

## SST89E564RD/SST89V564RD/SST89E554RC/SST89V554RC

### 器件资料

#### 一、性能特点:

- I 通用 8 位 8051 系列兼容微处理器，带嵌入式存储器
  - 与 8051 的软件完全兼容
  - 开发工具兼容
  - 封装与引脚兼容
- I 工作电压：
  - SST89E564RD/SST89E554RC: 5V 操作，工作频率 0~40MHz
  - SST89V564RD/SST89V554RC: 3V 操作，工作频率 0-33MHz
- I 内部 RAM 共有 1Kbyte (256Byte + 768Byte)
- I 256Bytes 寄存器/数据 RAM
- I 内含两块高性能 SuperFlash 存储器 (EEPROM)
  - SST89E564RD/SST89V564RD:  
64Kbyte 的第一块和 8Kbyte 的第二块 (每块的扇区大小都是 128Byte)
  - SST89E554RC/SST89V554RC:  
32Kbyte 的第一块和 8Kbyte 的第二块 (每块的扇区大小都是 128Byte)
  - 用 SoftLock 可以独立地对每个块进行安全加锁。
  - 实现应用中再编程 (IAP: In-Application-Programming), 两块可同时操作。
  - 在 IAP 过程中可实现存储器的覆盖，支持中断的响应
- I 支持最大 64KByte 外部程序和数据存储空间
- I P1 的 5, 6, 7 引脚可驱动大电流 (每个可达 16mA)
- I 3 个 16 位定时/计数器 (T0, T1, T2)
- I 一个全双工、可编程串行通讯口 (UART)
  - 帧错误识别
  - 自动地址识别
- I 8 个中断源，4 个优先级
- I 带内部可编程看门狗 (WDT)
- I 4 个 8 位 I/O 口 (32 个 I/O 引脚)
- I 支持第二个 DPTR 寄存器
- I 标准的每个指令周期 12 个时钟，也可以倍频，以实现每个指令周期 6 个时钟
- I TTL 和 CMOS 电平全兼容
- I 支持掉电检测
- I 省电模式
  - Idle 模式
  - Power Down 模式，并由外部中断唤醒
- I 有三种封装: PDIP-40、PLCC-44 和 TQFP-44
- I 有工业级 (-40°C—+85°C) 和商用级 (0°C—+70°C)

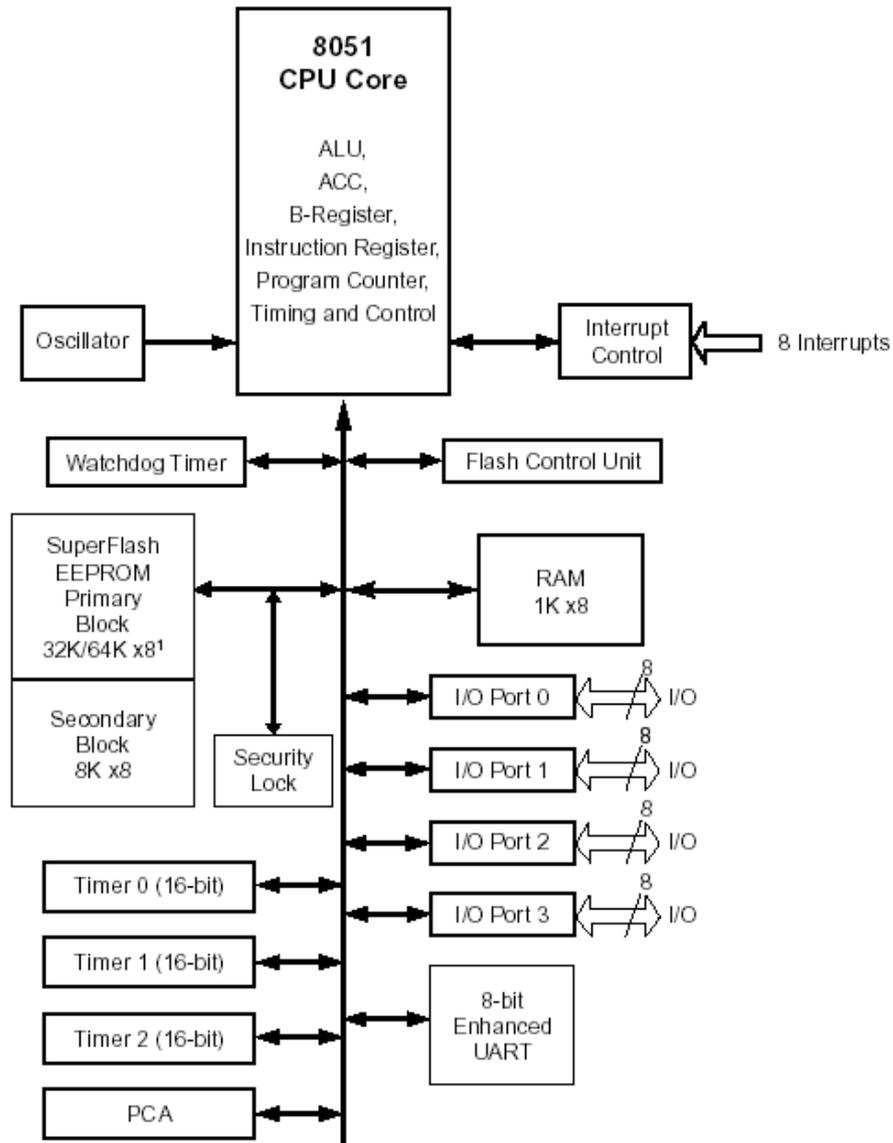
## 二、产品描述:

SST89E564RD、SST89V564RD、SST89E554RC 和 SST89V554RC 是 SST 公司 8 位微处理器 FlashFlex51 系列的成员，是采用先进的 SuperFlash CMOS 半导体技术设计和制造，这些器件是采用 8051 的指令集，并和标准的 8051 控制器管脚兼容。

器件带有 72/40Kbyte 的片内 FLASH EEPROM 存储器，使用了 SST 公司专利的 CMOS SuperFlash EEPROM 技术，存储器被分成两块独立的程序存储器，第一块 (BLOCK0) 占用 64/32Kbyte 的内部程序存储器空间，第二块 (BLOCK1) 占用 8Kbyte 的内部程序存储器空间。8Kbyte 的第二块 FLASH 可以映射到 64/32Kbyte 空间的低地址，还可以被隐藏和当成类似 EEPROM 的独立的数据存储器。FLASH 存储器可用标准的 87C5x OTP EPROM 编程器来烧录。在上电复位时，单片机可以配置成外部主机的从属设备，以源代码存入，也可以做外部主机的控制机，执行 IAP 操作。单片机已经预先烧录一段引导下装 (BOOT STRAP LOADER) 的代码，通过 IAP 操作，实现开始的用户程序代码烧录和以后的用户代码升级。CHIP-ERASE 操作会擦除该引导下装程序。

除了片内 72/40Kbyte 的 SuperFlash EEPROM 的程序存储器，单片机也可寻址外部 64Kbyte 程序空间；除了 1024Byte 片内 RAM，单片机也可寻址外部 64Kbyte RAM 空间。SST 公司的高可靠性、专利的 SuperFlash 技术和存储器单元结构有很多的优点，这些优点给我们的客户提供明显的成本和可靠性改善。

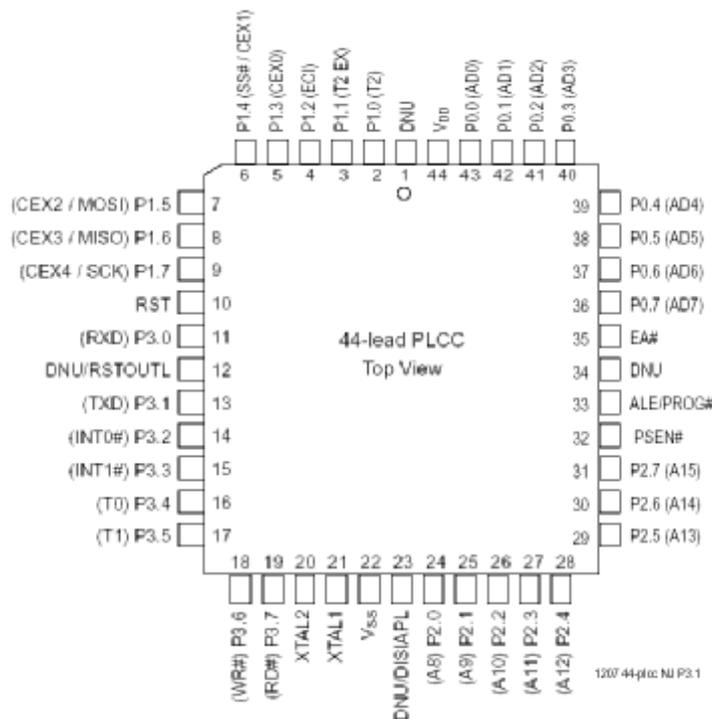
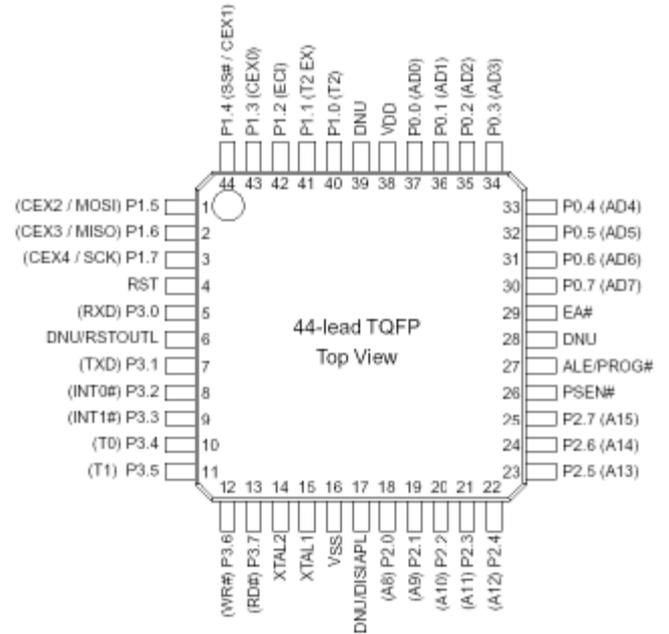
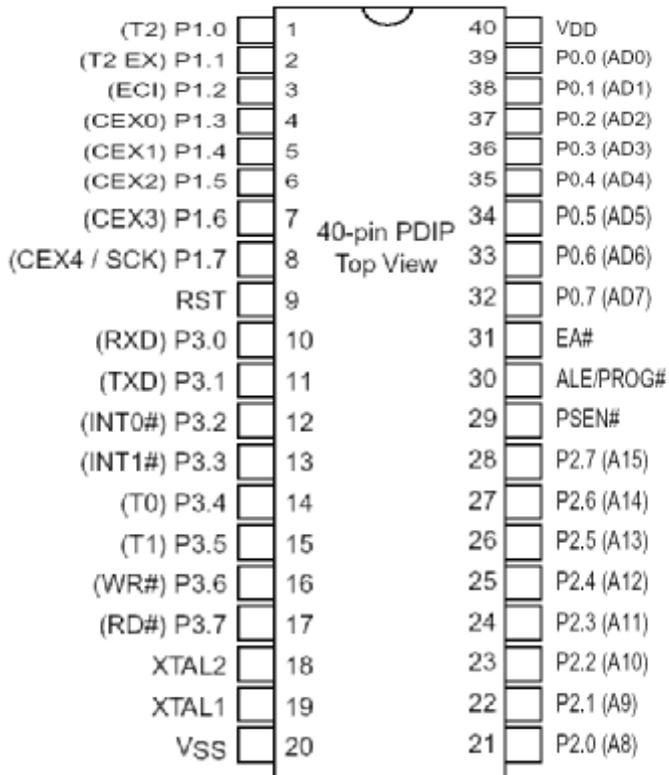
## 1.0 功能方框图



SST89E564RD/SST89V564RD: 64K x 8

SST89E554RD/SST89V554RD: 32K x 8

## 2.0 引脚定义图



1207 44-pin NI P3.1

## 2.1 引脚说明

符号	类型 <sup>1</sup>	名字和功能
P0[7:0]	I/O	Port 0: Port 0 是八位双向 I/O 端口, 当作输出口时每个引脚可以接收 LS TTL 电平输入, 可以写为 1 使其状态为悬浮用做高阻输入。P0 也可以在访问外部程序存储器时作地址的低字节, 在访问外部数据存储器时作数据总线, 此时通过内部强上拉输出 1。P0 在外部主机模式编程时接收代码, 外部主机模式校验是输出代码。在编程校验或是当作通用 I/O 口时需要外接上拉。
P1[7:0]	I/O, 带上拉	Port1: P1口是8位带内部上拉的双向I/O口, P1输出可以驱动LS TTL输入。向P1口写入1时P1口被内部上拉为高电平, 可用作输入口。当作为输入脚时被外部拉低的P1口会因为内部上拉而输出电流(见DC 电气特性)。P1[5, 6, 7]可以驱动16mA的大电流。P1口在外部主机模式编程和校验时接收低位地址数据。
P1[0]	I/O	T2: 定时/计数器 2 的外部计数输入/时钟输出。
P1[1]	I	T2EX: 定时/计数器 2 重载/捕捉/方向控制
P1[2]	I	ECI: 该信号是 PCA 的外部时钟输入(External Clock Input)。
P1[3]	I/O	CEX0: PCA 模块 0 的捕捉/比较外部 I/O。 每个捕捉/比较模块都要接到一个 P1 口做外部 I/O。
P1[4]	I/O	CEX1: PCA 模块 1 的捕捉/比较外部 I/O。
P1[5]	I/O	CEX2: PCA 模块 2 的捕捉/比较外部 I/O。
P1[6]	I/O	CEX3: PCA 模块 3 的捕捉/比较外部 I/O。
P1[7]	I/O	CEX4: PCA 模块 4 的捕捉/比较外部 I/O。
P2[7:0]	I/O, 带内部上拉	Port2: P2 口是 8 位带内部上拉的双向 I/O 口。向 P2 口写入 1 时 P2 口被内部上拉为高电平, 可用作输入口。当作为输入脚时, 被外部拉低的 P2 口会因为内部上拉而输出电流(见 DC 电气特性)。在访问外部程序存储器和外部数据时分别作为地址高位字节和 16 位地址(MOVX @DPTR), 此时通过内部强上拉传送 1。P2 口在外部主机模式编程和校验时接收一些控制信号和部分的低位地址数据。
P3[7:0]	I/O, 带内部上拉	Port3: P3口是8位带内部上拉的双向I/O口。P3口输出缓冲器可以驱动LS TTL输入。向P3口写入1时P3口被内部上拉为高电平, 可用作输入口。当作为输入脚时, 被外部拉低的P3口会因为内部上拉而输出电流(见DC 电气特性)。P3口在外部主机模式编程和校验时接收一些控制信号和部分的低位地址数据。
P3[0]	I	RXD: UART 串行输入口
P3[1]	O	TXD: UART 串行输出口
P3[2]	I	INT0#: 外部中断 0 输入
P3[3]	I	INT1#: 外部中断 1 输入
P3[4]	I	T0: 定时器 0 外部输入
P3[5]	I	T1: 定时器 1 外部输入
P3[6]	O	WR#: 外部数据存储器写信号
P3[7]	O	RD#: 外部数据存储器读信号
PSEN#	I/O	程序存储使能。是对外部程序的读选通脉冲, 当执行内部程序存储器时, PSEN#被激活(高)。当执行外部程序存储器代码时 PSEN#每个机器周期

		被激活两次，除了在每次访问外部数据存储器两次激活被忽略。当 RST 连续保持高电平或超过 10 个机器周期，PSEN# 从高变低的变化使单片机进入外部主机模式编程状态。
RST	I	<b>Reset:</b> 复位。当晶振在运行中，只要复位管脚出现 2 个机器周期高电平即可复位单片机。当 RST 输入高电平时，如果 PSEN# 从高变成低，单片机将外部主机模式，否则进入普通运行模式。
EA#	I	<b>External Access Enable:</b> 外部寻址使能。在访问整个外部程序存储器时 EA 必须外部置低，如果 EA 为高时将执行内部程序。但是安全加密第 4 级禁止 EA#，程序只能执行内部程序空间。EA# 脚能承受 12V 高电压 <sup>2</sup> （请参考电气参数）。
ALE/PROG#	I/O	<b>Address Latch Enable:</b> 地址锁存使能。在访问外部存储器时输出脉冲锁存地址的低字节。该引脚也是 FLASH 编程的脉冲输入（PROG#）。在正常情况下 ALE <sup>3</sup> 输出信号恒定为 1/6 振荡频率，并可用作外部时钟或定时。注意每次访问外部数据时，一个 ALE 脉冲将被忽略。ALE 可以通过置位 SFR 的 Auxiliary.0 禁止，置位后 ALE 只能在执行 MOVX 指令时被激活。
DNU	I/O	<b>Do not use:</b> 不要使用，必须保持悬浮。在图 2-2 和图 2-3 中 RTSOUTL 和 DISIAPL 脚在正常操作时在使用，但在外部主机模式必须悬浮。
RSTOUTL	O, 带内部上拉	在看门狗复位和掉电复位时变低。
DISIAPL	I	如果 DISIAPL 变低可以禁止 IAP 功能。在复位时取样和锁存，这是 IAP 的硬件禁止。复位后，管脚的任何变化都无效。
XTAL1	I	晶体 1: 反相振荡放大器输入和内部时钟发生电路输入
XTAL2	O	晶体 2: 反相振荡放大器输出
V <sub>DD</sub>	I	电源
V <sub>SS</sub>	I	地

注 1: I=Input - 输入; O=Output - 输出

2: 在 FLASH 编程时不需要接 12V 的编程电压。

### 3.0 存储器概述

#### 3.1 程序 FLASH 存储器

单片机内部有两块 FLASH 存储器，第一块(BLOCK0)是 64Kbyte，第二块(BLOCK1)是 8Kbyte。因为全部程序地址空间限制为 64Kbyte，SFCF[1:0]用来控制程序区的选择。请参考图 3-1 和图 3-2 的程序空间的配置。64K/32K x8 的第一块由 512/256 个扇区组成，每个扇区有 128Byte。8K x8 的第二块由 64 个扇区组成，每个扇区有 128Byte。

