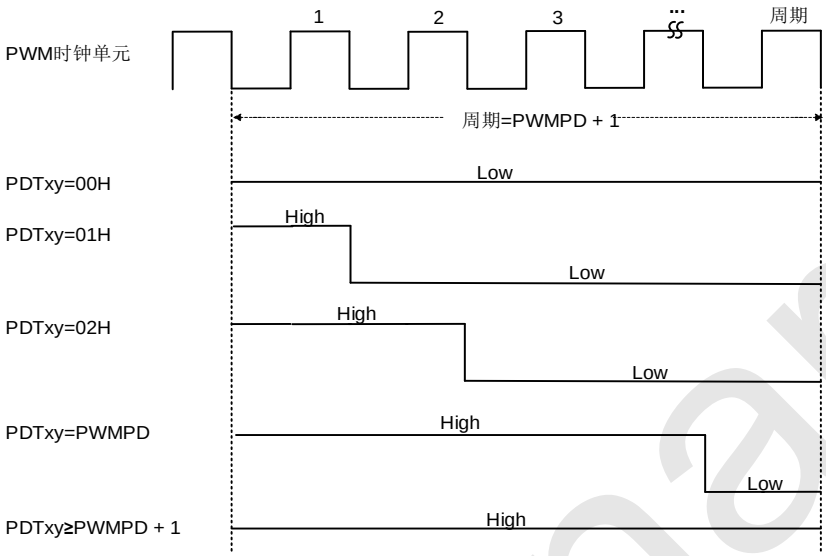
当 PWM 输出波形时，若需改变周期，可通过改变周期设置寄存器 PWMPD 的值实现。更改 PWMPD 的值，



Preliminary

周期不会立即改变，而是等待本周期结束，在下个周期改变，参考上图所示。

③ 周期和占空比的关系




周期与占空比关系图

周期和占空比的关系如上图所示。该结果的前提是 PWM 输出反向控制(INVx，x=0~7)初始为 0，若需得到相反结果，可置 INVx 为 1。

Preliminary

# 14 GP I/O

SC95F767X 提供了最多 30 个可控制的双向 GPIO 端口，输入输出控制寄存器用来控制各端口的输入输出状态，当端口作为输入时，每个 I/O 端口带有由 PxPHy 控制的内部上拉电阻。此 30 个 IO 同其他功能复用。I/O 端口在输入或输出状态下，从端口数据寄存器里读到的都是端口的实际状态值。

**注意：未使用及封装未引出的 IO 口均要设置为强推挽输出模式。**

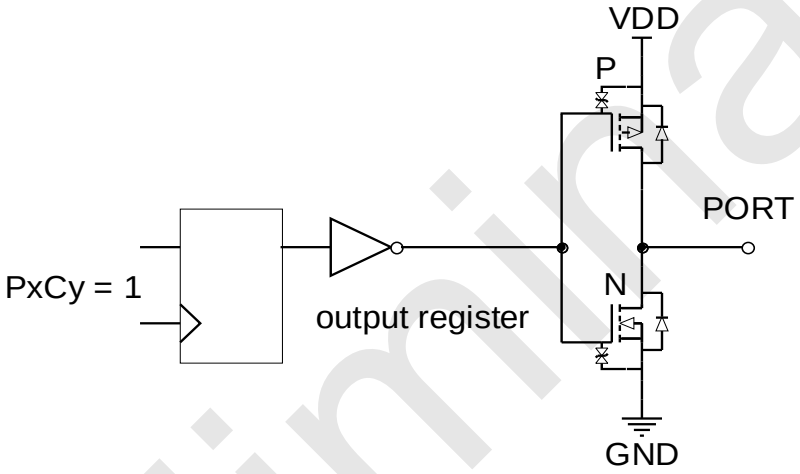
## 14.1 GPIO 结构图

**强推挽输出模式**

强推挽输出模式下，能够提供持续的大电流驱动：

- 除 P05/P20/P21 之外的 IO 驱动能力为：大于 10mA 的输出高，大于 50mA 的输出低。
- P05/P20/P21 驱动能可达到：大于 20mA 的输出高，大于 50mA 的输出低。

强推挽输出模式的端口结构示意图如下：

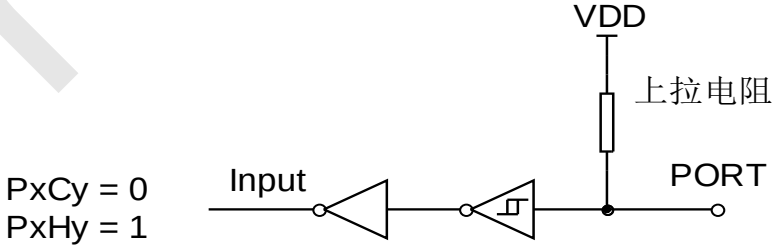



强推挽输出模式

**带上拉的输入模式**

带上拉的输入模式下，输入口上恒定接一个上拉电阻，仅当输入口上电平被拉低时，才会检测到低电平信号。

带上拉的输入模式的端口结构示意图如下：

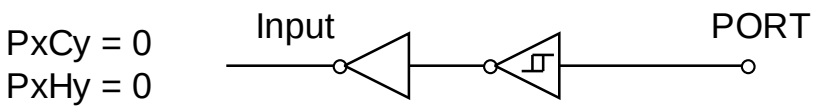



带上拉的输入模式

**高阻输入模式(Input only)**

高阻输入模式的端口结构示意图如下所示：

Preliminary




高阻输入模式

## 14.2 I/O 端口相关寄存器

**P0CON (9AH) P0 口输入/输出控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P0C7 | P0C6 | P0C5 | P0C4 | P0C3 | P0C2 | P0C1 | P0C0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**P0PH (9BH) P0 口上拉电阻控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P0H7 | P0H6 | P0H5 | P0H4 | P0H3 | P0H2 | P0H1 | P0H0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**P1CON (91H) P1 口输入/输出控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P1C7 | P1C6 | P1C5 | P1C4 | P1C3 | P1C2 | P1C1 | P1C0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**P1PH (92H) P1 口上拉电阻控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P1H7 | P1H6 | P1H5 | P1H4 | P1H3 | P1H2 | P1H1 | P1H0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**P2CON (A1H) P2 口输入/输出控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P2C7 | P2C6 | P2C5 | P2C4 | P2C3 | P2C2 | P2C1 | P2C0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**P2PH (A2H) P2 口上拉电阻控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P2H7 | P2H6 | P2H5 | P2H4 | P2H3 | P2H2 | P2H1 | P2H0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**P5CON (D9H) P5 口输入/输出控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | - | - | P5C1 | P5C0 |
| 读/写 | - | - | - | - | - | - | 读/写 | 读/写 |

Preliminary

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 上电初始值 | x | x | x | x | x | x | 0 | 0 |

**P5PH (DAH) P5 口上拉电阻控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | - | - | P5H1 | P5H0 |
| 读/写 | - | - | - | - | - | - | 读/写 | 读/写 |
| 上电初始值 | x | x | x | x | x | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **PxCy**<br>**（x=0~2,5, y=0~7）** | Px 口输入输出控制：<br>0：Pxy 为输入模式（上电初始值）<br>1：Pxy 为强推挽输出模式 |
| 7~0 | **PxHy**<br>**（x=0~2,5, y=0~7）** | Px 口上拉电阻设置，仅在 PxCy=0 时有效：<br>0：Pxy 为高阻输入模式（上电初始值），上拉电阻关闭；<br>1：Pxy 上拉电阻打开 |

**P0 (80H) P0 口数据寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**P1 (90H) P1 口数据寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**P2 (A0H) P2 口数据寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**P5 (D8H) P5 口数据寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | - | - | P5.1 | P5.0 |
| 读/写 | - | - | - | - | - | - | 读/写 | 读/写 |
| 上电初始值 | x | x | x | x | x | x | 0 | 0 |

**IOHCON0 (96H) IOH 设置寄存器 0 (读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P1H[1:0] | | P1L[1:0] | | P0H[1:0] | | P0L[1:0] | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~6 | **P1H[1:0]** | P1 高四位 IOH 设置<br>00：设置 P1 高四位 IOH 等级 0（最大）；<br>01：设置 P1 高四位 IOH 等级 1； |

Preliminary

| | | |
|---|---|---|
| | | 10：设置 P1 高四位 IOH 等级 2；<br>11：设置 P1 高四位 IOH 等级 3（最小）； |
| 5~4 | **P1L[1:0]** | P1 低四位 IOH 设置<br>00：设置 P1 低四位 IOH 等级 0（最大）；<br>01：设置 P1 低四位 IOH 等级 1；<br>10：设置 P1 低四位 IOH 等级 2；<br>11：设置 P1 低四位 IOH 等级 3（最小）； |
| 3~2 | **P0H[1:0]** | P0 高四位 IOH 设置<br>00：设置 P0 高四位 IOH 等级 0（最大）；<br>01：设置 P0 高四位 IOH 等级 1；<br>10：设置 P0 高四位 IOH 等级 2；<br>11：设置 P0 高四位 IOH 等级 3（最小）； |
| 1~0 | **P0L[1:0]** | P0 低四位 IOH 设置<br>00：设置 P0 低四位 IOH 等级 0（最大）；<br>01：设置 P0 低四位 IOH 等级 1；<br>10：设置 P0 低四位 IOH 等级 2；<br>11：设置 P0 低四位 IOH 等级 3（最小）； |

**IOHCON1 (97H) IOH 设置寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | P5H[1:0] | | P5L[1:0] | | P2H[1:0] | | P2L[1:0] | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~6 | **P5H[1:0]** | P54、P55 的 IOH 设置<br>00：设置 P54、P55 的 IOH 等级 0（IO 口驱动能力最大）；<br>01：设置 P54、P55 的 IOH 等级 1；<br>10：设置 P54、P55 的 IOH 等级 2；<br>11：设置 P54、P55 的 IOH 等级 3（IO 口驱动能力最小）； |
| 5~4 | **P5L[1:0]** | P5 低四位 IOH 设置<br>00：设置 P5 低四位 IOH 等级 0（IO 口驱动能力最大）；<br>01：设置 P5 低四位 IOH 等级 1；<br>10：设置 P5 低四位 IOH 等级 2；<br>11：设置 P5 低四位 IOH 等级 3（IO 口驱动能力最小）； |
| 3~2 | **P2H[1:0]** | P2 高四位 IOH 设置<br>00：设置 P2 高四位 IOH 等级 0（IO 口驱动能力最大）；<br>01：设置 P2 高四位 IOH 等级 1；<br>10：设置 P2 高四位 IOH 等级 2；<br>11：设置 P2 高四位 IOH 等级 3（IO 口驱动能力最小）； |
| 1~0 | **P2L[1:0]** | P2 低四位 IOH 设置<br>00：设置 P2 低四位 IOH 等级 0（IO 口驱动能力最大）；<br>01：设置 P2 低四位 IOH 等级 1；<br>10：设置 P2 低四位 IOH 等级 2；<br>11：设置 P2 低四位 IOH 等级 3（IO 口驱动能力最小）； |

Preliminary

# 15 UART0

SC95F767X支持一个全双工的串行口，可方便用于同其它器件或者设备的连接，例如Wifi模块电路或其它UART通信接口的驱动芯片等。UART0的功能及特性如下：

1. 三种通讯模式可选：模式 0、模式 1 和模式 3；
2. 可选择定时器 1 或定时器 2 作为波特率发生器；
3. 发送和接收完成可产生中断 RI/TI，该中断标志需要软件清除。

**SCON (98H) 串口控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~6 | **SM0~1** | 串行通信模式控制位<br>00：模式 0，8 位半双工同步通信模式，在 RX 引脚上收发串行数据。TX 引脚用作发送移位时钟。每帧收发 8 位，低位先接收或发送；<br>01：模式 1，10 位全双工异步通信，由 1 个起始位，8 个数据位和 1 个停止位组成，通信波特率可变；<br>10：保留；<br>11：模式 3，11 位全双工异步通信，由 1 个起始位，8 个数据位，一个可编程的第 9 位和 1 个停止位组成，通信波特率可变。 |
| 5 | **SM2** | 串行通信模式控制位 2，此控制位只对模式 2，3 有效<br>0：每收到一个完整的数据帧就置位 RI 产生中断请求；<br>1：收到一个完整的数据帧时，只有当 RB8=1 时才会置位 RI 产生中断请求。<br>**波特率倍率设置位，仅在模式 0（SM0~1 = 00）有效：**<br>0：串行端口在系统时钟的 1/12 下运行<br>1：串行端口在系统时钟的 1/4 下运行 |
| 4 | **REN** | 接收允许控制位<br>0：不允许接收数据；<br>1：允许接收数据。 |
| 3 | **TB8** | 只对模式 2、3 有效，为发送数据的第 9 位 |
| 2 | **RB8** | 只对模式 2、3 有效，为接收数据的第 9 位 |
| 1 | **TI** | 发送中断标志位 |
| 0 | **RI** | 接收中断标志位 |

**SBUF (99H) 串口数据缓存寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SBUF[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **SBUF[7:0]** | **串口数据缓存寄存器**<br>SBUF 包含两个寄存器：一个发送移位寄存器和一个接收锁存器，写入 SBUF 的数据将送至发送移位寄存器，并启动发送流程，读 SBUF 将返回接收锁存器中的内容。 |

**PCON (87H) 电源管理控制寄存器(只写、*不可读 *)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|

Preliminary

| 符号 | SMOD | - | - | - | RST | - | STOP | IDL |
|---|---|---|---|---|---|---|---|---|
| 读/写 | 只写 | - | - | - | 只写 | - | 只写 | 只写 |
| 上电初始值 | 0 | x | x | x | n | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **SMOD** | 当 SM0~1 = 01（UART0 模式 1）或 SM0~1 = 11（UART0 模式 3），波特率倍率设置位：<br>0: 串行端口在系统时钟的 1 分频下运行<br>1: 串行端口在系统时钟的 16 分频下运行 |

## 15.1 串口通信的波特率

方式 0 中，波特率可编程为系统时钟的 1/12 或 1/4：

- SM2=0，串行端口在系统时钟的 1/12 下运行；
- SM2=1，串行端口在系统时钟的 1/4 下运行。

在方式 1 和方式 3 中，串行端口时钟源可编程为系统时钟的 1 分频或 16 分频，由 SMOD(PCON.7)位决定。当 SMOD 为 0 时，串行端口在系统时钟的 1 分频下运行。当 SMOD 为 1 时，串行端口在系统时钟的 16 分频下运行。串行端口时钟源确定后，再由定时器 1 或定时器 2 设置波特率的溢出率：

- 当 TCLK(TXCON.4)和 RCLK(TXCON.5)位为均为 0，则定时器 1 为波特率发生器方式，UART0 的波特率溢出率由[TH1,TL1]设定。公式如下，注意：当定时器 1 作为波特率发生器，定时器 1 必须停止计数，即 TR1=0：
  - SMOD = 0： $\text{BaudRate} = \frac{\text{fsys}}{[\text{TH1,TL1}]}$; (注意：[TH1,TL1] 必须大于 0x0010)
  - SMOD = 1： $\text{BaudRate} = \frac{1}{16} * \frac{\text{fsys}}{[\text{TH1,TL1}]}$;
- 当 TCLK(TXCON.4)或 RCLK(TXCON.5)其中任何一位为 1，则定时器 2 为波特率发生器方式，UART0 的波特率溢出率由[RCAP2H、RCAP2L]设定，公式如下：
  - SMOD = 0： $\text{BaudRate} = \frac{\text{fsys}}{[\text{RCAP2H,RCAP2L}]}$; (注意：[RCAP2H,RCAP2L] 必须大于 0x0010)
  - SMOD = 1： $\text{BaudRate} = \frac{1}{16} * \frac{\text{fsys}}{[\text{RCAP2H,RCAP2L}]}$;

Preliminary

# 16 SPI/TWI/UART 三选一串行接口 USCI

SC95F767X 内部集成了三选一串行接口电路（简称 USCI），可方便 MCU 与不同接口的器件或者设备的连接。用户可通过配置寄存器 OTCON 的 USMD[1:0] 位将 USCI 接口配置为 SPI、TWI 和 UART 中任意一种通信模式。其特点如下：

1. SPI 模式可配置为主模式或从属模式中的一种
2. TWI 模式通信可配置为主模式或从属模式
3. UART 模式可工作在模式 1（10 位全双工异步通信）和模式 3（11 位全双工异步通信）

USCI 接口相关控制寄存器如下：

| 符号 | 地址 | 说明 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Reset 值 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| US0CON0 | 95H | USCI0 控制寄存器 0 | US0CON0[7:0] | | | | | | | | 00000000b |
| US0CON1 | 9DH | USCI0 控制寄存器 1 | US0CON1[7:0] | | | | | | | | 00000000b |
| US0CON2 | 9EH | USCI0 控制寄存器 2 | US0CON2[7:0] | | | | | | | | 00000000b |
| US0CON3 | 9FH | USCI0 控制寄存器 3 | US0CON3[7:0] | | | | | | | | 00000000b |
| US1CON0 | A4H | USCI1 控制寄存器 0 | US1CON0[7:0] | | | | | | | | 00000000b |
| US1CON1 | A5H | USCI1 控制寄存器 1 | US1CON1[7:0] | | | | | | | | 00000000b |
| US1CON2 | A6H | USCI1 控制寄存器 2 | US1CON2[7:0] | | | | | | | | 00000000b |
| US1CON3 | A7H | USCI1 控制寄存器 3 | US1CON3[7:0] | | | | | | | | 00000000b |
| US2CON0 | C4H | USCI2 控制寄存器 0 | US2CON0[7:0] | | | | | | | | 00000000b |
| US2CON1 | C5H | USCI2 控制寄存器 1 | US2CON1[7:0] | | | | | | | | 00000000b |
| US2CON2 | C6H | USCI2 控制寄存器 2 | US2CON2[7:0] | | | | | | | | 00000000b |
| US2CON3 | C7H | USCI2 控制寄存器 3 | US2CON3[7:0] | | | | | | | | 00000000b |

具体配置方式如下：

**OTCON (8FH) 输出控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | USMD1[1:0] | | USMD0[1:0] | | - | - | - | - |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | - | - | - | - |
| 上电初始值 | 0 | 0 | 0 | 0 | x | x | x | x |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~6 | **USMD1[1:0]** | **USCI1 通信模式控制位**<br>00：USCI1 关闭<br>01：USCI1 设置为 SPI 通信模式；<br>10：USCI1 设置为 TWI 通信模式；<br>11：USCI1 设置为 UART 通信模式； |
| 5~4 | **USMD0[1:0]** | **USCI0 通信模式控制位**<br>00：USCI0 关闭<br>01：USCI0 设置为 SPI 通信模式；<br>10：USCI0 设置为 TWI 通信模式；<br>11：USCI0 设置为 UART 通信模式； |

**TMCON (8EH) 定时器频率控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | USMD2[1:0] | | - | - | - | - | T1FD | T0FD |
| 读/写 | 读/写 | 读/写 | - | - | - | - | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | x | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~6 | **USMD2[1:0]** | **USCI2 通信模式控制位**<br>00：USCI2 关闭<br>01：USCI2 设置为 SPI 通信模式；<br>10：USCI2 设置为 TWI 通信模式；<br>11：USCI2 设置为 UART 通信模式； |

Preliminary

注意：

一个 USCI 接口可以通过 USMD 设置为不同的通信模式，这些通信模式均有对应的操作寄存器组。不同通信模式的控制寄存器组共用了同一组映射地址，但各组之间的操作相互独立，设置其中一种通信模式下的控制寄存器，不会影响到其它通信模式下的寄存器组里的值。

例如：

- 设置 USMD0 = 01，USCI0 为 SPI 通信接口，在此模式下设置 US0CON0 (95H) = 0x80H；
- 接着设置 USMD0 =11，USCI0 为 UART 通信接口，在此模式下设置 US0CON0 (95H) = 0x0FH；
- 再设置 USMD0 = 01，USCI0 再设置回 SPI 通信接口，在此模式下读取 US0CON0 (95H)，应为 0x80H；
- 再设置 USMD0 =11，USCI0 再设置回 UART 通信接口，在此模式下读取 US0CON0 (95H)，应为 0x0FH。

## 16.1 SPI

串行外部设备接口(简称 SPI)是一种高速串行通信接口，允许 MCU 与外围设备(包括其它 MCU)进行全双工，同步串行通信。

其中 USCI0 的 SPI 接口具有 16 位 8 级 FIFO 缓存，发送接收独立，即用户可以实现：

- 连续向 SPI 发送缓存（US0CON2、US0CON3）写入 8 个或 8 个以内的 16 位发送数据，SPI 发送的时候，最先写入的数据也最先被发送。当用户写入 FIFO 的数据被发送完成，发送缓存器空标志 TXE 置 1；若 FIFO 的数据已满，则写入冲突标志位 WCOL 置位，用户无法向 FIFO 写入数据，直至 FIFO 内的数据被发送出去、FIFO 不满，用户才能写入数据。当 FIFO 内的数据全部发送完毕才置起中断标志 SPIF。
- 连续从 SPI 接收缓存（US0CON2、US0CON3）读取 8 个或 8 个以内的 16 位接收数据，最先接收到的数据也最先被读取到。

| 对比 / BIT 位 | USCI1/2 的 SPI | USCI0 的 SPI |
|---|---|---|
| TXE | 发送缓冲器为空，则置 1 | 写入 FIFO 的数据发完，则置 1 |
| WCOL | 当一帧正在发送时，再写入会置 1，且写不进 | 写满 FIFO 后会置 1，FIFO 再写写不进 |
| SPIF | 发送完成，中断标志置起 | FIFO 内的数据全部发送完毕才置起中断标志 |

SC95F767X 的三个 SPI 接口 SPI0/1/2 均可设为主模式或从模式。

### 16.1.1 SPI0/1/2

USMDn[1:0] = 01 (n=1,2)，三选一串行接口 USCI 配置为 SPI 接口，即 SPI0/1/2：

- USTXn 作为 MOSI 信号
- USRXn 作为 MISO 信号
- USCKn 作为 CLK 信号

SPI0/1/2 均可设为主模式或从模式。

#### 16.1.1.1 SPI0/1/2 操作相关寄存器

**US0CON0 (95H) SPI0 控制寄存器(读/写)**
**US1CON0 (A4H) SPI1 控制寄存器(读/写)**
**US2CON0 (C4H) SPI2 控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SPEN | - | MSTR | CPOL | CPHA | SPR2 | SPR1 | SPR0 |
| 读/写 | 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | x | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **SPEN** | **SPI 使能控制位**<br>0：关闭 SPI<br>1：打开 SPI |
| 5 | **MSTR** | **SPI 主从选择位**<br>0： SPI 为从设备 |



Preliminary

| | | |
|---|---|---|
| | | 1： SPI 为主设备 |
| 4 | **CPOL** | **时钟极性控制位**<br>0：SCK 在空闲状态下为低电平<br>1：SCK 在空闲状态下为高电平 |
| 3 | **CPHA** | **时钟相位控制位**<br>0：SCK 周期的第一沿采集数据<br>1：SCK 周期的第二沿采集数据 |
| 2~0 | **SPR[2:0]** | **SPI 时钟速率选择位**<br>000：$f_{SYS}$<br>001：$f_{SYS}/2$<br>010：$f_{SYS}/4$<br>011：$f_{SYS}/8$<br>100：$f_{SYS}/16$<br>101：$f_{SYS}/32$<br>110：$f_{SYS}/64$<br>111：$f_{SYS}/128$ |
| 1 | **-** | 保留 |

**US0CON1 (9DH) SPI0 状态寄存器(读/写)**
**US1CON1 (A5H) SPI1 状态寄存器(读/写)**
**US2CON1 (C5H) SPI2 状态寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SPIF | WCOL | - | - | TXE | DORD | SPMD | TBIE |
| 读/写 | 读/写 | 读/写 | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **SPIF** | **SPI 数据传送标志位**<br>0：由软件清 0<br>1：表明已完成数据传输，由硬件置 1 |
| 6 | **WCOL** | **写入冲突标志位**<br>0：由软件清 0，表明已处理写入冲突<br>1：由硬件置 1，表明检测到一个冲突 |
| 3 | **TXE** | **发送缓存器空标志**<br>0：发送缓存器不空<br>1：发送缓存器空，必须由软件清零 |
| 2 | **DORD** | **传送方向选择位**<br>0：MSB 优先发送<br>1：LSB 优先发送 |
| 1 | **SPMD** | **SPI 传输模式选择：**<br>0：8 位模式<br>1：16 位模式 |
| 0 | **TBIE** | **发送缓存器中断允许控制位**<br>0：TXE=1 时，不允许产生中断<br>1：TXE=1 时，将产生 SPI 中断 |
| 5, 4 | **-** | 保留 |

**SPDL**
**US0CON2 (9EH) SPI0 数据缓存寄存器低字节(读/写)**



Preliminary

**US1CON2 (A6H) SPI1 数据寄存器低字节(读/写)**
**US2CON2 (C6H) SPI2 数据寄存器低字节(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SPD[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **SPD[7:0]** | **SPI 数据缓存寄存器低字节（8/16 位模式）**<br>写入数据寄存器 SPD 的数据低字节<br>读取数据寄存器 SPD 的数据低字节 |

**SPDH**
**US0CON3 (9FH) SPI0 数据缓存寄存器高字节(读/写)**
**US1CON3 (A7H) SPI1 数据寄存器高字节(读/写)**
**US2CON3 (C7H) SPI2 数据寄存器高字节(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SPD[15:8] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **SPD[15:8]** | **SPI 数据缓存寄存器高字节（仅用于 16 位模式）**<br>写入数据寄存器 SPD 的数据高字节<br>读取数据寄存器 SPD 的数据高字节<br>**注意：SPI 设置为 16 位模式时，必须先写入高字节，后写入低字节，低字节写入后数据立刻送入发送缓存。** |

### 16.1.1.2 SPI0/1/2 信号描述

**主输出从输入(MOSI)：**

该路信号连接主设备和一个从设备。数据通过 MOSI 从主设备串行传送到从设备，主设备输出，从设备输入。

**主输入从输出(MISO)：**

该路信号连接从设备和主设备。数据通过 MISO 从从设备串行传送到主设备，从设备输出，主设备输入。当 SPI 配置为从设备并未被选中，从设备的 MISO 引脚处于高阻状态。

**SPI 串行时钟(SCK)：**

SCK 信号用作控制 MOSI 和 MISO 线上输入输出数据的同步移动。每 8 时钟周期线上传送一个字节。如果从设备未被选中，SCK 信号被此从设备忽略。

### 16.1.1.3 SPI0/1/2 工作模式

SPI 可配置为主模式或从属模式中的一种。SPI 模块的配置和初始化通过设置 SPI 控制寄存器 USnCON0 (n=0~2)和 SPI 状态寄存器 USnCON1 来完成。配置完成后，通过设置 SPI 数据寄存器 USnCON2，USnCON3(以下简称 SPD)来完成数据传送。

在 SPI 通讯期间，数据同步地被串行的移进移出。串行时钟线(SCK)使两条串行数据线(MOSI 和 MISO)上数据的移动和采样保持同步。如果从设备没有被选中，则不能参与 SPI 总线上的活动。

当 SPI 主设备通过 MOSI 线传送数据到从设备时，从设备通过 MISO 线发送数据到主设备作为响应，这就实现了在同一时钟下数据发送和接收的同步全双工传输。发送移位寄存器和接收移位寄存器使用相同的特殊功能器地址，对 SPI 数据寄存器 SPD 进行写操作将写入发送移位寄存器，对 SPD 进行读操作将获得接收移位寄存器的数据。

有些设备的SPI接口会引出SS脚（从设备选择引脚，低有效），与SC95F767X的SPI通信时， SPI总线上其它设备的SS脚的连接方式需根据不同的通信模式进行连接。下表列出了SC95F767X的SPI不同通信模式下，SPI总线上其它设备SS脚的连接方式：


Preliminary

| SC95F767X SPI | SPI 总线上其它设备 | 模式 | 从机的 SS（从设备选择引脚） |
|---|---|---|---|
| 主模式 | 从模式 | 一主一从 | 拉低 |
| | | 一主多从 | SC95F767X 引出多根 I/O，分别接至从机的 SS 脚。在数据传送之前，从设备的 SS 引脚必须被置低 |
| 从模式 | 主模式 | 一主一从 | 拉高 |

**SPI0/1/2 主模式**

- **SPI0/1/2 主模式启动：**

  SPI 主设备控制 SPI 总线上所有数据传送的启动。当 SPI 控制寄存器 USnCON0 (n=0~2)中的 MSTR 位置 1 时，SPI 在主模式下运行，只有一个主设备可以启动传送。

- **SPI0/1/2 主模式发送：**

  在 SPI 主模式下，对 SPD 进行以下操作：在 8 位模式下写一个字节数据到 SPDL 或在 16 位模式下先将高字节写入 SPDH，再将低字节写入 SPDL，数据将会写入发送移位缓冲器。如果发送移位寄存器已经存在一个数据，那么主 SPI 产生一个 WCOL 信号以表明写入太快。但是在发送移位寄存器中的数据不会受到影响，发送也不会中断。另外如果发送移位寄存器不为空，那么主设备立即按照 SCK 上的 SPI 时钟频率串行地移出发送移位寄存器中的数据到 MOSI 线上。当传送完毕，数据传输完整标志位 SPIF 被置 1。如果 SPI 中断被允许，当 SPIF 位置 1 时，也会产生一个中断。

- **SPI0/1/2 主模式接收：**

  当主设备通过 MOSI 线传送数据给从设备时，相对应的从设备同时也通过 MISO 线将其发送移位寄存器的内容传送给主设备的接收移位寄存器，实现全双工操作。因此，SPIF 标志位置 1 即表示传送完成也表示接收数据完毕。从设备接收的数据按照 MSB 或 LSB 优先的传送方向存入主设备的接收移位寄存器。当一个字节的数据完全被移入接收寄存器时， 处理器可以通过读 SPD 获得该数据。