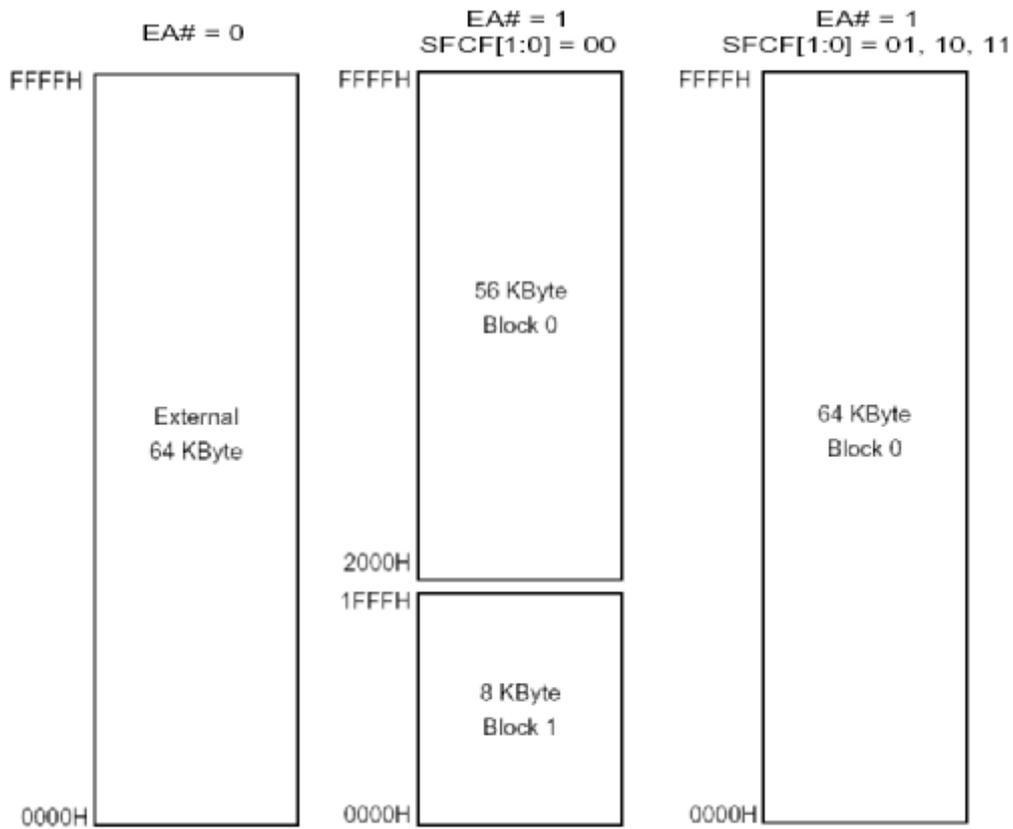


图 3-1 SST89E564RD/SST89V564RD 程序空间配置

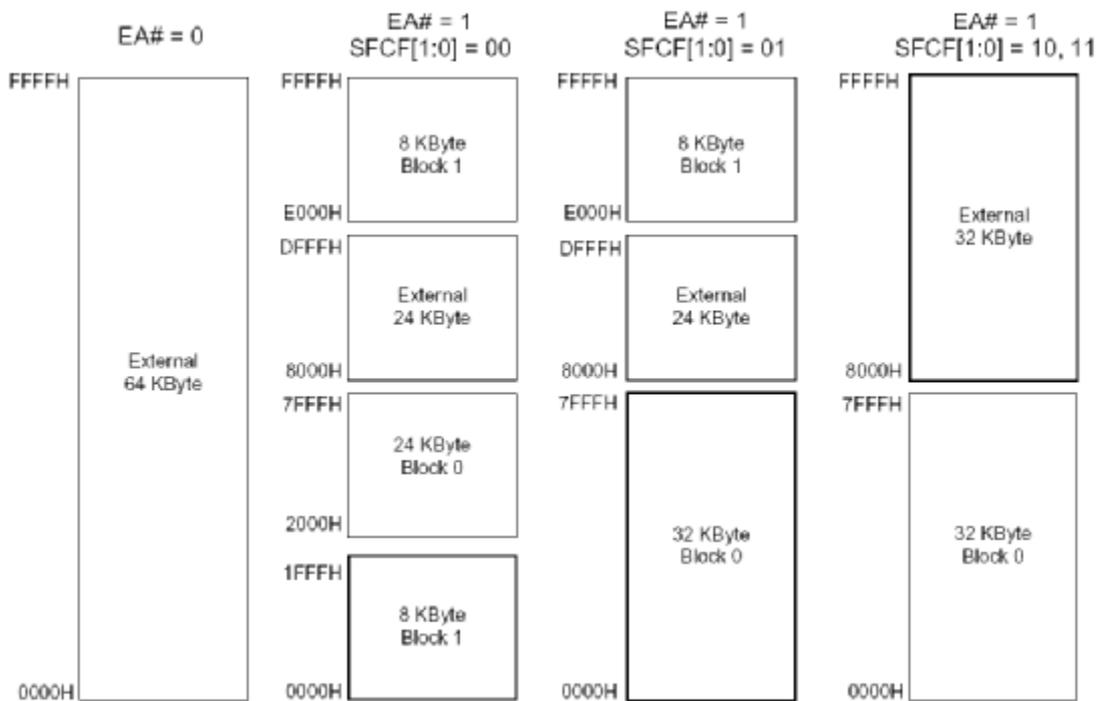


图 3-2 SST89E554RD/SST89V554RD 程序空间配置

### 3.2 程序存储器块的切换

单片机的程序存储器块切换功能使 BLOCK1 或 BLOCK0 的低 8Kbyte 占用程序地址空间的最低 8Kbyte 地址。SFCF[1:0]控制程序存储器的块切换。

表 3-1: SFCF 的值及对应的程序存储器切换 (SST89E/V564RD)

SFCF[1:0]	程序存储器块切换
01, 10, 11	BLOCK1 被隐藏, 程序计数器看不到。 BLOCK1 仅通过对 0000H-1FFFH 的 IAP 操作来访问。
00	BLOCK1 覆盖了程序地址空间的低 8K, 占用地址 0000H-1FFFH。当 PC 指针落到 0000H-1FFFH 时, 将会从 BLOCK1 取指令, 而不是从 BLOCK0。在 0000H-1FFFH 以外的地址则是用 BLOCK0。这时 BLOCK0 的 0000H-1FFFH 地址可以通过 IAP 操作来访问。

表 3-2: SFCF 的值及对应的程序存储器切换 (SST89E/V554RC)

SFCF[1:0]	程序存储器块切换
10, 11	BLOCK1 被隐藏, 程序计数器看不到。 BLOCK1 仅通过对 E000H-FFFFH 的 IAP 操作来访问。
01	BLOCK0 和 BLOCK1 都能被程序计数看见。 BLOCK0 占用 0000H-7FFFH, BLOCK1 占用 E000H-FFFFH。
00	BLOCK1 覆盖了程序地址空间的低 8K, 占用地址 0000H-1FFFH。当 PC 指针落到 0000H-1FFFH 时, 将会从 BLOCK1 取指令, 而不是从 BLOCK0。在 0000H-1FFFH 以外的地址则是用 BLOCK0。这时 BLOCK0 的 0000H-1FFFH 地址可以通过 IAP 操作来访问。

#### 3.2.1 程序存储器块切换的复位初始值

程序存储器块切换的复位初始值是由起动配置位 SC0 及 SC1 的状态确定。SC0 和 SC1 位可通过外部主机模式命令或 IAP 模式的命令来编程。请看表 4-2 和表 4-7。RESET 完成后, SFCF[0]可以由程序动态地修改, 改变 SFCF[0]值不会改变 SC0 位。

在改变 SFCF[0]的值时需要小心, 因为会使不同的物理空间映射到逻辑程序地址空间, 使用者必须防止在地址范围 0000H 到 1FFFH 执行块切换指令。

表 3-3 在不同 RESET 条件下的 SFCF 值 (SST89E/V554RC)

SC1	SC0	SFCF[1:0]的状态		
		上电或外部复位	WDT 看门狗复位 或掉电复位	软件复位
U (1)	U (1)	00	x0	10
U (1)	P (0)	01	x1	11
P (0)	U (1)	10	10	10
P (0)	P (0)	11	11	11

注: P=Programmed – 编程 (位逻辑状态=0)  
U=Unprogrammed – 无编程 (位逻辑状态=1)

表 3-4 在不同 RESET 条件下的 SFCF 值 (SST89E/V564RD)

SC0	SFCF[1:0]的状态		
	上电或外部复位	WDT 看门狗复位 或掉电复位	软件复位
U (1)	00	x0	10
P (0)	01	x1	11

注: P=Programmed – 编程 (位逻辑状态=0)  
U=Unprogrammed – 无编程 (位逻辑状态=1)

### 3.3 数据 RAM 存储器

数据 RAM 有 1024 字节的内部存储器, RAM 能寻址到 64Kbyte 的外部数据存储器。

### 3.4 扩展数据 RAM 的寻址

SST89E/V554RC 和 SST89E/V564RD 都有 1K 的 RAM, 对内部数据存储器有四个部分:

- 1、RAM 的低 128Byte (00H 到 7FH) 可以直接或间接寻址。
- 2、RAM 的高 128Byte (80H 到 FFH) 可以间接寻址。
- 3、特殊功能寄存器 (80H 到 FFH) 只能直接寻址。
- 4、768Byte 的扩展 RAM (00H 到 2FFH) 由转移外部指令 MOVX 间接寻址和清除 EXTRAM 位 (请参考下面的 Auxiliary Register(AUXR))。

由于高位 128Byte 字节和 SFR 占用相同的地址, RAM 必须间接访问, RAM 和 SFR 空间尽管有相同的地址, 但是它们在物理上是分开的。当指令访问高位 128Byte (高于 7FH), 单片机访问 SFR 还是 RAM 由指令的类型, 如果是间接, 将访问 RAM; 如果是直接, 将访问 SFR。请看下面例子:

直接寻址:

MOV @R0, #data ; R0 的值是 90H

寄存器 R0 指向 90H, 90H 是在高位地址的范围, 在 #data 的数据将写到 90H 的 RAM 的位置, 而不是 P1 口。

直接寻址:

MOV 90H, #data ;写数据到 P1 口

在 #data 的数据写到 P1 口, 指令直接写到地址将写到 SFR。

要访问扩展 RAM, 必须清除 EXTRAM 位, 并且用 MOVX 指令, 额外的 768 字节是在芯片内部, 占用外部存储器的开始 768 字节 (地址 000H 到 2FFH)。

当 EXTRAM=0, 扩展 RAM 通过 MOVX 指令和 R0、R1 的组合使用实现间接寻址。访问扩展 RAM 不会影响 P0、P3.6 (WR#)、P3.7 (RD#) 或 P2。EXTRAM=0 时访问扩展 RAM 的例子如下:

**扩展 RAM 访问 (只能间接寻址):** MOVX @DPTR, A ; DPTR 的内容是 0A0H

DPTR 指向 0A0H, A 的数据写到扩展 RAM 的 0A0H 地址而不是外部存储器, 用 MOVX 指令访问高于 2FFH 的外部存储器将读取外部存储器 (0300H 到 FFFFH), 和标准 8051 相同, P0 和 P2 做数据/地址总线, P3.6 和 P3.7 做写/读的时序信号。

如果 EXTRAM=1, MOVX @Ri 和 MOVX @DPTR 执行和标准 8051 相同, 用 MOVX @Ri 在 P0 口提供了带多个数据的 8 位地址, 其它的输出口可以用来输出更高的地址位, 这样提供了外部寻址的功能, 用 MOVX @DPTR 指令产生 16 位的地址, 使外部地址寻址达到 64K。P2 口是高 8 位地址 (DPH), P0 口是低 8 位地址和数据共用。MOVX @Ri 和 MOVX @DPTR 产生外部存储器必要的读/写信号 (P3.6 - WR# 和 P3.7 - RD#)。表 3-5 列出了外部数据存储器 RD#、WR# 的变化和 EXTRAM 的关系。堆栈指针可位于内部 256 字节 RAM 中的任意位置。堆栈不能位于扩展 RAM 中。

**表 3-5: 不同 EXTRAM 的外部数据存储器 RD#, WR#**

AUXR	MOVX @DPTR, A or MOVX A, @DPTR		MOVX @Ri, A or MOVX A, @Ri
	ADDR < 0300H	ADDR >= 0300H	ADDR = Any
EXTRAM=0	RD# / WR# not asserted	RD# / WR# asserted	RD# / WR# not asserted <sup>注</sup>
EXTRAM=1	RD# / WR# asserted	RD# / WR# asserted	RD# / WR# asserted

注: 访问扩展 RAM 的地址限制在 0 到 0FFH, 不能访问 100H 到 02FFH。

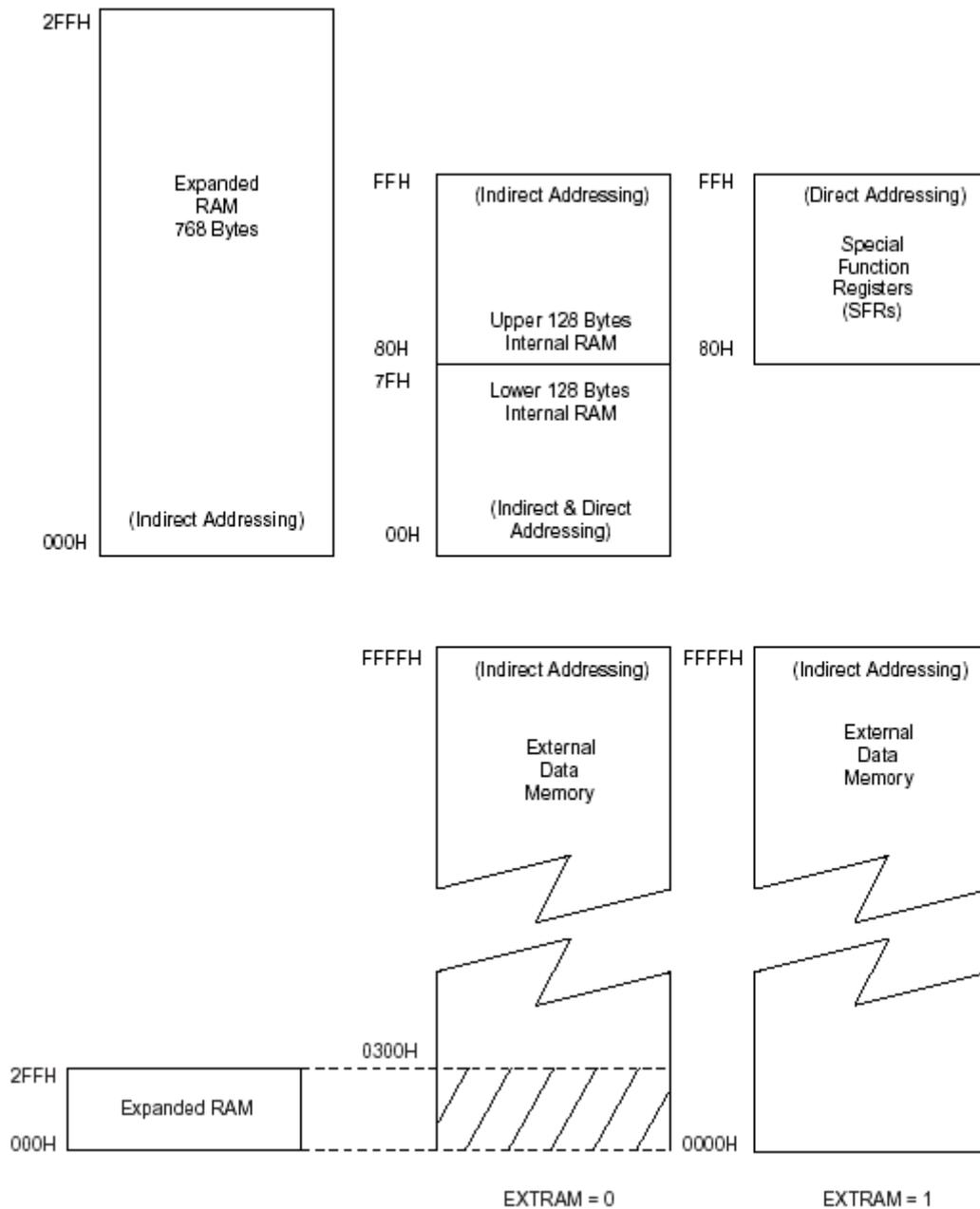


图 3-3 内部和外部数据存储器结构

### 3.5 双数据指针

本器件有两个16位数据指针，在AUXR1的DPS位决定选择那一个数据指针，当DPS=0，选择DPTR0；当DPS=1，选择DPTR1。通过执行在AUXR1的INC指令能对两个数据指针快速切换（请看图3-4）。

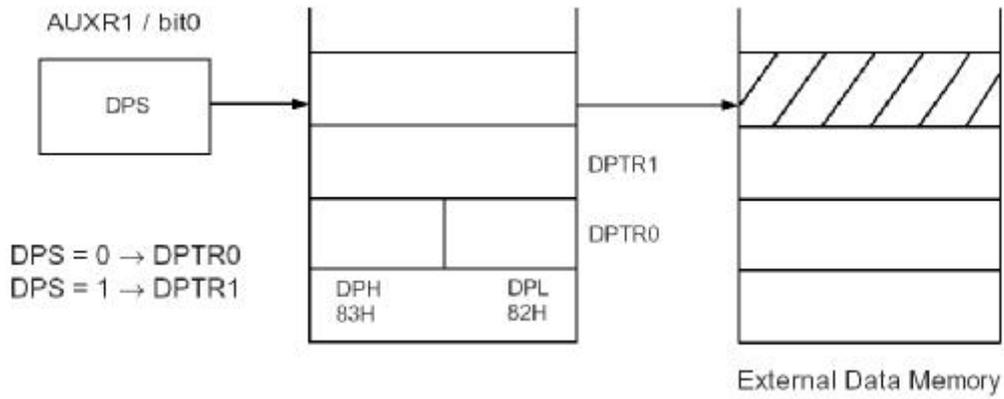


图 3-4 双数据指针组织图

### 3.6 特殊功能寄存器

FlashFlex51 单片机系列的大部分特性是由在 SFR 存储器中的特殊功能寄存器控制，请看表 3-6 的 SFR 图，而表 3-7 到表 3-11 列出每一个 SFR 的说明和 RESET 值。