**从模式**

- **模式启动：**

  当 SPI 控制寄存器 USnCON0 寄存器中的 MSTR 位清 0，SPI 在从模式下运行。

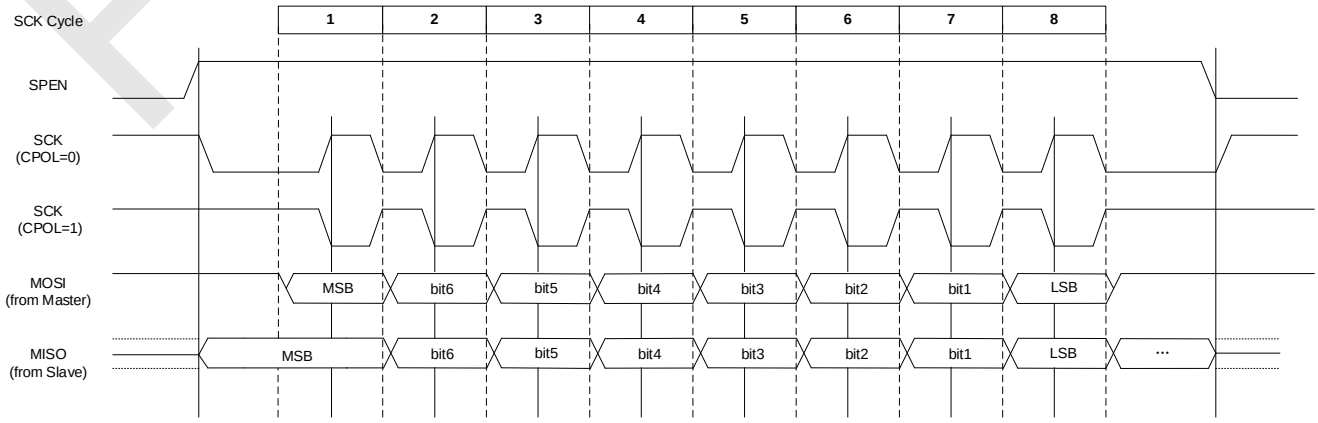
- **发送与接收：**

  从属模式下，按照主设备控制的 SCK 信号，数据通过 MOSI 引脚移入，MISO 引脚移出。一个位计数器记录 SCK 的边沿数，当接收移位寄存器移入 8 位数据(一个字节)同时发送移位寄存器移出 8 位数据(一个字节)，SPIF 标志位被置 1。数据可以通过读取 SPD 寄存器获得。如果 SPI 中断被允许，当 SPIF 置 1 时，也会产生一个中断。此时接收移位寄存器保持原有数据并且 SPIF 位置 1，这样 SPI 从设备将不会接收任何数据直到 SPIF 清 0。SPI 从设备必须在主设备开始一次新的数据传送之前将要传送的数据写入发送移位寄存器。如果在开始发送之前未写入数据，从设备将传送“0x00”字节给主设备。如果写 SPD 操作发生在传送过程中，那么 SPI 从设备的 WCOL 标志位置 1，即如果传送移位寄存器已经含有数据。但是移位寄存器的数据不受影响，传送也不会被中断。

**传送形式**

通过软件设置 SPI 控制寄存器 USnCON0（n=1,2）的 CPOL 位和 CPHA 位，用户可以选择 SPI 时钟极性和相位的四种组合方式。CPOL 位定义时钟的极性，即空闲时的电平状态，它对 SPI 传输格式影响不大。CPHA 位定义时钟的相位，即定义允许数据采样移位的时钟边沿。在主从通讯的两个设备中，时钟极性相位的设置应一致。

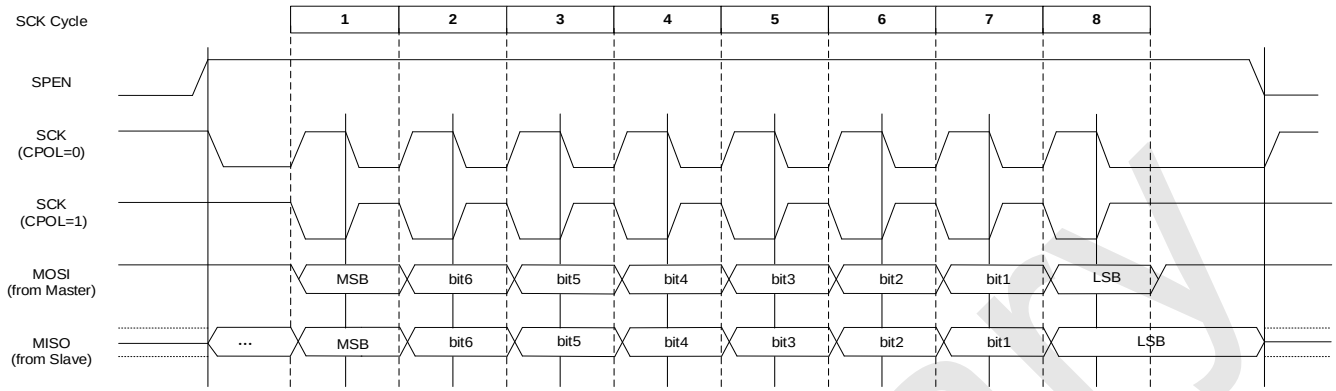
当 CPHA = 0，SCK 的第一个沿捕获数据，从设备必须在 SCK 的第一个沿之前将数据准备好。




CPHA = 0 数据传输图

Preliminary

当 CPHA = 1，主设备在 SCK 的第一个沿将数据输出到 MOSI 线上，从设备把 SCK 的第一个沿作为开始发送信号，SCK 的第二沿开始捕获数据，因此用户必须在第一个 SCK 的两个沿内完成写 SPD 的操作。这种数据传输形式是一个主设备一个从设备之间通信的首选形式。




CPHA = 1 数据传输图

**出错检测**

在发送数据序列期间写入 SPD 会引起写冲突，SPI 状态寄存器 USnCON1 中的 WCOL 位置 1。WCOL 位置 1 不会引起中断，发送也不会中止。WCOL 位需由软件清 0。

Preliminary

## 16.2 TWI

USMDn[1:0] = 10，n=0~2 三选一串行接口 USCI 配置为 TWI 接口：

- USTXn 作为 SDA 信号
- USCKn 作为 CLK 信号

SC95F767X 在 TWI 通信时可根据应用需要设定为主机或从机模式。

**US0CON0 (95H) TWI0 控制寄存器 0(读/写)**
**US1CON0 (A4H) TWI1 控制寄存器 0(读/写)**
**US2CON0 (C4H) TWI2 控制寄存器 0(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TWEN | TWIF | MSTR | GCA | AA | STATE[2:0] | | |
| 读/写 | 读/写 | 读/写 | 读 | 读 | 读/写 | 读 | 读 | 读 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **TWEN** | TWI 使能控制<br>0：关闭 TWI<br>1：打开 TWI |
| 6 | **TWIF** | TWI 中断标志位<br>1. 由软件清零<br>2. 在下列条件下，中断标志位由硬件置 1：<br>1) 主机模式：<br>① 发送启动信号<br>② 发送完地址帧<br>③ 接收或发送完数据帧<br>2) 从机模式：<br>① 第一帧地址匹配成功<br>② 成功接收或发送 8 位数据<br>③ 接收到重复起始条件<br>④ 从机收到停止信号 |
| 5 | **MSTR** | 主从标志位<br>0：从模式<br>1：主模式<br>说明：<br>1. 当 TWI 接口向总线发出起始条件后，会自动切换为主模式，同时硬件将该位置位；<br>2. 当总线上检测到一个停止条件时，硬件清除该位。 |
| 4 | **GCA** | 通用地址响应标志位<br>0：非响应通用地址<br>1：当 GC 置 1，同时通用地址匹配时该位由硬件置 1，并自动清零 |
| 3 | **AA** | **应答使能位**<br>**0：无应答，返回 UACK**（应答位为高电平）<br>**1：在接收到一个匹配的地址或数据后返回一个应答 ACK** |
| 2~0 | **STATE[2:0]** | 状态机状态标志位<br>从机模式：<br>000：从机处于空闲状态，等待 TWEN 置 1，检测 TWI 启动信号。当从机接收到停止条件后跳会转到此状态<br>001：从机正在接收第一帧地址和读写位（第 8 位为读写位，1 为读，0 为写）。从机接收到起始条件后会跳转到此状态<br>010：从机接收数据状态 |

Preliminary

| | | |
|---|---|---|
| | | 011：从机发送数据状态<br>100：在从机发送数据状态中，当主机回 UACK 时跳转到此状态，等待重新启动信号或停止信号。<br>101：从机处于发送状态时，将 AA 写 0 会进入此状态，等待重新启动信号或停止信号。<br>110：从机的地址与主机发送的地址不匹配会跳转到此状态，等待新的起始条件或停止条件。<br><br>主机模式：<br>000：状态机为空闲状态<br>001：主机发送起始条件或主机正在发送从设备地址<br>010：主机发送数据<br>011：主机接收数据<br>100：主机发送停止条件或接收到从机的 UACK 信号 |

**US0CON1 (9DH) TWI0 控制寄存器 1(读/写)**
**US1CON1 (A5H) TWI1 控制寄存器 1(读/写)**
**US2CON1 (C5H) TWI2 控制寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TXnE/RXnE | **STRETCH** | STA | STO | TWCK[3:0] | | | |
| 读/写 | 只读 | **读/写** | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| **7** | **TXnE/RXnE** | 发送/接收完成标志位<br>以下情况，TXnE/RXnE 被置 1<br>主机模式：<br>① 主机发送地址帧（写），且收到从机的 ACK<br>② 主机发送完数据，且接收到从机 ACK<br>③ 主机接收到数据，且主机回从机 ACK<br>从机模式：<br>① 从机接收地址帧（读），且和从机地址（TWA）匹配<br>② 从机接收到数据，且从机回主机 ACK<br>③ 从机发送完数据，且接收到主机 ACK（AA=1）<br>对 TWIDAT 进行读写操作会清除此标志位。 |
| **6** | **STRETCH** | 允许时钟延长（从模式）<br>0：禁止时钟延长<br>1：允许时钟延长，主机需要支持时钟延长功能<br>说明：在数据传输完成之后，且 ACK 为 0，此时时钟延长发生 |
| 5 | STA | 起始位<br>置"1"产生起始条件，TWI 将切换为主机模式<br>软件可以设置或清除该位，或当起始条件发出后，由硬件清除。 |
| 4 | STO | 主机模式停止位<br>主机模式下置"1"，在当前字节传输或起始条件发出后产生停止条件<br>软件可以设置或清除该位，或当检测到停止条件时，由硬件清除。 |
| 3~0 | TWCK[3:0] | 主机模式下 TWI 的通信速率设定：<br>0000：$f_{SYS}$ /1024<br>0001：$f_{SYS}$ /512<br>0010：$f_{SYS}$ /256<br>0011：$f_{SYS}$ /128<br>0100：$f_{SYS}$ /64 |

Preliminary

| | | |
|---|---|---|
| | | 0101：$f_{SYS}$ /32<br>0110：$f_{SYS}$ /16<br>其它：保留<br>**注意：**<br>**1. 从机模式下设定无效。时钟频率最高为 400kHz；**<br>**2. TWI 的时钟源跟随系统时钟 $f_{SYS}$** |
| 7 | - | 保留 |

**US0CON2 (9EH) TWI0 地址寄存器(读/写)**
**US1CON2 (A6H) TWI1 地址寄存器(读/写)**
**US2CON2 (C6H) TWI2 地址寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TWA[6:0] | | | | | | | GC |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~1 | **TWA[6:0]** | TWI 地址寄存器<br>TWA[6:0]不能写为全 0，00H 为通用地址寻址专用。主机模式下设定无效 |
| 0 | **GC** | TWI 通用地址使能<br>0： 禁止响应通用地址 00H<br>1： 允许响应通用地址 00H |

**US0CON3 (9FH) TWI0 数据缓存寄存器(读/写)**
**US1CON3 (A7H) TWI1 数据缓存寄存器(读/写)**
**US2CON3 (C7H) TWI2 数据缓存寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | TWDAT[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **TWDAT[7:0]** | TWI 数据缓存寄存器 |

### 16.2.1 信号描述

**TWI 时钟信号线（SCL）**

该时钟信号由主机发出，连接到所有的从机。每 9 个时钟周期传送一个字节数据。前 8 个周期作数据的传送，最后一个时钟作为接收方应答时钟。空闲时应为高电平，由 SCL 线上的上拉电阻拉高。

**TWI 数据信号线（SDA）**

SDA 是双向信号线，空闲时应为高电平，由 SDA 线上的上拉电阻拉高。

### 16.2.2 从机工作模式

- **模式启动：**

  当 TWI 使能标志位打开（TWEN = 1），同时接收到主机发送的启动信号时，模式启动。

  从机从空闲模式（STATE[2:0] = 000）进入接收第一帧地址（STATE[2:0] = 001）状态，等待主机的第一帧数据。第一帧数据由主机发送，包括了 7 位地址位和 1 位读写位，TWI 总线上所有从机都会收到主机的第一帧数据。主机发送完第一帧数据后释放 SDA 信号线。若主机所发地址与某一从机自身地址寄存器中的值相同，说明该从机被选中，被选中的从机会判断接总线上的第 8 位，即数据读写位（=1，读命令；=0，写命令），然后占用 SDA 信号线，在 SCL 的第 9 个时钟周期给主机一个低电平的应答信号，之后会释放总线。从机被选中后，会根据读写位的不同而进入不同的状态：

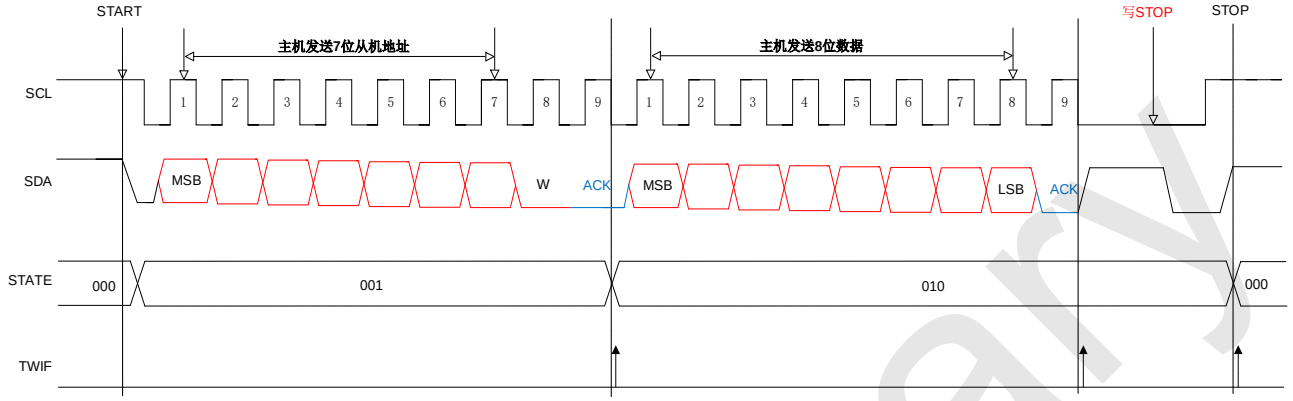
- **非通用地址响应，从机接收模式：**

  如果第一帧接收到的读写位是写（0），则从机进入到从机接收状态（STATE[2:0] = 010）等待接收主机

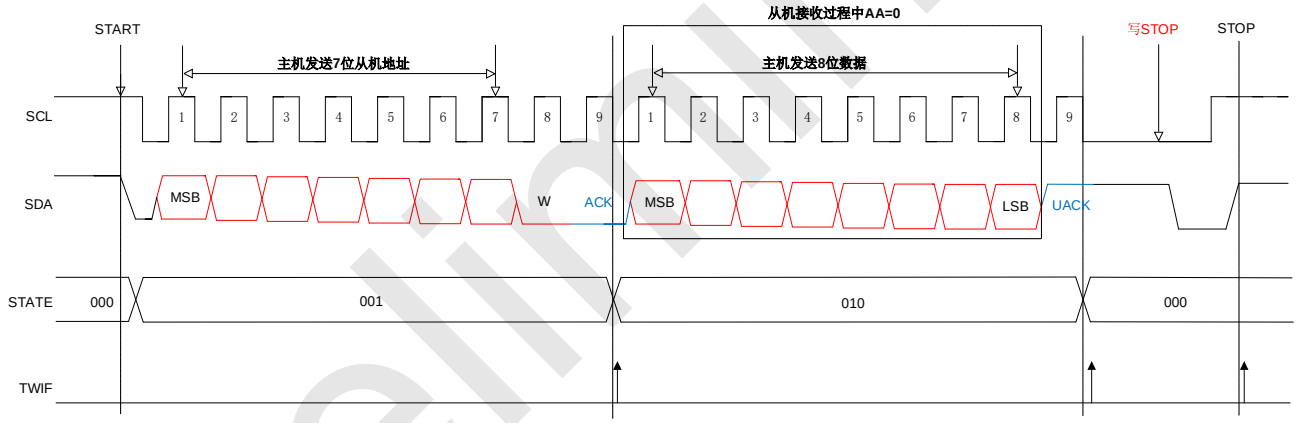

Preliminary

发送的数据。主机每发送 8 位，都要释放总线，等待第 9 个周期从机的应答信号。

1. 如果从机的应答信号是低电平，主机的通信可以有以下三种方式：
   1) 继续发送数据；
   2) 重新发送启动信号（start），此时从机重新进入接收第一帧地址（STATE[2:0] = 001）状态；
   3) 发送停止信号，表示本次传输结束，从机回到空闲状态，等待主机下一次的启动信号。



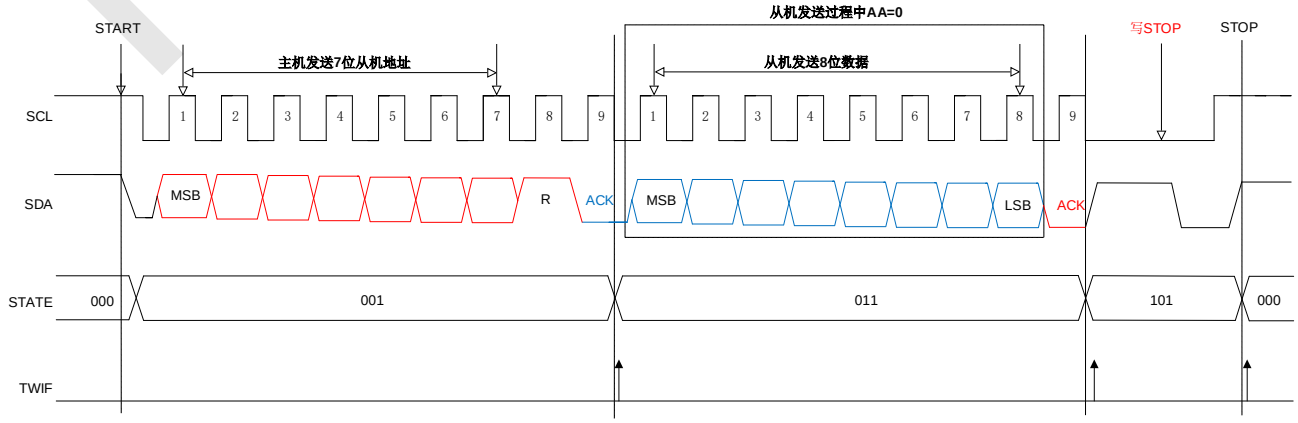

2. 如果从机应答的是高电平（在接收过程中，从机寄存器中的 AA 值改写为 0），表示当前字节传输完以后，从机会主动结束本次传输，回到空闲状态（STATE[2:0] = 000），不再接收主机发送的数据。




- **非通用地址响应，从机发送模式：**

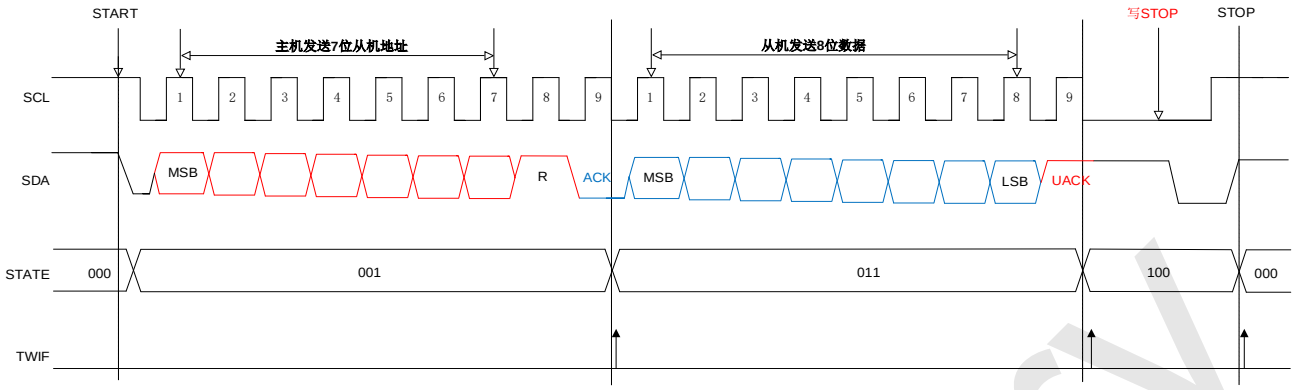
  如果第一帧接收到的读写位是读（1），则从机会占用总线，向主机发送数据。每发送 8 位数据，从机释放总线，等待主机的应答：

  1. 如果主机应答的是低电平，则从机继续发送数据。在发送过程中，如果从机寄存器中的 AA 值被改写为 0，则传输完当前字节从机会主动结束传输并释放总线，等待主机的停止信号或重新启动信号（STATE[2:0] = 101）。




Preliminary

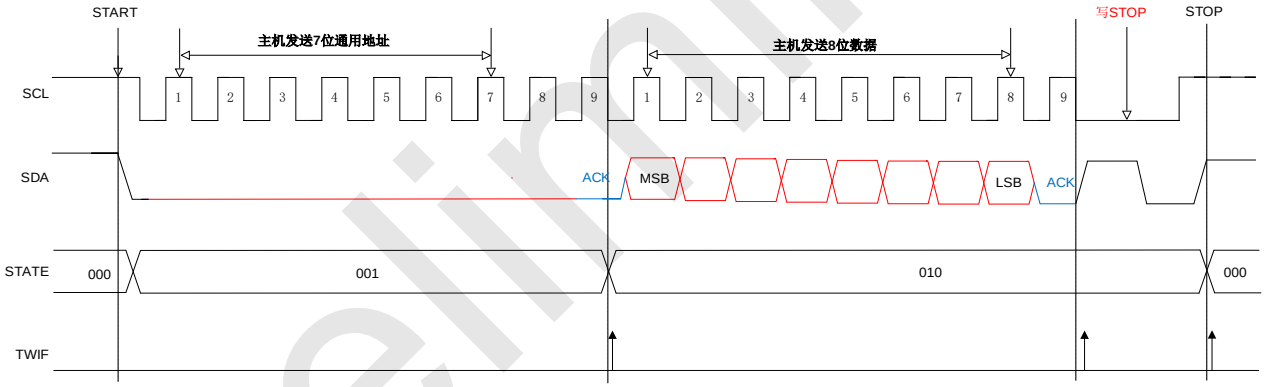
2. 如果主机应答的是高电平，则从机 STATE[2:0] = 100，等待主机的停止信号或重新启动信号。




- **通用地址的响应：**

  GC=1 时，此时通用地址允许使用。从机进入到接收第一帧地址（STATE[2:0] = 001）状态，接收的第一帧数据中的地址位数据为 0x00，此时所有从机响应主机。主机发送的读写位是必须是写（0），所有从机接收后进入接收数据（STATE[2:0] = 010）状态。主机每发送 8 个数据释放一次 SDA 线，并读取 SDA 线上的状态：

  1. 如果有从机应答，则主机的通信可以有以下三种方式：
     1) 继续发送数据；
     2) 重新启动；
     3) 发送停止信号，结束本次通讯。




  2. 如果无从机应答，则 SDA 为空闲状态。

**注意：在一主多从模式下使用通用地址时，主机发送的读写位不能为读（1）状态，否则除发送数据的设备，总线上其它设备均会响应。**

### 16.2.3 从机模式操作步骤

1. 配置 USMDn[1:0]，选择 TWI 模式；
2. 配置 TWIn 控制寄存器 USnCON0 和 USnCON1；
3. 配置 TWI 地址寄存器 USnCON2；
4. 如果从机接收数据，则等待 USnCON0 中的中断标志位 TWIF 置 1。从机每接收到 8 位数据，TWIF 会被置 1。中断标志位 TWIF 需手动清零；
5. 如果从机发送数据，则要将待发送的数据写进 TWDAT 中，TWI 会自动将数据发送出去。每发送 8 位，中断标志位 TWIF 就会被置 1。

### 16.2.4 主机工作模式

- **模式启动：**

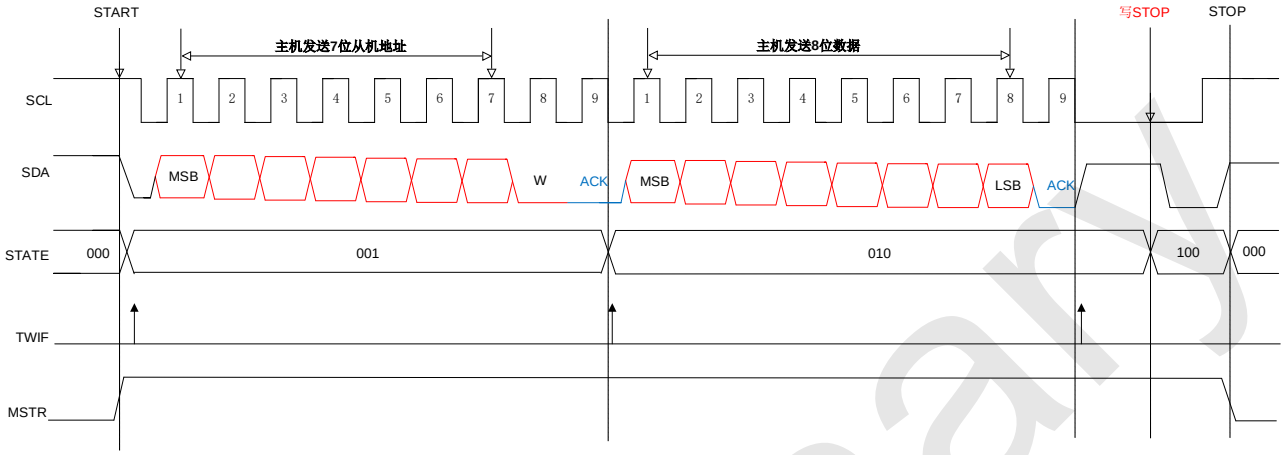
  当 TWI 接口向总线发出起始条件后，会自动切换为主模式，同时硬件将 MSTR 位置 1。主机状态位 STATE[2:0]从 000 切换到 001，同时中断条件 TWIF 被置 1。

- **TWI 主机发送模式：**

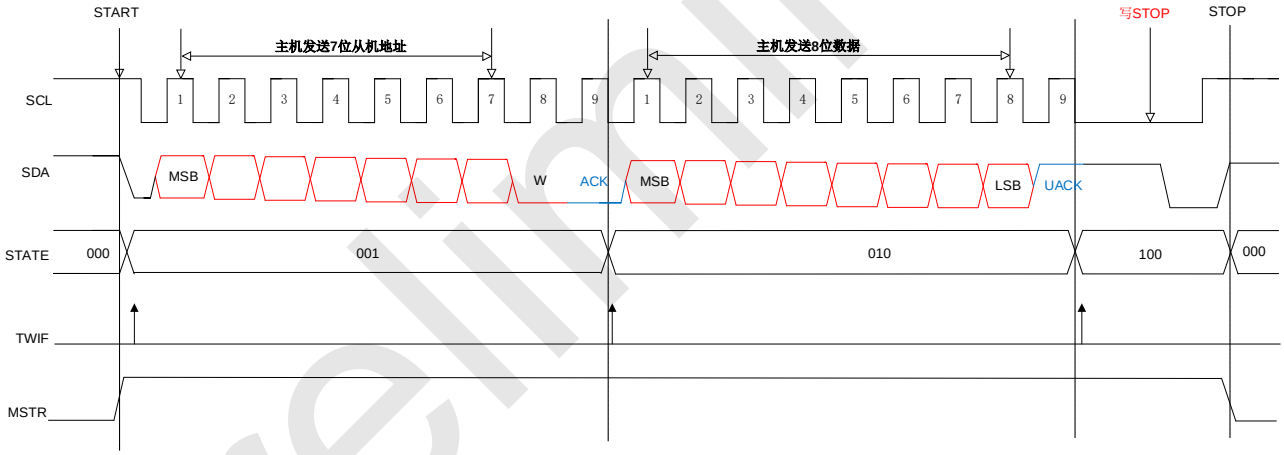
Preliminary

主机发送模式下，主机发送的第一帧数据包括了 7 位地址位（被选中的从机地址）和 1 位读写位（=0，写命令），TWI 总线上所有从机都会收到主机的第一帧数据。主机发送完第一帧数据后释放 SDA 信号线。被选中的从机在 SCL 的第 9 个时钟周期给主机一个应答信号，之后会释放总线并进入到从机接收状态等待接收主机发送的数据。主机每发送 8 位，都要释放总线，等待第 9 个周期从机的应答信号。

1. 如果从机应答低电平，主机可以继续发送数据。也可以重新发送启动信号：




2. 如果从机应答高电平，表示当前字节传输完以后，从机会主动结束本次传输，不再接收主机发送的数据，主机 STATE[2:0]从发送数据状态 010 切换为 100：



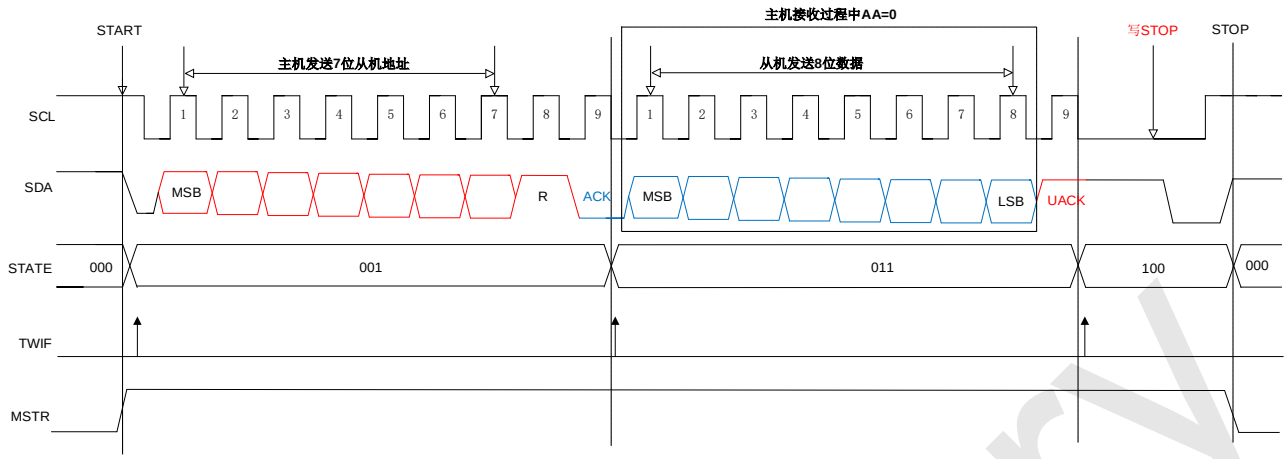

- **TWI 主机接收模式：**

主机发送模式下，主机发送的第一帧数据包括了 7 位地址位（被选中的从机地址）和 1 位读写位（=1，读命令），TWI 总线上所有从机都会收到主机的第一帧数据。主机发送完第一帧数据后释放 SDA 信号线。被选中的从机在 SCL 的第 9 个时钟周期给主机一个应答信号，之后会占用总线，向主机发送数据。每发送 8 位数据，从机释放总线，等待主机的应答。主机接收到从机地址匹配成功后的应答信号 ACK，并开始接收从机数据（STATE=011）：

1. 若主机应答位使能（AA=1），则每接收到一 BYTE 数据，主机回复应答信号 ACK，TWIF 被置位；
2. 在接收最后一 byte 数据前，若应答使能位关闭（AA=0），则主机接收完最后一 byte 数据后回复 UACK，然后主机可发送停止信号。