表 3-6 FlashFlex51 特殊功能寄存器

8 个字节									
F8H	IPA <sup>1</sup>	CH	CCAP0H	CCAP1H	CCAP2H	CCAP3H	CCAP4H		FFH
F0H	B <sup>1</sup>							IPAH	F7H
E8H	IEA <sup>1</sup>	CL	CCAP0L	CCAP1L	CCAP2L	CCAP3L	CCAP4L		EFH
E0H	ACC <sup>1</sup>								E7H
D8H	CCON <sup>1</sup>	CMOD	CCAPM0	CCAPM1	CCAPM2	CCAPM3	CCAPM4		DFH
D0H	PSW <sup>1</sup>								D7H
C8H	T2CON <sup>1</sup>	T2MOD	RCAP2L	RCAP2H	TL2	TH2			CFH
C0H	WDTC <sup>1</sup>								C7H
B8H	IP <sup>1</sup>	SADEN							BFH
B0H	P3 <sup>1</sup>	SFCF	SFCM	SFAL	SFAH	SFDT	SFST	IPH	B7H
A8H	IE <sup>1</sup>	SADDR							AFH
A0H	P2 <sup>1</sup>		AUXR1						A7H
98H	SCON <sup>1</sup>	SBUF							9FH
90H	P1 <sup>1</sup>								97H
88H	TCON <sup>1</sup>	TMOD	TL0	TL1	TH0	TH1	AUXR		8FH
80H	P0 <sup>1</sup>	SP	DPL	DPH		WDTD		PCON	87H

注 1: 可位寻址的 SFR

表 3-7: 与 CPU 相关的 SFR

符号	说明	地址	位功能和位地址								复位值
ACC <sup>1</sup>	累加器	E0H	ACC[7:0]								00H
B <sup>1</sup>	B 寄存器	F0H	B[7:0]								00H
PSW <sup>1</sup>	程序状态字	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H

SP	堆栈指针	81H	SP[7:0]								07H
DPL	指针低地址	82H	DPL[7:0]								00H
DPH	指针高地址	83H	DPH[7:0]								00H
IE <sup>1</sup>	中断使能	A8H	EA	EC	ET2	ES	ET1	EX1	ET0	EX0	00H
IEA <sup>1</sup>	中断使能 A	E8H	-	-	-	-	EBO	-	-	-	xxxx0xxx <sup>b</sup>
IP <sup>1</sup>	中断优先级	B8H	-	PPC	PT2	PS	PT1	PX1	PT0	PX0	x0000000 <sup>b</sup>
IPH	中断优先级高字节	B7H	-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	x0000000 <sup>b</sup>
IPA <sup>1</sup>	中断优先寄存器 A	F8H	-	-	-	-	PBO	-	-	-	xxxx0xxx <sup>b</sup>
IPAH	中断优先寄存器 A 高字节	F7H	-	-	-	-	PBOH	-	-	-	xxxx0xxx <sup>b</sup>
PCON	电源控制寄存器	87H	SMOD1	SMOD0	BOF	POF	GF1	GF0	PD	IDL	00010000 <sup>b</sup>
AUXR	辅助功能寄存器	8EH	-	-	-	-	-	-	EXTRAM	AO	xxxxxx00 <sup>b</sup>
AUXR1	辅助功能寄存器 1	A2H	-	-	-	-	GF2	0	-	DPS	xxxx00x0 <sup>b</sup>

注 1: 可位寻址的 SFR

表 3-8: Flash 存储器编程 SFR

符号	说明	地址	位功能和位地址								复位值
SFCF	SuperFlash 配置	B1H	-	IAPEN	-	-	-	-	SWR	BSEL	x0xxxx00b
SFCM	SuperFlash 命令	B2H	FIE	FCM[6:0]							00H
SFAL	SuperFlash 低位地址	B3H	SuperFlash 低位地址寄存器 A7-A0 (SFAL)								00H
SFAH	SuperFlash 高位地址	B4H	SuperFlash 高位地址寄存器 A15-A8 (SFAH)								00H
SFDT	SuperFlash 数据	B5H	SuperFlash 数据寄存器								00H
SFST	SuperFlash 状态	B6H	SB1_i	SB3_i	SB3_i	-	EDC_i	FLASH_BUSY	-	-	000x00xxb

表 3-9: 看门狗 SFR

符号	说明	地址	位功能和位地址								复位值
WDTC <sup>1</sup>	看门狗定时器控制	C0H	-	-	-	WDOUT	WDRE	WDTS	WDT	SWDT	xxx00x00b
WDTD	看门狗定时器数据/加载	85H	看门狗定时器数据/加载								00H

注 1: 可位寻址的 SFR

表 3-10: 定时器/计数器 SFR

符号	说明	地址	位功能和位地址								复位值
TMOD	定时器模式	89H	Timer1				Timer0				00H
			GATE	C/T#	M1	M0	GATE	C/T#	M1	M0	
TCON <sup>1</sup>	定时器控制	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TH0	定时器 0 高字节	8CH	TH0[7:0]								00H
TL0	定时器 0 低字节	8AH	TL0[7:0]								00H

	节										
TH1	定时器1高字节	8DH	TH1[7:0]								00H
TL1	定时器1低字节	8BH	TL1[7:0]								00H
T2CON <sup>1</sup>	定时器2控制	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#	00H
T2MOD#	定时器2模式	C9H	-	-	-	-	-	-	T2OE	DCEN	xxxxxx00b
TH2	定时器2高字节	CDH	TH2[7:0]								00H
TL2	定时器2底字节	CCH	TL2[7:0]								00H
RCAP2H	定时器2捕获高字节	CBH	RCAP2H[7:0]								00H
RCAP2L	定时器2捕获低字节	CAH	RCAP2L[7:0]								00H

注 1: 可位寻址的 SFR

表 3-11: 接口 SFR

符号	说明	地址	位功能和位地址								复位值
SBUF	串口数据缓冲区	99H	SBUF[7:0]								不确定
SCON <sup>1</sup>	串行口控制	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H

SADDR	从地址	A9H	SADDR[7:0]								00H
SADEN	从地址屏蔽	B9H	SADEN[7:0]								00H
P0 <sup>1</sup>	P0 口	80H	P0[7:0]								FFH
P1 <sup>1</sup>	P1 口	90H	-	-	-	-	-	-	T2EX	T2	FFH
P2 <sup>1</sup>	P2 口	A0H	P2[7:0]								FFH
P3 <sup>1</sup>	P3 口	B0H	RD#	WR#	T1	T0	INT1#	INT0#	TXD	RXD	FFH

注 1: 可位寻址的 SFR

表 3-12: PCA SFR

符号	说明	地址	位功能和位地址								复位值
CH	PCA 计数器高字节	F9H	CH[7:0]								00H
CL	PCA 计数器低字节	E9H	CL[7:0]								00H
CCON <sup>1</sup>	PCA 计数器控制	D8H	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0	00x00000b
CMOD	PCA 计数器模式	D9H	CIDL	WDTE	-	-	-	CPS1	CPS0	ECF	00xxx000b
CCAP0H	模块 0 捕获高字节	FAH	CCAP0H[7:0]								00H
CCAP0L	模块 0 捕获低字节	EAH	CCAP0L[7:0]								00H
CCAP1H	模块 1 捕获高字节	FBH	CCAP1H[7:0]								00H
CCAP1L	模块 1 捕获低字节	EBH	CCAP1L[7:0]								00H
CCAP2H	模块 2 捕获高字节	FCH	CCAP2H[7:0]								00H
CCAP2L	模块 2 捕获低字节	ECH	CCAP2L[7:0]								00H
CCAP3H	模块 3	FDH	CCAP3H[7:0]								00H

	捕获高字节											
CCAP3L	模块 3 捕获低字节	EDH	CCAP3L[7:0]									00H
CCAP4H	模块 4 捕获高字节	FEH	CCAP4H[7:0]									00H
CCAP4L	模块 4 捕获低字节	EEH	CCAP4L[7:0]									00H
CCAPM0	模块 0 模式	DAH	-	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	x0000000b	
CCAPM1	模块 1 模式	DBH	-	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1	x0000000b	
CCAPM2	模块 2 模式	DCH	-	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2	x0000000b	
CCAPM3	模块 3 模式	DDH	-	ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3	x0000000b	
CCAPM4	模块 4 模式	DEH	-	ECOM4	CAPP4	CAPN4	MAT4	TOG4	PWM4	ECCF4	x0000000b	

注 1: 可位寻址的 SFR

### SuperFlash 配置寄存器 SFCF

位置	7	6	5	4	3	2	1	0	复位值
B1H	-	IAPEN	-	-	-	-	SWR	BSEL	x0xxx00b

标志 功能  
 IAPEN 允许 IAP 操作  
 0: 不允许执行 IAP 命令  
 1: 允许执行 IAP 命令  
 SWR 软件复位, 请参考 10.2 部分的“软件复位”  
 BSEL 程序空间块切换位, 请看图 3-1, 图 3-2, 表 3-3 和表 3-4

### SuperFlash 命令寄存器 (SFCM)

位置	7	6	5	4	3	2	1	0	复位值
B2H	FIE	FCM6	FCM5	FCM4	FCM3	FCM2	FCM1	FCM0	00H

标志 功能  
 FIE 允许 Flash 中断  
 0: INT1#没有被重新指定  
 1: INT1#被重新指定为标示 IAP 操作完成。  
 外部 INT1#中断被忽略。  
 FCM[60] Flash 操作命令  
 000\_0001b Chip-Erase

000\_1011b Sector-Erase  
 000\_1101b Block-Erase  
 000\_1100b Byte-Verify<sup>1</sup>  
 000\_1110b Byte-Program  
 000\_1111b Prog-SB1  
 000\_0011b Prog-SB2  
 000\_0101b Prog-SB3  
 000\_1001b Prog-SC0  
 000\_1001b Prog-SC1  
 000\_1000b Enable-Clock-Double  
 其它组合不会执行。

注1: Byte-Verify有一个机器周期等待时间, 所以不管FIE是何值都不会产生INT1#中断。

**SuperFlash 地址寄存器 (SFAL)**

位置	7	6	5	4	3	2	1	0	复位值
B3H	SuperFlash 低位地址寄存器								00H
标志	功能								
SFAL	连接 Flash 存储器块的邮箱寄存器, 低位地址寄存器								

**SuperFlash 地址寄存器 (SFAH)**

位置	7	6	5	4	3	2	1	0	复位值
B4H	SuperFlash 高位地址寄存器								00H
标志	功能								
SFAH	连接 Flash 存储器块的邮箱寄存器, 高位地址寄存器								

**SuperFlash 数据寄存器 (SFDT)**

位置	7	6	5	4	3	2	1	0	复位值
B5H	SuperFlash 数据寄存器								00H
标志	功能								
SFDT	连接 Flash 存储器块的邮箱寄存器, 数据寄存器								

**SuperFlash 状态寄存器 (SFST) - 只读寄存器**

位置	7	6	5	4	3	2	1	0	复位值
B6H	SB1_i	SB2_i	SB3_i	-	EDC_i	FLASH_BUSY	-	-	xxxxx0xxb
标志	功能								
SB1_i	保密位 1 状态 (SB1 位取反)								
SB2_i	保密位 2 状态 (SB2 位取反)								
SB3_i	保密位 3 状态 (SB3 位取反)								
	请参考表 9-1 的加密选项								
EDC_I	时钟倍频状态								
	0: 12 时钟一个机器周期								
	1: 6 时钟一个机器周期								
FLASH_BUSY	Flash 操作完成的查询位								

0: 已经完成了 IAP 操作

1: IAP 操作正在进行中

### Interrupt Enable 中断使能 (IE)

位置	7	6	5	4	3	2	1	0	复位值
A8H	EA	EC	ET2	ES	ET1	EX1	ET0	EX0	00H

标志	功能
EA	全局禁止位， 如果EA=0，禁止所有中断。 如果EA=1，通过置位或清除使能位，对应的每个中断被使能或禁止。
EC	PCA中断使能位
ET2	定时器2中断使能位
ES	串行口中断使能位
ET1	定时器1中断使能位
EX1	外部中断1使能位
ET0	定时器0中断使能位
EX0	外部中断0使能位

### Interrupt Enable A 中断使能 A (IEA)

位置	7	6	5	4	3	2	1	0	复位值
E8H	-	-	-	-	EBO	-	-	-	xxxx0xxx <sub>b</sub>

标志	功能
EBO	掉电中断使能 1: 允许中断 0: 禁止中断

### Interrupt Priority 中断优先级 (IP)

位置	7	6	5	4	3	2	1	0	复位值
B8H	-	PPC	PT2	PS	PT1	PX1	PT0	PX0	x0000000 <sub>b</sub>

标志	功能
PPC	PCA中断优先级控制位
PT2	定时器2中断优先级控制位
PS	串行口中断优先级控制位
PT1	定时器1中断优先级控制位
PX1	外部中断1中断优先级控制位
PT0	定时器0中断优先级控制位
PX0	外部中断0中断优先级控制位

### Interrupt Priority High 中断优先级高位 (IPH)

位置	7	6	5	4	3	2	1	0	复位值
B7H	-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	x0000000 <sub>b</sub>

标志	功能
PPCH	PCA中断优先级控制位高

PT2H	定时器2中断优先级控制位高
PSH	串行口中断优先级控制位高
PT1H	定时器1中断优先级控制位高
PX1H	外部中断1中断优先级控制位高
PT0H	定时器0中断优先级控制位高
PX0H	外部中断0中断优先级控制位高

#### Interrupt Priority A 中断优先级 A (IPA)

位置	7	6	5	4	3	2	1	0	复位值
F8H	-	-	-	-	PBO	-	-	-	xxxx0xxx <sub>b</sub>
标志	功能								
PBO	掉电中断优先位								

#### Interrupt Priority A High 中断优先级 A 高位 (IPAH)

位置	7	6	5	4	3	2	1	0	复位值
F7H	-	-	-	-	PBOH	-	-	-	xxxx0xxx <sub>b</sub>
标志	功能								
PBOH	掉电中断优先位高								

#### Auxiliary Register 辅助寄存器 (AUXR)

位置	7	6	5	4	3	2	1	0	复位值
8EH	-	-	-	-	-	-	EXTRAM	AO	xxxxxx00 <sub>b</sub>
标志	功能								
EXTRAM	内部/外部 RAM 存取								
	0: 使用MOVX @Ri/@DPTR访问内部RAM, 范围在00H到2FFH, 300H以上的地址访问外部RAM								
	1: 外部RAM								
AO	禁止/使能ALE								
	0: 以恒定的1/3振荡器频率发出ALE信号, 12时钟模式下为1/6fosc								
	1: ALE 仅当执行MOVX 或MOVC 指令时才有效								

#### Auxiliary Register1 辅助寄存器 1 (AUXR1)

位置	7	6	5	4	3	2	1	0	复位值
A2H	-	-	-	-	GF2	0	-	DPS	xxxx00x0 <sub>b</sub>
标志	功能								
GF2	通用的用户定义的标志位								
DPS	DPTR寄存器选择位, 用于切换指针DPTR0和DPTR1								
	0: 选择DPTR0								
	1: 选择DPTR1								

#### 看门狗定时器控制寄存器 (WDTC)

位置	7	6	5	4	3	2	1	0	复位值
C0H	-	-	-	WDOUT	WDRE	WDTS	WDT	SWDT	xxx00000 <sub>b</sub>
标志	功能								

- WDOUT** 看门狗输出允许  
 0: 看门狗复位不通过 **RESET** 脚输出  
 1: 如果设了 **WDRE**, 看门狗可以复位, 并从 **RESET** 脚输出 32 个时钟
- WDRE** 看门狗定时器复位允许位  
 0: 禁止看门狗定时器复位  
 1: 允许看门狗定时器复位
- WDTS** 看门狗定时器复位标志位  
 0: 外部硬件复位或上电复位可以清除这位, 向标志位写 1 有可清除这位, 如果芯片由于看门狗定时器溢出而复位, 标志位继续保持。  
 1: 看门狗溢出, 硬件设该位
- WDT** 看门狗定时器刷新  
 0: 当做了刷新, 硬件重置该位。  
 1: 软件设该位让看门狗定时器刷新。
- SWDT** 开始看门狗定时器  
 0: 停止 **WDT**。  
 1: 开始 **WDT**。

**看门狗定时器数据/重加载寄存器 (WDTD)**

位置	7	6	5	4	3	2	1	0	复位值
85H	看门狗定时器数据/装载								00H
标志	功能								
WDTD	看门狗定时器初始/装载值。设了 <b>WDT</b> 后新值才有效								

**PCA 定时器/计数器控制寄存器 (CCON) - 可位寻址**

位置	7	6	5	4	3	2	1	0	复位值
D8H	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0	00x00000b
标志	功能								
CF	PCA 计数器溢出标志, 计数器溢出时由硬件置位, <b>CMOD</b> 中 <b>ECF</b> 置位时 <b>CR</b> 为中断标志位。CF 可由硬件或软件置位, 但只能由软件清零。								
CR	PCA 计数器运行控制位, 由软件置位将 PCA 计数器打开, 要关闭 PCA 计数器必须由软件清零。								
-	保留位, 编程时勿将其置位。								
CCF4	PCA 模块 4 中断标志位。当产生匹配或捕获时由硬件置位。必须由软件清零。								
CCF3	PCA 模块 3 中断标志位。当产生匹配或捕获时由硬件置位。必须由软件清零。								
CCF2	PCA 模块 2 中断标志位。当产生匹配或捕获时由硬件置位。必须由软件清零。								
CCF1	PCA 模块 1 中断标志位。当产生匹配或捕获时由硬件置位。必须由软件清零。								
CCF0	PCA 模块 0 中断标志位。当产生匹配或捕获时由硬件置位。必须由软件清零。								

**PCA 定时器/计数器模式寄存器 (CMOD) - 可位寻址**

位置	7	6	5	4	3	2	1	0	复位值
D9H	CIDL	WDTE	-	-	-	CPS1	CPS0	ECF	00xxx000b
标志	功能								

- CIDL 计数器空闲控制:  
 CIDL=0时, PCA在空闲模式下继续工作。  
 CIDL=1时, PCA在空闲模式下关闭。
- WDTE 看门狗定时器使能:  
 WDTE=0时, 禁止PCA模块4的看门狗功能。  
 WDTE=1时, 使能。
- 保留位, 编程时勿将其置位
- CPS1 PCA计数脉宽选择位1
- CPS0 PCA计数脉宽选择位0

CPS1	CPS0	选择PCA输入*	
0	0	0	内部时钟6 时钟模式下为fosc/6(12 时钟模式下为fosc/12)
0	1	1	内部时钟6 时钟模式下为fosc/2(12 时钟模式下为fosc/4)
1	0	2	定时器0 溢出
1	1	3	ECI/P1.2 脚输入的外部时钟(6 时钟模式最大为fosc/4,12 时钟模式下为fosc/8)