主机接收模式下，主动释放总线方式如下：

Preliminary




## 16.2.5 主机模式操作步骤

1. 配置 USMDn[1:0]，选择 TWI 模式；
2. 配置 TWIn 控制寄存器 USnCON0：TWEN = 1，使能 TWI
3. 配置 TWIn 控制寄存器 USnCON1：配置 TWI 通信速率（TWCK[3:0]），将起始位 STA 置“1”
4. 配置 TWIn 地址寄存器 USnCON3：将“从机地址+读写位”写入 TWDAT，总线上发出地址帧
5. 如果主机接收数据，则等待 USnCON0 中的中断标志位 TWIF 置 1。主机每接收到 8 位数据，中断标志位会被置 1。中断标志位需手动清零；
6. 如果主机发送数据，则要将待发送的数据写进 TWDAT 中，TWI 会自动将数据发送出去。每发送 8 位，中断标志位 TWIF 就会被置 1。
7. 数据收发送完成，主机可发送停止条件（STO=1），主机状态切换为 000。或发送重复起始信号，开始新一轮的数据传输。

注意：主机产生 stop 之后主机的 TWIF 不会置位！

Preliminary

## 16.3 UART

USMDn[1:0] = 11，n=0~2三选一串行接口USCI配置为UART接口。可方便用于同其它器件或者设备的连接，例如Wifi模块电路或其它UART通信接口的驱动芯片等。其功能及特性如下：

1. 三种通讯模式可选：模式0、模式1和模式3；
2. 独立波特率发生器；
3. 发送和接收完成可产生中断RI/TI，该中断标志需要软件清除，清除方式为“写1清零”；

USCI 配置为 UART 接口时：

- USTXn 作为 TX 信号
- USRXn 作为 RX 信号

**US0CON0 (95H) 串口 1 控制寄存器(读/写)**
**US1CON0 (A4H) 串口 2 控制寄存器(读/写)**
**US2CON0 (C4H) 串口 3 控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 1 清零 | 读/写 1 清零 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 6~7 | **SM0~1** | 串行通信模式控制位<br>00：模式 0，8 位半双工同步通信模式，在 RX 引脚上收发串行数据。TX 引脚用作发送移位时钟。每帧收发 8 位，低位先接收或发送；<br>01：模式 1，10 位全双工异步通信，由 1 个起始位，8 个数据位和 1 个停止位组成，通信波特率可变；<br>10：保留；<br>11：模式 3，11 位全双工异步通信，由 1 个起始位，8 个数据位，一个可编程的第 9 位和 1 个停止位组成，通信波特率可变。 |
| 5 | **SM2** | 串行通信模式控制位 2，此控制位只对模式 3 有效<br>0：每收到一个完整的数据帧就置位 RI 产生中断请求；<br>1：收到一个完整的数据帧时，只有当 RB8=1 时才会置位 RI 产生中断请求。<br>**波特率倍率设置位，仅在模式 0（SM0~1 = 00）有效：**<br>0：串行端口在系统时钟的 1/12 下运行<br>1：串行端口在系统时钟的 1/4 下运行 |
| 4 | **REN** | 接收允许控制位<br>0：不允许接收数据；<br>1：允许接收数据。 |
| 3 | **TB8** | 只对模式 3 有效，为发送数据的第 9 位 |
| 2 | **RB8** | 只对模式 3 有效，为接收数据的第 9 位 |
| 1 | **TI** | 发送中断标志位 |
| 0 | **RI** | 接收中断标志位 |

**US0CON1 (9DH) 串口 1 波特率控制寄存器低位(读/写)**
**US1CON1 (A5H) 串口 2 波特率控制寄存器低位(读/写)**
**US2CON1 (C5H) 串口 3 波特率控制寄存器低位(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | BAUDL [7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Preliminary

**US0CON2 (9EH) 串口 1 波特率控制寄存器高位(读/写)**
**US1CON2 (A6H) 串口 2 波特率控制寄存器高位(读/写)**
**US2CON2 (C6H) 串口 3 波特率控制寄存器高位(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | BAUDH [7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **BAUD1 [15:0]** | **USCI 串口波特率控制**<br>$\text{BaudRate} = \frac{\text{fsys}}{[\text{BAUDH}, \text{BAUDL}]}$<br>**注意：[BAUDH,BAUDL] 必须大于 0x0010** |

**US0CON3 (9FH) 串口 1 数据缓存寄存器(读/写)**
**US1CON3 (A7H) 串口 2 数据缓存寄存器(读/写)**
**US2CON3 (C7H) 串口 3 数据缓存寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | SBUF[7:0] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~0 | **SBUF[7:0]** | **串口数据缓存寄存器**<br>SBUF 包含两个寄存器：一个发送移位寄存器和一个接收锁存器，写入 SBUF 的数据将送至发送移位寄存器，并启动发送流程，读 SBUF 将返回接收锁存器中的内容。 |

Preliminary

# 17 模数转换 ADC

SC95F767X 集成有 13 路 12 位高精度 1M 高速 ADC，外部的 12 路 ADC 和 IO 口的其它功能复用，内部的一路可接至 1/4 $V_{DD}$，配合内部 2.048V 、1.024V 或 2.4V 参考电压用于测量 VDD 电压。1MHz 超高速采样时钟，采样到完成转换的总时间低至 2μs。

SC95F767X 的 ADC 参考电压可以有 4 种选择：

① VDD 管脚(即直接是内部的 $V_{DD}$)；

② 内部 Regulator 输出的参考电压精准的 2.048V。

③ 内部 Regulator 输出的参考电压精准的 1.024V。

④ 内部 Regulator 输出的参考电压精准的 2.4V。

## 17.1 ADC 相关寄存器

**ADCCON (ADH) ADC 控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ADCEN | ADCS | EOC/ADCIF | ADCIS[4:0] | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | n |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **ADCEN** | 启动 ADC 的电源<br>0：关闭 ADC 模块电源<br>1：开启 ADC 模块电源 |
| 6 | **ADCS** | ADC 开始触发控制（ADC Start）<br>对此 bit 写 “1”, 开始做一次 ADC 的转换，即该位只是 ADC 转换的触发信号。此位只可写入 1 有效。<br>**注意：对 ADCS 写“1”后，到中断标志 EOC/ADCIF 置起前不要对 ADCCON 寄存器进行写操作** |
| 5 | **EOC/ADCIF** | 转换完成/ADC 中断请求标志(End Of Conversion / ADC Interrupt Flag)<br>0：转换尚未完成<br>1：ADC 转换完成。需用户软件清除<br>ADC 转换完成标志 EOC：当使用者设定 ADCS 开始转换后, 此位会被硬件自动清除为 0； 当转换完成后, 此位会被硬件自动置为 1；<br>ADC 中断请求标志 ADCIF：<br>此位同时也当作是 ADC 中断的中断请求标志，如果用户使能 ADC 中断， 那么在 ADC 的 中断 发生后, 用户必须用软件清除此位。 |
| 4~0 | **ADCIS[4:0]** | ADC 输入通道选择(ADC Input Selector)<br>00000：选用 AIN0 为 ADC 的输入<br>00001：选用 AIN1 为 ADC 的输入<br>00010：选用 AIN2 为 ADC 的输入<br>00011：选用 AIN3 为 ADC 的输入<br>00100：选用 AIN4 为 ADC 的输入<br>00101：选用 AIN5 为 ADC 的输入<br>00110：选用 AIN6 为 ADC 的输入<br>00111：选用 AIN7 为 ADC 的输入<br>01000：选用 AIN8 为 ADC 的输入<br>01001：选用 AIN9 为 ADC 的输入<br>01010：选用 AIN10 为 ADC 的输入<br>01011：选用 AIN11 为 ADC 的输入<br>01100~11110：保留<br>11111： ADC 输入为 1/4 $V_{DD}$, 可用于测量电源电压 |

Preliminary

**ADCCFG2 (B5H) ADC 设置寄存器 2(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | LOWSP[2:0] | | | - | - |
| 读/写 | - | - | - | 读/写 | 读/写 | 读/写 | - | - |
| 上电初始值 | x | x | x | 0 | 0 | 0 | x | x |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 4~2 | **LOWSP[2:0]** | ADC 采样时间选择：<br>100：采样时间为 3 个 ADC 采样时钟，（约 100ns @Fsys= 32MHz）<br>101：采样时间约 6 个 ADC 采样时钟，（约 200ns @Fsys= 32MHz）<br>110：采样时间约 16 个 ADC 采样时钟，（约 500ns @Fsys= 32MHz）<br>111：采样时间约 32 个 ADC 采样时钟，（约 1000ns @Fsys= 32MHz）<br><br>说明：<br>1. ADC 采样时钟频率 Fadc = Fsys<br>2. ADC 从采样到完成转换的总时间 $T_{ADC}$= 采样时间 + 转换时间，其中转换时间固定为 950ns。 |
| 7~5，1~0 | **-** | 保留 |

**ADCCFG0 (ABH) ADC 设置寄存器 0(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | EAIN7 | EAIN6 | EAIN5 | EAIN4 | EAIN3 | EAIN2 | EAIN1 | EAIN0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**ADCCFG1 (ACH) ADC 设置寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | - | - | - | EAIN11 | EAIN10 | EAIN9 | EAIN8 |
| 读/写 | - | - | - | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | x | x | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 0 | **EAINx (x=0~11)** | **ADC 端口设置寄存器**<br>0：设定 AINx 为 IO 口<br>1：设定 AINx 为 ADC 输入，并自动将上拉电阻移除。 |

**OP_CTM1 (C2H@FFH) Customer Option 寄存器 1(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | VREFS[1:0] | | OP_BL | DISJTG | IAPS[1:0] | | LDSIZE[1:0] | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 只读 | |
| 上电初始值 | n | n | n | n | n | n | n | n |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7~6 | **VREFS[1:0]** | **参考电压选择(初始值从 Code Option 调入，用户可修改设置)**<br>00：设定 ADC 的 VREF 为 VDD；<br>01：设定 ADC 的 VREF 为内部准确的 2.048V；<br>10：设定 ADC 的 VREF 为内部准确的 1.024V；<br>11：设定 ADC 的 VREF 为内部准确的 2.4V |

**ADCVL (AEH) ADC 转换数值寄存器(低位)(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|

Preliminary

| 符号 | ADCV[3:0] | | | | - | - | - | - |
|---|---|---|---|---|---|---|---|---|
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | - | - | - | - |
| 上电初始值 | 0 | 0 | 0 | 0 | x | x | x | x |

**ADCVH (AFH) ADC 转换数值寄存器(高位)(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | ADCV[11:4] | | | | | | | |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 11~4 | **ADCV[11:4]** | ADC 转换值的高 8 位数值 |
| 3~0 | **ADCV[3:0]** | ADC 转换值的低 4 位数值 |

**IE (A8H) 中断使能寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | EA | EADC | ET2 | - | ET1 | EINT1 | ET0 | EINT0 |
| 读/写 | 读/写 | 读/写 | 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | 0 | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 6 | **EADC** | ADC 中断使能控制<br>0： 不允许 EOC/ADCIF 产生中断<br>1： 允许 EOC/ADCIF 产生 中断 |

**IP (B8H) 中断优先权寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | - | IPADC | IPT2 | - | IPT1 | IPINT1 | IPT0 | IPINT0 |
| 读/写 | - | 读/写 | 读/写 | - | 读/写 | 读/写 | 读/写 | 读/写 |
| 上电初始值 | x | 0 | 0 | x | 0 | 0 | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 6 | **IPADC** | ADC 中断优先权选择<br>0：设定 ADC 的中断优先级是 “低”<br>1：设定 ADC 的中断优先级是 “高” |

## 17.2 ADC 转换步骤

用户实际进行 ADC 转换所需要的操作步骤如下：

① 设定 ADC 输入管脚；（设定 AINx 对应的位为 ADC 输入，通常 ADC 管脚会预先固定）

② 设定 ADC 参考电压 Vref，设定 ADC 转换所用的频率；

③ 开启 ADC 模块电源；

④ 选择 ADC 输入通道；(设置 ADCIS 位，选择 ADC 输入通道)

⑤ 启动 ADCS，转换开始；

⑥ 等待 EOC/ADCIF=1，如果 ADC 中断使能，则 ADC 中断会产生，用户需要软件清 0 EOC/ADCIF 标志；

⑦ 从 ADCVH、ADCVL 获得 12 位数据，先高位后低位，一次转换完成；

⑧ 如不换输入通道，则重复 5~7 的步骤，进行下一次转换。

**注意：在设定 IE[6](EADC)前，使用者最好用软件先清除 EOC/ADCIF，并且在 ADC 中断服务程序执行完时，也清除该 EOC/ADCIF，以避免不断的产生 ADC 中断。**

Preliminary

# 18 硬件 CRC 模块

SC95F767X 内建了 1 个硬件 CRC 模块，CRC 执行计算过程中，CPU 保持程序计数器，CRC 计算完成后，程序计数器才继续执行之后的指令。

该模块有两种计算模式：

**硬件 CRC 模式一：对指定数据进行 CRC 运算处理：**

将需要进行 CRC 计算的数据写入 CRC 数据寄存器 CRCREG，当需要读取 CRC 计算结果时，再从 CRCDRn(n = 0~3)读出。

单独一个 byte 计算 CRC 需要 8 个系统时钟，即 0.25μs@32MHz。

**硬件 CRC 模式二：对 APROM 进行 CRC 运算处理：**

可用来实时生成 APROM（即 64 Kbytes Flash ROM）的 32 位 CRC 值，该值和理论值比较，可监测程序区的内容是否正确。CRC 理论值不需要用户计算，烧录软件会根据载入的代码及 Code 区域设置项自动完成计算并在烧录时通过烧写器将 4 bytes 的 CRC32 计算结果写入 CRC 结果存储区，具体操作方法见烧录工具使用手册。

对 64 Kbytes APROM 计算 CRC 需要约 16.5ms@32MHz。

**注意：在 LDROM 中启动硬件 CRC 时，模式二无效。**

**SC95F767X 的硬件 CRC 参数模型：**

| | |
|---|---|
| CRC 算法名称 | CRC-32/MPEG-2 |
| 多项式公式 | $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$ |
| 数据宽度 | 32bit |
| 初始值 | 0xFFFFFFFF |
| 结果异或值 | 0x00000000 |
| 输入值反转 | false |
| 输出值反转 | false |
| LSB/MSB | MSB |

**CRC 使用注意事项：**

1. CRCDRn 写入数据和读出不是同一数据；
2. 硬件计算所得的 CRC 值是整个程序区数据（注意，这里不包括 IAP 区域！）的 32 位 CRC 校验值。若地址单元中有用户上次操作后的残留值，会导致 CRC 值与理论值不符。因此，建议用户对整片 Flash ROM 进行擦除后再烧录代码以保证 CRC 值与理论值一致；
3. 硬件 CRC 计算范围不包含 IAP 区域；
4. 用户如需进行 CRC 运算操作，请参考《SC95FXX7X CRC 操作库资料包》；

## 18.1 CRC 校验操作相关寄存器

**OPERCON (EFH) 运算控制寄存器(读写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | OPERS | MD | - | - | - | - | CRCRST | CRCSTA |
| 读/写 | 读/写 | 读/写 | - | - | - | - | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | x | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 1 | CRCRST | CRCDR 寄存器复位(Q31~Q0)<br>对此 bit 写 “1”，即可将 CRCDR 复位为全 1 |
| 0 | CRCSTA | CRC 硬件计算启动位<br>对此 bit 写 “1”，开始做一次 Check sum 计算。此位只可写入 1 有效。 |

CRC 的数据寄存器 CRCDRn(n = 0~3)的读写操作由 CRCINX 和 CRCREG 两个寄存器进行控制，各 CRCDRn 的具体位置由 CRCINX 确定，如下表所示：

| 符号 | 地址 | 说明 | | 上电初始值 |
|---|---|---|---|---|
| CRCINX | FCH | CRC 指针 | CRCINX[7:0] | 00000000b |
| CRCREG | FDH | CRC 寄存器 | CRCREG[7:0] | nnnnnnnnb |

| 符号 | 地址 | 说明 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|



Preliminary

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| CRCDR3 | 03H@FDH | CRC 数据寄存器 3 | Q31 | Q30 | Q29 | Q28 | Q27 | Q26 | Q25 | Q24 |
| CRCDR2 | 02H@FDH | CRC 数据寄存器 2 | Q23 | Q22 | Q21 | Q20 | Q19 | Q18 | Q17 | Q16 |
| CRCDR1 | 01H@FDH | CRC 数据寄存器 1 | Q15 | Q14 | Q13 | Q12 | Q11 | Q10 | Q9 | Q8 |
| CRCDR0 | 00H@FDH | CRC 数据寄存器 0 | Q7 | Q6 | Q5 | Q4 | Q3 | Q2 | Q1 | Q0 |

CRCDRn(n = 0~3)的 bit 位相关描述如下：

| 编号位 | 符号位 | 说明 |
|---|---|---|
| Bit31~0 | Qx<br>(x = 0~31) | 硬件 CRC 模式一：对指定数据进行 CRC 运算处理：<br>1. 首先必须写 CRCRST，将 CRCDR 复位为全 1<br>2. 当 CRCREG 被写入时，硬件自动计算 CRC 结果，并继续存放于 CRCDR 内<br>3. 当需要时，即时读出 CRC 计算结果<br><br>硬件 CRC 模式二：对 APROM 进行 CRC 运算处理：<br>4. 由 CRCSTA 启动，此时 CPU 自动进入 IDLE<br>5. 自动复位 CRCDR 为全 1：<br>**硬件 CRC 计算范围不包含 IAP 区域。**CRC 的计算范围根据 IAPS[1:0]的值分为四种：<br>① IAPS[1:0]=00（Flash ROM 最后 0K 可 IAP）：0000H ~ 最后 0K 之前<br>② IAPS[1:0]=01（Flash ROM 最后 1K 可 IAP）：0000H ~ 最后 1K 之前<br>③ IAPS[1:0]=10（Flash ROM 最后 2K 可 IAP）：0000H ~ 最后 2K 之前<br>④ IAPS[1:0]=11（Flash ROM 全部可 IAP）：全部 Flash ROM<br>6. 结束后 CPU 自动退出 IDLE，即可读取 CRC 计算结果<br><br>注意：写入数据和读出不是同一数据。 |

操作 CRC 相关 SFR 时 CRCINX 寄存器存放相关 CRCTION 寄存器的地址，CRCREG 寄存器存放对应的值。读 CRCREG 前需要先设定 CRCINX 再读取，每读完一次，CRCINX 自动加 1(0~3 循环)。

**硬件 CRC 校验操作例程**

用户如需进行 CRC 运算操作，请参考《SC95FXX7X CRC 操作库资料包》

**注意：禁止向 CRCINX 寄存器写入 CRC 寄存器地址之外的数值！否则会造成系统运行异常！**

# 19 乘除法器

SC95F767X 提供了 1 个 16 位的乘除法器，由扩展累加器 EXA0~EXA3、扩展 B 寄存器 EXB 和运算控制寄存器 OPERCON 组成。可取代软件进行 16 位×16 位乘法运算和 32 位/16 位除法运算。

| 符号 | 地址 | 说明 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Reset 值 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| EXA0 | E9H | 扩展累加器 0 | EXA [7:0] | | | | | | | | 00000000b |
| EXA1 | EAH | 扩展累加器 1 | EXA [15:8] | | | | | | | | 00000000b |
| EXA2 | EBH | 扩展累加器 2 | EXA [23:16] | | | | | | | | 00000000b |
| EXA3 | ECH | 扩展累加器 3 | EXA [31:24] | | | | | | | | 00000000b |
| EXBL | EDH | 扩展 B 寄存器 L | EXB [7:0] | | | | | | | | 00000000b |
| EXBH | EEH | 扩展 B 寄存器 H | EXB [15:8] | | | | | | | | 00000000b |

**OPERCON (EFH) 运算控制寄存器(读/写)**

| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 符号 | OPERS | MD | - | - | - | - | CRCRST | CRCSTA |
| 读/写 | 读/写 | 读/写 | - | - | - | - | 读/写 | 读/写 |
| 上电初始值 | 0 | 0 | x | x | x | x | 0 | 0 |

| 位编号 | 位符号 | 说明 |
|---|---|---|
| 7 | **OPERS** | 乘除法器运算开始触发控制（Operater Start） |

Preliminary

<table>
<tr><td></td><td></td><td>对此 bit 写 “1”, 开始做一次乘除法计算，即该位只是乘除法器开始计算的触发信号，当该位为零这说明计算已完成。此位只可写入 1 有效。</td></tr>
<tr><td>6</td><td>MD</td><td>乘除法选择<br>0：乘法运算，被乘数和乘数的写入、乘积的读取如下：
<table>
<tr><td>字节<br>运算数</td><td>字节 3</td><td>字节 2</td><td>字节 1</td><td>字节 0</td></tr>
<tr><td>被乘数 16bit</td><td>-</td><td>-</td><td>EXA1</td><td>EXA0</td></tr>
<tr><td>乘数 16bit</td><td>-</td><td>-</td><td>EXBH</td><td>EXBL</td></tr>
<tr><td>乘积 32bit</td><td>EXA3</td><td>EXA2</td><td>EXA1</td><td>EXA0</td></tr>
</table>
1：除法运算，被除数和除数的写入、商和余数的读取如下：
<table>
<tr><td>字节<br>运算数</td><td>字节 3</td><td>字节 2</td><td>字节 1</td><td>字节 0</td></tr>
<tr><td>被除数 32bit</td><td>EXA3</td><td>EXA2</td><td>EXA1</td><td>EXA0</td></tr>
<tr><td>除数 16bit</td><td>-</td><td>-</td><td>EXBH</td><td>EXBL</td></tr>
<tr><td>商 32bit</td><td>EXA3</td><td>EXA2</td><td>EXA1</td><td>EXA0</td></tr>
<tr><td>余数 16bit</td><td>-</td><td>-</td><td>EXBH</td><td>EXBL</td></tr>
</table></td></tr>
</table>

注：
1. 在执行运算操作过程中，禁止对 EXA 和 EXB 数据寄存器执行读或写动作。
2. 乘除法器运算转换所需时间为 16/$f_{SYS}$。

Preliminary

# 20 电气特性

## 20.1 极限参数

| 符号 | 参数 | 最小值 | 最大值 | UNIT |
|---|---|---|---|---|
| VDD/VSS | 直流供电电压 | -0.3 | 5.5 | V |
| Voltage ON any Pin | 任一管脚输入/输出电压 | -0.3 | $V_{DD}$+0.3 | V |
| $T_A$ | 工作环境温度 | -40 | 105 | ℃ |
| $T_{STG}$ | 储存温度 | -55 | 125 | ℃ |
| $I_{VDD}$ | 流过 VDD 的电流值 | - | 200 | mA |
| $I_{VSS}$ | 流过 VSS 的电流值 | - | 200 | mA |

## 20.2 推荐工作条件

| 符号 | 参数 | 最小值 | 最大值 | UNIT | 系统时钟频率 |
|---|---|---|---|---|---|
| $V_{DD}$ | 工作电压 | 2.0 | 5.5 | V | 32MHz |
| $T_A$ | 工作环境温度 | -40 | 105 | ℃ | |

## 20.3 Flash ROM 参数

| 符号 | 参数 | 最小值 | 典型值 | 最大值 | UNIT | 条件 |
|---|---|---|---|---|---|---|
| $N_{END}$ | 擦写次数 | 100,000 | - | - | Cycles | |
| $T_{DR}$ | 数据保存时间 | 100 | - | - | Years | $T_A$ = +25℃ |
| $T_{S\text{-}Erase}$ | 单个 Sector 擦除时间 | - | 5 | - | ms | $T_A$ = +25℃ |
| $T_{Erase}$ | 全擦时间 | 30 | - | 40 | ms | $T_A$ = +25℃ |
| $T_{Write}$ | 单个 byte 写入时间 | - | 30 | - | μs | $T_A$ = +25℃ |

## 20.4 直流电气特性

**($V_{DD}$ = 5V，$T_A$ = +25℃，除非另有说明)**

| 符号 | 参数 | 最小值 | 典型值 | 最大值 | 单位 | 测试条件 |
|---|---|---|---|---|---|---|
| 电流 | | | | | | |
| $I_{op1}$ | 工作电流 | - | 4.5 | - | mA | $f_{SYS}$=32MHz |
| $I_{op2}$ | 工作电流 | - | 3 | - | mA | $f_{SYS}$=16MHz |
| $I_{op3}$ | 工作电流 | - | 2 | - | mA | $f_{SYS}$=8MHz |
| $I_{op4}$ | 工作电流 | - | 1.5 | - | mA | $f_{SYS}$=4MHz |
| $I_{pd1}$ | 待机电流（Power Down 模式） | - | 2 | - | μA | |
| $I_{IDL1}$ | 待机电流（IDLE 模式） | - | 3 | - | mA | $f_{SYS}$=32MHz |
| $I_{BTM}$ | Base Timer 工作电流 | - | 1.3 | 3 | μA | BTMFS[3:0]= 1000 每 4.0 s 产生一个中断 |
| $I_{WDT}$ | WDT 电流 | - | 1.3 | 3 | μA | WDTCKS[2:0]= 000 WDT 溢出时间 500ms |
| IO 口特性 | | | | | | |
| $V_{IH1}$ | 输入高电压 | 0.7$V_{DD}$ | - | $V_{DD}$+0.3 | V | GPIO |
| $V_{IL1}$ | 输入低电压 | -0.3 | - | 0.3$V_{DD}$ | V | |

Preliminary

| | | | | | | |
|---|---|---|---|---|---|---|
| $V_{IH2}$ | 输入高电压 | $0.8V_{DD}$ | - | $V_{DD}$ | V | 施密特触发输入:<br>RST<br>tCK / tDIO<br>UART0 输入 RX0<br>USCI 信号输入口<br>INT0~2<br>PWM 故障检 FLT<br>Timer 时钟输入口<br>Timer 捕获口 |
| $V_{IL2}$ | 输入低电压 | -0.2 | - | $0.2V_{DD}$ | V | |
| $I_{OL1}$ | 输出低电流 | - | 27 | - | mA | $V_{Pin}$=0.4V |
| $I_{OL2}$ | 输出低电流 | - | 50 | - | mA | $V_{Pin}$=0.8V |
| $I_{OHSPI0A}$ | SPI0 信号口：<br>USCK0（P05）<br>USTX0（P20）<br>USRX0（P21）<br>输出高电流@ $V_{Pin}$=4.3V | - | 22 | - | mA | 仅适用于 SPI0 数据传输时 |
| $I_{OH1}$ | 输出高电流@ $V_{Pin}$=4.3V | - | 11 | - | mA | Pxyz=0,$I_{OH}$ 等级 0 |
| | | - | 9 | - | mA | Pxyz=1,$I_{OH}$ 等级 1 |
| | | - | 6 | - | mA | Pxyz=2,$I_{OH}$ 等级 2 |
| | | - | 3 | - | mA | Pxyz=3,$I_{OH}$ 等级 3 |
| $I_{OHSPI0B}$ | SPI0 信号口：<br>USCK0（P05）<br>USTX0（P20）<br>USRX0（P21）<br>输出高电流@ $V_{Pin}$=4.7V | - | 12 | - | mA | 仅适用于 SPI0 数据传输时 |
| $I_{OH2}$ | 输出高电流@ $V_{Pin}$=4.7V | - | 6 | - | mA | Pxyz=0,$I_{OH}$ 等级 0 |
| | | - | 4 | - | mA | Pxyz=1,$I_{OH}$ 等级 1 |
| | | - | 3 | - | mA | Pxyz=2,$I_{OH}$ 等级 2 |
| | | - | 1 | - | mA | Pxyz=3,$I_{OH}$ 等级 3 |
| $R_{PH1}$ | 上拉电阻 | - | 30 | - | kΩ | |

**($V_{DD}$ = 3.3V，$T_A$ = +25℃， 除非另有说明)**

| 符号 | 参数 | 最小值 | 典型值 | 最大值 | 单位 | 测试条件 |
|---|---|---|---|---|---|---|
| 电流 | | | | | | |
| $I_{op5}$ | 工作电流 | - | 4 | - | mA | $f_{SYS}$=32MHz |
| $I_{op6}$ | 工作电流 | - | 3 | - | mA | $f_{SYS}$=16MHz |
| $I_{op7}$ | 工作电流 | - | 2 | - | mA | $f_{SYS}$=8MHz |
| $I_{op8}$ | 工作电流 | - | 1.5 | - | mA | $f_{SYS}$=4MHz |
| $I_{pd2}$ | 待机电流<br>（Power Down 模式） | - | 2 | - | μA | |
| $I_{IDL2}$ | 待机电流<br>（IDLE 模式） | - | 3 | - | mA | $f_{SYS}$=32MHz |
| IO 口特性 | | | | | | |
| $V_{IH3}$ | 输入高电压 | $0.7V_{DD}$ | - | $V_{DD}$+0.3 | V | |
| $V_{IL3}$ | 输入低电压 | -0.3 | - | $0.3V_{DD}$ | V | |
| $V_{IH4}$ | 输入高电压 | $0.8V_{DD}$ | - | $V_{DD}$ | V | 施密特触发输入:<br>RST<br>tCK / tDIO<br>UART0 输入 RX0<br>USCI 信号输入口<br>INT0~2<br>PWM 故障检 FLT<br>Timer 时钟输入口<br>Timer 捕获口 |
| $V_{IL4}$ | 输入低电压 | -0.2 | - | $0.2V_{DD}$ | V | |
| $I_{OL3}$ | 输出低电流 | - | 20 | - | mA | $V_{Pin}$=0.4V |

Preliminary

| | | | | | | |
|---|---|---|---|---|---|---|
| $I_{OL4}$ | 输出低电流 | - | 35 | - | mA | $V_{Pin}$=0.8V |
| $I_{OHSPI0C}$ | SPI0 信号口：<br>USCK0（P05）<br>USTX0（P20）<br>USRX0（P21）<br>输出高电流@ $V_{Pin}$=3.0V | - | 8.5 | - | mA | 仅适用于 SPI0 数据传输时 |
| $I_{OH3}$ | 输出高电流@ $V_{Pin}$=3.0V | - | 4 | - | mA | Pxyz=0,$I_{OH}$ 等级 0 |
| | | - | 3 | - | mA | Pxyz=1,$I_{OH}$ 等级 1 |
| | | - | 2 | - | mA | Pxyz=2,$I_{OH}$ 等级 2 |
| | | - | 1 | - | mA | Pxyz=3,$I_{OH}$ 等级 3 |
| $R_{PH2}$ | 上拉电阻 | - | 55 | - | kΩ | |

## 20.5 交流电气特性

**($V_{DD}$ = 2.4V ~ 5.5V，$T_A$ = 25℃， 除非另有说明)**

| 符号 | 参数 | 最小值 | 典型值 | 最大值 | 单位 | 测试条件 |
|---|---|---|---|---|---|---|
| $T_{osc1}$ | 外接高频振荡器起振时间 | - | 9 | - | ms | 外接 16MHz 晶振 |
| $T_{osc2}$ | 外接高频振荡器起振时间 | - | 18 | - | ms | 外接 8MHz 晶振 |
| $T_{osc3}$ | 外接高频振荡器起振时间 | - | 35 | - | ms | 外接 4MHz 晶振 |
| $T_{POR}$ | Power On Reset 时间 | - | 15 | - | ms | |
| $T_{PDW}$ | Power Down 模式唤醒时间 | - | 60 | 130 | μs | |
| $T_{Reset}$ | 复位脉冲宽度 | 18 | - | - | μs | 低电平有效 |
| $T_{LVR}$ | LVR 消抖时间 | - | 30 | - | μs | |
| $f_{HRC}$ | RC 振荡稳定性 | 31.68 | 32 | 32.32 | MHz | $V_{DD}$=2.0~5.5V<br>$T_A$=-40~105 ℃ |