\*fosc=振荡器频率

- ECF PCA使能计数器溢出中断:  
 ECF=1, 使能CCON 中的CF位, 产生中断。  
 ECF=0时禁止CF的功能。

#### PCA 比较/捕获模块模式寄存器 (CCAPMn)

位置	7	6	5	4	3	2	1	0	复位值
DAH	-	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	00xxx000b
DBH	-	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1	00xxx000b
DCH	-	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2	00xxx000b
DDH	-	ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3	00xxx000b
DEH	-	ECOM4	CAPP4	CAPN4	MAT4	TOG4	PWM4	ECCF4	00xxx000b

#### 标志 功能

- GF2 通用的用户定义的标志位  
 - 保留位, 编程时勿将其置位。
- ECOMn 比较器使能位。ECOMn=1 使能比较器功能。
- CAPPn 上升沿捕获。CAPPn=1使能上升沿捕获。
- CAPNn 下降沿捕获。CAPNn=1使能下降沿捕获。
- MATn 匹配: MATn=1时, PCA计数器值与该模块比较/捕获寄存器值的匹配使CCON 中的CCFn 位置位, 标志中断。
- TOGn 触发: TOGn=1时, PCA计数器值与该模块比较/捕获寄存器值的匹配使CEXn 脚的电平发生翻转
- PWMn 脉宽调制模式。PWMn=1使能CEXn 脚作为脉宽调制信号的输出。
- ECCFn 使能CCF中断。使能CCON中的比较/捕获标志CCFn以产生中断

**电源控制寄存器 (PCON)**

位置	7	6	5	4	3	2	1	0	复位值
87H	SMOD1	SMOD0	BOF	POF	GF1	GF0	PD	IDL	00010000b

标志            功能

SMOD1        双倍波特率位。当 SMOD1=1，定时器 1 用来产生波特率，串口用模式 1、2 和 3。

SMOD0        FE/SM0 选择位  
0: SM0[7]=SM0  
1: SM0[7]=FE

BOF            掉电保护状态位。不会被其它复位影响。BOF 可以由软件清除，上电复位也可以清 BOF 位。

0: 没有掉电保护  
1: 发生掉电保护

POF            上电保护状态位。不会被其它复位影响。 POF 可以由软件清除。  
0: 没有上电复位  
1: 发生上电复位。

GF1            通用标志位。

GF0            通用标志位

PD             掉电位。从掉电模式出来后硬件清除。  
0: 没有启动掉电模式  
1: 启动掉电模式

IDL            空闲 (IDLE) 模式位。从空闲模式出来后硬件清除。  
0: 没有启动空闲模式  
1: 启动空闲模式。

**串口控制寄存器 (SCON)**

位置	7	6	5	4	3	2	1	0	复位值
98H	SMO/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00000000b

标志            功能

FE             通过设 SMOD0=1 来访问 FE 位

0: 没有帧错误  
1: 帧错误。当检测到一个无效停止位时，通过 UART 接收器设置该位，但它必须由软件清零。

SM0            串口模式位0。要使该位有效，PCON寄存器中的SMOD0必须置0。

SM1            串口模式位1。

SM0	SM1	模式	说明	波特率
0	0	0	同步移位寄存器	fosc/12 (12时钟模式) 或 fosc/6 (6时钟模式)
0	1	1	8位UART	可变
1	0	2	9位UART	fosc/64 或 fosc/32(12时钟模式); fosc/32或fosc/16(6时钟模式)
1	1	3	9位UART	可变

Fosc=振荡频率

SM2	模式2或3的自动地址识别使能位。在模式2或3中，若SM2=1，且接收到的第9位数据（RB8）是0，则RI（接收中断标志）不会被激活。在模式1中，若SM2=1且没有接收到有效的停止位，则RI不会被激活。在模式0中，SM2必须是0。
REN	允许接收位。由软件置位或清除。 REN=1时允许接收。 REN=0时禁止接收。
TB8	模式2和3中发送的第9位数据，可以按需要由软件置位或清除。
RB8	模式2和3中已接收的第9位数据，在模式1中，或sm2=0，RB8是已接收的停止位，在模式0中，RB8未用。
TI	发送中断标志。模式0中，在发送完第8位数据时，由硬件置位。其它模式中，在发送停止位之初，由硬件置位。在任何模式中，都必须由软件来清除TI。
RI	接收中断标志。模式0中，接收第8位结束时由硬件置位。其它模式中，在接收停止位的中间时刻，由硬件置位。在任何模式(SM2所述情况除外)必须由软件清除RI。

### 定时器/计数器 2 控制寄存器（T2CON）

位置	7	6	5	4	3	2	1	0	复位值
C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#	00H

<b>标志</b>	<b>功能</b>
TF2	定时器 2 溢出标志。定时器 2 溢出时置位，必须由软件清除。当 RCLK 或 TCLK=1 时，TF2 将不会置位。
EXF2	定时器 2 外部标志。当 EXEN2=1 且 T2EX 的负跳变产生捕获或重装时，EXF2 置位。定时器 2 中断使能时，EXF2=1 将使 CPU 从中断向量处执行定时器 2 中断子程序。EXF2 位必须用软件清零。在递增/递减计数器模式（DCEN=1）中，EXF2 不会引起中断。
RCLK	接收时钟标志。RCLK 置位时，定时器 2 的溢出脉冲作为串行口模式 1 和模式 3 的接收时钟。RCLK=0 时将定时器 1 的溢出脉冲作为接收时钟。
TCLK	发送时钟标志。TCLK 置位时，定时器 2 的溢出脉冲作为串行口模式 1 和模式 3 的发送时钟。TCLK=0 时，将定时器 1 的溢出脉冲作为发送时钟。
EXEN2	定时器 2 外部使能标志。当其置位且定时器 2 未作为串行口时钟时，允许 T2EX 的负跳变产生捕获或重装。EXEN2=0 时，T2EX 的跳变对定时器 2 无效。
TR2	定时器 2 启动/停止控制位。置 1 时启动定时器。
C/T2#	定时器/计数器选择。（定时器 2） 0：内部定时器（OSC/12或OSC/6） 1：外部事件计数器（下降沿触发）
CP/RL2#	捕获/重装标志。置位：EXEN2=1 时 T2EX 的负跳变产生捕获。清零：EXEN2=1 时定时器 2 溢出或 T2EX 的负跳变都可使定时器自动重装。当 RCLK=1 或 TCLK=1 时，该位无效且定时器强制为溢出时自动重装。

### 定时器/计数器 2 模式控制寄存器（T2MOD）

位置	7	6	5	4	3	2	1	0	复位值
C9H	-	-	-	-	-	-	T2OE	DCEN	xxxxxx00b

<b>标志</b>	<b>功能</b>
-	不可用，保留将来之用。
T2OE	定时器 2 输出使能位

DCEN 向下计数使能位。定时器 2 可配置成向上/向下计数器。

## 4.0 FLASH 存储器的编程

SST89 系列单片机可以通过以下两种方法进行编程和烧录:

- 外部主机编程模式
- 应用中编程 (IAP) 模式

### 4.1 外部主机编程模式

外部主机编程模式让用户不用 CPU, 直接烧录 FLASH 存储器。进入外部主机模式是通过 RST 输入保持高电平时在 PSEN#脚加由高电平到低电平。当 RST=1 和 PSEN#=0 时, 器件始终在外部主机模式。

在外部主机模式下, 需要先做 Read-ID 操作与器件“握手”, 完成了 Read-ID 后, 别的外部主机模式命令才有效。内部 FLASH 存储块通过重新定义的 I/O 脚来访问。

详细的外部主机编程模式请参考英文的器件资料, 在 <http://www.szks.net> 上。

### 4.2 应用中在编程 IAP 模式

IAP 模式简单地说就是 CPU 一边在某一个 block 中运行程序, 同时可以对另一个 block 进行擦除、写入、校验等操作。单片机提供了 72KByte 或 40KByte 的应用中再编程 FLASH 存储器, 在 IAP 中, CPU 进入 IAP 模式。两块 FLASH 存储器让 CPU 在一个块中执行用户代码, 而同时另一块被擦除或编程。CPU 还可以在内部所有 FLASH 被再编程时从外部存储器获取代码。MAILBOX 寄存器 (SFST、SFCM、SFAL、SFAH、SFDT、SFCF) 位于特殊功能寄存器 SFR 中, 控制和监视器件的擦除和编程过程。

表 4-7 列出了命令和它们有联系 mailbox 寄存器的设定。

#### 4.2.1 IAP 模式时钟源

在 IAP 模式时, CPU 核与 FLASH 控制器单元与外部时钟脱离, 内部产生一个振荡器提供编程和擦除的时序参考, 内部振荡器在需要时打开, FLASH 操作完成后关闭。

#### 4.2.2 IAP 模式存储器块的选择

由于地址范围是 16bit, 所以程序只能在 64KB 空间内寻址, 表 4-5 中, 块的选择 (EA#脚和 SFCF[1:0] 位的配置) 允许块 1 存储器覆盖块 0 存储器的低 8KB, 使块 1 可访问。从“不可见”存储器的代码不能当作源地址去编程另外的地址, 但是“不可见的”存储器可以通过邮箱寄存器被另外一块编程。

器件允许一个存储器块的 IAP 代码改写另一个存储器块, 但不能改写在同一个块的地址, 也就是说, 从 BLOCK1 发出的 IAP 操作, 它的目标地址已经默认在 BLOCK0; 从 BLOCK0 发出的 IAP 操作, 它的目标地址已经默认在 BLOCK1; 如果是外部程序空间的 IAP 操作, 它的目标地址决定于地址和块选择的状态。

表 4-5: SST89E/V564RD IAP 地址分解

EA#	SFCF[1:0]	IAP 指令的地址	目标地址	被编程的存储块
1	00	>= 2000H (Block 0)	>= 2000H (Block 0)	None <sup>1</sup>
1	00	>= 2000H (Block 0)	< 2000H (Block 1)	Block 1

1	00	< 2000H (Block 1)	Any (Block 0)	Block 0
1	01, 10, 11	Any (Block 0)	>= 2000H (Block 0)	None <sup>1</sup>
1	01, 10, 11	Any (Block 0)	< 2000H (Block 1)	Block 1
0	00	From external	>= 2000H (Block 0)	Block 0
0	00	From external	< 2000H (Block 1)	Block 1
0	01, 10, 11	From external	Any (Block 0)	Block 0

#### 4.2.3 IAP使能位

IAP使能位是SFCF[6]，允许IAP模式。除非这位被设定，否则所有的FLASH编程的IAP指令都被忽略。

#### 4.2.4 IAP模式命令

以下所有的 IAP 命令只能在 IAP 模式下执行。在所有的情况下，写 SFCM 寄存器就会启动 IAP 动作。如果被选择的存储块被加锁，IAP 则不能对它操作。

PROGRAM（编程）指令是往存储器写入新的数据。准备编程的存储器应该在擦除状态，FFH。如果存储器没有被擦除，应该先用 ERASE（擦除）命令进行擦除。再次提醒，不要改写（编程或擦除）当前运行代码的 FLASH 块，这会造成不可预料的编程行为及可能破坏程序代码。

##### 4.2.4.1 芯片擦除

芯片擦除命令 Chip-Erase 擦除两个存储器块的所有数据。这个命令只能当 EA#=0 时（运行外部存储器）有效。如果器件在 LEVEL4 的加密等级时芯片擦除命令也是无效的。在其它的情况下，本指令将忽略加密状态并擦除加密位和映射位（RE-MAP）。

程序示例（查询方式）：

```

    ORL   SFCF, #40H           ;IAP 使能
    MOV   SFDT, #55H
    MOV   SFCM, #01H          ; 芯片擦除的命令是 01H
    LCALL DONE?               ; 查询 SFST[2]位
    
```

```

DONE?:  MOV   A, SFST
        JB   ACC.2, Done?     ; 等待到擦除完成
        RET
    
```

##### 4.2.4.2 块擦除 Block-Erase

块擦除命令 Block-Erase 将擦除某一个存储器块的所有数据。要选择擦除某个块，对 SST89x564RD，请参考表 4-5；对于 SST89x554RC，如果 SFAH[7]=0b，选择的是 BLOCK0 的主存储块，如果 SFAH[7:4]=EH，选择的是 BLOCL1 的次存储块。

程序示例（查询方式）：

```

    ORL   SFCF, #40H           ; IAP 使能
    MOV   SFAH, #00H          ; 擦除 BLOCK 0, 擦除 BLOCK1 则是#FOH
    MOV   SFDT, #55H
    MOV   SFCM, #0DH          ; 块擦除的命令是 0DH
    LCALL DONE?               ; 查询 SFST[2]位
    
```

```

DONE?:  MOV   A, SFST
        JB   ACC.2, Done?     ; 等待到擦除完成
        RET
    
```

##### 4.2.4.3 扇区擦除 Sector-Erase

扇区擦除命令 Sector-Erase 将擦除某一个扇区的所有数据。每个扇区的大小都是 128Byte。由在 SFAH 和 SFAL 的值决定要擦除的扇区。

程序示例（查询方式）：

```

    ORL  SFCE, #40H           ; IAP 使能
    MOV  SFAH, #sector_addressH ; 给地址高位
    MOV  SFAL, #sector_addressL ; 给地址低位
    MOV  SFCM, #0BH          ; 扇区擦除的命令是 0BH
    LCALL DONE?              ; 查询 SFST[2]位
    
```

```

DONE?:  MOV  A, SFST
        JB  ACC.2, Done?      ; 等待到擦除完成
        RET
    
```

#### 4.2.4.4 字节编程 Byte-Program

字节编程命令 Byte-Program 可以往一个字节（Byte）写数据。由在 SFAH 和 SFAL 的值决定要擦除的地址，数据放在 SFDT。

程序示例（查询方式）：

```

    ORL SFCE, #40H           ; IAP 使能
    MOV  SFAH, #byte_addressH ; 放入地址高位
    MOV  SFAL, #byte_addressL ; 放入地址低位
    MOV  SFDT, #data          ; 放入数据
    MOV  SFCM, #0EH          ; 扇区擦除的命令是 0EH
    LCALL DONE?              ; 查询 SFST[2]位
    
```

```

DONE?:  MOV  A, SFST
        JB  ACC.2, Done?      ; 等待到擦除完成
        RET
    
```

#### 4.2.4.5 字节校读 Byte-Verify

字节校读命令 Byte-Verify 让用户检查 ERASE 或 PROGRAM 后是否正确。如果命令执行成功，将在 SFDT 返回数据。所以用户在使用本命令前需要检查之前的 FLASH 操作是否已经完成。字节校读命令的执行时间很短，不需要检查命令的结束和不产生中断。

程序示例（查询方式）：

```

    ORL SFCE, #40H           ; IAP 使能
    MOV  SFAH, #byte_addressH ; 放入地址高位
    MOV  SFAL, #byte_addressL ; 放入地址低位
    MOV  SFCM, #0CH          ; 字节校读命令是 0CH
    MOV  A, SFDT              ; 数据保持在 SFDT
    
```

#### 4.2.4.6 Prog-SB3, Prog-SB2, Prog-SB1

Prog-SB3、Prog-SB2 和 Prog-SB1 命令是用来烧写安全加密位（表 9-1），命令完成后，加密位马上有效。当加密位在非编程状态才能被编程。Prog-SB3、Prog-SB2 和 Prog-SB1 命令只能在 BLOCK1 或外部的程序空间。

Pro-SB1 程序示例（查询方式）：

```
ORL SFCF, #40H          ; IAP 使能
MOV  SFDT, #0AAH
MOV  SFCM, #0FH         ; 烧录 SB1 的命令是 0FH
LCALL DONE?            ; 查询 SFST[2]位
```

```
DONE?: MOV  A, SFST
       JB  ACC.2, Done?   ; 等待到擦除完成
       RET
```

Pro-SB2 程序示例 (查询方式):

```
ORL SFCF, #40H          ; IAP 使能
MOV  SFDT, #0AAH
MOV  SFCM, #03H         ; 烧录 SB2 的命令是 03H
LCALL DONE?            ; 查询 SFST[2]位
```

```
DONE?: MOV  A, SFST
       JB  ACC.2, Done?   ; 等待到擦除完成
       RET
```

Pro-SB3 程序示例 (查询方式):

```
ORL SFCF, #40H          ; IAP 使能
MOV  SFDT, #0AAH
MOV  SFCM, #05H         ; 烧录 SB3 的命令是 05H
LCALL DONE?            ; 查询 SFST[2]位
```

```
DONE?: MOV  A, SFST
       JB  ACC.2, Done?   ; 等待到擦除完成
       RET
```

#### 4.2.4.7 Prog-SC0, Prog-SC1

Prog-SC0 命令是用来烧写 SC0 位, 改完 SC0 后 BSEL 位没有立即有效, 直到有复位。SC0 位在没有编程状态才可以由本命令烧录。Prog-SC0 命令只能在 BLOCK1 或外部的程序空间执行。

Prog-SC1 命令是用来烧写 SC1 位, 改完 SC1 后 SFCF[1]位没有立即有效, 直到有复位。SC1 位在没有编程状态才可以由本命令烧录。Prog-SC1 命令只能在 BLOCK1 或外部的程序空间执行。

Pro-SC0 程序示例 (查询方式):

```
ORL SFCF, #40H          ; IAP 使能
MOV  SFAH, #5AH
MOV  SFDT, #0AAH
MOV  SFCM, #09H         ; 烧录 SC 的命令是 09H
LCALL DONE?            ; 查询 SFST[2]位
```

```
DONE?: MOV  A, SFST
       JB  ACC.2, Done?   ; 等待到擦除完成
       RET
```

Pro-SC1 程序示例（查询方式）：

```

ORL SFCH, #40H          ; IAP 使能
MOV  SFAH, #0AAH
MOV  SFDT, #0AAH
MOV  SFCH, #09H        ; 烧录 SC 的命令是 09H
LCALL DONE?           ; 查询 SFST[2]位
    
```

```

DONE?:  MOV  A, SFST
        JB  ACC.2, Done?    ; 等待到擦除完成
        RET
    
```

#### 4.2.4.8 时钟倍频CLOCK-DOUBLE

时钟倍频命令是让 MCU 在 6 个时钟跑一个机器周期。标准的是 12 个时钟跑一个机器周期。

Pro-SC0 程序示例（查询方式）：

```

ORL SFCH, #40H          ; IAP 使能
MOV  SFAH, #55H
MOV  SFDT, #0AAH
MOV  SFCH, #08H        ; 烧录 SC 的命令是 09H
LCALL DONE?           ; 查询 SFST[2]位
    
```

```

DONE?:  MOV  A, SFST
        JB  ACC.2, Done?    ; 等待到擦除完成
        RET
    
```