## 20.6 ADC 电气特性

**($T_A$ = 25℃，除非另有说明)**

| 符号 | 参数 | 最小值 | 典型值 | 最大值 | 单位 | 测试条件 |
|---|---|---|---|---|---|---|
| $V_{AD1}$ | 供电电压 1 | 2.7 | 5.0 | 5.5 | V | Vref = 2.048V |
| $V_{AD2}$ | 供电电压 2 | 2.0 | 5.0 | 5.5 | V | Vref = 1.024V<br>或<br>Vref = $V_{DD}$ |
| $V_{AD3}$ | 供电电压 3 | 2.7 | 5.0 | 5.5 | V | Vref = 2.4V |
| $V_{REF1}$ | 内部基准 2.048V | 2.028 | 2.048 | 2.070 | V | $V_{DD}$ = 2.7~5.5V |
| $V_{REF2}$ | 内部基准 1.024V | 1.004 | 1.024 | 1.044 | V | $V_{DD}$ = 2.0~5.5V |
| $V_{REF3}$ | 内部基准 2.4V | 2.37 | 2.40 | 2.45 | V | $V_{DD}$ = 2.7~5.5V |
| $N_R$ | 精度 | - | 12 | - | bit | GND≤$V_{AIN}$≤$V_{DD}$ |
| $V_{AIN}$ | ADC 输入电压 | GND | - | $V_{DD}$ | V | |
| $R_{AIN}$ | ADC 输入电阻 | 1 | - | - | MΩ | $V_{IN}$=5V |
| $I_{ADC1}$ | ADC 转换电流 1 | - | - | 2 | mA | ADC 模块打开<br>$V_{DD}$=5V |
| $I_{ADC2}$ | ADC 转换电流 2 | - | - | 1.8 | mA | ADC 模块打开<br>$V_{DD}$=3.3V |
| DNL | 微分非线性误差 | - | - | ±3 | LSB | $V_{DD}$=5V<br>$V_{REF}$=5V |
| INL | 积分非线性误差 | - | - | ±3 | LSB | |
| $E_Z$ | 偏移量误差 | - | ±4 | - | LSB | |

Preliminary

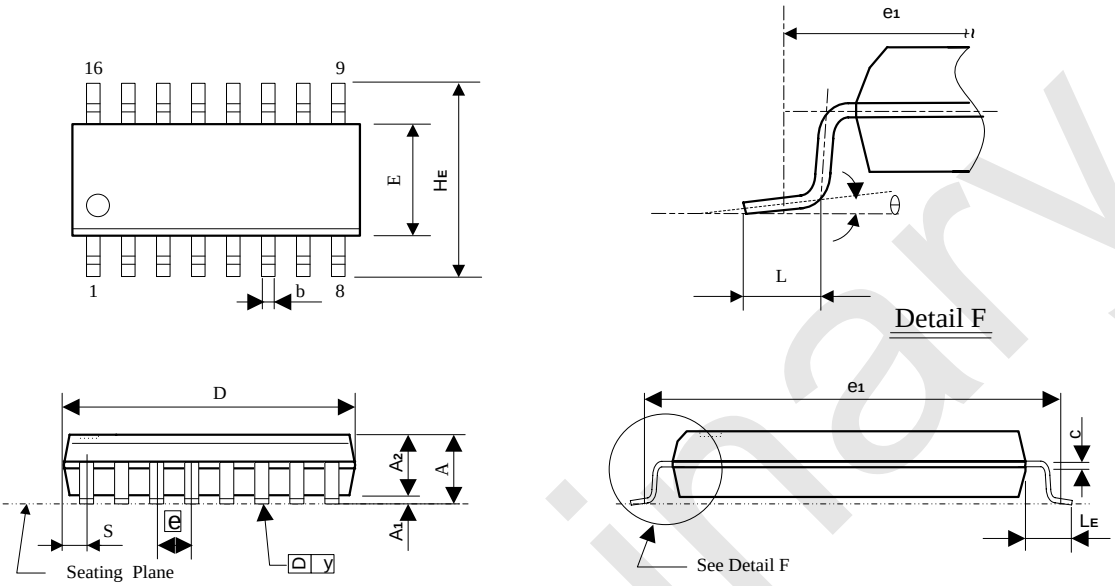
| | | | | | | |
|---|---|---|---|---|---|---|
| $E_F$ | 满刻度误差 | - | ±4 | - | LSB | |
| $E_{AD}$ | 总绝对误差 | - | ±4 | - | LSB | |
| $T_{ADC1}$ | ADC 转换时间 1 | - | 1.1 | 1.4 | μs | $f_{SYS}$=32MHz LOWSP[2:0] = 100 |
| $T_{ADC2}$ | ADC 转换时间 2 | - | 1.2 | 1.5 | μs | $f_{SYS}$=32MHz LOWSP[2:0] = 101 |
| $T_{ADC3}$ | ADC 转换时间 3 | - | 1.5 | 1.9 | μs | $f_{SYS}$=32MHz LOWSP[2:0] = 110 |
| $T_{ADC4}$ | ADC 转换时间 4 | - | 2.0 | 2.6 | μs | $f_{SYS}$=32MHz LOWSP[2:0] = 111 |

Preliminary

# 21 封装信息

**SC95F7671M16U**

**SOP16L(150mil)** 外形尺寸 单位:毫米

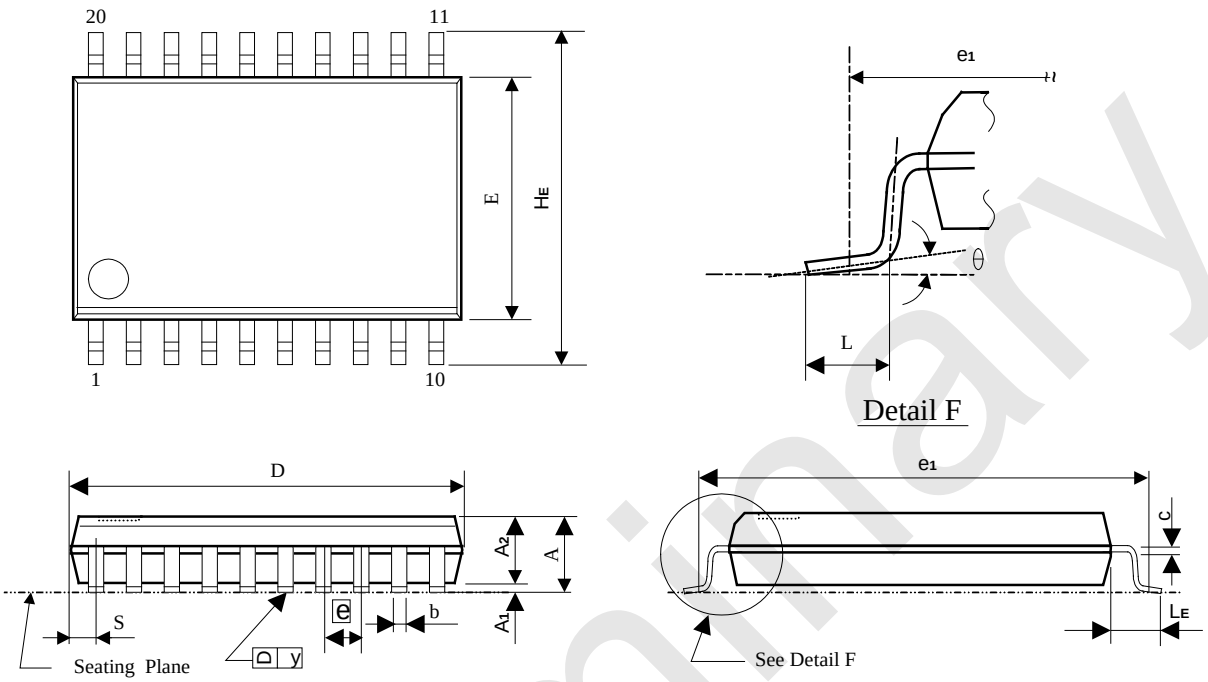



| 符号 | mm(毫米) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 1.500 | 1.625 | 1.750 |
| A1 | 0.050 | 0.1375 | 0.225 |
| A2 | 1.30 | 1.45 | 1.55 |
| b | 0.38 | 0.43 | 0.48 |
| c | 0.20 | 0.23 | 0.26 |
| D | 9.70 | 9.90 | 10.10 |
| E | 3.70 | 3.90 | 4.10 |
| HE | 5.80 | 6.00 | 6.20 |
| e | 1.27(BSC) | | |
| L | 0.50 | 0.65 | 0.80 |
| LE | 0.95 | 1.05 | 1.15 |
| θ | 0° | - | 8° |

Preliminary

**SC95F7672M20U**

**SOP20L(300mil)** 外形尺寸 单位:毫米

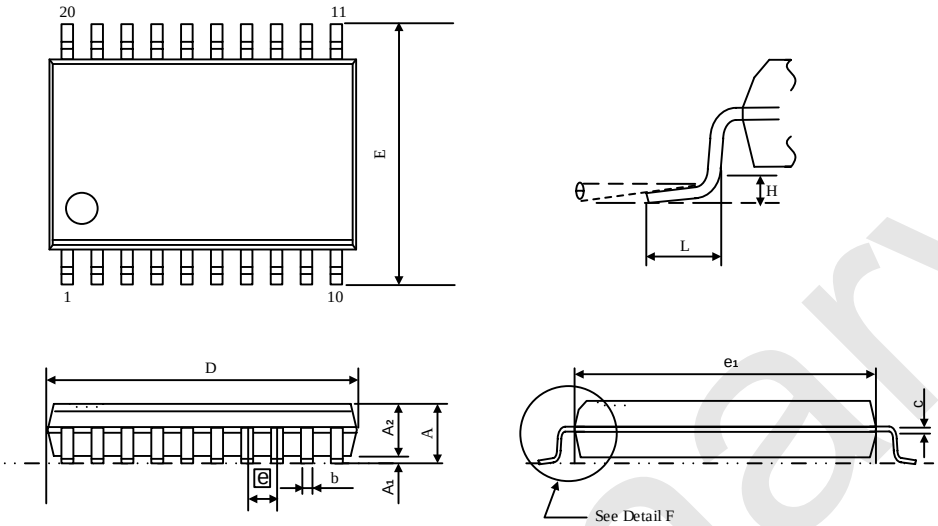



| 符号 | mm(毫米) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 2.40 | 2.56 | 2.65 |
| A1 | 0.100 | 0.200 | 0.300 |
| A2 | 2.240 | 2.340 | 2.440 |
| b | 0.35 | -- | 0.47 |
| c | 0.25 | -- | 0.31 |
| D | 12.60 | 12.80 | 13.00 |
| E | 7.30 | 7.50 | 7.70 |
| HE | 10.100 | 10.300 | 10.500 |
| e | 1.27(BSC) | | |
| L | 0.700 | 0.850 | 1.000 |
| LE | 1.30 | 1.40 | 1.50 |
| θ | 0° | - | 8° |

Preliminary

**SC95F7672X20U**

**TSSOP20L** 外形尺寸 单位:毫米

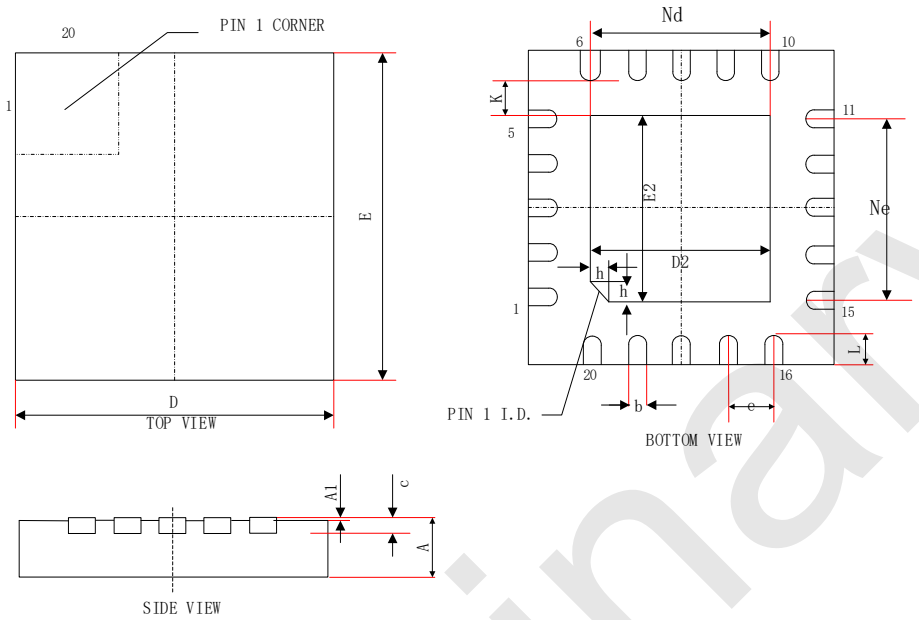



| 符号 | mm(毫米) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | - | - | 1.200 |
| A**1** | 0.050 | - | 0.150 |
| A**2** | 0.800 | - | 1.050 |
| b | 0.190 | - | 0.300 |
| c | 0.090 | - | 0.200 |
| D | 6.400 | - | 6.600 |
| E | 6.20 | - | 6.60 |
| e1 | 4.300 | - | 4.500 |
| e | 0.65(BSC) | | |
| L | - | - | 1.00 |
| θ | 0° | - | 8° |
| H | 0.05 | - | 0.15 |

Preliminary

**SC95F7672Q20R**

**QFN20 L(3*3)** 外形尺寸　　单位:毫米

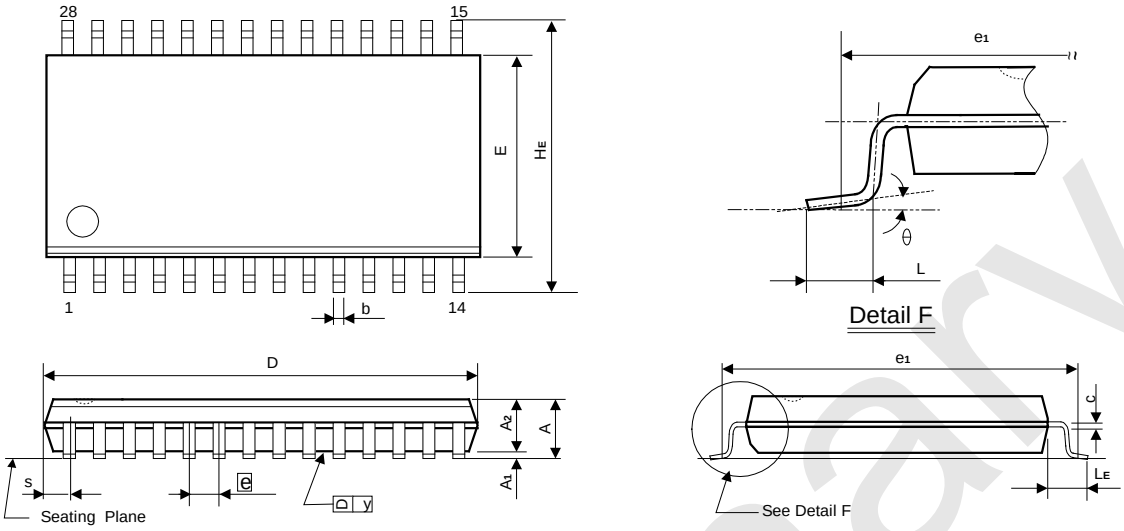



| 符号 | mm(毫米) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.50 | 0.55 | 0.60 |
| A1 | 0 | 0.02 | 0.05 |
| b | 0.15 | 0.20 | 0.25 |
| c | 0.15REF | | |
| D | 2.90 | 3.00 | 3.10 |
| D2 | 1.60 | 1.70 | 1.80 |
| e | 0.40BSC | | |
| Ne | 1.60BSC | | |
| Nd | 1.60BSC | | |
| E | 2.90 | 3.00 | 3.10 |
| E2 | 1.60 | 1.70 | 1.80 |
| L | 0.25 | 0.30 | 0.35 |
| h | 0.20 | 0.25 | 0.30 |
| K | 0.30 | 0.35 | 0.40 |

Preliminary

**SC95F7673M28U**

**SOP28L(300mil)外形尺寸　单位：毫米**

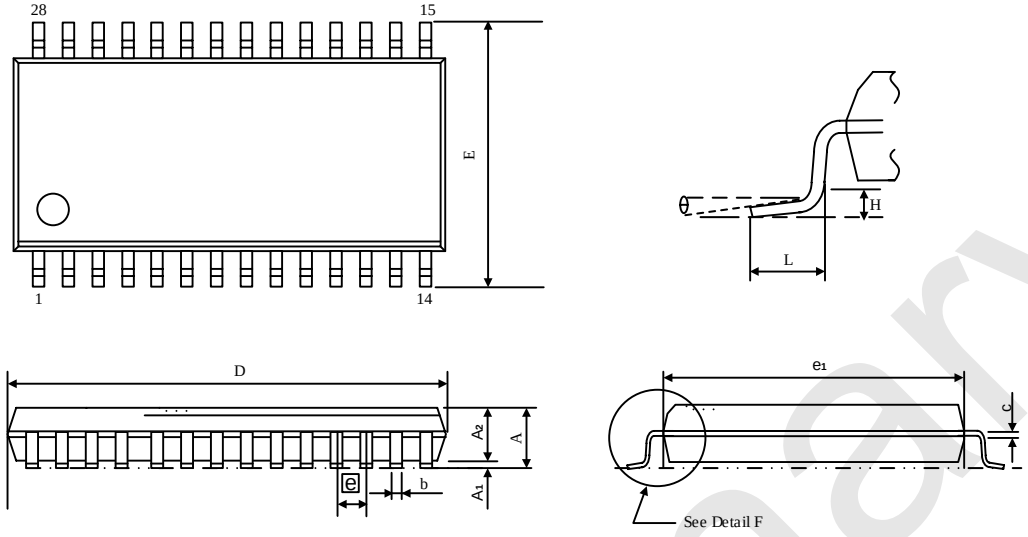



| 符号 | mm(毫米) | | |
|---|---|---|---|
| | 最小 | 标准 | 最大 |
| A | 2.40 | 2.56 | 2.65 |
| A**1** | 0.100 | 0.200 | 0.300 |
| A**2** | 2.240 | 2.340 | 2.440 |
| b | 0.39 | --- | 0.48 |
| C | 0.254(BSC) | | |
| D | 17.80 | 18.00 | 18.20 |
| E | 7.30 | 7.50 | 7.70 |
| HE | 10.100 | 10.300 | 10.500 |
| e | 1.270(BSC) | | |
| L | 0.7 | 0.85 | 1.0 |
| LE | 1.3 | 1.4 | 1.5 |
| θ | 0° | - | 8° |

Preliminary

**SC95F7673X28U**

**TSSOP28L** 外形尺寸　　单位：毫米

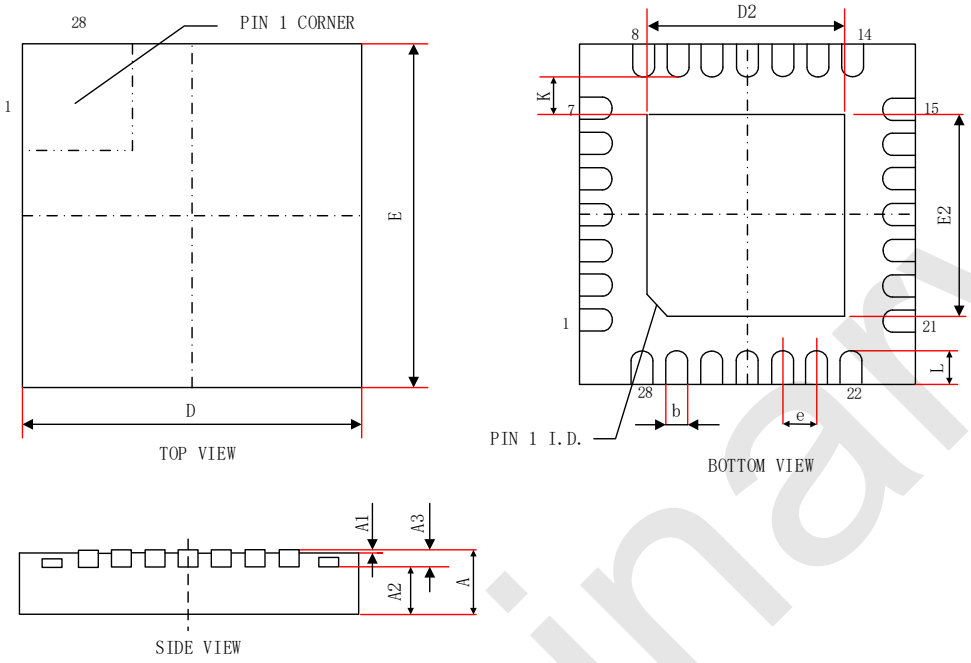



| 符号 | mm(毫米) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | - | - | 1.200 |
| A**1** | 0.050 | - | 0.150 |
| A**2** | 0.800 | - | 1.050 |
| b | 0.190 | - | 0.300 |
| c | 0.090 | - | 0.200 |
| D | 9.600 | - | 9.800 |
| E | 6.250 | - | 6.550 |
| e1 | 4.300 | - | 4.500 |
| e | 0.65(BSC) | | |
| L | - | - | 1.0 |
| θ | 0° | - | 8° |
| H | 0.05 | - | 0.25 |

Preliminary

**SC95F7673Q28R**

**QFN28(4X4)外形尺寸　　单位：毫米**

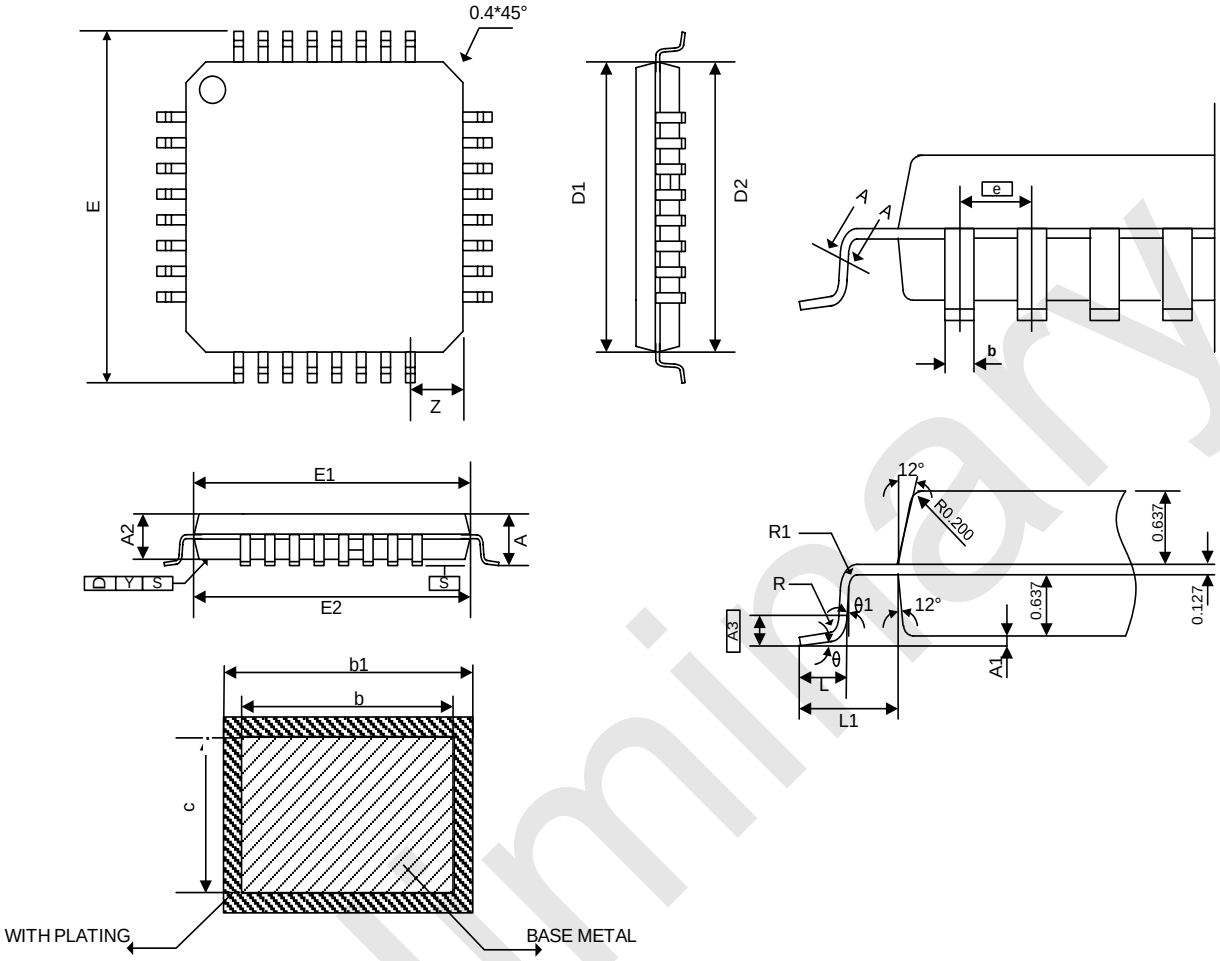



| 符号 | mm(毫米) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.7 | 0.75 | 0.8 |
| A1 | 0 | 0.02 | 0.05 |
| A2 | - | 0.55 | - |
| A3 | 0.203 REF | | |
| b | 0.18 | 0.23 | 0.28 |
| D | 3.90 | 4.00 | 4.10 |
| E | 3.90 | 4.00 | 4.10 |
| e | 0.45 BSC | | |
| D2 | 2.5 | 2.6 | 2.7 |
| E2 | 2.5 | 2.6 | 2.7 |
| L | 0.25 | 0.35 | 0.45 |
| K | 0.35 REF | | |

Preliminary

**SC95F7675P32R**

**LQFP32 (7X7)外形尺寸** **单位: 毫米**




| 符号 | mm(毫米) | | |
|---|---|---|---|
| | 最小 | 标准 | 最大 |
| A | 1.45 | 1.55 | 1.65 |
| A1 | 0.01 | -- | 0.21 |
| A2 | 1.30 | 1.4 | 1.5 |
| A3 | -- | 0.254 | -- |
| b | 0.30 | 0.35 | 0.41 |
| b1 | 0.31 | 0.37 | 0.43 |
| c | 0.12 | 0.13 | 0.14 |
| D1 | 6.85 | 6.95 | 7.05 |
| D2 | 6.90 | 7.00 | 7.10 |
| E | 8.80 | 9.00 | 9.20 |
| E1 | 6.85 | 6.95 | 7.05 |
| E2 | 6.90 | 7.00 | 7.10 |
| e | -- | 0.8 | -- |
| L | 0.43 | -- | 0.75 |
| L1 | 0.90 | 1.0 | 1.10 |
| R | 0.1 | -- | 0.25 |

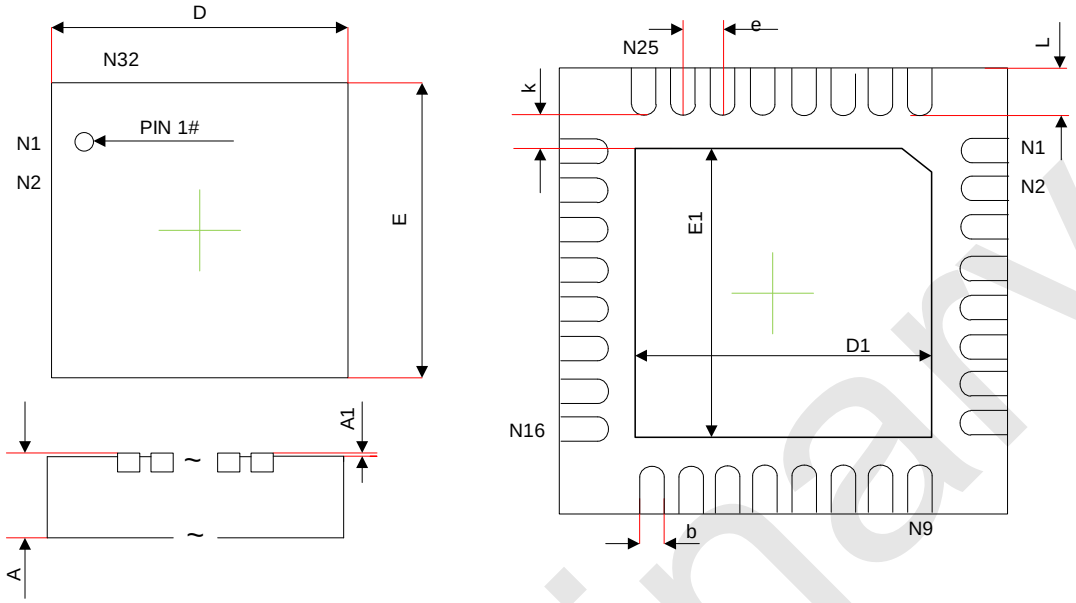
Preliminary

| | | | |
|---|---|---|---|
| R1 | 0.1 | -- | -- |
| θ | 0° | -- | 10° |
| θ1 | 0° | -- | -- |
| y | -- | -- | 0.1 |
| Z | -- | 0.70 | -- |

Preliminary

**SC95F7675Q32R**

**QFN32 (5X5)外形尺寸　　单位：毫米**




| 符号 | mm(毫米) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.70 | 0.75 | 0.80 |
| A1 | -- | 0.02 | 0.05 |
| b | 0.18 | 0.25 | 0.30 |
| D | 4.90 | 5.00 | 5.10 |
| E | 4.90 | 5.00 | 5.10 |
| e | 0.5 BSC | | |
| k | 0.4 REF | | |
| D1 | 3.30 | 3.45 | 3.60 |
| E1 | 3.30 | 3.45 | 3.60 |
| L | 0.30 | 0.40 | 0.50 |

Preliminary

# 22 规格更改记录

| 版本 | 记录 | 日期 |
|---|---|---|
| V0.4 | 修改 IAP/CRC 章节描述 | 2024 年 1 月 12 日 |
| V0.3 | 1. 完善 STOP/IDLE/IAP/CRC 例程操作语句描述：原至少 8 个 NOP 指令，现修正为至少 12 个 NOP 指令<br>2. 修正管脚定义列表笔误 | 2023 年 4 月 10 日 |
| V0.2 | 1. 完善 BTM 低频定时器控制寄存器描述：原中断时间档位最高为 4S，现更正为 32S<br>2. 完善 ADCCFG1 设置寄存器描述：原 ADC 通道为 10 个，现更正为 12 个 | 2023 年 1 月 30 日 |
| V0.1 | 初版 | 2021 年 12 月 10 日 |

Preliminary

# 声明

深圳市赛元微电子股份有限公司（以下简称赛元）保留随时对赛元产品、文档或服务进行变更、更正、增强、修改和改进的权利，恕不另行通知。赛元认为提供的信息是准确可信的。本文档信息于 2021 年 12 月开始使用。在实际进行生产设计时，请参阅各产品最新的数据手册等相关资料。

Preliminary