#### 4.2.5 查询POLLING

FLASH操作的完成可以通过查询FLASH\_BUSY位（SFST[2]），当FLASH\_BUSY位为0时，器件完成FLASH的操作，准备下一个操作。MOVC指令也可以用来验证编程和擦除操作是否完成，如果MOVC指令读的FLASH块正忙，MOVC指令将出错。

#### 4.2.6 中断结束

把SFCH[7]设为1，将选择中断结束方式，这样会产生一个中断（INT1），提示FLASH操作完成。在这个情况下，INT1#脚变成内部中断源，所以在IAP时，INT1#不能当成外部中断源1，只能做为普通的I/O脚。为了用中断标示FLASH操作结束，IE寄存器的EX1和EA位需要设；TCON的IT1位也需要设，以做延触发检测。

表 4-6 SST89E/V564RD 的 IAP 指令<sup>1</sup>

操作	SFCH[6:0]	SFDT[7:0]	SFAH[7:0]	SFAL[7:0]
整片擦除 Chip-Erase <sup>3</sup>	01H	55H	X4	X
块擦除 Block-Erase <sup>5</sup>	0DH	55H	AH	X
扇区擦除 Sector-Erases	0BH	X	AH6	AL7
字节编程 Byte-Programs	0EH	D18	AH	AL
字节校读 Byte-Verify (Read) <sup>5</sup>	0CH	DO8	AH	AL
Prog-SB1 <sup>9</sup>	0FH	AAH	X	X
Prog-SB2 <sup>9</sup>	03H	AAH	X	X

Prog-SB3 <sup>9</sup>	05H	AAH	X	X
Prog-SC0 <sup>9</sup>	09H	AAH	5AH	X
时钟倍频 Clock-Doubles	08H	AAH	55H	X

- 1、SFCF[6]=1 允许 IAP 指令；SFCF[6]=0 禁止 IAP 指令。
- 2、FLASH 操作完成的中断/查询选择。  
SFCM[7]=1：中断方式；=0：查询方式
- 3、Chip-Erase 是 IAP 指令中唯一在 EA#=0（外部程序空间）工作。
- 4、X 可以是 V<sub>IL</sub> 或 V<sub>IH</sub>。
- 5、参考表 4-5。
- 6、AH 是地址高位
- 7、AL 是地址低位
- 8、DI=数据输入，DO=数据输出，其它值是 HEX 格式。
- 9、指令要在 BLOCK1 或外部地址空间。

表 4-7 SST89E/V554RC 的 IAP 指令<sup>1</sup>

操作	SFCM[6:0] <sup>2</sup>	SFDT[7:0]	SFAH[7:0]	SFAL[7:0]
整片擦除 Chip-Erase <sup>3</sup>	01H	55H	X <sup>4</sup>	X
块擦除 Block-Erase	0DH	55H	AH	X
扇区擦除 Sector-Erase	0BH	X	AH <sup>5</sup>	AL <sup>6</sup>
字节编程 Byte-Program	0EH	DI <sup>7</sup>	AH	AL
字节校读 Byte-Verify (Read) <sup>8</sup>	0CH	DO <sup>7</sup>	AH	AL
Prog-SB1 <sup>9</sup>	0FH	AAH	X	X
Prog-SB2 <sup>9</sup>	03H	AAH	X	X
Prog-SB3 <sup>9</sup>	05H	AAH	X	X
Prog-SC0 <sup>9</sup>	09H	AAH	5AH	X
Prog-SC1 <sup>9</sup>	09H	AAH	AAH	
时钟倍频 Clock-Double <sup>9</sup>	08H	AAH	55H	X

- 1、SFCF[6]=1 允许 IAP 指令；SFCF[6]=0 禁止 IAP 指令。
- 2、FLASH 操作完成的中断/查询选择。
- 3、Chip-Erase 是 IAP 指令中唯一在 EA#=0（外部程序空间）工作。
- 4、X 可以是 V<sub>IL</sub> 或 V<sub>IH</sub>。
- 5、AH 是地址高位
- 6、AL 是地址低位
- 7、DI=数据输入，DO=数据输出，其它值是 HEX 格式。
- 8、SFAH[7:5]=111b 选择块 1；SFAH[7]=0b 选择块 0。
- 9、指令要在 BLOCK1 或外部地址空间。

## 5.0 定时器/计数器

### 5.1 定时器

器件有 3 个 16 位寄存器做定时器或事件计数器。3 个定时器或计数器分别是 T0、T1 和 T2。每个都有一对 8 位特殊功能寄存器指定。各自的寄存器是 TL0, TH0, TL1, TH1, TL2, TH2。

### 5.2 定时器设置

参考表 3-10, 对应 T0、T1 和 T2 的 TMOD、TCON 和 T2CON 寄存器的设置。下面的表提供设置 T0、

T1、T2 的 TMOD 值。除了在波特率发生器模式，T2CON 不包括 TR2 位的设置，TR2 位需要单独设置来启动定时器。

**表 5-1: 定时器/计数器 0**

	模式	功能	TMOD	
			内部控制 (注 1)	外边控制 (注 2)
作为定时器	0	13 位定时器	00H	08H
	1	16 位定时器	01H	09H
	2	8 位自动重装	02H	0AH
	3	两个 8 位定时器	03H	0BH
作为计数器	0	13 位定时器	04H	0CH
	1	16 位定时器	05H	0DH
	2	8 位自动重装	06H	0EH
	3	两个 8 位定时器	07H	0FH

注 1: 由软件控制设定和清除 TR0 位来控制定时器的开始和关闭。

2: 硬件控制时, 在 INT0# (P3.2) 从 1 到 0 的变化控制定时器的开始和关闭。

**表 5-2: 定时器/计数器 1**

	模式	功能	TMOD	
			内部控制 (注 1)	外边控制 (注 2)
作为定时器	0	13 位定时器	00H	80H
	1	16 位定时器	10H	90H
	2	8 位自动重装	20H	A0H
	3	两个 8 位定时器	30H	B0H
作为计数器	0	13 位定时器	40H	C0H
	1	16 位定时器	50H	D0H
	2	8 位自动重装	60H	E0H
	3	两个 8 位定时器	-	-

注 1: 由软件控制设定和清除 TR1 位来控制定时器的开始和关闭。

2: 硬件控制时, 在 INT1# (P3.3) 从 1 到 0 的变化控制定时器的开始和关闭。

**表 5-3: 定时器/计数器 2**

	模式	TMOD	
		内部控制 (注 1)	外边控制 (注 2)
作为定时器	16 位重装	00H	08H
	16 位捕获	10H	09H
	波特率发生器接收和发送相同波特率	34HH	36H
	只接收	24H	26H
	只发送	14H	16H
作为计数器	16 位	02H	0AH
	自动重装	03H	0BH

注 1: 仅当定时器溢出时捕获和重装。

2: 当定时器/计数器溢出并且 T2EX (P1.1) 发生电平负跳变时产生捕获和重装 (定时器 2 用于波

特率发生器模式时除外)。

### 5.3 可编程时钟输出

可从 P1.0 编程输出 50% 占空比的时钟信号。P1.0 除了作为常规 I/O 口外，还有两个可选功能。它可编程为：

1. 用于定时器/计数器 2 的外部时钟输入。
2. 12 时钟模式时，在 16MHz 操作频率下输出频率为 61Hz 到 4MHz 的 50% 占空比的时钟信号（6 时钟模式时为 122Hz~8MHz）。

要将定时器/计数器 2 配置为时钟发生器，C/#T2 (T2CON.1) 必须清零，T2MOD 中的 T2OE 位必须置位。启动定时器 2 必须将 TR2 (T2CON.2) 置位。时钟输出频率由振荡器频率和定时器 2 捕获寄存器的重新装入值确定，公式如下：

$$n \times \frac{\text{振荡器频率}}{(65536 - \text{RCAP2H}, \text{RCAP2L})} \quad \begin{matrix} n=2 \text{ (6 时钟模式)} \\ n=4 \text{ (12 时钟模式)} \end{matrix}$$

(65536 - RCAP2H, RCAP2L) = RCAP2H 和 RCAP2L 的内容作为一个 16 位无符号整数

在时钟输出模式中，定时器 2 的翻转将不会产生中断，这和它作为波特率发生器时相似。定时器 2 可同时作为波特率发生器和时钟发生器。但需要注意的是，波特率和时钟输出频率相同。

## 6.0 串口 I/O

### 6.1 全双工增强型 UART

SST89E/V54RC 和 SST89E/V564RD 有一个全双工端口，可以同时接收和发送数据，串口的发送和接收寄存器都是通过特殊功能寄存器 SBUF 进行访问，写入 SBUF 的数据装入发送寄存器，对 SBUF 的读获得接收寄存器的内容。

串口有四种操作模式，由串口控制寄存器 SCON 的 SM0 和 SM1 位选择。每个模式中，把 SBUF 寄存器作为目标寄存器的指令将启动数据发送。接收时，在模式 0 下，当 SCON 的接收中断 RI 标志位被清除和 SCON 的接收允许/禁止 (REN) 位被设定，开始接收；在其它模式，当 SCON 的 REN 位已设，接收在收进来的开始位开始。

#### 6.1.1 帧错误检测

帧错误检测功能允许接收控制器检测模式 1、2、3 的正确停止位，串口线噪音或两个 CPU 同时发射可能丢掉停止位。进入 PCON 寄存器把 SMOD0 改为 1 (看图 6-1) 将选择帧错误检测。如果丢了一个停止位，将给帧错误位 FE 设值，软件可以在每次接收后查 FE 位检测是否有数据出错，一旦 FE 位被设定，只能由软件清除。有效的停止位不会清 FE，当允许 FE，RI 在停止位上升，而不是最后的数据位。

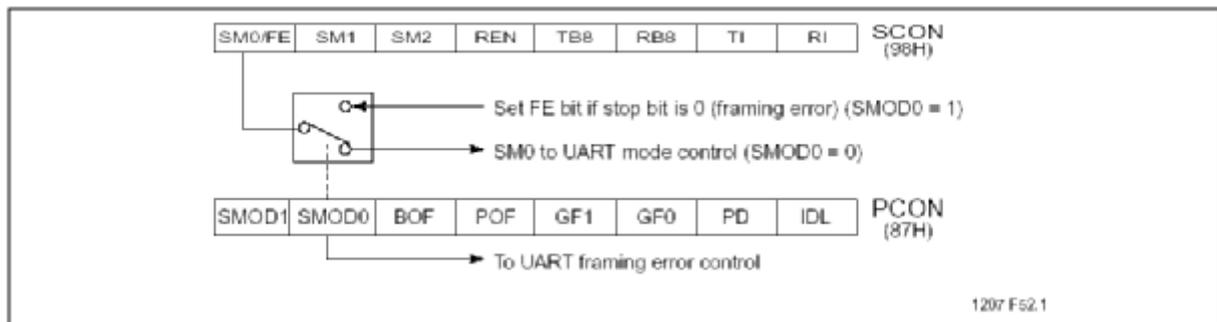


图 6-1 帧错误方框图

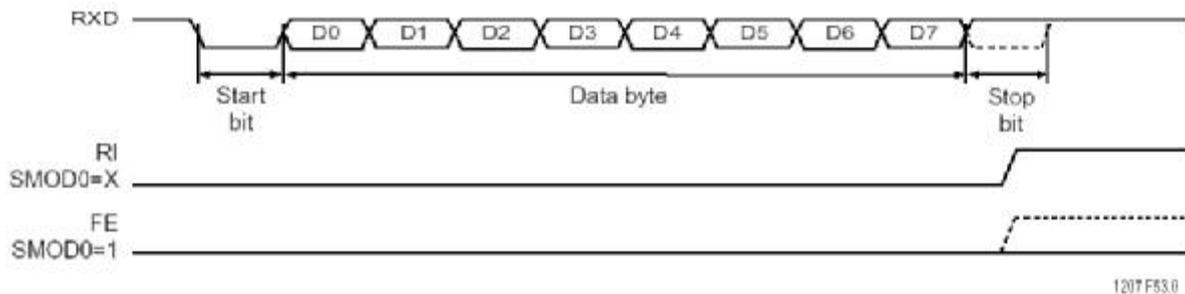


图 6-2 UART 口在模式 1 的时序

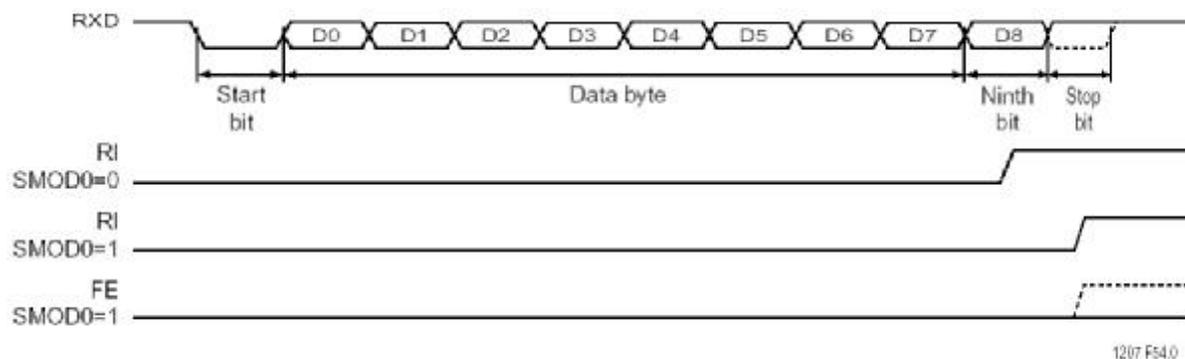


图 6-3 UART 口在模式 2 和 3 的时序

### 6.1.2 自动地址识别

自动地址识别可以帮助 MCU 与多个串口设备通讯，减少单片机的时间和功耗。每个设备可以挂在一起，共用同一个串口连接，但有自己的地址。器件只有接收自己的地址才会响应，从而减少软件消耗去比较地址。自动地址识别还可以结合 IDLE 模式降低系统的功耗。由于可以有多个从机连到一个主机，只有一个从机从 IDLE 模式被唤醒，回应主机的发送，自动地址识别 AAR 可以允许其它的从机保持 IDLE 模式而只要一个会中断。通过限制中断的次数，整个系统的功耗会减少。

主机有两种方式和从机通讯：一组从机和所有的从机。和一组从机通讯，主机给出的地址叫“给定”地址，和所有从机通讯，主机给的地址叫“广播”地址。

AAR 可以配置成模式 2 或 3（9 位模式），设 SCON 的 SM2 位。每个从机有自己的 SM2 位设置，等待一个地址字（第 9 位=1）。当接收数据等于“给定”地址或“广播”地址，接收中断标记 RI 才会置位。接着从机清 SM2 位，允许接收从主机来的数据字节（第 9 位=0）。（当第 9 位=1，主机送的是地址；第 9 位=0，主机送的是实际的数据。）如果采用模式 1，第 9 位变成是停止位。当接收到的命令帧地址符合器件的地址，RI 位被设定，由合法停止位终止。注意不能用模式 0，在模式 0 下设 SCON 的 SM2 位是无效的。

每个从机的独立地址是由特殊功能寄存器 SADDR 决定，SADDR 表示从机地址，SADEN 是一个掩码，表示地址屏蔽，用于定义 SADDR 内哪几位需使用而哪几位不考虑。看下面的例子：

#### Slave 1

SADDR = 11110001

SADEN = 11111010

GIVEN = 11110x0x

#### Slave 2

SADDR = 11110011

SADEN = 11111001

GIVEN = 11110xx1

### 6.1.2.1 用“给定”地址选择从机

“给定”地址的位被 SADEN 用 0 屏蔽后变成不考虑，而是 1 的位置则变成 ANDED。这种不考虑的特性让用户有自己定义地址的灵活性，用给定的地址选择更多的从机。

从上面的例子看，从机 1 给了地址 1111 0001 (SADDR)。SADEN 用来屏蔽“给定”地址位，以方便从机 1 和从机 2。在例子中，从机 1 的最末位是不考虑，而从机 2 的最末位是 1，要和两个从机通讯，主机需要发一个最末位是 1 的地址（如 1111 0001），因为从机 1 的最末位是不考虑而从机 2 的最末位要等于 1。如果只要和从机 1 通讯，主机就发出一个最末位为 0 的地址（如 1111 0000），而从机 2 的最末位要等于 1。请看下面的选择：

只选从机 1		
从机 1	给定地址	可能的地址
	1111 0x0x	1111 0000 1111 0100

只选从机 2		
从机 2	给定地址	可能的地址
	1111 0xx1	1111 0111 1111 0011

选从机 1 和 2		
从机 1 和 2	可能的地址	
	1111 0001 1111 0101	

如果加了第 3 个从机，如下：

#### Slave 3

SADDR = 1111 1001

SADEN = 1111 0101

GIVEN = 1111 x0x1

只选从机 3		
从机 3	给定地址	可能的地址
	1111 x0x1	1111 1011 1111 1001

用户可以用上面的地址来选择从机 3。选择从机 2 和从机 3 的组合是：

选从机 2 和 3		
从机 2 和 3	可能的地址	
	1111 0011	

### 6.1.2.2 用“广播”地址选择从机

用“广播”地址，主机可以和所有的从机通讯。它是由 SADDR 和 SADEN 的逻辑“或”组成的，结果为“0”的当成不考虑。

$$\begin{array}{rcl}
 & \text{Slave 1} & \\
 1111\ 0001 & = & \text{SADDR} \\
 +1111\ 1010 & = & \text{SADEN} \\
 \hline
 1111\ 1x11 & = & \text{广播地址}
 \end{array}$$

单片机复位后，SADDR 和 SADEN 为“0”，这样可以使“给定”地址是不考虑，“广播”地址也是不考虑，从而关闭自动地址识别模式。

## 7.0 看门狗定时器

单片机内部有一个可编程的看门狗（WDT），用来防止软件的死锁和自动保护。为了保护系统，防止软件的死锁，用户软件要在一个给定的时间刷新 WDT。在 WDRE=1 时，如果软件在给定时间没有刷新 WDT，内部将产生一个硬件复位。所以 WDT 可以用来设计成如果程序没有正确运行，就 WDT 溢出。

WDT 使用系统时钟（XTAL1）作为时间的基准，所以严格来讲，它是一个看门狗计数器，而不是定时器。WDT 寄存器在 344,064 个晶振时钟后增加。时间基准寄存器（WDTD）的高 8 位当作重装寄存器。

WDT 溢出将把 WDTS 标志置位，WDT 复位不能改变它的值，用户程序写“1”可以清 WDTS。图 7-1 是 WDT 的方框图，WDT 和 WDTD 两个 SFR 控制看门狗操作。在 IDLE 模式，WDT 操作相应地挂起，从 IDLE 模式出来后恢复。

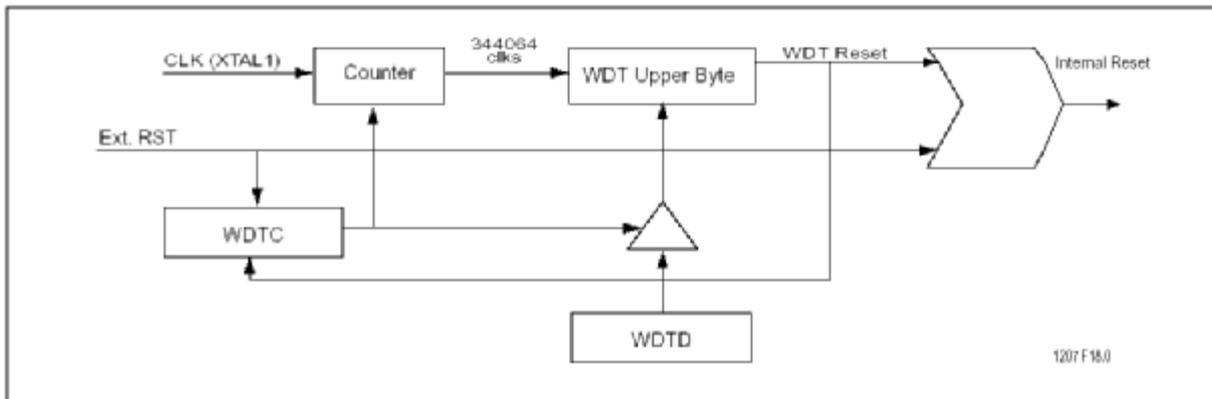


图 7-1 看门狗定时器方框图

WDT 的溢出的时间由下面公式计算：

$$\text{时间} = (255 - \text{WDTD}) * 344064 * 1/f_{\text{CLK (XTAL1)}}$$

WDTD 是在 WDTD 寄存器的值， $f_{\text{OSC}}$  是振荡频率。

## 8.0 可编程计数器阵列 PCA

SST89E/V554RC 和 SST89E/V564RD 的可编程计数器阵列是有 5 个 16 位捕获/比较模块组成的特殊定时器，每个模块都可以经编程实现以下 4 种模式：捕获模式、软件定时器模式、高速输出模式和 PWM 脉宽调制模式。每个模块都有一个 P1 口管脚与之对应，模块 1 连接到 P1.3（CEX0），模块 1 连接到 P1.4（CEX1），模块 2 连接到 P1.5（CEX2），模块 3 连接到 P1.6（CEX3），模块 4 连接到 P1.7（CEX4）。PCA 的配置请看图 8-1。

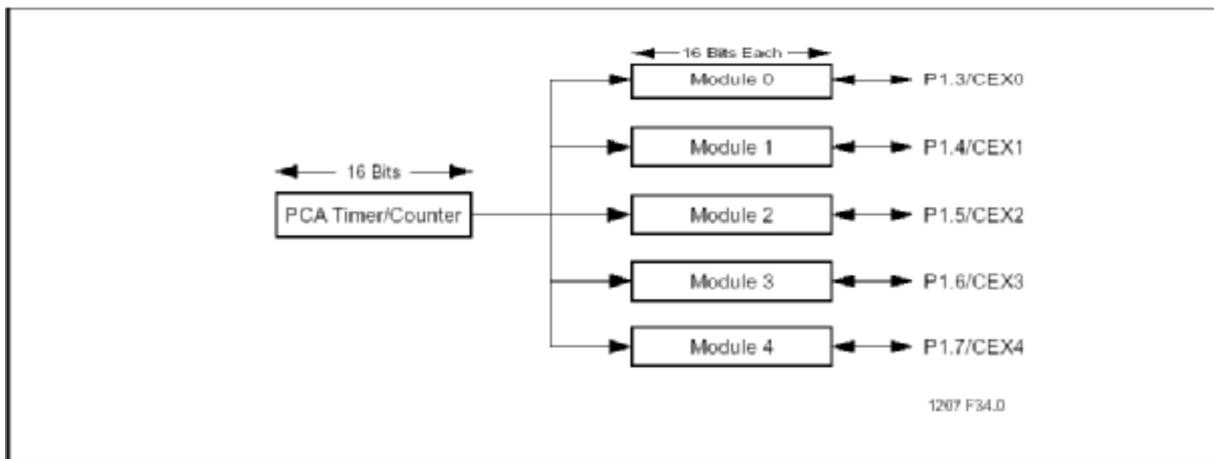


图 8-1 PCA 定时/计数器和比较/捕获模块

### 8.1 PCA概述

PCA 功能比普通的单片机相比提供更强计时性，同时更少 CPU 的干预。用它可以减少软件的开销和改进精度。PCA 的 5 个模块有一个专用的定时/计数器做时间基准。没有使用到的模块可以做普通的 I/O 脚。每个模块都可以编程为下面的模式：

- 1、上升和/或下降沿捕获
- 2、软件定时器
- 3、高速输出
- 4、看门狗（只有模块 4 能实现）
- 5、脉宽调制（PWM）

### 8.2 PCA定时/计数器

PCA 定时器是一个自己跑的 16 位定时器，由寄存器 CH 和 CL（计数值的高位和低位）组成。PCA 定时器是 5 个模块的公共时间基准，可编程为：1/6 振荡器频率，1/2 振荡器频率，定时器 0 溢出和 ECI 脚（P1.2）输入。定时器的计数源由 CMOD 中的 CPS1 和 CPS0 决定。

表 8-1 PCA 定时/计数源

CPS1	CPS0	12 时钟模式	6 时钟模式
0	0	$F_{osc}/12$	$F_{osc}/6$
0	1	$F_{osc}/4$	$F_{osc}/2$
1	0	定时器 0 溢出	定时器 0 溢出
1	1	ECI 脚的外部中断（最大速率= $f_{osc}/8$ ）	ECI 脚的外部中断（最大速率= $f_{osc}/4$ ）

表 8-2 PCA 定时/计数器输入

PCA 定时/计数器模式	时钟增加	
	12MHz0	16MHz
模式 0: $f_{osc}/12$	1 $\mu$ sec	0.75 $\mu$ sec
模式 1:	330 nsec	250 nsec
模式 2: 定时器 0 溢出 <sup>(注 1)</sup>		
定时器 0 编程为		
8 位模式	256 $\mu$ sec	192 $\mu$ sec

16 位模式	65 msec	49 $\mu$ sec
8 位自动重装	1 到 255 $\mu$ sec	0.75 到 191 $\mu$ sec
模式 3: 外部输出最大	0.66 $\mu$ sec	0.50 $\mu$ sec

注 1: 在模式 2, 定时器 0 的溢出中断不需要使能。

CMOD 的 4 个可能定时器模式 (带中断和不带中断) 见下表, 列表假使 PCA

表 8-3 CMOD 的值

PCA 计数脉冲选择	CMOD 值	
	不带中断使能	带中断使能
内部时钟, $f_{osc}/12$	00H	01H
内部时钟, $f_{osc}/4$	02H	03H
定时器 0 溢出	04H	05H
P1.2 的外部时钟	06H	07H

CCON 寄存器和所有的 PCA 定时器功能有关, 它包含了 PCA 定时器和所有模块的运行控制位和标志。要运行 PCA, CR 位必须由软件置位, 清零该位则关闭 PCA。当 PCA 计数器溢出, CF 位 (CCON.7) 置位, 此时如果 CMOD 寄存器的 ECF 位是 1 则产生中断。CF 位只能由软件清零。每个模块有自己的定时器中断或捕获中断位 (模块 0 是 CCF0, 模块 4 是 CCF4 等)。当发生匹配/捕获时置位, 只能由软件清零。

### 8.3 比较/捕获模块

每个 PCA 模块有一个 SFR 对应。这些寄存器是: 模块 0 的 CCAPM0, 模块 1 的 CCAPM1 等, 它们每个保留 7 位, 用来控制将要执行模块的模式。ECCF 位 (CCAPMn.0,  $n=0, 1, 2, 3, 4$ , 对应不同的模块) 使 CCON 的 CCF 标志使能, 当符合或比较发生时产生中断。PWM (CCAPMn.1) 进入脉冲宽度调制模式。TOG 位 (CCAPMn.2) 置位, 当 PCA 计数器与模块的捕获/比较寄存器匹配时, CEX 输出发生翻转。MAT 位 (CCAPMn.3) 置位, 当 PCA 计数器与模块的捕获/比较寄存器匹配时, CCON 的 CCFn 位置位。

CAPN (CCAPMn.4) 和 CAPP (CCAPMn.5) 决定捕获输入是被上升沿或下降沿激活。CAPN 位使能下降沿, CAPP 位使能上升沿。如果两个位都置位, 那么在在下降和上升沿均可使能产生捕获。寄存器的最后一位 ECOM (CCAPMn.6) 置位, 使能比较器功能。表 8-5 是 CCAPMn 设置对应不同的 PCA 功能。

有两个额外的寄存器与每个 PCA 模块有关: CCAPnH 和 CCAPnL。它们保存用于发生捕获或比较的 16 位计数值。当模块用于 PWM 模式时, 这两个寄存器用来控制输出信号的占空比。看图 8-1。

表 8-4 PCA 高位和低位寄存器的比较/捕获模块

名称	说明	直接地址	位地址, 符号, 或可选口的功能		复位值
			MSB	LSB	
CCAP0H	PCA 模块 0 比较/ 捕获寄存器	FAH	CCAP0H[7:0]		00H
CCAP0L		EAH	CCAP0L[7:0]		00H
CCAP1H	PCA 模块 1 比较/ 捕获寄存器	FBH	CCAP1H[7:0]		00H
CCAP1L		EBH	CCAP1L[7:0]		00H
CCAP2H	PCA 模块 2 比较/ 捕获寄存器	FCH	CCAP2H[7:0]		00H
CCAP2L		ECH	CCAP2L[7:0]		00H
CCAP3H	PCA 模块 3 比较/ 捕获寄存器	FDH	CCAP3H[7:0]		00H
CCAP3L		EDH	CCAP3L[7:0]		00H

CCAP4H	PCA 模块 4 比较/ 捕获寄存器	FEH	CCAP4H[7:0]	00H
CCAP4L		EEH	CCAP4L[7:0]	00H

表 8-5 PCA 模块模式

不带中断使能								
<sup>-1</sup>	ECOMy <sup>2</sup>	CAPPy <sup>2</sup>	CAPNy <sup>2</sup>	MATy <sup>2</sup>	TOGy <sup>2</sup>	PWMy <sup>2</sup>	ECCFy <sup>2</sup>	模块功能
-	0	0	0	0	0	0	0	没有操作
-	0	1	0	0	0	0	0	16 位捕获, CEXn 上升沿触发
-	0	0	1	0	0	0	0	16 位捕获, CEXn 下降沿触发
-	0	1	1	0	0	0	0	16 位捕获, CEXn 上升/下降沿触发
-	1	0	0	1	0	0	0	比较: 软件定时器
-	1	0	0	1	1	0	0	比较: 高速输出
-	1	0	0	0	0	1	0	比较: 8 位 PWM
-	1	0	0	1	0 或 1 <sup>3</sup>	0	0	比较: PCA WDT (CCAPM4) 4

注 1、用户不能写“1”到该保留位，从保留位读的值是不确定的。

2、y=0, 1, 2, 3, 4。

3、0 是禁止翻转，1 是允许翻转。

4、对于 PCA WDT 模式，也可以置 CMOD 寄存器的 WDTE 位，从而允许复位输出信号。

表 8-6 PCA 模块模式

带中断使能								
<sup>-1</sup>	ECOMy <sup>2</sup>	CAPPy <sup>2</sup>	CAPNy <sup>2</sup>	MATy <sup>2</sup>	TOGy <sup>2</sup>	PWMy <sup>2</sup>	ECCFy <sup>2</sup>	模块功能
-	0	1	0	0	0	0	1	16 位捕获, CEXn 上升沿触发
-	0	0	1	0	0	0	1	16 位捕获, CEXn 下降沿触发
-	0	1	1	0	0	0	1	16 位捕获, CEXn 上升/下降沿触发
-	1	0	0	1	0	0	1	比较: 软件定时器
-	1	0	0	1	1	0	1	比较: 高速输出
-	1	0	0	0	0	1	X <sup>3</sup>	比较: 8 位 PWM
-	1	0	0	1	0 或 1 <sup>4</sup>	0	X <sup>5</sup>	比较: PCA WDT (CCAPM4) <sup>5</sup>

注 1、用户不能写“1”到该保留位，从保留位读的值是不确定的。

2、y=0, 1, 2, 3, 4。

3、不需要 PCA 中断来产生 PWM。

4、0 是禁止翻转，1 是允许翻转。

5、让看门狗定时器使能会影响看门狗定时器的功能。

4、对于 PCA WDT 模式，也可以置 CMOD 寄存器的 WDTE 位，从而允许复位输出信号。

### 8.3.1 捕获模式

捕获模式是用来捕获在模块捕获寄存器的 PCA 定时/计数器值 (CCAPnH 和 CCAPnL)。捕获可发生在上升沿、下降沿、或上升/下降都捕获，要把某个 PCA 模块设成捕获模式，一个或多个 CCAPM 的 CAPN 位和 CAPP 为要置位。当 CEX 脚有一个正式的跳变发生，PCA 硬件把 PCA 计数寄存器 (CH 和 CL) 的 16 位值 装入模块的捕获寄存器 (CCAPnL 和 CCAPnH)。如果模块的 CCON 寄存器 CCFn 位

和 CCAPMn 寄存器的 ECCFn 被置位，那么产生将产生一个中断。在中断服务子程序，16 位捕获值必须存到 RAM 里直到下一个事件捕获发生。如果一个并发捕获发生，原来的捕获值会丢掉。在标志事件后标志已经由硬件置位，用户需要在软件里清标志。

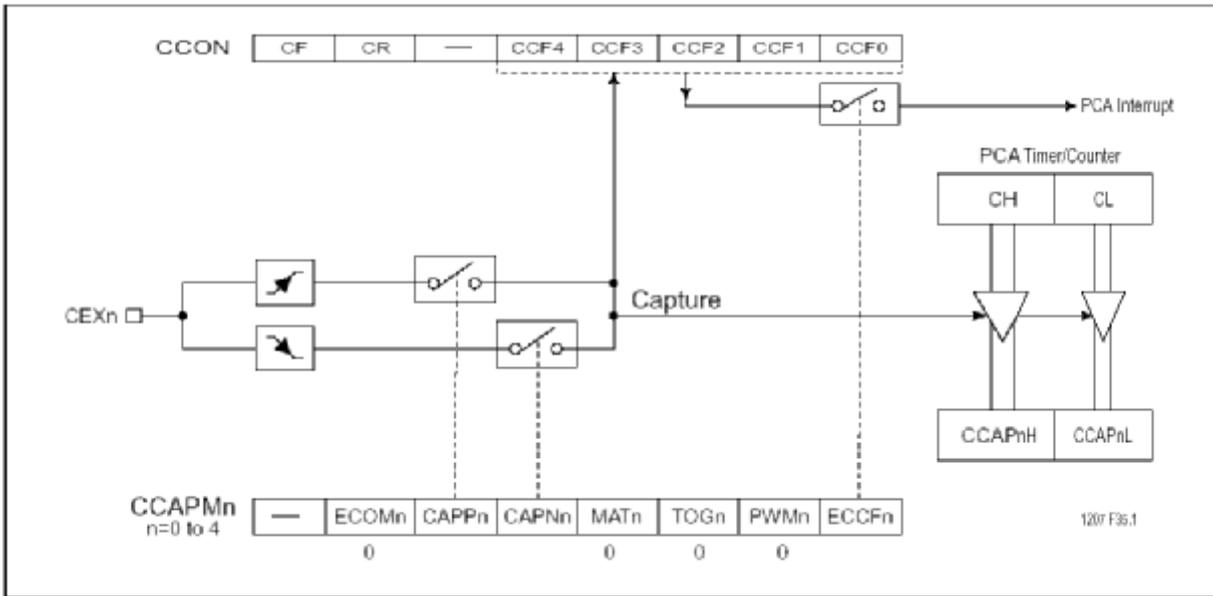


图 8-2 PCA 捕获模式

### 8.3.2 16位软件定时器模式

16 位软件定时器模式用在需要在一个时间间隔内触发中断子程序，它通过对 CCAPMn 寄存器的 ECOM 和 MAT 位置位来启动。PCA 定时器与模块的捕获寄存器（CCAPnL 和 CCAPnH）比较，如果 CCFn（CCON SFR）和 ECCFn（CCAPMn SFR）都被置位，当比较匹配时会产生一个中断。如果需要，可以在中断子程序中把一个新的 16 位比较值装入 CCAPnH 和 CCAPnL。用户要注意，硬件会在这些寄存器被更新时临时地禁止比较功能，从而避免发生非法比较。所以，建议用户先写低位 CCAPnL 禁止比较功能，然后写高位 CCAPnH 重新使能。

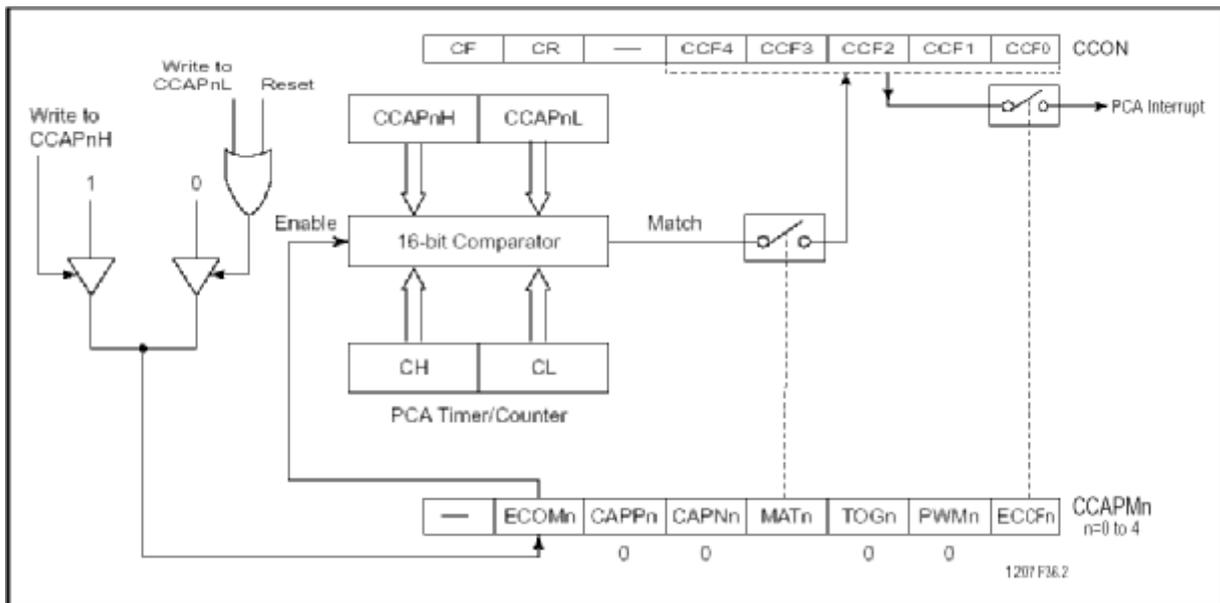


图 8-3 PCA 比较模式（软件定时器）

### 8.3.3 高速输出模式

高速输出模式是在 PCA 定时器和在比较寄存器的预置值匹配时做翻转。本模式下，每次 PCA 计数器（CH 和 CL）与捕获寄存器（CCAPnH 和 CCAPnL）匹配时 CEX 输出脚会翻转。用户通过对模块的 CCAPMn 的 TOG、MAT 和 ECOM 位置位进入该模式。高速输出模式比触发脚的精度要高，因为触发是在执行中断前产生。使用高速输出时，中断是可选可不选。只有在用户希望改变时间做下一个触发，才需要改变比较寄存器，否则，当 PCA 定时器翻转并和上一个比较值匹配时，发生下一个翻转。

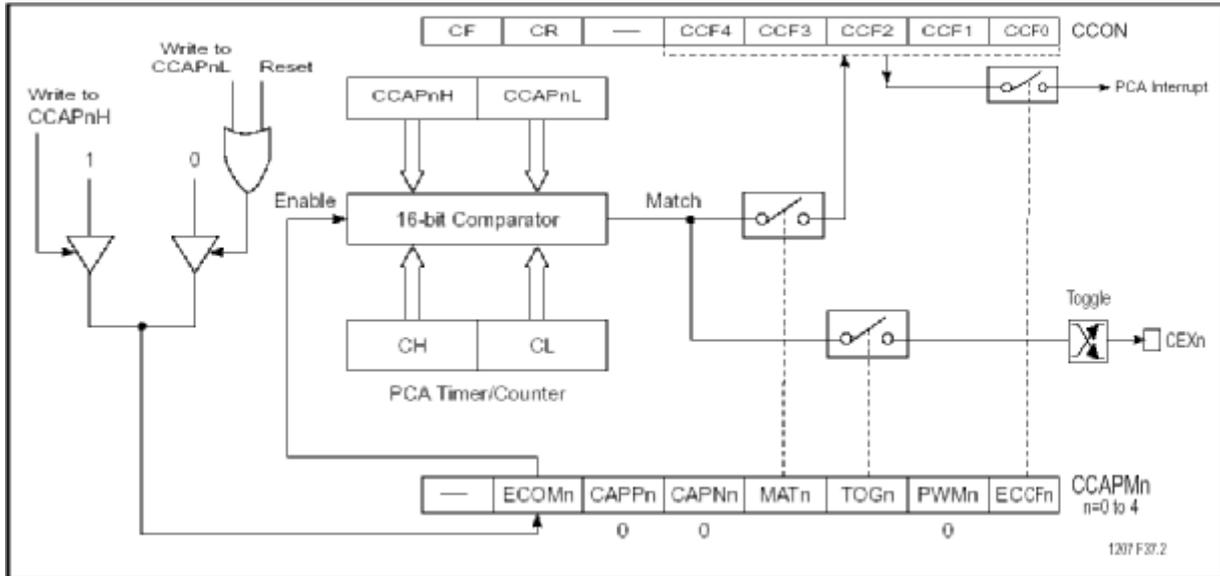


图 8-4 PCA 高速输出模式

### 8.3.4 脉冲宽度调制

脉冲宽度调制（PWM）模式用来产生一个 8 位 PWM，它通过比较 PCA 定时器的低位（CL）和比较寄存器的低位（CCAPnL）产生。当  $CL < CCAPnL$  时输出为低；当  $CL \geq CCAPnL$  是输出为高。要进入这个模式，可以对 CCAPMn 的 PWM 和 ECOM 位置位。（看图 8-5 和表 8-7）

在 PWM 模式，输出的频率决定于 PCA 定时器的源。由于只有一对 CH 和 CL 寄存器，所有模块共享 PCA 定时器和频率。输出的占空比由装在高位 CCAPnH 的值控制。由于写入 CCAPnH 寄存器不是同步的，写入高位的值不会马上装入 CCAPnL 做比较，直到输出的另一个周期（当 CL 从 255 到 0 翻转）。用下面的公式计算任何占空比的 CCAPnH 值：

$$CCAPnH = 256 \times (1 - \text{占空比})$$

CCAPnH 是 8 位整数，占空比是分数。

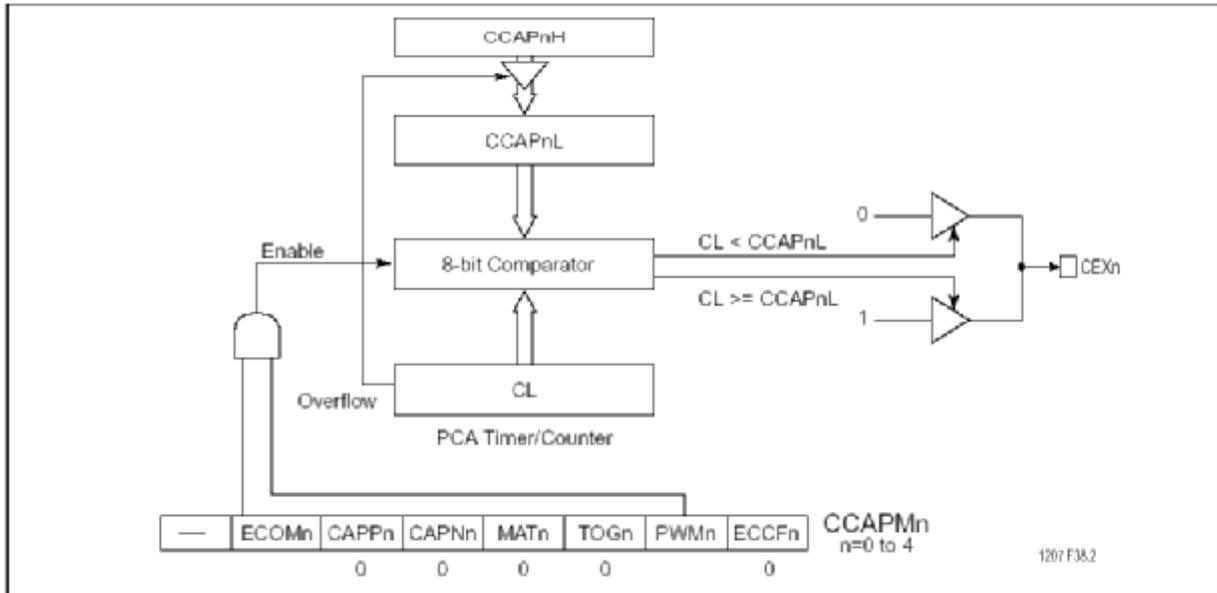


图 8-5 PCA 脉冲宽度调制模式

表 8-7 脉冲宽度调制频率

PCA 定时器模式	PWM 频率	
	12MHz	16MHz
1/12 振荡频率	3.9KHz	5.2KHz
1/4 振荡频率	11.8KHz	15.6KHz
定时器 0 溢出:		
8 位	15.5Hz	20.3Hz
16 位	0.06Hz	0.08Hz
8 位自动重装	3.9KHz 到 15.3KHz	5.2KHz 到 20.3KHz
外部输入 (最大)	5.9KHz	7.8KHz

### 8.3.5 PCA看门狗定时器

由 PCA 实现的看门狗模式可以改善系统的稳定性而不需要增加芯片的成本。看门狗定时器对于那些受噪音、电源杂波或静电放电影响的系统有好处，也可以用来防止软件的死锁。只有 PCA 模块 4 可编程为看门狗，如果不做看门狗，该模块也可做其它模式。

用户将一个 16 位数值预先装入比较寄存器，与其它比较模式一样，该 16 位值与 PCA 定时器值比较，如果相同则产生一个内部复位，但不会将管脚 RST 拉高。为了防止发生复位，用户有下列 3 个选择：

- 1、 周期性改变比较的数值，让它不会与 PCA 定时器的值相同。
- 2、 周期性改变 PCA 定时器的值，使之不会与比较的数值相同。
- 3、 在发生匹配之前将 WDTE 位清零，然后再重新使它。

前两个选项更为可行，因为看门狗定时器从来不像第 3 项那样被禁止。如果程序指针跑飞，最终会发生匹配并产生内部复位信号。如果其它模块在使用时，最好不用第 2 个选项，因为 PCA 定时器是所有模块的时间基准，改变它对其它模块都有影响。因此，在大多数应用中，第 1 项是最好的。

下面是看门狗定时器的初始化代码，模块 4 配置为比较模式，同时 CMOD 中的 WDTE 位必须置位。用户的软件必须周期性改变 (CCAP4H, CCAP4L)，防止发生复位。

Init\_Watchdog:

```

MOV  CCAPM4, #4CH      ; 模块 4 为比较模式
MOV  CCAP4L, #0FFH    ; 先写低位字节
MOV  CCAP4H, #0FFH    ; 在 PCA 定时器计数到 FFFF 前必须改变此比较值
ORL  CMOD, #40H       ; 置位 WDTE 位以使能看门狗定时器, 不改变
                        ; CMOD 的其它位
    
```

; -----  
; 主程序, 周期性调用子程序 WATCHDOG  
; -----

WATCHDOG:

```

CLR  ER                ; 关中断
MOV  CCAP4L, #00       ; 下一个比较值与当前 PCA 定时器值相差不超过 255 个计数
MOV  CCAP4H, CH
SETB EA
    
```

; -----  
该程序不能作为一个中断服务程序, 因为如果程序指针跑飞并进入一个死循环, 仍然执行中断看门狗复位, 这样起不到使用看门狗目的, 应当在主程序内调用该程序。

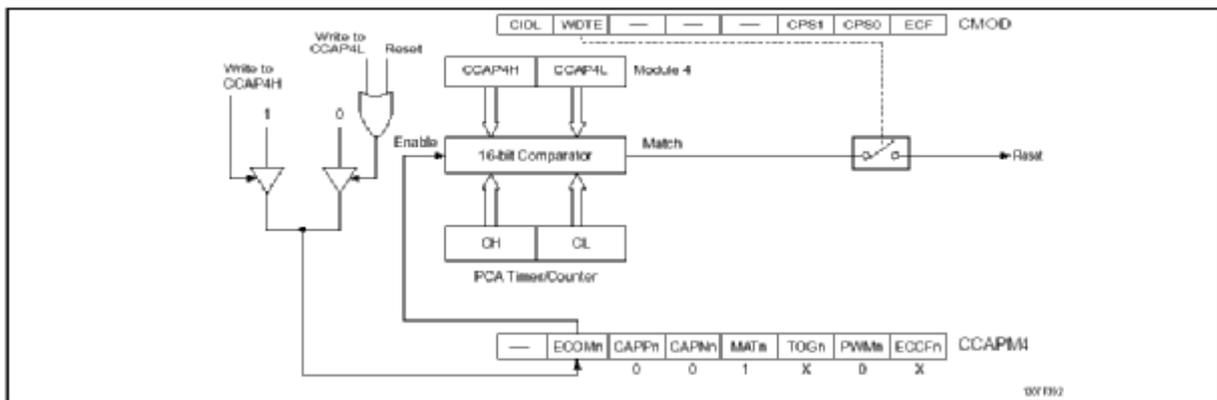


图 8-6 PCA 看门狗定时器 (只要模块 4)

## 9.0 安全加锁

安全加锁是为了避免软件侵权和防止 FLASH 的内容被非法读取, 也可以防止内部 FLASH 被意外擦除和编程导致代码破坏。SST89E/V554RC 和 SST89E/V564RD 有两种加密的方式: 硬加密和软加密。

### 9.1 硬加密

当硬加密被激活, 从非加密或软加密地址执行的 MOV<sub>VC</sub> 和 IAP 指令不能读取硬加密空间的内容(看表 9-2)。硬加密可以锁两块 FLASH 空间或 8Kbyte 的 FLASH 存储块 (BLOCK1)。所有外部主机和 IAP 指令 (除了整片擦除 CHIP-ERASE) 对硬加密的块都无效。

### 9.2 软加密

软加密允许 FLASH 的内容杂安全情况下被修改。这个加密的选择让用户在安全环境通过 IAP 模式升级软加密存储块的程序代码。例如, BLOCK1 (8K) 存储器块被加锁 (硬加密或软加密), BLOCK0 被软加密, 在 BLOCK1 的代码可以改写 BLOCK0, 以下从加密块执行的 IAP 指令能对软加密的存储块操作: 块擦除、扇区擦除、字节编程和字节校读。在外部模式时, 软加密和硬加密一样。

### 9.3 安全加密状态

控制器件加密状态的有三个位, 在 SFST[7:5], 请看图 9-1 和表 9-1 中, 加密状态有 4 个不同的等级。第一级, 没有一个加密位被烧写, 两个存储块没有加密; 在第二级, 两个块都被加密成不能被编

程，但还能通过字节校读命令读取内容；在第三级，有三个不同的选择：块 1 硬加密/块 0 软加密，两块都是软加密，和两块都是硬加密。两个块都是软加密与第二级相同，除了块 1 的不能读。第四级是最高加密等级，不能读取/改写内部存储器或从外部存储器启动。如何编程安全加密等级，请参考外部主机模式和 IAP 的章节。

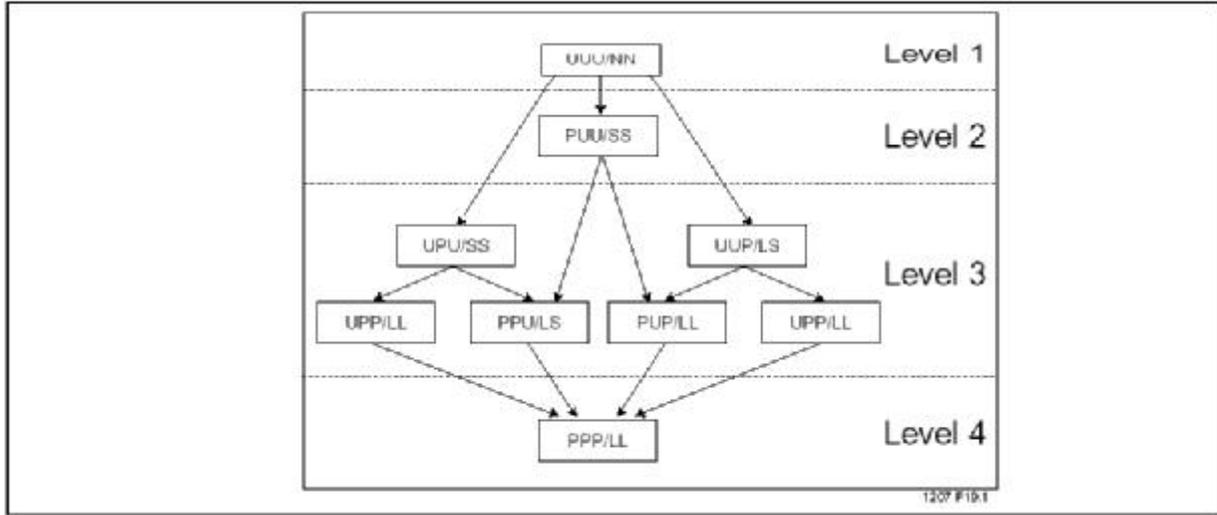


图 9-1 安全加密等级

注：P=被编程（位逻辑状态=0），U=没被编程（位逻辑状态=1），N=未加密，L=硬加密，S=软加密

表 9-1：安全加密选择

等级	加密位			加密状态		加密类型	
	SFST[7:5]	SB1	SB2	SB3	BLOCK1		BLOCK0
1	000	U	U	U	不加密	不加密	没有启动加密
2	100	P	U	U	软加密	软加密	外部程序空间执行的 MOVC 指令读取内部存储器，EA#在复位时被取样和锁存，改写 FLASH 也被禁止。
3	011	U	P	P	硬加密	硬加密	等级 2 加上禁止校读，两个块加密。
	101	P	U	P			
	010	U	P	U	软加密	软加密	等级 2 加上禁止校读。在块 1 的代码可以改块 0。反之一样。
	110	P	P	U	硬加密	软加密	等级 2 加上禁止校读。在块 1 的代码可以改块 0。
001	U	U	P				
4	111	P	P	P	硬加密	硬加密	和等级 3 的硬加密/硬加密一样，但是 CPU 只从内部程序执行代码，而不管 EA#的状态。

1、P=被编程（位逻辑状态=0），U=未编程（位逻辑状态=1）

2、SFST[7:5]=加密位（SB1<sub>i</sub>, SB2<sub>i</sub>, SB3<sub>i</sub>）

#### 9.4 在加密状态的读操作

加密位 SB1、SB2 和 SB3 的状态在读操作被加密禁止后还能读取，读取的方式有三种：

- 1、外部主机模式：读回=00H（加密）
- 2、IAP 指令：读回=之前的 SFDT 数据

3、 MOVc: 读回=FFH (空)

表 9-2: 加密下的存取表

等级	SFST[7:5]	源地址 <sup>1</sup>	目标地址 <sup>2</sup>	允许字节校读		允许 MOVc		
				外部主机	IAP	564RD	554RC	
4	111b 两块都被硬加密	块 0/1	块 0/1	N	N	Y	Y	
			外部	N/A	N/A	N	Y	
		外部	块 0/1	N	N	N	N	
			外部	N/A	N/A	N	Y	
3	011b/101b 两块都被硬加密	块 0/1	块 0/1	N	N	Y	Y	
			外部	N/A	N/A	N	Y	
		外部	块 0/1	N	N	N	N	
			外部	N/A	N/A	Y	Y	
	001b/110b 块 0=软加密, 块 1=硬加密	块 0	块 0	N	N	Y	Y	
			块 1	N	N	N	N	
			外部	N/A	N/A	N	Y	
		块 1	块 0	N	Y	Y	Y	
			块 1	N	N	Y	Y	
			外部	N/A	N/A	N	Y	
		外部	块 0/1	N	N	N	N	
			外部	N/A	N/A	Y	Y	
			块 0	块 0	N	N	Y	Y
				块 1	N	Y	Y	Y
		外部		N/A	N/A	N	Y	
		010b 块 0/1=软加密	块 1	块 0	N	Y	Y	Y
	块 1			N	N	Y	Y	
	外部			N/A	N/A	N	Y	
	外部		块 0/1	N	N	N	N	
			外部	N/A	N/A	Y	Y	
块 0			块 0	Y	N	Y	Y	
	块 1	Y	Y	Y	Y			
	外部	N/A	N/A	N	Y			
100b 块 0/1=软加密	块 1	块 0	Y	Y	Y	Y		
		块 1	Y	N	Y	Y		
		外部	N/A	N/A	N	Y		
	外部	块 0/1	Y	N	N	N		
		外部	N/A	N/A	Y	Y		
		块 0	块 0	Y	N	Y	Y	
块 1	Y		Y	Y	Y			
外部	N/A		N/A	N	Y			
1	000b 未加密	块 0	块 0	Y	N	Y	Y	
			块 1	Y	Y	Y	Y	
			外部	N/A	N/A	N	Y	
		块 1	块 0	Y	Y	Y	Y	
			块 1	Y	N	Y	Y	
			外部	N/A	N/A	N	Y	

		外部	块 0/1	Y	Y	N	Y
			外部	N/A	N/A	Y	Y

- 1、 MOVC 或 IAP 指令的地址。
- 2、 准备要读取的地址。

### 10.0 复位

系统的复位将把 MCU 初始化，并且开始执行在地址 0000H 的程序 2 代码。器件的复位输入脚是 RST 脚。振荡稳定后，给 RST 脚加一个不少于 2 个机器周期（24 个时钟）的逻辑高电平，将把器件复位。ALE 和 PSEN#在复位时拉高。在复位时，ALE 和 PSEN#输出高电平，来完成正确的复位。这个不能被外部的零件影响。器件在工作时，系统的复位不会影响片内的 1Kbyte RAM，但如果是掉电的话，RAM 的内容就不确定了。复位后，所有的特殊功能寄存器恢复到默认值（看表 3-7 到 3-11）。

#### 10.1 上电复位

在初始上电，管脚都在随机状态，直到振荡器开始，内部复位的算法将把所有的引脚轻微的拉高。不通过合法的复位来启动器件会使 MCU 从一个不确定地址开始执行程序。这样不确定的状态可能会破坏 FLASH 的代码。

电压加给器件后，RST 脚要保持高位尽可能长，等到振荡器启动（对低频率振荡器通常要 2 微秒），加多 2 个机器周期做正确的上电复位。增加 RST 信号的一种方法是加一个 RC 电路，如图 10-1，RST 通过一个 10μF 电容接 V<sub>DD</sub> 和一个 8.2KΩ 电阻到 V<sub>SS</sub>。用了 RC 电路要防止 V<sub>DD</sub> 的上升时间不能超过 1ms，振荡器启动时间不超过 10ms。

低频振荡器的启动时间比较慢，需要延长复位信号，以配合较慢的启动时间。上电检测在上电时发生，在达到 BROWN-OUT 检测的水平。上电初始化后，在 PCON 寄存器的 POF 标志被置位，POF 标志保持有效，直到被软件清除。请参考 PCON 寄存器定义。

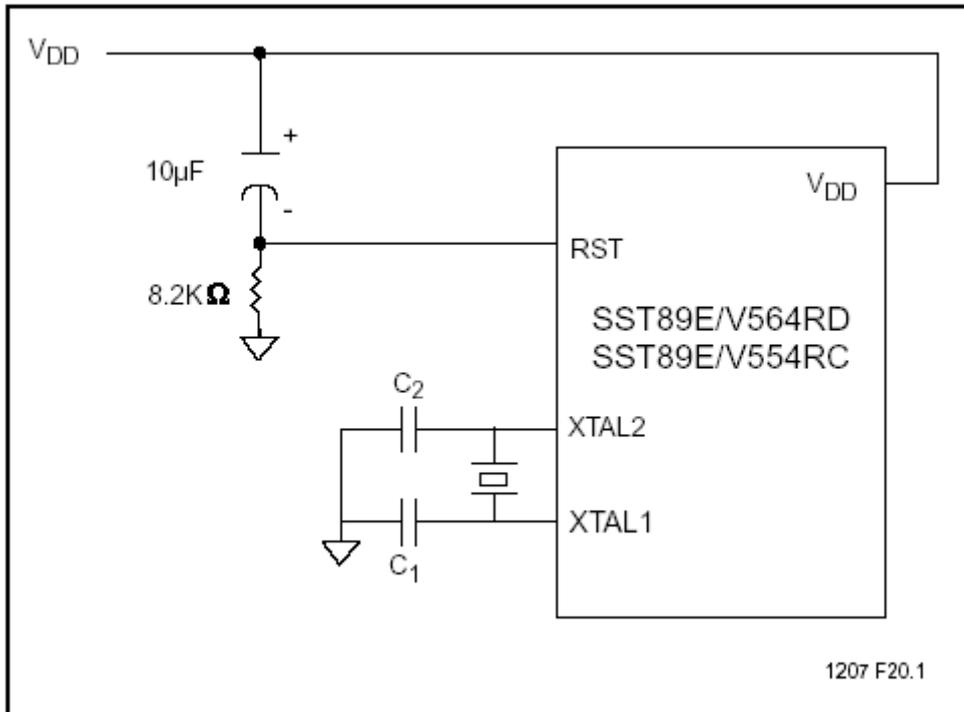


图 10-1: 上电复位电路

#### 10.2 软件复位

把 SFCF[1] (SWR) 从“0”转为“1”，将执行软件复位。复位后程序计数器变为地址 0000H，所

有 SFR 变成初始值，除了 SFCF[1] (SWR)，WDTC[2] (WDTS)，RAM 的数据也不会改变。

### 10.3 掉电检测 (Brown-out Detection) 复位

器件内部有一个掉电检测电路，防止电压  $V_{DD}$  的波动对系统造成影响。SST89E564RD 的内部掉电检测阈值是 3.85V，SST89E564RD 的内部掉电检测阈值是 2.35V，详细电压参数请看表 13-6 和 13-7。

当  $V_{DD}$  掉到阈值以下，掉电检测电路产生掉电中断，但 CPU 继续工作，直到供电电压回到掉电检测电压  $V_{BOD}$ 。掉电检测的默认操作是产生处理器复位。 $V_{DD}$  必须在  $V_{BOD}$  以下保持 4 个时钟时间，掉电检测电路才会响应。掉电中断由 IEA 寄存器的 EBO 位 (地址 E8H，位 3) 置位来使能。EBO 位被置位，发生掉电条件，将产生掉电中断，并执行 004BH 的程序。掉电中断被响应后，需要软件把 EBO 清除。在掉电环境下，清除 EBO 位可正确地把器件复位。如果掉电中断没有使能，掉电环境把程序重新开始。

### 10.4 中断优先级和查询顺序

器件支持 8 个中断源，4 个优先级。表 10-1 概括了支持的中断和顺序。

表 10-1: 中断服务顺序

说明	中断标志	矢量地址	中断使能	中断优先	服务优先	唤醒/掉电
外部中断 0	IE0	0003H	EX0	PX0/H	1 (最高)	是
掉电	-	004BH	EBO	PBO/H	2	否
T0	TF0	000BH	ET0	PT0/H	3	否
外部中断 1	IE1	0013H	EX1	PX1/H	4	是
T1	TF1	001BH	ET1	PT1/H	5	否
PCA	CF/CCFn	0033H	EC	PPCH	6	否
UART	TI/RI	0023H	ES	PS/H	7	否
T2	TF2, EXF2	002BH	ET2	PT2/H	8	否

## 11.0 省电模式

本器件提供两种省电模式，分别是 IDLE 模式和 POWER-DOWN 模式。请看表 11-1。

### 11.1 空闲 (IDLE) 模式

把 PCON 寄存器的 IDL 位置位将进入 IDLE 模式。在 IDLE 模式，程序计数器 PC 停止，系统时钟继续在跑，所有中断和外围设备保持工作状态，片内 RAM 和特殊功能寄存器保持数据。器件通过系统中断或硬件复位退出 IDLE 模式。如果通过系统中断退出，中断的开始要清除 IDL 位，然后退出 IDLE 模式。执行完中断服务子程序后，被中断的程序重新执行在启动 IDLE 模式的指令后的指令。如果用硬件复位退出，操作和上电复位一样。

### 11.2 掉电 (Power-down) 模式

把 PCON 寄存器的 PD 位置位将进入 POWER-DOWN 模式。在 POWER-DOWN 模式，时钟停止，外部中断可以响应，SRAM 的内容保持，此时最小的 VDD 是 2.0V。

器件通过外部中断或硬件复位可退出 POWER-DOWN 模式。中断开始清除 PD 位，然后退出 POWER-DOWN 模式。保持外部中断脚为低将重新启动晶振，信号要保持 1024 个时钟周期再变回高电平。当中断信号恢复为高时，首先执行中断服务子程序。硬件复位则与上电复位类似。为了准确地退出 POWER-DOWN 模式，复位或外部中断必须要在 VDD 恢复到正常工作电压后才工作。一定要保持 VDD 电压足够长时间以使振荡器重新启动并稳定下来。

表 11-1: 省电模式

模式	初始化	MCU 状态	退出
----	-----	--------	----

<p><b>IDLE</b></p>	<p>软件 (对 PCON 的 IDL 置位) MOV PCON, #01H;</p>	<p>时钟在跑。中断, 串口和定时/计数器在工作。程序计数器停止。ALE 和 PSEN# 在高电平。所有的寄存器不改变。</p>	<p>允许中断或硬件复位。中断的开始要清除 IDL 位和退出 IDLE 模式。ISR RETI 指令后, 程序回到调用 IDLE 模式的位置。用户可以考虑在调用 POWER-DOWN 模式后放 2 到 3 个 NOP 指令, 以减少出问题。硬件复位重起器件的情况和上电复位一样。</p>
<p><b>POWER-DOWN</b></p>	<p>软件 (对 PCON 的 PD 置位) MOV PCON, #02H;</p>	<p>时钟停止。片内 SRAM 和 SFR 数据保持。ALE 和 PSEN#在低电平。可以允许外部中断工作。</p>	<p>外部时钟或硬件复位。中断开始先清除 PD 位和退出 POWER-DOWN 模式, ISR RET 指令后程序恢复到调用模式时的指令。用户可以考虑在调用 POWER-DOWN 模式后放 2 到 3 个 NOP 指令, 以减少出问题。硬件复位重起器件的情况和上电复位一样。</p>

## 12.0 系统时钟和时钟选择

### 12.1 时钟输出选择和晶振电容值的推荐值

图 12-1 是内部反向放大器 (XTAL1, XTAL2) 的输入和输出, 可以用来做片内振荡器。如果输入外部时钟源时, XTAL2 不接, XTAL1 接驱动脚。上电后, 由于放大器的相互作用和反馈电容, 外部振荡器当作一个更高的电容接到 XTAL1。但是, 一旦外部信号达到 VIL 和 VIH 的规格时, 电容值不能超过 15pF。晶体生产厂家、工作电压和其它因素可能会造成各个应用的差异。C1 和 C2 应该根据每个设计做响应的调整。表 12-1 列出了 C1 和 C2 对应不同频率的典型值。更多关于片内振荡器的设计, 请看网站 <http://www.szks.net> 的内容。

表 12-1: 不同晶体类型下 C1 和 C2 的推荐值

晶体	C1=C2
石英	20-30pF
陶瓷	40-50pF

### 12.2 时钟倍频选择

在默认时, 器件每个机器周期跑 12 个时钟 (x 1 模式), 也可以选择每个机器周期跑 6 个时钟的倍频方式。请参考表 12-2。

时钟倍频模式可以通过外部主机或 IAP 模式使能。倍频模式只是把系统时钟和内部 FLASH 加倍, 也就是 EA#=1, 在访问外部存储器和外围电路时要注意。晶体输出 (XTAL2) 也不会倍频。

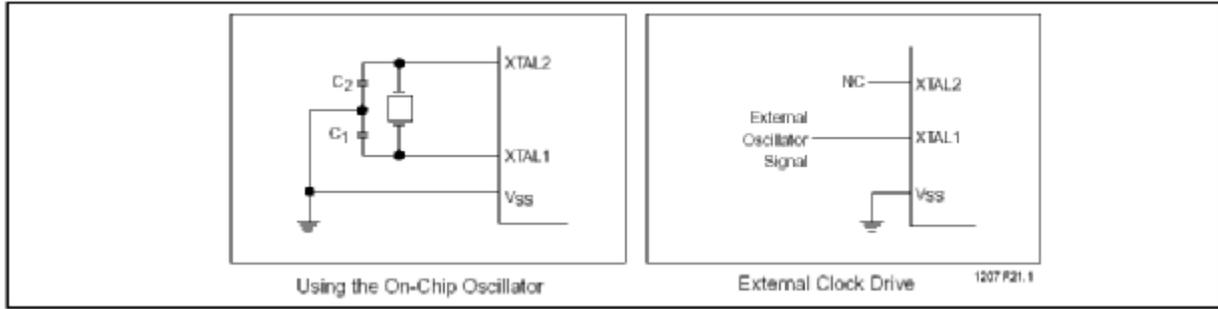
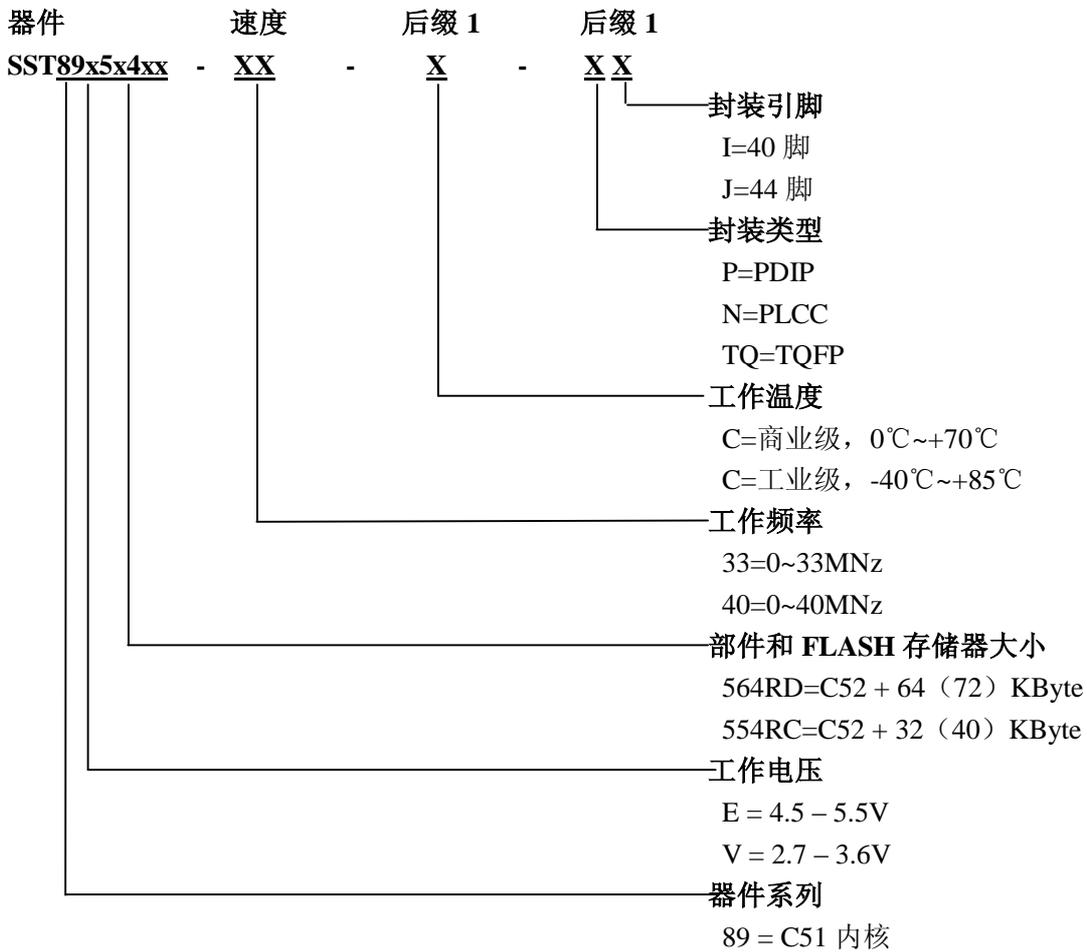


图 12-1: 振荡器特性

表 12-2: 时钟倍频

器 件	标准模式		倍频模式 (x 2)	
	每个机器周期的 时钟	最大外部时钟频 率 (MHz)	每个机器周期的 时钟	最大外部时钟频 率 (MHz)
SST89E564RD/554RC	12	40	6	20
SST89V564RD/554RC	12	16	6	16

### 13.0 产品订购信息



### 14.1 详细型号

SST89E564RD-40-C-PI	SST89E564RD-40-C-NJ	SST89E564RD-40-C-TQJ
SST89E564RD-40-I-PI	SST89E564RD-40-I-NJ	SST89E564RD-40-I-TQJ
SST89V564RD-40-C-PI	SST89V564RD-40-C-NJ	SST89V564RD-40-C-TQJ
SST89V564RD-40-I-PI	SST89V564RD-40-I-NJ	SST89V564RD-40-I-TQJ
SST89E554RD-40-C-PI	SST89E554RD-40-C-NJ	SST89E554RD-40-C-TQJ
SST89E554RD-40-I-PI	SST89E554RD-40-I-NJ	SST89E554RD-40-I-TQJ
SST89V554RD-40-C-PI	SST89V554RD-40-C-NJ	SST89V554RD-40-C-TQJ
SST89V554RD-40-I-PI	SST89V554RD-40-I-NJ	SST89V554RD-40-I-TQJ